

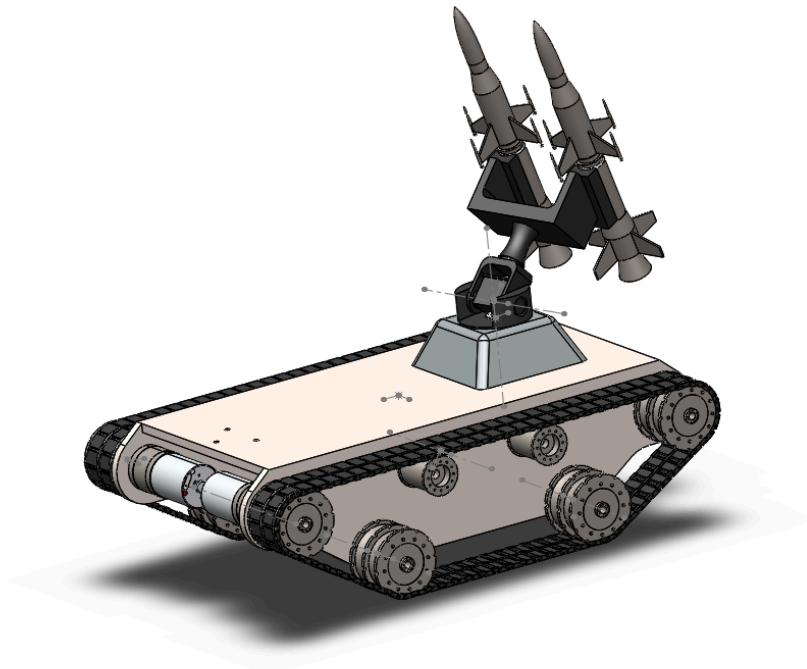
BÁO CÁO THỰC HÀNH GIỮA KỲ LẬP TRÌNH ROBOT VỚI ROS

Giảng viên: TS. Lê Xuân Lực, KS. Dương Văn Tân
Sinh viên: Nguyễn Văn Nam - 22027523

Đề tài: Thiết kế, mô phỏng Robot sử dụng bánh tank, 3 sensor (Lidar, Camera, GPS) , 2 khớp tay máy Rotate

