

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



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

**Sinh viên thực hiện:**

HỌ VÀ TÊN	Vũ Đình Đức
Mã sinh viên	22027518

# I. TỔNG QUAN VỀ PROJECT GIỮA KÌ

## 1. Dạng robot

-Loại di chuyển : carlike-4 bánh skid-steer

-Tay máy :

+Khớp 1 :revolute

+Khớp 2: prismatic

-Các cảm biến : Camera,imu,encoder,lidar

## 2. Động học

### a. Mô hình xe

- Robot của em gồm 4 bánh kiểu skid-steer (giống xe tăng ) trong đó 2 bánh cùng bên nhận cùng vận tốc và di chuyển cùng hướng

+ Hai bánh trái (front\_left\_joint, back\_left\_joint) quay cùng tốc độ ( $\omega_{left}$ ).

+ Hai bánh phải (front\_right\_joint, back\_right\_joint) quay cùng tốc độ ( $\omega_{right}$ ).

-ĐỘNG HỌC THUẬN

+Vận tốc tuyến tính :

$$V=r/2(\omega_{left}+\omega_{right})$$

+Vận tốc góc :

$$\omega=(r/L)(\omega_{right}-\omega_{left})$$

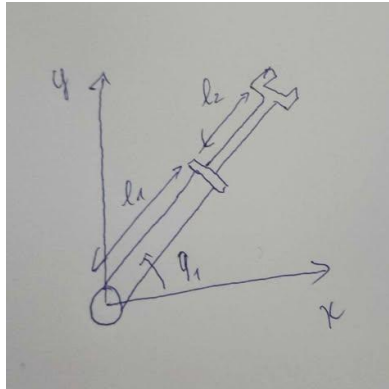
-Động học ngược : vận tốc dài và vận tốc góc được lấy từ topic /cmd\_vel

Vận tốc bánh xe được tính bằng:

$$\omega_{left} = \frac{1}{r} \left( v - \frac{L}{2} \omega \right)$$

$$\omega_{right} = \frac{1}{r} \left( v + \frac{L}{2} \omega \right)$$

## b .Tay máy



### Vị trí của khớp 2 (gốc của link 2):

- Link 1 quay một góc  $q_1$  quanh trục  $z_0$ , nên tọa độ của khớp 2 (gốc của link 2) trong hệ tọa độ gốc là:

$$x_{joint2} = l_1 \cos(q_1)$$

$$y_{joint2} = l_1 \sin(q_1)$$

### Vị trí của điểm cuối (end-effector):

- Link 2 tịnh tiến một khoảng  $q_2$  dọc theo trục của link 1 (hướng  $x_1$ ).
- Trục  $x_1$  của link 1 có hướng  $(\cos(q_1), \sin(q_1))$ , nên tọa độ của điểm cuối (end-effector) là:

$$x_e = l_1 \cos(q_1) + q_2 \cos(q_1) = (l_1 + q_2) \cos(q_1)$$

$$y_e = l_1 \sin(q_1) + q_2 \sin(q_1) = (l_1 + q_2) \sin(q_1)$$

## 3. Kích thước

- Thân xe : 210 x 180 x 10 mm

(gắn với động cơ bằng các thanh hình chữ u)

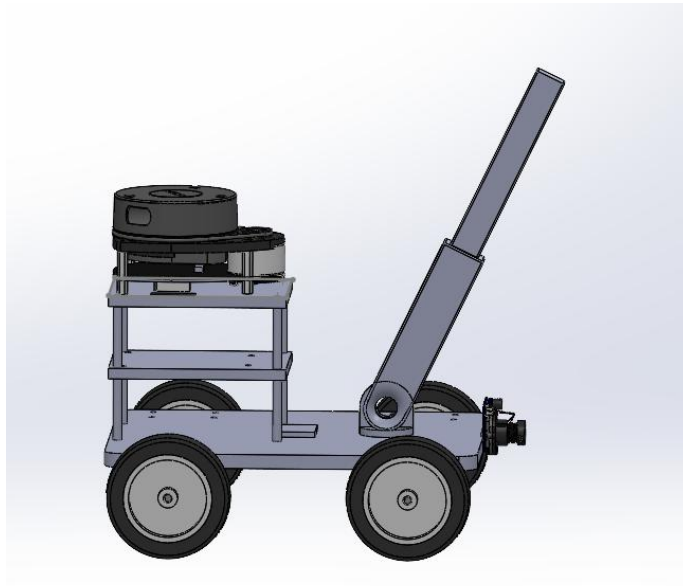
- Bánh xe : đường kính 70 mm, chiều ngang 20 mm

