Kai Bing Goh, William Delaporte, Yi Gao & Yixiang Gao
981436, 915100, 1208446, 990803
Department of Mechanical Engineering
The University of Melbourne
VIC 3010, Australia
kaibingg@student.unimelb.edu.au
wdelaporte@student.unimelb.edu.au
yigao4@student.unimelb.edu.au
yixiang1@student.unimelb.edu.au

Robotics Systems Assignment 2 Report
Deriving Jacobian Matrix and Evaluating Torque Required

MCEN90028 : Dr. Justin Fong & Dr. Tomislav Bacek
Due –*Friday, April 15, 2022*

Table of Content

1 Introduction

The aim of this report is to document the process of developing building blocks that are necessary for motion control of a robotic manipulator as outlined in the previous report. As such, the derivation of DH table, link length selection via workspace plots and inverse kinematics is not the focus of this report, however it will be presented where necessary for the ease of understanding.

Objectives

To ensure that the provided motors will not experience torque exceeding its maximum torque stated in the data sheets, and to prevent overloading that could lead to burnt motors, procedures to predict how torque will be applied on the system should be carried out in the design phase of the manipulator before any manufacture should take place. Hence, this leads to the main objectives of this report as listed below:

- Derive a Jacobian matrix $J(q)$ for the robot which maps the joint velocities into end-effector velocity by expressing it in a specific coordinate frame.
- Evaluate if joint torques required to lift the inertia of the robot alongside the load within the entire workspace are achievable and falls within the motors' torque range.

2 Methodology

2.1 Deriving the Jacobian Matrix Relating the Joint Velocity to End-Effector Velocity

2.1.1 Robot Kinematics

As discussed in the last report, the robot consists of two main arm sections, mounted on a rotating base, and with a grabber attached to the end. A diagram of the robot labelled with the different link reference frames is given in figure 1.

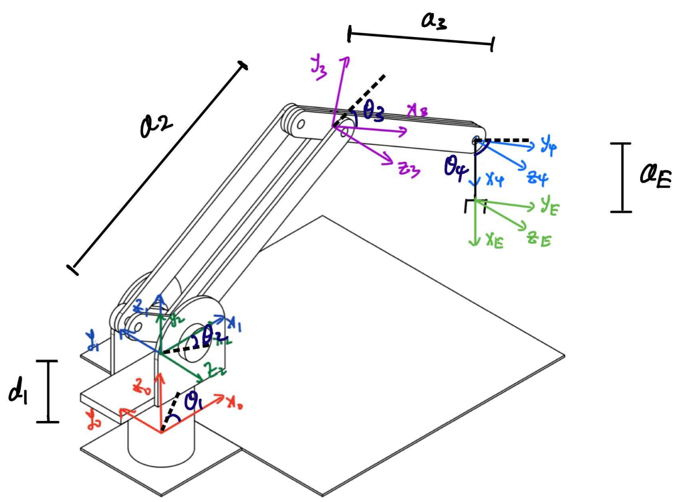

Figure 1: Schematic Diagram of Robot

a_{i-1}	α_{i-1}	d_i	θ_i
0	0	d_1	Q_1
0	90	0	$90 - Q_2$
a_2	0	0	$-Q_3$
a_3	0	0	$-Q_4$
a_E	0	0	0

Table 1: DH table with parameters

Performing Denavit-Hartenberg analysis on this structure results in table 1, which can then be used to derive the transformation matrix from the inertial frame to the end-effector frame, given in equation 1.

$$\begin{pmatrix} \sigma_2 & \cos(Q_4)\,\sigma_8 + \sin(Q_4)\,\sigma_7 & \sin(Q_1) & a_3\,\sigma_7 + a_E\,\sigma_2 + a_2\cos(Q_1)\,\sigma_9 \\ \sigma_1 & \cos(Q_4)\,\sigma_6 + \sin(Q_4)\,\sigma_5 & -\cos(Q_1) & a_3\,\sigma_5 + a_E\,\sigma_1 + a_2\,\sigma_9\sin(Q_1) \\ -\cos(Q_4)\,\sigma_3 - \sin(Q_4)\,\sigma_4 & \cos(Q_4)\,\sigma_4 - \sin(Q_4)\,\sigma_3 & 0 & d_1 - a_2\,\sigma_{10} - a_E\left(\cos(Q_4)\,\sigma_3 + \sin(Q_4)\,\sigma_4\right) - a_3\,\sigma_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where

