UR5 control using ROS2

sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y

sudo apt-get install python3-colcon-ros

sudo apt-get install python3-colcon-bash

sudo apt update && sudo apt install curl gnupg2 lsb-release

sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(source /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null

sudo apt update

sudo apt-get upgrade

sudo apt install ros-foxy-desktop
sudo apt install ros-foxy-moveit
sudo apt install ros-foxy-ros2-control
sudo apt install ros-foxy-rqt-service-caller
sudo apt install ros-foxy-warehouse-ros-mongo

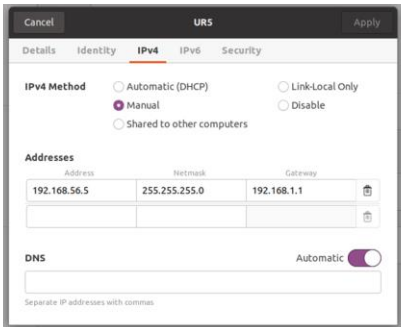
mkdir ur5
cd ur5
mkdir ros2
cd ros2
mkdir src
cd src

git clone --recurse-submodules -j8 https://github.com/ashacs2/ur5_interface.git -b ros2
git clone --recurse-submodules -j8 https://github.com/ashacs2/ur5_ros2_interface.git -b ros2
gedit ~/.bashrc
source ~/ur5/ros2/install/setup.bash

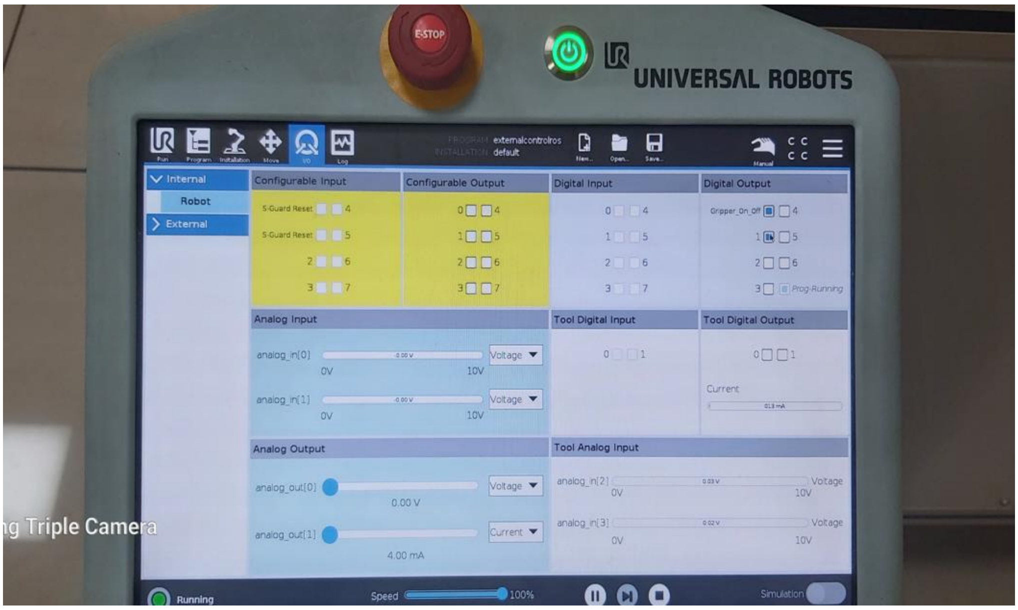
cd ur5/ros2/
rosdep install --from-paths src --ignore-src -r -y

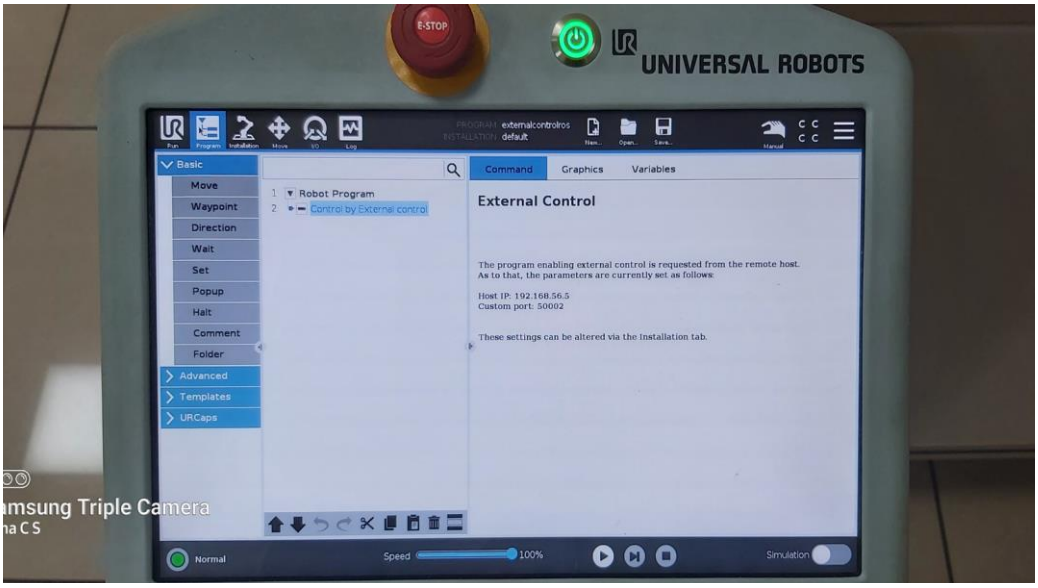
cd ..
colcon build --symlink-install --cmake-args -DCMAKE_EXPORT_COMPILE_COMMANDS=1

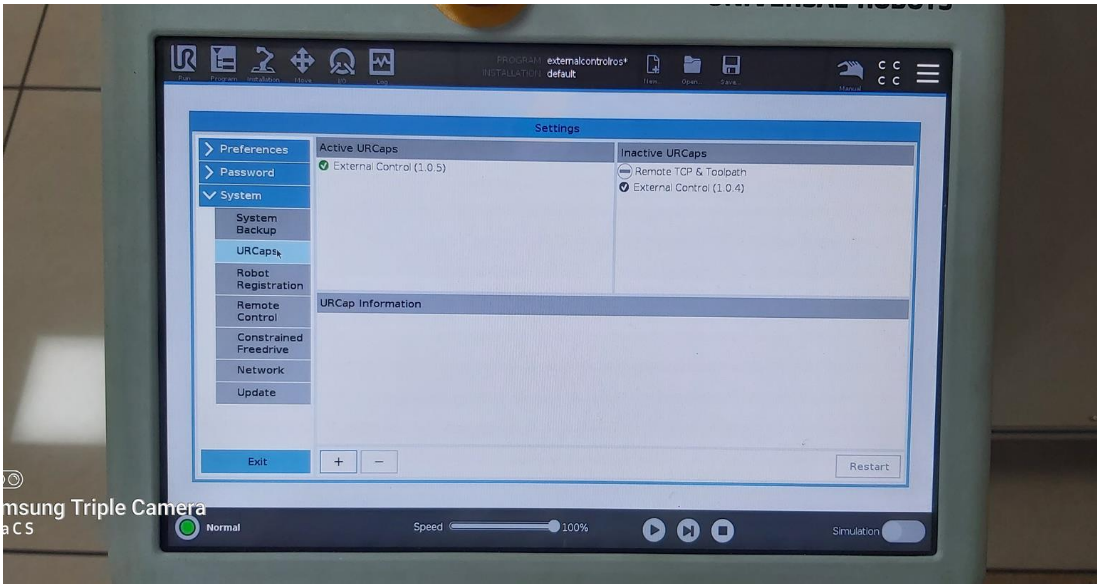
Connect an ethernet cable from UR5 to PC.

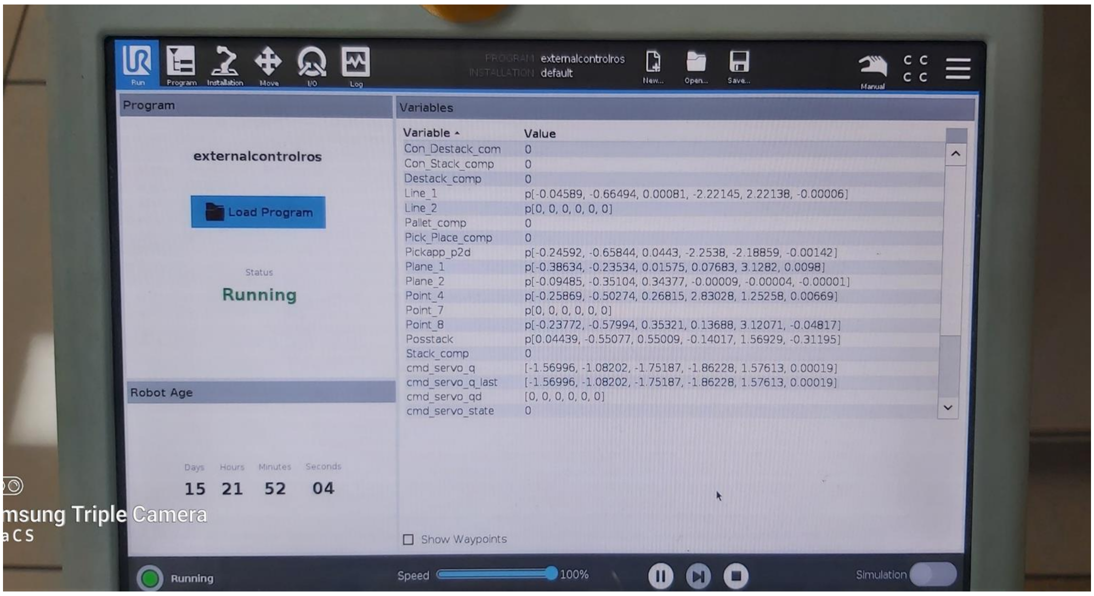


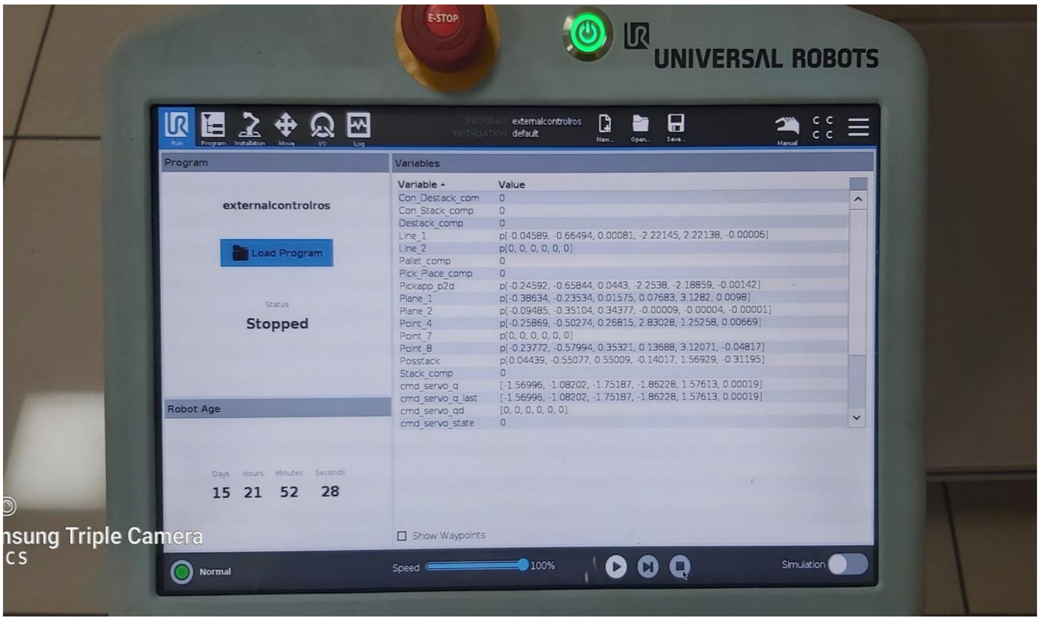
Settings in Teach Pendent:

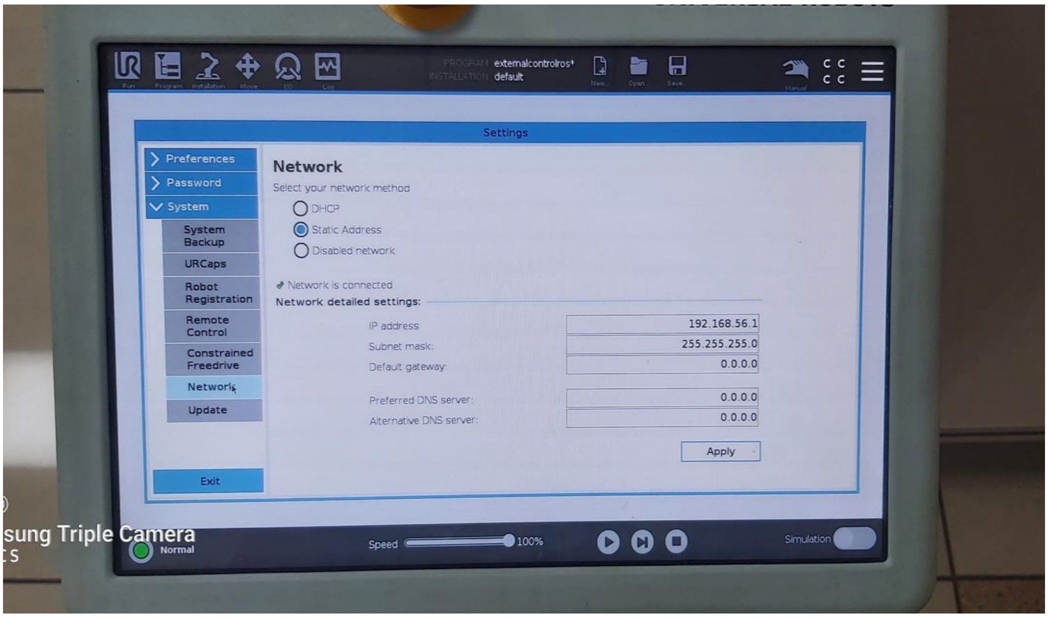


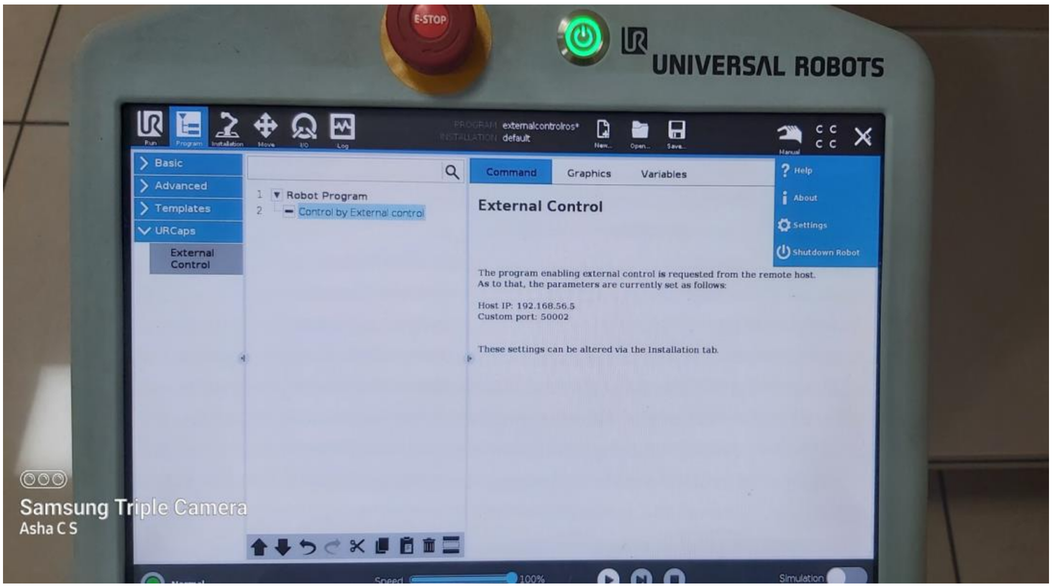

E-STOP
UNIVERSAL ROBOTS
PROGRAM externalcontrolros
INSTALLATION default
Basic
Move
Waypoint
Direction
Wait
Set
Popup
Halt
Comment
Folder
Advanced
Templates
URCaps
1 Robot Program
2 Control by External control
Command
Graphics
Variables
External Control
The program enabling external control is requested from the remote host.
As to that, the parameters are currently set as follows:
Host IP: 192.168.56.5
Custom port: 50002
These settings can be altered via the Installation tab.
Normal
Speed
100%
Simulation


PROGRAM externalcontrolros*
INSTALLATION default
Settings
Preferences
Password
System
System Backup
URCaps
Robot Registration
Remote Control
Constrained Freedrive
Network
Update
Active URCaps
External Control (1.0.5)
Inactive URCaps
Remote TCP & Toolpath
External Control (1.0.4)
URCap Information
Exit
Restart
Normal
Speed
100%
Simulation

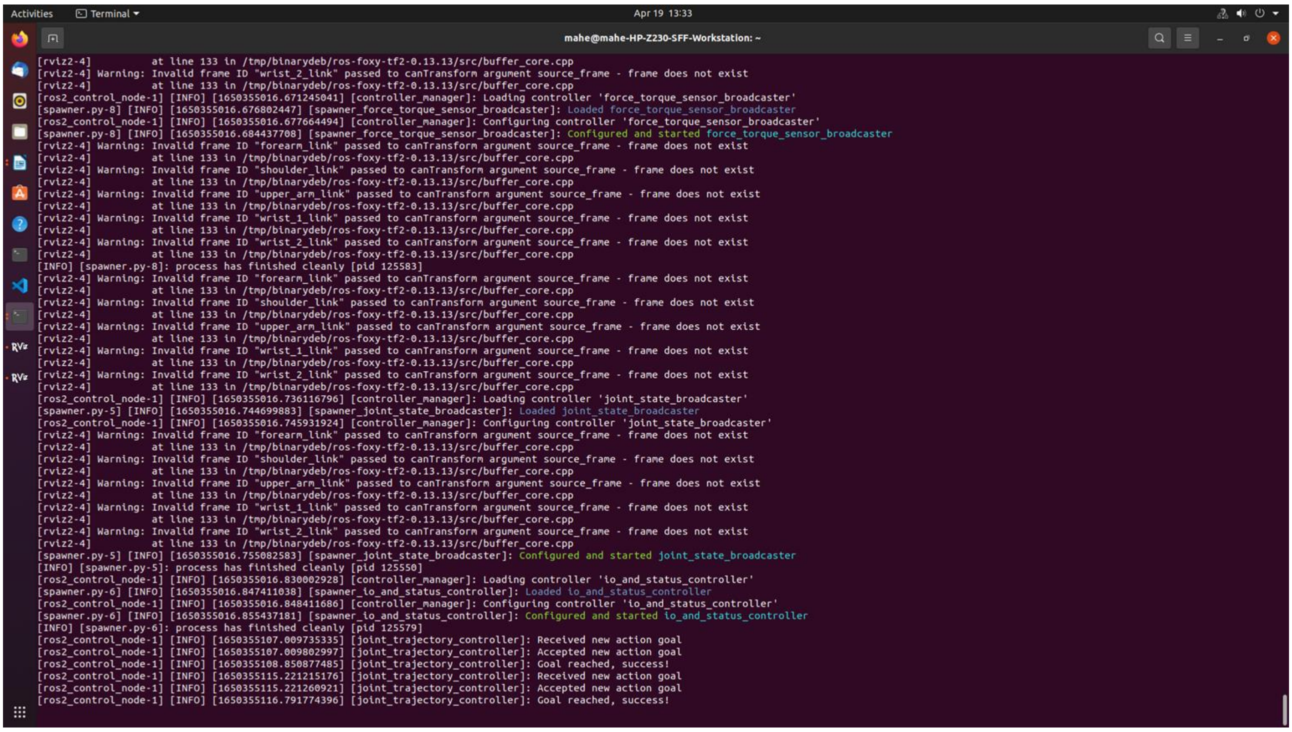

PROGRAM externalcontrolros
INSTALLATION default
Program
externalcontrolros
Load Program
Status
Running
Robot Age
Days Hours Minutes Seconds
15 21 52 04
Variables
Variable Value
Con_Destack_com 0
Con_Stack_comp 0
Destack_comp 0
Line_1 p[-0.04589, -0.66494, 0.00081, -2.22145, 2.22138, -0.00006]
Line_2 p[0, 0, 0, 0, 0, 0]
Pallet_comp 0
Pick_Place_comp 0
Pickapp_p2d p[-0.24592, -0.65844, 0.0443, -2.2538, -2.18859, -0.00142]
Plane_1 p[-0.38634, -0.23534, 0.01575, 0.07683, 3.1282, 0.0098]
Plane_2 p[-0.09485, -0.35104, 0.34377, -0.00009, -0.00004, -0.00001]
Point_4 p[-0.25869, -0.50274, 0.26815, 2.83028, 1.25258, 0.00669]
Point_7 p[0, 0, 0, 0, 0, 0]
Point_8 p[-0.23772, -0.57994, 0.35321, 0.13688, 3.12071, -0.04817]
Posstack p[0.04439, -0.55077, 0.55009, -0.14017, 1.56929, -0.31195]
Stack_comp 0
cmd_servo_q [-1.56996, -1.08202, -1.75187, -1.86228, 1.57613, 0.00019]
cmd_servo_q_last [-1.56996, -1.08202, -1.75187, -1.86228, 1.57613, 0.00019]
cmd_servo_qd [0, 0, 0, 0, 0, 0]
cmd_servo_state 0
Show Waypoints
Running
Speed 100%
Simulation
msung Triple Camera

