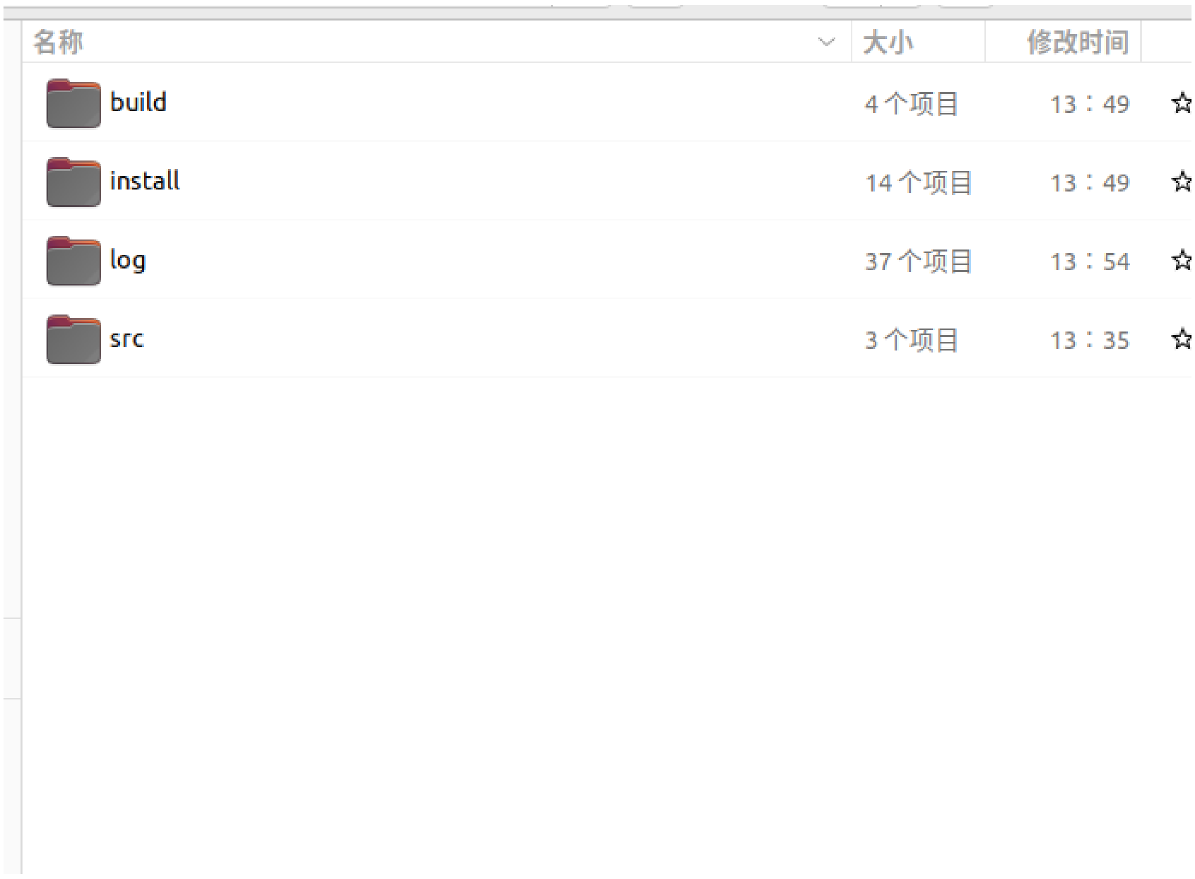
二轮差速小车建图与导航

创建工作空间



名称	大小	修改时间
build	4 个项目	13：49
install	14 个项目	13：49
log	37 个项目	13：54
src	3 个项目	13：35

编写URDF统一机器人建模语言

在src中的urdf文件中编写机器人模型语言，来创建一个二轮差速小车

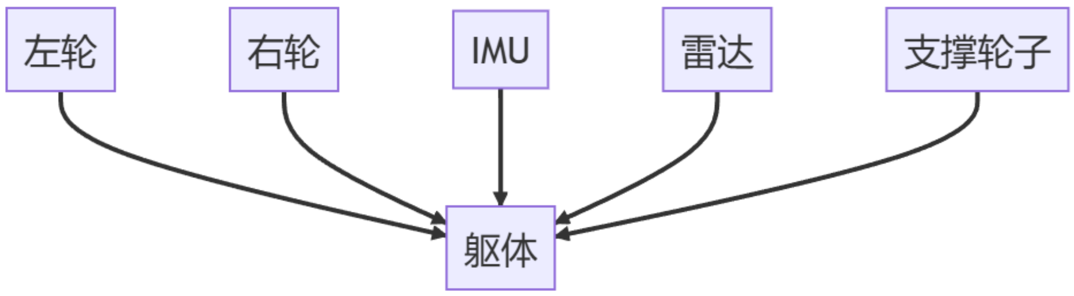
```
<?xml version="1.0"?>
<robot name="fishbot">

  <!-- 定义机器人基础结构的根部 -->
  <!-- Robot Footprint -->
  <link name="base_footprint"/>

  <!-- 连接base_footprint和base_link的固定关节 -->
  <joint name="base_joint" type="fixed">
    <parent link="base_footprint"/>
    <child link="base_link"/>
    <!-- 定义了关节的初始位置和姿态 -->
    <origin xyz="0.0 0.0 0.076" rpy="0 0 0"/>
  </joint>

  <!-- 定义机器人主要的基础链接 -->
  <link name="base_link">
    <!-- 定义视觉属性 -->
    <visual>
      <!-- 指定视觉几何的原点和姿态 -->
      <origin xyz="0 0 0.0" rpy="0 0 0"/>
      <!-- 定义几何形状，这里使用了一个圆柱体 -->
      <geometry>
        <cylinder length="0.12" radius="0.10"/>
      </geometry>
      <!-- 定义材质和颜色 -->
      <material name="blue">
        <color rgba="0.1 0.1 1.0 0.5" />
      </material>
    </visual>
    <!-- 定义碰撞属性，用于物理仿真中的碰撞检测 -->
    <collision>
      <origin xyz="0 0 0.0" rpy="0 0 0"/>
      <geometry>
        <cylinder length="0.12" radius="0.10"/>
      </geometry>
      <material name="blue">
        <color rgba="0.1 0.1 1.0 0.5" />
      </material>
    </collision>
    <!-- 定义惯性属性，用于物理仿真 -->
    <inertial>
      <mass value="0.2"/>
      <inertia ixx="0.0122666" ixy="0" ixz="0" iyy="0.0122666" iyz="0"
izz="0.02"/>
    </inertial>
  </link>

  <!-- 定义激光传感器的链接 -->
  <link name="laser_link">
    <!-- 视觉属性 -->
    <visual>
      <origin xyz="0 0 0" rpy="0 0 0"/>
      <geometry>
        <cylinder length="0.02" radius="0.02"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </visual>
    <!-- 碰撞属性 -->
    <collision>
      <origin xyz="0 0 0" rpy="0 0 0"/>
      <geometry>
        <cylinder length="0.02" radius="0.02"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </collision>
    <!-- 惯性属性 -->
    <inertial>
      <mass value="0.1"/>
      <inertia ixx="0.000190416666667" ixy="0" ixz="0" iyy="0.0001904" iyz="0"
izz="0.00036"/>
    </inertial>
```