$\sigma_1 = \cos(Q_4)\,\sigma_5 - \sin(Q_4)\,\sigma_6$

$\sigma_2 = \cos(Q_4)\,\sigma_7 - \sin(Q_4)\,\sigma_8$

$\sigma_3 = \cos(Q_3)\,\sigma_{10} + \sigma_9\sin(Q_3)$

$\sigma_4 = \cos(Q_3)\,\sigma_9 - \sin(Q_3)\,\sigma_{10}$

$\sigma_5 = \cos(Q_3)\,\sigma_9\sin(Q_1) - \sin(Q_1)\,\sin(Q_3)\,\sigma_{10}$

$\sigma_6 = \cos(Q_3)\,\sin(Q_1)\,\sigma_{10} + \sigma_9\sin(Q_1)\,\sin(Q_3)$

$\sigma_7 = \cos(Q_1)\,\cos(Q_3)\,\sigma_9 - \cos(Q_1)\,\sin(Q_3)\,\sigma_{10}$

$\sigma_8 = \cos(Q_1)\,\cos(Q_3)\,\sigma_{10} + \cos(Q_1)\,\sigma_9\sin(Q_3)$

$\sigma_9 = \cos\left(Q_2 - \frac{\pi}{2}\right)$

$\sigma_{10} = \sin\left(Q_2 - \frac{\pi}{2}\right)$

2.1.2 Derivation of Jacobian

In order to derive the Jacobian matrix relating end-effector velocity to joint velocity, it was decided to work within frame 1, the frame attached to the rotating base of the robotic arm. This was in order to simplify the analysis, and since the angle of the base has no effect on the torque requirements of the arm, it was deemed safe to ignore.

The Jacobian consists of two components - the translational component and the angular component. The elements of the translational component are found by taking the cross product of the axis of rotation for each joint and the displacement of that joint. The angular component is simply composed of the rotational axes, expressed in the frame being used for the analysis. Equation 2 shows the form this takes for the robot.

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}, \; J_v = \begin{bmatrix} \hat{z_1} \times \vec{p_{1E}} & \hat{z_2} \times \vec{p_{2E}} & \hat{z_3} \times \vec{p_{3E}} \end{bmatrix}, \; J_\omega = \begin{bmatrix} \hat{z_1} & \hat{z_2} & \hat{z_3} \end{bmatrix} \tag{2}$$

Here, $\hat{z_i}$ is the unit vector for the axis of rotation of joint i and $\vec{p_{iE}}$ is the displacement vector of joint i relative to joint 1, all expressed in frame 1. $\hat{z_i}$ was expressed in frame 1 using the appropriate series of rotation matrices, while $\vec{p_{iE}}$ was derived geometrically in frame 1 for each joint. Using MATLAB to perform the calculation results in matrix given in equation 3.

$$J = \begin{bmatrix} 0 & \frac{27\,\sin\left(Q_2-\frac{\pi}{2}\right)}{100} - \frac{23\,\sin\left(Q_3-Q_2+\frac{\pi}{2}\right)}{100} & -\frac{23\,\sin\left(Q_3-Q_2+\frac{\pi}{2}\right)}{100} \\ \sigma_1 & 0 & 0 \\ 0 & \sigma_1 & \frac{23\,\cos\left(Q_3-Q_2+\frac{\pi}{2}\right)}{100} \\ 0 & 0 & 0 \\ 0 & -1 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{3}$$

where

$$\sigma_1 = \frac{23\,\cos\left(Q_3 - Q_2 + \frac{\pi}{2}\right)}{100} + \frac{27\,\cos\left(Q_2 - \frac{\pi}{2}\right)}{100}$$

