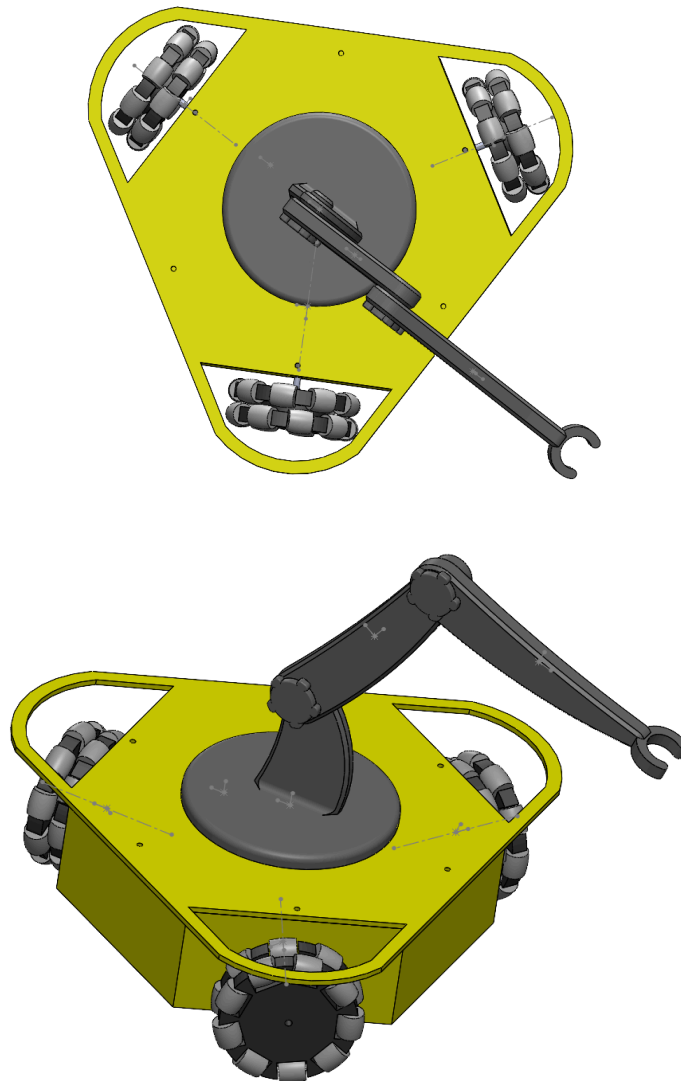


## Báo cáo giữa kỳ ROS Xây dựng mô phỏng và điều khiển Robot

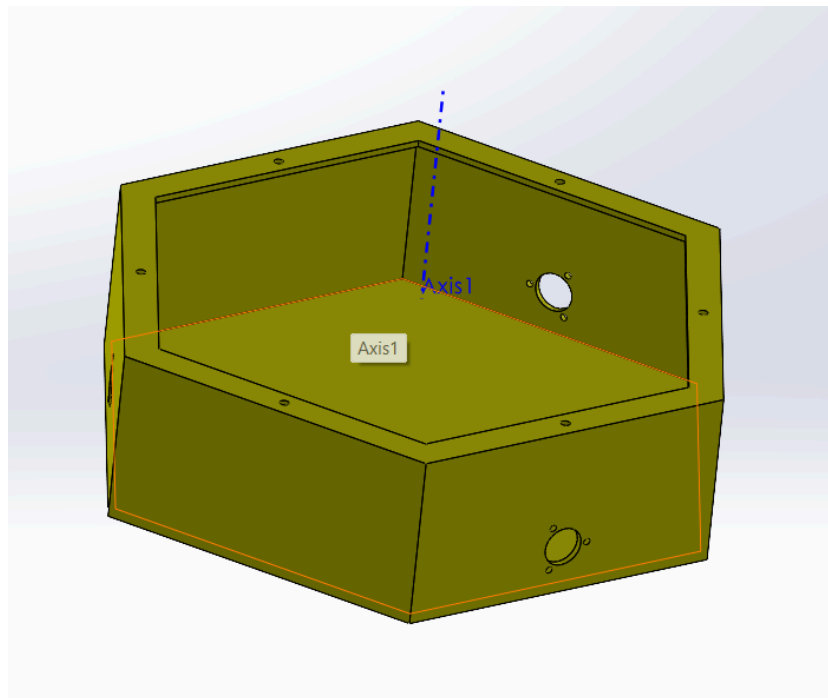
### 1. Thành phần cấu tạo



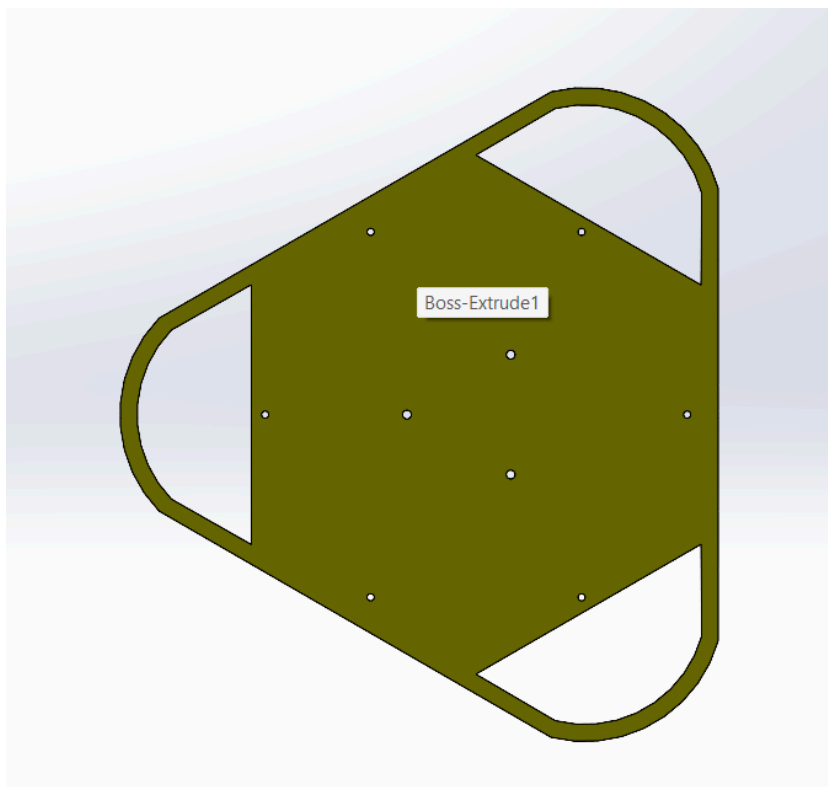
Các hình ảnh tổng quan về robot

- + Tổng quan: Đây là Robot sử dụng 3 bánh xe omni, có khả năng di chuyển theo mọi hướng bằng cách thay đổi tốc độ và hướng của từng bánh xe mà không cần thay đổi hướng thân xe. Điều này đạt được nhờ sử dụng ba bánh xe omni-directional, mỗi bánh được điều khiển bởi một động cơ DC có encoder.

+ Các thành phần chính của Robot:

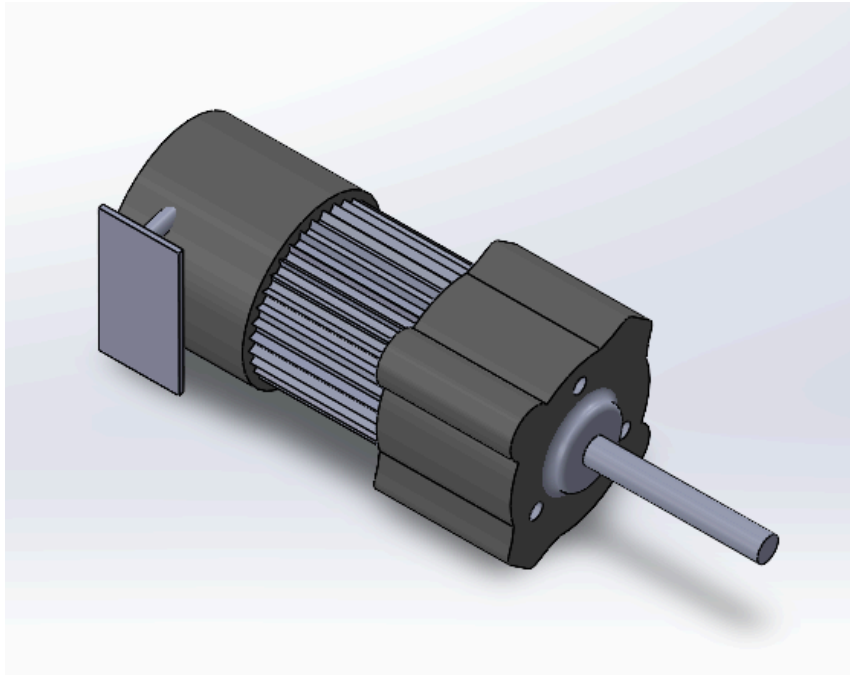


Khung xe

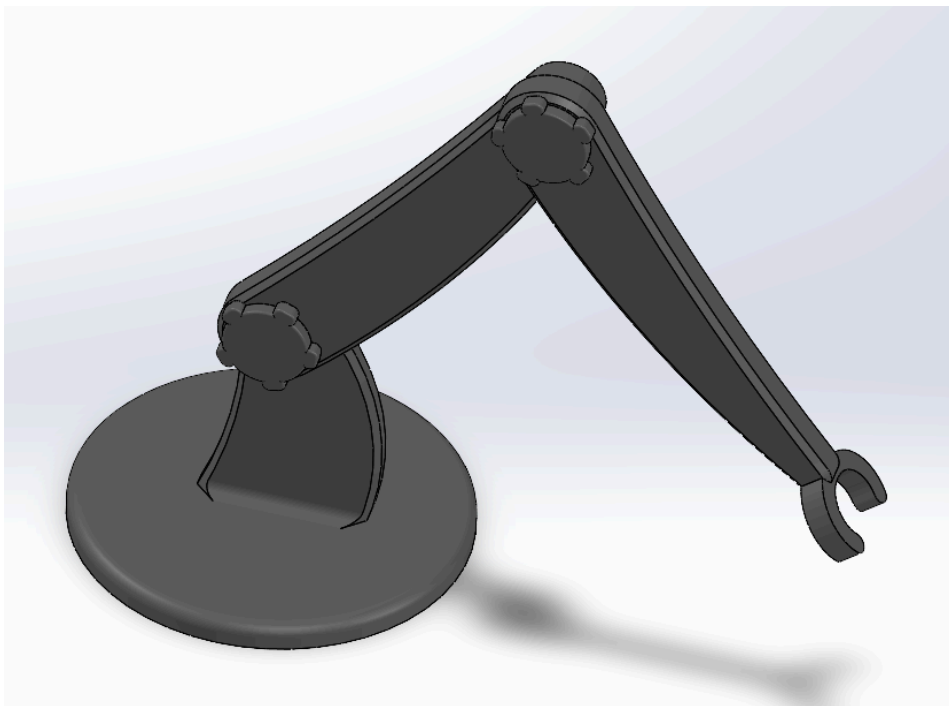


Nắp xe

(Bộ phận dùng để kết nối khung xe và tay máy)



Động cơ



Tay máy 2 bậc tự do quay

### + Phân tích động học:

- Robot này sử dụng ba bánh omni bố trí đối xứng nhau (góc  $120^\circ$  giữa các bánh) để tạo ra chuyển động toàn phương.

- Các biến số chính:

- $\mathbf{V} = [v_x, v_y, \omega]^T$ : Vector vận tốc của robot trong hệ tọa độ toàn cục.
  - $v_x$ : Vận tốc dọc trục X.
  - $v_y$ : Vận tốc dọc trục Y.
  - $\omega$ : Vận tốc góc (quay quanh trục Z).
- $\omega_i$  ( $i = 1, 2, 3$ ): Tốc độ góc của các bánh xe.
- $R$ : Bán kính bánh xe.
- $L$ : Khoảng cách từ tâm robot đến bánh xe.

- Động học thuận xác định vận tốc robot khi biết tốc độ góc của các bánh. Ta giả sử hệ tọa độ đặt tại tâm robot, với các bánh xe bố trí như sau:

1. Bánh 1: Góc  $0^\circ$  so với trục x.
2. Bánh 2: Góc  $120^\circ$ .
3. Bánh 3: Góc  $-120^\circ$ .

Vận tốc tiếp tuyến của mỗi bánh xe được xác định bởi phương trình:

$$\mathbf{V} = R * \boldsymbol{\omega}$$

Chiếu vận tốc từng bánh lên trục X và Y, ta có hệ phương trình:

$$\begin{bmatrix} \sin 0^\circ & -\cos 0^\circ & -L \\ \sin 120^\circ & -\cos 120^\circ & -L \\ \sin(-120^\circ) & -\cos(-120^\circ) & -L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = R \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & -L \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & -L \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & -L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = R \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

