EE 183 DA: DESIGN OF ROBOTIC SYSTEMS

Lab report 1

JANUARY 25, 2018

CUONG HOANG
004847900

1 Introduction

A kinematic linkage is a robot that has a set of rigid bodies called links interconnected by a set of joints to a fixed base and a special part that performs specific tasks called the end-effector. In this lab, we are going to find a real-world kinematic linkage that has 4 or more joints. Then we need to analyze the forward and inverse kinematics of that object.

We choose to consider the left human shoulder and elbow as an object to be analyzed. As illustrated in Figure 1, the right human shoulder can be considered as a combination of 3 (1 DOF) joints j1, j2 and j3, and the elbow with 2 joints j4 and j5. For the purpose this assignment, we will separate shoulder into 3 individual joints and rearrange them to another position for simplification. We also assume the hand (end effector) is attached right into the elbow as a rigid object without joint 5 in frame 5. The distance from shoulder to elbow is labeled as L_2 (30 cm) and the distance from elbow to the hand is L_4 (20 cm). The simplified schematic for the arm is shown in Figure 2.

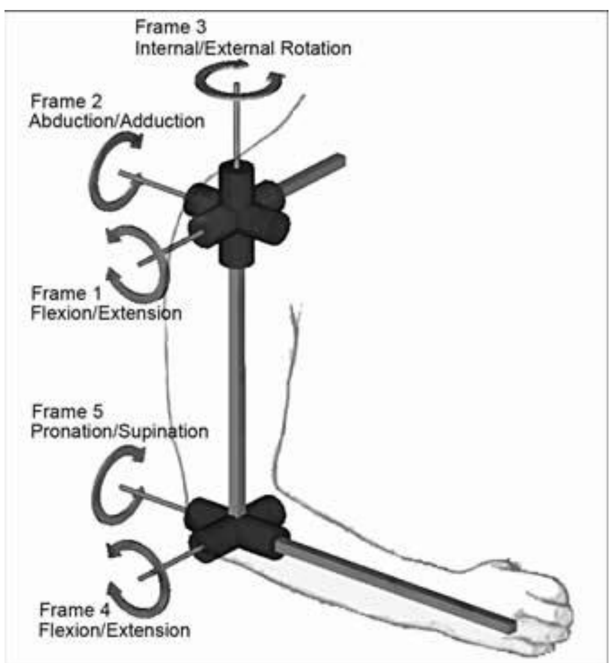

Figure 1.
Kinematics diagram of proposed biomechanical model showing frame designation for joints' modeled degrees of freedom.

