This report is part of Dynamics of Non Linear Robotic Systems [DNLRS] course for first year master students at Innopolis University. In this report I am working on Mitsubishi robot RV-1A. First of all, I am deriving the robot jacobian matrix using Skew theory and Numerical method and implementing both methods using Matlab R2021a and it is available on GitHub.

1 Jacobian using screw theory

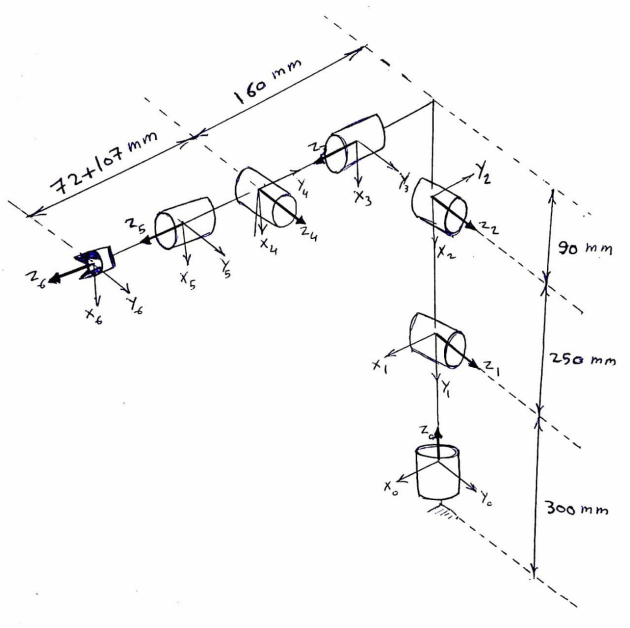

Figure 1: Mitsubishi robot RV-1A scheme.

First of all we have to calculate these transformations:

$$T_1^0 = R_z(q_1) \cdot T_z(0.3) \tag{1}$$

$$T_2^0 = T_1^0 \cdot R_y(q_2) \cdot T_z(0.25) \tag{2}$$

$$T_3^0 = T_2^0 \cdot R_y(q_3) \cdot T_z(0.09) \cdot T_x(0.160) \tag{3}$$

$$T_4^0 = T_3^0 \cdot R_x(q_4) \tag{4}$$

$$T_5^0 = T_4^0 \cdot R_y(q_5) \tag{5}$$

$$T_6^0 = T_5^0 \cdot R_x(q_6) \cdot T_x(0.072 + 0.107) \tag{6}$$

The Jacobian for revolute joints is calculated using the following equation:

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

where O_i is the positional vector from T_i transformation matrix, and Z_i is the rotation matrix from T_i transformation cross product the column vector corresponds to the axis of rotation. The Jacobian will be written in the following format:

$$J = \begin{bmatrix} J_1 & J_2 & J_3 & J_4 & J_5 & J_6 \end{bmatrix}$$

2 Jacobian using Numerical method

First of all we need to calculate the Forward Kinematics:

$$T = R_z(q_1)\cdot T_z(0.3)\cdot R_y(q_2)\cdot T_z(0.25) R_y(q_3)\cdot T_z(0.09)\cdot T_x(0.160)\cdot R_x(q_4)\cdot R_y(q_5)\cdot R_x(q_6)\cdot T_x(0.179) \tag{7}$$

Then we get the rotation matrix R from T so now we can calculate its inverse. After that we compose it with a homogeneous matrix with zero position vector and let it be T_0. We can generalize the solution for each joint using the following structure:

$$\dot{T}_i = T_{left} \cdot \dot{H}_i \cdot T_{right} \cdot T_0 \tag{8}$$

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

$$J_i = \begin{bmatrix} \dot{T}_i(1,4) \\ \dot{T}_i(2,4) \\ \dot{T}_i(3,4) \\ \dot{T}_i(3,2) \\ \dot{T}_i(1,3) \\ \dot{T}_i(2,1) \end{bmatrix} \tag{9}$$

3 Singularity Analysis

A robot singularity is a configuration in which the robot end-effector becomes blocked in certain directions.
We can talk discuss snigularity in different approaches as we mentioned previously in the lecture:

(a) Algebra: Jacobian matrix becomes singular.

(b) Geometry: The joint screws (lines) are linearly dependent.

(c) Kinematics:The manipulator (instantaneously) loses one or more degrees of freedom.

(d) Statics:There exists one or more wrenches that can be resisted without turning on the actuators

So can use the Jacobian matrix to check the singularities by one of these three methods:

(a) The Jacobian matrix determinate is equal to Zero.

(b) By using the SVD function on Matlab and dividing the highest value of matrix S on its lowest value and if the answer is then this is a singularity case. In other words, we can say if diagonal matrix S from SVD has extremely small values then this is a singularity case.

(c) Checking the rank of the Jacobian.

I have entered many initial cases of q and checked their singularities using these methods above, and one of the singularity that I have faced is:

$$q = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ \frac{-\pi}{2} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

This singularity is called $ElbowSingularity$, in which occurs when the wrist center (the point where the axes of joints 4, 5 and 6 intersect) lies on the plane passing through the axes of joints 2 and 3. We can say that, in an elbow singularity, the arm is fully stretched as it shown in Figure 2.

There is two other types of singularities: Wrist and shoulder singularity, and we can check any situation of the arm using my code. All files can be found on GitHub

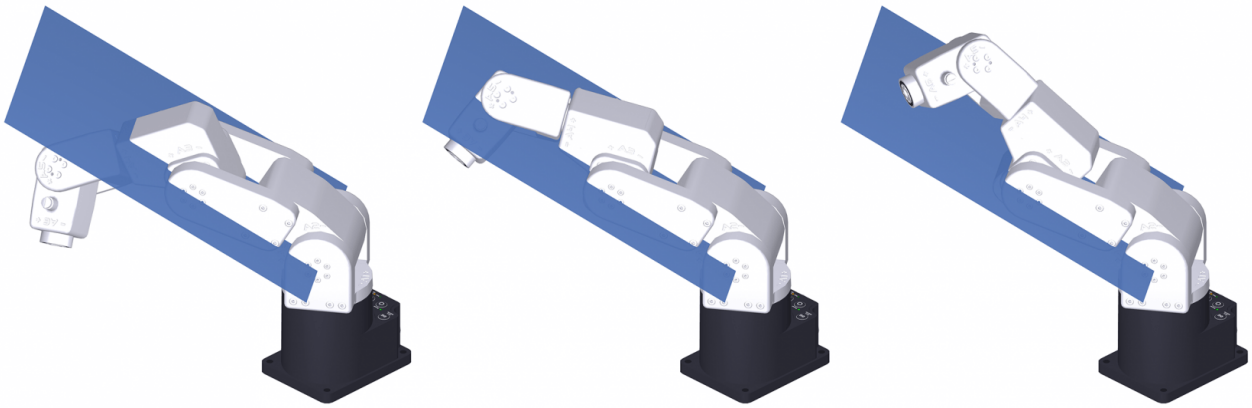

Figure 2: Elbow singularity (center) and elbow-up (left) and elbow-down (right) conditions.