ME 639

INTRODUCTION TO ROBOTICS

Assignment 3 & 4 (Part – 1)

Submitted by:

Rhitosparsha Baishya
23310039
PhD (Mechanical Engineering)

CONTENTS

Question 1

A **singular configuration** is a configuration where certain directions of motion at the end effector become impossible or problematic. It can be defined as a state where bounded end-effector velocities correspond to unbounded joint velocities. When a robot reaches singularity configuration, there is no unique solution for the inverse kinematic problem.

A singularity configuration can be detected in the following ways:

1. When the determinant of the Jacobian approaches zero, the manipulator reaches a near-singularity configuration.
2. If the rank of the Jacobian is less than the number of degrees of freedom of the robot, it indicates that the robot is at or near singularity.
3. Eigen values approaching zero also indicate singularity configurations.

Question 3

Source Code: ([Link to GitHub]())

```
import numpy as np

def compute_jacobian_and_velocity(links, dh_parameters, joint_types=None):
    if joint_types is None:
        # Default assumption: All joints are revolute
        joint_types = ["R"] * len(links)

    if len(links) != len(dh_parameters) or len(links) != len(joint_types):
        raise ValueError("Input lengths do not match")

    # Initialize the transformation matrix
    T = np.eye(4)

    # Initialize the Jacobian matrix
    J = np.zeros((6, len(links)))

    # Initialize the end-effector position
    end_effector_position = np.zeros(3)

    # Initialize the end-effector velocity
    end_effector_velocity = np.zeros(6)

    for i in range(len(links)):
        a, alpha, d, theta = dh_parameters[i]
        if joint_types[i] == 'R':  # Revolute joint
            theta = links[i]

        # Calculate the transformation matrix
        A = np.array([[np.cos(theta), -np.sin(theta) * np.cos(alpha),
np.sin(theta) * np.sin(alpha), a * np.cos(theta)],
                      [np.sin(theta), np.cos(theta) * np.cos(alpha), -
np.cos(theta) * np.sin(alpha), a * np.sin(theta)],
                      [0, np.sin(alpha), np.cos(alpha), d],
                      [0, 0, 0, 1]])

        # Update the transformation matrix
        T = np.dot(T, A)

        # Calculate the rotational part of the Jacobian
        z_i_minus_1 = T[:3, 2]
        end_effector_position = T[:3, 3]
        J[:3, i] = np.cross(z_i_minus_1, end_effector_position)
```

```
        # Calculate the translational part of the Jacobian
        J[3:, i] = z_i_minus_1

    return J, end_effector_position, end_effector_velocity

# Default values:
links = [0.1, 0.2, 0.15]
dh_parameters = [
    [0.1, 0, 0.05, 0.1],
    [0.05, 0, 0, 0.2],
    [0.03, 0, 0, 0.15]
]
joint_types = ["R", "R", "R"]

choice = int(input("Do you want to :\n1. Use default values?\n2. Enter your
own values?\nEnter option no. : "))

n = int(input("\nEnter the number of links : "))

# Take inputs if user decides to input their own values
if choice == 2:
    print("\nEnter the following values row-wise, separated by spaces :")
    links =  list(map(float, input("Enter link lengths in m  : ").split()))
    e2 =  list(map(float, input("Enter values of DH Parameters (Format : a,
alpha, d, theta) : ").split()))
    dh_parameters = np.array(e2).reshape(n, 4)
    joint_types =  list(map(str, input("Enter joint types (R or P) :
").split()))

print("\nLink lengths (m) :")
print(links)
print("\nDH Parameters (Format : a, alpha, d, theta) : ")
print(dh_parameters)
print("\nJoint types : ")
print(joint_types)

jacobian, end_effector_position, end_effector_velocity =
compute_jacobian_and_velocity(links, dh_parameters, joint_types)

print("\nManipulator Jacobian:")
print(jacobian)
print("\nEnd-effector Position:")
print(end_effector_position)
print("\nEnd-effector Velocity:")
print(end_effector_velocity)
```

Output:

```
Do you want to :
1. Use default values?
2. Enter your own values?
Enter option no. : 2

Enter the number of links : 2

Enter the following values row-wise, separated by spaces :
Enter link lengths in m  : 2 2
Enter values of DH Parameters (Format : a, alpha, d, theta) : 2 0 0 0.7854 2 0 0 0.7854
Enter joint types (R or P) : R R

Link lengths (m) :
[2.0, 2.0]

DH Parameters (Format : a, alpha, d, theta) :
[[2.     0.     0.     0.7854]
 [2.     0.     0.     0.7854]]

Joint types :
['R', 'R']

Manipulator Jacobian:
[[-1.81859485 -0.30498986]
 [-0.83229367 -2.13958091]
 [ 0.          0.        ]
 [ 0.          0.        ]
 [ 0.          0.        ]
 [ 1.          1.        ]]

End-effector Position:
[-2.13958091  0.30498986  0.        ]

End-effector Velocity:
[0. 0. 0. 0. 0. 0.]
```

