

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Khoa: Điện tử viễn thông



**BÁO CÁO GIỮA KỲ
LẬP TRÌNH ROBOT VỚI ROS**

Họ và tên:
Trần Hoàng Thắng - 22023160
Hà Nội – 2025

1. Giới thiệu

Dự án sử dụng ROS2 Humble thiết kế mô phỏng mobile robot 2 bánh vi sai. Robot tích hợp các cảm biến như camera, imu, lidar và hệ thống tay máy 2 bậc tự do và điều khiển qua bàn phím.

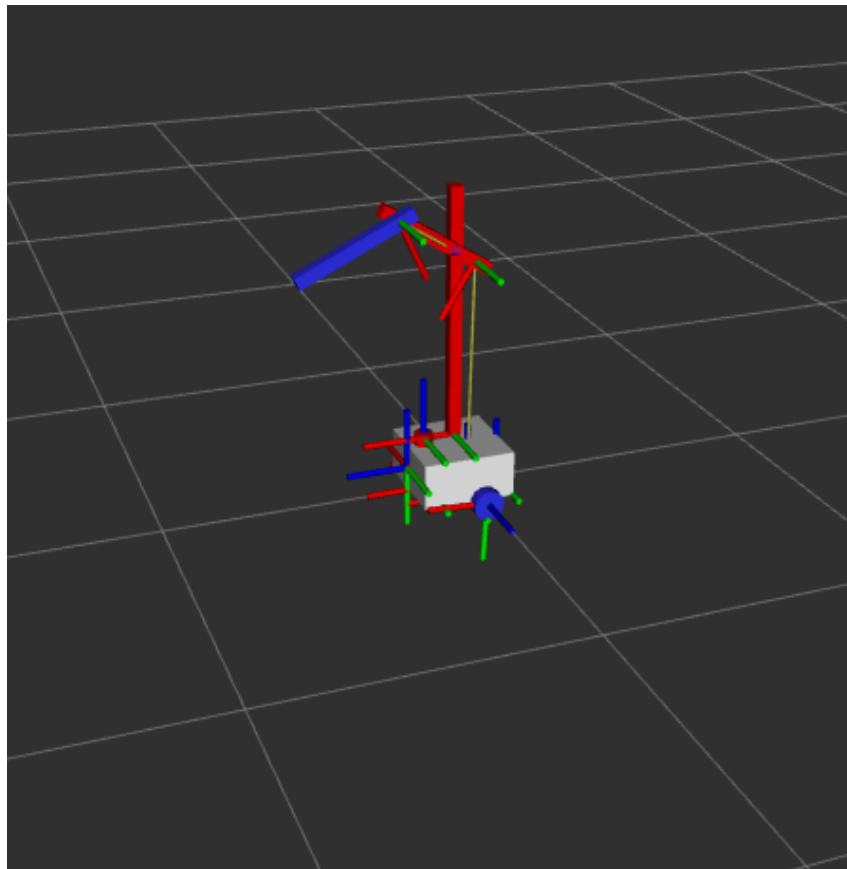
2. Mô tả

Các bộ phận trên robot:

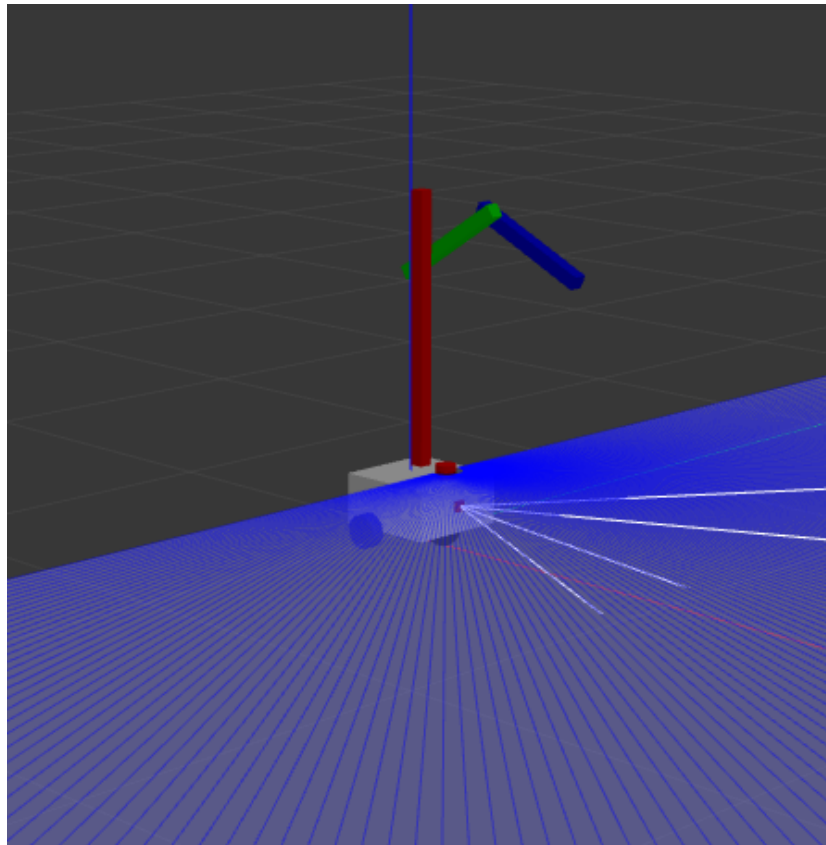
- Chassis: là phần thân robot, nơi chứa các mạch điều khiển, máy tính nhúng, bộ nguồn,...
- Hai bánh xe gắn vào động cơ ở chassis.
- Cảm biến imu gắn vào chassis. Thực tế, imu thường được đặt cách xa các động cơ, dây điện, pin,... hay được bao bọc bởi nhựa, nhôm hoặc các vật liệu không nhiễm từ để cách ly từ trường do cấu tạo của imu có từ kế hoạt động dựa trên từ trường trái đất. Nếu không cách ly từ trường thì cảm biến dễ bị nhiễu dẫn đến sai số. Imu có tác dụng lấy odometry cho robot.
- Lidar 180 độ được đặt trước tay máy để thực hiện các tác vụ như slam hay localization.
- Camera gắn ở trước xe.
- Cánh tay 2 bậc tự do.

