華中科技大學

· 路径规划 算法文档 ·

基于ROS的智能体强化学习导航与避障研究

专业 ： 人工智能

班级 ： 人工智能本硕博 2001 班

姓名 ： 罗亚文

学号 ： U202015237

指导老师: 何顶新

报告日期: 2023-06-16

人工智能与自动化学院

目录

目录

1 题目要求及环境搭建

1.1 题目要求：

1.在ROS框架下实现机器人导航避障仿真。

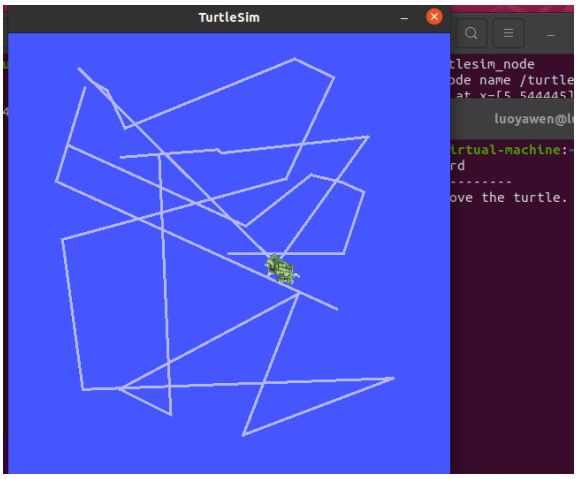
2.使用导航功能包实现，可使用提供的teb的导航功能包(参数需自己调整)，如何使用见README文件，也可自选其他功能包。

3.调整teb算法规划或其他规划算法参数，录制类似演示视频中仿真环境+rviz视角的运行视频。

4.整理完成路径规划算法文档，包括算法原理、参数含义、参数如何调整、如何在仿真环境中部署等内容。

1.2 环境搭建：

- 本次实验基于虚拟机VMware配置了Ubuntu20.04环境。
- 在Ubuntu20.04系统中安装了对应的ROS Noetic Ninjemys。
  - 运行实验验证ROS系统正确安装：



- 安装teb_local_planner依赖库，build并run程序。
  - 具体的依赖库为：
    - navigation stack：`sudo apt install ros-$ROS_DISTRO-navigation`
    - teb_local_planner：`sudo apt install ros-$ROS_DISTRO-teb-local-planner`
  - 在build以及run过程遇见的bug及解决方案如下：

① 运行gazebo时出现“[gazebo-2] process has died [pid 7920, exit code 255.....“

```
[gazebo-2] process has died [pid 7920, exit code 255, cmd /opt/ros/melodic/lib/g
```