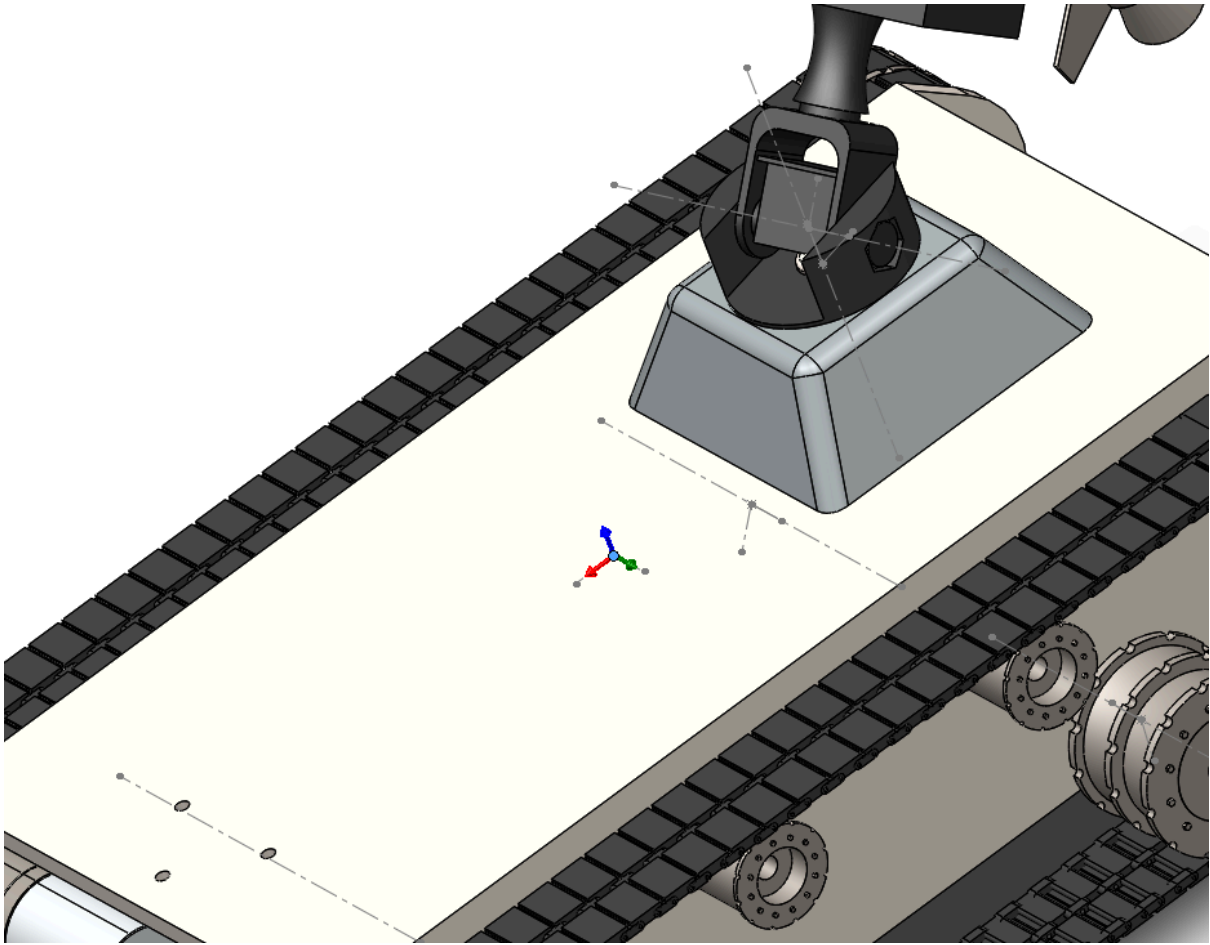
I. Thiết kế

1. Thiết kế Solidwork

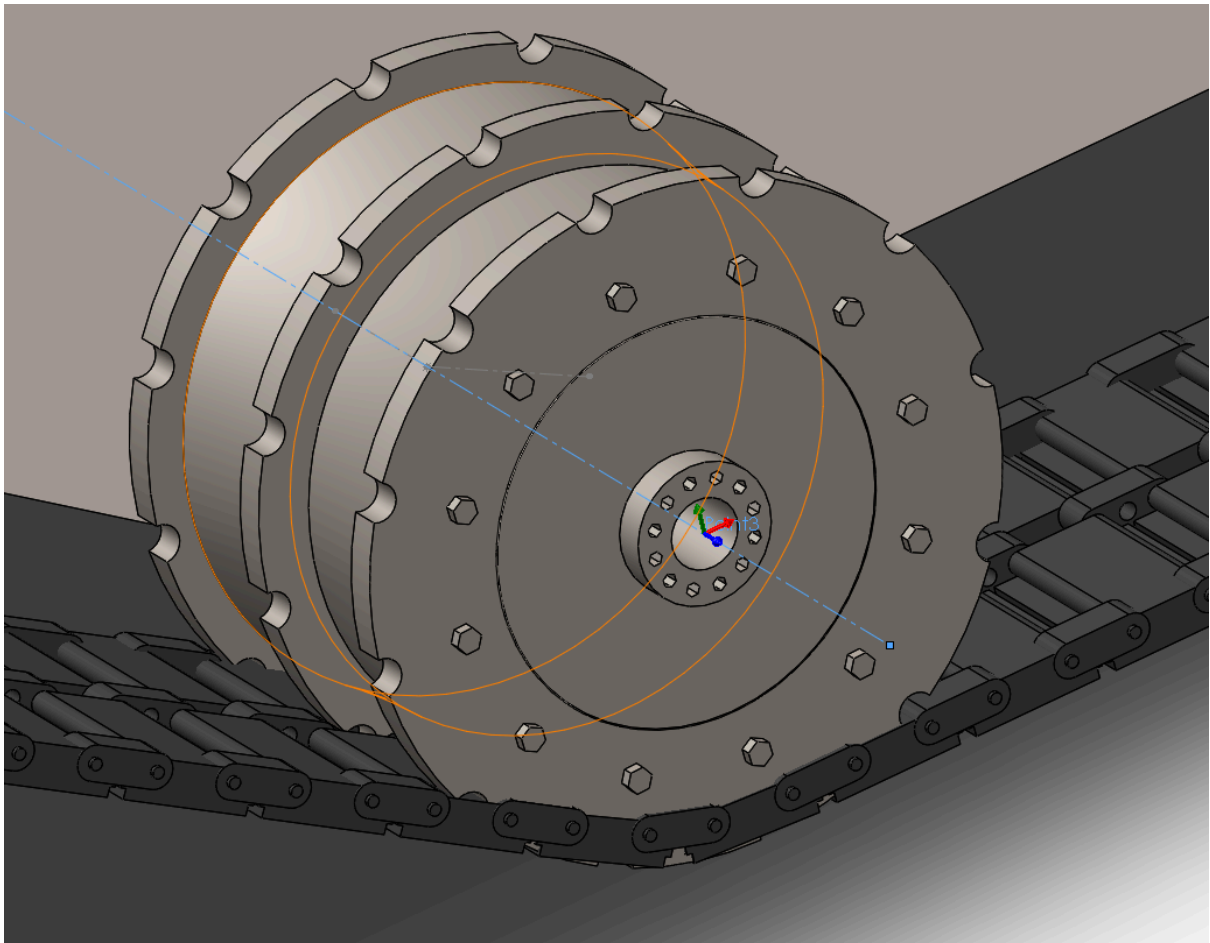


