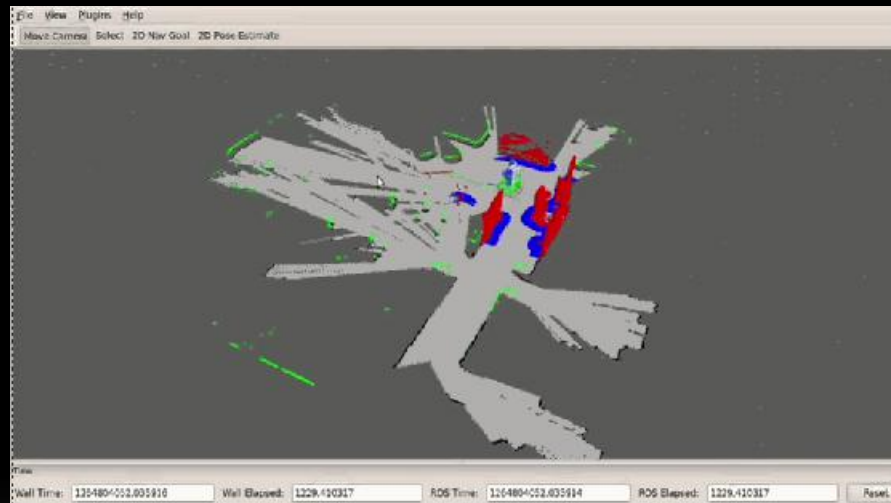


ROS组件

----Navigation

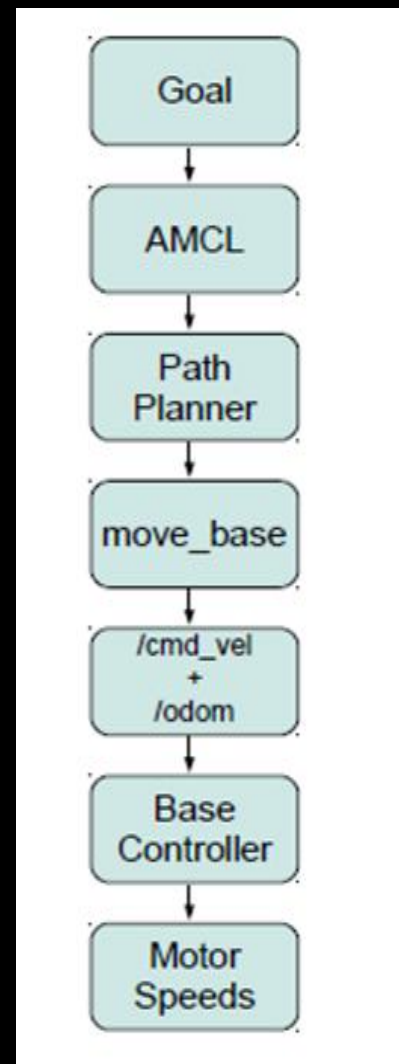
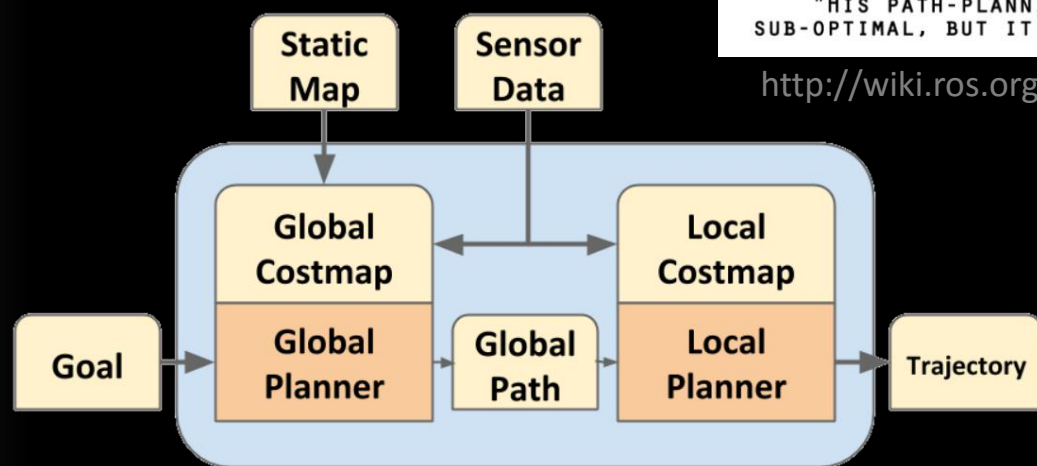


