

移动机器人地图构建与自主导航实验指导书

实验项目编号 01089201008

一、实验目的

1. 了解常用建图导航功能包优缺点、熟悉建图测试流程

二、实验内容及要求

1. 实验内容

- ①了解机器人常用建图功能包，熟悉实机建图和导航操作。

2. 实验要求

- ①提供机器人建图功能包计算图级。
- ②提供机器人导航功能包计算图级。
- ③提供建立的地图和 Rviz 显示的代价地图。
- ④第二次实验课开始，每次实验时必须提交上次实验的实验报告。最后一次实验报告由实验后两星期内提交，否则不再受理。

三、实验原理

1、机器人建图及导航操作

SLAM (simultaneous localization and mapping), 也称为 CML (Concurrent Mapping and Localization), 即时定位与地图构建, 或并发建图与定位。

问题可以描述为: 将一个机器人放入未知环境中的未知位置, 是否有办法让机器人一边逐步描绘出此环境完全的地图, 同时一边决定

机器人应该往哪个方向行进。

例如扫地机器人就是一个很典型的 SLAM 问题，所谓完全的地图（a consistent map）是指不受障碍行进到房间可进入的每个角落。SLAM 最早由 Smith、Self 和 Cheeseman 于 1988 年提出。由于其重要的理论与应用价值，被很多学者认为是实现真正全自主移动机器人的关键。

当你来到一个陌生的环境时，为了迅速熟悉环境并完成自己的任务（比如找饭馆，找旅馆），你应当做以下事情：

- a. 用眼睛观察周围地标如建筑、大树、花坛等，并记住他们的特征（特征提取）
- b. 在自己的脑海中，根据双目获得的信息，把特征地标在三维地图中重建出来（三维重建）
- c. 当自己在行走时，不断获取新的特征地标，并且校正自己头脑中的地图模型（bundle adjustment or EKF）
- d. 根据自己前一段时间行走获得的特征地标，确定自己的位置（trajectory）
- e. 当无意中走了很长一段路的时候，和脑海中的以往地标进行匹配，看一看是否走回了原路（loop-closure detection）。实际这一步可有可无。

下表为 2D 激光 SLAM 方案特点

年份	方案	传感器	特点
1988	EKF-SLAM	2D 激光	构建特征地图，计算量复杂，鲁棒

			性较差。
2002	FastSLAM	2D 激光	最早实时输出栅格地图；消耗内存，粒子耗散严重。
2007	Gmapping	2D 激光	缓解粒子耗散；非常依赖于里程计信息。
2010	Optimal RBPF	2D 激光	进一步减少粒子退化问题。
2010	Karto SLAM	2D 激光	首个图优化框架开源方案，认识到稀疏性；时间消耗大。
2010	CoreSLAM	2D 激光	最小损失函数的算法。
2011	Hector-SLAM	2D 激光	不需里程计数据；强旋转漂移，初值敏感，难以闭环。
2012	LagoSLAM	2D 激光	图优化 SLAM 系统，最小化非线性非凸代价函数。
2016	Cartographer	2D 激光	CSM 与梯度优化的前端，图优化的后端，加速的闭环检测。

四、实验步骤

1、移动机器人建图及导航实验步骤

①启动机器人 SLAM 节点

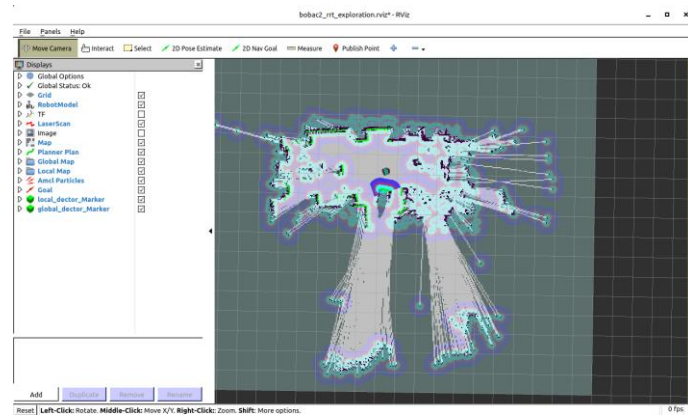
- 打开机器人 slam 节点开始建图

```
roslaunch bobac2_slam bobac2_slam.launch
```

- 开启新终端打开手柄控制节点

```
roslaunch bobac2_joy bobac2_joy.launch
```

- 使用手柄控制小车进行移动进行建图



②保存地图

- 进入地图保存路径

```
roscd bobac2_navigation/maps
```

或

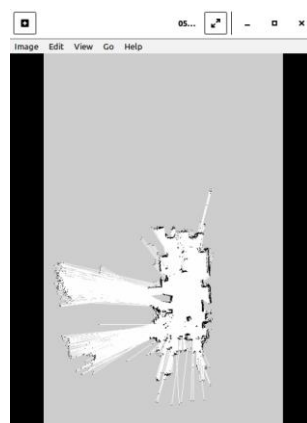
```
cd bobac2_ws/src/bobac2_navigation/maps
```