Hình ảnh minh họa mô hình

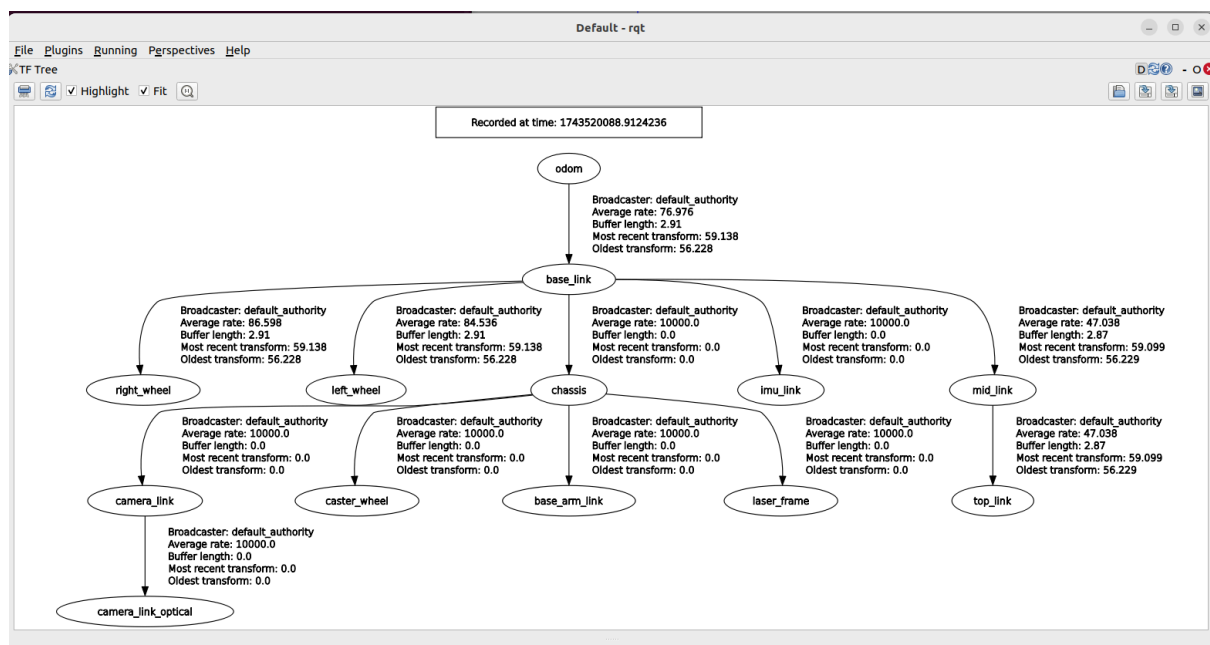
(ở đây em vẽ trực tiếp vào urdf file nên robot trông đơn giản)



