

# BÁO CÁO ROS- XÂY DỰNG, MÔ PHỎNG VÀ ĐIỀU KHIỂN ROBOT

Họ và tên: Đinh Mạnh Tuyên

Mã sinh viên: 22027548

## I. Thiết kế robot

### 1. Dạng robot thiết kế:

Yêu cầu thiết kế robot Omni 3 bánh, có khả năng di chuyển linh hoạt nhờ bánh Omni. Robot có cánh tay 2 khớp quay để chuyển động, cùng các cảm biến Camera, LIDAR và GPS để quan sát, lập bản đồ và định vị.

### 2. Chuyển động của xe:

Sử dụng thuật toán động học ngược (inverse kinematics) để tính vận tốc từng bánh dựa trên vận tốc mong muốn ( $v_x$ ,  $v_y$ ,  $\omega$ ).

Công thức động học ngược:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & L \\ -\sin(\theta_2) & \cos(\theta_2) & L \\ -\sin(\theta_3) & \cos(\theta_3) & L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

Với ba bánh Omni (omni\_1, omni\_2, omni\_3) được gắn trên base\_link với các trục quay nghiêng (lần lượt là  $0.5, -0.866, 0.5$ ,  $-0.866, 0.5, -0.866, 0$ ;  $0.5, 0.866, 0.5$ ,  $0.866, 0.5, 0.866, 0$ ; và  $-1, 0, 0$ ,  $-1, 0, 0$ ), thuật toán này cho phép robot di chuyển linh hoạt theo mọi hướng và xoay tại chỗ.

### 3. Chuyển động của tay máy:

Áp dụng thuật toán PID (Proportional-Integral-Derivative) để điều khiển vị trí các khớp (part\_1\_joint, part\_2\_joint, part\_3\_joint)

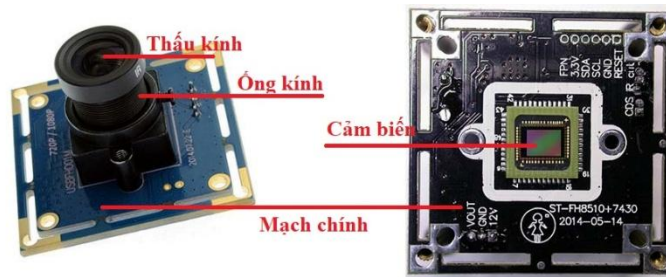
Thuật toán PID được sử dụng để điều khiển vị trí của các khớp trên cánh tay robot, bao gồm part\_1\_joint, part\_2\_joint, và part\_3\_joint

Công thức:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

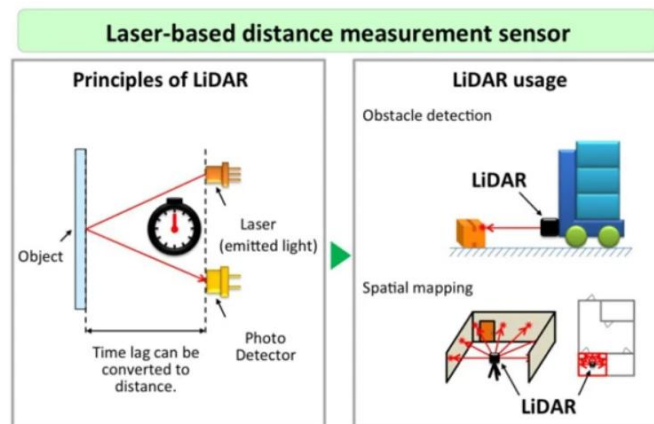
### 4. Camera:

Cảm biến camera hoạt động bằng cách thu nhận ánh sáng qua ống kính, sau đó lọc màu (thường dùng bộ lọc Bayer với các điểm ảnh đỏ, lục, lam). Mỗi pixel trên cảm biến chứa photodiode để chuyển ánh sáng thành điện tích, điện tích này được chuyển đổi thành tín hiệu số nhờ bộ ADC. Bộ xử lý hình ảnh (ISP) sẽ tổng hợp thông tin màu sắc, điều chỉnh độ sáng, độ tương phản và tạo ra ảnh kỹ thuật số



