**Project Report – Centaur Right Arm**

Overview:

This project is a robotic system designed for simulation and motion planning using **ROS2** **(Humble)** and **Gazebo**. The project uses **MoveIt2** for trajectory and motion planning, **KDL** for dynamics, and **Gazebo** for simulation. The repository follows a **colcon** **ROS2** **workspace** structure, containing packages for robot description, configuration, and motion planning.

Dependencies:

The project has been tested on **Ubuntu 22.04.3 LTS** and depends on the following software:

1. **ROS2 (humble)** - Framework for robot operation

Installation - <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html>

1. **Gazebo** **Classic**- Robot simulation environment

* Note: Gazebo Classic goes **EOL in Jan 2025**, install the latest version supported for ROS Humble.

1. **KDL** - Used for dynamics and motion interpolation
2. **MoveIt2** - Used for motion planning

Installation -<https://moveit.picknik.ai/main/doc/tutorials/getting_started/getting_started.html>

Repository Structure:

This repository is organized as a **colcon ROS2 workspace**, consisting of the following packages:

**diablo\_bot**

* *config/* – Configuration files for dynamics, controller manager, and kinematics solver.
* *description*/ – URDF description, ROS controllers, and Gazebo plugin files.
* *meshes/* - STL files for visualization in Gazebo and Rviz.
* *scenes/* – Includes scene objects for simulation.
* *launch/* – Launch files for Gazebo and Rviz.

**moveit\_config\_diablo**

* *config/* – Includes SRDF, URDF, KDL kinematics solver, and helper methods for MoveIt2.
* *launch/* – Launch files for MoveIt2 motion planning.

Installation & Setup

1. Install **ROS2** **Humble** by following the official installation guide.
2. Install **Gazebo** (latest version supported for ROS Humble).
3. Install **MoveIt2** using the provided installation guide.
4. Clone the repository and build the workspace using colcon

git clone git@github.com:Mystique03/diablo\_ws.git

cd diablo\_ws

colcon build --symlink-install

source install/setup.bash

Usage:

1. **Visualize Robot in Rviz:**

# Terminal 1

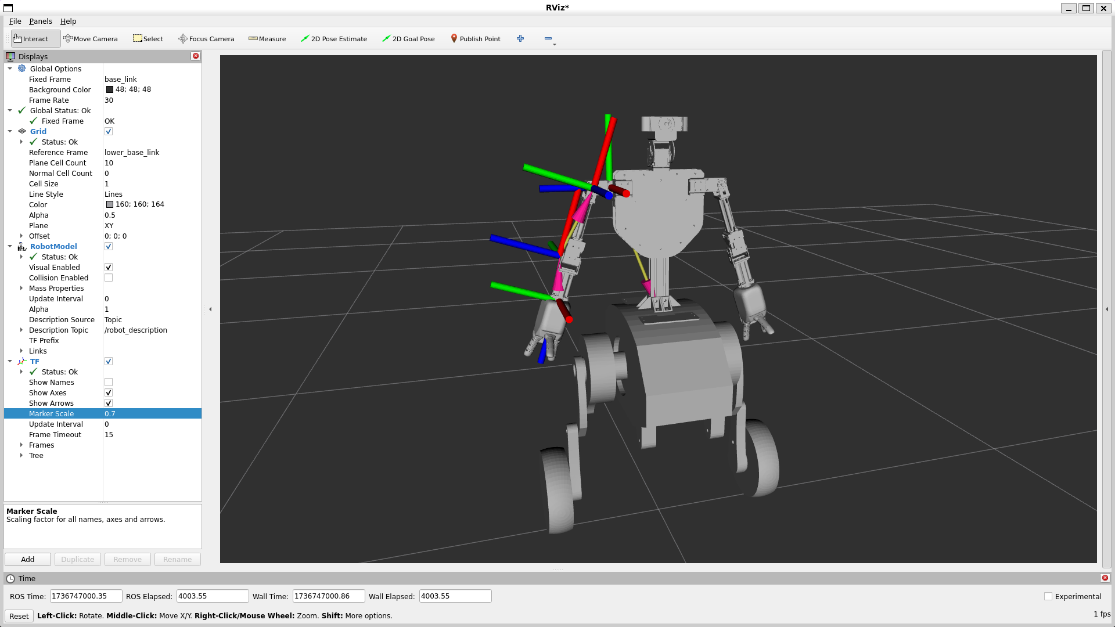
ros2 launch diablo\_bot rsp.launch.py

# Terminal 2

rviz2

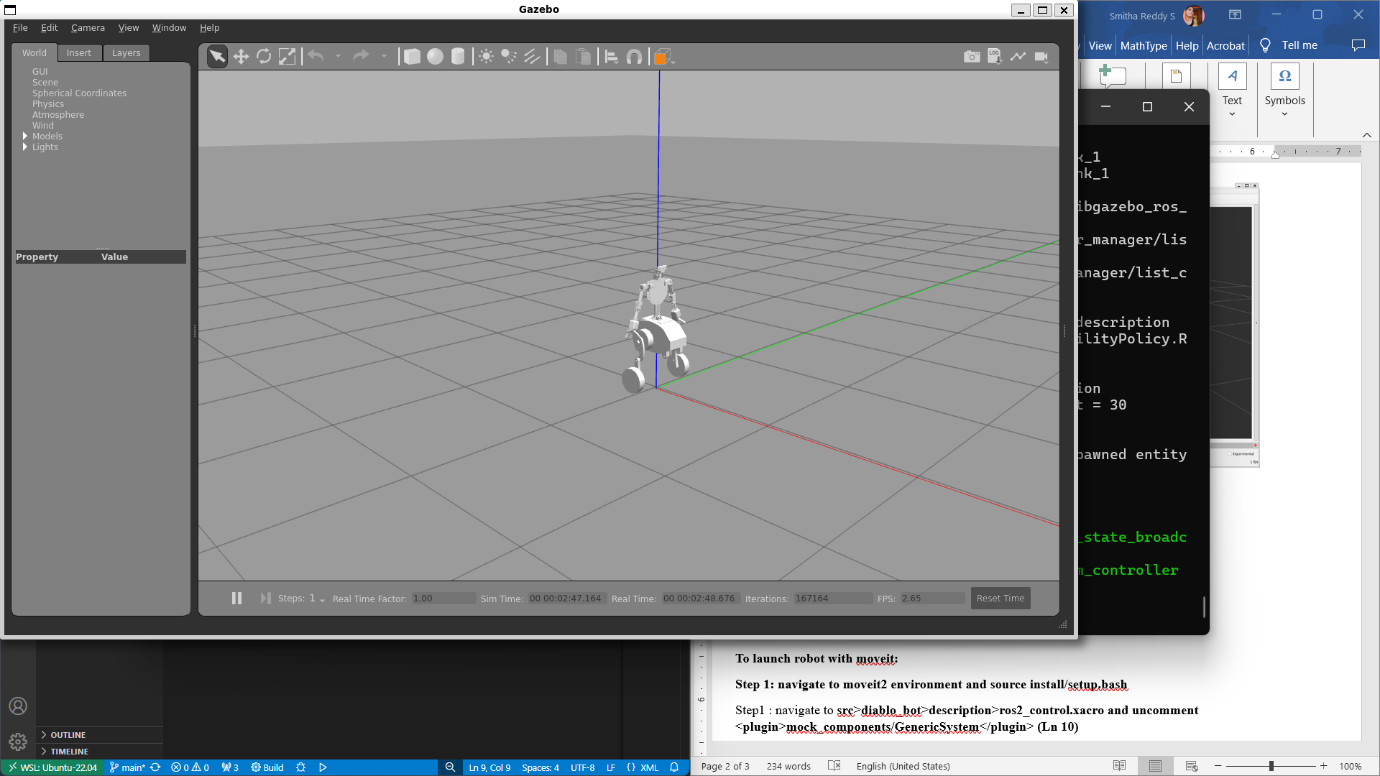
# Terminal 3

ros2 run joint\_state\_publisher\_gui joint\_state\_publisher\_gui



1. **Visualize Robot in Gazebo:**

ros2 launch diablo\_bot gz.launch.py



1. **To Launch Robot with ROS Control:**

Step 1 : Navigate to diablo\_bot>description>ros2\_control.xacro and uncomment <plugin>gazebo\_ros2\_control/GazeboSystem</plugin> (Ln 7)

