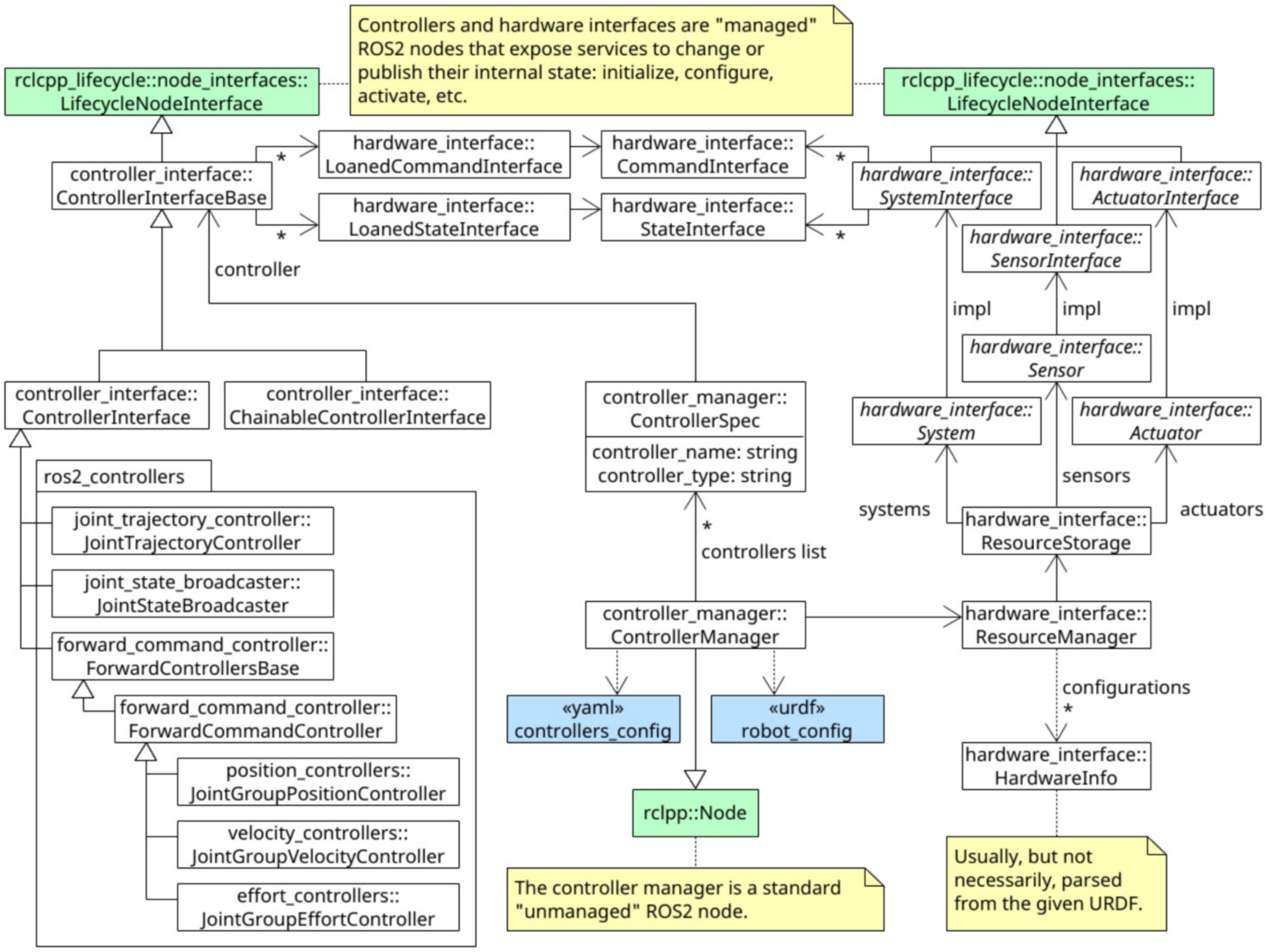
MANIPULATOR DESIGN AND CONTROL

Theory: If we drive a robot to a specific position by giving command to robot motors. It often fails to reach the goal precisely. There are several factors that influence robot motion such as robot dynamics, manufacturing tolerances, signal distortions, external forces (friction, payload, surface texture), environmental factors, etc. Robot control is used to manage, command, and direct the robot.



References:

https://github.com/ros-controls/control.ros.org/tree/humble

https://control.ros.org/master/index.html

http://wiki.ros.org/urdf/Tutorials/Adding%20Physical%20and%20Collision%20Properties%20to%20a%20URDF%20Model

Add extensions in Visual Studio

ros
ros snippet
xml
xml tools
urdf
xml complete
icons

Install the following packages

Execute in Terminal #1
sudo apt-get install ros-humble-teleop-twist-keyboard
sudo apt-get install ros-humble-joint-state-publisher*
sudo apt-get install ros-humble-joint-trajectory-controller
sudo apt-get install ros-humble-controller-manager
sudo apt install ros-humble-gazebo-*
sudo apt install ros-humble-gazebo-msgs
sudo apt install ros-humble-gazebo-ros
sudo apt install ros-humble-gazebo-ros2-control-demos
sudo apt install ros-humble-ros2-control
sudo apt install ros-humble-ros2-controllers
sudo apt install ros-humble-ros2controlcli
sudo apt install ros-humble-xacro
sudo apt install ros-humble-gazebo-dev
sudo apt install ros-humble-gazebo-plugins

cd asha_ws/src
ros2 pkg create --build-type ament_python lab5 --dependencies rclpy

cd ..
colcon build --packages-select lab5

cd asha_ws/src/lab5/
mkdir urdf
cd urdf
touch arm.urdf

```
<?xml version="1.0"?>
<robot name="arm">
<!-- 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>
```

```xml
            <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">
```

```xml
    <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="0.4" 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>
```

```xml
    <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/asha/asha_ws/src/lab5/config/control.yaml</parameters>
    </plugin>
  </gazebo>
```

```xml
    <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>
```

```
cd ..
mkdir launch
cd launch
touch rviz.launch.py
```

```python
from launch import LaunchDescription
from launch_ros.actions import Node



def generate_launch_description():

    urdf_file = urdf = '/home/asha/asha_ws/src/lab5/urdf/arm.urdf'
```

```
    joint_state_publisher_node = Node(
        package="joint_state_publisher_gui",
        executable="joint_state_publisher_gui",
        name="joint_state_publisher_gui",
        output="screen",
        arguments=[urdf_file]
    )
    robot_state_publisher_node = Node(
        package="robot_state_publisher",
        executable="robot_state_publisher",
        name="robot_state_publisher",
        output="screen",
        arguments=[urdf_file]
    )
    rviz_node = Node(
        package="rviz2",
        executable="rviz2",
        name="rviz2",
        output="screen"
    )

    nodes_to_run = [
        joint_state_publisher_node,
        robot_state_publisher_node,
        rviz_node
    ]

    return LaunchDescription(nodes_to_run)
```

touch gazebo.launch.py

```
import os
from launch import LaunchDescription
from launch.actions import ExecuteProcess
from launch_ros.actions import Node

def generate_launch_description():
    urdf_file = '/home/asha/asha_ws/src/lab5/urdf/arm.urdf'

    return LaunchDescription(
        [
            ExecuteProcess(
                cmd=["gazebo","-s","libgazebo_ros_factory.so",],
                output="screen",
            ),
            Node(
                package="gazebo_ros",
                executable="spawn_entity.py",
                arguments=["-entity","urdf_tutorial","-b","-file", urdf_file],
            ),
```

```
            Node(
                package="robot_state_publisher",
                executable="robot_state_publisher",
                output="screen",
                arguments=[urdf_file],
            ),

        ]
    )
```

```
cd ~/asha_ws/src/lab5
mkdir config
cd config
touch control.yaml
```

```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
```

```
touch arm.launch.py
edit arm.launch.py
```

```
import os
from launch import LaunchDescription
from launch.actions import ExecuteProcess, IncludeLaunchDescription,
RegisterEventHandler
from launch_ros.actions import Node
from launch.event_handlers import OnProcessExit
from launch.launch_description_sources import PythonLaunchDescriptionSource
from ament_index_python.packages import get_package_share_directory

import xacro

def generate_launch_description():

    urdf_file = '/home/asha/asha_ws/src/lab5/urdf/arm.urdf'
    controller_file = '/home/asha/asha_ws/src/lab5/config/control.yaml'
    robot_description = {"robot_description": urdf_file}

    gazebo = IncludeLaunchDescription(
        PythonLaunchDescriptionSource([os.path.join(
            get_package_share_directory('gazebo_ros'), 'launch'), '/gazebo.launch.py']),
    )

    doc = xacro.parse(open(urdf_file))
    xacro.process_doc(doc)
    params = {'robot_description': doc.toxml()}

    node_robot_state_publisher = Node(
        package='robot_state_publisher',
        executable='robot_state_publisher',
        output='screen',
        parameters=[params]
    )

    spawn_entity = Node(package='gazebo_ros', executable='spawn_entity.py',
        arguments=["-entity","lab5","-b","-file", urdf_file],
        output='screen'
    )

    load_joint_state_controller = ExecuteProcess(
        cmd=['ros2', 'control', 'load_controller', '--set-state', 'active', 'joint_state_broadcaster'],
        output='screen'
    )

    load_joint_trajectory_controller = ExecuteProcess(
        cmd=['ros2', 'control', 'load_controller', '--set-state', 'active',
'joint_trajectory_controller'],
        output='screen'
    )
```

```
    return LaunchDescription(
        [
            RegisterEventHandler(
                event_handler=OnProcessExit(
                    target_action=spawn_entity,
                    on_exit=[load_joint_state_controller],
                )
            ),

            RegisterEventHandler(
                event_handler=OnProcessExit(
                    target_action=load_joint_state_controller,
                    on_exit=[load_joint_trajectory_controller],
                )
            ),

            gazebo,

            node_robot_state_publisher,

            spawn_entity,

            Node(
                package="controller_manager",
                executable="ros2_control_node",
                parameters=[robot_description, controller_file],
                output="screen"
            )
        ]
    )
```

