7CCEMROB **Robotic Systems**
Coursework 1

Robot Kinematics Analysis

Date: 28/02/2023

Abstract

Robotics is a field that is rapidly evolving and expanding, with numerous applications in various industries, including manufacturing, healthcare, and agriculture. This report focuses on three tasks that involve analysing the kinematics of robotic systems. The first two tasks involve assigning frames to robot links, constructing the Denavit-Hartenberg (D-H)[1] table for the given data, computing the forward and inverse kinematics of the given robots, finding the Jacobian matrices, and checking for singularities. The third task involves creating a robot simulation in MuJoCo and analysing its kinematics. The report provides a detailed explanation of each of these tasks, along with the methodology and results obtained. The report also discusses the significance of kinematics analysis in robotics and its various applications in real-world scenarios. Overall, this report provides valuable insights into the kinematics analysis of robotic systems and highlights the potential of this field for future research and development.

Table of contents

List of Figures

1. Introduction

Robotics is a rapidly growing field that combines computer science, engineering, and mathematics to develop intelligent machines that can perform tasks autonomously. Robotics has a wide range of applications in various industries, including manufacturing, healthcare, and agriculture. The development of robotics has led to the creation of advanced technologies that have improved productivity, efficiency, and safety in various industries. There is what can be considered a sub-field of robotics, called "robotic kinematic analysis", where the interest is the kinematic geometry of the types of multi-degree-of-freedom mechanisms used in robotics devices such as mechanical arms and legs.

The problem statement for this report is to analyse the given robots in terms of their forward kinematics, inverse kinematics, and Jacobian matrices. Forward kinematics involves calculating the position and orientation of the robot's end effector given the joint angles. Inverse kinematics involves calculating the joint angles required to position the end effector at a desired location. Jacobian matrices are used to calculate the linear and angular velocities of the end effector given the joint velocities. Additionally, a model of the robot will be simulated on MuJoCo (Multi-Joint dynamics with Contact). It is a physics engine used to simulate intricate dynamic systems and can also be used for modelling and managing the dynamics of physical systems, especially those involving interacting articulated and deformable bodies. The simulation will be analysed to determine the kinematic properties of the robot and how they relate to the physical properties of the robot. MuJoCo has a wide array of features, that will be utilised, of those which will be of great use to for the task is generalized coordinates combined with modern contact dynamics.

The results of this report are expected to provide insights into the kinematics analysis of robotic systems, including the importance of forward kinematics, inverse kinematics, and Jacobian matrices in the operation of robots. The simulation on MuJoCo is expected to provide a visual representation of the movement of the robot, which can be used to analyse the behaviour of the robot under different conditions. The results obtained will be useful for the development of more efficient and effective robotic systems for various industries.

2. Literature Review

The assignment of frames is a fundamental concept in the field of robotics, as it allows for the description of the position and orientation of each joint of a robotic system relative to a fixed reference frame. This is achieved by assigning a coordinate frame to each joint of the robot, and specifying the relationship between these frames using the Denavit-Hartenberg (DH) notation. The DH notation is a widely used convention that simplifies the mathematical representation of the kinematic structure of robotic systems. This notation involves assigning four parameters, namely the link length, the link twist, the link offset, and the joint angle, to each joint of the robot. By using the DH notation, the forward kinematics of the robotic system can be computed efficiently.

The main source of reference for this report has been the course reference material and the book by *J. Craig "Introduction to Robotics: Mechanics and Control."* [3] In the text, Craig uses the modified DH convention to assign frames to robot links, whereas the course material follows the standard or classical DH convention, the differences of which are mentioned in the appendix.

In their paper, Constantine et al. [4] analysed an industrial robot using two different methods, i.e. using the DH notation as we will do in this report, but also using the Robotic Toolbox ® developed by Peter Corke. The forward kinematics of a robotic system involves calculating the position and orientation of the robot's end effector given the joint angles. This is achieved by using the DH parameters to transform each joint frame to the end effector frame. The computation of the forward kinematics is crucial for controlling the motion of the robot, as it allows for the determination of the position and orientation of the robot's end effector. The inverse kinematics of a robotic system involves calculating the joint angles required to position the end effector at a desired location. This is achieved by using the DH parameters to transform the end effector frame to each joint frame, and then solving for the joint angles. The computation of the inverse kinematics is crucial for controlling the motion of the robot, as it allows for the determination of the joint angles required to achieve a desired end effector position and orientation.

