



# 硕士 学位 论文

## MASTER'S DISSERTATION

### ( 学术 学位 )

论文题名 生物同步康复机制理论与应用

作者姓名 张小明

学科专业 电气工程

指导教师 郭大师 教授

2024 年 6 月



中图分类号：TM464

UDC：621.3

学校代码：10216

密级：公开

## 学术学位硕士学位论文

# 生物同步康复机制理论与应用

硕 士 研 究 生：张小明

导 师：郭大师 教授

申 请 学 位：工学硕士

学 科 专 业：电气工程

所 属 学 院：电气工程学院

答 辩 日 期：2024 年 6 月

授 予 学 位 单 位：燕山大学



# **Bio-Syncretic Rehabilitation Mechanism Theory and Application**

A dissertation submitted to  
Yanshan University  
in partial fulfillment of the requirement for the degree of  
Master of Engineering  
in Electrical Engineering  
By  
Zhang Xiaoming  
Supervisor: Professor Guo Dashi

**School of Electrical Engineering, Yanshan University**

**June, 2024**



## 燕山大学硕士学位论文原创性声明

本人郑重声明：此处所提交的 硕士学位论文《生物同步康复机制理论与应用》，是本人在导师指导下，在燕山大学攻读 硕士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作作出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字： 日期： 年 月 日

## 燕山大学硕士学位论文使用授权书

《生物同步康复机制理论与应用》系本人在燕山大学攻读 硕士学位期间在导师指导下完成的 硕士学位论文。本论文的研究成果归燕山大学所有，本论文的研究内容不得以其他单位的名义发表。本人完全了解燕山大学关于保存、使用学位论文的规定，同意学校保留并向有关部门递交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权燕山大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

保密 ，在 年解密后适用本授权书。

本学位论文属于

不保密 。

(请在以上相应方框内打“√”)

作者签名： 日期： 年 月 日

导师签名： 日期： 年 月 日



摘要

---

## 生物同步康复机制理论与应用

摘要：

关键词：关键词 1；关键词 2；……；关键词 8

分类号：TM464

# Bio-Syncretic Rehabilitation Mechanism Theory and Application

**Abstract:** The rehabilitation mechanism is the foundation for a rehabilitation robot to realize its motion, and the quality of the rehabilitation mechanism decides the rehabilitation effect of patients with the rehabilitation robot. The research of rehabilitation mechanism is the key content in rehabilitation engineering. The bio-syncretic rehabilitation mechanism (BSRM) is a new type of modern mechanism composed of human body and machine, and it can meet the design requirements of rehabilitation robot more effectively because of the interaction and harmonization between human body and machine during the rehabilitation exercise. This paper presents the relative systematic research on BSRM and its application. The main contents of this paper are as follows:

The concept of rehabilitation mechanism is proposed based on the relationship between human body mechanism and mechanical mechanism in the rehabilitation kinematics system, and the characteristics, degree of freedom, classification and symbol representation of this new kind of mechanism is introduced. The approaches to type synthesis of parallel BSRM and series BSRM are developed based on screw theory and the characteristic of BSRM, and some parallel BSRMs for ankle joint rehabilitation and series BSRMs for knee joint rehabilitation are got.

**Keywords:** keyword1; keyword2; .....; keyword8

**Classification:** 621.3

# 目 次

<b>摘要</b> .....	I
<b>Abstract</b> .....	II
<b>第1章 绪论</b> .....	1
1.1 研究背景与意义 .....	1
1.2 运动规划技术国内外研究现状 .....	2
1.2.1 基于图搜索的运动规划方法 .....	2
1.2.2 基于采样的运动规划方法 .....	3
1.2.3 基于曲线插值的运动规划方法.....	3
1.2.4 基于优化的运动规划方法 .....	4
1.3 铰接车运动规划技术国内外研究现状 .....	5
1.4 本文研究的主要内容 .....	6
1.5 本文结构安排 .....	7
<b>第2章 铰接车建模与运动规划系统设计</b> .....	9
2.1 运动规划系统总体方案设计 .....	9
2.2 铰接车运动学模型建立 .....	9
2.3 基于最优控制的运动规划问题建模 .....	11
2.4 最优控制问题的解法 .....	12
2.4.1 极小值原理.....	12
2.4.2 基于打靶法的数值求解.....	13
2.4.3 基于配点法的数值求解.....	14
2.5 非线性规划问题的求解 .....	14
2.5.1 罚函数法 .....	15
2.5.2 非线性规划问题的初始解 .....	15
2.6 本章总结 .....	16
<b>第3章 无人铰接车微分平坦全局轨迹规划方法</b> .....	17
3.1 铰接模型微分平坦性 .....	18
3.1.1 铰接车模型的平坦输出 .....	18
3.2 轨迹的参数化表达形式 .....	19
3.2.1 MINCO 样条曲线基本原理.....	19
3.2.2 起点和终点约束方程 .....	21
3.2.3 航点的连续性约束方程.....	22
3.2.4 轨迹的闭式解 .....	23

---

3.3 基于安全走廊的安全约束分析 .....	23
3.3.1 凸多边形约束的数学表达 .....	24
3.4 单阶段无约束轨迹规划 .....	26
3.4.1 铰接车的最大曲率 .....	26
3.4.2 优化问题的建立 .....	26
3.4.3 起点和终点航向角保证 .....	28
3.4.4 优化问题求解 .....	29
3.5 多阶段全局轨迹规划 .....	30
3.5.1 多阶段规划的问题描述 .....	31
3.5.2 站点的换向处理 .....	31
3.6 本章总结 .....	32
<b>第4章 基于安全走廊的无人铰接车局部规划方法 .....</b>	<b>33</b>
4.1 铰接车的一般性结构建模 .....	33
4.2 铰接车 Hybrid A* 算法的创新改进 .....	34
4.2.1 节点拓展策略改进 .....	35
4.2.2 代价函数设计 .....	37
4.2.3 碰撞检测算法设计 .....	38
4.3 基于 NMPC 和安全走廊的轨迹规划 .....	39
4.3.1 NMPC 问题构建 .....	39
4.3.2 运动学约束 .....	40
4.3.3 安全走廊的防碰撞约束处理 .....	42
4.3.4 变量的边界约束 .....	44
4.3.5 时间正则化约束 .....	44
4.3.6 迭代求解框架 .....	45
4.4 本章小结 .....	45
<b>第5章 无人铰接车运动规划系统仿真实验 .....</b>	<b>47</b>
5.1 仿真环境介绍 .....	48
5.1.1 ROS 简介 .....	48
5.1.2 运动规划软件框架设计 .....	49
5.2 全局规划算法测验 .....	50
5.2.1 点到点路径规划 .....	51
5.2.2 多点路径规划测试 .....	52
5.3 局部规划算法测验 .....	54
5.3.1 自主泊车场景 .....	55
5.3.2 侧方位停车场景 .....	58

## 目 次

---

---

5.4 穿越复杂地形实验.....	59
5.5 本章总结 .....	59
参考文献 .....	61
附录 A .....	62
附录 B .....	63
攻读硕士学位期间取得的成果.....	64
致谢.....	65



## 第1章 绪论

### 1.1 研究背景与意义

随着科技的不断进步与社会对智能化技术需求的增加，自动驾驶技术在多个领域得到了广泛应用。从无人驾驶汽车到无人驾驶清扫车，智能化交通工具逐步融入日常生活中，改变了传统的交通方式，提高了工作效率，保障了安全性。无人驾驶技术的快速发展为城市环卫、仓储物流等多个领域提供了新的解决方案，尤其是在复杂环境中的智能作业能力，使得人力资源的需求大幅减少，降低了事故发生的概率，推动了社会的可持续发展。

铰接车作为一种具有独特结构的车辆，其在车辆动力学和控制领域具有较高的研究价值。不同于常规车辆，铰接车通过中间铰接部分实现转向，具有较强的灵活性和转弯能力，但也因此带来了更复杂的运动学约束和控制问题。铰接车的运动规划涉及到如何高效、安全地控制其在复杂环境中的行驶轨迹，特别是在低速、精确行驶的应用场景中，如何确保车辆能够稳定执行任务，同时避免与环境中的障碍物发生碰撞，是研究的核心问题。近年来，铰接车的应用逐渐扩展至仓储搬运、城市环卫等多个领域。在环卫作业中，铰接车具有较高的机动性和灵活性，能够应对复杂的城市道路环境，尤其是在狭窄的街道和交通繁忙的区域，铰接车凭借其独特的转向方式能够完成更为精细的作业任务。因此，研究铰接车的运动规划技术，尤其是在复杂环境下的路径规划与速度控制，具有重要的现实意义。

铰接车运动规划包括两个关键部分：全局规划和局部规划。全局规划旨在为铰接车生成一条长远的路径，它主要为铰接车提供宏观的导航基准。局部规划跟随全局规划，并实时在铰接车周围生成一条满足车辆非完整性的无碰撞轨迹。两个规划层使得系统技能适应大范围导航，又可以适应局部环境的变化。

本课题旨在研究铰接车在不同场景下的运动规划问题，包括全局规划和局部规划两个部分。通过分析铰接车的运动学模型，结合车辆的工作环境和任务要求，本文将设计适应铰接车特性的轨迹生成方法，以提升铰接车的运动性能，保证其在实际应用中的高效性、安全性与稳定性。

## 1.2 运动规划技术国内外研究现状

### 1.2.1 基于图搜索的运动规划方法

早期研究聚焦静态结构化环境中的最短路径求解。Dijkstra 算法 [\[1\]](#) 通过广度优先策略首次系统化解决带权图的最短路径问题，但其盲目搜索特性导致时间复杂度高达  $O(n^2)$ ，难以应用于大规模场景。Hart 等人 [\[2\]](#) 提出的 A\* 算法创新性地引入启发式函数  $h(Xiaoming)$ ，将搜索方向引导至目标节点，在保证路径最优性的前提下将复杂度降低至  $O(b^d)$  ( $b$  为分支因子， $d$  为搜索深度)，奠定了启发式搜索的理论基础。然而，经典 A\* 算法仍面临两个关键限制：一是内存占用随搜索空间指数增长，二是启发函数设计依赖先验环境知识，难以应对动态障碍物或未知地形。

动态环境下的增量式搜索技术突破解决了部分实时重规划难题。Stentz 的 D\* 算法 [\[3\]](#) 采用反向搜索机制与增量更新策略，仅在环境变化时局部修正路径，将重规划耗时从  $O(n^2)$  降至  $O(k)$  ( $k$  为变化区域节点数)，使机器人能在未知地形中实时避障。但 D\* 算法仍存在路径振荡问题，且在复杂动态场景中频繁更新导致累积误差。Koenig 等人提出的 LPA\* 通过维护优先队列的键值一致性，实现历史搜索信息的智能复用，使得连续规划场景的搜索速度大幅度提升，但其内存管理机制仍无法有效应对长期运行中的状态爆炸问题。Likhachev 团队同期提出的 AD 与 ARA 算法引入次优性容忍系数，允许算法在有限时间内输出渐进优化的路径，为实时性要求严苛的无人机导航提供了解决方案，但路径质量与计算速度的定量关系仍需人工调参，缺乏自适应能力。

Bulitko 的 LRTS [\[4\]](#) 通过剪枝冗余状态与动态调整搜索深度，将实时启发式搜索的响应延迟控制在毫秒级，但其剪枝策略可能丢失潜在最优路径。Harabor 的 JPS 算法 [\[5\]](#) 针对栅格地图提出“跳跃点”理论，通过识别路径对称性跳过大量无效节点，在保持最优性的前提下将搜索速度提升 10 倍以上，但该方法仅适用于均匀网格环境，且无法处理动态障碍物。Koenig 与 Likhachev 的 RTAA\* [\[6\]](#) 结合在线学习机制，利用实际移动代价动态更新启发函数，使智能体在未知地形中的路径收敛速度得到提升，但其学习过程需要大量交互数据，在稀疏奖励场景中表现受限。

### 1.2.2 基于采样的运动规划方法

基于采样的路径规划方法在地图中随机生成采样路径点，然后将被障碍物占据的采样点从候选序列中去除，最后相邻的采样点进行连接，可以得到一路径。与图搜索的方法不同，基于采样的规划方法更适合在高维空间中加速求解路径。在概率上基于采样的路径规划算法具有完备性，只要采样空间中解存在，那么在足够的时间内就一定能够找到可行解。

LaValle 提出的 RRT (Rapidly-exploring Random Trees)，首次将随机采样与树形结构结合，通过偏向未探索区域的生长策略有效应对高维构型空间和非完整约束问题。RRT 的概率完备性 奠定了采样算法的理论基础，但其生成的路径质量无法保证最优。Kuffner 与 LaValle [1] 提出 RRT-Connect，通过双向树生长与贪心连接策略显著提升单次查询效率，解决了传统 RRT 在复杂环境中收敛速度慢的问题，成为实时机器人导航的重要工具。然而，这一阶段的算法仍局限于可行解搜索，缺乏对路径最优性的理论保证。针对动态环境下的实时规划需求，Bruce 与 Veloso [2] 提出 ERRT (Execution-extended RRT)，引入路径点缓存和自适应成本惩罚机制，在重规划中复用历史路径信息，解决了传统 RRT 在动态障碍物场景下的重复计算问题，首次实现了机器人系统的实时在线规划。然而，ERRT 的路径优化依赖启发式规则，缺乏全局最优性保证。Gammell 等人 [3] 的 Informed RRT 通过椭圆启发式采样将搜索空间限制在可能改进当前解的椭圆区域内，显著提升收敛速度与最终路径质量，首次在理论上证明采样算法可渐进趋近最优解。这一工作揭示了传统 RRT 因全局均匀采样导致的效率瓶颈，为后续最优化导向算法奠定了基础。Gammell 等人 [4] 的 BIT\* (Batch Informed Trees) 融合图搜索 (A\* 启发式) 与批处理采样，在隐式随机几何图 (RGG) 上执行增量搜索，解决了高维空间中传统算法收敛缓慢的问题。这些工作标志着采样算法从“可行解搜索”迈入“最优解保证”的新阶段，但其计算复杂度与高维场景的适应性仍面临挑战。

### 1.2.3 基于曲线插值的运动规划方法

基于曲线插值的运动规划相比于基于图搜索和采样的规划方法，具有路径平滑性的特点。早期研究多集中于静态环境下的几何路径求解，RS 曲线就是一个典型的工作，它奠定了车辆运动规划的理论基础。该研究在 Dubins 曲线基础上引入倒车机

制, 证明了包含至多两个尖点的 CCSSCC 型路径(圆-圆-直线-圆-圆)即可构成最优解。通过构建 48 种候选路径形式, 解决了双向行驶车辆在最小转弯半径约束下的最短路径问题。但该方法局限于二维平面几何推导, 未考虑动态障碍物和实时计算需求, 且候选路径的枚举式搜索在复杂场景下计算效率较低。随着自动驾驶技术的发展, 2017 年百度 Apollo 团队在 EM Motion Planner 中提出分层运动规划框架, 标志着路径规划开始向动态环境适应性和多约束协同优化方向演进。该系统采用 Frenet 坐标系下的路径-速度迭代优化, 结合动态规划粗搜索与二次规划细优化, 首次实现了交通规则、障碍物决策与运动平滑性的多目标协同。其创新性地将决策问题转化为可行轨迹生成, 通过构建凸可行空间保障解的存在性。Gao 等人 [1] 基于安全飞行走廊和贝塞尔曲线, 使用优化的方法去构建时间最优轨迹, 推动了无人机领域的实时规划进展。该方法创新性地将快速行进法(FMM)与 ESDF 距离场结合, 构建速度自适应的时空索引路径, 解决了传统方法时间分配不合理的痛点。通过贝塞尔曲线的凸包特性, 首次实现了轨迹整体安全性和高阶动力学约束的严格保证。实验证明其能在未知环境中实时生成安全轨迹, 但飞行走廊膨胀策略在狭窄空间可能过于保守, 且未有效处理动态障碍物的预测与避让。Nguyen 团队提出多面体导航工具箱, 将路径规划抽象为凸多面体序列内的 B 样条优化问题。通过将 B 样条转换为等效贝塞尔表示, 显著提升了局部凸性约束的紧密度, 使线性约束控制点即可保证全局安全性。该工作构建了从栅格地图到多面体地图的完整流程, 支持最小控制点、代数解等多种优化模式。但多面体序列的构建依赖静态环境假设, 在动态场景中需要频繁重规划, 且高维凸优化问题的实时性仍有提升空间。

#### 1.2.4 基于优化的运动规划方法

2015 年 Li 等人 [2] 首次提出基于时间最优控制的动态优化框架, 将平行、垂直等规则停车场景与不规则障碍物场景统一建模。该方法严格描述车辆运动学、机械约束与碰撞条件, 通过内点法求解非线性优化问题。其创新性在于构建了开放框架, 允许用户自定义约束与优化目标。但缺陷在于计算复杂度高, 难以满足实时性需求。Kondak 等 [3] 提出将路径规划转化为非线性最优控制问题, 利用序列二次规划(SQP)直接优化轨迹, 引入人工势场处理障碍物碰撞, 支持全维车辆模型。然而, 其势场函数设计依赖经验参数, 且未解决非凸优化对初值敏感的问题。Arantes 等 [4] 针对离散时间步间的“跳跃碰撞”问题, 提出基于混合整数线性规划(MILP)的连续时间

碰撞避免编码。通过约束相邻状态位于障碍物同侧，确保轨迹段安全性，并将机会约束引入风险分配框架，平衡路径安全性与保守性。该方法显著提升了非凸环境下的路径安全性，但 MILP 的 NP-hard 特性限制了其扩展性，难以应对大规模障碍物场景。Zhang 与 Borrelli [1] 进一步提出基于强对偶定理的平滑碰撞约束重构，将非凸避障条件转化为可微优化问题，支持全维物体与复杂障碍物，并通过符号距离函数量化侵入程度，实现“最小侵入”轨迹生成。但障碍物需分解为凸集，预处理成本较高。Zhang 等提出分层优化框架 H-OBCA [1]，融合混合 A 的全局搜索与优化碰撞避免（OBCA）的局部精细化。混合 A 基于简化模型快速生成粗路径，OBCA 利用全维模型优化平滑性与动态可行性，解决了非凸优化依赖初值的瓶颈。实验表明其在狭窄场景中可实现实时规划，且轨迹可被底层控制器精准跟踪。然而，其性能受限于混合 A\* 的路径质量，复杂障碍布局下可能陷入局部最优。

### 1.3 铰接车运动规划技术国内外研究现状

铰接车路径规划的研究历程可追溯至 2010 年代初期的几何建模与静态环境规划阶段。早期的代表性工作如 Choi 与 Huhtala 提出基于 Bézier 曲线的分层规划框架，其核心贡献在于将参数化曲线与搜索算法结合，通过离线生成满足非完整约束的运动基元库（包含曲率连续的前进/倒车路径片段），再结合 A\* 算法在线搜索全局路径，并引入非线性优化修正关节角度超限问题。紧随其后的 Sarata 团队针对轮式装载机的 V 型作业路径提出对称 Clothoid 曲线连接策略，通过数学证明得出 Clothoid 曲线的曲率线性变化特性可消除传统线-弧路径的曲率突变问题，但其固定路径模式依赖预设的装载点与卸载点位置，难以适应砂石堆形状动态变化的非结构化场景。随着实时计算能力的提升，2010 年研究重点转向动力学模型集成与在线优化算法。Nayl 等（2017）首次将模型预测控制（MPC）与 Bug 算法结合，提出基于铰接车误差动力学模型的滚动优化框架，通过实时调整前轮转向角与车速实现动态避障。该方法在仿真中成功处理了突然出现的移动障碍物，路径跟踪误差控制在  $\pm 0.15m$  以内，但未解决多周期任务的长时优化问题，且在狭窄通道中易因局部最优陷入死锁。Chen 等对比了简化单车模型与双体铰接模型的规划效果，发现复杂模型在双向倒车场景中可生成更短的安全路径，但计算耗时增加，揭示了模型精度与实时性之间的固有矛盾。Hong 团队进一步引入 3D 地形高程数据构建动力约束，提出动态规划（DP）与 MPC 结合的混合方法，在轮式装载机的崎岖地形实验中实现能耗降低，但其离

散化状态变量导致高维空间下的计算复杂度呈指数级增长，难以扩展至多关节车辆。Kawabe 等设计的分级优化框架将全局任务分解为铲取点选择、卸载点规划与局部路径生成子问题，并首次引入深度强化学习（DRL）实现动态砂石堆的自适应装载路径规划。Zhao 等采用 Chebyshev 伪谱法对重型铰接车垂直泊车问题进行时间最优轨迹优化，通过精确建模非完整约束，使得求解的精度大幅提高。Shi 团队提出的自适应 MPC 跟踪系统通过曲率扰动补偿机制，在非均匀路面下将路径跟踪的横向位移误差有所降低，但其依赖 RTK-GPS 的高精度定位信号，在卫星遮挡的室内仓储场景中性能显著下降。

不难看出，基于优化的运动规划算法在近期一直是研究的热点，原因在于其相较于基于搜索和采样的规划方法，能够求解出更加圆滑且满足各种约束的轨迹。其缺点是在大量约束的情况下求解会变得很困难。但是在当下计算机高速发展，优化问题的求解时间逐渐不再是问题的瓶颈。所以基于优化的轨迹规划方法是目前为止最值得研究的一个方向之一。

## 1.4 本文研究的主要内容

本研究针对无人铰接车辆在复杂非结构化环境中的轨迹规划问题，使用全局与局部规划结合，旨在解决传统路径规划算法因忽略铰接车强非完整性与运动学约束导致的路径不可跟踪、避障安全性不足等核心问题。

本课题的研究内容和解决的关键问题如下：

(1) 基于微分平坦全局轨迹规划方法。

第一，我们将微分平坦理论引入到铰接车的轨迹规划中，使得系统的输入和状态变量可以用这组平坦变量的多阶导数表示，用于简化轨迹规划问题，并加强求解的数值稳定性。

第二，我们将铰接车的前桥和后桥建模为两个凸结构相连，基于安全走廊来表示铰接车的安全避障问题。此外，我们引入 Minco (Minimum Control Effort Spline) 样条曲线，建立基于样条优化的轨迹规划问题，将原本高维度的优化问题转化为低维度且安全避障的优化问题。

