MAE 547

MODELLING AND CONTROL OF ROBOTS

FINAL PROJECT REPORT

FORWARD & INVERSE KINEMATICS AND TRAJECTORY PLANNING IN A 6 DOF COBOT ROBOT

Submitted by:

Ashish Paka
ASU ID: 1231152574
Course: M.S. in Robotics and Autonomous Systems
(MAE Concentration)

Akshay Sopan Mahalle
ASU ID: 1229915429
Course: M.S. in Robotics and Autonomous Systems
(MAE Concentration)

Manaswi Thinagaran
ASU ID: 1229978752
Course: M.S. in Aerospace Engineering
(Controls and Dynamics, Aerodynamics)

CONTENTS

INTRODUCTION

In the rapidly evolving landscape of robotics, the integration of collaborative robots (COBOTs) has become instrumental in enhancing industrial processes, augmenting human capabilities, and ensuring efficient task execution. This project delves into the dynamic realm of robotics, specifically focusing on the simulation of a 6 Degrees of Freedom (6 DOF) COBOT. The fundamental principles of forward kinematics, inverse kinematics, and trajectory planning are harnessed to propel the capabilities of this COBOT, offering a comprehensive exploration into its spatial movements and task execution methodologies.

PURPOSE AND OBJECTIVES

The primary purpose of this project is to comprehend and demonstrate the intricate concepts of forward kinematics, inverse kinematics, and trajectory planning within the context of a 6 DOF COBOT. These concepts lay the foundation for understanding how a robot perceives and executes movements in its operational space. The objectives include:

- **Forward Kinematics Simulation**

  In this simulation, the COBOT demonstrates its predictive prowess by computing the end-effector's precise position and orientation based on given joint angles. This foundational capability showcases the robot's understanding of its own kinematic structure and its ability to foresee its spatial configuration in response to joint movements.

- **Inverse Kinematics Simulation**

  Investigating COBOT's prowess in inverse kinematics involves assessing its ability to compute the necessary joint angles to achieve a predefined end-effector position and orientation. This simulation showcases the robot's adaptability in solving the complex mathematical problem of determining joint configurations for desired Cartesian coordinates.

- **Trajectory Planning**

  The trajectory planning simulation delves into the optimization of the COBOT's path within its workspace. This involves crafting a trajectory that ensures not only efficient and smooth movements but also adheres to predefined constraints. By exploring various optimization techniques, the COBOT showcases its capability to navigate dynamically changing environments with agility and precision.

- **Operational Space Plotting**

The operational space plotting segment involves visually mapping the COBOT's workspace. Through 3D and 2D projections, this simulation illustrates the robot's reachability set, offering insights into its spatial capabilities. This visualization aids in understanding the robot's operational limits and guides further refinements in motion planning.

- **Analytical and Linear Jacobian Calculation**

The analytical and linear Jacobian calculation simulation delves into the mathematical foundations of COBOT's motion. By computing both the analytical and linear Jacobian matrices, this simulation provides a comprehensive insight into the robot's kinematic properties. Understanding these matrices is crucial for analyzing the robot's sensitivity to changes in joint configurations and optimizing its motion.

- **Joint Velocities and Accelerations Plots**

The joint velocities and accelerations plots offer a dynamic visualization of COBOT's motion along a predefined trajectory. This simulation provides a quantitative assessment of the robot's speed and responsiveness. By examining these plots, one can gauge the smoothness and stability of the COBOT's movements, essential for ensuring safe and efficient operation in real-world scenarios.

By achieving these objectives, the project aims to contribute to the advancement of robotics knowledge, providing insights into the practical application of kinematics and trajectory planning in the realm of COBOTs.

ROBOT DESCRIPTION

The 6 Degrees of Freedom (6 DOF) COBOT selected for this project is characterized by a kinematic structure comprising six interconnected links, each distinguished by specific parameters that define its geometry and kinematics. The lengths of these links, denoted as $L1$, $L2$, $L3$, $L4$, $L5$, and $L6$, play a pivotal role in determining the physical dimensions and range of motion within the robotic system.

