FR- Home task (Differential kinematics)

Task:

1) Derive the jacobian matrix for your robot model..
2) Implement this for your robot model from the previous assignment.
3) Use the Jacobian matrix to check for singularities.

References:

It is recommended to refer to these references for information about implementation of Jacobian and singularity analysis.

1) B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, "Robotics: Modelling, Planning and Control", 3rd Edition, Springer, 2009

2) Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, Robot Dynamics and Control, Second Edition, John Wiley & Sons, Inc. 2008

Submission:

You need to submit code and report: .

1) Function Jacobian(frames):
    a) Frames: list of frames from forward kinematics solution
    b) Output: Jacobian matrix
2) Function Check_singular(jacobian):
    a) Output: a boolean value “true” if the manipulator indicates singularity and “false” if the manipulator is not in singularity.
3) Function cartesian_vel(jacobian, q_dot):
    a) Takes jacobian matrix and vector of joint velocities.
    b) Output: a vector of velocities in cartesian space.

In your report:

- A scheme of your robot model.
- Derivation of your solution.

Bonus:

You can do either of these two tasks (if you do both, you will get marks only for one of them)

Bonus task (1): (recommended for those who could implement animation of robot)

Drive the robot's end-effector from one pose to another using the inverse jacobian solution.

You have the rule for mapping joint velocity to end-effector velocity:

$$\dot{x} = J(q)\,\dot{q}$$

Assume the robot is at position X_0, you need to drive the robot from X_0 to pose X_1 with inverse jacobian solution as follows:

$$J^{-1}(q)\,(X_1 - X_0) = \dot{q}$$

Bonus task (2):

- Implement Jacobian matrix numerically (you need further reading about deriving orientation)
- Validate the solution of numerical approach with the solution you implemented in the main task (both should have the same answer)