Group7 Lab1
Introduction to Dobots

Andy Tsai
ctsai67@asu.edu
1233653410

Lalgudi Shivakumar Ganesh Rathnam
glalgudi@asu.edu
1233693450

Michael Bogale
mabogale@asu.edu
1235712207

March 7, 2025

Abstract

This report presents the development and manipulation of a digital twin for the Dobot Magician Lite using ROS 2, Gazebo, and Rviz2. Forward kinematics were computed and validated in Rviz2, while inverse kinematics were solved using MATLAB and validated using real world robot. The results were cross-verified between simulation and analytical calculations, confirming the model's accuracy for motion planning and control.

1 Introduction

This labratory activity focuses on developing a digital twin of the Dobot Magician Lite robotic arm using desired methods, in our case: ROS 2, Gazebo, and Rviz2. The simulation environment replicates the robot's behavior, allowing for forward and inverse kinematics analysis. Forward kinematics determine the end-effector position given joint angles, while inverse kinematics compute feasible joint angles for a desired end-effector position. MATLAB was used for solving inverse kinematics, and results were verified through simulation to ensure accuracy. This study demonstrates the importance of digital twins in robotic motion planning and control.

2 Method

We followed six major steps to complete this laboratory activity.

1. **Tutorials**: The first and most crucial step was reviewing the necessary tutorials. This provided a fundamental understanding of various approaches to implementing robot simulation.

2. **Model Recreation**: Using the *Create a URDF with ROS2* tutorial and the provided CAD file, we created the Dobot Magician Lite model in a URDF file. The model was then exported as a folder containing the URDF file and other necessary components for ROS2 simulation.

3. **ROS2 Environment Setup**: After generating the required files, a proper ROS2 workspace was configured. First, ROS2 and essential software such as Rviz2 and Gazebo were installed. The generated folder (containing the URDF file) was moved to `~/workspace/src/`, and key configuration files (e.g., `CMakeLists.txt`, `package.xml`, and the URDF file) were modified to properly define the robot model and build the workspace.

4. **Robot Simulation Implementation**: The workspace was built using `colcon build`, sourced, and then the simulation was launched in Rviz2 and Gazebo.

5. **Robot Control**: A Python script was developed to control the robot's movement. The script subscribed to the end-effector node to retrieve its final position.

6. **Inverse Kinematics**: Mathematical equations defining the relationship between joint angles were formulated and solved using MATLAB. The results were validated by comparing them with the actual Dobot Magician Lite.

3 Results

1. **Tutorials**: After completing the tutorials, we chose **ROS2** as our platform for implementing the simulation.

2. **Model Recreation**: Although this step seemed straightforward, finalizing the **URDF file** took several days. The primary challenge was that the provided CAD file was in **STEP format**, which is a fixed model that does not allow modifications. To address this, we reconstructed an **assembly file** using individual model parts and redefined the relationships between **links, joints, and coordinate frames**.

3. **ROS2 Environment Setup**: The **ROS2 environment** was set up on **Ubuntu 22.04** to support simulation and development.

4. **Robot Simulation Implementation**: The **Dobot Magician Lite** model was successfully simulated (Figure 1). Using **Rviz2**, we obtained the coordinates of each joint (Figure 2) and visualized the **link relationships** (Figure 3).

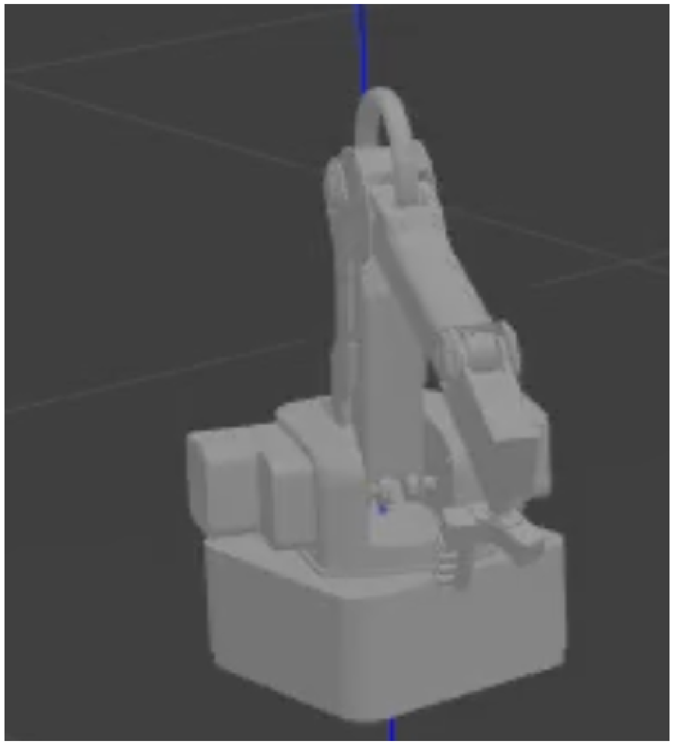

Figure 1: Simulation model of the Dobot Magician Lite.