In the project setup, each link's length represents the spatial distance between consecutive joints, establishing the foundational structure for COBOT’s kinematics. The lengths are defined as 1.

This length parameter is crucial in creating an accurate and representative model of the robot's physical attributes within the simulation environment.

The robot's kinematic structure is articulated using revolute joints, each defined by a set of parameters encapsulated in **Revolute** objects. The parameters include:

- **d**: Displacement along the joint axis.
- **a**: Link length, representing the distance from the joint to the next joint.
- **alpha**: Twist angle, indicating the angle of rotation about the joint axis.
- **offset**: Joint offset, accounting for any initial angle or displacement.

The choice of revolute joints allows for rotational motion around specified axes, aligning with the typical characteristics of robotic manipulators.

To facilitate the modeling of the COBOT in MATLAB, the **SerialLink** function from the Robotics Toolbox is employed. This function serves as a cornerstone for constructing a comprehensive representation of the robot's kinematic structure. The array of **Revolute** joint objects is passed as input to the **SerialLink** function, resulting in the creation of a model named "6 DOF COBOT UNIVERSAL ROBOTICS MANIPULATOR."

This kinematic model, established through the integration of lengths, revolute joints, and the **SerialLink** function, forms the basis for subsequent simulations and analyses in the project. It provides a robust platform for exploring forward kinematics, inverse kinematics, and trajectory planning algorithms within a simulated environment, aligning with best practices in robotics research and development. The defined parameters and joint characteristics ensure an accurate representation of the COBOT's physical attributes, enabling a thorough investigation of its spatial movements and operational capabilities.

PROCEDURE

1) Simulating Trajectory Planning

Objective:

The primary objective of this phase is to orchestrate a robust trajectory planning mechanism for the 6 Degrees of Freedom (6 DOF) Collaborative Robot (COBOT). The trajectory planning process is foundational in robotics, ensuring precise and efficient maneuvering of the robotic manipulator between predefined points within its operational space. This intricate implementation integrates inverse kinematics, trajectory generation, forward kinematics, and visualization techniques, collectively forming a comprehensive trajectory planning solution.

Process:

Initiating the trajectory planning process involves direct user interaction, a crucial element in the interactive nature of COBOT's operation. The input function serves as the gateway to acquire Cartesian coordinates for user-defined points A, B, and C. These coordinates stand as critical waypoints, defining spatial objectives for COBOT's traversal.

Transitioning from Cartesian coordinates to joint angles constitutes the next stage, guided by the principles of inverse kinematics. The pivotal robot.ikine function facilitates this conversion, enabling the computation of joint configurations aligned with the desired end-effector positions. The 'mask' parameter, applied here, affords full control over all joints, ensuring a comprehensive mapping between Cartesian coordinates and joint angles. This step is foundational in establishing the kinematic relationship governing the COBOT's end-effector and its joint configurations.

With joint configurations for points A, B, and C secured, the trajectory generation process takes center stage. The jtraj function undertakes this responsibility, generating joint space trajectories between the computed configurations for points A to B and B to C. The granularity of the trajectory, influenced by the number of steps (set to 25 in this instance), becomes a critical consideration impacting the smoothness and precision of the COBOT's movement. This parameter plays a pivotal role in achieving optimal trajectory planning for robotic systems.

Transitioning from trajectory generation, forward kinematics becomes the focal point. Leveraging the robot.fkine function, forward kinematics is executed on the combined joint space trajectory. This results in a sequence of transformation matrices, each encapsulating the pose of the end-effector at different junctures along the trajectory. Extracting the translational components from these matrices using the transl function provides Cartesian positions for the end-effector in space.