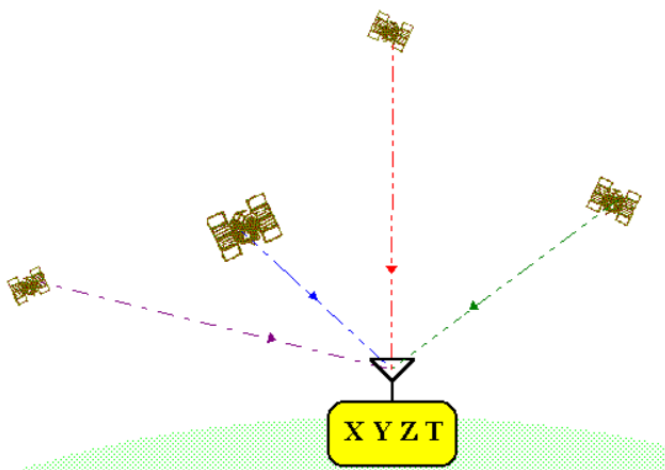
## 5. Cảm biến lidar:

Cảm biến LiDAR hoạt động bằng cách phát các chùm tia laser vào môi trường, đo thời gian tia laser phản xạ trở lại để tính toán khoảng cách (Time of Flight). Dữ liệu từ hàng nghìn điểm quét mỗi giây được tổng hợp tạo thành đám mây điểm 3D (point cloud), giúp xây dựng bản đồ không gian chính xác.



## 6. GPS:

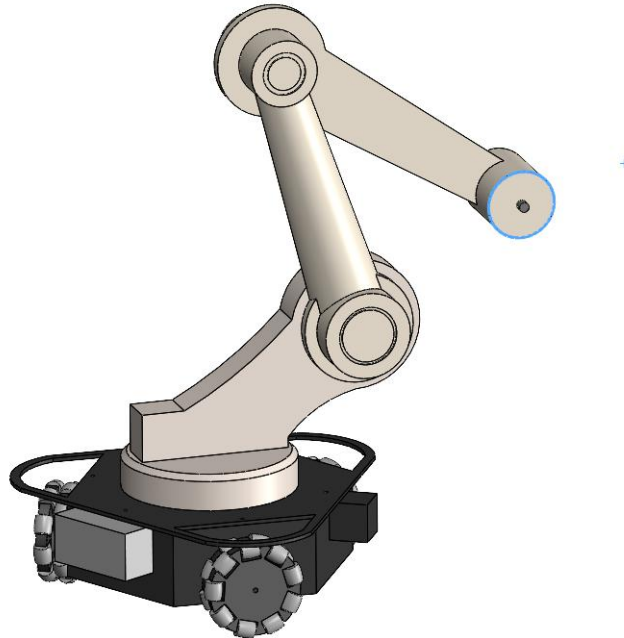
Trong mô hình robot, cảm biến GPS hoạt động bằng cách nhận tín hiệu từ các vệ tinh để xác định vị trí chính xác (kinh độ, vĩ độ, độ cao) thông qua phương pháp tam giác hóa.



## II. Thiết kế trong Solidwork

### 1. Thiết kế mô hình 3D

Phần khung theo hình lục giác để có thể phân bố đều 3 bánh xe, phần tay máy thiết kế 2 khớp xoay phù hợp với yêu cầu đề bài, dưới đây là hình ảnh thiết kế trong Solidwork:



## 2. Xuất file urdf:

Hệ trục tọa độ xác định vị trí và hướng của từng bộ phận robot

Base\_link: Thành phần chính (khung cơ sở) của robot

Lidar: Cảm biến LiDAR gắn cố định vào base\_link qua lidar\_joint

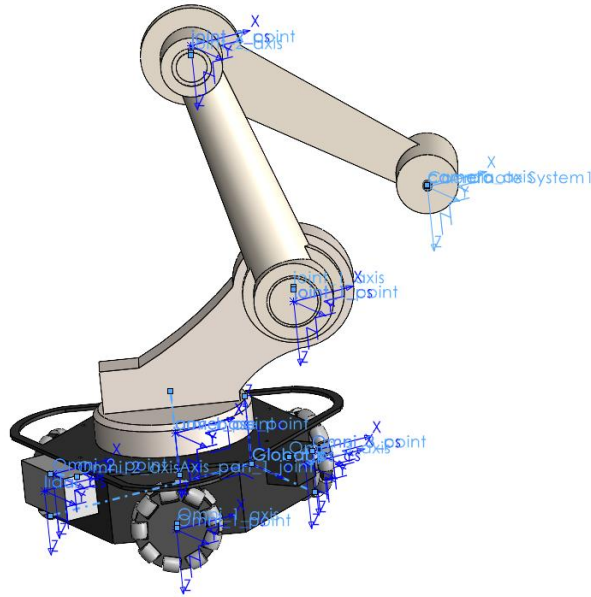
GPS: Cảm biến GPS gắn cố định vào base\_link qua gps\_joint

