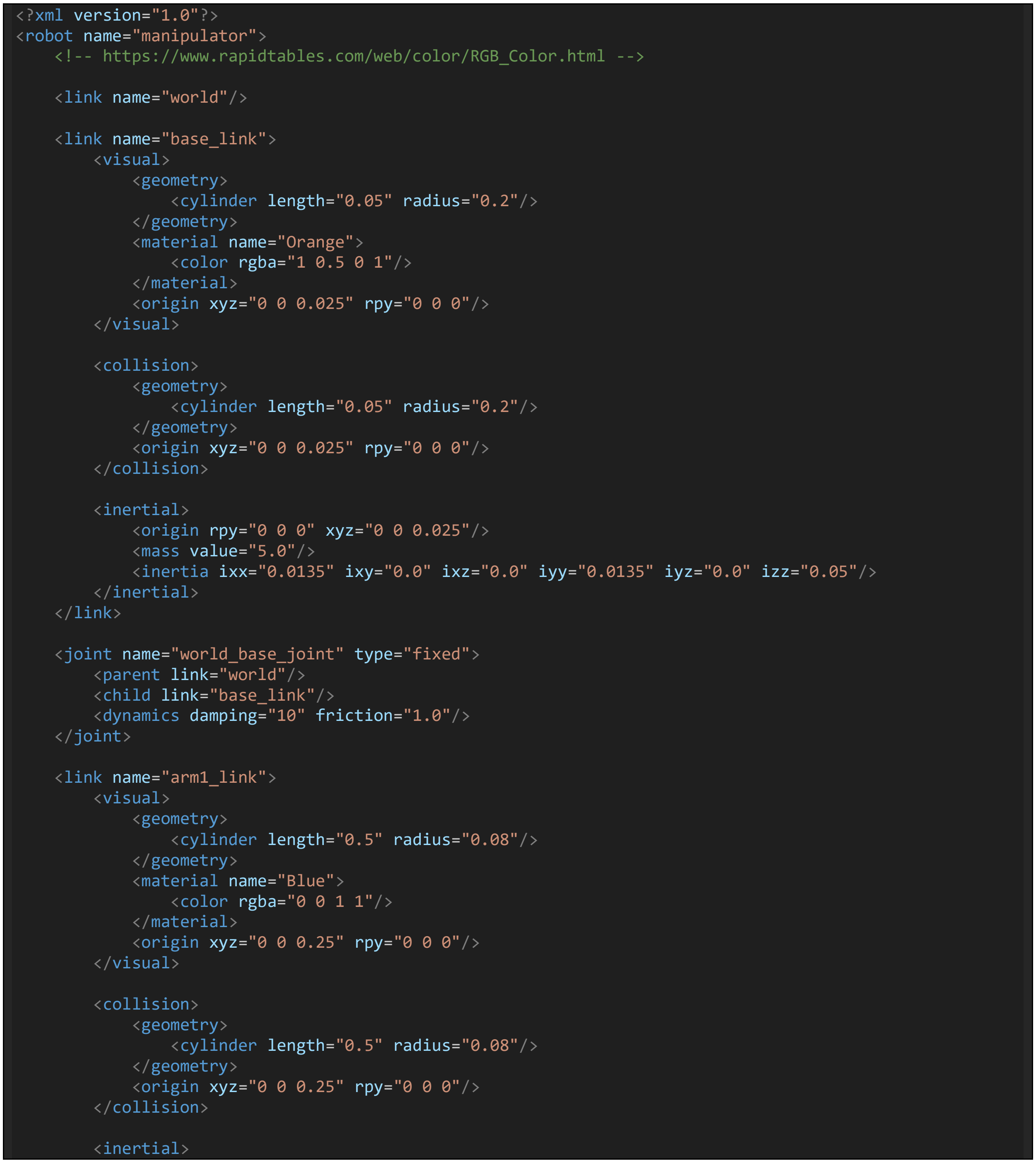
Lab 5 – Report

Meet Kansara – 220929270 Roll no. 54

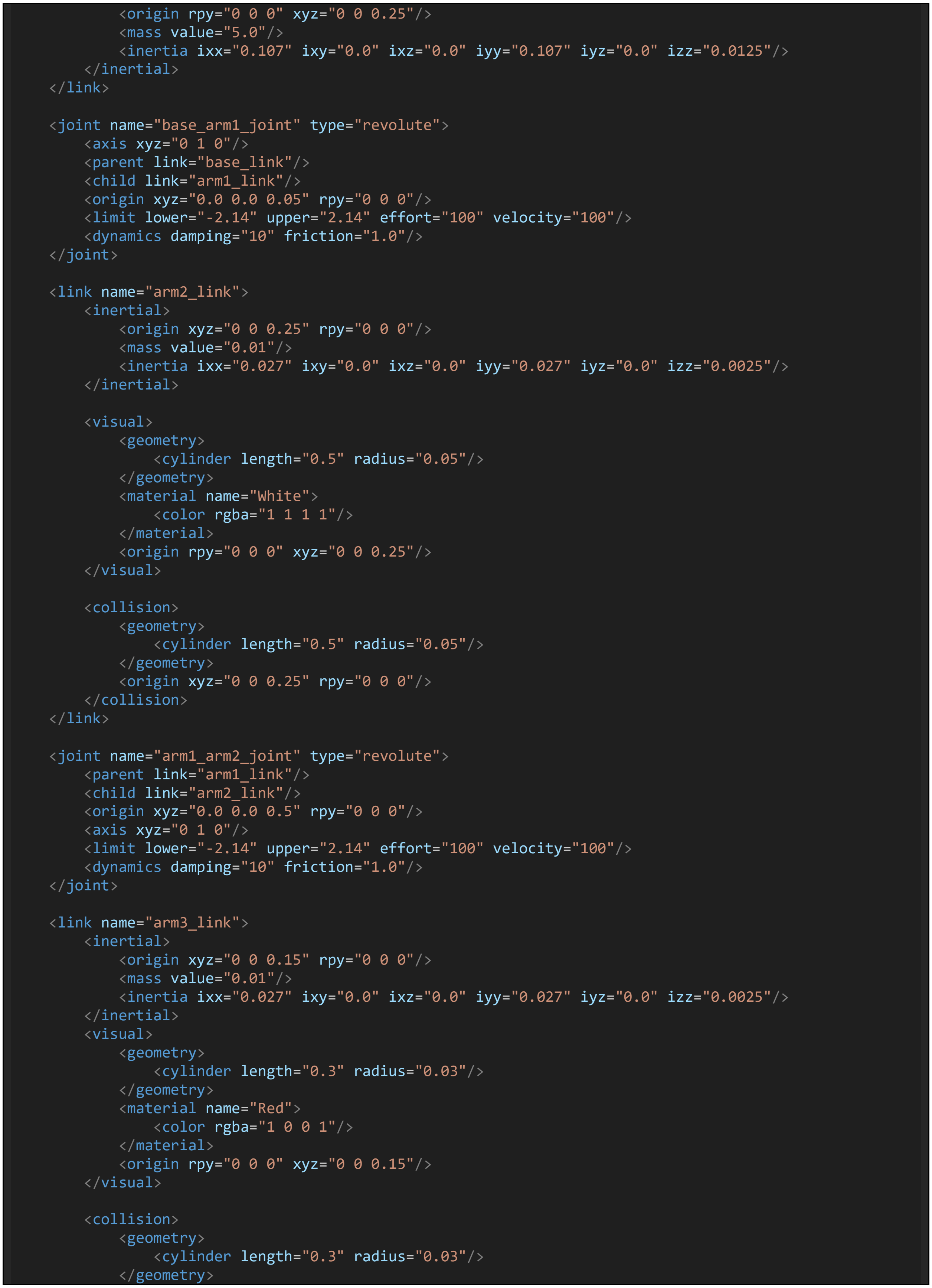
Aim: To simulate a manipulator using forward kinematics controllers in Rviz and Gazebo.

Code Execution and analysis:

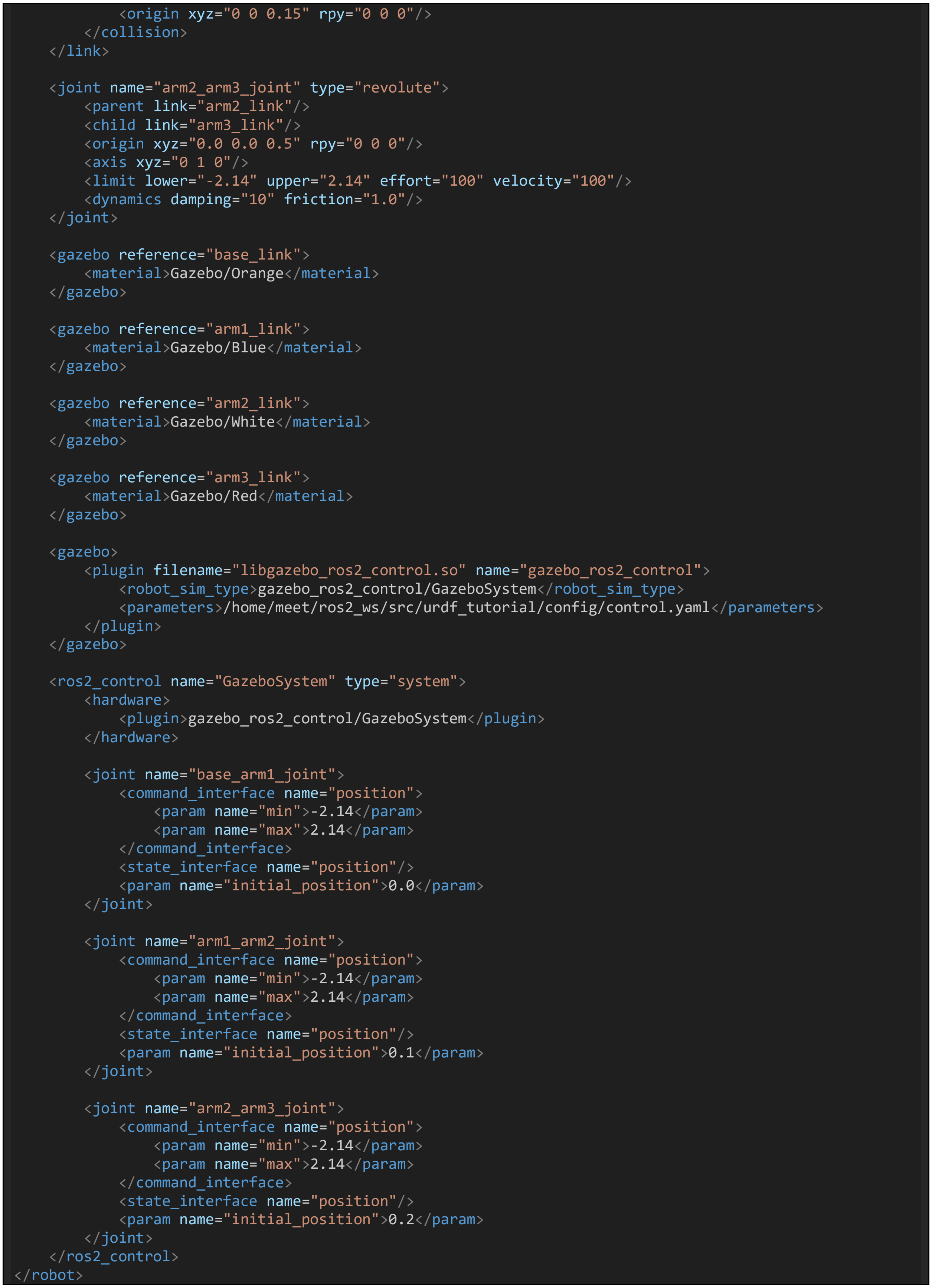
URDF file:



```
<?xml version="1.0"?>
<robot name="manipulator">
    <!-- https://www.rapidtables.com/web/color/RGB_Color.html -->

    <link name="world"/>

    <link name="base_link">
        <visual>
            <geometry>
                <cylinder length="0.05" radius="0.2"/>
            </geometry>
            <material name="Orange">
                <color rgba="1 0.5 0 1"/>
            </material>
            <origin xyz="0 0 0.025" rpy="0 0 0"/>
        </visual>

        <collision>
            <geometry>
                <cylinder length="0.05" radius="0.2"/>
            </geometry>
            <origin xyz="0 0 0.025" rpy="0 0 0"/>
        </collision>

        <inertial>
            <origin rpy="0 0 0" xyz="0 0 0.025"/>
            <mass value="5.0"/>
            <inertia ixx="0.0135" ixy="0.0" ixz="0.0" iyy="0.0135" iyz="0.0" izz="0.05"/>
        </inertial>
    </link>

    <joint name="world_base_joint" type="fixed">
        <parent link="world"/>
        <child link="base_link"/>
        <dynamics damping="10" friction="1.0"/>
    </joint>

    <link name="arm1_link">
        <visual>
            <geometry>
                <cylinder length="0.5" radius="0.08"/>
            </geometry>
            <material name="Blue">
                <color rgba="0 0 1 1"/>
            </material>
            <origin xyz="0 0 0.25" rpy="0 0 0"/>
        </visual>

        <collision>
            <geometry>
                <cylinder length="0.5" radius="0.08"/>
            </geometry>
            <origin xyz="0 0 0.25" rpy="0 0 0"/>
        </collision>

        <inertial>
```