Step 2: Launch the robot with ROS Control

ros2 launch diablo\_bot diablo.launch.py

1. **To Launch Robot with Moveit:**

Step 1: Source MoveIt2 environment:

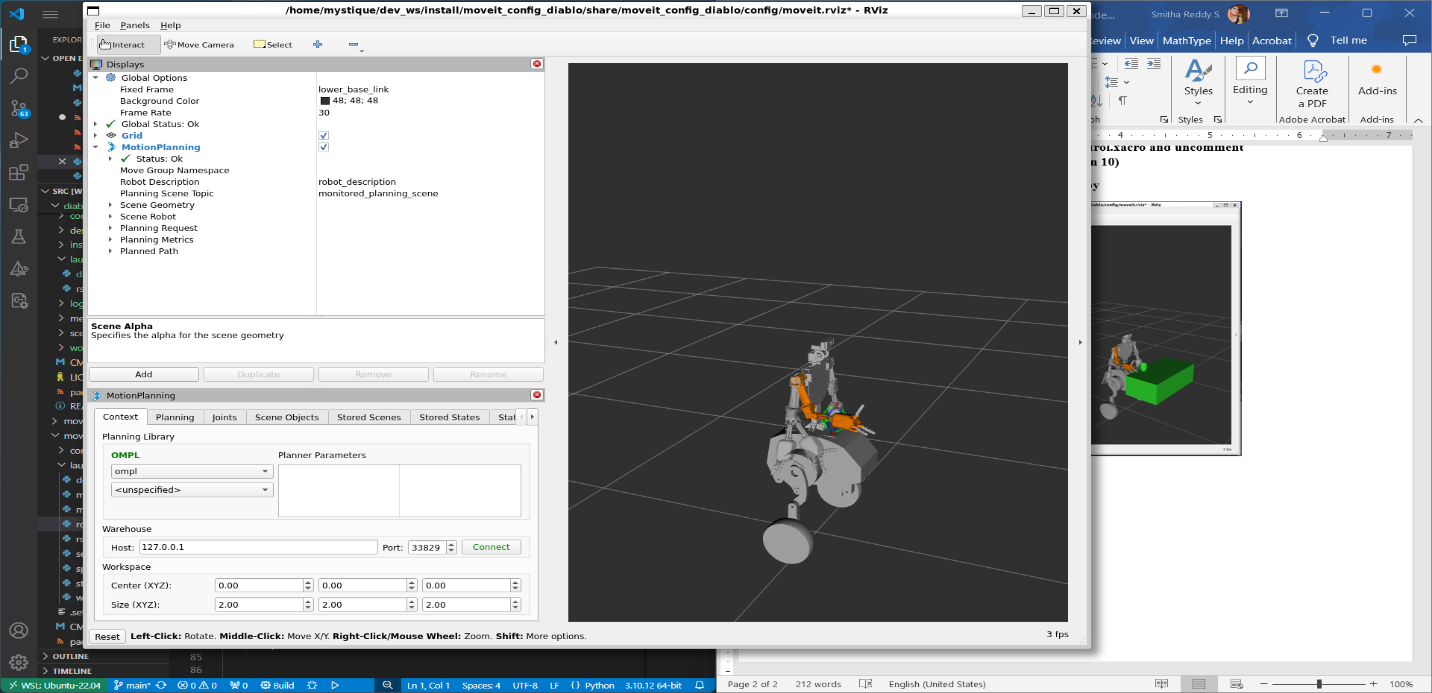
source <your\_moveit2\_package>/install/setup.bash