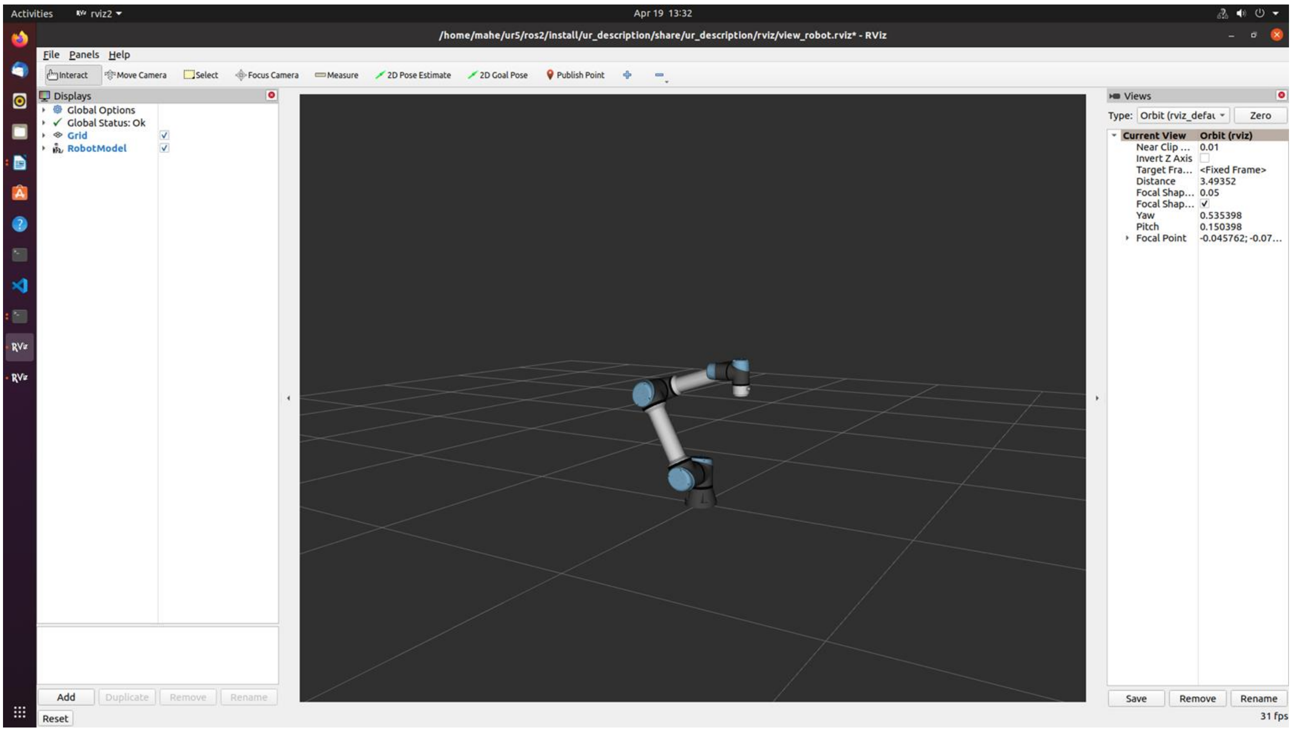

E-STOP
UNIVERSAL ROBOTS
PROGRAM externalcontrolros
INSTALLATION default
Program
externalcontrolros
Load Program
Status
Stopped
Robot Age
Days Hours Minutes Seconds
15 21 52 28
Variables
Variable Value
Con_Destack_com 0
Con_Stack_comp 0
Destack_comp 0
Line_1 p[-0.04589, -0.66494, 0.00081, -2.22145, 2.22138, -0.00006]
Line_2 p[0, 0, 0, 0, 0, 0]
Pallet_comp 0
Pick_Place_comp 0
Pickapp_p2d p[-0.24592, -0.65844, 0.0443, -2.2538, -2.18859, -0.00142]
Plane_1 p[-0.38634, -0.23534, 0.01575, 0.07683, 3.1282, 0.0098]
Plane_2 p[-0.09485, -0.35104, 0.34377, -0.00009, -0.00004, -0.00001]
Point_4 p[-0.25869, -0.50274, 0.26815, 2.83028, 1.25258, 0.00669]
Point_7 p[0, 0, 0, 0, 0, 0]
Point_8 p[-0.23772, -0.57994, 0.35321, 0.13688, 3.12071, -0.04817]
Posstack p[0.04439, -0.55077, 0.55009, -0.14017, 1.56929, -0.31195]
Stack_comp 0
cmd_servo_q [-1.56996, -1.08202, -1.75187, -1.86228, 1.57613, 0.00019]
cmd_servo_q_last [-1.56996, -1.08202, -1.75187, -1.86228, 1.57613, 0.00019]
cmd_servo_qd [0, 0, 0, 0, 0, 0]
cmd_servo_state 0
Show Waypoints
Normal
Speed 100%
Simulation
nsung Triple Camera


externalcontrolros*
default
Settings
Preferences
Password
System
System Backup
URCaps
Robot Registration
Remote Control
Constrained Freedrive
Network
Update
Exit
Network
Select your network method
DHCP
Static Address
Disabled network
Network is connected
Network detailed settings:
IP address
192.168.56.1
Subnet mask:
255.255.255.0
Default gateway:
0.0.0.0
Preferred DNS server:
0.0.0.0
Alternative DNS server:
0.0.0.0
Apply
Normal
Speed
100%
Simulation



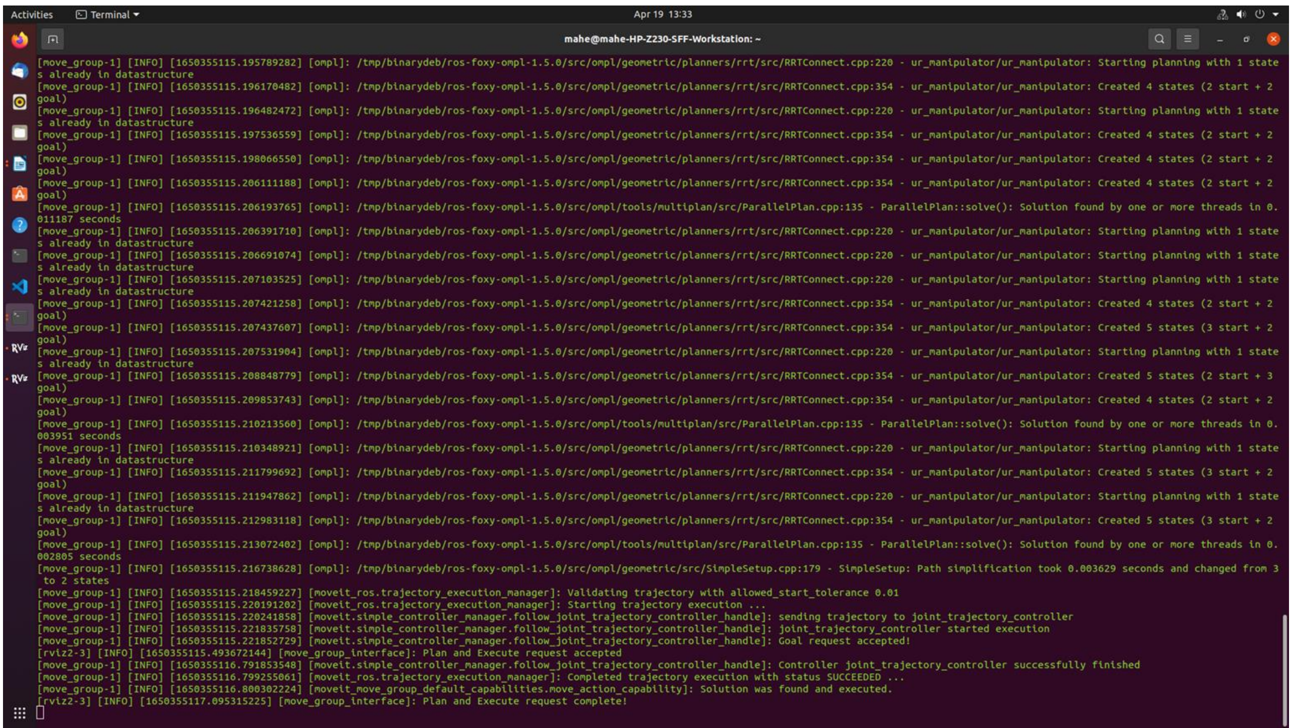
ros2 launch ur_bringup ur5e.launch.py robot_ip:=192.168.56.1

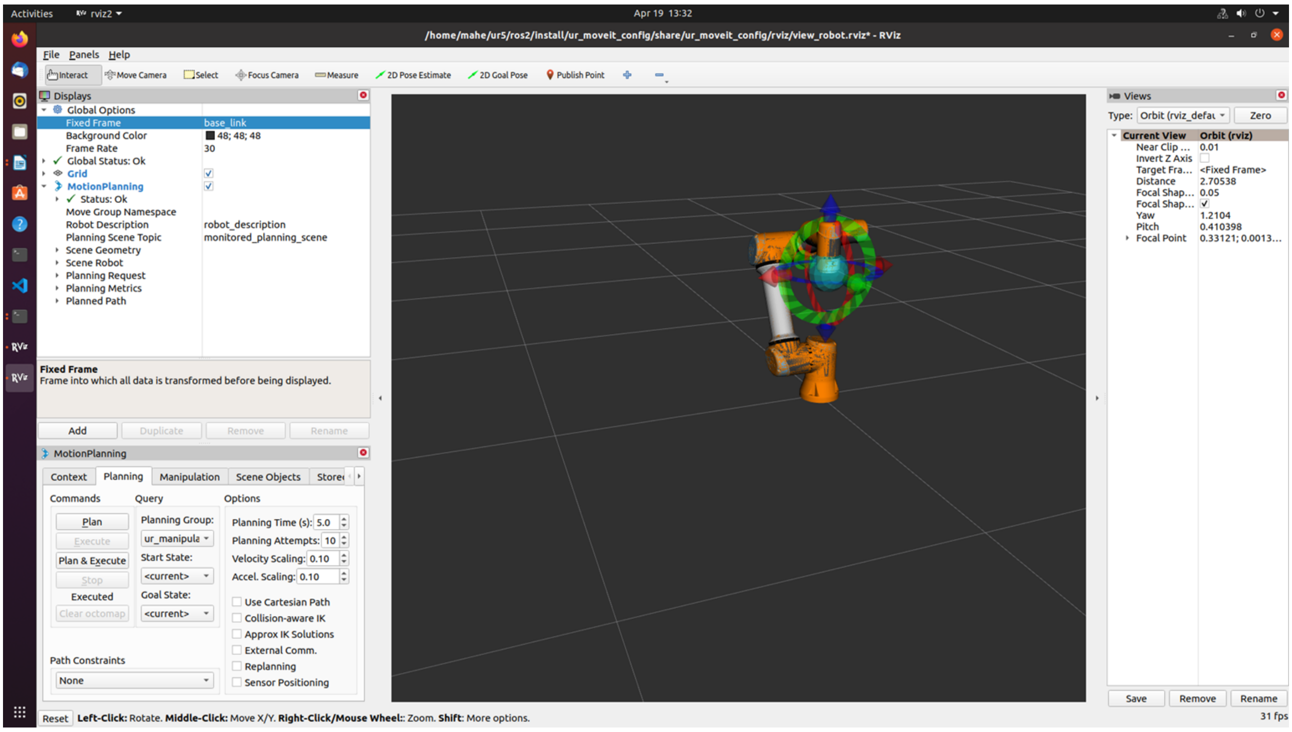




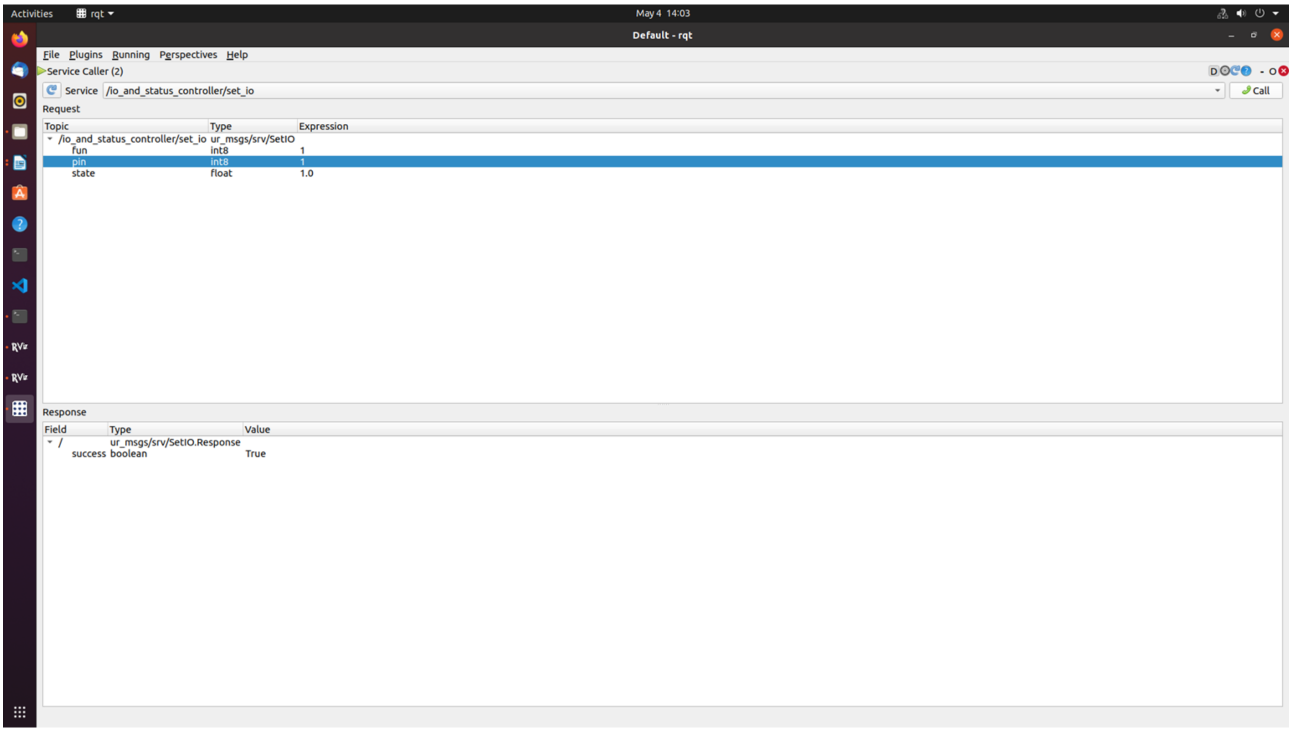
Set in manual mode (1234)
Open externalcontrolros program
Run the program

