Robotics—Lab1

1. Direct Kinematics

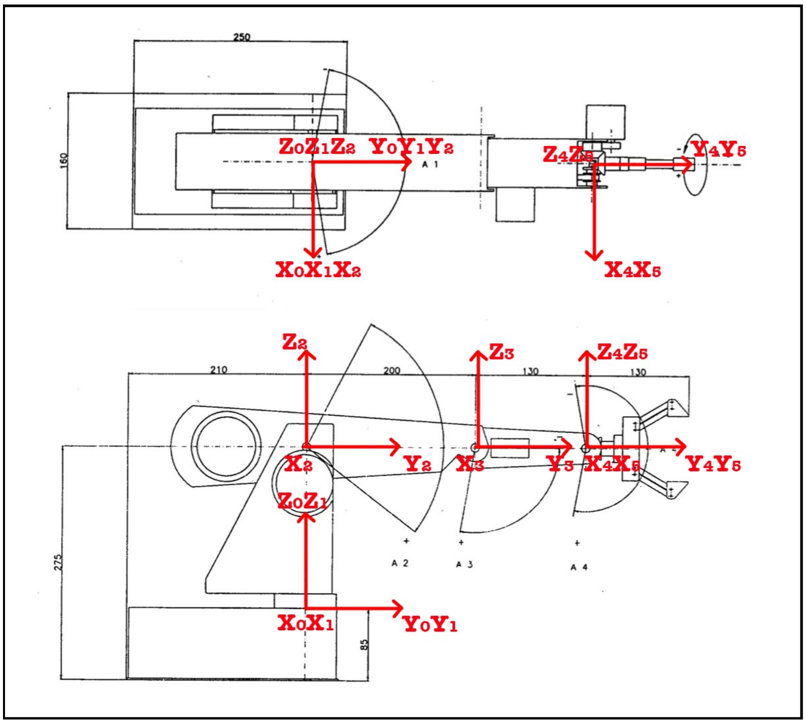
1.1 Outline

Direct Kinematics is an approach from which we can get the final position and orientation of any part of ROB3/TR5 robot. With the given 5 angles and the graph of the robot, rotation matrix is composed to describe the movements of the robot. Finally, the position described in base frame and orientation of every part of robot can be figured out.

1.2 Establishment of the coordinate system

To describe the position and the orientation of all parts of the robot, serials of coordinate system should be established as described in figure below. As seen in the picture, the base frame is established at the bottom of the robot.

And then, the other coordinate systems can also be established to describe the other part of the robot.



1.3 Composition of rotation matrices

Rotation matrices are needed to be composed to describing the orientation with respect to the coordinate system. Due to the rotation of any arm is around a fixed axis of the corresponding coordinate system, the rotation matrices can be compose in the following way.

The rotation from X_0-Y_0-Z_0 to X_1-Y_1-Z_1 around Z_0 axis:

$$ {}^{0}_{1}R = \begin{pmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} $$

The rotation from X_1-Y_1-Z_1 to X_2-Y_2-Z_2 around X_2 axis:

$$ {}_2^1R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{pmatrix} $$

The rotation from X_2-Y_2-Z_2 to X_3-Y_3-Z_3 around X_3 axis:

$$ {}_3^2R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_2 & -\sin\theta_2 \\ 0 & \sin\theta_2 & \cos\theta_2 \end{pmatrix} $$

The rotation from X_3-Y_3-Z_3 to X_4-Y_4-Z_4 around X_4 axis:

$$ {}_4^3R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & \sin\theta_3 & \cos\theta_3 \end{pmatrix} $$

The rotation from X_4-Y_4-Z_4 to X_5-Y_5-Z_5 around Y_5 axis:

$$ {}_5^4R = \begin{pmatrix} \cos\theta_4 & 0 & \sin\theta_4 \\ 0 & 1 & 0 \\ -\sin\theta_4 & 0 & \cos\theta_4 \end{pmatrix} $$

1.4 Homogeneous transformations

In order to achieve a compact representation of the relationship between the coordinates of the same point in two or more different frames, homogeneous transformations is needed to achieve the goal to present the position and orientation in the base frame.