Execute in Terminal #1

colcon build --packages-select lab5 --symlink-install

Execute in Terminal #1

ros2 launch lab5 rviz.launch.py

Execute in Terminal #2

ros2 launch lab5 gazebo.launch.py
ctrl+c

Execute in Terminal #3

ros2 launch lab5 arm.launch.py

cd asha_ws/src/lab5/lab5/
touch controller.py
chmod +x controller.py

```
#!/usr/bin/env python3


#colcon build --packages-select lab5 –symlink-install
```

```
#ros2 run lab5 control --ros-args -p end_location:=[0.15,0.5,-0.2]

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_ =[0.0, 0.0, 0.75]
        #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()
```

Edit setup.py as

```
from setuptools import find_packages, setup
import os
from glob import glob
package_name = 'lab5'

setup(
```

```
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
        (os.path.join('share',package_name),glob('launch/*')),
        (os.path.join('share',package_name),glob('urdf/*')),
        (os.path.join('share',package_name),glob('config/*')),

    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='asha',
    maintainer_email='asha@todo.todo',
    description='TODO: Package description',
    license='TODO: License declaration',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            'control=lab5.controller:main'
        ],
    },
))
```

Execute in Terminal #1

colcon build --packages-select lab5 –symlink-install

Execute in Terminal #1

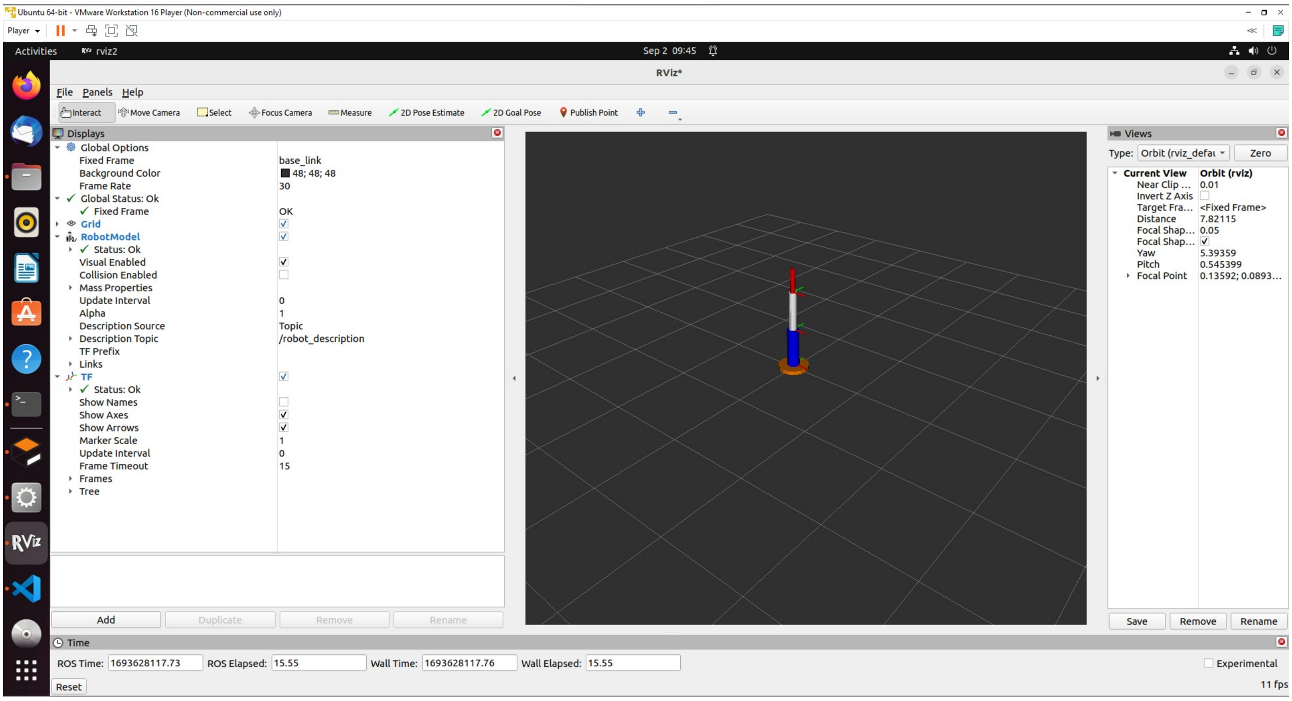
ros2 launch lab5 rviz.launch.py

Execute in Terminal #2

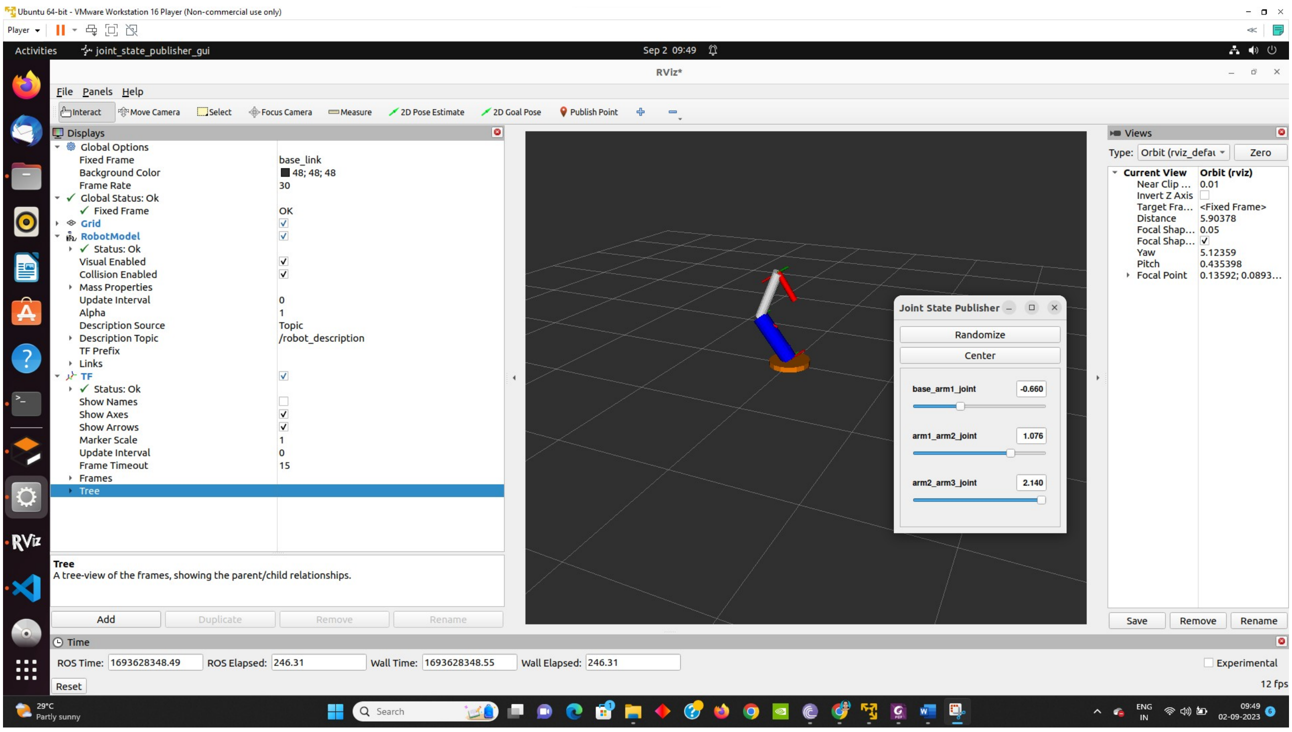
ros2 launch lab5 arm.launch.py

Execute in Terminal #3

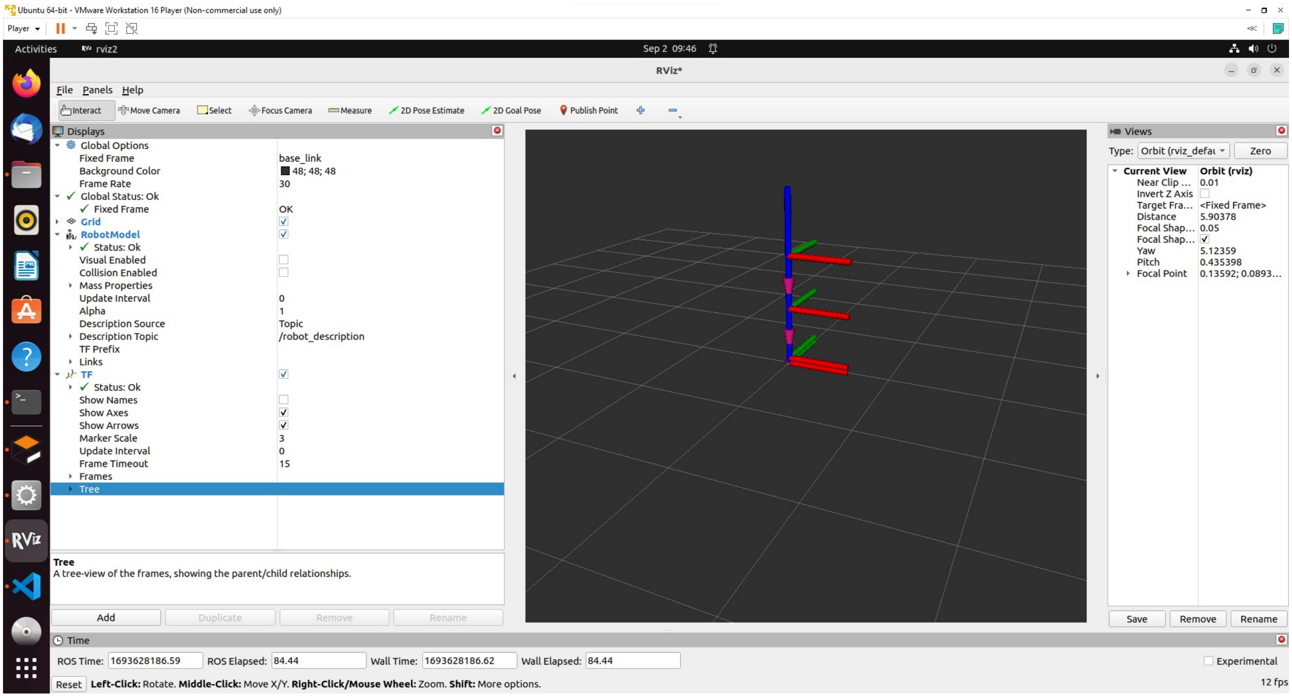
ros2 run lab5 control

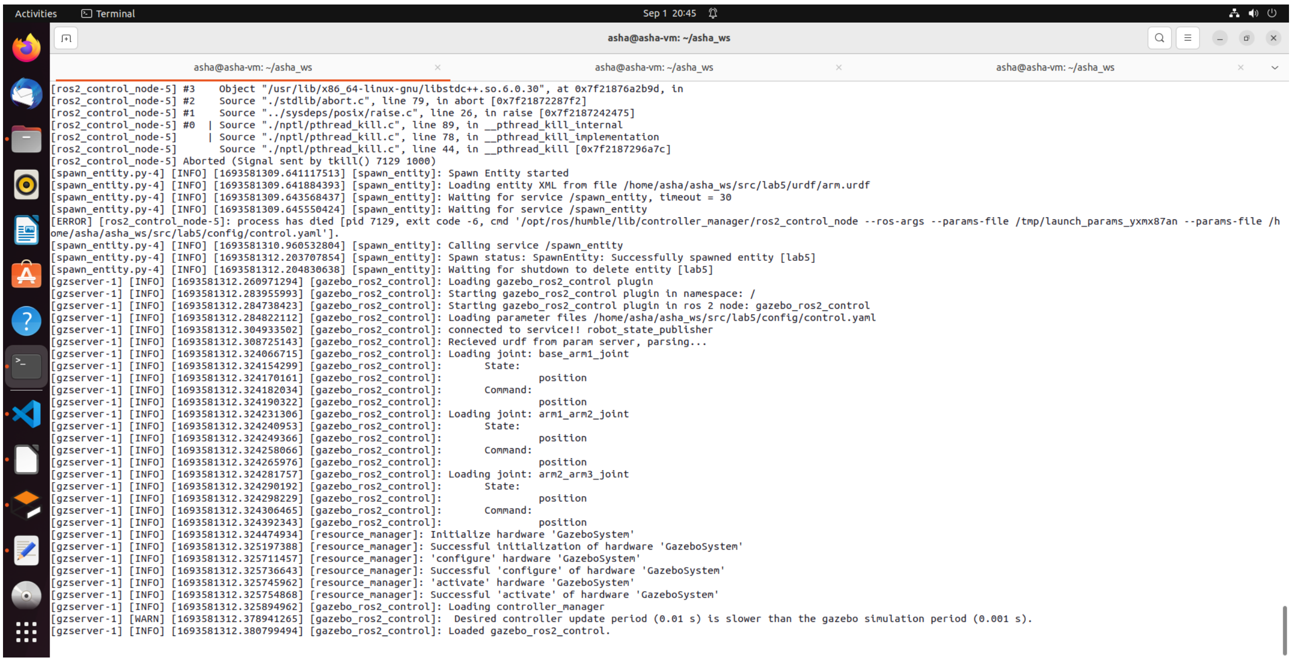


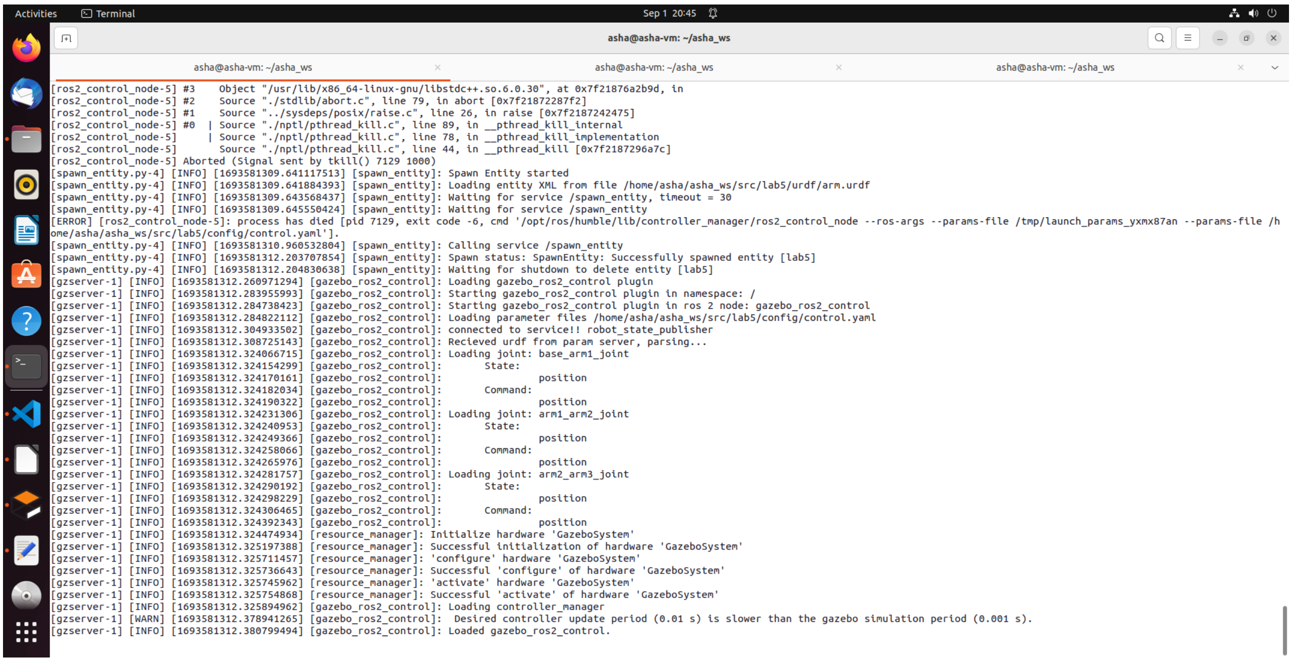
Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Activities
rviz2
Sep 2 09:45
RViz*
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Goal Pose Publish Point
Displays
Global Options
Fixed Frame base_link
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Mass Properties
Update Interval 0
Alpha 1
Description Source Topic
Description Topic /robot_description
TF Prefix
Links
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale 1
Update Interval 0
Frame Timeout 15
Frames
Tree
Add Duplicate Remove Rename
Views
Type: Orbit (rviz_defau Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 7.82115
Focal Shap... 0.05
Focal Shap...
Yaw 5.39359
Pitch 0.545399
Focal Point 0.13592; 0.0893...
Save Remove Rename
Time
ROS Time: 1693628117.73 ROS Elapsed: 15.55 Wall Time: 1693628117.76 Wall Elapsed: 15.55
Experimental
Reset
11 fps

