前言

问题

- 在 open space 空间下规划出一条自车到停车位的无碰撞轨迹 满足平滑约束 可跟踪
- 考虑动态障碍物约束
- 在路径不可用的情况下 具备重规划能力
- 重规划时能够做到无缝切换 即从原路径无缝切换到重规划路径 无明显体感
- 规划频率 10HZ

预实现目标

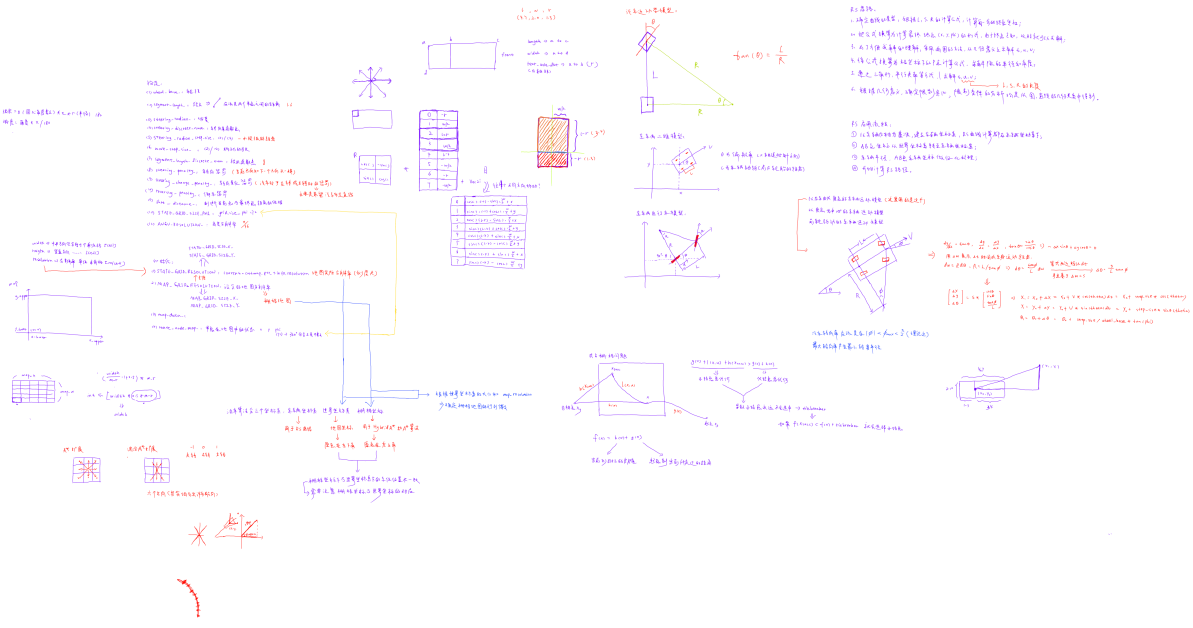
- 在 100ms 内规划出一条轨迹(包括速度)
- 路径满足平滑约束 无 kappa 过大的问题
- 规划成功率 > 80%
- 可在 ROS 中可视化看到仿真结果 包括控制

预计时间

- 2 个月

相关思路

图解(.pdf)