内容概要

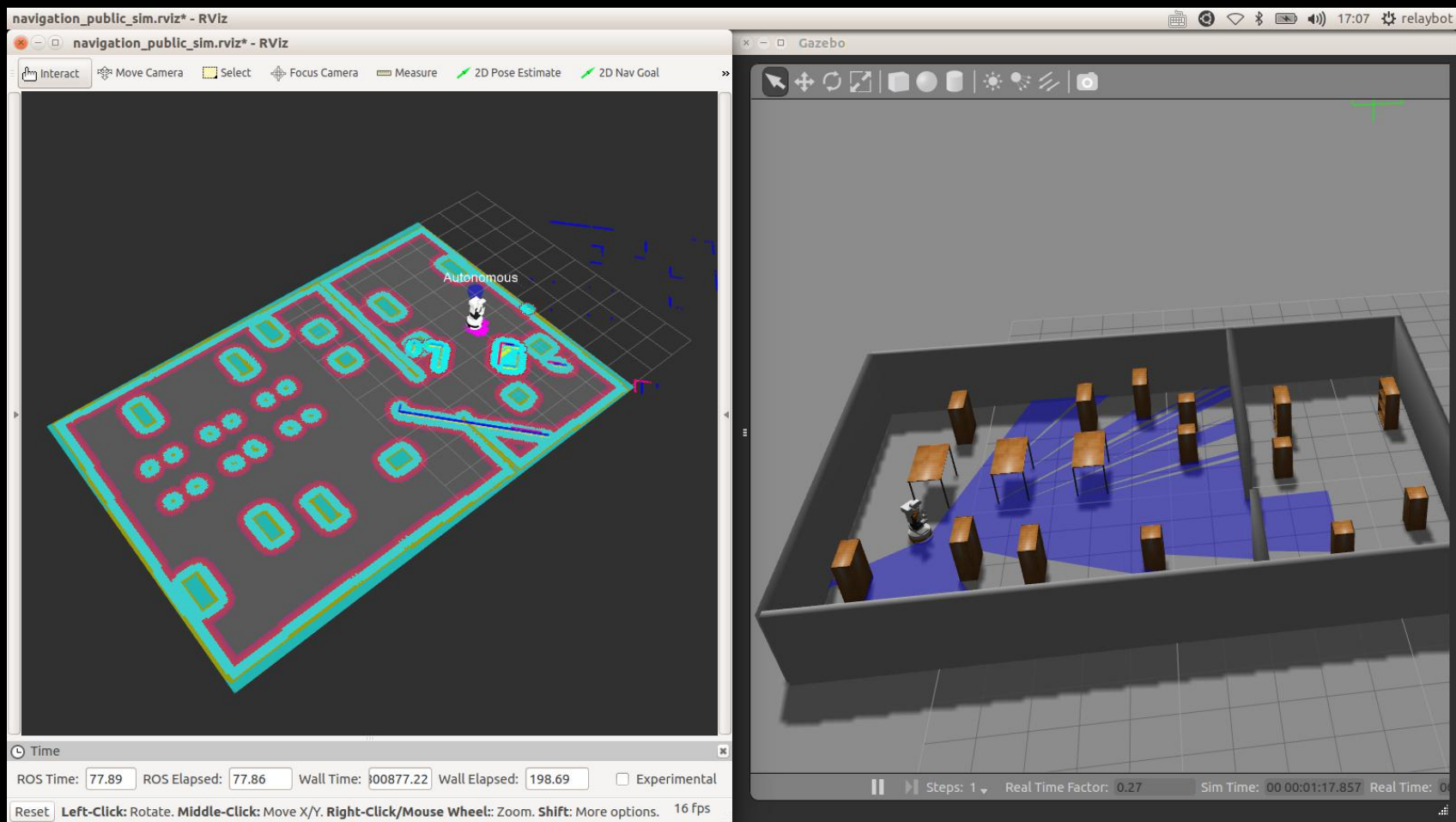
- Navigation概念
- Navigation应用



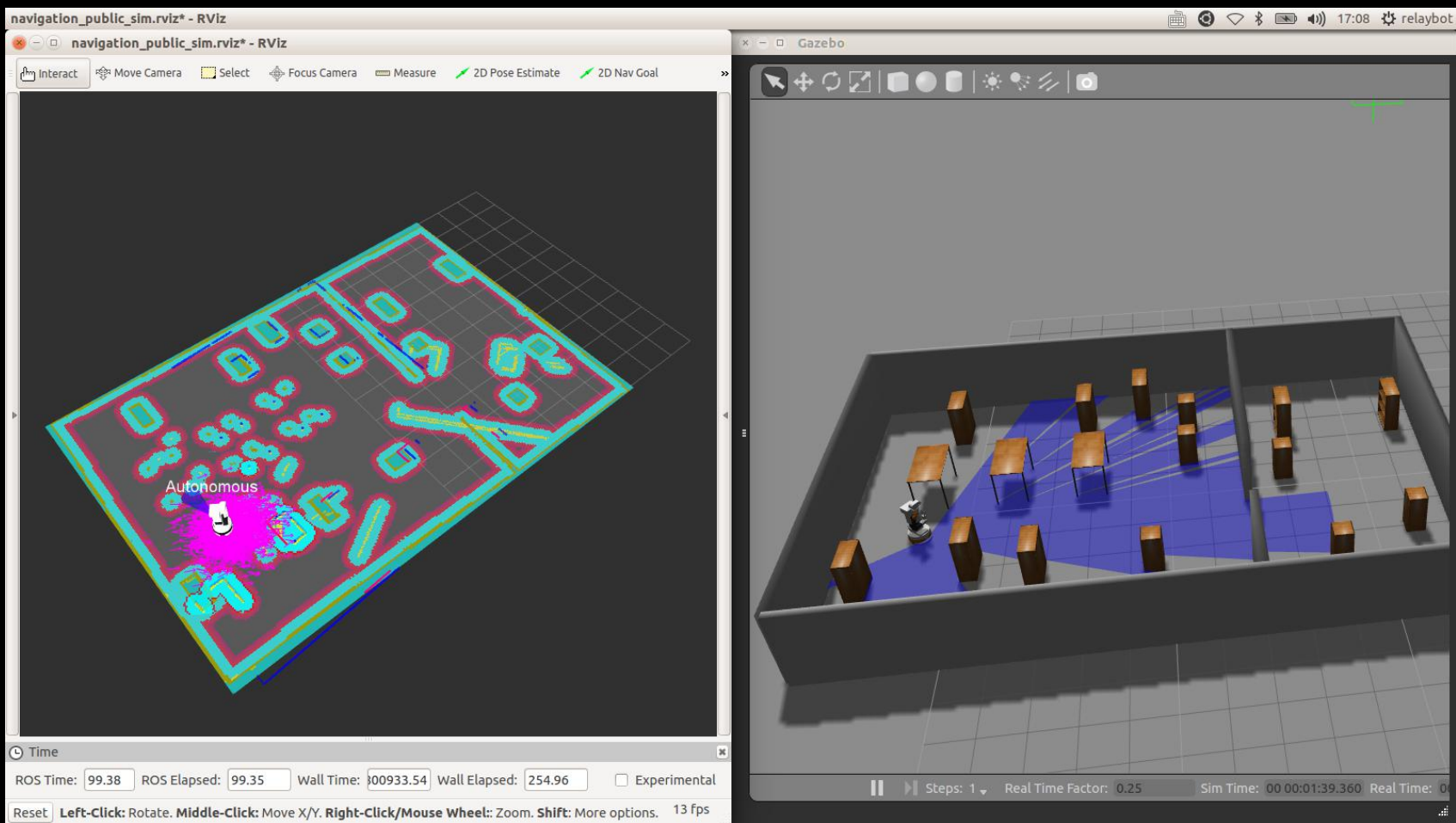
<http://wiki.ros.org/navigation>



