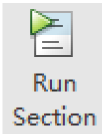
Robotic project1

Student Id: 312512061

Name： 劉郁昇

Interface description:

We use MATLAB R2001b as our work platform.

Open the file “robotic_project1” and press the run section button 

.

Command window will show

```
Q1:Forward kinematics:
please enter the joint variable(in degree)
theta1   (-160~160),  theta2   (-125~125), theta3   (-135~135),
theta4   (-140~140),  theta5   (-100~100), theta6   (-260~260):
```

Users should input an array containing six values, which represent six angles.

E.g. Input: [20 20 20 20 20 20]

Result is shown in figure1.

Then program will show

```
Q2 : Inverse Kinematic:
Please enter Cartesian point:
```

User should input a joint variables matrix.

E.g. Input: [0.1058 -0.6425 0.7589 0.5776;
0.7019 0.5889 0.4007 0.3688;
-0.7044 0.4903 0.5133 0.1968;
0 0 0 1.0000;]

Result is shown in figure3.

With the above method, we can cross-validate forward kinematics and inverse kinematics.

Result:

Task1:

```
Q1:Forward kinematics:
please enter the joint variable(in degree)
theta1  (-160~160),  theta2  (-125~125), theta3  (-135~135),
theta4  (-140~140),  theta5  (-100~100), theta6  (-260~260):
[20 20 20 20 20 20]
[n o a p]:
    0.1058   -0.6425    0.7589    0.5776
    0.7019    0.5889    0.4007    0.3688
   -0.7044    0.4903    0.5133    0.1968
         0         0         0    1.0000

output:
    0.5776    0.3688    0.1968   34.8424   59.1189   27.8338
```

Fig1. result with input [20 20 20 20 20 20]

```
Q1:Forward kinematics:
please enter the joint variable(in degree)
theta1  (-160~160),  theta2  (-125~125), theta3  (-135~135),
theta4  (-140~140),  theta5  (-100~100), theta6  (-260~260):
[20.000 -53.6857 165.1699 171.5666 52.9026 -136.0080]
theta3 is out of range!
theta4 is out of range!
[n o a p]:
    0.1058   -0.6425    0.7589    0.5749
    0.7019    0.5889    0.4007    0.3678
   -0.7044    0.4903    0.5133    0.2081
         0         0         0    1.0000

output:
    0.5749    0.3678    0.2081   34.8424   59.1189   27.8338
```

Fig2. result with input [20 -53.6857 165.1699 171.5666 52.9026 -136.0080]

Task2:

```
Q2 : Inverse Kinematic:
Please enteer Cartesian point:
    [0.1058   -0.6425    0.7589    0.5776;
    0.7019    0.5889    0.4007    0.3688;
   -0.7044    0.4903    0.5133    0.1968;
         0         0         0    1.0000;]
Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
   20.0007   20.0041   19.9897   19.9957   20.0025   20.0074

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
 theta2 is out of range!
 -134.8839 -127.2069   19.9897   19.1820   50.8906 -166.6135

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
 theta3 is out of range!
 theta4 is out of range!
   20.0007  -52.7931  165.2995  171.6782   53.9171 -136.1984

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
theta2 is out of range!
theta3 is out of range!
theta4 is out of range!
 -134.8839 -200.0041  165.2995  146.2406   27.3089   56.4731

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
theta4 is out of range!
   20.0007   20.0041   19.9897 -160.0043  -20.0025 -159.9926

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
theta2 is out of range!
theta4 is out of range!
 -134.8839 -127.2069   19.9897 -160.8180  -50.8906   13.3865

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
theta3 is out of range!
   20.0007  -52.7931  165.2995   -8.3218  -53.9171   43.8016

Corresponding variable (theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8)
theta2 is out of range!
theta3 is out of range!
 -134.8839 -200.0041  165.2995  -33.7594  -27.3089 -123.5269
```

Figure3. q2 result

Program architecture description:

Task1:

User Input:

```
%Q1
fprintf("Q1:Forward kinematics:\n")
% let user enter six angle
fprintf("please enter the joint variable(in degree)\n")
fprintf("theta1   (-160~160),  theta2   (-125~125), theta3   (-135~135),\n")
input_str = input("theta4   (-140~140),  theta5   (-100~100), theta6   (-260~260):\n",'s');
vector_input = str2num(input_str);%let the string type to float type

ranges = [-160, 160; -125, 125; -135, 135; -140, 140; -100, 100; -260, 260];
%check range
for i = 1:6
    if vector_input(i) < ranges(i, 1) || vector_input(i) > ranges(i, 2)
        fprintf("theta%d is out of range!\n", i);
    end
end
%end of check
```

- The code prompts the user to enter six joint angles within specified ranges.
- Input validation is performed to check if the entered angles fall within the defined ranges.