第三，设计曲率约束松弛函数，从而将路径规划问题转化为无约束优化问题，利用 L-BFGS 算法高效求解，确保轨迹满足最大转向半径与避障要求。

(2) 基于 NMPC 和安全走廊的局部轨迹规划研究。

第一,在优化的框架下,我们将轨迹规划问题建模为非线性模型预测控制(NMPC)问题,这是一个典型的最优控制框架。

第二,由于传统的避障约束带来的严重的非凸性,促使优化问题的求解变得极其困难。为此,同全局规划一样,我们引入安全走廊保证凸性。研究在走廊下如何重构避障约束,降低优化求解的难度,使数值解接近最优解。

第三,由于大量的约束和优化变量会使得问题极难求解,产生维度灾难。为此,我们设计外罚函数策略和微分同胚变换。在不改变原问题凸性的情况下,将问题转换为无约束的优化,大幅增强求解速度。

## 1.5 本文结构安排

本文的结构安排和各章的内容如下:

**第一章: 绪论。**本章节主要阐述无人铰接车在复杂环境(如港口、矿山)中的运动规划研究背景与意义,分析铰接车非完整约束带来的轨迹规划挑战。系统梳理国内外运动规划技术的研究现状,包括基于图搜索、采样、曲线插值与优化的方法,以及铰接车专用算法的进展与局限性。最后,明确本文研究目标与创新点,提出融合全局与局部规划的优化框架,并概述全文的章节安排。

**第二章: 无人铰接车运动学模型建立和运动规划系统设计。**本章节构建铰接车轨迹规划的数学模型,推导其运动学方程与转弯半径。将运动规划分为全局和局部两个部分,基于最优控制问对规划问题建模。阐述问题的求解方法,包括动态规划与模型预测控制(MPC)的结合策略,并将问题离散化为非线性规划(NLP)形式。

**第三章: 无人铰接车微分平坦全局轨迹规划方法。**基于铰接车微分平坦特性,以Minco样条参数化轨迹,结合RRT-Connect生成初始路径;设计凸多边形安全走廊约束,通过松弛函数处理曲率与避障条件,利用L-BFGS算法实现高效优化。针对多阶段任务(如码头装卸),给出换向点处理机制,并通过仿真验证算法在复杂场景下的有效性。

**第四章: 基于安全走廊的无人铰接车局部轨迹规划方法。**本章节针对传统Hybrid A\*无法适应铰接式车辆问题,提出改进方法并将其应用于我们的算法的热启动。构建基于安全走廊的NMPC框架,以舒适性、能耗与时间为优化目标,集成精确碰撞检测算法(Bresenham栅格检测)与高阶运动学约束,通过IPOPT求解器实现动态避障与实时轨迹优化。

第五章：无人铰接车运动规划系统仿真实验。基于 ROS 机器人系统和 RVIZ 仿真可视化软件，搭建仿真平台。针对第三章的全局规划和第四章的局部规划，分别设计有障碍物躲避、泊车、侧边停车、狭窄空间穿行和对向掉头等仿真场景。以分别验证两个算法的可行性。

第六章：总结。本章节总结研究内容和研究成果，分析相应的创新点。对目前存在的问题进行讨论，对未来的研究工作进行展望。

## 第2章 铰接车建模与运动规划系统设计

本章节将首先我们的运动规划系统框架，它包括有全局规划和局部规划两个部分。然后基于最优控制问题框架给出轨迹规划问题的求解方法。介绍外点法和内点法在优化问题中的使用。

### 2.1 运动规划系统总体方案设计

受开源机器人框架 ROS 的启发，我们的运动规划系统采用全局-局部双级递进架构实现任务分解。在全局层面，规划过程解耦为路径搜索与轨迹优化两个阶段：首先基于 RRT/A\*/JPS 等前沿搜索算法快速生成无碰撞粗解路径，随后通过基于能量最优代价构建优化模型，在保障路径连续性的同时实现平滑性提升与运动学上的可行性。局部规划层面向实时运动控制需求，提出改进型 Hybrid A\* 算法生成满足车辆运动学约束的可行路径，进而构建速度剖面作为轨迹优化初值。最终通过非线性模型预测控制（NMPC）框架实施时空联合优化，其安全边界通过动态安全走廊技术进行在线更新，确保规划结果同时满足实时性要求与严格的安全准则。运动规划系统的总体方案如下：

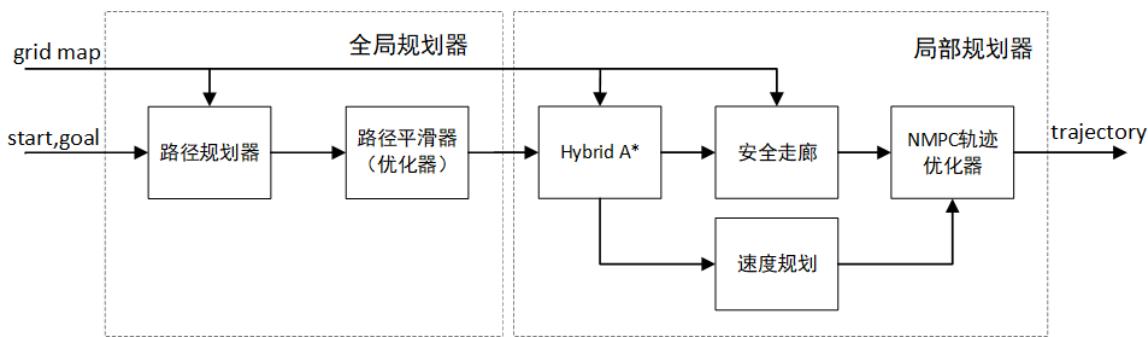


图 2-1 运动规划算法框架

### 2.2 铰接车运动学模型建立

铰接车结构上由前桥和后桥组成，中间通过液压或者舵机驱动铰接角完成转向。本文介绍的铰接车结构模型如图2-2。我们假设以前轴中点作为参考点，前轴的航向角为  $\theta_1$ ，后轴航向角为  $\theta_2$ 。其中  $O$  点为瞬时转向圆心， $p_f$  为前轴中点 pr 为后轴中

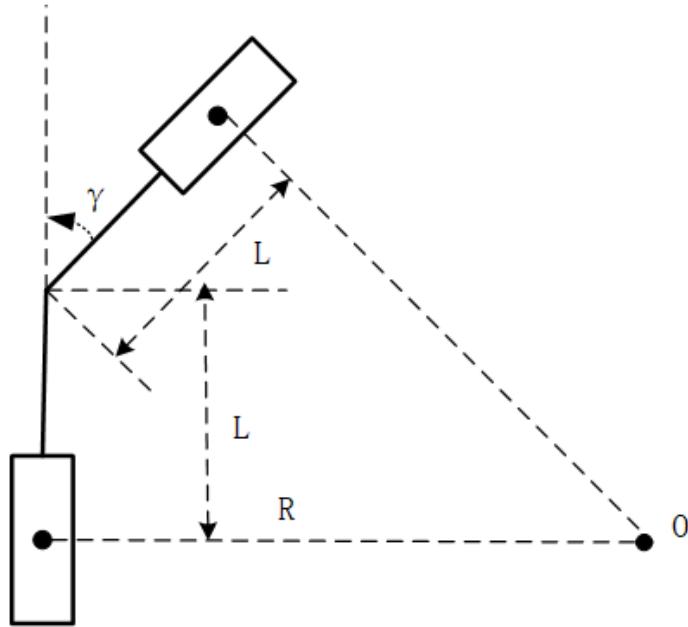


图 2-2 铰接车简易模型

点,  $l_f$  是前轴长度,  $l_r$  是后轴长度,  $\gamma$  是铰接角。为了推出铰接车的运动学模型, 我们做出两个假设。(1) 铰接车辆始终保持低速运动, 忽略动力学特性。(2) 假设车辆不发生侧滑。首先由于车辆不会发生打滑, 所以车辆运动时前后车轮完全满足车辆的滚动约束, 即速度方向和车辆的朝向保持一致。可以得到方程组2-1。

$$\begin{aligned}\dot{x}_1 &= v \cos \theta_1 \\ \dot{y}_1 &= v \sin \theta_1 \\ \dot{x}_1 \sin \theta_1 - \dot{y}_1 \cos \theta_1 &= 0 \\ \dot{x}_2 \sin \theta_2 - \dot{y}_2 \cos \theta_2 &= 0\end{aligned}\tag{2-1}$$

由于前桥和后桥通过铰接角刚性连接, 我们可以推导出前桥参考点和后桥参考点的坐标变换公式如下:

$$\begin{aligned}x_1 &= x_2 + l_2 \cos \theta_2 + l_1 \cos \theta_1 \\ y_1 &= y_2 + l_2 \sin \theta_2 + l_1 \sin \theta_1 \\ \theta_2 &= \theta_1 - \gamma\end{aligned}\tag{2-2}$$

以前桥为参考点, 定义  $(x, y, theta, gamma)$  为铰接车的位姿, 他们分别代表着  $x_f, y_f, \theta_1, \gamma$ 。在我们的研究中铰接车的前桥和后桥相等, 即  $l_f = l_r = L = 1.3m$ 。联合上式我们可

以得到铰接车的运动学模型为:

$$\frac{d}{dt} \begin{bmatrix} x_f \\ y_f \\ \theta \\ \gamma \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ vtan(\gamma/2)/L + \omega/(cos\gamma + 1) \\ \omega \end{bmatrix} \quad (2-3)$$

其中  $v, w$  是输入量, 分别代表这前轴速度和铰接角速度。其次我们还可以推导出铰接车的转弯半径为:

$$R = \frac{v}{\dot{\theta}} = v \cdot L \frac{1 + cos\gamma}{vsin\gamma + L\omega} \quad (2-4)$$

特别的, 如果我们忽略铰接车的动态转向, 只保留静态转向, 即  $\omega = 0$ :

$$R_{steady} = \frac{L}{tan(\gamma/2)} \quad (2-5)$$

### 2.3 基于最优控制的运动规划问题建模

本文设计的全局和局部规划器都使用了基于优化的方法, 将问题构建在最优控制的框架下求解。最优控制的一大特点就是可以清晰的把各种约束表示在问题里, 例如运动学方程约束、状态受限约束和执行器受限约束, 因此最优控制在轨迹规划中被广泛关注。假设系统状态方程为  $\dot{x}(t) = f(x(t), u(t))$ , 最优控制就是从可选择的容许控制集  $U$  中, 找到一个控制矢量  $u(t)$ , 使得系统到终端时刻时, 性能指标  $J$  取最小值, 满足条件的控制成为最优控制  $u^*(t)$ , 对应的状态方程的解成为最优轨线  $x^*(t)$ 。最优控制侧重于得到系统的最优控制量, 轨迹规划与之不同, 它侧重于得到系统的状态量, 即最优轨线。轨迹规划在最优控制的表述下属于代价函数是综合型或鲍尔扎型的性能指标, 它通常包括能量消耗和终点代价两个部分。

基于连续系统的鲍尔扎型最优控制可以将轨迹规划表述为如下:

$$\begin{aligned} \min_{x(t), u(t), t} \quad & J = \Phi[x(t_f)] + \int_{t_0}^{t_f} L[x(t), u(t), t] dt \\ \text{s.t.} \quad & \dot{x} = f(x(t), u(t)) \\ & x_{min} \leq x(t) \leq x_{max}, u_{min} \leq u(t) \leq u_{max}, t \in [t_0, t_f] \\ & x(t_0) = x_{start}, x(t_f) = x_{goal}, \\ & h(x(t)) \cap C_{occ} = \emptyset \end{aligned} \quad (2-6)$$

其中  $t_f$  是终端时刻,  $\Phi[x(t_f)]$  是终端代价, 也就是对目标的偏差。 $x(t)$  和  $u(t)$  分别代表状态量和控制量,  $L[x(t), u(t), t]$  表示过程代价, 在轨迹优化中可以是能量消耗, 也就是对控制量的代价。

对于离散系统, 一般代价泛函可以表示为:

$$J = \Phi[x(N)] + \sum_{k=k_0}^{N-1} L[x(k), u(k), k] \quad (2-7)$$

$x(t)$  和  $u(t)$  表示车辆的状态量和控制量,  $J$  是最优控制的目标函数。在约束中,  $\dot{x}(t) = f(x(t), u(t))$  是车辆的运动学方程,  $x_{min} \leq x(t) \leq x_{max}$ ,  $u_{min} \leq u(t) \leq u_{max}$  分别代表系统的状态受限和输入受限,  $x(t_0) = x_{start}$ ,  $x(t_f) = x_{goal}$  表示起点和终点约束。 $h(x(t)) \cap C_{occ} = \emptyset$  表示车辆的避障约束。其中  $C_{occ}$  表示障碍物区域。下面将详细介绍铰接车轨迹规划中的各个组成部分。

## 2.4 最优控制问题的解法

最优控制问题的求解方法主要可分为两类。一类是 解析解法, 基于变分法原理与极小值原理的经典分析方法。这类方法能够通过严格的数学推导获得控制律的显式解析解, 具有理论完备性强的特点。然而其应用范围受限于问题复杂度, 当系统动力学方程非线性程度加剧或路径约束增多时, 解析解的推导将面临难以逾越的数学障碍, 导致方法失效。另一类是 数值解法, 采用离散化策略将连续最优控制问题转化为非线性规划问题求解。此类方法虽能处理复杂约束条件与非线性系统, 但随着精度要求的提升(包括时间离散密度与状态量精度), 会导致优化变量维度呈指数级增长。由此产生的计算复杂度将显著延长求解时间, 严重影响控制系统的实时性表现。这两类方法呈现不同的特性, 解析解法在简单场景中具有理论优势却缺乏工程适用性。而数值解法具广泛适应性但受制于“维度灾难”带来的计算瓶颈。

### 2.4.1 极小值原理

作为最优控制理论的基石之一, 庞特里亚金于 1956 年建立的极小值原理突破了经典变分法的约束条件限制。该原理通过引入协态变量构建哈密顿函数体系, 在容许控制域内寻找使哈密顿函数极小的最优控制序列。相较于传统变分法, 其核心创新在于允许控制变量存在闭集约束, 并能有效处理终端状态受限的边值问题。在工

程实践中，该原理为航天器轨道转移、机器人轨迹规划等典型最优控制问题提供了严格的判据，但在强非线性系统中的应用常受限于两点边值问题的解析求解困难，这推动了后续数值方法的发展。

### 2.4.2 基于打靶法的数值求解

前文探讨的极值条件理论虽然为最优控制律的存在性提供了严格判据，但在实际应用中面临显著的计算瓶颈。当系统动力学呈现强非线性特征或存在多类路径约束时，基于庞特里亚金原理的解析求解往往难以实现。针对此类复杂场景，需要建立具有工程可实现性的数值优化策略。本节阐述的打靶法即通过离散化处理将连续最优控制问题转化为有限维参数优化问题，不仅保持了最优化条件，同时为应用成熟的数值优化算法（如序列二次规划、内点法等）奠定了基础。

该方法的核心思想在于，将状态微分方程在时间域上离散为代数方程组，进而构造具有可计算结构的非线性规划模型。具体而言，打靶法对控制量进行离散化，状态量作为控制量的函数，将原问题转化为关于离散节点的控制量寻优过程。这种方法只优化控制变量，变量的维度一般不会很高，对于最优控制有很大的应用场景。但是对于轨迹规划他就显得不是那么的时候，因为它并不会直接获得系统的状态，系统的状态需要由优化后的控制输入根据运动学模型进行航迹推演。因此，打靶法更适合轨迹跟踪这一类控制问题。打靶法的优化问题如下：

$$\begin{aligned} \min_{u_0, \dots, u_N} & J(u_0, u_1, \dots, u_N, x_1, x_2, \dots, x_N) \\ & x_{k+1} = x_k + f(x_k, u_k) \cdot \Delta t, \quad k = 0, \dots, N \end{aligned} \tag{2-8}$$

$$x_0 = x_{start}$$

如上式，打靶法只优化控制变量，所以代价函数中的状态量全部时关于控制量  $u$  的函数。若想获得轨迹，还需要将优化出来的控制变量  $u_0, \dots, u_k$  使用运动学方程  $x_{k+1} = x_k + f(x_k, u_k)$  递推得到。下一小节我将介绍一种更为直接的数值方法，它可以直接获得系统的状态，也就是轨迹。

### 2.4.3 基于配点法的数值求解

与打靶法不同，打靶法只离散控制量，而配点法通过离散控制量和状态量将连续最优控制问题转化为非线性规划问题（nonlinear programming problem, NLP），其数值鲁棒性在工程领域广受认可。由于直接将状态量纳入到优化变量，所以配点法相较于打靶法更适合用来做轨迹规划。这种方法在配置节点处要求满足动力学方程约束，配点之间使用多项式或者其他曲线进行差值。当配点较为密集时，配点之间也可以近似用直线连接。

本次设计中我们使用配点法求解，所以我们先对连续最优控制问题进行离散化，离散化后可以表示如下：

$$\begin{aligned}
 & \min_{x(t), u(t), t} J = \Phi[x(N)] + \sum_{k=0}^{N-1} L[x(k), u(k), k] \\
 & \text{s.t. } x(k+1) = x(k) + \Delta t \cdot f(x(k), u(k)) \\
 & \quad x_{\min} \leq x(k) \leq x_{\max}, u_{\min} \leq u(k) \leq u_{\max}, k \in 0, N-1 \\
 & \quad x(0) = x_{start}, x(N) = x_{goal}, \\
 & \quad h(x(k)) \cap C_{occ} = \emptyset
 \end{aligned} \tag{2-9}$$

上述式子中约束条件依次为系统动力学约束、状态受限约束、输入受限约束、起点终点约束和路径约束（安全避障约束）。

第三章和第四章中我们会分别针对全局规划和局部规划对公式2-9设计各自的具体目标函数和约束形式，并使用非线性优化求解器进行求解。非线性优化求解器我们将在下一节中介绍。

## 2.5 非线性规划问题的求解

在上一节中我们把最优控制离散化为了一个非线性规划（NLP）问题，但是这个问题一般具有很强的非线性和非凸性。一般处理 NLP 问题的求解方法有很多，包含增广拉格朗日乘子法（ALM）、交叉方向乘子法（ADMM）、罚函数法、有效集法和可行方向法等。这些所有的方法都假设了问题和约束都是凸的，目前对于非凸问题还没有有效的求解方法。由于罚函数在工程中表现优异，在本设计中也将使用罚函数。

### 2.5.1 罚函数法

无约束问题的求解有梯度一类的方法和牛顿法一类的方法，这些方法被公认可有效解决无约束优化问题。所以对于带约束的优化问题我们一般可以使用罚函数把它化为无约束优化问题。罚函数有内罚函数和外罚函数两种，其代表性求解器分别是 IPOPT 和 G2O。考虑如下一般的优化问题：

$$\begin{aligned} & \min_x J(x) \\ & s.t. g(x) < 0 \\ & h(x) = 0 \end{aligned} \tag{2-10}$$

公式 [?] 中包含通用的两种约束，等式约束和不等式约束。我们可以直接把这两种约束使用罚函数加到代价函数中：

$$\min_x J(x) + Pen1(g(x)) + Pen2(h(x)) \tag{2-11}$$

罚函数 Pen1 针对于处理不等式约束，它必须具有的特性是在可行域外时代价尽量大，在可行域内时代价尽量小。罚函数 Pen2 针对于处理等式约束，它必须具有的特性是在可行点外时代价尽量大，在可行点内时代价尽量小。在大多数时候，我们其实并不关心使用内罚还是外罚，我们更关心的是找到两个合适的罚函数来使得问题求解起来更加的鲁棒。

### 2.5.2 非线性规划问题的初始解

罚函数仅仅是将带约束的问题转化为了无约束优化问题，求解这么一个无约束优化问题还需要一个迭代初始解。在凸优化中，迭代初始解我们很关心它的质量，问题的凸特性会使得初始解迫向最优解迭代，初始解的不同只是影响收敛快慢而已。但是在我们的非线性规划问题中，问题是强非凸的，所以不同的初始解可能最终收敛到的极值并不一样。所以找到一个优秀的初始解往往可以使得最终的求解结果接近最优解。如图2-3所示，非线性规划算法处理初始轨迹 A1 后将迭代优化至轨迹 A2，而选取初始方案 B1 时则会收敛于轨迹 B2。通过图示对比可明显观察到，采用初始轨迹 A1 能使系统收敛至路径长度更优的局部极小值。该现象验证了初始参数设定对优化结果具有决定性作用，因此初始解的获取策略是极为关键的。

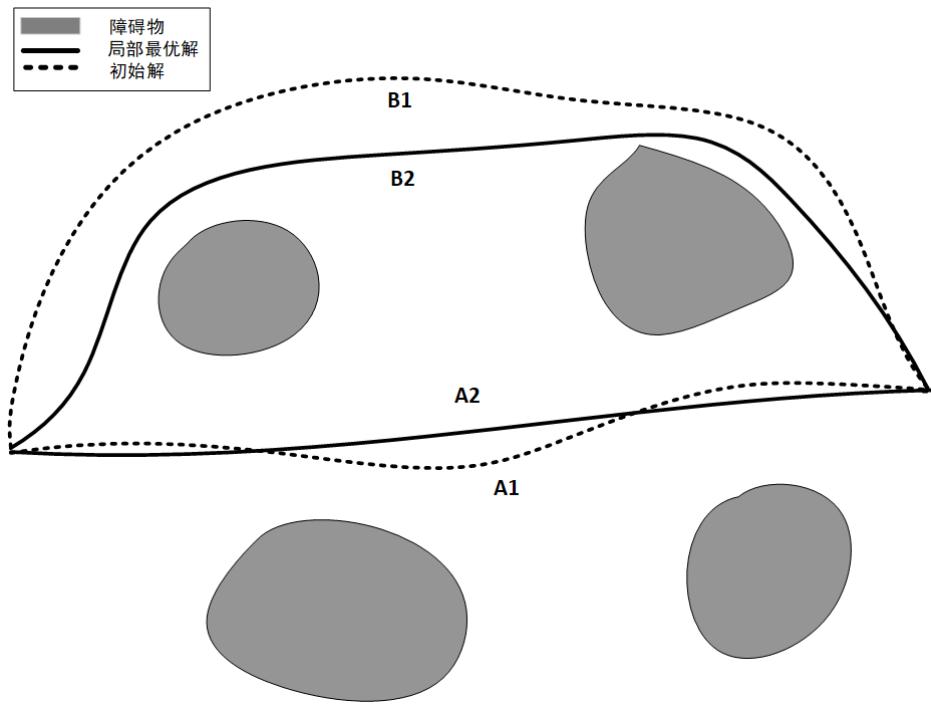


图 2-3 不同初解

## 2.6 本章总结

本章系统构建了铰接车运动规划的理论框架与求解体系。通过全局-局部分层规划架构实现任务分解，全局层基于搜索算法生成粗解路径，结合能量最优进行平滑优化；局部层通过改进型 Hybrid A\* 算法生成满足运动学约束的轨迹初值，并利用非线性模型预测控制（NMPC）进行时空联合优化，辅以动态安全走廊技术实时更新安全边界，兼顾规划效率与安全性。

