EXPERIMENT 3: MANIPULATOR JACOBIAN

The purpose of this experiment is to study the relationship between the linear and angular velocity of the end-effector of a robot, and the velocity of the individual joints of a robot arm. The following topics will be studied in this experiment.

Topics Covered

- Linear and angular velocities of rigid bodies
- The notion of Jacobian
- Singularities
- Static forces and Jacobian

Prerequisites

- The robot has been setup and tested. See the Quick Start Guide for details.
- You have access to the User Manual.
- You are familiar with the basics of MATLAB® and SIMULINK®.

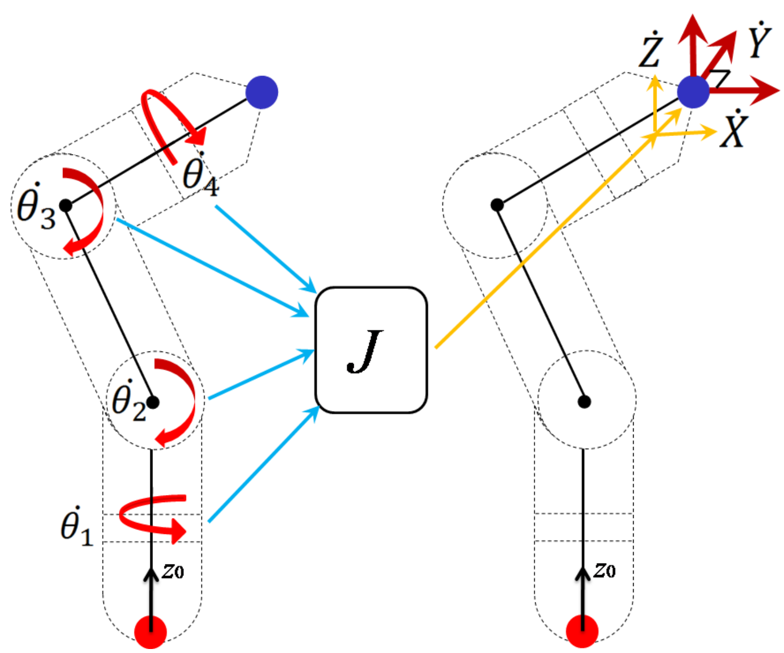

Figure 0.1: Manipulator Jacobian is used to determine the end-effector speed or forces (Cartesian Space) of the robot arm given individual joint speeds or torques (Joint Space).

1 Pre-Lab Preparations

1.1 Background

In this experiment, we study the linear and angular velocity of manipulators as well as the static forces and moments acting on manipulators. Both of these topics are closely related to a property of manipulators called *Jacobian* or *Velocity Kinematics*.

1.1.1 Linear and Angular Velocity

The linear and angular velocity of a rigid body can be achieved by differentiating position vectors. We use the following notation for the derivative of Q relative to frame B, expressed in frame B.

$$^{B}V_{Q} = \frac{d}{dt}{}^{B}Q = \lim_{\Delta t \to 0} \frac{^{B}Q(t+\Delta t) - {}^{B}Q(t)}{\Delta t} \tag{1.1}$$

To describe this velocity vector in frame A, we can write

$$^{A}(^{B}V_{Q}) = {}^{A}_{B}R\,{}^{B}V_{Q}. \tag{1.2}$$

If the origin of frame B is also linearly moving relative to frame A, we have

$$^{A}V_{Q} = {}^{A}V_{Borig} + {}^{A}_{B}R\,{}^{B}V_{Q}. \tag{1.3}$$

While linear velocity describes an attribute of a **point** in space, angular velocity describes an attribute of a rigid **body** (to which a frame is attached). Therefore, the angular velocity vector describes the rotational motion of a frame.

A notation similar to linear velocity is defined for angular velocity. Here $^{C}(^{A}\Omega_{B})$ indicates the rotation of frame B relative to frame A described in frame C.

