Fundamentals of Robotics HW 3

Ahmad Hamdan

November 12, 2020

Abstract

This report is part of Fundamentals of Robotics course for 3d year students at Innopolis University. In this report I am working on KR 10 R1100-2 manipulator designed by KUKA, where I am calculating the robot Jacobian using both Skew theory and Numerical method. And in the final step I am analysing the Kinematic Singularities. All the methods are explained in this report while the implementation is done using Matlab and it is available on GitHub.

1 Jacobian using Skew Theory

The kinematic scheme for KR 10 R1100-2 is described in Figure 1.

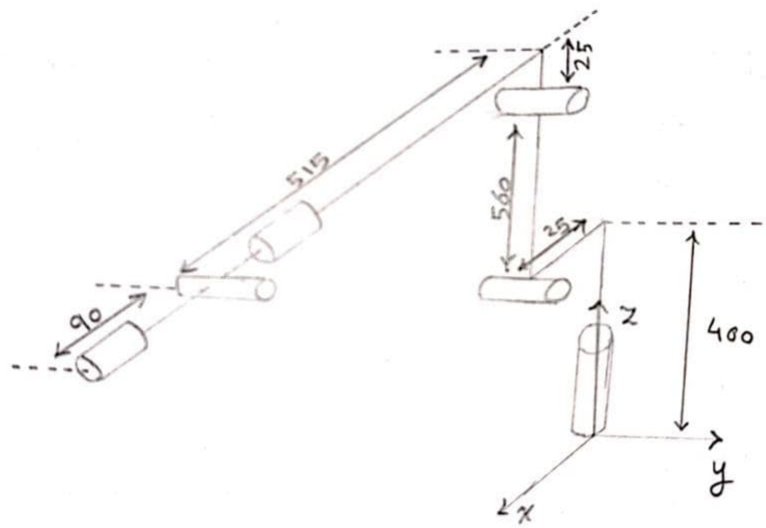

Figure 1: KR 10 R1100-2 Kinematic Scheme

For this method we will calculate the following transformations:

$$T_1 = T_z(0.4)R_z(q_1)T_x(0.025)$$

$$T_2 = T_1R_y(q_2)T_z(0.56)$$

$$T_3 = T_2R_y(q_3)T_z(0.025)T_x(0.515)$$

$$T_4 = T_3R_x(q_4)$$

$$T_5 = T_4R_y(q_5)$$

$$T_6 = T_5R_x(q_6)$$

From above we can see that I defined T_i as the transformation from the base frame to the joint q_i frame. The Jacobian for revolute joints is calculated using the following equation:

$$J_i = \begin{bmatrix} Z_{i-1} \times (O_n - O_{i-1}) \\ Z_{i-1} \end{bmatrix}$$

Where O_i is the positional vector from the T_i transformation matrix. On the other hand, Z_i is the rotation matrix from T_i transformation cross product the column vector corresponds to the axis of rotation.

Finally the Jacobian will be written in the following format:

$$J = \begin{bmatrix} J_1 & J_2 & J_3 & J_1 & J_2 & J_3 \end{bmatrix}$$

2 Jacobian using Numerical Derivatives

Firstly we calculate the Forward Kinematics :

$$T = T_z(0.4)R_z(q_1)T_x(0.025)R_y(q_2)T_z(0.56)R_y(q_3)T_z(0.025)T_x(0.515)R_x(q_4)R_y(q_5)R_x(q_6)T_x(0.09)$$

Then we get the rotation matrix R from T and calculate its inverse. After that we compose with a homogeneous matrix with zero position vector and I will name it T_0

Now for each joint we compute the following:

$$\dot{T}_i = T_{left}\dot{H}_iT_{right}T_0$$

Where $\dot{H}_i$ is the derivative of the transformation that depends on the ith joint. Now we can obtain J as follows:

$$J_i = \begin{bmatrix} \dot{T}_i(1,4) \\ \dot{T}_i(2,4) \\ \dot{T}_i(3,4) \\ \dot{T}_i(3,2) \\ \dot{T}_i(1,3) \\ \dot{T}_i(2,1) \end{bmatrix}$$

