

基于 TurtleBot3 的仿真

通过 TurtleBot3 的示例，了解整个仿真的过程

基于 TurtleBot3 的仿真

前提

- 基于 Ubuntu18.04
- 需提前安装 ROS 环境，对应版本为 ros-melodic-desktop-full

1 Gazebo 仿真

<1> 安装 TurtleBot3 相关的包

```
1 | sudo apt install ros-melodic-turtlebot3
```

<2> 设置 TurtleBot3 机器人模型

```
1 | echo "export TURTLEBOT3_MODEL=burger" >> ~/.bashrc
2 | source ~/.bashrc
```

<3> 启动 TurtleBot3 Gazebo 仿真环境

```
1 | roslaunch turtlebot3_gazebo turtlebot3_house.launch
```

注意，如果第一次运行，需要下载资源会耗费一些时间

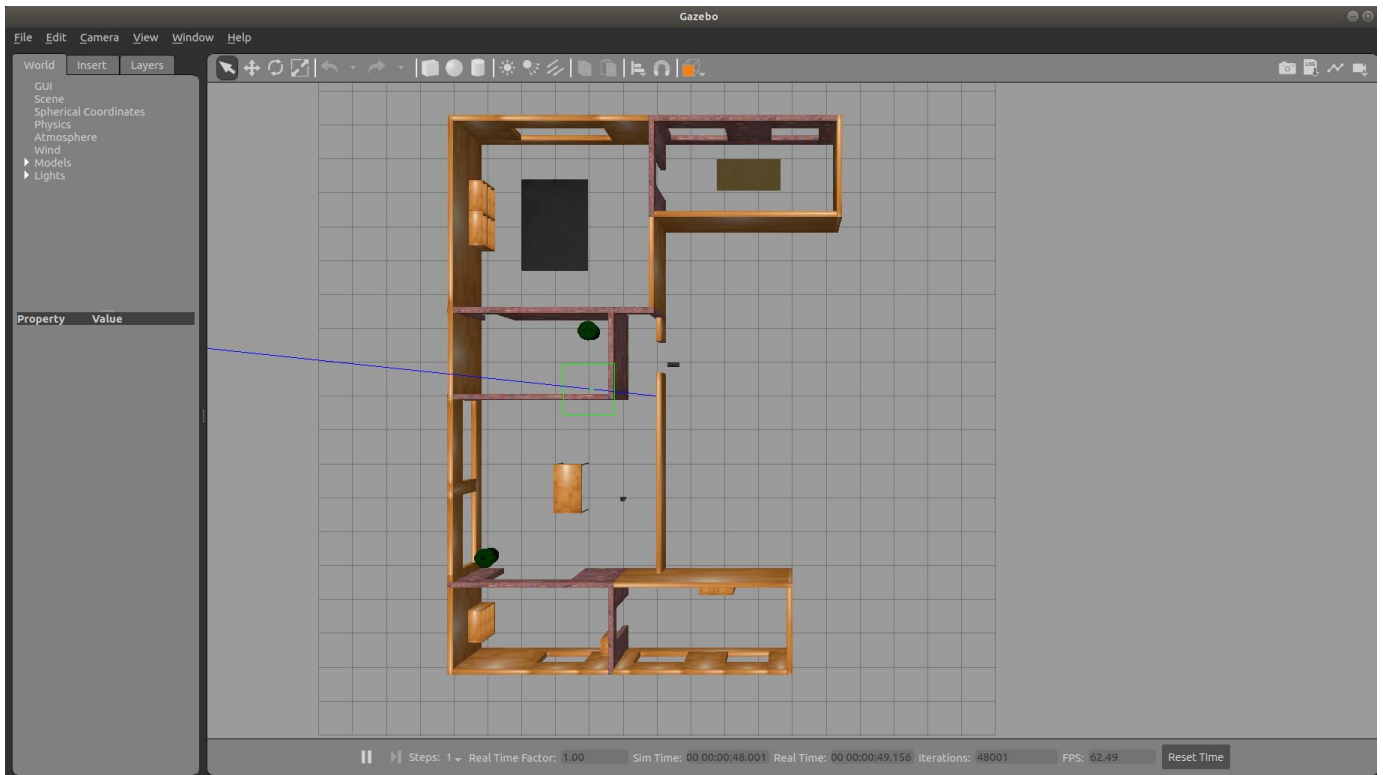
另外，如果是在虚拟机上运行，启动 Gazebo 可能会有报如下错误：

VMware: vmw_ioctl_command error Invalid argument.

