

Báo cáo giữa kỳ môn học lập trình ROS

họ và tên : Nguyễn Hữu Huy - msv : 22027534

giảng viên : TS. Lê Xuân Lực

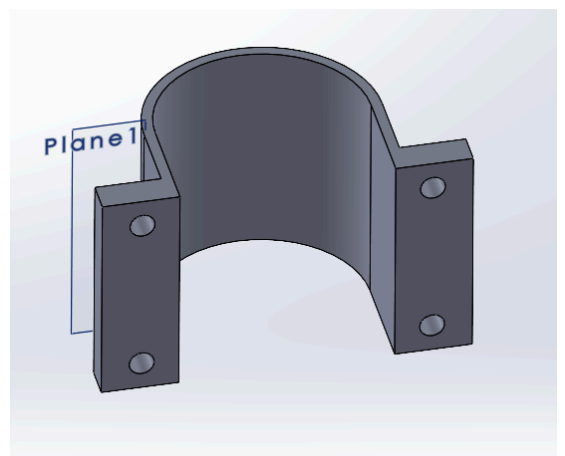
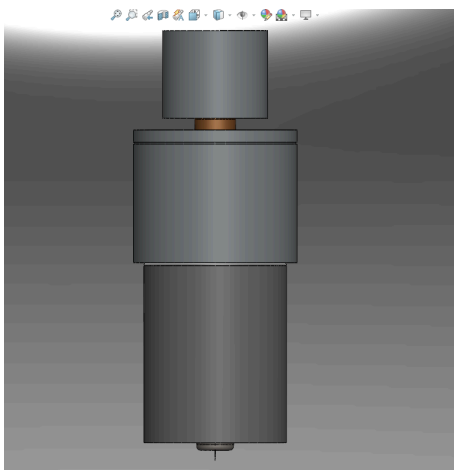
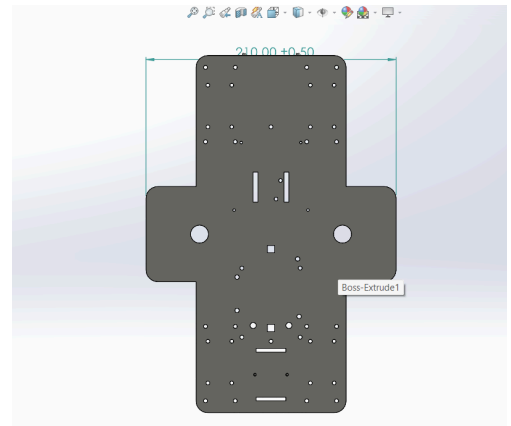
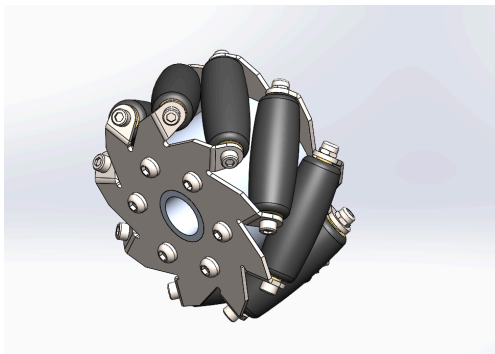
Ks. Dương Văn Tân

Đề tài : Mô phỏng robot 4 bánh Mecanum với tay máy 2 bậc tự do với ROS

I. Thiết kế robot :

để thiết kế được robot mecanum này thì em đã sử dụng bánh mecanum 100mm và tay máy 2 bậc tự do với 2 khớp quay .

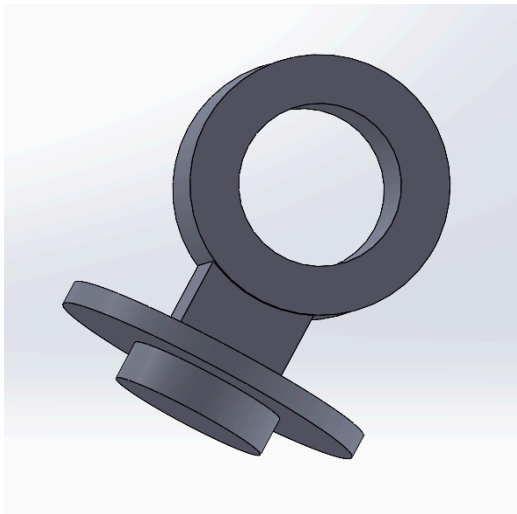
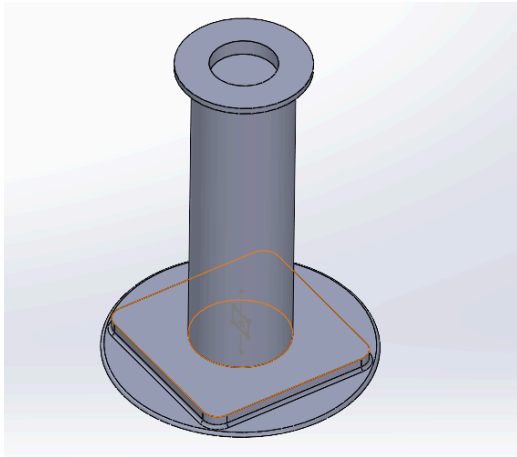
thiết kế phần khung bao gồm các thành phần sau :