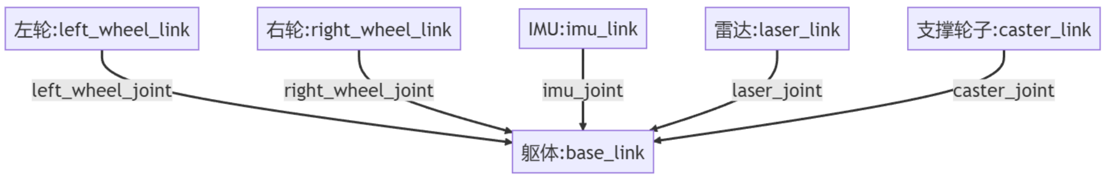
```xml
  </link>

  <!-- 定义激光传感器的关节，固定到基础链接上 -->
  <joint name="laser_joint" type="fixed">
    <parent link="base_link" />
    <child link="laser_link" />
    <!-- 指定激光传感器的位置 -->
    <origin xyz="0 0 0.075" />
  </joint>

  <!-- 定义IMU传感器的链接 -->
  <link name="imu_link">
    <visual>
      <origin xyz="0 0 0.0" rpy="0 0 0"/>
      <geometry>
        <box size="0.02 0.02 0.02"/>
      </geometry>
    </visual>
    <collision>
      <origin xyz="0 0 0.0" rpy="0 0 0"/>
      <geometry>
        <box size="0.02 0.02 0.02"/>
      </geometry>
    </collision>
    <inertial>
      <mass value="0.1"/>
      <inertia ixx="0.000190416666667" ixy="0" ixz="0" iyy="0.0001904" iyz="0"
izz="0.00036"/>
    </inertial>
  </link>

  <!-- 定义IMU传感器的关节，固定到基础链接上 -->
  <joint name="imu_joint" type="fixed">
    <parent link="base_link" />
    <child link="imu_link" />
    <!-- 指定IMU传感器的位置 -->
    <origin xyz="0 0 0.02" />
  </joint>

  <!-- 定义左轮的链接 -->
  <link name="left_wheel_link">
    <visual>
      <origin xyz="0 0 0" rpy="1.57079 0 0"/>
      <geometry>
        <cylinder length="0.04" radius="0.032"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </visual>
    <collision>
      <origin xyz="0 0 0" rpy="1.57079 0 0"/>
      <geometry>
        <cylinder length="0.04" radius="0.032"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </collision>
    <inertial>
      <mass value="0.2"/>
      <inertia ixx="0.000190416666667" ixy="0" ixz="0" iyy="0.0001904" iyz="0"
izz="0.00036"/>
    </inertial>
  </link>

  <!-- 定义右轮的链接 -->
  <link name="right_wheel_link">
    <visual>
      <origin xyz="0 0 0" rpy="1.57079 0 0"/>
      <geometry>
        <cylinder length="0.04" radius="0.032"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </visual>
    <collision>
      <origin xyz="0 0 0" rpy="1.57079 0 0"/>
      <geometry>
        <cylinder length="0.04" radius="0.032"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </collision>
    <inertial>
      <mass value="0.2"/>
      <inertia ixx="0.000190416666667" ixy="0" ixz="0" iyy="0.0001904" iyz="0"
izz="0.00036"/>
    </inertial>
  </link>

  <!-- 定义左轮的连续旋转关节 -->
  <joint name="left_wheel_joint" type="continuous">
    <parent link="base_link" />
    <child link="left_wheel_link" />
    <!-- 指定左轮的位置 -->
    <origin xyz="-0.02 0.10 -0.06" />
    <!-- 指定轮子的旋转轴 -->
    <axis xyz="0 1 0" />
  </joint>

  <!-- 定义右轮的连续旋转关节 -->
  <joint name="right_wheel_joint" type="continuous">
    <parent link="base_link" />
    <child link="right_wheel_link" />
    <!-- 指定右轮的位置 -->
    <origin xyz="-0.02 -0.10 -0.06" />
    <!-- 指定轮子的旋转轴 -->
    <axis xyz="0 1 0" />
  </joint>

  <!-- 定义机器人的脚轮（被动轮）的链接 -->
  <link name="caster_link">
    <visual>
      <origin xyz="0 0 0" rpy="1.57079 0 0"/>
      <geometry>
        <sphere radius="0.016"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </visual>
    <collision>
      <origin xyz="0 0 0" rpy="1.57079 0 0"/>
      <geometry>
        <sphere radius="0.016"/>
      </geometry>
      <material name="black">
        <color rgba="0.0 0.0 0.0 0.5" />
      </material>
    </collision>
    <inertial>
      <mass value="0.02"/>
      <inertia ixx="0.000190416666667" ixy="0" ixz="0" iyy="0.0001904" iyz="0"
izz="0.00036"/>
    </inertial>
  </link>

  <!-- 定义脚轮的固定关节 -->
  <joint name="caster_joint" type="fixed">
    <parent link="base_link" />
    <child link="caster_link" />
    <!-- 指定脚轮的位置 -->
    <origin xyz="0.06 0.0 -0.076" />
    <axis xyz="0 1 0" />
  </joint>

  <!-- Gazebo相关配置 -->
  <gazebo reference="caster_link">
    <material>Gazebo/Black</material>
  </gazebo>

  <!-- 脚轮的物理属性设置 -->
  <gazebo reference="caster_link">
    <mu1 value="0.0"/>
    <mu2 value="0.0"/>
    <kp value="1000000.0" />
    <kd value="10.0" />
    <!-- <fdir1 value="0 0 1"/> -->
  </gazebo>

  <!-- Gazebo插件：差速驱动插件 -->
  <gazebo>
```

```xml
    <plugin name='diff_drive' filename='libgazebo_ros_diff_drive.so'>
      <ros>
        <namespace>/</namespace>
        <remapping>cmd_vel:=cmd_vel</remapping>
        <remapping>odom:=odom</remapping>
      </ros>
      <update_rate>30</update_rate>
      <!-- 关联左右车轮的关节 -->
      <left_joint>left_wheel_joint</left_joint>
      <right_joint>right_wheel_joint</right_joint>
      <!-- 机器人运动学参数 -->
      <wheel_separation>0.2</wheel_separation>
      <wheel_diameter>0.065</wheel_diameter>
      <!-- 控制器的最大扭矩和加速度 -->
      <max_wheel_torque>20</max_wheel_torque>
      <max_wheel_acceleration>1.0</max_wheel_acceleration>
      <!-- 里程计输出配置 -->
      <publish_odom>true</publish_odom>
      <publish_odom_tf>true</publish_odom_tf>
      <publish_wheel_tf>false</publish_wheel_tf>
      <odometry_frame>odom</odometry_frame>
      <robot_base_frame>base_footprint</robot_base_frame>
    </plugin>

    <!-- Gazebo插件：关节状态发布器 -->
    <plugin name="fishbot_joint_state"
filename="libgazebo_ros_joint_state_publisher.so">
      <ros>
        <remapping>~/out:=joint_states</remapping>
      </ros>
      <update_rate>30</update_rate>
      <!-- 指定要发布的关节 -->
      <joint_name>right_wheel_joint</joint_name>
      <joint_name>left_wheel_joint</joint_name>
    </plugin>
  </gazebo>

  <!-- 定义激光传感器的材质 -->
  <gazebo reference="laser_link">
    <material>Gazebo/Black</material>
  </gazebo>

  <!-- 配置IMU传感器 -->
  <gazebo reference="imu_link">
    <sensor name="imu_sensor" type="imu">
      <plugin filename="libgazebo_ros_imu_sensor.so" name="imu_plugin">
        <ros>
          <namespace>/</namespace>
          <remapping>~/out:=imu</remapping>
        </ros>

 <initial_orientation_as_reference>false</initial_orientation_as_reference>
      </plugin>
      <always_on>true</always_on>
      <update_rate>100</update_rate>
```

```
        <visualize>true</visualize>
        <imu>
          <angular_velocity>
            <x>
              <noise type="gaussian">
                <mean>0.0</mean>
                <stddev>2e-4</stddev>
                <bias_mean>0.0000075</bias_mean>
                <bias_stddev>0.0000008</bias_stddev>
              </noise>
            </x>
            <y>
              <noise type="gaussian">
                <mean>0.0</mean>
                <stddev>2e-4</stddev>
                <bias_mean>0.0000075</bias_mean>
                <bias_stddev>0.0000008</bias_stddev>
              </noise>
            </y>
            <z>
              <noise type="gaussian">
                <mean>0.0</mean>
                <stddev>2e-4</stddev>
                <bias_mean>0.0000075</bias_mean>
                <bias_stddev>0.0000008</bias_stddev>
              </noise>
            </z>
          </angular_velocity>
          <linear_acceleration>
            <x>
              <noise type="gaussian">
                <mean>0.0</mean>
                <stddev>1.7e-2</stddev>
                <bias_mean>0.1</bias_mean>
                <bias_stddev>0.001</bias_stddev>
              </noise>
            </x>
            <y>
              <noise type="gaussian">
                <mean>0.0</mean>
                <stddev>1.7e-2</stddev>
                <bias_mean>0.1</bias_mean>
                <bias_stddev>0.001</bias_stddev>
              </noise>
            </y>
            <z>
              <noise type="gaussian">
                <mean>0.0</mean>
                <stddev>1.7e-2</stddev>
                <bias_mean>0.1</bias_mean>
                <bias_stddev>0.001</bias_stddev>
              </noise>
            </z>
          </linear_acceleration>
        </imu>
```

```
      </sensor>
    </gazebo>

    <!-- 配置激光传感器 -->
    <gazebo reference="laser_link">
      <sensor name="laser_sensor" type="ray">
        <always_on>true</always_on>
        <visualize>true</visualize>
        <update_rate>5</update_rate>
        <pose>0 0 0.075 0 0 0</pose>
        <ray>
          <scan>
            <horizontal>
              <samples>360</samples>
              <resolution>1.000000</resolution>
              <min_angle>0.000000</min_angle>
              <max_angle>6.280000</max_angle>
            </horizontal>
          </scan>
          <range>
            <min>0.120000</min>
            <max>3.5</max>
            <resolution>0.015000</resolution>
          </range>
          <noise>
            <type>gaussian</type>
            <mean>0.0</mean>
            <stddev>0.01</stddev>
          </noise>
        </ray>
        <plugin name="laserscan" filename="libgazebo_ros_ray_sensor.so">
          <ros>
            <remapping>~/out:=scan</remapping>
          </ros>
          <output_type>sensor_msgs/LaserScan</output_type>
          <frame_name>laser_link</frame_name>
        </plugin>
      </sensor>
    </gazebo>

</robot>
```





