

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



Môn học: Lập trình robot với ROS- RBE3017_1

Họ và tên:

Nguyễn Bảo Long-22027537

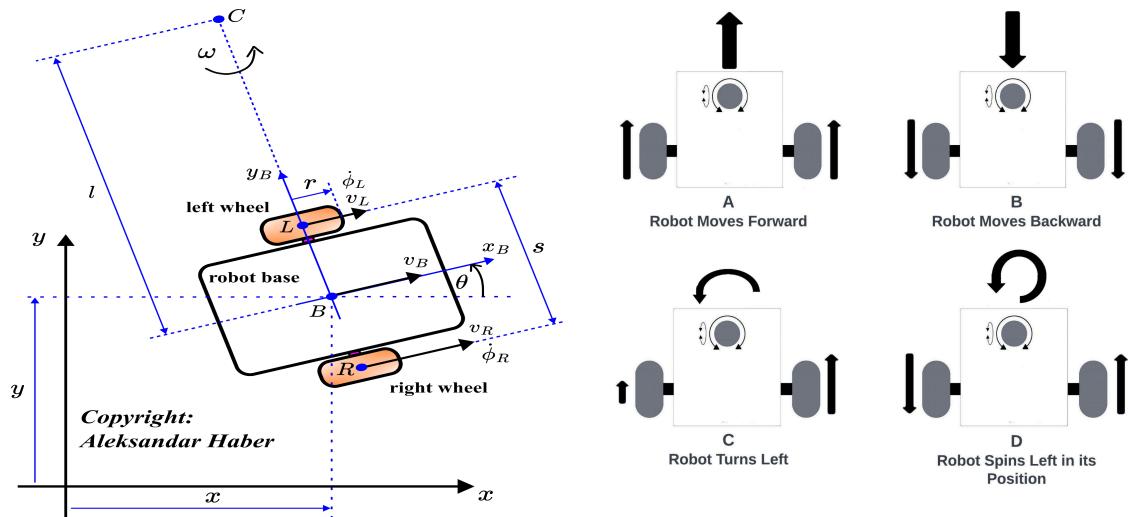
**BÁO CÁO ROS GIỮA KÌ: XÂY DỰNG, MÔ PHỎNG VÀ ĐIỀU
KHIỂN ROBOT**

I. Thiết kế robot

- Dựa theo những yêu cầu sau:
 - Differential drive
 - Tay máy 1 : Rotation
 - Tay máy 2: Rotation
 - Cảm biến hoạt động: Camera, Lidar, GPS

=> Mục tiêu dự án: Thiết kế một robot differential drive 2 bánh vi sai có khả năng di chuyển linh hoạt trong môi trường giả lập Gazebo. Robot cần tích hợp cả cảm biến bao gồm Camera, Lidar, GPS và có hệ thống tay máy để thực hiện các nhiệm vụ mô phỏng.

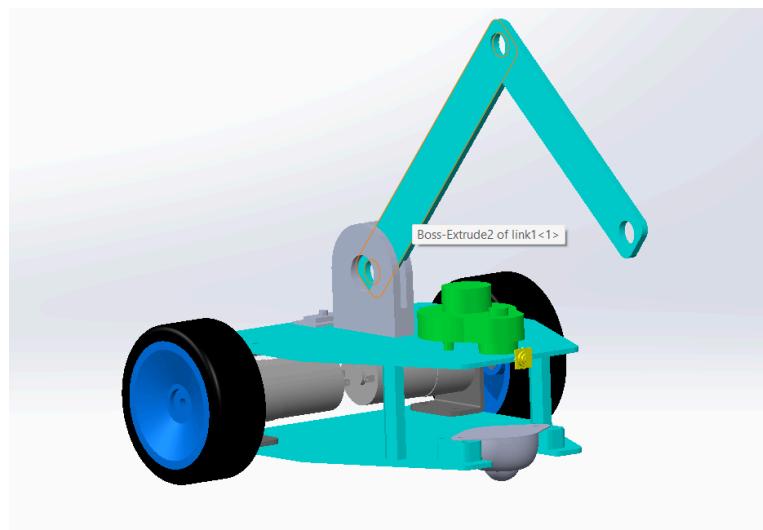
- Dạng robot 2 bánh vi sai:
- Sử dụng 2 bánh xe độc lập, mỗi bánh được điều khiển bởi động cơ riêng , thường kết hợp với bánh xe tự do để giữ thăng bằng. Sự khác biệt về tốc độ giữa 2 bánh xe cho phép robot di chuyển và xoay linh hoạt
- Động học robot 2 bánh vi sai:



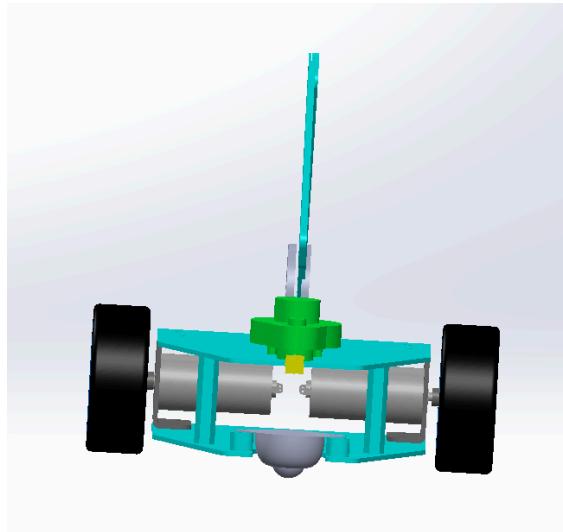
- Khi 2 bánh xe quay cùng tốc độ và cùng hướng, robot sẽ di chuyển tiến hoặc lùi theo đường thẳng
- Khi 2 bánh xe quay với tốc độ bằng nhau nhưng ngược chiều, robot sẽ xoay quanh trục trung tâm của nó
- Khi 2 bánh xe quay cùng hướng nhưng với tốc độ khác nhau, robot sẽ di chuyển theo một cung tròn. Bán kính vòng xoay phụ thuộc vào mức độ chênh lệch tốc độ giữa 2 bánh xe

