Introduction to Universal Robot 5

Varshini Rao Venkateshwara Rao
[1]Robotics and Autonomous Systems (AI), Arizona State University, Arizona, USA

Article Info

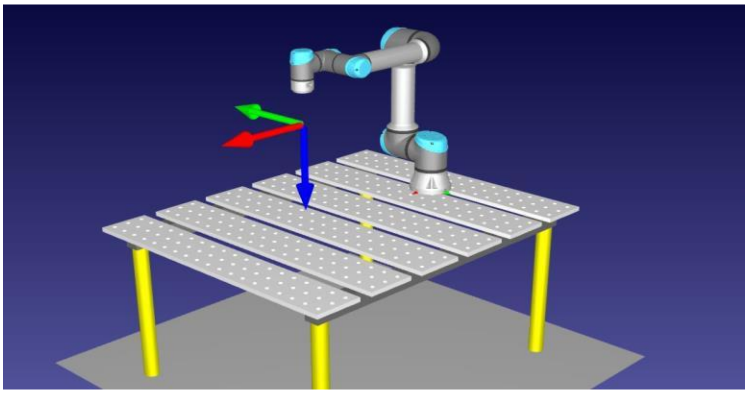
UR5 LAB 1

Keywords:

Universal Robot UR5
Way Points
Polyscope
RoboDK
URScript
End Effector
Programming Environment
Target Position
Joint Angles
Robot control

ABSTRACT (10 PT)

The Universal Robot 5 is a cutting-edge industrial robot that has improved the applications of collaborative robot in the field of automation and robotics. It has a very simple user interface for easy integration with in various industries. The introductory lab of UR5 was aimed for programming the robot using three different kinds of programming environment (Polyscope, RoboDK and URScript). The main tasks of this lab was to learn how to move the robot through different way points and finally draw a cube by moving the robot arm through it's 8 vertices. Through these exercises, we gained a deeper understanding of the robot's kinematics, workspace, and various programming paradigms.



1. INTRODUCTION

Understanding how to program and control robots is a fundamental skill for engineers and researchers working in this field. In this lab, we had the opportunity to work with the Universal Robots UR5, a versatile and collaborative robotic arm, to explore different programming environments and concepts.

The lab tasks helped us gain understanding on three different key areas:

- **Robotic Programming Environments:**
  The lab task consisted of using the Polyscope for moving and programming the robot in real time. The RoboDK simulation software created an understanding of simulating the industrial robot and implement the required environment for performing the tasks in different real environments through simulation. The URScript task made us understand, how to control the robot only using the codes. Understanding the capabilities of each environment is crucial for effectively utilizing the robot in various applications.

- **Workspace Analysis:**
  An important aspect of working with robots is assessing their workspace, which defines the region within which they can perform tasks. We explored the limitations and possibilities of the UR5's workspace, considering the robot's physical mounting and reach.

- **Robot Kinematics:**
  We delved into the realms of joint and Cartesian space, two essential coordinate systems for robot control. This allowed us to grasp how robot positions and orientations are specified, and how these choices impact the robot's behavior during task execution.

The robot's workspace is inherently limited by its physical mounting location on the table. The key limitations observed during the tasks included:

- **Limited Reach:** The robot's arm length, coupled with its base position, imposed constraints on its reach. The workspace was confined within the reachable volume determined by the arm's kinematic structure.
- **Obstacle Interference:** Fixed obstacles or obstructions on the lab table restricted certain robot movements. We had to ensure that the robot's path was free from collisions with the surroundings.
- **Z-Axis Limitation:** The height at which the robot was mounted on the table restricted its vertical reach and prevented it from accessing positions outside the designated Z-axis limits.

2. OBJECTIVE

The objective of this lab is to become familiar with the UR5 robot and to operate it using Polyscope, RoboDK and URScript. The tasks of the lab was to move the robot using Polyscope and free drive from one way point to other, The robot to be moved using RoboDK simulation and implementing them in real robot, The same task is to be performed using URScripts and finally specifying waypoints in both joint and Cartesian space to create a virtual cube.