2. Cách đặt trục và xuất URDF

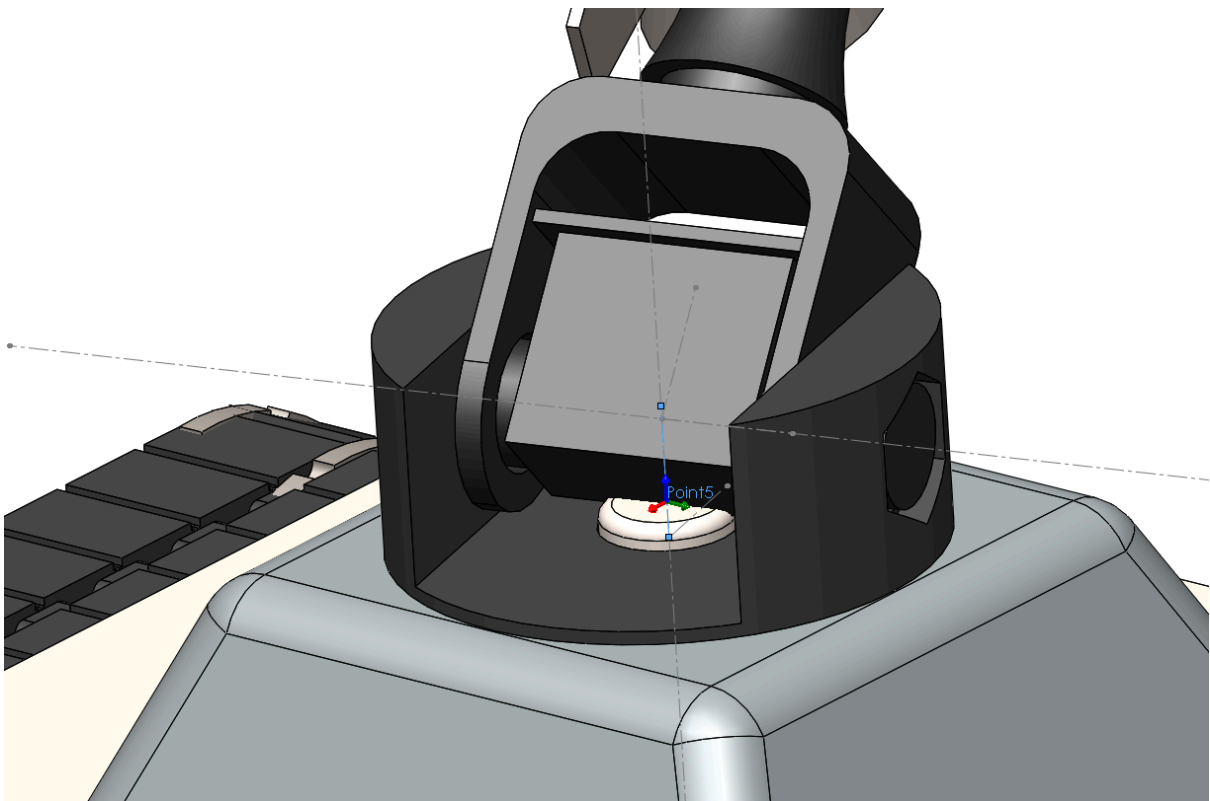
- Trục đỏ: X
- Trục xanh dương: Z
- Trục xanh lá: Y
- a. Base_link



