



移动机器人系列课程

机器人定位与导航系列



深圳市史河机器人科技有限公司

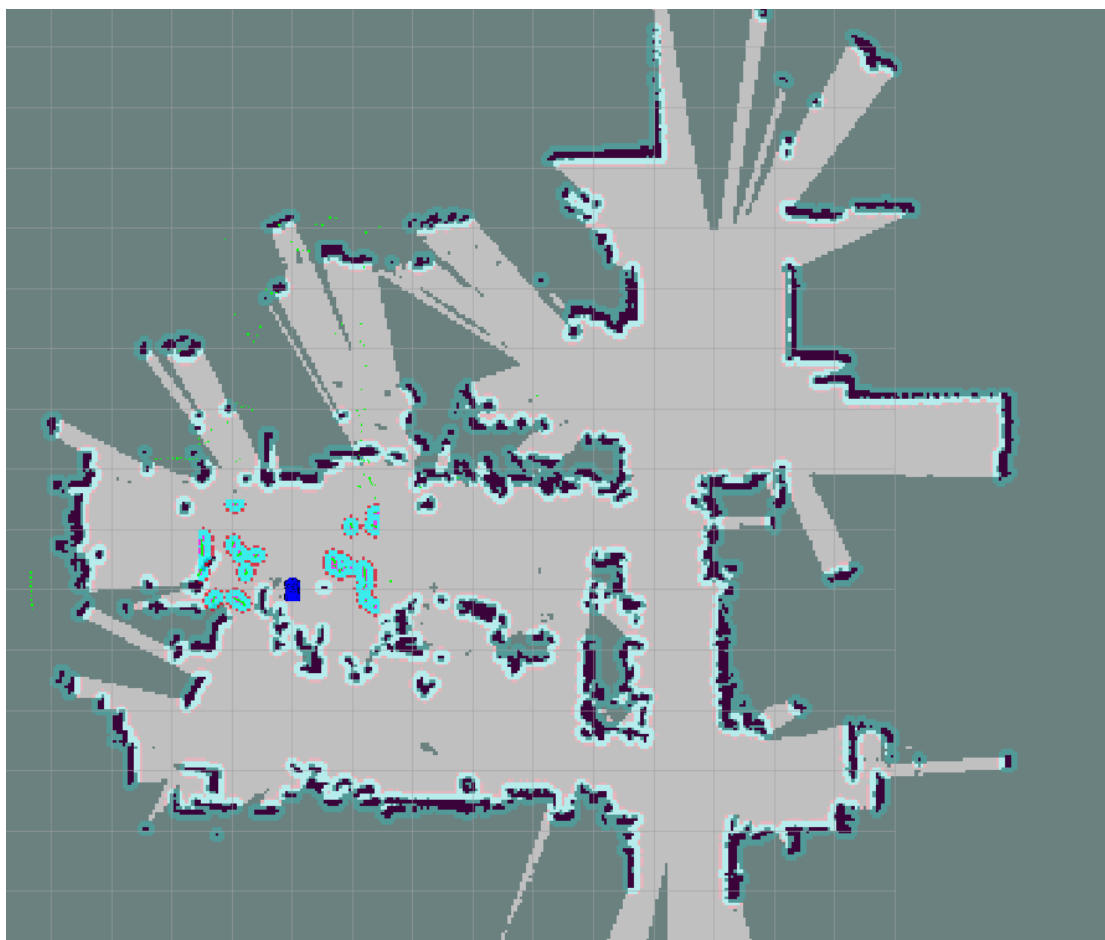
SLAM 导航实践操作

在之前的内容我们运用了 Cartographer 算法进行了建图。本节的内容是在之前建立的地图基础上进行的导航实践，如果没有建图请进行建图操作。

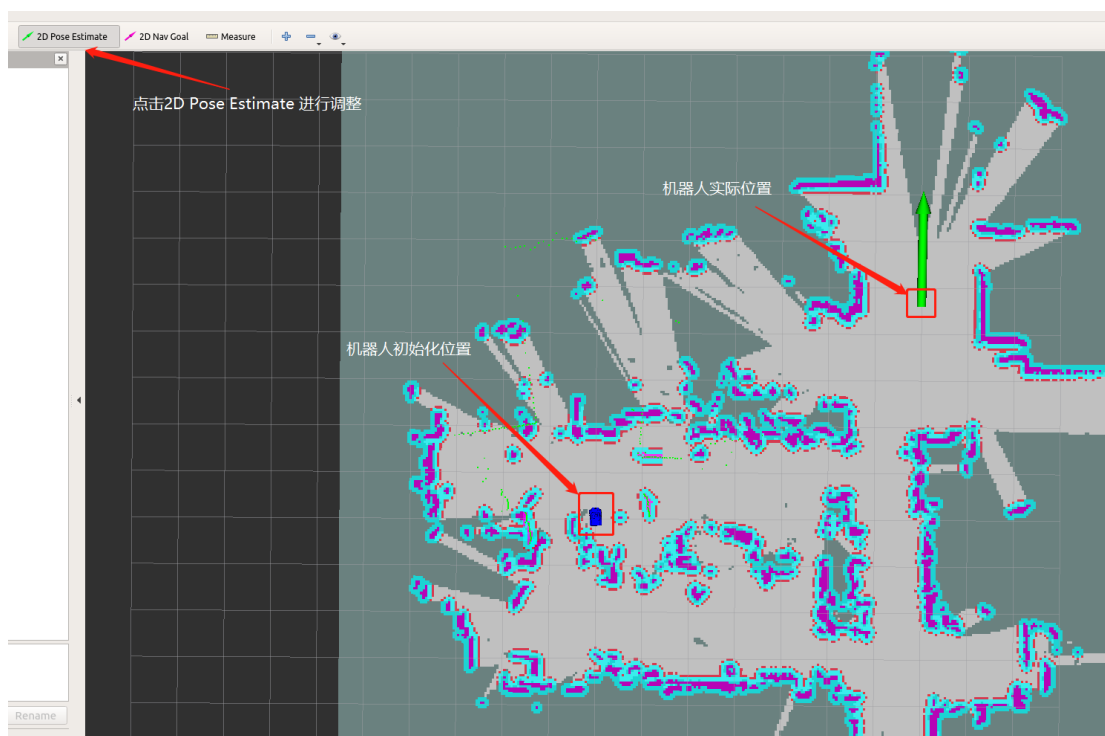
1. 运行导航窗口

打开一个终端，输入以下指令加载之前保存的地图。

```
roslaunch robuster_mr_navigation navigation.launch
```



这个加载需要一点时间，加载完成后我们能发现，机器人的实际位置与地图中的位置不一样，有偏差。解决这个问题方法为：用[2D Pose Estimate]按钮手动给定一个正确的初始位置。点击[2D Pose Estimate]按钮，然后将鼠标放置到机器人在地图中实际应该的位置，按住鼠标并拖动鼠标来完成机器人朝向的设置，最后让机器人原地旋转 3 圈，完成位置的设定。