3. METHODOLOGY

<u>LAB1</u>

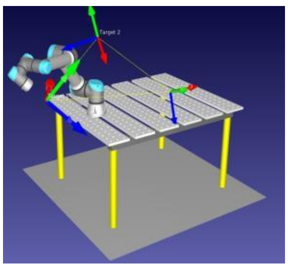
- **Task 1 (Moving to Waypoints in PolyScope):**

In this task, during our experimentation in PolyScope, we found that joint space control enabled precise control over each joint's angle, allowing the robot to follow a specific path efficiently. In Cartesian space, we could specify the robot's end-effector position and orientation directly, simplifying path planning and enabling smoother movements. The choice between these spaces significantly affected the robot's motion, with joint space offering fine-grained control and Cartesian space simplifying task programming.
First, we tried to add some way points to the robot using the free drive. But at the time of movement, the robot could not reach the position due to its joint space limitation. The workspace was confined within the reachable volume determined by the arm's kinematic structure.
So, finally the robot is given some other way points. Then the robot simulation is checked in the simulation in PolyScope itself. Finally, the robot moves through the specified way points.

The specified waypoint positions are:

	X (mm)	Y (mm)	Z (mm)	RX (rad)	RY (rad)	RZ (rad)
Pose 1	-575	-350	300	2.10	1.11	0.63
Pose 2	-240	-445	650	1.57	-1.57	-1.57
Pose 3	400	-400	200	2.79	-0.16	0

Task 2 (Moving the simulation using RoboDK):



The task 2 was to perform the same movement of the robot using the RoboDK simulation software.

- The UR5 model is imported in the software along with a table station with it.
- Then three target positions are recorded using the target button.
- The main program consists of the joint space movement type (in this

case it is linear).

- ➢ The target points are linked to three different linear joint spaces.
- ➢ The program is executed to check the simulation of the movement

Fig: UR5 Pose1

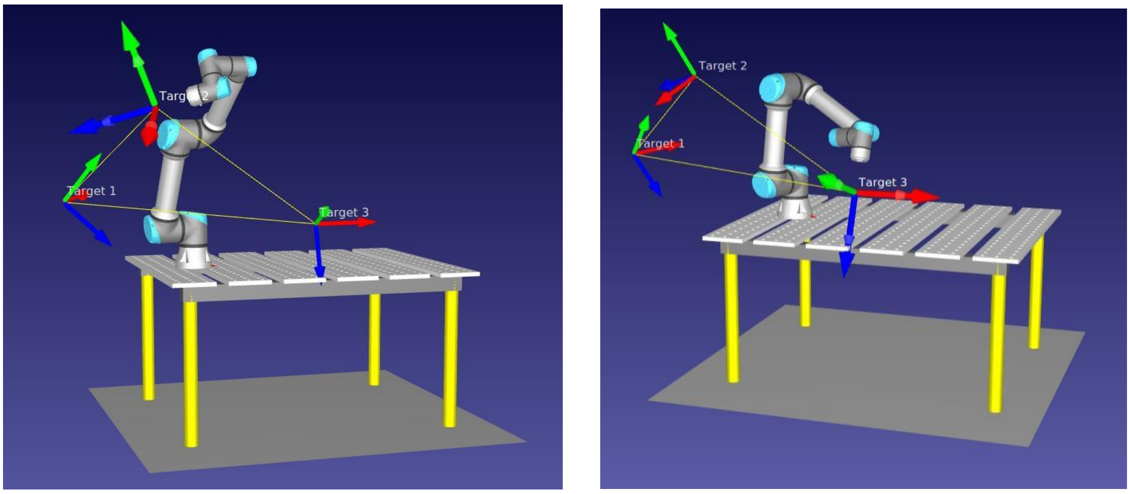


Fig: UR5 Positions in RoboDK simulation

- **Task 3 (Writting the program in URScript):**

  - ➢ In task 3, the positions of the robot is defined using the URScript.
  - ➢ The program is written in such a way that the file can be stored in the Polyscope.
  - ➢ Finally the program is run in the real robot.
  - ➢ The robot repeats the same positions using the URScript file and goes to the specified way points.