2.1.3 Jacobian Matrix Validation

MATLAB provides a function *jacobian()* that can directly generate the Jacobian Matrix from the position matrix [X;Y;Z]. This function will do the partial derivative for $[Q_1; Q_2; Q_3; Q_4]$. When we calculated the Jacobian, we assume the gripper to be always perpendicular to the ground, so we ignore Q_4 and add the weight of gripper and chess piece as a single extra force. Equation 4 shows the result of directly deriving the position matrix.

$$\begin{pmatrix} 0 & -\frac{23\sin(\theta_2+\theta_3)}{100} - \frac{27\sin(\theta_2)}{100} & -\frac{23\sin(\theta_2+\theta_3)}{100} \\ \sigma_1 & 0 & 0 \\ 0 & \sigma_1 & \frac{23\cos(\theta_2+\theta_3)}{100} \end{pmatrix} \tag{4}$$

where

$$\sigma_1 = \frac{23\cos(\theta_2+\theta_3)}{100} + \frac{27\cos(\theta_2)}{100}$$

From the DH table, $\theta_1 = Q_1,\ \theta_2 = pi/2 - Q2,\ \theta_3 = -Q_3,\ \theta_4 = -Q_4$. By substituting the values $([Q_1, Q_2, Q_3] = [0, pi/2 - pi/3, pi/3])$, the results that are obtained from both methods are verified to be the same, which is shown in equation 5. Therefore, the Jacobian is correct.

$$\begin{pmatrix} 0 & -\frac{27\sqrt{3}}{200} & 0 \\ \frac{73}{200} & 0 & 0 \\ 0 & \frac{73}{200} & \frac{23}{100} \end{pmatrix} \tag{5}$$

2.2 Evaluating the joint torque required to overcome gravity associated with the task

2.2.1 Derivation of Jacobian Matrices for each Center of Mass

The derivation process is similar as deriving Jacobian matrix for end-effector, by using the definition of Jacobian Matrix as shown in equation 6. The difference is that we need to take every gravity center into account.

$$J_{ci} = \begin{bmatrix} J_{vci} \\ J_{\omega ci} \end{bmatrix}, i = 1, 2, 3 \tag{6}$$

Equations below are used for deriving the Jacobian matrices for each center of mass.

$$J_{vc1} = \begin{bmatrix} \hat{z_1} \times \vec{p_{1c1}} & 0_{3\times1} & 0_{3\times1} \end{bmatrix},\ J_{\omega} = \begin{bmatrix} \hat{z_1} & 0_{3\times1} & 0_{3\times1} \end{bmatrix} \tag{7}$$

$$J_{vc2} = \begin{bmatrix} \hat{z_1} \times \vec{p_{1c2}} & \hat{z_2} \times \vec{p_{2_{c2}}} & 0_{3\times1} \end{bmatrix},\ J_{\omega} = \begin{bmatrix} \hat{z_1} & \hat{z_2} & 0_{3\times1} \end{bmatrix} \tag{8}$$

$$J_{vc3} = \begin{bmatrix} \hat{z_1} \times \vec{p_{1c3}} & \hat{z_2} \times \vec{p_{2_{c3}}} & \hat{z_3} \times \vec{p_{3_{c3}}} \end{bmatrix},\ J_{\omega} = \begin{bmatrix} \hat{z_1} & \hat{z_2} & \hat{z_3} \end{bmatrix} \tag{9}$$