- Động học nghịch giúp xác định tốc độ bánh xe khi biết vận tốc của robot.  
 Từ phương trình trên, ta lấy ma trận nghịch đảo:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 0 & -1 & -L \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & -L \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & -L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

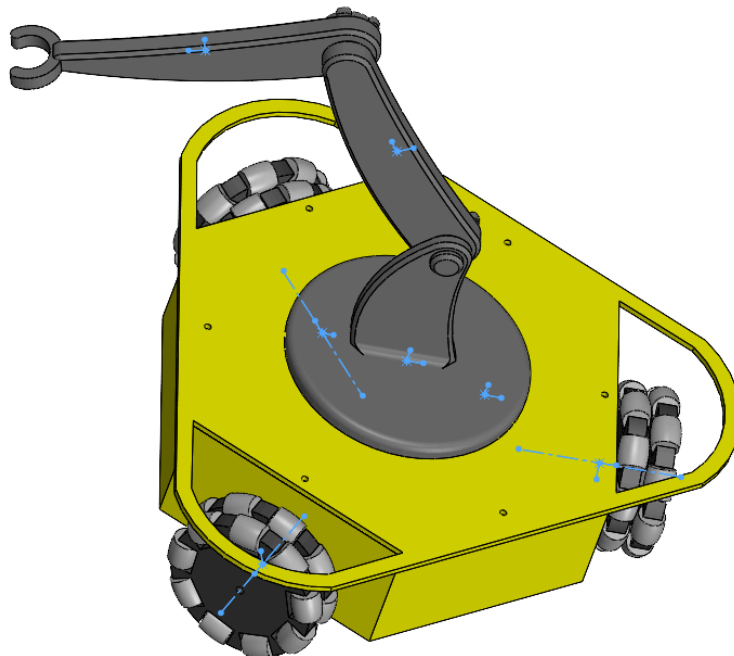
Từ đó, tính tốc độ từng bánh:

$$\omega_1 = \frac{-v_y - L\omega}{R}$$

$$\omega_2 = \frac{\frac{\sqrt{3}}{2}v_x + \frac{1}{2}v_y - L\omega}{R}$$

$$\omega_3 = \frac{-\frac{\sqrt{3}}{2}v_x + \frac{1}{2}v_y - L\omega}{R}$$

- Ta đặt các hệ trục tọa độ như hình vẽ:



## 2. Cấu trúc của file urdf, liên kết của các link, các cảm biến

+ Base link:

```
<link name="base_link">
  <inertial>
    <origin xyz="-0.0514950390834328 0.00372680362943543
-0.0212713777766178" rpy="0 0 0" />
    <mass value="5.546524151294" />
    <inertia ixx="0.0390256274086559"
ixy="7.53990510164341E-17" ixz="-9.49819895659914E-18"
iyy="0.0390256274084877" iyz="-4.52445032425781E-14"
izz="0.0727431794136434" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh
filename="package://robot_kin/meshes/base_link.STL" />
    </geometry>
    <material name="">
      <color rgba="0.647058823529412 0.619607843137255
0.588235294117647 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh
filename="package://robot_kin/meshes/base_link.STL" />
    </geometry>
  </collision>
</link>
```

+ Các Link: omni1, omni2, omni3, link1, link2, link3

```
<link name="omni_1">
  <inertial>
    <origin xyz="2.10248485288389E-15 -4.64905891561784E-16
-0.01500000000000003" rpy="0 0 0" />
    <mass value="0.208733977688544" />
    <inertia ixx="5.75297014634535E-05"
ixy="-3.89635155736978E-20" ixz="5.02608940799584E-19"
iyy="5.75297014634539E-05" iyz="-1.7427007015891E-19"
izz="9.13704714530288E-05" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/omni_1.STL"
/>
    </geometry>
    <material name="">
      <color rgba="0.752941176470588 0.752941176470588
0.752941176470588 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/omni_1.STL"
/>
    </geometry>
  </collision>
</link>
```

```
link name="omni_2">
  <inertial>
    <origin xyz="2.51881848711832E-15 -2.0122792321331E-16
-0.015000000000000004" rpy="0 0 0" />
    <mass value="0.208733977688543" />
    <inertia ixx="5.75297014634536E-05"
ixy="1.35525271560688E-20" ixz="5.37279608401087E-19"
iyy="5.75297014634537E-05" iyz="-3.87797460613561E-19"
izz="9.13704714530288E-05" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/omni_2.STL"
/>
    </geometry>
    <material name="">
      <color rgba="0.752941176470588 0.752941176470588
0.752941176470588 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/omni_2.STL"
/>
    </geometry>
  </collision>
</link>
```



```

link name="omni_3">
  <inertial>
    <origin xyz="2.04697370165263E-15 -4.57966997657877E-16
-0.01500000000000004" rpy="0 0 0" />
    <mass value="0.208733977688544" />
    <inertia ixx="5.75297014634535E-05"
ixy="-5.59041745187838E-20" ixz="5.01350143462744E-19"
iyy="5.75297014634538E-05" iyz="-2.67520268147213E-19"
izz="9.13704714530288E-05" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/omni_3.STL"
/>
    </geometry>
    <material name="">
      <color rgba="0.752941176470588 0.752941176470588
0.752941176470588 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/omni_3.STL"
/>
    </geometry>
  </collision>
</link>

