MAE204 Lab Project 2

Dihan Liu, Kaiyu Liu, Lingyi Wei, Meng Chin Chiang

1.Abstract:

The purpose of the second lab is to try out the forward kinematics functionality using the UR3e robot arm.

2.Problem formulation and procedure:

We wrote a MATLAB script to calculate forward kinematics, then utilized the provided script templates to verify our script by moving the robot to the following joint angles : [-20°, -40°, 60°, 10°, 30°, 10°], [5°, 10°, -30°, 230°, -50°, 150°], [25°, -88°, 59°, 66°, -18°, -153°].

Our target angle ϴ can be represented as:

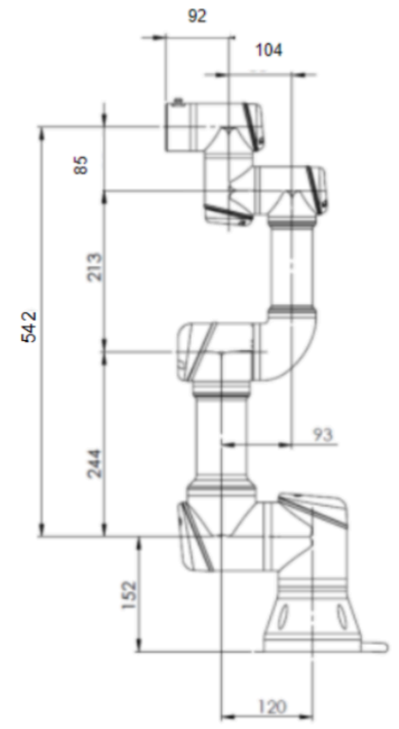
$$\theta = \{\theta_1, \theta_2, ..., \theta_6\}$$

Transformation function T can be represented as:

$$T(\theta) = e^{[S_1]\theta_1} ... e^{[S_6]\theta_6} M$$

3.Experiments and Results:

Given the dimensions of the robot arms



We firstly computed the 6 screw axes and configuration transformation matrix M as below.

S1 = [0, 0, 1, -300, 0, 0]
S2 = [0, 1, 0, -152-88, 0, 0]
S3 = [0, 1, 0, -152-88, 0, 244]
S4 = [0, 1, 0, -152-88, 0, 244+213]
S5 = [0, 0, -1, 300-120+93-104, 244+213, 0]
S6 = [0, 1, 0, -152-88+85, 0, 244+213]

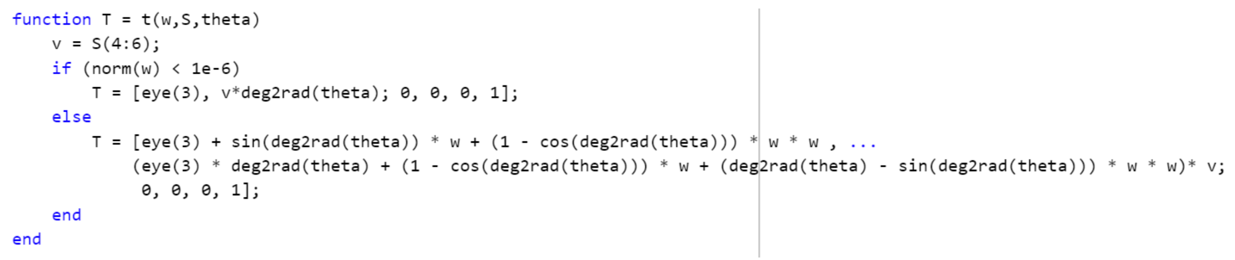
M=[1 0 0 244+213;
0 1 0 -(300-120+93-104-92-78);
0 0 1 152+88-85;
0 0 0 1];

Secondly, we wrote a function t to compute the matrix exponential for each screw axes. According to the formula from the course:

$$e^{[S]\theta} = \begin{bmatrix} e^{[\omega]\theta} & \left(I\theta + (1-\cos\theta)[\omega] + (\theta - \sin\theta)[\omega]^2\right)v \\ 0 & 1 \end{bmatrix}.$$

$$\begin{aligned} R &= e^{[\hat{\omega}_1]\theta_1} \\ &= I + \sin\theta_1[\hat{\omega}_1] + (1-\cos\theta_1)[\hat{\omega}_1]^2 \end{aligned}$$

Thus, function t can be written as,



```
function T = t(w,S,theta)
    v = S(4:6);
    if (norm(w) < 1e-6)
        T = [eye(3), v*deg2rad(theta); 0, 0, 0, 1];
    else
        T = [eye(3) + sin(deg2rad(theta)) * w + (1 - cos(deg2rad(theta))) * w * w , ...
            (eye(3) * deg2rad(theta) + (1 - cos(deg2rad(theta))) * w + (deg2rad(theta) - sin(deg2rad(theta))) * w * w)* v;
             0, 0, 0, 1];
    end
end
```