- **Task 4 (Drawing an imaginary cube using UR5):**

  - ➢ The task4 is to draw a virtual cube with the UR5 robot.
  - ➢ The way points are the 8 vertices of the virtual cube.
  - ➢ First, the robot moves from an initial position to one of the corners of the cube using the joint space movement (moveJ).
  - ➢ Then the robot program uses the linear movement (moveL) command to move in a straight line.
  - ➢ The robot then traces all the way points and draws a virtual cube.
  - ➢ Finally, it gets back to its initial position.

Tracing the cube

Points	X (mm)	Y (mm)	Z (mm)	RX (rad)	RY (rad)	RZ (rad)
1	-140.07	-609.05	424.95	3.080	0.585	0.035
2	-140.06	-659.00	424.98	3.080	0.585	0.035
3	-90.03	-658.96	425.01	3.080	0.585	0.035
4	-90.04	-608.98	425.01	3.080	0.585	0.035
5	-140.07	-609.05	424.95	3.080	0.585	0.035
6	-90.05	-609.02	324.98	3.080	0.585	0.035
7	-140.05	-609.00	324.96	3.080	0.585	0.035
8	-140.09	-658.98	324.97	3.080	0.585	0.035
9	-90.04	-658.97	325.02	3.080	0.584	0.035

points (vertices) that define the two cubes you traced. These points are specified in millimeters (mm) for X, Y, and Z coordinates and in radians (rad) for Rx, Ry, and Rz rotations. The points are selected in a systematic manner to create two cubes:

- Points 1 to 5 define the vertices of the first cube, and Points 6 to 9 define the vertices of the second cube.
- The X, Y, and Z coordinates determine the positions of the vertices in 3D space.
- The Rx, Ry, and Rz rotations determine the orientation of the cubes.

This configuration ensures that the two cubes have distinct positions and orientations while sharing common vertices. It allows the robot to trace both cubes with consistent gripper orientation while changing the (x, y, z) portion of the pose, as required by the task.

For the volume of the cube:

The side length of the combined cube is given by the Euclidean distance between the two diagonally opposite corners of the cube. In this case, we can consider the distance between Points 1 and 6.

The volume of the combined cube is then V = d^3, where d is the calculated side length.
d = sqrt ((-140.06 - (-90.05))^2 + (-659.00 - (-609.00))^2 + (424.98 - 324.96)^2)
d = sqrt (2500.0025 + 2500.0025 + 10000.0004) ≈ sqrt (15000.0054) ≈ 122.474 mm
(approximately)
So, the volume of the cube (V) is approximately:

- V ≈ (122.474 mm) ^3 ≈ 1,803,438.54 cubic millimeters (mm^3)
- The cube has a volume of approximately 1,803,438.54 cubic millimeters (mm^3).
- This indicates that the robot's workspace, when considering the entire configuration, can accommodate a cube of this size and shape.
- The calculated volume provides an understanding of the robot's effective workspace, which is quite substantial, allowing it to manipulate objects within this volume efficiently.