DH Parameters Initialization:

```
d_array = [0 0 0.149 0.433 0 0];
a_array = [0 0.432 -0.02 0 0 0];
alpha = [-1/2 0 1/2 -1/2 1/2 0];%
                                %

% Initialize transformation matrix
T = eye(4);
```

- Parameters such as 'd_array', 'a_array', and 'alpha' are initialized, representing the DH table values for the robot arm.

Transformation Matrix Calculation:

```
%build A1*A2*...A6
for i =1:6
    angle_radian = deg2rad(vector_input(i));% cos and sin function receive randian but not angle
    A_i = [cos(angle_radian), -sin(angle_radian)*cospi(alpha(i)), sin(angle_radian)*sinpi(alpha(i)), a_array(i)*cos(angle_radian);
           sin(angle_radian), cos(angle_radian)*cospi(alpha(i)) , -cos(angle_radian)*sinpi(alpha(i)), a_array(i)*sin(angle_radian);
           0, sinpi(alpha(i)), cospi(alpha(i)) , d_array(i);
           0, 0, 0, 1];
    T = T * A_i;
end
```

- The code computes the transformation matrix 'T' using the DH parameters and user-provided joint angles.

Deriving Euler Angles and Final Output:

```
angle3_radian = atan2(T(2,3),T(1,3));
angle3_degree = rad2deg(angle3_radian);

angle2_radian = atan2(cos(angle3_radian)*T(1,3)+sin(angle3_radian)*T(2,3),T(3,3));
angle2_degree = rad2deg(angle2_radian);

angle1_radian = atan2(-sin(angle3_radian)*T(1,1)+cos(angle3_radian )*T(2,1) ,-sin(angle3_radian)*T(1,2)+cos(angle3_radian)*T(2,2));
angle1_degree = rad2deg(angle1_radian);

output  = [T(1,4) T(2,4) T(3,4) angle1_degree angle2_degree angle3_degree];
```

- Euler angles (angle1_degree, angle2_degree, angle3_degree) are calculated based on elements of the transformation matrix 'T'. These angles represent the orientation of the end-effector.

Displaying Results:

```
%display result
fprintf("[n o a p]:\n")
disp(T)
fprintf("output:\n")
disp(output)
```

- The resulting transformation matrix 'T' and the computed output (position and orientation) are displayed to the user.

Task2

Input:

```
fprintf("Q2 : Inverse Kinematic:\n");
n_o_a_p = input("Please enter Cartesian point:\n");
nx = n_o_a_p(1,1); ny = n_o_a_p(2,1); nz = n_o_a_p(3,1);
ox = n_o_a_p(1,2); oy = n_o_a_p(2,2); oz = n_o_a_p(3,2);
ax = n_o_a_p(1,3); ay = n_o_a_p(2,3); az = n_o_a_p(3,3);
px = n_o_a_p(1,4); py = n_o_a_p(2,4); pz = n_o_a_p(3,4);


T_in = [nx ox ax px;
        ny oy ay py ;
        nz oz az pz;
        0  0  0  1;] ;
```

- The code starts by taking Cartesian coordinates (**n_o_a_p**) as input, representing the desired position and orientation of the end-effector of the robot.

Defining Transformation Matrix:

- It creates a transformation matrix **T_in** based on the input Cartesian coordinates.

Calculating Joint Angles:

- It computes joint angles (**theta1**, **theta2**, **theta3**, **theta4**, **theta5**, **theta6**) using trigonometric functions and geometric relationships. This process involves multiple steps for each joint angle and often results in multiple possible solutions due to the redundancy in the robot's degrees of freedom.