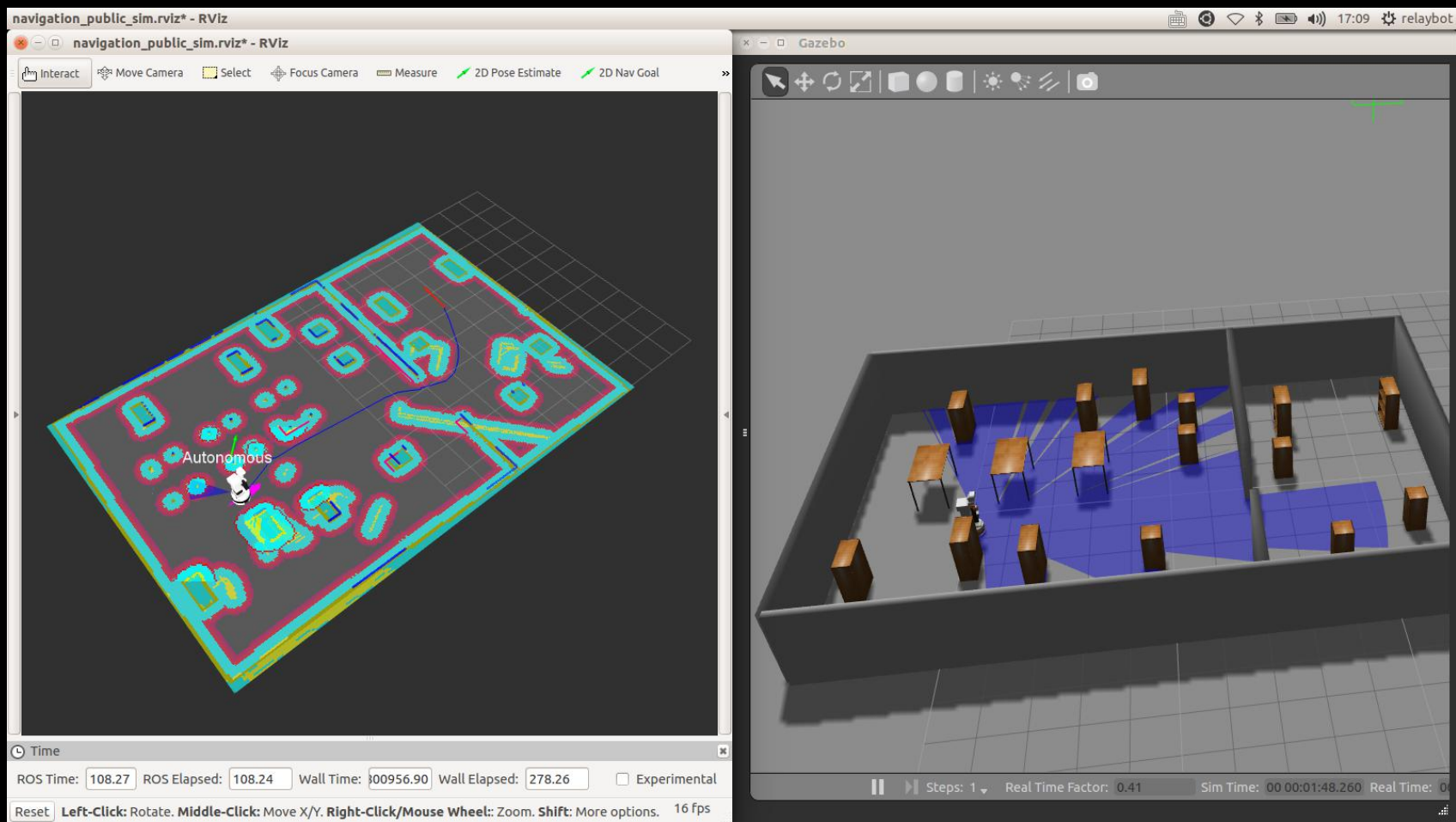
示例



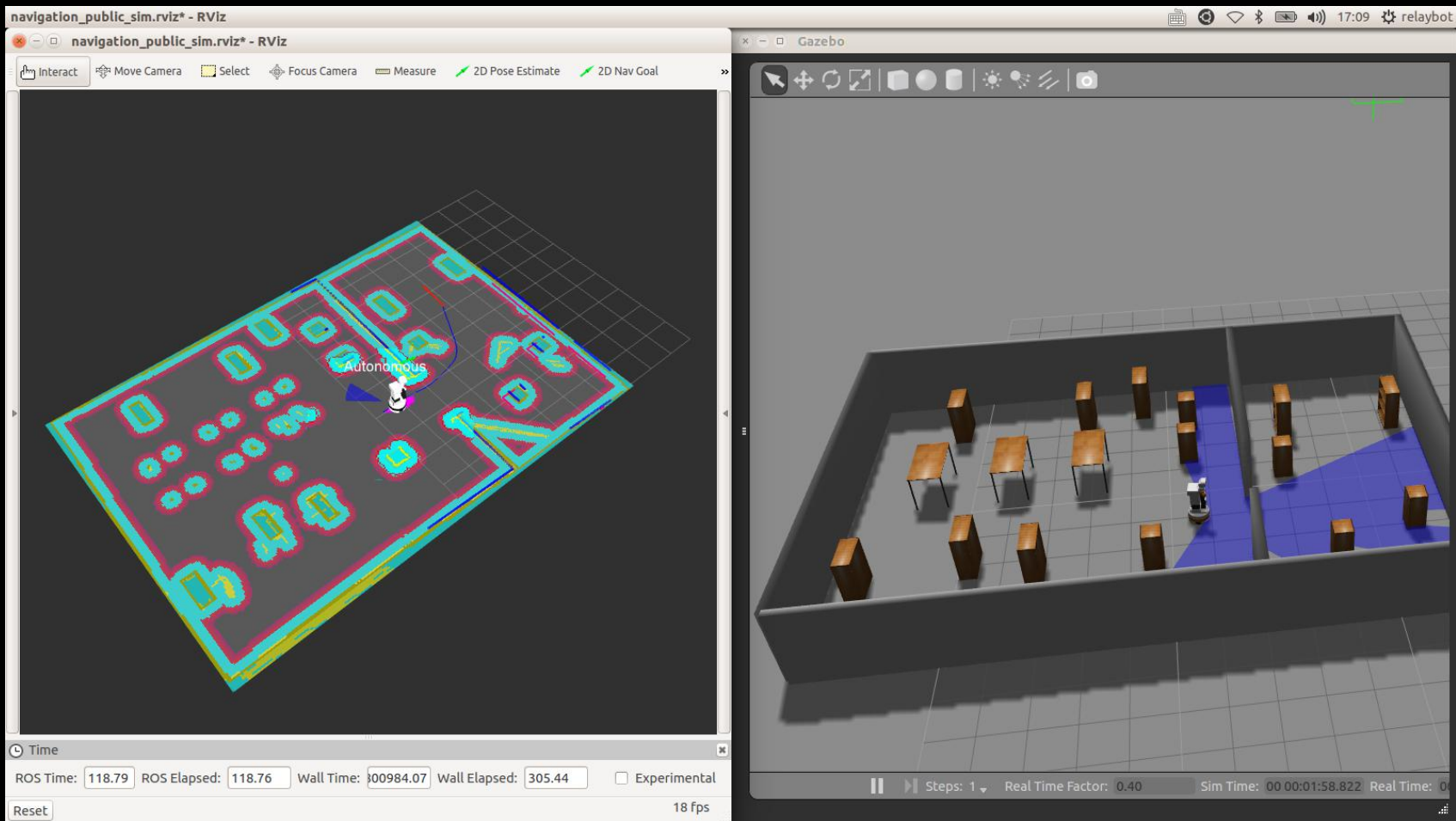
示例



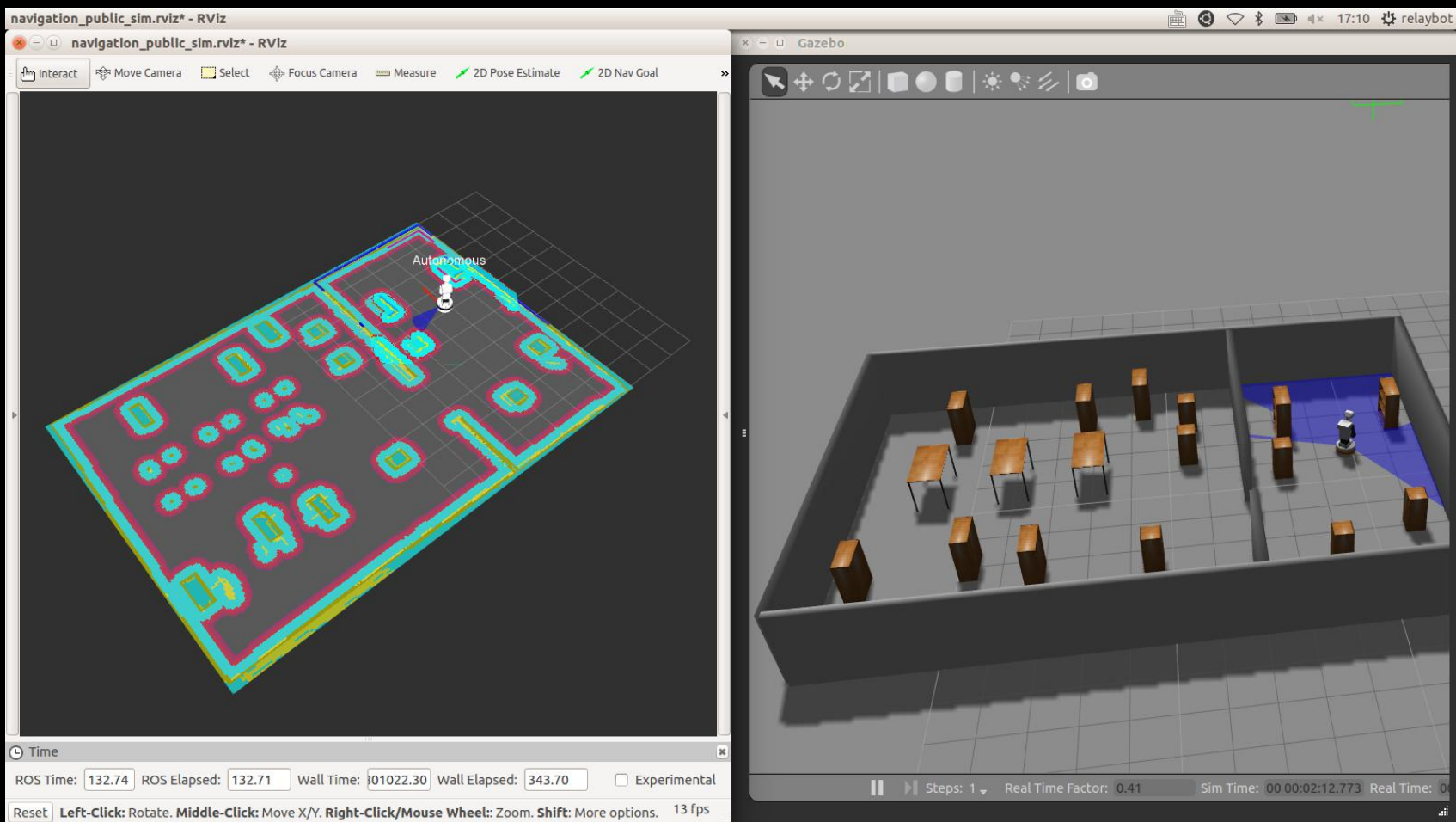
示例