RVIZ2可视化移动机器人模型

```
mkdir launch
touch display_rviz2.launch.py
```

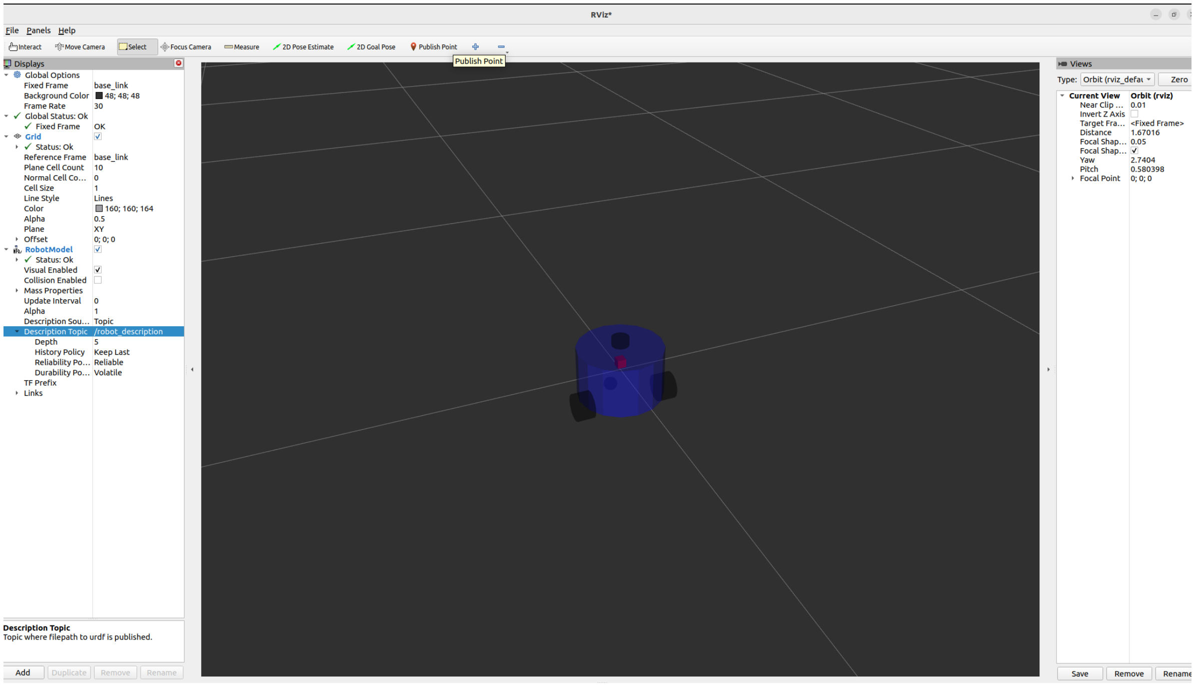
编写launch文件代码

```python
import os
from launch import LaunchDescription
from launch.substitutions import LaunchConfiguration
from launch_ros.actions import Node
from launch_ros.substitutions import FindPackageShare


def generate_launch_description():
    package_name = 'fishbot_description'
    urdf_name = "fishbot_base.urdf"

    ld = LaunchDescription()
    pkg_share = FindPackageShare(package=package_name).find(package_name)
    urdf_model_path = os.path.join(pkg_share, f'urdf/{urdf_name}')

    robot_state_publisher_node = Node(
        package='robot_state_publisher',
        executable='robot_state_publisher',
        arguments=[urdf_model_path]
        )

    joint_state_publisher_node = Node(
        package='joint_state_publisher_gui',
        executable='joint_state_publisher_gui',
        name='joint_state_publisher_gui',
        arguments=[urdf_model_path]
        )

    rviz2_node = Node(
        package='rviz2',
        executable='rviz2',
        name='rviz2',
        output='screen',
        )

    ld.add_action(robot_state_publisher_node)
    ld.add_action(joint_state_publisher_node)
    ld.add_action(rviz2_node)

    return ld
```

修改setup.py

```python
from setuptools import setup
from glob import glob
import os

package_name = 'fishbot_description'

setup(
    name=package_name,
    version='0.0.0',
    packages=[package_name],
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
        (os.path.join('share', package_name, 'launch'),
glob('launch/*.launch.py')),
        (os.path.join('share', package_name, 'urdf'), glob('urdf/**')),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='root',
    maintainer_email='root@todo.todo',
    description='TODO: Package description',
    license='TODO: License declaration',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
        ],
    },
)
```

运行结果



Global Options	
Fixed Frame	base_link
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
✓ Status: Ok	
Reference Frame	base_link
Plane Cell Count	10
Normal Cell Co...	0
Cell Size	1
Line Style	Lines
Color	■ 160; 160; 164
Alpha	0.5
Plane	XY
Offset	0; 0; 0
RobotModel	✓
✓ Status: Ok	
Visual Enabled	✓
Collision Enabled	☐
Mass Properties	
Update Interval	0
Alpha	1
Description Sou...	Topic
Description Topic	/robot_description
Depth	5
History Policy	Keep Last
Reliability Po...	Reliable
Durability Po...	Volatile
TF Prefix	
Links	

Description Topic
Topic where filepath to urdf is published.

Type: Orbit (rviz_defau Zero

Current View	**Orbit (rviz)**
Near Clip ...	0.01
Invert Z Axis	☐
Target Fra...	<Fixed Frame>
Distance	1.67016
Focal Shap...	0.05
Focal Shap...	✓
Yaw	2.7404
Pitch	0.580398
Focal Point	0; 0; 0

使用GAZEBO进行机器人仿真

1. 物理属性的定义

1.1 碰撞检测

碰撞检测是仿真环境中物体交互的基础。通过检测物体之间是否发生碰撞，可以模拟真实世界中的物理现象。URDF文件中使用`<collision>`标签来定义物体的碰撞属性。

- `<origin>`: 定义碰撞几何体的位置和方向。
- `<geometry>`: 定义用于碰撞检测的几何形状。
- `<material>`: 定义碰撞几何体的材料属性，便于在Gazebo中可视化碰撞包围体的形状。

一个典型的碰撞描述如下：

```
<collision>
  <origin xyz="0 0 0.0" rpy="0 0 0"/>
  <geometry>
    <cylinder length="0.12" radius="0.10"/>
  </geometry>
  <material name="blue">
    <color rgba="0.1 0.1 1.0 0.5" />
  </material>
</collision>
```

1.2 旋转惯量

旋转惯量矩阵用于描述物体的惯性，在动力学仿真中非常重要。URDF文件中使用`<inertial>`标签来定义物体的质量和惯性属性。

- `<mass>`: 定义物体的质量。
- `<inertia>`: 定义物体的旋转惯量矩阵，该矩阵是一个对称矩阵，URDF中仅需定义其上三角部分。

一个典型的惯性描述如下：

```
<inertial>
  <mass value="0.2"/>
  <inertia ixx="0.0122666" ixy="0" ixz="0" iyy="0.0122666" iyz="0" izz="0.02"/>
</inertial>
```

1.3 摩擦力和刚性系数

摩擦力和刚性系数在仿真中用于描述物体与地面或其他物体之间的相互作用。URDF文件中使用`<gazebo>`标签并结合`mu1`、`mu2`、`kp`和`kd`参数来配置摩擦力和刚性系数。

- `mu1`、`mu2`: 分别表示摩擦力的两个方向分量。
- `kp`: 刚性系数，控制物体的硬度。
- `kd`: 阻尼系数，控制物体的阻尼效果。

摩擦力和刚性系数的定义如下：

```
<gazebo reference="caster_link">
  <mu1 value="0.0"/>
  <mu2 value="0.0"/>
  <kp value="1000000.0" />
  <kd value="10.0" />
</gazebo>
```