LAB2

- **Task1 (Controlling the Gripper using Polyscope):**

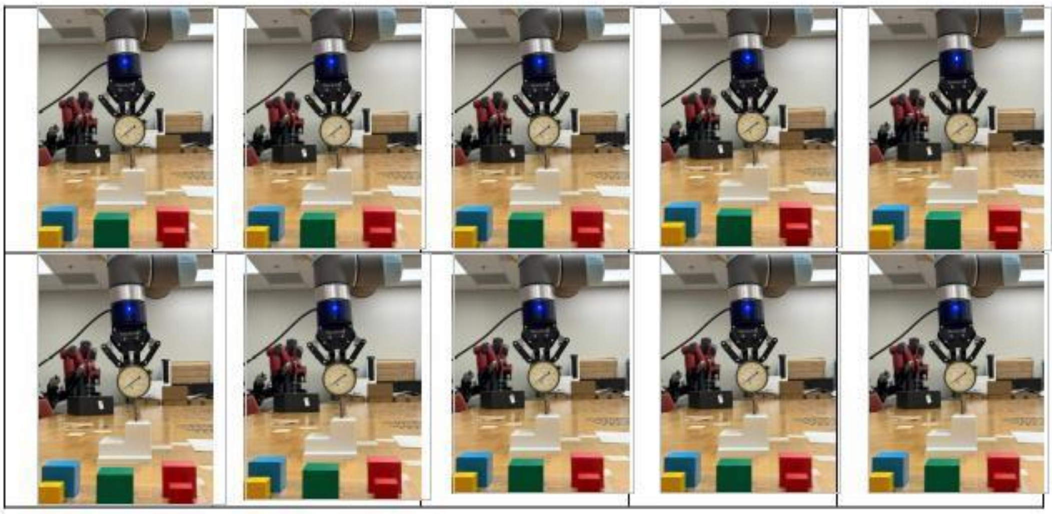
1. The objective of thr task is to learn how to open and close the gripper of the robot. Using polyscope.
2. The gripper control option in the polyscope helps to toggle the gripper status from ON to OFF.
3. The polyscope also has the function to exert required ammount of force to hold any objects and we learnt to set them using the polyscope.
4. Also, the "rq_close()" and "rq_close_and_wait()" functions of the URScript does the same work through the scripting.
5. This task served as a foundation for future tasks and labs that involve more complex gripping tasks and the utilization of the gripper's functions.
6. It is important to keep track of the number of times the gripper has been opened and closed, which can be recorded from the gripper's operating log or by using the relevant commands.

- **Task2 (Assess Straigtness and the repeatability of UR5):**

1. First,We needed to locate the faces of the 3D printed block where the edge of the granite block was traced.
2. Two points are selected as a waypoint near the two vertexes of the block to trace the distance of the edge.
3. The digital indicator (pressure) is positioned to be perpendicular to the face of the surface attaching to the end effector of the UR5.
4. The polyscope program is created to follow the two wayopints and check the digital indicator for the changes in the pressure during following the edge line and its been repeated for assessing the repeatability.
5. After that the face of the granite block is traced using 4 waypoints at the four vertexes of the face.

6. The same process of measuring the pressure throughout the edge following is repeated to check the straigtness and the repeatability of the UR5.

- **Results:**



The robot has a good amount of repeatability and straightness but as the surface was 3D printed and not so smooth so the value of the pressure changes in some of the iteration near some of the edges.

2.3 LAB 3

The objective of this lab is to familiarize oneself with the UR5 robot and the ROBOTIQ adaptive robot gripper for gripping tasks. The tasks encompass the procedures of opening and closing the gripper, calibrating the gripper's force, executing pick-and-place actions, and inserting/removing a dowel pin.