-Khớp 1 : 100 x 25 x 20 mm

-Khớp 2 : 90 x25 x20 mm

## II. Bản vẽ SOLIDWORKS

### 1. Tổng thể



Gồm các chi tiết :

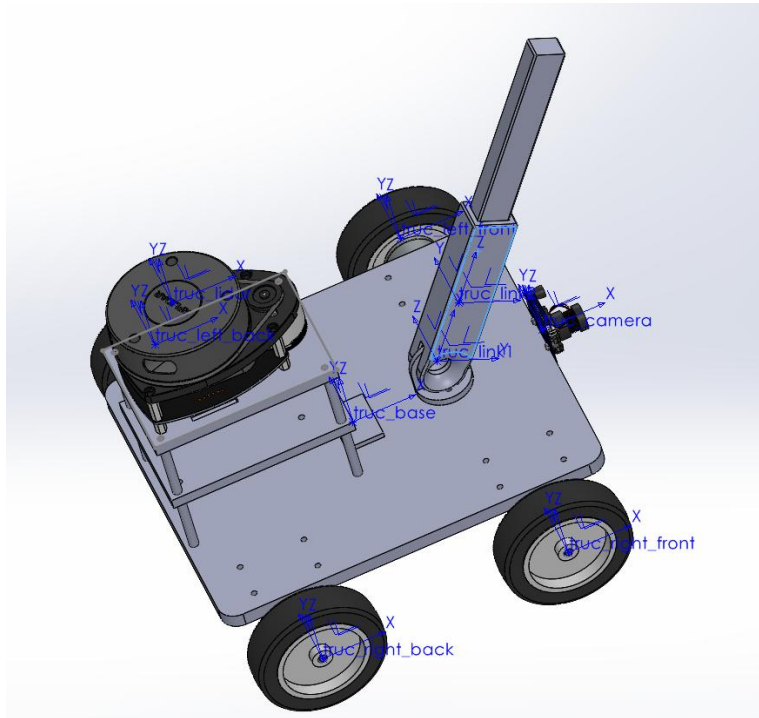
- +4 bánh xe
- +Thân xe
- +Tay máy
- +Động cơ
- +Cảm biến
- +Giá đỡ

## 2. Cách đặt trục tọa độ

- Bánh xe :
  - + Gốc tọa độ đặt tại tâm xoay của khớp
  - + Trục y theo chiều xoay của bánh xe
  - +Trục x theo chiều di chuyển của xe
  - +Trục z hướng lên

- Tay máy
  - +Khớp 1: trục z theo trục xoay của khớp, trục x hướng theo chiều tay máy , trục y theo quy tắc bàn tay phải
  - +khớp 2: trục z theo chiều trùng với x của khớp 1, trục x vuông góc với khớp 1, trục y theo quy tắc bàn tay phải

- +Các cảm biến: đặt theo bánh xe , gốc tọa độ nằm ở tâm của cảm biến



### III. URDF ,gazebo

#### 1. URDF

- Cấu trúc file

```
duc@duc-Nitro-AN515-45:~/catkin_ws$ check_urdf /home/duc/catkin_ws/src/xe_4/urdf/xe_4.urdf
robot name is: xe_4
----- Successfully Parsed XML -----
root Link: base_link has 8 child(ren)
  child(1): back_left_link
  child(2): back_right_link
  child(3): camera_link
  child(4): front_left_link
  child(5): front_right_link
  child(6): imu_link
  child(7): lidar_Link
  child(8): link1_link
            child(1): link2 link
```

- Tổng quan Tổng quan về file URDF

+Cấu trúc chính:

**Phần thân chính:** Gồm base\_link và 4 bánh xe (front\_left\_link, front\_right\_link, back\_left\_link, back\_right\_link).

**Tay máy:** Gồm 2 link (link1\_link, link2\_link) với 2 khớp (link1\_joint - revolute, link2\_joint - prismatic).

## Cảm biến:

- IMU (imu\_link).
- Camera (camera\_link).
- LiDAR (lidar\_Link).