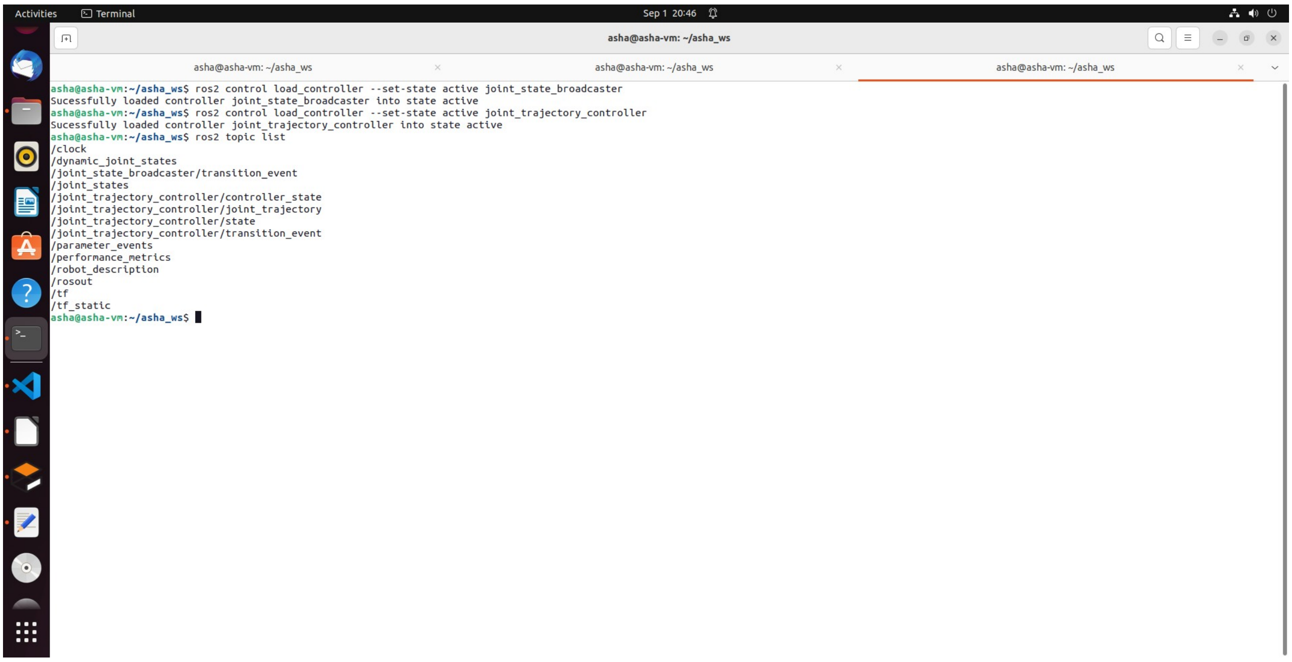


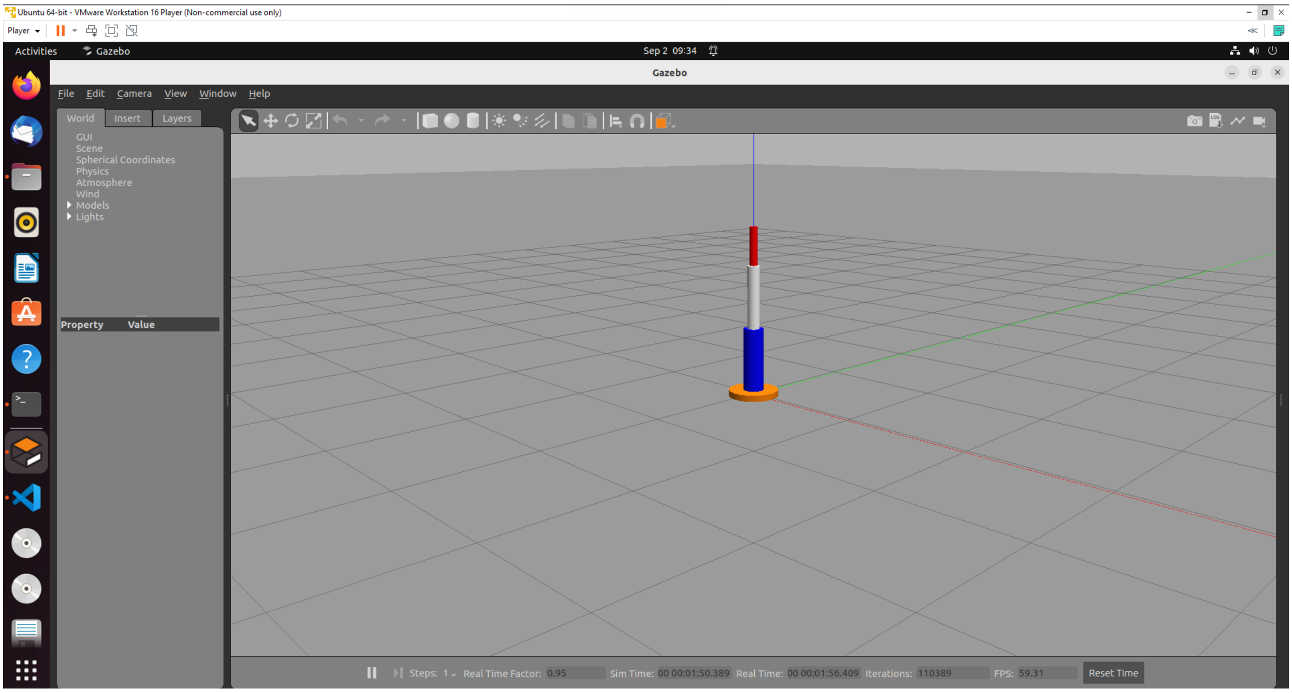
Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Activities
joint_state_publisher_gui
Sep 2 09:49
RViz*
File Panels Help
Displays
Global Options
Fixed Frame base_link
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Mass Properties
Update Interval 0
Alpha 1
Description Source Topic
Description Topic /robot_description
TF Prefix
Links
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale 1
Update Interval 0
Frame Timeout 15
Frames
Tree
Tree
A tree-view of the frames, showing the parent/child relationships.
Joint State Publisher
Randomize
Center
base_arm1_joint -0.660
arm1_arm2_joint 1.076
arm2_arm3_joint 2.140
Views
Type: Orbit (rviz_defau Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Target Fra... <Fixed Frame>
Distance 5.90378
Focal Shap... 0.05
Yaw 5.12359
Pitch 0.435398
Focal Point 0.13592; 0.0893...
Time
ROS Time: 1693628348.49 ROS Elapsed: 246.31 Wall Time: 1693628348.55 Wall Elapsed: 246.31
Experimental
Reset
12 fps
29°C Partly sunny
Search
ENG IN
09:49 02-09-2023

