Báo cáo giữa kì môn học: Lập trình Robot với ROS

Họ và tên sinh viên: Nguyễn Mạnh Tường 22027536

Đề tài lựa chọn
Loại di chuyển: Car-like
Tay máy khớp 1: Rotation
Tay máy khớp 2: Rotation
Cảm biến : LIDAR, Camera, Encoder

I.Dạng robot,động học, kích thước

- Mô hình động học kinematic của robot carlike :

Mô hình của robot này được mô tả bởi các phương trình sau:

$$\dot{x} = v\cos(\theta)$$

$$\dot{y} = v\sin(\theta)$$

$$\dot{\theta} = \frac{v}{L}\tan(\delta)$$

trong đó:

- (x, y) là tọa độ của robot trên mặt phẳng.
- θ là góc hướng của robot so với trục x.
- v là vận tốc tuyến tính của robot.
- δ là góc lái (góc của bánh xe trước so với thân xe).
- L là khoảng cách giữa hai trục bánh xe (wheelbase).

II.Thiết kế solidworks, cách đặt hệ trục tọa độ

Bước 1: Xác định hệ trục tọa độ của ROS

Trong URDF (ROS), hệ trục tọa độ chuẩn là:

X: Hướng về phía trước.

Y: Hướng sang phải.

Z: Hướng lên trên.

Bước 2: Đặt lại hệ trục trong SolidWorks

1. Chỉnh lại hệ tọa độ gốc (Origin & Axes)

   Khi thiết kế mô hình trong SolidWorks, đảm bảo rằng hệ trục chính (Front, Top, Right) tương ứng với hệ trục của ROS.

   Nếu hệ trục bị lệch,có thể sử dụng Move/Copy Bodies hoặc tạo một Coordinate System mới.

2. Tạo một hệ tọa độ mới (Coordinate System)

   Vào Insert → Reference Geometry → Coordinate System.

   Chọn điểm gốc (Origin).

   Đặt trục X, Y, Z sao cho phù hợp với tiêu chuẩn của ROS.

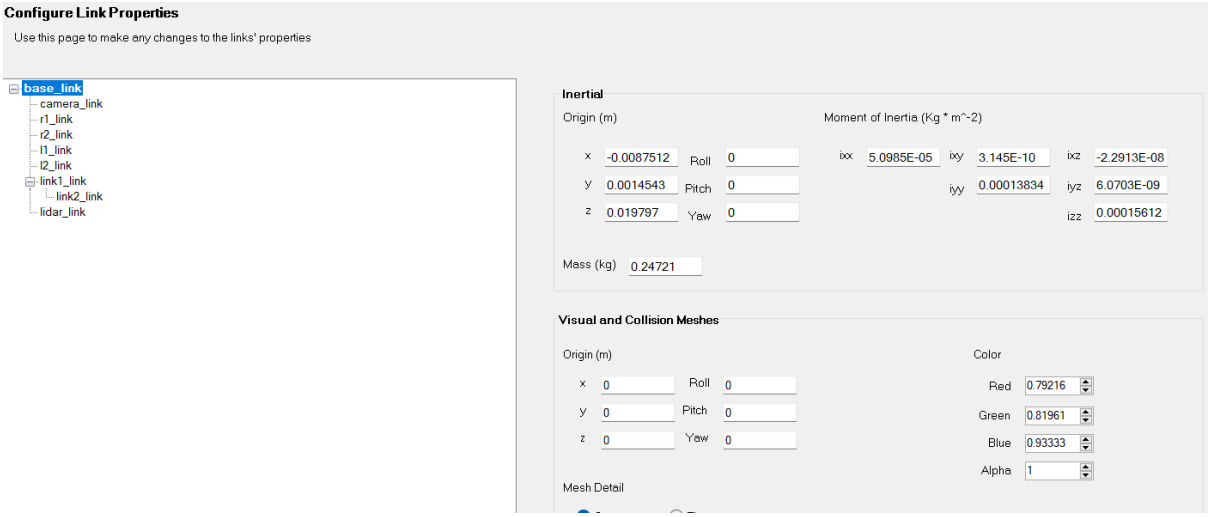
Bước 3: Xuất file STL/STEP từ SolidWorks

Khi xuất STL, đảm bảo chọn Output Coordinate System là hệ tọa độ mới đã tạo.

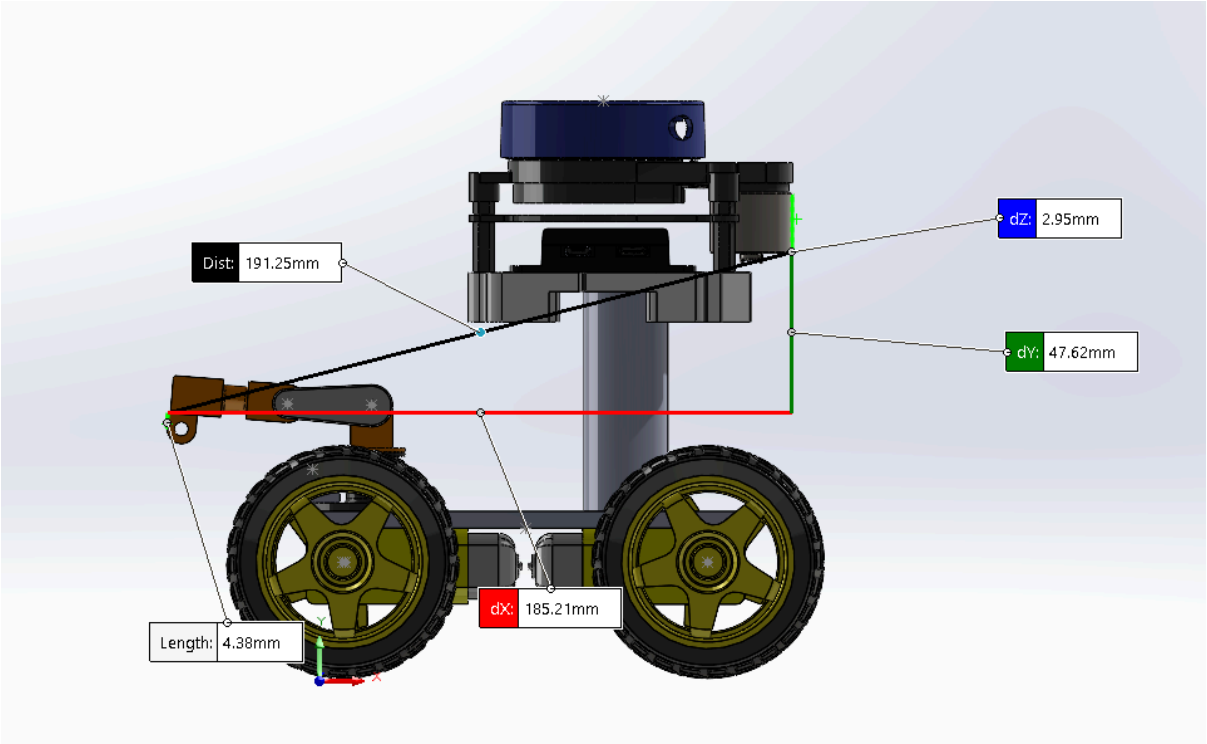
Nếu sử dụng plugin SW2URDF,cũng có thể đặt lại hệ trục tọa độ trong quá trình xuất.