To better understand the concept of inverse kinematics calculations, please refer to the following Mathematical Operations Instructions. We'll list all possibilities for clarity.

```
% theta1 have two values
theta1_1_radian = atan2(py,px)-atan2(d_array(3),sqrt(px^2+py^2-
d_array(3)^2 )) ;
theta1_1_degree = rad2deg(theta1_1_radian);
theta1_2_radian = atan2(py,px)-atan2(d_array(3),-sqrt(px^2+py^2-
d_array(3)^2 )) ;
theta1_2_degree = rad2deg(theta1_2_radian);

% theta3 also have two values
M = (px^2+py^2+pz^2-
(a_array(2)^2+a_array(3)^2+d_array(3)^2+d_array(4)^2))/(2*a_array(2));
theta3_1_radian = atan2(M,sqrt(a_array(3)^2+d_array(4)^2-M^2)) -
atan2(a_array(3),d_array(4));
theta3_1_degree = rad2deg(theta3_1_radian);

theta3_2_radian = atan2(M,-sqrt(a_array(3)^2+d_array(4)^2-M^2)) -
atan2(a_array(3),d_array(4));
theta3_2_degree = rad2deg(theta3_2_radian);
```

theta2 will have four case because theta1 and theta3 both have 2 value.

```
% theta2  have four values (because theta2 = theta23 - theta3 )
% case1 theta1_1 theta3_1
A = cos(theta1_1_radian)*px+sin(theta1_1_radian)*py;
B = pz;
p1 = sqrt(A^2+B^2);
C = a_array(3)+a_array(2)*cos(theta3_1_radian);
theta23_radian = acos(C/p1)-atan2(B,A);
theta23_degree = rad2deg(theta23_radian);
```

```
theta2_1_radian = theta23_radian - theta3_1_radian;
theta2_1_degree = rad2deg(theta2_1_radian);
```

case2 to case4 is similar to case1

When it comes to theta 4, it become total eight cases.

```
%theta 4 have 2 value , thus original_output * num(theta) = 8
%case 1
theta4_1_radian = atan2(T4_6_1(2,3),T4_6_1(1,3)) ;
if theta4_1_radian>0
    theta4_5_radian = theta4_1_radian-pi;
else
    theta4_5_radian = theta4_1_radian+pi;
end
theta4_1_degree = rad2deg(theta4_1_radian);
theta4_5_degree = rad2deg(theta4_5_radian);
output_q2_1(4) = theta4_1_degree;
output_q2_5 = output_q2_1;
output_q2_5(4) = theta4_5_degree;

%theta 5
theta5_1_radian = atan2(T5_6_1(1,3),T5_6_1(2,3)) ;
theta5_1_degree = rad2deg(theta5_1_radian);
output_q2_1(5) = theta5_1_degree;
%theta 6
%case 1
theta6_1_radian = atan2(T4_6_1(3,2),-T4_6_1(3,1));
theta6_1_degree = rad2deg(theta6_1_radian);
output_q2_1(6) = theta6_1_degree;
```

Checking Joint Angle Ranges:

- It then checks whether the calculated joint angles fall within predefined ranges. If they are within the permissible ranges, it displays the corresponding joint angles; otherwise, it shows a message indicating that the joint angles are out of the acceptable range.

Output:

- Finally, it displays the joint angles (**theta1**, **theta2**, **theta3**, **theta4**, **theta5**, **theta6**) for each of the calculated solutions based on the input Cartesian coordinates.

Mathematical operations instructions

Joint	d	a	α	θ
1	0	0	-90°	0
2	0	a_2	0	0
3	d_3	a_3	90°	0
4	d_4	0	-90°	0
5	0	0	90°	0
6	0	0	0	0

$-160^\circ \leq \theta_1 \leq 160^\circ$

$-125^\circ \leq \theta_2 \leq 125^\circ$

$-135^\circ \leq \theta_3 \leq 135^\circ$

$-140^\circ \leq \theta_4 \leq 140^\circ$

$-100^\circ \leq \theta_5 \leq 100^\circ$

$-260^\circ \leq \theta_6 \leq 260^\circ$

where $d_3 = 0.149$

$d_4 = 0.433$

$a_2 = 0.432$

$a_3 = -0.02$

$$A_n = \begin{bmatrix} c\theta_n & -s\theta_n c\alpha_n & s\theta_n s\alpha_n & a_n c\theta_n \\ s\theta_n & c\theta_n c\alpha_n & -c\theta_n s\alpha_n & a_n s\theta_n \\ 0 & s\alpha_n & c\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_n^{-1} = \begin{bmatrix} c\theta_n & s\theta_n & 0 & 0 \\ -s\theta_n c\alpha_n & c\theta_n c\alpha_n & s\alpha_n & 0 \\ s\theta_n s\alpha_n & -c\theta_n s\alpha_n & c\alpha_n & 0 \\ a_n c\theta_n & a_n s\theta_n & d_n & 1 \end{bmatrix}$$