Firstly, the orientation matrix can be easily generated in the way below:

$$ {}_5^0R = {}_1^0R \times {}_2^1R \times {}_3^2R \times {}_4^3R \times {}_5^4R $$

Then the position of the target point can be transformed by the following equation:

$$ {}^{n-1}P = {}_n^{n-1}R\, {}^{n-1}P + {}^{n-1}P_{org} \qquad n \in \{1,2,3,4,5\} $$

Knowing the coordinate of each joint in the corresponding frame, the joint's coordinate can be easily calculated by iterative using the equation above until n=1. So the position of each joint can be determined.

1.5 Euler angles

Euler angles ZYZ can be applied to describe the known orientation in our program. First, the orientation matrix should be in the form of:

$$ {}_5^0R = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} $$

Then, the three Euler Angles ZYZ are defined in this equation:

$$ R(\varphi) \ = Rz(\varphi)Ry'(\theta)Rz''(\psi) = \begin{pmatrix} c_\varphi c_\theta c_\psi - s_\varphi s_\psi & -c_\varphi c_\theta s_\psi - s_\varphi c_\psi & c_\varphi s_\theta \\ s_\varphi c_\theta c_\psi + c_\varphi s_\psi & -s_\varphi c_\theta s_\psi + c_\varphi c_\psi & s_\varphi c_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{pmatrix} $$

Compare the expression below, we can solve the direct kinematics problem with Euler angles ZYZ:

$$ {}_5^0R \equiv R(\varphi) $$

2. Inverse Kinematics

2.1 Outline

Inverse Kinematics allows us to compute the rotation angle of each joint along its certain axis under the circumstances of receiving two parameters, the position and orientation matrix of the end-effector as the input.

In this approach, we divide the solution into three parts, the first section is to determine the P_2 and θ_3 based on the given condition. The second part is to figure out the P_2, θ_0, θ_1 and θ_2 by using the plane geometric method, and the last one is to implement the geometric method to locate the 4^{th} point.

Since the P_0 and P_1 won't change its position because of the fixed structure, the function will finally return all the corresponding solutions in the angles space.

2.1 The first part:

By given the position of P_3 point and the final orientation, we can get the position of the point P_2 and the θ_3 in this scenario.

It is rather evident that the direction of the A_{34} is identical with the vector of the y-axis in the final orientation. Knowing that, we extract the elements in the second column of the orientation, and then normalize the vector to 1. Along with the given position of the P_3 point, the P_2 point can be figured out by apply the following relationship:

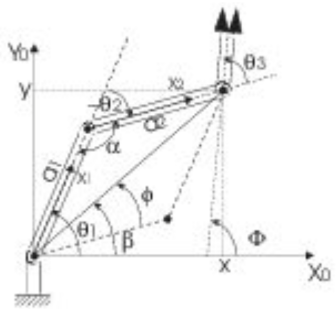
$$ \mathrm{P}_3 = P - 130 \times \overline{V} $$

In terms of the solution to θ_3, firstly, we should find out two vectors, the first one with the direction along the A_{23} and the other one is with the direction along the A_{34}, by implementing the geometric method, we can determine θ_3 with the given relationship below:

$$\theta_3 = \cos^{-1}\left(\frac{\overrightarrow{A_{23}} \bullet \vec{V}}{|\overrightarrow{A_{23}}| \times |\vec{V}|}\right)$$

2.2 The second part:

Given the condition of the position of P_3, applying plane geometric method shown in the slides of lecture6, and through simple calculation, we can get P_2, θ_0, θ_1 and θ_2 in this part.



The figure above gives all what we want to know, we pack all the functions in the auxiliaryFunction2.

Firstly, we can determine the θ_2 by:

$$\theta_2 = \cos^{-1}\left(\frac{x_1^2 + y_1^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

Then we introduce two assistant angles which call β and ϕ, with the calculation below, we can get the value of them by:

$$\beta = \tan^{-1}\left(\frac{y_1}{x_1}\right)$$

$$\phi = \cos^{-1}\left(\frac{x_1^2 + y_1^2 + l_1^2 - l_2^2}{2 l_1 \sqrt{x_1^2 + y_1^2}}\right)$$

After that, θ_1 is easy to reach simply by:

$$\theta_1 = \beta + \phi$$

Since P_0 can only rotate along its z axis, so θ_0 can be determined by x and y coordinates of the projection of P_3 in the XOY plane through the calculation:

$$\theta_0 = -\tan^{-1}\left(\frac{x}{y}\right)$$

Finally, with all the parameters we have, P_2 can be easily located.