针对铰接车运动学特性，建立低速非侧滑条件下的四维状态模型，推导了前/后桥坐标变换关系与瞬时转向半径方程，为轨迹优化提供核心运动学约束。基于最优控制理论，将轨迹规划问题转化为多约束非线性规划问题，对比解析解法与数值解法的适用性，提出采用配点法直接优化状态/控制变量，并通过罚函数法处理非凸约束，解决了传统方法在复杂场景下的维度灾难与收敛性问题。

## 第3章 无人铰接车微分平坦全局轨迹规划方法

在传统的全局规划问题中，目标是找到从起点到终点的最优路径，最优的指标一般是路径最短。例如 A\* 算法和 Dijkstra 算法可以在栅格地图中搜索出路径代价最小的路径，RRT\* 系列算法在概率完备下得到路径也是距离最短的。然而，这些全局规划算法并不会考虑车身的朝向和曲率等信息。它们仅仅针对全向或者差速机器人会非常有效，面对具有强非完整性的机器人来说就显得吃力了。考虑以下的场景：

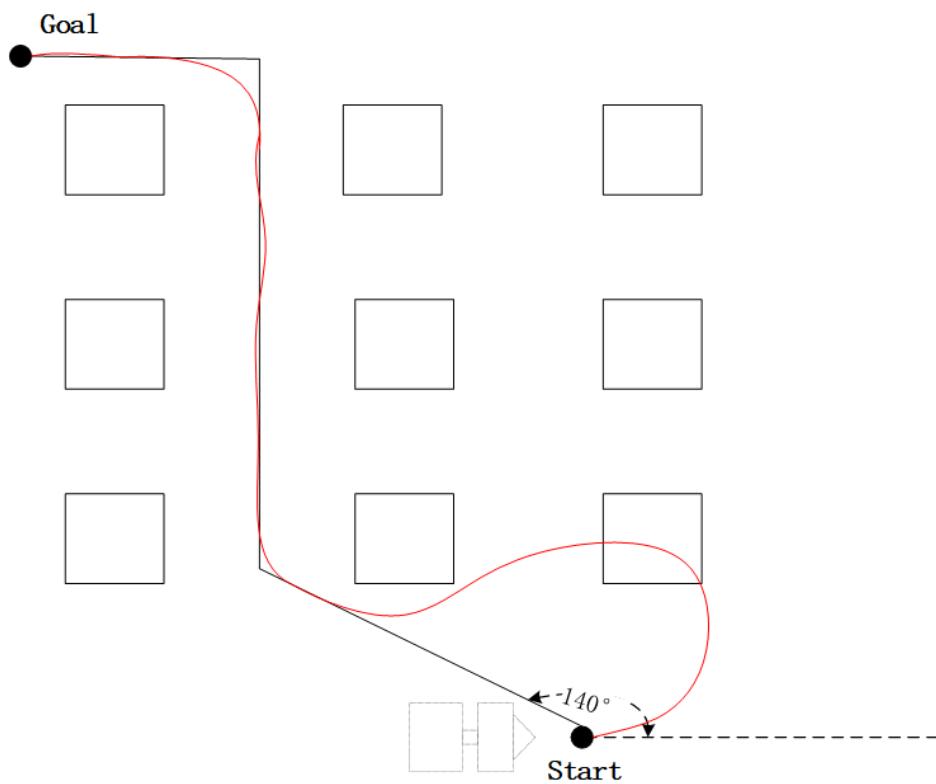


图 3-1 路径规划算法特例

如图3-1，需要规划起点 Start 到终点 Goal 的路径。一般的算法可能规划出图中黑色的路径，那是路径最短的。可以看到路径的起点切向和机器人的朝向相差 140 度。对于差速机器人跟踪这条路径的话，他会先原地转向然后继续跟踪其他点。对于全向机器人它直接可以顺着路径走，不必掉头。但是对于类车转向和铰接转向底盘，这两种底盘在转向时不仅有严格的非完整性，同时转向半径非常大。这会直接导致在开始的一段路径跟踪时会走出图中红色的路径，严重不贴合全局路径。这会带来

出乎意料的后果，例如撞向障碍物或走出地图边缘。本章将基于优化理论和 Minco 样条，提出一种适用于铰接车辆的全局路径优化方法，可以在整段路径上保证严格的转弯半径约束并保证安全躲避碰撞。

### 3.1 铰接模型微分平坦性

#### 3.1.1 铰接车模型的平坦输出

微分平坦性为非线性控制提供了一种“捷径”，通过平坦输出的参数化，可以将复杂的轨迹优化问题转化为简单的代数计算。它在机器人、航空航天和自动化领域有广泛应用，是现代控制理论中连接几何方法与实际工程的重要工具。下面给出微分平坦的定义：

**微分平坦系统：**对于一个非线性系统，如果能找一组输出变量，使得所有的状态变量输出变量都可以用这组输出变量及其多阶导数所表示，则称这个系统为微分平坦系统。

对于铰接车来说，实际工作环境往往非常复杂（比如障碍物多、空间狭窄）。这时候做全局路径规划，不需要用特别精确的运动学模型——因为模型太精确反而会增加计算量，影响实时性。我们完全可以把需要精细调整的部分交给局部规划来处理，这样既省资源又合理。在这里我们忽略动态转向，只保留静态转向模型。令  $v$  和  $\gamma$  为系统输入，简化的系统模型为：

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= v \tan(\gamma/2)/L \\ \dot{\gamma} &= \omega\end{aligned}\tag{3-1}$$

对于上述的铰接车的运动模型，我们选择  $x, y$  作为平坦输出，这在轨迹规划中可以带来很好的性质。因为  $x, y$  就是我们需要的轨迹，是具有实际的物理意义的。下

面我们给出装载机的微分平坦系统:

$$\begin{aligned}
 v &= \eta \sqrt{\dot{x}^2 + \dot{y}^2}, \\
 \theta &= \arctan2(\eta \dot{y}, \eta \dot{x}), \\
 \gamma &= 2\arctan(\eta(\dot{x}\ddot{y} - \dot{y}\ddot{x})L / (\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}), \\
 k &= (\dot{x}\ddot{y} - \dot{y}\ddot{x}) / (\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}.
 \end{aligned} \tag{3-2}$$

其中  $x$  和  $y$  是系统的平坦输出变量,  $(v, \theta, \gamma, k)$  是我们选取的状态变量。 $k$  为轨迹的曲率, 值得指出的是曲率其实也是铰接角的函数, 但是这里只在路径层面表示。 $n \in \{-1, 1\}$  是车辆运动的方向, 分别代表前进和后退两种情况。

## 3.2 轨迹的参数化表达形式

在自动驾驶中参数化曲线的方法有很多, 例如五次多项式、贝塞尔曲线、B 样条、螺旋线、Dubin 曲线和 RS 曲线。MINCO 曲线作为一种基于五次多项式的曲线类型, 它在处理微分平坦系统上有着天然的优势【文献】。我们这一小节将介绍 minco 曲线的原理, 并且运用它处理我们的铰接模型运动规划问题。

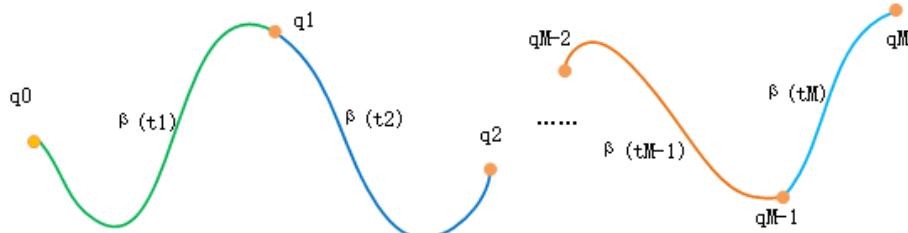


图 3-2 minco

### 3.2.1 MINCO 样条曲线基本原理

在【3.1】节中我们给出了铰接式车辆模型的平坦系统和平坦输出  $x(t), y(t)$ , 在这一小节中我们将详细介绍在平坦输出下的两种轨迹优化形式——BVP 和 BIVP。

#### (1) BVP (Boundary Value Problem)

在多项式规划中一个典型的 BVP (Boundary Value Problem) 问题可以表示为:

$$\begin{aligned} & \min \int_0^T u(t)^T W u(t) dt \\ \text{s.t. } & z^s(t) = u(t), \forall t \in [0, T], \\ & z^{[s-1]}(t_0) = \bar{z}_s, z^{[s-1]}(t_M) = \bar{z}_e. \end{aligned} \quad (3-3)$$

其中  $z(t)$  是样条曲线, 它可以是我们上述提到的贝塞尔曲线、B 样条或者多项式形式。 $u(t)$  是系统的输入, 它是  $z(t)$  关于时间的  $s$  阶导数。 $\bar{z}_s$  和  $\bar{z}_e$  分别是起点状态和终点状态。对于以上最小控制输入问题, 控制量  $u(t)$  是平坦变量的  $s$  阶次导数, 则 BVP 问题的最优解是一个  $2s - 1$  次多项式【文献】。

当  $s = 3$  时这是一个 minimumJerk (最小化加速度代价) 问题, 他的最优解是一个五次多项式  $\blacksquare$ 。轨迹曲线  $z(t)$  可以被表示为关于时间  $t$  多项式函数  $\beta(t)$ :

$$\begin{aligned} z = \beta(t) &= \lambda^T(t)c = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5 \\ &= [1 \ t \ t^2 \ t^3 \ t^4 \ t^5] \cdot [c_0 \ c_1 \ c_2 \ c_3 \ c_4 \ c_5]^T \end{aligned} \quad (3-4)$$

其中  $c$  是多项式的系数向量, 可以由起点和终点的各阶状态解出。假设轨迹的时间段是  $T$ , 起点和终点带来的约束可以写为:

$$\begin{bmatrix} x_s \\ v_s \\ a_s \\ x_e \\ v_e \\ a_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \quad (3-5)$$

$$= A_F(t)c$$

在上式中,  $x_s, v_s, a_s$  分别是轨迹起点的位置、速度和加速度,  $x_e, v_e, a_e$  对应为终点的状态。需要说明的是, 上述讨论的边值问题 (BVP) 仅针对一维情况。而实际应用中, 平坦变量通常是二维的 (如平面坐标系中的 x 和 y 方向), 此时可以将其分解为两个独立的一维 BVP 的组合——即分别在 x 和 y 方向求解对应的边界条件, 最终通过坐标合成得到二维轨迹。这种解耦方法的核心假设是两个维度在平坦域内无耦合约束。

## (2) BIVP

为了保证轨迹的灵活性，在实际的规划问题中轨迹规划被表述为多个 BVP 的统一形式，即 BIVP 问题：

$$\begin{aligned}
 & \min_{z(t)} \int_{t_0}^{t_M} u^T(t) \mathbf{W} u(t) dt \\
 & \text{s.t. } z^{(s)}(t) = u(t) \quad \forall t \in [t_0, t_M] \\
 & z^{[s-1]}(t_0) = \bar{z}_s, z^{[s-1]}(t_M) = \bar{z}_e \\
 & z^{[d_i-1]}(t_i) = \bar{z}_i, 1 \leq i < M \\
 & t_{i-1} < t_i, 1 \leq i \leq M.
 \end{aligned} \tag{3-6}$$

对于以上问题，其最优化条件指出：对与最小化  $s$  阶导数的 BIVP 问题，他的最优解的单段轨迹一定是一个  $2s - 1$  次多项式，且这个解具有  $s + 1$  阶导数的连续性【文献】。这意味着我们在处理 minimumJerk ( $s = 3$ ) 的轨迹时，这个轨迹具有 snap 阶 ( $s = 4$ ) 阶的连续性。因此 BIVP 问题的闭式解完全可以由起点状态、终点状态、中间位置和中间变量的连续性约束所得到。如图3-2，由  $M + 1$  个航点可以生成  $M$  段五次多项式轨迹，每个多项式有 6 个系数，一共有  $6M$  个系数。这意味着我们需要  $6M$  个约束才能唯一确定这个轨迹。

### 3.2.2 起点和终点约束方程

假设我们共有  $M + 1$  个航点  $q_0, \dots, q_M$ ，其中起点  $q_0 = [x_s, y_s, v_{sx}, v_{sy}, a_{sx}, a_{sy}]$  和终点  $q_M = [x_e, y_e, v_{ex}, v_{ey}, a_{ex}, a_{ey}]$  完全已知。由此，一共可以得到 6 个约束方程，起点的约束方程为：

$$\begin{bmatrix} x_0 & y_0 \\ v_{0x} & v_{0y} \\ a_{0x} & a_{0y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{0x}^1 & c_{0y}^1 \\ c_{1x}^1 & c_{1y}^1 \\ c_{2x}^1 & c_{2y}^1 \\ c_{3x}^1 & c_{3y}^1 \\ c_{4x}^1 & c_{4y}^1 \\ c_{5x}^1 & c_{5y}^1 \end{bmatrix} \implies b_0 = F_0 \cdot c_1 \tag{3-7}$$

终点的约束方程为：

$$\begin{bmatrix} x_M & y_M \\ v_{Mx} & v_{My} \\ a_{Mx} & a_{My} \end{bmatrix} = \begin{bmatrix} 1 & T_M & T_M^2 & T_M^3 & T_M^4 & T_M^5 \\ 0 & 1 & 2T_M & 3T_M^2 & 4T_M^3 & 5T_M^4 \\ 0 & 0 & 2 & 6T_M & 12T_M^2 & 20T_M^3 \end{bmatrix} \begin{bmatrix} c_{0x}^M & c_{0y}^M \\ c_{1x}^M & c_{1y}^M \\ c_{2x}^M & c_{2y}^M \\ c_{3x}^M & c_{3y}^M \\ c_{4x}^M & c_{4y}^M \\ c_{5x}^M & c_{5y}^M \end{bmatrix} \Rightarrow b_M = F_M \cdot c_M \quad (3-8)$$

### 3.2.3 航点的连续性约束方程

除了已知起点和终点之外，我们还必须保证 BIVP 问题 M-1 个中间航点的连续性。根据 Minco 的理论，minimumJerk 问题 ( $s=3$ ) 具有 snap 阶的连续性。对于每一段多项式轨迹，我们可以求出它的各阶导数为：

$$\begin{bmatrix} x(t) & y(t) \\ v_x(t) & v_y(t) \\ a_x(t) & a_y(t) \\ jerk_x(t) & jerk_y(t) \\ snap_x(t) & snap_y(t) \end{bmatrix} = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \\ 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \\ 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \\ 0 & 0 & 0 & 6 & 24t & 60t^2 \\ 0 & 0 & 0 & 0 & 24 & 120t \end{bmatrix} \begin{bmatrix} c_{0x}^i & c_{0y}^i \\ c_{1x}^i & c_{1y}^i \\ c_{2x}^i & c_{2y}^i \\ c_{3x}^i & c_{3y}^i \\ c_{4x}^i & c_{4y}^i \\ c_{5x}^i & c_{5y}^i \end{bmatrix} \Rightarrow G(t)c_i = \mathbf{b}(t) \quad (3-9)$$

则第  $i$  段轨迹和第  $i+1$  段轨迹的连续性约束方程可以写为：

$$G(T_i)c_i - G(0)c_{i+1} = 0 \quad (3-10)$$

由于一共有  $M-1$  个中间航点，每一个航点带来 5 个约束，所以连续性约束一共有  $5M-5$  个。

### 3.2.4 轨迹的闭式解

令中间航点为  $q_1(x_1, y_1), q_2(x_2, y_2), \dots, q_{M-1}(x_{M-1}, y_{M-1})$ 。如果中间航点是已知的，则可以带来  $M-1$  个航点的位置约束：

$$q_i = \begin{bmatrix} x(T_i) & y(T_i) \end{bmatrix} = \beta(T_i)c_i \quad (3-11)$$

现在我们将中间航点位置约束和连续性约束写在一起：

$$\begin{aligned} \begin{bmatrix} \beta(T_i) & 0 \\ G(T_i) & -G(0) \end{bmatrix} \cdot \begin{bmatrix} c_i \\ c_{i+1} \end{bmatrix} &= \begin{bmatrix} x(T_i) & y(T_i) \\ 0 & 0 \end{bmatrix} \\ \implies \begin{bmatrix} E_i, F_i \end{bmatrix} \cdot \begin{bmatrix} c_i \\ c_{i+1} \end{bmatrix} &= \begin{bmatrix} q_i \\ 0 \end{bmatrix} \end{aligned} \quad (3-12)$$

综合起点、终点和航点的约束我们可以得到  $6M$  个约束方程：

$$\begin{bmatrix} F_0 & 0 & 0 & \cdots & 0 \\ E_1 & F_1 & 0 & \cdots & 0 \\ 0 & E_2 & F_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & F_{M-1} \\ 0 & 0 & 0 & \cdots & E_M \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{M-1} \\ c_M \end{bmatrix} = \begin{bmatrix} b_0 \\ q_1 \\ 0 \\ \vdots \\ 0 \\ b_M \end{bmatrix} \quad (3-13)$$

$$\implies M_{2Ms \times 2Ms} \mathbf{c} = \mathbf{b}$$

系数矩阵我们表示为  $M$ ，由  $M$  和  $b$  可以唯一确定一组多项式系数  $c$ ，即轨迹的参数形式。

值得一提的是，在  $M$  矩阵中  $T_i$  是第  $i$  段多项式的总时间，它可以提前由一些其他的算法来给出，这样  $M$  就是完全已知的矩阵了。由于我们做的是路径规划，所以我们把  $T_i$  都归一化处理作为常数。

## 3.3 基于安全走廊的安全约束分析

在上一节中我们已经对铰接车的轨迹进行了参数化，这样就自然的处理了微分方程带来的约束问题，保证了轨迹的平滑性。但是在路径规划中还有一个典型的问题

题需要处理，那就是障碍物躲避。无人车辆在非结构或半结构的场景中还需要面临安全性带来的种种挑战，包括避开其他工程机械车辆和环境中的种种障碍物。

无人车的避障和地图的表示形式是分不开的，选择恰当的地图表示形式往往可以让避障算法发挥到很好的性能。常见的地图表示方法有占据栅格地图、点云地图、距离场、特征地图、拓扑地图、场图以及语义地图。规划算法中的避障部分一般根据地图的表示差异而有所不同。对于栅格地图而言，由于其均匀的栅格大小和快速的索引方式，使得它在几乎所有的避障算法中首当其冲，例如 ROS (Robot Operating System) 框架中就实现了基于栅格地图搜索的 A\* 算法和 Dijkstra 算法。点云地图是对空间中障碍物的直接表示，常见的激光 slam 算法建立的地图一般都是这种类型，基于此种地图的规划器一般需要结合 KDTrees(最近邻) 搜索离当前位置最近的障碍物并避开。其次是特征地图，特征地图并不对整个地图空间进行离散，而是在地图上仅仅保留稀疏的特征。这些地图类型在不同的场景下有着不同的优势，针对某种类型的地图往往需要选择行之有效的规划器，减小规划的空间和时间复杂度。

考虑到施工现场的结构不确定，往往地面上会有各种各样的杂物，为了充分利用栅格地图的快速索引能力并提高通用性，我们选择使用 2 维空间下的占据栅格地图作为规划的地图类型。

### 3.3.1 凸多边形约束的数学表达

对于凸包地图区间一般有两种表示方法，一种是把障碍物的轮廓用一个凸多边形近似模拟。另一种是则相反，将地图中安全的区域用凸多边形近似。在本次设计中，我们选择将地图安全区域用凸多边形来近似，这样铰接车的避障问题就转换为了铰接车是否在凸包内。凸多边形的表示方法有很多种类型，例如矩形和不规则凸多边形。在广义上，球体和椭球体也属于凸体。

Gao 等人使用固定方向的长方体安全走廊对轨迹进行优化，但是这种方法固定了安全走廊的拓展方向，在障碍物密集的环境中避障效果会大幅削弱 [1]。Li 等人提出的 STC (Safe Travel Corridors) 算法基于当前路径段的方向进行矩形迭代扩展，生成方式更加灵活 [2]。Deits 等人提出了 IRIS 算法 (Iterative Region Inflation by Semidefinite Programming)，该方法通过不断膨胀椭球的长短半径，求得与障碍物的切平面，进而构造凸多面体 [3]。每次迭代使用最大体积内接椭球作为起始点，然而此过程需要多次求解半正定规划，计算开销较大。Liu 等人提出了 SFC 算法 (Safe Flight Corridors)，

该算法基于线段不断求解包含该线段且不与障碍物相交的最大椭球，并通过椭球与障碍物的切平面构建凸多面体，从而形成安全走廊[4]。

我们使用的空间表示体是不规则凸多边形，它的生成方法参考文献[4]。它相较于矩形的可以把空间填充的更充实，更逼近于障碍物，从而带来更大的解空间。下图是矩形走廊和不规则凸包走廊的对比图。

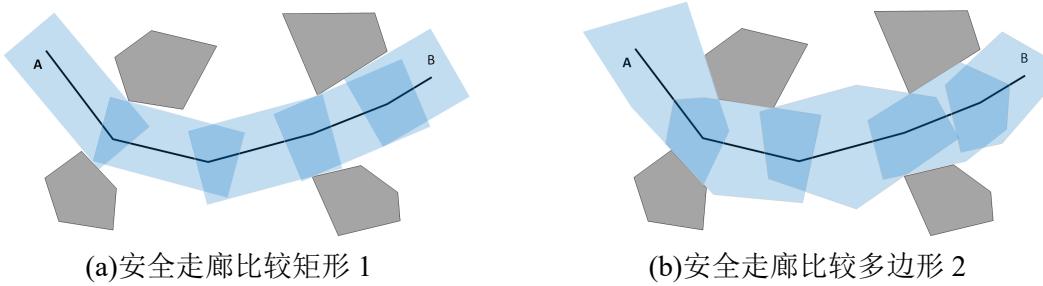


图 3-3 安全走廊的比较

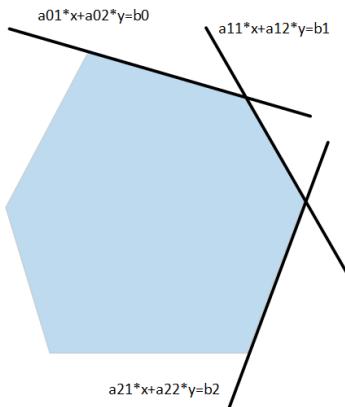


图 3-4 凸包的数学表达关系

凸多边形的区域由一组半空间交集组成。对应 2D 场景凸包就是由一组直线围成的。如图3-4所示。因此，在封闭区域内的点必然满足以下不等式约束：

$$H_z = \{p \mid Ap < b\}, \quad (3-14)$$

$$A = [a_0, \dots, a_z, \dots a_{n_z-1}]_{(n_z-1) \times 2}^T, \quad b = [b_0, \dots, b_z, \dots, b_{n_z-1}]_{(n_z-1) \times 1}^T.$$