Two part in this project

<i> forward kinematics <ii> Inverse kinematic

<i> forward kinematics

$$A_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 c\alpha_1 & s\theta_1 s\alpha_1 & a_1 c\theta_1 \\ s\theta_1 & c\theta_1 c\alpha_1 & -c\theta_1 s\alpha_1 & a_1 s\theta_1 \\ 0 & s\alpha_1 & c\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_1^{-1} = \begin{bmatrix} c\theta_1 & s\theta_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c\theta_2 & -s\theta_2 c\alpha_2 & s\theta_2 s\alpha_2 & a_2 c\theta_2 \\ s\theta_2 & c\theta_2 c\alpha_2 & -c\theta_2 s\alpha_2 & a_2 s\theta_2 \\ 0 & s\alpha_2 & c\alpha_2 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0.432c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & 0.432s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2^{-1} = \begin{bmatrix} c\theta_2 & s\theta_2 & 0 & 0 \\ -s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a_2 c\theta_2 & a_2 s\theta_2 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_2 & s\theta_2 & 0 & 0 \\ -s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.432c\theta_2 & 0.432s\theta_2 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} c\theta_3 & -s\theta_3 c\alpha_3 & s\theta_3 s\alpha_3 & a_3 c\theta_3 \\ s\theta_3 & c\theta_3 c\alpha_3 & -c\theta_3 s\alpha_3 & a_3 s\theta_3 \\ 0 & s\alpha_3 & c\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_3 & 0 & s\theta_3 & -0.02c\theta_3 \\ s\theta_3 & 0 & -c\theta_3 & -0.02s\theta_3 \\ 0 & 1 & 0 & 0.149 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3^{-1} = \begin{bmatrix} c\theta_3 & s\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s\theta_3 & -c\theta_3 & 0 & 0 \\ a_3 c\theta_3 & a_3 s\theta_3 & d_3 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_3 & s\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s\theta_3 & -c\theta_3 & 0 & 0 \\ -0.02c\theta_3 & -0.02s\theta_3 & 0.149 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c\theta_4 & -s\theta_4 c\alpha_4 & s\theta_4 s\alpha_4 & a_4 c\theta_4 \\ s\theta_4 & c\theta_4 c\alpha_4 & -c\theta_4 s\alpha_4 & a_4 s\theta_4 \\ 0 & s\alpha_4 & c\alpha_4 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & 0.433 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_4^{-1} = \begin{bmatrix} c\theta_4 & s\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s\theta_3 & c\theta_4 & 0 & 0 \\ 0 & 0 & d_4 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_4 & s\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s\theta_3 & c\theta_4 & 0 & 0 \\ 0 & 0 & 0.433 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c\theta_5 & -s\theta_5 c\alpha_5 & s\theta_5 s\alpha_5 & a_5 c\theta_5 \\ s\theta_5 & c\theta_5 c\alpha_5 & -c\theta_5 s\alpha_5 & a_5 s\theta_5 \\ 0 & s\alpha_5 & c\alpha_5 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5^{-1} = \begin{bmatrix} c\theta_5 & s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 c\alpha_6 & s\theta_6 s\alpha_6 & a_6 c\theta_6 \\ s\theta_6 & c\theta_6 c\alpha_6 & -c\theta_6 s\alpha_6 & a_6 s\theta_6 \\ 0 & s\alpha_6 & c\alpha_6 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_6^{-1} = \begin{bmatrix} c\theta_6 & s\theta_6 & 0 & 0 \\ -s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}^{5}_{6}T = A_5A_6 = \begin{bmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_5c_6 & -c_5s_6 & s_5 & 0 \\ s_5c_6 & -s_5s_6 & -c_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{4}_{6}T = A_4A_5A_6 = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_5c_6 & -c_5s_6 & s_5 & 0 \\ s_5c_6 & -s_5s_6 & -c_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4c_5c_6-s_4s_6 & -c_4c_5s_6-s_4c_6 & c_4s_5 & 0 \\ s_4c_5c_6+c_4s_6 & -s_4c_5s_6+c_4c_6 & s_4s_5 & 0 \\ -s_5c_6 & s_5s_6 & c_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}_{6}T = A_3A_4A_5A_6 = \begin{bmatrix} c\theta_3 & 0 & s\theta_3 & -a_3c\theta_3 \\ s\theta_3 & 0 & -c\theta_3 & -a_3s\theta_3 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_4c_5c_6-s_4s_6 & -c_4c_5s_6-s_4c_6 & c_4s_5 & 0 \\ s_4c_5c_6+c_4s_6 & -s_4c_5s_6+c_4c_6 & s_4s_5 & 0 \\ -s_5c_6 & s_5s_6 & c_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ = \begin{bmatrix} c_3(c_4c_5c_6-s_4s_6)-s_3s_5c_6 & -c_3(c_4c_5s_6+s_4c_6)+s_3s_5s_6 & c_3c_4s_5+s_3c_5 & s_3d_4-a_3c_3 \\ s_3(c_4c_5c_6-s_4s_6)+c_3s_5c_6 & -s_3(c_4c_5s_6+s_4c_6)-c_3s_5s_6 & s_3c_4s_5-c_3c_5 & -d_4c_3-a_3s_3 \\ s_4c_5c_6+c_4s_6 & -s_4c_5s_6+c_4c_6 & s_4s_5 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{6}T = A_2A_3A_4A_5A_6 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3(c_4c_5c_6-s_4s_6)-s_3s_5c_6 & -c_3(c_4c_5s_6+s_4c_6)+s_3s_5s_6 & c_3c_4s_5+s_3c_5 & s_3d_4-a_3c_3 \\ s_3(c_4c_5c_6-s_4s_6)+c_3s_5c_6 & -s_3(c_4c_5s_6+s_4c_6)-c_3s_5s_6 & s_3c_4s_5-c_3c_5 & -d_4c_3-a_3s_3 \\ s_4c_5c_6+c_4s_6 & -s_4c_5s_6+c_4c_6 & s_4s_5 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ = \left[\begin{matrix} c_2\left[c_3(c_4c_5c_6-s_4s_6)-s_3s_5c_6\right]-s_2\left[s_3(c_4c_5c_6-s_4s_6)+c_3s_5c_6\right] & c_2\left[-c_3(c_4c_5s_6+s_4c_6)+s_3s_5s_6\right]+s_2\left[s_3(c_4c_5s_6+s_4c_6)+c_3s_5s_6\right] \\ s_2\left[s_3(c_4c_5c_6-s_4s_6)+c_3s_5c_6\right]+c_2\left[s_3(c_4c_5c_6-s_4s_6)+c_3s_5c_6\right] & s_2\left[-c_3(c_4c_5s_6+s_4c_6)+s_3s_5s_6\right]+c_2\left[-s_3(c_4c_5s_6+s_4c_6)-c_3s_5s_6\right] \\ s_4c_5c_6+c_4s_6 & -s_4c_5s_6+c_4c_6 \\ 0 & 0 \end{matrix} \right. $$