```

```

<link name="link_1">
  <inertial>
    <origin xyz="-0.00197622872489094 -4.49272203594166E-05
0.00633251365290859" rpy="0 0 0" />
    <mass value="0.214783356206923" />

```

```

    <inertia ixx="0.000365790344438595"
ixy="1.90730142866856E-08" ixz="1.46206270755554E-05"
iyy="0.000380010525298812" iyz="8.07220952141231E-07"
izz="0.000475014888435784" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/link_1.STL"
/>
    </geometry>
    <material name="">
      <color rgba="0.498039215686275 0.498039215686275
0.498039215686275 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/link_1.STL"
/>
    </geometry>
  </collision>
</link>

```

```

<link name="link_2">
  <inertial>
    <origin xyz="0.0734879371385836 -0.00334262643425878
0.00783116598654759" rpy="0 0 0" />
    <mass value="0.108863827343016" />
    <inertia ixx="1.52983726752256E-05"
ixy="3.25169315859978E-06" ixz="2.96831173668244E-06"
iyy="0.000349212764999917" iyz="-1.20507616431351E-07"
izz="0.000359580214053703" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />

```

```

    <geometry>
      <mesh filename="package://robot_kin/meshes/link_2.STL"
/>

    </geometry>
    <material name="">
      <color rgba="0.498039215686275 0.498039215686275
0.498039215686275 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/link_2.STL"
/>
    </geometry>
  </collision>
</link>

```

```

<link name="link_3">
  <inertial>
    <origin xyz="0.0679394800286911 -0.00254732956132153
0.00791117403147978" rpy="0 0 0" />
    <mass value="0.0922267153628876" />
    <inertia ixx="1.11543901229101E-05"
ixy="3.03180485829024E-06" ixz="2.56993305226668E-06"
iyy="0.00031313631090358" iyz="-1.19027339136059E-07"
izz="0.000318250360418558" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://robot_kin/meshes/link_3.STL"
/>
    </geometry>
    <material name="">

```

```

        <color rgba="0.498039215686275 0.498039215686275
0.498039215686275 1" />
    </material>
</visual>
<collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
        <mesh filename="package://robot_kin/meshes/link_3.STL"
/>
    </geometry>
</collision>
</link>

```

## + Các Joint:

```

joint name="joint_omni_1" type="continuous">
    <origin xyz="-0.136496944367027 -0.143500815062323
-0.0472051656727379" rpy="-1.57079632679489 -1.18691023115661
2.61799387799149" />
    <parent link="base_link" />
    <child link="omni_1" />
    <axis xyz="0 0 1" />
    <dynamics damping="0.1" friction="0.1" />
</joint>

```

```

joint name="joint_omni_2" type="continuous">
    <origin xyz="-0.136496944367024 0.150954422321192
-0.0472051656727389" rpy="-1.5707963267949 0.532099877009558
0.523598775598294" />
    <parent link="base_link" />
    <child link="omni_2" />
    <axis xyz="0 0 1" />
    <dynamics damping="0.1" friction="0.1" />
</joint>

```

```

joint name="joint_omni_3" type="continuous">
  <origin xyz="0.118508771484474 0.00372680362943003
-0.0472051656727379" rpy="-1.5707963267949 0.854297069502276
-1.57079632679491" />
  <parent link="base_link" />
  <child link="omni_3" />
  <axis xyz="0 0 1" />
  <dynamics damping="0.1" friction="0.1" />
</joint>