2. 物理属性配置

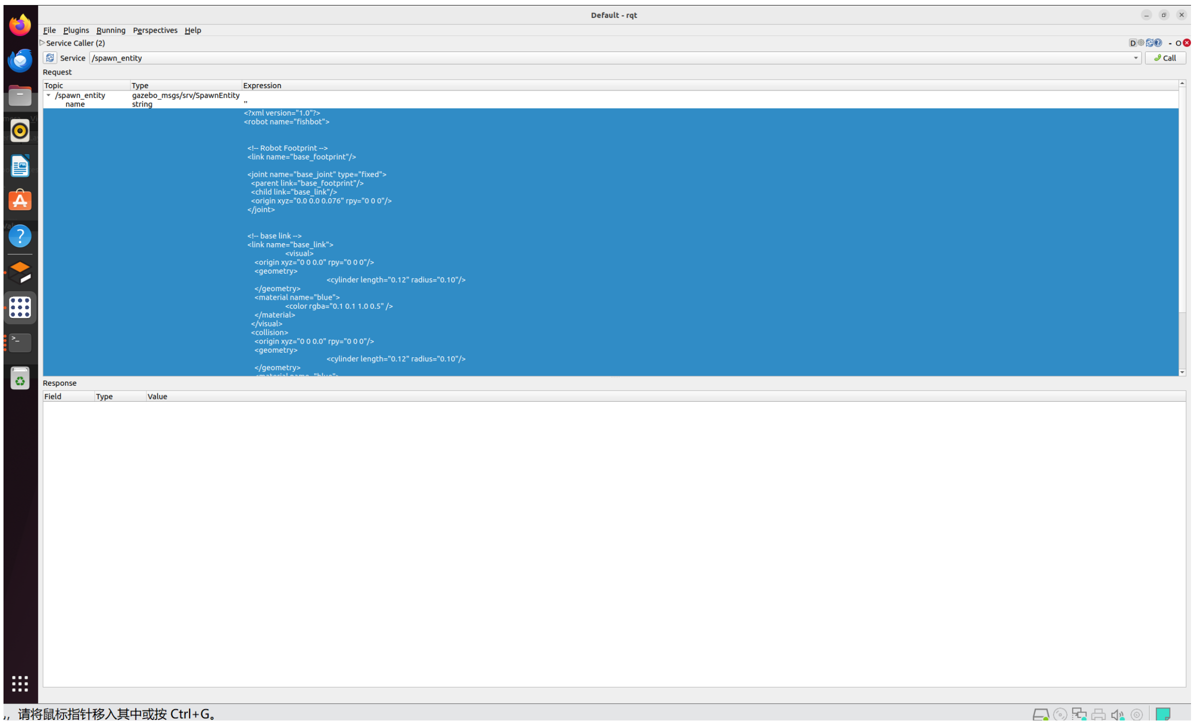
基于上述物理属性定义方法，我们为FishBot机器人模型的各个`link`添加了物理属性。以下是FishBot的`base_link`的完整定义，包括视觉、碰撞和惯性属性。

```
<link name="base_link">
  <visual>
    <origin xyz="0 0 0.0" rpy="0 0 0"/>
    <geometry>
      <cylinder length="0.12" radius="0.10"/>
    </geometry>
    <material name="blue">
      <color rgba="0.1 0.1 1.0 0.5" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0.0" rpy="0 0 0"/>
    <geometry>
      <cylinder length="0.12" radius="0.10"/>
    </geometry>
    <material name="blue">
      <color rgba="0.1 0.1 1.0 0.5" />
    </material>
  </collision>
  <inertial>
    <mass value="0.2"/>
    <inertia ixx="0.0122666" ixy="0" ixz="0" iyy="0.0122666" iyz="0"
izz="0.02"/>
  </inertial>
</link>
```

在这个配置中：

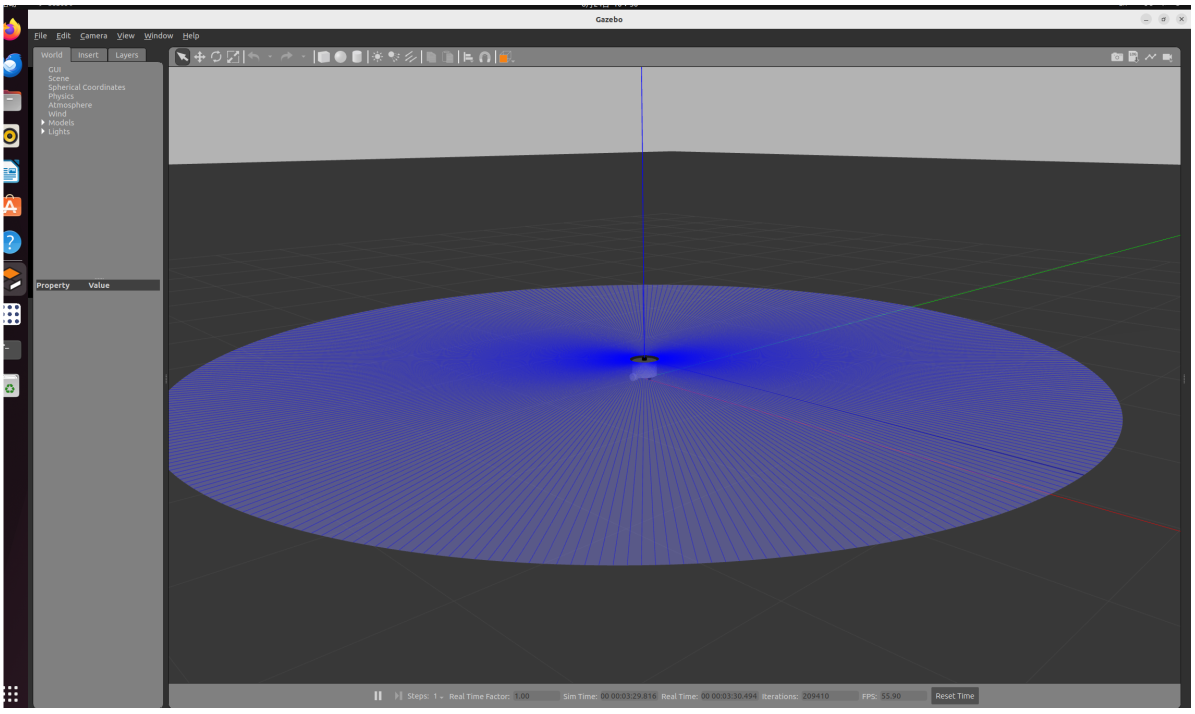
- `visual`标签定义了`base_link`的视觉外观，使用了蓝色透明圆柱体。
- `collision`标签定义了用于碰撞检测的包围体，形状和视觉几何相同。
- `inertial`标签定义了`base_link`的质量（0.2kg）和惯性矩阵，计算得出它的旋转惯性。

使用qrt加载机器人模型



```
Topic            Type                           Expression
/spawn_entity    gazebo_msgs/srv/SpawnEntity
  name           string                         ''
                                                <?xml version="1.0"?>
                                                <robot name="fishbot">


                                                 <!-- Robot Footprint -->
                                                 <link name="base_footprint"/>

                                                 <joint name="base_joint" type="fixed">
                                                  <parent link="base_footprint"/>
                                                  <child link="base_link"/>
                                                  <origin xyz="0.0 0.0 0.076" rpy="0 0 0"/>
                                                 </joint>


                                                 <!-- base link -->
                                                 <link name="base_link">
                                                       <visual>
                                                   <origin xyz="0 0 0.0" rpy="0 0 0"/>
                                                   <geometry>
                                                             <cylinder length="0.12" radius="0.10"/>
                                                   </geometry>
                                                   <material name="blue">
                                                       <color rgba="0.1 0.1 1.0 0.5" />
                                                   </material>
                                                  </visual>
                                                  <collision>
                                                   <origin xyz="0 0 0.0" rpy="0 0 0"/>
                                                   <geometry>
                                                             <cylinder length="0.12" radius="0.10"/>
                                                   </geometry>
```