- 使用 map_server 保存地图至上述路径

```
roslaunch map_server map_saver -f <文件名>
```

- 妥善保存建好的地图

```
eog <文件名>.pgm
```



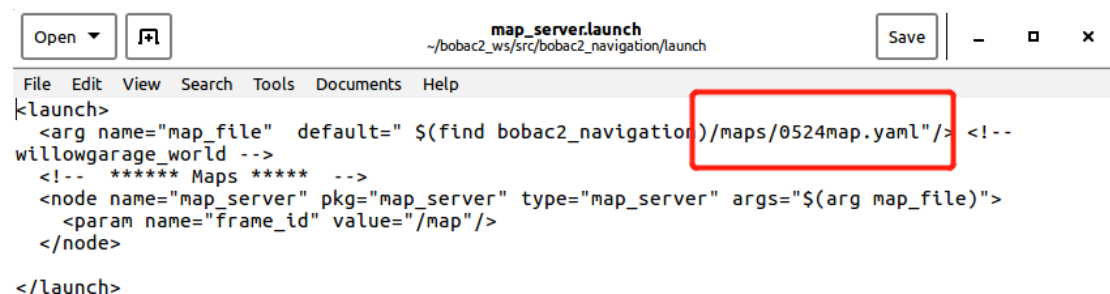
③移动机器人导航实验

- 打开 map_server.launch 文件进行地图修改

```
cd bobac2_ws/src/bobac2_navigation/launch
```

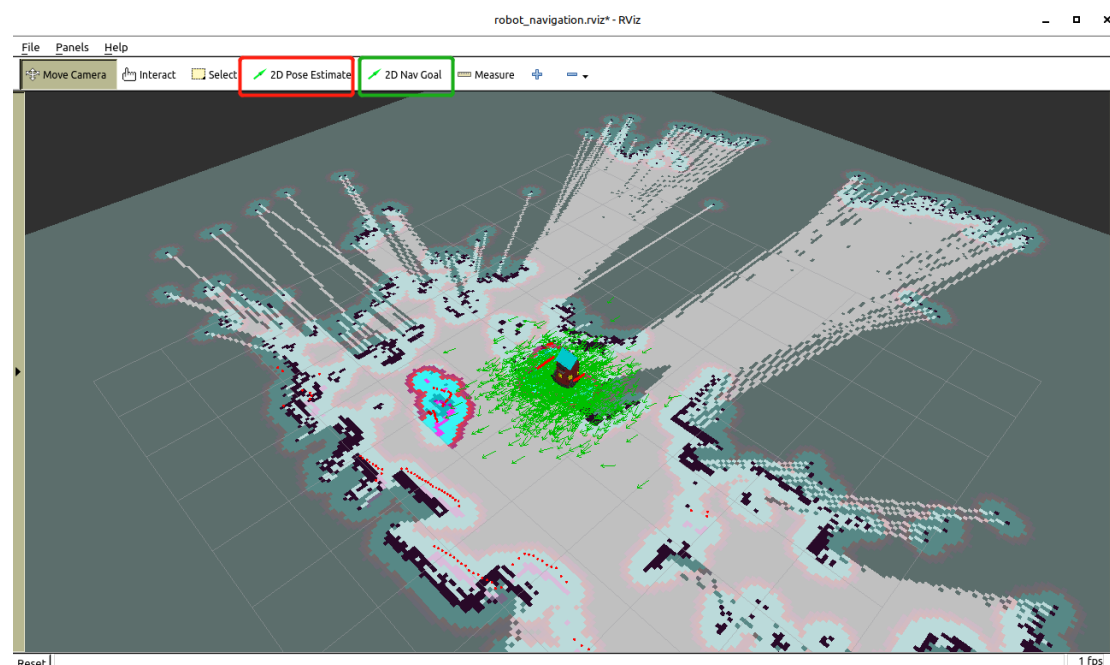
```
gedit map_server.launch
```

红色部分替换为自己保存的地图



➤ 打开机器人导航功能包，将出项 rviz 界面

```
roslaunch boba2_navigation boba2_nav_2d.launch
```

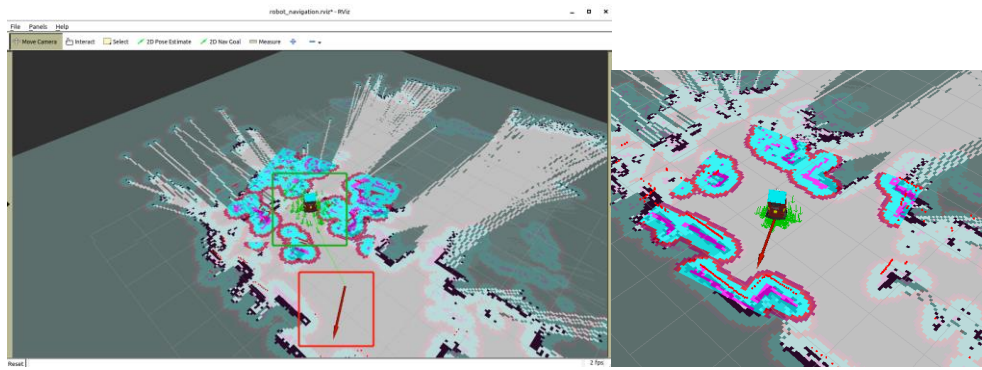


红色“2D Pose Estimate”： 是手动给机器人估计当前位置。

绿色“2D Nav Goal”： 让机器人导航到目标位置去。

机器人下方的绿色箭头为机器人位姿粒子点云。

使用 2D Nav Goal 给定机器人导航目标如下图红框所示，机器人会自动行进至目标位姿。目标点为红色箭头，绿色直线为规划的全局路径。



五、实验结果（图片/表格）

- a) 机器人构建的二维地图；
- b) 机器人导航的 `rviz` 截图（可见代价地图、全局规划轨迹）；
- c) 机器人建图时的实体照片