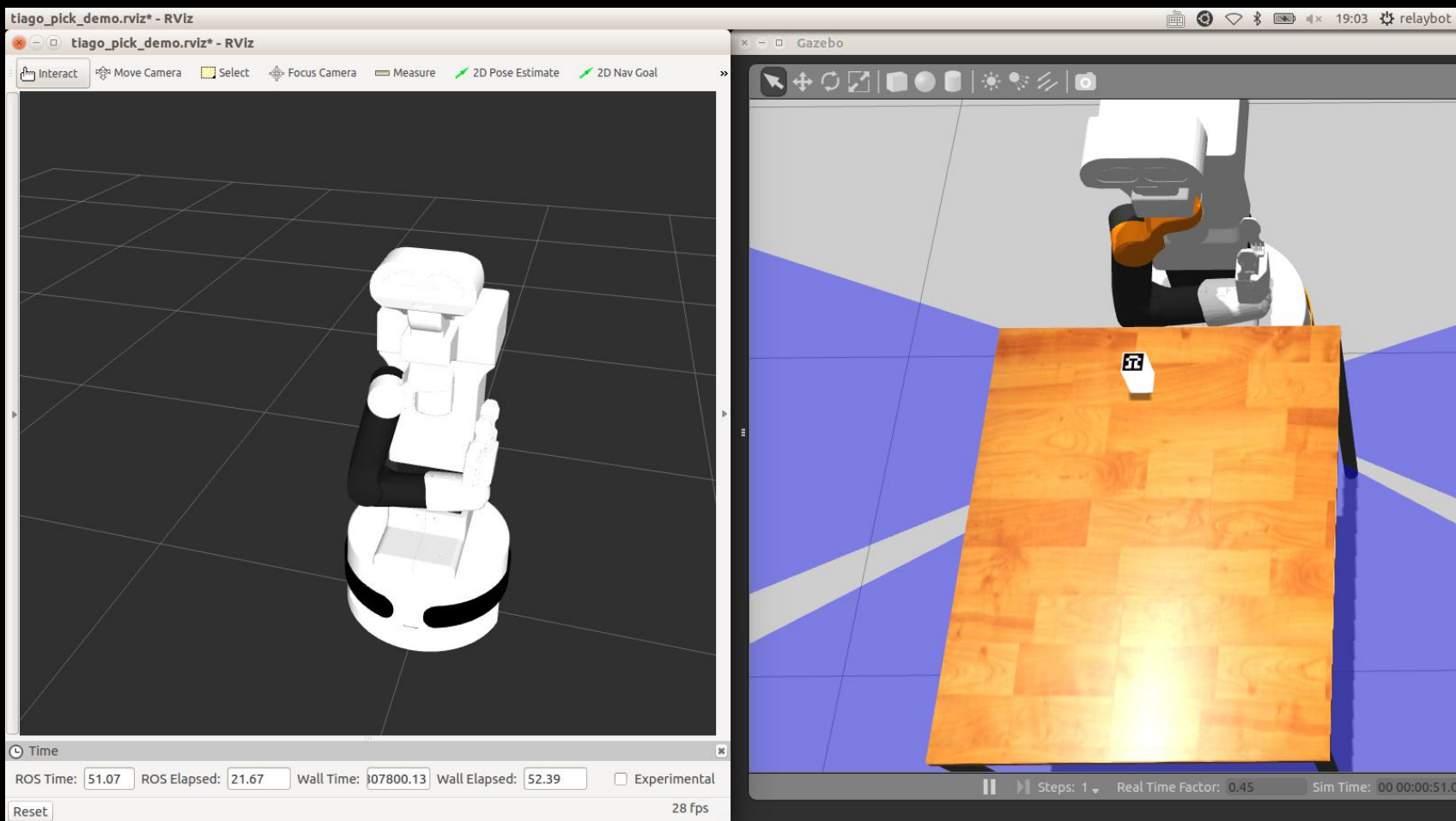
示例



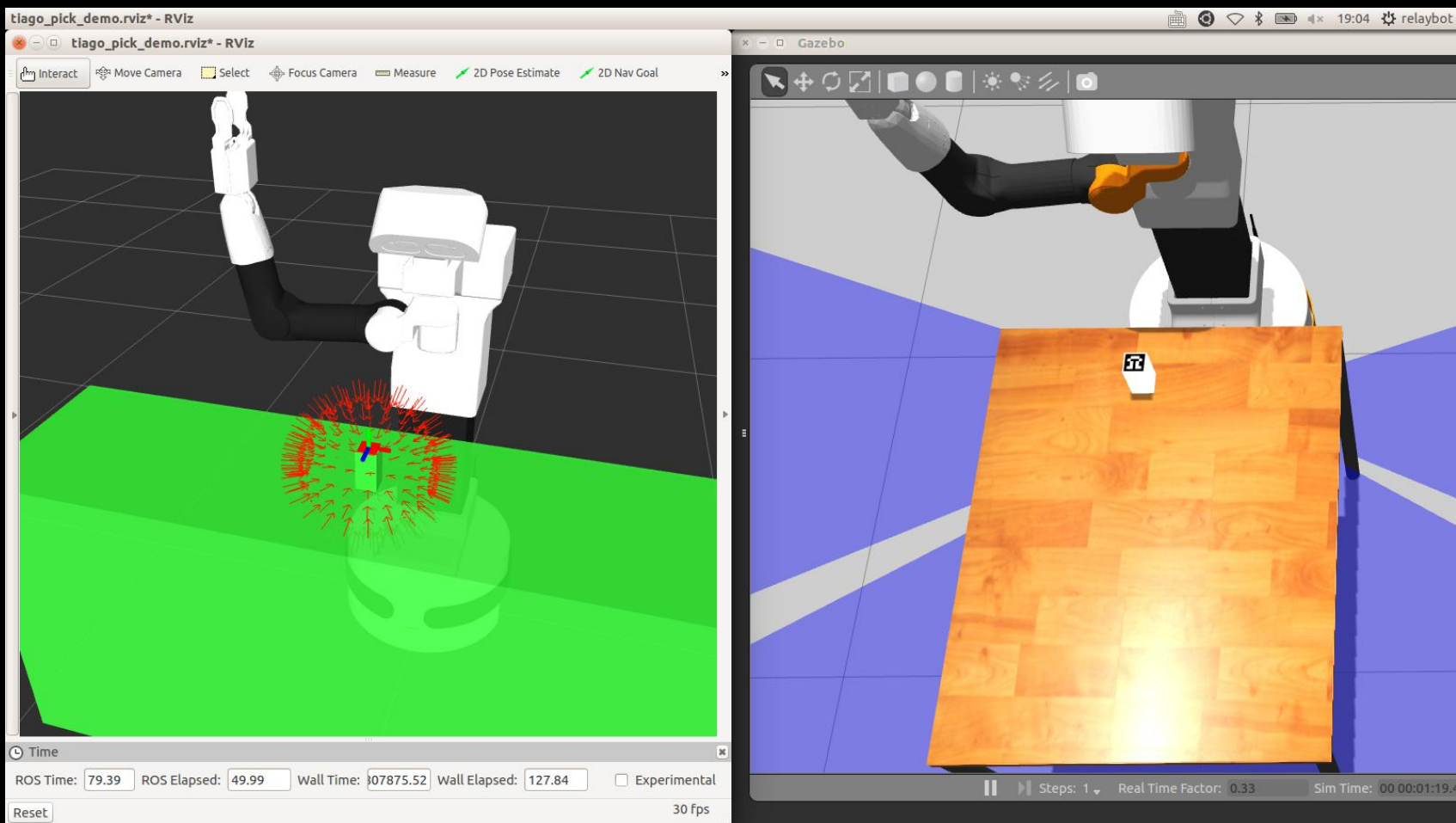
示例



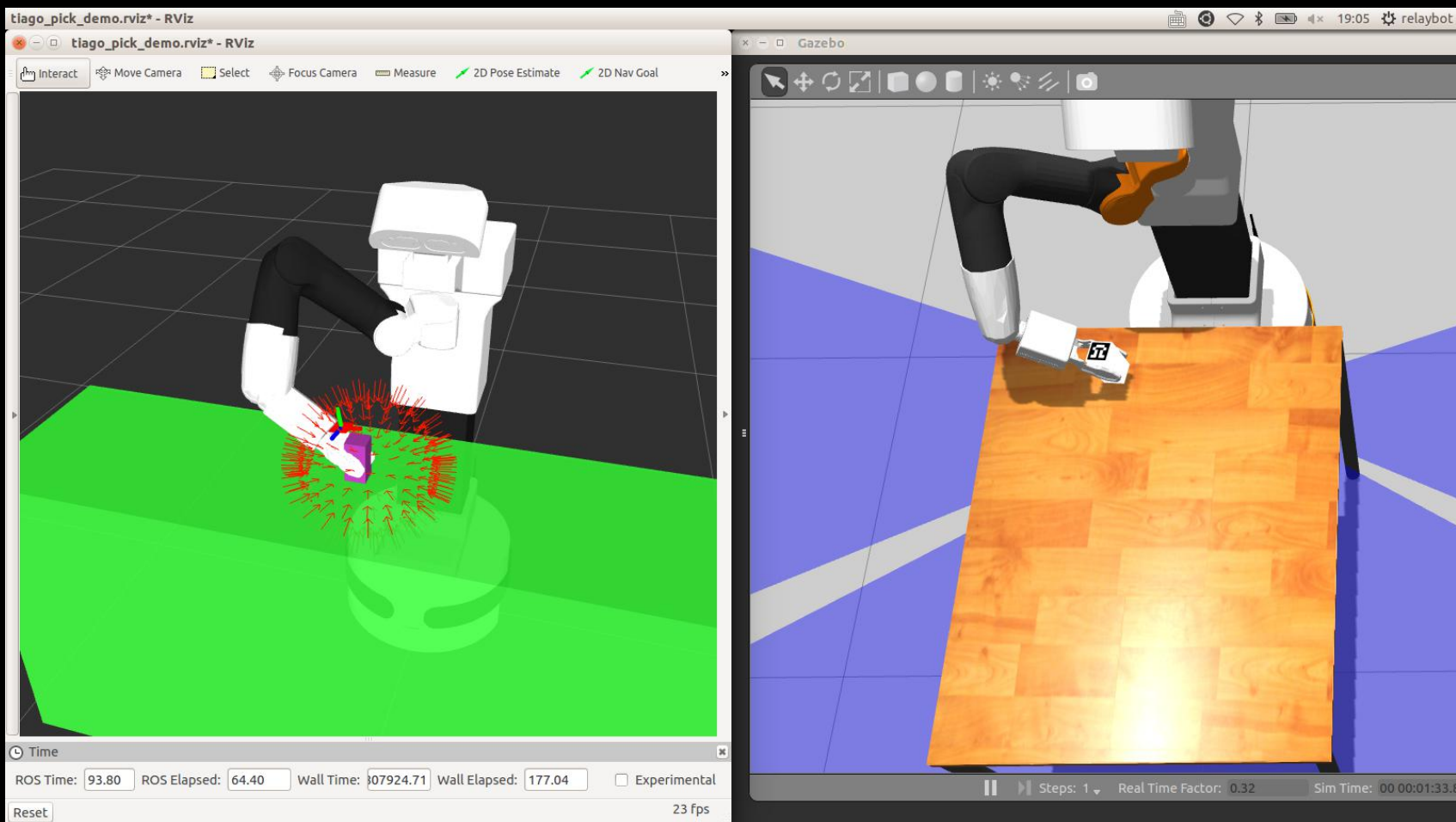
示例



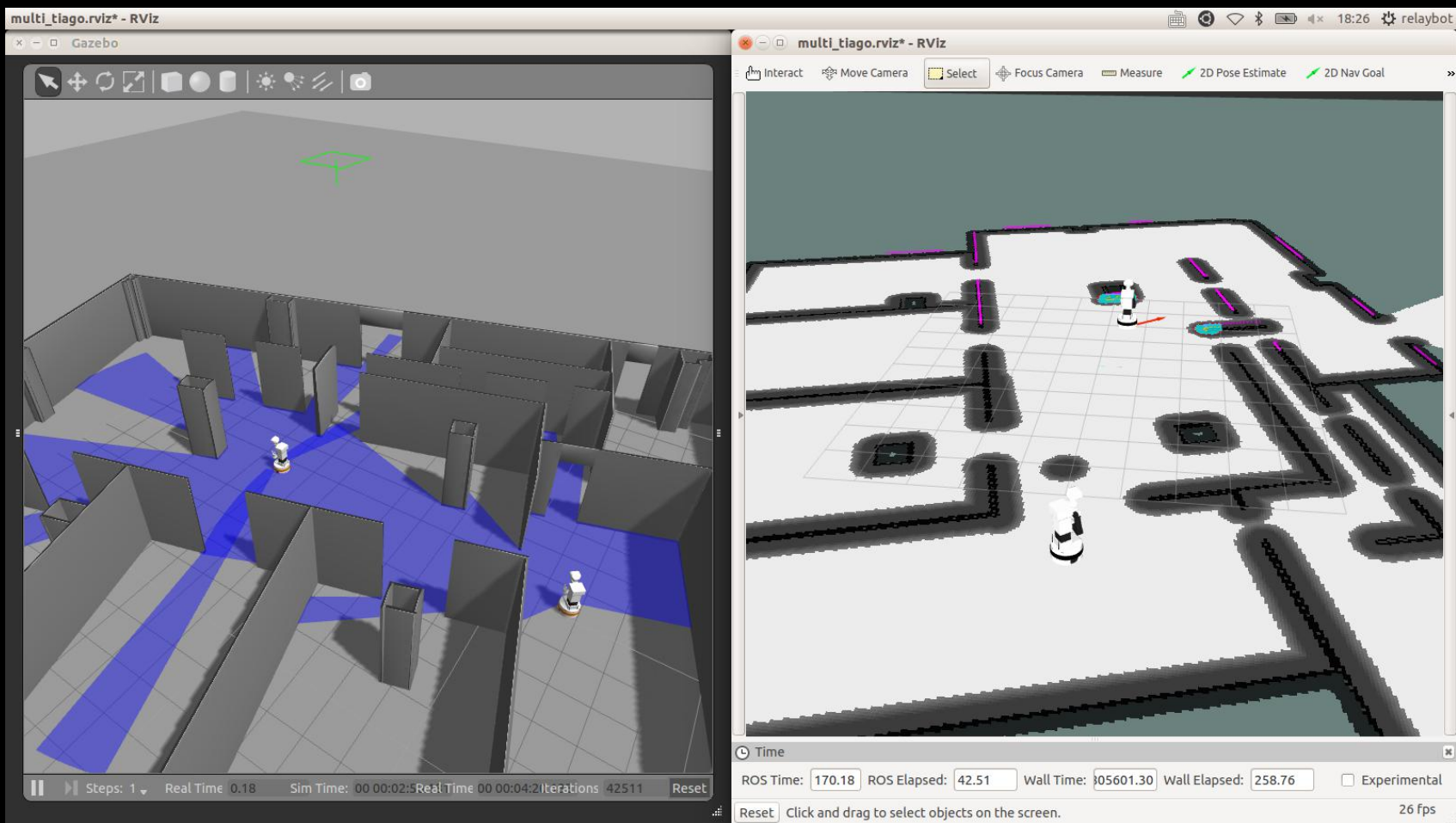