$$ \left. \begin{matrix} c_2\left[c_3c_4s_5+s_3c_5\right]-s_2\left[s_3c_4s_5-c_3c_5\right] & c_2\left[s_3d_4-a_3c_3\right]+s_2\left[d_4c_3+a_3s_3\right]+a_2c_2 \\ s_2\left[c_3c_4s_5+s_3c_5\right]+c_2\left[s_3c_4s_5-c_3c_5\right] & s_2\left[s_3d_4-a_3c_3\right]-c_2\left[d_4c_3+a_3s_3\right]+a_2s_2 \\ s_4s_5 & d_3 \\ 0 & 1 \end{matrix} \right] $$

$$
{}^{1}_{6}T = A_1 \cdot {}^{2}_{6}T = \begin{bmatrix} C\theta_1 & 0 & -S\theta_1 & 0 \\ S\theta_1 & 0 & C\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^{2}_{6}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$
where.

$$n_x = C_1\left\{C_2\left[C_3(C_4C_5C_6 - S_4S_6) - S_3S_5C_6\right] - S_2\left[S_3(C_4C_5C_6 - S_4S_6) + C_3S_5C_6\right]\right\} - S_1\left[S_4C_5C_6 + C_4S_6\right]$$

$$n_y = S_1\left\{C_2\left[C_3(C_4C_5C_6 - S_4S_6) - S_3S_5C_6\right] - S_2\left[S_3(C_4C_5C_6 - S_4S_6) + C_3S_5C_6\right]\right\} + C_1\left[S_4C_5C_6 + C_4S_6\right]$$

$$n_z = -\left\{S_2\left[S_3(C_4C_5C_6 - S_4S_6) + C_3S_5C_6\right] + C_2\left[S_3(C_4C_5C_6 - S_4S_6) + C_3S_5C_6\right]\right\}$$