Omni\_1, Omni\_2, Omni\_3: Ba bánh xe gắn vào base\_link qua các khớp liên tục (continuous joint), cho phép robot di chuyển đa hướng với trục quay riêng

Part\_1, Part\_2, Part\_3: Các đoạn tay robot nối tiếp nhau. Part\_1 gắn vào cover qua khớp quay liên tục, part\_2 và Part\_3 nối với nhau qua khớp quay với giới hạn góc  $\pm 1.57$  rad, điều khiển vị trí trong Gazebo

Camera: Camera gắn cố định vào part\_3

Hệ quy chiếu: Hệ trục tuân theo chuẩn ROS (X hướng về trước, Y sang trái, Z lên trên). Các giá trị xyz và rpy trong joint mô tả cách hệ tọa độ của link con dịch chuyển và xoay so với link cha, tạo thành cấu trúc phân cấp



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

#### 1. Cấu trúc mô hình robot

```
<?xml version="1.0" encoding="utf-8"?>
<robot name="robot_giua_ki">
  <link name="base_link">
    <inertial>
      <origin xyz="0.129903810567668 0.0749999999999997 0.0270932977125576" rpy="0 0 0" />
      <mass value="8.0" />
      <!-- Phần inertia -->
    </inertial>
    <visual>
      <origin xyz="0 0 0" rpy="0 0 0" />
      <geometry>
        <mesh filename="package://robot_giua_ki/meshes/base_link.STL" />
      </geometry>
      <!-- Phần material -->
    </visual>
    <collision>
      <origin xyz="0 0 0" rpy="0 0 0" />
      <geometry>
        <mesh filename="package://robot_giua_ki/meshes/base_link.STL" />
      </geometry>
    </collision>
  </link>
  <!-- Các link và joint khác tiếp theo -->
</robot>
```

File URDF mô tả robot bắt đầu từ phần thân chính là `base_link`, đại diện cho khung xe. `Base_link` có khối lượng 8 kg, với trọng tâm lệch nhẹ so với gốc tọa độ của nó. Hình dạng hiển thị và vùng va chạm của `base_link` được xác định bằng file STL, cùng gốc tọa độ và không xoay.

Đây là bộ phận nền tảng, nơi các thành phần khác như bánh xe, cảm biến, và tay máy sẽ được gắn vào thông qua các khớp, tạo thành cấu trúc hoàn chỉnh của robot trong không gian 3D

## 2. Plugin

- Plugin điều khiển ROS (Gazebo ROS Control):  
Plugin libgazebo\_ros\_control.so cho phép điều khiển robot qua ROS, với không gian tên /robot\_giua\_ki, giúp kết nối các khớp (joint) với bộ điều khiển ROS

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

- Plugin cho LiDAR:  
Plugin libgazebo\_ros\_laser.so gắn vào lidar, xuất dữ liệu quét khoảng cách qua topic /robot\_giua\_ki/lidar/scan, mô phỏng cảm biến LiDAR trong Gazebo

```
<gazebo reference="lidar">
  <sensor name="lidar_sensor" type="ray">
    <!-- Cấu hình cảm biến -->
    <plugin name="lidar_controller" filename="libgazebo_ros_laser.so">
      <topicName>/robot_giua_ki/lidar/scan</topicName>
      <frameName>lidar</frameName>
    </plugin>
  </sensor>
</gazebo>
```

- Plugin cho Camera:

```
<gazebo reference="camera">
  <sensor name="camera_sensor" type="camera">
    <!-- Cấu hình cảm biến -->
    <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
      <alwaysOn>true</alwaysOn>
      <updateRate>10</updateRate>
      <cameraName>camera1</cameraName>
      <imageTopicName>/robot_giua_ki/camera/image_raw</imageTopicName>
      <cameraInfoTopicName>/robot_giua_ki/camera/camera_info</cameraInfoTopicName>
      <frameName>camera</frameName>
    </plugin>
  </sensor>
</gazebo>
```

Camera Plugin: Plugin libgazebo\_ros\_camera.so gắn vào camera, xuất hình ảnh qua topic /robot\_giua\_ki/camera/image\_raw và thông tin camera qua /robot\_giua\_ki/camera/camera\_info, mô phỏng camera thực tế

- Plugin cho GPS:

```

<gazebo reference="gps">
  <sensor name="gps_sensor" type="gps">
    <!-- Cấu hình cảm biến -->
    <plugin name="gps_controller" filename="libgazebo_ros_gps.so">
      <topicName>/robot_giua_ki/gps/fix</topicName>
      <frameName>gps</frameName>
    </plugin>
  </sensor>
</gazebo>

```

Plugin libgazebo\_ros\_gps.so gắn vào gps, cung cấp dữ liệu vị trí, mô phỏng cảm biến GPS.