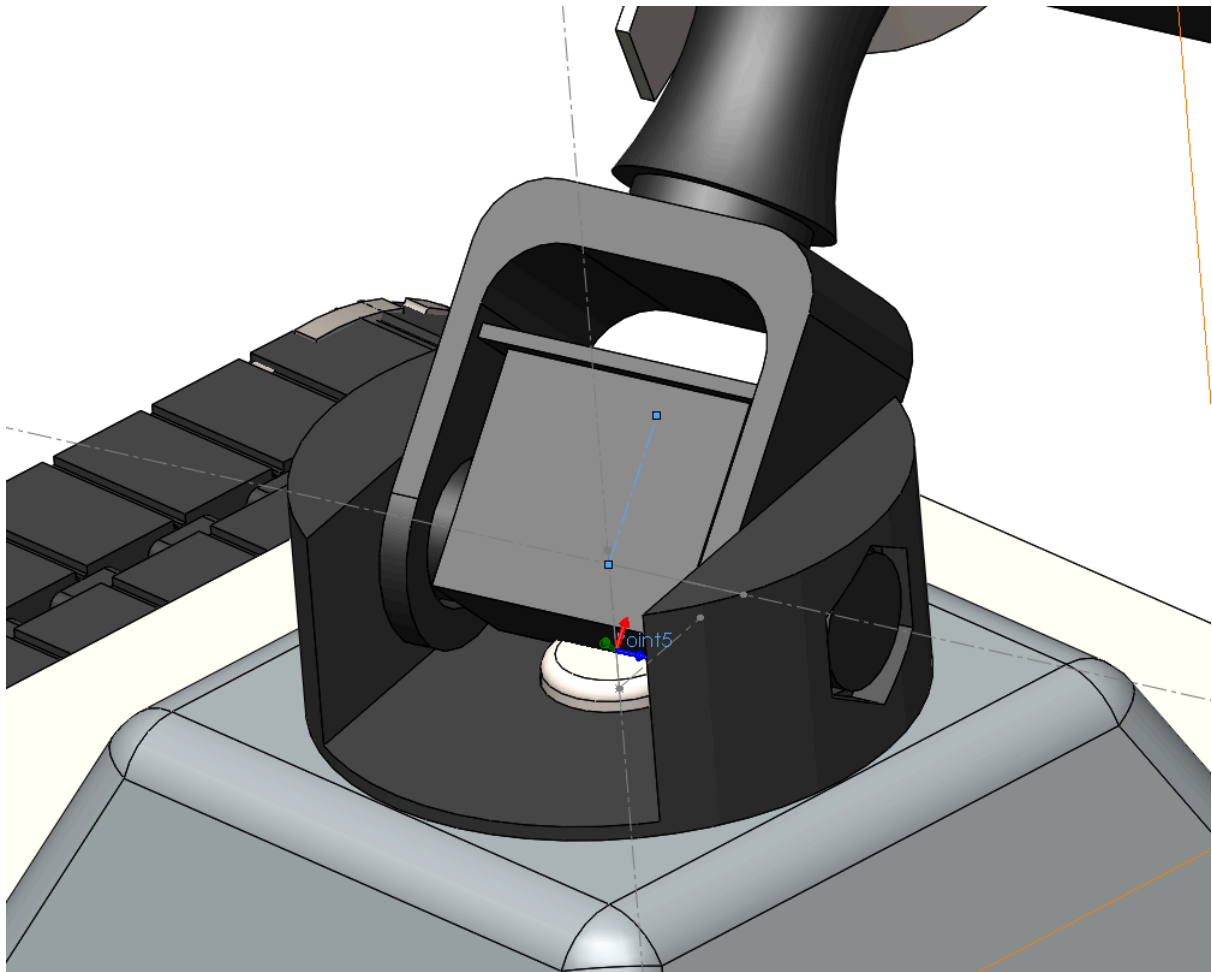
b. Bánh xe



c. Link 1

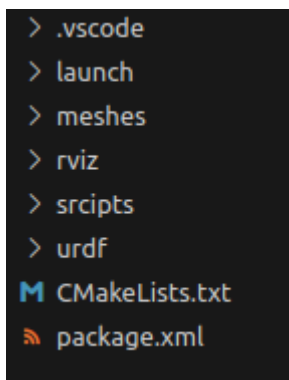


d. Link 2



II. Cấu trúc Mô phỏng

1. Cấu trúc Package



- File launch: File chứa các chương trình chạy chính: display rviz, gazebo, main.
- File meshes: File chứa các file STL của model khi xuất sang URDF
- File rviz: Lưu lại config để mở lại sau khi đã setup trong rviz và gazebo
- File scripts: File code bằng C++ để control
- File urdf: chứa model nguyên bản từ Solidwork sang

- File Xacro: Chia nhỏ URDF và cấu hình thêm cho dễ làm việc

2. Cấu trúc file Xacro

a. File Model.xacro

Đây là file thường chứa mô tả tổng quát của robot, bao gồm các thông số về hình dạng, kích thước, khối lượng, thông tin va chạm (collision), hình ảnh (visual) và khối lượng (inertial).

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
  <xacro:macro name="model">
    <xacro:property name="r200_cam_rgb_px" value="0.005"/>
    <xacro:property name="r200_cam_rgb_py" value="0.018"/>
    <xacro:property name="r200_cam_rgb_pz" value="0.013"/>
    <xacro:property name="r200_cam_depth_offset" value="0.01"/>
    <link name="base_link">
      <inertial>
        <origin xyz="-0.00861750035525078 1.29855654087783E-05 -0.0142636839736506" rpy="0 0 0" />
        <mass value="4.6363793056219" />
        <inertia ixx="0.00845760819479726" ixy="-4.14692026594335E-10" ixz="0.000979173377282238"
          iyy="0.0175509925437596" iyz="1.9027397639913E-10" izz="0.0230329110525293" />
      </inertial>
