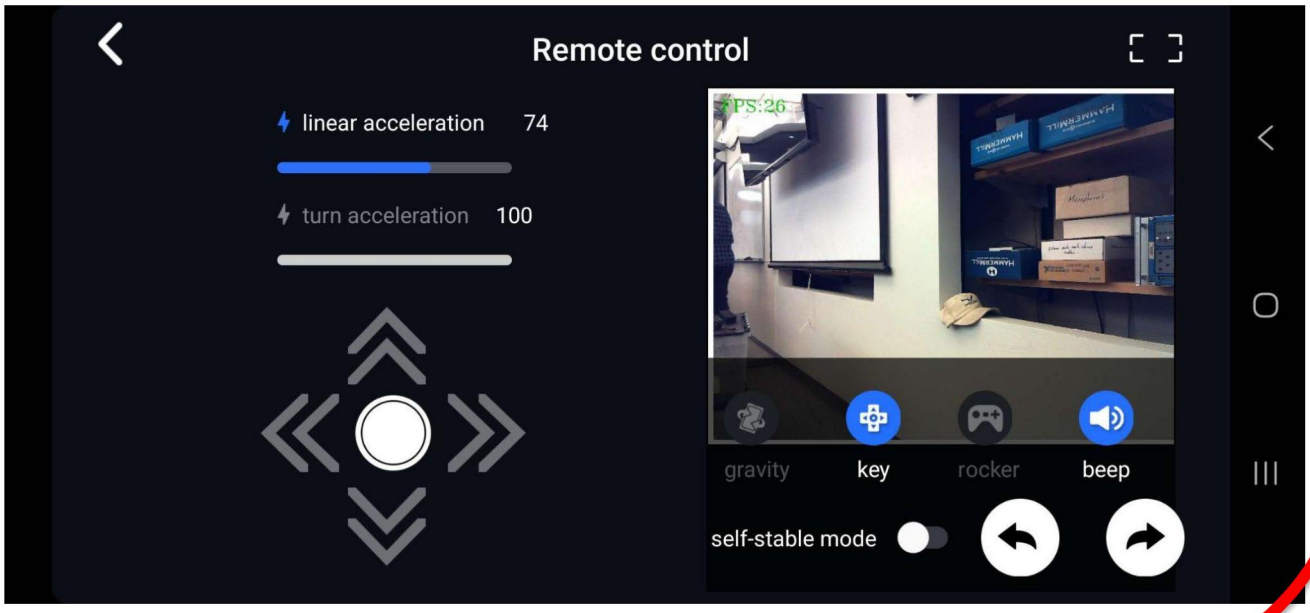
LAB5 Deliverables
It contains everything for lab5
The specified deliverables are highlighted with large red squares

Task1

Testing with the App

Action4:

Action5:

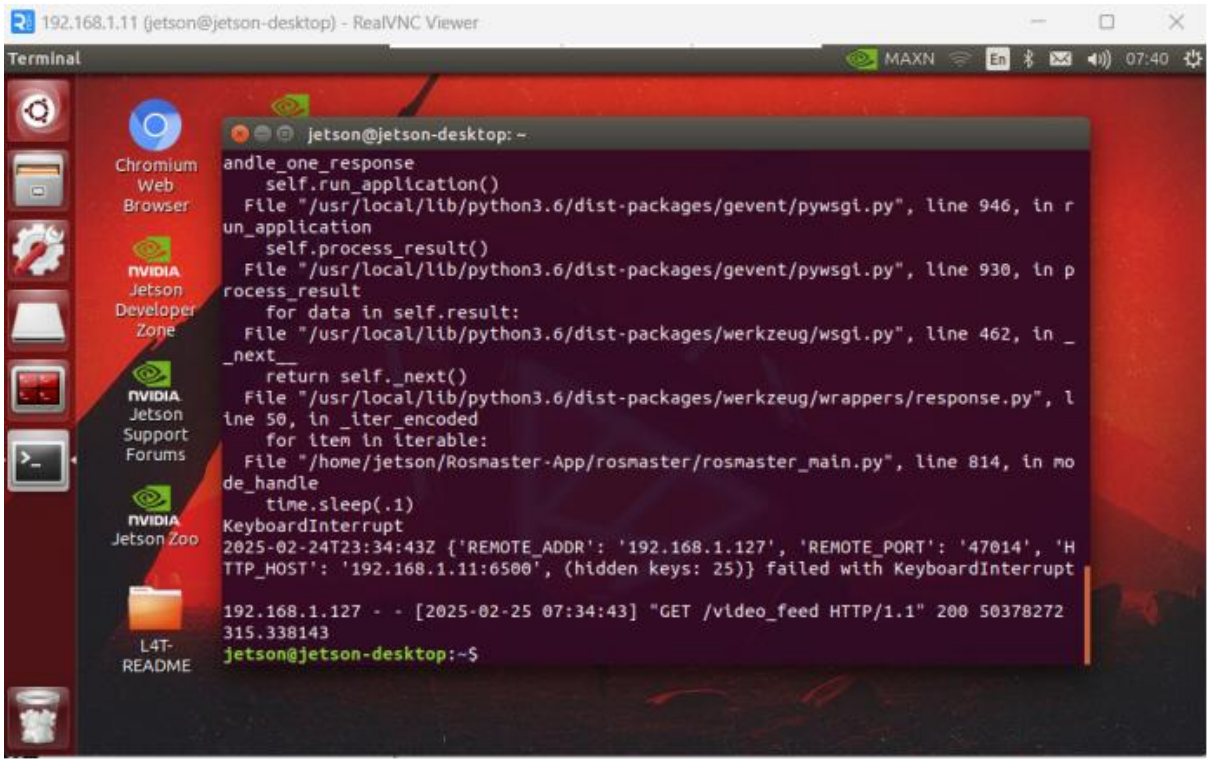


REQUIRED Question: Based on all of this testing, what are some pros and cons of using omni-directional wheels?

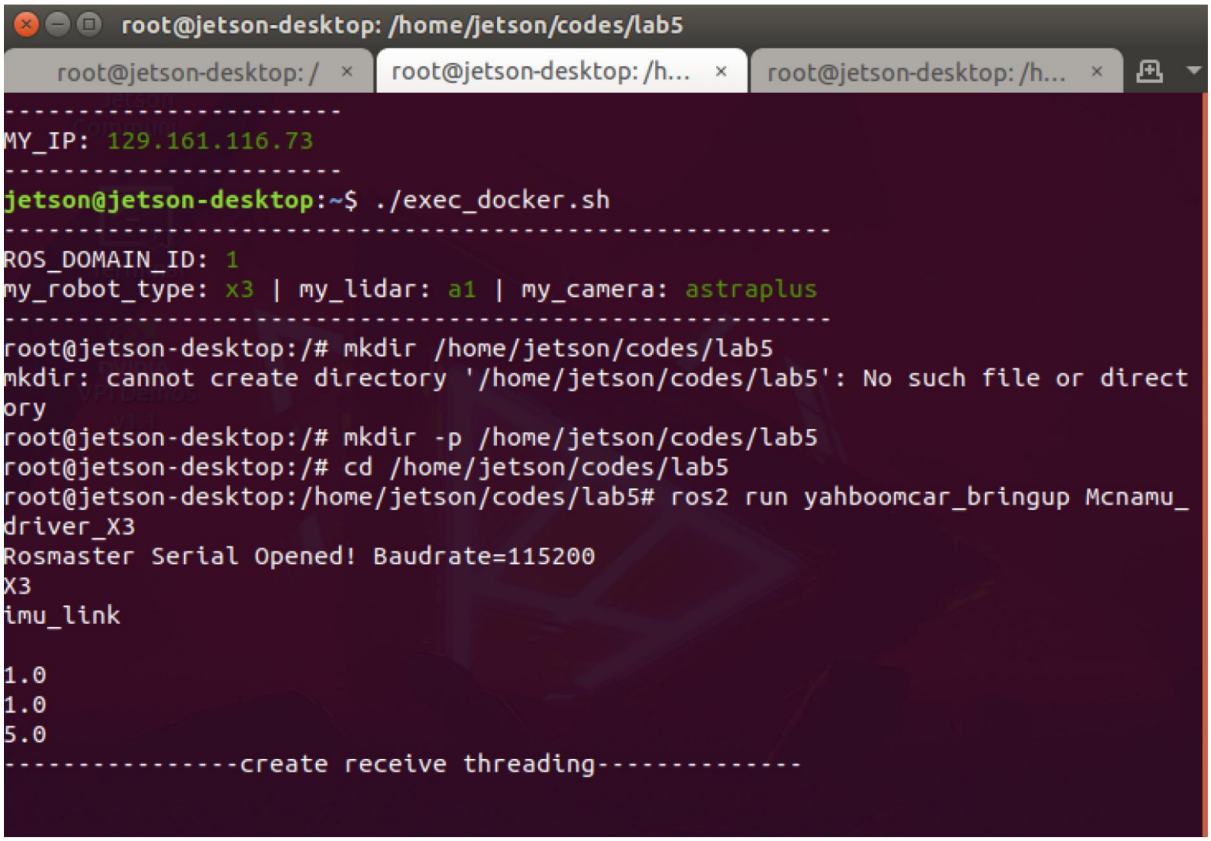
Omnidirectional wheels can move forward, backward, sideways, and rotate in place. They are used in places with limited space or frequent direction changes. But the contact area with the ground is small, so on uneven or smooth surfaces, they may slip or shake. When the load is large or the ground is not flat, energy use increases, and load capacity is lower than traditional wheels.