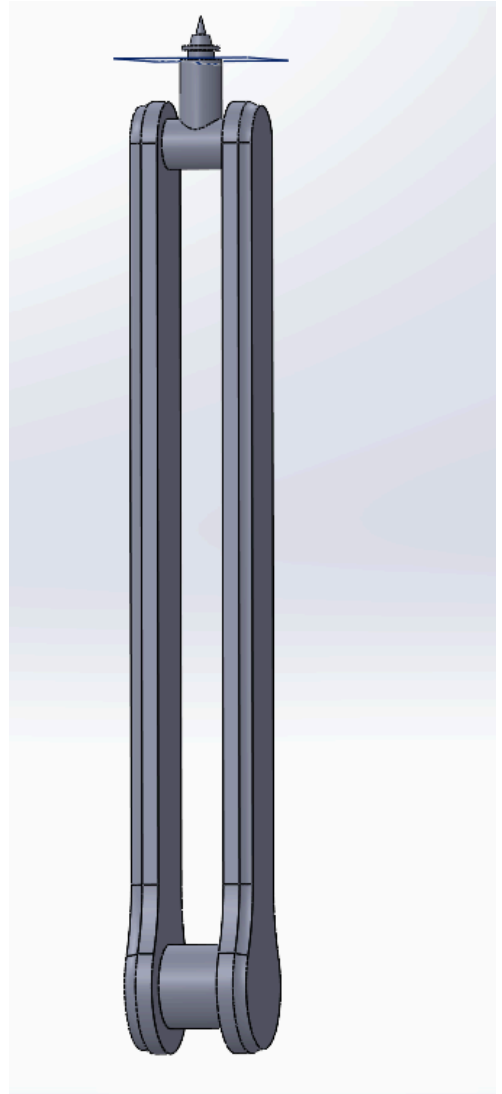
I.2. thiết kế tay máy :

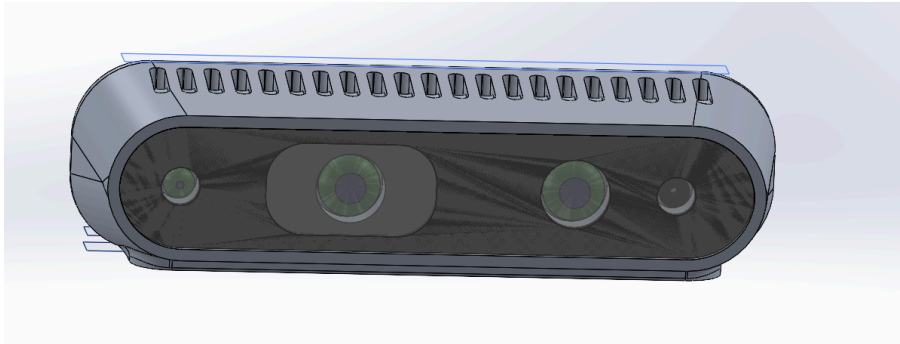
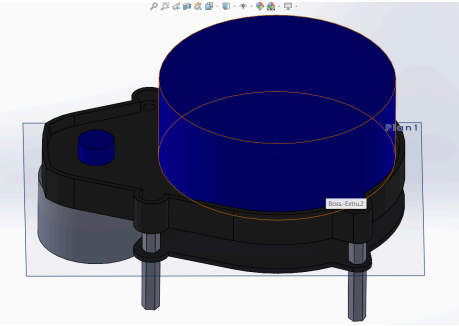
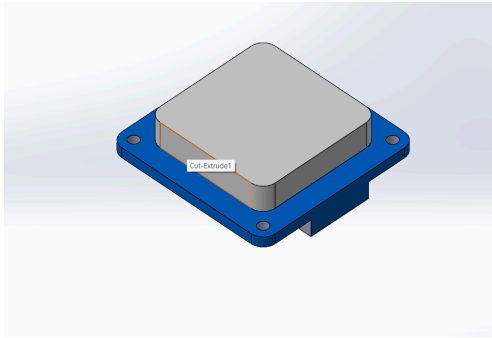
thiết kế sẽ gồm 3 khâu như sau :

phần trục của tay máy, là khớp cố định :