2.3 The third part:

We firstly initialize the value of θ4 as 0, and create an auxiliary vector $\overrightarrow{nA}$
Then we use the same method as we apply in the first part:

$$\theta_4 = \cos^{-1}\left(\frac{v_{11}v_{21} + v_{12}v_{22} + v_{13}v_{23}}{|\overrightarrow{V_1}| \times |\overrightarrow{V_2}|}\right)$$

Therefore, the rotation angle along the y axis at P_4, which is also defined as θ_4, can be solved.

3. Accuracy analysis:

3.1 Test with random value

Direct Kinematics:

>> [P,A]=directKinematics()
Please input the rotation angle around z1 axis
that should be in range of (-80,80):rand*160-80

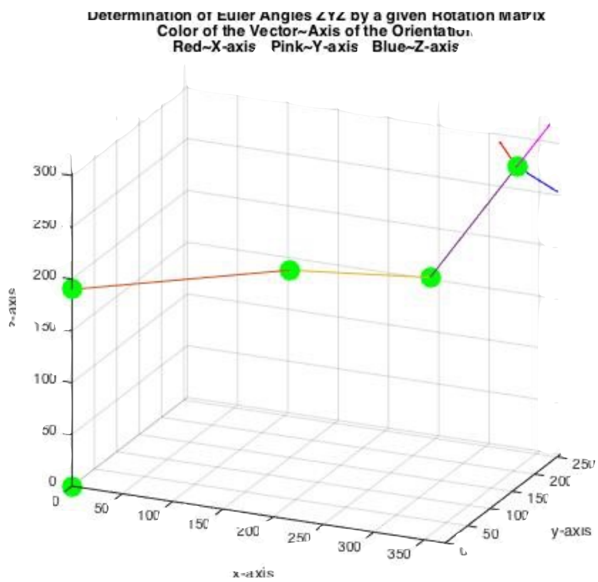
Please input the rotation angle around x1 axis
that should be in range of (-40,60):rand*100-40

Please input the rotation angle around x2 axis
that should be in range of (-100,0):rand*100-100

Please input the rotation angle around x3 axis
that should be in range of (-100,100):rand*200-100

Please input the rotation angle around y1 axis
that should be in range of (0,200):rand*200

Orientation in Euler Angles ZYZ:
The rotation angle around z1-axis is 47.63°
The rotation angle around y1-axis is 126.86°
The rotation angle around z2-axis is 99.09°



The coordinates of the end-effector in the base frame and the Matrix of the Orientation is:
P =

343.9831
220.8486
286.1487

A =

-0.6657	0.5159	0.5392
0.7354	0.3312	0.5912
0.1264	0.7901	-0.5999

Inverse Kinematics:
>> inverseKinematics(P,A)
Corresponding solutions in the angles space:

Firstly, the rotation angle along the axis z1, Theta0 = -57.30°
Then, the rotation angle along the axis x1, Theta1 = 2.18°
Next, the rotation angle along the axis x2, Theta2 = -8.43°
After that, the rotation angle along the axis x3, Theta3 = 58.44°
Finally, the rotation angle along the axis y1, Theta4 = 191.90°

3.2 Test with specific value

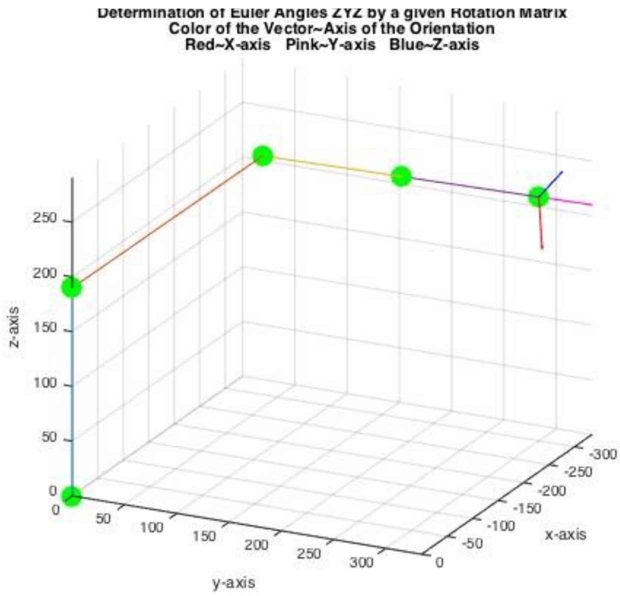
Direct Kinematics:
>> [P,A]=directKinematics()
Please input the rotation angle around z1 axis that should be in range of (-80,80):45