示例



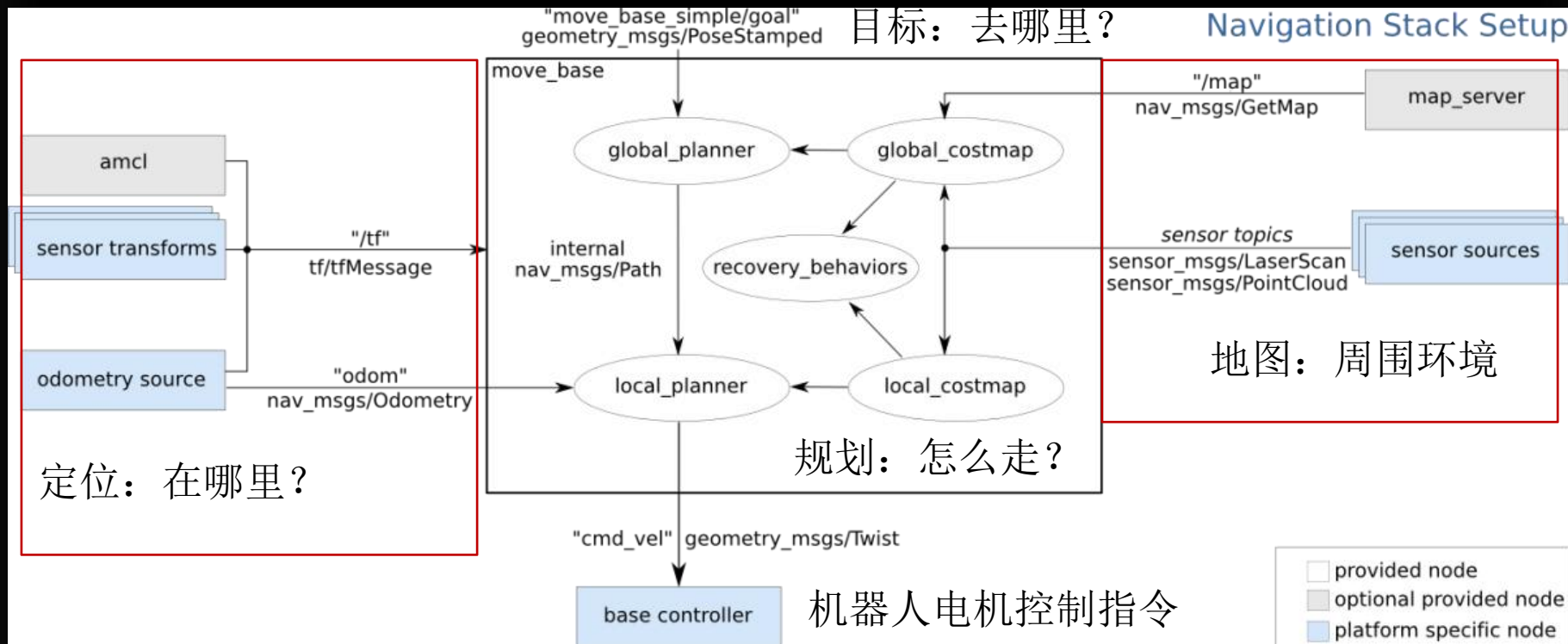
示例



示例

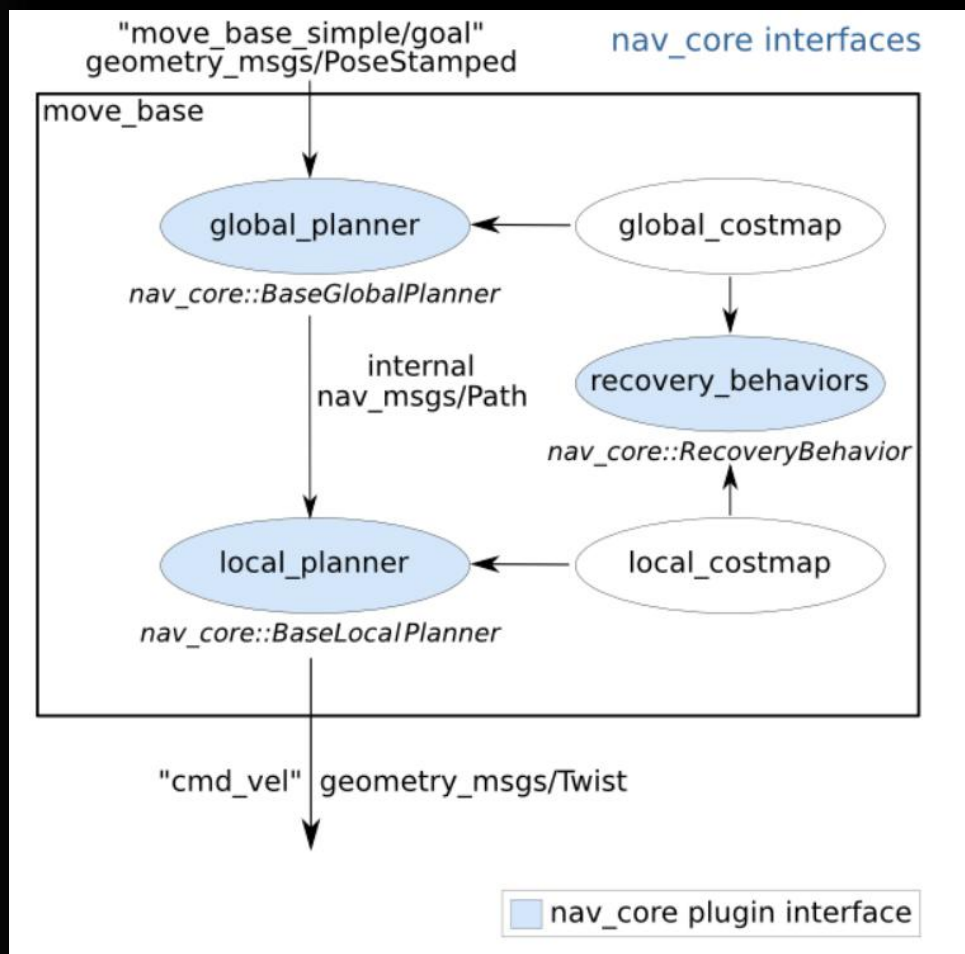


机器人配置



白色的部分是必须且已实现的组件，灰色的部分是可选且已实现的组件，蓝色的部分是必须为每一个机器人平台创建的组件。

导航功能包核心



nav_core: 使用 `pluginlib` 定义了各个组件的接口

costmap: 通过不同的 `costmap cost` 取值方法, 不同的传感器输入, 不同的 `layer` 叠加可以最大程度上定制 `navigation stack`