```

+ Cấu hình camera:

```

!-- Camera Link -->
<link name="camera_link">
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <mass value="0.1"/>
    <inertia ixx="0.001" ixy="0" ixz="0" iyy="0.001" iyz="0"
izz="0.001"/>
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <box size="0.01 0.01 0.01" />
    </geometry>
    <material name="Blue">
      <color rgba="0 0 1 1"/>
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <box size="0.05 0.05 0.05"/>
    </geometry>
  </collision>
</link>

```

```

<!-- Fixed Joint to attach Camera to Link 3 -->
<joint name="camera_joint" type="fixed">
  <origin xyz="0 0 0.05" rpy="0 0 0"/>
  <parent link="link_3"/>
  <child link="camera_link"/>
</joint>

```

+ Plugin camera:

```

<!-- Gazebo Camera Sensor -->
<gazebo reference="camera_link">
  <sensor type="camera" name="camera_sensor">
    <always_on>true</always_on>
    <update_rate>30</update_rate>
    <camera>
      <horizontal_fov>1.3962634</horizontal_fov>
      <image>
        <width>640</width>
        <height>480</height>
        <format>R8G8B8</format>
      </image>
      <clip>
        <near>0.1</near>
        <far>100</far>
      </clip>
    </camera>
    <plugin name="gazebo_ros_camera_controller"
filename="libgazebo_ros_camera.so">
      <robotNamespace>/robot_kin</robotNamespace>
      <cameraName>camera</cameraName>
      <imageTopicName>image_raw</imageTopicName>
      <cameraInfoTopicName>camera_info</cameraInfoTopicName>
      <frameName>camera_link</frameName>
      <updateRate>30</updateRate>
    </plugin>
  </sensor>
</gazebo>

```

+ Trong file URDF:

Mỗi link đại diện cho một bộ phận của robot và có thể chứa các thuộc tính:

- **Inertial**: Thông tin quán tính (khối lượng, ma trận quán tính) giúp mô phỏng động lực học.
- **Visual**: Thông tin hiển thị, bao gồm hình dạng 3D hoặc màu sắc.
- **Collision**: Thông tin về va chạm, dùng để xác định vùng ảnh hưởng khi mô phỏng vật lý.