## Plugin Gazebo:

- Plugin điều khiển bánh xe: libgazebo\_ros\_skid\_steer\_drive.so.
- Plugin điều khiển tay máy: libgazebo\_ros\_joint\_pose\_trajectory.so.
- Plugin cho cảm biến: IMU, camera, LiDAR.

## Chi tiết:

- Mỗi link có <inertial>, <visual>, và <collision> để định nghĩa khối lượng, hình dạng hiển thị, và va chạm.
- Các joint định nghĩa mối quan hệ giữa các link (parent-child) và loại chuyển động (quay, tịnh tiến, cố định)
- IMU : topic : /imu , giúp cung cấp dữ liệu về gia tốc và hướng
- Camera : topic:/camera1/image\_raw giúp cung cấp hình ảnh thực tế
- Lidar : topic ;/scan giúp cung cấp dữ liệu quét 2D

-File URDF Định nghĩa cấu trúc vật lý của robot (link, joint), các thuộc tính quán tính (inertial), hình học (visual, collision), và các plugin Gazebo để mô phỏng trong ROS/Gazebo.

### -Các plugin sử dụng

```
<plugin name="skid_steer_drive_controller" filename="libgazebo_ros_skid_steer_drive.so">
  <leftFrontJoint>front_left_joint</leftFrontJoint>
  <rightFrontJoint>front_right_joint</rightFrontJoint>
  <leftRearJoint>back_left_joint</leftRearJoint>
  <rightRearJoint>back_right_joint</rightRearJoint>
  <wheelSeparation>0.173316</wheelSeparation>
  <wheelDiameter>0.08</wheelDiameter>
  <torque>20</torque>
  <commandTopic>cmd_vel</commandTopic>
  <odometryTopic>odom</odometryTopic>
  <odometryFrame>odom</odometryFrame>
  <robotBaseFrame>base_link</robotBaseFrame>
  <publishWheelTF>true</publishWheelTF>
  <publishWheelJointState>true</publishWheelJointState>
  <updateRate>100.0</updateRate>
  <publishOdomTF>true</publishOdomTF>
</plugin>
```

- Điều khiển robot 4 bánh kiểu skid-steer (2 bánh trái và 2 bánh phải).
- Chuyển đổi lệnh vận tốc từ topic /cmd\_vel (dạng geometry\_msgs/Twist) thành vận tốc quay của các bánh xe.
- Publish dữ liệu odometry lên topic /odom (dạng nav\_msgs/Odometry).
- Publish trạng thái khớp lên topic /joint\_states (dạng sensor\_msgs/JointState).

```

<plugin name="camera_controller" filename="libgazebo_ros_camera.so">
  <alwaysOn>true</alwaysOn>
  <updateRate>0.0</updateRate>
  <cameraName>xe_4/camera1</cameraName>
  <imageTopicName>image_raw</imageTopicName>
  <cameraInfoTopicName>camera_info</cameraInfoTopicName>
  <frameName>camera_link</frameName>
</plugin>
<plugin name="camera_controller" filename="libgazebo_ros_camera.so">
  <alwaysOn>true</alwaysOn>
  <updateRate>0.0</updateRate>
  <cameraName>xe_4/camera1</cameraName>
  <imageTopicName>image_raw</imageTopicName>
  <cameraInfoTopicName>camera_info</cameraInfoTopicName>
  <frameName>camera_link</frameName>
</plugin>
<plugin filename="libgazebo_ros_imu_sensor.so" name="imu_plugin">
  <topicName>imu</topicName>
  <bodyName>imu_link</bodyName>
  <updateRateHZ>10.0</updateRateHZ>
  <gaussianNoise>0.0</gaussianNoise>
  <xyzOffset>0 0 0</xyzOffset>
  <rpyOffset>0 0 0</rpyOffset>
  <frameName>imu_link</frameName>
</plugin>
<plugin name="gazebo_ros_laser" filename="libgazebo_ros_laser.so">
  <topicName>/scan</topicName>
  <frameName>lidar_Link</frameName>
</plugin>

<plugin name="arm_controller" filename="libgazebo_ros_joint_pose_trajectory.so">
  <robotNamespace>/xe_4</robotNamespace>
  <updateRate>100.0</updateRate>
  <topicName>arm_controller/command</topicName>
  <jointName>link1_joint, link2_joint</jointName>
</plugin>

```

## 2. File gazebo

- file gazebo.launch
- Khởi động Gazebo server (gzserver) và Gazebo GUI (gzclient).
- Spawn robot xe\_4 vào Gazebo từ file URDF.
- Thiết lập transform giữa base\_link và base\_footprint.
- Load tham số điều khiển từ file controller.yaml.
- Khởi động các bộ điều khiển ( joint\_state\_controller).