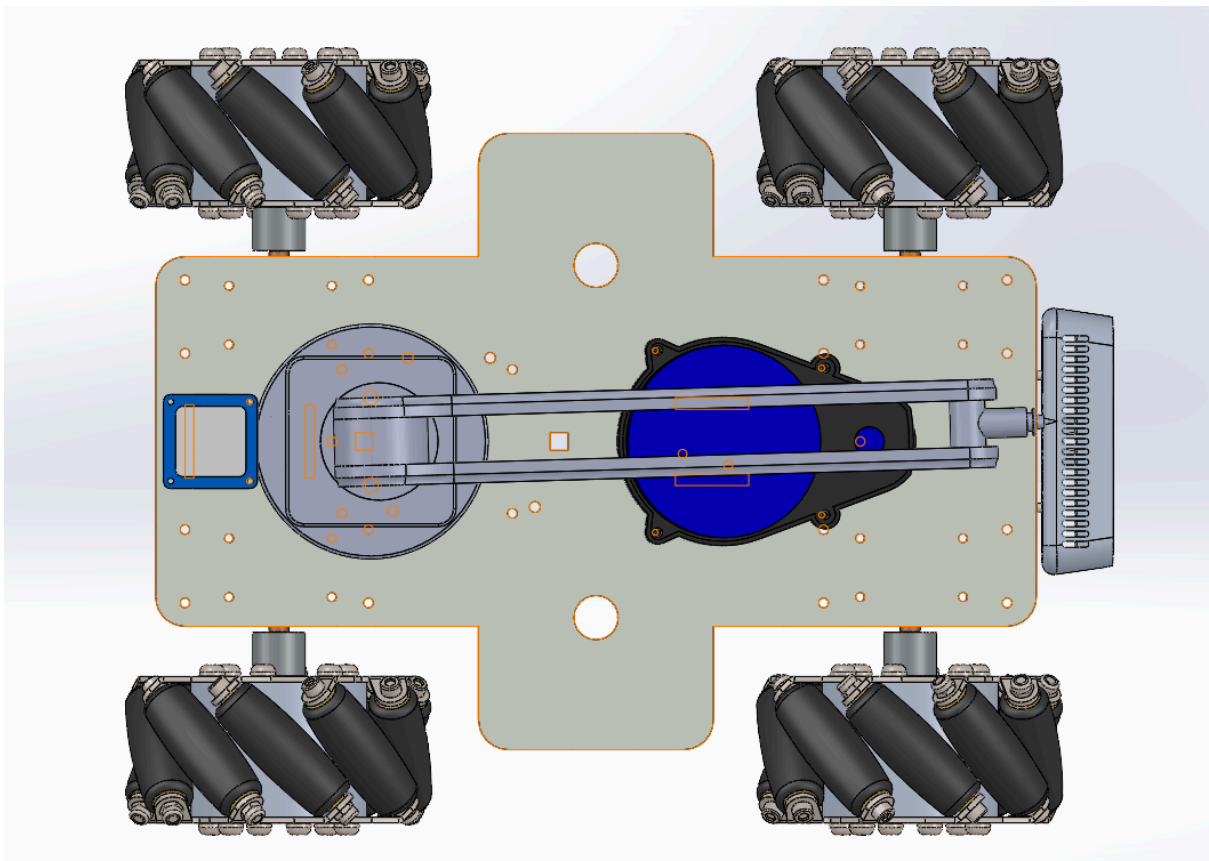


I.3. Các cảm biến sử dụng :

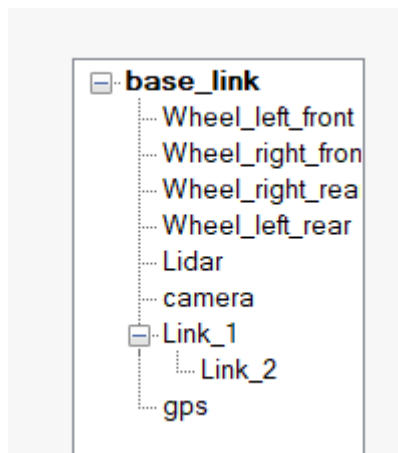
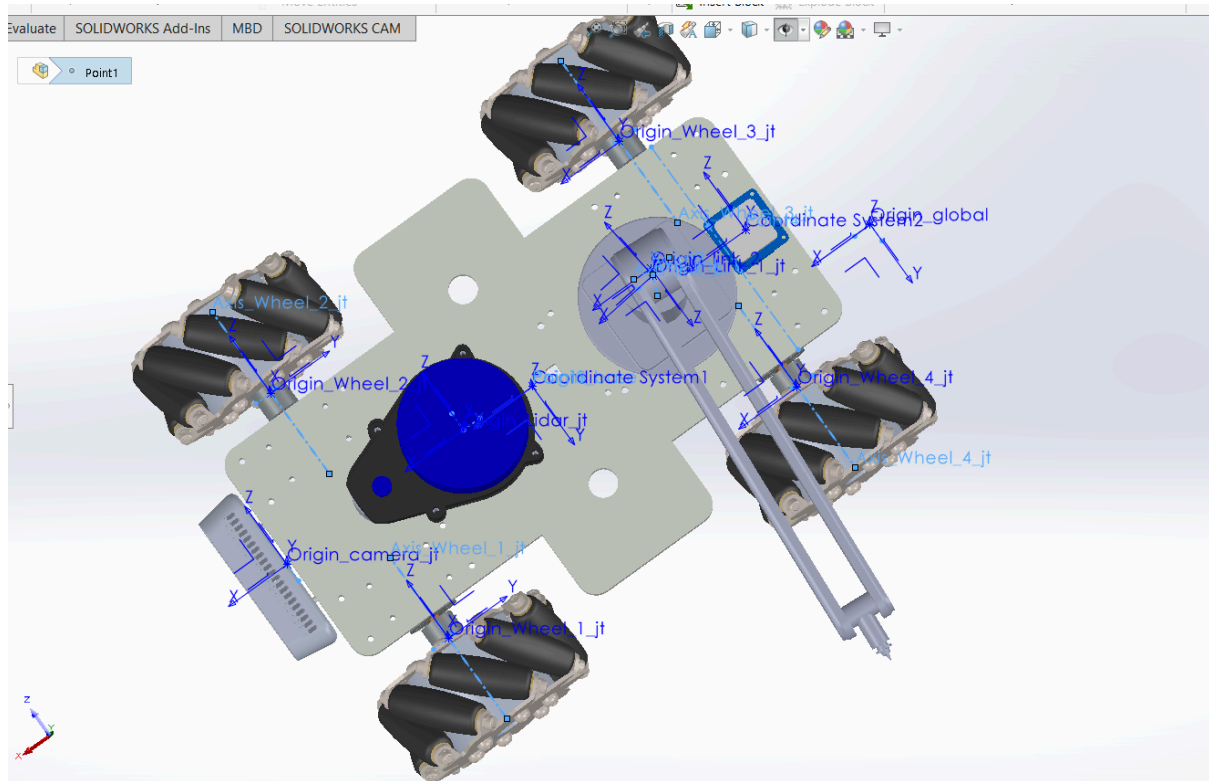