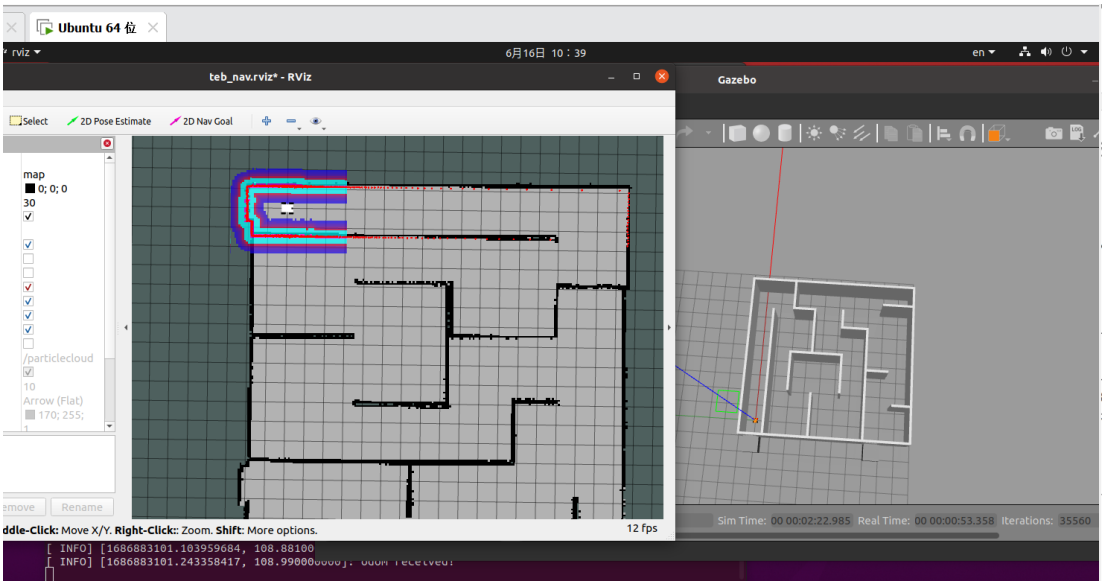
解决方案：killall gzserver

② ros中启动rviz显示段错误：[rviz-3] process has died [pid 17193, exit code -6

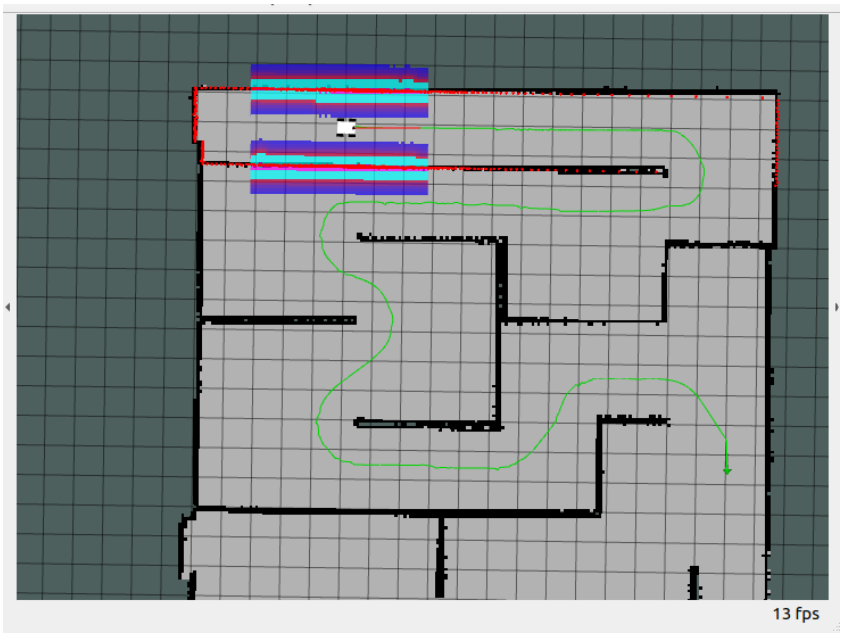
解决方案：

运行以下五条命令
1 sudo add-apt-repository ppa: ubuntu-x-swat/updates
2 sudo apt-get update
3 sudo apt-get dist-upgrade
4 glxinfo | grep “OpenGL version”
5 sudo apt install ppa-purge && sudo ppa-purge ppa:ubuntu-x-swat/updates
如果依然显示错误，运行
glxinfo | grep OpenGL查看自己OpenGL版本，我的版本显示
OpenGL version string: 1.4 Mesa 20.0.8
所以是OpenGL版本是mesa导致rviz启动不了。这时需要关闭硬件加速器。运行
export LIBGL_ALWAYS_SOFTWARE=1
然后再次启动rviz就可以了