Using Docker in VNC Viewer

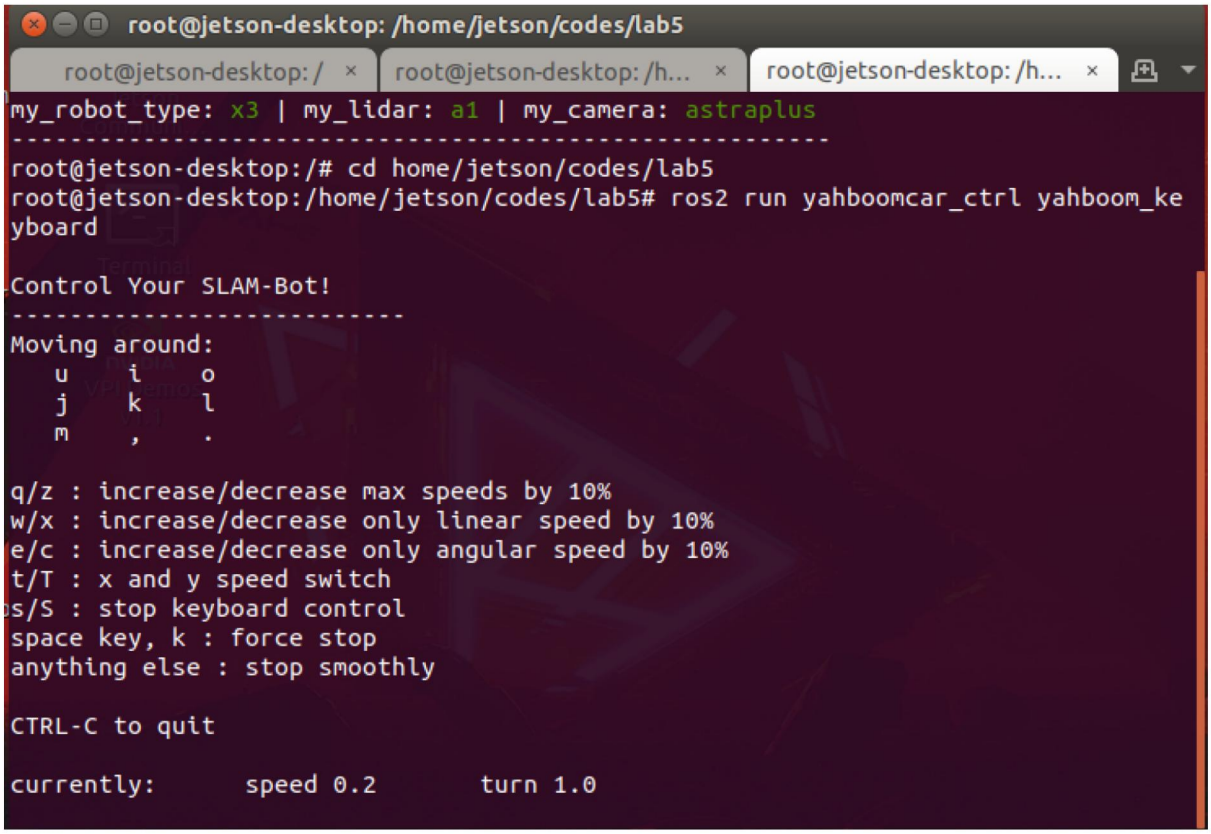
Action2



Action4 and Action7



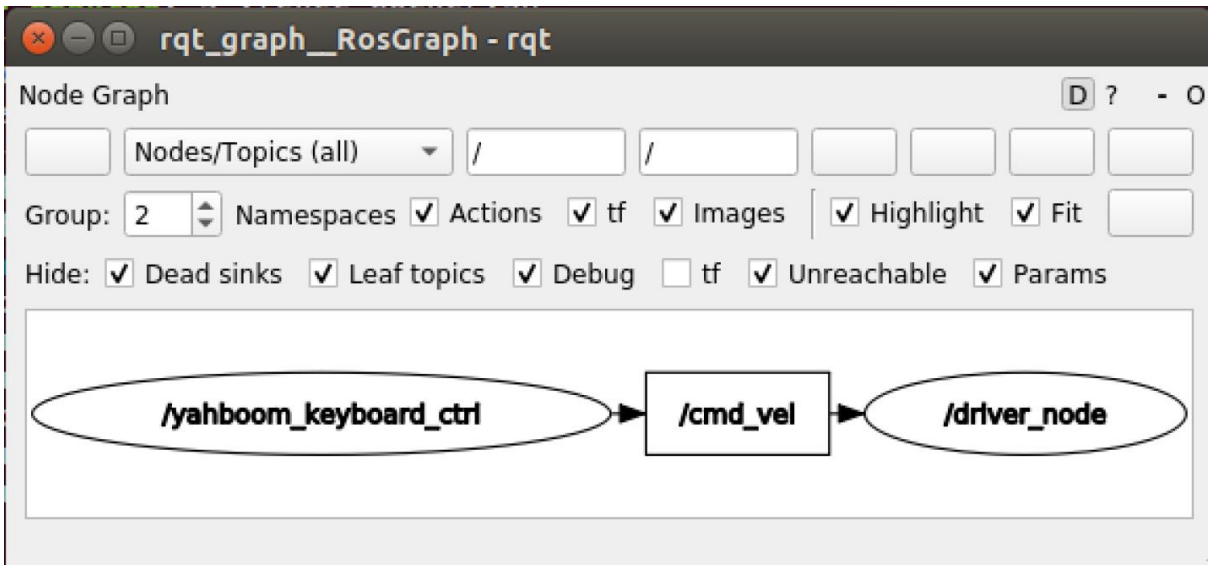
Action8



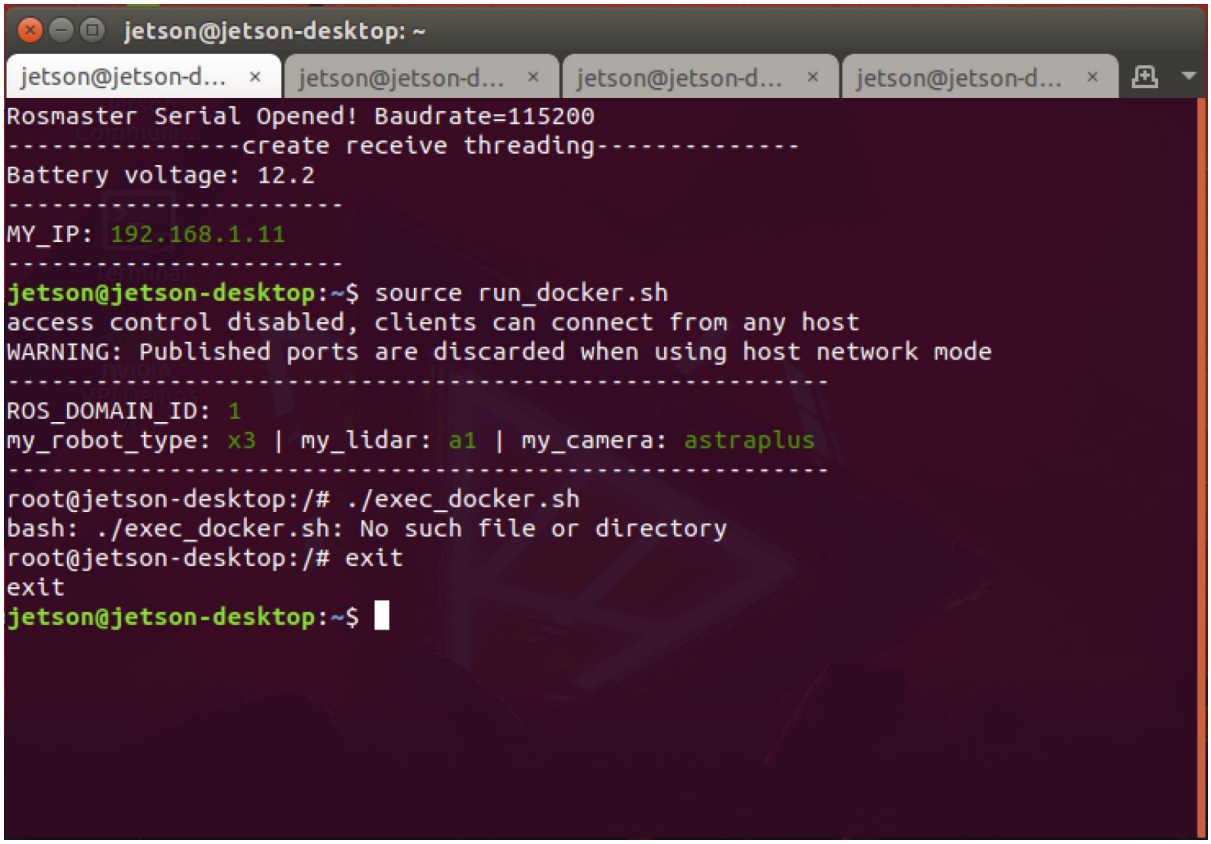
REQUIRED Question: Based on the previous step, what is the coordinate system for the robot?

The robot uses a right-hand coordinate system. In this system, the X-axis represents the forward direction, the Y-axis represents the lateral movement, and the Z-axis points upward. The Z-axis usually indicates the rotation axis.