Robot model được dựng trong Rviz



Robot model được đặt vào môi trường Gazebo



Đây là tf tree của robot

```
tth@tth: ~/gk_test ws 62x29
tth@tth:~/gk_test_ws$ ros2 topic list
/camera/camera_info
/camera/image_raw
/clicked_point
/clock
/cmd_vel
/dynamic_joint_states
/goal_pose
/imu_plugin/out
/initialpose
/joint_base_mid_position_controller/controller_state
/joint_base_mid_position_controller/joint_trajectory
/joint_base_mid_position_controller/state
/joint_base_mid_position_controller/transition_event
/joint_mid_top_position_controller/controller_state
/joint_mid_top_position_controller/joint_trajectory
/joint_mid_top_position_controller/state
/joint_mid_top_position_controller/transition_event
/joint_state_broadcaster/transition_event
/joint_states
/odom
/parameter_events
/performance_metrics
/robot_description
/rosout
/scan
/tf
/tf_static
tth@tth:~/gk_test_ws$
```

Các topic của robot

Đây là các topic khi chạy mô phỏng robot trong môi trường gazebo:

- Các topic /camera/* là các topic về camera đã gắn vào
- /imu_plugin/out là topic về imu đã gắn vào
- /scan là topic của về lidar
- /join* là các topic về tay máy

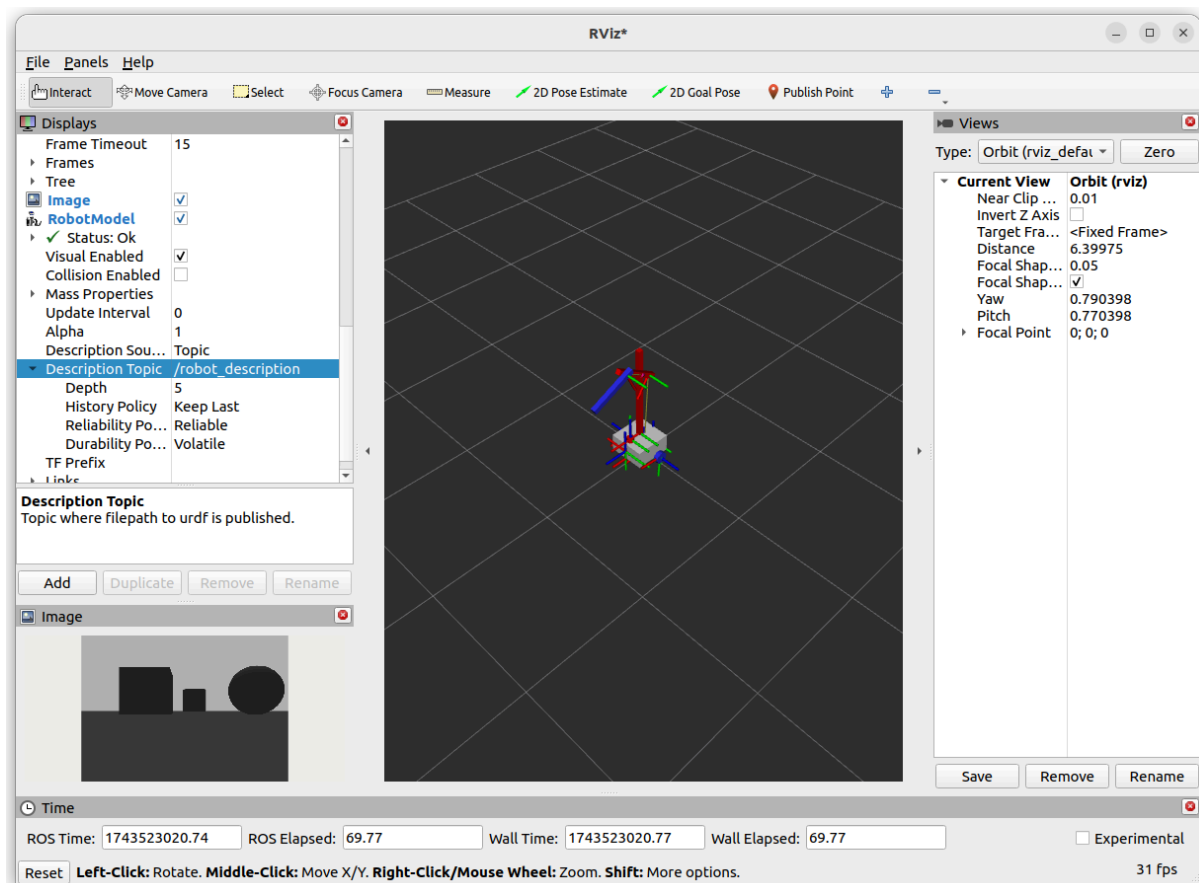
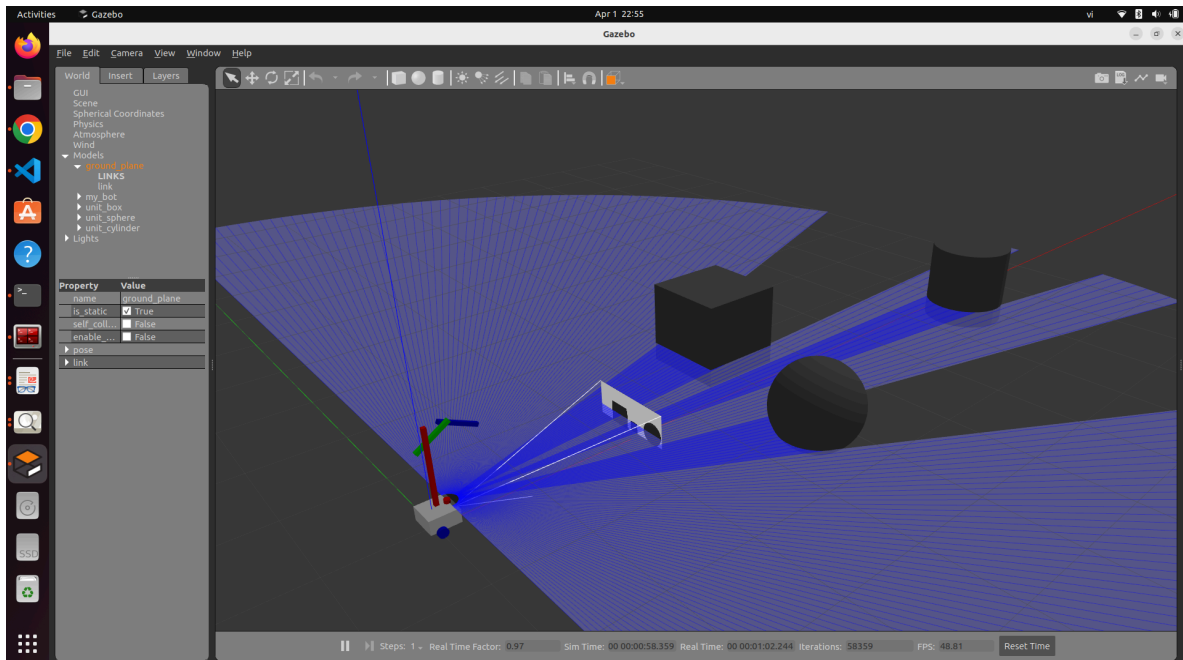
Các .xacro file các thành phần của robot:

```
▼ description
  🔊 arm_control.xacro
  🔊 arm.gazebo.xacro
  🔊 arm.xacro
  🔊 camera.xacro
  🔊 gazebo_control.xacro
  🔊 gk_core.xacro
  🔊 gk.urdf.xacro
  🔊 imu.xacro
  🔊 inertial_macros.xacro
  🔊 lidar.xacro
  🔊 materials.xacro
```