可通过执行如下命令解决

```
1 | echo " export SVGA_VGPU10=0" >> ~/.bashrc
```

Gazebo仿真效果如下



2 SLAM 仿真

提示：在 Gazebo 仿真的基础上，继续执行如下

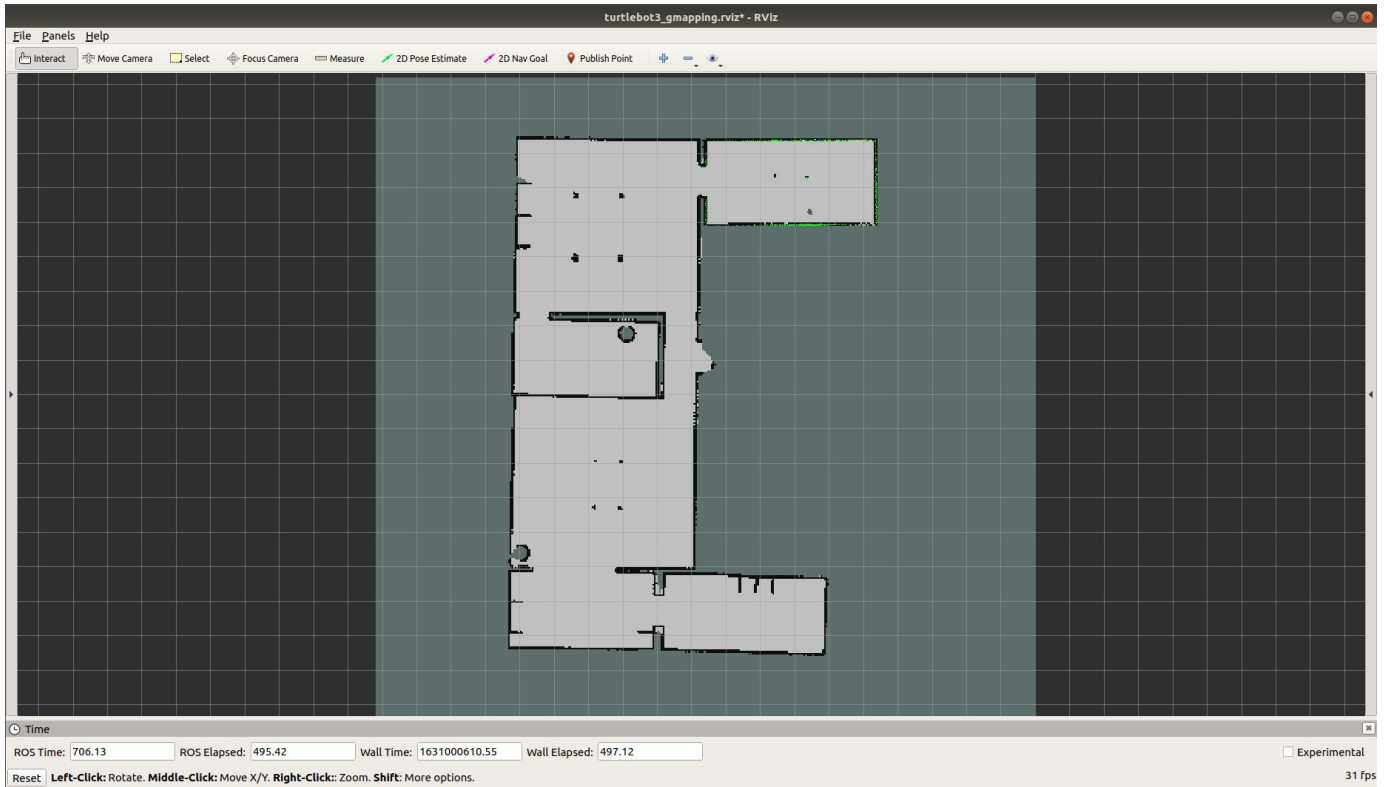
<1> 运行 TurtleBot3 SLAM 节点，使用 gmapping，启动 Rviz

```
1 | roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping
```

