

# Navigation

AI 航团队

## 1. Navigation

Navigation 是机器人最基本的功能之一，ROS 为我们提供了一整套 Navigation 的解决方案，包括全局与局部的路径规划、代价地图、异常行为恢复、地图服务器等等，这些开源工具包极大地减少了我们开发的工作量，任何一套移动机器人硬件平台经过这套方案就可以快速部署实现。

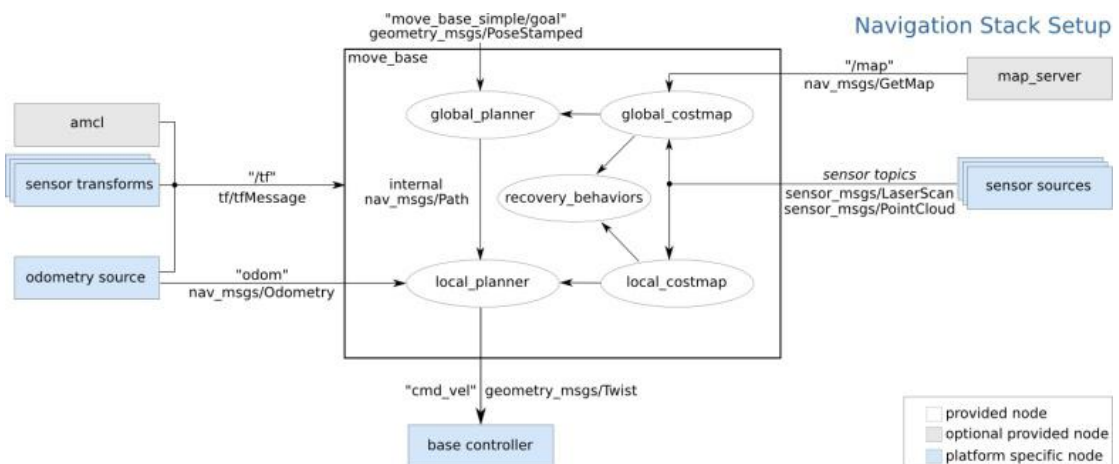
## 2. Navigation Stack

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。

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。

包名	功能
amcl	定位
fake_localization	定位
map_server	提供地图
move_base	路径规划节点
nav_core	路径规划的接口类，包括 base_local_planner、base_global_planner 和 recovery_behavior 三个接口
base_local_planner	实现了 Trajectory Rollout 和 DWA 两种局部规划算法
dwa_local_planner	重新实现了 DWA 局部规划算法
carrot_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
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
costmap_2d	二维代价地图
voxel_grid	三维小方块（体素网格）
robot_pose_ekf	机器人位姿的卡尔曼滤波



机器人导航中最重要的 stack 就是图中采用方框勾画的 move\_base 节点，在实际导航任务中，需要启动此节点，此节点订阅数据，并可以发布车体运行路径和速度。此节点包含了很多插件，包括 global\_planner、local\_planner、global\_costmap、local\_costmap、recovery\_behaviors。这些插件用于负责一些更细微的任务：全局规划、局部规划、全局地图、局部地图、恢复行为。而每一个插件其实也都是一个 package，放在 Navigation Stack 里。

### 3. 订阅

- 1) /tf:提供的 tf 包括 map\_frame、odom\_frame、base\_frame 以及机器人各关节之间的完成的一棵 tf 树。
- 2) /odom:里程计信息
- 3) /scan 或 /pointcloud:传感器的输入信息，最常用的是激光雷达(sensor\_msgs/LaserScan 类型)，也有用点云数据(sensor\_msgs/PointCloud)的。

4) /map: 地图, 可以由 SLAM 程序来提供, 也可以由 map\_server 来指定已知地图。

以上四个 Topic 是必须持续提供给导航系统的, 下面一个是可随时发布的 topic:

move\_base\_simple/goal: 目标点位置。

需要注意的是对于路径规划而言, 地图和定位是已知的, 是需要其他节点提供

## 4. 发布

/cmd\_vel: geometry\_msgs/Twist 类型, 为每一时刻规划的速度信息。