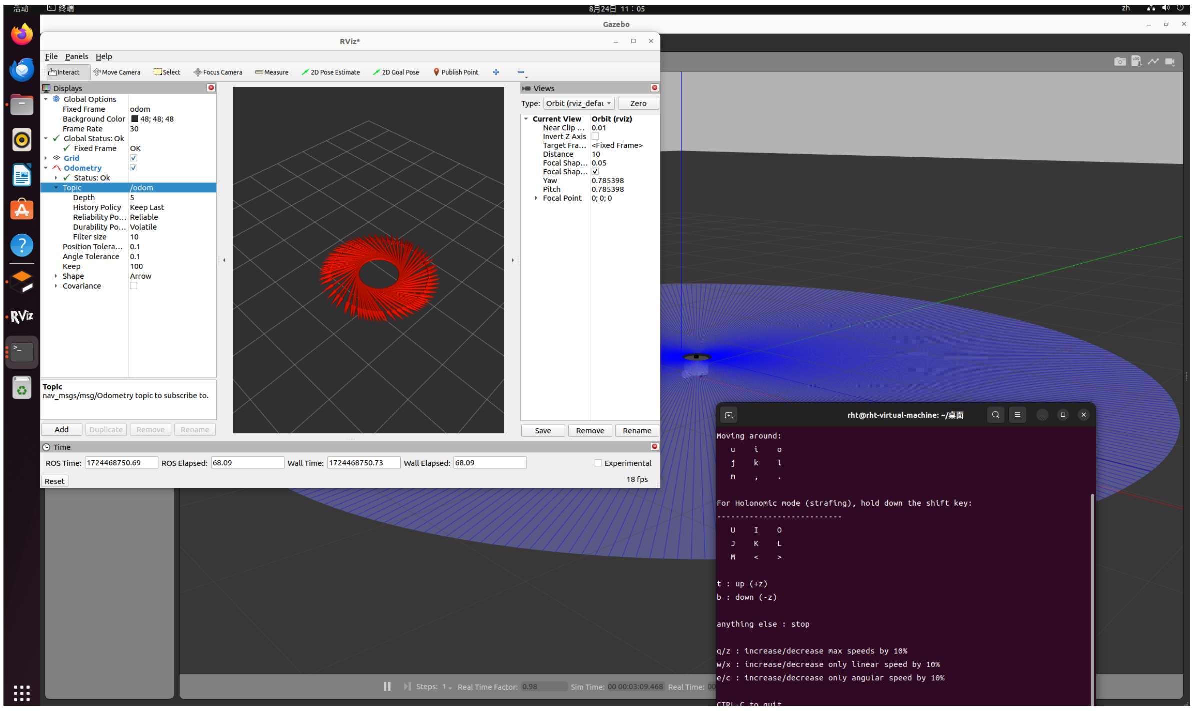
```
Field    Type    Value
```



编写 `gazebo.launch.py` 文件

```python
import os
from launch import LaunchDescription
from launch.actions import ExecuteProcess
from launch_ros.actions import Node
from launch_ros.substitutions import FindPackageShare

def generate_launch_description():
    robot_name_in_model = 'fishbot'
    package_name = 'fishbot_description'
    urdf_name = "fishbot_gazebo.urdf"

    ld = LaunchDescription()
    pkg_share = FindPackageShare(package=package_name).find(package_name)
    urdf_model_path = os.path.join(pkg_share, f'urdf/{urdf_name}')
    gazebo_world_path = os.path.join(pkg_share, 'world/fishbot.world')

    # Start Gazebo server with the specified world file
    start_gazebo_cmd = ExecuteProcess(
        cmd=['gazebo', '--verbose', '-s', 'libgazebo_ros_init.so', '-s',
'libgazebo_ros_factory.so', gazebo_world_path],
        output='screen')

    # Launch the robot
    spawn_entity_cmd = Node(
        package='gazebo_ros',
        executable='spawn_entity.py',
        arguments=['-entity', robot_name_in_model, '-file', urdf_model_path],
        output='screen')

    # Start Robot State publisher
    start_robot_state_publisher_cmd = Node(
        package='robot_state_publisher',
        executable='robot_state_publisher',
        arguments=[urdf_model_path]
    )

    # Launch RViz
    start_rviz_cmd = Node(
        package='rviz2',
        executable='rviz2',
        name='rviz2',
        output='screen',
        # arguments=['-d', default_rviz_config_path]
    )

    # Add all actions to the launch description
    ld.add_action(start_gazebo_cmd)
    ld.add_action(spawn_entity_cmd)
    ld.add_action(start_robot_state_publisher_cmd)
    ld.add_action(start_rviz_cmd)

    return ld
```

两轮差速运动测试



Global Options	
Fixed Frame	odom
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
Odometry	✓
✓ Status: Ok	
Topic	/odom
Depth	5
History Policy	Keep Last
Reliability Po…	Reliable
Durability Po…	Volatile
Filter size	10
Position Tolera…	0.1
Angle Tolerance	0.1
Keep	100
Shape	Arrow
Covariance	☐

Topic
nav_msgs/msg/Odometry topic to subscribe to.

Type: Orbit (rviz_defau | Zero

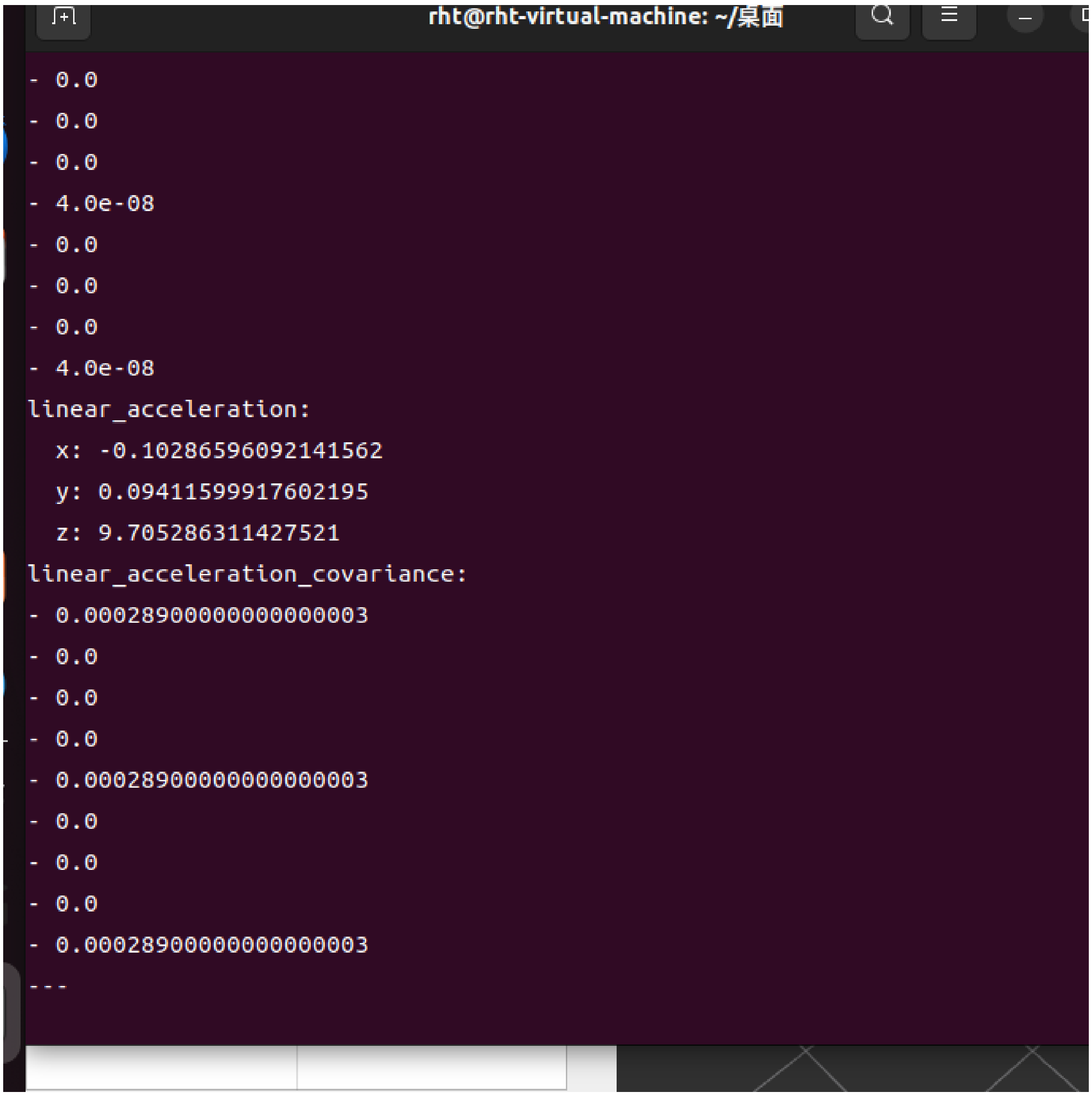
Current View	Orbit (rviz)
Near Clip …	0.01
Invert Z Axis	☐
Target Fra…	<Fixed Frame>
Distance	10
Focal Shap…	0.05
Focal Shap…	✓
Yaw	0.785398
Pitch	0.785398
Focal Point	0; 0; 0

Time
ROS Time: 1724468750.69 ROS Elapsed: 68.09 Wall Time: 1724468750.73 Wall Elapsed: 68.09 ☐ Experimental
18 fps

rht@rht-virtual-machine: ~/桌面

```
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%

CTRL-C to quit
```