3. Methodologies

Assignment of frames: The assignment of frames is a critical step in the kinematic analysis of robotic systems. It involves defining a coordinate system or a frame for each joint of the robot. The choice of frames is important as it determines the accuracy of the kinematic analysis. The frames are usually assigned based on the physical structure of the robot, and each frame is typically located at the joint axis. The frames can be defined using CAD software or by manual measurement. In this report, the frames are assumed to be already assigned to the given robots.

Denavit-Hartenberg Notation: The Denavit-Hartenberg (DH) notation is a standard method for describing the kinematic structure of robotic systems. This notation involves assigning four parameters, namely the link length, the link twist, the link offset, and the joint angle, to each joint of the robot. These parameters are used to construct a kinematic chain that relates the position and orientation of the end-effector of the robot to the joint angles. In this report, we use the modified DH notation, which involves assigning the x-axis of each joint frame along the joint axis and the z-axis perpendicular to both the x-axis of the current joint and the x-axis of the previous joint.

Forward Kinematics: Forward kinematics involves determining the position and orientation of the end-effector of the robot given the joint angles. The forward kinematics equations are derived from the DH parameters and the transformation matrices between the adjacent frames. The forward kinematics equations are used to determine the position and orientation of the robot's end-effector, which is essential for controlling the robot's motion. In this report, we compute the forward kinematics using the modified DH notation.

Inverse Kinematics: Inverse kinematics involves determining the joint angles required to achieve a desired position and orientation of the robot's end-effector. Inverse kinematics is important for controlling the motion of the robot and is used in applications such as trajectory planning and motion control. Inverse kinematics can be computed using numerical methods such as Newton-Raphson or gradient descent algorithms. This report uses an analytical approach to derive the inverse kinematics equations for the given robots.

Jacobian Matrices: The Jacobian matrix is a fundamental tool in robotics and is used to relate the velocities of the robot's joints to the velocity of the end-effector. The Jacobian matrix is computed by differentiating the forward kinematics equations concerning the joint angles. The Jacobian matrix is important for applications such as motion planning and control. In this report, we compute the Jacobian matrices for the given robots and analyse their singularities.

Singularities: A singularity is a configuration of the robot where the Jacobian matrix becomes singular, which means that the robot's motion cannot be controlled. This means it is a configuration in which the robot loses its ability to move in certain directions or along certain axes. In other words, the robot becomes locked or stuck and its movement is restricted or limited. Singularities can cause problems in motion planning and control and can limit the performance of the robot. In this report, we analyse the singularities of the given robots and discuss their implications for motion planning and control.

4. Task 1

Consider the robot described by the D-H table below:

i	θi	di	$ai{-}1$	$\alpha i{-}1$
1	0	$q1$	0	0
2	$q2$	0	0	-90
3	0	$q3$	0	0

4.1. Solution:

In this task, we assume that the Denavit – Hartenberg table is described as per the classical convention.

1) The sketch of the robot is shown below:

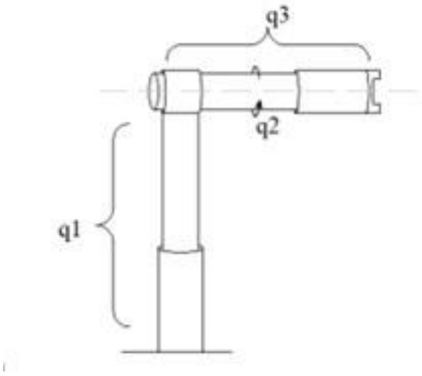

Figure1: Sketch of Robot

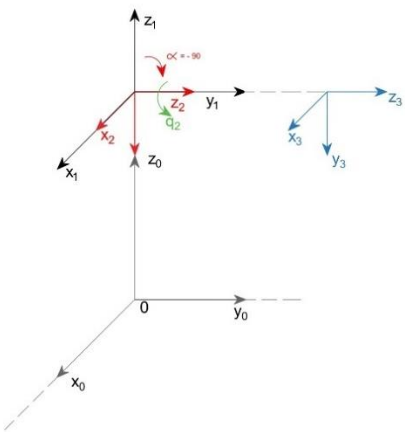

Figure 2: Frame Coordinate Systems

2) To compute the forward Kinematics

We can solve it using the General formula

$$^{i-1}T_i = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a.cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a.sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let cos(q1) = c_1, sin(q1) = s_1, cos(q2) = c_2, sin(q2) = s_2, cos(q3) = c_3, sin(q3) = s_3