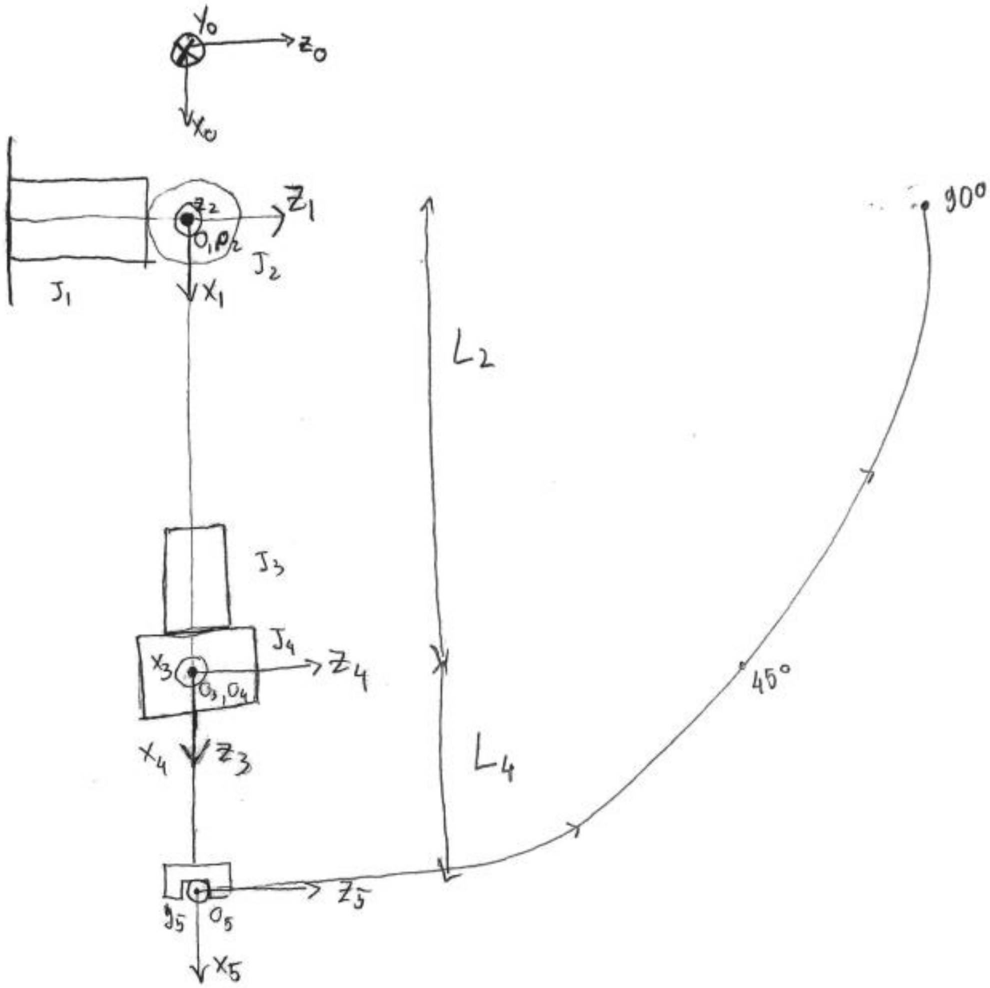

Figure 2: Simplified schematic for the left arm

In the operational space, our robot arm will lift the hand to an angle of 90 degrees compare to initial position and in the same direction with $\mathbf{Z_1}$ as shown in figure 2. This motion requires only the rotation of joint 2 to achieve the purpose.

Normally, the end effector requires 6DOF state variables to describe its state: 3DOF rotational variable $\mathbf{R = \{\theta, \varphi, \psi\}}$ and 3DOF translational variables $\mathbf{r = \{r_x, r_y, r_z\}}$. In our experiment, because the human arm only has rotational joints, the maximum DOF is 3.

2 Methods

Denavit-Hartenberg parameters

To compute the forward kinematics and inverse kinematics, the Denavit-Hartenberg parameters for the linkage of the robotic arm have to be approximated. First, we will set up the geometry configurations of our arm base on the following rules:

1) $\mathbf{Z_i}$ along joint i axis
2) $\mathbf{X_i}$ mutual perpendicular to $\mathbf{Z_i}$ and $\mathbf{Z_{i-1}}$
3) Origin i at intersection of $\mathbf{X_i}$ and $\mathbf{Z_i}$
4) Use right hand rule to find $\mathbf{Y_i}$

Then the 4 D-H parameters are defined as follow:

a_{i-1}: distance $(\mathbf{Z_{i-1}}, \mathbf{Z_i})$ along $\mathbf{X_{i-1}}$

α_{i-1}: angle $(\mathbf{Z_{i-1}}, \mathbf{Z_i})$ about $\mathbf{X_{i-1}}$

d_i: distance $(\mathbf{X_i}, \mathbf{X_{i-1}})$ along $\mathbf{Z_i}$

θ_i: angle $(\mathbf{X_i}, \mathbf{X_{i-1}})$ about $\mathbf{Z_i}$

All the analyses are shown in figure 2 above.

We measure the value of D-H parameters as follow:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	Θ_1
2	$\pi/2$	0	0	Θ_2
3	$\pi/2$	0	30	Θ_3
4	$\pi/2$	0	0	Θ_4
5	0	20	0	0

Forward Kinematic

Forward kinematics is used to compute the position of the end-effector from specified values for the joint parameters.

First, the rotational and translational variables make up the 3 by 4 transformation matrix **T** that we use to describe the state of the end effector in its operational space **b** to be:

$$\boldsymbol{T}_b = [\boldsymbol{R_b}\ \boldsymbol{d_b}] = \begin{bmatrix} x_x & y_x & z_x & d_x \\ x_y & y_y & z_y & d_y \\ x_y & y_z & z_z & d_y \end{bmatrix}$$

In general, the forward kinematics problem is simply to find such $\boldsymbol{T_n^0} = \boldsymbol{T_1^0 T_2^1 .. T_n^{n-1}}$ and then calculate

$$\boldsymbol{p^o} = \boldsymbol{T_n^o p^n}$$

To find the position of our end effector in the world space **o**, we use the following equation:

$$\begin{bmatrix} \boldsymbol{p^o} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R_b^o} & \boldsymbol{d_b^o} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p^b} \\ 1 \end{bmatrix}$$

The last row is added to make the math easier.

Inverse Kinematic

Inverse Kinematic is opposite to forward kinematic. In this case, we will try to figure out the combination of joints' parameters that will move our end effector to a specific position. We use method of gradient descent as discussed in class to solve this problem.

First, we define the end effector position vector **x** in world frame is a function of **q** joint space variables. So at $\boldsymbol{q_o}$:

$$\boldsymbol{x_o} = \boldsymbol{f}(\boldsymbol{q_o})$$

then we compare current position with our goal position

$$\boldsymbol{\delta x} = \boldsymbol{x} - \boldsymbol{x_o}$$

We then use those values and Jacobian matrix to compute a new **q** and repeat those steps above until we get our expected values.