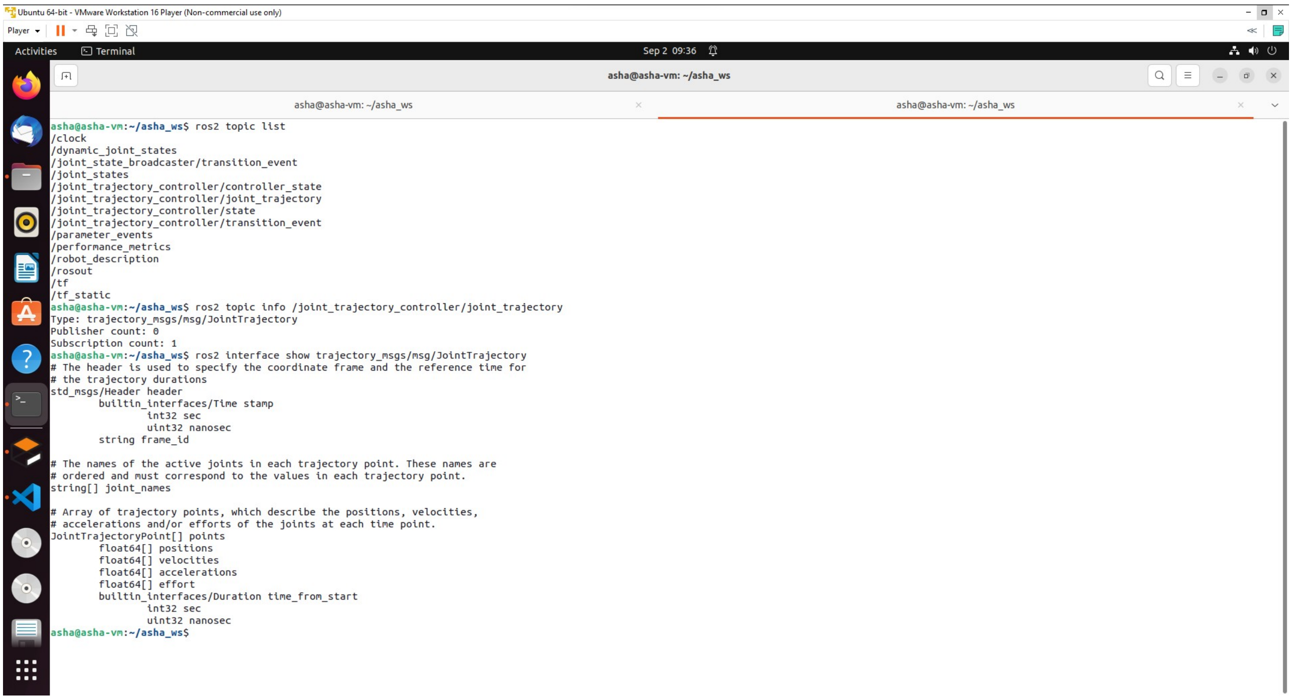

Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Activities
rviz2
Sep 2 09:46
RViz*
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Goal Pose
Publish Point
Displays
Global Options
Fixed Frame base_link
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Mass Properties
Update Interval 0
Alpha 1
Description Source Topic
Description Topic /robot_description
TF Prefix
Links
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale 3
Update Interval 0
Frame Timeout 15
Frames
Tree
Tree
A tree-view of the frames, showing the parent/child relationships.
Add
Duplicate
Remove
Rename
Views
Type: Orbit (rviz_defau
Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z Axis
Target Fra... <Fixed Frame>
Distance 5.90378
Focal Shap... 0.05
Focal Shap...
Yaw 5.12359
Pitch 0.435398
Focal Point 0.13592; 0.0893...
Save
Remove
Rename
Time
ROS Time: 1693628186.59
ROS Elapsed: 84.44
Wall Time: 1693628186.62
Wall Elapsed: 84.44
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.
12 fps