Step 2: Navigate to diablo\_bot/description/ros2\_control.xacro and uncomment

<plugin>mock\_components/GenericSystem</plugin>

Step 3: Launch MoveIt2

ros2 launch moveit\_config\_diablo demo.launch.py



1. **Launch Joint State Publisher GUI with Gazebo**

# Terminal 1

ros2 run diablo\_bot diablo.launch.py

# Terminal 2

ros2 run diablo\_bot js\_to\_trajectory.py

# Terminal 3

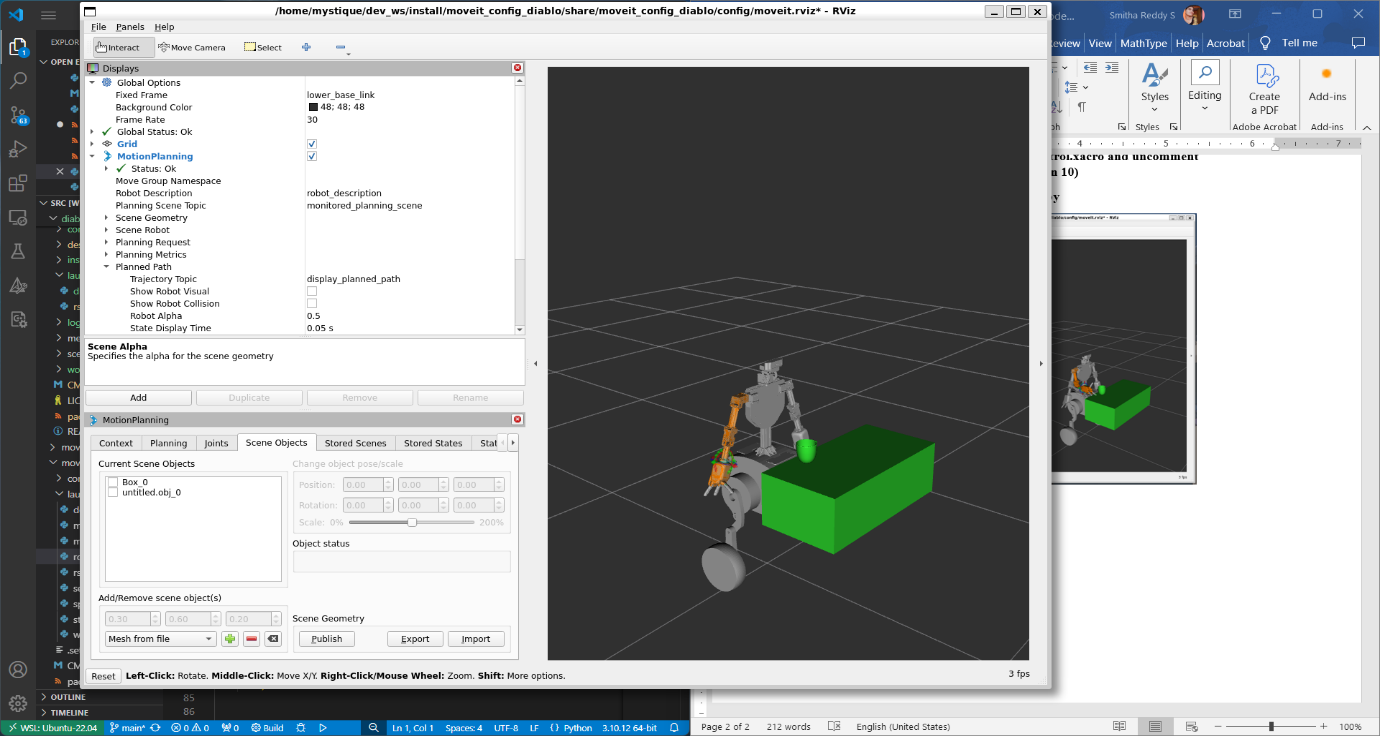
ros2 run joint\_state\_publisher\_gui joint\_state\_publisher\_gui

1. **Launch Table & Mug Scene:**

Step 1: launch robot using above moveit commands

Step 2: In **Rviz**, Under **Scene Objects** import the scene

diablo\_bot>scenes>coffee\_mug.scene



1. **Run Inverse Kinematics**

# Terminal 1

ros2 run diablo\_bot diablo.launch.py

# Terminal 2

ros2 run diablo\_bot ikSolver.py

# Terminal 3

ros2 topic pub /desired\_end\_effector\_position geometry\_msgs/msg/Point "{x: 0.0, y: 0.0, z: 0.0}"

1. **Run Forward Kinematics**

# Terminal 1

ros2 run diablo\_bot diablo.launch.py

# Terminal 2

ros2 run diablo\_bot dh\_fkSolver.py

1. **Modifying MoveIt Setup**

Step 1 : Source the Moveit environment

source moveit/install/setup.bash

Step 2: Run Moveit Setup Assistant

ros2 run moveit\_setup\_assistant moveit\_setup\_assistant