Action11

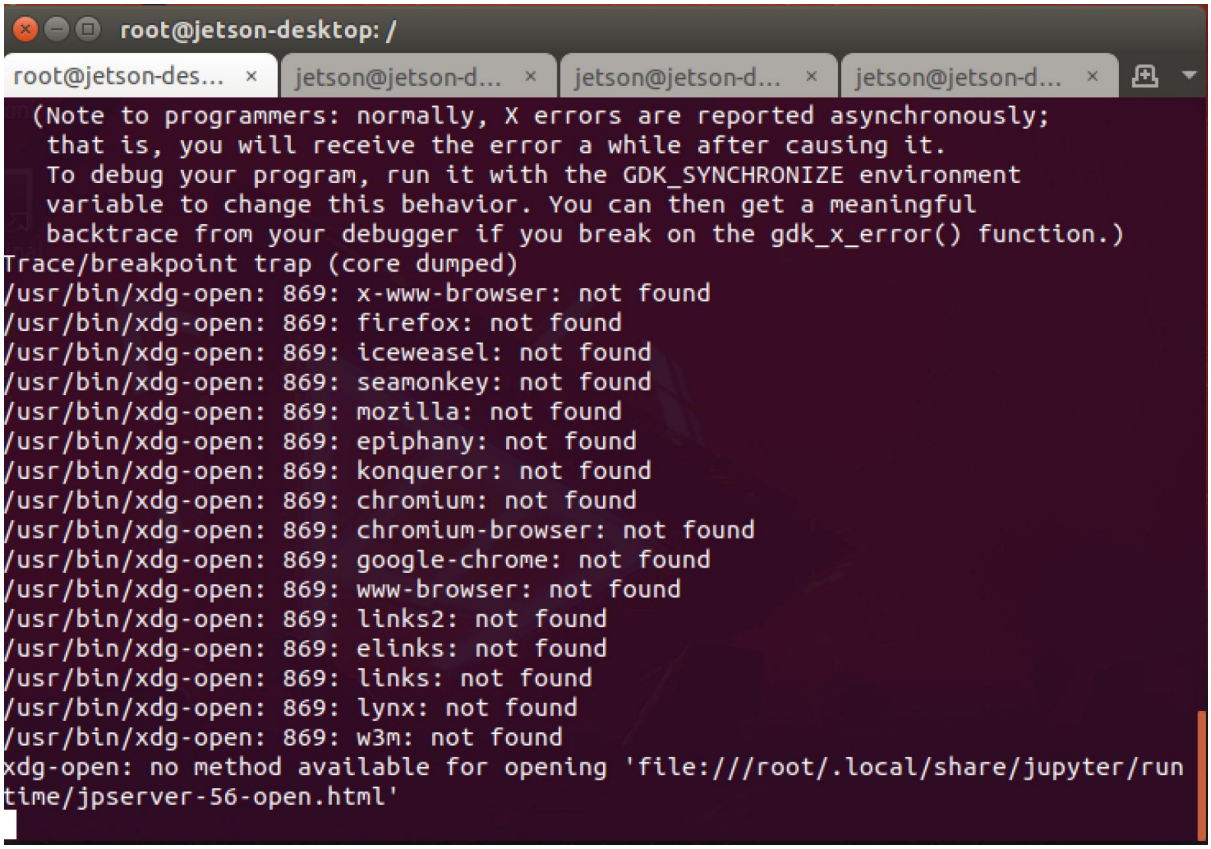


Action12

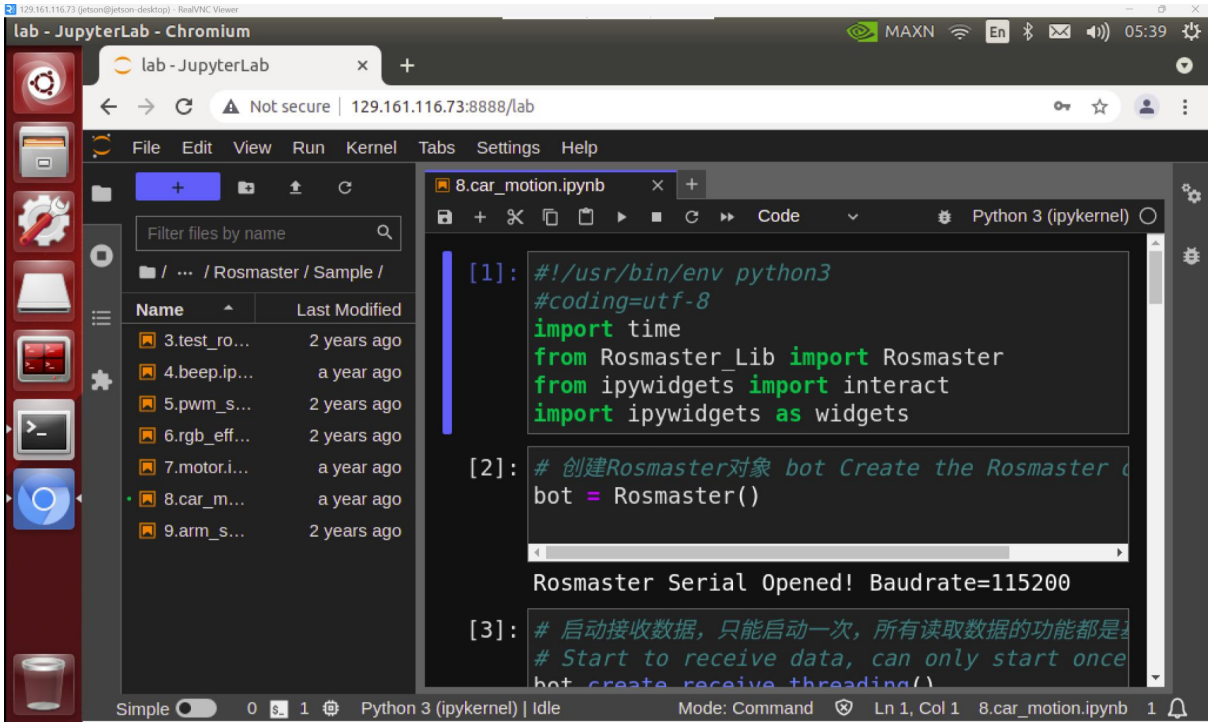


Using Jupyter Lab

Action3

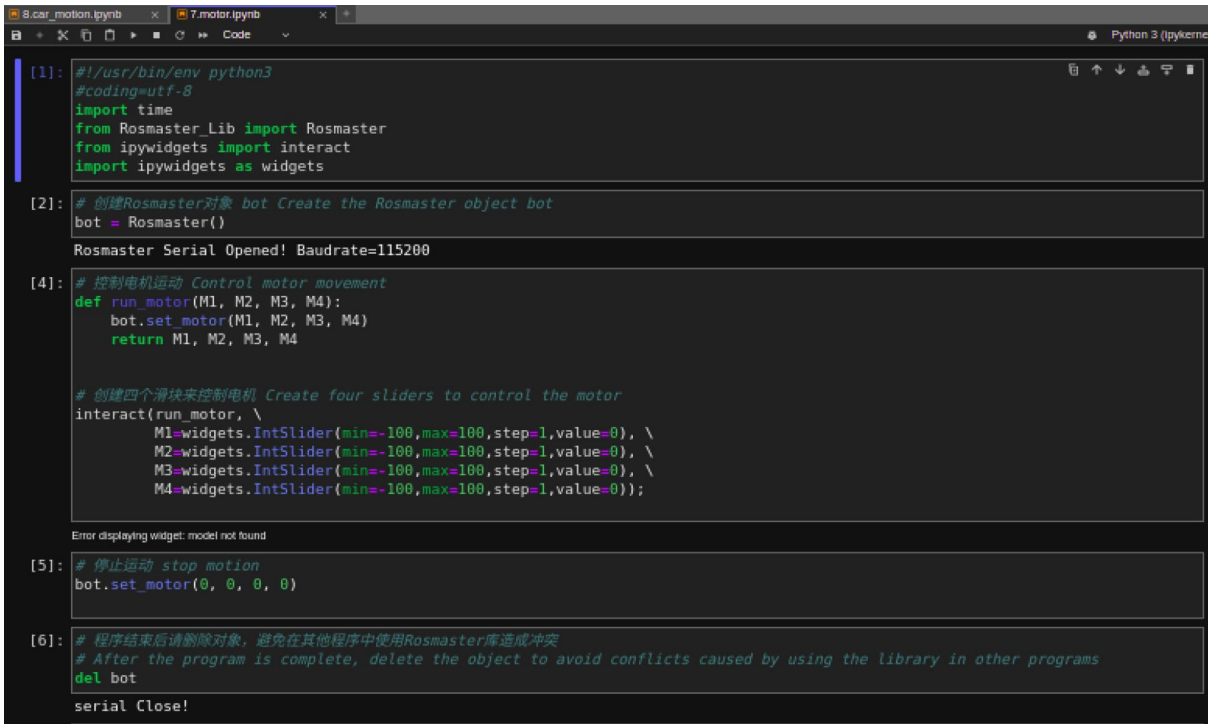


Action4



Action5

motor



car motion

[1]:
```
#!/usr/bin/env python3
#coding=utf-8
import time
from Rosmaster_Lib import Rosmaster
from ipywidgets import interact
import ipywidgets as widgets
```

[2]:
```
# 创建Rosmaster对象 bot Create the Rosmaster object bot
bot = Rosmaster()
```

```
Rosmaster Serial Opened! Baudrate=115200
```

[3]:
```
# 启动接收数据，只能启动一次，所有读取数据的功能都是基于此方法
# Start to receive data, can only start once, all read data function is based on this method
bot.create_receive_threading()
```

自动上报数据开关

Switch of automatic data reporting

[4]:
```
# 开启自动发送数据
# enable=True，底层扩展板会每隔40毫秒发送一次数据。enable=False，则不发送。
# forever=True永久保存，=False临时作用。
# Enable automatic data sending
# If enable=True, the underlying expansion module sends data every 40 milliseconds.  If enable=False, the port is not sent.
# Forever =True for permanent, =False for temporary
enable = True
bot.set_auto_report_state(enable, forever=False)
```

[5]:
```
# 关闭自动发送数据
# enable=True，底层扩展板会每隔40毫秒发送一次数据。enable=False，则不发送。
# forever=True永久保存，=False临时作用。
# Disable automatic data sending
# If enable=True, the underlying expansion module sends data every 40 milliseconds.  If enable=False, the port is not sent.
# Forever =True for permanent, =False for temporary
enable = False
bot.set_auto_report_state(enable, forever=False)
```

[6]:
```
# 清除单片机自动发送过来的缓存数据 Clear the cache data automatically sent by the MCU
bot.clear_auto_report_data()
```

小车控制速度相关参数设置

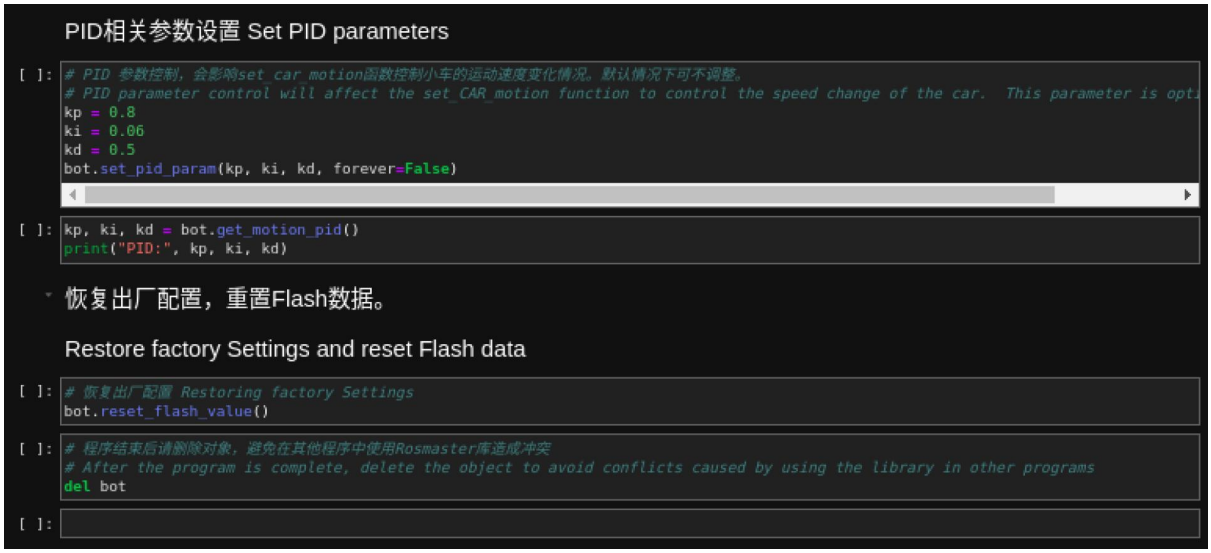
[9]:
```
# 控制电机运动 Control motor movement
def car_motion(V_x, V_y, V_z):
    speed_x= V_x / 10.0
    speed_y = V_y / 10.0
    speed_z = V_z / 10.0
    bot.set_car_motion(speed_x, speed_y, speed_z)
    return speed_x, speed_y, speed_z


# 创建三个滑块来控制小车的速度 Create three sliders to control the speed of the car
interact(car_motion, \
         V_x=widgets.IntSlider(min=-10,max=10,step=1,value=0), \
         V_y=widgets.IntSlider(min=-10,max=10,step=1,value=0), \
         V_z=widgets.IntSlider(min=-50,max=50,step=1,value=0));
```

```
Error displaying widget: model not found
```

[ ]:
```
# 停止运动 stop motion
bot.set_car_motion(0, 0, 0)
```

[ ]:
```
# 获取小车线速度和角速度数据
# Obtain the linear velocity and angular velocity data of the car
try:
    while True:
        V_x, V_y, V_z = bot.get_motion_data()
        print("speed:", V_x, V_y, V_z)
        bot.clear_auto_report_data()
        time.sleep(.1)
except KeyboardInterrupt:
    pass
```



REQUIRED Question: Based on these scripts, what is the minimal code needed to control the robot motors? What are the inputs for the key functions? What is the key difference between the scripts?

Among these scripts, the minimal code to control the robot motors includes three parts: importing the library from Rosmaster_Lib import Rosmaster, creating the object bot = Rosmaster(), and using bot.set_motor(M1, M2, M3, M4) to set the speeds of the four motors. The inputs for set_motor() are the speed values for each motor, typically in the range of -100 to 100. Another key function is set_car_motion(Vx, Vy, Vz), which takes linear velocities in the X and Y directions and angular velocity around the Z-axis to control the robot's movement. The main difference between the scripts lies in the control method: some use direct motor speed control, some use overall motion control, and others handle serial communication, automatic data reporting, and PID parameter configuration.