Task 1: Gripper Opening and Closing

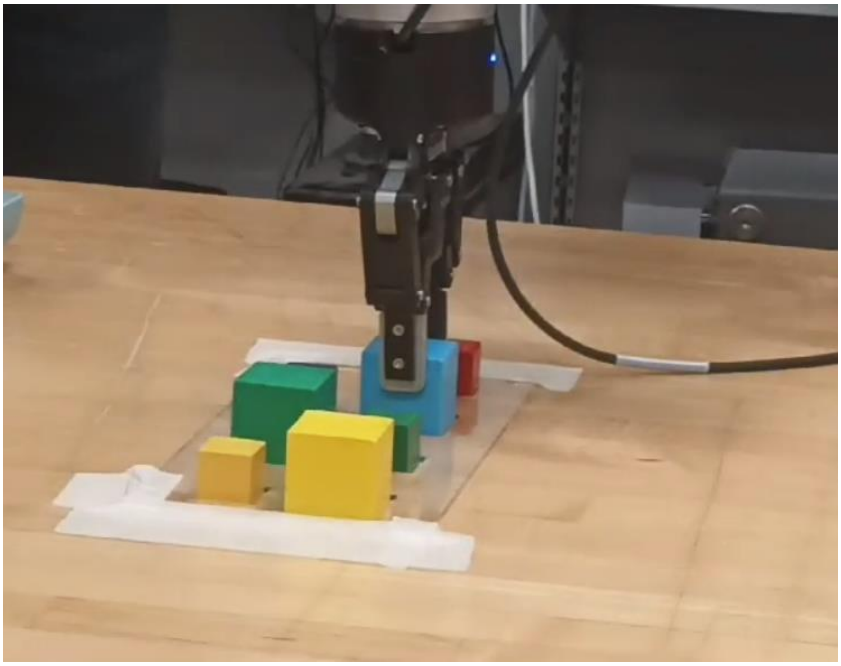
- The robot is powered on, and the gripper manual is followed to perform gripper opening and closing actions.
- Exploration of additional gripper functions is carried out using URScript.
- Optionally, the tally of gripper open and close operations throughout its lifespan can be determined.

Task 2: Grasping the Load Cell Assembly and Calibration

- The load cell's location on the table is identified.
- Calibration of the gripper's force is conducted in Newtons, with consideration of speed and force settings.
- A program is authored to grip the load cell at varying force and speed settings, with data recording.

Task 3: Object Pick-and-Place

- A program is crafted to transfer blocks from one pallet to another without causing damage.
- The determination of coordinate transformations from the robot's base frame to the pallets is essential.
- A program is written, incorporating variables for flexible positioning and orientation of the pallets.



Task 4: Dowel Pin Insertion and Removal

- A program is devised to insert and remove a 10mm dowel pin into holes of different sizes.
- A specific sequence is adhered to for transferring the pin between holes within the mating plate.

2.4 Lab 4

The objective of this lab task is to assemble a flashlight using the UR5 robot, a pneumatic chuck, and various flashlight components. This assembly process involves precise movements and torque control to ensure a successful assembly.

Lab Procedure:

Setup:

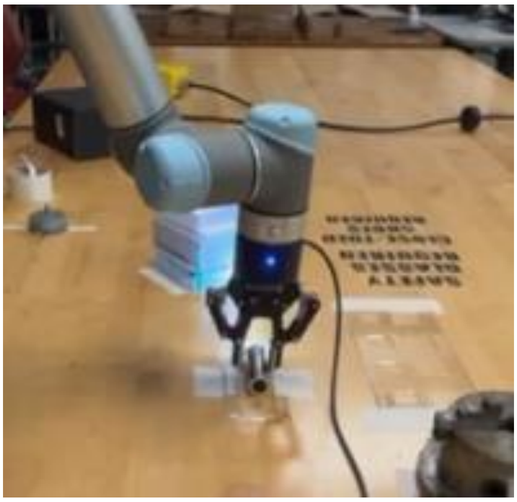
The assembly process is initiated by clamping the head of the flashlight into the pneumatic chuck, which is controlled by digital I/O in the robot controller.
Flashlight Components: A tray is used to place four flashlight components.

Assembly Steps:

The assembly steps consist of the following:
a. The head is placed into the chuck.
b. The chuck is clamped using Digital I/O.
c. The barrel is positioned onto the head.
d. The barrel is screwed onto the head until a torque of 3 Nm is achieved, with torque monitored using get_joint_torques().
e. The battery pack is inserted into the barrel, ensuring correct polarity.
f. The tail cap is placed onto the barrel.
g. The tail cap is aligned and screwed until a torque of 2 Nm is reached.
h. The assembly is unclamped with Digital I/O.
i. The assembled flashlight is removed from the chuck and placed in the flashlight tray.