Please input the rotation angle around x1 axis that should be in range of (-40,60):30

Please input the rotation angle around x2 axis that should be in range of (-100,0):-45

Please input the rotation angle around x3 axis that should be in range of (-100,100):0

Please input the rotation angle around y1 axis that should be in range of (0,200):45



Orientation in Euler Angles ZYZ:
The rotation angle around z1-axis is 59.51°
The rotation angle around y1-axis is 46.92°
The rotation angle around z2-axis is -20.75°

The coordinates of the end-effector in the base frame and the Matrix of the Orientation is:
P =

-300.0578
300.0578
222.7070

A =

0.6294	-0.6830	0.3706
0.3706	0.6830	0.6294
-0.6830	-0.2588	0.6830

Inverse Kinematics:
>> inverseKinematics(P,A)
Corresponding solutions in the angles space:

Firstly, the rotation angle along the axis z1, Theta0 = 45.00°
Then, the rotation angle along the axis x1, Theta1 = 30.00°
Next, the rotation angle along the axis x2, Theta2 = -45.00°
After that, the rotation angle along the axis x3, Theta3 = 0.00°
Finally, the rotation angle along the axis y1, Theta4 = 45.00°

4. Code documentation

The Matlab™ code is made up by 5 parts:

- directKinematics.m
- inverseKinematics.m
- auxiliaryFunction1.m
- auxiliaryFunction2.m
- draw.m

You could always execute 【directKinematics.m】 and 【inverseKinematics.m】 , other three functions will be automatically called when you execute the two main functions. The matrices below are all in three dimension, so the position and orientation are presented separately throughout the program. Input three-dimensional matrix, and you will get the right answer.

4.1 directKinematics.m

function [P, A] = directKinematics()

Parameters: none

Return Values:

1. P: The position of the end-effector point
2. A: The Orientation of the last arm

Input Values:

1. θ0: the rotation angle along the axis z_1
2. θ1: the rotation angle along the axis x_1
3. θ2: the rotation angle along the axis x_2
4. θ3: the rotation angle along the axis x_3
5. θ4: the rotation angle along the axis y_1

Output Values:

1. Orientation in Euler Angles ZYZ
2. A painting that shows the shape of the robot

4.2 inverseKinematics.m

function [ ] = inverseKinematics(P,A)

Parameters:

1. P: The position of the end-effector point
2. A: The Orientation of the last arm

Return Values: none

Input Values: none

Output Values:

1. Five angles corresponding the position and orientation of the robot
2. A painting that shows the shape of the robot

4.3 auxiliaryFunction1.m

function [P,A] = auxiliaryFunction1(Theta0,Theta1,Theta2,Theta3,Theta4)

Parameters:

1. Theta0: the rotation angle along the axis z_1
2. Theta1: the rotation angle along the axis x_1
3. Theta2: the rotation angle along the axis x_2
4. Theta3: the rotation angle along the axis x_3
5. Theta4: the rotation angle along the axis y_1

Return Values:

1. P: The position of the end-effector point
2. A: The Orientation of the last arm

Input Values: none

Output Values: none

4.4 auxiliaryFunction2.m

function [n1,n2,n3] = auxiliaryFunction2(x,y,z)

Parameters:

1.x: x axis of the end-effort point
2.y: y axis of the end-effort point
3.z: z axis of the end-effort point

Return Values:

1. n1: the rotation angle along the axis z_1
2. n2: the rotation angle along the axis x_1
3. n3: the rotation angle along the axis x_2

Input Values: none
Output Values: none

4.5 draw.m

function [] = draw(P3,P4,P5,rotationMatrix)

Parameters:

1. P3: the position of the origin point of frame x_3-y_3-z_3
2. P4: the position of the origin point of frame x_4-y_4-z_4
3. P5: the position of the end-effector point
4. rotationMatrix: the orientation of the end-effector

Return Values: none
Input Values: none
Output Values: A painting that shows the shape of the robot

5 Group member

Name	Jialin Wang	Cong Peng
IST Number	ist423929	ist423925
Course	MEEC	MEEC
E-mail	buptjialin@icloud.com	cleverleypc@icloud.com