Task2

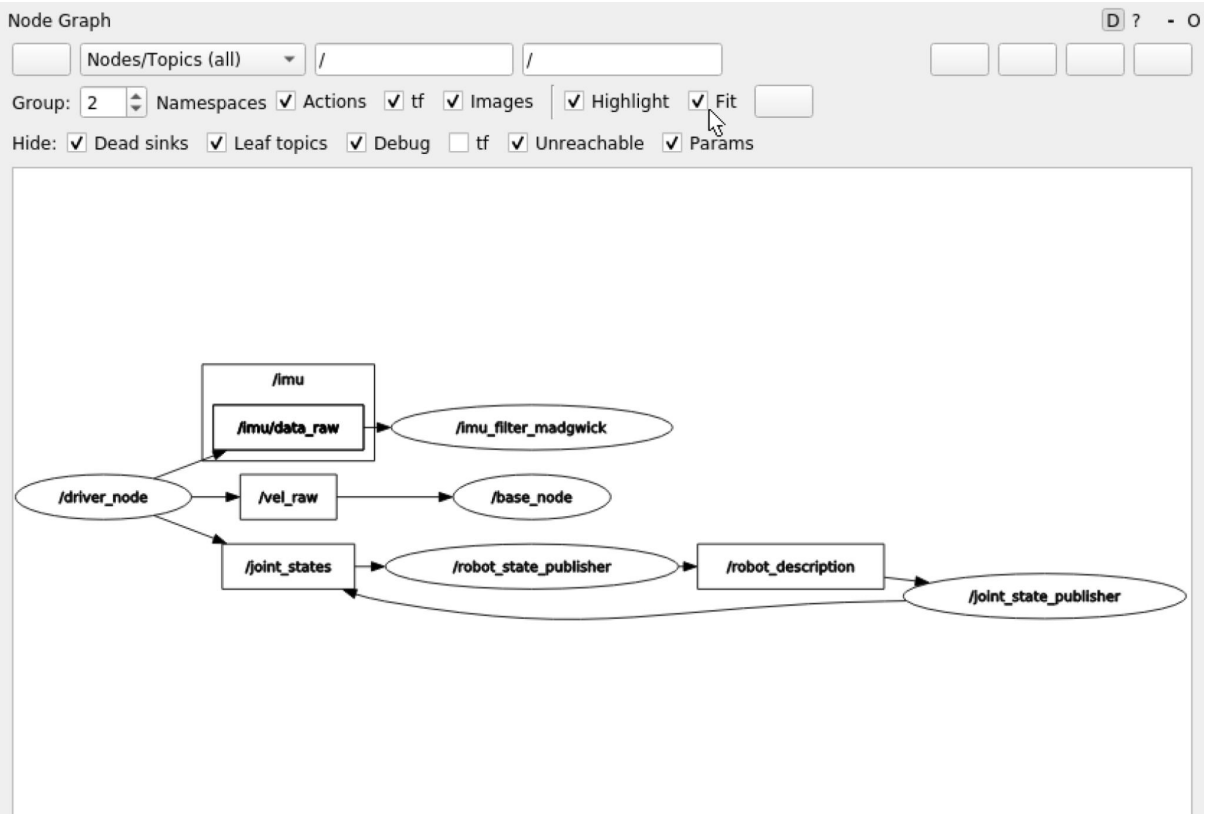
Action1

```
[INFO] [joint_state_publisher-1]: process started with pid [184]
[INFO] [robot_state_publisher-2]: process started with pid [186]
[INFO] [Mcnamu_driver_X3-3]: process started with pid [188]
[INFO] [base_node_X3-4]: process started with pid [190]
[INFO] [imu_filter_madgwick_node-5]: process started with pid [192]
[INFO] [ekf_node-6]: process started with pid [195]
[INFO] [yahboom_joy_X3-7]: process started with pid [197]
[robot_state_publisher-2] Parsing robot urdf xml string.
[robot_state_publisher-2] Link base_link had 7 children
[robot_state_publisher-2] Link back_left_wheel had 0 children
[robot_state_publisher-2] Link back_right_wheel had 0 children
[robot_state_publisher-2] Link imu_link had 0 children
[robot_state_publisher-2] Link camera_link had 0 children
[robot_state_publisher-2] Link front_left_wheel had 0 children
[robot_state_publisher-2] Link front_right_wheel had 0 children
[robot_state_publisher-2] Link laser_link had 0 children
[robot_state_publisher-2] [INFO] [1742330116.638329606] [robot_state_publisher]: got segment back_left_wh
eel
[robot_state_publisher-2] [INFO] [1742330116.638652733] [robot_state_publisher]: got segment back_right_w
heel
[robot_state_publisher-2] [INFO] [1742330116.638709400] [robot_state_publisher]: got segment base_footpri
nt
[robot_state_publisher-2] [INFO] [1742330116.638745494] [robot_state_publisher]: got segment base_link
[robot_state_publisher-2] [INFO] [1742330116.638776016] [robot_state_publisher]: got segment camera_link
[robot_state_publisher-2] [INFO] [1742330116.638808464] [robot_state_publisher]: got segment front_left_w
heel
[robot_state_publisher-2] [INFO] [1742330116.638839349] [robot_state_publisher]: got segment front_right_
wheel
[robot_state_publisher-2] [INFO] [1742330116.638868829] [robot_state_publisher]: got segment imu_link
[robot_state_publisher-2] [INFO] [1742330116.638897683] [robot_state_publisher]:
 got segment laser_link
[imu_filter_madgwick_node-5] [INFO] [1742330116.839984364] [imu_filter_madgwick]
: Starting ImuFilter
[imu_filter_madgwick_node-5] [INFO] [1742330116.848947505] [imu_filter_madgwick]
: Using dt computed from message headers
[imu_filter_madgwick_node-5] [INFO] [1742330116.849064485] [imu_filter_madgwick]
: The gravity vector is kept in the IMU message.
[imu_filter_madgwick_node-5] [INFO] [1742330116.892042780] [imu_filter_madgwick]
: Imu filter gain set to 0.100000
[imu_filter_madgwick_node-5] [INFO] [1742330116.932280012] [imu_filter_madgwick]
: Gyro drift bias set to 0.000000
[imu_filter_madgwick_node-5] [INFO] [1742330116.941769667] [imu_filter_madgwick]
: Magnetometer bias values: 0.000000 0.000000 0.000000
[joint_state_publisher-1] [INFO] [1742330118.826324684] [joint_state_publisher]:
 Waiting for robot_description to be published on the robot_description topic...
[imu_filter_madgwick_node-5] [INFO] [1742330119.111409501] [imu_filter_madgwick]
: First IMU message received.
```

REQUIRED Question: Based on the printout to the terminal, what are some sensors the robot has?

From the terminal output, the robot has an IMU sensor and a laser sensor. The IMU is activated through the imu_filter_madgwick node and receives data from the gyroscope, accelerometer, and magnetometer. The laser sensor is identified by the laser_link segment, indicating the presence of a laser module.

