ECE 470: Forward Kinematics

Yuchuan Zhu
NetID: yuchuan5
Partner's Name: Fengkai Chen
TA: Peixin Chang
Section: Tuesday 4PM
Date Submitted: March 17, 2020

Introduction:

In Lab 4, we will implement the exponential forward kinematics on the UR3 robot arm to find out the transformation matrix with respect to the space frame we choose. We will design a Python program to implement this lab.

Objectives:

The purpose of this lab is to implement the exponential forward kinematics on the UR3 robot to estimate the end-effector pose given a set of joint angles. In this lab we will:

- Solve the exponential forward kinematic equations for the UR3.
- Write a Python function that moves the UR3 to a configuration specified by the user.
- Compare the exponential forward kinematic estimation with the actual robot movement.

Method:

First, we choose the space frame and initial state as the following figure shows:

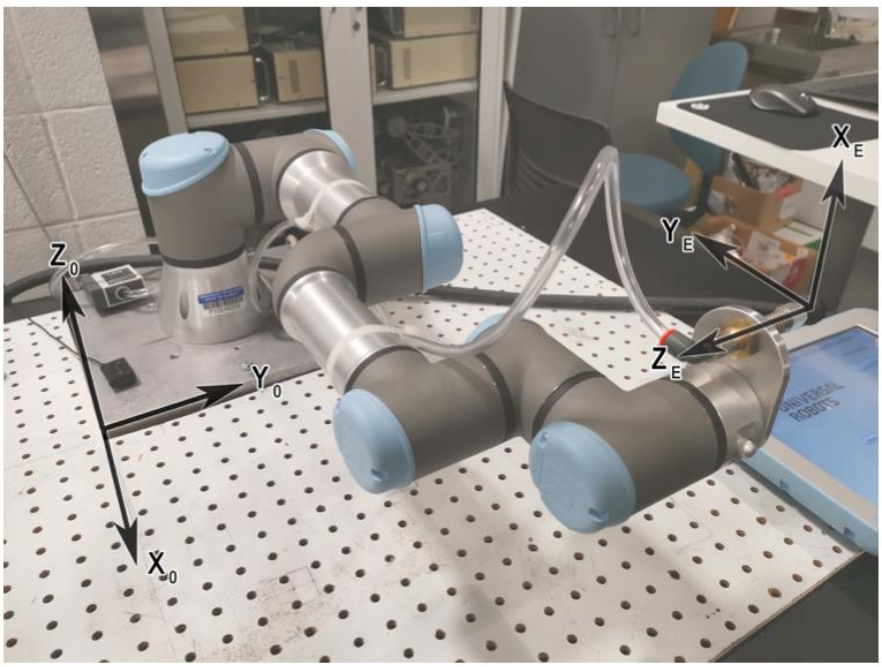
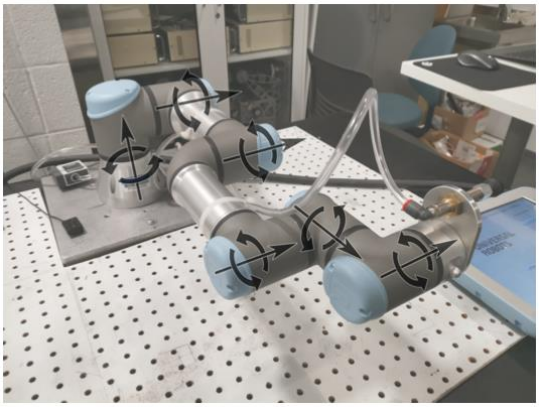
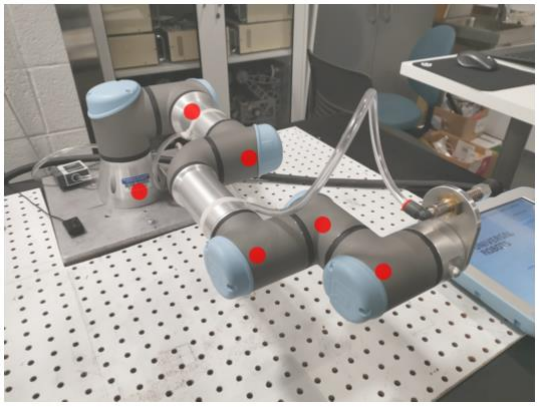