      <visual>
        <origin xyz="0 0 0" rpy="0 0 0" />
        <geometry>
          <mesh filename="package://tank_rocket/meshes/base_link.STL" />
        </geometry>
        <material name="white">
          <color rgba="1.0 1.0 1.0 1.0" />
        </material>
      </visual>
      <collision>
        <origin xyz="0 0 0" rpy="0 0 0" />
        <geometry>
          <mesh filename="package://tank_rocket/meshes/base_link.STL" />
        </geometry>
      </collision>
    </link>

    <!-- Right Front Wheel -->
    <link name="right_front">
      <inertial>
        <origin xyz="-3.46944695195361E-17 -3.46944695195361E-17 0.0150066275369927" rpy="0 0 0" />
        <mass value="0.16560407234294" />
        <inertia ixx="2.27667842545529E-05" ixy="-1.14502982470056E-20" ixz="-2.02571919411905E-21"
          iyy="2.27667842545529E-05" iyz="7.60563347761972E-21" izz="2.44403723155893E-05" />
      </inertial>
      <visual>
        <origin xyz="0 0 0" rpy="0 0 0" />
        <geometry>
          <mesh filename="package://tank_rocket/meshes/right_front.STL" />
        </geometry>
        <material name="silver">
          <color rgba="0.7725490196078432 0.788235294117647 0.7803921568627451 1.0" />
        </material>
      </visual>
      <collision>
```

b. File Gazebo.xacro

File này mở rộng mô tả từ model.xacro bằng cách thêm các yếu tố dành riêng cho môi trường mô phỏng Gazebo, như thông số động lực học, các plugin điều khiển, cảm biến mô phỏng và các thiết lập vật lý khác.