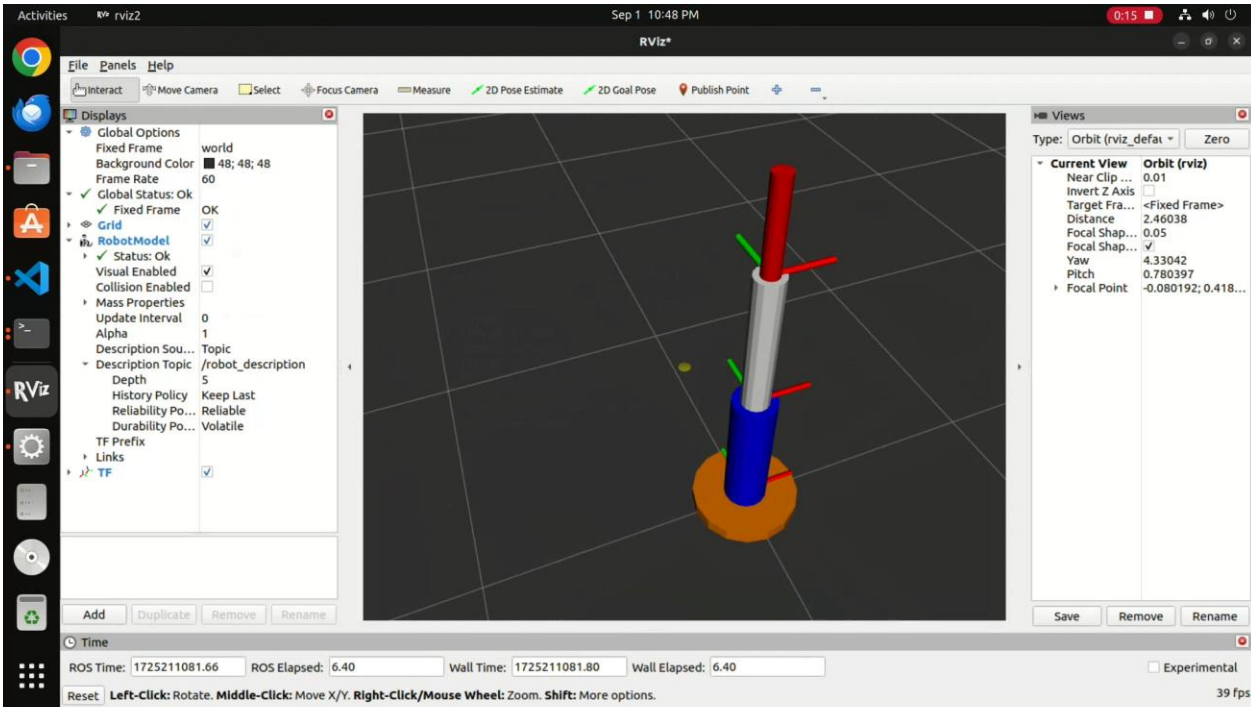


```
            <origin rpy="0 0 0" xyz="0 0 0.25"/>
            <mass value="5.0"/>
            <inertia ixx="0.107" ixy="0.0" ixz="0.0" iyy="0.107" iyz="0.0" izz="0.0125"/>
        </inertial>
    </link>

    <joint name="base_arm1_joint" type="revolute">
        <axis xyz="0 1 0"/>
        <parent link="base_link"/>
        <child link="arm1_link"/>
        <origin xyz="0.0 0.0 0.05" rpy="0 0 0"/>
        <limit lower="-2.14" upper="2.14" effort="100" velocity="100"/>
        <dynamics damping="10" friction="1.0"/>
    </joint>

    <link name="arm2_link">
        <inertial>
            <origin xyz="0 0 0.25" rpy="0 0 0"/>
            <mass value="0.01"/>
            <inertia ixx="0.027" ixy="0.0" ixz="0.0" iyy="0.027" iyz="0.0" izz="0.0025"/>
        </inertial>

        <visual>
            <geometry>
                <cylinder length="0.5" radius="0.05"/>
            </geometry>
            <material name="White">
                <color rgba="1 1 1 1"/>
            </material>
            <origin rpy="0 0 0" xyz="0 0 0.25"/>
        </visual>

        <collision>
            <geometry>
                <cylinder length="0.5" radius="0.05"/>
            </geometry>
            <origin xyz="0 0 0.25" rpy="0 0 0"/>
        </collision>
    </link>

    <joint name="arm1_arm2_joint" type="revolute">
        <parent link="arm1_link"/>
        <child link="arm2_link"/>
        <origin xyz="0.0 0.0 0.5" rpy="0 0 0"/>
        <axis xyz="0 1 0"/>
        <limit lower="-2.14" upper="2.14" effort="100" velocity="100"/>
        <dynamics damping="10" friction="1.0"/>
    </joint>

    <link name="arm3_link">
        <inertial>
            <origin xyz="0 0 0.15" rpy="0 0 0"/>
            <mass value="0.01"/>
            <inertia ixx="0.027" ixy="0.0" ixz="0.0" iyy="0.027" iyz="0.0" izz="0.0025"/>
        </inertial>
        <visual>
            <geometry>
                <cylinder length="0.3" radius="0.03"/>
            </geometry>
            <material name="Red">
                <color rgba="1 0 0 1"/>
            </material>
            <origin rpy="0 0 0" xyz="0 0 0.15"/>
        </visual>

        <collision>
            <geometry>
                <cylinder length="0.3" radius="0.03"/>
            </geometry>
```



```
            <origin xyz="0 0 0.15" rpy="0 0 0"/>
        </collision>
    </link>

    <joint name="arm2_arm3_joint" type="revolute">
        <parent link="arm2_link"/>
        <child link="arm3_link"/>
        <origin xyz="0.0 0.0 0.5" rpy="0 0 0"/>
        <axis xyz="0 1 0"/>
        <limit lower="-2.14" upper="2.14" effort="100" velocity="100"/>
        <dynamics damping="10" friction="1.0"/>
    </joint>

    <gazebo reference="base_link">
        <material>Gazebo/Orange</material>
    </gazebo>

    <gazebo reference="arm1_link">
        <material>Gazebo/Blue</material>
    </gazebo>

    <gazebo reference="arm2_link">
        <material>Gazebo/White</material>
    </gazebo>

    <gazebo reference="arm3_link">
        <material>Gazebo/Red</material>
    </gazebo>

    <gazebo>
        <plugin filename="libgazebo_ros2_control.so" name="gazebo_ros2_control">
            <robot_sim_type>gazebo_ros2_control/GazeboSystem</robot_sim_type>
            <parameters>/home/meet/ros2_ws/src/urdf_tutorial/config/control.yaml</parameters>
        </plugin>
    </gazebo>

    <ros2_control name="GazeboSystem" type="system">
        <hardware>
            <plugin>gazebo_ros2_control/GazeboSystem</plugin>
        </hardware>

        <joint name="base_arm1_joint">
            <command_interface name="position">
                <param name="min">-2.14</param>
                <param name="max">2.14</param>
            </command_interface>
            <state_interface name="position"/>
            <param name="initial_position">0.0</param>
        </joint>

        <joint name="arm1_arm2_joint">
            <command_interface name="position">
                <param name="min">-2.14</param>
                <param name="max">2.14</param>
            </command_interface>
            <state_interface name="position"/>
            <param name="initial_position">0.1</param>
        </joint>

        <joint name="arm2_arm3_joint">
            <command_interface name="position">
                <param name="min">-2.14</param>
                <param name="max">2.14</param>
            </command_interface>
            <state_interface name="position"/>
            <param name="initial_position">0.2</param>
        </joint>
    </ros2_control>
</robot>
```