Bước 4: Chỉnh lại URDF nếu cần

Nếu sau khi nhập vào ROS mà hệ trục vẫn chưa đúng, có thể sửa file URDF bằng cách chỉnh sửa thuộc tính `xyz` và `rpy` trong `<origin>` của từng link.



Đặt hệ trục tọa độ cho các link của robot



Kích thước

III. Mô tả file urdf, liên kết của các link, các cảm biến, mô tả gazebo

Các Link:

1. base_link – Thân chính của robot.
2. camera_link – Link chứa camera.
3. r1_link – Link cánh tay phải 1.
4. r2_link – Link cánh tay phải 2.
5. l1_link – Link cánh tay trái 1.
6. l2_link – Link cánh tay trái 2.
7. link1_link – Một bộ phận liên kết khác.
8. link2_link – Một bộ phận liên kết khác.
9. lidar_link – Link chứa cảm biến LiDAR.

Các Joint:

1. camera_joint – Kết nối camera với robot.
2. r1_joint – Khớp nối cánh tay phải 1.
3. r2_joint – Khớp nối cánh tay phải 2.

4. l1_joint – Khớp nối cánh tay trái 1.
5. l2_joint – Khớp nối cánh tay trái 2.
6. link1_joint – Khớp nối liên kết 1.
7. link2_joint – Khớp nối liên kết 2.
8. lidar_joint – Khớp nối cảm biến LiDAR.

Giải thích ý nghĩa các Joint

1. camera_joint: Gắn camera cố định vào base_link, không có chuyển động.
2. r1_joint & r2_joint: Hai khớp nối cánh tay phải, có thể quay quanh trục.
3. l1_joint & l2_joint: Hai khớp nối cánh tay trái, có thể quay tương tự cánh tay phải.
4. link1_joint & link2_joint: Các khớp nối liên kết phần trên của robot.
5. lidar_joint: Cố định cảm biến LiDAR trên thân robot.

IV.Cơ chế điều khiển trên gazebo

1. Điều khiển chuyển động của robot

Robot sử dụng cơ chế differential drive (truyền động vi sai) để di chuyển, với bốn bánh xe được gắn vào base_link thông qua các khớp liên tục (continuous joints):

> Bánh xe bên phải: r1_joint (gắn với r1_link) và r2_joint (gắn với r2_link).
>
> Bánh xe bên trái: l1_joint (gắn với l1_link) và l2_joint (gắn với l2_link).

Plugin điều khiển

Plugin diff_drive_controller (libgazebo_ros_diff_drive.so) được sử dụng để điều khiển chuyển động:

> Chức năng chính: Chuyển đổi lệnh vận tốc tuyến tính (linear velocity) và góc (angular velocity) từ topic ROS /cmd_vel thành vận tốc quay cho các bánh xe.

Cấu hình quan trọng:

Khớp điều khiển: leftJoint (l1_joint, l2_joint) và rightJoint (r1_joint, r2_joint).

Khoảng cách giữa các bánh xe (wheelSeparation): 0.1566 m.

Đường kính bánh xe (wheelDiameter): 0.07 m.

Topic điều khiển: /cmd_vel.

Topic xuất bản dữ liệu odometry: /odom.

Cách hoạt động: Khi một lệnh vận tốc được gửi đến /cmd_vel (ví dụ: thông qua geometry_msgs/Twist), plugin tính toán và áp dụng vận tốc quay cho các bánh xe trái và phải để robot di chuyển tiến, lùi, hoặc xoay.

Cách điều khiển

Gửi lệnh vận tốc đến topic /cmd_vel bằng ROS (ví dụ: sử dụng rostopic pub hoặc một node điều khiển).

Ví dụ lệnh để robot di chuyển thẳng với vận tốc 0.2 m/s:

rostopic pub /cmd_vel geometry_msgs/Twist '{linear: {x: 0.2, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}'

2. Điều khiển cơ cấu (cánh tay hoặc bộ phận khác)

Robot có hai liên kết link1_link và link2_link, được kết nối qua các khớp liên tục:

link1_joint: Gắn link1_link vào base_link.

link2_joint: Gắn link2_link vào link1_link.

Cơ chế điều khiển

Các khớp này sử dụng ROS Control thông qua plugin gazebo_ros_control (libgazebo_ros_control.so).

Transmission: Cả link1_joint và link2_joint được cấu hình với transmission_interface/SimpleTransmission và giao diện phần cứng hardware_interface/PositionJointInterface. Điều này cho phép điều khiển vị trí của các khớp.

Cách hoạt động: Các controller ROS (ví dụ: position_controllers/JointPositionController) được sử dụng để gửi lệnh vị trí đến các khớp thông qua ROS.

3. Mô phỏng và sử dụng dữ liệu từ camera

Camera được gắn vào camera_link (gắn cố định với base_link qua camera_joint).

Plugin mô phỏng

Plugin camera_controller (libgazebo_ros_camera.so) được sử dụng:

Cấu hình:

Tần số cập nhật: 30 Hz.

Độ phân giải: 640x480.

Topic xuất bản: /camera/image_raw (hình ảnh) và /camera/camera_info (thông tin camera).

Chức năng: Mô phỏng camera và cung cấp dữ liệu hình ảnh cho ROS.

Cách sử dụng

Đăng ký topic /camera/image_raw để nhận dữ liệu hình ảnh (ví dụ: dùng rqt_image_view hoặc node xử lý hình ảnh).

---

4. Mô phỏng và sử dụng dữ liệu từ lidar

Lidar được gắn vào lidar_link (gắn cố định với base_link qua lidar_joint).

Plugin mô phỏng

Plugin gazebo_ros_head_rplidar_controller (libgazebo_ros_laser.so) được sử dụng:

Cấu hình:

Tần số cập nhật: 30 Hz.

Góc quét: 0 đến 360 độ (0 đến 6.28319 rad).

Phạm vi: 0.12 m đến 10 m.

Topic xuất bản: /scan.

Chức năng: Mô phỏng lidar và cung cấp dữ liệu quét (scan) dưới dạng sensor_msgs/LaserScan.

Cách sử dụng

Đăng ký topic /scan để nhận dữ liệu quét (ví dụ: dùng rviz hoặc node xử lý dữ liệu lidar).