=> Dựa vào các yêu cầu trên thiết kế robot bằng phần mềm Solidworks như sau

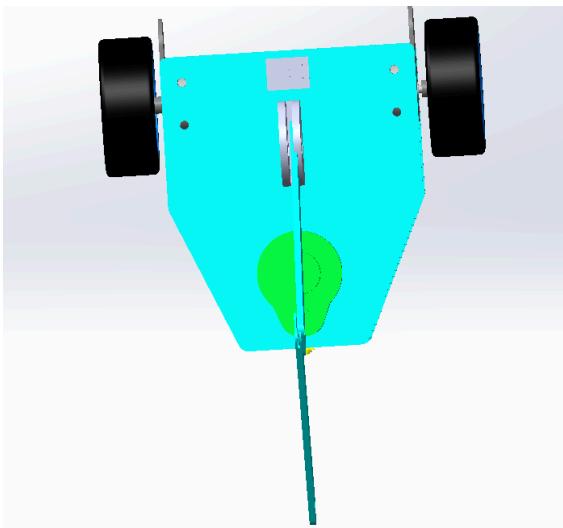
- Tổng quan mô hình xe hoàn chỉnh



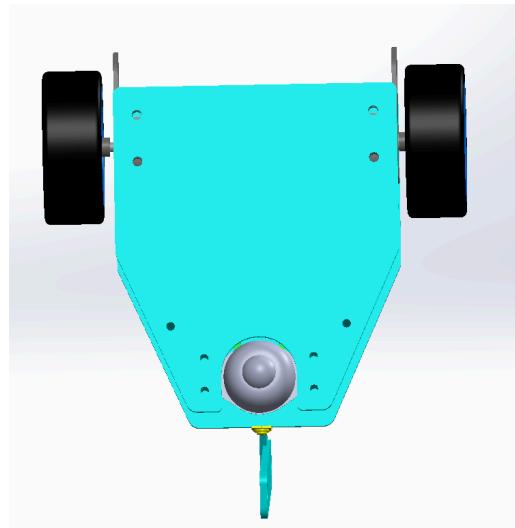
- Mặt trước robot



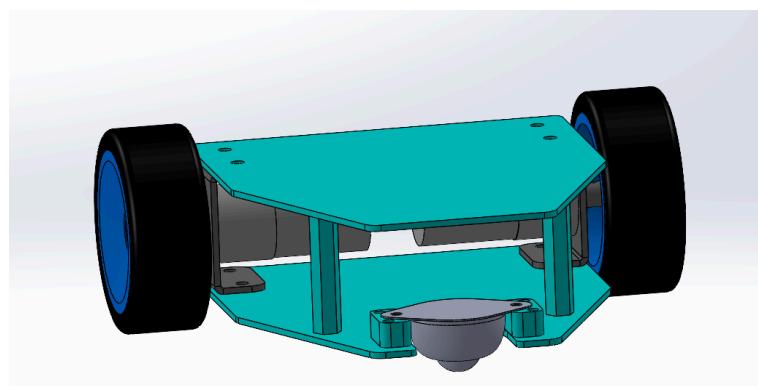
- Mặt trên robot



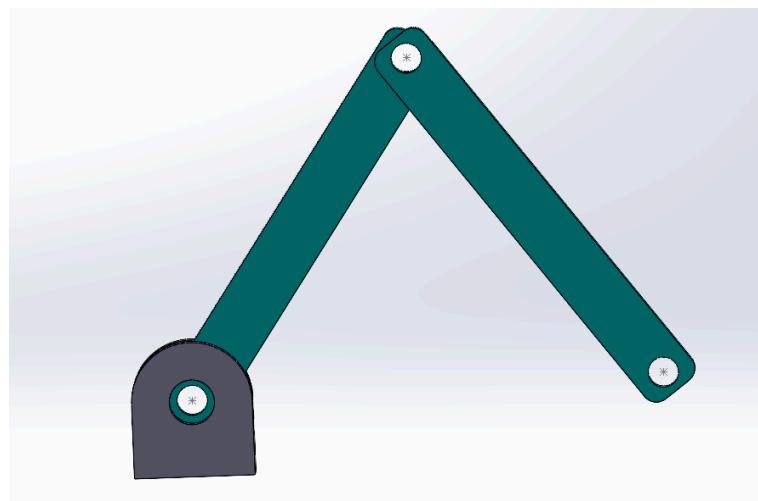
- Mặt dưới robot



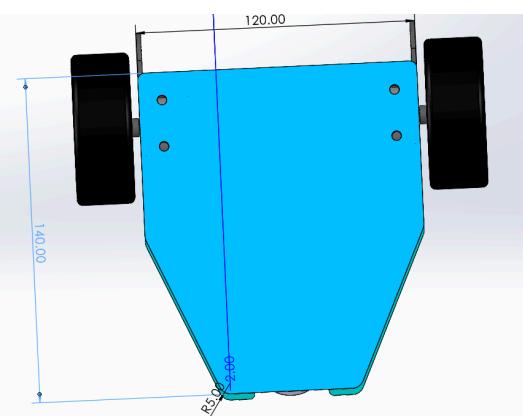
- Khung xe chính

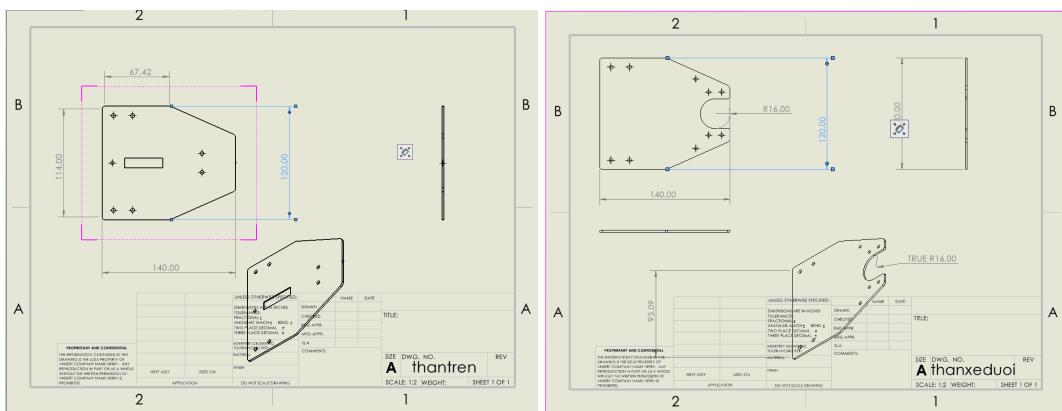


- + 2 bánh đơn : Gắn liền ở 2 bên giúp robot di chuyển
- + Thân trên + thân dưới: Khung chứa toàn bộ hệ thống cảm biến, khớp trục,..
- + Bánh tự do (Caster wheel)
- Tay máy

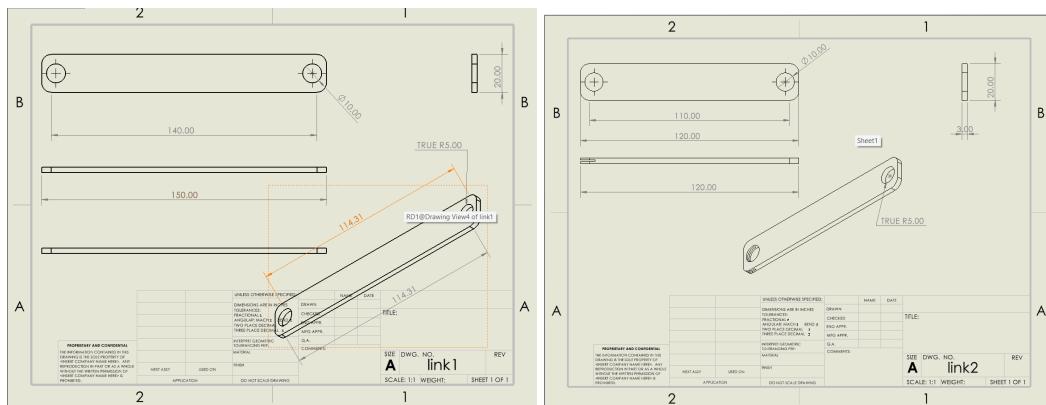
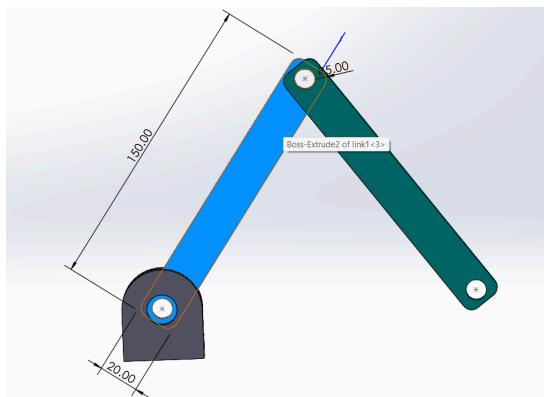


- + Gồm 1 khớp nối cố định với thân trên
- + 2 khớp quay
- Kích thước khung xe

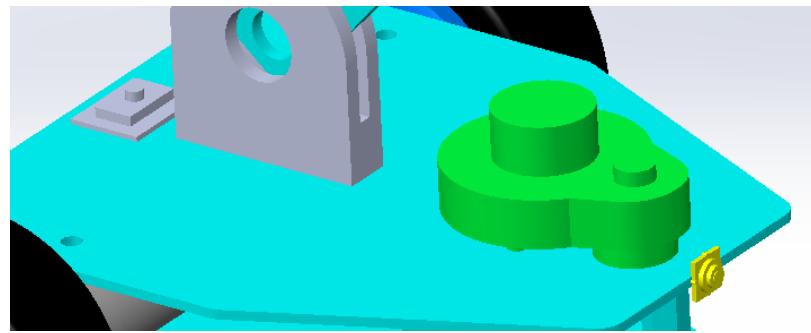




- Kích thước tay máy



- Các cảm biến đi kèm: (Lidar, Camera, GPS)
- Camera : Hỗ trợ quan sát, nhận diện đối tượng và theo dõi môi trường xung quanh
- Lidar: quét Laser để tạo bản đồ 2D,3D trong các môi trường
- GPS: Xác định vị trí toàn cầu của xe, kết hợp với robot để biết được mình đang ở đâu