planner: 针对不同的应用场景可以使用不同的 `planner` 或者自己定制

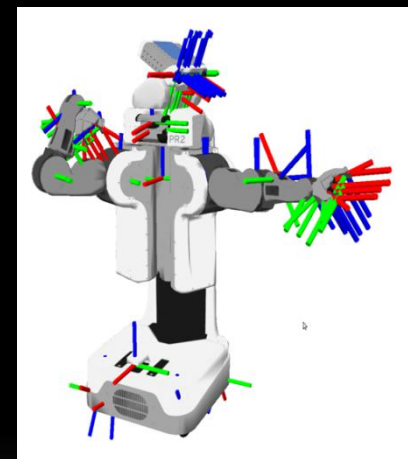
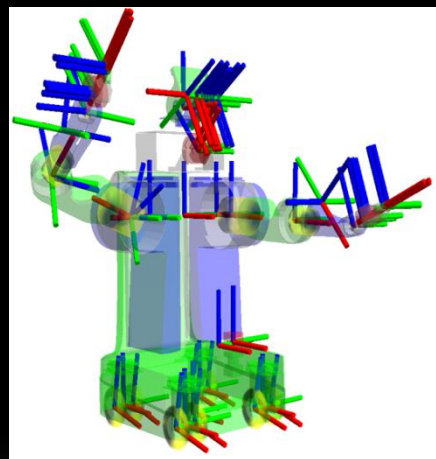
重要功能包

tf包

机器人系统一般都包含很多坐标系，如世界坐标系，机体坐标系，机械手坐标系，头坐标系等。tf是一个让使用者可以跟踪所有这些坐标系，它采用树型数据结构，根据时间缓冲并维护各个参考系的坐标变换关系，可以在任意时间完成2个参考系的坐标变换。

常用操作：

1. 监听坐标变换
2. 广播坐标变换



重要功能包

move_base包

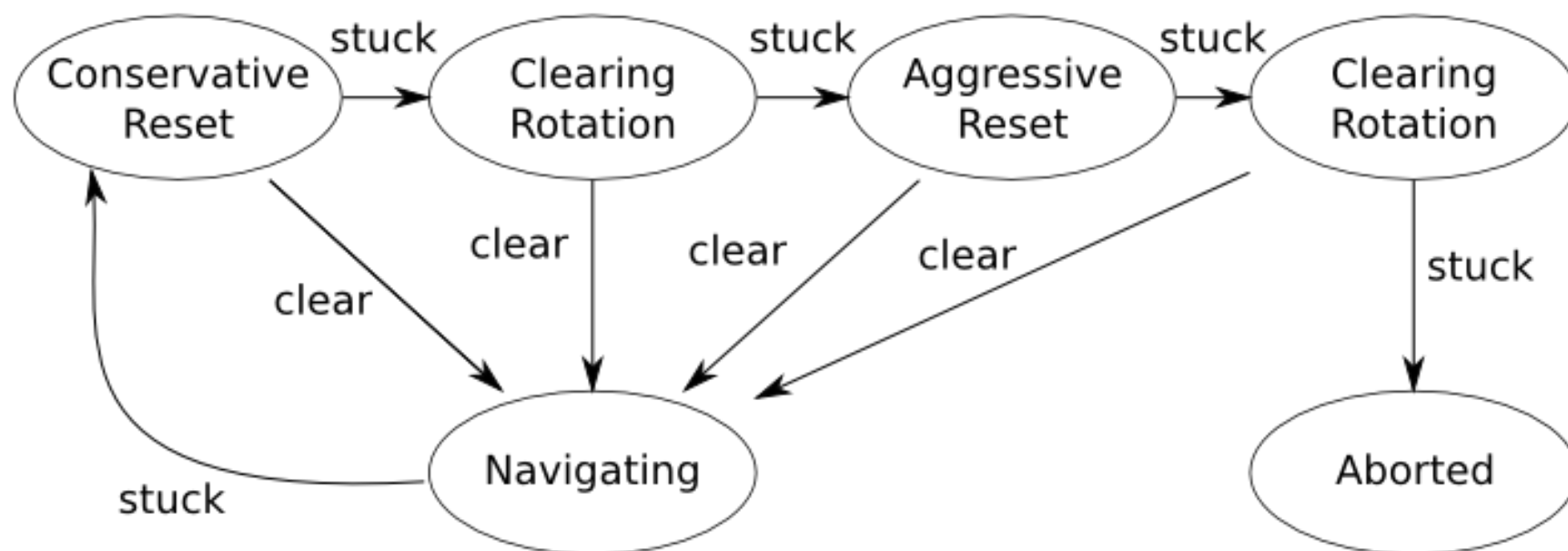
全局路径规划就是根据给定的目标位置进行总体路径的规划。局部实时规划就是根据附近障碍物进行躲避路线的规划。

一般先会通过全局路径规划，规划出一条从机器人当前位置到目标位置的全局路线。该包采用了Dijkstra算法，计算最小路径。局部实时规划是利用base_local_planner包实现的。该包使用Trajectory Rollout 和Dynamic Window approaches算法计算机器人每个周期内应该行驶的速度和角度（dx, dy, dtheta velocities）。base_local_planner这个包通过地图数据，通过算法搜索到达目标的多条路径，利用一些评价标准（是否会撞击障碍物，所需要的时间等等）选取最优的路径，并且计算所需要的实时速度和角度。

重要功能包

move_base包

move_base Default Recovery Behaviors



重要功能包

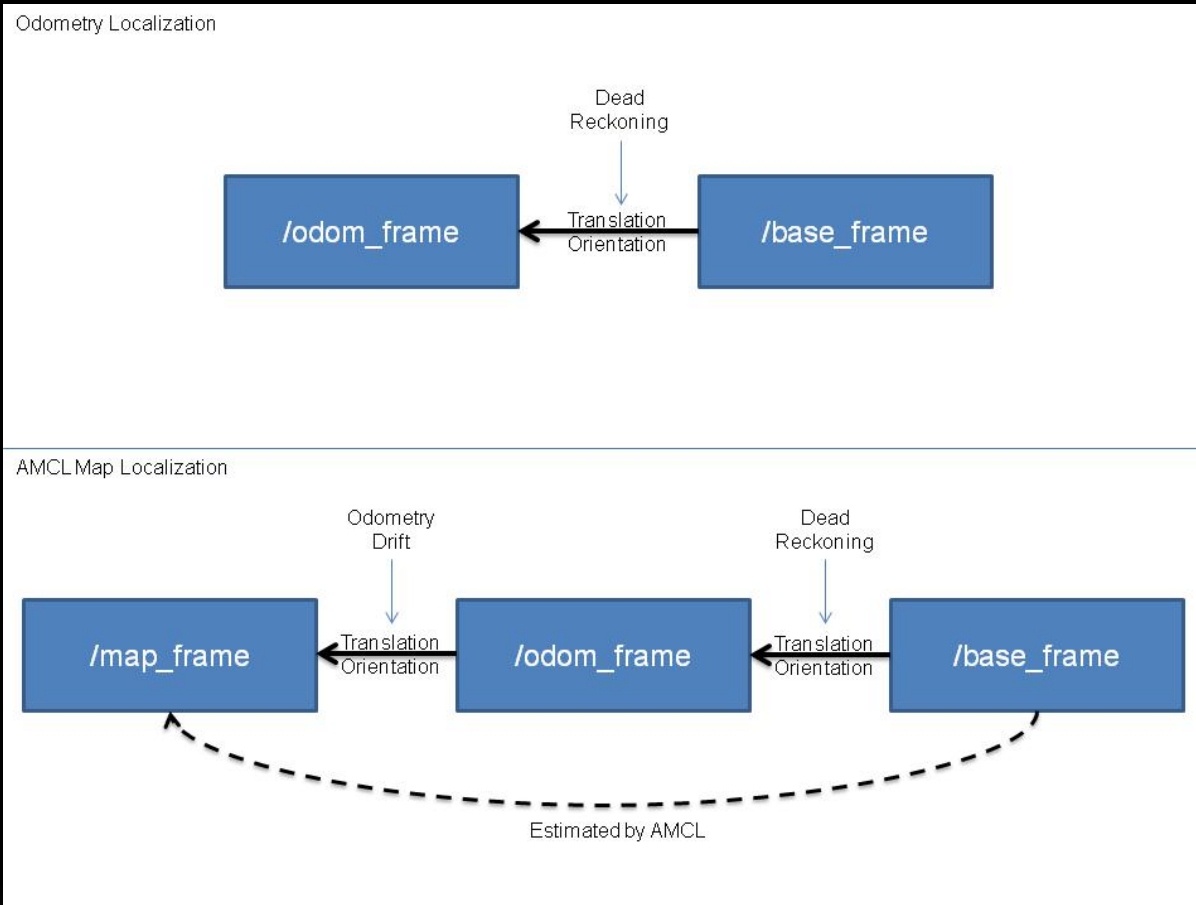
amcl包

amcl是移动机器人二维环境下的概率定位系统,其中针对已有的地图使用粒子滤波器跟踪机器人的位置。如果要详细的了解该算法,最好看下概率机器人这本书(**Probabilistic Robotics**)。

