机器人导航总结

1. 自己总结：

* 任务类型：找东西（先去某个特定位置找，没找到再扩大搜索范围）、到某个坐标点、遍历整个区域作路径规划
* 空间覆盖，扫地机器人会面临不同的地方，沿墙边，避障、大的自由空间等

1. 反应式导航：现状是很多机器人在完成任务时并不依赖任何地图，如寻着光源前行、沿着地上白线走、沿着墙壁穿越迷宫和沿随机路线走
2. 基于地图的类人导航方法支持更复杂的机器人任务，需要地图和定位
3. 基于地图的路径规划常用算法：规划和查询，常只考虑距离，但有时需要考虑运动学和动力学产生的成本，如颠簸带来的磨损、能否控制电机沿着路径走。

假设条件是：机器人子网格中运动，且只占一格单元格（机器人可能不止占一个单元格）；机器人不含任何非完整约束，且可以移动到相邻单元格（运动学和动力学不满足全方向运动能力）；机器人可以确定它在平面上的位置（slam）；机器人可以使用地图计算它要走的路径（路径规划）

* 距离变换
* A\*、D\*（A\*的升级版）：允许成本地图改变后重新计算路径，但他不支持改变目标。若B为目标，那以B为中心产生规划，然后根据不同A查询
* 沃罗诺伊路线图法：细化或骨架化也与距离变换一样计算量很大的迭代算法，但它友好为我们展示了自由空间寻找路径的方法和原理。根据障碍物生成骨架，那么A、B都可以改变
* 概率路线图方法（PRM）：首先在自由空间中找N个随机点，然后用直线将每个点与其相近的邻点连接，不穿过障碍物，这样就形成点数最少且开环的网络图。查询时，从起点移动到离他最近的节点，然后按路线图走到离终点最近的节点，最后从节点走到终点
* 快速搜索随机树（rapidly-exploring random，RRT）:考虑了机器人的运动模型，也是一种概率算法

1. Dijkstra算法也即：单源最短路径 算法。算法的基本思想是：每次找到离源点最近的一个顶点，然后以该顶点为中心进行扩展，最终得到源点到其余所有点的最短路径
2. 为了避免机器人与障碍物碰撞，可以使用闵可夫斯基和（minkowski sum）使障碍物膨胀
3. 导航时通过使用两种代价地图存储周围环境中的障碍信息：一种用于全局路径规划，一种用于本地路径规划和实时避障
4. 1