**global\_planner**

   在目前的ROS版本中，机器人全局路径规划使用的是navfn包，这在move\_base的默认参数中可以找到 base\_global\_planner  (string, default: "navfn/NavfnROS")。而在navigation的源代码中还有一个global\_planner的包，该包的源文件夹(navigation-hydro-devel\global\_planner\src )下已经有了A\*,Dijkstra等算法的实现。可是navfn的源程序中也有这两个算法的实现，貌似根本就没用到global\_planner这个文件夹下的源程序。因此最开始直接看用于move\_base全局导航的程序时有点一头雾水，为什么有两个用于全局导航的包在ROS里面？到底这两个包navfn和global\_planner是什么关系？

       可以将navfn包和global\_planner包理解成一个并列关系，因为他们两个都是用来做全局规划的，两个包里面也都实现了A\*,Dijkstra算法。那是不是意味着这两者中的一个包就是多余的呢？其实不是，早期的开发中是用navfn包做导航的，那时候并没有global\_planner这个包，并且在navfn的源代码里可以看到这个包默认是使用Dijkstra做全局路径规划，并且有A\*的代码，那为什么没有使用A\*呢？幸好有人在ROS answers里问了这个问题，也引来了众开发者回答：

      根据12年bhaskara的回答，意思是navfn里的A\*算法存在bug，没人有时间去弄（在ros的各种答案里经常可以看到开发者说没时间弄，他们确实也相当的忙），直到13年David Lu 才完成了这部分工作，重新发布了global\_planner包，修改好的代码封装性更强，更清晰明了。因此，也可以认为global\_planner是navfn的替代者。也有人问David Lu为什么没用global\_planner替代掉navfn？他的回答是为了和以前兼容。因此可以看到源代码中两个包都在，并且move\_base的那个全局变量参数默认的是navfn，也就是说没用global\_planner。那么如何使用global\_planner包而不是navfn包呢？按理来说，只要将move\_base的参数base\_global\_planner用global\_planner/PlannerCore替代就行了。如：

|  |
| --- |
| <nodepkg="move\_base" type="move\_base" respawn="false"name="move\_base" output="screen">   <param name="base\_global\_planner"value="global\_planner/GlobalPlanner"/> |

插件用来通知ROS我们将使用它，具体过程见官方wiki(ROS的插件机制)。弄完插件机制以后，我们就可以像上面一样将其用参数的形式直接传过去了。之前的bgp\_plugin.xml有bug，因此就出现了那个问题。

       可是，我随便按照自己的思路去写global 或者local planner的函数就行吗？显然ROS有一个标准。你必须按照它提供给你的模板去实现你自己的算法，这些模板就是基类。如果觉得抽象，可以先通过官方的文档来了解插件是如何工作的？

     要写global 或者local planner的插件，须按照ROS提供的模板去实现，而这正是nav\_core这个包存在的意义。在navigation的源代码中你会看到这个nav\_core包中仅仅只有几个头文件，正是这些头文件提供了多个模板： nav\_core::BaseGlobalPlanner，nav\_core::BaseLocalPlanner， nav\_core::RecoveryBehavior，在官方的wiki文档里可以看到他们的相关介绍。所以按照这些模板的标准形式去写自己的planner算法就行了。

       了解了这些，也十分有助于我们去看别人写的各种planner的插件，如使用遗传算法的global planner，sbql global planner以及eband local planner等，当然也包括官方提供的用来完善之前遗留程序问题的插件包global\_planner包和dwa\_local\_planner包。