```

link_0.xacro / xacro / gazebo / gazebo
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
  <xacro:macro name="gazebo">
    <xacro:arg name="camera_visual" default="false"/>
    <xacro:arg name="laser_visual" default="false"/>

    <gazebo reference="base_link">
      <kp>100000.0</kp>
      <kd>100.0</kd>
      <mu1>1.5</mu1>
      <mu2>1.5</mu2>
      <fdirl>1 0 0</fdirl>
      <maxVel>1.0</maxVel>
      <minDepth>0.001</minDepth>
      <material>Gazebo/White</material>
    </gazebo>

    <gazebo reference="link_1">
      <kp>100000.0</kp>
      <kd>100.0</kd>
      <mu1>1.5</mu1>
      <mu2>1.5</mu2>
      <fdirl>1 0 0</fdirl>
      <maxVel>1.0</maxVel>
      <minDepth>0.001</minDepth>
      <material>Gazebo/Silver</material>
    </gazebo>

    <gazebo reference="link_2">
      <kp>100000.0</kp>
      <kd>100.0</kd>
      <mu1>1.5</mu1>
      <mu2>1.5</mu2>
      <fdirl>1 0 0</fdirl>
      <maxVel>1.0</maxVel>
      <minDepth>0.001</minDepth>
      <material>Gazebo/Silver</material>
    </gazebo>

    <gazebo reference="right_front">
      <kp>100000.0</kp>
      <kd>100.0</kd>
      <mu1>1.5</mu1>
      <mu2>1.5</mu2>
      <fdirl>1 0 0</fdirl>
      <maxVel>1.0</maxVel>
      <minDepth>0.001</minDepth>
      <material>Gazebo/Silver</material>
    </gazebo>
  </macro>
</robot>

```

c. File gps.xacro

File này để cấu hình vật lý cho Sensor GPS và định dạng Plugin cho nó

```

1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3   <xacro:macro name="sensor_gps" params="prefix parent *origin">
4     <joint name="${prefix}_joint" type="fixed">
5       <axis xyz="0 1 0"/>
6       <xacro:insert_block name="origin"/>
7       <parent link="${parent}"/>
8       <child link="${prefix}_base_link"/>
9     </joint>
10    <link name="${prefix}_base_link">
11      <inertial>
12        <mass value="0.001"/>
13        <origin rpy="0 0 0" xyz="0 0 0"/>
14        <inertia ixx="0.0001" ixy="0" ixz="0" iyy="0.000001" iyz="0" izz="0.0001"/>
15      </inertial>
16      <visual>
17        <origin rpy="0 0 0" xyz="0 0 0"/>
18        <material name="orange">
19          <color rgba="1.0 0.4235 0.0392 1.0"/>
20        </material>
21        <geometry>
22          <mesh filename="package://tank_rocket/meshes/antenna_3G016.stl" scale="0.25 0.25 0.25"/>
23        </geometry>
24      </visual>
25      <collision>
26        <origin rpy="0 0 0" xyz="0 0 0"/>
27        <geometry>
28          <mesh filename="package://tank_rocket/meshes/antenna_3G016.stl" scale="0.25 0.25 0.25"/>
29        </geometry>
30      </collision>
31    </link>
32    <gazebo reference="${prefix}_base_link">
33      <material>Gazebo/Orange</material>
34      <gravity>true</gravity>
35    </gazebo>
36    <gazebo>
37      <plugin filename="libhector_gazebo_ros_gps.so" name="${prefix}_controller">
38        <alwaysOn>1</alwaysOn>
39        <updateRate>10</updateRate>
40        <bodyName>${prefix}_base_link</bodyName>
41        <!-- must be the link of the gps device, not the base_link or base_footprint -->
42        <frameId>${prefix}_base_link</frameId>
43        <topicName>gps</topicName>
44        <!-- Appleton Tower coordinates -->
45        <referenceLatitude>55.944400</referenceLatitude>
46        <referenceLongitude>-3.18629</referenceLongitude>
47        <!-- To set heading in ENU orientation (degrees) -->
48        <referenceHeading>90</referenceHeading>
49        <velocityTopicName>gps_velocity</velocityTopicName>

```

d. master.xacro - file này sẽ include toàn bộ các file xacro khác