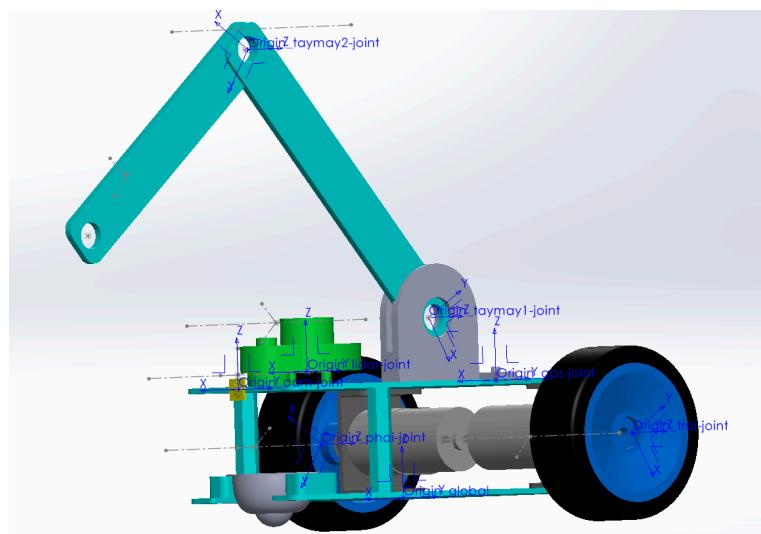
- Cách đặt hệ trục tọa độ

- Mô hình solidworks khi xuất sang file URDF được xác định như sau với hệ trục tọa độ tương ứng:

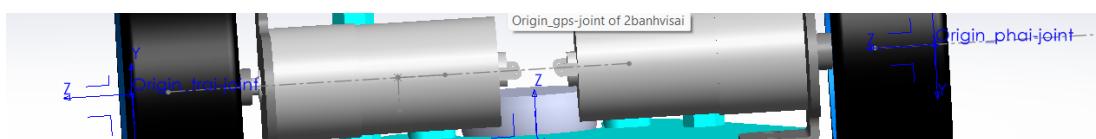
Trục X: Hướng về phía trước robot

Trục Y: Hướng sang trái hoặc phải của robot

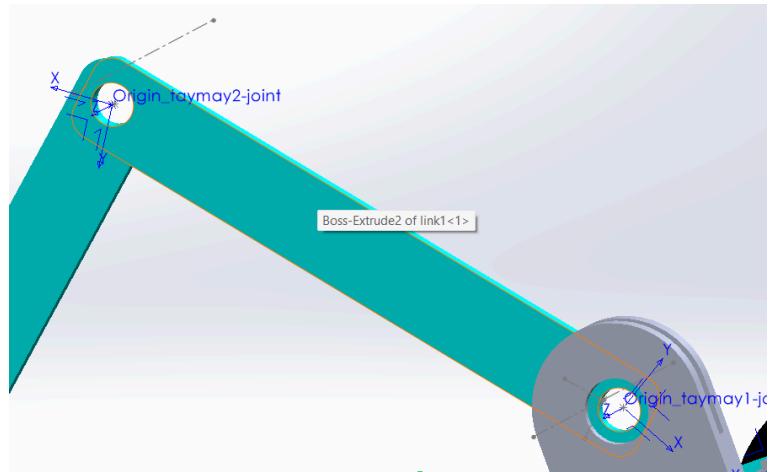
Trục Z: Hướng lên hoặc xuống



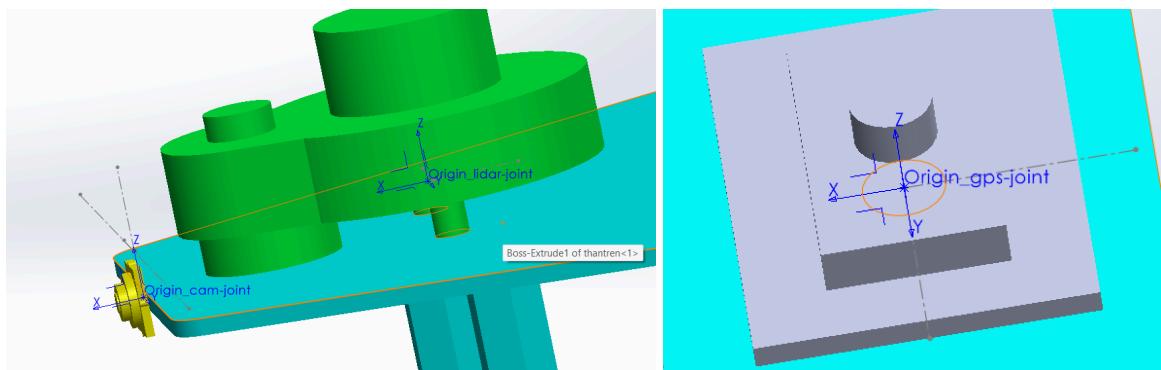
- + Origin Global: hệ tọa độ cố định trong không gian, đặt tại tâm của thân xe robot. Trục X hướng về phía trước, trục Z hướng lên trên và trục Y hướng sang bên hông
- + Tại 2 bánh xe trái và phải, đặt trục tọa độ tại tâm của 2 bánh xe sao cho quay quanh trục Z



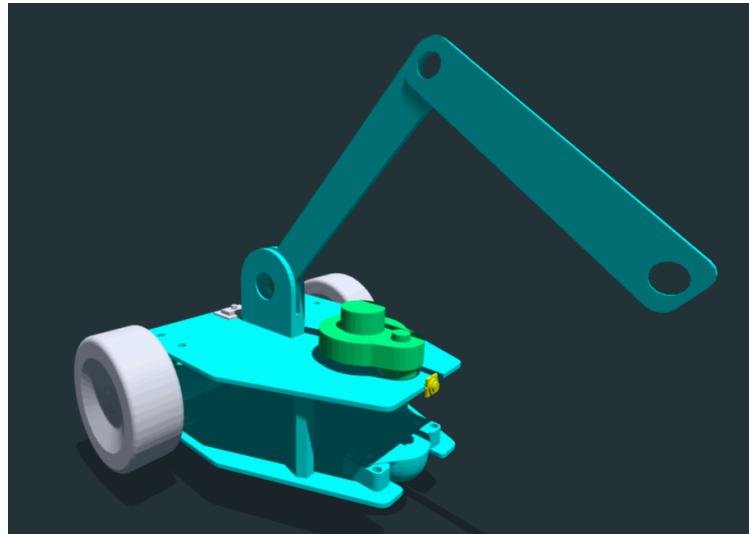
- + Tại tay máy 1 và tay máy 2 đặt tại các khớp của cánh tay robot với trục tọa độ điều chỉnh theo hướng của cánh tay sao cho quay quanh trục Z



- + Tại các cảm biến cố định, đặt hệ trục tọa độ tại tâm của các cảm biến và chiều X Y Z như hệ tọa độ cố định của thân xe robot



- Tùy chỉnh các thông số khi export to URDF như : dạng bánh xe (continuous), tay máy (revolute), cảm biến (fixed), các thông số của link và bánh xe
=> Xuất sang file URDF và check trên link git



đã xây dựng thành công

II. Xây dựng, mô phỏng và điều khiển robot trên phần mềm gazebo

- Cấu trúc file, cấu trúc thư mục

Mục	Mô tả	Thời gian
config	update_project	6 hours ago
launch	update_project	6 hours ago
meshes	update_project	6 hours ago
rviz	update_rviz	3 hours ago
scripts	update_project	6 hours ago
urdf	update_project	6 hours ago
CMakeLists.txt	update_project	6 hours ago
README.md	Initial commit	6 hours ago
export.log	update_project	6 hours ago
package.xml	update_project	6 hours ago

Cấu trúc thư mục và các file liên quan

- config/: Chứa các file cấu hình (.yaml) các thông số như PID,..
- launch/: Chứa các file (.launch) để khởi chạy mô phỏng gazebo, rviz, các node điều khiển
- meshes/: Chứa các file (.STL) mô hình 3D của robot
- rviz/: Chứa các file (.rviz) cấu hình config cho mtrg rviz
- scripts/: Chứa các file (.py) dùng để điều khiển topic của robot
- urdf/: Chứa các file mô tả robot URDF/XACRO
- Cmakelist.txt & package.xml : Chứa các file cần thiết để build package trong ROS