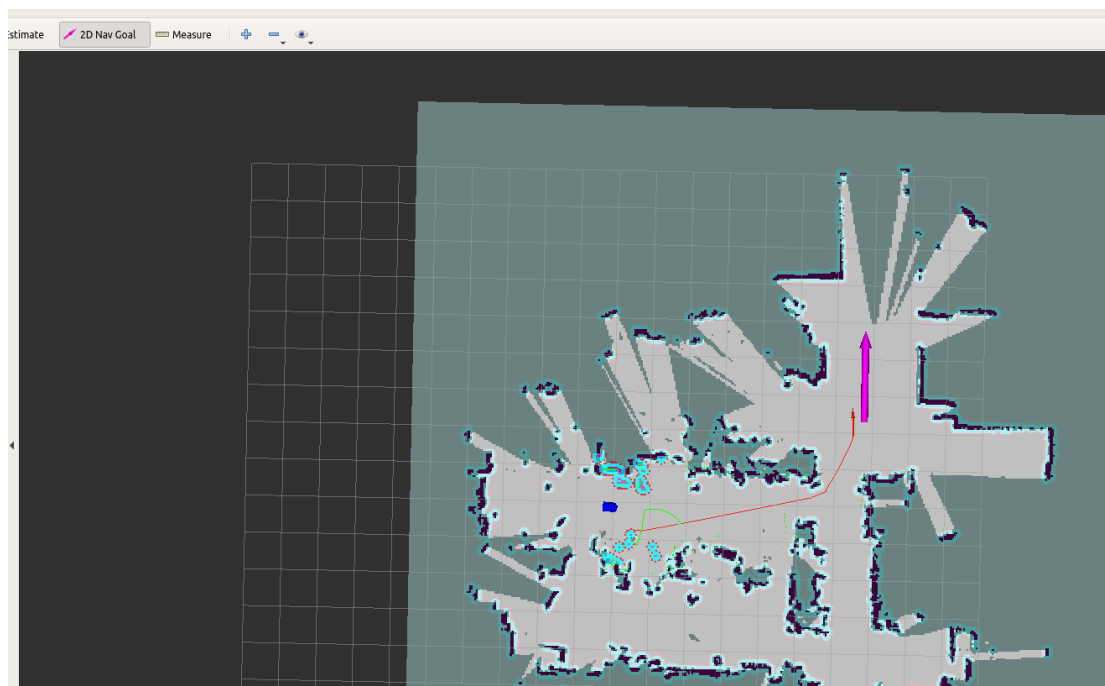
2. 2D Nav Goal 进行实施导航避障

右键点击 2D Nav Goal 按钮，在地图上右键点击选择点，按住不松调整机器人终点姿态方向，松开右键，开始导航。如果不小心中选错点，可以在调整方向的时候按下 Esc 键取消位置选择。如果不小心中点击太快还没调整好机器人的终端姿态就开始导航了，可以利用遥控控制下机器人，就退出当前导航。

在导航的时候也会存在路径规划出错的情况，出现这种情况我们先让机器人停止。出现导航路径规划出错的情况，有可能是地图建的还不太完全，例如如果你选择下面地图中比较

边缘的位置就有可能存在路径规划不成功现象。如果想要解决这个问题,需要重新进行建图,将需要导航的位置包含进地图。

还有一种比较常见的现象是,机器人没有识别到某些障碍物,导致无法执行导航的路径规划,这是激光雷达的原因,受限于本身视角和安装位置,有些地方是扫描不到。这样在建图的时候有些物体是没有扫描进地图,进而在路径规划的过程中出现路径实际无法通过的现象。



3. Teb_local_planner 参数配置

在前文中我们应用 2D Nav Goal 进行导航,但是该如何对导航的参数进行调整优化?

例如:如何修改导航速度呢?接下来我们将了解这个问题。

在介绍导航 ROS 框架时,我们介绍了 BR280 全局规划采用的是 global_planner,局部路径规划采用的是 teb_local_planner。在具体应用中 global_planner 在配置时几乎不怎么需要配置,需要具体配置的是局部路径规划,就 BR280 机器人而言就是具体如何配置

teb_local_planner。完整的介绍可以参考 ROS 官方的配置文档，

http://wiki.ros.org/teb_local_planner/Tutorials。

输入下面命令打开 teb_local_planner_params.yaml 所在的文件夹。

```
cd
```

```
~/mr300_ws/src/robuster_mr/robuster_mr_navigation/param/2d/outdoor_configs
```

我们也可以用 sublime 按照路径进行打开查看。下面将逐项解析各项参数的含义。

3.1 Trajectory

```
5 # Trajectory
6
7 teb_autosize: True
8 dt_ref: 0.45
9 dt_hysteresis: 0.05
10 global_plan_overwrite_orientation: True
11 max_global_plan_lookahead_dist: 3.0
12 feasibility_check_no_poses: 2
```

teb_autosize，默认值：True。优化期间允许改变轨迹的时域长度，TEB 通过状态搜索树寻找最优路径，而 dt_ref 则是最优路径上的两个相邻姿态即位置、速度、航向信息，可通过 TEB 可视化在 rviz 中看到）的默认距离此距离不固定，规划器自动根据速度大小调整这一距离，速度越大，相邻距离自然越大，较小的值理论上可提供更高精度。