Manipulator model in RViz:



Global Options	
Fixed Frame	world
Background Color	48; 48; 48
Frame Rate	60
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
RobotModel	☑
✓ Status: Ok	
Visual Enabled	☑
Collision Enabled	☐
Mass Properties	
Update Interval	0
Alpha	1
Description Sou…	Topic
Description Topic	/robot_description
Depth	5
History Policy	Keep Last
Reliability Po…	Reliable
Durability Po…	Volatile
TF Prefix	
Links	
TF	☑

Type: Orbit (rviz_defau

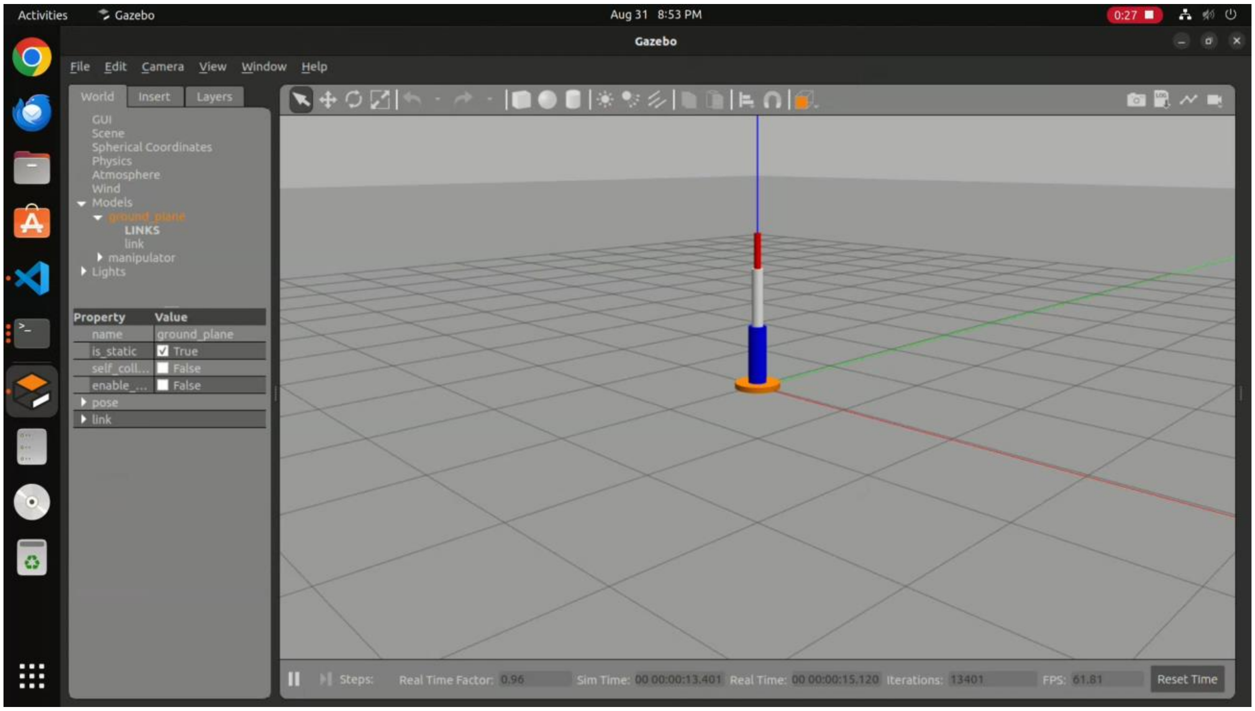
Current View	Orbit (rviz)
Near Clip …	0.01
Invert Z Axis	☐
Target Fra…	<Fixed Frame>
Distance	2.46038
Focal Shap…	0.05
Focal Shap…	☑
Yaw	4.33042
Pitch	0.780397
Focal Point	-0.080192; 0.418…

ROS Time: 1725211081.66 ROS Elapsed: 6.40 Wall Time: 1725211081.80 Wall Elapsed: 6.40 ☐ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:** Zoom. **Shift:** More options.

39 fps

Manipulator model in Gazebo:

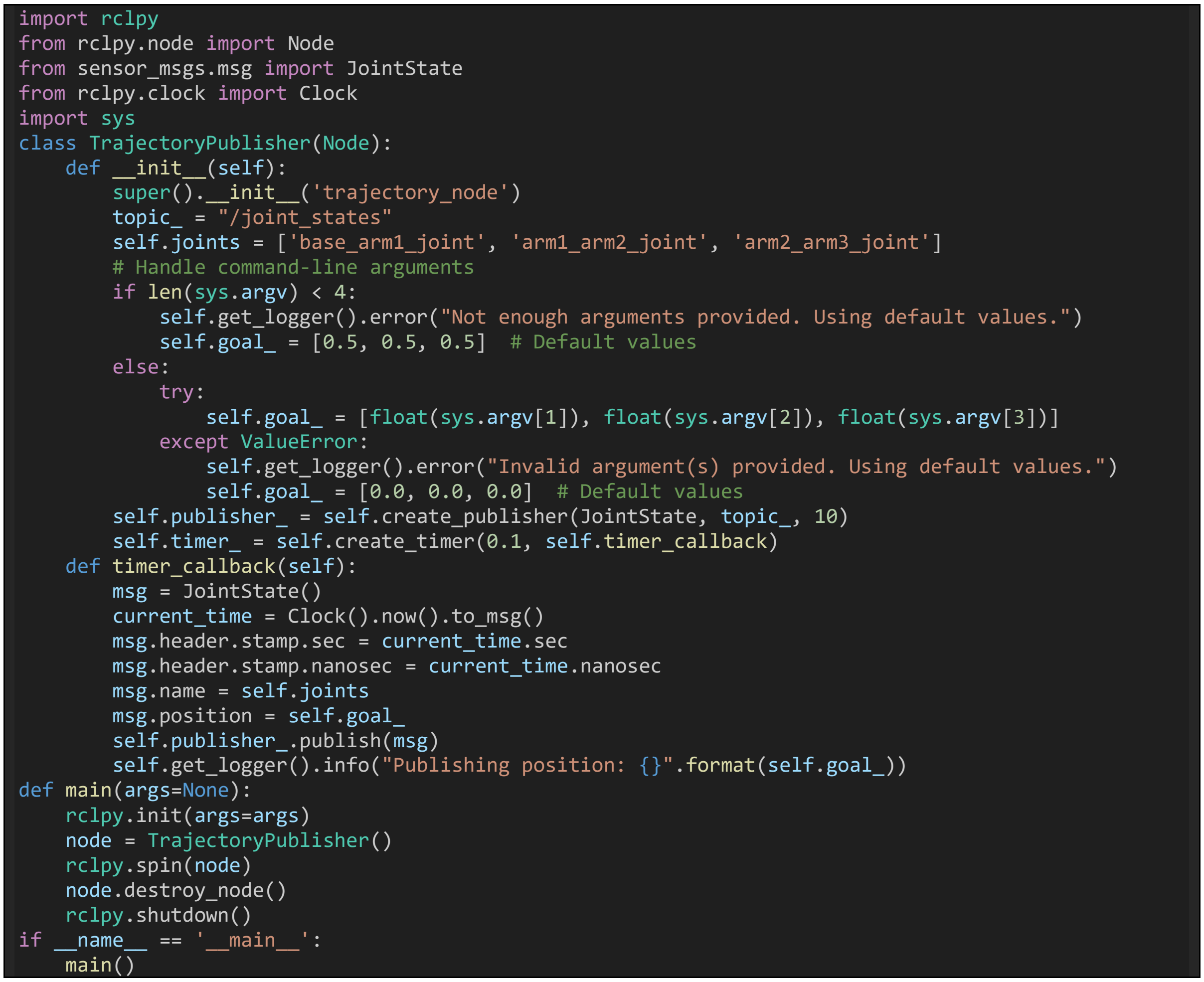


GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
manipulator
Lights

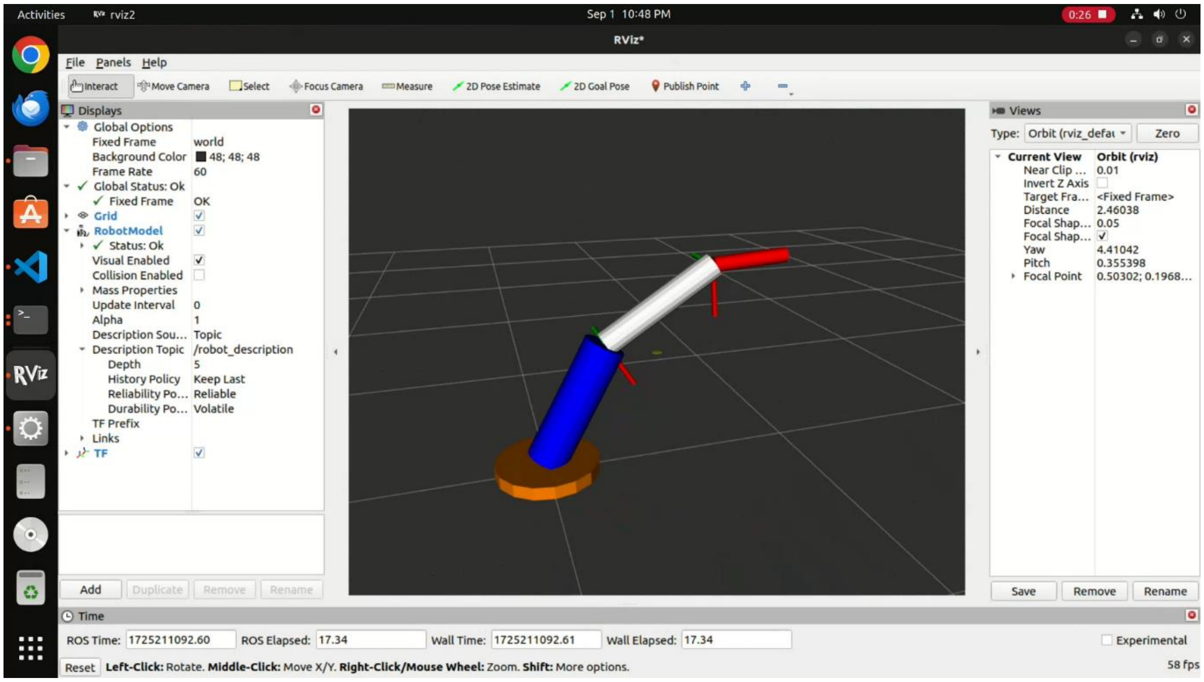
Property	Value
name	ground_plane
is_static	☑ True
self_coll…	☐ False
enable_…	☐ False
pose	
link	

Steps: Real Time Factor: 0.96 Sim Time: 00 00:00:13.401 Real Time: 00 00:00:15.120 Iterations: 13401 FPS: 61.81

1. Forward kinematics controller for manipulator in RViz:



```python
import rclpy
from rclpy.node import Node
from sensor_msgs.msg import JointState
from rclpy.clock import Clock
import sys
class TrajectoryPublisher(Node):
    def __init__(self):
        super().__init__('trajectory_node')
        topic_ = "/joint_states"
        self.joints = ['base_arm1_joint', 'arm1_arm2_joint', 'arm2_arm3_joint']
        # Handle command-line arguments
        if len(sys.argv) < 4:
            self.get_logger().error("Not enough arguments provided. Using default values.")
            self.goal_ = [0.5, 0.5, 0.5]  # Default values
        else:
            try:
                self.goal_ = [float(sys.argv[1]), float(sys.argv[2]), float(sys.argv[3])]
            except ValueError:
                self.get_logger().error("Invalid argument(s) provided. Using default values.")