Figure 1. "zero" configuration and frame placement for UR3

And we use the joints in lab manual:





Therefore, we can find out the M:

```
M = [0 -1  0 0.39;
     0  0 -1 0.401;
     1  0  0 0.2155;
     0  0  0 1];
```

Next, we write the codes to calculate $v_1 \sim v_6$ from $\omega_1 \sim \omega_6$ and $q_1 \sim q_6$ we find using equation $v = -\omega \times q$:

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
ω	(0 0 1)	(0 1 0)	(0 1 0)	(0 1 0)	(1 0 0)	(0 1 0)
q	(-0.15 0.15 0.01)	(-0.15 0.27 0.162)	(0.094 0.27 0.162)	(0.307 0.177 0.162)	(0.307 0.26 0.162)	(0.39 0.26 0.162)
v	(0.15 0.15 0)	(-0.162 0 0.15)	(-0.162 0 0.094)	(-0.162 0 0.307)	(0 0.162 -0.26)	(-0.162 0 0.39)

```
S = [0  0  0  0  1  0;
     0  1  1  1  0  1;
     1  0  0  0  0  0;
     0.15  -0.162  -0.162  -0.162  0  -0.162;
     0.15  0  0  0  0.162  0;
     0  0.15  0.094  0.307  -0.26  0.39];
```

Then to find the screw matrix of every joint, we define a function to calculate it:

```
def skew(S):
    return np.array([[0,-S[2],S[1],S[3]],[S[2],0,-S[0],S[4]],[-S[1],S[0],0,S[5]],[0,0,0,0]])
```

Next we use the equation $T = e^{[S_1]\theta_1} \dots e^{[S_6]\theta_6} M$ to calculate the transformation matrix T:

```
theta = np.array([theta1,theta2,theta3,theta4,theta5,theta6])
T = np.eye(4)

M, S = Get_MS()

for i in range(6):
    SS = skew(S[:,i])
    T = np.dot(T,expm(SS*theta[i]))
T = np.dot(T,M)
```

As for the codes of Matlab's symbolic toolbox to generate the final homogeneous transform T, we have upload it on Github whose link is attached at the Appendix part.

$T = e^{[S_1]\theta_1} \dots e^{[S_6]\theta_6} M$

$e^{[S_1]\theta_1} =$

```
[          exp(-theta1*1i)/2 + exp(theta1*1i)/2, - (exp(-theta1*1i)*1i)/2 + (exp(theta1*1i)*1i)/2, 0, - 3/20 + exp(-theta1*1i)*(3/40 + 3i/40) + exp(theta1*1i)*(3/40 - 3i/40)]
[ (exp(-theta1*1i)*1i)/2 - (exp(theta1*1i)*1i)/2,            exp(-theta1*1i)/2 + exp(theta1*1i)/2, 0,   3/20 - exp(theta1*1i)*(3/40 + 3i/40) - exp(-theta1*1i)*(3/40 - 3i/40)]
[                                              0,                                               0, 1,                                                                              0]
[                                              0,                                               0, 0,                                                                              1]
```

$e^{[S_2]\theta_2} =$

```
[           exp(-theta2*1i)/2 + exp(theta2*1i)/2, 0, (exp(-theta2*1i)*1i)/2 - (exp(theta2*1i)*1i)/2,    3/20 - exp(theta2*1i)*(3/40 - 81i/1000) - exp(-theta2*1i)*(3/40 + 81i/1000)]
[                                              0, 1,                                               0,                                                                             0]
[ - (exp(-theta2*1i)*1i)/2 + (exp(theta2*1i)*1i)/2, 0,          exp(-theta2*1i)/2 + exp(theta2*1i)/2,  81/500 - exp(theta2*1i)*(81/1000 + 3i/40) - exp(-theta2*1i)*(81/1000 - 3i/40)]
[                                              0, 0,                                               0,                                                                             1]
```