Now, if the frame B rotates with the rotational velocity $^{A}\Omega_{B}$ relative to frame A, the motion of this vector as viewed in Frame A can be written as follows

$$^{A}V_{Q} = {}^{A}V_{Borig} + {}^{A}_{B}R\,{}^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\,{}^{B}Q \tag{1.4}$$

1.1.2 Velocity Propagation for Serial Link Manipulators and Manipulator Jacobian

The following equations can be used to propagate the linear and angular velocities for revolute-joint manipulators.

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_{i} \tag{1.5}$$
$$^{i+1}v_{i+1} = {}^{i+1}_{i}R\,({}^{i}v_{i} + {}^{i}\omega_{i} \times {}^{i}P_{i+1}) \tag{1.6}$$

Applying these equations successively from link to link, we can compute the linear and angular velocity of the end-effector frame with respect to the base frame. These equations can be written in a compact matrix format that relates the linear and angular velocities of the end-effector in Cartesian space to the joint angles of the robot arm

$$\dot{X} = \begin{bmatrix} v \\ \omega \end{bmatrix} = J\dot{q} \tag{1.7}$$

where v is the linear velocity of the end-effector frame, w is the angular velocity of the end-effector frame, $\dot{q}$ is the vector of the joint angles and matrix J is called the Jacobian matrix and can be written as follows (for revolute joint manipulators)

$$J = \begin{bmatrix} J_1 & ... & J_N \end{bmatrix}, \quad J_i = J = \begin{bmatrix} J_{vi} \\ J_{\omega i} \end{bmatrix}, \quad J_{vi} = z_{i-1} \times (O_n - O_{i-1}), \quad J_{\omega i} = z_{i-1} \tag{1.8}$$

where O_j is the centre of frame j and z_i is the joint axis related to frame i.

Alternatively, in order to achieve the linear velocity components of the Jacobian, J_{vi}, we can calculate the partial derivative of the position vector of the end-effector of the robot with respect to the joint angles. The angular velocity component of the Jacobian is simply z_{i-1}.

1.1.3 Static Forces

The Jacobian matrix J also relates the forces and moments that a manipulator exerts at its end-effector, F, to the corresponding torques required at the joints, τ, as shown below.

$$\tau = J^T F \tag{1.9}$$

This theorem can be proved using the principle of virtual work.

1.1.4 Manipulator Singularities

As discussed, the Jacobian matrix J defines a mapping between the joint velocities ($\dot{q}$) and the end-effector velocities $\dot{X}$, which is a function of the arm pose determined by joint angles q. The Jacobian defines the linear transformation $dX = Jdq$ between the differentials or infinitesimal motions. To achieve a desired motion for the end-effector, the inverse solution is required. However, in some configurations, the rank of matrix J decreases; such configurations are called singularities. Therefore, singularities represent configurations or poses from which certain directions in the space may be unattainable by the robot arm.

At singularities, bounded end-effector velocities may correspond to unbounded or infinite joint velocities. Similarly, at singularities, bounded end-effector forces and torques may correspond to unbounded or infinite joint torques. These solutions result in undesired and unsafe motion of the robot.

1.2 Pre-Lab Exercise

1. Discuss the procedure you would follow to derive the Jacobian of the 4DOF MICO robot. Derive $J_{\omega i}$ for i = 1 and 2 at DH home (where all joint angles are zero).

   Note: you are not required to go through matrix calculations for the rest of the Jacobian matrix.

2. Write a MATLAB function to receive a transformation matrix of a tool and the joint angle and calculate the Jacobian matrix. What is the Jacobian matrix at zero DH when there is no tool attached?

   Hint: you can use the available DH() function in your code to derive ${}^{i-1}_{i}T$ and multiply these matrices to derive ${}^{0}_{i}T$.

3. Describe different possibilities for singularity of the 4DOF MICO arm. What would happen in case of singularities?

2 In-Lab Exercise

2.1 Simulation

The QUARC model for this exercise is "MICO_Jacobian_Simulation.mdl" a snapshot of which shown in Figure 2.1.