                self.goal_ = [0.0, 0.0, 0.0]  # Default values
        self.publisher_ = self.create_publisher(JointState, topic_, 10)
        self.timer_ = self.create_timer(0.1, self.timer_callback)
    def timer_callback(self):
        msg = JointState()
        current_time = Clock().now().to_msg()
        msg.header.stamp.sec = current_time.sec
        msg.header.stamp.nanosec = current_time.nanosec
        msg.name = self.joints
        msg.position = self.goal_
        self.publisher_.publish(msg)
        self.get_logger().info("Publishing position: {}".format(self.goal_))
def main(args=None):
    rclpy.init(args=args)
    node = TrajectoryPublisher()
    rclpy.spin(node)
    node.destroy_node()
    rclpy.shutdown()
if __name__ == '__main__':
    main()
```



Global Options
Fixed Frame: world
Background Color: 48; 48; 48
Frame Rate: 60
✓ Global Status: Ok
✓ Fixed Frame: OK
Grid ☑
RobotModel ☑
✓ Status: Ok
Visual Enabled ☑
Collision Enabled □
Mass Properties
Update Interval: 0
Alpha: 1
Description Sou...: Topic
Description Topic: /robot_description
Depth: 5
History Policy: Keep Last
Reliability Po...: Reliable
Durability Po...: Volatile
TF Prefix
Links
TF ☑