Question 4

Output for Stanford Manipulator:

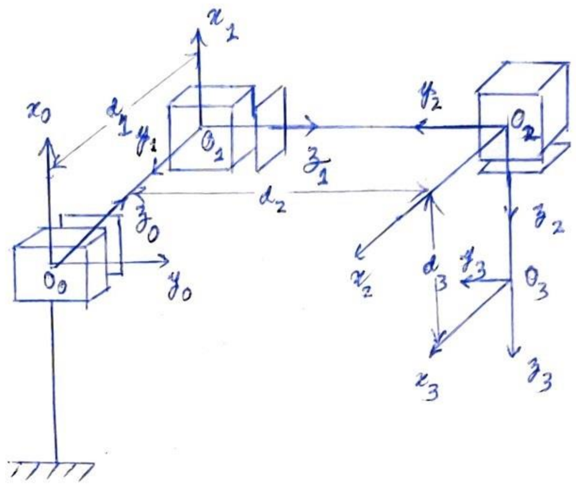
```
Do you want to :
1. Use default values?
2. Enter your own values?
Enter option no. : 2

Enter the following values row-wise, separated by spaces :
Enter link lengths in m  : 1 1 1
Enter values of DH Parameters (Format : a, alpha, d, theta) : 0 -1.5707 0 0.7853 0 1.5707 1 0.7853 0 0 2 0
Enter joint types (R or P) : R R P

Link lengths (m) :
[1.0, 1.0, 1.0]

DH Parameters (Format : a, alpha, d, theta) :
[[ 0.     -1.5707  0.      0.7853]
 [ 0.      1.5707  1.      0.7853]
 [ 0.      0.      2.      0.    ]]

Joint types :
['R', 'R', 'P']

Manipulator Jacobian:
[[ 0.00000000e+00 -2.91858374e-01 -2.91858374e-01]
 [ 0.00000000e+00 -4.54692506e-01 -4.54692506e-01]
 [-0.00000000e+00  8.41470977e-01  8.41470977e-01]
 [-8.41470981e-01  4.54611450e-01  4.54611450e-01]
 [ 5.40302303e-01  7.08097340e-01  7.08097340e-01]
 [ 9.63267947e-05  5.40302310e-01  5.40302310e-01]]

End-effector Position:
[0.06775192 1.95649698 1.08070095]

End-effector Velocity:
[0. 0. 0. 0. 0. 0.]
```

Output for SCARA Manipulator:

```
Do you want to :
1. Use default values?
2. Enter your own values?
Enter option no. : 2

Enter the following values row-wise, separated by spaces :
Enter link lengths in m  : 1 1 1
Enter values of DH Parameters (Format : a, alpha, d, theta) : 0 -1.5707 0 0.7853 0 1.5707 1 0.7853 0 0 2 0
Enter joint types (R or P) : R R P

Link lengths (m) :
[1.0, 1.0, 1.0]

DH Parameters (Format : a, alpha, d, theta) :
[[ 0.     -1.5707  0.      0.7853]
 [ 0.      1.5707  1.      0.7853]
 [ 0.      0.      2.      0.    ]]

Joint types :
['R', 'R', 'P']

Manipulator Jacobian:
[[ 0.00000000e+00 -2.91858374e-01 -2.91858374e-01]
 [ 0.00000000e+00 -4.54692506e-01 -4.54692506e-01]
 [-0.00000000e+00  8.41470977e-01  8.41470977e-01]
 [-8.41470981e-01  4.54611450e-01  4.54611450e-01]
 [ 5.40302303e-01  7.08097340e-01  7.08097340e-01]
 [ 9.63267947e-05  5.40302310e-01  5.40302310e-01]]

End-effector Position:
[0.06775192 1.95649698 1.08070095]

End-effector Velocity:
[0. 0. 0. 0. 0. 0.]
```

Question 5

Q5



DH parameters ;

Link	d	θ	a	α
1	d_1	0	0	$-\pi/2$
2	d_2	$\pi/2$	0	$\pi/2$
3	d_3	$-\pi/2$	0	0

$$H_0^1 = \begin{bmatrix} R_{x,-\pi/2} & T_{z,d_1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_1^2 = \begin{bmatrix} R_1^2 & d_1^2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^3 = \begin{bmatrix} R_2^3 & d_2^3 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