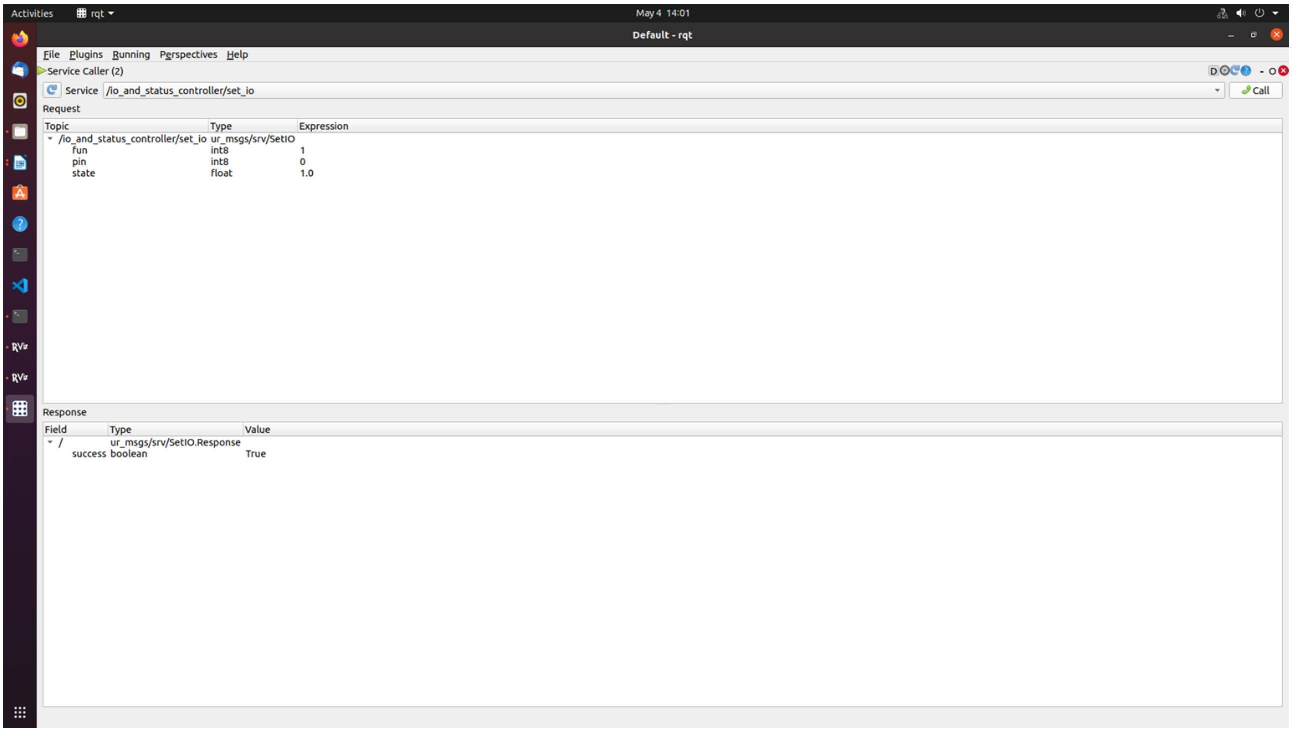
ros2 launch ur_bringup ur_moveit.launch.py ur_type:=ur5e robot_ip:=192.168.56.1

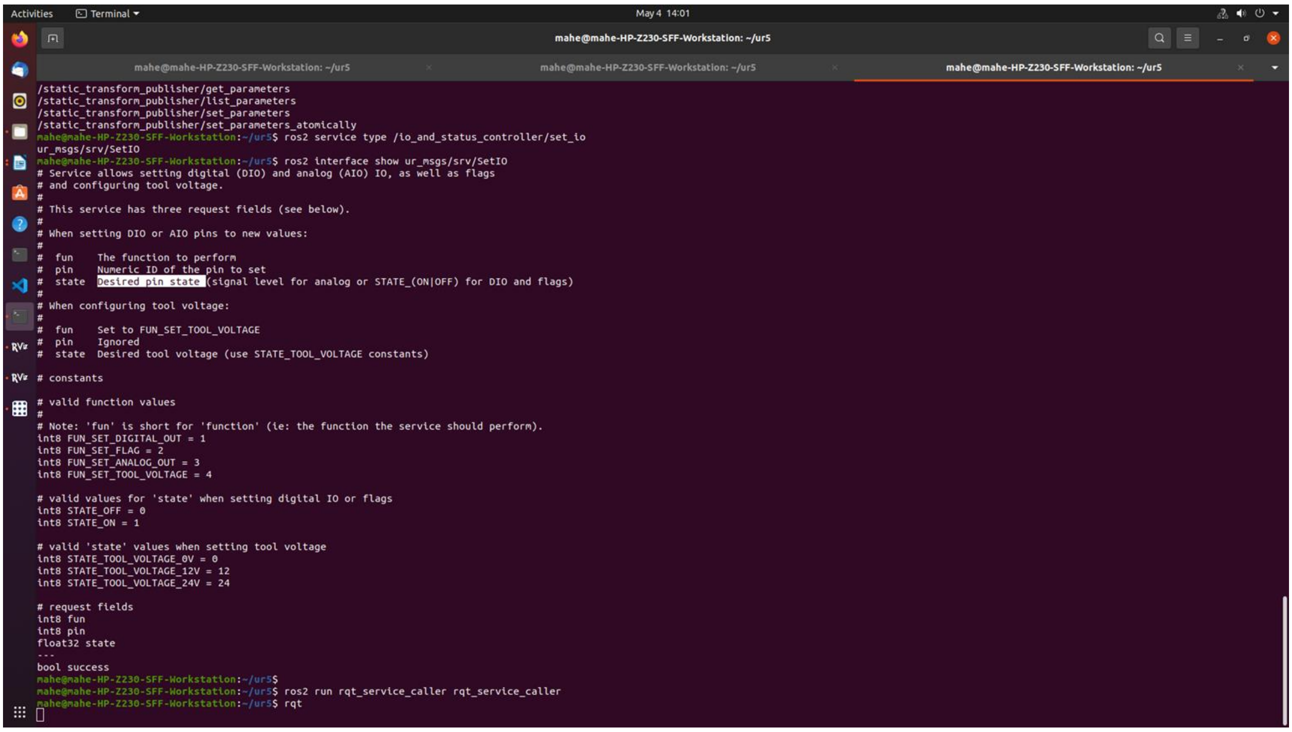




type >>
rqt
plugins→service caller
select io_and_status_controller/set_io
set the pin as 0 for gripper
state as 1.0 to set the gripper off
state as 0.0 to set the gripper off


Activities
rqt
May 4 14:03
Default - rqt
File Plugins Running Perspectives Help
Service Caller (2)
Service /io_and_status_controller/set_io
Call
Request
Topic Type Expression
/io_and_status_controller/set_io ur_msgs/srv/SetIO
fun int8 1
pin int8 1
state float 1.0
Response
Field Type Value
/ ur_msgs/srv/SetIO.Response
success boolean True


Activities
rqt
May 4 14:01
Default - rqt
File Plugins Running Perspectives Help
Service Caller (2)
Service /io_and_status_controller/set_io
Call
Request
Topic Type Expression
/io_and_status_controller/set_io ur_msgs/srv/SetIO
fun int8 1
pin int8 0
state float 1.0
Response
Field Type Value
/ ur_msgs/srv/SetIO.Response
success boolean True



```
/static_transform_publisher/get_parameters
/static_transform_publisher/list_parameters
/static_transform_publisher/set_parameters
/static_transform_publisher/set_parameters_atomically
mahe@mahe-HP-Z230-SFF-Workstation:~/ur5$ ros2 service type /io_and_status_controller/set_io
ur_msgs/srv/SetIO
mahe@mahe-HP-Z230-SFF-Workstation:~/ur5$ ros2 interface show ur_msgs/srv/SetIO
# Service allows setting digital (DIO) and analog (AIO) IO, as well as flags
# and configuring tool voltage.
#
# This service has three request fields (see below).
#
# When setting DIO or AIO pins to new values:
#
#  fun    The function to perform
#  pin    Numeric ID of the pin to set
#  state  Desired pin state (signal level for analog or STATE_(ON|OFF) for DIO and flags)
#
# When configuring tool voltage:
#
#  fun    Set to FUN_SET_TOOL_VOLTAGE
#  pin    Ignored
#  state  Desired tool voltage (use STATE_TOOL_VOLTAGE constants)

# constants

# valid function values
#
# Note: 'fun' is short for 'function' (ie: the function the service should perform).
int8 FUN_SET_DIGITAL_OUT = 1
int8 FUN_SET_FLAG = 2
int8 FUN_SET_ANALOG_OUT = 3
int8 FUN_SET_TOOL_VOLTAGE = 4

# valid values for 'state' when setting digital IO or flags
int8 STATE_OFF = 0
int8 STATE_ON = 1

# valid 'state' values when setting tool voltage
int8 STATE_TOOL_VOLTAGE_0V = 0
int8 STATE_TOOL_VOLTAGE_12V = 12
int8 STATE_TOOL_VOLTAGE_24V = 24

# request fields
int8 fun
int8 pin
float32 state
---
bool success
mahe@mahe-HP-Z230-SFF-Workstation:~/ur5$
mahe@mahe-HP-Z230-SFF-Workstation:~/ur5$ ros2 run rqt_service_caller rqt_service_caller
mahe@mahe-HP-Z230-SFF-Workstation:~/ur5$ rqt
```

Edit test_goal_publishers_config file as

```
publisher_scaled_joint_trajectory_controller:
  ros__parameters:

    controller_name: "scaled_joint_trajectory_controller"
    wait_sec_between_publish: 6

    goal_names: ["pos1", "pos2", "pos3", "pos4"]
    pos1: [-1.76, -2.129, -1.117, -1.483, 1.57, 0.1745]
    pos2: [-1.76, -2.3911, -1.448, -0.8726, 1.57, 0.17]
    pos3: [-1.76, -2.129, -1.117, -1.483, 1.57, 0.1745]
    pos4: [-1.76, -2.3911, -1.448, -0.8726, 1.57, 0.17]

    joints:
      - shoulder_pan_joint
      - shoulder_lift_joint
      - elbow_joint
      - wrist_1_joint
      - wrist_2_joint
      - wrist_3_joint
```

```
    check_starting_point: true
    starting_point_limits:
      shoulder_pan_joint: [-3.14,3.14]
      shoulder_lift_joint: [-3.14,3.14]
      elbow_joint: [-3.14,3.14]
      wrist_1_joint: [-3.14,3.14]
      wrist_2_joint: [-3.14,3.14]
      wrist_3_joint: [-3.14,3.14]

publisher_joint_trajectory_controller:
  ros__parameters:

    controller_name: "joint_trajectory_controller"
    wait_sec_between_publish: 6

    goal_names: ["pos1", "pos2", "pos3", "pos4"]
    pos1: [-1.76, -2.129, -1.117, -1.483, 1.57, 0.1745]
    pos2: [-1.76, -2.3911, -1.448, -0.8726, 1.57, 0.17]
    pos3: [-1.76, -2.129, -1.117, -1.483, 1.57, 0.1745]
    pos4: [-1.76, -2.3911, -1.448, -0.8726, 1.57, 0.17]

    joints:
      - shoulder_pan_joint
      - shoulder_lift_joint
      - elbow_joint
      - wrist_1_joint
      - wrist_2_joint
      - wrist_3_joint

    check_starting_point: true
    starting_point_limits:
      shoulder_pan_joint: [-3.14,3.14]
      shoulder_lift_joint: [-3.14,3.14]
      elbow_joint: [-3.14,3.14]
      wrist_1_joint: [-3.14,3.14]
      wrist_2_joint: [-3.14,3.14]
      wrist_3_joint: [-3.14,3.14]
```

```
cd ur5
colcon build –packages-select ur_bringup
ros2 launch ur_bringup test_joint_trajectory_controller.launch.py
```

Content

1. Preliminaries

1.1 ROS 2 distribution

ROS 2 Foxy is highly recommended.