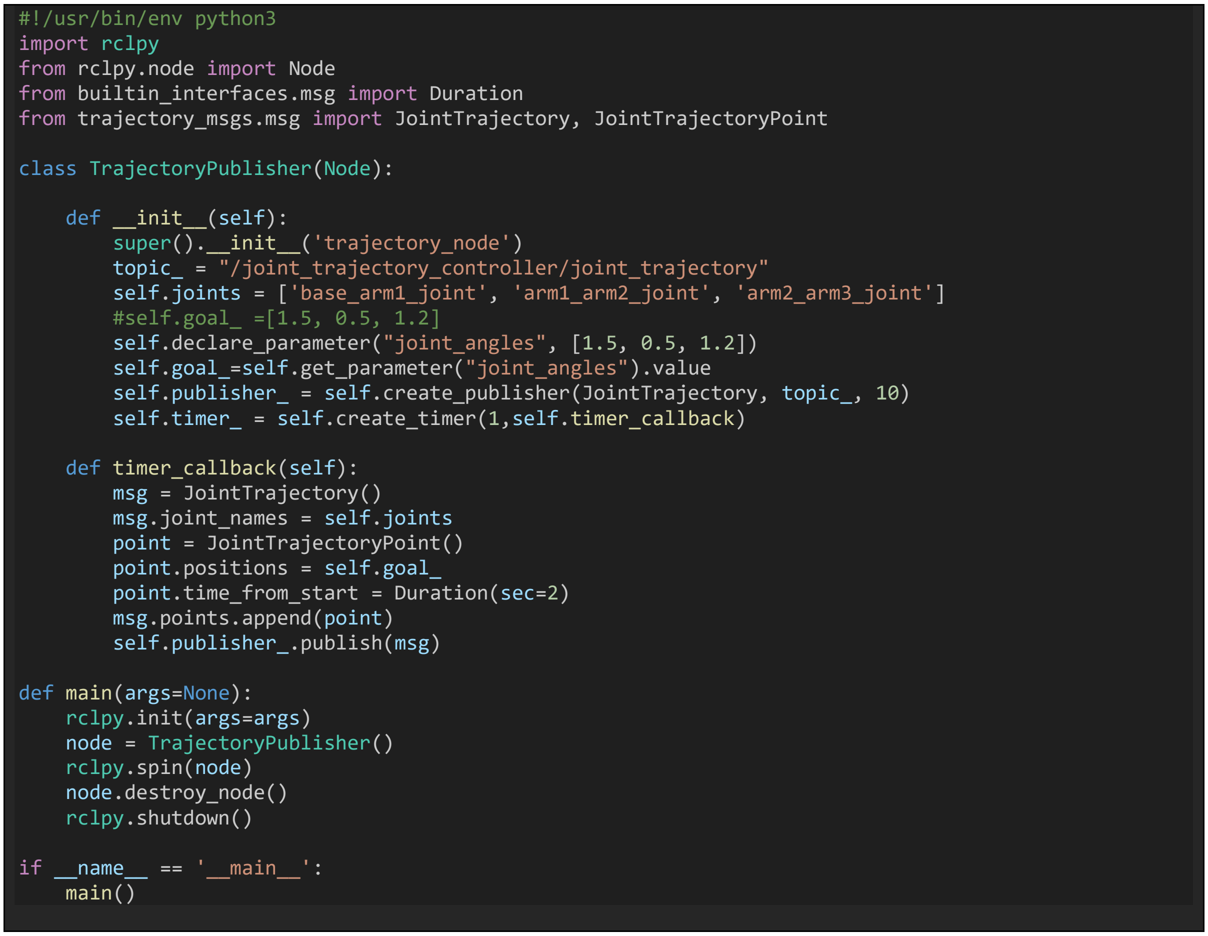
Type: Orbit (rviz_defaı
Current View: Orbit (rviz)
Near Clip ...: 0.01
Invert Z Axis □
Target Fra...: <Fixed Frame>
Distance: 2.46038
Focal Shap...: 0.05
Focal Shap... ☑
Yaw: 4.41042
Pitch: 0.355398
Focal Point: 0.50302; 0.1968...

ROS Time: 1725211092.60 ROS Elapsed: 17.34 Wall Time: 1725211092.61 Wall Elapsed: 17.34 □ Experimental

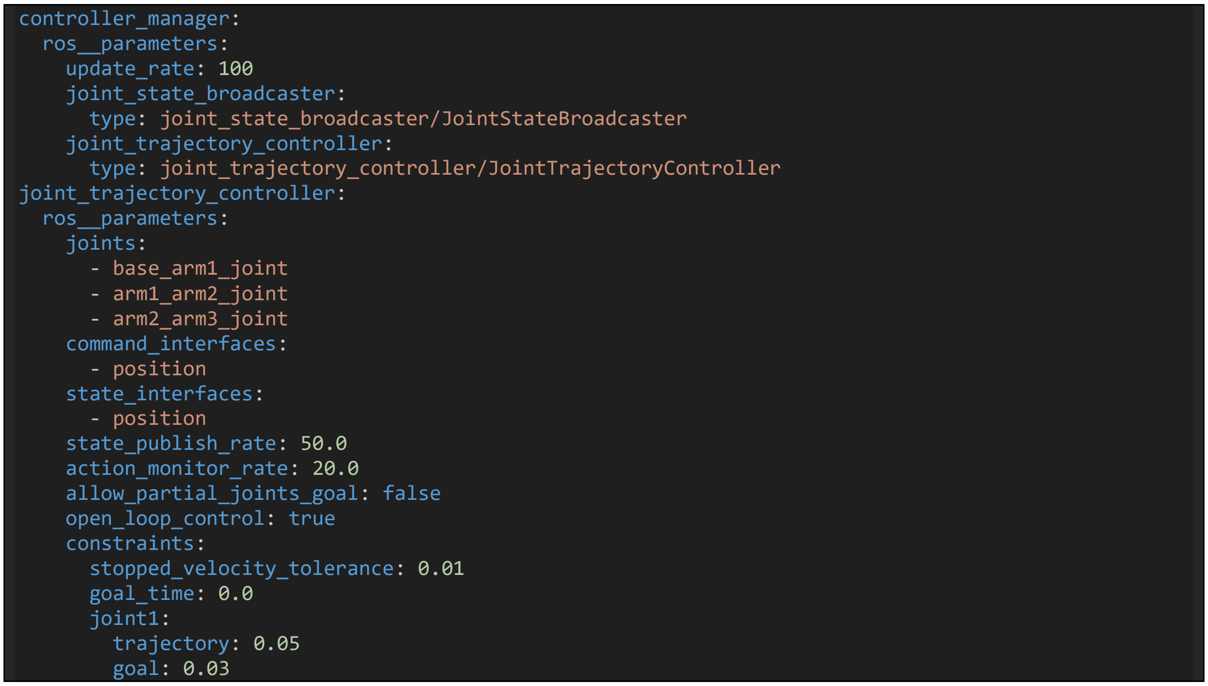
Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:** Zoom. **Shift:** More options.