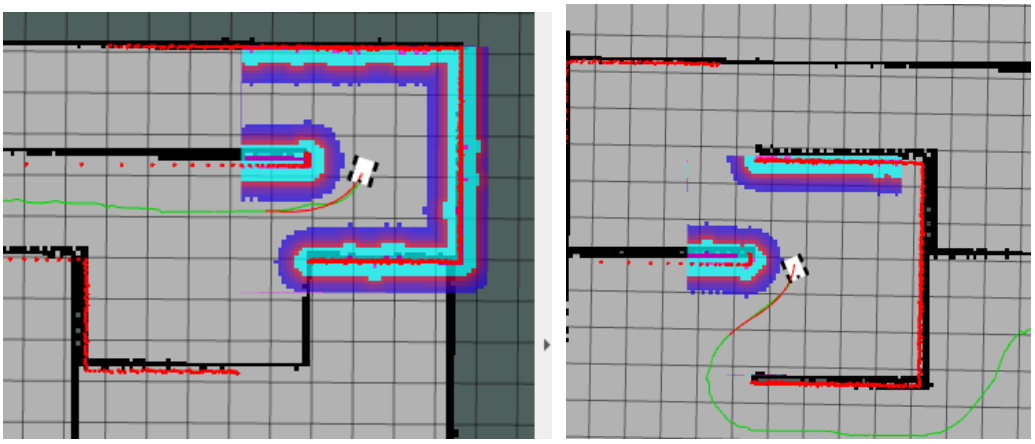
- 安装并编译完成后，运行效果如下图：



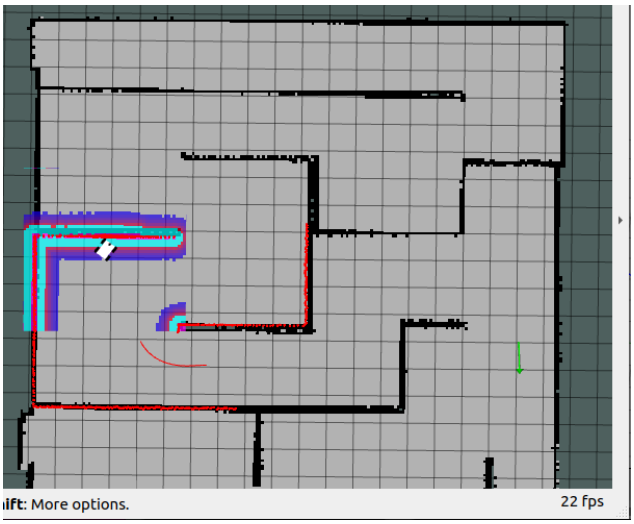
- 在使用默认参数的情况下，先指定目标位置看看效果：
  - 在指定目标位置后，其规划路径如下（绿色为规划出来的路径）：



- 原始参数有一些缺点，由下图可以看出，在一些拐弯的地方会非常慢，且控制不够精确，会和目标路径产生较大的位置偏差。



- 该参数还存在控制效率不高，且容易使得小车失控的情况，如下图所示：



2 teb_local_planner 算法文档

2.1 算法简介

Time elastic band算法（teb），考虑了运动在时间方面的动态约束，如有限的机器人速度和加速度，在加权多目标优化框架中进行求解。在优化过程中，大多数目标都是局部的，因为它们只依赖于几个邻近的中间状态，这使得系统矩阵非常的稀疏，从而在大规模约束下基于最小二乘法进行优化成为可能。该方法鲁棒性强，计算效率高，能够实时生成最优机器人轨迹，将一系列由路径点组成的初始路径转化为一条显式依赖于时间的轨迹，从而实现对机器人的实时控制。

Elastic band[2]的主要思想是修正初始路径，将其视为一个受内外力平衡的弹性橡皮筋，在保持与障碍物的距离的同时，尽可能的绷紧路径。然而，动力学约束尚未被认定是路径修正的一个目标。传统的方法是对路径进行平滑处理，例如使用样条曲线来获得动力学可行的轨迹。Timed elastic band[1]的方法，在elastic band的基础上增加了时间信息，从而允许考虑机器人的动力学约束和直接修正轨迹而不是路径。通过考虑时间信息Timed elastic band方法还可以用来控制机器人的速度和加速度。