$$o_x = C_1\left\{C_2\left[-C_3(C_4C_5S_6 + S_4C_6) + S_3S_5S_6\right] + S_2\left[S_3(C_4C_5S_6 + S_4C_6) + C_3S_5S_6\right]\right\} - S_1\left[-S_4C_5S_6 + C_4C_6\right]$$

$$o_y = S_1\left\{C_2\left[-C_3(C_4C_5S_6 + S_4C_6) + S_3S_5S_6\right] + S_2\left[S_3(C_4C_5S_6 + S_4C_6) + C_3S_5S_6\right]\right\} + C_1\left[-S_4C_5S_6 + C_4C_6\right]$$

$$o_z = -\left\{S_2\left[-C_3(C_4C_5S_6 + S_4C_6) + S_3S_5S_6\right] + C_2\left[-S_3(C_4C_5S_6 + S_4C_6) - C_3S_5S_6\right]\right\}$$

$$a_x = C_1\left\{C_2\left[C_3C_4S_5 + S_3C_5\right] - S_2\left[S_3C_4S_5 - C_3C_5\right]\right\} - S_1(S_4S_5)$$

$$a_y = S_1\left\{C_2\left[C_3C_4S_5 + S_3C_5\right] - S_2\left[S_3C_4S_5 - C_3C_5\right]\right\} + C_1(S_4S_5)$$

$$a_z = -\left\{S_2\left[C_3C_4S_5 + S_3C_5\right] + C_2\left[S_3C_4S_5 - C_3C_5\right]\right\}$$

$$
\begin{aligned}
p_x &= C_1\left\{C_2\left[S_3d_4 - a_3C_3\right] + S_2\left[d_4C_3 + a_3S_3\right] + a_2C_2\right\} - S_1d_3 \\
&= C_1\left\{C_2S_3\underline{\underline{d_4}} - C_2\underline{a_3}C_3 + S_2C_3\underline{\underline{d_4}} + S_2\underline{\underline{a_3}}S_3 + a_2C_2\right\} - S_1d_3 \\
&= C_1\left\{d_4(C_2S_3 + S_2C_3) + a_3(S_2S_3 - C_2C_3) + a_2C_2\right\} - S_1d_3
\end{aligned}
$$

$$
\begin{aligned}
p_y &= S_1\left\{C_2\left[S_3d_4 - a_3C_3\right] + S_2\left[d_4C_3 + a_3S_3\right] + a_2C_2\right\} + C_1d_3 \\
&= S_1\left\{C_2S_3\underline{\underline{d_4}} - C_2\underline{\underline{a_3}}C_3 + S_2C_3\underline{\underline{d_4}} + S_2\underline{\underline{a_3}}S_3 + a_2C_2\right\} + C_1d_3 \\
&= S_1\left\{d_4(C_2S_3 + S_2C_3) + a_3(-C_2C_3 + S_2S_3) + a_2C_2\right\} + C_1d_3
\end{aligned}
$$

$$
\begin{aligned}
p_z &= -\left\{S_2\left[S_3d_4 - a_3C_3\right] - C_2\left[d_4C_3 + a_3S_3\right] + a_2S_2\right\} \\
&= -\left\{S_2S_3\underline{\underline{d_4}} - S_2\underline{\underline{a_3}}C_3 - C_2C_3\underline{\underline{d_4}} - C_2\underline{\underline{a_3}}S_3 + a_2S_2\right\} \\
&= -\left\{d_4(S_2S_3 - C_2C_3) + a_3(-S_2C_3 - C_2S_3) + a_2S_2\right\}
\end{aligned}
$$

⟨iii⟩ Inverse kinematic

θ_1:

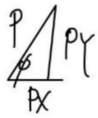
$$T_6 = \begin{bmatrix} f_{11} & f_{12} & f_{13} & p_x \\ f_{21} & f_{22} & f_{23} & p_y \\ f_{31} & f_{32} & f_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^{-1} T_6 = {}^1T_6 = A_2 A_3 A_4 A_5 A_6$$

$$\begin{bmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} & p_x \\ f_{21} & f_{22} & f_{23} & p_y \\ f_{31} & f_{32} & f_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^1T_6 = \begin{bmatrix} & & & s_{23}d_4 + c_{23}a_3 + a_2c_2 \\ & & & -c_{23}d_4 + s_{23}a_3 + a_2s_2 \\ & & & d_3 \\ & & & 1 \end{bmatrix}$$