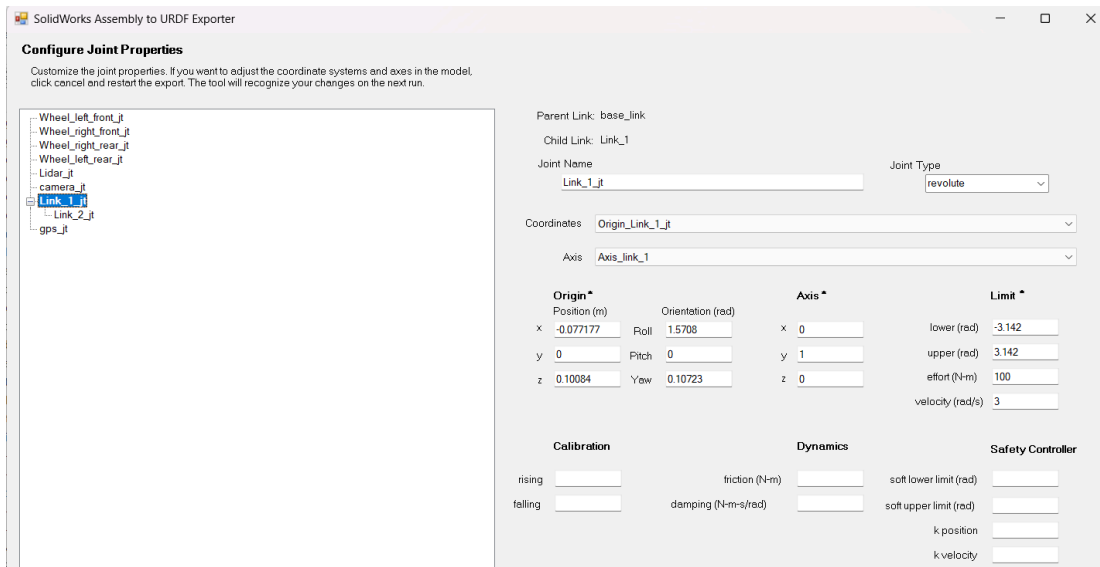


sau khi lắp ghép ta sẽ có thiết kế như sau :