$$^{b}T_{ee} = {}^{0}T_1\ {}^{1}T_2\ {}^{2}T_3$$

$$^{0}T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}T_2 = \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ s_2 & 0 & c_2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{2}T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{b}T_{ee} = {}^{0-}T_3 = \begin{bmatrix} c_2 & 0 & -s_2 & -q3.s_2 \\ s_2 & 0 & c_2 & q3.c_2 \\ 0 & -1 & 0 & q1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We know that

$$^{b}T_{ee} = \begin{bmatrix} ^{b}R_{ee} & ^{b}t_{ee} \\ 0 & 1 \end{bmatrix}, \text{ where}$$

$^{b}R_{ee}$ *is the Rotation Matrix from base to end effector and* $^{b}t_{ee}$ *is the translation from base to end effector.*

Therefore $^{b}t_{ee} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -q3.s_2 \\ q3.c_2 \\ q1 \end{bmatrix}$ is the translation from the base to the end effector.

3) <u>Inverse Kinematics</u>

In order to find the robot joint values, we consider the final position of the end effector as calculated above.

x = $-q3.s_2$, y= $q3.c_2$, z = $q1$

To find q3: Consider $x^2 + y^2$

$$x^2 = (-q3.s_2)^2$$
$$y^2 = (q3.c_2)^2$$
$$x^2 + y^2 = (-q3.s_2)^2 + (q3.c_2)^2$$
$$x^2 + y^2 = (q3)^2({s_2}^2 + {c_2}^2) = q3^2$$

Therefore, q3 = $\pm\sqrt{x^2 + y^2}$

If $x^2 + y^2 = 0$, q3 would have one solution i.e. q3=0 and q2 would have infinite number of solutions

If $x^2 + y^2 \neq 0$, q3 would have 2 solutions q3 = $\sqrt{x^2 + y^2}$ and q3 = $-\sqrt{x^2 + y^2}$

sin(q2) = $-\frac{a}{q3}$, cos(q2) = $\frac{b}{q3}$

q2 = $arctan2(sin(q2), cos(q2))$ = $arctan2(-\frac{a}{q3}, \frac{b}{q3})$

4) <u>Jacobian Matrix:</u>

$$\frac{df}{dq} = J(q) = \begin{bmatrix} \frac{dx1}{dq1} & \frac{dx1}{dq2} & \cdots & \frac{dx1}{dqn} \\ \frac{dx2}{dq1} & \frac{dx2}{dq2} & \cdots & \frac{dx2}{dqn} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dxm}{dq1} & \frac{dxm}{dq2} & \cdots & \frac{dxm}{dqn} \end{bmatrix} = \begin{bmatrix} \frac{dx}{dq1} & \frac{dx}{dq2} & \frac{dx}{dq3} \\ \frac{dy}{dq1} & \frac{dy}{dq2} & \frac{dy}{dq3} \\ \frac{dz}{dq1} & \frac{dz}{dq2} & \frac{dz}{dq3} \end{bmatrix} = \begin{bmatrix} 0 & -q3.c_2 & -s_2 \\ 0 & q3.s_2 & c_2 \\ 1 & 0 & 0 \end{bmatrix}$$

In order to obtain singularities, $|J| = 0$

$$|J| = -c_2{}^2.q3 - s_2{}^2.q3 = -q3(c_2{}^2 + s_2{}^2) = -q3$$

Hence, to obtain a singularity q3 = 0

4.2. Result:

As can be seen from the above analysis, the robot described by the DH table comprises of two prismatic joints and one revolute joint. By using forward and inverse kinematics, we can obtain the values for the joints corresponding to the final position of the end effector. From the calculations, we can conclude that if the value of q3 would be zero, the robot, would be confined to a single position and would be able to move, i.e., a singularity would be obtained.