dt_ref，最小值：0.01、默认值：0.3、最大值：1.0。局部路径规划的解析度，将 dt_ref 设置为与车辆长度大致是可行的。

dt_hysteresis，允许改变的时域解析度的浮动范围，一般为 dt_ref 的 10% 左右。

global_plan_overwrite_orientation，默认值：True。覆盖全局路径中局部路径点的朝向。

feasibility_check_no_poses，最小值：0、默认值：5、最大值：50。检测位姿可到达的时间间隔。

3.2 Robot

```
14 # Robot
15
16 max_vel_x: 0.2
17 max_vel_y: 0.0
18 max_vel_x_backwards: 0.05
19 max_vel_theta: 0.2 #0.7
20
21 acc_lim_x: 0.6
22 acc_lim_theta: 0.6
23
24 min_turning_radius: 0.15
25 wheelbase: 0.202
26 footprint_model: # types: "point", "circular", "two_circles", "line", "polygon"
27   type: "polygon"
28   #line_start: [0.0,0.0]
29   #line_end: [0.57,0.0]
30   vertices: [[0.2, 0.135], [-0.2, 0.135], [-0.2, -0.135], [0.2, -0.135]]
```

max_vel_x, 最小值: 0.01、默认值: 0.4、最大值: 100。最大 x 前向速度。

max_vel_y, 最小值: 0、默认值: 0、最大值: 100。最大 y 方向速度。

max_vel_x_backwards, 最小值: 0.01、默认值: 0.2、最大值: 100。最大 x 后退速度。将此速度设置为 0 或者负数将导致错误。

max_vel_theta, 最小值: 0.01、默认值: 0.3、最大值: 100。最大转向角速度

acc_lim_x, 最小值: 0.01、默认值: 0.5、最大值: 100。最大 x 加速度约束。ax_vel_x 和 acc_lim_x 这是速度与加速度约束。注意加速度同样约束减速过程。若电机性能良好且速度不快可以立即刹车, 可直接将 acc_lim_x 设置为 0, 表示没有约束。若电机不能承受阶跃输入或者响应时间很长, 则应当设置加速度限制。

acc_lim_theta, 最小值: 0.01、默认值: 0.5、最大值: 100。max_vel_theta 和 acc_lim_theta 角速度和角加速度约束。正确配置下最小转弯半径应在低速时生效, 而角速度限制应作用在车辆高速行驶时。角速度约束可以防止高重心转弯过快翻车, 角加速度限制可以避免车轮打滑, 车辆失控。对于低速平稳运行的车辆可以不约束此两项, 将二者设为一个足够大的值即可。

min_turning_radius, 最小值: 0、默认值: 0、最大值: 50。车类机器人的最小转弯

半径。为避免规划出不可能实现的移动路径，请动手测量实际车辆的转弯半径。此参数事实上约束了舵机的最大转角。有些车辆转向性能不佳，前轮实际转过角度小于舵机角度，则应当给指令转角乘上一增益后再控制舵机，否则车辆将总是不能实现设置的最小转弯半径。请注意此项应当与最大角速度配合设置。

Wheelbase，最小值：-10、默认值：0、最大值：10。驱动轴和转向轴之间的距离，对于后轮式机器人，该值可能为负值。

footprint_model，底盘模型。

3.3 GoalTolerance

```
31 # GoalTolerance
32
33 xy_goal_tolerance: 0.1
34 yaw_goal_tolerance: 0.1
35 free_goal_vel: False
36
```

xy_goal_tolerance，最小值：0.001、默认值：0.2、最大值：10。目标 xy 偏移容忍度

yaw_goal_tolerance，最小值：0.001、默认值：0.1、最大值：3.2。目标角度。偏移容忍度 xy_goal_tolerance 和 yaw_goal_tolerance 目标跟踪误差。根据车辆运行精度设置。如果车辆转向间隙特性很大，则不应设置严格的航向限制。

free_goal_vel: False，允许机器人以最大速度驶向目的地，为 False 时，车到达终点时的目标速度为 0；TEB 是时间最优规划器。缺少目标速度约束将导致车辆“全速冲线”，即使前方有一堵墙也是如此（因为撞墙的时刻不在规划器考虑范围内了）。不过此选项在竞速比赛中比较有用。

3.4 Obstacles

```
37 # Obstacles
38
39 min_obstacle_dist: 0.15 #0.3
40 inflation_dist: 0.2 #0.3
41 include_costmap_obstacles: True
42 costmap_obstacles_behind_robot_dist: 0.3 #1.0 #0.3
43 obstacle_poses_affected: 15
44 costmap_converter_plugin: ""
45 costmap_converter_spin_thread: True
46 costmap_converter_rate: 5
```