- gk.urdf.xacro: đây là file tổng include các .xacro file con
- gk_core.xacro: đây là file chứa base_link và các link gắn với base_link như chassis, left_wheel, right_wheel

- arm.xacro: đây là urdf cho tay máy
- arm_control.xacro và arm.gazebo.xacro là các file gắn plugin và màu arm
- inertial_macros.xacro là file cấu hình khối lượng cho phần xe (cấu hình khối lượng riêng cho xe và tay máy để dễ hiệu chỉnh khi gắn cả tay máy vào xe mà xe bị lật do quán tính hoặc trọng lực)
- Các .xacro file còn lại là các cảm biến và cấu hình plugin cho chúng

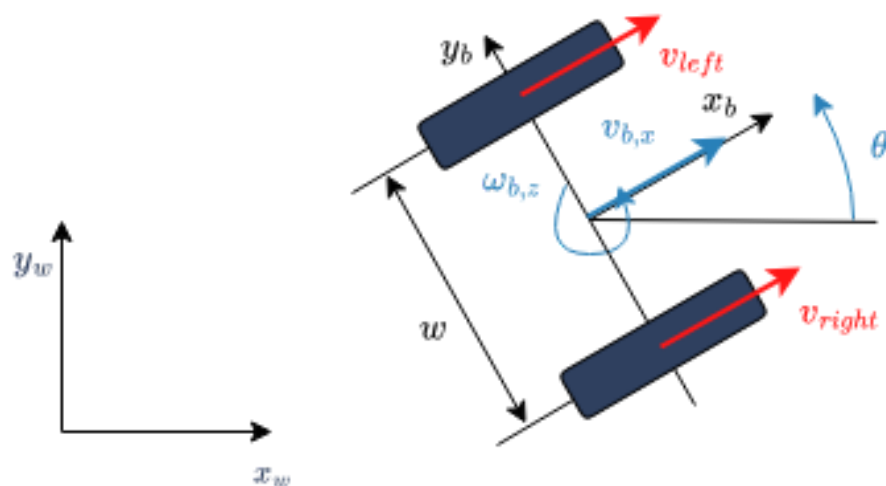
3. Cảm biến hoạt động



(camera nhận thấy các chương ngại vật ở môi trường Gazebo)

```
ttht@ttht: ~  
- 0.0  
- 0.0  
angular_velocity:  
  x: 0.0005244334702240404  
  y: -8.247334024543989e-05  
  z: 3.1767042207691894e-06  
angular_velocity_covariance:  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
linear_acceleration:  
  x: -0.00042777920266362296  
  y: -0.016703761660696557  
  z: 9.79947295956713  
linear_acceleration_covariance:  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
- 0.0  
---  
ttht@ttht:~$
```