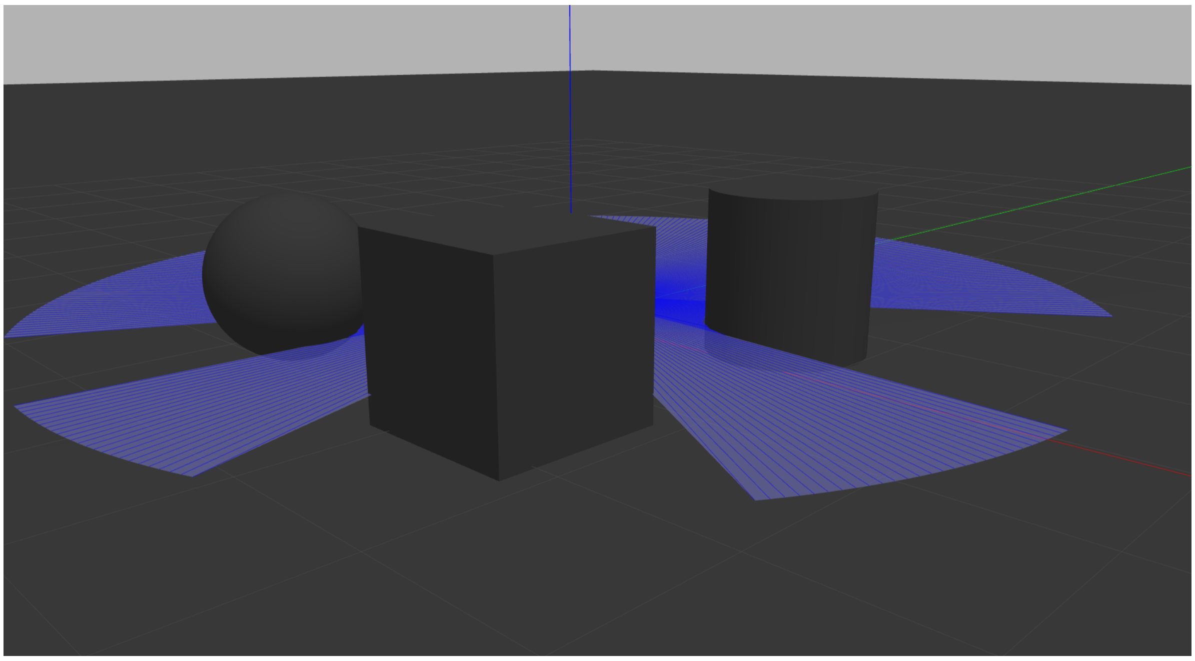
imu测试

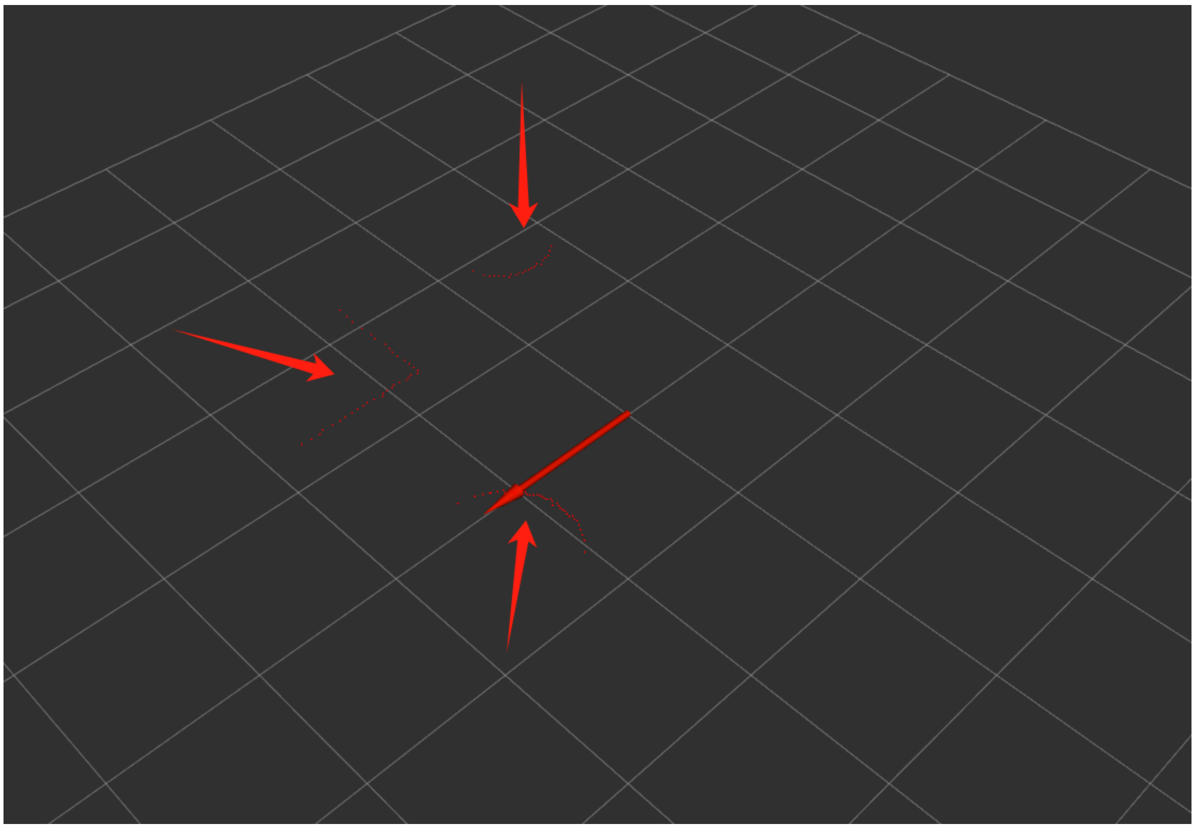


rht@rht-virtual-machine: ~/桌面

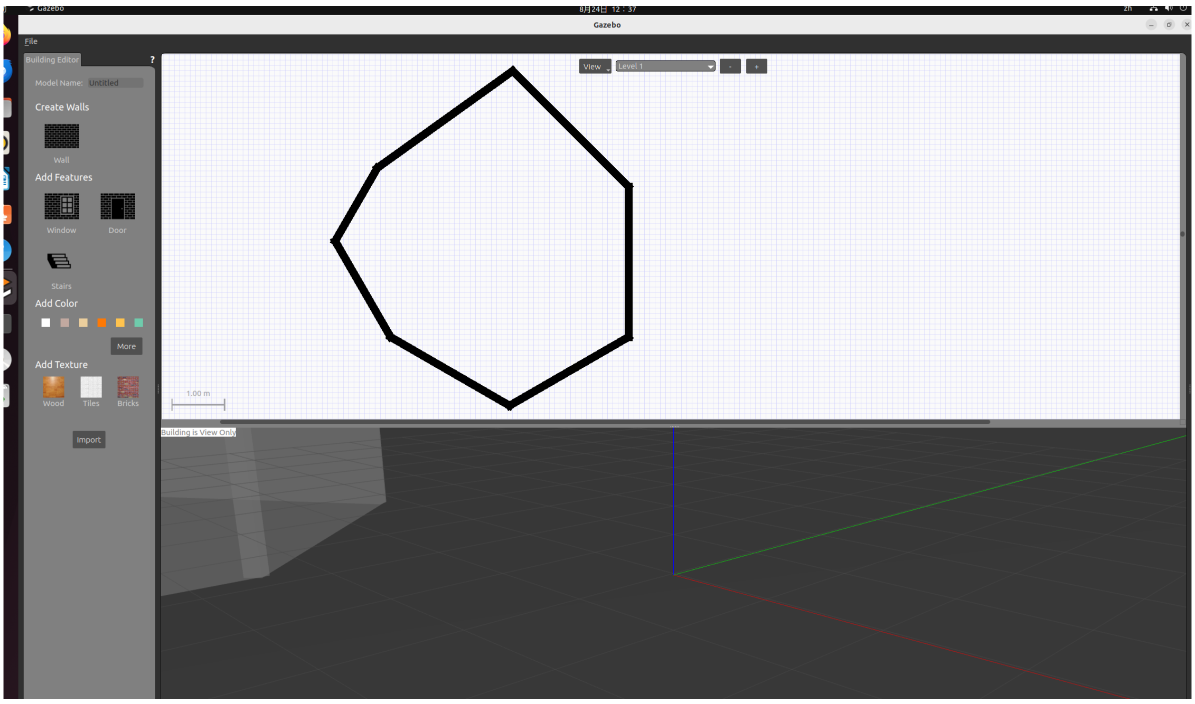
```
- 0.0
- 0.0
- 0.0
- 4.0e-08
- 0.0
- 0.0
- 0.0
- 4.0e-08
linear_acceleration:
  x: -0.10286596092141562
  y: 0.09411599917602195
  z: 9.705286311427521
linear_acceleration_covariance:
- 0.00028900000000000003
- 0.0
- 0.0
- 0.0
- 0.00028900000000000003
- 0.0
- 0.0
- 0.0
- 0.00028900000000000003
---
```