### 3. Cảm biến

- Cảm biến lidar:

```

<link name="lidar">
  <inertial>
    <origin xyz="0.02 0 0" rpy="0 0 0" />
    <mass value="0.2" />
    <!-- Phần inertia rút gọn -->
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_giua_ki/meshes/lidar.STL" />
    </geometry>
    <material name="">
      <color rgba="0.79 0.82 0.93 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_giua_ki/meshes/lidar.STL" />
    </geometry>
  </collision>
</link>
<joint name="lidar_joint" type="fixed">
  <origin xyz="0.2998 0.0713 0.055" rpy="3.1416 0 3.1416" />
  <parent link="base_link" />
  <child link="lidar" />
  <axis xyz="0 0 0" />
</joint>
<gazebo reference="lidar">
  <sensor name="lidar_sensor" type="ray">
    <pose>0 0 0 0 0 0</pose>
    <visualize>true</visualize>
    <update_rate>10</update_rate>
  <ray>
    <scan>
      <horizontal>
        <samples>360</samples>
        <resolution>1</resolution>
        <min_angle>-3.1416</min_angle>
        <max_angle>3.1416</max_angle>

```

```

    </horizontal>
  </scan>
  <range>
    <min>0.1</min>
    <max>10.0</max>
    <resolution>0.01</resolution>
  </range>
</ray>
<plugin name="lidar_controller" filename="libgazebo_ros_laser.so">
  <topicName>/robot_giua_ki/lidar/scan</topicName>
  <frameName>lidar</frameName>
</plugin>
</sensor>
</gazebo>

```

LiDAR gắn cố định lên base\_link, quét 360 độ xung quanh, đo khoảng cách từ 0.1 m đến 10 m, cập nhật 10 lần mỗi giây. Dữ liệu quét được gửi qua topic /robot\_giua\_ki/lidar/scan để robot biết xung quanh có gì. Plugin libgazebo\_ros\_laser.so giúp nó hoạt động trong Gazebo.

- **Cảm biến Camera:**

```

<link name="camera">
  <inertial>
    <origin xyz="0.0023 -0.0044 0" rpy="0 0 0" />
    <mass value="0.000785" />
    <!-- Phần inertia rút gọn -->
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_giua_ki/meshes/camera.STL" />
    </geometry>
    <material name="">
      <color rgba="0.79 0.82 0.93 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_giua_ki/meshes/camera.STL" />
    </geometry>
  </collision>
</link>
<joint name="camera_joint" type="fixed">
  <origin xyz="0.1841 0.2036 0.1532" rpy="0 0 0" />
  <parent link="part_3" />
  <child link="camera" />
  <axis xyz="0 0 0" />
</joint>
<gazebo reference="camera">
  <sensor name="camera_sensor" type="camera">
    <pose>0 0 0 0 0 0</pose>
    <visualize>true</visualize>
  </sensor>
</gazebo>

```

```

<update_rate>10</update_rate>
<camera>
  <horizontal_fov>1.0472</horizontal_fov>
  <image>
    <width>640</width>
    <height>480</height>
    <format>R8G8B8</format>
  </image>
  <clip>
    <near>0.1</near>
    <far>100.0</far>
  </clip>
</camera>
<plugin name="camera_controller"
filename="libgazebo_ros_camera.so">
  <alwaysOn>true</alwaysOn>
  <updateRate>10</updateRate>
  <cameraName>camera1</cameraName>

<imageTopicName>/robot_giua_ki/camera/image_raw</imageTopicName>

<cameraInfoTopicName>/robot_giua_ki/camera/camera_info</cameraInfo
TopicName>
  <frameName>camera</frameName>
</plugin>
</sensor>
</gazebo>

```

Camera gắn vào part\_3 (phần cuối tay máy). Nó chụp ảnh độ phân giải 640x480, góc nhìn ngang 60 độ, thấy được từ 0.1 m đến 100 m, cập nhật 10 lần mỗi giây. Ảnh được gửi qua topic /robot\_giua\_ki/camera/image\_raw, còn thông tin camera qua /robot\_giua\_ki/camera/camera\_info. Plugin libgazebo\_ros\_camera.so giúp nó quay phim trong Gazebo