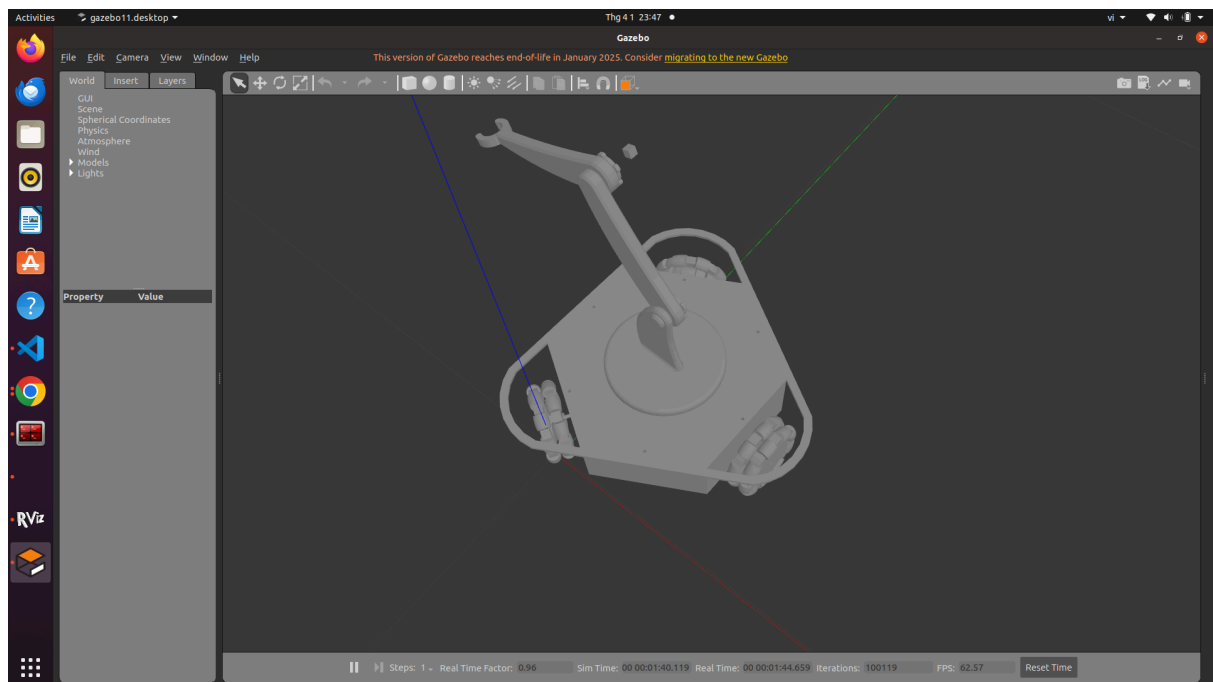
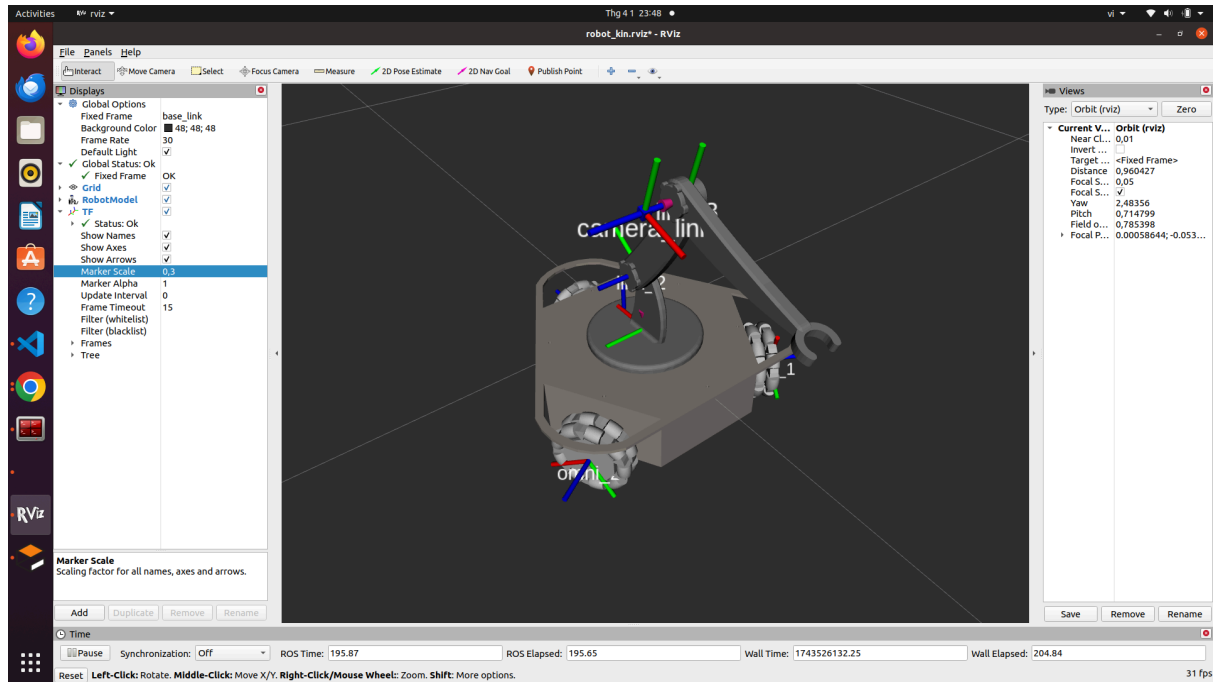
Các **joint** kết nối các **link** lại với nhau. Một **joint** có thể có các loại:

- **fixed** (cố định)
- **continuous** (xoay liên tục)
- **revolute** (xoay có giới hạn)
- **prismatic** (trượt tuyến tính)
- **floating** (tự do 6 bậc)
- **planar** (di chuyển trên mặt phẳng)

Mỗi joint có các thuộc tính:

- **origin**: Vị trí và góc quay của khớp.
- **parent** và **child**: Xác định hai link được kết nối.
- **axis**: Trục chuyển động của khớp.
- **limit**: (chỉ dùng với **revolute** và **prismatic**) Giới hạn chuyển động.
- **dynamics**: Damping và ma sát.

### 3. Kết quả chạy mô phỏng



- Hình ảnh Robot hiển thị đầy đủ chi tiết trong 2 phần mềm Gazebo và Rviz
- Camera cũng đã hiển thị nhưng chưa cấu hình được