```

1 <?xml version="1.0"?>
2 <robot name="tank_rocket" xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4   <xacro:include filename="$(find tank_rocket)/urdf/xacro/model.xacro"/>
5   <xacro:model />
6
7   <xacro:include filename="$(find tank_rocket)/urdf/xacro/gazebo.xacro"/>
8
9   <xacro:include filename="$(find tank_rocket)/urdf/xacro/gps.xacro"/>
10  <xacro:sensor_gps prefix="gps" parent="base_link">
11    <origin xyz="0 0 0" rpy="0 0 0"/>
12  </xacro:sensor_gps>
13
14 </robot>
15

```

3. Cấu trúc File Tank_rocket.launch

a. tank_rocket_model.launch

```

<launch>

  <node pkg="gazebo_ros" type="gzserver" name="gazebo" output="screen">
    <param name="use_sim_time" value="true"/>
  </node>
  <node pkg="gazebo_ros" type="gzclient" name="gazebo_gui" output="screen" />

  <param name="robot_description" command="$(find xacro)/xacro '$(find tank_rocket)/urdf/xacro/master.xacro'" />

  <node name="gazebo_ros" pkg="gazebo_ros" type="spawn_model"
    args="-urdf -param robot_description -model tank_rocket -x 0 -y 0 -z 0.2" />

  <node pkg="robot_state_publisher" type="robot_state_publisher" name="robot_state_publisher">
    <param name="publish_frequency" type="double" value="50.0"/>
  </node>

  <node pkg="joint_state_publisher" type="joint_state_publisher" name="joint_state_publisher" />

  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find tank_rocket)/rviz/tank_rocket.rviz" />
</launch>

```

b. teleop.launch

```

<launch>
  <node pkg="teleop_twist_keyboard" type="teleop_twist_keyboard.py"
    name="control_keyboard" output="screen">
    <remap from="cmd_vel" to="/cmd_vel"/>
  </node>
</launch>

```

4. File Srcrips


