ME 5250 Fall 2025 Project – II

6-DOF UR3 Robot Manipulation **December 2025**

1. Introduction

The use of industrial robots on industrial production lines has considerably increased production efficiency, putting greater emphasis on the precision and intelligence of industrial robot activities. Currently, many industrial robots are spot welding robots, arc welding robots, spray robots, and palletizing robots. The general steps for different types of robotic arms to work are as follows [1]:

1. Figuring out the workspace.
2. Detection for Object pose.
3. Planning the trajectory for end effectors

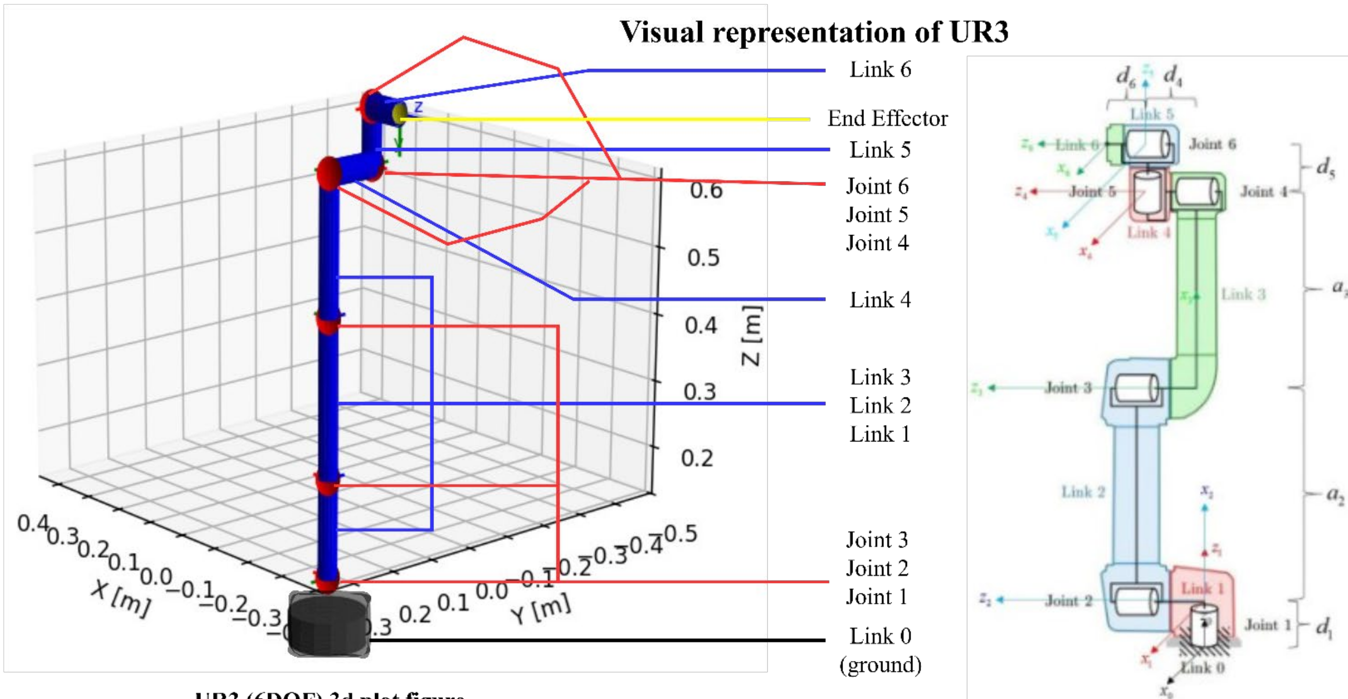
After completing the above three steps, different tasks can be achieved by setting specific functions for different types of end effectors of the robotic arm. For this kind of tasks, we use robotic arms called "Serial robotic manipulators", which are widely used in industrial and collaborative robotics due to their large workspaces, flexible motion capabilities, and ease of control. Accurate modeling of robot kinematics is essential for tasks such as trajectory planning, motion control, and manipulation in task space. In particular, the ability to compute forward and inverse kinematics enables the robot to follow prescribed end-effector paths while keeping desired orientation constraints.