其中  $a_i$  是第  $i$  个半空间的法向量， $a_i$  和  $b_i$  唯一去确定一个超平面约束。对于任意在多边形内的点应当满足公式3-14带来的不等式约束。多边形的生成方法参考文献 4。

### 3.4 单阶段无约束轨迹规划

在 3.2 节中，我们介绍了 MINCO 曲线的参数化方法。针对全局路径规划并不需要时间维度的信息，我们重点关注路径层面的两个核心约束条件：首先是路径曲率必须满足由铰接车最大转向角推导出的最小曲率，其次是路径须满足避障要求。关于避障约束的实现细节，已在 3.3 节通过安全走廊构造方法进行了系统阐述。

#### 3.4.1 铰接车的最大曲率

常见的路径规划算法得到的路径是有拐角的，对于具有转向半径约束的车辆来说并不能完全的跟踪。忽略动态转向带来的曲率影响，铰接式车辆的最大曲率由静态转向模型确定。因此在进行轨迹规划时必须考虑最大曲率的约束条件：

$$k_{max} = \frac{1}{R_{max}} = \frac{\tan(\gamma_{max}/2)}{L} \quad (3-15)$$

本次设计中我们的最大转向角度为 30 度，轴距  $L$  为 1.3m，即最大转向半径为 4.85m，最大曲率为 0.2。曲线曲率的计算公式为：

$$k = (\dot{x}\ddot{y} - \dot{y}\ddot{x}) / (\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}} \quad (3-16)$$

令  $p = [x, y]^T$ ，则：

$$k = \frac{\ddot{p}^T B \dot{p}}{\|p\|_2^3}, B = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (3-17)$$

$$\frac{\partial k}{\partial \dot{p}} = \frac{B^T \ddot{p}}{\|p\|_2^3} - 3 \frac{\ddot{p}^T B \dot{p}}{\|p\|_2^5} \dot{p}, \frac{\partial k}{\partial \ddot{p}} = \frac{B \dot{p}}{\|p\|_2^3} \quad (3-18)$$

#### 3.4.2 优化问题的建立

根据公式3-13，轨迹的参数可以由两个要素唯一确定：航点向量  $\mathbf{b}$  与系数矩阵  $M$ 。由于这两个参数能够唯一表征整条轨迹，在构建优化问题时，将航点向量  $\mathbf{b}$  直接做为优化变量是很自然的选择。我们把曲率和安全约束离散化的施加在问题中，当

离散化的点足够多时轨迹就是足够安全的。于是可以得到以下优化问题的表达形式：

$$\begin{aligned}
 & \min_{q_1, q_2, \dots, q_{M-1}} J(q) \\
 & s.t. -k_m - k_{ij} \leq 0 \\
 & \quad k_{ij} - k_m \leq 0 \\
 & \quad A_i p_{ij} - b_i < 0, \quad i \in [1, N], j \in [1, m_i]
 \end{aligned} \tag{3-19}$$

其中  $c$  轨迹多项式是系数矩阵， $k_{ij}$  的计算方法见公式3-17。如图  $k_m$  是最大曲率， $N$  是轨迹的段数， $m_i$  是第  $i$  段轨迹的离散点数。 $p_{ij} = \beta^T \left( \frac{T_i}{m_i} \cdot j \right) \cdot c_i$ ，是第  $i$  段多项式轨迹的第  $j$  个点， $c_i$  通过  $\mathbf{c} = M^{-1}\mathbf{b}$  计算得到。针对于曲率和安全这两个不等式约束，我们引入一个松弛函数将它们作为软约束加入到代价中，松弛函数为：

$$S(x) = \begin{cases} 0 & x \leq 0, \\ -\frac{1}{2a^3}x^4 + \frac{1}{a^2}x^3 & 0 < x \leq a \\ x - \frac{a}{2} & a < x. \end{cases} \tag{3-20}$$

$$S'(x) = \begin{cases} 0 & x \leq 0, \\ -\frac{2}{a^3}x^3 + \frac{3}{a^2}x^2 & 0 < x \leq a \\ 1 & a < x. \end{cases}$$

其中  $a$  是一个已知参数，我们一般选取为优化问题迭代的停止条件值。如图3-5，是该松弛函数图。

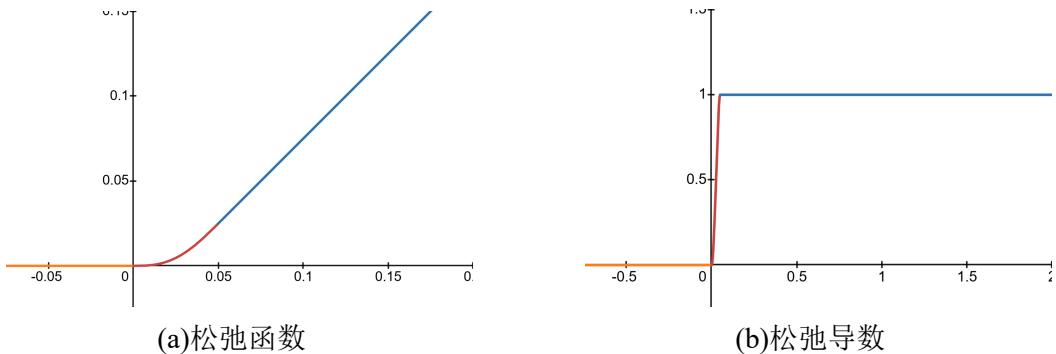


图 3-5 松弛函数

在本课题实验中，我们设置的  $a$  为 0.05。这个松弛函数在  $x$  的负半轴的值为 0，也就意味着不等式约束的代价为 0。在  $0 < x \leq a$  时，代价以一个四次函数上升，越靠近原点代价越小。当  $x > a$  时代价值以斜率为 1 直线上升。另外这个分段函数还有

非常好的一个特性，在定义域上是  $C_2$  微分同胚的。这就意味着它的梯度和 Hessian 矩阵是完全连续的，在优化时我们不必考虑额外的光滑函数。综上，问题被转化成了一个无约束的优化问题：

$$\min_{q_1, q_2, \dots, q_{M-1}} J(q) = \sum_{i=1}^N \sum_{j=1}^{m_i} w_k \cdot S(-k_m - k_{ij}) + w_k \cdot S(-k_m + k_{ij}) + w_c \cdot S(A_i p_{ij} - b_i) \quad (3-21)$$

利用矩阵求导的链式法则我们可以求得各个部分的梯度信息，需要指出的是我们矩阵求导遵循分母布局。

$$\begin{aligned} \frac{\partial S(-k_m - k_{ij})}{\partial c_i} &= -\left(\frac{\partial \dot{p}_{ij}}{\partial c_i} \frac{\partial k_{ij}}{\partial \dot{p}_{ij}} + \frac{\partial \ddot{p}_{ij}}{\partial c_i} \frac{\partial k_{ij}}{\partial \ddot{p}_{ij}}\right) S'(-k_m - k_{ij}), \\ \frac{\partial S(-k_m + k_{ij})}{\partial c_i} &= \left(\frac{\partial \dot{p}_{ij}}{\partial c_i} \frac{\partial k_{ij}}{\partial \dot{p}_{ij}} + \frac{\partial \ddot{p}_{ij}}{\partial c_i} \frac{\partial k_{ij}}{\partial \ddot{p}_{ij}}\right) S'(-k_m + k_{ij}) \\ \frac{\partial S(A_i p_{ij} - b_i)}{\partial c_i} &= \frac{\partial p_{ij}}{\partial c_i} A_i^T S'(A_i p_{ij} - b_i) \end{aligned} \quad (3-22)$$

我们利用上式求出了  $J$  关于  $c$  的梯度，于是由  $b = Mc$ ，可以推导出： $\frac{\partial b}{\partial q} M^{-T} = \frac{\partial c}{\partial q}$ ，因此  $J$  关于  $q$  的梯度为：

$$\frac{\partial J}{\partial q} = \frac{\partial c}{\partial q} \frac{\partial J}{\partial c} \quad (3-23)$$

### 3.4.3 起点和终点航向角保证

我们使用 Minco 样条对轨迹的 x 和 y 方向分别规划，然后再合成一个轨迹。同样对于给定起点  $start(x_s, y_s, v_s, a_s)$  和终点  $end(x_e, y_e, v_e, a_e)$ ，需要计算出 x 和 y 两个方向上的初始状态分量。对应为起点  $q_0 = [x_0, y_0, v_{0x}, v_{0y}, a_{0x}, a_{0y}]$  和终点  $q_M = [x_M, y_M, v_{Mx}, v_{My}, a_{Mx}, a_{My}]$ 。其中，加速度的分解值我们令其为 0，这在路径规划中是合理的。但是对于速度的分解我们还需要详细讨论，因为其涉及到航向角  $\theta$ 。

由运动学模型我们可以求得两个方向上的速度：

$$\begin{aligned} v_{0x} &= v_s \cos \theta_s \\ v_{0y} &= v_s \sin \theta_s \end{aligned} \quad (3-24)$$

对于初始速度  $v_s$  不为零的情况， $v_{0x}, v_{0y}$  都不等于零，我们带入优化问题就可以直接保证车辆初始朝向和轨迹的起点切向在一条直线上。但是当  $v_s = 0$  时，得到  $v_{0x} =$

$v_{0y} = 0$ 。此时用平坦模型计算  $\theta_s$ :

$$\theta_s = \text{atan2}(v_{sy}, v_{sx}) = \text{atan2}(0, 0) \quad (3-25)$$

这时发现其实当  $v_s = 0$  时轨迹曲线的切线方向是没有定义的，这是一个奇异点。为了处理这种奇异情况，我们给当  $v_s$  加上一个微小正量保证它的有效性：

$$\begin{aligned} v_{sx} &= (v_s + \epsilon) \cos \theta_s \\ v_{sy} &= (v_s + \epsilon) \sin \theta_s, \quad 0 < \epsilon < 1e-5. \end{aligned} \quad (3-26)$$

这样就可以保证车辆的起点朝向施加在轨迹上，同样由于  $\epsilon$  是一个极小量，它几乎不会影响轨迹起点的初始速度。对于终点的处理方法和起点一样。

#### 3.4.4 优化问题求解

对于如式3-21的无约束优化问题，有很多成熟的算法可以求解，例如最速下降法、牛顿法、共轭梯度法等。一般的优化算法总体上可以分为两类，一类是基于一阶信息的梯度法，另一类是基于二阶信息的牛顿法。牛顿法除了需要提供代价函数的梯度信息以外，还需要提供代价函数的 Hessian 矩阵。对于复杂优化问题而言解析计算 Hessian 是非常困难的。

LBFGS 算法作为一种高效的拟牛顿优化方法，在优化过程中会不断地迭代逼近出一个 Hessian 矩阵，省去了手动推导的难度。在大规模无约束优化问题中有显著的优势。其核心优势在于内存效率和计算效率，仅存储最近的迭代信息，有效处理高维问题，同时快速收敛于最优解。综合众多因素我们选择使用 LBFGS 优化算法最为本优化问题的优化求解算法。下面

第一，初始解的构造。由于使用优化的方法生成轨迹，所以我们需要一个快速生成航点的路径规划算法，为我们的算法提供软起动初值。我们不关心这个路径规划算法是否最优，但是它需要以较低的时间复杂度运行，因为我们的优化过程可以带来足够的最优性。我们推荐使用 RRTConnect 或者 JPS 算法，这两者都具有快速生成路径的能力。在本课题中使用 RRTConnect 生成路径。区别于传统的 RRT 算法，RRTConnect 同时从起点和终点搜索路径，可以显著提高搜索的效率，减少搜索的次数。我们使用 RRTConnect 算法搜索出从起点到终点的一条路径  $\hat{path}$ ，由此可以构

造初始解，其中包含  $M + 1$  个路径点。

$$\hat{path} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \dots \\ x_M & y_M \end{bmatrix} \longrightarrow q_{init}^- = \begin{bmatrix} q_1^T \\ q_2^T \\ \dots \\ q_{M-1}^T \end{bmatrix} \quad (3-27)$$

第二，构造系数矩阵  $M$ 。我们归一化  $T_i$ ，则  $M$  矩阵是完全已知的。对于系  $\mathbf{b}$  中的  $b_0$  和  $b_M$  可以通过起点  $q_0$  和终点  $q_M$  构建。

第三，构造安全走廊。沿着路径  $\hat{path}$  膨胀出安全走廊。

第四，构建代价函数、构建梯度函数，送入求解器求解得到最优解  $q_{opt}$ 。

第五，获取最优路径参数  $\mathbf{c}$ 。通过  $q_{opt}$  构建  $\mathbf{b}$ ，再通过  $\mathbf{c} = M^{-1}\mathbf{b}$  得到系数  $\mathbf{c}$ 。

第六，得到路径最优化表达式。 $path(t_i) = \beta(t_i)c_i$ ，其中  $t_i$  是第  $i$  条轨迹多项式中的时刻， $0 \leq t_i \leq 1$ 。通过这个公式我们采样得到我们需要的轨迹。

下面给出我们算法的伪代码：

---

**Algorithm 1** Global Planning

**Require:**  $\mathbf{p}_{start}, \mathbf{p}_{end}, \text{map}$

**Ensure:**  $\mathbf{path}$

- 1:  $\hat{path}, q_{init} \leftarrow RRTConnect(\mathbf{p}_{start}, \mathbf{p}_{end}, \text{map})$
  - 2:  $corridor \leftarrow getCorridor(\text{map}, \hat{path})$
  - 3:  $cons \leftarrow getConstraint(M, \kappa_{min,max}, cors)$
  - 4:  $costFcn \leftarrow getCost(cons)$
  - 5:  $grad \leftarrow getGrad(cons)$
  - 6:  $q_{opt} \leftarrow \text{lbfgs}(q_{init}, costFcn, grad)$
  - 7:  $\mathbf{c} = M^{-1}\mathbf{b}, \mathbf{b} \leftarrow q_{opt}$
  - 8: **return**  $path(t_i) = \beta(t_i)c_i$
- 

### 3.5 多阶段全局轨迹规划

在上一节中我们描述了完整的算法结构和细节，可以完成一个点对点的路径规划问题。对于一些更加复杂的场景，例如码头搬运、舱体清卸、矿山运料等多种复杂的工况下，铰接车需要依次去往各个区域规定完成卸货。这样的场景往往需要面向任务设计一个框架。因为这些场景需要更灵活的运动模式，例如倒车。这一节我们将针对码头搬运场景设计一个周期性的规划流程。

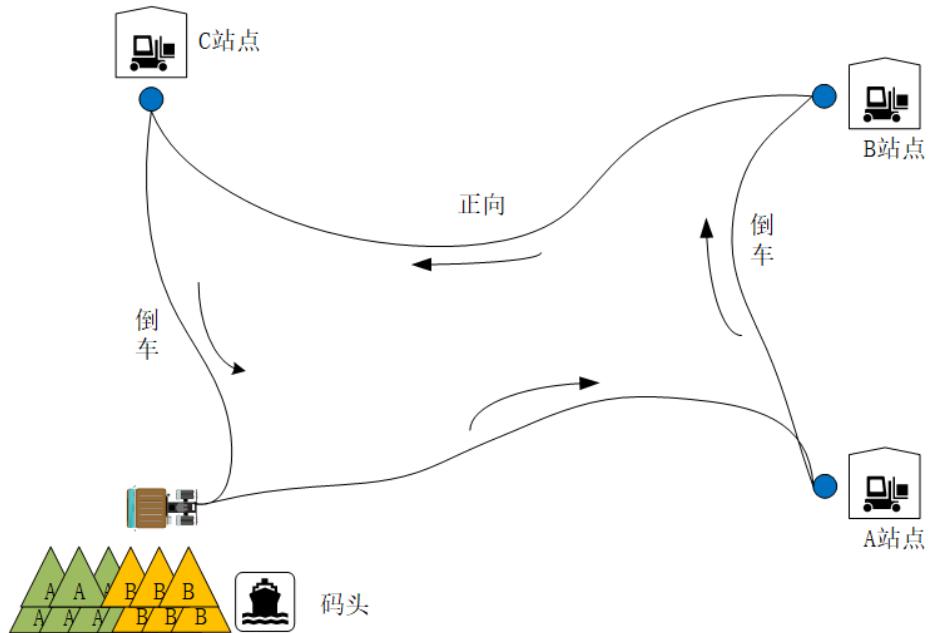


图 3-6 全局规划

### 3.5.1 多阶段规划的问题描述

考虑3-6的搬运问题，我们需要铰接车从码头出发，装货完成去往A点进行卸货，然后去往B点，再去C点，最后回到码头重新装货。这个搬运周期一共需要四次规划，且搬运过程中由于环境等因素规划过程需要倒车。在整个规划周期中铰接车一共需要经过四个站点，分别为码头、A站点、B站点和C站点。在整个过程中受到路径受到两个方面的限制。第一，每一段轨迹的起点的切线方向必须和铰接车保持一致，这就是车辆的非完整约束，这在上一节中已经处理。第二，如果在站点处需要求车辆发生换向，则换向后的轨迹的航向和换向前应该相反。

### 3.5.2 站点的换向处理

在中间站点处的是否换向包含两种情况，一种是站点前的车辆运动方向和站点后运动方向相同，则不需要换向。另一种是站点前的车辆运动方向和站点后运动方向相反，则需要换向，此站点也被称为换向点。图3-7是表示的是车正向行驶穿越站点的情况，经过站点时不用倒车。我们在规划AB段路径时B点作为终点，在规划BC段路径时B点作为起点车辆的运动方向和车辆的朝向相同。图3-8是表示的是车正向行驶到达站点，再反向倒车。与图3-7不同的是，在规划BC段路径时B点作为

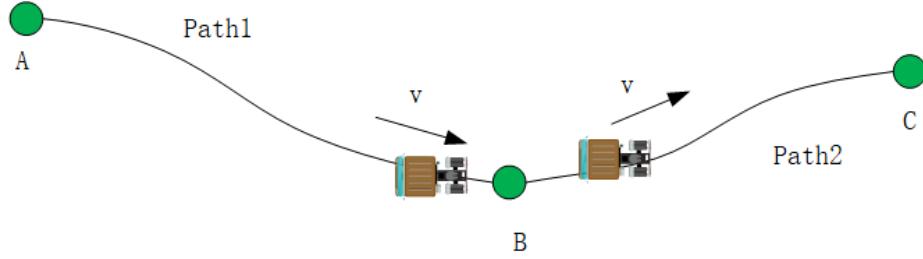


图 3-7 换向 1

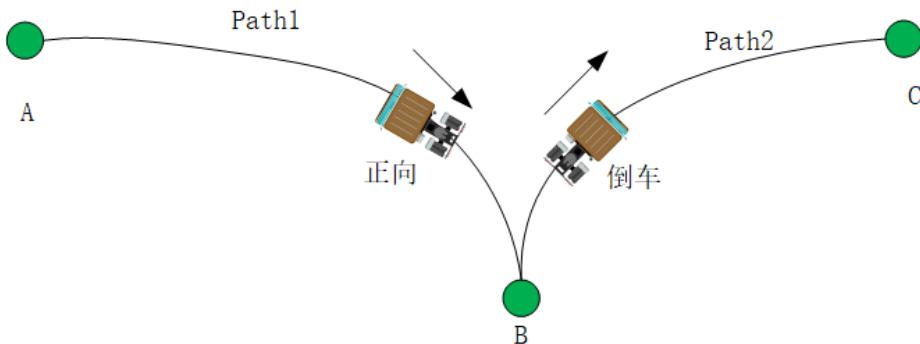


图 3-8 换向 2

起点车辆的运动方向和车辆的朝向相反，而 AB 段相同，因此规划 BC 段时 B 点作为起点需要换向。

### 3.6 本章总结

本章针对无人铰接车辆的轨迹规划难题，提出了一种基于微分平坦理论与 Minco 样条的全局优化方法，通过多维度约束融合与高效求解策略，实现了符合运动学特性、安全避障与实时性的规划框架。首先基于铰接车简化运动学模型，选取笛卡尔坐标  $(x, y)$  作为微分平坦输出，推导出包含速度、航向角、铰接角及曲率的显式表达式，将非完整约束转化为平坦空间代数关系。其次，采用 Minco 样条对轨迹进行参数化建模，通过建立包含起点/终点状态、中间航点位置及高阶导数连续性的 BIVP 约束体系，得出基于航点向量的闭式解形式。针对避障需求，设计基于 SFC 算法的动态凸多边形安全走廊生成机制，通过凸多边形约束实现障碍物规避。进一步提出曲率约束松弛函数，将铰接角物理极限映射为轨迹曲率硬约束，构建包含安全走廊与运动学约束的非线性优化模型，利用 LBFGS 算法实现高效求解。在下一章中我们将针对精细化的避障，介绍局部规划方法。

## 第4章 基于安全走廊的无人铰接车局部规划方法

在这一章中我们提出一种基于安全走廊的时空联合轨迹规划方法，这种方法采用最优控制（Optimal Control）的框架对问题建模。第一节中我们将对铰接车外观做一个通用性的建模，可以有效加强碰撞检测的速度。第二节中介绍针对于铰接车改进的 Hybrid A\* 算法，它将用于我们轨迹规划算法的热启动。第三节介绍规划问题中涉及到的所有约束并进行处理。第四节我们对优化问题进行求解。

### 4.1 铰接车的一般性结构建模

为了保持优化问题的凸性，将车辆底盘形状建模为两个矩形的连接，从而将装载机的避障问题转化为多边形几何碰撞问题。车辆形状分为前轴和后轴两部分，共由九个顶点 A,B,C,⋯,H 和 O<sub>1</sub>。这些顶点在车辆车身坐标系中的表示如图4-1。