$\Rightarrow$ $-s_1p_x + c_1p_y = d_3$

let $p_x = p\cos\phi$ $\quad p_y = p\sin\phi$



$p = \sqrt{p_x^2 + p_y^2}$ $\quad \phi = \mathrm{Atan2}(p_y, p_x)$

$c_1 s\phi - s_1 c\phi = \frac{d_3}{p}$

$\sin(\phi - \theta_1) = \frac{d_3}{p}$ $\quad \therefore \cos(\phi - \theta_1) = \pm\sqrt{1 - \frac{d_3^2}{p^2}}$

$\therefore \phi - \theta_1 = \mathrm{Atan2}\left[\frac{d_3}{p}, \pm\sqrt{1 - \frac{d_3^2}{p^2}}\right]$

$\underline{\theta_1 = \mathrm{Atan2}(p_y, p_x) - \mathrm{Atan2}\left(d_3, \pm\sqrt{p_x^2 + p_y^2 - d_3^2}\right)}$ # where $d_3 = a149$

θ_3

$c_1p_x + s_1p_y = s_{23}d_4 + c_{23}a_3 + a_2c_2$ (1)

$-p_z = -c_{23}d_4 + s_{23}a_3 + a_2s_2$ (2)

$-s_1p_x + c_1p_y = d_3$ (3)

$①^2 + ②^2 + ③^2$

$\Rightarrow$

$$a_3 C_3 + d_4 S_3 = \frac{P_x^2 + P_y^2 + P_z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} = M$$

$$\therefore \theta_3 = \mathrm{Atan2}\left(M, \pm\sqrt{a_3^2 + d_4^2 - M^2}\right) - \mathrm{Atan2}(a_3, d_4)$$

$$= \mathrm{Atan2}\left(M, \pm\sqrt{(-0.02)^2 + (0.433)^2 - M^2}\right) - \mathrm{Atan2}(-0.02, 0.433)$$ #

θ_2: $T_3^{-1} T_6 = {}^3T_6 = A_4 A_5 A_6$

$$\begin{bmatrix} C_1C_{23} & S_1C_{23} & -S_{23} & -a_3 - a_2C_3 \\ -S_1 & C_1 & 0 & -d_3 \\ C_1S_{23} & S_1S_{23} & C_{23} & -a_2S_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} — & — & — & 0 \\ — & — & — & 0 \\ — & — & — & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C_1C_{23}P_x + S_1C_{23}P_y - S_{23}P_z = a_3 + a_2C_3$$

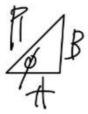
$$\Rightarrow \underbrace{(C_1P_x + S_1P_y)}_{A} C_{23} - S_{23}\underbrace{P_z}_{B} = \underbrace{a_3 + a_2C_3}_{C}$$

let $A = C_1P_x + S_1P_y$

$B = P_z \quad C = a_3 + a_2C_3$

$$AC_{23} - BS_{23} = C$$

let $A = P_1\cos\phi \quad B = P_1\sin\phi$



$P_1 = \sqrt{A^2 + B^2}$

$$P_1\cos\phi\, C_{23} - P_1\sin\phi\, S_{23} = C$$

$$\Rightarrow P_1\cos(\phi + \theta_{23}) = C$$

$$\phi + \theta_{23} = \cos^{-1}\left(\frac{C}{P_1}\right) \qquad \theta_{23} = \cos^{-1}\left(\frac{C}{P_1}\right) - \phi$$

$$\Rightarrow \theta_{23} = \cos^{-1}\left(\frac{C}{P_1}\right) - \mathrm{atan2}(B, A)$$

$$= \arccos\left(\frac{a_3 + a_2C_3}{\sqrt{(C_1P_x + S_1P_y)^2 + (-P_z)^2}}\right) - \mathrm{atan2}(P_z, C_1P_x + S_1P_y)$$

$$C_1S_{23}P_x + S_1S_{23}P_y + C_{23}P_z = d_4 + a_2S_3$$

$$\Rightarrow \theta_2 = \theta_{23} - \theta_3$$

#

$$^4T_6=\begin{bmatrix} C_4C_5C_6-S_4S_6 & -C_4C_5S_6-S_4C_6 & C_4S_5 & 0 \\ S_4C_5C_6+C_4S_6 & -S_4C_5S_6+C_4C_6 & S_4S_5 & 0 \\ -S_5C_6 & S_5S_6 & C_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad *$$