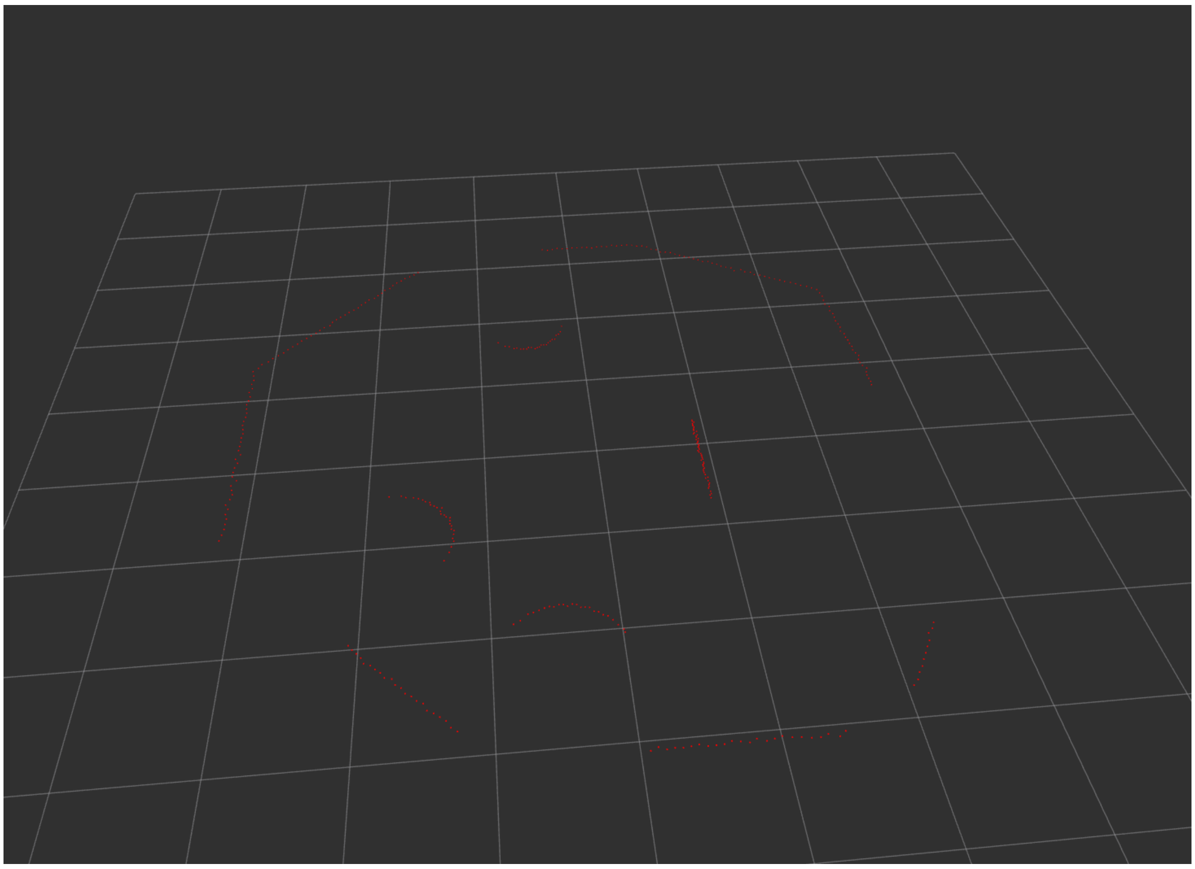
激光雷达测试

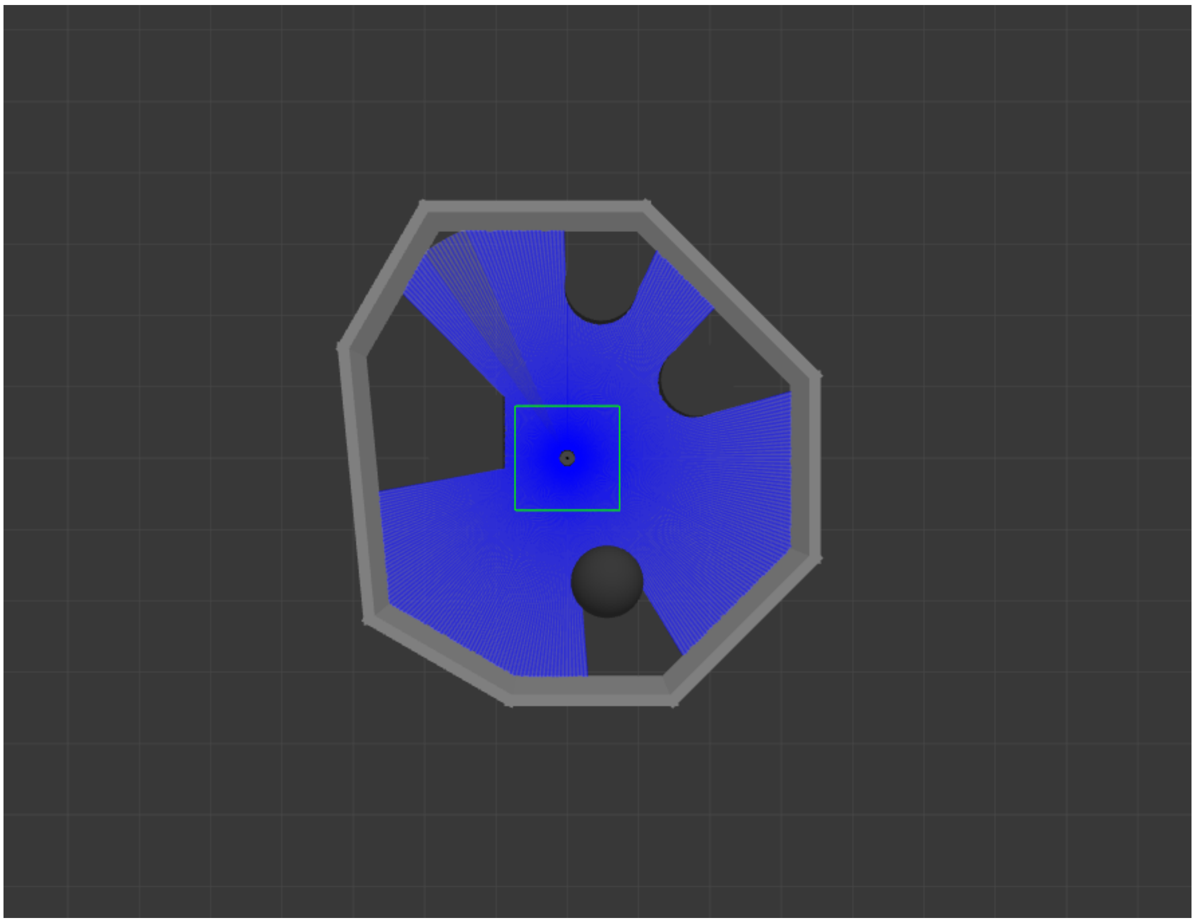




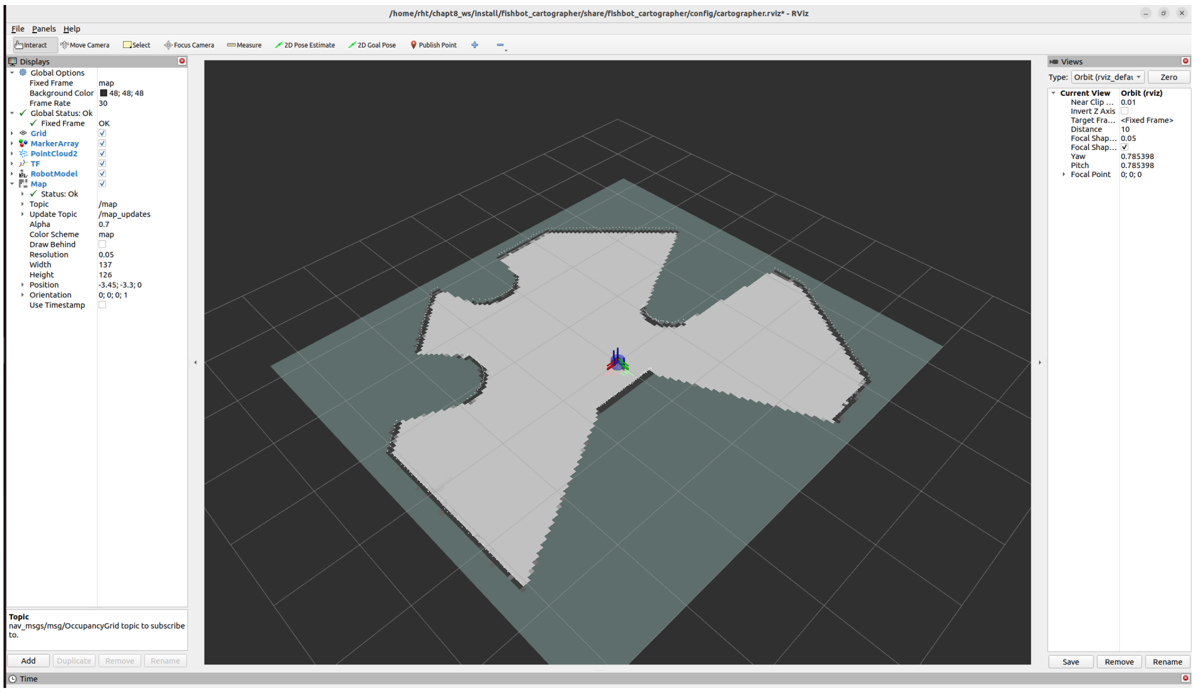
gazebo仿真环境搭建


Gazebo
8月24日 12：37
Gazebo
File
Building Editor
Model Name: Untitled
Create Walls
Wall
Add Features
Window
Door
Stairs
Add Color
More
Add Texture
Wood
Tiles
Bricks
Import
View
Level 1
-
+
1.00 m
Building is View Only





建图



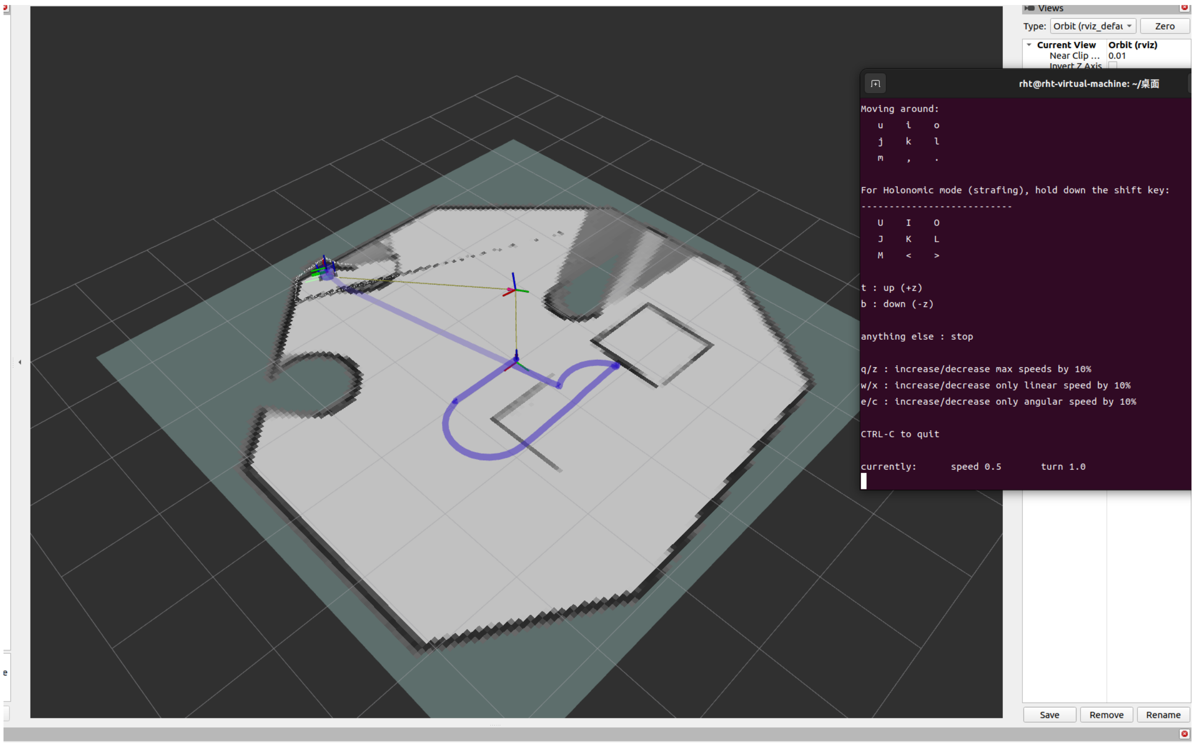
/home/rht/chapt8_ws/install/fishbot_cartographer/share/fishbot_cartographer/config/cartographer.rviz* - RViz

Global Options	
Fixed Frame	map
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
MarkerArray	✓
PointCloud2	✓
TF	✓
RobotModel	✓
Map	✓
✓ Status: Ok	
Topic	/map
Update Topic	/map_updates
Alpha	0.7
Color Scheme	map
Draw Behind	☐
Resolution	0.05
Width	137
Height	126
Position	-3.45; -3.3; 0
Orientation	0; 0; 0; 1
Use Timestamp	☐

Topic