2.2 算法原理

传统的elastic band[2]使用n个机器人的中间位姿$x_i = (x_i, y_i, \beta_i)^T \epsilon R^2 \times S^1$的序列来描述，表示为包括了机器人在相关帧（{地图}，图2）中的位置x_i, y_i和方位角β_i的表达：

$$Q = \{\mathbf{x}_i\}_{i=0\dots n} \quad n \in \mathbb{N} \qquad (1)$$

“timed elastic band”在两个连续configuration之间增加了时间间隔，产生n-1个时差ΔT_i：

$$\tau = \{\Delta T_i\}_{i=0\dots n-1} \qquad (2)$$

时差表示机器人从一个configuration依次过渡到下一个configuration所需的时间。TEB被定义为两个序列的元组：

$$B := (Q, \tau) \qquad (3)$$

关键思想是通过实时优化一个加权多目标函数，在configuration和时间间隔方面调整和优化TEB：

$$f(B) = \sum_k \gamma_k f_k(B) \quad (4)$$

$$B^* = \underset{B}{\operatorname{argmin}} f(B) \quad (5)$$

TEB的目标函数分为两类：用惩罚函数表示的速度和加速度限制等约束和与轨迹有关的目标，如路径最短、路径最快或与障碍物的距离，可自由使用的稀疏约束优化算法的实现在机器人框架（ROS）中不能方便的获取到，因此，在“timed elastic band”中，采用一个分段连续的、可微的代价函数来计算违反约束的惩罚：

$$e_\Gamma(x, x_r, \epsilon, S, n) \simeq \begin{cases} (\frac{x-(x_r-\epsilon)}{S})^n & \text{if } x > x_r - \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

在算法中所使用的具体约束如下：

- **避免撞到静态或动态障碍物：**

  目标函数取决于与路标或障碍物z_j之间的最小间隔$d_{min,j}$。对于路标点，距离从上界以最大目标半径r_{pmax}约束,在障碍物的情况下，距离从下界以最小距离r_{pmin}约束:

$$f_{path} = e_\Gamma(d_{min,j}, r_{p_{max}}, \epsilon, S, n) \quad (7)$$

$$f_{ob} = e_\Gamma(-d_{min,j}, -r_{o_{min}}, \epsilon, S, n) \quad (8)$$

- **速度和加速度的动态约束：**通过类似于几何约束的惩罚函数来描述。平均平移速度和旋转速度根据下式计算：

$$v_i \simeq \frac{1}{\Delta T_i} \left\| \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix} \right\| \quad (9)$$

$$\omega_i \simeq \frac{\beta_{i+1} - \beta_i}{\Delta T_i} \quad (10)$$

欧式距离是两个连续位姿之间圆弧路径真实长度的充分近似。 加速度涉及到两个连续的平均速度，因此考虑具有两个相应时间间隔的三个连续configurations：

$$a_i = \frac{2(v_{i+1} - v_i)}{\Delta T_i + \Delta T_{i+1}} \quad (11)$$

考虑到移动机器人是两轮差分驱动的，可以根据以下公式计算车轮的速度与机器人中心的平移速度和旋转速度之间的关系：

$$v_{w_r,i} = v_i + L\omega_i \quad (12)$$

$$v_{w_l,i} = v_i - L\omega_i \quad (13)$$

- **速度和加速度的动态约束：**通过类似于几何约束的惩罚函数来描述：具有差分驱动的机器人仅有两个局部自由度，因此，它们只能在机器人当前方向上运动。这一运动学约束导致了机器人的平滑路径是由一系列分段圆弧组成的。其目标函数为：

$$f_k(\mathbf{x}_i, \mathbf{x}_{i+1}) = \left\| \left[\begin{pmatrix} \cos\beta_i \\ \sin\beta_i \\ 0 \end{pmatrix} + \begin{pmatrix} \cos\beta_{i+1} \\ \sin\beta_{i+1} \\ 0 \end{pmatrix} \right] \times \mathbf{d}_{i,i+1} \right\|^2$$