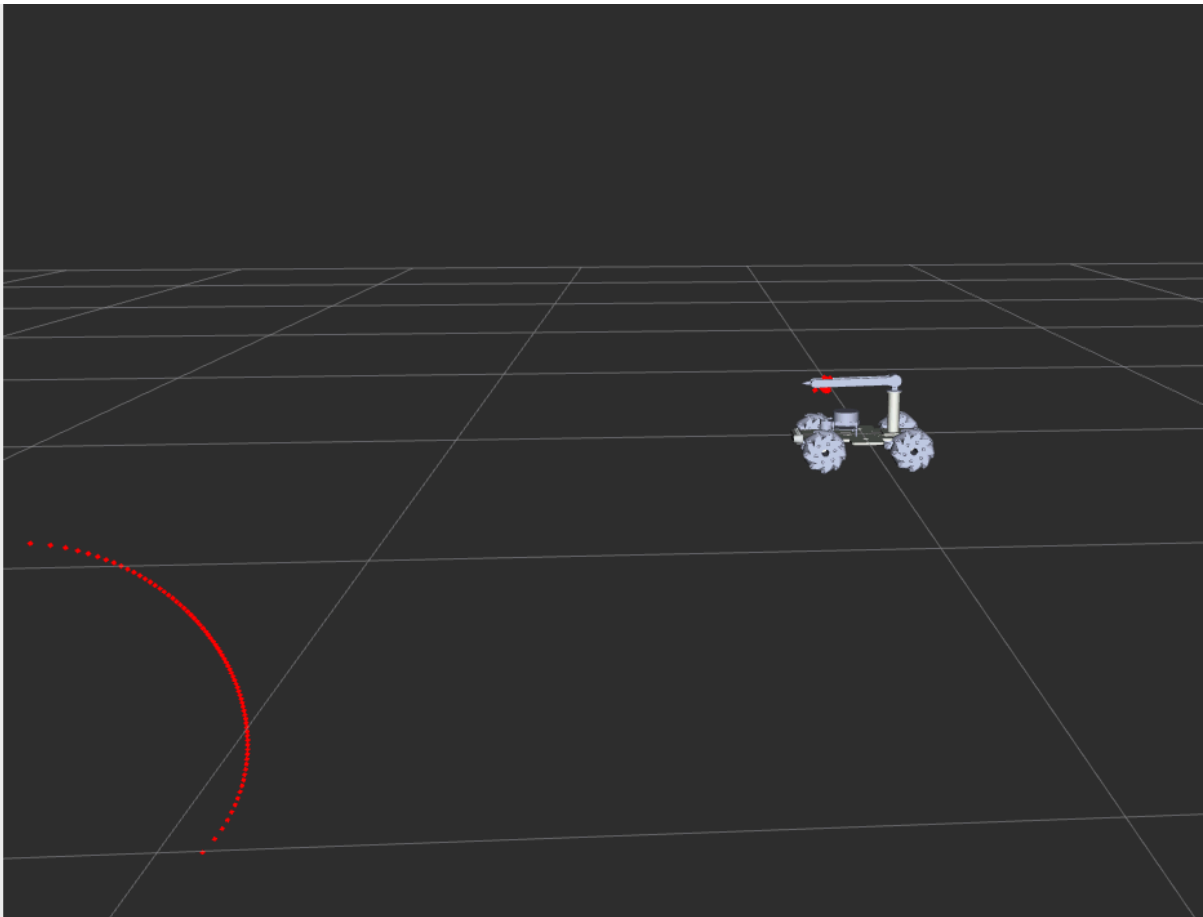
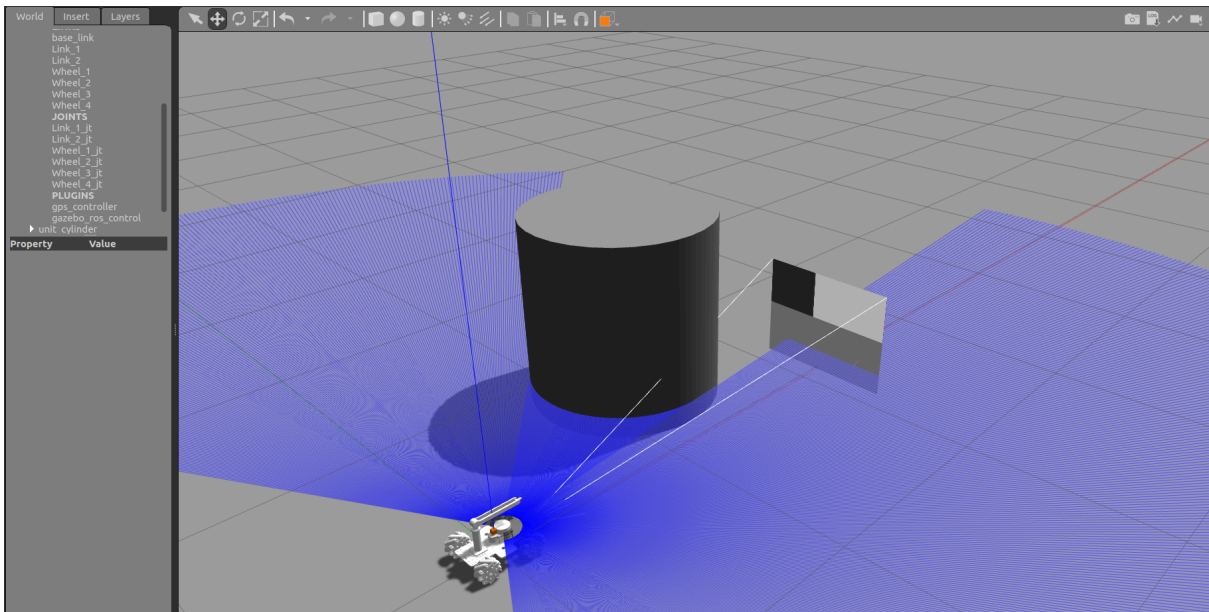
đặt trục các link :

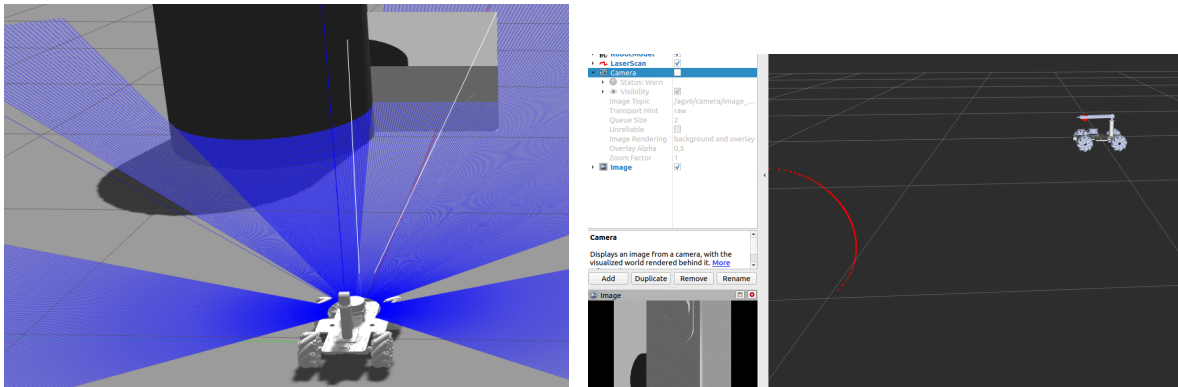




xuất file urdf : kiểm tra view urdf web ta kiểm tra các khớp có hoạt động đúng như mong đợi