nav_msgs/msg/OccupancyGrid topic to subscribe to.

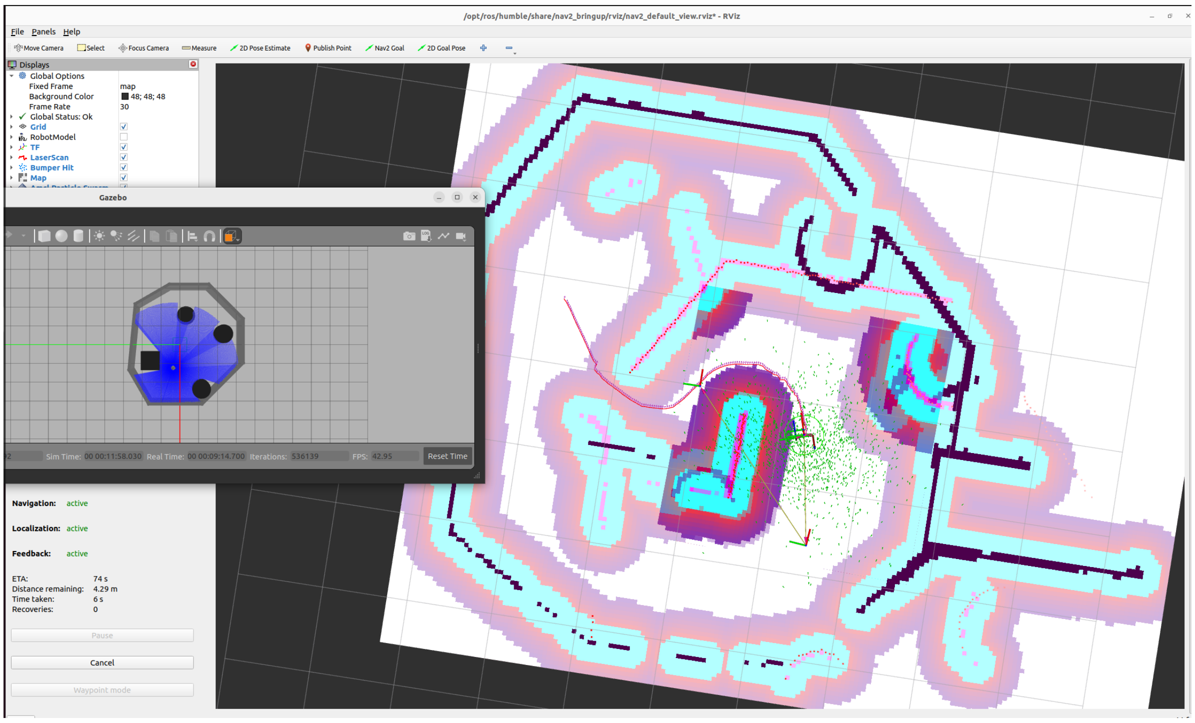
Type: Orbit (rviz_defau Zero

Current View	Orbit (rviz)
Near Clip ...	0.01
Invert Z Axis	☐
Target Fra...	<Fixed Frame>
Distance	10
Focal Shap...	0.05
Focal Shap...	✓
Yaw	0.785398
Pitch	0.785398
Focal Point	0; 0; 0



```
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%

CTRL-C to quit

currently:      speed 0.5       turn 1.0
```

导航



```
Global Options
  Fixed Frame          map
  Background Color     48; 48; 48
  Frame Rate           30
Global Status: Ok
Grid
RobotModel
TF
LaserScan
Bumper Hit
Map
```

```
Sim Time: 00 00:11:58.030  Real Time: 00 00:09:14.700  Iterations: 536139  FPS: 42.95
```

```
Navigation:     active
Localization:   active
Feedback:       active

ETA:                 74 s
Distance remaining:  4.29 m
Time taken:          6 s
Recoveries:          0
```