- Cảm biến GPS:

```

<link name="gps">
  <inertial>
    <origin xyz="-0.01 -0.0173 0" rpy="0 0 0" />
    <mass value="0.064" />
    <!-- Phần inertia rút gọn -->
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_giua_ki/meshes/gps.STL" />
    </geometry>
    <material name="">
      <color rgba="0.25 0.25 0.25 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />

```



```

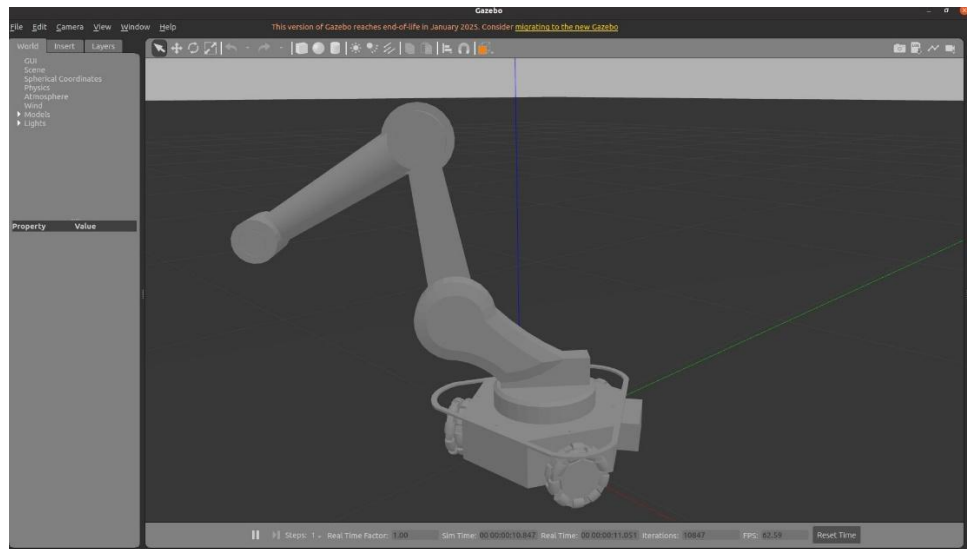
    <geometry>
      <mesh filename="package://robot_giua_ki/meshes/gps.STL" />
    </geometry>
  </collision>
</link>
<joint name="gps_joint" type="fixed">
  <origin xyz="0.0449 0.2221 0.06" rpy="3.1416 0 3.1416" />
  <parent link="base_link" />
  <child link="gps" />
  <axis xyz="0 0 0" />
</joint>
<gazebo reference="gps">
  <sensor name="gps_sensor" type="gps">
    <pose>0 0 0 0 0 0</pose>
    <update_rate>10</update_rate>
  <gps>
    <referenceLatitude>37.7749</referenceLatitude>
    <referenceLongitude>-122.4194</referenceLongitude>
    <referenceAltitude>0.0</referenceAltitude>
    <noise>
      <type>gaussian</type>
      <mean>0.0</mean>
      <stddev>0.01</stddev>
    </noise>
  </gps>
  <plugin name="gps_controller" filename="libgazebo_ros_gps.so">
    <topicName>/robot_giua_ki/gps/fix</topicName>
    <frameName>gps</frameName>
  </plugin>
</sensor>
</gazebo>