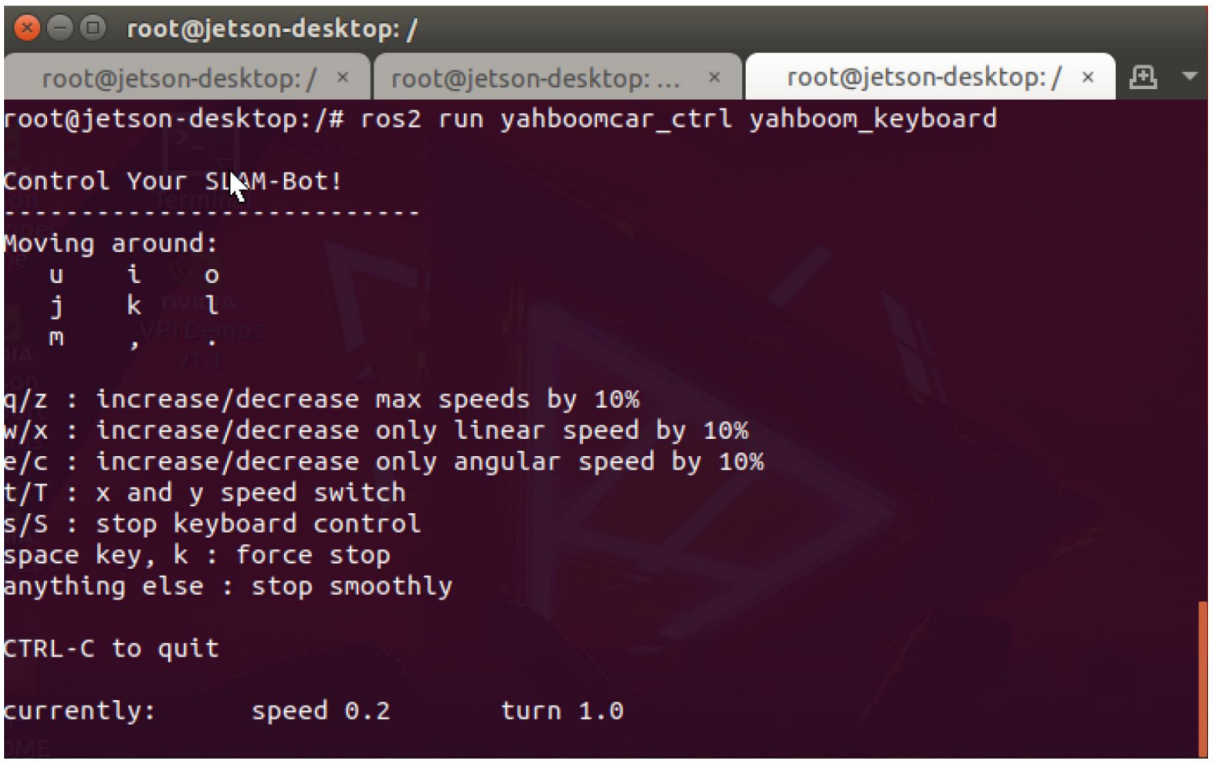
Action3



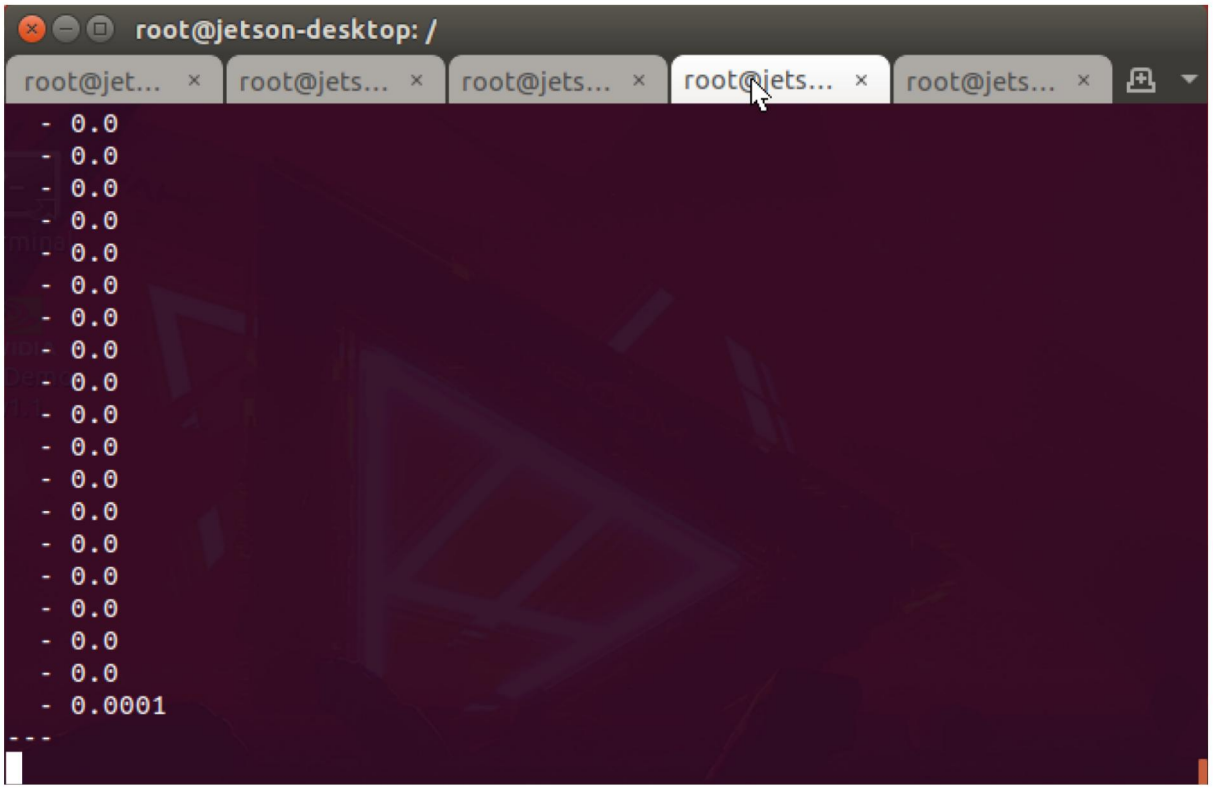
Action4

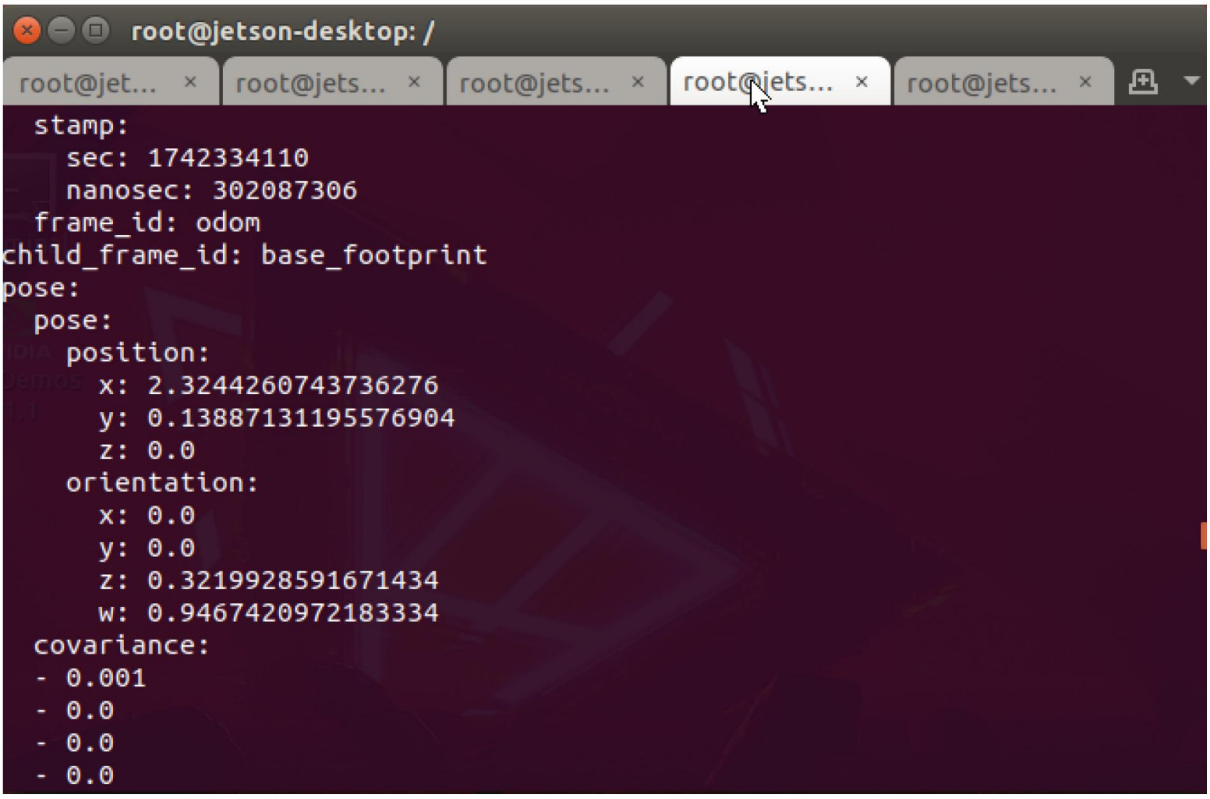
```
root@jetson-desktop:/# cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboom
car_bringup/launch/
root@jetson-desktop:~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_bringu
p/launch# ls -l
total 12
-rw-r--r-- 1 root root 4039 May 25  2023 yahboomcar_bringup_R2_launch.py
-rw-r--r-- 1 root root 4070 Apr 21  2023 yahboomcar_bringup_X1_launch.py
-rw-r--r-- 1 root root 4033 Jul 24  2023 yahboomcar_bringup_X3_launch.py
```