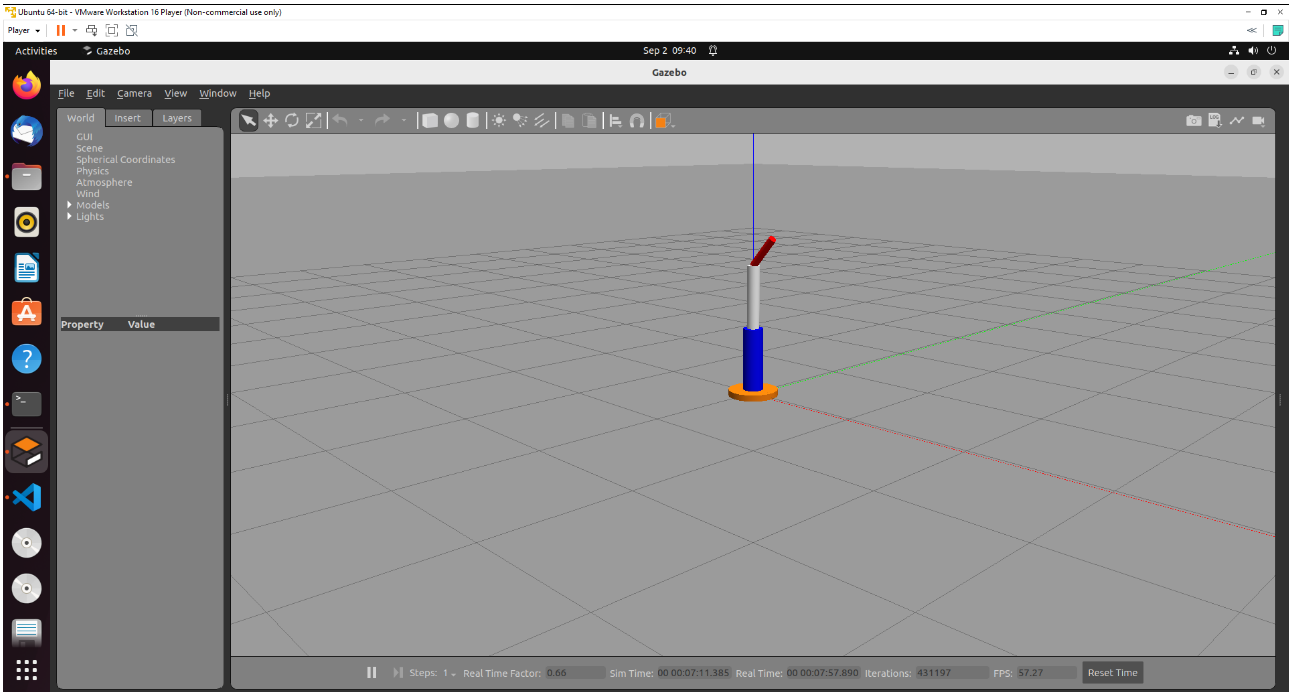

Activities
Terminal
Sep 1 20:45
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
[ros2_control_node-5] #3 Object "/usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30", at 0x7f21876a2b9d, in
[ros2_control_node-5] #2 Source "./stdlib/abort.c", line 79, in abort [0x7f21872287f2]
[ros2_control_node-5] #1 Source "../sysdeps/posix/raise.c", line 26, in raise [0x7f2187242475]
[ros2_control_node-5] #0 | Source "./nptl/pthread_kill.c", line 89, in __pthread_kill_internal
[ros2_control_node-5] | Source "./nptl/pthread_kill.c", line 78, in __pthread_kill_implementation
[ros2_control_node-5] Source "./nptl/pthread_kill.c", line 44, in __pthread_kill [0x7f2187296a7c]
[ros2_control_node-5] Aborted (Signal sent by tkill() 7129 1000)
[spawn_entity.py-4] [INFO] [1693581309.641117513] [spawn_entity]: Spawn Entity started
[spawn_entity.py-4] [INFO] [1693581309.641884393] [spawn_entity]: Loading entity XML from file /home/asha/asha_ws/src/lab5/urdf/arm.urdf
[spawn_entity.py-4] [INFO] [1693581309.643568437] [spawn_entity]: Waiting for service /spawn_entity, timeout = 30
[spawn_entity.py-4] [INFO] [1693581309.645550424] [spawn_entity]: Waiting for service /spawn_entity
[ERROR] [ros2_control_node-5]: process has died [pid 7129, exit code -6, cmd '/opt/ros/humble/lib/controller_manager/ros2_control_node --ros-args --params-file /tmp/launch_params_yxmx87an --params-file /home/asha/asha_ws/src/lab5/config/control.yaml'].
[spawn_entity.py-4] [INFO] [1693581310.960532804] [spawn_entity]: Calling service /spawn_entity
[spawn_entity.py-4] [INFO] [1693581312.203707854] [spawn_entity]: Spawn status: SpawnEntity: Successfully spawned entity [lab5]
[spawn_entity.py-4] [INFO] [1693581312.204830638] [spawn_entity]: Waiting for shutdown to delete entity [lab5]
[gzserver-1] [INFO] [1693581312.260971294] [gazebo_ros2_control]: Loading gazebo_ros2_control plugin
[gzserver-1] [INFO] [1693581312.283955993] [gazebo_ros2_control]: Starting gazebo_ros2_control plugin in namespace: /
[gzserver-1] [INFO] [1693581312.284738423] [gazebo_ros2_control]: Starting gazebo_ros2_control plugin in ros 2 node: gazebo_ros2_control
[gzserver-1] [INFO] [1693581312.284822112] [gazebo_ros2_control]: Loading parameter files /home/asha/asha_ws/src/lab5/config/control.yaml
[gzserver-1] [INFO] [1693581312.304933502] [gazebo_ros2_control]: connected to service!! robot_state_publisher
[gzserver-1] [INFO] [1693581312.308725143] [gazebo_ros2_control]: Recieved urdf from param server, parsing...
[gzserver-1] [INFO] [1693581312.324066715] [gazebo_ros2_control]: Loading joint: base_arm1_joint
[gzserver-1] [INFO] [1693581312.324154299] [gazebo_ros2_control]: State:
[gzserver-1] [INFO] [1693581312.324170161] [gazebo_ros2_control]: position
[gzserver-1] [INFO] [1693581312.324182034] [gazebo_ros2_control]: Command:
[gzserver-1] [INFO] [1693581312.324190322] [gazebo_ros2_control]: position
[gzserver-1] [INFO] [1693581312.324231306] [gazebo_ros2_control]: Loading joint: arm1_arm2_joint
[gzserver-1] [INFO] [1693581312.324240953] [gazebo_ros2_control]: State:
[gzserver-1] [INFO] [1693581312.324249366] [gazebo_ros2_control]: position
[gzserver-1] [INFO] [1693581312.324258066] [gazebo_ros2_control]: Command:
[gzserver-1] [INFO] [1693581312.324265976] [gazebo_ros2_control]: position
[gzserver-1] [INFO] [1693581312.324281757] [gazebo_ros2_control]: Loading joint: arm2_arm3_joint
[gzserver-1] [INFO] [1693581312.324290192] [gazebo_ros2_control]: State:
[gzserver-1] [INFO] [1693581312.324298229] [gazebo_ros2_control]: position
[gzserver-1] [INFO] [1693581312.324306465] [gazebo_ros2_control]: Command:
[gzserver-1] [INFO] [1693581312.324392343] [gazebo_ros2_control]: position
[gzserver-1] [INFO] [1693581312.324474934] [resource_manager]: Initialize hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325197388] [resource_manager]: Successful initialization of hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325711457] [resource_manager]: 'configure' hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325736643] [resource_manager]: Successful 'configure' of hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325745962] [resource_manager]: 'activate' hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325754868] [resource_manager]: Successful 'activate' of hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325894962] [gazebo_ros2_control]: Loading controller_manager
[gzserver-1] [WARN] [1693581312.378941265] [gazebo_ros2_control]: Desired controller update period (0.01 s) is slower than the gazebo simulation period (0.001 s).
[gzserver-1] [INFO] [1693581312.380799494] [gazebo_ros2_control]: Loaded gazebo_ros2_control.

