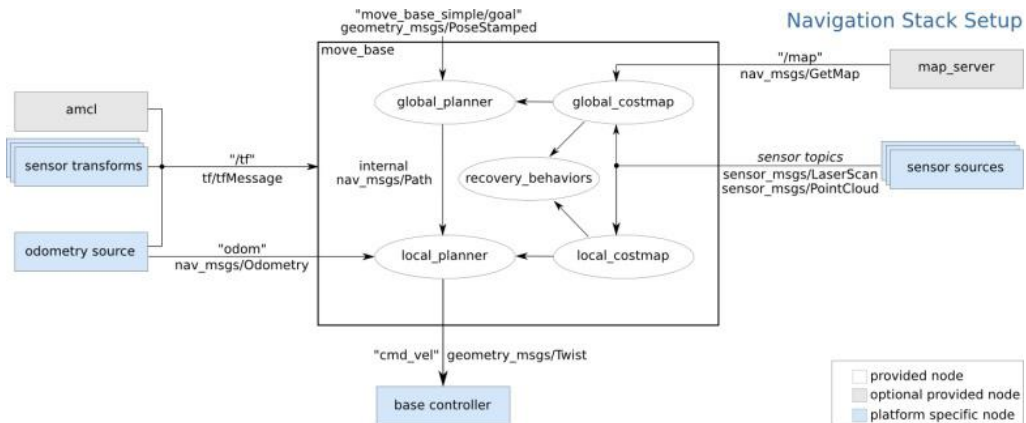


move_base 介绍

AI 航团队

1. 总体框架



在总体框架图中可以看到，move_base 提供了 ROS 导航的配置、运行、交互接口，它为一个 action 提供了实现的途径（详情参见 actionlib 包）。假如在地图内给定一个目标，这个操作将尝试控制一个移动基座到达这个目标。

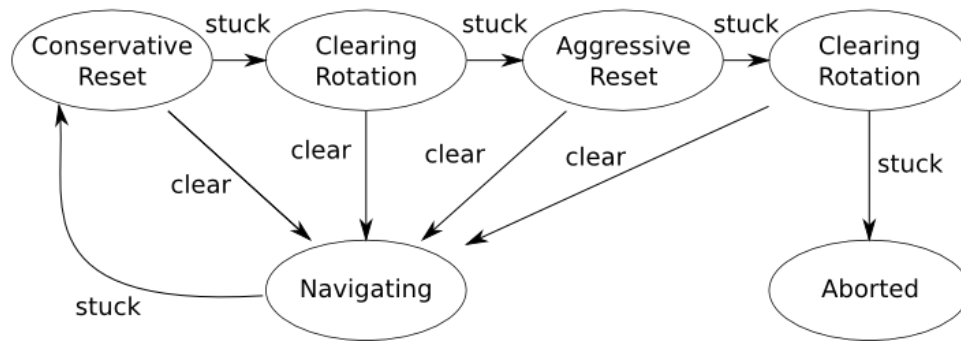
move_base 节点将全局路径和局部路径规划程序链接在一起，以完成其全局导航任务。

该包提供了作为导航功能包集的主要组件的 move_base ROS 节点。此节点及其配置选项的详细说明如下。

move_base 节点的高级视图及其与其他组件的交互如上所示。蓝色部分基于机器人平台而变化，所有平台都能提供可选的灰色部分和不可少的白色部分。

2. move_base 节点进行流程

move_base Default Recovery Behaviors



- 1) 在机器人上正确运行 `move_base` 节点（详情请看 [navigation stack documentation](#)），来让机器人尝试到达在使用者允许的误差范围内目标位置。
- 2) 在没有动态障碍的情况下，`move_base` 节点最终运行结果要么是接近了规划的目标，要么是给用户返回失败信号。
- 3) 当机器人认为它是卡住了，`move_base` 节点可以选择执行修复操作。默认情况下，`move_base` 节点采用如下所述行动来清理出周围空间：

首先，用户指定区域以外的障碍物将从机器人的地图中清除。如果失败，机器人会更激进的将其可以就地旋转的矩形框外面的障碍物全部移除。

紧接着下一个就地旋转操作会被继续执行。如果所有这些操作都失效，机器人会认为目标设定不恰当并通知用户放弃操作。

这些修复操作可以通过 `recovery_behaviors` 参数来配置，也可以 `recovery_behavior_enabled` 参数 `disable`。

3. move_base 插件

- 1) `base_local_planner` 插件：

`base_local_planner`: 实现了 Trajectory Rollout 和 DWA 两种局部规划算法。

`dwa_local_planner`: 实现了 DWA 局部规划算法，可以看作是 `base_local_planner` 的改进版本。

- 2) `base_global_planner` 插件：

`parrot_planner`: 实现了较简单的全局规划算法

`navfn`: 实现了 Dijkstra 和 A*全局规划算法

`global_planner`: 重新实现了 Dijkstra 和 A*全局规划算法，可以看作 `navfn` 的改进版

3) `recovery_behavior` 插件:

`clear_costmap_recovery`: 实现了清除代价地图的恢复行为

`rotate_recovery`: 实现了旋转的恢复行为

`move_slow_and_clear`: 实现了缓慢移动的恢复行为

除了以上三个需要指定的插件外, 还有一个 `costmap` 插件, 该插件默认已经选择好, 无法更改。

以上所有的插件都是继承于 `nav_core` 里的接口, `nav_core` 属于一个接口 `package`, 它只定义了三种插件的规范, 也可以说定义了三种接口类, 然后分别由以上的插件来继承和实现这些接口。因此如果你要研究路径规划算法, 不妨研究一下 `nav_core` 定义的路径规划工作流程, 然后仿照 `dwa_local_planner` 或其他插件来实现。

除了以上三个需要指定的插件外, 还有一个 `costmap` 插件, 该插件默认已经选择好, 默认即为 `costmap_2d`, 不可更改, 但 `costmap_2d` 提供了不同的 `Layer` 可以供我们设置, 在 9.3 节我们会进行介绍。

在这里插件的概念并不是我们抽象的描述, 而是在 ROS 里 `catkin` 编译系统能够认出的, 并且与其他节点能够耦合的 C++ 库, 插件是可以动态加载的类, 也就是说插件不需要提前链接到 ROS 的程序上, 只需在运行时加载插件就可以调用其中的功能。

4. `move_base` 的 topic 和 service

`move_base` 输入输出的 Topic 在 315 节已经做了介绍, 这里不再赘述。

`move_base` 包含的 Service 包括:

`make_plan`: `nav_msgs/GetPlan` 类型, 请求为一个目标点, 响应为规划的轨迹, 但不执行该轨迹。

`clear_unknown_space`: `std_srvs/Empty` 类型, 允许用户清除未知区域地图。

`clear_costmaps`: `std_srvs/Empty` 类型, 允许用户清除代价地图上的障碍物。