3 Result

As we have mentioned in introduction that we wanted to perform a movement of the arm to an angle of 90 degrees compare to the initial position. This movement only requires the rotation of joint 2. For demonstration purpose, we separate it in 3 stages. The first stage is at rest (initial position, 0 degree), the second stage is the angle of 45 degrees and the last stage is our goal position 90 degrees.

Here are our inputs and outputs for forward kinematic:

```
A = (FK (0, 0, 0, 0))
B = (FK (0, 0.785398, 0, 0))
C = (FK (0, 1.5708, 0, 0))
```

```
[[ 20.]
 [  0.]
 [-30.]]
[[35.3553379 ]
 [ 0.        ]
 [-7.07107359]]
[[29.99992654]
 [ 0.        ]
 [20.0001102 ]]
```

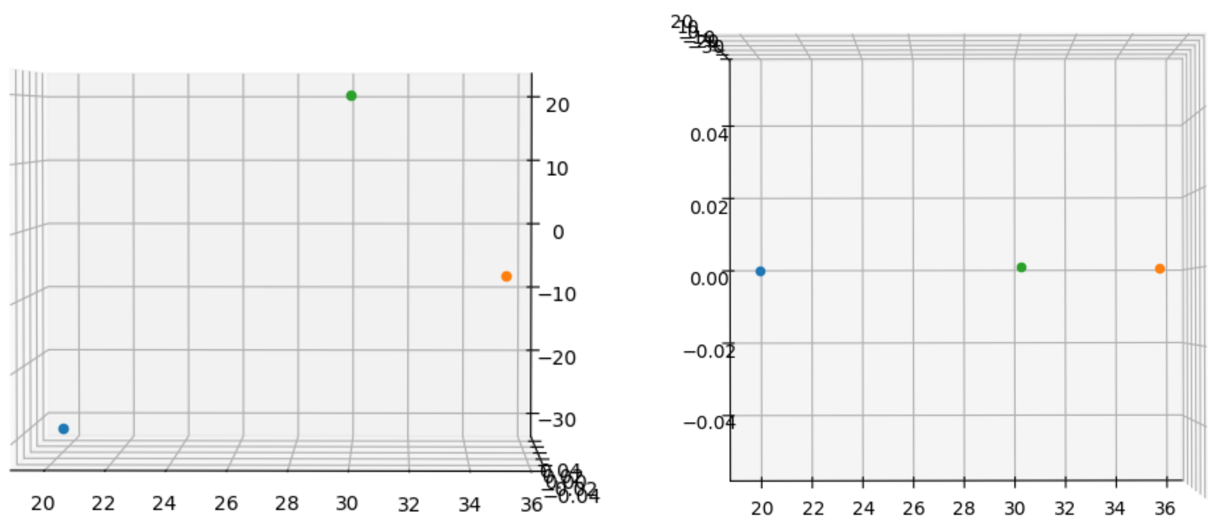

Figure 3: Trajectory of the end-effector in operation space

Blue dot is the initial position of end effector. Orange dot is the position at 45 degrees. Green dot is the position at 90 degrees which is our goal.

As we can see from figure 3, the end effector ends up at position above the initial position as we expected. Because our arm only rotates in the x-z plan respect to the world coordinate, its y value does not change over time. We can confirm this by look at the outputs and they are consistent with our guess.

By inputting our desired position which is obtained from forward kinematic to inverse kinematic, we get back the joint spaces that require to move the end effector to our goal position:

```
A =IK(numpy.matrix([[20],[0],[-30]]))
B =IK(numpy.matrix([[35.3553379],[0],[-7.07107359]]))
C =IK(numpy.matrix([[29.99992654],[0],[20.0001102]]))
```

```
[0. 0. 0. 0.]
[1.37814299e-17 7.85397869e-01 2.44274730e-17 6.28318512e+00]
[6.17520824e-17 1.57080289e+00 9.01145459e-17 6.28287103e+00]
```

The trajectory which is created by FK is not a straight line but an arc in operational space. To create a more accurate trajectory, we need to add more rotational joints or translational joints to our robot. Those joints will add more DOF, so they allow our robot arm moves more freely.

- This lab took me 15 hours to finish. All the sections are challenging because they required in depth understanding to complete.
- For this lab, I worked with Khoi Luc. I wrote the python code for forward kinematic and T matrix. Khoi wrote the code for Jacobian matrix and inverse kinematic. I would say that I have done 45% and Khoi have done 55% of the total work.
- **Github: https://github.com/cuongcan2020/EE-183DA-Lab-1**

References

1. Journal of Rehabilitation Research & Development

   https://www.rehab.research.va.gov/jour/07/44/1/abdullah.html