Assembled Flashlight:



Recommendations:

Maintaining consistency in component orientation is of utmost importance. Threading can be time-consuming; manual assistance is allowed, but for video demonstrations, autonomous completion by the robot is preferred. Initial torque readings may be high but are expected to decrease over time. Commence assembly at lower speeds and gradually increase efficiency. Consider implementing gripper release methods to minimize part misalignment. Fine-tune part positions for enhanced accuracy. The inclusion of delays in the process can be advantageous. Save multiple versions of your code and keep a record of modifications. Slower robot movements during part insertion can enhance precision. The use of poses defined by joint angles is recommended to avoid exceeding joint limits.

Collaborative Robotics:

Collaborative robotics, often referred to as "cobots," is a type of robotics technology that focuses on the collaboration between robots and humans in a shared workspace. Unlike traditional industrial robots that are typically kept behind safety cages and are isolated from human workers, collaborative robots are designed to work alongside people, enhancing productivity, efficiency, and safety in various applications.

Key characteristics of collaborative robotics include:

1. Safety Features: Cobots are equipped with advanced safety features to prevent collisions and injuries. This can include sensors that detect the presence of humans and stop or slow down the robot's movements when a person is in close proximity.
2. Ease of Programming: Collaborative robots are typically designed to be user-friendly, with easy-to-use interfaces that allow non-experts to program them. This makes them accessible to a wider range of industries and applications.
3. Force and Tactile Sensing: Some cobots are equipped with force and tactile sensors, allowing them to adapt to different tasks and work more closely with human operators. This enables them to handle delicate tasks or adjust their movements based on the environment.
4. Reduced Footprint: Collaborative robots are often compact, which allows them to work in confined spaces or on production lines with limited real estate.
5. Diverse Applications: Cobots are used in a wide range of industries, including manufacturing, logistics, healthcare, agriculture, and more. They are employed in tasks such as assembly, material handling, welding, inspection, and even as personal assistants for individuals with disabilities.

Collaborative robotics has the potential to improve efficiency and safety in many industries, as these robots can handle repetitive or physically demanding tasks, while humans can focus on more complex and cognitive aspects of the work. This technology continues to evolve, with ongoing advancements in sensors, artificial intelligence, and human-robot interaction to make cobots even more capable and versatile.

Elbow Configuration:

The UR5 is a popular collaborative robotic arm developed by Universal Robots. It has a standard articulated robotic arm configuration, which consists of several joints. Specifically, the UR5 has six joints, allowing for six degrees of freedom (DOF), which makes it highly flexible and capable of performing a wide range of tasks. Here is a description of the elbow configuration of the UR5:

1. Base (Joint 1): The first joint is located at the base of the robot, where it attaches to a fixed point or a mobile platform. This joint allows the UR5 to rotate horizontally, usually in a full 360-degree range, enabling the arm to move left and right.
2. Shoulder (Joint 2): The second joint is known as the shoulder joint and is connected to the base. It allows the robot arm to lift up and down, moving vertically. This joint's motion provides the vertical reach of the UR5.
3. Elbow (Joint 3): The third joint is the elbow joint, and this is where the term "elbow configuration" typically refers. It allows the arm to bend or extend. The motion at this joint gives the UR5 its elbow-like flexibility.
4. Wrist 1 (Joint 4): The fourth joint, often referred to as the wrist joint or wrist 1, is positioned after the elbow joint. It allows the robot's arm to rotate horizontally around its own axis, enabling it to orient the end-effector in different directions.
5. Wrist 2 (Joint 5): The fifth joint, known as wrist 2, is located after wrist 1. It provides another rotational degree of freedom, allowing the end-effector to tilt or pitch in different directions.
6. Wrist 3 (Joint 6): The sixth and final joint, wrist 3, is situated at the end of the robot arm. It offers a final rotation capability, typically allowing the end-effector to roll or twist.