In this project, a **6-degree-of-freedom UR3 collaborative robot** is selected as the test platform. The UR3 robot is representative of modern lightweight industrial manipulators and has well-documented geometric parameters, making it suitable for analytical kinematic modeling. Using published Denavit–Hartenberg (DH) parameters, an analytical forward kinematics model is derived and implemented, we also implement A numerical inverse Kinematics (IK) solver based on Newton's method, it is developed to compute joint configurations corresponding to desired end-effector poses.

2. Robot Modeling and Forward Kinematics

2.1 UR3 Robot Description

The UR3 robot is a 6-DOF serial manipulator with six revolute joints arranged in an anthropomorphic configuration. The robot consists of a vertical base joint followed by shoulder, elbow, and three wrist joints, allowing full positioning and orientation control of the end effector. A schematic representation of the UR3 kinematic structure and frame assignments is shown in Fig. 1. We can see the end effectors TCO as an RGB frame clearly in Fig. 2.



UR3 (6DOF) 3d plot figure

Schematic and frames assignment of UR3

Fig. 1. Visual Representation of UR3 robotic arm

To model the robot, the **standard Denavit–Hartenberg convention** is used. Each joint is associated with four parameters: link length a_{i-1}, link twist α_{i-1}, link offset d_i, and joint angle θ_i. The DH parameters adopted in this project are taken directly from published UR3 modeling resources and are summarized in Table 1 [2][3].

TABLE I. THE DH PARAMETERS TABLE OF UR3 ROBOT

Link	θ_i *(radians)*	a_{i-1} *(meters)*	d_i *(meters)*	α_{i-1} *(radians)*
Link_1	0	0	0.1519	π /2
Link_2	0	-0.24365	0	0
Link_3	0	-0.21325	0	0
Link_4	0	0	0.11235	π/2
Link_5	0	0	0.08535	-π/2
Link_6	0	0	0.0819	0

All joints are revolute, so the joint variables are the angles θ_i.
As shown in fig 1 the partial kinematic diagram of the UR3 robot is established based on the DH parameters of the first and second joints.[3]

2.2 Forward Kinematics Formulation

Using the DH parameters [2], the homogeneous transformation from frame $i-1$ to frame i is given by:

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

The forward kinematics of the UR3 robot is obtained by multiplying the six successive transformations:

$$^{0}T_6(q) = \prod_{i=1}^{6} {}^{i-1}T_i(q).$$

This transformation fully describes the pose of the end effector relative to the base frame, including both position and orientation.

3. Jacobian and Inverse Kinematics

3.1 Geometric Jacobian

To solve inverse kinematics numerically, **geometric Jacobian** is needed. For a serial manipulator with all revolute joints, the Jacobian columns are given by [3]:

$$J_v^{(i)}(q) = z_{i-1}(q) \times \big(p_e(q) - p_{i-1}(q)\big)$$

where:

- p_i is the origin of joint i expressed in the base frame,
- z_i is the axis of rotation of joint i expressed in the base frame,
- p_e is the end-effector position.

The full Jacobian has the structure:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} = \begin{bmatrix} J_v^{(1)} & J_v^{(2)} & \cdots & J_v^{(6)} \\ J_\omega^{(1)} & J_\omega^{(2)} & \cdots & J_\omega^{(6)} \end{bmatrix}.$$

This Jacobian maps joint velocities to the end-effector spatial twist:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J(q)\,\dot{q},$$

3.2 Newton-Based Inverse Kinematics

Inverse Kinematics is solved using an iterative Newton-Raphson method. Given a desired end-effector pose Td= [Rd, pd], the error vector is defined as:

$$e = \begin{bmatrix} p_d - p \\ \mathrm{Log}\big(R_d R^T\big) \end{bmatrix}$$

where the orientation error is computed using the matrix logarithm map from SO (3).[3][4]

At each iteration, the joint update is computed as:

$$\Delta q = J^T \left(J J^T + \lambda I\right)^{-1} e$$

where λ is a small damping factor for numerical stability. The joint angles are updated until the position and orientation errors fall below specified tolerances.