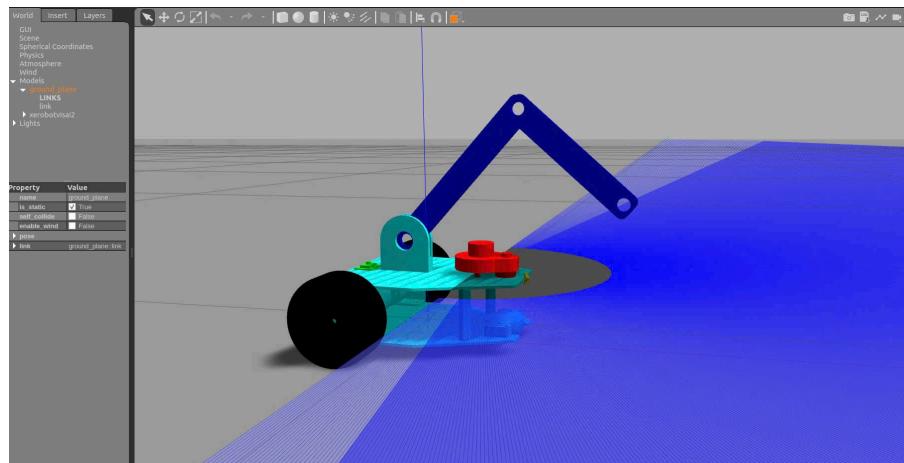
- Sau khi hoàn thành URDF, mô phỏng robot trong gazebo
- Mô tả file URDF sau khi xuất từ bên solidworks
 - + Mô tả cấu trúc của robot bao gồm các liên kết (base_link; trai_link; phai_link; taymay1_link; taymay2_link;...) và các khớp (joints) tương ứng
 - + Ngoài ra mô tả giới hạn giữa các khớp (limit); thông số động học (inertial) ; trục quay (axis); damping và friction (dynamics)
- Chính sửa file gazebo.launch để mở 2 môi trường và các cảm biến

```

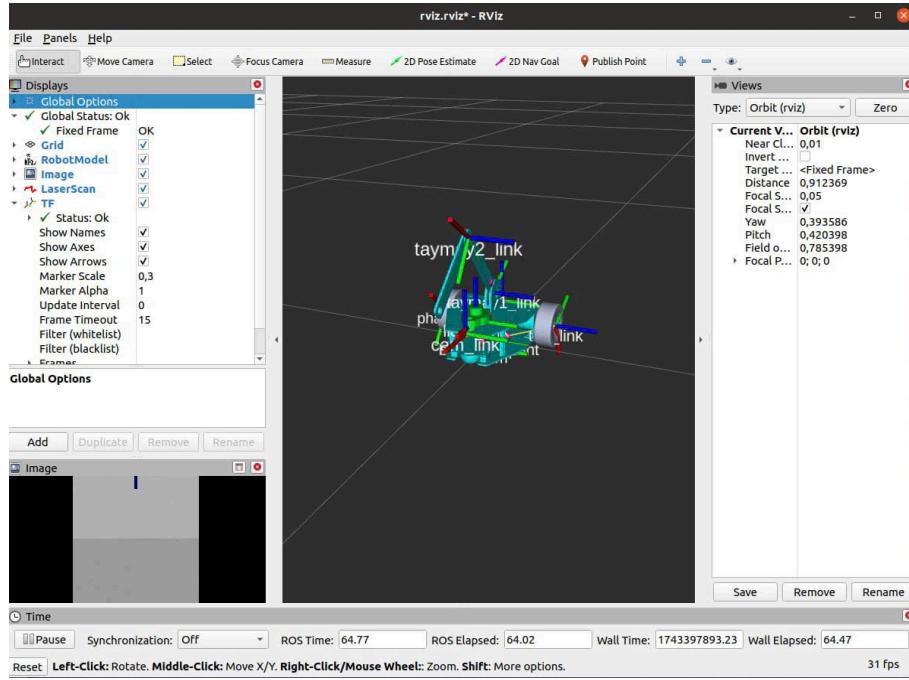
launch > gazebo.launch
1  <launch>
2   <include file="$(find gazebo_ros)/launch/empty_world.launch" />
3   <param name="robot_description" command="$(find xacro)/xacro $(find xerobotvisai2)/urdf/xerobotvisai2.urdf" />
4   <param name="use_sim_time" value="true"/>
5   <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" />
6   <node name="tf_footprint_base" pkg="tf" type="static_transform_publisher" args="#0 0 0 0 base_link base_footprint 40" />
7   <node name="spawn_model" pkg="gazebo_ros" type="spawn_model" args="--param robot_description:=urdf -model xerobotvisai2" output="screen" />
8   <node name="fake_joint_calibration" pkg="rostopic" type="rostopic" args="/pub /calibrated std_msgs/Bool true" />
9   <rosparam file="$(find xerobotvisai2)/config/arm.yaml" command="load"/>
10  <node name="controller_spawner" pkg="controller_manager" type="spawner" args="joint_state_controller taymay1_joint_position_controller taymay2_joint_posi
11  <node name="joint_state_publisher_gui" pkg="joint_state_publisher_gui" type="joint_state_publisher_gui" />
12  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find xerobotvisai2)/rviz/rviz.rviz" required="false" launch-prefix="bash -c 'sleep 3; $0 $@'" />
13 </launch>

```

- Chạy file gazebo.launch để hiển thị robot trong môi trường gazebo và rviz



hiển thị thành công robot model trong gazebo



hiển thị thành công robot model trong rviz và các cảm biến

- Gõ lệnh rostopic list để kiểm tra các topic trong gazebo

```
/home/longnb/catkin_ws/src/xerobotvisai2/launch/gazebo.launch
rlongnb@longnb:~$ rostopic list
/calibrated
/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
/gps/fix
/gps/fix/position/parameter_descriptions
/gps/fix/position/parameter_updates
/gps/fix/status/parameter_descriptions
/gps/fix/status/parameter_updates
/gps/fix/velocity/parameter_descriptions
/gps/fix_velocity
/initialpose
/joint_states
/move_base_simple/goal
/odom
/rosout
/rosout_agg
/rrbot/camera1/camera_info
/rrbot/camera1/image_raw
/rrbot/camera1/image_raw/compressed
/rrbot/camera1/image_raw/compressed/parameter_descriptions
/rrbot/camera1/image_raw/compressed/parameter_updates
/rrbot/camera1/image_raw/compressedDepth
/rrbot/camera1/image_raw/compressedDepth/parameter_descriptions
/rrbot/camera1/image_raw/compressedDepth/parameter_updates
/rrbot/camera1/image_raw/theora
/rrbot/camera1/image_raw/theora/parameter_descriptions
/rrbot/camera1/image_raw/theora/parameter_updates
/rrbot/camera1/parameter_descriptions
/rrbot/camera1/parameter_updates
/scan
/taymay1_joint_position_controller/command
/taymay2_joint_position_controller/command
/tf
/tf_static
Longnb@longnb:~$
```

Hiển thị thành công các topic điều khiển bánh xe, tay máy, lidar; camera và gps

- Thêm các plugin để điều khiển robot với các cảm biến tương ứng
- Tích hợp plugin camera: Cho phép robot quan sát môi trường xung quanh trong mô phỏng

```

<gazebo reference="cam_link">
  <sensor type="camera" name="camera1">
    <update_rate>30.0</update_rate>
    <camera name="head">
      <horizontal_fov>1.3962634</horizontal_fov>
      <image>
        <width>800</width>
        <height>800</height>
        <format>R8G8B8</format>
      </image>
      <clip>
        <near>0.02</near>
        <far>300</far>
      </clip>
      <noise>
        <type>gaussian</type>
        <mean>0.0</mean>
        <stddev>0.007</stddev>
      </noise>
    </camera>
    <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
      <alwaysOn>true</alwaysOn>
      <updateRate>0.0</updateRate>
      <cameraName>rrbot/camera1</cameraName>
      <imageTopicName>image_raw</imageTopicName>
      <cameraInfoTopicName>camera_info</cameraInfoTopicName>
      <frameName>camera_link</frameName>
      <hackBaseline>0.07</hackBaseline>
      <distortionK1>0.0</distortionK1>
      <distortionK2>0.0</distortionK2>
      <distortionK3>0.0</distortionK3>
      <distortionT1>0.0</distortionT1>
      <distortionT2>0.0</distortionT2>
    </plugin>
  </sensor>
</gazebo>