The combination of these six joints and degrees of freedom gives the UR5 its high dexterity and versatility. It can perform a wide range of tasks, from simple pick-and-place operations to more complex motions and orientations needed for tasks like assembly, welding, and other industrial applications. The UR5 is known for its ease of programming and its collaborative capabilities, making it a valuable tool in various industries.

Compared and contrasted the three programming methods—PolyScope, RoboDK, and URScript based on experiences during the lab:

PolyScope:

- PolyScope offers a user-friendly graphical interface that is intuitive for beginners. It simplifies robot programming, making it accessible to users with limited programming experience.
- It provides a high-level of control, enabling both joint and Cartesian space programming. However, it may have limitations for advanced users seeking extensive customization.
- PolyScope is well-suited for quick setup and execution of tasks, making it ideal for introductory tasks and learning purposes.

RoboDK:

- RoboDK offers a powerful simulation environment, making it excellent for off-line programming. The interface can be more complex for novices, but it provides advanced capabilities.
- RoboDK excels in off-line programming, enabling detailed control over robot paths and simulations. It supports complex tasks and extensive customization.
- RoboDK is valuable for tasks that require precise path planning and simulation. It is suitable for advanced projects and in-depth robot control.

URScript:

- URScript is a text-based language, requiring programming skills. It offers the highest level of control and flexibility but comes with a steeper learning curve.
- URScript provides extensive customization and fine-grained control, making it suitable for complex and specialized applications.
- URScript is ideal for projects that demand advanced programming, custom algorithms, and specific control requirements. It may be the preferred choice for advanced robotics work.

4. USABLE WORKSPACE OF UR5 ROBOT

The usable workspace of the Universal Robots UR5 robot refers to the volume within which the robot arm can reach and perform tasks effectively. The UR5 has a spherical work envelope due to its articulated arm configuration with six degrees of freedom. The specific dimensions of the usable workspace can vary depending on the robot's installation, mounting position, and reach, but I can provide some general information.

The UR5 has a maximum reach of approximately 850 millimeters (about 33.5 inches). This reach is often measured from the robot's base to the center of its wrist, allowing it to access points within this radius. However, it's important to note that the usable workspace isn't just a simple spherical volume, as the orientation of the end-effector can affect where it can perform tasks effectively.

In practical terms, the usable workspace of the UR5 can be described as follows:

- Spherical Reach: The robot arm can reach any point within a sphere with a radius of approximately 850 mm, assuming the arm is fully extended.
- Work Height: The UR5 can also reach heights above the floor level, allowing it to perform tasks at various elevations.
- Dexterity: The articulated arm allows the UR5 to reach points at different angles and orientations within its reach, including overhead, below its base, and at various angles to the base.

- Collaborative Workspace: The UR5 is designed for collaborative work with humans, which means it can effectively reach points within its workspace while ensuring safety when working in proximity to people.
- Adjustable Reach: The reach and effective workspace can be adjusted by mounting the robot in different positions, using external tooling, or combining it with linear tracks or other automation equipment to extend its reach.

While the UR5 has a generous working envelope, the effective workspace can also be influenced by factors such as the size and shape of the end-effector or tooling attached to the robot, as well as any obstacles or restrictions in the environment.

5. CONCLUSION

The lab tasks and the initial module helped us to gain the understanding about the specifications and how to use and program the UR5 collaborative robot. The key concepts from the lab are:

- The knowledge about the UR5 industrial robot specifications.
- Difference between Joint and Cartesian space using the Move commands and setting the way points.
- Simulation of the robot work before implementing in an actual robot in Polyscope as well as in RoboDk.
- Tracing a the vertices of a cube in a linear motion and getting the idea of the three different kinds of motions during the tracing process.
- The report discusses the advantages and limitations of the UR5 robot. The comparision between three different programming techniques are also been discussed. Also, it the robot workspace is discussed.
- The module helped us to discuss about the safe interaction between the human and robot workspace.