5. Task 2

5.1. Solution

1) Assignment of Frames and D-H Table:

Task 2 has been solved using both the Classical DH method and the modified DH method

i) Using Classical Method

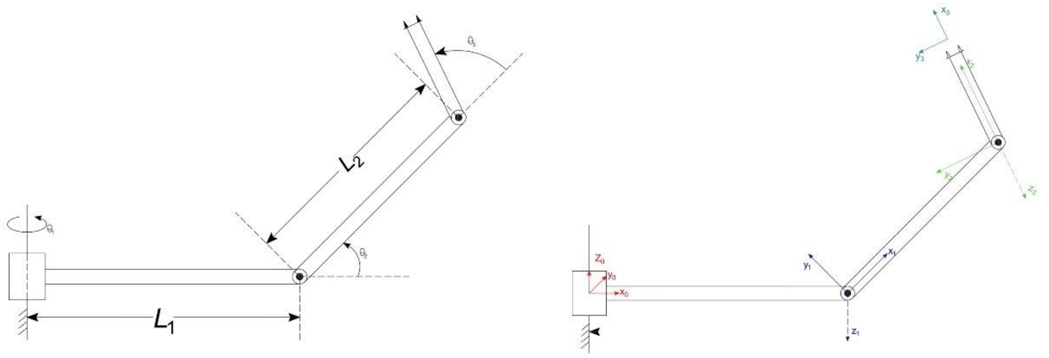

Figure 3: Assignment of Frames as per classical D-H convention

i	θi	di	a_i	α_i
1	θ_1	0	L_1	90º
2	θ_2	0	L_2	0
3	θ_3	0	0	0

2) Forward Kinematics

$${}^bT_w = {}^0T_3 = {}^0T_1\ {}^1T_2\ {}^2T_3$$

$$^{0}T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & s_1 & L_1.c_1 \\ s_1 & 0 & -c_1 & L_1.s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{1}T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2 & -s_2 & 0 & L_2.c_2 \\ s_2 & c_2 & 0 & L_2.s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{2}T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{b}T_w = \begin{bmatrix} c_1 & 0 & s_1 & L_1.c_1 \\ s_1 & 0 & -c_1 & L_1.s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & L_2.c_2 \\ s_2 & c_2 & 0 & L_2.s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{b}T_w = \begin{bmatrix} c_1.c_{23} & -c_1.s_{23} & s_1 & c_1.(L_2.c_2 + L_1) \\ s_1.c_{23} & -s_1.s_{23} & -c_1 & s_1.(L_2.c_2 + L_1) \\ s_{23} & c_{23} & 0 & L_2.s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ii) Using the Modified DH method

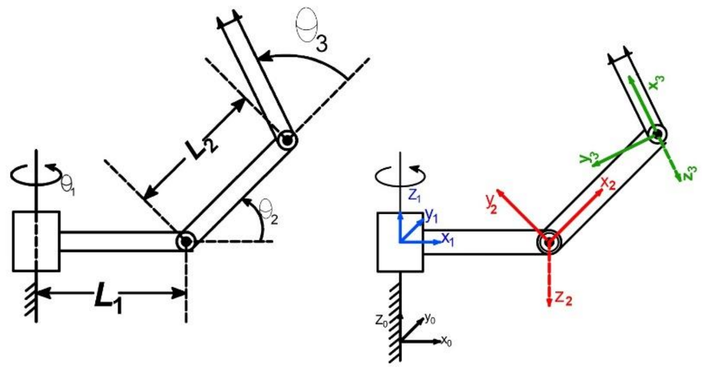

Figure 4: Assignment of Frames as per modified D-H convention

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	θ_1	0	L_1	90º
2	θ_2	0	L_2	0
3	θ_3	0	0	0

The General Transformation Matrix is:

$$^{i-1}T_i = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) & 0 & a_{i-1} \\ sin(\theta_i)cos(\alpha_{i-1}) & cos(\theta_i)cos(\alpha_{i-1}) & -sin(\alpha_{i-1}) & -d_i.sin(\alpha_{i-1}) \\ sin(\theta_i)sin(\alpha_{i-1}) & cos(\theta_i)sin(\alpha_i) & cos(\alpha_{i-1}) & d_i.cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^1T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^2T_3 = \begin{bmatrix} 1 & 0 & 0 & L_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^bT_w = \ ^0T_3 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^bT_w = \begin{bmatrix} c_1.c_{23} & -c_1.s_{23} & s_1 & c_1.(L_2.c_2 + L_1) \\ s_1.c_{23} & -s_1.s_{23} & -c_1 & s_1.(L_2.c_2 + L_1) \\ s_{23} & c_{23} & 0 & L_2.s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It is noticed that the Transformation Matrix obtained in both the methods is the same.

3) <u>Inverse Kinematics</u>

$^bT_w = {}^0T_1 \ {}^1T_2 \ {}^2T_3$
$^bT_w = \ ^0T_3$

$$^0T_3 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^0T_3 = \begin{bmatrix} c_1.c_{23} & -c_1.s_{23} & s_1 & c_1.(L_2.c_2 + L_1) \\ s_1.c_{23} & -s_1.s_{23} & -c_1 & s_1.(L_2.c_2 + L_1) \\ s_{23} & c_{23} & 0 & L_2.s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For θ_1
We equate elements (1,3) and (2,3): $s_1 = R_{13}$ and $-c_1 = R_{23}$

$\therefore \ \theta_1 = arctan2(R_{13}, R_{23})$
If both $R_{13} = 0 \ and \ R_{23} = 0, then \ \theta_1 \ has \ no \ feasible \ solution$

For θ_2:
We equate elements (1,4): $P_x = c_1(c_2L_2 + L_1)$
We equate elements (2,4): $P_y = s_1(c_2L_2 + L_1)$

If $c_1 \neq 0$ then $c_2 = \frac{1}{L_2}(\frac{P_x}{c_1} - L_1)$

Else $c_2 = \frac{1}{L_2}(\frac{P_y}{s_1} - L_1)$

Equating elements (3,4): $P_z = S_2 L_2$

$\therefore \theta_2 = atan2(\frac{P_z}{L_2}, c_2)$

For θ_3:
Equating elements (3,1): $S_{23} = R_{31}$
Equating elements (3,2): $C_{23} = R_{32}$

$\therefore\ \theta_3 = arctan2(R_{31}, R_{32}) - \theta_2$

4) Jacobian Matrix
The Jacobian of the manipulator can be derived, as shown in the previous part, there is:

$$^0T_3 = \begin{bmatrix} c_1.c_{23} & -c_1.s_{23} & s_1 & c_1.(L_2.c_2 + L_1) \\ s_1.c_{23} & -s_1.s_{23} & -c_1 & s_1.(L_2.c_2 + L_1) \\ s_{23} & c_{23} & 0 & L_2.s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now considering the 4th reference frame:

$$^3T_4 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\therefore\ \ ^0T_4 = \ ^0T_3\ ^3T_4$

With the following information it is possible to find, $^0J(\theta)$, via differentiating, $^0P_{YORG}$.

$$^0T_4 = \begin{bmatrix} c_1.c_{23} & -c_1.s_{23} & s_1 & c_1.(L_1 + L_2.c_2 + L_3.c_{23}) \\ s_1.c_{23} & -s_1.s_{23} & -c_1 & s_1.(L_2.c_2 + L_1 + L_3.c_{23}) \\ s_{23} & c_{23} & 0 & L_2.s_2 + L_3.s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J(q) = \begin{bmatrix} \frac{dx}{dq1} & \frac{dx}{dq2} & \frac{dx}{dq3} \\ \frac{dy}{dq1} & \frac{dy}{dq2} & \frac{dy}{dq3} \\ \frac{dz}{dq1} & \frac{dz}{dq2} & \frac{dz}{dq3} \end{bmatrix}$$

$$J(q) = \begin{bmatrix} -s_1.(L_1 + L_2.c_2 + L_3.c_{23}) & -c_1.(L_2.s_2 + L_3.s_{23}) & -c_1.L_3.s_{23} \\ c_1.(L_1 + L_2.c_2 + L_3.c_{23}) & -s_1.(L_2.s_2 + L_3.s_{23}) & -s_1.L_3.s_{23} \\ 0 & L_2.c_2 + L_3.c_{23} & L_3.c_{23} \end{bmatrix}$$

Also, $^4J(\theta)$, can be calculated as $^4_0R^0J(\theta)$.

$$^1W_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}, \ ^1V_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$^2W_2 = {}^2_1R\,^1W_1 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} C_2 & 0 & S_2 \\ -S_2 & 0 & C_2 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix}$$

$$^2W_2 = \begin{bmatrix} S_2\dot{\theta}_1 \\ C_2\dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \ ^2V_2 = {}^2_1R(\,^1V_1 + \ ^1W_1 + \ ^1P_2)$$

$$^2V_2 = \begin{bmatrix} C_2 & 0 & S_2 \\ -S_2 & 0 & C_2 \\ 0 & -1 & 0 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ L_1\dot{\theta}_1 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ -L_1\dot{\theta}_1 \end{bmatrix}$$

$$^3W_3 = {}^3_2R\,^2W_2 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} C_3 & S_3 & 0 \\ -S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_2\dot{\theta}_1 \\ C_2\dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix}$$

$$^3W_3 = \begin{bmatrix} S_{23}\dot{\theta}_1 \\ C_{23}\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} \qquad ^3V_3 = {}^3_2R(\,^2V_2 + \ ^1W_1 + \ ^1P_2)$$

$$^3V_3 = \begin{bmatrix} C_3 & S_3 & 0 \\ -S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ -L_1\dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ L_2\dot{\theta}_2 \\ -C_2L_1\dot{\theta}_1 \end{bmatrix} \right)$$

$$^3V_3 = \begin{bmatrix} S_3L_2\dot{\theta}_2 \\ C_3L_2\dot{\theta}_2 \\ -L_1\dot{\theta}_1 - C_2L_2\dot{\theta}_1 \end{bmatrix}$$

$$^4W_4 = \ ^3W_3$$

$$^4V_4 = {}^4_3R(\,^3V_3 + \ ^3W_3 + \ ^3P_y) = \ ^3V_3 + \ ^3W_3 + \ ^3P_4$$

$$^4V_4 = \begin{bmatrix} S_2L_2\dot{\theta}_2 \\ C_3L_2\dot{\theta}_2 \\ -L_1\dot{\theta}_1 - C_2L_2\dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ L_3(\dot{\theta}_2 + \dot{\theta}_3) \\ -C_{23}L_3\dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} S_2L_2\dot{\theta}_2 \\ C_3L_2\dot{\theta}_2 + L_3(\dot{\theta}_2 + \dot{\theta}_3) \\ -L_1\dot{\theta}_1 - C_2L_2\dot{\theta}_1 - C_{23}L_3\dot{\theta}_1 \end{bmatrix}$$

Now differentiating each element w.r.t $\dot{\theta}_1, \dot{\theta}_2$ and $\dot{\theta}_3$

$$^4J(\theta) = \begin{bmatrix} 0 & L_2.S_3 & 0 \\ 0 & L_2.C_3 & L_3 \\ -(L_1 + L_2.C_2 + L_3.C_{23})0 & 0 & 0 \end{bmatrix}$$

5.2. Results:

As can be seen from the above analysis, on performing forward Kinematics, both the DH conventions yield the same transformation matrix. In this solution we have used the Rotation Matrix to calculate the value of $\theta_1$and the translation matrix to calculate the values of $\theta_2 and\ \theta_3$. We have also obtained the Jacobian matrix using two methods.

6. <u>Task 3</u>

For task 3 we have chosen to analyse The UR5e Robot, created by Universal Robots [5]

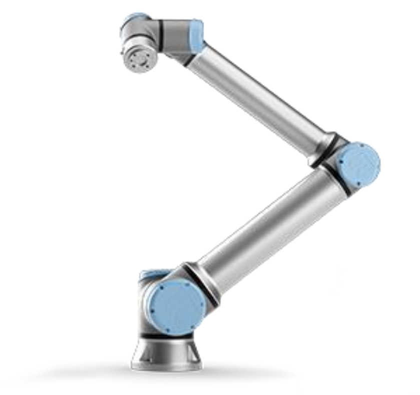

Figure 5: UR5e Robot Arm

The UR5e robot arm is a versatile and flexible collaborative robot that is widely used in industrial applications. One of the key challenges in working with the UR5e is developing and testing control algorithms and motion planning strategies that can help improve its performance. This is where MuJoCo comes in, as it provides a powerful platform for simulating and analysing the behaviour of the UR5e.