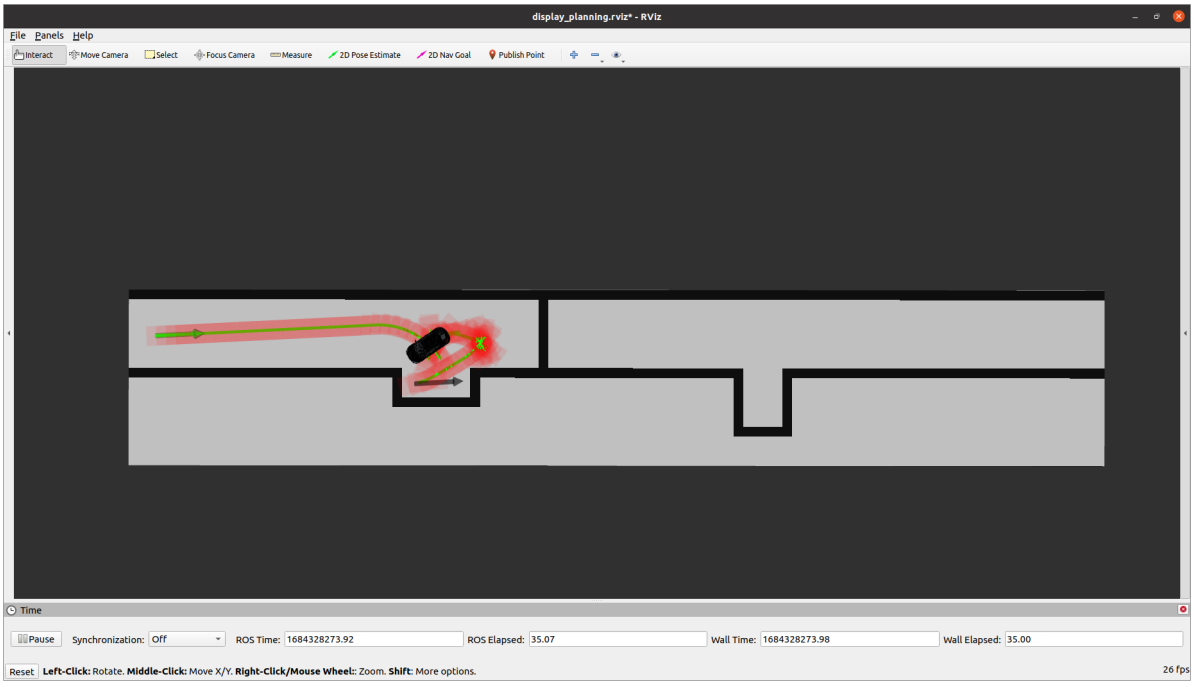
效果

垂直车位停车

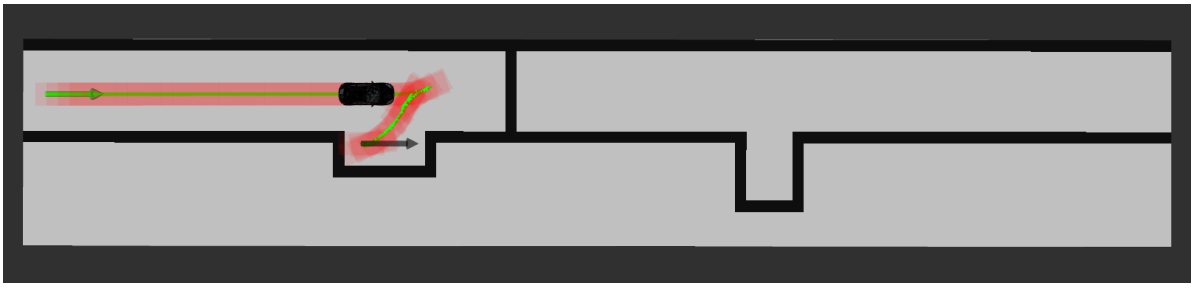
垂直车位停车-1

垂直停车-2

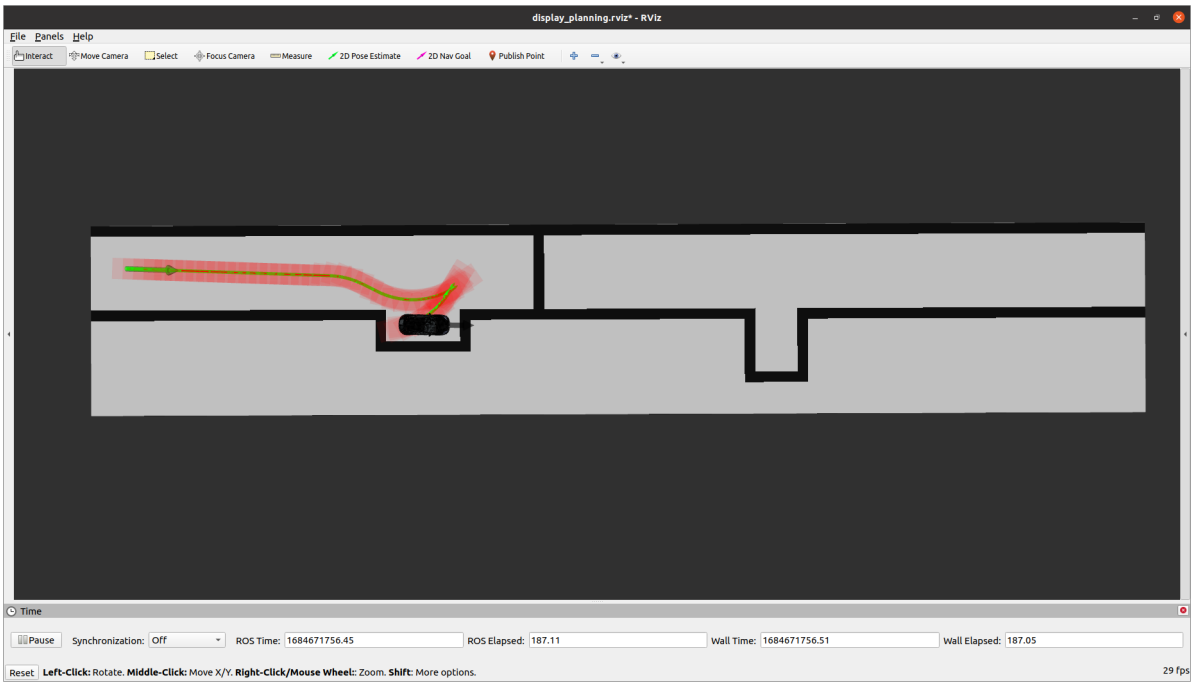
水平车位停车



水平车位停车-1



水平车位停车-2



时间

垂直车位停车-1花费

```
[ INFO] [1685258178.190189913]:  - ->  Time in Hybrid A star is 61.626475 ms, path length: 33.652524
```