Visualization emerges as a critical component for trajectory validation and comprehension. MATLAB's plotting functions play a pivotal role in this context. Points A, B, and C are visibly marked on the plot, providing a graphical representation of the COBOT's initial and final positions. Simultaneously, the trajectory is visually depicted through a sequence of robot configurations, each corresponding to a specific point along the trajectory. This visual feedback serves a dual purpose - validating the correctness of the trajectory and offering an intuitive understanding of the COBOT's movement between user-defined points.

In summary, the trajectory planning process unfolds as an intricate interplay of user interaction, inverse kinematics, trajectory generation, forward kinematics, and visualization. This phase lays the groundwork for subsequent explorations into the COBOT's operational space, analytical insights through Jacobian computation, and dynamic analyses involving joint velocities and accelerations.

Functions Used:

- **input:** Obtains user input for Cartesian coordinates.
- **robot.ikine:** Computes inverse kinematics for joint angles.
- **jtraj:** Generates joint space trajectories.
- **robot.fkine and transl:** Performs forward kinematics and extracts translational components.
- **MATLAB plotting functions:** Visualizes robot movement and trajectory.

2) Simulating/Plotting the Operational Workspace

Objective: Explore the operational workspace to comprehend the spatial capabilities of the 6 DOF COBOT by systematically varying joint angles and mapping the reachable space.

Process: Initiate the process by establishing a reachability set, a conceptual container representing end-effector positions for various joint configurations. The size of this set is defined by the linspace function, systematically varying joint angles through nested loops to ensure a thorough exploration of the joint space. Forward kinematics, facilitated by robot.fkine, calculates the transformation matrix and extracts the end-effector position for each joint configuration. This positional data contributes to a comprehensive map of the COBOT's operational workspace.

Visualization: Utilize MATLAB's plotting functions to visually represent the operational workspace through scatter plots in 3D and 2D projections. Each point corresponds to an end-effector position, providing insights into the spatial capabilities along the X, Y, and Z axes. The 3D scatter plot offers a holistic view, while 2D projections focus on specific planes.

Contextual Relevance: This exploration sets the stage for subsequent analytical investigations. Understanding the COBOT's spatial reach is crucial for optimizing its deployment in real-world scenarios, ensuring effective reach to desired locations within its operational envelope. This detailed exploration provides essential context for further analyses, such as Jacobian computation and dynamic analyses.

Functions Used:

- **linspace:** Defines a range of joint angles.
- **Loops in MATLAB:** Iterates over joint angle combinations.
- **robot.fkine:** Computes forward kinematics for end-effector positions.
- **MATLAB plotting functions:** Visualize operational workspace in 3D and 2D projections.

3) Analytical and Linear Jacobian Calculation

Objective: Compute analytical and linear Jacobians to understand joint and end-effector velocity relationships in the 6 DOF COBOT.

Process: Define COBOT's kinematic structure using SerialLink with specified link lengths and joint parameters. Use robot.jacob0 to efficiently calculate the Jacobian matrix for instantaneous motion characteristics.

Advanced Analysis: Configure robot.jacob0 with 'rpy' parameter for analytical Jacobian considering rotational components. Simultaneously, derive the linear Jacobian with 'trans' parameter, focusing on translational components.

Functions Used:

- **SerialLink:** Defines COBOT's kinematic structure.
- **robot.jacob0:** Computes Jacobians for rotational (RPY) and linear components.

4) Joint Velocities and Accelerations Along the Path

Objective: The objective of this phase is to analyze joint velocities and accelerations along a specified trajectory for the 6 DOF COBOT, offering critical insights into its dynamic behavior. This involves generating joint trajectories from point A to B and B to C, utilizing the jtraj function to calculate velocities and accelerations at each trajectory point.

Process: Initiating this phase involves trajectory generation using the jtraj function, yielding joint trajectories, velocities, and accelerations. The combined trajectory from point A to B and B to C forms the basis for dynamic analysis. MATLAB's plotting functions visualize joint velocities and accelerations over time, utilizing subplots for clarity. The first subplot illustrates joint velocities, while the second provides insights into joint accelerations.