`min_obstacle_dist`, 最小值: 0、默认值: 0.5、最大值: 10.0。和障碍物最小距离。

`inflation_dist`, 最小值: 0、默认值: 0.6、最大值: 15.0。障碍物膨胀距离。

`include_costmap_obstacles`, 默认值: `False`。`costmap` 中的障碍物是否被直接考虑。

`costmap_obstacles_behind_robot_dist`, 最小值: 0、默认值: 1.5、最大值: 20.0。

规划时考虑后面 `n` 米内的障碍物。

`obstacle_poses_affected`, 最小值: 0、默认值: 30、最大值: 100。障碍物姿态受影响程度 0-30

`costmap_converter_plugin`, 激活 Costmap Converter 插件, 缺省名称采用系统默认名称。TEB 默认情况下不使用 Costmap Converter。事实上, 此插件可以在复杂场景下极大提高运算效率, 尤其是处理激光雷达分散的测量数据时。因为将障碍物视为系列孤立点效率极低。原始的代价地图是由栅格地图中单元格组成, 用于表示障碍物, 但单元格占用的计算资源较大, 故采用插件将单元格转换成点, 线, 多边形表示。

`costmap_converter_spin_thread`, 默认 `True`。如果为 `true`, 则 `costmap` 转换器将以不同的线程调用其回调队列。

`costmap_converter_rate`, 定义 `costmap_converter` 插件处理当前 `costmap` 的频率, 该值不高于 `costmap` 更新率。

3.5 Optimization

```
48 # Optimization
49
50 no_inner_iterations: 5
51 no_outer_iterations: 4
52 optimization_activate: True
53 optimization_verbose: False
54 penalty_epsilon: 0.03
55 weight_max_vel_x: 1
56 weight_max_vel_theta: 3 #1
57 weight_acc_lim_x: 1
58 weight_acc_lim_theta: 1
59 weight_viapoint: 5
60 weight_kinematics_nh: 1000 #1000
61 weight_kinematics_forward_drive: 1000
62 weight_kinematics_turning_radius: 0.0
63 weight_optimaltime: 1
64 weight_obstacle: 50
65 weight_dynamic_obstacle: 10 # not in use yet
66 #alternative_time_cost: False # not in use yet
67 selection_alternative_time_cost: False
```

no_inner_iterations, 最小值: 0、默认值: 5、最大值: 100。被外循环调用后内循环执行优化次数。

no_outer_iterations, 最小值: 0、默认值: 4、最大值: 100。执行的外循环的优化次数执行的外循环的优化次数。每次外部循环迭代都会根据所需的时间分辨率 dt_ref 自动调整轨迹的大小, 并调用内部优化器 (执行 no_inner_iterations)。因此, 每个计划周期中求解程序迭代的总数是两个值的乘积。

optimization_activate, 默认值: True。激活优化。

optimization_verbose, 默认值: False。打印优化过程详情。

penalty_epsilon, 最小值: 0、默认值: 0.1、最大值: 1.0。对于硬约束近似, 在惩罚函数中添加安全范围。

weight_max_vel_x, 最大 x 速度权重。

weight_max_vel_theta, 最大角速度权重。

weight_acc_lim_x, 最大 x 加速度权重。

weight_acc_lim_theta, 最大角速度权重。

weight_kinematics_nh, 非完整运动学的优化权重。

weight_kinematics_forward_drive, 优化过程中, 迫使机器人只选择前进方向, 差速轮适用。

weight_kinematics_turning_radius, 优化过程中, 车型机器人的最小转弯半径的权重

weight_optimaltime, 优化过程中, 基于轨迹的时间上的权重 weight_shortest_path: 0

weight_obstacle, 优化过程中, 和障碍物最小距离的权重 0~50

weight_inflation, 优化过程中, 膨胀区的权重

weight_dynamic_obstacle, 优化过程中, 和动态障碍物最小距离的权重

weight_dynamic_obstacle_inflation, 优化过程中, 和动态障碍物膨胀区的权重 0~50

selection_alternative_time_cost, 默认 False。则将时间成本 (时间差平方的总和) 替换为总过渡时间 (时间差之和)。

3.6 Homotopy Class Planner

```
69 # Homotopy Class Planner
70
71 enable_homotopy_class_planning: False
72 enable_multithreading: True
73 simple_exploration: False
74 max_number_classes: 4
75 roadmap_graph_no_samples: 15
76 roadmap_graph_area_width: 5
77 h_signature_prescaler: 0.5
78 h_signature_threshold: 0.1
79 obstacle_keypoint_offset: 0.1
80 obstacle_heading_threshold: 0.45
81 visualize_hc_graph: False
82 global_plan_viapoint_sep: 0.1
```