II. Chương trình :

bước đầu :

chuyển đổi file urdf sang file xacro để dễ quản lý :

sử dụng các tính năng của Xacro như macro, tham số hóa và tái sử dụng code để làm cho file gọn gàng

tận dụng các macro để định nghĩa các thành phần lặp lại như bánh xe (wheels) như sau :

```
robot_agv00 > urdf > robot_agv00.xacro
2  <robot name="robot_agv00" xmlns:xacro="http://www.ros.org/wiki/xacro">
3    <xacro:property name="wheel_offset_x" value="0.107"/>
4    <xacro:property name="wheel_offset_y" value="0.076"/>
5    <xacro:property name="wheel_z" value="-0.019"/>
6    <xacro:property name="arm_base_x" value="-0.07716"/>
7    <xacro:property name="arm_base_z" value="0.10084"/>
8    <xacro:property name="link_1_length" value="0.035395"/>
9
10
11    <!-- Macro cho bánh xe mecanum -->
12    <xacro:macro name="mecanum_wheel" params="name parent x y z roll pitch yaw axis_x axis_y axis_z">
13      <link name="${name}">
14        <inertial>
15          <origin xyz="0 0 ${z}" rpy="0 0 0"/>
16          <mass value="0.266614887622473"/>
17          <inertia ixx="8.0544307064588E-05" ixy="0" ixz="0" iyy="8.05443070625978E-05" iyz="0" izz="9.31811521324714E-05"/>
18        </inertial>
19        <visual>
20          <origin xyz="0 0 0" rpy="0 0 0"/>
21          <geometry>
22            <mesh filename="package://robot_agv00/meshes/${name}.STL"/>
23          </geometry>
24          <material name="wheel_color">
25            <color rgba="0.792156862745098 0.819607843137255 0.933333333333333 1"/>
26          </material>
27        </visual>
28        <collision>
29          <origin xyz="0 0 0" rpy="0 0 0"/>
30          <geometry>
31            <mesh filename="package://robot_agv00/meshes/${name}.STL"/>
32          </geometry>
33        </collision>
34      </link>
35    </xacro:macro>
36  </robot>
```

```

        </geometry>
    </collision>
</link>
<joint name="${name}_jt" type="continuous">
    <parent link="${parent}" />
    <child link="${name}" />
    <origin xyz="${x} ${y} ${wheel_z}" rpy="${roll} ${pitch} ${yaw}" />
    <axis xyz="${axis_x} ${axis_y} ${axis_z}" />
</joint>
<!-- Thêm Gazebo transmission cho bánh xe -->
<transmission name="${name}_trans">
    <type>transmission_interface/SimpleTransmission</type>
    <joint name="${name}_jt">
        <hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInterface>
    </joint>
    <actuator name="${name}_motor">
        <mechanicalReduction>1</mechanicalReduction>
        <hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInterface>
    </actuator>
</transmission>
</xacro:macro>

<!-- Macro cho cảm biến cố định -->
<xacro:macro name="fixed_sensor" params="name parent x y z roll pitch yaw mass ixx ixy ixz iyy iyz izz">
    <link name="${name}">
        <inertial>

```

add cảm biến lidar ::

```

<!-- Gắn LiDAR -->
<xacro:fixed_sensor name="Lidar" parent="base_link" x="0.041732395661752" y="0" z="0.027" roll="0" pitch="-${pi/2}" yaw=
    mass="0.146695752131392" ixx="0.000125682524275729" ixy="0" ixz="1.88658076337422E-05"
    iyy="0.000109673997590948" iyz="0" izz="6.22721120792819E-05"/>

<!-- Add Gazebo plugin for Lidar -->
<gazebo reference="Lidar">
    <sensor name="laser" type="ray">
        <pose>0.015 0 0 0 ${pi/2} 0</pose>
        <ray>
            <scan>
                <horizontal>
                    <!-- The URG-04LX-UG01 has 683 steps with 0.35139 Degree resolution -->
                    <resolution>1</resolution>
                    <max_angle>2.0944</max_angle> <!-- 120 Degree -->
                    <min_angle>-2.0944</min_angle> <!-- -120 Degree -->
                    <samples>683</samples>
                </horizontal>
            </scan>
            <range>
                <min>0.08</min>
                <max>5</max>
                <resolution>0.01</resolution>
            </range>
        </ray>
        <plugin name="laser" filename="libgazebo_ros_laser.so">
            <robotNamespace></robotNamespace>
            <topicName>Lidar/scan</topicName>
            <frameName>Lidar</frameName>
        </plugin>
    </sensor>
</gazebo>

```



```

<!-- Add Gazebo plugin for Camera -->
<gazebo reference="camera">
  <sensor type="camera" name="camera">
    <pose>0 0 0 -${pi/2} 0</pose>
    <update_rate>30.0</update_rate>
    <camera name="head">
      <horizontal_fov>1.3962634</horizontal_fov>
      <image>
        <width>640</width>
        <height>480</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>camera</cameraName>
      <imageTopicName>image_raw</imageTopicName>
      <cameraInfoTopicName>camera_info</cameraInfoTopicName>
      <frameName>camera</frameName>
      <hackBaseline>0.07</hackBaseline>
    </plugin>
  </sensor>
</gazebo>