Quy trình điều khiển robot trong Gazebo

Để điều khiển robot trong Gazebo, thực hiện các bước sau:

1. Khởi động Gazebo:

   Tải mô hình robot từ tệp URDF bằng roslaunch (ví dụ: roslaunch gazebo_ros empty_world.launch kết hợp với spawn URDF).

2. Chạy các node ROS:

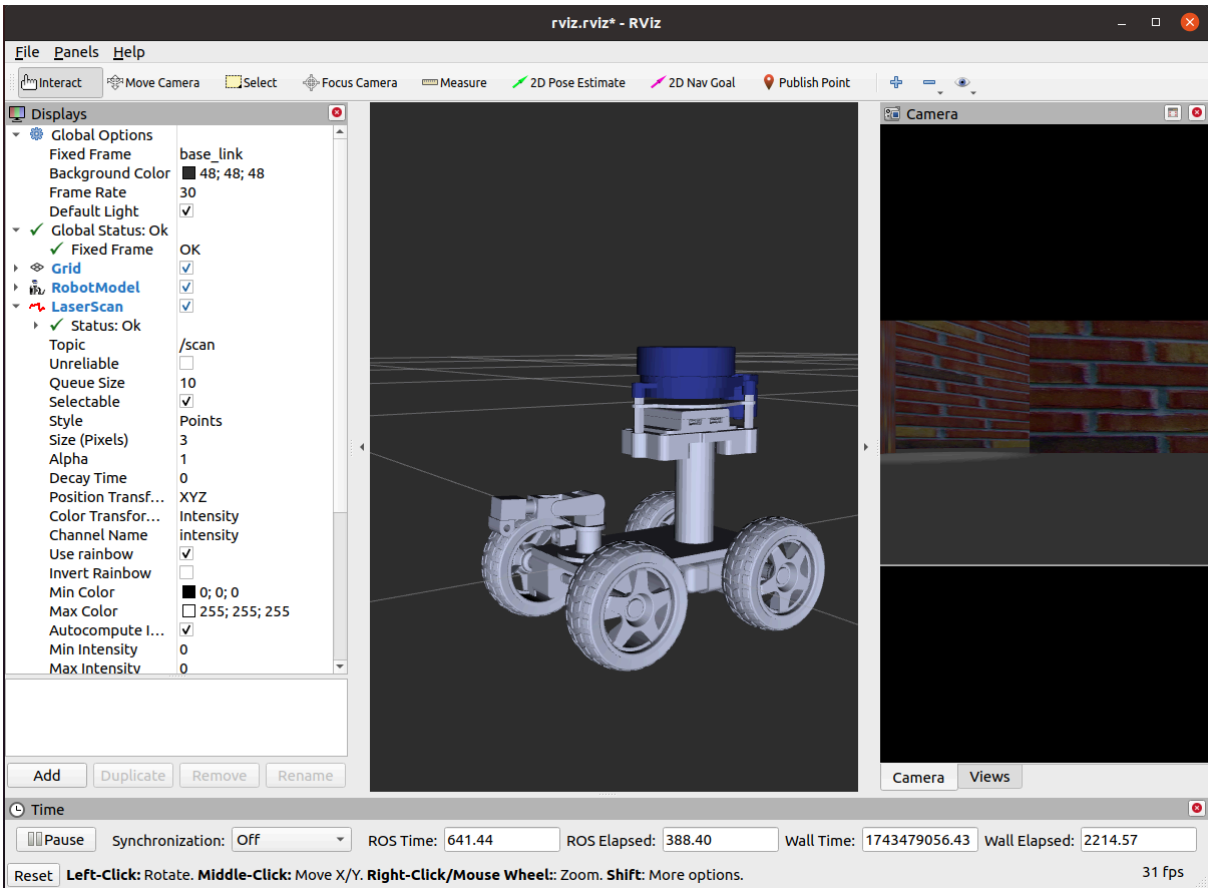
   Node gửi lệnh đến /cmd_vel cho chuyển động.

   Node controller cho link1_joint và link2_joint.

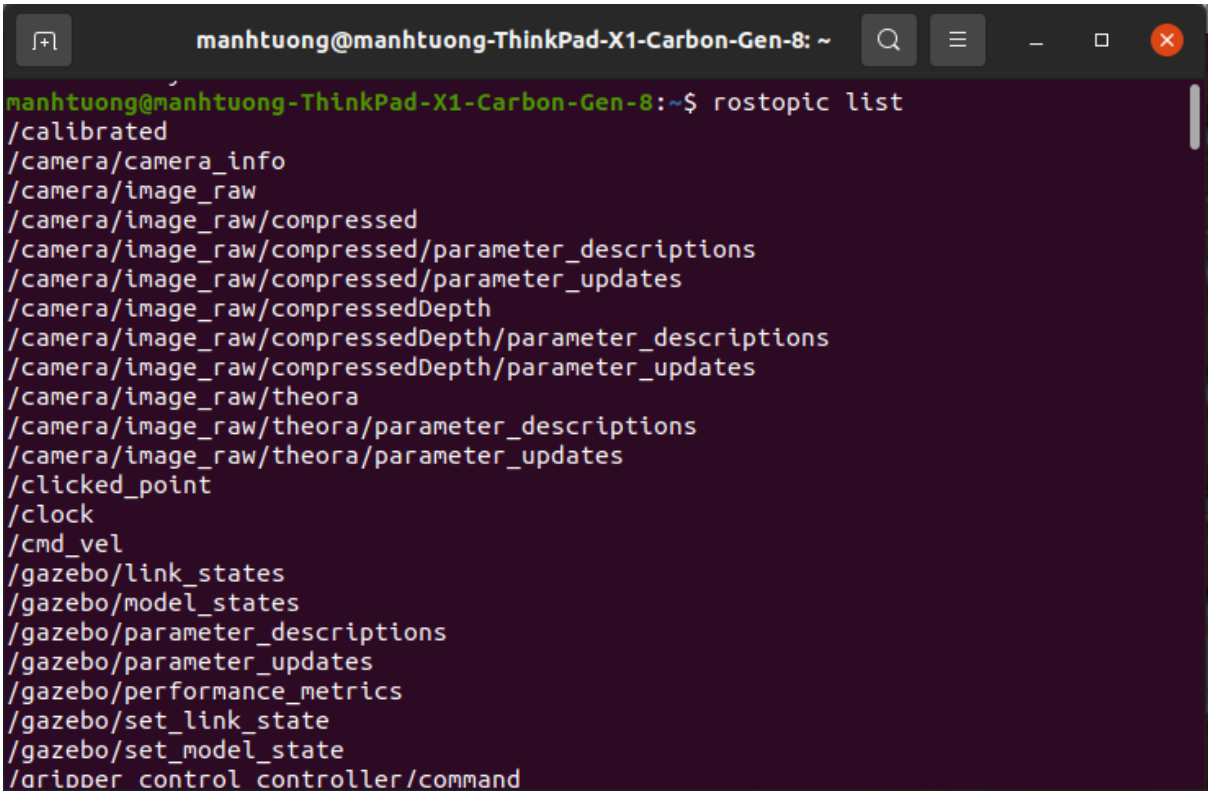
3. Xử lý dữ liệu cảm biến:

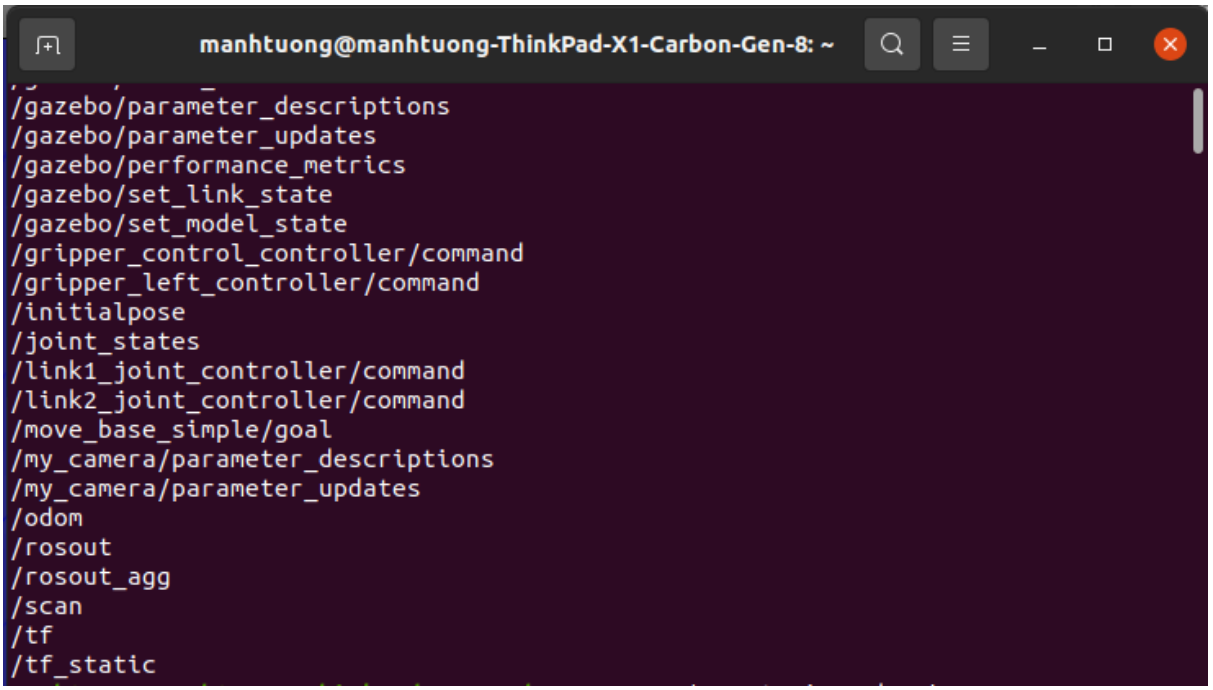
   Sử dụng dữ liệu từ /camera/image_raw và /scan cho các ứng dụng như điều hướng, lập bản đồ.

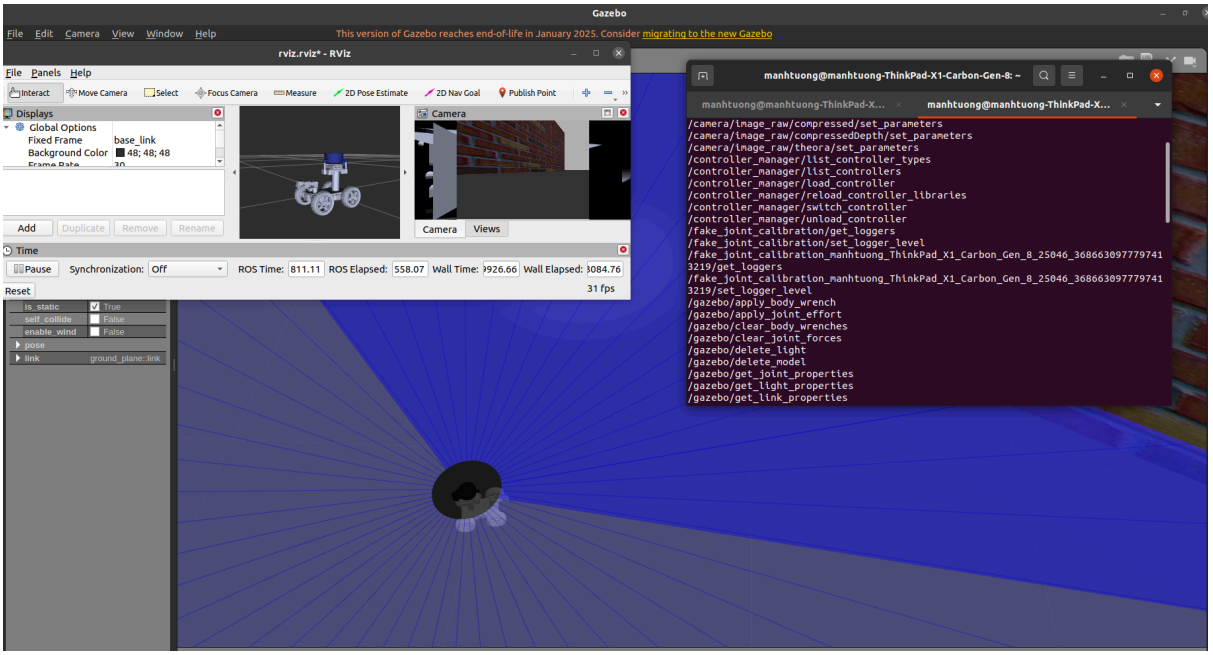
V.Các thành phần chính của code, structure folder dự án