```

Dựa vào cam_link từ solidworks, thêm plugin cho camera

- Tích hợp plugin Lidar: Cho phép robot quét môi trường 3D

```

<gazebo reference="lidar_link">
  <sensor type="ray" name="lidar_sensor">
    <pose>0 0 0 0 0 0</pose>
    <visualize>true</visualize>
    <update_rate>30</update_rate>
    <ray>
      <scan>
        <horizontal>
          <samples>720</samples>
          <resolution>1</resolution>
          <min_angle>-1.5708</min_angle>
          <max_angle>1.5708</max_angle>
        </horizontal>
      </scan>
      <range>
        <min>0.1</min>
        <max>30.0</max>
        <resolution>0.01</resolution>
      </range>
      <noise>
        <type>gaussian</type>
        <mean>0.0</mean>
        <stddev>0.01</stddev>
      </noise>
    </ray>
    <plugin name="gazebo_ros_laser" filename="libgazebo_ros_laser.so">
      <topicName>/scan</topicName>
      <frameName>lidar_link</frameName>
    </plugin>
  </sensor>
</gazebo>

```

Dựa vào lidar_link bên solidworks, thêm plugin cho lidar

- Tích hợp plugin GPS: Cho phép robot xác định vị trí

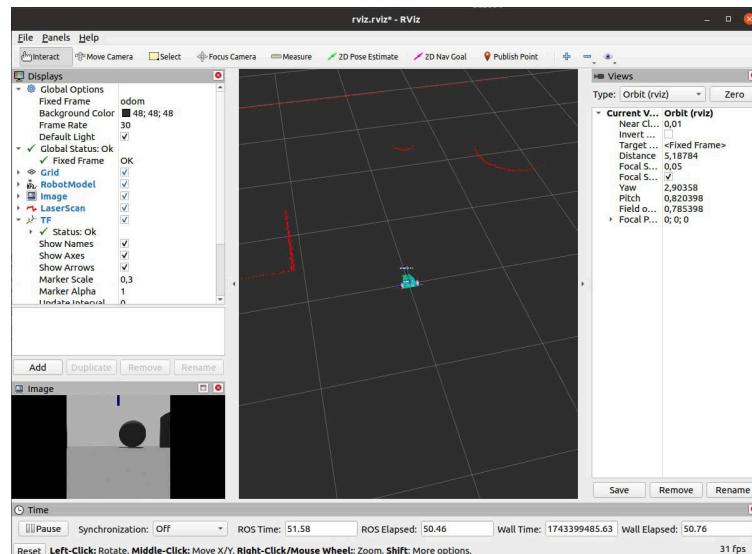
```

<gazebo>
  <plugin name="gazebo_ros_gps" filename="libhector_gazebo_ros_gps.so">
    <updateRate>4.0</updateRate>
    <bodyName>gps_link</bodyName>
    <frameId>gps_link</frameId>
    <topicName>/gps/fix</topicName>
    <velocityTopicName>/gps/fix_velocity</velocityTopicName>
    <referenceLatitude>-30.06022459407145675</referenceLatitude>
    <referenceLongitude>-51.173913575780311191</referenceLongitude>
    <referenceHeading>90</referenceHeading>
    <referenceAltitude>10</referenceAltitude>
    <offset>0 0 0</offset>
    <drift>0.001 0.001 0.001</drift>
    <gaussianNoise>0.05 0.05 0.05</gaussianNoise>
    <velocityDrift>0.001 0.001 0.001</velocityDrift>
    <velocityGaussianNoise>0.5 0.5 0.5</velocityGaussianNoise>
  </plugin>
</gazebo>

```

Dựa vào gps_link bên solidworks, thêm plugin cho GPS

- Sau khi add thành công các plugin, chạy file launch để hiển thị thành công các topic của cảm biến



lidar và camera hoạt động trên rviz

```

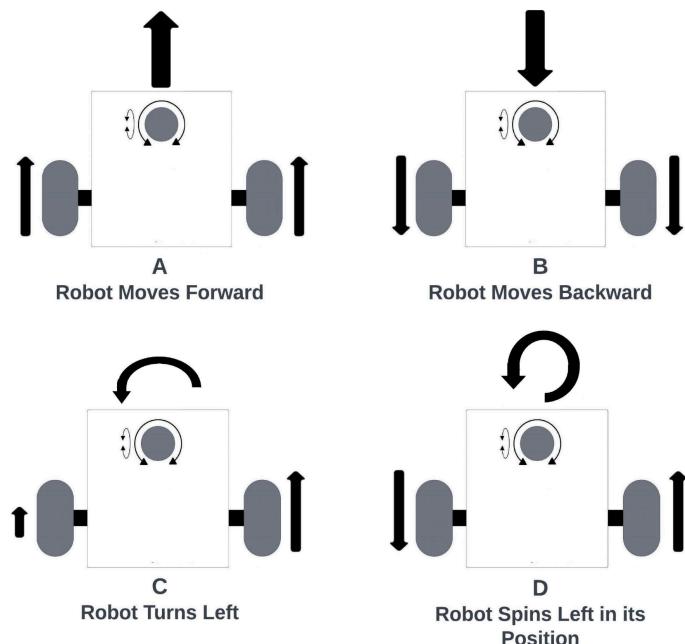
longitude: -51.17390449765464
latitude: -30.06022503967702
altitude: 9.98389601406678
position_covariance: [0.002501000000000006, 0.0, 0.0, 0.0, 0.002501000000000006]
6, 0.0, 0.0, 0.0, 0.002501000000000006]
position_covariance_type: 2
---
header:
  seq: 437
  stamp:
    secs: 109
    nsecs: 500000000
  frame_id: "gps_link"
status:
  status: 0
  service: 0
longitude: -51.17390495925893
latitude: -30.060224381377136
altitude: 9.941584272382386
position_covariance: [0.002501000000000006, 0.0, 0.0, 0.0, 0.002501000000000006]
6, 0.0, 0.0, 0.0, 0.002501000000000006]
position_covariance_type: 2
---

```

gps hoạt động

- **Điều khiển robot trong ROS**

- Cách di chuyển của robot 2 bánh vi sai



Minh họa di chuyển của 2 bánh vi sai

- Add plugin gazebo_ros_control:

```

<gazebo>
  <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
    <robotNamespace>/</robotNamespace>
    <robotSimType>gazebo_ros_control/DefaultRobotHWSim</robotSimType>
    <legacyModeNS>false</legacyModeNS>
    <controlPeriod>0.001</controlPeriod>
  </plugin>
</gazebo>
</robot>

```

- + Plugin cho phép Gazebo kết nối với ROS Control, cho phép sử dụng ros_control để điều khiển các joint của robot trong Gazebo thông qua các controller như:
 - **effort_controllers/JointEffortController**
 - **position_controllers/JointPositionController**
 - **velocity_controllers/JointVelocityController**
- + Các controller cho phép gửi các lệnh, tín hiệu lên các topic để điều khiển các joint bánh xe và tay máy dễ dàng hơn
- Add plugin differential-drive-controller để điều khiển xe robot

```

<gazebo>
  <plugin name="differential_drive_controller" filename="libgazebo_ros_diff_drive.so">
    <updateRate>50</updateRate>
    <leftJoint>trai_joint</leftJoint>
    <rightJoint>phai_joint</rightJoint>
    <wheelSeparation>0.5380</wheelSeparation>
    <wheelDiameter>0.2410</wheelDiameter>
    <wheelAcceleration>1.0</wheelAcceleration>
    <wheelTorque>20</wheelTorque>
    <commandTopic>cmd_vel</commandTopic>
    <odometryTopic>odom</odometryTopic>
    <odometryFrame>odom</odometryFrame>
    <robotBaseFrame>base_link</robotBaseFrame>
    <odometrySource>1</odometrySource>
    <publishWheelTF>true</publishWheelTF>
    <publishOdom>true</publishOdom>
    <publishWheelJointState>true</publishWheelJointState>
    <legacyMode>false</legacyMode>
  </plugin>
</gazebo>

```

Plugin trên được sử dụng trong mô hình robot vi sai, giúp nó có thể nhận lệnh vận tốc từ ROS và xuất dữ liệu odometry để theo dõi vị trí của robot

- Plugin ROS transmission cho 2 tay máy

```

<transmission name="taymay1_joint_transmission">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="taymay1_joint">
    <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
  </joint>
  <actuator name="link1_servo">
    <mechanicalReduction>1.0</mechanicalReduction>
  </actuator>
</transmission>

<transmission name="taymay2_joint_transmission">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="taymay2_joint">
    <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
  </joint>
  <actuator name="link2_servo">
    <mechanicalReduction>1.0</mechanicalReduction>
  </actuator>
</transmission>