(we have mentione d above)

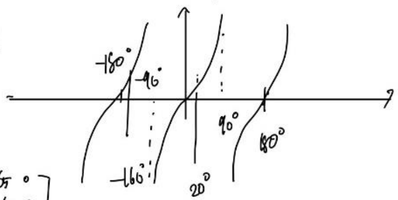
From C_4S_5, S_4S_5 in $*$

$$\tan\theta_4=\frac{S_4S_5}{C_4C_5}=\frac{S_4}{C_4} \quad \text{if only } S_5\neq 0$$

$$\theta_4=\operatorname{atan2}\left(A_3^{-1}A_2^{-1}A_1^{-1}T_{in}(2,3),\ A_3^{-1}A_2^{-1}A_1^{-1}T_{in}(1,3)\right)$$

#

θ_4 will have two solution
it also has relation with θ_6



$$^5T_6=A_5A_6=\begin{bmatrix} C_5C_6 & -C_5S_6 & S_5 & 0 \\ S_5C_6 & -S_5S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\tan\theta_5=\frac{S_5}{-C_5}$$

$$\theta_5=\operatorname{atan2}\left(A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1}T_{in}(1,3),\ A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1}T_{in}(2,3)\right)$$

#

$$\tan\theta_6=\frac{S_5S_6}{-S_5C_6} \quad \text{if } S_5\neq 0$$

$$\Rightarrow \theta_6=\operatorname{atan2}\left({}^5T_6(3,1),\ {}^5T_6(3,2)\right)$$

#

若 $\theta_5=0$ we have $\begin{cases} -C_4S_6-S_4C_6=-S_{46} \\ -S_4S_6+C_4C_6=C_{46} \end{cases}$

$$\Rightarrow -\tan 46=\frac{{}^4T_6(1,2)}{{}^4T_6(2,2)} \quad \theta_{46}=\operatorname{atan2}\left(-{}^4T_6(1,2),\ {}^4T_6(2,2)\right)$$

$$\Rightarrow \theta_6=\theta_{46}-\theta_4$$

#

Discuss the advantages and disadvantages of two types of inverse kinematics (algebraic method, geometric method)

Algebraic Method:

Advantages:

1. **Precision and Accuracy:** Algebraic methods often provide precise and accurate solutions to inverse kinematics problems, especially in complex robotic systems with multiple degrees of freedom.
2. **Suitability for Analytical Solutions:** They are suitable for systems with simple geometries or when an analytical solution is feasible, making it easier to derive formulas.
3. **Mathematical Rigor:** These methods rely on mathematical equations and matrices, offering a rigorous and well-defined approach to solving kinematic problems.

Disadvantages:

1. **Complexity with Nonlinear Systems:** Algebraic methods might struggle with nonlinear systems or mechanisms with complex structures, leading to difficulties in finding closed-form solutions.
2. **Computational Intensity:** Solving algebraic equations for systems with high degrees of freedom can be computationally intensive, requiring substantial processing power and time.
3. **Singularity and Multiple Solutions:** They may encounter singular configurations or instances where multiple solutions exist, complicating the interpretation of results and requiring careful handling.

Geometric Method:

Advantages:

1. **Intuitive Visualization:** Geometric methods often offer a more intuitive approach, using geometric constructions and graphical representations to solve inverse kinematics problems.
2. **Adaptability to Complex Structures:** They can be more adaptable to complex structures and non-standard robotic mechanisms, as they rely on geometric relationships.
3. **Efficiency for Certain Configurations:** For certain configurations and simpler systems, geometric methods can be quicker to implement and more straightforward to visualize.

Disadvantages:

1. **Limited Applicability:** Geometric methods might struggle with highly complex systems or those with numerous joints and constraints, where finding a precise geometric solution becomes challenging.
2. **Sensitivity to Perturbations:** Small errors or inaccuracies in measurements or calculations can significantly impact the accuracy of solutions obtained through geometric methods.
3. **Difficulty in Iterative Solutions:** When iterative methods are necessary, geometric approaches might lack the efficiency of numerical techniques, leading to longer computational times.

In practice, the choice between these methods often depends on the specific characteristics of the robotic system, the required accuracy, computational resources available, and the complexity of the task at hand. Many times, a hybrid approach combining both methods or utilizing numerical techniques might offer the best compromise between accuracy and computational efficiency.