`enable_homotopy_class_planning`, 默认 `True`。激活并行规划（因为一次优化多个轨迹，占用更多的 CPU 资源。

`enable_multithreading`, 默认值 `True`。激活多个线程，以便在不同的线程中规划每个轨迹。

`simple_exploration: False`

`max_number_classes`, 最小值：1、默认值：5、最大值：100。考虑到的不同轨迹的最大数量。

`roadmap_graph_no_samples`, 最小值：1、默认值：15、最大值：100。指定为创建路线图而生成的样本数。

`roadmap_graph_area_width`, 最小值：0.1、默认值：5.0、最大值：20.0。指定该区域的宽度。

`h_signature_prescaler`, 最小值：0.2、默认值：1.0、最大值：1.0。 $(0.2 < \text{value} \leq 1)$ 缩放用于区分同伦类的内部参数（H-signature）。警告：只能减少此参数，如果在局部 `costmap` 中遇到太多障碍物的情况，请勿选择极低值，否则无法将障碍物彼此区分开线

缩放用于区分同伦类的内部参数 (H-signature) 。

`h_signature_threshold`, 最小值: 0.0、默认值: 0.1、最大值: 1.0。如果实部和虚部的差都低于规定的阈值, 则假定两个 h 签名相等。

`obstacle_keypoint_offset`: 0.1

`obstacle_heading_threshold`, 最小值: 0.0、默认值: 0.45、最大值: 1.0。指定障碍标头和目标标头之间的标量积的值, 以便将(障碍)考虑到勘探中

`visualize_hc_graph`, 默认值: False。可视化创建的图形, 用于探索不同的轨迹 (在 rviz 中检查标记消息)

`global_plan_viapoint_sep`, 设置为正值(考虑全局 points) 含义是在 global_planner 轨迹上间隔多远会有一个 viapoint 插入。设置为 1 就是 1m 一个 viapoint。

3.7 参数修改举例

在修改 `teb_local_planner_params` 参数是, 每次修改需要编译重启导航才能生效。每次修改参数, 可以将原参数进行注释, 不用删除, 以免发生忘记原始参数的情况。下面以修改导航速度为例进行说明。

修改文件红框内数值数值, 注释掉原数值, 添加新数值。

```
13
14 # Robot
15
16 max_vel_x: 0.3 #0.2
17 max_vel_y: 0.0
18 max_vel_x_backwards: 0.05
19 max_vel_theta: 0.5 #0.2 #0.7
20
21 acc_lim_x: 0.6
22 acc_lim_theta: 0.6
23
24 min_turning_radius: 0.15
25 wheelbase: 0.202
```

参数修改说明: 我们看到影响到导航速度主要为 `max_vel_x`、`max_vel_theta`。

保存文件并编译。

```
cd ~/mr300_ws && catkin_make -j4
```

然后再启动导航，查看结果。

4. 总结

本文的内容是针对 BR280 的参数调整的角度来编写的，其中又些参数 BR280 中并没有用到，如果了解全部的参数内容，请参阅官方说明链接：

http://wiki.ros.org/teb_local_planner#Parameters

更多关于 Teb_local_planner 的资料可以参考官方教程：

http://wiki.ros.org/teb_local_planner/Tutorials

本文总结一些可以提高TEB规划器性能的参数调整策略,供遇到类似问题的读者参考。

- 关闭多路径并行规划（效果非常显著）
- 使用 Costmap Converter （非常显著）
- 降低迭代次数（no_inner/outer_iterations）（显著）
- 降低 max_lookahead_distance （一般）
- 减小局部耗费地图的大小 （显著）
- 增大规划周期和控制周期 （影响效果）
- 使用单点 footprint，配合最小障碍物距离约束 （不太显著且影响效果）



— 深圳史河机器人科技有限公司 —



深圳史河机器人科技有限公司

www.robotplusplus.com.cn

深圳市龙华区龙华街道清华社区建设东路青年创业园C栋4层412-415房