```

định nghĩa các thành phần truyền động cho 2 khớp tay máy 1, 2

- + sử dụng giao diện điều khiển vị trí (Position Joint Interface) và cơ cấu truyền động đơn giản (Simple Transmission)
- File arm.yaml

```

config > ! arm.yaml
1 joint_state_controller:
2   type: joint_state_controller/JointStateController
3   publish_rate: 50
4
5 taymay1_joint_position_controller:
6   type: position_controllers/JointPositionController
7   joint: taymay1_joint
8   pid: {p: 3.0, i: 0.01, d: 0.1}
9   initial_position: 0.0
10
11 taymay2_joint_position_controller:
12   type: position_controllers/JointPositionController
13   joint: taymay2_joint
14   pid: {p: 3.0, i: 0.01, d: 0.1}
15   initial_position: 0.0
16

```

định nghĩa các bộ điều khiển cho các tay máy của robot

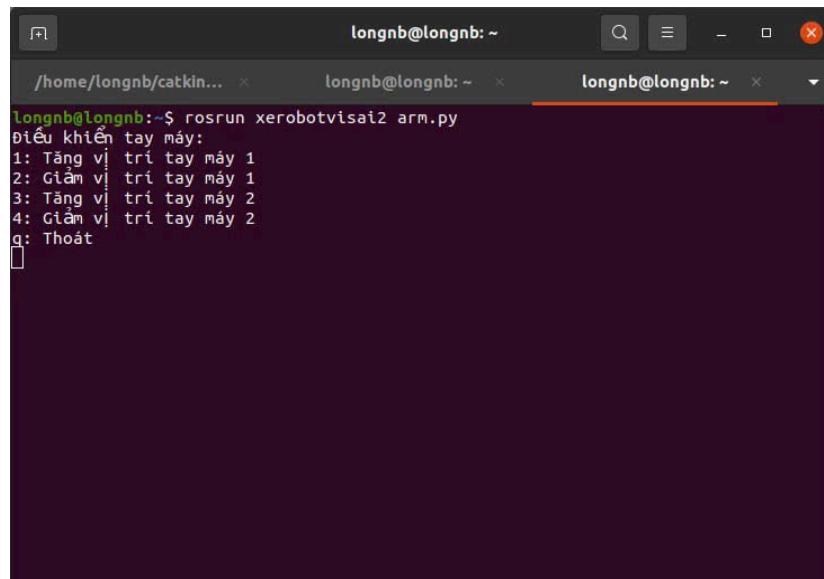
- + Kết hợp thêm bộ số PID để giảm thiểu sai số, độ giật và giúp tay máy di chuyển chính xác linh hoạt hơn
- File arm.py

```

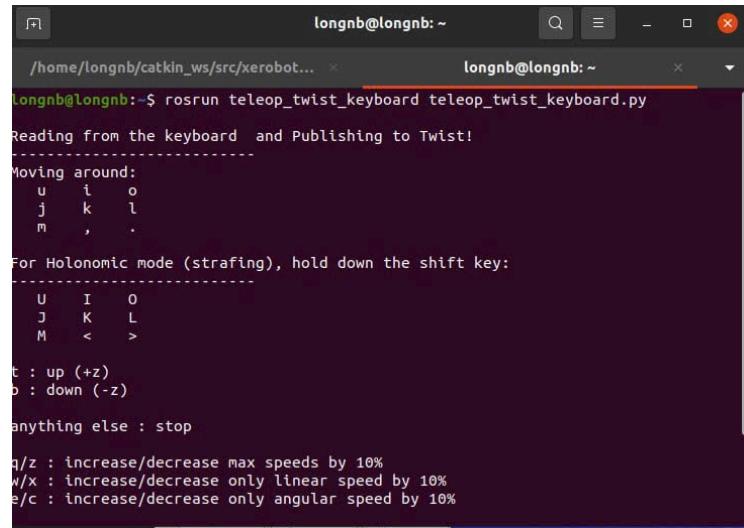
18     def arm_controller():
19         rospy.init_node('arm_controller', anonymous=True)
20         arm_1_pub = rospy.Publisher('/taymay1_joint_position_controller/command', Float64, queue_size=10)
21         arm_2_pub = rospy.Publisher('/taymay2_joint_position_controller/command', Float64, queue_size=10)
22         arm_1_pos = 0.0
23         arm_2_pos = 0.0
24         step = 0.5
25         print("Điều khiển tay máy:")
26         print("1: Tăng vị trí tay máy 1")
27         print("2: Giảm vị trí tay máy 1")
28         print("3: Tăng vị trí tay máy 2")
29         print("4: Giảm vị trí tay máy 2")
30         print("q: Thoát")
31         rate = rospy.Rate(50)
32         while not rospy.is_shutdown():
33             key = get_key()
34             if key == '1':
35                 arm_1_pos += step
36                 if arm_1_pos > 1.57:
37                     arm_1_pos = 1.57
38             elif key == '2':
39                 arm_1_pos -= step
40                 if arm_1_pos < -1.57:
41                     arm_1_pos = -1.57
42             elif key == '3':
43                 arm_2_pos += step
44                 if arm_2_pos > 1.57:
45                     arm_2_pos = 1.57
46             elif key == '4':
47                 arm_2_pos -= step
48                 if arm_2_pos < -1.57:
49                     arm_2_pos = -1.57
50             elif key == 'q':
51                 break
52             arm_1_pub.publish(arm_1_pos)
53             arm_2_pub.publish(arm_2_pos)
54             rospy.loginfo("Tay máy 1: %.2f rad, Tay máy 2: %.2f rad", arm_1_pos, arm_2_pos)
55             rate.sleep()

```

- + Khởi tạo các publisher là arm1_pub và arm2_pub
- + arm_pos lưu vị trí hiện tại của tay máy
- + step xác định mức tăng giảm tay máy khi nhấn phím điều khiển
- + Sử dụng bàn phím để điều khiển tăng giảm vị trí của tay máy 1 và tay máy 2
- + 1 : Tăng vị trí tay máy 1
- + 2: Giảm vị trí tay máy 1
- + 3: Tăng vị trí tay máy 2
- + 4: Giảm vị trí tay máy 2
- + Gửi dữ liệu và in thông tin ra