$e^{[S_3]\theta_3} =$

```
[           exp(-theta3*1i)/2 + exp(theta3*1i)/2, 0, (exp(-theta3*1i)*1i)/2 - (exp(theta3*1i)*1i)/2,  47/500 - exp(theta3*1i)*(47/1000 - 81i/1000) - exp(-theta3*1i)*(47/1000 + 81i/1000)]
[                                              0, 1,                                               0,                                                                             0]
[ - (exp(-theta3*1i)*1i)/2 + (exp(theta3*1i)*1i)/2, 0,          exp(-theta3*1i)/2 + exp(theta3*1i)/2,  81/500 - exp(theta3*1i)*(81/1000 + 47i/1000) - exp(-theta3*1i)*(81/1000 - 47i/1000)]
[                                              0, 0,                                               0,                                                                             1]
```

$e^{[S_4]\theta_4} =$

```
[           exp(-theta4*1i)/2 + exp(theta4*1i)/2, 0, (exp(-theta4*1i)*1i)/2 - (exp(theta4*1i)*1i)/2, 307/1000 - exp(theta4*1i)*(307/2000 - 81i/1000) - exp(-theta4*1i)*(307/2000 + 81i/1000)]
[                                              0, 1,                                               0,                                                                              0]
[ - (exp(-theta4*1i)*1i)/2 + (exp(theta4*1i)*1i)/2, 0,          exp(-theta4*1i)/2 + exp(theta4*1i)/2,   81/500 - exp(theta4*1i)*(81/1000 + 307i/2000) - exp(-theta4*1i)*(81/1000 - 307i/2000)]
[                                              0, 0,                                               0,                                                                              1]
```

$e^{[S_5]\theta_5} =$

```
[ 1,                                              0,                                               0,                                                                              0]
[ 0,          exp(-theta5*1i)/2 + exp(theta5*1i)/2, - (exp(-theta5*1i)*1i)/2 + (exp(theta5*1i)*1i)/2,   13/50 - exp(theta5*1i)*(13/100 + 81i/1000) - exp(-theta5*1i)*(13/100 - 81i/1000)]
[ 0, (exp(-theta5*1i)*1i)/2 - (exp(theta5*1i)*1i)/2,            exp(-theta5*1i)/2 + exp(theta5*1i)/2,  81/500 - exp(theta5*1i)*(81/1000 - 13i/100) - exp(-theta5*1i)*(81/1000 + 13i/100)]
[ 0,                                              0,                                               0,                                                                              1]
```

$e^{[S_6]\theta_6} =$

```
[           exp(-theta6*1i)/2 + exp(theta6*1i)/2, 0, (exp(-theta6*1i)*1i)/2 - (exp(theta6*1i)*1i)/2, 39/100 - exp(theta6*1i)*(39/200 - 81i/1000) - exp(-theta6*1i)*(39/200 + 81i/1000)]
[                                              0, 1,                                               0,                                                                              0]
[ - (exp(-theta6*1i)*1i)/2 + (exp(theta6*1i)*1i)/2, 0,          exp(-theta6*1i)/2 + exp(theta6*1i)/2,  81/500 - exp(theta6*1i)*(81/1000 + 39i/200) - exp(-theta6*1i)*(81/1000 - 39i/200)]
[                                              0, 0,                                               0,                                                                              1]
```

Data and Results:

Results for two test points:

	Joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$	Calculated vector d
Test Point 1	(35, -35, 25, -20, -90, 0)	(-0.1424, 0.0489, -0.0350)
Test Point 2	(10, -25, 35, -45, -90, 10)	(-0.1226, 0.1598, 0.3579)

Conclusion:

The reason why the error occurs we think is that the length of every joint is not totally the same as the length in real world. And the movement of UR3 robot may not be as accurately as we set. But the final error is quite small, so we think the result is good.

In conclusion, this lab is not as complicated as before because most codes have been written in previous homework. The most difficult part of this lab, I think, is to use Matlab's symbolic toolbox to generate the final homogeneous transform because we mainly use Python to finish the lab. But finally, we did it.

References:

Lab manual and Lab Report Guidelines

Appendix:

Code for this lab (including code of Matlab, named *calculateT.m*):
https://github.com/ECE470Yuchuan5/ECE470Lab.git