```

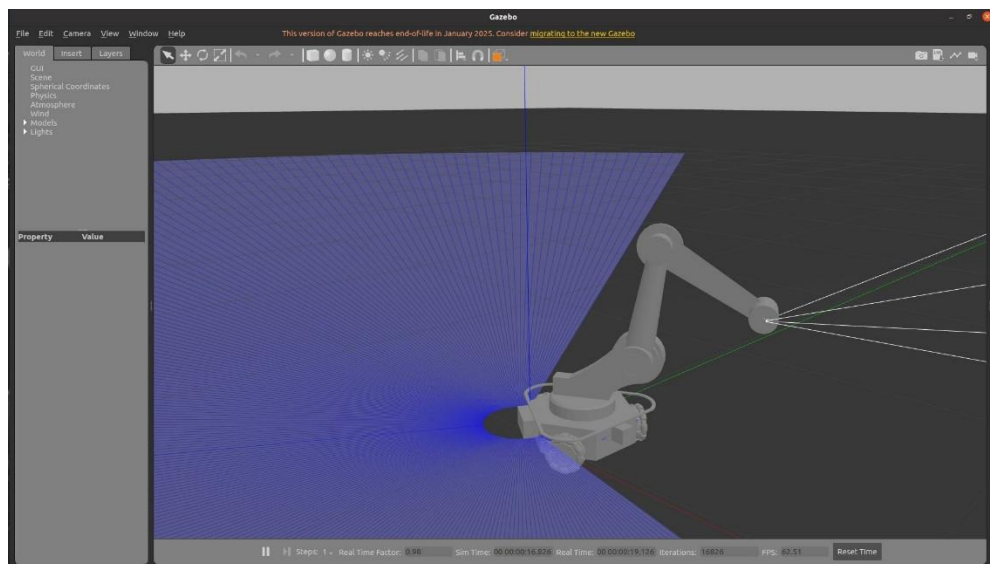
GPS gắn cố định lên base\_link. Nó cho biết vị trí (kinh độ, vĩ độ, độ cao) dựa trên tham chiếu, cập nhật 10 lần mỗi giây, có chút nhiễu nhỏ. Dữ liệu vị trí gửi qua topic /robot\_giua\_ki/gps/fix. Plugin libgazebo\_ros\_gps.so làm nó chạy được trong Gazebo

## IV. Kết quả

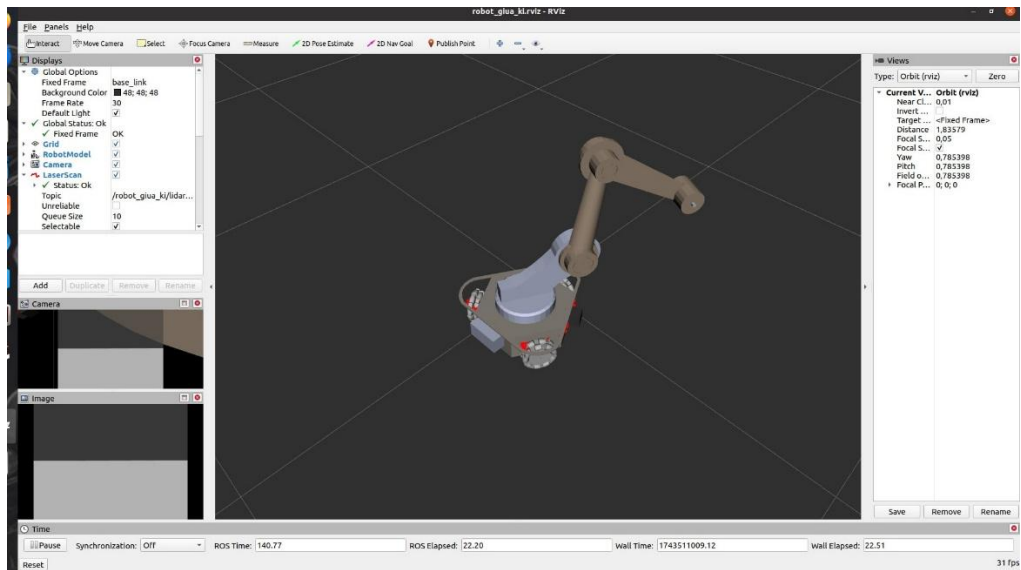
### 1. Hình ảnh mô hình trong Gazebo



## 2. Hình ảnh mô hình chạy và quét xung quanh trong gazebo



## 3. Hình mô hình trong RVIZ



## V. Tổng kết

Robot được thiết kế đầy đủ với các bộ phận chính như thân xe, bánh xe omni, tay máy, cùng các cảm biến LiDAR, Camera, và GPS. Tuy nhiên, do tay máy có các khớp với độ cao cố định, cảm biến LiDAR gắn trên `base_link` bị che khuất một phần bởi `part_1`, `part_2`, hoặc `part_3`, làm giảm hiệu quả quét của tia laser. Thân xe khá nhỏ gọn, không đủ không gian để đặt LiDAR ở vị trí tối ưu, đây là hạn chế trong thiết kế ban đầu. Robot có thể điều khiển qua ROS và Gazebo, nhưng chuyển động vẫn còn gặp vấn đề khi tiến lên hoặc lùi lại. Lỗi này có thể xuất phát từ việc tính toán thông số cho bộ điều khiển chưa chính xác, hoặc trục axis của các bánh omni trong URDF chưa được căn chỉnh hoàn hảo.