

- 前期准备知识
- 一些参考文档
- 传感器启动
- 建图
 - 使用cartographer_3d 建图
 - 使用cartographer_3d 纯定位模式
 - 使用move_base 导航
- 一些参数调整

前期准备知识

- Ros 的使用， roslaunch， launch文件， rviz 订阅话题 可视化话题
- Cartographer_3d 原理
- EKF 原理（融合轮速计和imu信息）
- urdf配置， cartographer_ros/cartographer_ros/urdf/my_3d.urdf 配置小车和imu、雷达的位姿关系， imu需要z轴朝下， 因为Cartographer_3d 需要这个信息来判断地面所在方向

一些参考文档

- [Cartographer 3D实时建图教程\(超详细\) - 镭神C16 + IMU（一） - 知乎 \(zhihu.com\)](#)
- (21条消息) [cartographer 3D点云建图教程_cartography 建图_小肉包老师的博客-CSDN博客](#)
- (21条消息) [cartographer跑自己的数据+3d建图篇\(Lidar+imu\)_叫我李先生的博客-CSDN博客](#)
- [ROS学习笔记-机器人导航\(仿真\)2-路径规划move_base - 知乎 \(zhihu.com\)](#)
- [dgt001m_can驱动包使用说明dgt00.docx \(文档， 使用前先运行 yhs_can_control/start_car.sh\) can盒灯全亮并在闪就是正常运行](#)

传感器启动

- imu 参考(21条消息) [三驰惯导IMU传感器100D2的测试使用_杰尼君的博客-CSDN博客](#)
- rslidar 参考(21条消息) [Ubuntu18.04 安装速腾聚创最新驱动RSLidar_SDK采集 XYZIRT格式的激光点云数据 --SLAM不学无术小问题_rslidar驱动_^oprater^的博客-](#)

- 小车的启动参考docx文档

建图

使用cartographer_3d 建图

1. 录制bag （需要录制话题 imu、 yhs左右轮速信息（看docx文档的话题）、激光雷达的话题）
2. 回放bag
3. 运行 wheel_odom/launch/start.launch 发布轮速计话题
4. 运行 robot_pose_ekf/launc/ekf.launch 发布融合了imu 轮速计信息的 odom话题
5. 运行 cartographer_ros/cartographer_ros/launch/my_3d.launch 注意需要设置use sim time 为true，因为回放的bag和真实时间不一样，相当于在模拟
6. 停止建图，保存地图

```
rosservice call /finish_trajectory 0 //结束路径0 的建图
rosservice call /write_state "{filename:
'${HOME}/Downloads/mymap.pbstream'}" //保存地图
```

使用cartographer_3d 纯定位模式

1. 修改文件occupancy_node_main.cc 注释 171行
//occupancy_grid_publisher_.publish(*msg_ptr);不然你纯定位过程建立的子图会覆盖在之前建好的地图上。并重新编译 cartographer，参考定位的CSDN博客
2. 启动各种传感器 imu 雷达 dgt车 轮速计
3. 运行 wheel_odom/launch/start.launch 发布轮速计话题
4. 运行 robot_pose_ekf/launc/ekf.launch 发布融合了imu 轮速计信息的 odom话题
5. 运行
cartographer_ros/cartographer_ros/launch/my_demo_backpack_3d_localiztion.launch 注意修改mapserver 和 pbstream指定对应的map

使用move_base 导航

1. 运行 car_control/launch/start.launch 使得车辆可以被move_base 控制

一些参数调整

- 由于设计的参数比较多，所以很多参数的设置需要具体情况具体设置，例如话题的名称，可能需要在launch文件里面remap或者是读取
- 建图和定位的参数调整类似，都在lua文件里面，可以看launch文件里面指定的是哪一个lua文件

`POSE_GRAPH.optimize_every_n_nodes =20` // 这个数值越低后端优化的频率越高，但是效果不一定最好，数值越高优化频率就越低

`POSE_GRAPH.constraint_builder.sampling_ratio =0.35`

`POSE_GRAPH.constraint_builder.min_score =0.42` // 和建图的效果相关，可以调整来判断不同的效果

`POSE_GRAPH.constraint_builder.global_localization_min_score =0.46`

在纯定位时有一个比较关键的参数：

`pose_publish_period_sec = 0.2`, 降低pose发布的频率可以降低movebase路径的改变频率，导航的时候会更加平顺稳定

movebase里的costmap的参数，以及导航路径的更新发布频率可以参考知乎，局部路径规划需要稍短一点避障会更好，costmap的更新频率也需要稍微低一点，路径规划的更新频率也会降低（但不是越低越好，太低可能会直接装上障碍物），导航效果也就更好