In the equations, $\hat{z_i}$ is the joint axis and $\vec{p_{kci}}$ are vectors from joint k pointing to the centre of gravity of link i. In our design, we have three links and currently assume the centres of mass are in the middle of each linkage. (Further adjustment may need if we get more precise positions of the centres of mass) Thus, the vectors are as shown below:

$$\begin{aligned} \vec{p_{1_{c11}}} &= \begin{bmatrix} 0 & 0 & -l_1/2 \end{bmatrix} \\ \vec{p_{2_{c22}}} &= \begin{bmatrix} l_2/2 & 0 & 0 \end{bmatrix} \\ \vec{p_{3_{c33}}} &= \begin{bmatrix} l_3/2 & 0 & 0 \end{bmatrix} \end{aligned} \tag{10}$$

where l_1, l_2, l_3 are length of links.
With the help of MATLAB, we get the Jacobian matrices as follows:

$$J_{c1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{11}$$

$$J_{c2} = \begin{pmatrix} 0 & -\frac{27\,\sin\left(Q_2-\frac{\pi}{2}\right)}{200} & 0 \\ \frac{27\,\cos\left(Q_2-\frac{\pi}{2}\right)}{200} & 0 & 0 \\ 0 & \frac{27\,\cos\left(Q_2-\frac{\pi}{2}\right)}{200} & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{12}$$

$$J_{c3} = \begin{pmatrix} 0 & \sigma_3 - \sigma_2 - \frac{27\,\sigma_7}{100} & \sigma_3 - \sigma_2 \\ \sigma_1 & 0 & 0 \\ 0 & \sigma_1 & \sigma_4 + \sigma_5 \\ 0 & 0 & 0 \\ 0 & -1 & -1 \\ 1 & 0 & 0 \end{pmatrix}$$

where

$$\begin{aligned} \sigma_1 &= \tfrac{27\,\sigma_6}{100} + \sigma_4 + \sigma_5 \\ \sigma_2 &= \tfrac{23\,\cos(Q_3)\,\sigma_7}{200} \\ \sigma_3 &= \tfrac{23\,\sigma_6\,\sin(Q_3)}{200} \\ \sigma_4 &= \tfrac{23\,\cos(Q_3)\,\sigma_6}{200} \\ \sigma_5 &= \tfrac{23\,\sin(Q_3)\,\sigma_7}{200} \\ \sigma_6 &= \cos\left(Q_2 - \tfrac{\pi}{2}\right) \\ \sigma_7 &= \sin\left(Q_2 - \tfrac{\pi}{2}\right) \end{aligned} \tag{13}$$

2.2.2 Calculate Maximum Torque Required Across Workspace for each Joint

Weight estimation

Each link in our design has two pieces of Acrylic plate, where the first and second link has length of 27cm or 23 cm respectively, both with a width of 3cm and thickness of 6mm. Meanwhile, the density of acrylic is found to be 1.18 g/cm^3.

Because of the design, where we placed all of the motors at the base, the weight on link 1 is the heaviest, but that will not affect the torque on that joint as nothing is acting against the rotation except for friction, which can be negligible with appropriate use of bearings that will be installed in our robot. Secondly, link 3 is controlled by a four-bar parallel linkage. To simplify the analysis, we consider the entire body of the parallel linkage to be a part of link 2, whereas the remaining link will be the body of link 3 as shown in figure 2.

As we rotate link 3, the geometry of the parallel linkages will cause the center of mass of link 2 to shift. The true centre of mass is located in the middle of the parallelogram, can be expressed in terms of the joint angles. However, due to limited resources, a more simplified approach is taken where we assume all of its weight acts on the center of link 2. To compensate for this approximation, it is assumed that the entire weight of the parallel linkage will have a worst-case weight of 3.5 times that of a single arm section, rather than the 2 sections that actually make it up.

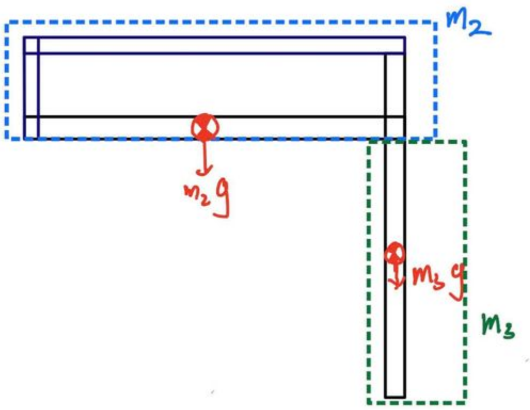

Figure 2: mass position

Before calculating the forces and torques, the mass of each component can be summarised as below:

- Mass of link 1: m_1=0.5kg
- Mass of link 2: m_2=3.5× 0.0489*kg
- Mass of link 3: m_3=0.0414× 2kg
- Mass of gripper: m_4=0.0489× 2 + 0.012kg
- Mass of chess: m_{chess}=0.004 kg

Forces and torques

In order to evaluate if the joint torque required within the to reach the entire workspace is within the torque range produced by the motors in a quasi-static manner, and for a robotic manipulator that is non-redundant ($n = m$), we use the relationship:

$$\tau = J^T \mathcal{F} \tag{14}$$

The force acting at the end-effector due to the mass of the chess piece and end-effector can be represented in frame 1 as shown in equation 15. This will later be used in equation 17.

$$\mathcal{F} = \begin{bmatrix} 0 \\ 0 \\ -(m_4 + m\_chess) \times g \end{bmatrix} \tag{15}$$

Now, we can find the resulting torques acting on all joints in order to overcome gravity (eq 16) and external forces (eq 17) using the equations below.

$$\tau_g = J_{vc1}^T \times \begin{bmatrix} 0 \\ 0 \\ -(m_1) \times g \end{bmatrix} + J_{vc2}^T \times \begin{bmatrix} 0 \\ 0 \\ -(m_2) \times g \end{bmatrix} + J_{vc3}^T \times \begin{bmatrix} 0 \\ 0 \\ -(m_3) \times g \end{bmatrix} \tag{16}$$

$$\tau_f = J_v^T \times \begin{bmatrix} 0 \\ 0 \\ -(m_4 + m\_chess) \times g \end{bmatrix} \tag{17}$$

Combining the torques that will required for overcoming both the gravity and external forces, the total torque τ_{total} can be obtained.

$$\tau_{total} = \tau_g + \tau_f \tag{18}$$

Maximum Torques for the Joints Across the Entire Workspace

With the gravity components and the Jacobian matrices, we can calculate resultant torques on each individual joint. Traversing the workspace by iterating through the joint angles, we can calculate all the torques required to keep static equilibrium for the respective angles by using MATLAB. Figure 3 below shows how the iteration is performed, where the maximum torque can be found by choosing the maximum value of all the possible torque results for each joint.

```
i = 1;
for q2 = 0:pi/18:pi/2
    for q3 = 0:pi/18:deg2rad(170)
        aa = subs(total_tau,[Q_1 Q_2 Q_3],[0 q2 q3]);
        Tau_1(i) = double(aa(1));
        Tau_2(i) = double(aa(2));
        Tau_3(i) = double(aa(3));
        i = i + 1;
    end
end
Tau_1_max = max(abs(Tau_1))
Tau_2_max = max(abs(Tau_2))
Tau_3_max = max(abs(Tau_3))
```

Figure 3: MATLAB Code of Iteration for Torque

As a result, the maximum torques for joints 1, 2, and 3 are found to be $[0; 1.0976; 0.3502]Nm$ respectively. These values are all safely smaller than the stall torque of $16.5kgcm \approx 1.62Nm$ for the Feetech SCS15 motors.

3 Conclusion

In summary, this report has outlined the derivation of the Jacobian matrices and validated that the results are correct. Moreover, procedures for estimations of the weights and center of mass have been clearly presented, including necessary assumptions and countermeasures for simplifying a complicated mathematical problem. This information is then used to calculate the possible torques required for every angle and to identify the maximum value. It is found that the design has satisfied the torque output limitations of the provided motors. As such, future improvements can be made in obtaining the expression for center of mass of link 2, and actual mass of the acrylic plates can be used in the calculations when the parts are manufactured for a more accurate calculation of the maximum torques.

Moving on, the generation of trajectory of our robot will be explored, where an algorithm will be constructed based on the initial and final end-effector pose and velocities, but also taking into account of new constraints that will be faced.