Visual Analysis: MATLAB's plotting functions enable a nuanced examination of dynamic behavior. Peaks and troughs in velocity profiles reveal periods of rapid and slow movement, while acceleration profiles highlight moments of significant force application. This visual information is crucial for refining control strategies, optimizing trajectories, and ensuring the COBOT adheres to operational and safety requirements.

Functions Used:

- **jtraj:** Generates joint trajectories, velocities, and accelerations.
- **MATLAB Plotting Functions:** Visualize joint velocities and accelerations.

This analysis acts as a pivotal bridge between trajectory planning and dynamic control, offering engineers and researchers the necessary insights to fine-tune and optimize COBOT’s behavior across diverse operational scenarios.

CURRENT AND FUTURE WORK:

- ***Integration of RRT for Motion Planning:*** Strategically integrating Rapidly-exploring Random Trees (RRT) into trajectory planning reflects a commitment to advanced motion planning algorithms. This technical pursuit involves adapting existing algorithms to seamlessly integrate RRT, addressing challenges like defining tree structures and optimizing collision checks. Beyond implementation, it's a systematic exploration into motion planning nuances guided by the ambition to unravel complexities.
- ***Algorithmic Optimization for Efficiency:*** Ongoing exploration of algorithmic optimization signifies a graduate student's quest for computational elegance. Exploring parallel processing and leveraging GPU capabilities reflects a commitment to understanding the intricate interplay between algorithms and computational efficiency. This journey goes beyond faster execution times, embodying a commitment to mastering algorithmic intricacies and pushing computational boundaries.
- ***Enhancement of Simulink Model:*** Continual refinement of the Simulink model showcases dedication to practical learning. Ongoing efforts involve fine-tuning parameters, introducing advanced dynamics, and ensuring a high-fidelity representation aligned with theoretical underpinnings. This extends beyond simulation, representing a dynamic synergy between theory and application.
- ***Shortcomings and Learning Opportunities:*** While ambitious, the project faced constraints that became valuable learning opportunities. Lack of access to robot libraries for simulating existing robots posed challenges in exploring methods like RRT. Additionally, optimizing for computing time with six links presented challenges with increasing complexity. Despite constraints, the project leveraged simulations and theoretical models for learning. Moving forward involves seeking collaboration for access to physical robots and devising strategies to optimize computational efficiency with increasing link complexity.

RESULTS

The results of the project showcase a successful implementation of trajectory planning, operational workspace exploration, Jacobian analysis, and dynamic behavior assessment for the 6 Degrees of Freedom (6 DOF) Collaborative Robot (COBOT). The trajectory planning phase demonstrated COBOT's ability to navigate between user-defined points in its operational space. Through the use of inverse kinematics, trajectory generation, and forward kinematics, the robot moved seamlessly from point A to B and B to C, as visualized in the plotted trajectories.

```
Enter coordinates for point A [x, y, z]: [0.5, 0.5, 0.5]
Enter coordinates for point B [x, y, z]: [-0.5, -0.5, 0.5]
Enter coordinates for point C [x, y, z]: [0.5, -0.5, -0.5]
```