Action5

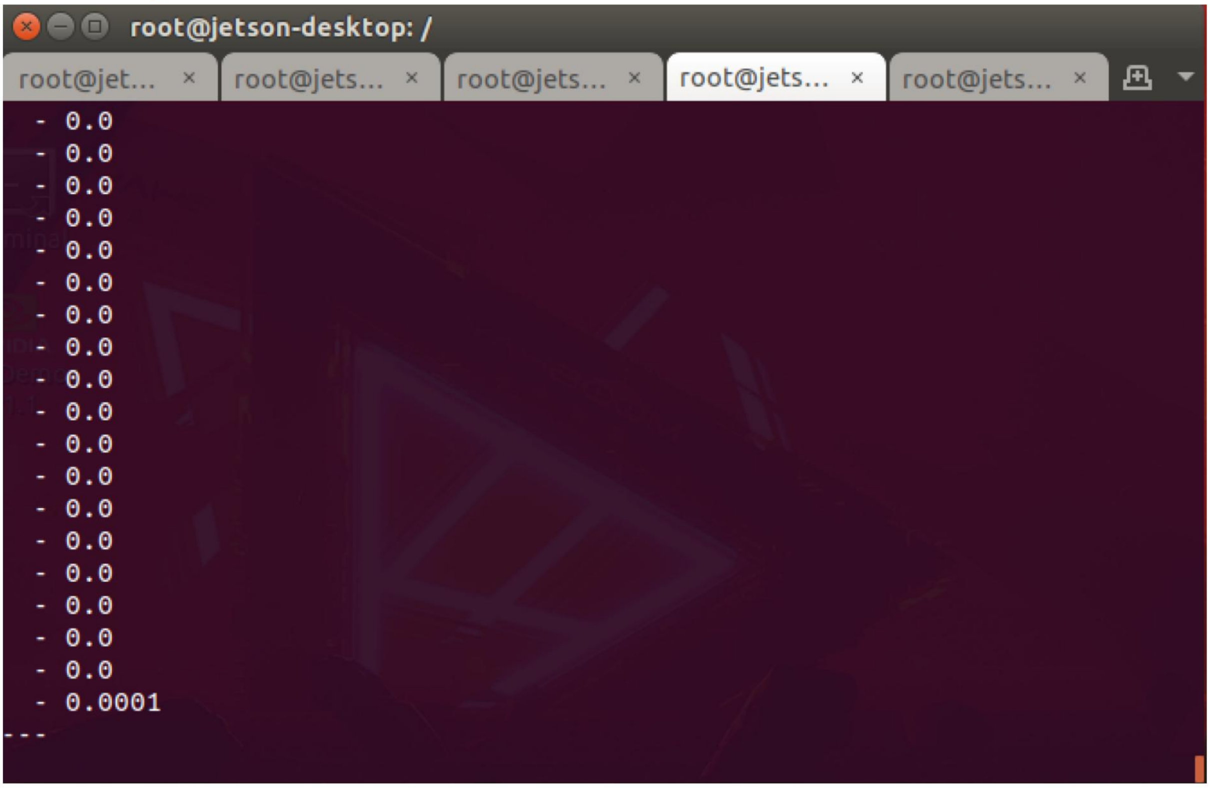


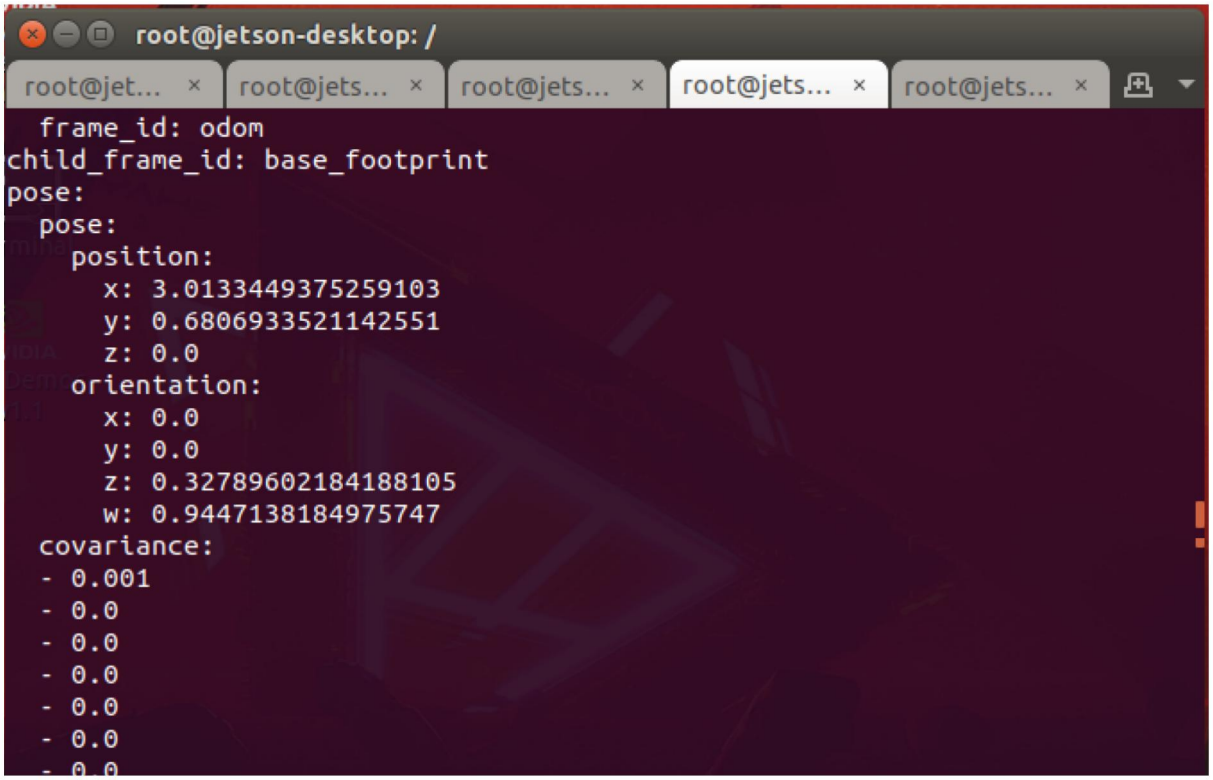
Action6



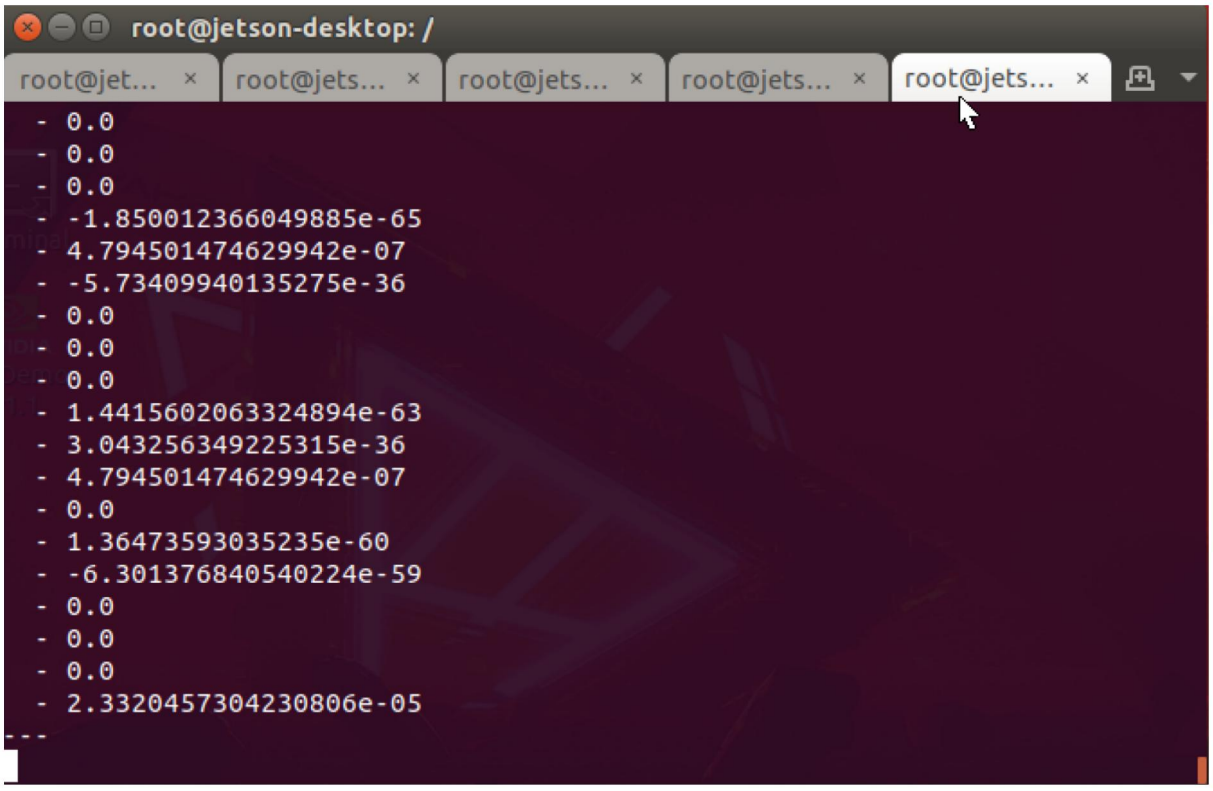


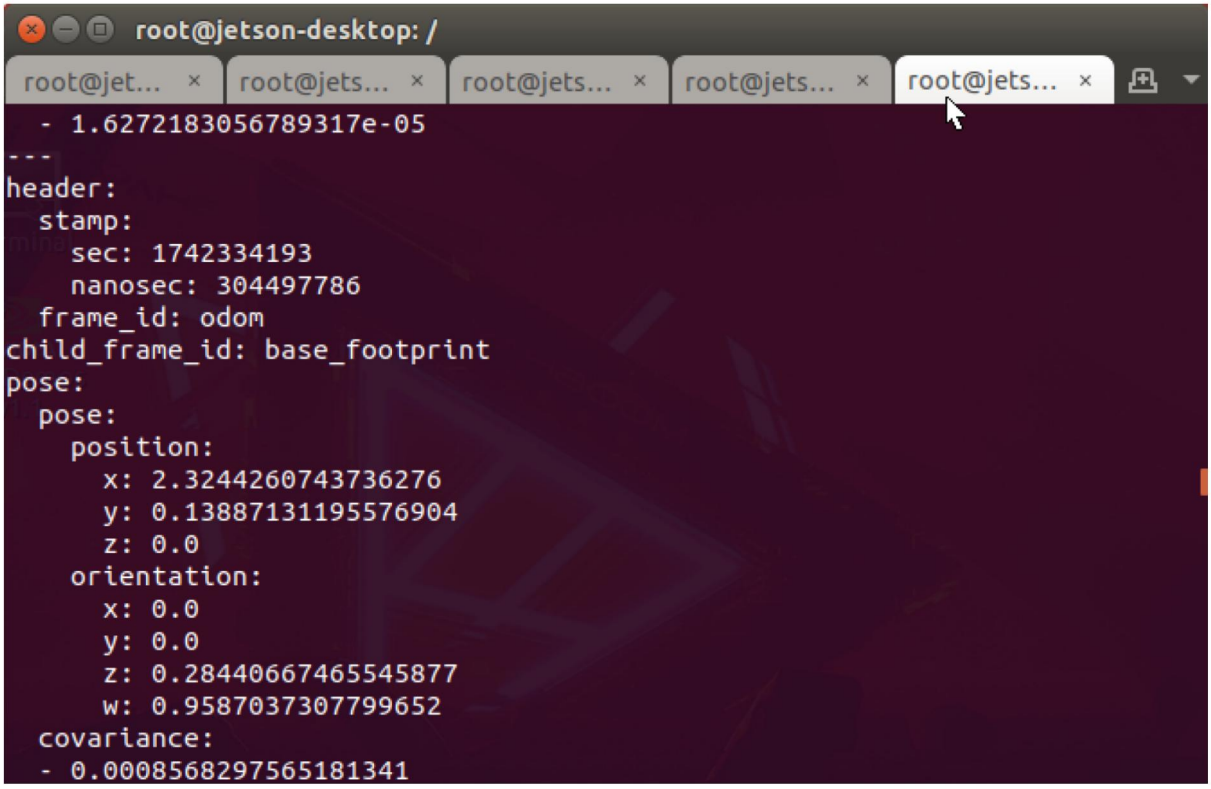
after moving



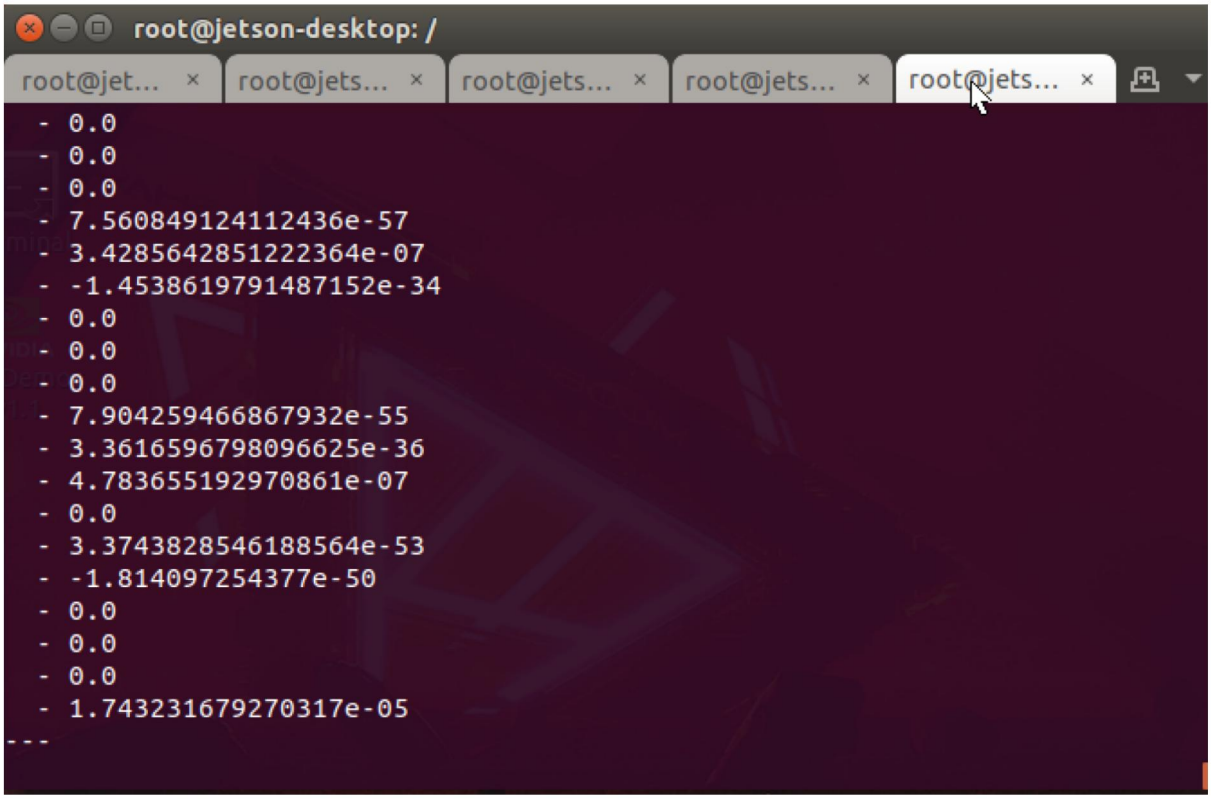

root@jetson-desktop: /
root@jet...
root@jets...
root@jets...
root@jets...
root@jets...
frame_id: odom
child_frame_id: base_footprint
pose:
pose:
position:
x: 3.0133449375259103
y: 0.6806933521142551
z: 0.0
orientation:
x: 0.0
y: 0.0
z: 0.32789602184188105
w: 0.9447138184975747
covariance:
- 0.001
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0

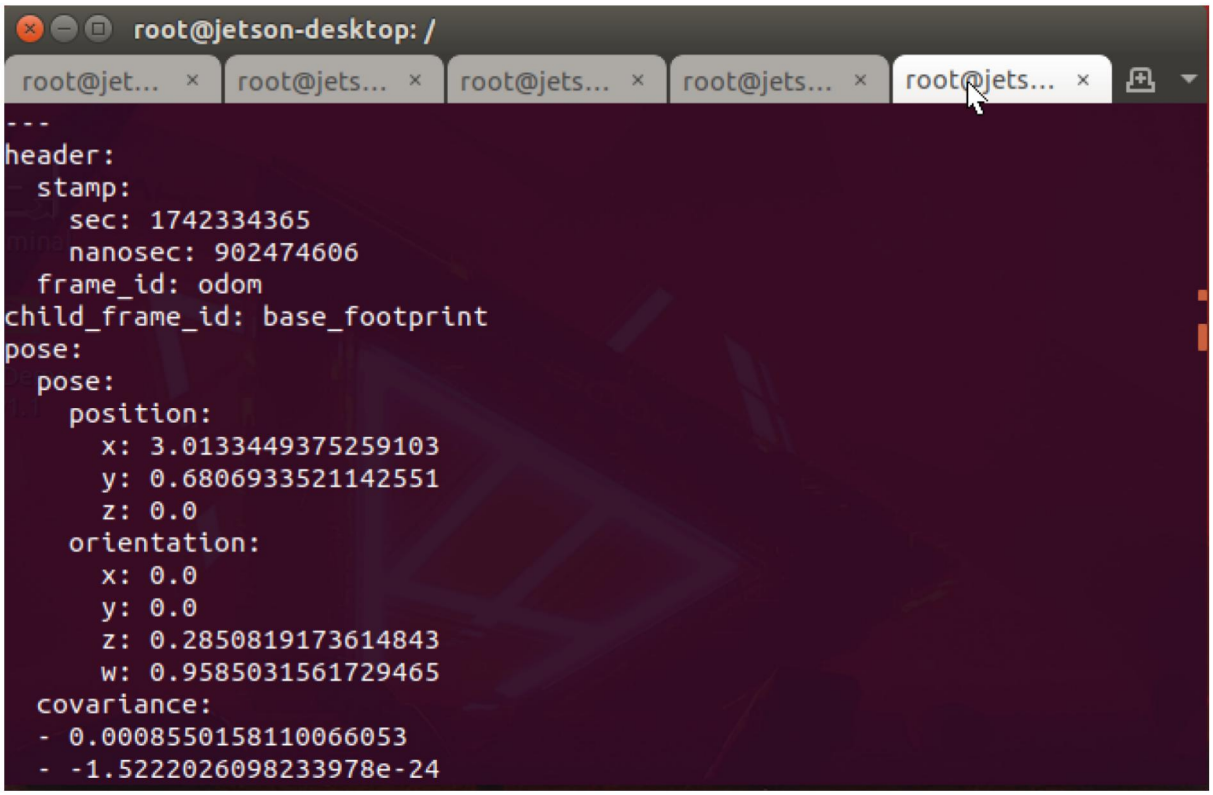
no raw


root@jetson-desktop: /
root@jet...
root@jets...
root@jets...
root@jets...
root@jets...
- 0.0
- 0.0
- 0.0
- -1.850012366049885e-65
- 4.794501474629942e-07
- -5.73409940135275e-36
- 0.0
- 0.0
- 0.0
- 1.44156020633248 94e-63
- 3.043256349225315e-36
- 4.794501474629942e-07
- 0.0
- 1.36473593035235e-60
- -6.301376840540224e-59
- 0.0
- 0.0
- 0.0
- 2.3320457304230806e-05



after moving



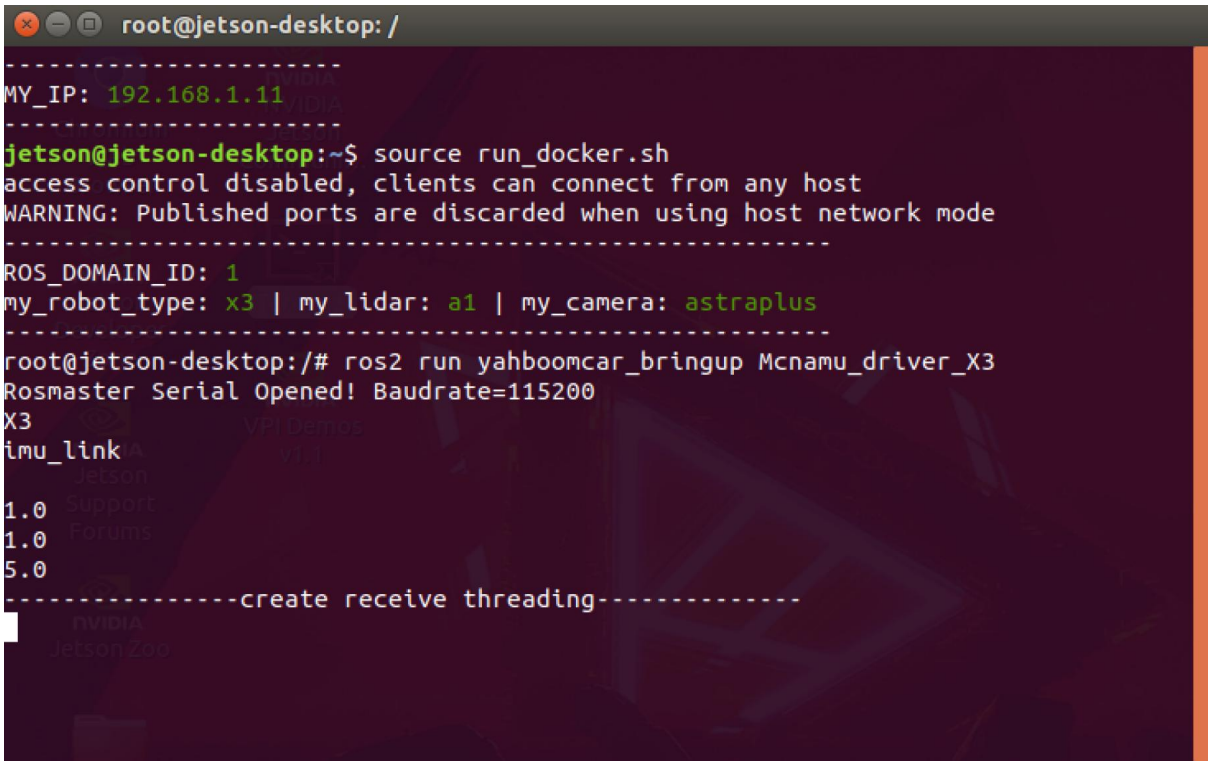


REQUIRED Question: Based on the previous steps and any other ROS tools that you use, what is this example doing? What are some key topics and message types used in the example?

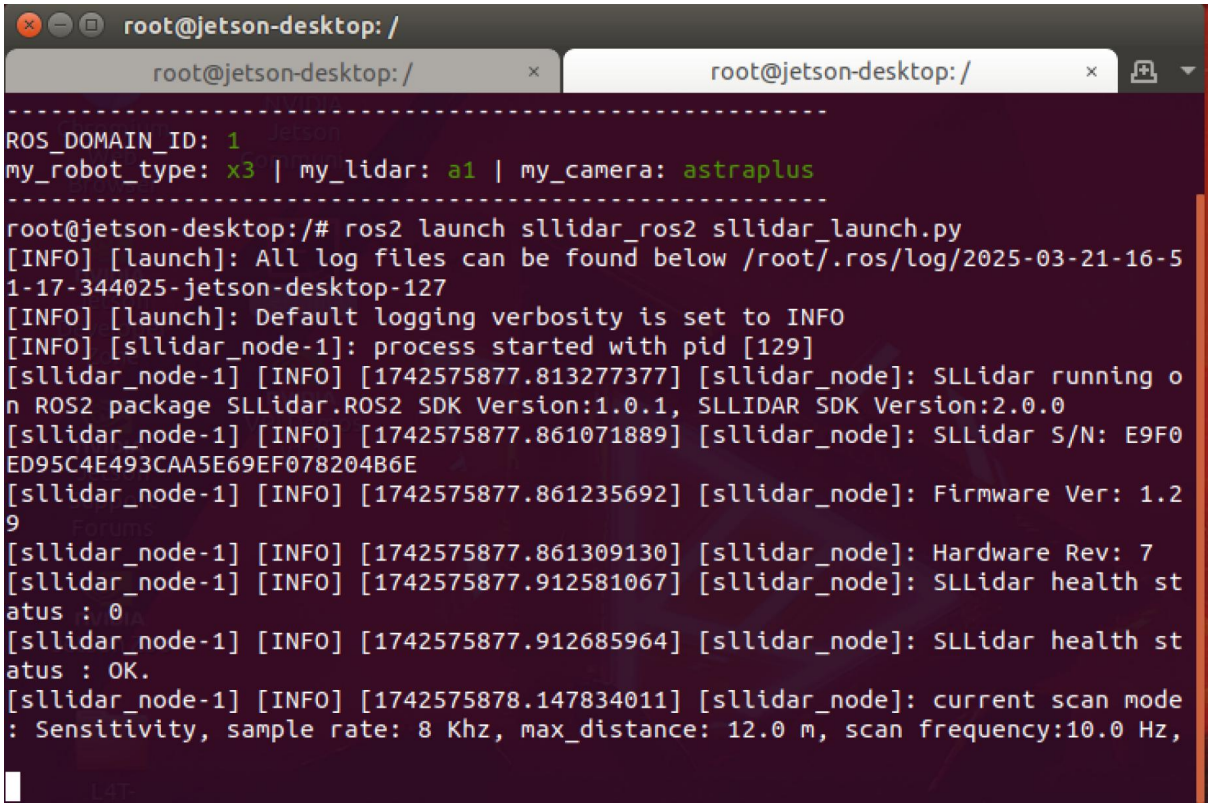
This example shows how multiple sensors and user input are used to produce various frames or states for the robot. The topics odom_raw and odom are used here, each providing a frame based on sensor readings and user commands. Other launch files may define different transform functions, which can result in different frames or states even when given the same user input.

Task3

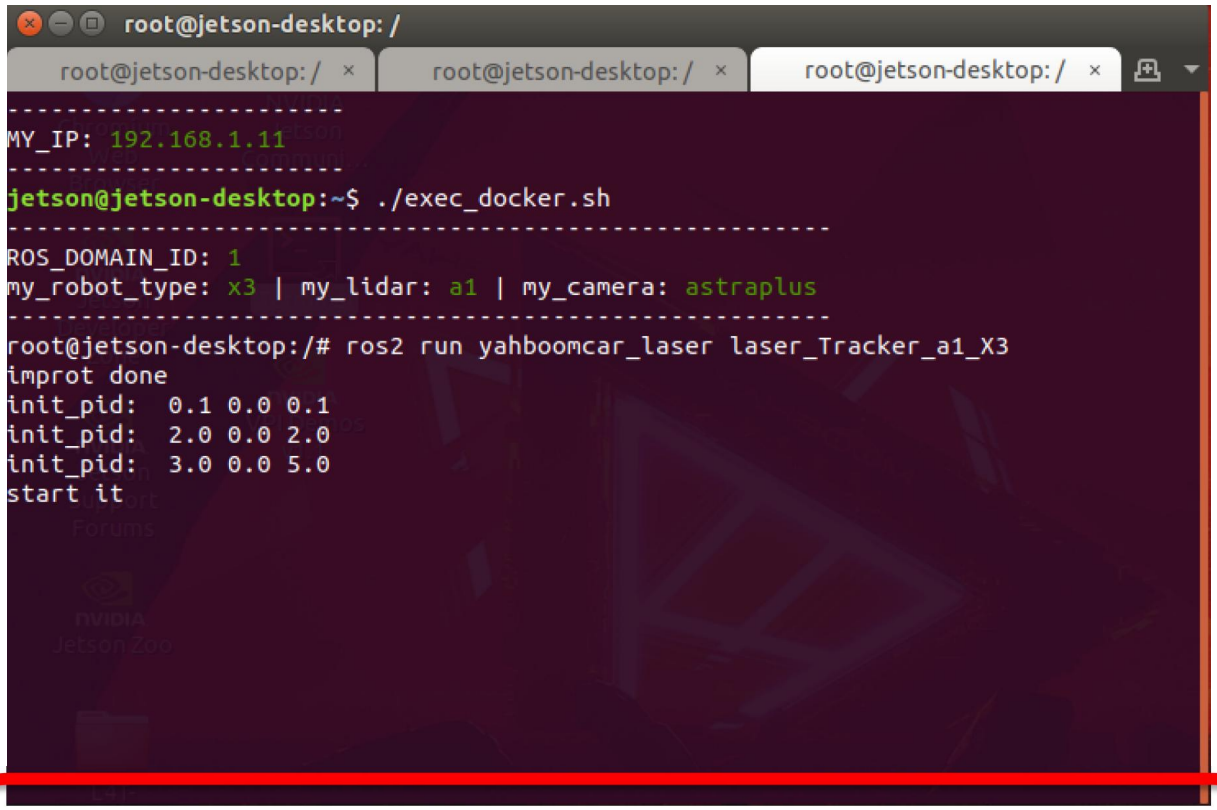
Action2



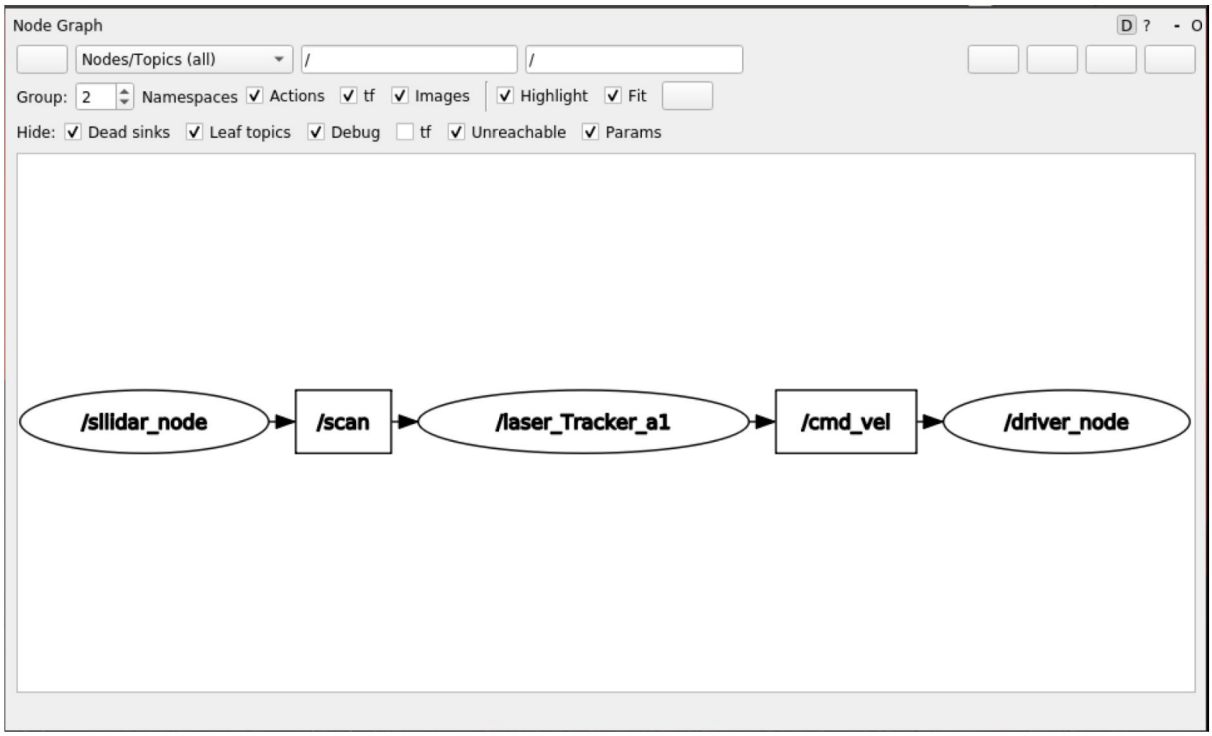
Action3



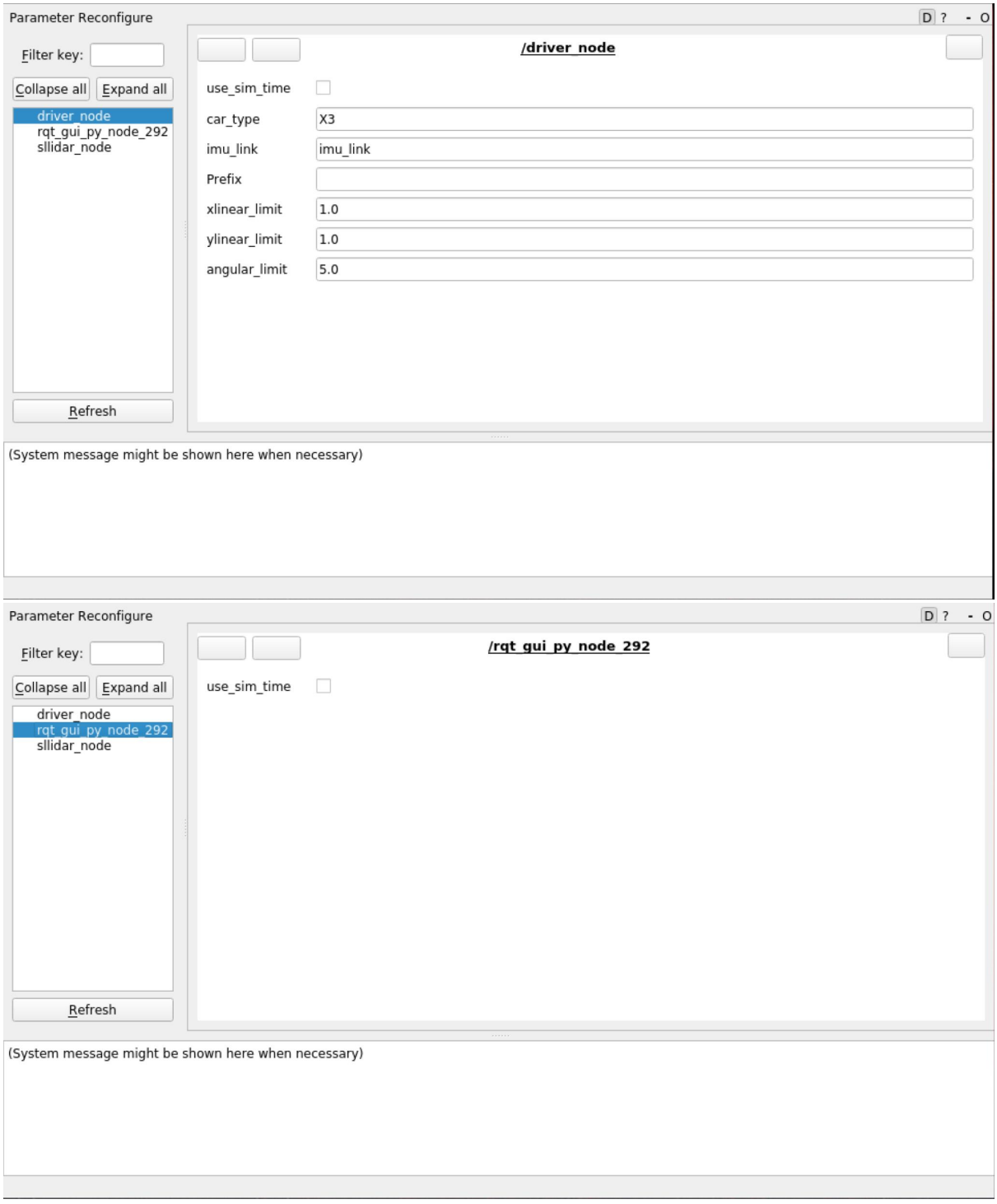
Action4

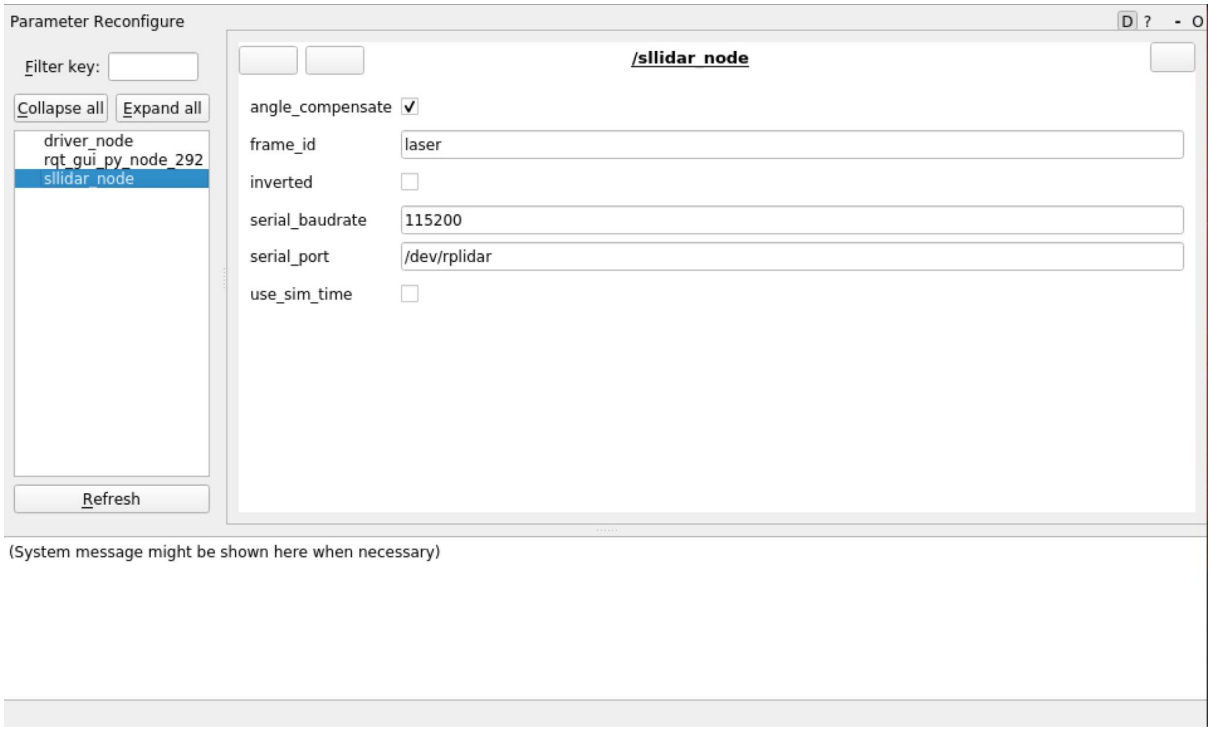


Action5



Action7


Parameter Reconfigure
Filter key:
Collapse all
Expand all
driver_node
rqt_gui_py_node_292
sllidar_node
Refresh
/driver_node
use_sim_time
car_type
X3
imu_link
imu_link
Prefix
xlinear_limit
1.0
ylinear_limit
1.0
angular_limit
5.0
(System message might be shown here when necessary)
Parameter Reconfigure
Filter key:
Collapse all
Expand all
driver_node
rqt_gui_py_node_292
sllidar_node
Refresh
/rqt_gui_py_node_292
use_sim_time
(System message might be shown here when necessary)



after changing parameters