58 fps

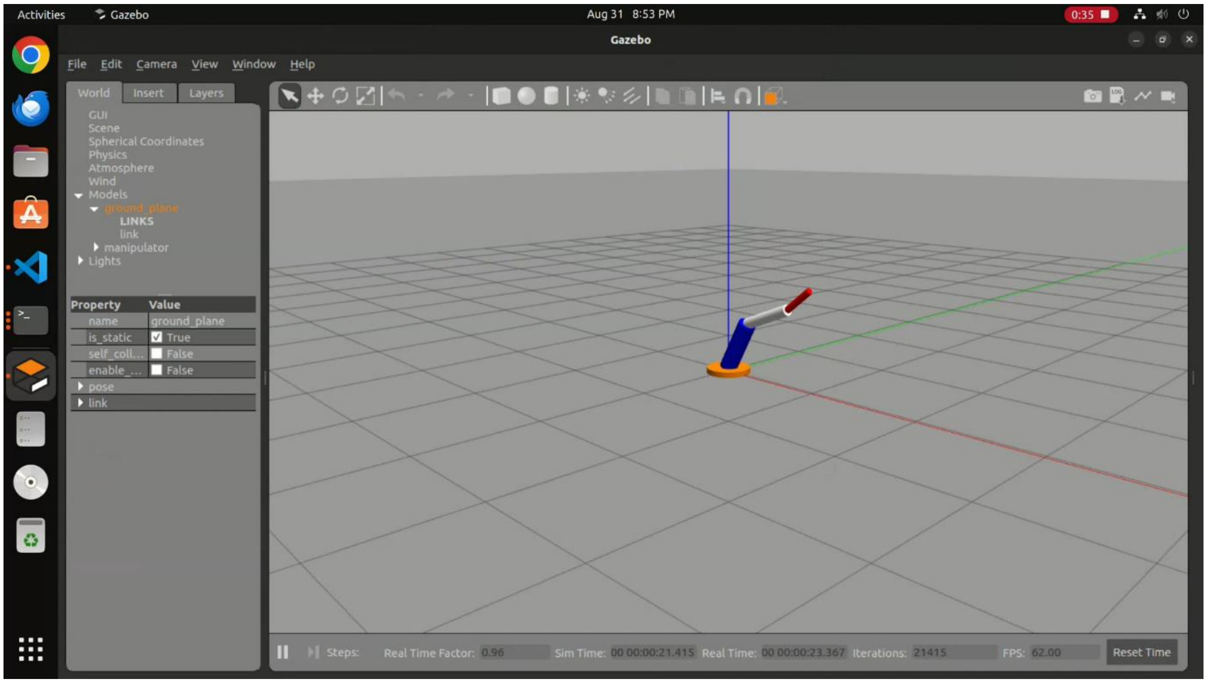
2. Forward kinematics controller for manipulator in Gazebo:



```python
#!/usr/bin/env python3
import rclpy
from rclpy.node import Node
from builtin_interfaces.msg import Duration
from trajectory_msgs.msg import JointTrajectory, JointTrajectoryPoint

class TrajectoryPublisher(Node):

    def __init__(self):
        super().__init__('trajectory_node')
        topic_ = "/joint_trajectory_controller/joint_trajectory"
        self.joints = ['base_arm1_joint', 'arm1_arm2_joint', 'arm2_arm3_joint']
        #self.goal_ =[1.5, 0.5, 1.2]
        self.declare_parameter("joint_angles", [1.5, 0.5, 1.2])
        self.goal_=self.get_parameter("joint_angles").value
        self.publisher_ = self.create_publisher(JointTrajectory, topic_, 10)
        self.timer_ = self.create_timer(1,self.timer_callback)

    def timer_callback(self):
        msg = JointTrajectory()
        msg.joint_names = self.joints
        point = JointTrajectoryPoint()
        point.positions = self.goal_
        point.time_from_start = Duration(sec=2)
        msg.points.append(point)
        self.publisher_.publish(msg)

def main(args=None):
    rclpy.init(args=args)
    node = TrajectoryPublisher()
    rclpy.spin(node)
    node.destroy_node()
    rclpy.shutdown()

if __name__ == '__main__':
    main()
```



```yaml
controller_manager:
  ros__parameters:
    update_rate: 100
    joint_state_broadcaster:
      type: joint_state_broadcaster/JointStateBroadcaster
    joint_trajectory_controller:
      type: joint_trajectory_controller/JointTrajectoryController
joint_trajectory_controller:
  ros__parameters:
    joints:
      - base_arm1_joint
      - arm1_arm2_joint
      - arm2_arm3_joint
    command_interfaces:
      - position
    state_interfaces:
      - position
    state_publish_rate: 50.0
    action_monitor_rate: 20.0
    allow_partial_joints_goal: false
    open_loop_control: true
    constraints:
      stopped_velocity_tolerance: 0.01
      goal_time: 0.0
      joint1:
        trajectory: 0.05
        goal: 0.03
```



Property	Value
name	ground_plane
is_static	☑ True
self_coll...	□ False
enable_...	□ False

Real Time Factor: 0.96 Sim Time: 00 00:00:21.415 Real Time: 00 00:00:23.367 Iterations: 21415 FPS: 62.00

Conclusion: Simulations of forward kinematics controllers for manipulator have been successfully conducted in RViz and Gazebo.