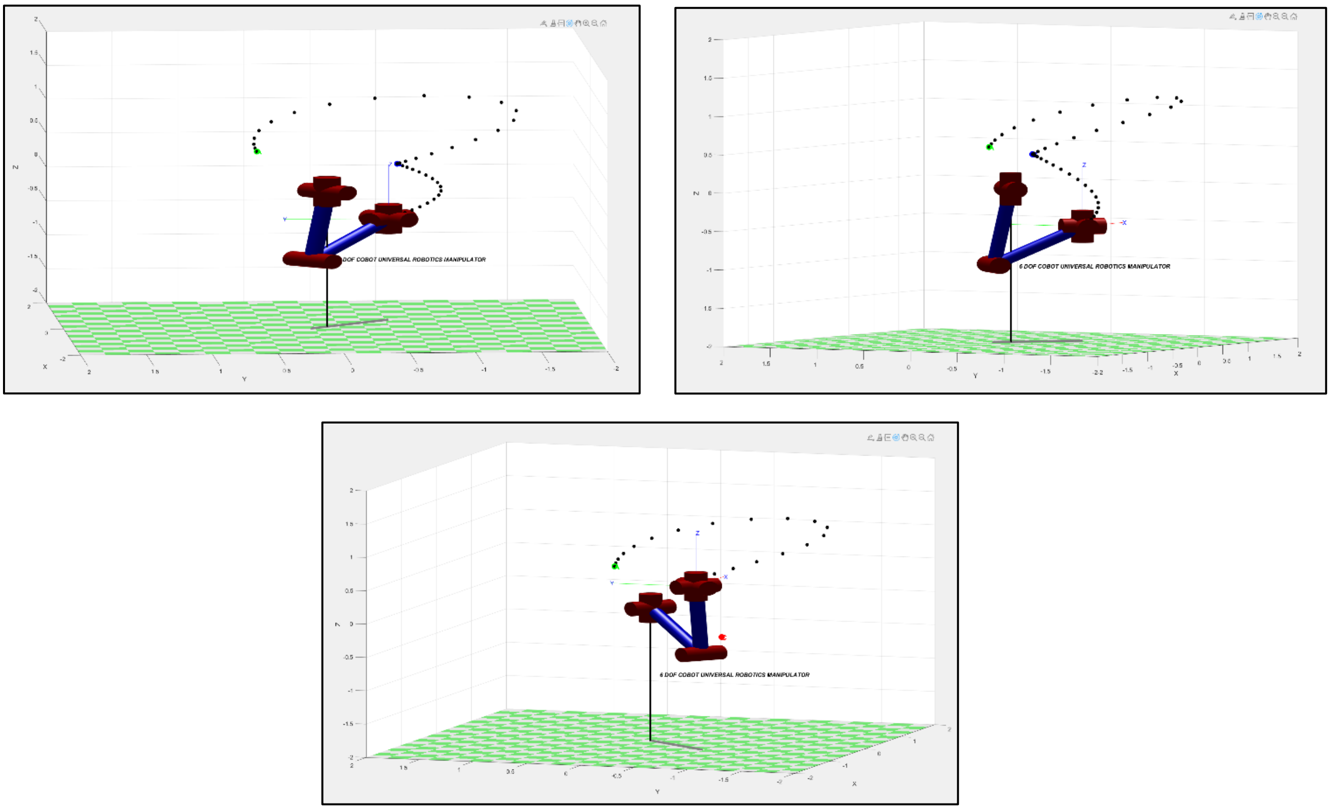

Figure 1. (top) User entered coordinates of 3 points A, B and C. (rest) Show the robot arm moving from A-B-C and showing the movement through the planned trajectory.

The operational workspace exploration provided valuable insights into the spatial capabilities of COBOT. The scatter plots in 3D and 2D projections effectively illustrated the reachability set of the robot, offering a comprehensive understanding of its operational envelope. This exploration serves as a foundational step for informed decision-making in deploying COBOT in diverse real-world scenarios.