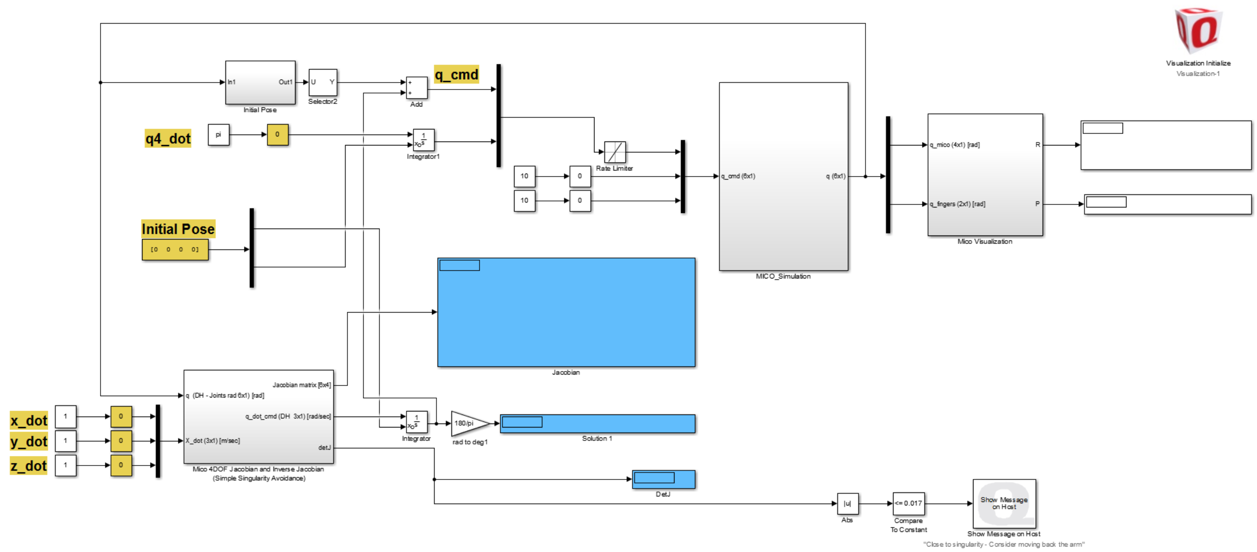

Figure 2.1: Snapshot of the controller model "MICO_Jacobian_Simulation.mdl"

Make sure that the slider gains for x_dot, y_dot, z_dot and q4_dot inputs are set to zero and the initial pose is set to [0 0 0 0]. Compile and run the model. The Quanser 3D Viewer should open, showing a visualization of the robot. Go through the following steps and answer the corresponding questions.

1. What is the Jacobian matrix at DH zero? Compare this with the output of your function in the Pre-lab section.
2. What is the meaning of the last column of the J matrix? Discuss it using the virtual robot motion and configuration.
3. Stop the model and change the initial pose to [0 $-\pi/2$ 0 0]. Run the model again. Now slowly change the x_dot input command to +0.01 using the slider gain. What do you observe? Do the same thing with the y_dot and z_dot commands and discuss your observations.
4. Describe the model. In particular describe how the joint position commands are generated from the x_dot commands.
5. Using the Jacobian when the joint angles are $q_0 = [0, -\pi/2,\ 0,\ 0]$, what would be the required torque be for the four joints of the robot to apply -5 N at its end-effector along the z axis (with no moments)? Discuss your results.

Once you feel comfortable with the above exercises using the virtual robot, go to the next section.

2.2 Experiment

The QUARC model for this exercise is "MICO_Jacobian_Experiment.mdl" the snapshot of which shown in Figure 2.2.