## IV. Cơ chế điều khiển và thuật toán điều khiển trên gazebo

### a. Điều khiển robot di động (4 bánh skid-steer)

-Plugin điều khiển:

+Tên: skid\_steer\_drive\_controller.

+Chức năng:



- Chuyển đổi lệnh vận tốc từ topic /cmd\_vel (dạng geometry\_msgs/Twist) thành vận tốc quay của 4 bánh xe.
- Publish dữ liệu odometry lên topic /odom (dạng nav\_msgs/Odometry).

### + Tham số

- wheelSeparation và wheelDiameter: Định nghĩa kích thước robot để tính toán động học.
- torque: Lực xoắn tối đa áp dụng cho bánh xe (20 Nm).
- commandTopic và odometryTopic: Topic đầu vào (/cmd\_vel) và đầu ra (/odom).
- updateRate: Tần số cập nhật (100 Hz).

### +Cơ chế điều khiển

- **Đầu vào:** Plugin nhận lệnh từ topic /cmd\_vel với: linear.x: Vận tốc tuyến tính (m/s) và angular.z: Vận tốc góc (rad/s).
- **Xử lý:** Plugin sử dụng động học ngược (inverse kinematics) để tính vận tốc quay của bánh trái ( $\omega_{left}$ ) và bánh phải ( $\omega_{right}$ ): hai bánh trái quay với vận tốc  $\omega_{left}$ , hai bánh phải quay với vận tốc  $\omega_{right}$

### +Script điều khiển robot di động:

Control.py :

- **Chức năng:** Gửi lệnh /cmd\_vel để điều khiển robot di động.
- **Topic:** /cmd\_vel (publish geometry\_msgs/Twist).

### +Thuật toán điều khiển

**Bước 1: Nhận lệnh từ /cmd\_vel:** Lấy  $v$  (linear.x) và  $\omega$  (angular.z).

**Bước 2: Tính động học ngược:** Tính vận tốc quay của bánh trái và phải

**Bước 3: Áp dụng vận tốc:**

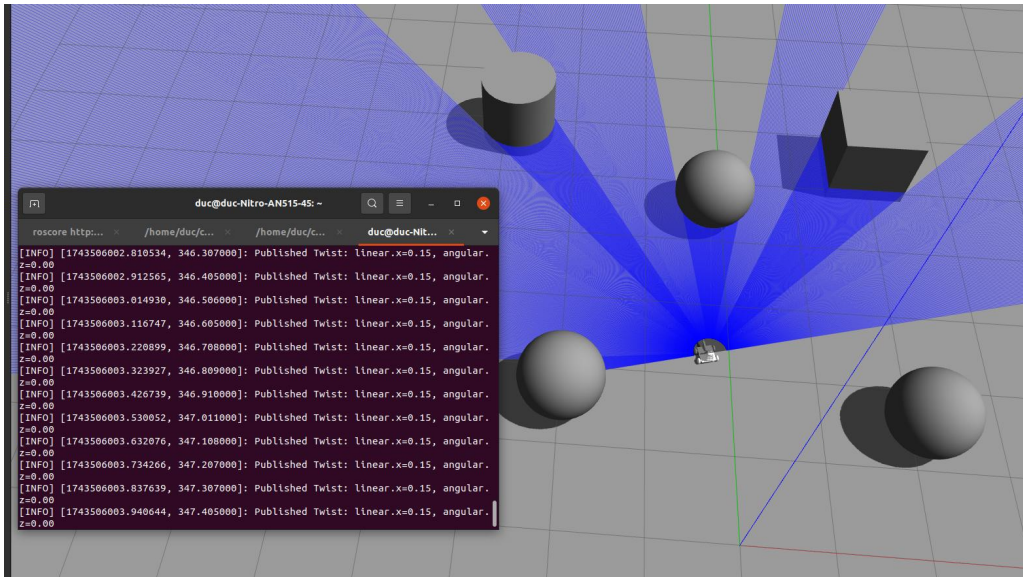
- Áp dụng  $\omega_{left}$  cho front\_left\_joint và back\_left\_joint.
- Áp dụng  $\omega_{right}$  cho front\_right\_joint và back\_right\_joint.