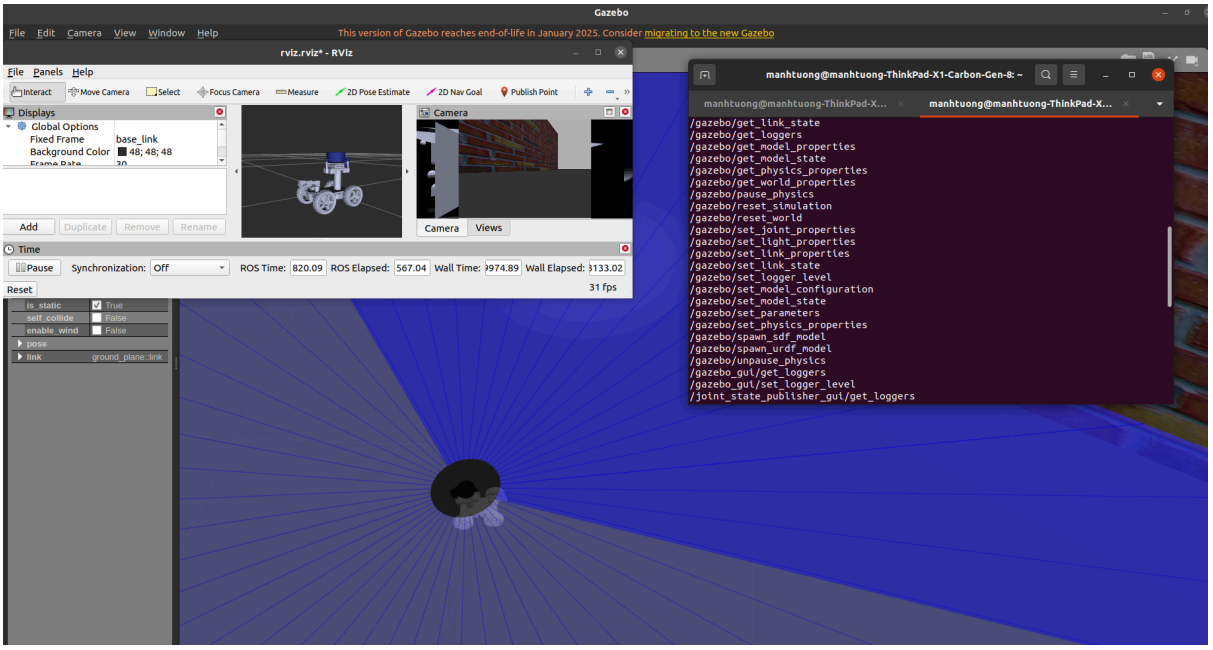


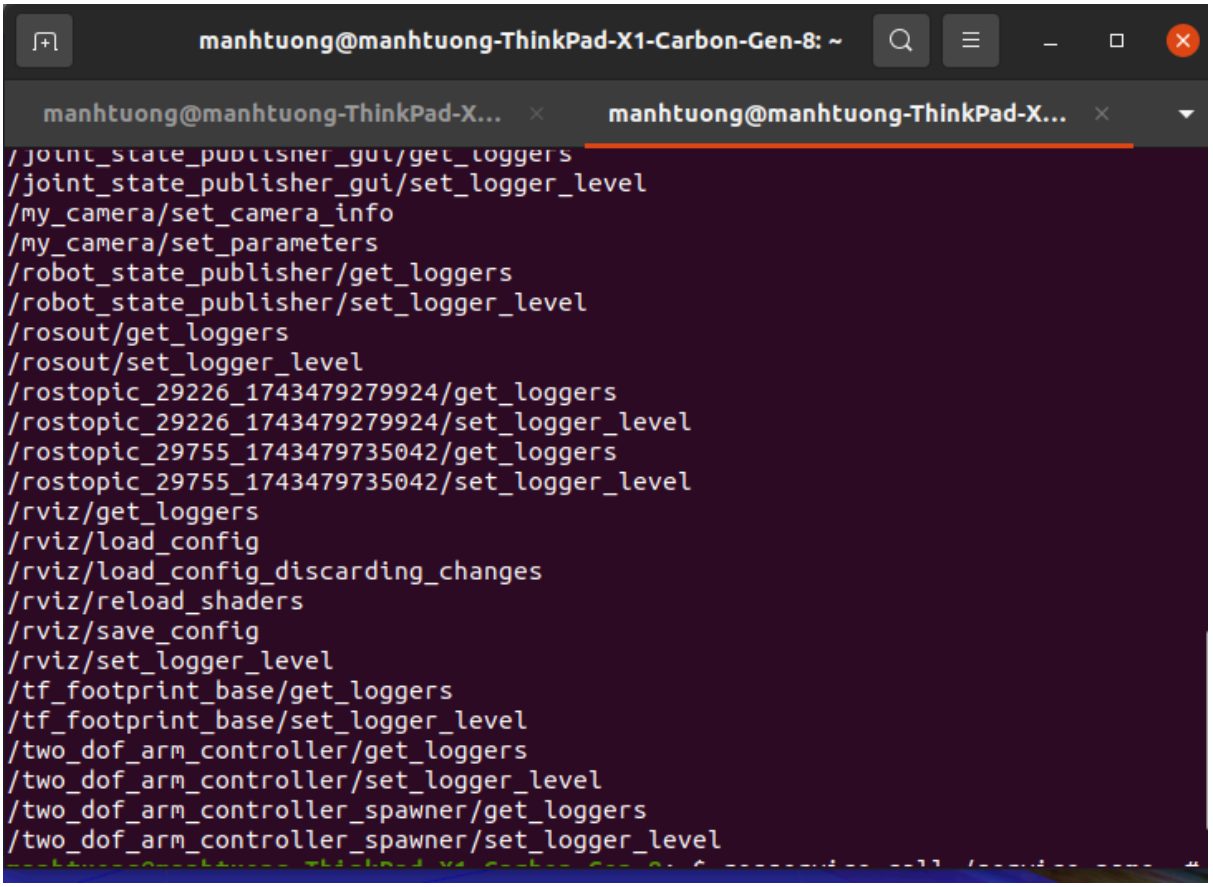
Các topic được sử dụng, Khi khởi chạy `roscore`, một số topic quan trọng sẽ được tạo ra để phục vụ việc trao đổi dữ liệu giữa các node.