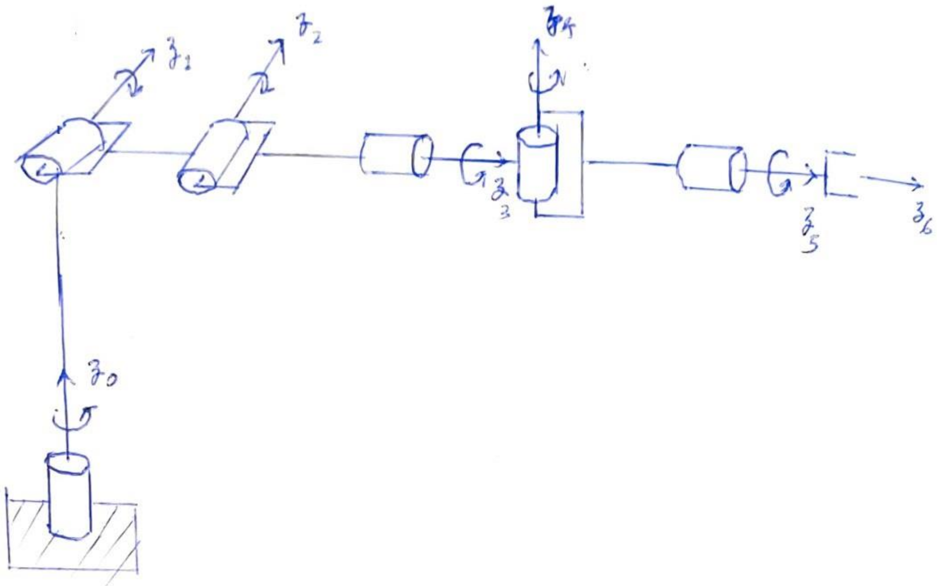
$$H_0^3 = H_0^1 H_1^2 H_2^3$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 & d_3 \\ -1 & 0 & 0 & d_2 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question 6

Q6



DH parameters:

Link	d	θ	a	α
1	0	θ_1	0	$\pi/2$
2	0	θ_2	a_2	0
3	0	θ_3	a_3	0
4	0	θ_4	0	$-\pi/2$
5	0	θ_5	0	0
6	d_6	θ_6	0	0

$$A_1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \quad T_0^6 = \begin{bmatrix} n_{11} & n_{12} & n_{13} & d_x \\ n_{21} & n_{22} & n_{23} & d_y \\ n_{31} & n_{32} & n_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $n_{11} = c_1 (c_5 c_6 c_{234} - s_6 s_{234}) - s_1 s_5 c_6$

$n_{12} = -c_1 (c_5 s_6 c_{234} + c_6 s_{234}) + s_1 s_5 s_6$

$n_{13} = c_1 s_5 c_{234} + s_1 c_5$

$d_x = a_2 c_1 c_2 + a_3 c_1 c_{23} + d_6 (c_1 s_5 c_{234} + s_1 c_5)$

$n_{21} = c_1 s_5 s_6 + s_1 c_5 c_6 c_{234} - s_1 s_6 s_{234}$

$n_{22} = - c_1 s_5 s_6 - s_1 c_5 s_6 c_{234}$

$n_{23} = - c_1 c_5 + s_1 s_5 c_{234}$

$$d_y = a_2 s_1 c_2 + a_3 s_1 c_{23} - d_6 \left(c_1 c_5 + s_1 s_5 c_{234} \right)$$

$$r_{31} = s_6 c_{234} + c_5 s_6 s_{234}$$

$$r_{32} = c_6 s_{234} - c_5 s_6 s_{234}$$

$$r_{33} = s_5 s_{234}$$

$$d_z = a_2 s_2 + a_3 c_2 s_{23} + d_6 s_5 s_{234}$$

$$\Rightarrow \quad T_0^6 = \begin{bmatrix} -c_6 s_5 & s_5 s_6 & c_5 & d_3 + d_6 c_5 \\ -c_4 c_5 c_6 + s_4 s_6 & c_4 c_5 s_6 + c_6 s_4 & -c_4 s_5 & d_2 - d_6 c_4 s_5 \\ -c_4 s_6 - c_5 c_6 s_4 & -c_4 c_6 + c_5 s_4 s_6 & -s_4 s_5 & d_1 - d_6 s_4 s_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question 7

Direct Drive	**Remotely-Driven**	**5-bar Parallelogram Arrangement**
Both the links are directly actuated by motors present at each joint. Advantages: • It allows precise end-effector position control. • Independent control is possible.	The first link is directly actuated by motor present at the joint and the second link is connected to the motor by a gearing or belt-pulley system. Advantages: • Wiring and control is simpler. • Less weight.	Both the links are connected in a parallelogram configuration with a common base. Advantages: • Ensures that end-effector remains parallel to base. • Very stable.