**Bước 4: Tính odometry:**

- Dùng động học thuận để tính lại  $v$  và  $\omega$  từ vận tốc thực tế của bánh xe:
- Tích phân  $v$  và  $\omega$  để tính vị trí  $(x, y, \theta)$  của robot
- Publish  $(x, y, \theta)$  lên /odom.



## Kết quả thực thi



### -Tóm tắt file điều khiển

- +**Khởi tạo**: Khởi tạo node teleop\_twist\_keyboard và publisher cho /cmd\_vel.
- +**Nhận phím**: Đọc lệnh từ bàn phím
- +**Tạo lệnh vận tốc**: Dựa trên phím nhấn, tính vận tốc tuyến tính (linear.x) và vận tốc góc (angular.z).
- +**Publish**: Publish message Twist lên /cmd\_vel

### b .Điều khiển tay máy

#### +Plugin điều khiển:

Tên: arm\_controller.

+**Chức năng**: Điều khiển 2 khớp của tay máy (link1\_joint, link2\_joint) dựa trên lệnh từ topic /arm\_controller/command.

#### +Cơ chế điều khiển:

##### Đầu vào:

- o Plugin nhận lệnh từ topic /arm\_controller/command (dạng trajectory\_msgs/JointTrajectory).
- o Lệnh này chứa các giá trị mục tiêu cho  $q_1$  (góc quay của link1\_joint) và  $q_2$  (khoảng cách tịnh tiến của link2\_joint).

##### Xử lý:

- o Plugin sử dụng bộ điều khiển nội bộ (thường là bộ điều khiển PID) để điều chỉnh vị trí của các khớp đến giá trị mục tiêu.
- o Các tham số PID được định nghĩa trong file controller.yaml

**Đầu ra**: Plugin cập nhật trạng thái khớp lên /joint\_states (vị trí và vận tốc của link1\_joint, link2\_joint).

## +Thuật toán

**Bước 1: Nhận lệnh từ /arm\_controller/command:**

Lấy giá trị mục tiêu  $q_1$  và  $q_2$  từ message JointTrajectory.

**Bước 2: Điều khiển PID:** Sử dụng bộ điều khiển PID để điều chỉnh vị trí của link1\_joint và link2\_joint đến giá trị mục tiêu.

- $K_p, K_i, K_d$  : Tham số PID từ controller.yaml.

**Bước 3: Cập nhật trạng thái:** Plugin cập nhật vị trí thực tế của các khớp lên /joint\_states.

Tóm tắt file điều khiển

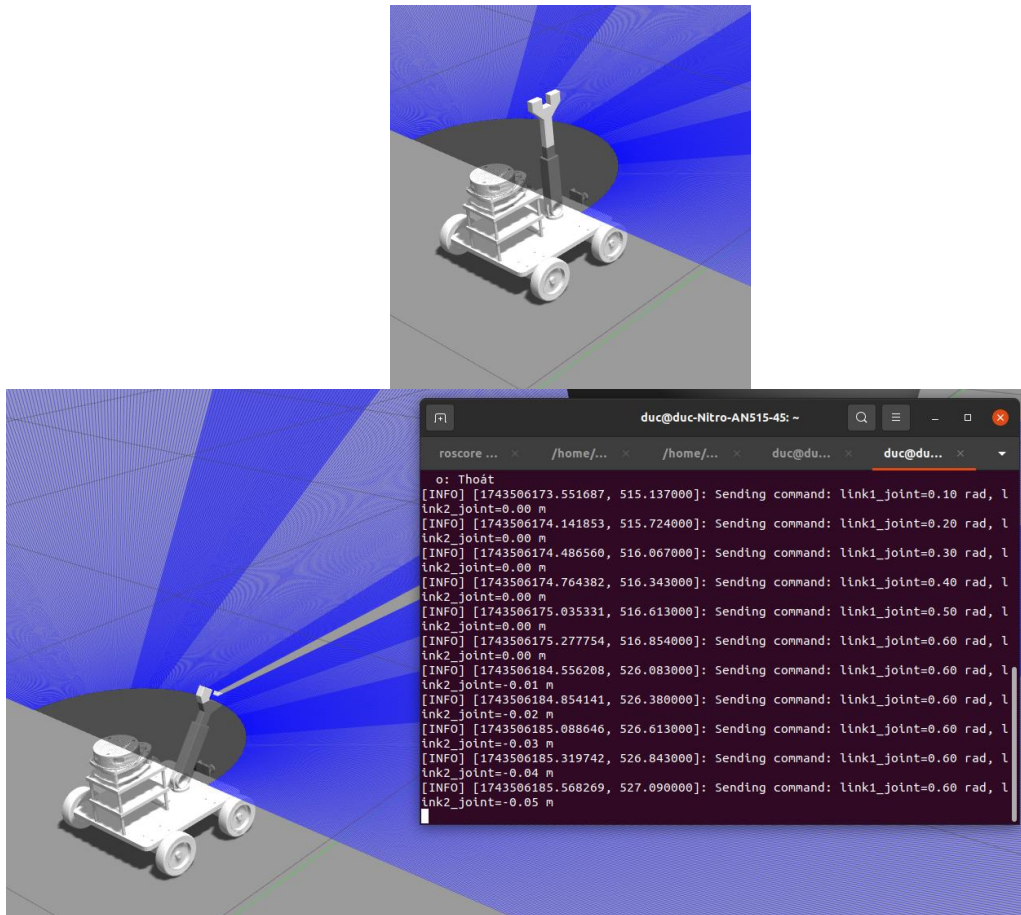
**Khởi tạo:** Khởi tạo node end\_effector\_publisher và publisher cho /end\_effector\_position.