4. Task-Space Trajectories and Results

4.1 Task-Space Trajectory Definition

Few Trajectories are defined directly in task space as sequences of cartesian end-effector positions spaced at approximately 1mm resolution. The end-effector as shown in fig 2, Is held fixed through the path to show constrained task-space control. The included trajectories shaped paths are 1. Planar Infinity, 2. Hexagonal, 3. Star-shaped, 4. 3D Elliptical trajectory.

4.2 Visualization and workspace Analysis

The Robot arm's workspace is visualized using "**Monter Carlo sampling**" approach. Random joint configurations were generated, and the corresponding end-effector positions are computed using forward kinematics, repeating this process produces a point cloud representation of the reachable workspace.

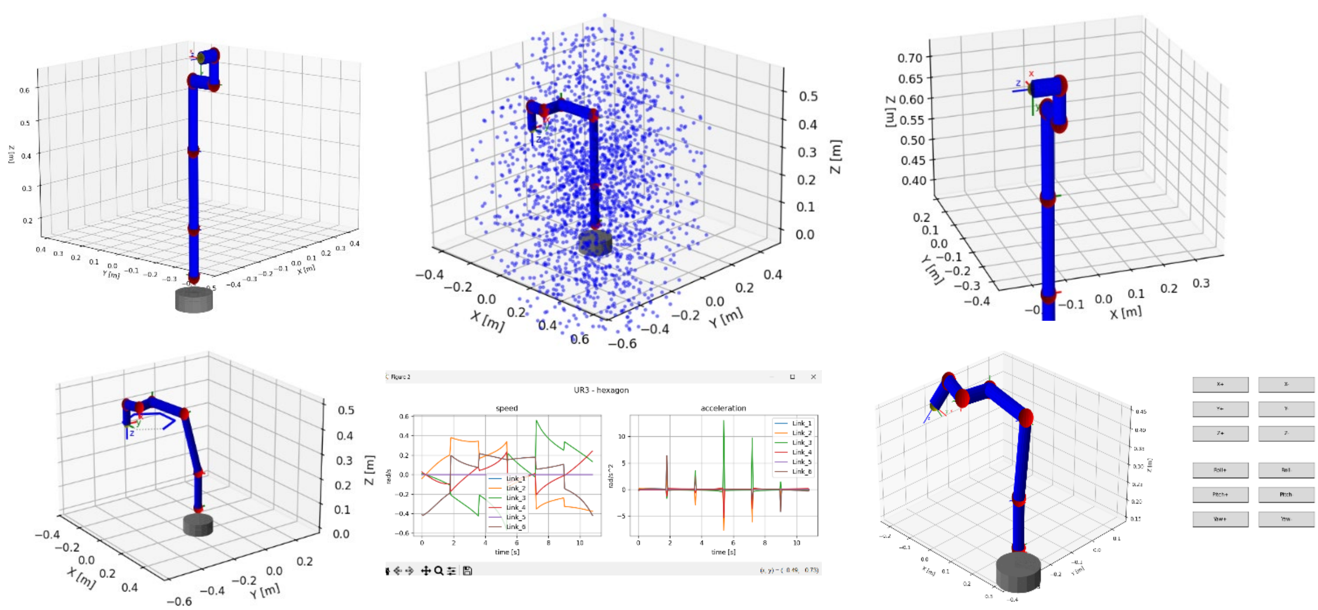


Fig.2. Overview: The figure includes the robotic arm - 1. in its home pose, 2.Workspace of the UR3, 3. RGB coordinate frame at EE, 4. EE of the robot following shaped path & Joint Velocities and accelerations are computed numerically to analyze motion smoothness during path following, 5. A simple Interactive Robot mover.

5. References

1. https://ieeexplore.ieee.org/document/9541299
2. https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/
3. https://ieeexplore.ieee.org/document/10335347
4. https://ieeexplore.ieee.org/document/7844896