Thirdly, we complete the content of lab2.m for calculating the end effector transformation matrix T :

S1 = [0, 0, 1, -300, 0, 0]';
S2 = [0, 1, 0, -152-88, 0, 0]';
S3 = [0, 1, 0, -152-88, 0, 244]';
S4 = [0, 1, 0, -152-88, 0, 244+213]';
S5 = [0, 0, -1, 300-120+93-104, 244+213, 0]';

```
S6 = [0, 1, 0, -152-88+85, 0, 244+213]';
M = [1 0 0 244+213;
   0 1 0 -(300-120+93-104-92-78);
   0 0 1 152+88-85;
   0 0 0 1];
T =  t(VecToso3(S1(1:3)),S1, theta_1)*t(VecToso3(S2(1:3)),S2, theta_2)*t(VecToso3(S3(1:3)),S3,
theta_3)*t(VecToso3(S4(1:3)),S4, theta_4)*t(VecToso3(S5(1:3)),S5, theta_5)*t(VecToso3(S6(1:3)),S6,
theta_6)*M;
disp('The transformation matrix T is:')
disp(T)
fprintf('The position of the end-effector is %4.2f, %4.2f, %4.2f (mm) \n',T(1,4),T(2,4),T(3,4))
end
```

After inputting [0 ,0 ,0, 0, 0, 0] into the script, we found that the T matrix equaled the M matrix.Then, we inputted three angles [-20°, -40°, 60°, 10°, 30°, 10°], [5°, 10°, -30°, 230°, -50°, 150°], [25°, -88°, 59°, 66°, -18°, -153°] into the program, the results including the positions of the end effectors are:

```
The transformation matrix T is:
    0.5338    0.7031    0.4698  488.1204
   -0.7264    0.6657   -0.1710 -181.5812
   -0.4330   -0.2500    0.8660  207.8777
         0         0         0    1.0000

The position of the end-effector is 488.12, -181.58, 207.88 (mm)
The transformation matrix T is:
    0.7871    0.6049    0.1207  572.5198
   -0.5971    0.6982    0.3950   -8.7193
    0.1547   -0.3830    0.9107  278.9785
         0         0         0    1.0000

The position of the end-effector is 572.52, -8.72, 278.98 (mm)
The transformation matrix T is:
   -0.2494   -0.6256   -0.7392  -31.5197
   -0.4201    0.7577   -0.4995    8.2383
    0.8725    0.1860   -0.4517  550.8469
         0         0         0    1.0000

The position of the end-effector is -31.52, 8.24, 550.85 (mm)
```

Physical measurements of the end-effector position:

The input angles	x-coordinate	y-coordinate	z-coordinate
[-20°, -40°, 60°, 10°, 30°, 10°]	490	-179	197
[5°, 10°, -30°, 230°, -50°, 150°]	572	0	266
[25°, -88°, 59°, 66°, -18°, -153°]	-45	10	547

All in mm.

The error is calculated using simulated result minus measured result and shown in the following chart.

The input angles	x-result	x-error	y-result	y-error	z-result	z-error
[-20°, -40°, 60°, 10°, 30°, 10°]	488.12	1.88	-181.58	-2.58	207.88	-10.88
[5°, 10°, -30°, 230°, -50°, 150°]	572.52	-0.52	-8.72	8.72	278.98	-12.98
[25°, -88°, 59°, 66°, -18°, -153°]	-31.52	-13.48	8.24	1.76	550.85	-3.85

All in mm.

4.Summary

1. This lab assignment taught the importance of forward kinematics calculating the position and orientation of the end effector from the joint angles. Comparing the measurements and the results from the forward kinematics script, we can obtain the errors between them.

There might be several possible reasons for these errors:

1) Model simplification or assumption errors: The model may have simplified some of the actual complex factors, such as neglecting flexible joints, gear backlash, or mechanical wear.
2) Parameter errors: These differences may originate from manufacturing errors, assembly errors, or wear and tear over long-term usage.

3) The data acquisition from the device is unstable, which is manifested by a jitter in the data display on the operation panel, leading to errors in the collected data. Even at the zero position, the data collected by the device cannot be completely zeroed out.

2. In addition, we need to understand the concept of the matrix exponential method for calculating the T matrix. In the process of coding, we directly used the exponential of the screw axis to calculate the matrix exponential without using the formula given in the book, which led to wrong results. The reason for the mistake is that the matrix exponential is a concept of an operation method, and cannot be directly calculated using exponential operations.

3. When coding in Matlab, it is important to pay attention to the conversion between degrees and radians. MATLAB's default input for angles is in radians, while the angles used in the experiment are in degrees. Using degrees directly will lead to calculation errors. Therefore, it is necessary to use the function deg2red() to convert the angles during input.

4. When constructing the Screw Axis Matrix from the image data of a robot's structure, we should not assume that two components are parallel simply because they appear to be. The specific positional relationship must be calculated through the connections between the structures.