**Tạo dữ liệu:**

- Tạo message PointStamped với (xe,ye)=(0.05,0.03).
- Frame tham chiếu: base\_link.

**Publish:** Publish message PointStamped lên /end\_effector\_position.

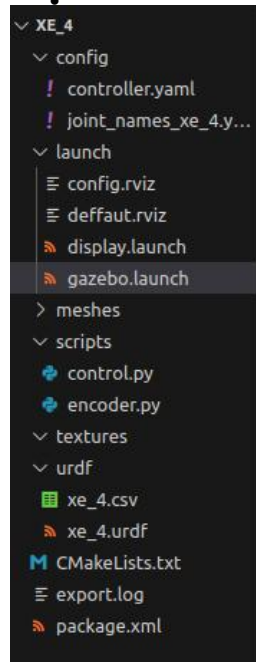
## Kết quả thực thi



Vấn đề gặp phải : - khi khởi chạy tay máy , tay máy không giữ được vị trí mà bị rơi tự do và khó lên lại vị trí

=> Hướng giải quyết :cập nhật vị trí một cách liên tục , luôn đặt vị trí ở (0,0) khi mới chạy

## V. CẤU TRÚC DỰ ÁN



### Các folder chính

-Folder config chứa các thông số của xe liên quan đến tên khớp, tham số điều khiển

-Folder launch chứa file launch gazebo.launch để khởi chạy gazebo và mô hình robot , display.launch để khởi chạy rviz , config.rviz để lưu cài đặt khi mở rviz

-Folder meshes để lưu bản vẽ trong solidworks

-Folder scripts chứa các file điều khiển

### Các topic phản hồi khi chạy

```

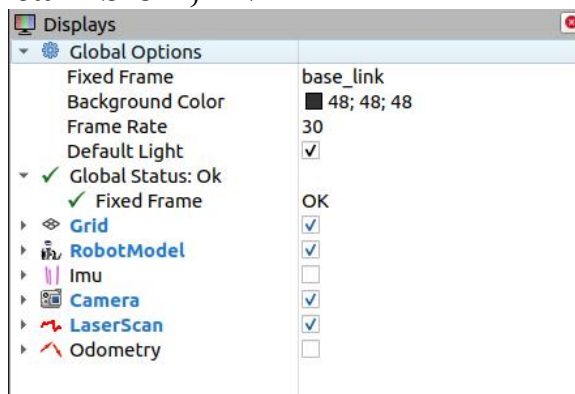
duc@duc-Nitro-AN515-45:~$ rostopic list
/clicked_point
/clock
/cmd_vel
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/performance_metrics
/gazebo/set_link_state
/gazebo/set_model_state
/imu
/initialpose
/joint_states
/move_base_simple/goal
/odom
/rosout
/rosout_agg
/scan
/tf
/tf_static
/xe_4/arm_controller/command
/xe_4/camera1/camera_info
/xe_4/camera1/image_raw
/xe_4/camera1/image_raw/compressed
/xe_4/camera1/image_raw/compressed/parameter_descriptions
/xe_4/camera1/image_raw/compressed/parameter_updates
/xe_4/camera1/image_raw/compressedDepth
/xe_4/camera1/image_raw/compressedDepth/parameter_descriptions
/xe_4/camera1/image_raw/compressedDepth/parameter_updates
/xe_4/camera1/image_raw/theora
/xe_4/camera1/image_raw/theora/parameter_descriptions
/xe_4/camera1/image_raw/theora/parameter_updates
/xe_4/camera1/parameter_descriptions
/xe_4/camera1/parameter_updates
duc@duc-Nitro-AN515-45:~$