4. Động học và điều khiển robot 2 bánh vi sai



Động học chuyển động của robot 2 bánh vi sai có công thức tính như sau:

$$v_{b,x} = \frac{v_{right} + v_{left}}{2}$$

$$\omega_{b,z} = \frac{v_{right} - v_{left}}{w}$$

(với $v_{b,x}$ là vận tốc dài và $\omega_{b,z}$ là vận tốc xoay)

$$v_{left} = v_{b,x} - \omega_{b,z}w/2$$

$$v_{right} = v_{b,x} + \omega_{b,z}w/2$$

(với v_{left} là vận tốc bánh xe bên trái và v_{right} là vận tốc bánh xe bên phải)

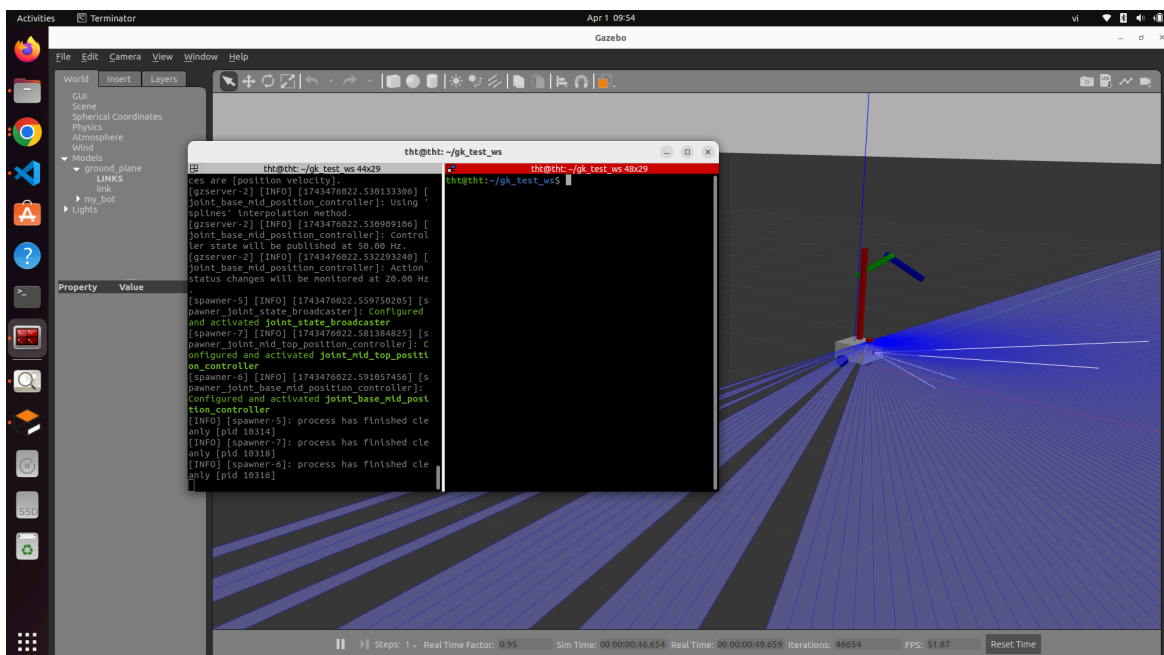
Ta có phương trình quy đổi từ vận tốc dài và vận tốc góc sub từ topic /cmd_vel sang vận tốc 2 bánh xe là:

```
def compute_wheel_velocities(self, linear_vel, angular_vel):
    left_wheel_vel = (linear_vel - (WHEEL_BASE / 2.0) * angular_vel) / WHEEL_RADIUS
    right_wheel_vel = (linear_vel + (WHEEL_BASE / 2.0) * angular_vel) / WHEEL_RADIUS
    return left_wheel_vel, right_wheel_vel
```