<2> 运行运动控制程序，通过 'W'、'X'、'A'、'D'、'S' 分别控制 TurtleBot3 前、后、左、右、停，对照 Gazebo 来扫描建图

```
1 | roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

Rviz 建图效果如下



<3> 保存制图

```
1 | rosrn map_server map_saver -f ~/map
```

3 导航仿真

提示：退出前面运行的程序

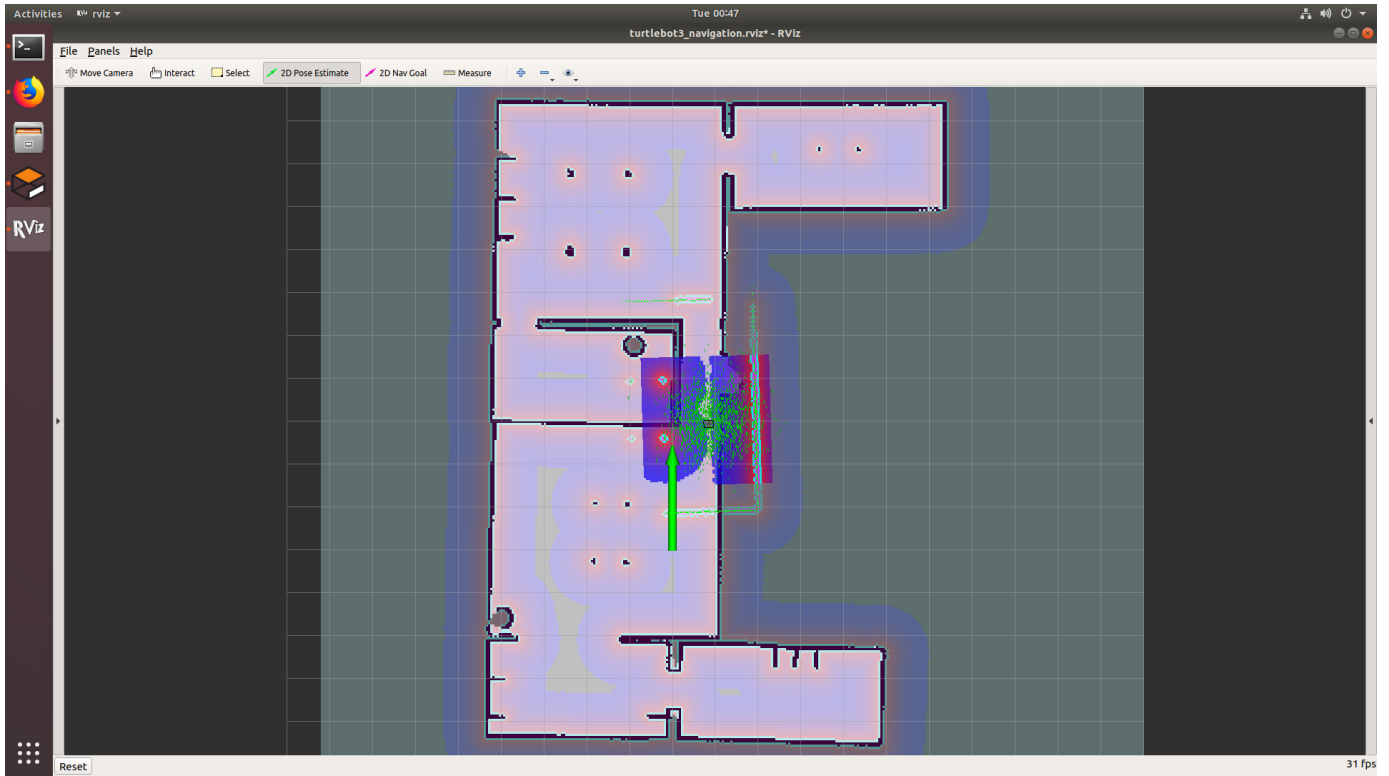
<1> 启动 TurtleBot3 Gazebo 仿真环境