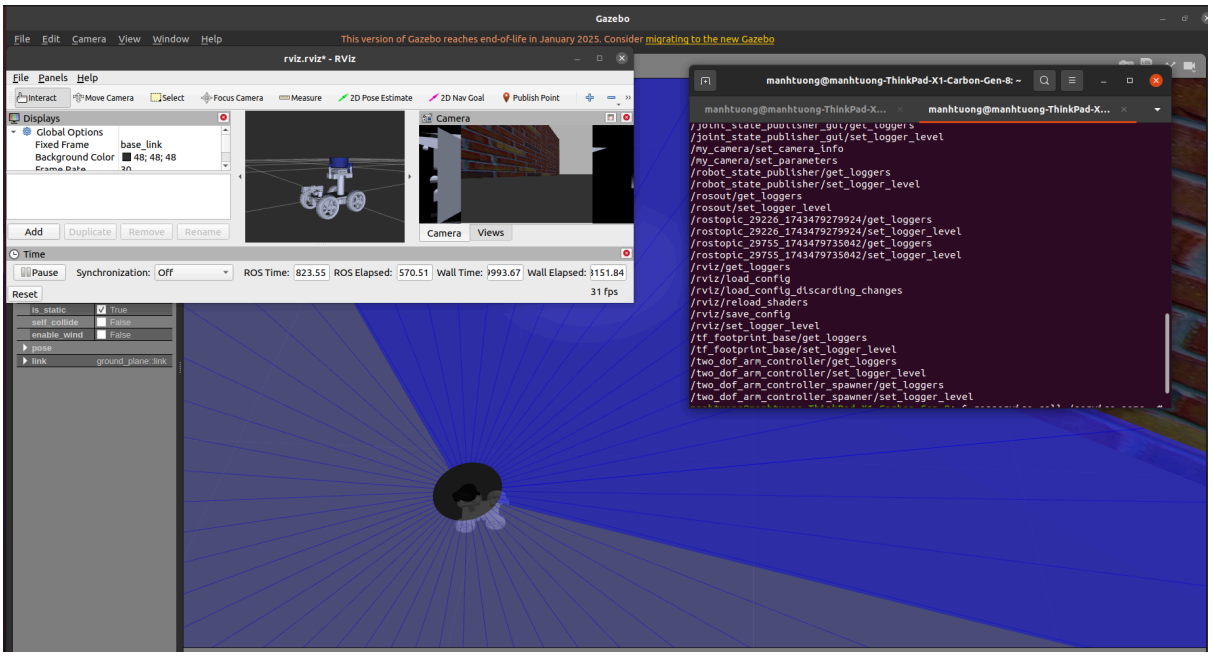

manhtuong@manhtuong-ThinkPad-X1-Carbon-Gen-8: ~
manhtuong@manhtuong-ThinkPad-X1-Carbon-Gen-8:~$ rostopic list
/calibrated
/camera/camera_info
/camera/image_raw
/camera/image_raw/compressed
/camera/image_raw/compressed/parameter_descriptions
/camera/image_raw/compressed/parameter_updates
/camera/image_raw/compressedDepth
/camera/image_raw/compressedDepth/parameter_descriptions
/camera/image_raw/compressedDepth/parameter_updates
/camera/image_raw/theora
/camera/image_raw/theora/parameter_descriptions
/camera/image_raw/theora/parameter_updates
/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
/gripper_control_controller/command

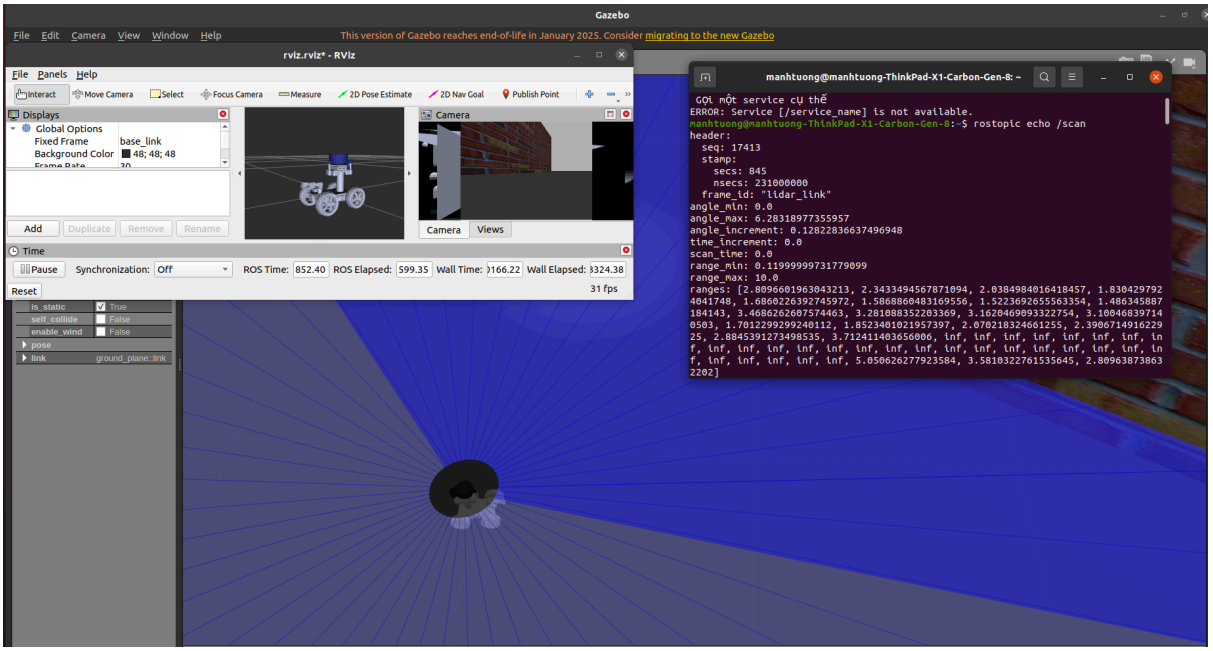

manhtuong@manhtuong-ThinkPad-X1-Carbon-Gen-8: ~
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/performance_metrics
/gazebo/set_link_state
/gazebo/set_model_state
/gripper_control_controller/command
/gripper_left_controller/command
/initialpose
/joint_states
/link1_joint_controller/command
/link2_joint_controller/command
/move_base_simple/goal
/my_camera/parameter_descriptions
/my_camera/parameter_updates
/odom
/rosout
/rosout_agg
/scan
/tf
/tf_static