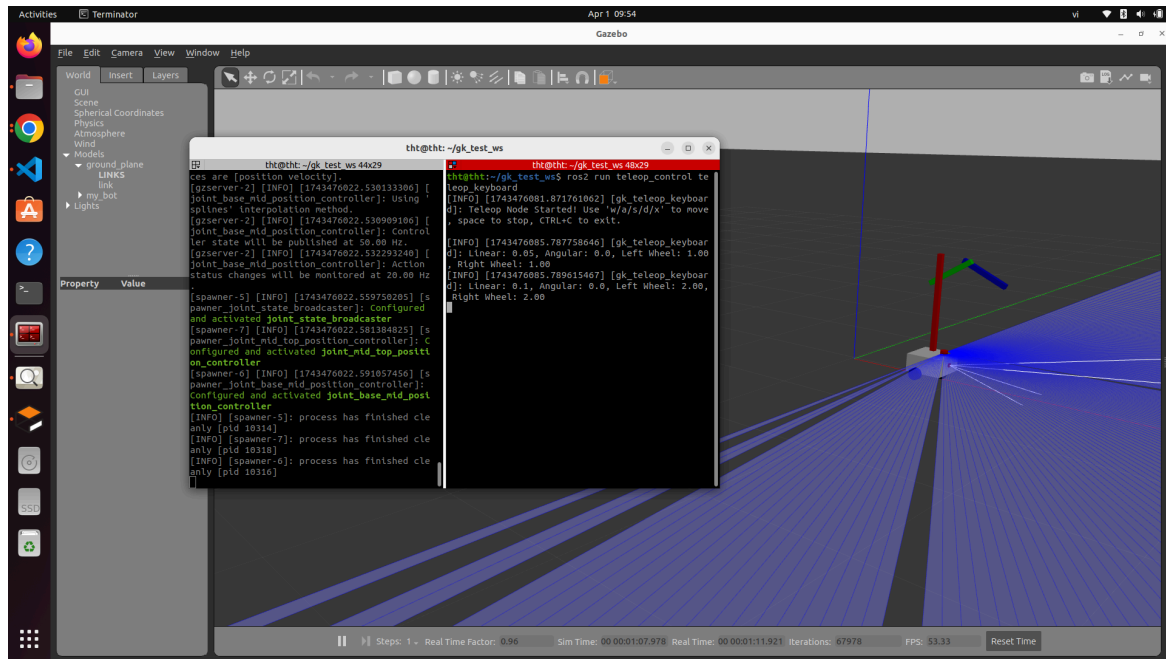
với các params được giải thích trên dòng code

```
# Robot parameters
WHEEL_RADIUS = 0.05 # r = 5 cm
WHEEL_BASE = 0.35 # Distance between wheels (0.175 * 2)
```

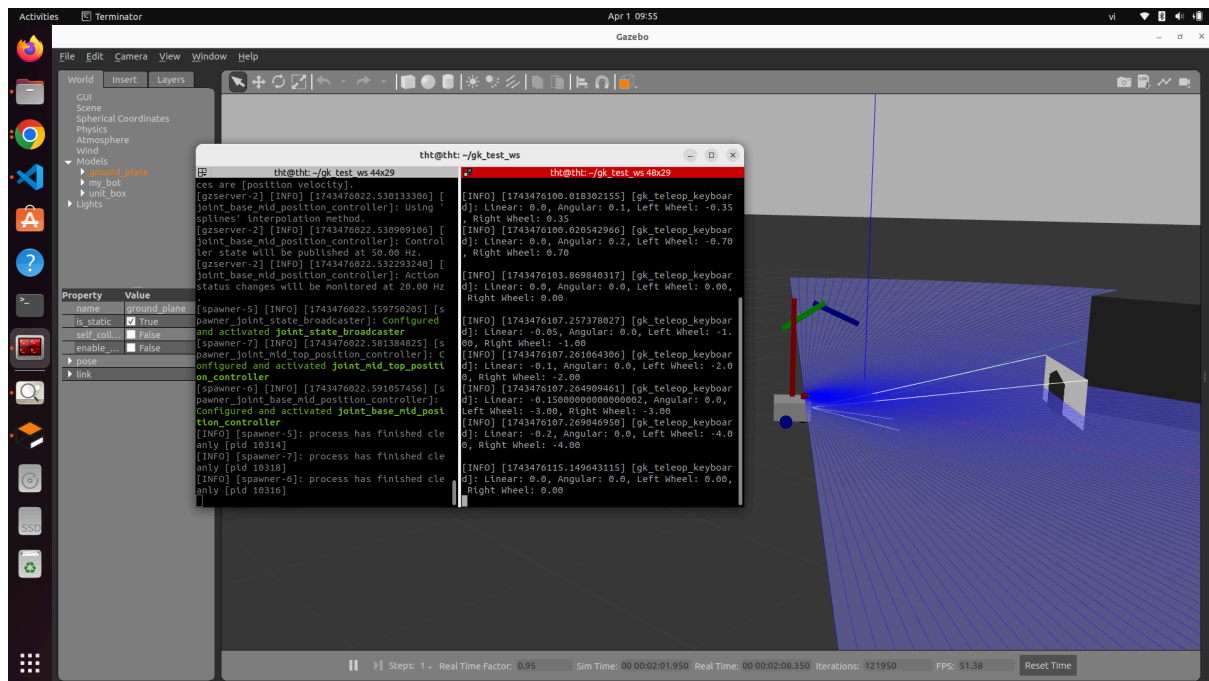
Kết quả:



=>



=>



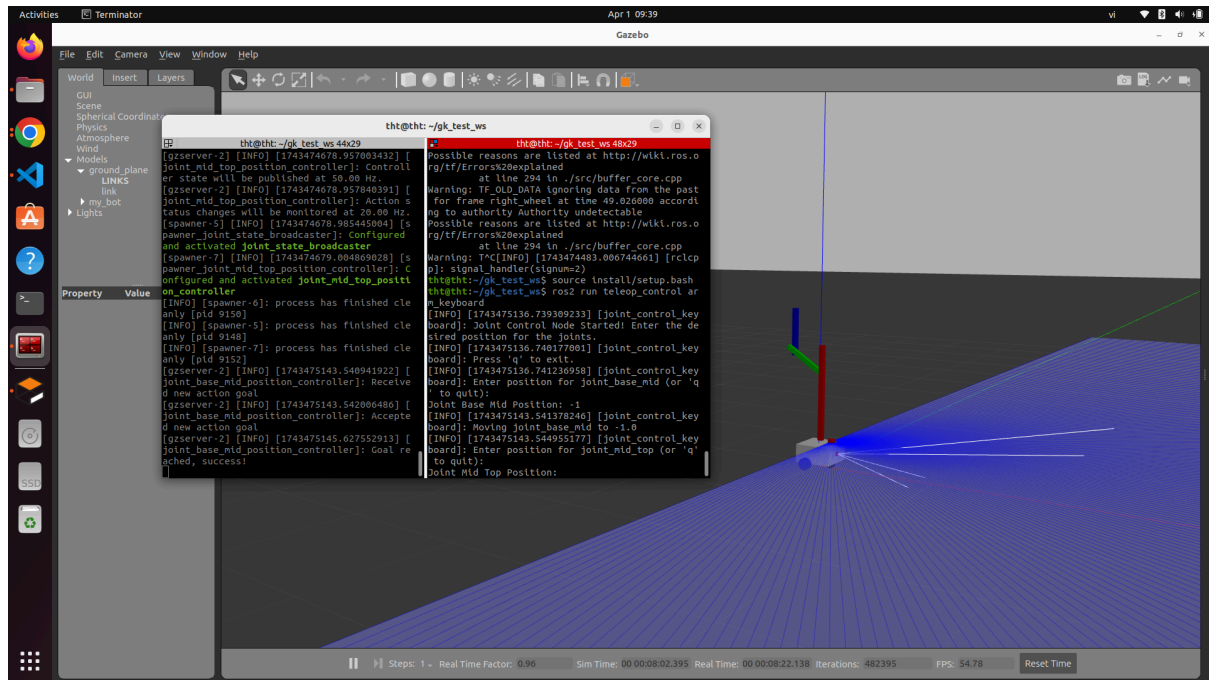
(3 ảnh là 3 vị trí khác nhau đã qua điều khiển của robot)

Code:

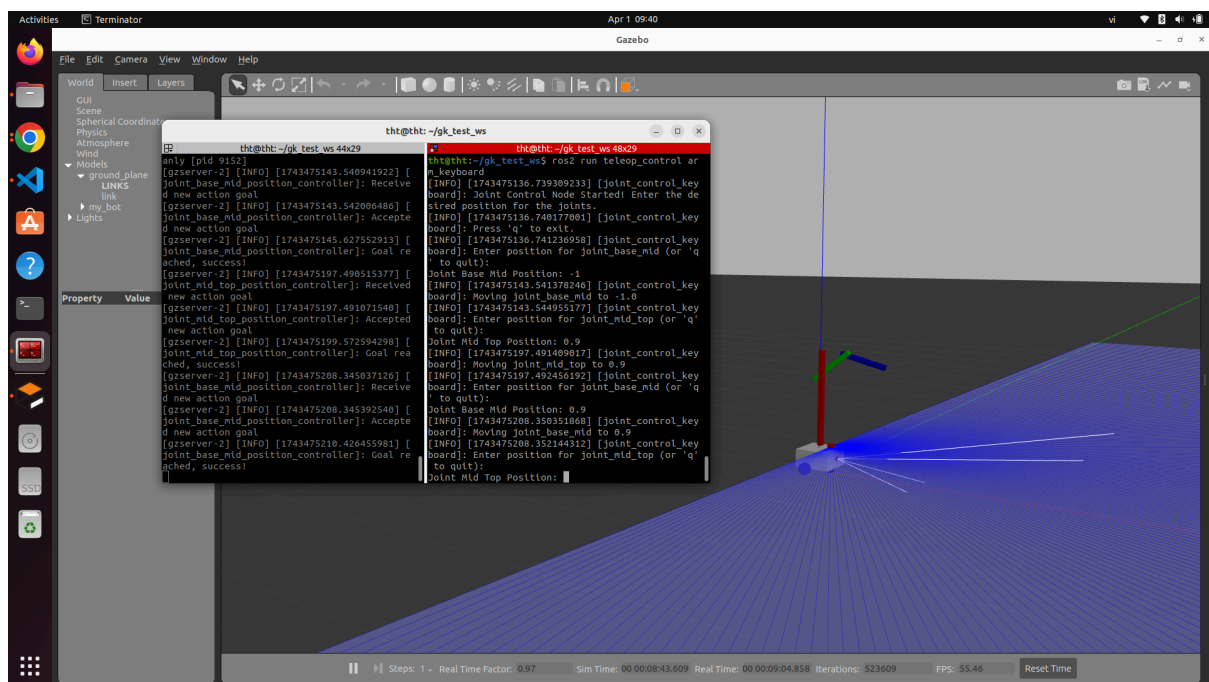
https://github.com/TrH-Thang/gk_diffbot2/blob/main/teleop_control/teleop_control/gk_teleop_keyboard.py