Using MuJoCo, researchers and engineers can build detailed models of the UR5e robot arm and simulate its behaviour under a wide range of conditions. This can be useful for testing new control algorithms, evaluating the impact of different sensors and actuators, and analysing the robot's performance under various loads and operating conditions.

MuJoCo offers a range of advanced features that make it particularly well-suited for simulating and analysing the behaviour of the UR5e. For example, it includes support for inverse kinematics, which allows users to specify the desired end-effector position and orientation, and then compute the joint angles needed to achieve that position. It also includes support for dynamic simulation, which allows users to model the impact of external forces on the robot arm, such as those caused by contact with objects or changes in the environment.

1) Sketch of UR5e Robot

Below is a sketch of the UR5e Robot, [6] drawn from analysing the XML file

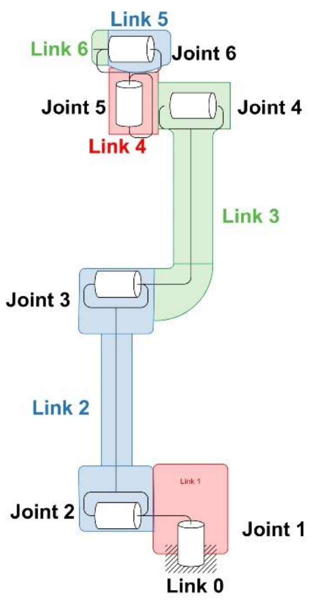

Figure 6: Sketch of UR5e Robot

2) Video Analysis of the UR5e Robot Arm

In this task, a video analysis of the UR5e robot arm in MuJoCo [7] was conducted. The video showcases the capabilities of MuJoCo as a powerful tool for simulating and analysing the behaviour of the robot arm. The analysis includes a description of the UR5e's physical characteristics, such as its joints and links, as well as its kinematic and dynamic properties. The video also demonstrates how MuJoCo can be used to simulate the motion of the robot arm.

7. Conclusion:

In the first task of this coursework, a sketch and the forward kinematics of a robot with 3 links were derived using D-H parameters. The joint values required to achieve a desired end-effector position were determined, and the manipulator Jacobian was computed for the robot. Based on the solutions obtained, it can be concluded that there are multiple possible joint configurations that can achieve a given end-effector position, as the robot has 3 degrees of freedom.

Furthermore, it was found that the manipulator Jacobian is a useful tool for analysing the robot's motion and determining its singular configurations, where the robot loses the ability to move in certain directions.

In the second task, the D-H table of a 3R nonplanar robot manipulator was created to define the coordinate frames of each joint and calculate the forward kinematics. The kinematic equations were used to obtain the inverse kinematics for the manipulator, enabling the determination of the joint angles necessary to reach a desired end-effector position. Then, the Jacobian matrix of the manipulator with respect to the end-effector position was derived, which allowed for the determination of the singular configurations of the robot.

In Task 3, a simulation of the UR5e robot arm was created in MuJoCo, allowing for the evaluation of its kinematic and dynamic properties under different operating conditions. The use of MuJoCo proved to be an invaluable tool for simulating and analysing the behaviour of the robot arm.

Overall, the fundamental concepts in robotics were explored through this coursework, including modelling of the kinematics of a robot manipulator, calculation of its forward and inverse kinematics, and analyzation of its singularity behaviour using the Jacobian matrix. These concepts are essential for designing and controlling robotic systems. These find widespread applications in various industries such as manufacturing, medical robotics, space exploration, autonomous vehicles, and human-robot collaboration.

8. References & Bibliography

[1] A. W. Wundheiler, 'Discussion: "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices" (Denavit, J., and Hartenberg, R. S., 1955, ASME J. Appl. Mech., 22, pp. 215–221)', *J. Appl. Mech.*, vol. 23, no. 1, pp. 152–153, Jun. 2021, doi: 10.1115/1.4011256.