Question 8

Q8

We know,

$$D(q)\ddot{q} + C(q,\dot{q}) + g(q) = \tau$$

For 2R Manipulator,

$$D(q) = \begin{bmatrix} m_1 l_1^2/4 + m_2 l_1^2 + I_1 & m_2 l_1 l_2/2 \cos(q_2 - q_1) \\ m_2 l_1 l_2/2 \cos(q_2 - q_1) & m_2 l_2^2/4 + I_2 \end{bmatrix}$$

$$C_{ijk} = \frac{1}{2}\left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ik}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right)$$

$$C_{111} = \frac{1}{2}\left(\frac{\partial d_{11}}{\partial q_1} + \cancel{\frac{\partial d_{11}}{\partial q_1}} - \cancel{\frac{\partial d_{11}}{\partial q_1}}\right) = \frac{1}{2}\left(\frac{\partial d_{11}}{\partial q_1}\right) = 0$$

$$C_{121} = C_{211} = \frac{1}{2}\left(\frac{\partial d_{11}}{\partial q_2} + \frac{\partial d_{21}}{\partial q_1} - \frac{\partial d_{12}}{\partial q_2}\right) = \frac{1}{2}\left(\frac{\partial d_{11}}{\partial q_2}\right) = 0$$

$$C_{221} = \frac{1}{2}\left(\frac{\partial d_{21}}{\partial q_1} + \frac{\partial d_{21}}{\partial q_2} + \frac{\partial d_{22}}{\partial q_1}\right) = \frac{\partial d_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = -m_2 l_1 l_2/2\, s_{2-1}$$

$$C_{112} = \frac{1}{2}\left(\frac{\partial d_{12}}{\partial q_1} + \frac{\partial d_{12}}{\partial q_1} - \frac{\partial d_{11}}{\partial q_2}\right) = \frac{\partial d_{12}}{\partial q_1} - \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = m_2 l_1 l_2/2\, s_{2-1}$$

$$C_{212} = C_{122} = \frac{1}{2}\left(\frac{\partial d_{22}}{\partial q_1} + \frac{\partial d_{12}}{\partial q_2} - \frac{\partial d_{21}}{\partial q_2}\right) = \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = 0$$

$$C_{222} = \frac{1}{2}\left(\frac{\partial d_{22}}{\partial q_2} + \cancel{\frac{\partial d_{22}}{\partial q_2}} - \cancel{\frac{\partial d_{22}}{\partial q_2}}\right) = \frac{1}{2}\frac{\partial d_{22}}{\partial q_2} = 0$$

Potential energy:

$$V = m_1 g\, l_1/2\, s_1 + m_2 g\left(l_1 s_1 + l_2/2\, s_2\right)$$

$$\phi_1 = \frac{\partial V}{\partial q_1} = \left(m_1 l_1/2 + m_2 l_1\right) g\, c_1$$

$$\phi_2 = \frac{\partial V}{\partial q_2} = m_2 g\, l_2/2\, c_2$$

∴ Final eq^n :

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + C_{221}\dot{q}_2^2 + \phi_1 = \tau_1$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + C_{112}\dot{q}_1^2 + \phi_2 = \tau_2$$

$$\Rightarrow \boxed{\begin{aligned}
&\left(m_1 l_1^2/4 + m_2 l_1^2 + I_1\right)\ddot{q}_1 + \left(m_2 l_1 l_2/2 \, C_{2-1}\right)\ddot{q}_2 \\
&\quad - \left(m_2 l_1 l_2/2 \, S_{2-1}\right)\dot{q}_2^2 + \left(m_1 l_1/2 + m_2 l_1\right) g \, C_1 = \tau_1 \\
&\qquad\qquad \text{and} \\
&\left(m_2 l_1 l_2/2 \, C_{2-1}\right)\ddot{q}_1 + \left(m_2 l_2^2/4 + I_2\right)\ddot{q}_2 \\
&\quad + \left(m_2 l_1 l_2/2 \, S_{2-1}\right)\dot{q}_1^2 + m_2 g \, l_2/2 \, C_2 = \tau_2
\end{aligned}}$$

∴ These are the same eq^n as used in mini-project//

Question 10

Q10

Suppose were given $D(q)$ and $V(q)$

Step - 1: compute the Christoffel symbols of first kind,

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right)$$

Step - 2: Define the terms

$$\phi_k = \frac{\partial V}{\partial q_k}$$

Step-3: Combine the terms as follows:

$$\sum_j d_{kj}(q)\ddot{q}_j + \sum_{i,j} c_{ijk}(q)\dot{q}_i\dot{q}_j + \phi_k(q) = \tau_k$$

where $k = 1, \dots, n$

In matrix form:

$$\boxed{D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau}$$

REFERENCES

1. Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control*. https://www.kramirez.net/Robotica/Tareas/Kinematics.pdf

2. Take Matrix input from user in Python: https://www.geeksforgeeks.org/take-matrix-input-from-user-in-python/

3. NumPy Documentation: https://numpy.org/doc/stable/

4. ChatGPT: https://chat.openai.com/