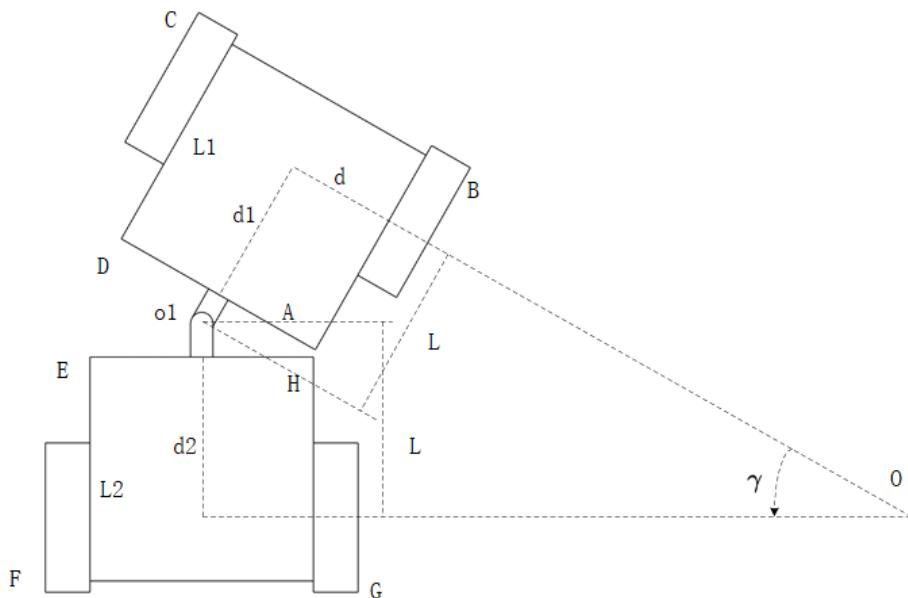


图 4-1 ppi

与动力学模型相一致，我们选取前后中点作为参考点  $(x_f, y_f)$ ，沿着车体方向作为

$x$  轴正方向。则 A,B,C, $\cdots$ ,H 和 O<sub>1</sub> 的相对坐标可以表示为:

$$\begin{aligned} l_A^T &= (-d_1, -width/2), l_B^T = (L_1 - L, -width/2), \\ l_C^T &= (L_1 - L, width/2), l_D^T = (-d_1, width/2), \\ l_E^T &= (d_2 - L, width/2), l_F^T = (-L_2, width/2), \\ l_G^T &= (-L_2, -width/2), l_H^T = (d_2 - L, -width/2), \\ l_{O_1}^T &= (-L, 0). \end{aligned} \quad (4-1)$$

其中,  $l_A, l_B, l_C, l_D, l_{O_1}$  表示相对于底盘前桥坐标系下的坐标。在世界坐标系中, 坐标可以表示为旋转矩阵  $R_\theta$  和平移向量  $\sigma$  的组合。因此, 根据4-1, 前桥的顶点集表示如下:

$$\varepsilon_f = \{p_f \in \mathbb{R}^2 \mid p_f = R_\theta l_f + \sigma, f = A, B, C, D\}, \quad (4-2)$$

在这里  $\sigma$  表示前轴参考点坐标, 表示为  $\sigma = (x_f, y_f)$ 。

$l_E, l_F, l_G, l_H$  是相对于铰接点坐标系下的坐标。在从世界坐标系到铰接点坐标系的变换中, 还涉及到车身坐标系, 需要应用旋转矩阵  $R_\theta$  和  $R_\gamma$ 。平移向量由世界坐标系中的铰接点给出, 记为  $\sigma_1$ 。我们可以得到后桥的坐标变换为:

$$\begin{aligned} p_r &= R_\gamma R_\theta l_b + \sigma_1 \\ \sigma_1 &= R_\theta l_{O_1} + \sigma \end{aligned} \quad (4-3)$$

则后桥的顶点集表示如下:

$$\varepsilon_r = \{p_r \in \mathbb{R}^2 \mid p_r = R_\gamma R_\theta l_r + R_\theta l_{O_1} + \sigma, r = E, F, G, H\}, \quad (4-4)$$

## 4.2 铰接车 Hybrid A\* 算法的创新改进

在大规模优化问题中, 影响求解质量的主要因素往往不是问题或约束的非线性特性, 而是问题是否具备凸性。遗憾的是, 大多数非线性规划问题都不具备凸性, 只有少数问题能够通过对偶方法、变换等手段转化为凸优化问题。直接求解这类问题时, 常常会陷入局部最优解, 难以保证找到全局最优解。然而, 对于这些非凸问题, 我们依然可以采用其他思路, 找到最优解或一个较为理想的解。关键在于, 尽管问题本身无法严格保证非凸性, 我们仍然可以使用一些全局最优算法, 通过寻找一个接近最优的初值来确保其位于全局最优解所在的凸空间内。接下来, 优化算法可以

沿着这个初值在局部凸空间内展开迭代，最终找到最优解。例如，在百度 Apollo 的轨迹规划中，轨迹优化基元采用了多项式形式，通过优化控制点来生成满足要求的轨迹。在此过程中，控制点的初值是通过动态规划搜索得到的。动态规划的作用在于，通过对状态空间的逐步分解，找到一个合理的凸空间，使得优化算法能够高效地进行局部最优解的搜索。Hybrid A\* 算法首次应用于 2007 年 DAPPA 城市挑战赛，并由 Dolgov 等人提出。与传统的 A\* 算法不同，Hybrid A\* 算法在三维空间中进行搜索，且在搜索过程中，邻居节点的拓展是通过动力学方程的前向推演来实现的。然而，传统的 Hybrid A\* 算法是基于  $x - y - \theta$  三维空间进行搜索的，这使得它并不适用于铰接车模型。铰接车由于其具有较为复杂的非线性运动特性，因此直接应用传统的 Hybrid A\* 算法会面临难以满足约束条件和路径质量不高的问题。为了使 Hybrid A\* 算法能够适应铰接车的路径规划问题，本节将提出一种创新性的改进方法。具体来说，我们将修改 Hybrid A\* 算法中的邻居节点拓展策略，结合铰接车的运动学特性，改进算法的搜索策略，以便更好地在铰接车的状态空间中找到一个次优的初解。该初解将在后续的优化过程中作为起点，进一步引导算法沿着合适的方向优化，最终得到一个更加符合实际需求的路径。通过这种方式，我们能够克服传统 Hybrid A\* 算法的局限性，为铰接车路径规划提供更为高效和精准的解决方案。

#### 4.2.1 节点拓展策略改进

铰接车的位形（configuration）传统汽车有显著差异，其位形不仅依赖于车辆的位置和方向  $(x, y, \theta)$ ，还受到铰接角  $\gamma$  的影响。假设铰接车当前的状态为  $x_0(x, y, \theta, \gamma)$ ，一般的 hybrid 算法将控制区间  $[-v_{max}, v_{max}]$  和  $[-\omega_{max}, \omega_{max}]$  进行离散化。并从当前状态  $x_0$  开始以输入采样值  $[v, \omega]$  在  $\Delta_t$  时间内进行运动学推演得到下一个状态，运动学推演见公式2-3，推演过程如图4-2a。图4-2a只简单列出了速度为 0.1 和 1 的情况。可以看出这种采样方式在  $|v|$  很小时，不管  $\omega$  为多少，所积分出来的轨迹都会聚在一起。红色轨迹有短又聚合，并没有充分的去探索周围空间。这是很浪费计算资源且没有必要的。第二，由于系统的输入是二维  $(v, \omega)$ ，所以算法的时间复杂度是  $O(N^2)$ 。下面我将针对节点采样进行改进，可以有效解决采样出来的轨迹聚合的问题，并以  $O(N)$  时间复杂度完成探索，某种意义上这也是一种轨迹的剪枝策略。

首先我们为了降低计算的复杂度，我们只使用铰接车的静态转向模型，系统的输入为  $(v, \gamma_u)$ 。我们不再以固定的时间  $\Delta t$  进行采样，我们以固定的行驶路程  $\Delta_s$  采样，

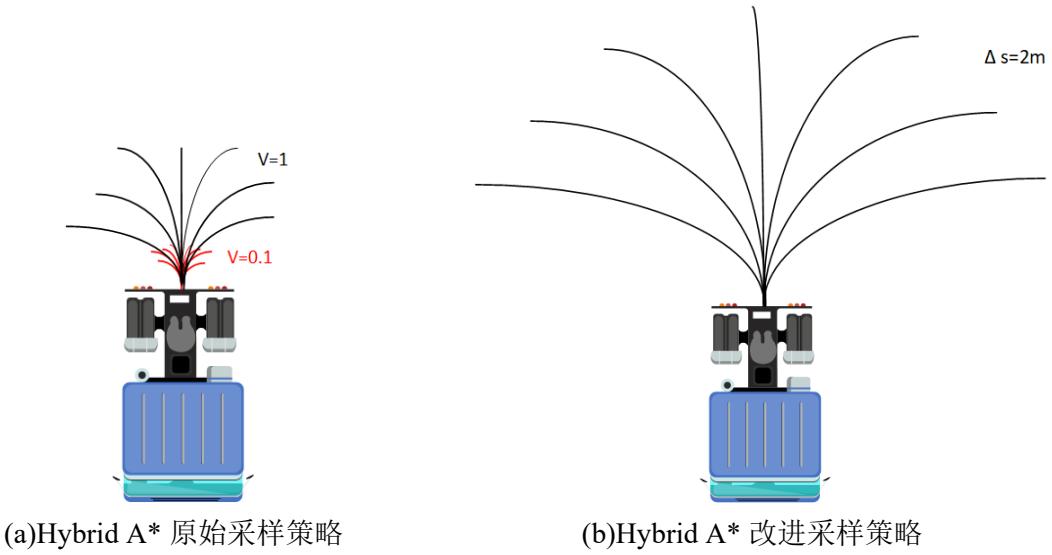


图 4-2 Hybrid A\* 原始采样策略和改进采样策略

于是系统的输入就只有  $\gamma_u$ 。这样做的本质是只在路径几何层面考虑规划问题。运动学推演方程如下：

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \\ \gamma_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta s \cdot \cos\theta_k \\ y_k + \Delta s \cdot \sin\theta_k \\ \theta_k + \frac{\Delta s}{L} \cdot \tan(\gamma_{uk}/2) \\ \gamma_{uk} \end{bmatrix} \quad (4-5)$$

其中  $\Delta s$  是采样的路径步长， $(x, y, theta, gamma)$  是铰接车的位形空间状态变量。 $\Delta s$  等于 2m 的推演过程如图4-2中的右图4-2b。

可以看出相对于原始推演策略来说我们的方式可以大大降低时间复杂度。本质上这其实也是速度  $v$  为固定常数时的特殊情况。此外为了加速搜索，hybrid A\* 在搜索过程中还引入了 RS (Reeds-Shepp 曲线)，进一步提高搜索效率。RS 曲线是一种用于求解二维平面中任意始末位姿之间最短路径的算法。该算法的核心思想是通过分析并归纳所有可能的路径类型，将最短路径的构造问题简化为 48 种不同的情况，并且对每种情况进行了枚举，涵盖了圆弧和直线段的各种排列组合方式。因此，RS 曲线可以为无障碍情况下的最短路径提供解析表达式，并能够快速地构造出符合运动学约束的可行路径。RS 曲线特别适用于涉及到转向半径约束的路径规划问题，尤其是在移动机器人和自动驾驶领域中，其广泛应用于生成车辆的路径规划。该算法通过考虑车辆的运动学约束（如最小转弯半径），能够有效地找到从起点到终点的最短

路径，并确保路径是可行的，符合实际的转向和行驶要求。值得指出的是，RS 曲线在 hybrid A\* 搜索的最后一步具有加速收敛到终点的效果。

### 4.2.2 代价函数设计

Hybrid A\* 算法在进行节点的扩展时，会从采样出来的节点中选出代价最低的一个节点作为下一个扩展节点。Hybrid A\* 算法的代价函数  $f$  分为累积代价  $g$  和启发式函数代价  $h$  两个部分，如公式4-6。其中累积代价  $g$  表征着起点到当前节点的路程代价，启发式代价  $h$  表征当前节点到终点的代价。

$$f = g + h \quad (4-6)$$

值得注意的一般当前点到终点的最小代价是未知的，所以理论上启发式函数  $h$  设计的越靠近最优代价则搜索效果越好。

我们的采样策略的输入变量是  $\gamma_u$ ，由于上一时刻的采样和这一时刻独立，所以可能出现  $\Delta\gamma$  过大的情况，也就是铰接角突变问题，这对于下层控制是不利的。所以我们在代价函数中必须含有  $\Delta\gamma$  项使得铰接角变化率尽量小，我们称之为转向惩罚。因此我们将下一个拓展的累积代价设计为：

$$g_{next} = g_{cur} + (x_{next} - x_{cur})^2 + (y_{next} - y_{cur})^2 + (\gamma_{u(next)} - \gamma_{u(cur)})^2 \quad (4-7)$$

其中  $g_{cur}$  为当前节点的累积代价， $g_{next}$  为即将拓展的邻居节点累积代价。 $x_{cur}, y_{cur}$  为当前节点位置信息， $x_{next}, y_{next}$  为即将拓展的邻居节点的位置信息。 $\gamma_{u(next)}$  为从当前节点拓展到邻居节点的所需要的转向输入， $\gamma_{u(cur)}$  为从上一节点拓展到当前节点的所需要的转向输入。

启发式代价函数  $h$  记录当前节点到终点的估计代价，hybrid A\* 算法的启发式代价一般分为两个部分：非完整约束启发式代价  $h_{nonh}$  和避障启发式代价  $h_{collision}$ 。其中非完整约束代价  $h_{nonh}$  是仅仅在考虑车辆有转弯半径且不考虑障碍物的情况下距离代价，通常可以直接由 RS 曲线闭式算出。避障启发式代价  $h_{collision}$  则是仅考虑

避障时的代价，这里我们使用无穷范数作为避障启发式代价。

$$\begin{aligned}
 h_{nonh} &= RS(x_{cur}, y_{cur}, x_{goal}, y_{goal}) \\
 h_{collision} &= \max(|x_{cur} - x_{goal}|, |y_{cur} - y_{goal}|) \\
 h &= h_{nonh} + h_{collision}
 \end{aligned} \tag{4-8}$$

#### 4.2.3 碰撞检测算法设计

在碰撞检测算法的设计中，我们通过检查车辆几何模型与栅格地图的是否有重叠来判断路径可行性，其核心原理基于车辆位姿变换与离散化障碍物检测。首先，算法利用旋转矩阵将车辆初始轮廓坐标变换到全局坐标系下，计算各顶点坐标后转换为地图栅格索引，以便在栅格地图中查询索引处是否由障碍物。随后，对车辆轮廓边进行线段离散化检测。

如图4-3所示，我们采用 Bresenham 算法遍历每条边的栅格路径，若任一栅格为障碍物或超出地图边界，则判定为碰撞。在路径搜索过程中，该模块嵌入节点扩展和 Reeds-Shepp 曲线连接阶段，逐段验证候选路径的安全性。其优势在于计算效率高且与栅格地图兼容性强，但受限于离散化误差可能忽略细小障碍物，且仅适用于静态环境。图4-3中左图为铰接车没有和障碍物发生碰撞的场景，右图则是发生碰撞的场景，其中红色栅格是车体轮廓和障碍物栅格重叠的区域。

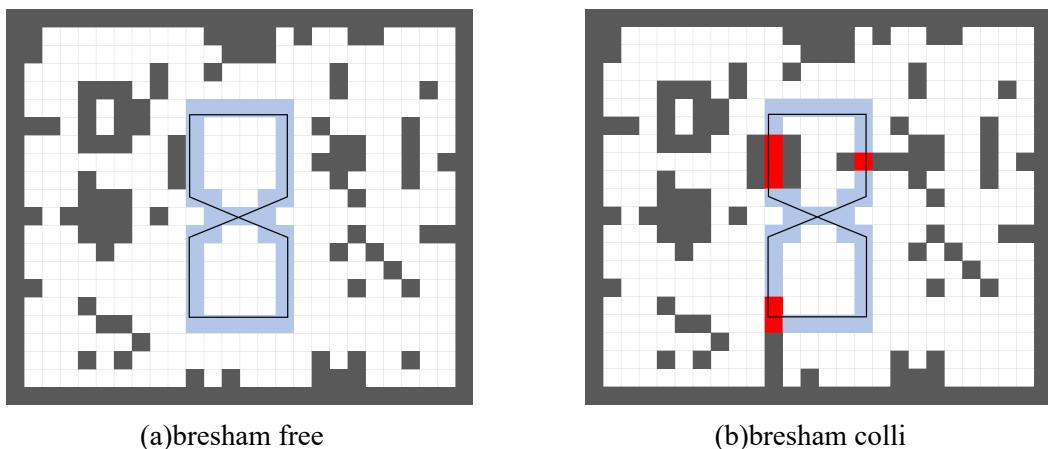


图 4-3 障碍物检测图

## 4.3 基于 NMPC 和安全走廊的轨迹规划

在 4.2 节中我们针对铰接车对 hybrid A\* 算法进行了改进，它将作为我们轨迹规划的热启动算法。本节中将介绍如何构建一个基于 NMPC 的轨迹规划算法，这是一个典型的 OCP 最优控制框架。

### 4.3.1 NMPC 问题构建

铰接式车辆运动规划的最优控制问题可以表示为其一般形式，包括成本函数以及多个等式或不等式约束。与典型的优化问题不同，基于最优控制的轨迹规划还受到一组动态微分方程的约束。一般能耗和时间最优的代价函数定义如下：

$$\begin{aligned} J(x(t), u(t)) &= \int_{t_0}^{t_f} \|u(t)\|^2 dt + w_t * \|t_f - t_0\|^2, \\ \|u(t)\|^2 &= w_{uj} * \|jerk(t)\|^2 + w_{uw} * \|\omega\|^2. \end{aligned} \quad (4-9)$$

其中  $w_t$ 、 $w_{uj}$  和  $w_{uw}$  分别为时间代价权重、舒适度代价权重和铰接角能耗代价权重。 $\|u(t)\|^2$  用作输入能量消耗的度量，而  $\|t_f - t_0\|^2$  确保时间持续时间的最优性，其中  $t_f$  和  $t_0$  分别为结束时间和开始时间。然后这个轨迹优化问题表示为：

$$\begin{aligned} [x^*(t), u^*(t)] &= \text{minimize } J(x(t), u(t)), \\ s.t. \dot{x}(t) - f(x(t), u(t)) &= 0, \\ x_l \leq x(t) \leq x_u, u_l \leq u(t) &\leq u_u, t \in [0, t_f], \\ x(t_0) = x_{start}, x(t_f) &= x_{goal}. \end{aligned} \quad (4-10)$$

在公式4-10的约束中， $\dot{x}(t) - f(x(t), u(t)) = 0$  表示运动方程约束，保证优化问题中轨迹的平滑性。变量  $x_l$  和  $x_u$  表示状态变量的下界和上界， $u_l$  和  $u_u$  表示控制变量的下界和上界。此外， $x_{start}$  和  $x_{goal}$  分别对应规划问题的起点状态和终点状态。对于如上 NMPC 形式的最优控制问题，处于问题的复杂度，我们并不在连续空间下进行求解。我们将它的状态方程进行离散化，求解它的最有离散子序列，即演变成一个 NLP

(Nonlinear Programming) 问题。我们定义如下形式的解向量形式:

$$\begin{aligned}\xi &= [\xi_x^T, \xi_y^T, \xi_\theta^T, \xi_\gamma^T, \xi_v^T, \xi_a^T, \xi_\omega^T, \xi_{jerk}^T, \xi_{\Delta t}^T]^T, \\ \xi_x &= [x(1), \dots, x(N-2)]^T, \\ \xi_\omega &= [\omega(0), \dots, \omega(N-2)]^T.\end{aligned}\quad (4-11)$$

对于轨迹上的  $N$  个点, 起点  $P_0$  和终点  $P_{N-1}$  是已知的。 $\xi_x$  的维度为  $n = N - 2$ ,  $\xi_y, \xi_\theta, \xi_\gamma, \xi_v, \xi_a$  一样。 $\xi_\omega$  的维度为  $m = N - 1$ ,  $\xi_{jerk}, \xi_{\Delta t}$  一样。

以上所构建的优化问题中所受到的约束有状态方程约束、状态受限、输入受限、起点约束和终点约束, 并没有包含避障约束。沿用安全走廊的方法, 和第三章中公式3-14一样, 避障约束是不等式的形式。我们假设  $g_f$  为铰接车前桥的不等式约束函数,  $g_r$  为铰接车后桥的不等式约束函数。则最终的轨迹规划问题可以表示为:

$$\begin{aligned}J(\xi) &= w_{uj} * \|\xi_{jerk}\|_2^2 + w_{uw} * \|\xi_\omega\|_2^2 + w_t * I_{N-1}^T \xi_{\Delta t}, \\ \text{s.t. } g_{kin}(k) &= 0, 1 \leq k \leq N-1, \\ g_f(k) &< 0, g_r(k) < 0, 1 \leq k \leq N-2, \\ \hat{\xi}_{min} &\leq \hat{\xi} \leq \hat{\xi}_{max}, \xi_{\Delta t min} < \xi_{\Delta t} < \xi_{\Delta t max}.\end{aligned}\quad (4-12)$$

其中  $\hat{\xi} = [\xi_\gamma^T, \xi_v^T, \xi_a^T, \xi_\omega^T, \xi_{jerk}^T]^T$ ,  $g_{kin}$  是状态方程约束。方程约束中未提到的变量都是无约束变量。

### 4.3.2 运动学约束

在第二章中我们介绍了铰接车的运动学模型, 模型的状态量为  $x, y, \theta, \gamma$ , 受到微分方程的约束这些变量是天然连续的。系统的输入为速度  $v$  和铰接角速度  $\omega$ , 他们并没有受到微分方程的约束, 这意味着  $v$  和  $\omega$  在优化过程中直接作为优化变量是会突变的。这对于下层控制来说是不利的。为了解决这个问题, 我们为系统引入高阶的连续性, 假设系统的输入为  $jerk$  (加加速度或者舒适度) 和  $\omega$ 。模型可以进一步

写为:

$$\dot{x} = f(x, u) \implies \frac{d}{dt} \begin{bmatrix} x_f \\ y_f \\ \theta \\ \gamma \\ v \\ a \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ vtan(\gamma/2)/L + \omega / (\cos \gamma + 1) \\ \omega \\ a \\ jerk \end{bmatrix} \quad (4-13)$$