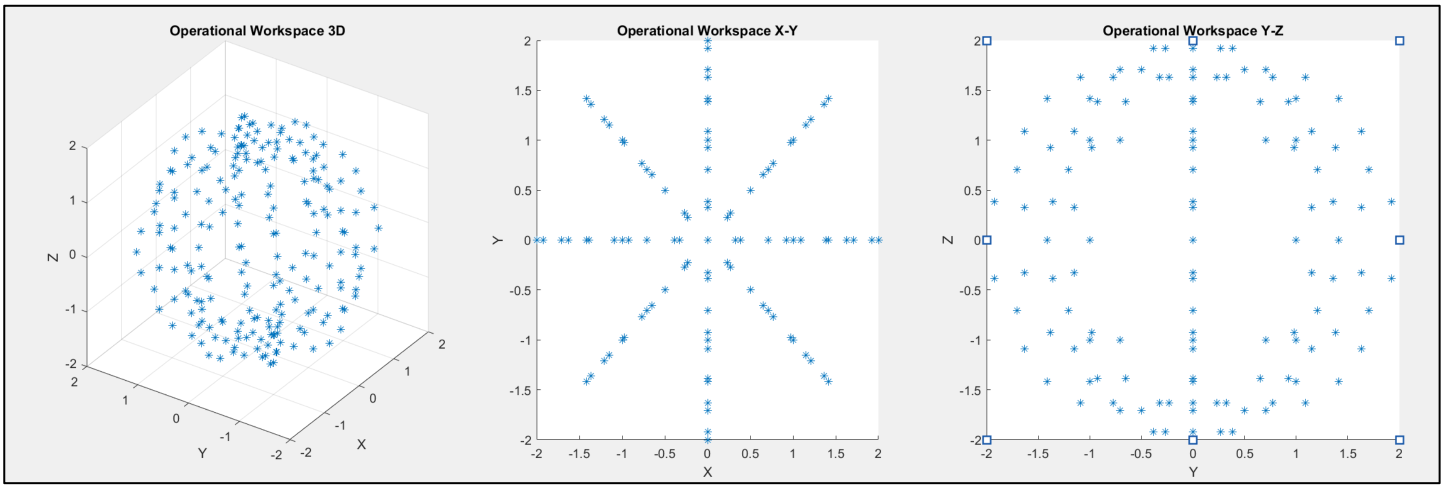

Figure 2. Operational Workspace scatter plot of end effector positions.

The analysis of the analytical and linear Jacobians further enhanced our understanding of COBOT's kinematics. The Jacobian matrices, calculated at a specific joint configuration, revealed the instantaneous motion characteristics of the robot. The inclusion of the rotational components in the analytical Jacobian added depth to the analysis, emphasizing the importance of considering all aspects of motion in robotic systems.

```
Jacobian:
         0         0         0         0         0         0
    2.0000    0.0000    0.0000         0         0         0
         0    2.0000    1.0000         0         0         0
         0         0         0         0         0         0
         0   -1.0000   -1.0000   -1.0000         0   -1.0000
    1.0000    0.0000    0.0000    0.0000    1.0000    0.0000

Linear Jacobian:
         0         0         0         0         0         0
    2.0000    0.0000    0.0000         0         0         0
         0    2.0000    1.0000         0         0         0
```

Figure 3. Analytical and Linear Jacobian matrices

Finally, the examination of joint velocities and accelerations along the specified trajectory provided crucial insights into COBOT's dynamic behavior. Visualizations of velocity and acceleration profiles enabled a nuanced understanding of the forces driving the robot's movement, facilitating optimizations for enhanced performance and control.