(ROS 2 Foxy Fitzroy, released June 5th, 2020, supported until May 2023) Installation: https://docs.ros.org/en/foxy/Installation/Ubuntu-Development-Setup.html

1.2 Ubuntu system

To match ROS 2 Foxy distribution, Ubuntu 20.04 is required. Installation: https://ubuntu.com/download/desktop

2. Introduction

This introduction is based on the official universal robot ROS 2 driver:
https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver

2.1 Download and build ROS 2 packages

Follow the steps below to build the required ROS 2 packages:

Step 1:
# source global ROS 2
$ gedit ~/.bashrc
At the last line, add "source ~/ros2_foxy/install/local_setup.bash"
$ source ~/.bashrc

Step 2:
# create a new ROS 2 workspace
$ export COLCON_WS=~/workspace/ros_ws_foxy_ur_driver
$ mkdir -p $COLCON_WS/src

Step 3:
Pull relevant packages, install dependencies, compile, and source the workspace
$ cd $COLCON_WS
$ git clone
https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver.
git src/Universal_Robots_ROS2_Driver
$ vcs import src --skip-existing --input
src/Universal_Robots_ROS2_Driver/Universal_Robots_ROS2_Driver
.repos
$ rosdep install --ignore-src --from-paths src -y -r
$ colcon build --cmake-args -DCMAKE_BUILD_TYPE=Release
$ source install/setup.bash

Step 4:
To use MoveIt, some additional packages should be added into workspace
$ cd $COLCON_WS
$ vcs import src --skip-existing --input
src/Universal_Robots_ROS2_Driver/MoveIt_Support.r
epos
$ vcs import src --skip-existing --input src/moveit2/moveit2.repos
$ rosdep install --ignore-src --from-paths src -y -r

$ colcon build --cmake-args -DCMAKE_BUILD_TYPE=Release

$ source install/setup.bash

2.2 Hardware setup: setting a ur_5e robot

2.2.1 Preparation

To enable external control of the UR robot from a control PC, you need to install the externalcontrol-1.0.5.urcap which can be found inside the resources folder of this driver: (ros_ws_foxy_ur_driver□src□Universal_Robots_ROS2_Driver□ur_r obot_driver□resources)
or download the latest from Universal_Robots_ExternalControl_URCap:
https://github.com/UniversalRobots/Universal_Robots_ExternalControl_URCap/relea ses
Note: For installing this URCap, a minimal PolyScope version 5.1 (for e-Series) is necessary.

2.2.2 Install an URCap on an e-Series robot

For installing the necessary URCap and creating a program, please see the individual tutorial on how to setup a CB3 robot or how to setup an e-Series robot

To install it you first have to copy it to the robot's programs folder which can be done using a USB stick.

Step 1:
On the welcome screen, click on the hamburger menu in the top-right corner and select Settings to enter the robot's setup. Select System and then URCaps to enter the URCaps installation screen.

Step 2:
Click the little plus sign at the bottom to open the file selector. You should see all URCap files stored inside the robot's programs folder. Select and open the externalcontrol-1.0.5.urcap file. Your URCaps view should now show the External Control in the list of active URCaps and a notification to restart the robot.

Step 3:
After the reboot you should find the External Control in URCaps tag inside Installation.

Step 4:
You should setup the IP address of the external PC which will be running the ROS 2 driver. Note that the robot and the external PC have to be in the same network, ideally in a direct connection with each other to minimize network disturbances. The custom port should be left untouched for now.

Step 5:
To use the new URCaps, create a new program and insert the External Controlprogram node into the program tree.

If you click on the command tab again, you'll see the settings entered inside the Installation. Check that they are correct, and then save the program. Your robot is now ready to be used together with this driver.

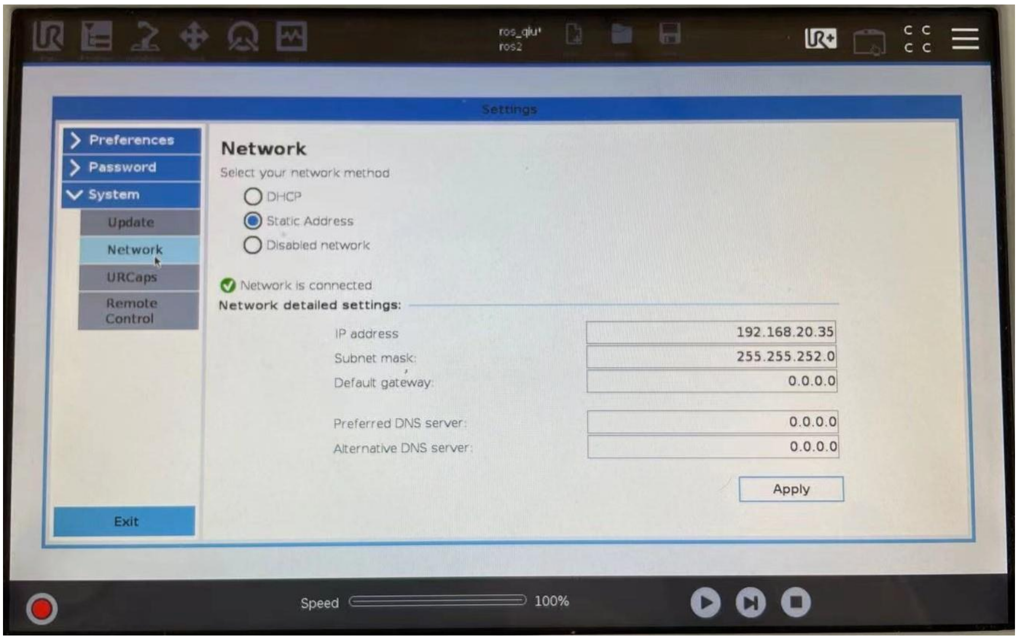
3. Examples

3.1 Preparation

First, the physical connection between the robot and the control PC should be established, e.g., connect the robot and the PC via a net cable.

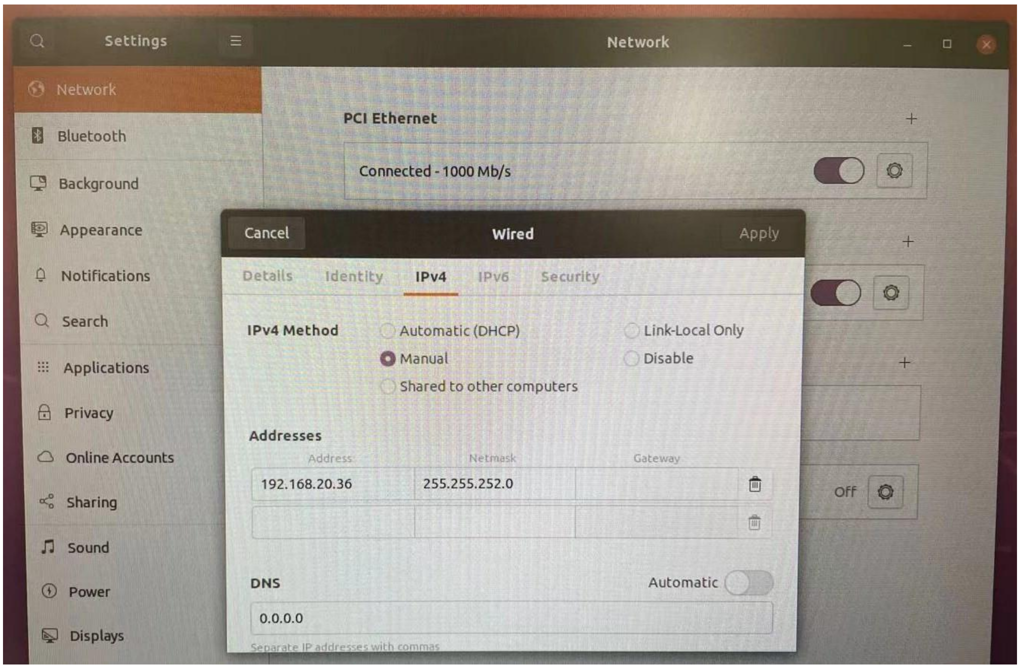
3.1.1 Set IP address of the robot

The IP address of the ur_5e robot is set as: 192.168.20.35.

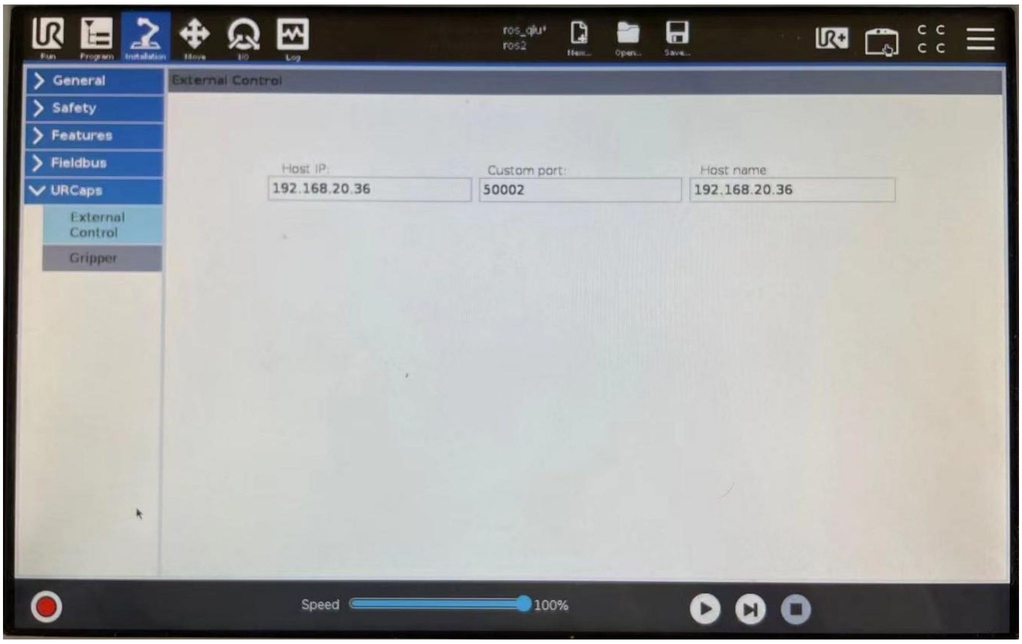


3.1.2 Set IP address of the control PC

First, configure the IP address of the control PC, which is set as: 192.168.20.36.



Second, update the information of External Control in URCaps in Installation.



As shown in the above figure, configure the Host IP and Host name.

3.2 Test joint trajectory controller

A ur_5e robot is controlled via joint_trajectory_controller by using a PC (ROS 2 Foxy with Ubuntu 20.04).

The following steps are recom-

mended: Step 1:

Power on the ur_5e robot, and confirm the connection between the robot and the PC.

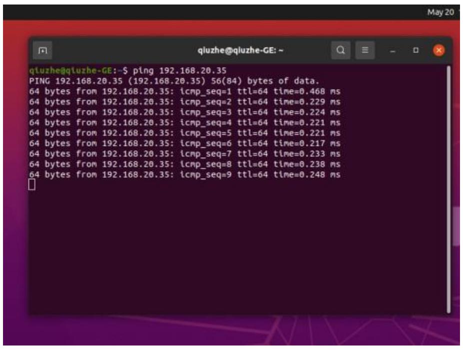
The IP address of the ur_5e robot is set as:

192.168.20.35. The IP address of the PC is set as:

192.168.20.36.