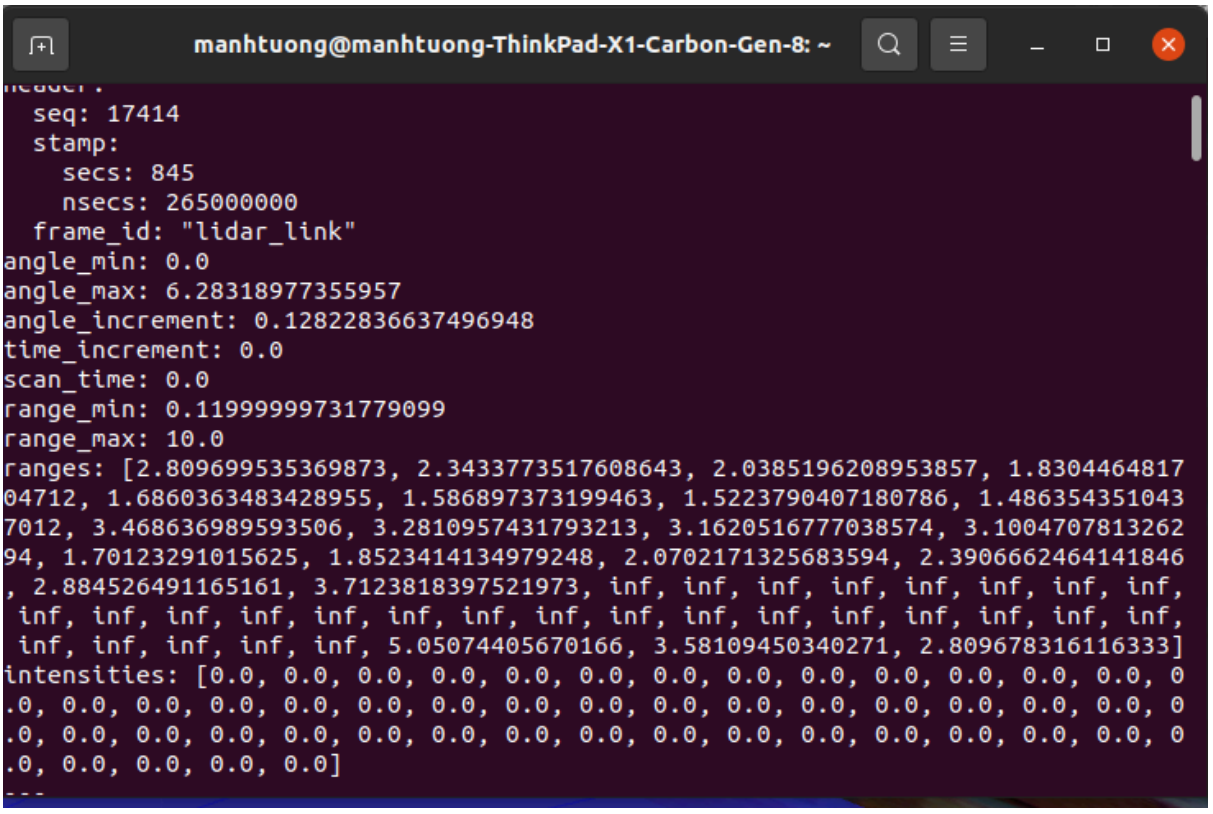

Gazebo
File Edit Camera View Window Help
This version of Gazebo reaches end-of-life in January 2025. Consider migrating to the new Gazebo
rviz.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Global Options
Fixed Frame base_link
Background Color 48; 48; 48
Add Duplicate Remove Rename
Camera
Camera Views
Time
Pause Synchronization: Off ROS Time: 811.11 ROS Elapsed: 558.07 Wall Time: 926.66 Wall Elapsed: 3084.76
Reset 31 fps
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
manhtuong@manhtuong-ThinkPad-X1-Carbon-Gen-8: ~
manhtuong@manhtuong-ThinkPad-X...
/camera/image_raw/compressed/set_parameters
/camera/image_raw/compressedDepth/set_parameters
/camera/image_raw/theora/set_parameters
/controller_manager/list_controller_types
/controller_manager/list_controllers
/controller_manager/load_controller
/controller_manager/reload_controller_libraries
/controller_manager/switch_controller
/controller_manager/unload_controller
/fake_joint_calibration/get_loggers
/fake_joint_calibration/set_logger_level
/fake_joint_calibration_manhtuong_ThinkPad_X1_Carbon_Gen_8_25046_368663097779741
3219/get_loggers
/fake_joint_calibration_manhtuong_ThinkPad_X1_Carbon_Gen_8_25046_368663097779741
3219/set_logger_level
/gazebo/apply_body_wrench
/gazebo/apply_joint_effort
/gazebo/clear_body_wrenches
/gazebo/clear_joint_forces
/gazebo/delete_light
/gazebo/delete_model
/gazebo/get_joint_properties
/gazebo/get_light_properties
/gazebo/get_link_properties


Gazebo
File Edit Camera View Window Help
This version of Gazebo reaches end-of-life in January 2025. Consider migrating to the new Gazebo
rviz.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Global Options
Fixed Frame base_link
Background Color 48; 48; 48
Add Duplicate Remove Rename
Camera
Camera Views
Time
Pause Synchronization: Off ROS Time: 820.09 ROS Elapsed: 567.04 Wall Time: 974.89 Wall Elapsed: 3133.02
Reset 31 fps
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
manhtuong@manhtuong-ThinkPad-X1-Carbon-Gen-8: ~
manhtuong@manhtuong-ThinkPad-X...
/gazebo/get_link_state
/gazebo/get_loggers
/gazebo/get_model_properties
/gazebo/get_model_state
/gazebo/get_physics_properties
/gazebo/get_world_properties
/gazebo/pause_physics
/gazebo/reset_simulation
/gazebo/reset_world
/gazebo/set_joint_properties
/gazebo/set_light_properties
/gazebo/set_link_properties
/gazebo/set_link_state
/gazebo/set_logger_level
/gazebo/set_model_configuration
/gazebo/set_model_state
/gazebo/set_parameters
/gazebo/set_physics_properties
/gazebo/spawn_sdf_model
/gazebo/spawn_urdf_model
/gazebo/unpause_physics
/gazebo_gui/get_loggers
/gazebo_gui/set_logger_level
/joint_state_publisher_gui/get_loggers


manhtuong@manhtuong-ThinkPad-X1-Carbon-Gen-8: ~
manhtuong@manhtuong-ThinkPad-X...
manhtuong@manhtuong-ThinkPad-X...
/joint_state_publisher_gui/get_loggers
/joint_state_publisher_gui/set_logger_level
/my_camera/set_camera_info
/my_camera/set_parameters
/robot_state_publisher/get_loggers
/robot_state_publisher/set_logger_level
/rosout/get_loggers
/rosout/set_logger_level
/rostopic_29226_1743479279924/get_loggers
/rostopic_29226_1743479279924/set_logger_level
/rostopic_29755_1743479735042/get_loggers
/rostopic_29755_1743479735042/set_logger_level
/rviz/get_loggers
/rviz/load_config
/rviz/load_config_discarding_changes
/rviz/reload_shaders
/rviz/save_config
/rviz/set_logger_level
/tf_footprint_base/get_loggers
/tf_footprint_base/set_logger_level
/two_dof_arm_controller/get_loggers
/two_dof_arm_controller/set_logger_level
/two_dof_arm_controller_spawner/get_loggers
/two_dof_arm_controller_spawner/set_logger_level


