动态调参，启动参数调节器：rosrun rqt\_reconfigure rqt\_reconfigure

参考文章：

Teb轨迹算法教程：https://www.ncnynl.com/archives/201809/2602.html

智能车竞赛——TEB轨迹规划算法的参数调试：<https://www.guyuehome.com/9811>

ROS - teb\_local\_planner 参数总结：<https://blog.csdn.net/Fourier_Legend/article/details/89398485>

Costmap\_2d文档：<http://wiki.ros.org/costmap_2d>

Ros导航调整指南：之前发的pdf文件（那里写的是dwa的，代价地图和amcl的调参可以参考下）

**costmap\_common\_params**

footprint:[[0.171,-0.128],[0.171,0.128],[-0.171,0.128],[-0.171,-0.128]]#样例包中的参数

obstacle\_layer:#障碍物层

inflation\_radius: 0.2

inflation\_layer:膨胀层

cost\_scaling\_factor: 15.0 # exponential rate at which the obstacle cost drops off (default: 10)

inflation\_radius: 0.2 # max. distance from an obstacle at which costs are incurred for planning paths.

#障碍物层和碰胀层都有 inflation\_radius（膨胀半径），这两个参数好像对结果没什么影响，我的理解是这个文件是全局代价地图和局部代价地图通用的参数，下面的全局代价地图有单独调这个碰胀半径，所以这个值影响不大

**global\_costmap\_params**

update\_frequency: 1.0 #更新和发布的频率，应该对性能会有影响，不过我没改过

publish\_frequency: 2.0

inflation\_radius: 0.101 #碰胀半径，实测0.1和0.101是比较好的值，0.101当定位飘了后容易导致无法规划出路径

**local\_costmap\_params**

#没改过

**teb\_local\_planner\_params**

**# Trajectory 轨迹相关参数**

dt\_ref: 0.3 # default 0.3 | 轨迹分辨率（单位秒）。小值导致精细分辨率，从而更好地逼近动力学模型，太高的值（> 0.6s）可能导致不再可行的轨迹

dt\_hysteresis: 0.03 #允许改变的时域解析度的浮动范围， 一般为 dt\_ref 的 10% 左右

max\_global\_plan\_lookahead\_dist: 1.0 #指定考虑优化的全局计划子集的最大长度 #向前规划的距离，尝试过调小，应该不是主要问题

**# Robot 机器人相关参数**

max\_vel\_x: 1.135 #4.0

max\_vel\_x\_backwards: 0.5

max\_vel\_y: 0.2

max\_vel\_theta: 2.185 #2.5 #4.0 # the angular velocity is also bounded by min\_turning\_radius in case of a carlike robot (r = v / omega)

acc\_lim\_y: 5.0

acc\_lim\_x: 9.25 #120 #4.5

acc\_lim\_theta: 7.15 #65 #3.5

footprint\_model: # types: "point", "circular", "two\_circles", "line", "polygon" #机器人足迹模型

type: "circular" #已换为, "polygon"

radius: 0.15 # for type "circular"

line\_start: [-0.1, 0.0] # for type "line" 线模型起始坐标

line\_end: [0.1, 0.0] # for type "line" 线模型尾部坐标

front\_offset: 0.2 # for type "two\_circles" 前圆心坐标

front\_radius: 0.3 # for type "two\_circles" 前圆半径

rear\_offset: 0.2 # for type "two\_circles" 后圆心坐标

rear\_radius: 0.2 # for type "two\_circles" 后圆半径

vertices: [ [-0.2,-0.2], [0.2,-0.2], [0.2,0.2], [-0.2,0.2] ] # for type "polygon" 多边形边界#可以尝试更改为长方形数据[[0.171,-0.128],[0.171,0.128],[-0.171,0.128],[-0.171,-0.128]]

**# GoalTolerance #目标容忍度相关，关系应该不大**

**# Obstacles #障碍物相关参数**

min\_obstacle\_dist: 0.02 # 与障碍物的最小期望距离 This value must also include our robot radius, since footprint\_model is set to "point". #允许到障碍物最小距离，应该与后面权重相关参数的weight\_obstacle相关

inflation\_dist: 0.15 # 障碍物周围的缓冲区，进缓冲区会导致规划器减速,需要比min\_obstacle\_dist大 #未找到合适的参数，过大导致过不去，过小导致撞墙

**# Optimization #权重相关参数** #不同博客，我们线上赛和北邮的在这些参数上差别较大，感觉比较关键，但没找到最好的

penalty\_epsilon: 0.1 #为硬约束近似的惩罚函数添加一个小的安全范围#为速度的约束提供一个缓冲的效果，就是在到达速度限制前会产生一定的惩罚让其提前减速达到缓冲的效果。

weight\_max\_vel\_x: 10.0 #满足最大允许平移速度的优化权重#后面还有最大角速度权重，以及加速度，角加速度权重，在整个运动过程中以主要以高速还是低速运行，则是看这些权重的分配

weight\_max\_vel\_theta: 76.0 #满足最大允许平移速度的优化权重

weight\_acc\_lim\_x: 10 #满足最大允许平移加速度的优化权重

weight\_acc\_lim\_theta: 95 #满足最大允许角加速度的优化权重

weight\_max\_vel\_y: 10.0 #横移关键参数

weight\_kinematics\_forward\_drive: 1000 #最大为1000，强制机器人仅选择正向#减小后小车更趋向正向移动，仿真中侧向和后退定位容易飘

weight\_optimaltime: 10 # must be > 0 #根据转换/执行时间对轨迹进行收缩的优化权重#调大能减少局部路径与最佳路径偏离过大问题，但局部路径还是很奇怪

weight\_obstacle: 100 #保持与障碍物的最小距离的优化权重 default: 50.0

weight\_inflation: 0.2 #膨胀半径权重

weight\_viapoint: 1000 #跟踪全据路径的权重

**# Homotopy Class Planner**

enable\_homotopy\_class\_planning: False #激活并行规划，因为一次优化多个轨迹，占用更多的CPU资源#已开启，效果不明显

**# Recovery #恢复行为，没改过**