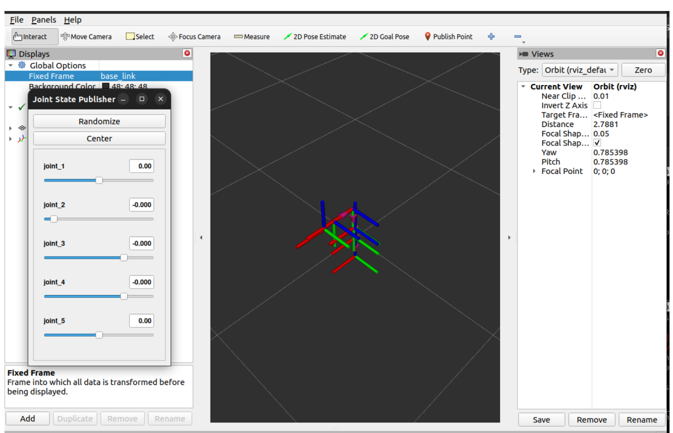

Figure 2: Joint coordinate frames obtained from Rviz2.

5. **Robot Control**: Forward kinematics were implemented using **Python scripts**, which subscribed to the **end-effector node** to extract its position and orientation (video included in the attached files).

6. **Inverse Kinematics**: The **inverse kinematics** model was verified through **MATLAB calculations**, and the results were validated with the **Dobot Magician Lite** (video included in the attached files).

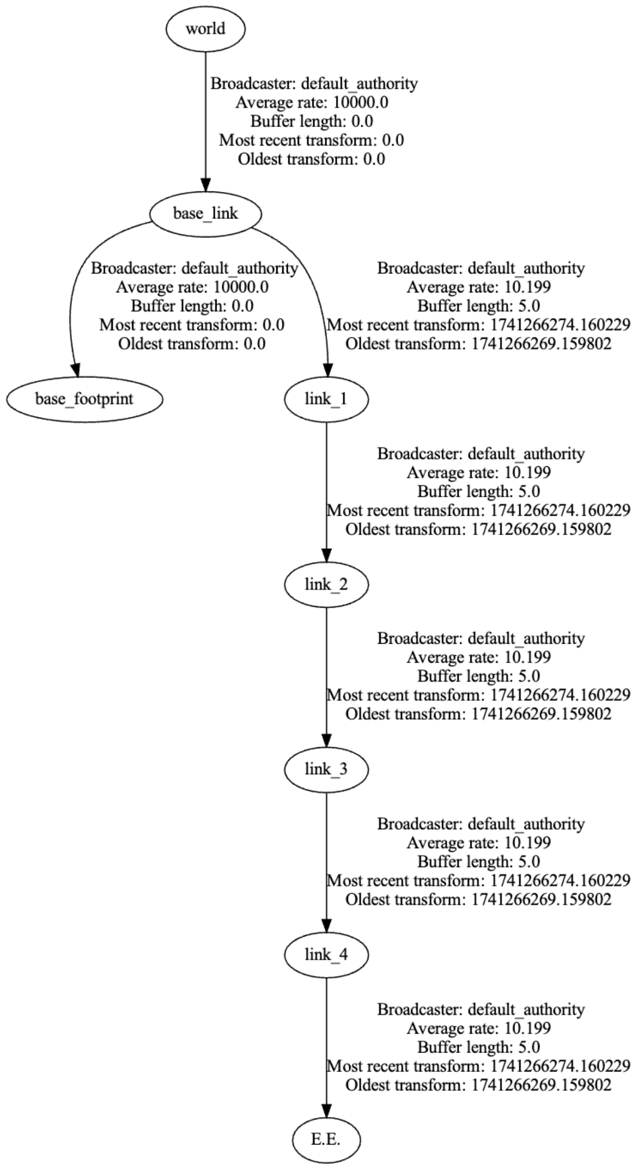

Figure 3: Link relationships within the model.

4 Conclusion

This lab successfully developed a digital twin of the Dobot Magician Lite, integrating ROS 2, Gazebo, Rviz2, and MATLAB to validate forward and inverse kinematics. The simulated results were consistent with analytical solutions, confirming the accuracy of the kinematic models. This approach enhances robotic system development by enabling pre-deployment validation, reducing errors in physical implementation. Future work may explore real-time control integration to further bridge the gap between simulation and real-world execution.

References

[1] Course Materials

[2] Create a URDF with ROS2

[3] Learning ROS basics

[4] Create a URDF with ROS2

[5] Tutorial on simulating robotic arm with solidworks and ROS