在这里  $x = [x_f, y_f, \theta, \gamma, v, a]^T$  为状态向量, 分别表示前轴中点位置、航向角、铰接角、速度、加速度。输入向量由  $u = [jerk, \omega]^T$  给出, 其中  $jerk$  为前轴中点加加速度(舒适度),  $\omega$  为铰接角速度。对于上述运动学方程我们对其进行离散化, 采用一阶离散运动学模型, 可表示为:

$$x(k) = x(k-1) + \Delta t(k-1) f(x(k-1), u(k-1)) \quad (4-14)$$

我们令  $g_{kin} = [g_x^T \ g_y^T \ g_\theta^T \ g_\gamma^T \ g_v^T \ g_a^T]^T$ , 其中每个  $g$  对应  $m = N - 1$  个等式约束。例如:

$$\begin{aligned} g_{x_f}(0) &= x_f(1) - x_{start} - \Delta t(0) \cdot v(0) \cos \theta(0), \\ g_{x_f}(1) &= x_f(2) - x_f(1) - \Delta t(1) \cdot v(1) \cos \theta(1), \\ &\dots \\ g_{y_f}(0) &= y_f(1) - x_{start} - \Delta t(0) \cdot v(0) \sin \theta(0), \\ g_{y_f}(1) &= y_f(2) - y_f(1) - \Delta t(1) \cdot v(1) \sin \theta(1), \\ &\dots \end{aligned} \quad (4-15)$$

使用惩罚函数将运动学约束纳入代价函数, 表达式如下:

$$\mathcal{G}_{kin} = w_{kin} * \|g_{kin}\|_2^2 \quad (4-16)$$

梯度为:

$$\frac{\partial \mathcal{G}_{kin}}{\partial \xi} = 2w_{kin} \cdot \frac{\partial g_{kin}^T}{\partial \xi} \cdot g_{kin} \quad (4-17)$$

其中  $\frac{\partial g_{\text{kin}}^T}{\partial \xi}$  表示所有运动约束向量的雅可比矩阵。假设第  $i$  类运动约束关于第  $j$  类变量的雅可比矩阵由  $\psi_{ij} = \frac{\partial g_i^T}{\partial \xi_j}$  给出。我们可以推出：

$$\frac{\partial g_{\text{kin}}^T}{\partial \xi} = \begin{bmatrix} \psi_E & 0 & 0 & 0 & 0 & 0 \\ 0 & \psi_E & 0 & 0 & 0 & 0 \\ -\psi_{x\theta} & -\psi_{y\theta} & \psi_E & 0 & 0 & 0 \\ 0 & 0 & -\psi_{\theta\gamma} & \psi_E & 0 & 0 \\ -\psi_{xv} & -\psi_{yv} & -\psi_{\theta v} & 0 & \psi_E & 0 \\ 0 & 0 & 0 & 0 & -\psi_{va} & \psi_E \\ 0 & 0 & -\psi_{\theta\omega} & -\psi_{\gamma\omega} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\psi_{ajerk} \\ -\psi_{x\Delta t} & -\psi_{y\Delta t} & -\psi_{\theta\Delta t} & -\psi_{\gamma\Delta t} & -\psi_{v\Delta t} & -\psi_{a\Delta t} \end{bmatrix} \quad (4-18)$$

其中  $\psi = [n \times n, 0_n] - [0_n, E_{n \times n}]$ , 其中  $E$  为单位矩阵, 且  $\psi_{ij} = \frac{\partial g_i^T}{\partial \xi_j}$ 。

### 4.3.3 安全走廊的防碰撞约束处理

在本节中, 我们将探讨避障约束在笛卡尔坐标系中的表述方法。我们仍然采用安全走廊模型对铰接车的避障进行全维建模, 安全走廊的构建方法参照第三章中的全局规划方案。与全局规划中的处理方式不同, 本节中我们将对车体结构进行建模。在全局规划中, 车体被简化为质点模型, 且车体结构的影响通过膨胀层来近似替代, 这种方法虽然较为粗略, 但其主要目的是快速生成一条长远的路径, 以为局部规划提供导航指引。而局部规划则负责生成更加精细化的轨迹。

首先, 在平面上沿前端粗解生成一系列凸多边形, 这些凸多边形依次相交形成安全走廊, 所有优化轨迹点都将被约束在该走廊内。安全走廊如图4-4所示。根据凸集的性质, 如果车辆的所有顶点都在凸包内, 那么整个车辆就一定在凸包内。我们把铰接车前桥和后桥共八个顶点全部带入不等式3-14, 可以得到如下安全约束:

$$\begin{cases} A(R_\theta l_f + \sigma) - b < 0, & f = A, B, C, D \\ A(R_\gamma R_\theta l_r + R_\theta l_{O_1} + \sigma) - b < 0, & r = E, F, G, H \end{cases} \quad (4-19)$$

需要注意的是, 所有顶点不一定都需要参与到避障约束中。由于铰接角的运动范围

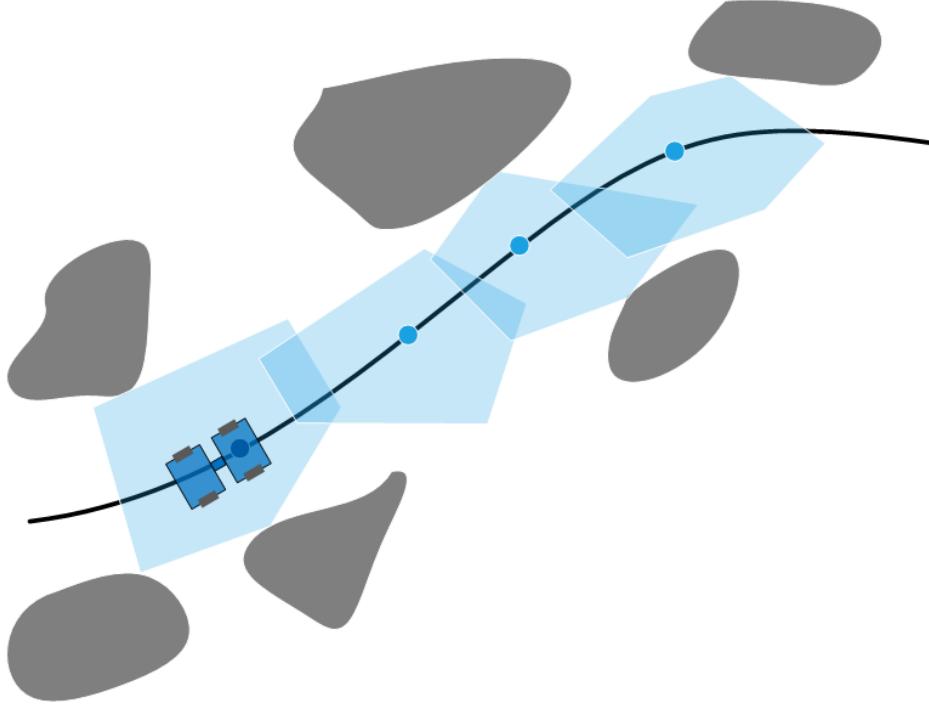


图 4-4 安全走廊

有限，仅利用前轴的四个顶点很多时候就足以实现有效的避障。为了方便表达，我们令

$$\begin{aligned} g_f(\sigma, \theta, \gamma) &= A(R_\theta l_f + \sigma) - b \\ g_b(\sigma, \theta, \gamma) &= A(R_\gamma R_\theta l_r + R_\theta l_{O_1} + \sigma) - b. \end{aligned} \quad (4-20)$$

这里  $R_{\theta+\gamma} = R_\theta R_\gamma$ ,  $R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ 。路径上第  $k$  个点处的走廊约束关于  $\sigma_k, \theta_k$  和  $\gamma_k$  的梯度由下式给出：

$$\begin{aligned} \frac{\partial g_f^T(k)}{\partial \sigma_k} &= A_k^T, \quad \frac{\partial g_f^T(k)}{\partial \gamma_k} = 0, \quad \frac{\partial g_f^T(k)}{\partial \theta_k} = l_f^T \frac{\partial R_\theta^T}{\partial \theta_k} A_k^T, \\ \frac{\partial g_r^T(k)}{\partial \sigma_k} &= A_k^T, \quad \frac{\partial g_r^T(k)}{\partial \gamma_k} = l_r^T \frac{\partial R_{\theta+\gamma}^T}{\partial \gamma_k} A_k^T, \\ \frac{\partial g_r^T(k)}{\partial \theta_k} &= l_r^T \frac{\partial R_{\theta+\gamma}^T}{\partial \theta_k} A_k^T + l_{O_1}^T \frac{\partial R_\theta^T}{\partial \theta_k} A_k^T. \end{aligned} \quad (4-21)$$

对于上述走廊的不等式约束，我们使用罚函数将约束加入到代价函数中。为了确保连续可微性，罚函数依旧使用公式3-20。为了方便表达，我们定义  $[]$  为逐元素运算符，例如  $[ \cdot ]$  表示逐元素乘法， $S[x]$  表示对  $x$  的所有元素进行函数  $S$  运算。于是

可以得到总体的走廊约束代价，其表达为：

$$\mathcal{G}_c = \mathcal{G}_f + \mathcal{G}_r = w_c \cdot \sum_{k=1}^{N-2} \sum_{i=0}^{d-1} (S[g_f(k)] + S[g_r(k)]). \quad (4-22)$$

其中  $d$  为矩阵  $A_k$  的行数。对公式4-22中的各项分别进行求导得到走廊的惩罚项的梯度：

$$\begin{aligned} \frac{\partial \mathcal{G}_{f,r}}{\sigma(k)} &= \sum_{i=0}^{d-1} A_k^T S' [g_{f,r}(k)] = A_k^T S' [g_{f,r}(k)] I_d, \\ \frac{\partial \mathcal{G}_f}{\xi_\theta(k)} &= l_f^T \frac{\partial R_\theta^T(k)}{\partial \theta(k)} A_k^T S' [g_f(k)] I_d, \\ \frac{\partial \mathcal{G}_r}{\xi_\gamma(k)} &= l_r^T \frac{\partial R_{\theta+\gamma}^T(k)}{\partial \gamma(k)} A_k^T S', \\ \frac{\partial \mathcal{G}_r}{\xi_\theta(k)} &= \left( l_r^T \frac{\partial R_{\theta+\gamma}^T(k)}{\partial \theta(k)} + l_{o1}^T \frac{\partial R_\theta^T(k)}{\partial \theta(k)} \right) A_k^T S' [g_r(k)] I_d, \\ \sigma(k) &= [\xi_x(k), \xi_y(k)]^T \cdot [g_r(k)] I_d \end{aligned} \quad (4-23)$$

#### 4.3.4 变量的边界约束

在实际的系统中，系统的输入和状态往往不是无边界的，例如铰接车的速度、加速度、加加速度、铰接角等变量，所以我们要对这些变量进行约束分析。由于边界约束同样和走廊约束一样是不等式约束，所以我们仍然采用走廊约束处理方法，使用松弛函数  $S(\cdot)$  加入到代价函数中。为方便起见，我们省略了下标，变量边界约束对应的罚函数项和其梯度如下：

$$\begin{aligned} \mathcal{G}_b &= w_b \cdot \sum_{k=0}^{cols} \left( S[\hat{\xi}_{min} - \hat{\xi}] + S[\hat{\xi} - \hat{\xi}_{max}] \right) \\ &= w_b \cdot I_{N-2}^T \left( S[\hat{\xi}_{min} - \hat{\xi}] + S[\hat{\xi} - \hat{\xi}_{max}] \right) \\ \frac{\partial \mathcal{G}_b}{\hat{\xi}} &= w_b \cdot \left( S'[\hat{\xi} - \hat{\xi}_{max}] - S'[\hat{\xi}_{min} - \hat{\xi}] \right) \end{aligned} \quad (4-24)$$

#### 4.3.5 时间正则化约束

在我们的轨迹优化问题中，每两个轨迹点之间的时间  $\Delta t$  是灵活变化且独立的，纳入了优化变量当中。它同样和走廊约束一样属于不等式约束，与之不同的是，时间项  $\Delta t$  必须严格大于 0，这是硬约束。使用松弛函数对时间项进行惩罚属于约束软

化的过程，在极端的场景下仍然可能出现  $\Delta t \geq 0$ ，即使在下一步数值迭代过程中  $\Delta t$  又回到小于 0。但是中间出现  $\Delta t \geq 0$  这是反只觉得，不符合物理的规律。所以避免这种情况，我们不能使用松弛函数处理时间正则化约束。为了解决以上问题，我们使用一个 sigmoid 变换将  $\Delta t$  的取值范围区间映射到负无穷到正无穷，变换后的变量记为  $\tau$ 。这个变换是微分同胚的，微分同胚变换前后不会为优化问题引入额外的极小值【文献】。

$$\Delta t(\tau) = \frac{\Delta t_{max}}{1 + e^{-\frac{\tau}{\Delta t_{max}}}}, \quad \Delta t'(\tau) = \frac{e^{-\frac{\tau}{\Delta t_{max}}}}{\left(1 + e^{-\frac{\tau}{\Delta t_{max}}}\right)^2} \quad (4-25)$$

这里  $-\infty < \tau < +\infty$ ,  $0 < \Delta t < \Delta t_{max}$ 。

#### 4.3.6 迭代求解框架

在上面几个小节，利用惩罚函数来表示成本函数中的运动学约束，并使用松弛函数来表示成本函数中的走廊约束，对于时间正则化，我们将时间变量的取值范围转换为整个实数轴。由于时间变量发生了转换，我们不妨重新表示以下优化变量。令最终的优化变量  $\eta = [\xi_x^T, \xi_y^T, \xi_\theta^T, \xi_\gamma^T, \xi_v^T, \xi_a^T, \xi_\omega^T, \xi_{jerk}^T, \xi_\tau^T]^T$ ，在这里  $\xi_\tau$  和最初优化变量  $\xi$  中的  $\xi_t$  是微分同胚的。于是代价函数可以重新写为：

$$\mathcal{J}(\eta) = J(\xi) + \mathcal{G}_{kin} + \mathcal{G}_c + \mathcal{G}_b \quad (4-26)$$

对应的梯度为

$$\frac{\partial \mathcal{J}}{\partial \xi_{jerk,\omega}} = 2w_{uj,uw} \cdot \xi_{jerk,\omega}, \quad \frac{\partial \mathcal{J}}{\partial \xi_\tau} = \Delta t[\xi_\tau] \quad (4-27)$$

在等式约束部分，公式4-22，我们推导出了运动学约束的雅可比矩阵。现在，我们同样需要把对时间变量的导数部分进行替换，替换为

$$\psi_{i\tau} = \frac{\partial g_i^T}{\partial \xi_\tau} = \frac{\partial g_i^T}{\partial \xi_{\Delta t}} \cdot \Delta t[\xi_\tau] \quad (4-28)$$

其中  $\mathcal{G}_{kin}, \mathcal{G}_c, \mathcal{G}_b$  由公式4-16, 4-22、4-24提供。

### 4.4 本章小结

在本章中，我们提出了一种基于安全走廊的无人铰接车局部轨迹规划方法，并通过最优控制框架对其进行建模与求解。首先，我们对铰接车的外观进行了通用性

的建模，并通过简化车辆形状为两个矩形的连接，转化了避障问题为几何碰撞问题。其次，提出了针对铰接车的 Hybrid A\* 算法改进，通过调整节点拓展策略并结合铰接车的运动学特性，优化了路径规划的效率与准确性。我们还设计了适用于铰接车的碰撞检测算法，利用栅格地图与 Bresenham 算法进行高效的障碍物检测。

在轨迹优化方面，我们构建了基于非线性模型预测控制 (NMPC) 的轨迹规划方法，定义了优化问题并加入了动力学约束、运动学约束及避障约束。通过引入适合铰接车的运动学模型和代价函数，确保了在求解过程中能够考虑车辆的舒适度与能效，同时解决了车辆输入突变的问题。下面给出我们算法的伪代码：

---

**Algorithm 2** Local Planning

---

**Require:**  $\mathbf{p}_{\text{start}}, \mathbf{p}_{\text{end}}, \text{map}$

**Ensure:**  $\mathbf{traj}^{\text{opt}}$

- 1:  $\mathbf{traj}^{\text{coarse}} \leftarrow \text{HybridAstar}(\mathbf{p}_{\text{start}}, \mathbf{p}_{\text{end}}, \text{map})$
  - 2:  $\xi_0 \leftarrow \text{getGuess}(\mathbf{traj}^{\text{coarse}})$
  - 3:  $\mathbf{cors} \leftarrow \text{getCorridor}(\text{map}, \mathbf{traj}^{\text{coarse}})$
  - 4:  $\mathbf{cons} \leftarrow \text{getConstraint}(\xi_{\min, \max}, \mathbf{cors}, \text{kinEqn})$
  - 5:  $\text{costFcn} \leftarrow \text{getCost}(\mathbf{cons})$
  - 6:  $\mathbf{grad} \leftarrow \text{getGrad}(\mathbf{cons})$
  - 7: **return**  $\mathbf{traj}^{\text{opt}} \leftarrow \text{lbfsgs}(\text{costFcn}, \mathbf{grad})$
- 

通过本章的研究，我们不仅改进了传统轨迹规划算法以适应铰接车的特殊运动特性，还设计了高效的碰撞检测与轨迹优化方案，为无人铰接车在复杂环境中的局部路径规划提供了有效的解决方案。

## 第5章 无人铰接车运动规划系统仿真实验

在前面章节中，我们全面介绍了全局轨迹规划和局部轨迹规划两个算法，他们组合成一个完整的运动规划系统。在这一章节中我们将把这两个算法集成到自主导航系统中，并在仿真中分别对这两个算法进行详细的分析和验证，随后在一个大的场景中展示我们整个运动规划系统的能力。

我们的实验仿真环境搭载 AMD Ryzen 9 7940HX 处理器，基于 Ubuntu 20.04 LTS 操作系统构建 ROS (Robot Operating System) Noetic 框架。核心算法采用 C++ 标准实现，充分发挥多核 CPU 的并行计算能力，并通过 Catkin 构建系统实现节点编译与依赖管理。表5-1给出我们的仿真车体的参数设置。

表 5-1 System Parameters

Parameter	Description	Value
$L$	Front axle/rear axle length $m$	1.3
$L_1$	Front axle length	1.8
$L_2$	Rear axle length	1.8
$d$	Half wheelbase	1.8
$d_1$	Distance from front axle to A	1.075
$d_2$	Distance from rear axle to H	1.075
$width$	Front/Rear axle width	2.1
$v_{min}$	Velocity lower-bound $m/s$	-3
$v_{max}$	velocity upper-bound	3
$a_{min}$	Acceleration lower-bound $m^2/s$	-2
$a_{max}$	Acceleration upper-bound	2
$jerk_{min}$	Jerk lower-bound $m^3/s$	-3
$jerk_{max}$	Jerk upper-bound	3
$\gamma_{min}$	Articulation angle lower-bound $rad$	-0.52
$\gamma_{max}$	Articulation angle upper-bound	0.52
$\omega_{min}$	Lower bound of articulation angular velocity $rad/s$	-0.2
$\omega_{max}$	Upper bound of articulation angular velocity	0.2

## 5.1 仿真环境介绍

### 5.1.1 ROS 简介

受实验条件限制,本研究采用 ROS (Robot Operating System) 开源框架进行算法仿真验证。ROS 是由斯坦福大学与 Willow Garage 公司联合开发的机器人开发平台,其分布式架构通过模块化设计简化了通信、数据可视化等流程。我们利用 ROS 的可视化工具 RViz 实现运动规划算法的三维动态演示,重点呈现路径几何特征与车体状态变化。需要说明的是,本文仿真仅基于 ROS 的运动学模拟功能(不含物理引擎),未涉及复杂动力学因素,可在保证实验效率的同时验证算法的基础性能。

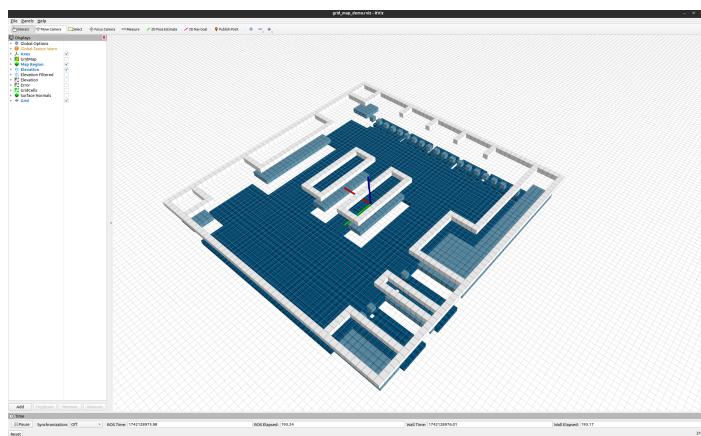


图 5-1 rviz 仿真界面

ROS 的主要特点包括:

- (1) 模块化与高复用性: ROS 采用节点 (Node) 的模块化设计理念,允许开发者将复杂系统拆分为独立的功能模块(如传感器驱动、运动控制、路径规划等),各模块通过松耦合的通信机制(话题、服务、动作)协同工作。此外他还提供了栅格地图、导航框架和坐标变换等实用模块,利于我们快速开发验证算法。
- (2) 分布式计算与跨平台支持: ROS 支持分布式架构,仿真中的感知、决策、控制等模块可部署于不同硬件设备,甚至结合容器化技术(如 Docker)实现跨平台运行。
- (3) 丰富的工具链与仿真生态: RViz(三维可视化工具)、rqt(图形化调试套件)等工具,形成完整的仿真-开发-调试闭环。可实时可视化点云、路径规划结果等数据,显著降低算法验证门槛。
- (4) 标准化通信与接口统一: ROS 定义了基于消息 (Message) 和服务 (Service)

的标准化通信接口，支持多种数据类型（如传感器数据、几何坐标、图像流）的传输。这种一致性使得仿真环境与实体机器人的代码迁移成本极低，确保研究成果可快速向实际系统转化。

(5) 开源社区：ROS 拥有全球最大的机器人开源社区（ROS Wiki、GitHub），提供数千个功能包，研究者可快速集成前沿算法，避免重复开发。

### 5.1.2 运动规划软件框架设计

开源的导航框架有很多，例如 ROS 自带的 navigation 就是一个非常受欢迎的框架，其次还有一个开源框架是 move\_base\_flex。navigation 框架以 move\_base 节点作为整个机器人的运动逻辑控制中心，集成了定位、路径规划、避障和运动控制等关键功能。它通过 AMCL 算法实现机器人在已知地图中的实时定位，利用 全局规划器生成最优路径，并通过局部规划器 实时调整路径以避开动态障碍物。其分层代价地图（global\_costmap 和 local\_costmap）动态更新环境信息，确保导航安全。支持灵活配置传感器、规划算法及地图参数，适用于服务机器人、仓储物流和工业自动化等场景，提供从建图到自主导航的一体化解决方案。但是目前正处于机器人高速发展的时期，navigation 会随着 ROS 版本的更替而快速更替，这给开发者带来了很多的不便。而 move\_base\_flex 的诞生解决了这个问题，该框架以向后兼容为目的，旨在让使用者可以不必关注于框架的改变，可以快速适配自己的机器人。