I have implemented it in Matlab as you notice in the next page:

```
T=FK(q);
R=T(1:3,1:3);
j1=Tz(0.4)*Rzd(q(1))*Tx(0.025)*Ry(q(2))*Tz(0.56)*Ry(q(3))*Tz(0.025)*Tx(0.515)
    *Rx(q(4))*Ry(q(5))*Rx(q(6))*Tx(0.09)*[inv(R) zeros(3,1); 0 0 0 1];

J1=[j1(1,4);j1(2,4);j1(3,4);j1(3,2);j1(1,3);j1(2,1)];

j2=Tz(0.4)*Rz(q(1))*Tx(0.025)*Ryd(q(2))*Tz(0.56)*Ry(q(3))*Tz(0.025)*Tx(0.515)
    *Rx(q(4))*Ry(q(5))*Rx(q(6))*Tx(0.09)*[inv(R) zeros(3,1); 0 0 0 1];

J2=[j2(1,4);j2(2,4);j2(3,4);j2(3,2);j2(1,3);j2(2,1)];

j3=Tz(0.4)*Rz(q(1))*Tx(0.025)*Ry(q(2))*Tz(0.56)*Ryd(q(3))*Tz(0.025)*Tx(0.515)
    *Rx(q(4))*Ry(q(5))*Rx(q(6))*Tx(0.09)*[inv(R) zeros(3,1); 0 0 0 1];

J3=[j3(1,4);j3(2,4);j3(3,4);j3(3,2);j3(1,3);j3(2,1)];

j4=Tz(0.4)*Rz(q(1))*Tx(0.025)*Ry(q(2))*Tz(0.56)*Ry(q(3))*Tz(0.025)*Tx(0.515)
    *Rxd(q(4))*Ry(q(5))*Rx(q(6))*Tx(0.09)*[inv(R) zeros(3,1); 0 0 0 1];

J4=[j4(1,4);j4(2,4);j4(3,4);j4(3,2);j4(1,3);j4(2,1)];

j5=Tz(0.4)*Rz(q(1))*Tx(0.025)*Ry(q(2))*Tz(0.56)*Ry(q(3))*Tz(0.025)*Tx(0.515)
    *Rx(q(4))*Ryd(q(5))*Rx(q(6))*Tx(0.09)*[inv(R) zeros(3,1); 0 0 0 1];

J5=[j5(1,4);j5(2,4);j5(3,4);j5(3,2);j5(1,3);j5(2,1)];

j6=Tz(0.4)*Rz(q(1))*Tx(0.025)*Ry(q(2))*Tz(0.56)*Ry(q(3))*Tz(0.025)*Tx(0.515)
    *Rx(q(4))*Ry(q(5))*Rxd(q(6))*Tx(0.09)*[inv(R) zeros(3,1); 0 0 0 1];

J6=[j6(1,4);j6(2,4);j6(3,4);j6(3,2);j6(1,3);j6(2,1)];

J=[J1 J2 J3 J4 J5 J6];
```

3 Singularity Analysis

Singularity cases can be determinde from the jacobian matrix using three methods:

1. Checking the rank of the Jacobian matrix
2. The diagonal matrix S from SVD has extremely small values.
3. The Jacobian matrix determinant is equal to zero

We have three type of singularities and I will discuss them one by one.

3.1 Shoulder Singularity

This case happens when the end-effector coincides with the axis of rotation of the base. Such case is considered singularity, no matter how we move the first joint it will not affect x and y position of the end-effector. example:

$$q = \begin{bmatrix} 0 \\ 0 \\ -\frac{\pi}{2} \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

3.2 Elbow Singularity

This case occurs once the manipulator extends fully, in such case the control is lost on several axis and directions. For example the robot can move on the Z axis neither the X nor Y axis. Example:

$$q = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ -\frac{\pi}{2} \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

3.3 Wrist Singularity

This case happens when the axis of joints 4 and 6 happen to be collinear. in this case 1 DOF is lsot. Example:

$$q = \begin{bmatrix} 0 \\ 1.8 \\ -\frac{\pi}{2} \\ -\frac{\pi}{2} \\ 0 \\ \frac{\pi}{2} \end{bmatrix}$$

I check singularities in my code using SVD and rank functions.

4 GitHub

All files can be found here