Gazebo
File Edit Camera View Window Help
This version of Gazebo reaches end-of-life in January 2025. Consider migrating to the new Gazebo
rviz.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Global Options
Fixed Frame base_link
Background Color 48; 48; 48
Add Duplicate Remove Rename
Camera
Camera Views
Time
Pause Synchronization: Off ROS Time: 823.55 ROS Elapsed: 570.51 Wall Time: 993.67 Wall Elapsed: 3151.84
Reset 31 fps
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
manhtuong@manhtuong-ThinkPad-X1-Carbon-Gen-8: ~
/joint_state_publisher_gui/set_logger_level
/my_camera/set_camera_info
/my_camera/set_parameters
/robot_state_publisher/get_loggers
/robot_state_publisher/set_logger_level
/rosout/get_loggers
/rosout/set_logger_level
/rostopic_29226_1743479279924/get_loggers
/rostopic_29226_1743479279924/set_logger_level
/rostopic_29755_1743479735042/get_loggers
/rostopic_29755_1743479735042/set_logger_level
/rviz/get_loggers
/rviz/load_config
/rviz/load_config_discarding_changes
/rviz/reload_shaders
/rviz/save_config
/rviz/set_logger_level
/tf_footprint_base/get_loggers
/tf_footprint_base/set_logger_level
/two_dof_arm_controller/get_loggers
/two_dof_arm_controller/set_logger_level
/two_dof_arm_controller_spawner/get_loggers
/two_dof_arm_controller_spawner/set_logger_level





1. Cấu trúc và ý nghĩa dữ liệu

Dữ liệu từ /scan bao gồm nhiều thông tin quan trọng:

Thông tin thời gian và nguồn gốc dữ liệu: Mỗi gói tin có một dấu thời gian giúp hệ thống biết chính xác khi nào dữ liệu được thu

thập. Nó cũng có một mã số thứ tự (sequence) để đánh dấu thứ tự của từng gói dữ liệu.

Phạm vi quét: LiDAR hoạt động theo nguyên tắc quét tia laser theo một góc nhất định. Ví dụ, nếu góc quét từ 0° đến 360°, thì dữ liệu sẽ bao gồm khoảng cách đo được tại từng góc trong phạm vi đó. Mỗi lần quét, LiDAR sẽ chia góc thành các phần nhỏ (ví dụ, mỗi lần đo cách nhau 1°).

Danh sách khoảng cách: Dữ liệu quan trọng nhất là danh sách các khoảng cách đo được tại mỗi góc quét. Ví dụ, nếu LiDAR quét từ trái sang phải, nó sẽ ghi nhận khoảng cách của vật thể từ gần đến xa theo từng góc quét. Nếu một điểm có giá trị 0.5m, nghĩa là có một vật thể ở khoảng cách 0.5m tại góc đó.

Cường độ phản xạ: Một số LiDAR còn cung cấp thông tin về cường độ phản xạ của tia laser. Nếu một bề mặt phản xạ mạnh (như kim loại), giá trị này sẽ cao, còn nếu bề mặt hấp thụ ánh sáng (như vải), giá trị này sẽ thấp.

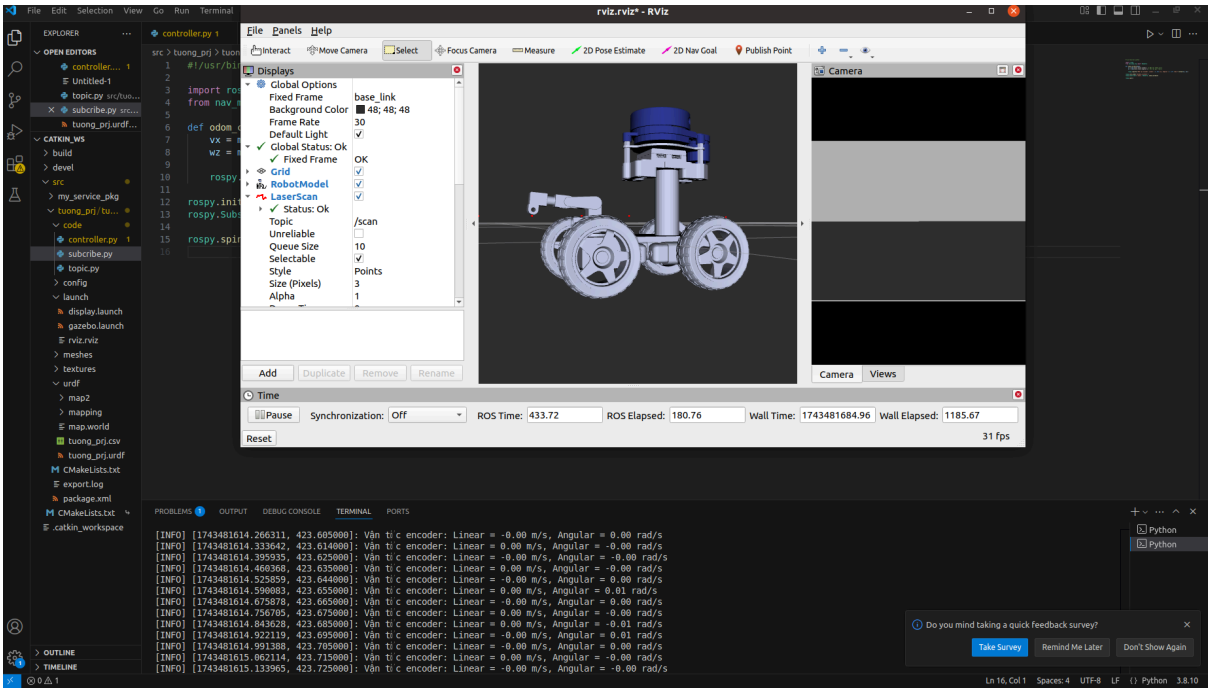
2. Ứng dụng của dữ liệu `/scan`

Dữ liệu này có thể được sử dụng trong nhiều ứng dụng thực tế:

Nhận diện vật thể phía trước: Nếu một số giá trị trong danh sách khoảng cách trở nên rất nhỏ (gần với khoảng cách tối thiểu mà cảm biến có thể đo), có nghĩa là có một vật thể rất gần với robot.

Điều hướng tránh vật cản: Robot có thể sử dụng dữ liệu để xác định đâu là hướng có thể di chuyển an toàn. Nếu một bên có nhiều điểm đo có giá trị nhỏ, robot sẽ tránh hướng đó.

Xây dựng bản đồ môi trường: Dữ liệu quét laser từ `/scan` có thể được kết hợp với các thuật toán như SLAM (Simultaneous Localization and Mapping) để tạo bản đồ môi trường xung quanh.



Cách thức hoạt động của encoder trong mô phỏng

Encoder trong mô phỏng hoạt động bằng cách lấy dữ liệu từ hệ thống động lực học của Gazebo. Khi bánh xe quay hoặc một khớp nào đó di chuyển, phần mềm sẽ tính toán và ghi lại các giá trị tương ứng, sau đó xuất ra dữ liệu thông qua các topic ROS như `/joint_states` hoặc `/odom`.

Ví dụ, nếu một bánh xe quay với vận tốc 1 rad/s, dữ liệu thu được sẽ chứa:

Vị trí góc của bánh xe (góc quay tích lũy theo thời gian).

Tốc độ góc (rad/s).