```

bật visualize để quan sát trong gazebo :

```
<visualize>true</visualize>
```

add gps :

tải plugin cần thiết :

sudo apt install ros-noetic-hector-gazebo-plugins

add plugin gps vào file urdf :

kiểm tra trong rviz hoặc rostopic list khi chạy gazebo ta sẽ thấy topic gps

để lấy dữ liệu từ Topic GPS ta dùng rostopic echo rồi trở đến dữ liệu cần lấy ra :

```

-
▼ /gps
  ▼ /position
    /parameter_descriptions
    /parameter_updates
  ▼ /status
    /parameter_descriptions
    /parameter_updates
  ▼ /velocity
    /parameter_descriptions
    /parameter_updates
  /gps_velocity
  /rosout
  /rosout_agg
  /tf
  /tf_static

```

/gps :

```
huy@huy: ~  
longitude: -3.1862913881136503  
altitude: 0.0708788228644004  
position_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]  
position_covariance_type: 2  
---  
header:  
  seq: 1952  
  stamp:  
    secs: 195  
    nsecs: 600000000  
  frame_id: "gps_base_link"  
status:  
  status: 0  
  service: 0  
latitude: 55.94439944167078  
longitude: -3.186291388658048  
altitude: 0.07089182538099274  
position_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]  
position_covariance_type: 2  
---  
header:  
  seq: 1953  
  stamp:  
    secs: 195  
    nsecs: 700000000  
  frame_id: "gps_base_link"  
status:  
  status: 0  
  service: 0  
latitude: 55.94439944057238  
longitude: -3.186291389294288  
altitude: 0.07089138379839145  
position_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]  
position_covariance_type: 2  
---  
|
```

gps_vehical

```
huy@huy: ~  
z: -0.006420028457387661  
---  
header:  
  seq: 2142  
  stamp:  
    secs: 214  
    nsecs: 600000000  
  frame_id: "gps_base_link"  
vector:  
  x: -0.0006078003320993258  
  y: 3.9869141506381456e-05  
  z: -0.0036099963968999247  
---
```

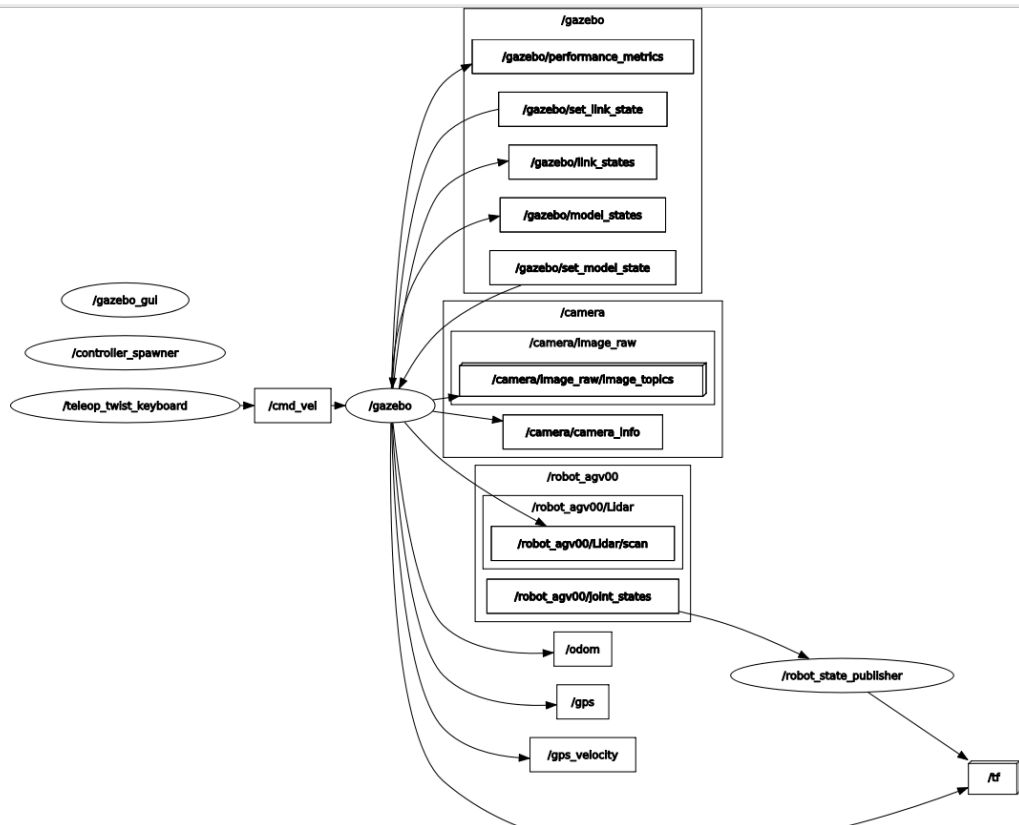
sang phần điều khiển :

em đã thực hiện 2 hướng điều khiển như sau :

dùng thư viện skid :

libgazebo_ros_skid_steer_drive.so hoặc thư viện gazebo_mecanum_plugins thì đều có thể chạy được bánh xe :

rqt_graph



cách thứ 2 là : tạo 1 file điều khiển, publish thông tin là các topic của 4 bánh xe :

tạo 1 file controller.yaml :