- **最快路径**：通过最小化所有时间间隔和的平方，可以轻松实现最快路径的目标：

$$f_k = (\sum_{i=1}^{n} \Delta T_i)^2$$

2.3 算法实现流程

Teb算法的流程图如下所示：

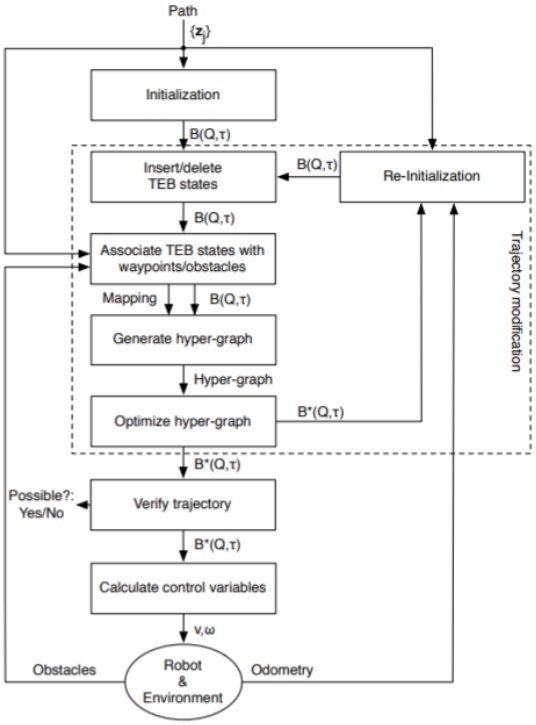


Figure 6: Control flow of TEB-implementation

在初始化阶段，通过添加关于动力学和运动学约束的默认时间信息，将初始路径变为初始轨迹。初始轨迹由分段的线性段组成，这些分段的线性段先进行纯旋转，再进行平移。在每次迭代中，算法动态地添加新的configuration或删除先前的configuration，以便将空间和时间分辨率调整到剩余的轨迹长度或规划的范围内。采用一些先验假设来避免振荡。优化问题被转换为hyper-graph问题，并使用稀疏系统大规模优化算法进行求解。

在确认能进行teb优化后，可以将计算出控制变量v和ω，直接对机器人的驱动系统进行控制。在每次新迭代之前，重新初始化阶段都会检查新的和更改中的路标，如果在解析short-range相机或激光扫描数据后接收到路标点，将非常有用。

2.4 算法参数含义

2.4.1 机器人配置参数

参数名称\默认值	含义
acc_lim_x\0.5	机器人的最大平移加速度
acc_lim_theta\0.5	机器人的最大角加速度
max_vel_x\0.4	机器人的最大平移速度
max_vel_x_backwards\0.2	机器人向后行驶时的最大绝对平移速度
max_vel_theta\0.3	机器人的最大角速度
max_vel_y\0.0	机器人的最大扫射速度（非完整机器人为0
min_turning_radius\0.0	类车机器人的最小转弯半径（对于差速驱动机器人，设置为零）
footprint_model/vertices \[[0.25, -0.05], […], …]）	此参数仅与“多边形”类型相关。它包含多边形顶点列表（每个顶点的二维坐标），不要重复末端的第一个顶点。

2.4.2 目标容差参数

参数名称\默认值	含义
xy_goal_tolerance\0.2	允许到目标位置的最终欧氏距离
yaw_goal_tolerance\0.2	允许的最终方向误差
free_goal_vel \False	移除目标速度约束，使机器人能够以最大速度到达目标

2.4.3 轨迹配置参数

参数名称\默认值	含义
dt_ref\0.3	轨迹的期望时间分辨率
dt_hysteresis\0.1	根据当前时间分辨率自动调整大小的滞后
global_plan_overwrite_ orientation\True	覆盖全局规划器提供的局部子目标的方向
allow_init_with_backwards _motion\False	如果为真，则当目标位于本地成本图中的起点之后时，可以使用向后运动初始化基本轨迹（仅配备后部传感器时才建议这样做）
max_global_plan_lookahead _dist\3.0	指定优化时考虑的全局计划子集的最大长度（累积欧氏距离）。实际长度由本地成本图大小和该最大界限的逻辑结合决定。可设置为零或负值以停用此限制。
feasibility_check_no_poses\4	在预测计划中指定每个采样间隔应检查可行性的姿势