垂直车位停车-2花费

```
[ INFO] [1685258235.927588045]:  - ->  Time in Hybrid A star is 28.100090 ms, path length: 19.133807
```

前期准备

环境配置和前期知识储备

- 预计 4 月 15 号完成

2023-04-08

- 配置 windows 下的 ros
  参考文章
  参考视频
- 安装过程并不顺利
- 安装 ros2
- 准备了解一下 ros 的相关概念和语法？
- ros 进行自动驾驶开发过程？

2023-04-09

- 网站老是崩 进行离线安装
- 下载又慢又容易崩
- 基于ros的路径规划
- ros入门
- ros入门文档
- 为了以后方便还是准备装 linux

- 之前 virtualbox 在 win11 出来的时候有 bug 就删掉了 现在应该已经修复了

2023-04-10

- 完成 linux 下的 ros 安装
- ros 集成开发环境搭建

2023-04-11 -> 04-16

- 了解 ros 的开发过程
- 了解 ros 的通信机制
- ros 的话题通信、服务通信和参数服务器
- ros 自定义头文件和源文件调用
- ros 的常用组件 --- TF 坐标变换 --- rosbag --- rqt 工具箱

机器人仿真

- 预计 4 月 20 号完成

2023-04-16 -> 04-18

- URDF 语法
- xacro
- gazebo

机器人导航

2023-04-18 -> 2023-04-20

- 这个应该是最重要的知识部分
- 导航模块
- 导航实现
- 接下来准备开始 coding

日志

代码参考

- 理解算法实现 -> 混合A*算法 的matlab复现
- 博客 -> 自动泊车实例

日期

2023-04-20

- 创建项目仓库
- 完成项目环境配置

2023-04-22

- 创建车辆模型
- 完成初始车辆姿态订阅
- 完成目标车辆姿态订阅

2023-04-23

- 实现对传入的车辆姿态信息解析

> 初始姿态使用PoseWithCovarianceStamped 而目标姿态使用PoseStamped 二者都是带有参考系和时间戳的姿态
> 主要的区别在于PoseWithCovarianceStamped消息类型包含一个协方差矩阵 用于表示姿态的不确定性

- 安装了 Eigen 但还是报错 No such file or directory -> 因为 eigen 库默认安装在 /usr/include/eigen3/Eigen 需映射到 /usr/include 路径下

```
sudo ln -s /usr/include/eigen3/Eigen /usr/include/Eigen
```

- 对传入的车辆信息进行初始化 并解析位姿信息
- 测试话题正常能否正常发布

2023-04-24

- 了解 `A*` 算法和 `混合A*` 算法
- 重温 `Dijkstra`算法
- 百度的 apollo 项目中的`混合A*` 算法的代码中 直接将拓展的距离设置成了栅格的对角线 也就是说子节点和父节点不会占据一个栅格 -> 是不是因为这样才不适合狭窄的空间下情况而不是`混合A*` 不适合？

2023-04-26

- 实现地图的订阅
- 继续了解 `混合A*` 算法

2023-05-07

- 创建规划策略(planning_method)相关文件(.h .cpp)
- 设置参数 -> 包括转向角、转向惩罚、倒车惩罚等
- 修改地图的消息类型

2023-05-08

- 设置车辆形状
- 抽象化状态节点
- 创建时间类 方便计算时间差

2023-05-09

- 开始编写`混合A*算法部分
- `混合A*` 初始化

2023-05-11

- 继续`混合A*`算法部分
- debug
- RS(Reed_Shepp)?

2023-05-12

- 测试程序
- debug
- 生成的路径发布失败(显示失败)

2023-05-13

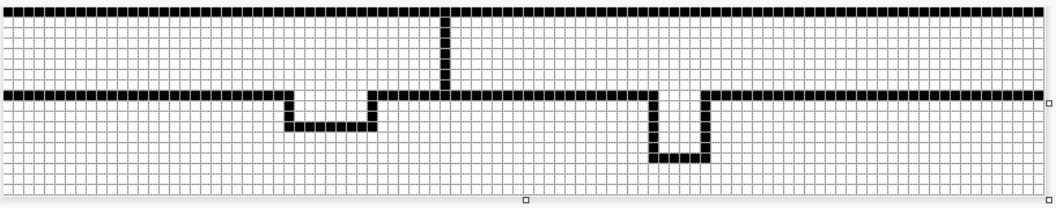
- 设置地图参数
- 实现地图在 `rviz` 中显示
- 节点搜索部分
- 启发函数

2023-05-15

- 继续 `search()` 部分
- 相邻节点搜索

2023-05-16

- 修改路径发布 `topic`
- 路径正常发布
- 绘制地图 设计垂直车位和水平车位



2023-05-17

- 路径障碍物碰撞检测
- 边界检测 `Bresenham` 算法

2023-05-21

- 修改一些参数
- 测试效果
- 规划时间和车辆的起始位置还是有关系的 如果生成的路径存在一些不必要的倒车和转弯的话 时间上的花费是很高的 可能是 3-4 倍