tức là mình sẽ lần lượt push các giá trị điều khiển lên các bánh xe wheel với type :

```
velocity_controllers
```

```

1  robot_agv0:
2      # Add joint_state_controller
3      joint_state_controller:
4          type: joint_state_controller/JointStateController
5          publish_rate: 50
6      # Điều khiển vận tốc cho bánh xe
7      Wheel_left_front_jt_controller:
8          type: velocity_controllers/JointVelocityController
9          joint: Wheel_left_front_jt
10         pid: {p: 10.0, i: 0.1, d: 0.01}
11      Wheel_right_front_jt_controller:
12          type: velocity_controllers/JointVelocityController
13          joint: Wheel_right_front_jt
14          pid: {p: 10.0, i: 0.1, d: 0.01}
15      Wheel_right_rear_jt_controller:
16          type: velocity_controllers/JointVelocityController
17          joint: Wheel_right_rear_jt
18          pid: {p: 10.0, i: 0.1, d: 0.01}
19      Wheel_left_rear_jt_controller:
20          type: velocity_controllers/JointVelocityController
21          joint: Wheel_left_rear_jt
22          pid: {p: 10.0, i: 0.1, d: 0.01}

```

thêm 1 cái pugin control :

```

<gazebo>
  <plugin name="gazebo_ros_control"
filename="libgazebo_ros_control.so">
    <robotNamespace>/agv6</robotNamespace>
  </plugin>
</gazebo>

```

phải cài đặt trước nếu như chưa cài:

```
sudo apt-get update
```

```
sudo apt-get install ros-noetic-ros-controllers
```

phần file urdf phải có các thẻ transimisson để thiết lập chuyển :

```

<transmission name="wheel_right_front_trans">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="Wheel_right_front_jt">

<hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInterface>

  </joint>
  <actuator name="wheel_right_front_motor">

<hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInterface>

    <mechanicalReduction>1</mechanicalReduction>
  </actuator>

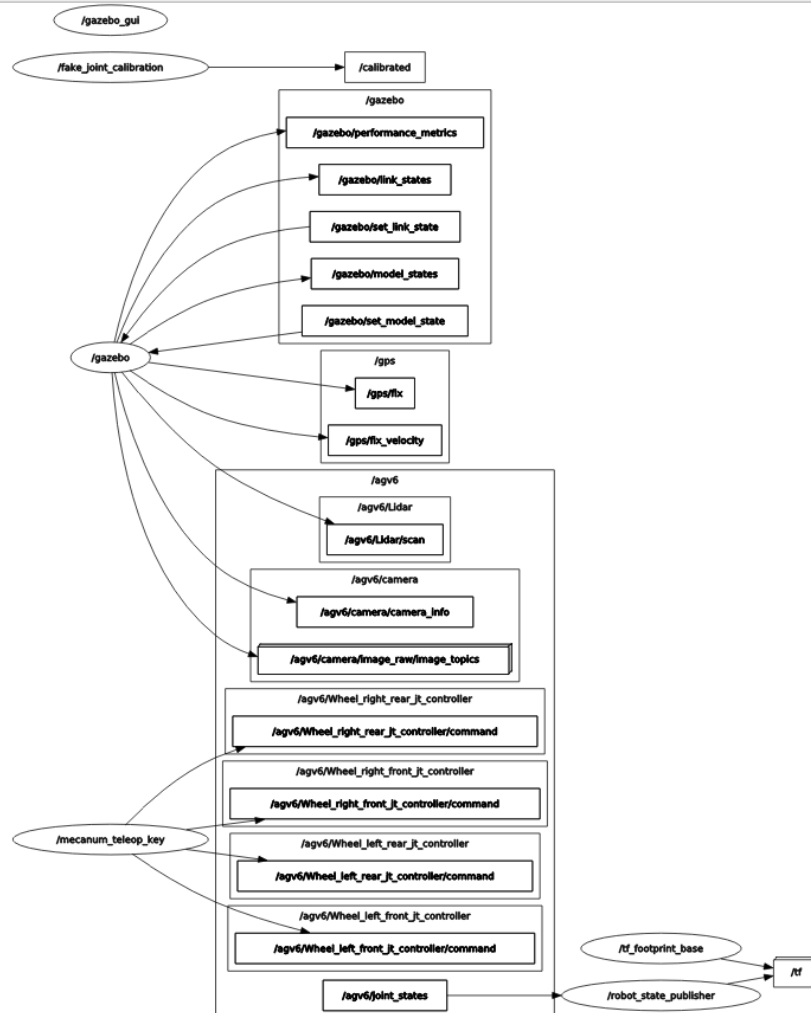
```

</transmission>

nó sẽ đóng vai trò quan trọng trong việc mô tả cách các khớp của robot được điều khiển bởi các bộ truyền động trong mô phỏng, khi sử dụng Gazebo, Nó là cầu nối giữa mô hình vật lý của robot (định nghĩa trong URDF) và file điều khiển

thực hiện theo sơ đồ sqt_graph như sau :

- chưa thể publish lên gazebo



III. kết quả mô phỏng :

- Đã mô phỏng cơ bản model trong rviz và gazebo cũng như đọc được topic của sensor.
- (Nếu gặp lỗi trong hiển thị gazebo có thể xem lại ma trận quá tính.
- Mô phỏng được chuyển động của bánh xe nhưng còn khá khó khăn.
- Trong phần điều khiển của bánh xe theo cách 2, không thể publish tốc độ lên gazebo video mô phỏng :

<https://drive.google.com/file/d/1NyqVJ-HP36hqlus1yQUIPR3WHFIUjBJR/view?usp=s>
haring