当前ROS上的算法能处理雷达扫描的数据和雷达地图,接受雷达地图、雷达扫描数据和坐标系的变换,输出位置估计。

重要功能包

amcl包



重要功能包

gmapping包

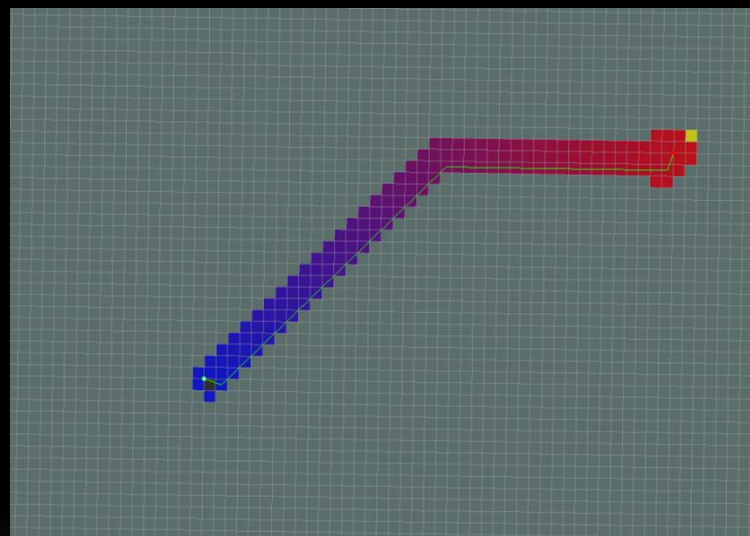
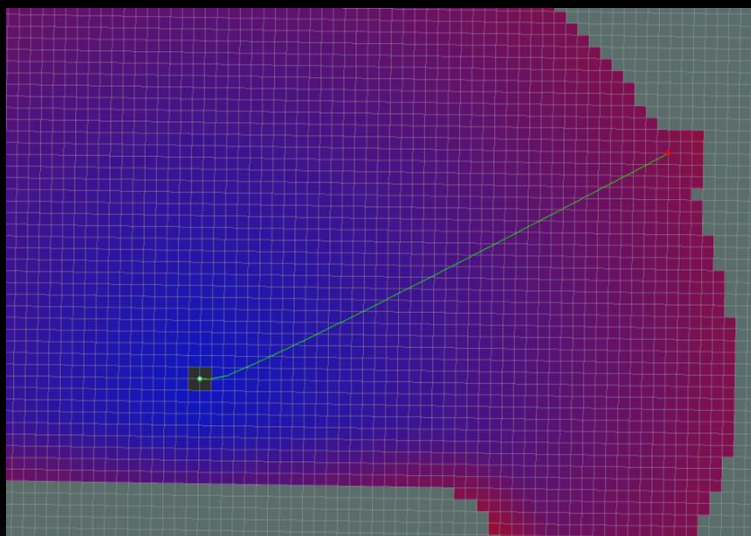
gmapping包提供基于雷达的slam，可以建立一个2D的占据方格地图。

使用时需要机器人提供里程计的数据，而且水平安装一个固定的激光雷达或将深度视觉信息转为雷达数据。

全局规划器

Dijkstra算法是典型最短路算法，用于计算一个节点到其他所有节点的最短路径。

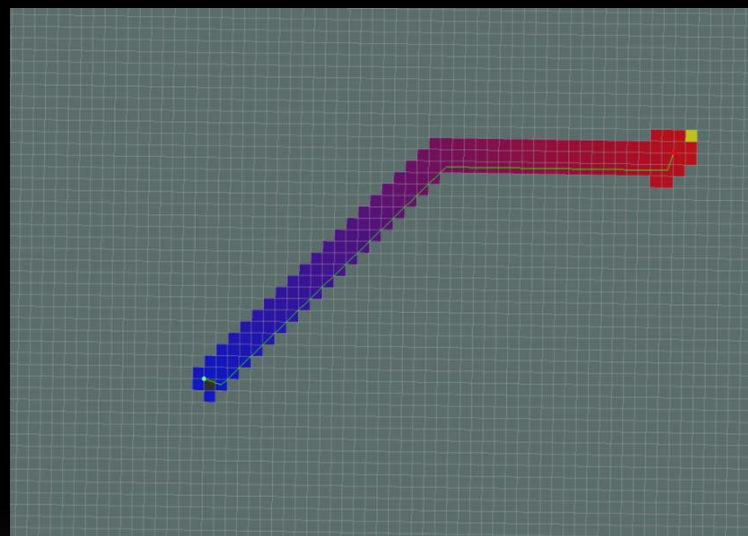
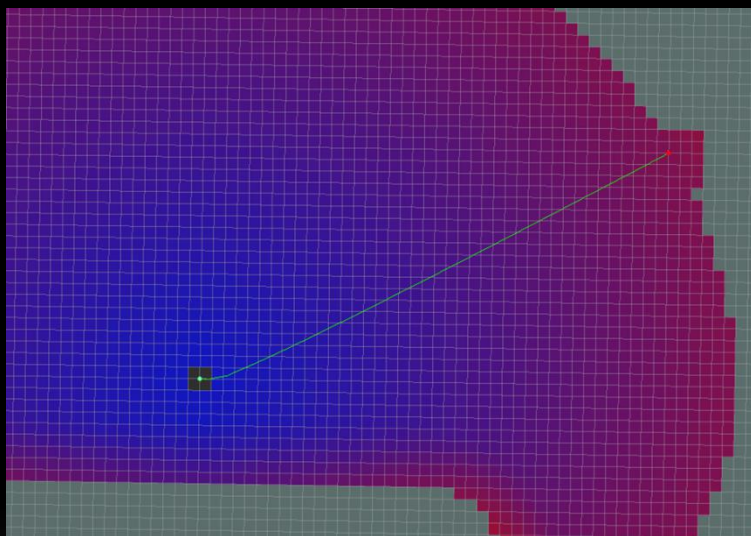
A*算法和Dijkstra算法一样可以搜索很大的区域，不过它多了一个启发函数来引导自己。



全局规划器

在A*算法中，有以下几种常用的启发函数：

- 曼哈顿距离
- 对角线距离
- 欧几里得距离
- 平方后的欧几里得距离



局部规划器

局部规划期使用Trajectory Rollout和Dynamic Window approaches算法计算机器人每个周期内应该行驶的速度和角度，有差别但流程类似，如下：

- 1 采样机器人当前的状态（ $dx, dy, dtheta$ ）；
- 2 针对每个采样的速度，计算机器人以该速度行驶一段时间后的状态，得出一条行驶的路线；
- 3 利用一些评价标准为多条路线打分；
- 4 根据打分，选择最优路径；
- 5 重复上面过程。

