costmap

AI 航团队

1. costmap

costmap 是 Navigation Stack 里的代价地图，它其实也是 move_base 插件，本质上是 C++的动态链接库，用过 catkin_make 之后生成.so 文件，然后 move_base 在启动时会通过动态加载的方式调用其中的函数。

2. 代价地图

之前我们在介绍 SLAM 时讲过 ROS 里的地图的概念，地图就是/map 这个 topic，它也是一张图片，一个像素代表了实际的一块面积，用灰度值来表示障碍物存在的可能性。然而在实际的导航任务中，光有一张地图是不够的，机器人需要能动态的把障碍物加入，或者清楚已经不存在的障碍物，有些时候还要在地图上标出危险区域，为路径规划提供更有用的信息。Navigation Stack 是一个 ROS 的 metapackage，里面包含了 ROS 在路径规划、定位、地图、异常行为恢复等方面的 package，其中运行的算法都堪称经典。Navigation Stack 的主要作用就是路径规划，通常是输入各传感器的数据，输出速度。一般我们的 ROS 都预装了 Navigation。

代价地图有如下特点：

costmap 简单来说就是为了在这张地图上进行各种加工，方便我们后面进行路径规划而存在的。那具体该如何实现 costmap 呢？在 ROS 中使用 costmap_2d 这个软件包来实现的，该软件包在原始地图上实现了两张新的地图。一个是 local_costmap，另外一个就是 global_costmap，根据名字大家就可以知道了，两张 costmap 一个是为局部路径规划准备的，一个是为全局路径规划准备的。无论是 local_costmap 还是 global_costmap，都可以配置多个图层，包括下面几种：

Static Map Layer：静态地图层，通常都是 SLAM 建立完成的静态地图。

Obstacle Map Layer：障碍地图层，用于动态的记录传感器感知到的障碍物信息。

Inflation Layer：膨胀层，在以上两层地图上进行膨胀（向外扩张），以避免机器人的

外壳会撞上障碍物。

Other Layers：你还可以通过插件的形式自己实现 costmap，目前已有 Social Costmap Layer、Range Sensor Layer 等开源插件。

可以同时选择多个 Layer 并存。

根据 move_base 的内部逻辑流程图得知，在进行路径规划时 costmap 是必不可少的。因此我们需要首先创建个 stdr_move_base 软件包，然后配置 costmap 相关的参数，这样 move_base 软件包内的路径规划器才能找到一条合适的路径控制机器人移动到达指定的目的地。

https://www.corvin.cn/829.html