```
1 | roslaunch turtlebot3_gazebo turtlebot3_house.launch
```

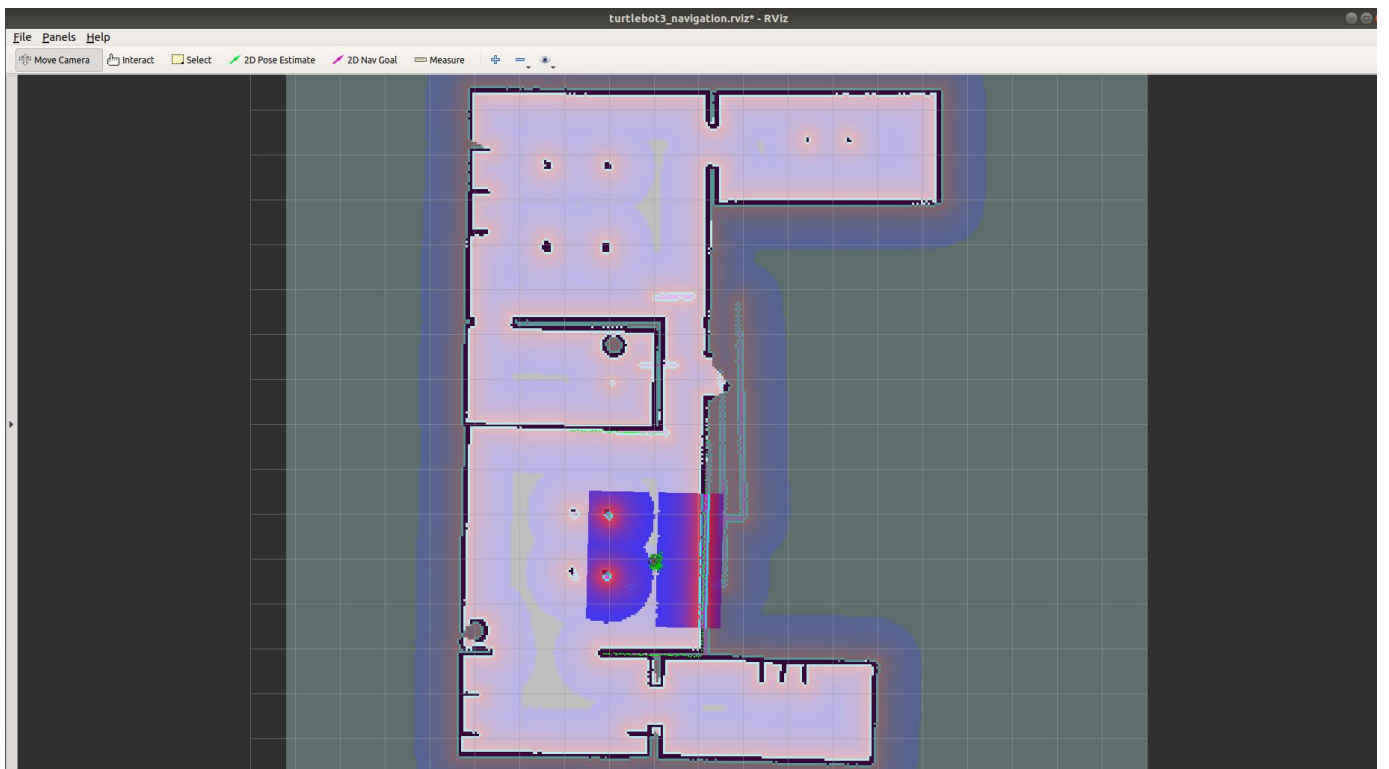
<2> 启动导航节点，加载上面建好的地图

