Assignment $\frac{3}{4}$

Shashank Ghosh
21110196

Q.1 A singular configuration in robotics refers to a specific arrangement of a robot's joints where it loses one or more degrees of freedom, resulting in limited or no unique solutions for its motion. Singular configurations are often identified by examining the robot's Jacobian matrix, where its determinant becomes zero, indicating the loss of manipulability. Detecting proximity to a singular configuration using the manipulator Jacobian involves observing abrupt changes or rapid fluctuations in the determinant or condition number of the Jacobian matrix as the robot approaches certain joint configurations. These changes signify an impending loss of dexterity or motion redundancy.

Q.3 In the provided code **Q3.py** i've written a function that takes the input 'dh_parameters' that needs to be provided explicitly(subroutine). So the code won't give any outputs, it will just return the jacobian and ee position and ee velocity.

Q.4

Stanford Manipulator (RRP Configuration)

For the RRP configuration of the Stanford manipulator, let's define the DH parameters and joint types:

DH parameters:

Link 1: alpha = 0, a = 0, d = 1, theta = $\theta 1$
Link 2: alpha = 0, a = 1, d = 0, theta = $\theta 2$
Link 3 (Prismatic): alpha = 0, a = 0, d = 0, theta = $\theta 3$ (Prismatic joint)

Joint types:

Revolute - Revolute - Prismatic

Therefore we can create dh_parameters for stanford manipulator as :-

```
[
    [0, 0, 1, 'theta1'],
    [0, 1, 0, 'theta2'],
    [0, 0, 0, 'theta3']
]
```

Passing this in the above code will return Jacobian, EE velocity and EE Position.

SCARA Manipulator

For the SCARA manipulator, the DH parameters and joint types:

DH parameters:

Link 1: alpha = 0, a = 1, d = 0, theta = $\theta 1$

Link 2: alpha = 0, a = 1, d = 0, theta = $\theta 2$ (Revolute)
Link 3: alpha = 0, a = 0, d = 1, theta = $\theta 3$
Joint types:
Revolute - Revolute - Prismatic

Similarly we can write dh_parameters for the purpose of verifying the code :-

```
dh_params_scara = [
            [0, 1, 0, 'theta1'],
            [0, 1, 0, 'theta2'],
              [0, 0, 1, 'theta3']
]
```
We can put values of theta1,2,3 according to our convenience.

Q.5
The 2R manipulator refers to a two-link manipulator composed of two revolute joints, often used in robotics. The three different configurations - direct drive, remotely-driven, and 5-bar parallelogram arrangement - can significantly affect the manipulator's characteristics and performance. Let's explore each configuration along with their key differences and advantages:

1. Direct Drive Configuration:
In a direct drive 2R manipulator, both joints are actuated directly, meaning each joint is equipped with its own motor or actuator. This arrangement allows independent control of each joint angle.

Advantages:

Individual Joint Control: Independent actuation of each joint provides precise control over each link's motion.
Flexibility: It allows for diverse motion trajectories and improved manipulability by controlling each joint separately.
Simplicity: Each joint has its motor or actuator, which simplifies control and kinematic analysis.
Key Difference: The primary distinction is the individual actuation of each joint, providing enhanced control and versatility.

2. Remotely-Driven Configuration:
In a remotely-driven 2R manipulator, one joint is directly actuated, while the second joint's motion is controlled indirectly, often through linkages or a transmission system connected to the primary actuator.

Advantages:

Simplicity and Cost: This configuration may reduce the number of actuators needed, thereby lowering cost and complexity.
Space Efficiency: Centralized actuation can save space, making it suitable for constrained environments.
Reduced Weight: Potentially lighter compared to direct drive due to fewer actuators.
Key Difference: One joint is directly actuated, while the other joint's motion is controlled indirectly through mechanical linkages or transmissions, simplifying the overall actuation system.

3. 5-Bar Parallelogram Arrangement:
This arrangement involves a parallelogram linkage in addition to the 2R manipulator's two links and joints. It typically incorporates an extra link and a passive joint to form a parallelogram structure.

Advantages:

Stability: The parallelogram structure offers increased stability and reduced sway during operation.
Enhanced Rigidity: The additional linkage can enhance rigidity, especially in certain orientations or configurations.
Load Distribution: It may distribute loads more evenly across the structure, potentially reducing stress on individual components.
Key Difference: The inclusion of a parallelogram structure aims to improve stability, rigidity, and load distribution, offering advantages in terms of stability and mechanical robustness.

Q.10 Given D(q) and V(q), we can include g(q) as well to incorporate for gravity effects and then we can complete the dynamics equation by writing $D(q)\ddot{q} + V(q)\dot{q} + G(q) = \tau$

Where T is the torque required at each joint. If we are given $q_{desired}$, We can use that in the equation as well.

Q.11 ————

Q.12 Code provided, the link lengths are assumed 2,3 and EE position as [3,4,5].

Q.13 Simple trigonometry is used to calculate the inverse kinematics of the SCARA Bot. code is attached.

Q.14 To compute the joint velocities from end-effector Cartesian velocities, we can utilize the relationship between the Jacobian matrix and the velocities of the end-effector and joints. The Jacobian relates the joint velocities to the end-effector velocities via the equation:

$$X' = J(q)q'$$

Note- In the code provided , i have not attached the subroutine required to calculate the Jacobian as it has been previously calculated. In the **Q14.py , i have only** used the

resulting jacobian from **Q3.py** and inverted it to multiply with ***X'*** vector (End effector velocity) to get ***q'*** vector.

Q.17 The inverse kinematics for a spherical wrist involves determining the joint angles of the wrist to achieve a desired end-effector orientation (rotation matrix). The method involves solving for three angles representing the wrist's orientation about the X, Y, and Z axes. The code **Q17.py** returns ——> θ_1= Rotation about **Z axis ,** θ_2 = Rotation about **Y axis ,** θ_3 = Rotation about **X axis**

Q.18 Code provided combines the subroutines for calculating forward kinematics, manipulator jacobians , it takes the dH Parameters as input. Also the code only works for PPP configuration and doesn't ask for joint type explicitly.

Q.19 Question Unclear. The inverse kinematics of a 3d printer is fairly straightforward. Like if the End Effector [XYZ] are [a,b,c] then joint variables are also [a,b,c] is some way.