由于本课题的重点在于运动规划，所以涉及到感知、定位和控制我们不做讨论。navigation 的规划框架由全局规划器、局部规划器、全局代价地图和局部代价地图构成。全局规划器利用全局代价地图提供的环境信息使用 A\* 或者 Dijkstra 算法完成路径规划。局部规划器使用局部代价地图提供的环境信息，使用 DWA 和 TEB 等算法进行局部轨迹规划。如图5-2是 navigation 的规划框架图。

受这 navigation 规划框架的启发，我们采用流水线的设计模式，针对于我们的仿真设计了有效的规划框架。图5-3是我们的规划框架图。相对于 navigation 的规划框架我们做了几个变化。第一，我们出于简化框架的目的去除了 recover 模块，它用来为规划提供恢复行为。第二，我们加入了 path\_smoother 模块为 RRT-planner 提供路径平滑功能，它们两个部分共同组成全局规划器。第三，我们将局部规划分解为前端 Hybrid A\* 和后端轨迹优化（trajectory optimizer）两个部分，这样可以提高足够的规划精度。

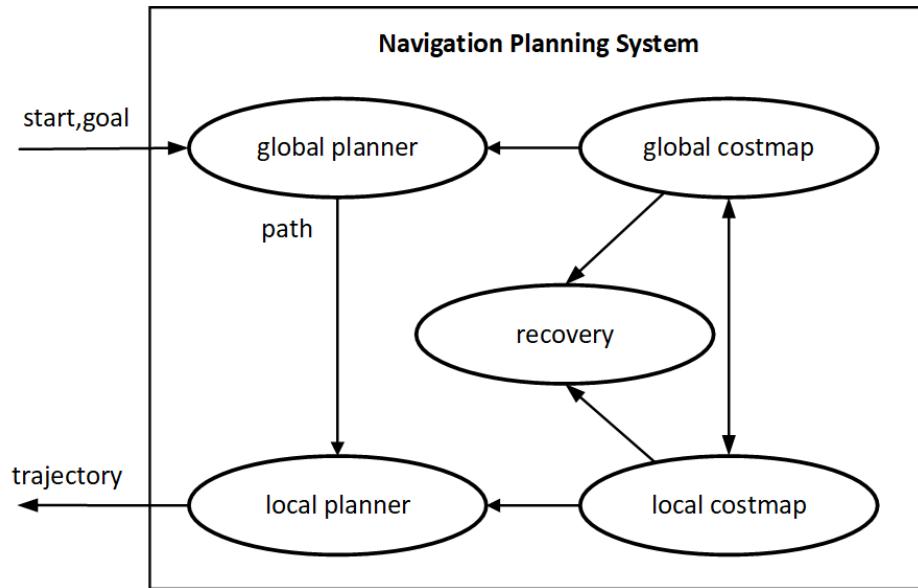


图 5-2 navigation 规划框架图

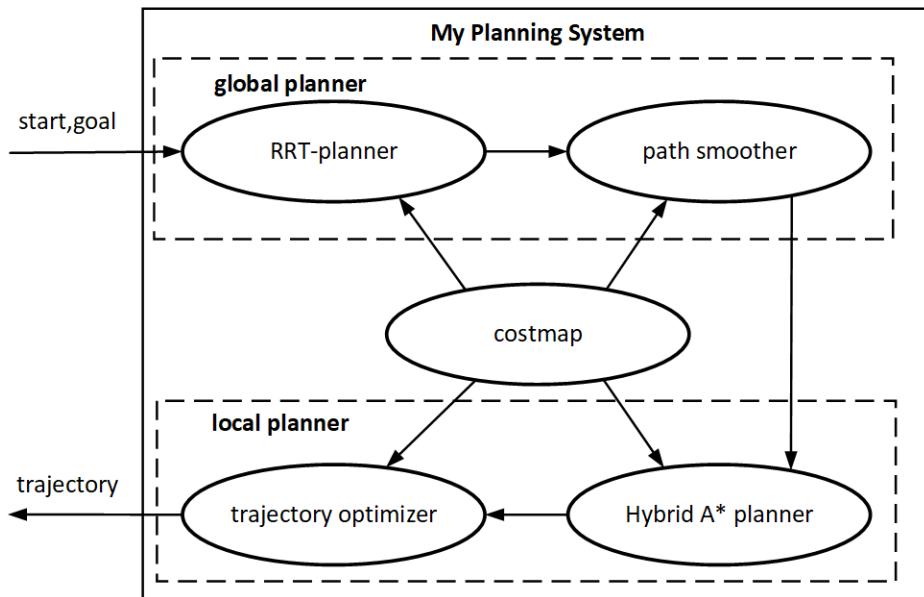


图 5-3 改进的规划框架图

## 5.2 全局规划算法测验

本节中我们对全局规划算法进行仿真测试，其理论原理已经在第三章中完成推导分析。我们将对单一点到点导航的场景和多点航行场景做全局规划以验证我们算法的有效性。在仿真中单点导航和多点导航使用的权重参数都相同，见表5-2。从表中可以看到我们设置的最大曲率为 0.1，最小曲率为-0.1。事实上我们的铰接车允许的最大曲率为 0.2，也就是铰接角为 30 度。但是为了留给后期局部轨迹规划一定的

表 5-2 Global Planner Parameters

Parameter	Description	Value
$k_{min}$	Minimum curvature	-0.1
$k_{max}$	Maximum curvature	0.1
$w_k$	Curvature constraint Penalty Weight	100
$w_c$	Corridor constraint penalty weight	10

优化裕量，全局我们设置的会比实际小。

### 5.2.1 点到点路径规划

如第三章所述，我们选用的前端路径搜索算法是 RRT-connect，然后基于搜索出来的路径进行路径优化。如图5-5是我们设置的障碍物场景，用来测试点到点的规划测试。图中黑色的细线是 RRTconnect 搜索出来的路径，蓝色的区域是沿着路径生成的安全走廊，红色的圆滑的路径则是我们优化算法优化出来的。可以看出我们优化出来的轨迹不仅是足够安全的和足够平滑的，同时转弯幅度（曲率）也在一定范围内，并且起点和终点都和车体的方向（图中箭头方向）一致。

现在我们简单对路径做一下仿真数据分析，图5-6中的右子图是我们路径的曲率曲线。为了方便对比细节，我们重新把路径轨迹在图中画出，图5-6中左子图是优化出来的全局路径，右子图是轨迹的曲率随路程 S 的曲线。可以看出在坐标 (80, 0) 左右对应路程 S=160 米时的曲率为最大，值为 0.097。曲率曲线的值始终小于 0.1，也就是曲率半径始终大于 10，这和我们的优化参数设置是对应的。

由于本算法分为前端搜索和后端优化两个部分，因此我们将算法的运行时间记录在了表5-3中，可以看出我们的优化算法在平均 200 米左右路程时，平均优化时间为 10.14ms，完全满足高实时性的要求。前端 RRTconnect 算法我们使用 ompl 【参考文献】库提供，平均求解时间在 6.07ms 左右。算法总的求解时间平均在 17.71ms 左右。

表 5-3 全局规划算法运行时间

算法阶段	最小值 (ms)	最大值 (ms)	平均值 (ms)
前端 RRTconnect	3.04	12.22	6.07
后端优化	0.38	21.44	10.14
总时间	4.32	24.18	17.71

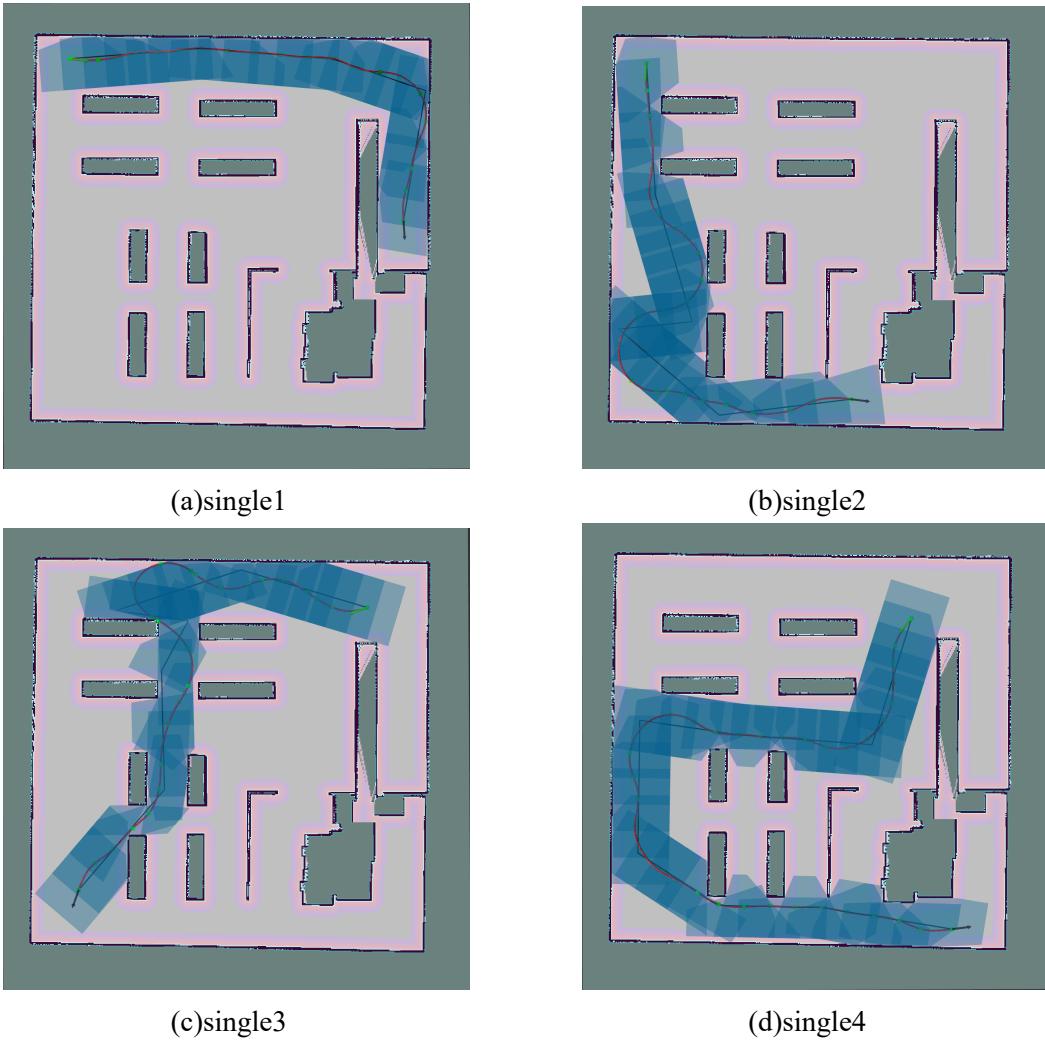


图 5-4 点到点路径规划仿真案例

### 5.2.2 多点路径规划测试

上一小节我们仿真测试了点到点的路径规划算法，这一节我们测试多点巡航的路径规划算法，每一个航点都是一个换向点。航点在 rviz 可视化界面中可以方便的用鼠标拖动。

如图5-7a【图中标注换向点】是我们多点航行的小测试。我们手动给出了一共 6 个航点 A、B、C、D、E 和 F，其中 A、F 是起点和终点，中间四个换向点。蓝色的区域是我们沿着前端搜索路径膨胀出来的安全走廊，红色的是我们优化出来的路径。可以看出，在换向点处连个路径的切线方向是相同，这意味着车辆在该点处可以正常完成换向，即从正向到倒车或从倒车到唤醒的转换。这个第二章中我们介绍的算法理论是契合的。

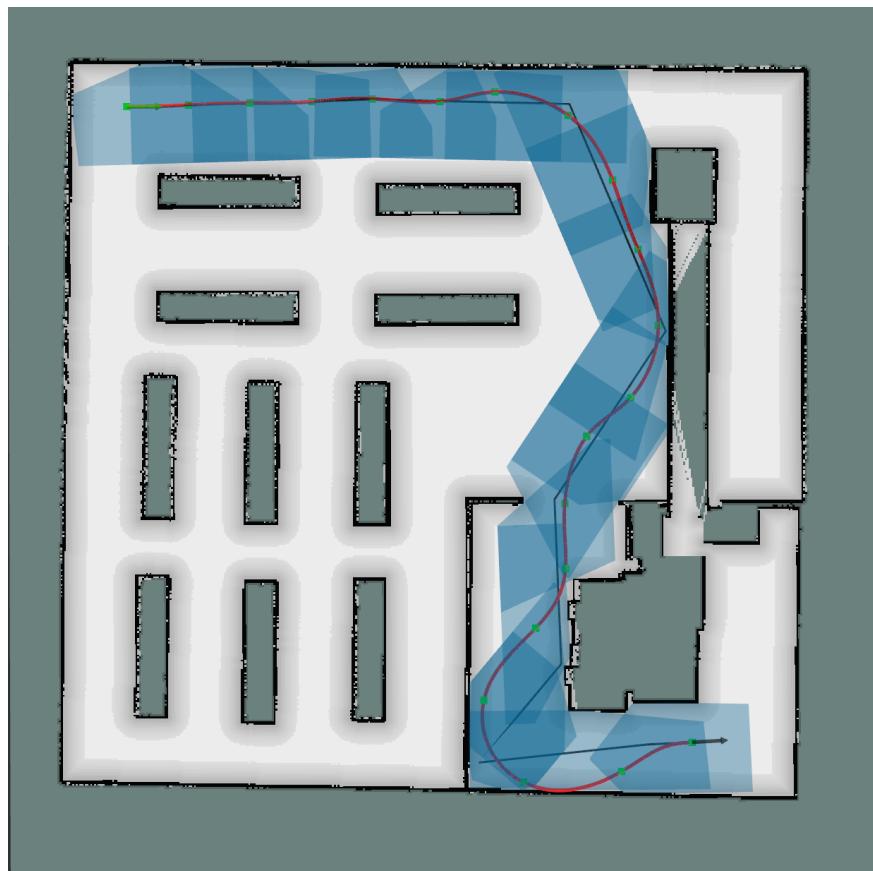


图 5-5 single

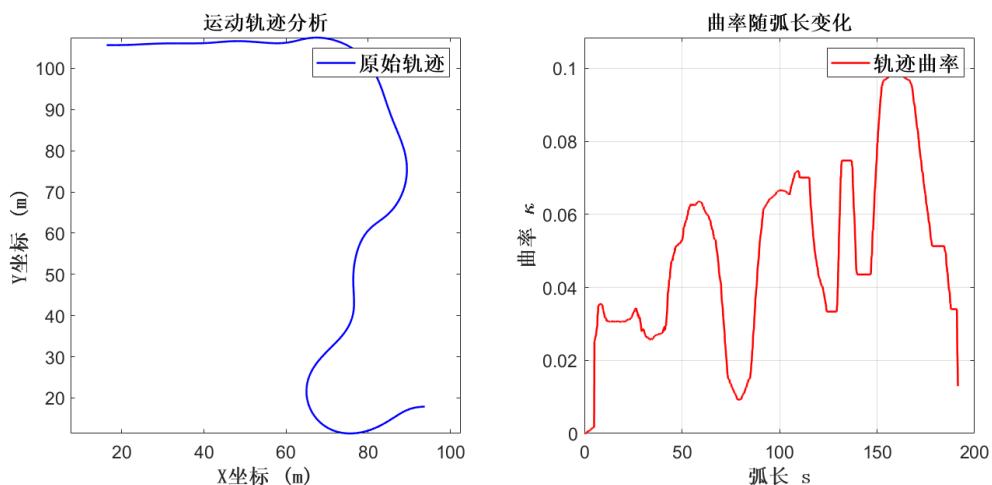


图 5-6 datasingle

图5-8是我们多航点路径的曲率曲线。和上一小节一样，我们重新把路径轨迹在图中画出，图5-8中左子图是优化出来的全局路径，右子图是轨迹的曲率随路程 S 的曲线。可以看出在路程 S=80 米、S=140 米和 S=210 米时的曲率曲率都接近 0.1，但

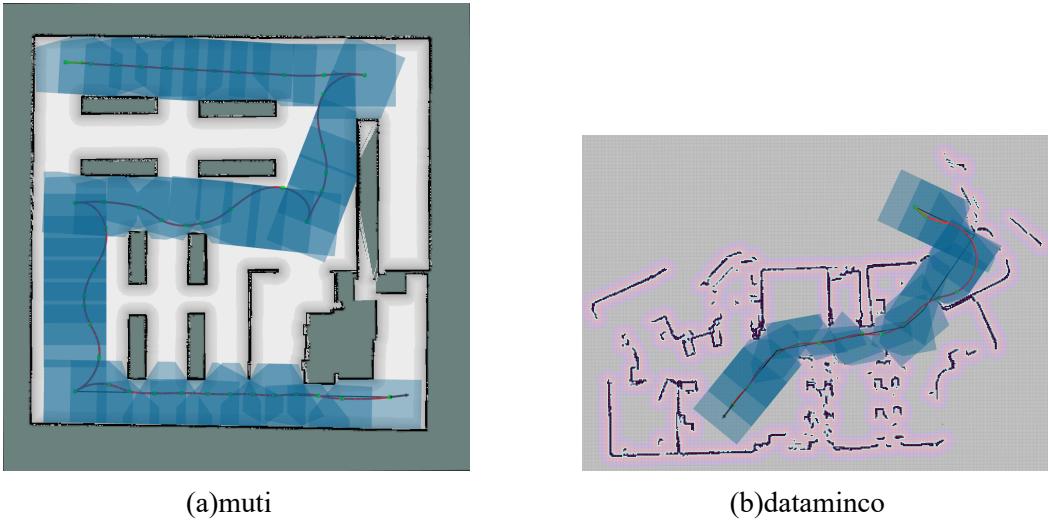


图 5-7 多点路径规划仿真案例

是始终未超出。曲率半径始终大于 10，这和我们的优化参数设置是对应的。可见我们的算法的稳定性是很好的。

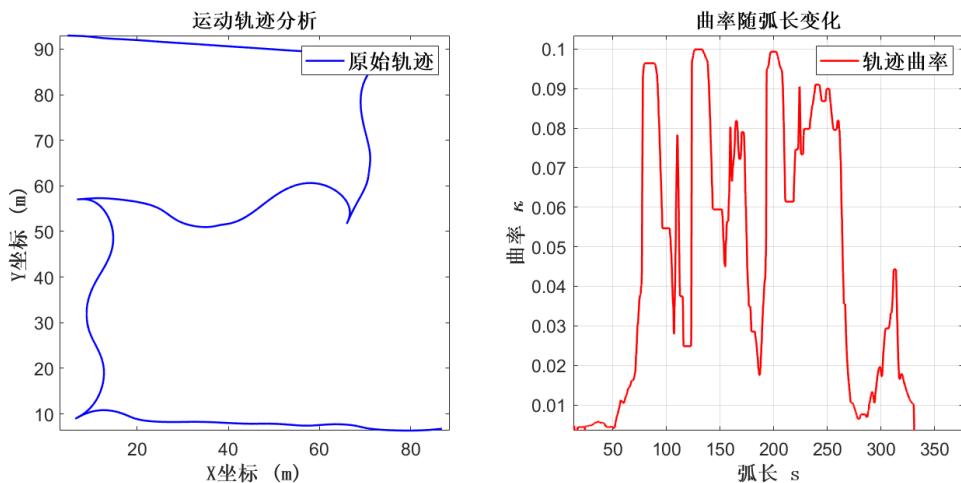


图 5-8 datamuti

### 5.3 局部规划算法测验

本文局部规划算法我们主要设置了两个实用场景用于验证，分别是自主泊车场景和侧方位停车场景。下面将介绍上述两种场景的仿真结果，详细的算法理论部分参考第四章。

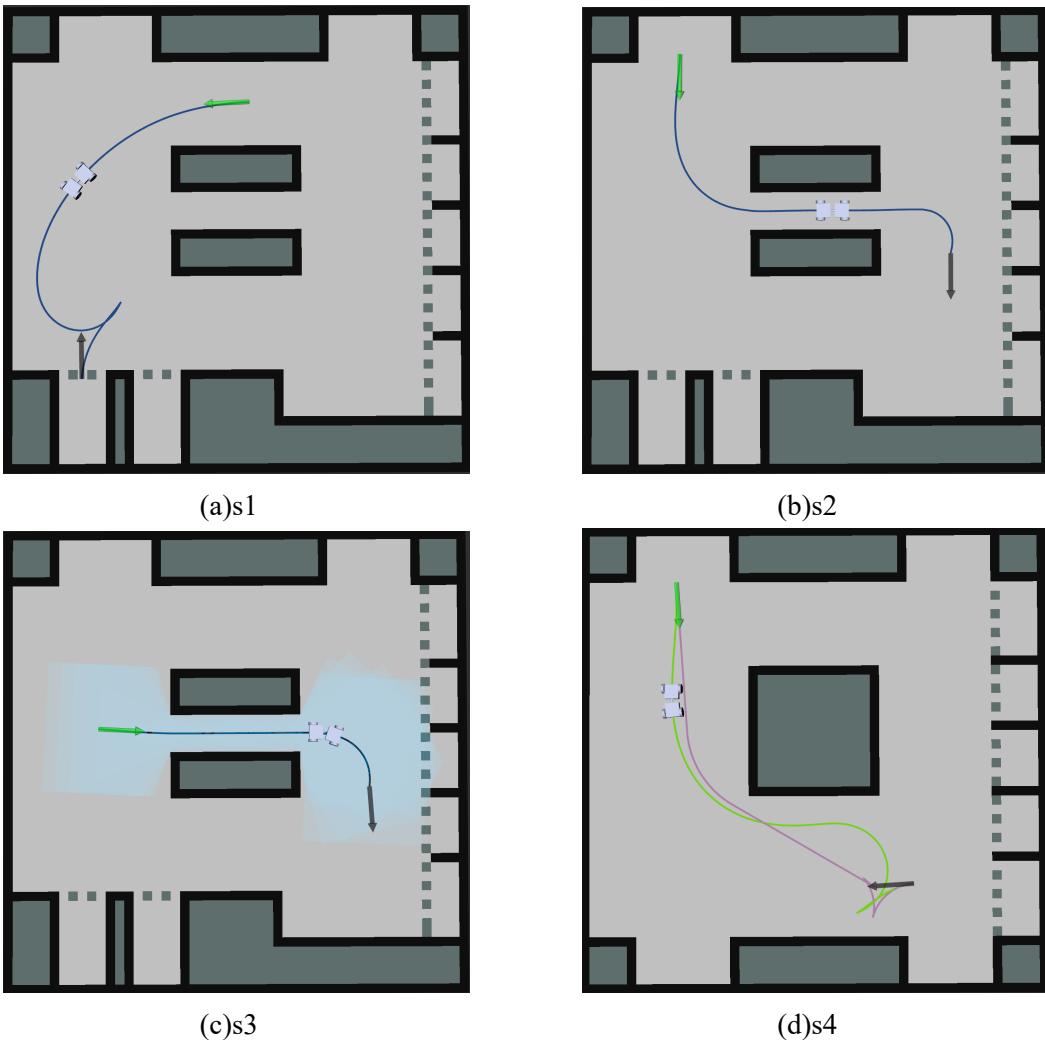


图 5-9 局部规划仿真案例

### 5.3.1 自主泊车场景

自主泊车类的算法要求车辆允许后退，一般的规划算法缺失前进和后退两种模式的区别，五次多项式等。但是我们的算法基于运动学模型，天然具备前进后退的切换能力。在泊车这种极为苛刻的条件仍然能够胜任。如图5-11是我们测试的泊车场景，铰接将从绿色箭头处开始泊车到灰色箭头。在图中，深蓝色细的轨迹是我用改进 Hybrid A\* 生成的，可以看出它在离终点近的时候进行三次换向才能成功完成泊车，蓝色的区域是我们沿着 Hybrid A\* 生成的轨迹膨胀出来安全区域，也就是安全走廊，铰接车在这个区域内就是碰撞避免的。绿色轨迹的是我们局部规划算法规划出来的轨迹，可以看出它是十分圆滑的，而且距离终点近的时候只进行了一次换向就完成了泊车。值得指出的是，我们的算法虽然使用 Hybrid A\* 生成的轨迹作为迭代粗