- Cài đặt gói teleop_twist_keyboard để điều khiển robot bằng bàn phím



```

longnb@longnb:~$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py
Reading from the keyboard and Publishing to Twist!
-----
Moving around:
 u   i   o
 j   k   l
 m   ,   .

For Holonomic mode (strafing), hold down the shift key:
-----
 U   I   O
 J   K   L
 M   <   >

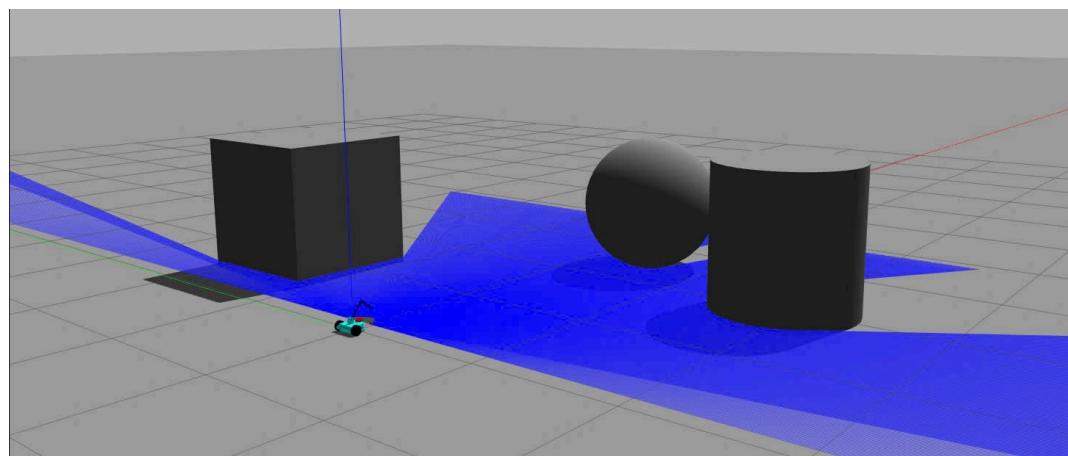
t : up (+z)
b : down (-z)

anything else : stop

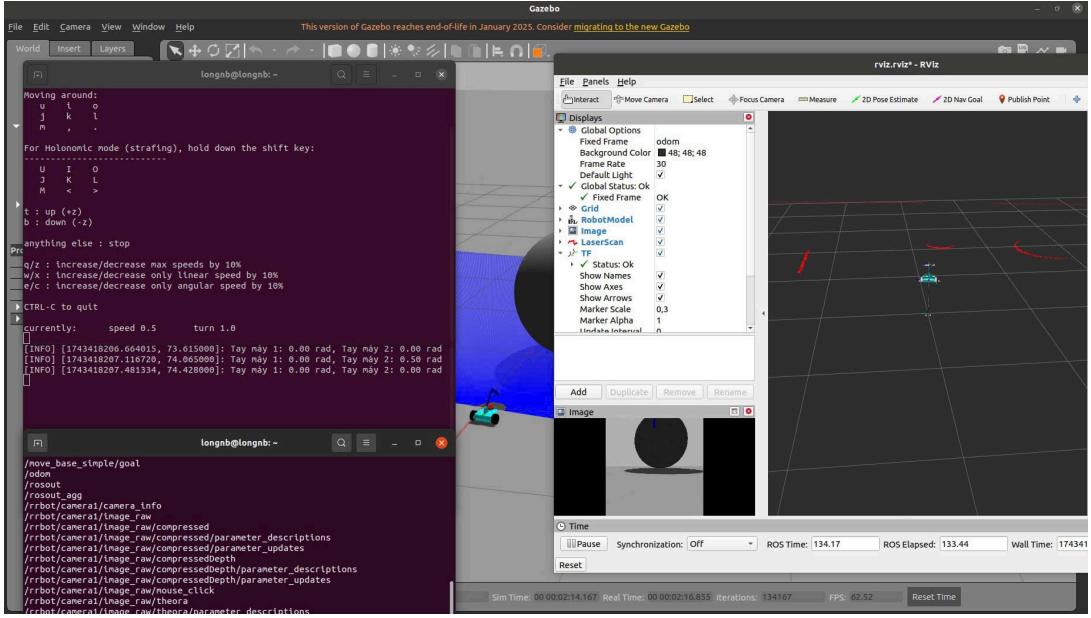
q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

```

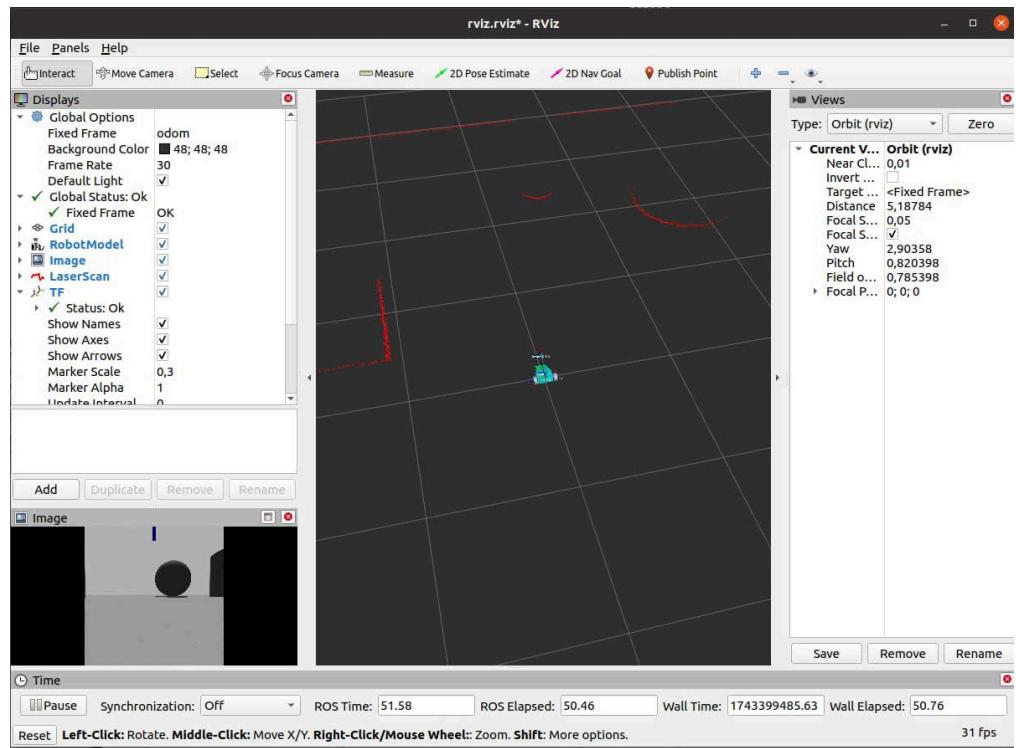
- Kết quả
- Sau khi khởi chạy gazebo.launch