2.4.4 障碍物参数

参数名称\默认值	含义
min_obstacle_dist\0.1	与障碍物的最小期望距离
inflation_dist\0.0（非动力学） \0.6(动力学)	非零惩罚成本障碍物周围的缓冲区
include_dynamic_obstacles\false	如果此参数设置为true，则非零速度障碍物的运动在优化过程中通过恒速模型进行预测和考虑
include_costmap_obstacles\True	指定是否应考虑本地成本图的障碍。标记为障碍物的每个单元格都被视为点障碍。因此，不要选择非常小的成本图分辨率，会增加计算时间。
costmap_obstacles_behind_ robot_dist\1.0	限制机器人后方规划时考虑的占用的本地成本图障碍物
obstacle_poses_affected\30	每个障碍物位置都附加到轨迹上最近的姿势，以保持距离。还可以考虑其他邻居。

2.4.5 优化参数

参数名称\默认值	含义
no_inner_iterations\5	每次调用的实际解算器迭代次数外环迭代
no_outer_iterations \4	每次外环迭代都会根据所需的时间分辨率自动调整轨迹大小参考并调用内部优化器（执行无内部迭代）
penalty_epsilon\0.1	为硬约束近似的惩罚函数添加一个小的安全裕度
weight_max_vel_x\2.0	满足最大允许平移速度的优化权重
weight_max_vel_theta\1.0	满足最大允许角速度的优化权重
weight_acc_lim_x \1.0	满足最大允许平移加速度的优化权重
weight_acc_lim_theta\1.0	满足最大允许角加速度的优化权重
weight_kinematics_nh\1000	满足非完整运动学的优化权重
weight_kinematics_forward _drive\1.0	用于强制机器人仅选择前进方向（正横向速度）的优化权重
weight_kinematics_turning_ radius\1.0	优化重量以实现最小转弯半径
weight_optimaltime\1.0	收缩轨道w.r.t过渡/执行时间的优化权重
weight_obstacle\1.0	与障碍物保持最小距离的优化权重
weight_inflation\0.1	通货膨胀惩罚的优化权重

2.4.5 不同拓扑中的并行规划参数

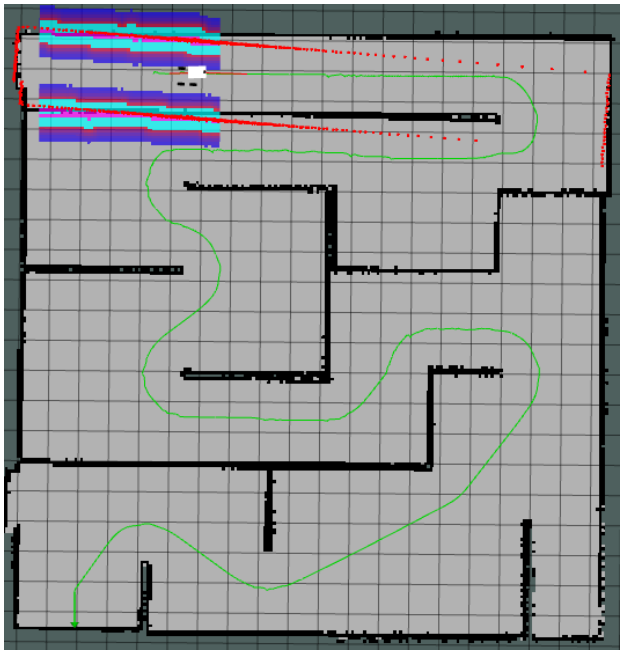
参数名称\默认值	含义
enable_homotopy_class _planning\True	在不同的拓扑中激活并行规划（需要更多的CPU资源，因为一次可以优化多个轨迹）
enable_multithreading\True	激活多线程以在不同线程中规划每个轨迹
max_number_classes\4	指定考虑的不同轨迹的最大数量（限制计算工作量）
selection_cost_hysteresis\1.0	采用新轨迹的代价比目前的路径代价需要低多少才得以采用
selection_obst_cost_scale\100.0	额外扩展障碍成本条款，仅用于选择“最佳”候选
selection_alternative_ time_cost\false	如果为true，时间成本（时间差平方和）将被总过渡时间（时间差之和）取代。
roadmap_graph_no_samples\15	指定为创建路线图而生成的样本数
roadmap_graph_area_width\6	随机关键点/航路点在起点和目标之间的矩形区域内采样,指定该区域的宽度
visualize_hc_graph\false	可视化为探索独特轨迹而创建的图形（在rviz中检查标记消息）