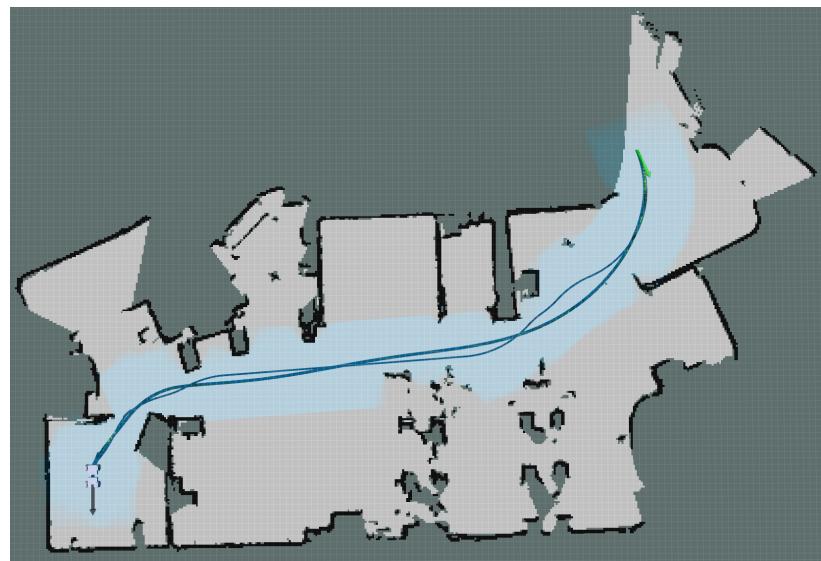


图 5-10 nmpcyuewa1

解，但是我们的轨迹质量却远远优于 Hybrid A\*，换向点大大减少。这说明我们的轨迹规划算法并不很依赖于初始解的质量，即使初始解质量并不是很好，我们的算法仍然能将轨迹拉回到一个不错的水平。

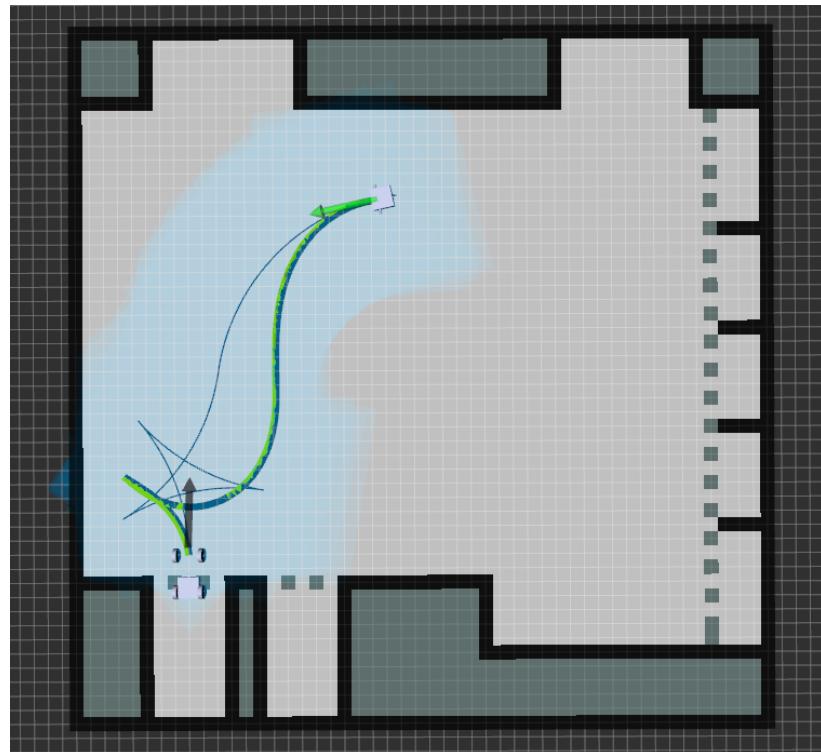


图 5-11 boche

上面我们对轨迹进行了可视化，下面我们来分析一下轨迹的仿真数据。如图5-

12是航向角、铰接角、车体速度和车体加速度随时间的变化图。从图中可以看出，我们的航向角和车速都是非常圆滑的，也就是可微的，这得益于我们的运动学约束。我们还可以发现一个小现象，对于铰接角和加速度它们只是连续的，有些时候他们的斜率仍然会发生小突变，这是因为他们和输入量铰接角速度和加加速度只保持有一阶导数的关系，由于输入量可以是不连续的，所以他们并不能保证是可微的。另外可以看出，铰接角范围在  $-0.5rad$   $0.5rad$  之间、车体速度保持在  $-1m/s$   $3m/s$  之间、加速度保持在  $-0.4m/s^2$   $0.4m/s^2$  之间，都在约束范围内。从速度曲线可以看出，在 0s 到 26s 之间铰接车速度为正，处于前进状态，在 26s 时速度减为 0m/s，此时铰接车到达换向点，对应可以看示意图5-11。此时从铰接角曲线可以看出，铰接角此时也为 0rad，也就是方向盘回正了。之后速度为负，一直倒车泊入车库。

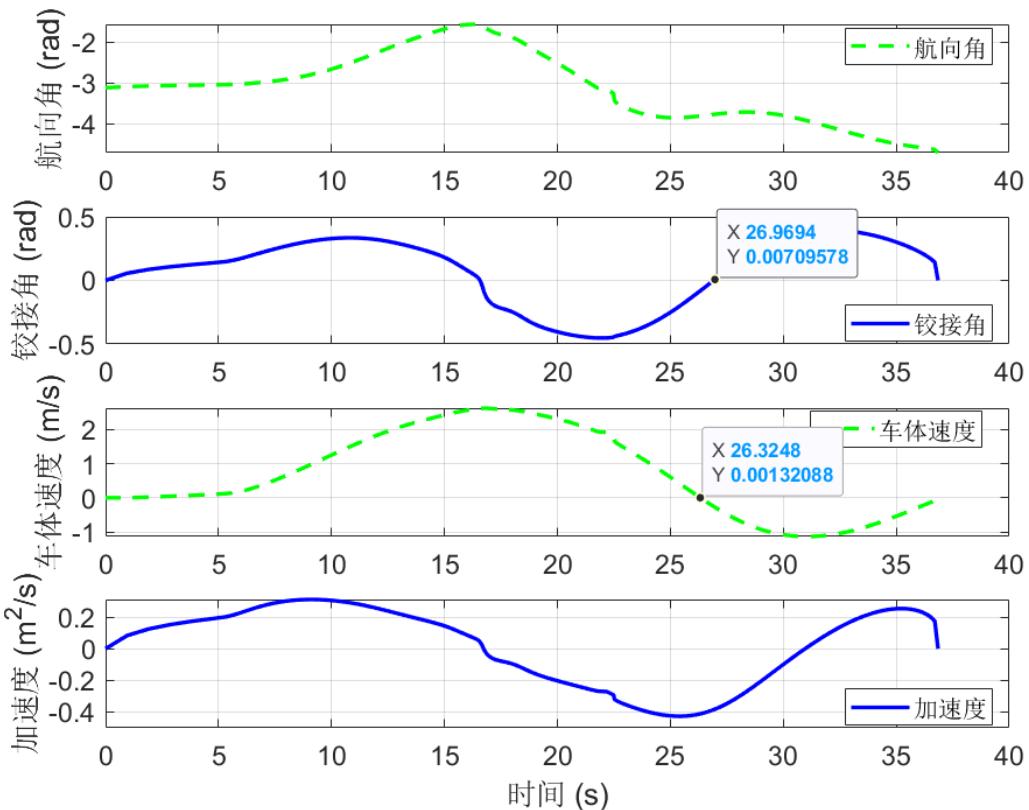


图 5-12 databocheestate

如图5-13，是我们优化出的轨迹的输入量，可以看到在 5s、17s 和 24s 时加速度都发生了突变，对应角速度也发生了突变。这是由系统的输入量并不会受到微分方程的约束。但是可以看出，输入变量和我们事先定下的边界约束时保持一致的，较

接角速度在-0.2rad/s 0.2rad/s, 加加速度在-0.3 0.3 之间。

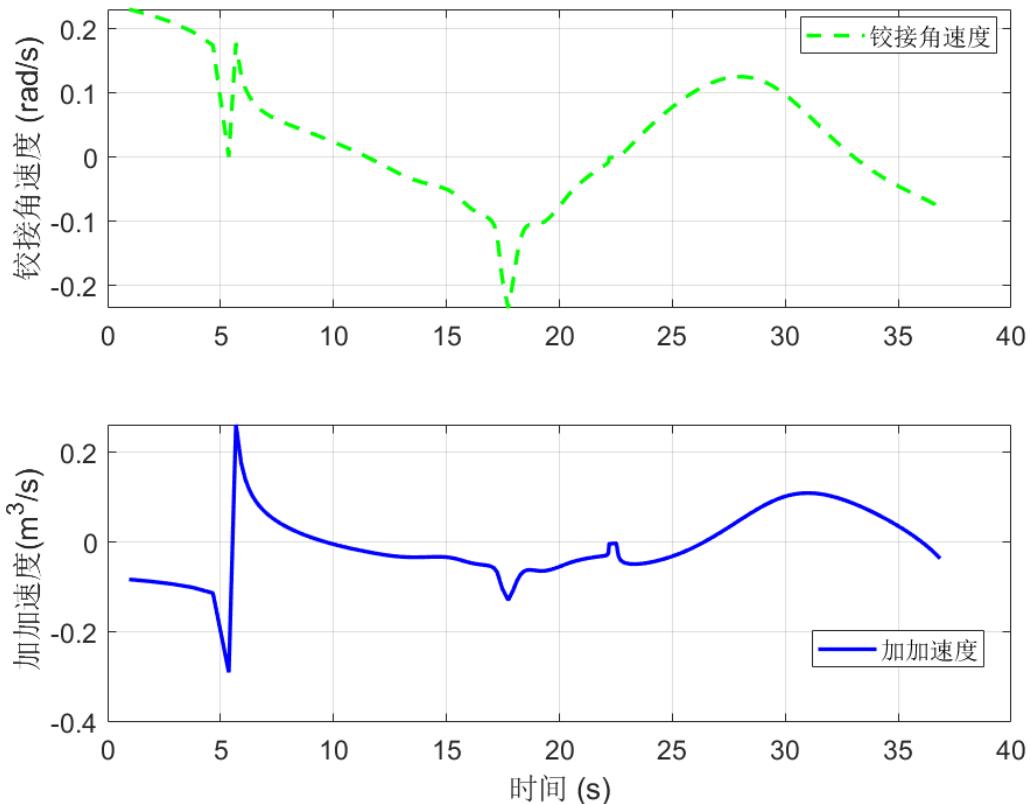


图 5-13 databocheinput

表5-4是局部规划算法的运行时间表。

表 5-4 局部规划算法运行时间

测试场景	规划长度 (m)	平均求解时间 (20 次)
绕障 1	20 30	78.2ms
绕障 2	30 40	106.0ms
绕障 3	40 50	147.8ms
泊车 1	30 40	207.4ms
泊车 2	40 50	515.2ms

### 5.3.2 侧方位停车场景

侧方位停车与自主泊车具有同等重要性，这一小节我们将做侧方位停车的仿真实验。如图5-14是我们侧边泊车的 rviz 仿真可视化图，其中细的轨迹是改进的 Hybrid

A\*生成的，它在距离终点很近时发生了一次泊车。蓝色区域是沿着细轨迹膨胀出来的安全走廊区域。较粗的轨迹是我们算法生成的最终轨迹。可以看出我们的轨迹对于该侧边停车场景并没有发生换向，一次就到位。由此可以看出我们算法的鲁棒性。

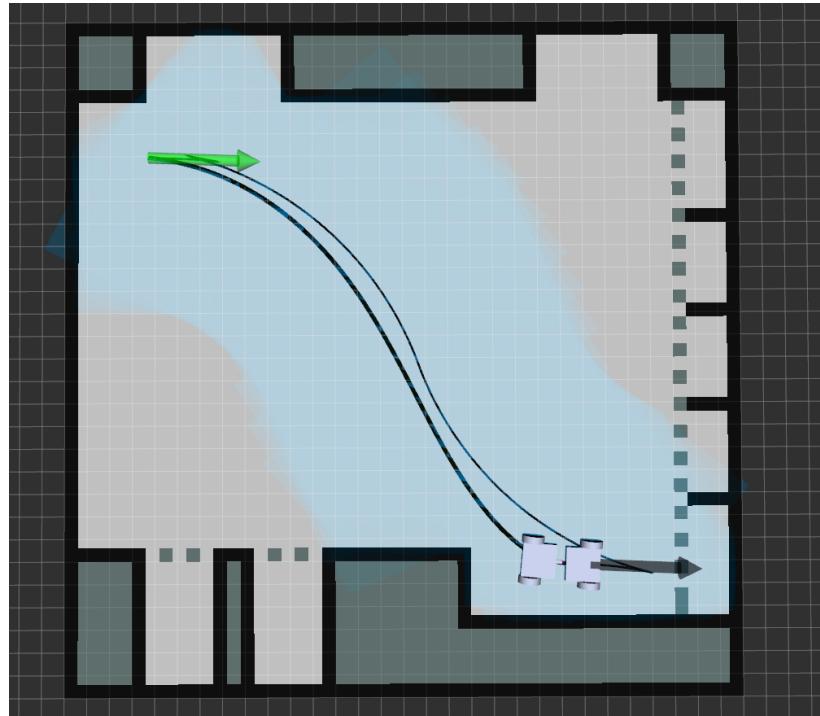


图 5-14 cebianboche

图5-15是侧边泊车的状态曲线图，包括航向角、铰接角、速度和加速度。可以从速度曲线看出，由于并没有发生换向，所以铰接车的速度始终保持为正。

## 5.4 穿越复杂地形实验

前面两节我们对全局规划和局部规划都做了详细的算法性能测试，为了多方面验证算法的性能，这一节我们将使用更真实的复杂地形来做一次仿真。我们对一个大的车间进行了建图，得到一个栅格地图，用这个地图来验证我们的算法。地图的像素是  $\text{cm}$ ，分辨率是  $\text{cm}$ ，长宽  $\text{m}$ 。图【很多图】是从不同的起点到到不同终点全局规划的路径。其中贴近障碍物的紫色区域是我们设置的膨胀层，目的是让前端RRTconnect算法搜索出的路径尽量远离障碍物，从而增大后端优化的成功率。

## 5.5 本章总结

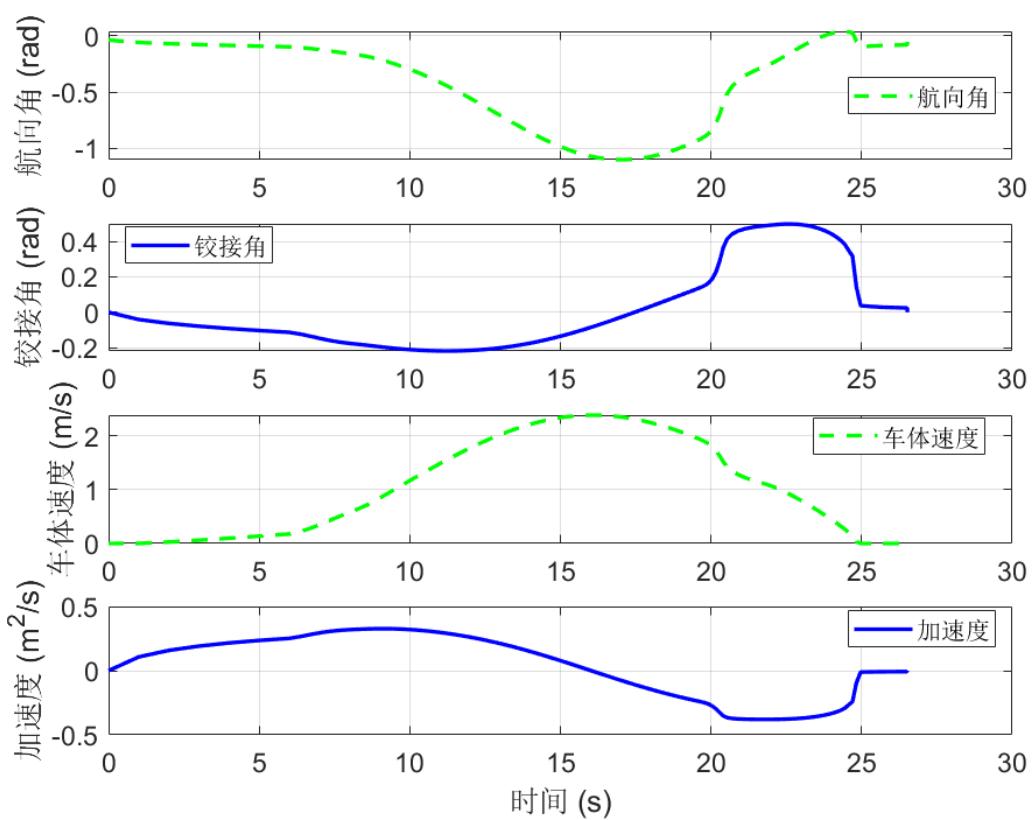


图 5-15 datacebianstate

## 参考文献

- [1] GAO F, WU W, LIN Y, et al. Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial[J/OL]. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018: 344-351. <https://api.semanticscholar.org/CorpusID:52283216>.
- [2] LI B, ACARMAN T, ZHANG Y, et al. Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework[J/OL]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8): 11970-11981. DOI: [10.1109/TITS.2021.310901](https://doi.org/10.1109/TITS.2021.310901).
- [3] DEITS R, TEDRAKE R. Computing large convex regions of obstacle-free space through semidefinite programming[M/OL]. Cham: Springer International Publishing, 2015: 109-124. [https://doi.org/10.1007/978-3-319-16595-0\\_7](https://doi.org/10.1007/978-3-319-16595-0_7).
- [4] LIU S, WATTERSON M, MOHTA K, et al. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments[J/OL]. IEEE Robotics and Automation Letters, 2017, 2(3): 1688-1695. DOI: [10.1109/LRA.2017.2663526](https://doi.org/10.1109/LRA.2017.2663526).

## 附录 A

$$\theta = \omega t$$

$$a = \cos \theta \quad (\text{A-1})$$

$$b = \sin \theta \quad (\text{A-2})$$

$$a^2 + b^2 = 1 \quad (\text{A-3})$$

$$\theta = \omega t$$

$$a = \cos \theta \quad (\text{A-4a})$$

$$b = \sin \theta \quad (\text{A-4b})$$

$$a^2 + b^2 = 1 \quad (\text{A-4c})$$

## 附录 B

$$\theta = \omega t$$

$$a = \cos \theta \quad (\text{B-1})$$

$$b = \sin \theta \quad (\text{B-2})$$

$$a^2 + b^2 = 1 \quad (\text{B-3})$$

$$\theta = \omega t$$

$$a = \cos \theta \quad (\text{B-4a})$$

$$b = \sin \theta \quad (\text{B-4b})$$

$$a^2 + b^2 = 1 \quad (\text{B-4c})$$

## 攻读硕士学位期间取得的成果

### 1. 发表的学术论文

- [1] ×××, ×××. 并联 2-RRR/UPRR 踝关节康复机器人机构及其运动学 [J]. 机器人, 2010, 32(1): 6-12.  
(EI 收录号: 20101212786168)
- [2] ×××, ×××. 空间并联机构连续刚度非线性映射研究 [J]. 机械工程学报, 2008, 44(8): 20-25. (EI  
收录号: 083911606237)
- [3] ×××, ×××. A sampling robot for high dust and strong corrosion environment[C]//International Conference on Robotic and Biomimetics, Tianjin, 2010: 828-832. (EI 收录号: 20111313856140)
- [4] ×××, ×××. 双重驱动四自由度并联机构型综合 [J]. 机械设计与研究, 2008, 24(1): 51-53.
- [5] … 要与参考文献格式一致!!!

### 2. 发表的专著/译著（无著作时此项不必列出）

- [1] ×××, ×××. 高等空间机构学 [M]. 北京: 机械工业出版社, 2010.
- [2] ×××, ×××. 空间并联机构导论 [M]. ×××, 译. 秦皇岛: 燕山大学出版社, 2020.

### 3. 申请及已获得的专利（无专利时此项不必列出）

- [1] ×××, ×××. 具有远程运动中心的三自由度转动并联机构: 中国, 200910073844.8 [P]. 2011-01-05.
- [2] ×××, ×××. 五自由度双重驱动并联机构: 中国, 200910075071.7 [P]. 2011-01-05.

### 4. 科研获奖（无奖励时此项不必列出）

- [1] ×××, ×××. 机器人机型综合及结构分析理论. XX 省科学技术二等奖, 2009.

致 谢

---

---

## 致 谢