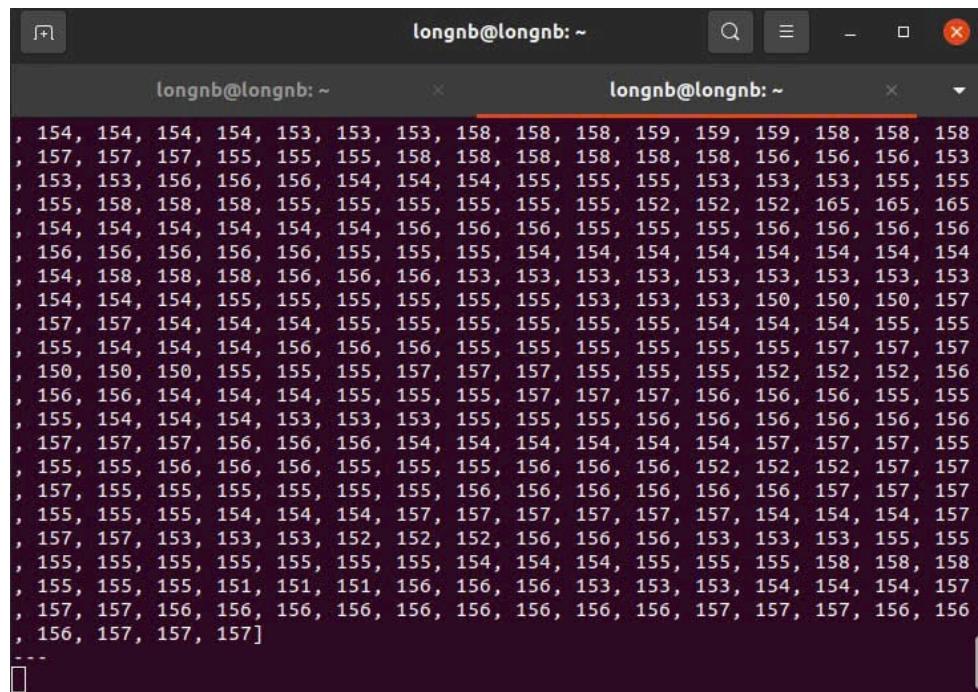
Add vật cản vào gazebo



- Rviz



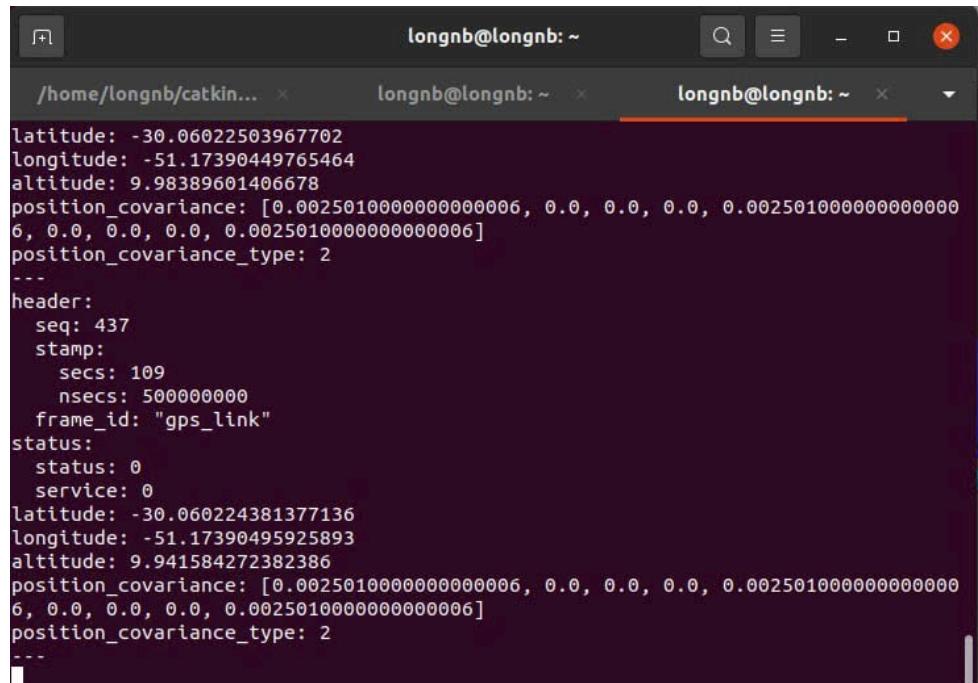
Topic của camera và lidar đã hiện thành công



```
longnb@longnb: ~
longnb@longnb: ~
, 154, 154, 154, 154, 153, 153, 153, 158, 158, 158, 158, 159, 159, 159, 158, 158, 158, 158
, 157, 157, 157, 155, 155, 155, 158, 158, 158, 158, 158, 158, 156, 156, 156, 156, 156, 153
, 153, 153, 156, 156, 156, 154, 154, 154, 155, 155, 155, 155, 153, 153, 153, 155, 155
, 155, 158, 158, 158, 155, 155, 155, 155, 155, 155, 155, 152, 152, 165, 165, 165
, 154, 154, 154, 154, 154, 154, 156, 156, 156, 155, 155, 155, 155, 156, 156, 156, 156
, 156, 156, 156, 156, 156, 155, 155, 154, 154, 154, 154, 154, 154, 154, 154, 154, 154
, 154, 158, 158, 158, 156, 156, 156, 153, 153, 153, 153, 153, 153, 153, 153, 153, 153
, 154, 154, 154, 155, 155, 155, 155, 155, 155, 153, 153, 153, 153, 150, 150, 150, 157
, 157, 157, 154, 154, 154, 155, 155, 155, 155, 155, 155, 154, 154, 154, 154, 155, 155
, 155, 154, 154, 154, 156, 156, 156, 155, 155, 155, 155, 155, 155, 157, 157, 157
, 150, 150, 150, 155, 155, 155, 157, 157, 157, 155, 155, 155, 152, 152, 152, 156
, 156, 156, 154, 154, 154, 155, 155, 155, 157, 157, 157, 156, 156, 156, 156, 155
, 155, 154, 154, 154, 153, 153, 153, 155, 155, 155, 155, 156, 156, 156, 156, 156
, 157, 157, 157, 156, 156, 156, 154, 154, 154, 154, 154, 154, 154, 157, 157, 157
, 155, 155, 156, 156, 156, 155, 155, 155, 156, 156, 156, 152, 152, 152, 157, 157
, 157, 155, 155, 155, 155, 155, 155, 156, 156, 156, 156, 156, 156, 156, 156, 157
, 155, 155, 155, 154, 154, 154, 157, 157, 157, 157, 157, 157, 154, 154, 154, 157
, 157, 157, 153, 153, 153, 152, 152, 152, 156, 156, 156, 153, 153, 153, 155, 155
, 155, 155, 155, 155, 155, 154, 154, 154, 154, 155, 155, 155, 158, 158, 158
, 155, 155, 155, 151, 151, 151, 151, 156, 156, 156, 153, 153, 153, 154, 154, 154, 157
, 157, 157, 156, 156, 156, 156, 156, 156, 156, 156, 156, 157, 157, 157, 156, 156
, 156, 157, 157, 157]
```

hiển thị dữ liệu camera

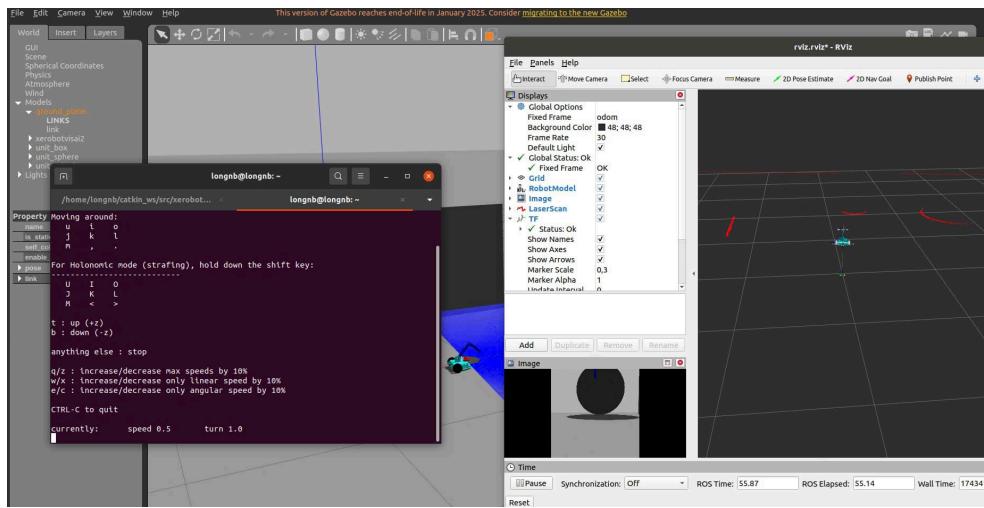
- Kiểm tra topic của GPS trong gazebo



```
longnb@longnb: ~
longnb@longnb: ~
latitude: -30.06022503967702
longitude: -51.17390449765464
altitude: 9.98389601406678
position_covariance: [0.002501000000000006, 0.0, 0.0, 0.0, 0.002501000000000006
6, 0.0, 0.0, 0.0, 0.002501000000000006]
position_covariance_type: 2
---
header:
  seq: 437
  stamp:
    secs: 109
    nsecs: 500000000
    frame_id: "gps_link"
status:
  status: 0
  service: 0
latitude: -30.060224381377136
longitude: -51.17390495925893
altitude: 9.941584272382386
position_covariance: [0.002501000000000006, 0.0, 0.0, 0.0, 0.002501000000000006
6, 0.0, 0.0, 0.0, 0.002501000000000006]
position_covariance_type: 2
---
```

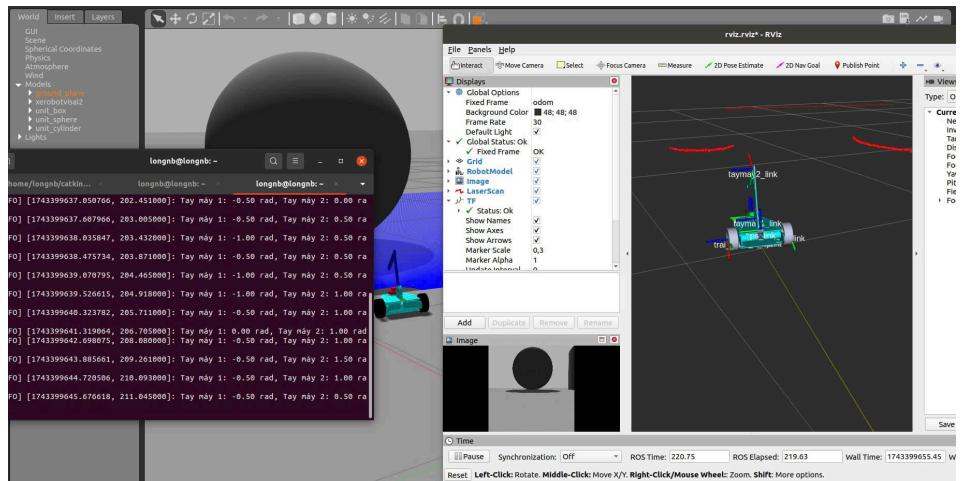
hiển thị dữ liệu gps

- Điều khiển xe robot bằng bàn phím



điều khiển robot thành công

- Điều khiển tay máy



điều khiển tay máy thành công

=> Robot di chuyển ổn định trong môi trường gazebo, thao tác với tay máy chính xác, linh hoạt và hiển thị đầy đủ các topic của cảm biến

video-ketqua