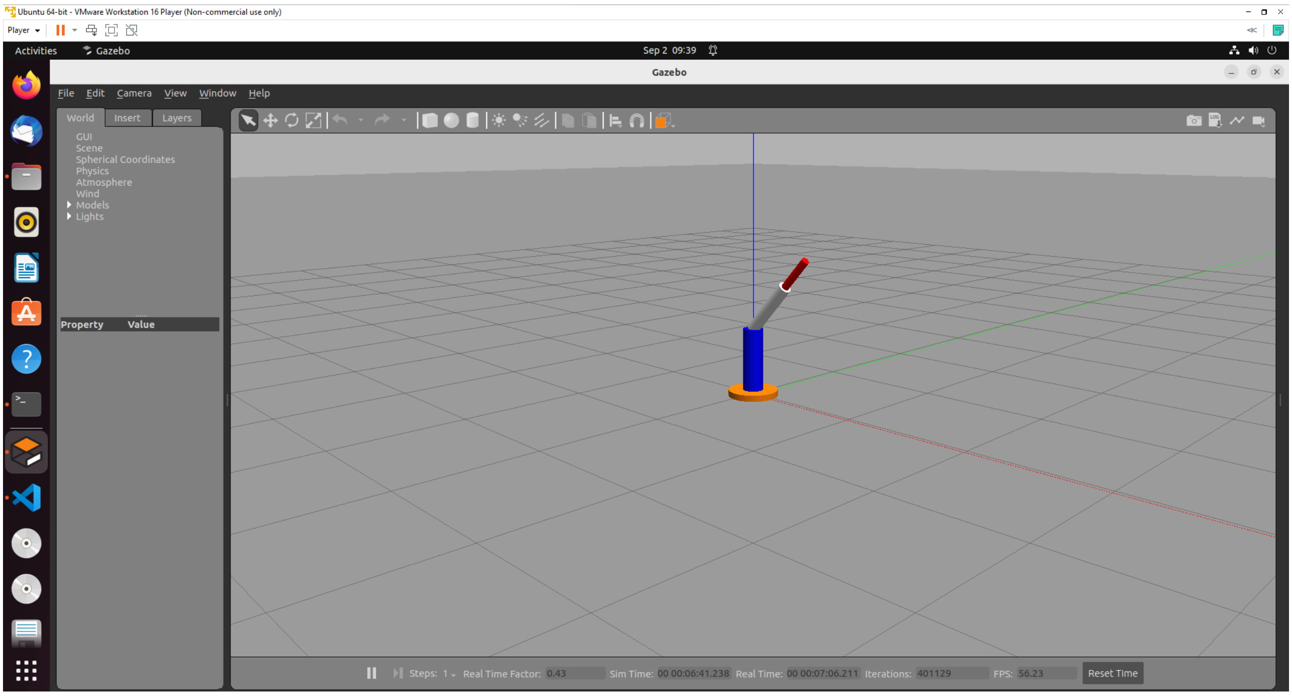

Activities
Terminal
Sep 1 20:45
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
[ros2_control_node-5] #3   Object "/usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.30", at 0x7f21876a2b9d, in
[ros2_control_node-5] #2   Source "./stdlib/abort.c", line 79, in abort [0x7f21872287f2]
[ros2_control_node-5] #1   Source "../sysdeps/posix/raise.c", line 26, in raise [0x7f2187242475]
[ros2_control_node-5] #0 | Source "./nptl/pthread_kill.c", line 89, in __pthread_kill_internal
[ros2_control_node-5] | Source "./nptl/pthread_kill.c", line 78, in __pthread_kill_implementation
[ros2_control_node-5]   Source "./nptl/pthread_kill.c", line 44, in __pthread_kill [0x7f2187296a7c]
[ros2_control_node-5] Aborted (Signal sent by tkill() 7129 1000)
[spawn_entity.py-4] [INFO] [1693581309.641117513] [spawn_entity]: Spawn Entity started
[spawn_entity.py-4] [INFO] [1693581309.641884393] [spawn_entity]: Loading entity XML from file /home/asha/asha_ws/src/lab5/urdf/arm.urdf
[spawn_entity.py-4] [INFO] [1693581309.643568437] [spawn_entity]: Waiting for service /spawn_entity, timeout = 30
[spawn_entity.py-4] [INFO] [1693581309.645550424] [spawn_entity]: Waiting for service /spawn_entity
[ERROR] [ros2_control_node-5]: process has died [pid 7129, exit code -6, cmd '/opt/ros/humble/lib/controller_manager/ros2_control_node --ros-args --params-file /tmp/launch_params_yxmx87an --params-file /h
ome/asha/asha_ws/src/lab5/config/control.yaml'].
[spawn_entity.py-4] [INFO] [1693581310.960532804] [spawn_entity]: Calling service /spawn_entity
[spawn_entity.py-4] [INFO] [1693581312.203707854] [spawn_entity]: Spawn status: SpawnEntity: Successfully spawned entity [lab5]
[spawn_entity.py-4] [INFO] [1693581312.204830638] [spawn_entity]: Waiting for shutdown to delete entity [lab5]
[gzserver-1] [INFO] [1693581312.260971294] [gazebo_ros2_control]: Loading gazebo_ros2_control plugin
[gzserver-1] [INFO] [1693581312.283955993] [gazebo_ros2_control]: Starting gazebo_ros2_control plugin in namespace: /
[gzserver-1] [INFO] [1693581312.284738423] [gazebo_ros2_control]: Starting gazebo_ros2_control plugin in ros 2 node: gazebo_ros2_control
[gzserver-1] [INFO] [1693581312.284822112] [gazebo_ros2_control]: Loading parameter files /home/asha/asha_ws/src/lab5/config/control.yaml
[gzserver-1] [INFO] [1693581312.304933502] [gazebo_ros2_control]: connected to service!! robot_state_publisher
[gzserver-1] [INFO] [1693581312.308725143] [gazebo_ros2_control]: Recieved urdf from param server, parsing...
[gzserver-1] [INFO] [1693581312.324066715] [gazebo_ros2_control]: Loading joint: base_arm1_joint
[gzserver-1] [INFO] [1693581312.324154299] [gazebo_ros2_control]:   State:
[gzserver-1] [INFO] [1693581312.324170161] [gazebo_ros2_control]:   position
[gzserver-1] [INFO] [1693581312.324182034] [gazebo_ros2_control]:   Command:
[gzserver-1] [INFO] [1693581312.324190322] [gazebo_ros2_control]:   position
[gzserver-1] [INFO] [1693581312.324231306] [gazebo_ros2_control]: Loading joint: arm1_arm2_joint
[gzserver-1] [INFO] [1693581312.324240953] [gazebo_ros2_control]:   State:
[gzserver-1] [INFO] [1693581312.324249366] [gazebo_ros2_control]:   position
[gzserver-1] [INFO] [1693581312.324258066] [gazebo_ros2_control]:   Command:
[gzserver-1] [INFO] [1693581312.324265976] [gazebo_ros2_control]:   position
[gzserver-1] [INFO] [1693581312.324281757] [gazebo_ros2_control]: Loading joint: arm2_arm3_joint
[gzserver-1] [INFO] [1693581312.324290192] [gazebo_ros2_control]:   State:
[gzserver-1] [INFO] [1693581312.324298229] [gazebo_ros2_control]:   position
[gzserver-1] [INFO] [1693581312.324306465] [gazebo_ros2_control]:   Command:
[gzserver-1] [INFO] [1693581312.324392343] [gazebo_ros2_control]:   position
[gzserver-1] [INFO] [1693581312.324474934] [resource_manager]: Initialize hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325197388] [resource_manager]: Successful initialization of hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325711457] [resource_manager]: 'configure' hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325736643] [resource_manager]: Successful 'configure' of hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325745962] [resource_manager]: 'activate' hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325754868] [resource_manager]: Successful 'activate' of hardware 'GazeboSystem'
[gzserver-1] [INFO] [1693581312.325894962] [gazebo_ros2_control]: Loading controller_manager
[gzserver-1] [WARN] [1693581312.378941265] [gazebo_ros2_control]: Desired controller update period (0.01 s) is slower than the gazebo simulation period (0.001 s).
[gzserver-1] [INFO] [1693581312.380799494] [gazebo_ros2_control]: Loaded gazebo_ros2_control.

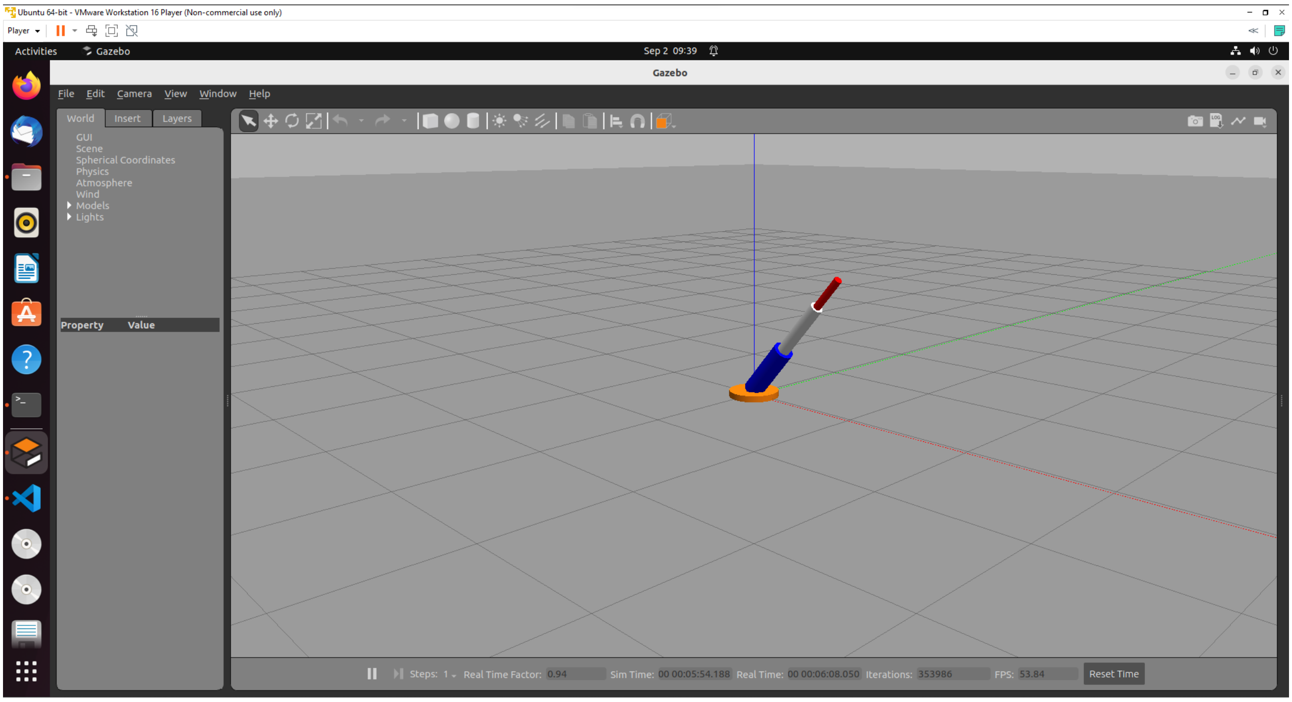

Activities
Terminal
Sep 1 20:46
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
asha@asha-vm: ~/asha_ws
asha@asha-vm:~/asha_ws$ ros2 control load_controller --set-state active joint_state_broadcaster
Sucessfully loaded controller joint_state_broadcaster into state active
asha@asha-vm:~/asha_ws$ ros2 control load_controller --set-state active joint_trajectory_controller
Sucessfully loaded controller joint_trajectory_controller into state active
asha@asha-vm:~/asha_ws$ ros2 topic list
/clock
/dynamic_joint_states
/joint_state_broadcaster/transition_event
/joint_states
/joint_trajectory_controller/controller_state
/joint_trajectory_controller/joint_trajectory
/joint_trajectory_controller/state
/joint_trajectory_controller/transition_event
/parameter_events
/performance_metrics
/robot_description
/rosout
/tf
/tf_static
asha@asha-vm:~/asha_ws$