```

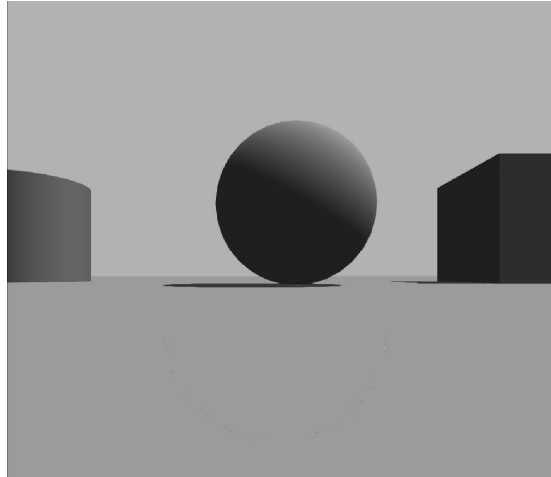
### -Các topic quan trọng

- /cmd\_vel: Nhận lệnh vận tốc (geometry\_msgs/Twist) để điều khiển robot.
- /odom: Publish dữ liệu odometry (nav\_msgs/Odometry) về vị trí và vận tốc.
- /joint\_states: Publish trạng thái khớp (sensor\_msgs/JointState) của bánh xe và tay máy
- /xe\_4/arm\_controller/command: Nhận lệnh điều khiển tay máy (trajectory\_msgs/JointTrajectory).
- /joint\_states: Cung cấp trạng thái thực tế của các khớp tay máy.
- /imu: Dữ liệu IMU (sensor\_msgs/Imu) về gia tốc và hướng.
- /xe\_4/camera1/\*: Dữ liệu hình ảnh (sensor\_msgs/Image), thông tin camera, và hình ảnh nền từ camera.
- /scan: Dữ liệu quét laser 2D (sensor\_msgs/LaserScan) từ LiDAR.

## VI. Dữ liệu cảm biến , Rviz



## 1. Camera



## 2. IMU

```

---
header:
  seq: 490
  stamp:
    secs: 49
    nsecs: 185000000
  frame_id: "imu_link"
orientation:
  x: -3.899743215094062e-05
  y: 2.0847554327566107e-05
  z: 0.9979155481182089
  w: -0.0645333779412996
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.06890679422950889
  y: 0.05848348797897446
  z: 0.018542673807809846
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: 0.07814545544324596
  y: -0.002874518877341604
  z: 9.351304810195888
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

```

### 3. Encoder(/odom)

```
header:
  seq: 10889
  stamp:
    secs: 109
    nsecs: 71000000
  frame_id: "odom"
child_frame_id: "base_link"
pose:
  pose:
    position:
      x: -1.8474572036959562
      y: -1.0912712174079067
      z: 0.0
    orientation:
      x: -7.632745047079195e-05
      y: -4.508577359183144e-05
      z: 0.6402426981409508
      w: 0.7481726886690663
  covariance: [0.0001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000000000000.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000000000000.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000000000000.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.01]
twist:
  twist:
    linear:
      x: -0.01105234986759758
      y: 0.0015178057873028276
      z: 0.0
    angular:
      x: 0.0
      y: 0.0
      z: 0.007172520567794685
  covariance: [0.0001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000000000000.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000000000000.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1000000000000.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.01]
```