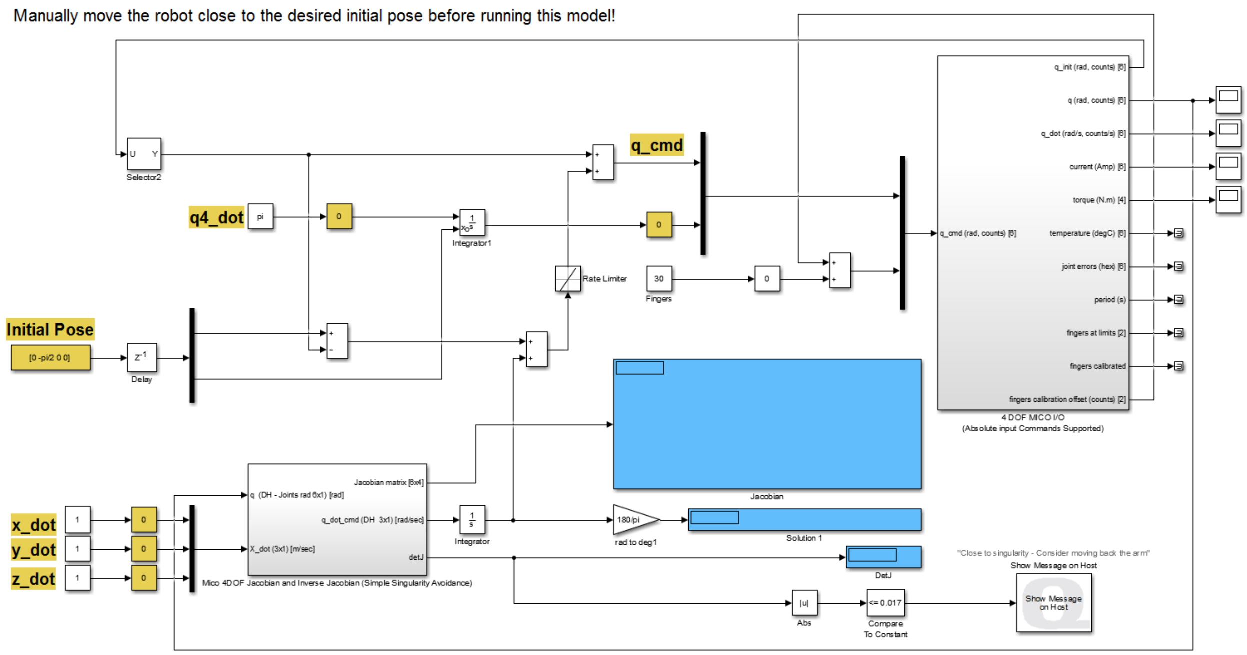

Figure 2.2: Snapshot of the controller model ”MICO_Jacobian_Experiment.mdl”

Turn off the robot and manually pose it to roughly $q = [0, -\pi/2, 0, 0]$ as was studied in the simulation section. Turn on the robot. Also make sure that the slider gains for the x_dot, y_dot, z_dot and q4_dot inputs are set to zero. Compile and run the robot, and it will go to the home pose $q = [0, -\pi/2, 0, 0]$.

Caution

Be sure to set the Ports in the 4-DOF MICO I/O block to the correct ports for your serial card. For more information, refer to the User Manual.

1. What is the Jacobian matrix of the robot in this pose? Compare it with the values from your Pre-lab and the simulation section. Explain your observations.
2. Change the x_dot command to 0.01 (1 cm/sec) by clicking and changing the corresponding slider gain.
3. Change the slider gain value back to zero after the robot moves about 10-15 cm.
4. Move the robot along y and z axis the same way you did for the x axis and stop the robot.
5. Send the robot to the home position using the manual switch (switch to home pose $q = [0, -\pi/2, 0, 0]$).
6. Find an object of a known weight (around 0.5 Kg). Using the finger control slider gains to open and close the fingers, hold the object.
7. Read the torque values (output from MICO I/O block). Compare the results with the ones you calculated in the previous section.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com