2.5 算法参数调整

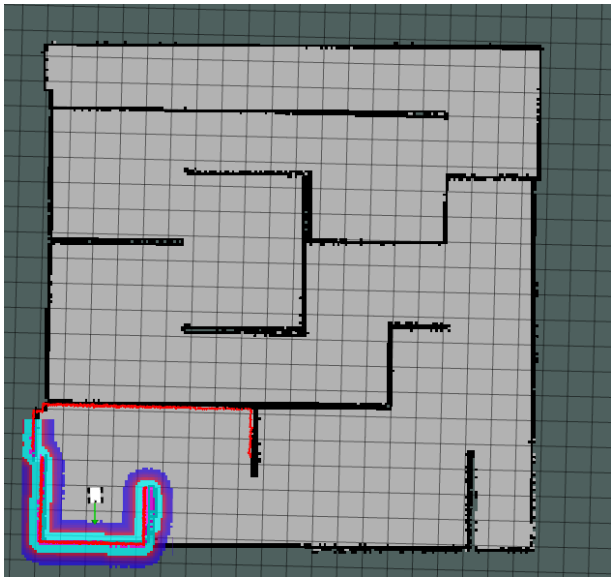
针对之前速度过慢，控制不稳的问题，主要调整的参数如下：

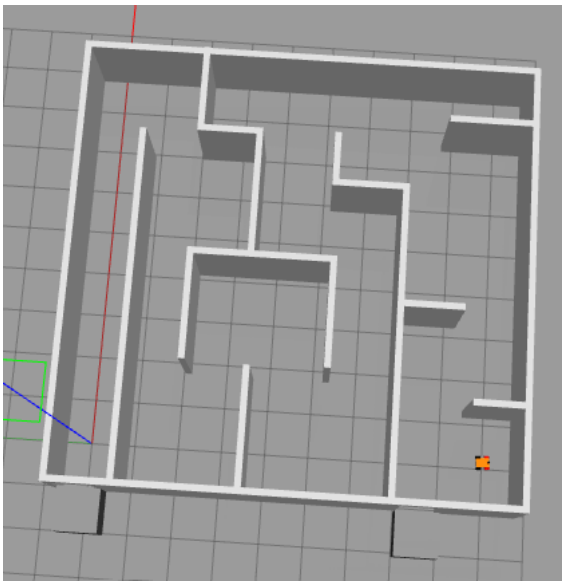
调整参数	调整前	调整后	调整目的
max_vel_x	0.4	5	**提高速度、角速度、加速度、角加速度上限，减少控制时间**
max_vel_theta	0.3	5	
acc_lim_x	0.5	3	
acc_lim_theta	0.5	3	
xy_goal_tolerance	0.2	0.1	**提高控制精度**
weight_max_vel_x	3	2	**降低最大速度的约束程度**
weight_dynamic_obstacle	50	20	**降低对障碍物的距离约束**

在调整参数之后，规划的路径更加圆滑，在拐弯处不会卡死，也避免了小车失控的问题：



小车可以较快到达终点：





调整参数之后的效果可以在录制的视频中观看。

2.6 算法仿真部署过程

这里再次总结一下算法仿真的部署过程：

teb_local_planner仿真部署步骤

安装依赖库

Dependencies:

- *ros-noetic*: `sudo apt install ros-noetic-desktop-full 20.04装这个 以下两个都要装`
- *navigation stack*: `sudo apt install ros-$ROS_DISTRO-navigation`
- *teb_local_planner*: `sudo apt install ros-$ROS_DISTRO-teb-local-planner`

Build过程

把course.tar.gz在主目录下,打开终端:

```
unzip course.zip
cd course/src
catkin_init_workspace
cd ..
catkin_make
gedit ~/.bashrc
source ~/course/devel/setup.bash
```

Run过程

```
roslaunch course run.launch
```

参考文献

[1] Rösmann C, Feiten W, Wösch T, et al. Trajectory modification considering dynamic constraints of autonomous robots[C]//ROBOTIK 2012; 7th German Conference on Robotics. VDE, 2012: 1-6.