Open Terminal 1:

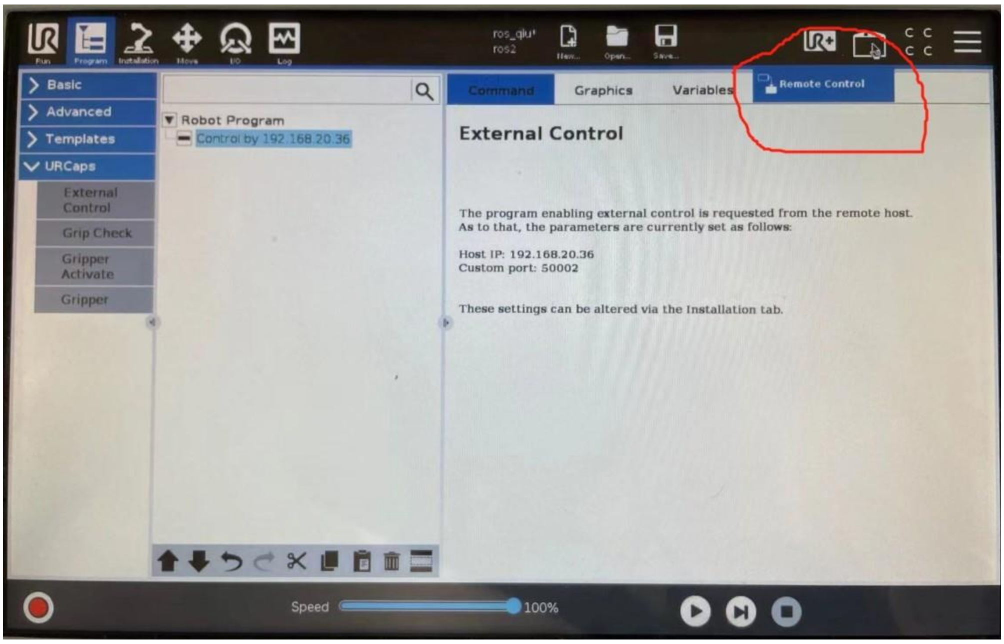
$ ping 192.168.20.35



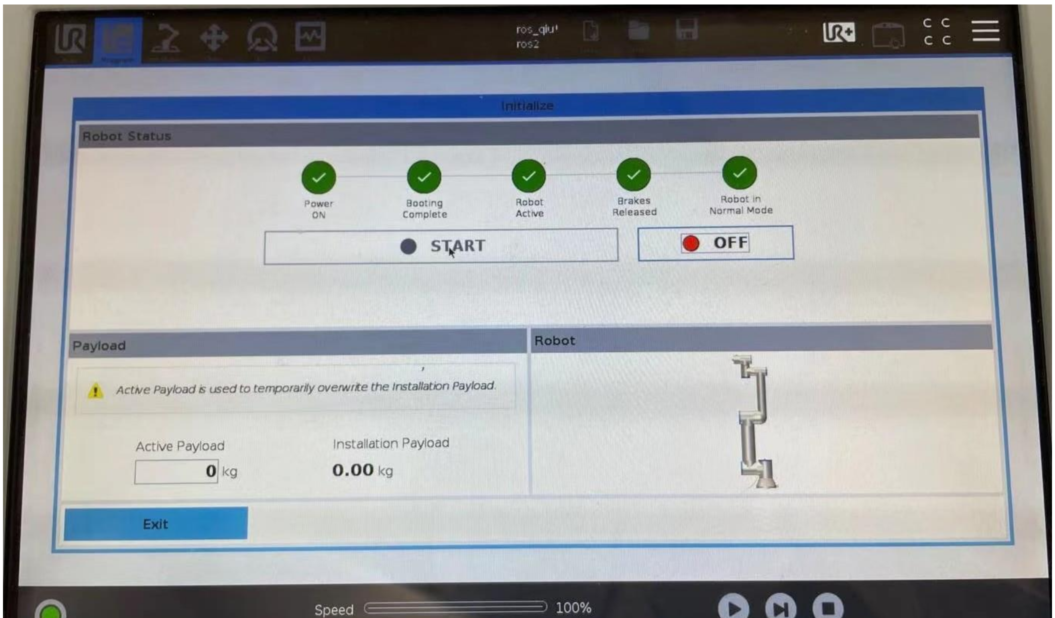
We can see the robot and the PC are successfully con-

nected. Step 2:

Set the robot control mode to the local control mode by using the teach-pendant.



As shown in the above figure, the local control mode has already set.
Step 3:
Start the robot.

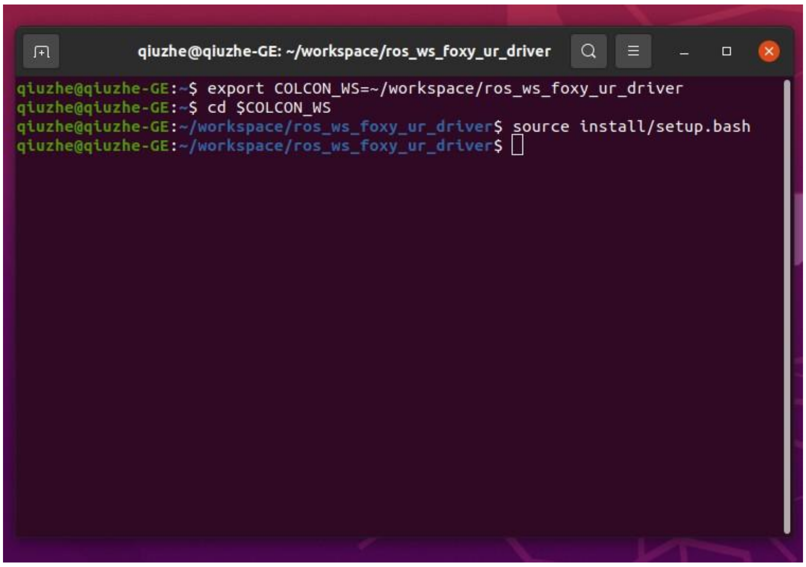


As shown in the above figure, the robot has already started. Step 4:
Start the robot driver. Remember source the bash file first. Open Terminal 2:

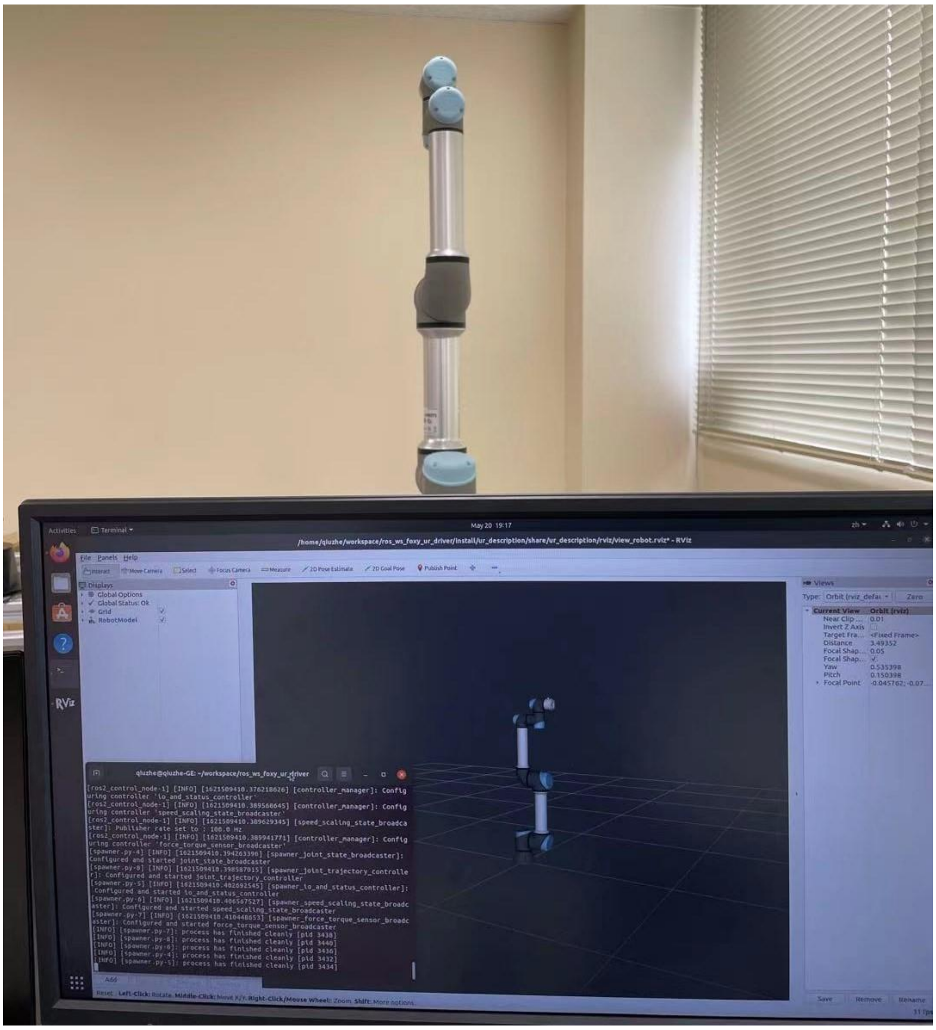
$ export COLCON_WS=~/workspace/ros_ws_foxy_ur_driver

$ cd $COLCON_WS

$ source install/setup.bash

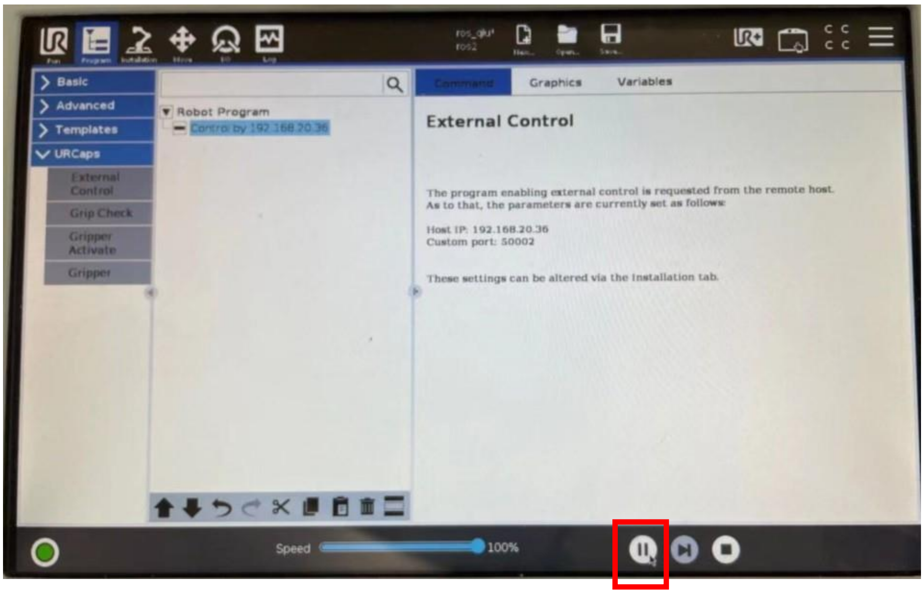


$ ros2 launch ur_bringup ur_control.launch.py ur_type:=ur5e robot_ip:=192.168.20.35 launch_rviz:=true



As shown in the above figure, the driver is successfully started.

Step 5:
Load Robot Program: External Control, and start it.



As shown in the above figure, the program is already started.

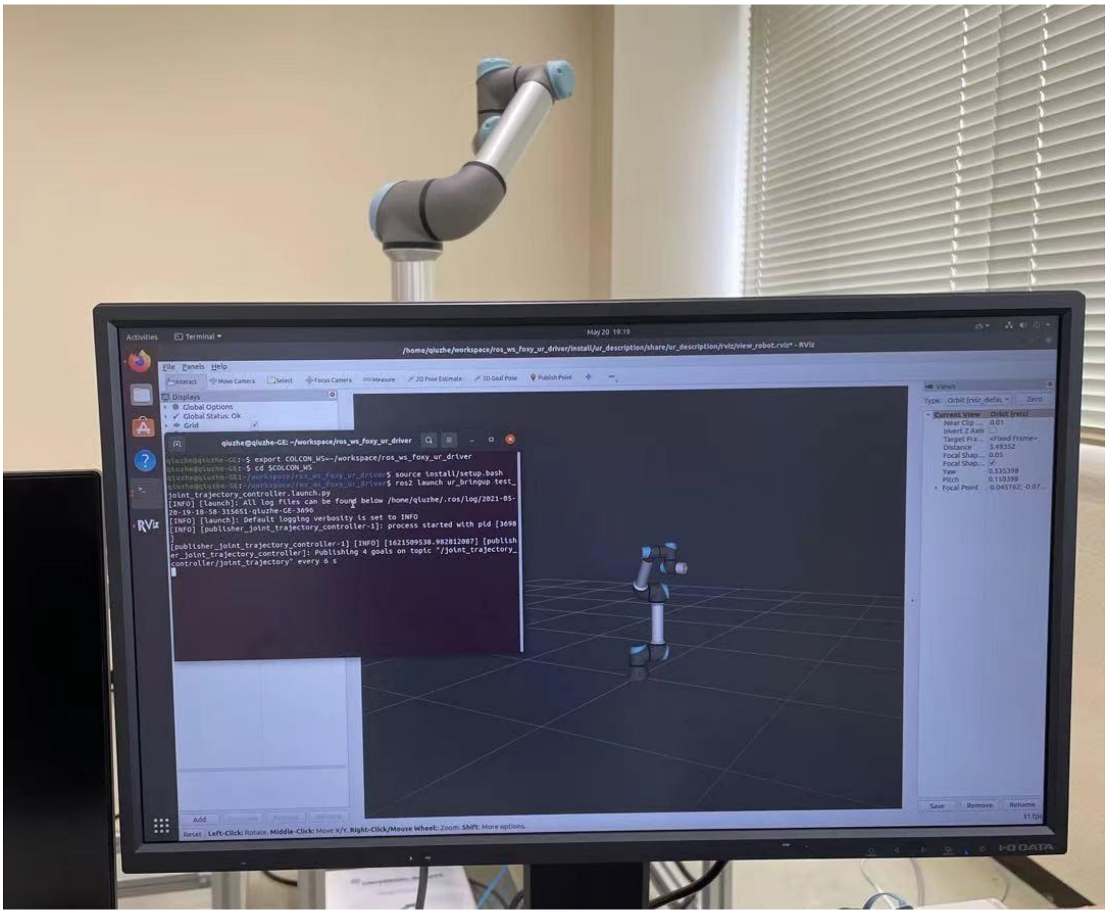
Step 6:
Start the Joint Trajectory Controller. Remember source the bash file first. Open Terminal 3:

$ export COLCON_WS=~/workspace/ros_ws_foxy_ur_driver

$ cd $COLCON_WS

$ source install/setup.bash

$ ros2 launch ur_bringup test_joint_trajectory_controller.launch.py



After a few seconds, the robot starts to move.

3.3 Test scaled_joint_trajectory_controller

A ur_5e robot is controlled via scaled_joint_trajectory_controller by using a PC (ROS 2 Foxy with Ubuntu 20.04).

The following steps are recom-

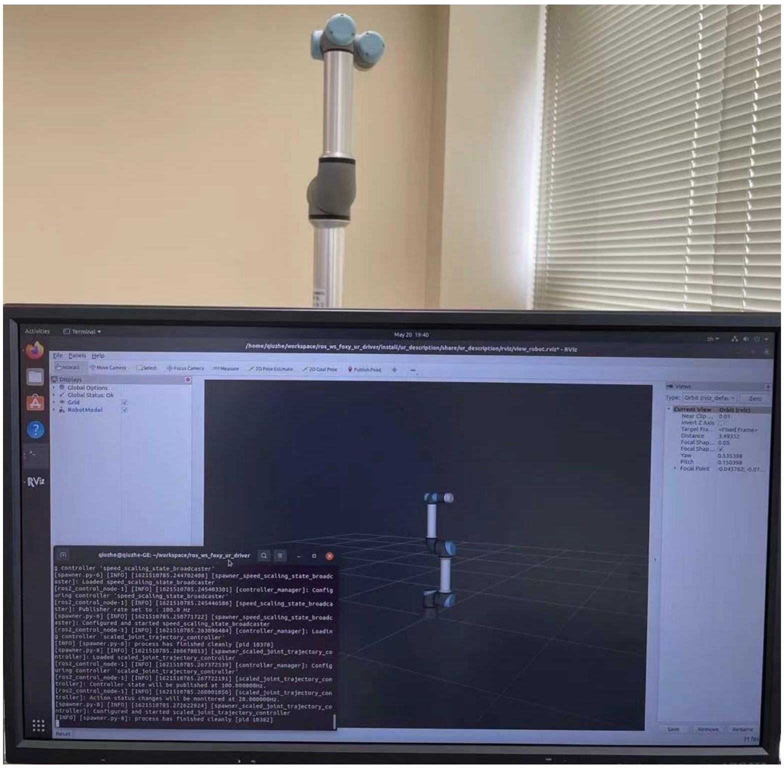
mended: Step 1:
Same as Step 1 of Example 3.2

Step 2:
Same as Step 2 of Example 3.2

Step 3:
Same as Step 3 of Example 3.2

Step 4:
Start the robot driver. Remember source the bash file first. Open Terminal 2:

```
$ export COLCON_WS=~/workspace/ros_ws_foxy_ur_driver
$ cd $COLCON_WS
$ source install/setup.bash
$ ros2 launch ur_bringup ur_control.launch.py ur_type:=ur5e robot_ip:=192.168.20.35 robot_controller:=scaled_joint_trajectory_controller
```



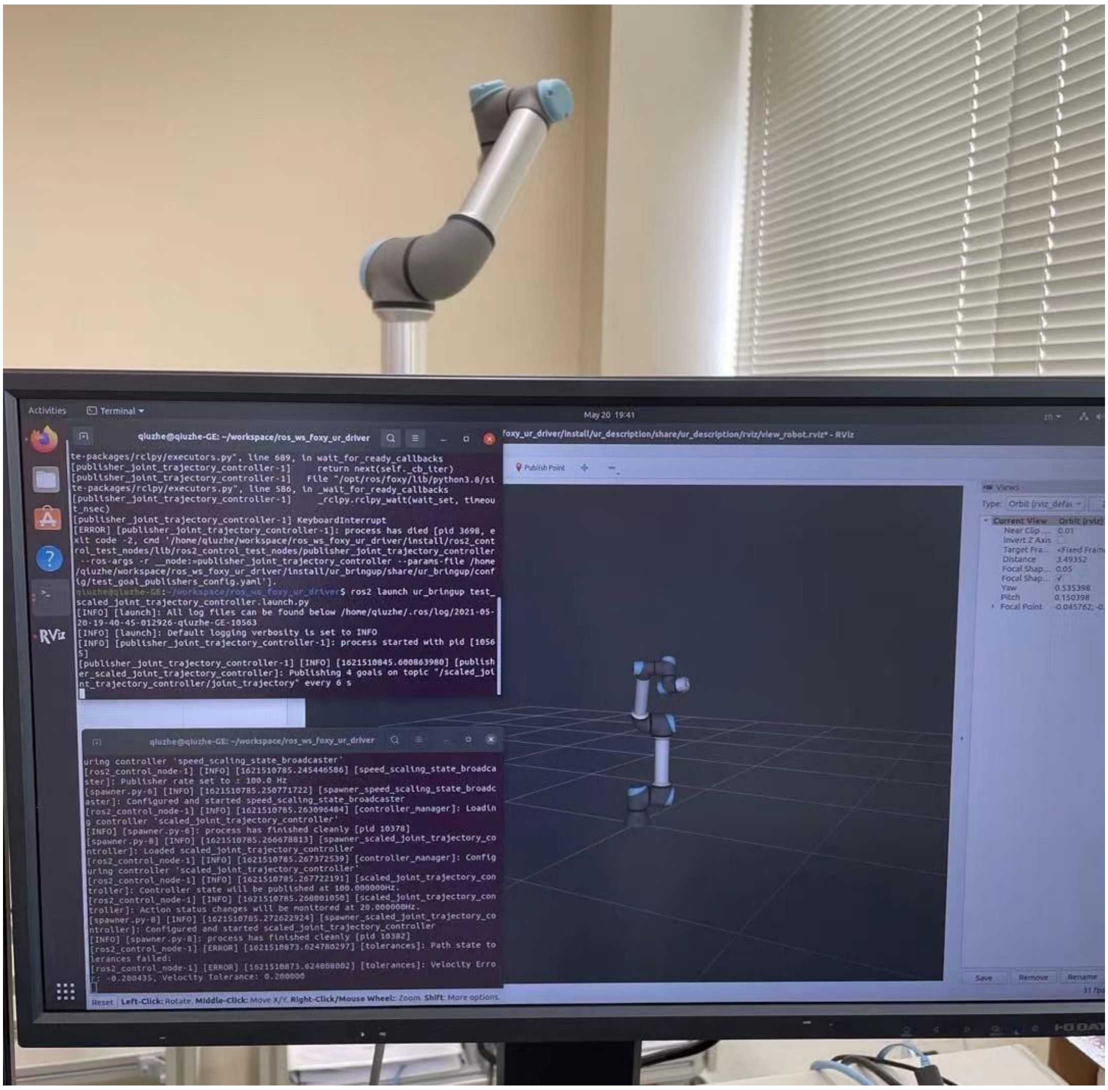
```
launch_rviz:=true
```

As shown in the above figure, the driver is already started.

Step 5:
Load Robot Program: External Control, and start it. Same as Step 5 of Example 3.2

Step 6:
Start the Scaled Joint Trajectory Controller. Remember source the bash file first. Open Terminal 3:

```
$ export COLCON_WS=~/workspace/ros_ws_foxy_ur_driver
$ cd $COLCON_WS
$ source install/setup.bash
$ ros2 launch ur_bringup test_scaled_joint_trajectory_controller.launch.py
```



After a few seconds, the robot starts to move.

3.4 Test MoveIt plugin

A ur_5e robot is controlled via MoveIt by using a PC (ROS 2 Foxy with Ubuntu 20.04).

The following steps are recom-

mended: Step 1:
Same as Step 1 of Example 3.2

Step 2:
Same as Step 2 of Example 3.2

Step 3:
Same as Step 3 of Example 3.2

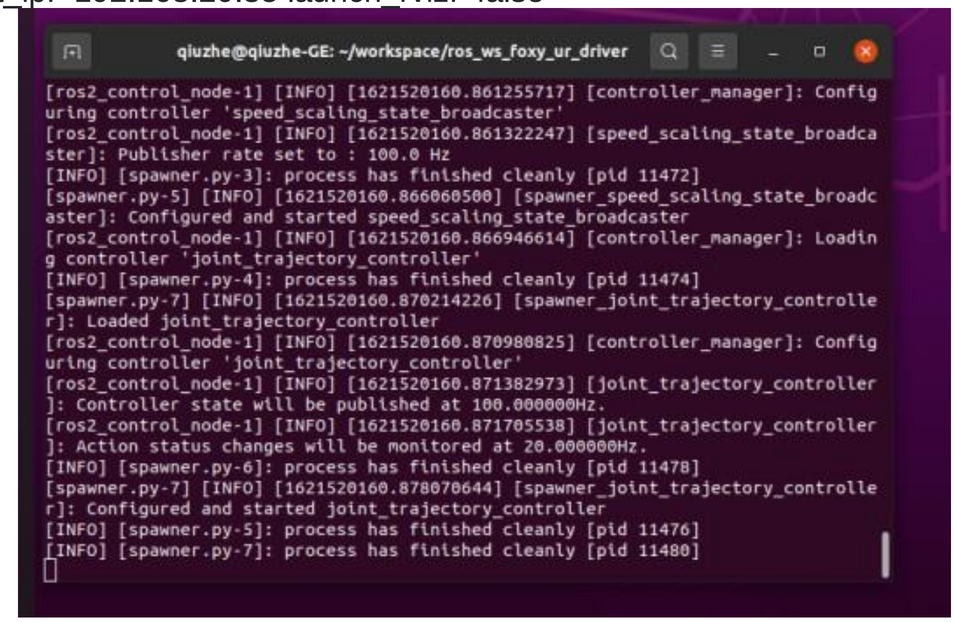
Step 4:
Start the robot driver. Remember source the bash file
first. Open Terminal 2:

$ export COLCON_WS=~/workspace/ros_ws_foxy_ur_driver

$ cd $COLCON_WS

$ source install/setup.bash

$ ros2 launch ur_bringup ur_control.launch.py ur_type:=ur5e robot_ip:=192.168.20.35 launch_rviz:=false



As shown in the above figure, the driver is successfully

started. Step 5:
Load Robot Program: External Control, and start it. Same as Step 5 of Example 3.2

Step 6:

Start the MoveIt example. Remember source the bash
file first. Open Terminal 3:

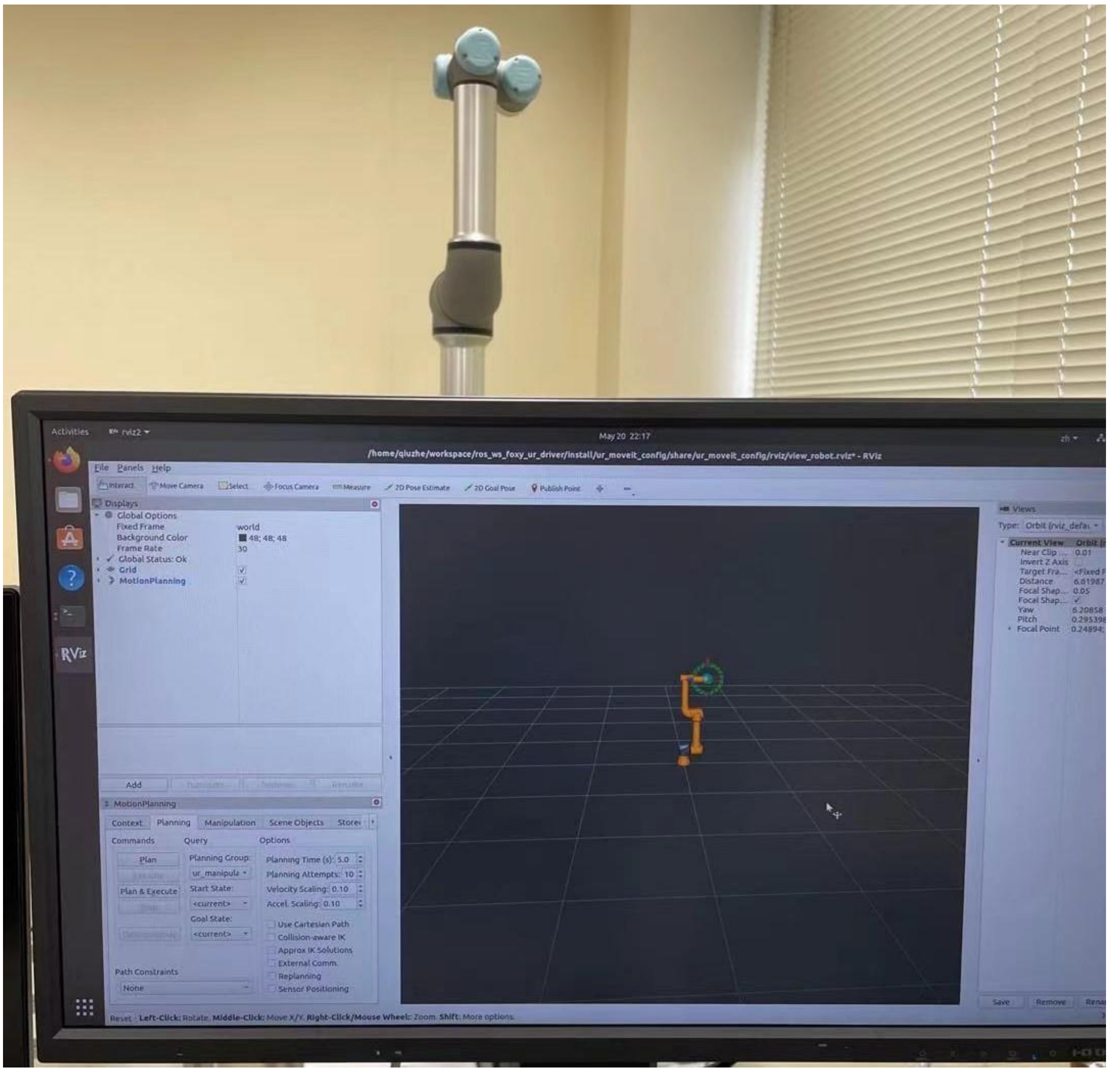
$ export COLCON_WS=~/workspace/ros_ws_foxy_ur_driver

$ cd $COLCON_WS

$ source install/setup.bash

$ ros2 launch ur_bringup ur_moveit.launch.py

ur_type:=ur5e robot_ip:=192.168.20.35 launch_rviz:=true

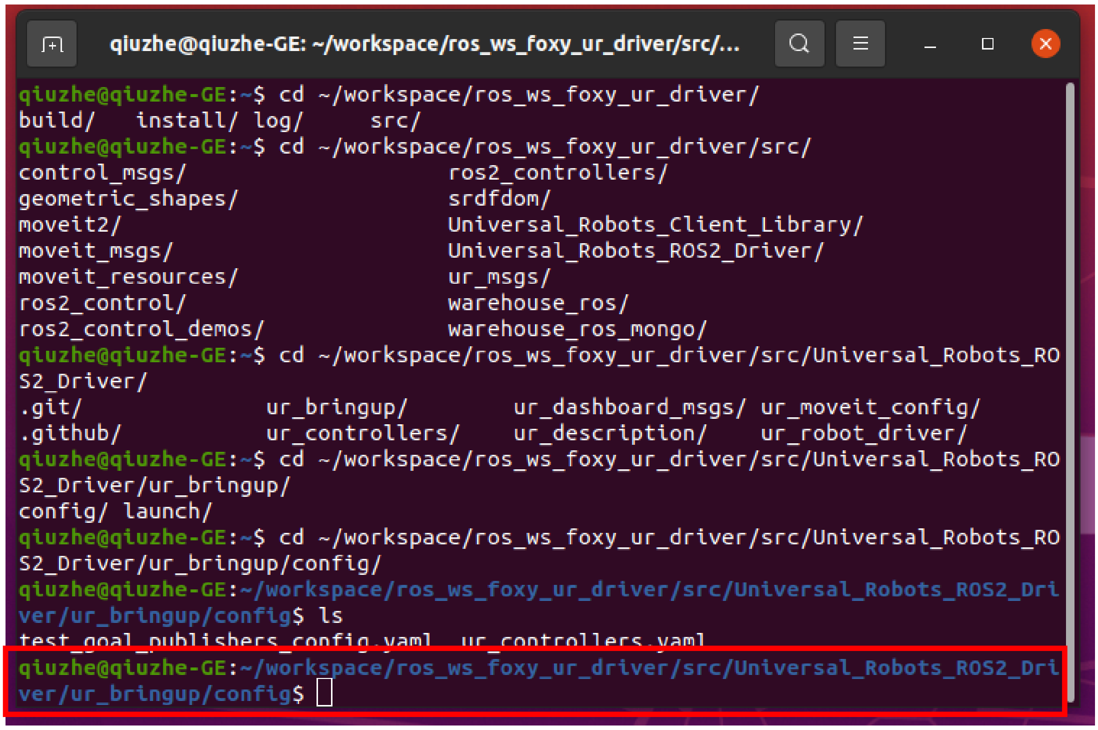


As shown in the above figure, now you can use the MoveIt Plugin in rviz2 to plan and execute trajectories with the robot.

3.5 Modified ROS 2 package of scaled-/joint trajectory controller

A ur_5e robot is controlled via modified joint_trajectory_controller by using a PC (ROS 2 Foxy with Ubuntu 20.04). Similarly, the same setting can be applied to the scaled_joint_trajectory_controller for controlling the ur_5e robot.

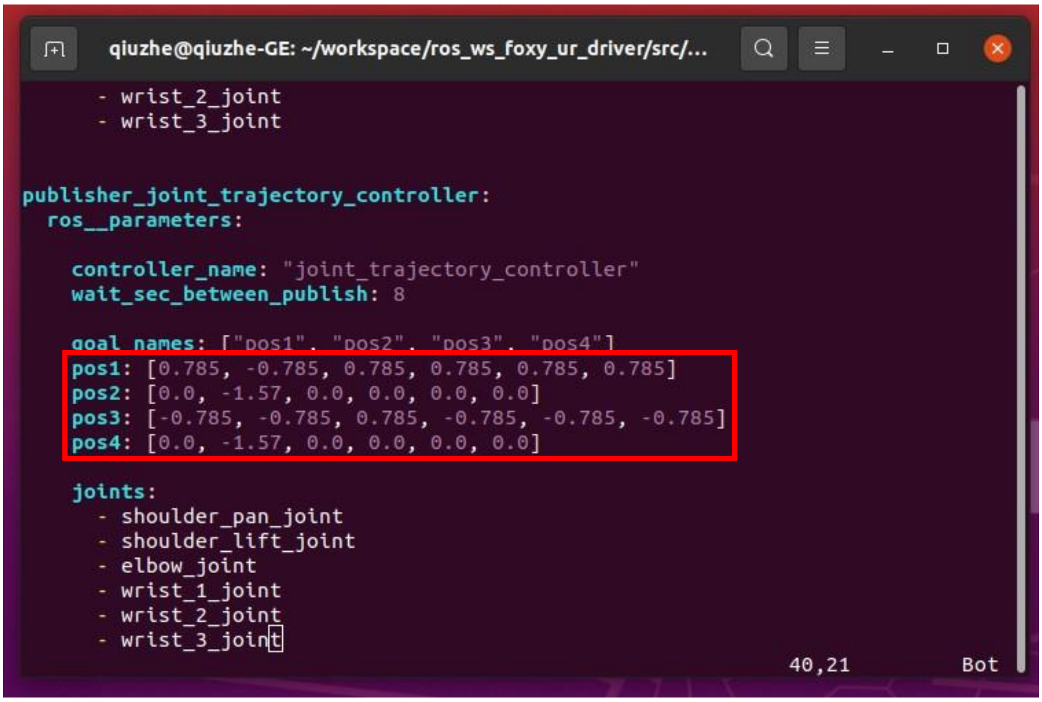
The following steps are recom-
mended: Step 1:
Set desired position of each joint
Open Terminal 1:
# Find the controller config file.
$ cd ~
/workspace/ros_ws_foxy_ur_driver/src/Universal_Robots_ROS2_Driver/



ur_bringup/config

# Modify the controller config file: change desired position of each joint.
$ vim test_goal_publishers_config.yaml

Six values of “pos” mean the desired position associated with six joints. In addition, “pos1-4” mean four desired joint trajectories for the ur_5e robot.



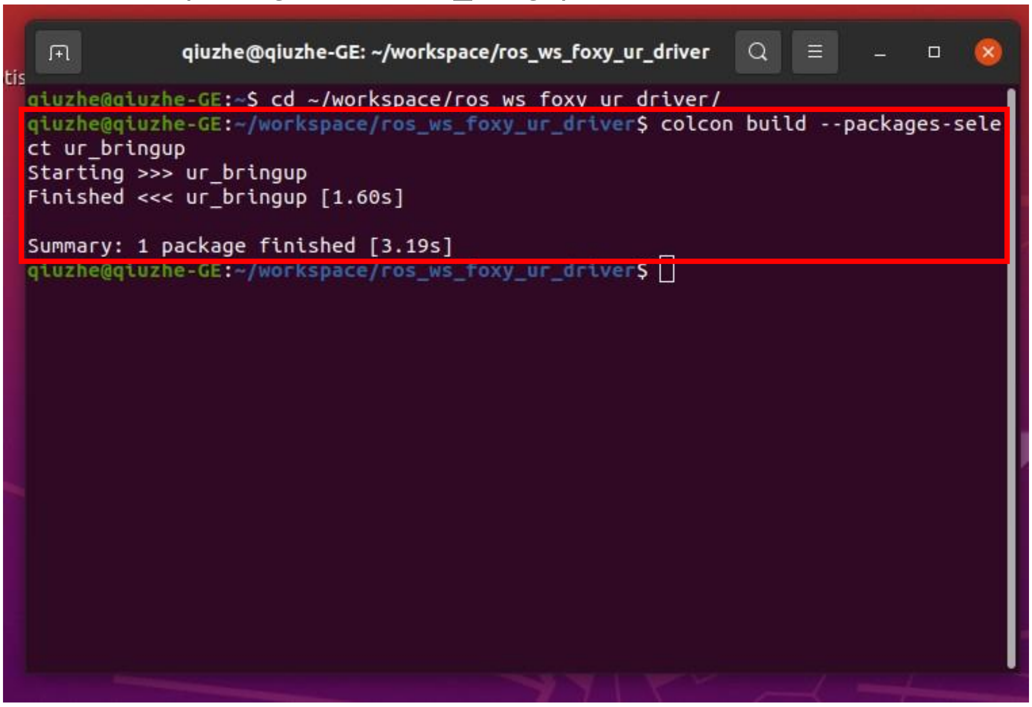
Step2:

The package should be recompiled after modification In Terminal 1:

$ cd ~ /workspace/ros_ws_foxy_ur_driver

Only compile the modified package to save time

$ colcon build –packages-select ur_bringup



Then, refer to Step1 to Step6 of Sec 3.2 to control the ur_5e robot. The performances of the joint trajectory controller are shown as follows.