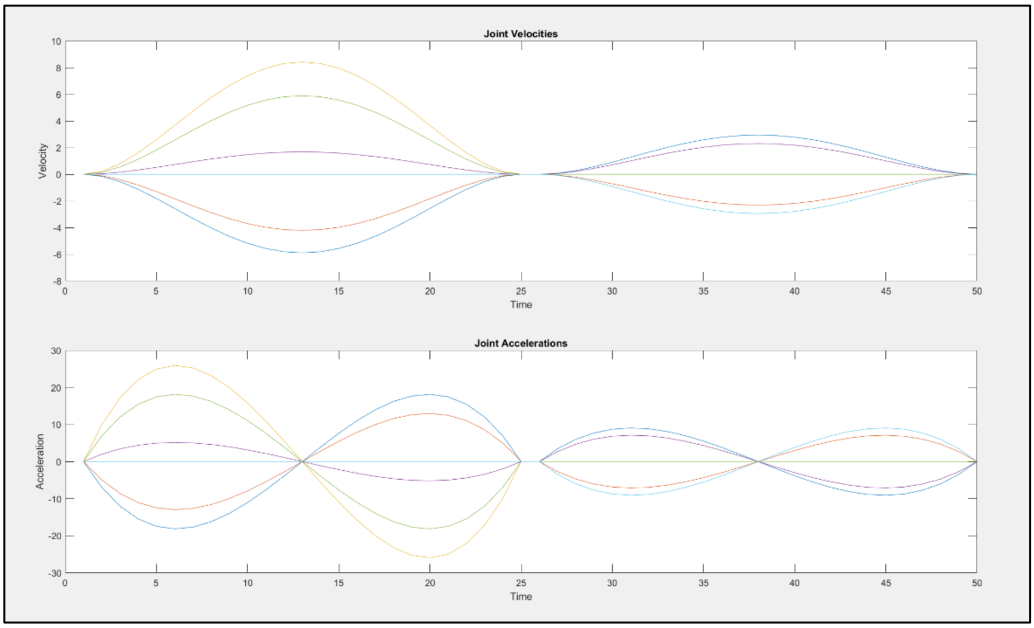

Figure 4. Plots of Joint Velocities and Joint Acceleration against time as the robot end effector moves from A-B-C.

CONCLUSION

In conclusion, the successful implementation of trajectory planning, operational workspace analysis, Jacobian computation, and dynamic behavior evaluation collectively demonstrates the capabilities and potential applications of the 6 DOF COBOT. We believe that the project's outcomes not only provide a solid foundation for future developments in robotics but also offer invaluable insights for enhancing the capabilities of robots.

The trajectory planning mechanism showcased COBOT's ability to perform precise movements between user-defined points. The operational workspace exploration provided a comprehensive understanding of the COBOT's spatial reach, guiding decisions on task assignments and workspace configurations.

The analysis of the analytical and linear Jacobians contributes to the fundamental understanding of COBOT's kinematics, essential for tasks involving manipulation and interaction with the environment. our collective efforts have significantly deepened our understanding of essential concepts in robotics, particularly Inverse Kinematics (IK), Forward

Kinematics (FK), and Jacobian matrices. Through the shared experience of hands-on implementation and simulations, we, as a team, immersed ourselves in the intricacies of these foundational principles, gaining profound insights into their implications for the motion and spatial awareness of robotic systems.

The practical application of trajectory planning became a collaborative endeavor, allowing us to explore optimization techniques collectively and ensure the smooth and efficient movements of the COBOT. This immersive experience has not only strengthened our theoretical foundation but has also fostered a practical intuition for solving complex problems in robotics.

As a group, this project serves as a dynamic learning platform, seamlessly integrating theoretical concepts with real-world applications and empowering us with the collective skills needed to navigate the complexities of collaborative robotics.

REFERENCES

1. MathWorks. (2022). Motion Planning for UR5e Robot Manipulator Using Rapidly-exploring Random Trees (RRT). Retrieved from [https://www.mathworks.com/help/supportpkg/urseries/ug/motion-planning-rbt-ur5e-simulation-manipulatorrrt-example.html]
2. Corke, P. (Year). Robotics Toolbox for MATLAB. [https://petercorke.com/toolboxes/robotics-toolbox/]
3. MathWorks. (Year). MATLAB Documentation. [https://www.mathworks.com/help/]
4. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). Robotics: Modelling, Planning and Control. Springer London. [https://www.springer.com/gp/book/9781849966344]
5. Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control. Pearson/Prentice Hall. [https://www.pearson.com/us/higher-education/program/Craig-Introduction-to-Robotics-Mechanics-and-Control-3rd-Edition/PGM63358.html]
6. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). Robot Modeling and Control. Wiley. [https://www.wiley.com/en-us/Robot+Modeling+and+Control-p-9780471649908]
7. Lynch, K. M., & Park, F. C. (2017). Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press. [http://www.modernrobotics.org/]