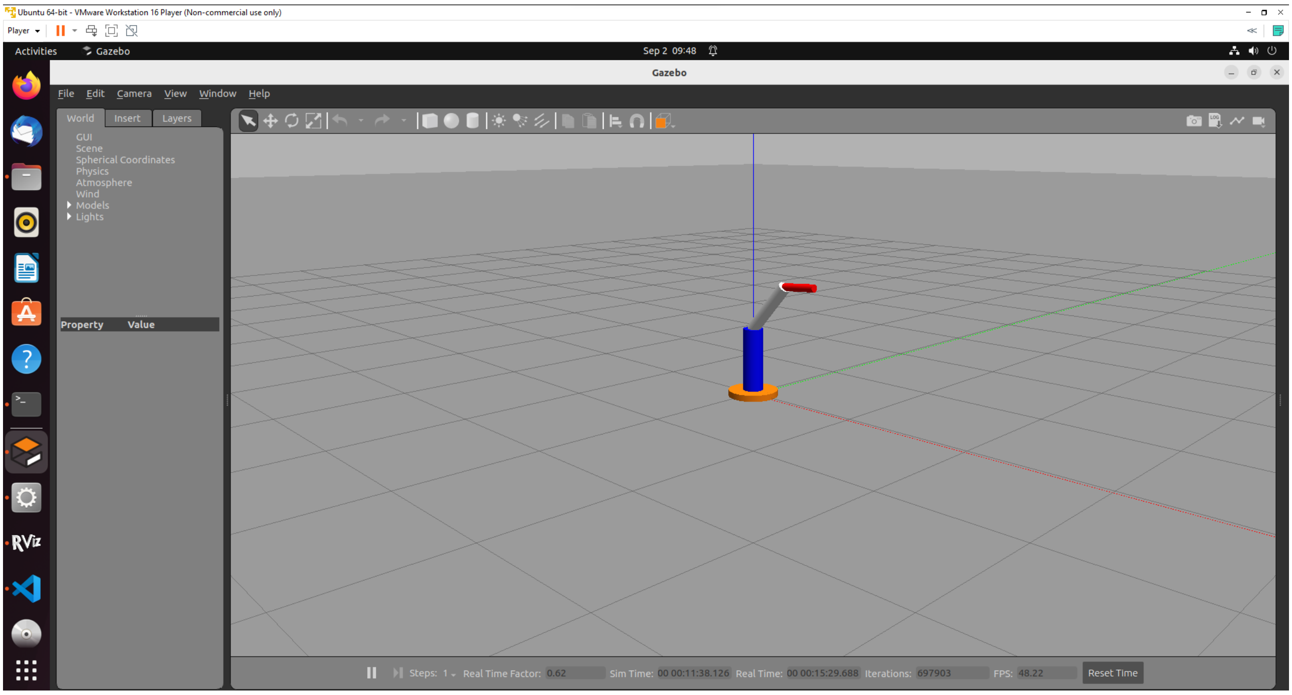

Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Activities
Gazebo
Sep 2 09:34
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Steps: 1 Real Time Factor: 0.95 Sim Time: 00 00:01:50.389 Real Time: 00 00:01:56.409 Iterations: 110389 FPS: 59.31 Reset Time

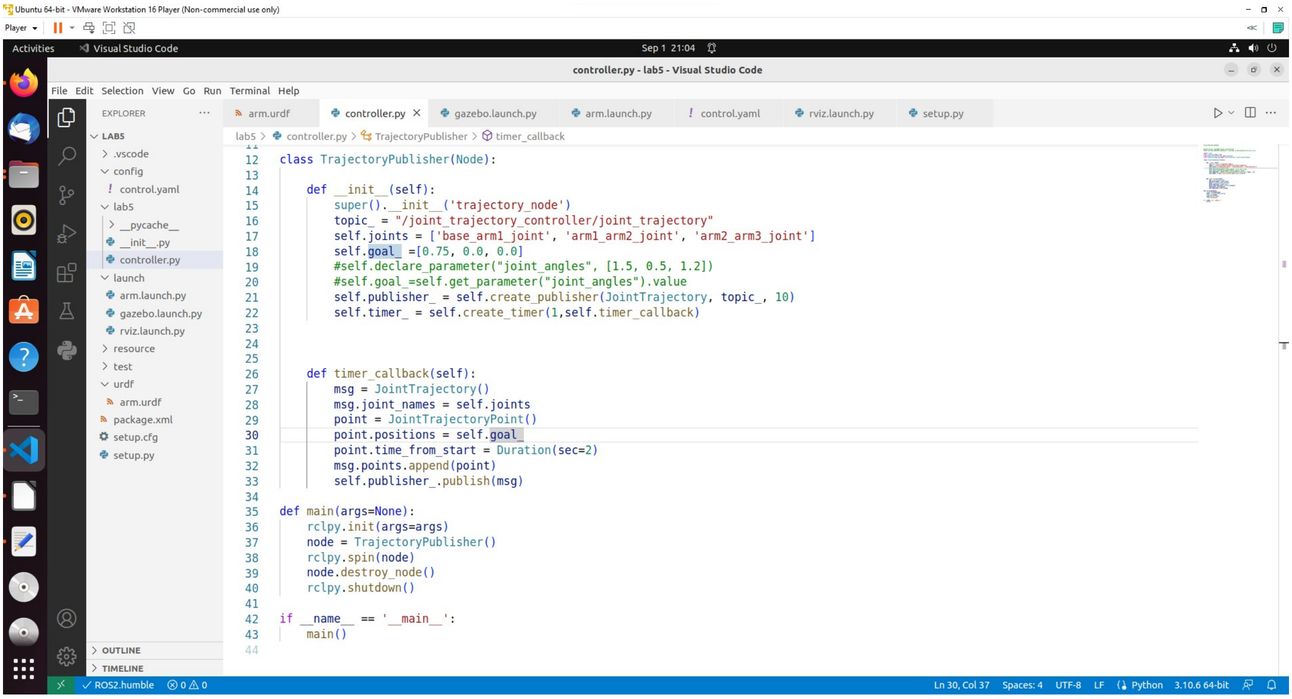

Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Activities
Terminal
Sep 2 09:36
asha@asha-vm: ~/asha_ws
asha@asha-vm:~/asha_ws$ ros2 topic list
/clock
/dynamic_joint_states
/joint_state_broadcaster/transition_event
/joint_states
/joint_trajectory_controller/controller_state
/joint_trajectory_controller/joint_trajectory
/joint_trajectory_controller/state
/joint_trajectory_controller/transition_event
/parameter_events
/performance_metrics
/robot_description
/rosout
/tf
/tf_static
asha@asha-vm:~/asha_ws$ ros2 topic info /joint_trajectory_controller/joint_trajectory
Type: trajectory_msgs/msg/JointTrajectory
Publisher count: 0
Subscription count: 1
asha@asha-vm:~/asha_ws$ ros2 interface show trajectory_msgs/msg/JointTrajectory
The header is used to specify the coordinate frame and the reference time for
the trajectory durations
std_msgs/Header header
builtin_interfaces/Time stamp
int32 sec
uint32 nanosec
string frame_id
The names of the active joints in each trajectory point. These names are
ordered and must correspond to the values in each trajectory point.
string[] joint_names
Array of trajectory points, which describe the positions, velocities,
accelerations and/or efforts of the joints at each time point.
JointTrajectoryPoint[] points
float64[] positions
float64[] velocities
float64[] accelerations
float64[] effort
builtin_interfaces/Duration time_from_start
int32 sec
uint32 nanosec
asha@asha-vm:~/asha_ws$


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Gazebo
Sep 2 09:40
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Steps: 1 Real Time Factor: 0.66 Sim Time: 00 00:07:11.385 Real Time: 00 00:07:57.890 Iterations: 431197 FPS: 57.27 Reset Time


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Gazebo
Sep 2 09:39
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Steps: 1 Real Time Factor: 0.43 Sim Time: 00 00:06:41.238 Real Time: 00 00:07:06.211 Iterations: 401129 FPS: 56.23 Reset Time


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Gazebo
Sep 2 09:39
Gazebo
File
Edit
Camera
View
Window
Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property
Value
Steps: 1
Real Time Factor: 0.94
Sim Time: 00 00:05:54.188
Real Time: 00 00:06:08.050
Iterations: 353986
FPS: 53.84
Reset Time


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Gazebo
Sep 2 09:48
Gazebo
File
Edit
Camera
View
Window
Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property
Value
RViz
Steps: 1
Real Time Factor: 0.62
Sim Time: 00 00:11:38.126
Real Time: 00 00:15:29.688
Iterations: 697903
FPS: 48.22
Reset Time



ROS2 parameters:

Parameters help to provide the values while running the code.

ros2 param list

```
#!/usr/bin/env python3


#colcon build --packages-select lab5 --symlink-install
#ros2 run lab5 control --ros-args -p end_location:=[0.15,0.5,-0.2]

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_ =[0.0, 0.0, 0.75]
        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()
```

Execute in Terminal #1

```
colcon build --packages-select lab5 --symlink-install
```

Execute in Terminal #2

```
ros2 launch lab5 arm.launch.py
ros2 control load_controller --set-state active joint_state_broadcaster
ros2 control load_controller --set-state active joint_trajectory_controller
```

Execute in Terminal #3

```
ros2 run lab5 control --ros-args -p joint_angles:=[0.5,0.5,0.2]
```

Exercise 1: Write a python code to move the manipulator to end location using inverse kinematics.

Exercise 2: Rewrite the complete code using xacro (optional).

Exercise 3: Replicate the process for UR5e robot given its urdf file (optional).

References

https://docs.ros.org/en/humble/Tutorials/URDF/Using-URDF-with-Robot-State-Publisher.html

https://github.com/benbongalon/ros2-urdf-tutorial/tree/master/urdf_tutorial

https://github.com/cra-ros-pkg/robot_localization/tree/humble-devel

https://github.com/ros/robot_state_publisher/tree/humble

https://github.com/ros/joint_state_publisher/tree/humble

http://gazebosim.org/tutorials?tut=ros_urdf