```

1  #!/usr/bin/env python3
2  import rospy
3  from geometry_msgs.msg import Twist
4  import sys, select, termios, tty
5  msg = ""
6  Reading from keyboard and Publishing to /cmd_vel!
7  -----
8  Moving around:
9  |   w
10 | a s d
11 |   x
12
13  CTRL-C to quit
14  """
15  speed_linear = 0.5
16  speed_angular = 1.0
17
18  def getKey():
19      tty.setraw(sys.stdin.fileno())
20      rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
21      if rlist:
22          key = sys.stdin.read(1)
23      else:
24          key = ' '
25      termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
26      return key
27
28  if __name__ == "__main__":
29      settings = termios.tcgetattr(sys.stdin)
30      rospy.init_node('control_keyboard')
31      pub = rospy.Publisher('/cmd_vel', Twist, queue_size=10)
32
33      x = 0
34      th = 0
35      try:
36          print(msg)
37          while not rospy.is_shutdown():
38              key = getKey()
39              if key == 'w':
40                  x = 1
41                  th = 0
42              elif key == 's':
43                  x = 0
44                  th = 0
45              elif key == 'x':
46                  x = -1
47                  th = 0
48              elif key == 'a':
49                  x = 0

```

5. Cách thức hoạt động

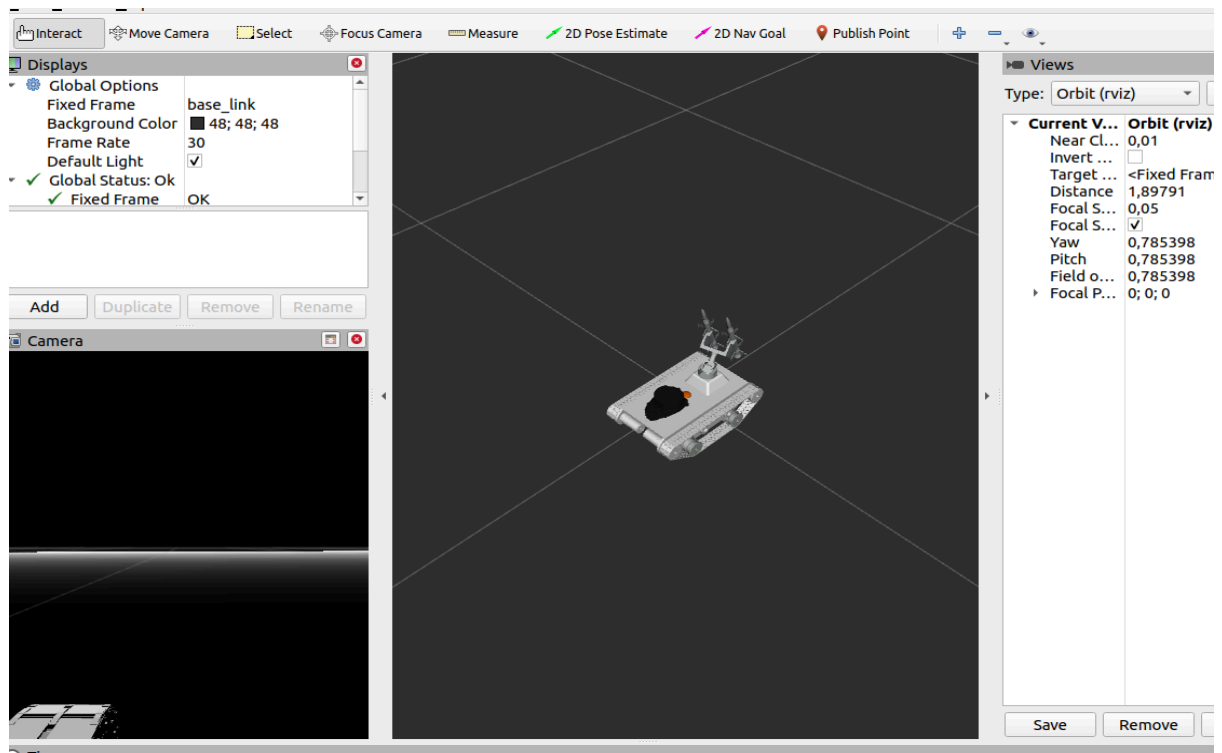
Node Tank_rocket_control.cpp thực hiện nhận giá trị từ bàn phím và publish lên cmd_vel sau đó plugin sẽ nhận dữ liệu rồi đẩy lên gazebo thực hiện chuyển động cho Robot.

III. Kết quả

1. Kết quả cảm biến

Dù đã thêm được các cảm biến vào nhưng em mới chỉ đọc được camera
và GPS thôi ạ! :(

+ Camera



+ GPS

```
header:
  seq: 65
  stamp:
    secs: 6
    nsecs: 700000000
  frame_id: "gps_base_link"
status:
  status: 0
  service: 0
latitude: 55.94439999288689
longitude: -3.1862900091654454
altitude: 0.07852866821301559
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: 66
  stamp:
    secs: 6
    nsecs: 800000000
  frame_id: "gps_base_link"
status:
  status: 0
  service: 0
latitude: 55.94439999292134
longitude: -3.186290009147229
altitude: 0.0785296692930736
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: 67
  stamp:
    secs: 6
    nsecs: 900000000
  frame_id: "gps_base_link"
status:
  status: 0
  service: 0
latitude: 55.944399992948085
longitude: -3.186290009132246
altitude: 0.07852942036902934
position_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
position_covariance_type: 2
---
```

2. Kết quả di chuyển

Hiện tại, khi em bấm nút di chuyển xe nó sẽ bị lật ngược xe, và đây là graph hoạt động:

