Lab 1 report

Jiahui Li

October 11, 2016

1 Introduction

Kinematics has two main topics, namely, forward kinematics and inverse kinematics. Forward kinematics refers to use kinematics equations of robots to compute the position of end effector and inverse kinematics is to use the position of end effector to determine the joints variables. These two parts can be widely used in robotics. In order to explore the relationship between FK and IK. We can find a physical kinematic linkage.
In my environment, I can't find a physical kinematic linkage that has 4 joints. But I find some foldable lamp have three joints and if I fix the lamp on the table and suppose I can only rotate it. it will add one more joint. The picture of lamp is shown below.

Figure 1: Physical kinematic linkage.

Here I outline the design of this linkage in terms of its joint space. The picture is shown below. From the picture, we can see all of this four joints are revolute joints. Three of them are parallel and the rest one not.

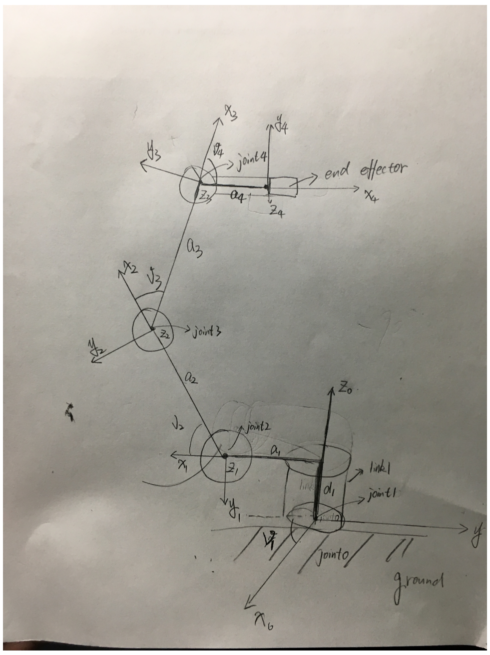

Figure 2: Outline of the linkage

The intended functionality of this linkage is to set the lamplight in a good position when people study, and people can carry it easily because of this foldable structure. As it is shown in the Figure1, the lamp arm is really nice.

In order to analyze the it, we choose a linear motion between two points, and these two points must be touchable by the end effector, which, in other words, the bulb of the lamp.More detail will be discussed in the following part.

2 Methods

In this section, I will compute the forward kinematics and implement inverse kinematics.

2.1 Forward kinematics

Since I find this lamp on the Amazon, I don't have the exact parameter values of this lamp. But we can define the Denavit-Hartenberg parameters as below:

- α : angle between axes z_{i-1} and z_i about axis x_i to be taken positive when rotation is made counter-clockwise
- a : distance between O_i and O_{i^1}
- θ : angle between axes x_{i-1} and x_{i-i} about axis z_{i-1} to be taken positive when rotation is made counter-clockwise.
- d : coordinate of O_{i^1} along z_{i-1}

Based on the figure 2, we can determine the Denavit-Hartenberg parameters for the linkage like below and I give these constant based on my experience, where θ is the variables.

Link	a_i	α_i	d_i	θ_i
1	5cm	-90	1	v_1
2	15cm	0	0	v_2
3	15cm	0	0	v_3
4	10cm	0	0	v_4

Table 1: The parameters of Denavit-Hartenberg

The transform matrix are also can be seen here.

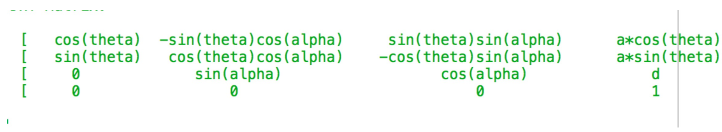

Figure 3: Outline of the linkage

In order to get the total transformation matrix, we can program this process. I use python to create this program. Please see it in code file.

2.2 Inverse kinematics

In this part, I will use inverse kinematics to get the joint variables by given the position of end effector. Inverse kinematics are much harder than forward kinematics, because there is no very accurate method that can exactly compute this problem. But there are several ways to solve these problem. One of the method that can get a very nice result of IK is pesudo-inverse method. The main point of this problem is to calculate the Jacobian matrix and its pesudo-inverse matrix. The figure shows how this method works and more details can be seen in the code file. And the results will be discussed in the results part.

```
while (e is too far from g){
    compute the Jacobian matrix J
    compute the pseudoinverse of the Jacobian matrix— J⁺
    compute change in joint DOFs: Δθ = J⁺ · Δe
    apply the change to DOFs, move a small step of αΔθ: θ = θ + αΔθ
}
```

Figure 4: Flow chart of inverse kinematic.

3 Results

In this section, I will present and analyze the results of your computation/experiments. In order to compare to the first problem, I tried two different points in the operational space. The position and the specified D-H parameters are shown below.

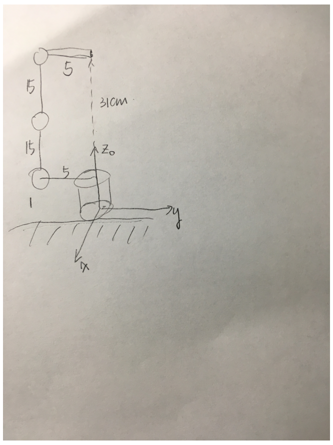

Figure 5: Position 1

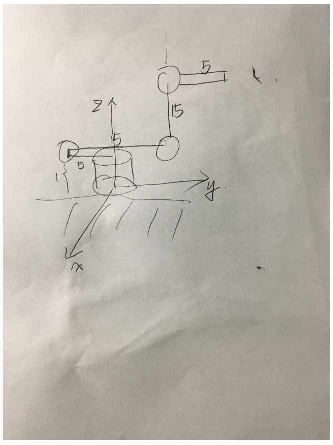

Figure 6: Position 2

Link	a_i	α_i	d_i	θ_i
1	5cm	-90	1	-90
2	15cm	0	0	-90
3	15cm	0	0	0
4	10cm	0	0	-90

Table 2: The parameters of Denavit-Hartenberg in case 1

Link	a_i	α_i	d_i	θ_i
1	5cm	-90	1	-90
2	15cm	0	0	-180
3	15cm	0	0	90
4	10cm	0	0	-90

Table 3: The parameters of Denavit-Hartenberg in case 2

In situation 1, the value of z should be 31cm and x =0, y=0. In situation 2, z should be 16 and y =15, x = 0. The program results are shown below. We can see that the results are pretty the same.

```
[[ -0.  -0.   1.  -0.]
 [  1.  -0.   0.  -0.]
 [  0.   1.   0.  31.]
 [  0.   0.   0.   1.]]
```

Figure 7: Program result of Position 1

```
....
[[ -0.  -0.   1.  -0.]
 [  1.  -0.   0.  15.]
 [  0.   1.   0.  16.]
 [  0.   0.   0.   1.]]
```

Figure 8: Program result of Position 2

And then I would like to plot the trajectory of this linkage. First I define that the range of joint0 is[-180,180], the range of joint1 is [-180,0], the range of joint2 is [0,180], and the joint3 is [-180,0]. Using the FK method in part2, we can get the 3D image below. But it can not represent the trajectory of the linkage very well. We can fix the joint1, in this way, it will become a 2D image, since the linkage only moves in the zoy plane. The 2D image is also shown below.

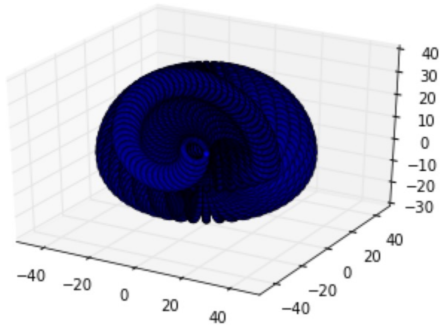

Figure 9: Program result of Position 1

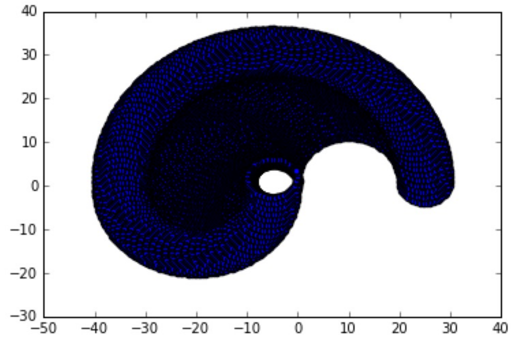

Figure 10: Program result of Position 2

Then I use IK method to compute the configuration parameters of the endpoints of the desired motion. And use FK to generate a trajectory in operational space arising from a linear interpolation between these in configuration space. The trajectory is shown below. As we can see from the picture, It doesn't work as well as I hope. From the figure and other program result, I can find that there is a trend that the point is moving from the origin of the point to the target position. But it moves away after several steps. I don't know why does it happen. I thought it is because there are some unstable points in the operational space that can not be touched by this method. But when I changed my target position. It still doesn't work very well. I'm thinking about where I am wrong. So I am still working on it.

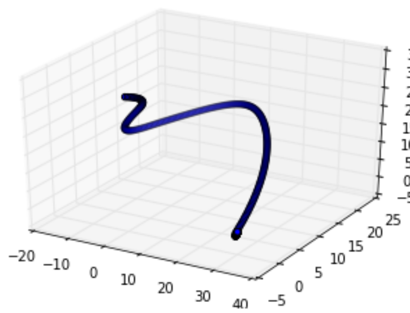

Figure 11: Trajectory of the end effector

Since I get two more days for this lab1, I wrote another code for Ik. This time, I solve this problem. The green one is the supposed linear motion, and the blue one is the trajectory what this IK method did. From the picture, we can see that the trajectory does not go directly to the target point, but it goes to that point finally. Then I checked the coordinates of these points, which are also give below. Since my target is[0,15,16] and my original point is [0,0,31], it make sense that z changed greatly at first. That is why in the figure z drops down quickly and then slowly goes to the target.

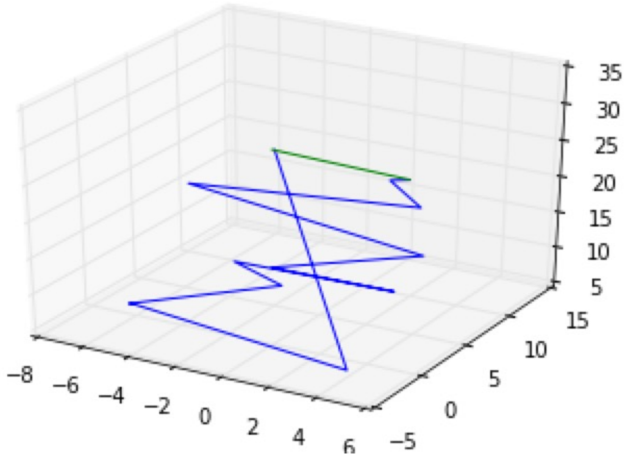

Figure 12: Trajectory of the end effector

ide ▲	Type	Size	
0	float64	1	0.0719999
1	float64	1	4.2080000
2	float64	1	-6.069
3	float64	1	-6.341999
4	float64	1	0.3930000
5	float64	1	-1.278
6	float64	1	5.0880000
7	float64	1	-0.071999
8	float64	1	4.2789999
9	float64	1	-7.229000
10	float64	1	1.9179999
11	float64	1	-0.771000
12	float64	1	0.0109999
13	float64	1	-0.0

Figure 13: Coordinates of x

ide ▲	Type	Size	
0	float64	1	0.070999999
1	float64	1	-2.69700000
2	float64	1	-0.002
3	float64	1	-0.001
4	float64	1	-0.02900000
5	float64	1	-0.87
6	float64	1	-0.04200000
7	float64	1	0.048000000
8	float64	1	5.277999999
9	float64	1	8.452
10	float64	1	11.237
11	float64	1	14.766
12	float64	1	14.97600000
13	float64	1	15.0

Figure 14: Coordinates of y

Index	Type	Size	
0	float64	1	30.9759999
1	float64	1	6.0
2	float64	1	5.88400000
3	float64	1	5.83999999
4	float64	1	12.9280000
5	float64	1	15.852
6	float64	1	15.2639999
7	float64	1	15.08
8	float64	1	15.365
9	float64	1	15.9250000
10	float64	1	15.997
11	float64	1	15.6140000
12	float64	1	15.9990000
13	float64	1	16.0

Figure 15: Coordinates of z

3.1 Conlusion

The goal of this lab is to understand forward and inverse kinematics of simple robotic designs.I have extracted the Denavit-Hartenberg parameters and use them to calculate the transformation between joint (configuration) space parameters and operational space parameters. And I also analyze that transformation to determine the inverse kinematics.Finally,I implement the inverse kinematics and use implementation to simulate a sample mission of the robot.

It takes me around three days to finish this lab. Since I don't know anyone in this class, I haven't find my partner to do this so I finish it by myself. I didn't get the right results at first, but since we have two more days for this lab, luckily I figured it out. Anyway it is an interesting topic and I like it.

References

[1] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani,Giuseppe Oriolo. Robotics modeling, planning and control. (2009)