[2] F. Piltan, A. Taghizadegan, and N. B. Sulaiman, 'Modeling and Control of Four Degrees of Freedom Surgical Robot Manipulator Using MATLAB/SIMULINK', *Int. J. Hybrid Inf. Technol.*, vol. 8, no. 11, pp. 47–78, Nov. 2015, doi: 10.14257/ijhit.2015.8.11.05.

[3] J. J. Craig, 'Mechanics and Control Third Edition'.

[4] D. Constantin, M. Lupoae, C. Baciu, and B. Ilie, *Forward Kinematic Analysis of an Industrial Robot*. 2015.

[5] 'Collaborative robots from UR | Start your automation journey'. https://www.universal-robots.com/products/ (accessed Feb. 24, 2023).

[6] K. Lee, J. Lee, B. Woo, J. Lee, Y.-J. Lee, and S. Ra, 'Modeling and Control of a Articulated Robot Arm with Embedded Joint Actuators', *2018 Int. Conf. Inf. Commun. Technol. Robot. ICT-ROBOT*, pp. 1–4, Sep. 2018, doi: 10.1109/ICT-ROBOT.2018.8549903.

[7] 'deepmind/mujoco'. DeepMind, Feb. 25, 2023. Accessed: Feb. 25, 2023. [Online]. Available: https://github.com/deepmind/mujoco

9. Appendix

Examples of Applications of Robotics

Robotic kinematic analysis can be applied to any type of robotic systems, particularly industrial robots, which have an end-effector or a manipulator at the end. The interest in applying principles of robotic kinematic analysis pertains to reducing labour costs, increasing economies of scales, and increasing the quality of batch production. However, for this to be feasible robotic designs must be improved upon; one key aspect of the design is considering the inverse robotic kinematic analysis, which can allow for optimum position and orientation of desired robotic manipulator (such as an arm).

Another example of how robotic kinematics has been applied is within the medical industry, which could have the advantages of reducing surgeon hand tremors, decreasing post-operative complications, and increasing dexterity during operation. There is research that aimed to utilize advanced robotic manipulators, where MATLAB has been used, where a robotic surgical arm has been created and optimised [2].

Furthermore, with the development of the mathematical theory behind robotic kinematics, the principles can be applied to simulation software such as MATLAB, where the robots can be prototyped. Robotic equations can be modelled in MATLAB to create a 3D visual simulation of the robot arm, hence there are many other software that can be used to model robotic equations other than MuJoCo.

The differences between Classical and Modified DH notation.

The Denavit-Hartenberg (DH) notation is a widely used method for describing the kinematic structure of robotic systems. This notation involves assigning four parameters, namely the link length, the link twist, the link offset, and the joint angle, to each joint of the robot. However, there are two variations of the DH notation: classical DH notation and modified DH notation. While both notations use the same four parameters, there are some key differences between them.

Classical DH notation involves assigning the z-axis of each joint frame along the joint axis, while the x-axis is perpendicular to both the z-axis of the current joint and the z-axis of the previous joint. This means that the orientation of the z-axis of each joint frame is dependent on the orientation of the previous joint, which can lead to ambiguity when the robot has multiple degrees of freedom. In classical DH notation, the link offset is defined as the distance between the z-axes of adjacent joint frames, while the link length is defined as the distance between the x-axes of adjacent joint frames.

On the other hand, modified DH notation involves assigning the x-axis of each joint frame along the joint axis, while the z-axis is perpendicular to both the x-axis of the current joint and the x-axis of the previous joint. This means that the orientation of the x-axis of each joint frame is independent of the orientation of the previous joint, which eliminates the ambiguity associated with classical DH notation. In modified DH notation, the link length is defined as the distance between the x-axes of adjacent joint frames, while the link offset is defined as the distance between the z-axes of adjacent joint frames.

Another difference between classical DH notation and modified DH notation is the way the joint angle is defined. In classical DH notation, the joint angle is defined as the rotation about the z-axis of the current joint frame, while in modified DH notation, the joint angle is defined as the rotation about the x-axis of the current joint frame. This means that the joint angle in modified DH notation is independent of the orientation of the previous joint, which further eliminates ambiguity.