### Thông tin từ encoder bằng code

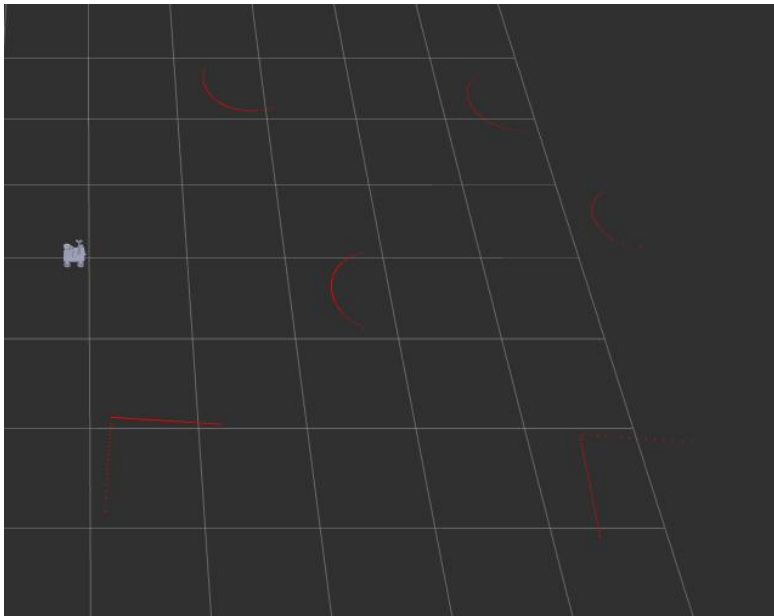


```

[INFO] [1743507242.283387, 190.789000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.284038, 190.790000]: Vận tốc: Linear = 0.007, Angular = 0.012
[INFO] [1743507242.284685, 190.790000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.285314, 190.791000]: Vận tốc: Linear = 0.007, Angular = 0.010
[INFO] [1743507242.286247, 190.792000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.286871, 190.792000]: Vận tốc: Linear = 0.007, Angular = 0.010
[INFO] [1743507242.297311, 190.802000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.298447, 190.803000]: Vận tốc: Linear = 0.007, Angular = 0.010
[INFO] [1743507242.342554, 190.847000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.345108, 190.848000]: Vận tốc: Linear = 0.007, Angular = 0.010
[INFO] [1743507242.346168, 190.851000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.346769, 190.851000]: Vận tốc: Linear = 0.009, Angular = 0.004
[INFO] [1743507242.347336, 190.852000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.347903, 190.852000]: Vận tốc: Linear = 0.008, Angular = 0.008
[INFO] [1743507242.348459, 190.853000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.348995, 190.853000]: Vận tốc: Linear = 0.007, Angular = 0.008
[INFO] [1743507242.349628, 190.854000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.350327, 190.855000]: Vận tốc: Linear = 0.007, Angular = 0.007
[INFO] [1743507242.357255, 190.862000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.358096, 190.862000]: Vận tốc: Linear = 0.007, Angular = 0.013
[INFO] [1743507242.398254, 190.902000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.399294, 190.904000]: Vận tốc: Linear = 0.008, Angular = 0.008
[INFO] [1743507242.400456, 190.905000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.401228, 190.906000]: Vận tốc: Linear = 0.009, Angular = 0.004
[INFO] [1743507242.401957, 190.906000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.402761, 190.907000]: Vận tốc: Linear = 0.010, Angular = -0.001
[INFO] [1743507242.403515, 190.908000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.404227, 190.909000]: Vận tốc: Linear = 0.004, Angular = 0.001
[INFO] [1743507242.407440, 190.912000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.408548, 190.913000]: Vận tốc: Linear = 0.012, Angular = -0.006
[INFO] [1743507242.417598, 190.922000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.419329, 190.923000]: Vận tốc: Linear = 0.005, Angular = 0.002
[INFO] [1743507242.458328, 190.962000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.459786, 190.964000]: Vận tốc: Linear = -0.005, Angular = -0.002
[INFO] [1743507242.460463, 190.964000]: Vị trí: x = -1.786, y = -1.169
[INFO] [1743507242.461123, 190.965000]: Vận tốc: Linear = 0.004, Angular = 0.002
[INFO] [1743507242.461886, 190.966000]: Vị trí: x = -1.786, y = -1.169

```

#### 4. Lidar



### VII. Tổng hợp và vấn đề cần khắc phục

#### 1 .Tổng hợp

- Mô hình xe khởi tạo thành công và chạy thành công trên gazebo và rviz
- File điều khiển có thể điều khiển xe và tay máy một cách liên tục
- Các topic hiển thị và chạy ổn định

#### 2 . Vấn đề cần khắc phục

- Bánh xe và tay máy di chuyển không đồng thời , bị xung đột
- Bánh xe di chuyển chưa đúng khoảng cách thực tế