```
1 | roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=$HOME/m
```

<3> 在上方工具栏点击“2D Pose Estimate”，对照 Gazebo TurtleBot3 位置，在 Rviz 选择起点

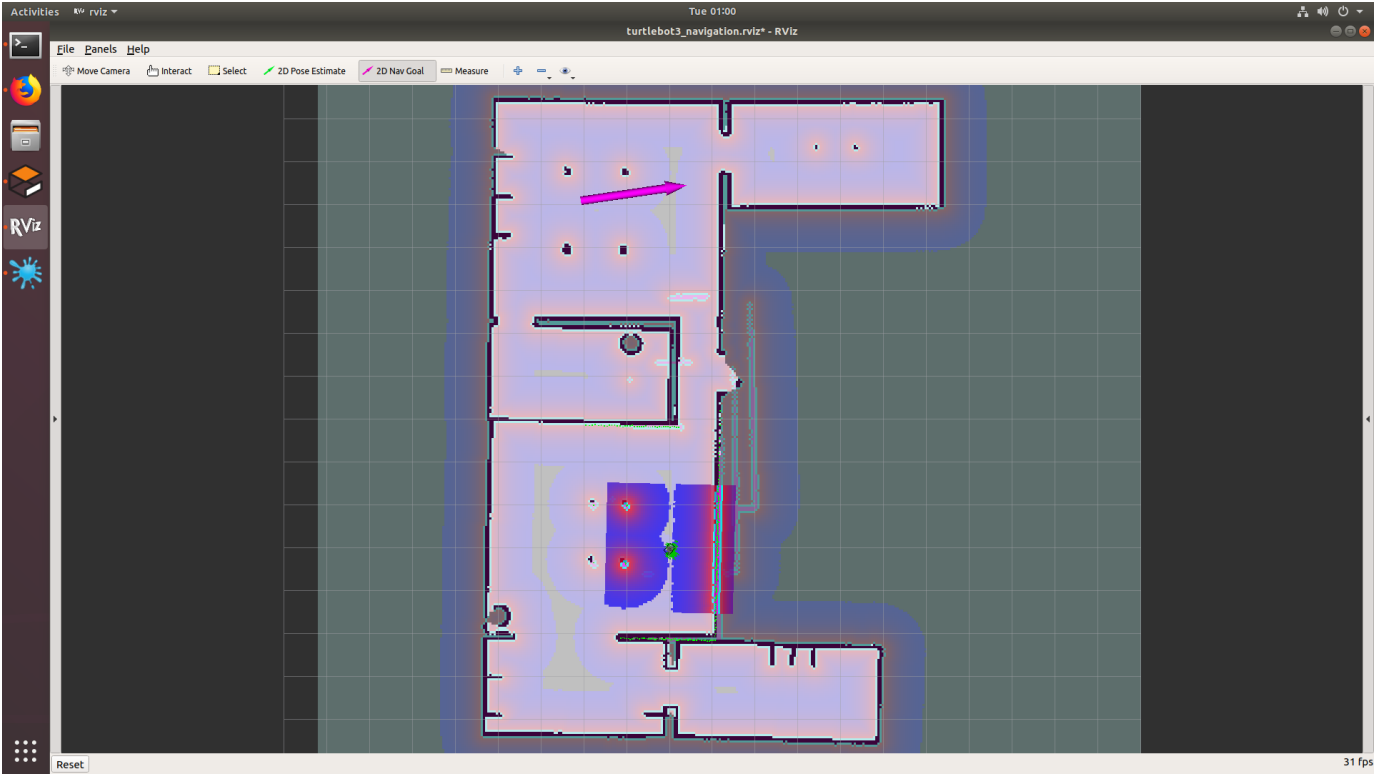


<4> 运行运动控制程序，通过 'A'、'D' 分别控制 TurtleBot3 旋转，收集环境信息，缩小 TurtleBot3 在地图上的估计位置，Rviz 中 TurtleBot3 附近的绿色小箭头会减少聚合



<5> 退出运动控制程序，防止后续导航过程中，误发送运动指令

<6> 在上方工具栏点击“2D Nav Goal”，在 Rviz 中选择导航目标点



导航到目标位置