5. Động học và điều khiển tay máy

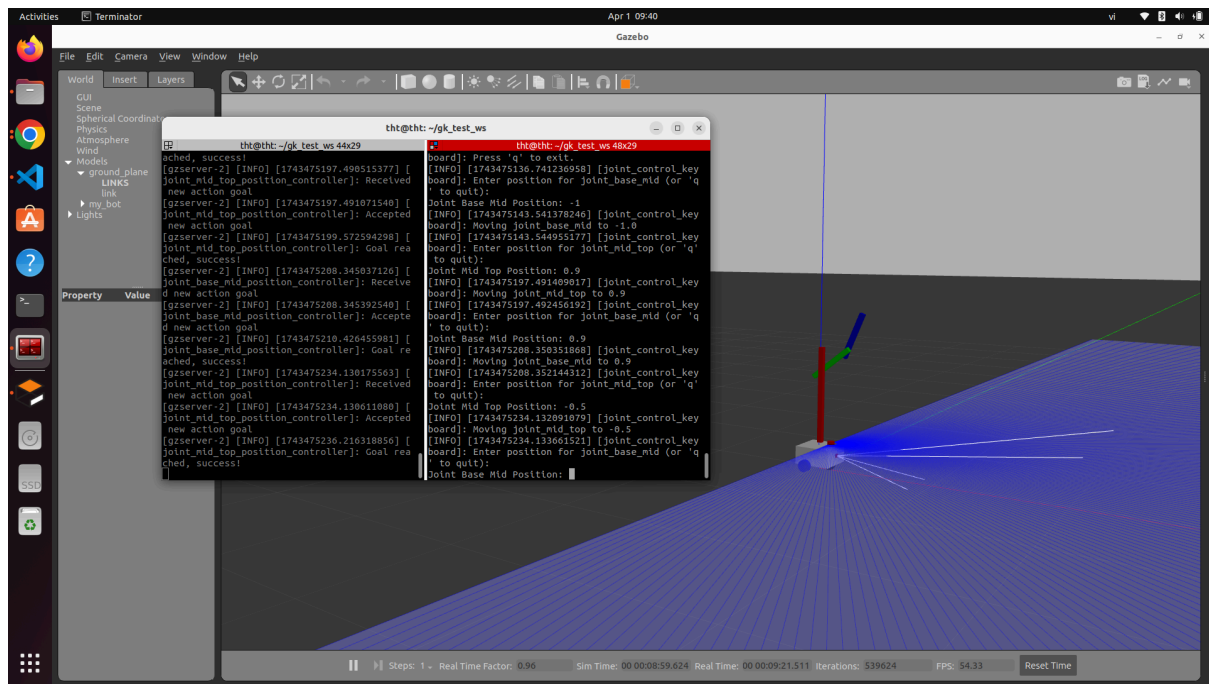
- Mô tả: Nhập vị trí khớp vào và tay máy sẽ di chuyển (tiền đề cho điều khiển tay máy khi biết vị trí cuối, sử dụng động học ngược tính vị trí các khớp, truyền vào node trên để điều khiển tay máy)



=>



=>



(Đây là 3 ảnh với 3 vị trí khớp sau khi truyền vị trí khớp vào bằng node điều khiển)

Code:

https://github.com/TrH-Thang/gk_diffbot2/blob/main/teleop_control/teleop_control/gk_arm_keyboard.py