[2] Quinlan, S.; Khatib, O. Elastic Bands: Connecting Path Planning and Control, IEEE Int. Conf. on Robotics and Automation (ICRA) (2), 1993

参考博客

[1] http://wiki.ros.org/rviz/Troubleshooting#Turning_off_hardware_acceleration

[2] https://www.jianshu.com/p/49684ed3978b

[3] https://blog.csdn.net/m0_60657960/article/details/129407002

[4] https://blog.csdn.net/Mr_l_i_u/article/details/115690782

[5] https://blog.csdn.net/mt528429/article/details/112757004

[6]http://wiki.ros.org/action/fullsearch/teb_local_planner?action=fullsearch&context=1

80&value=linkto%3A%22teb_local_planner%22

附录

调整后的参数文件如下：

```
TebLocalPlannerROS:
 odom_topic: odom
 map_frame: /map

 # Trajectory
 dt_ref: 0.3
 dt_hysteresis: 0.1
 global_plan_overwrite_orientation: True
 allow_init_with_backwards_motion: False
 max_global_plan_lookahead_dist: 3.0
 feasibility_check_no_poses: 5

 # Robot
 max_vel_x: 5 #velocity strain
 max_vel_x_backwards: 0.2
 max_vel_y: 0.0
 max_vel_theta: 5
 acc_lim_x: 3
 acc_lim_theta: 3
 min_turning_radius: 0.0 # diff-drive robot (can turn on place!)

 footprint_model:
  type: "polygon"
  vertices:
    [[0.165, 0.165], [0.165, -0.165], [-0.165, -0.165], [-0.165, 0.165]]

 # GoalTolerance
 xy_goal_tolerance: 0.1
 yaw_goal_tolerance: 0.1
 free_goal_vel: False

 # Obstacles
 min_obstacle_dist: 0.1 #0.5 # This value must also include our robot radius, since footprint_model is set to
"point".
 inflation_dist: 0.8
 dynamic_obstacle_inflation_dist: 0.6
```

include_dynamic_obstacles: True
include_costmap_obstacles: True
costmap_obstacles_behind_robot_dist: 1.5
obstacle_poses_affected: 30
costmap_converter parameters are defined in costmap_converter_params.yaml

Optimization
no_inner_iterations: 5
no_outer_iterations: 4
optimization_activate: True
optimization_verbose: False
penalty_epsilon: 0.1
weight_max_vel_x: 2
weight_max_vel_theta: 1
weight_acc_lim_x: 2
weight_acc_lim_theta: 2
weight_kinematics_nh: 1000
weight_kinematics_forward_drive: 1
weight_kinematics_turning_radius: 1
weight_optimaltime: 1
weight_obstacle: 20
weight_inflation: 0.3
weight_dynamic_obstacle: 50
weight_dynamic_obstacle_inflation: 0.3
weight_adapt_factor: 2

Homotopy Class Planner
enable_homotopy_class_planning: True
enable_multithreading: True
simple_exploration: False
max_number_classes: 4
selection_cost_hysteresis: 1.0
selection_obst_cost_scale: 1.0
selection_alternative_time_cost: True

roadmap_graph_no_samples: 15
roadmap_graph_area_width: 5
h_signature_prescaler: 0.5
h_signature_threshold: 0.1
obstacle_keypoint_offset: 0.1

obstacle_heading_threshold: 0.45

visualize_hc_graph: False

visualize_with_time_as_z_axis_scale: 0.2