



硕士 学位 论文

MASTER'S DISSERTATION

(学术 学位)

论文题名 基于安全走廊的无人铰接车全局及局部规划方法研究

作者姓名 杨远爽

学科专业 控制科学与工程

指导教师 华长春 教授

2025 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

基于安全走廊的无人铰接车全局及局部规划 方法研究

硕士研究生：杨远爽

导师：华长春 教授

申请学位：工学硕士

学科专业：控制科学与工程

所属学院：电气工程学院

答辩日期：2025年6月

授予学位单位：燕山大学

Research on Global and Local Planning Methods for Unmanned Articulated Vehicles Based on Safety Corridors

A dissertation submitted to
Yanshan University
in partial fulfillment of the requirement for the degree of
Master of Engineering
in Control Science and Engineering
By
Yang Yuanshuang
Supervisor: Professor Hua Changchun

School of Electrical Engineering, Yanshan University

June, 2025

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《基于安全走廊的无人铰接车全局及局部规划方法研究》，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

《基于安全走廊的无人铰接车全局及局部规划方法研究》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

基于安全走廊的无人铰接车全局及局部规划方法研究

摘要：无人铰接车凭借其灵活的转向结构，被广泛应用于如路面清洁、矿山开采等领域。在进行物料搬运和障碍物躲避时，铰接车的有效操作高度依赖于运动规划。本文分两个部分解决铰接车的运动规划问题：第一阶段涉及大范围环境中的全局路径规划，旨在确立大致行进路线；第二阶段则聚焦于小范围环境内的局部轨迹规划，以实现精确的避障。具体研究内容如下：

首先，对无滑移情况下的铰接车运动学模型进行了分析，并在此基础上，利用最优控制框架对轨迹规划问题进行了建模，详细阐述该问题的求解方法。

接着，针对于全局路径规划，提出一种基于铰接车微分平坦性的路径规划方法。算法以 RRT-Connect 算法为前端快速搜索航点，后端则基于 Minco 样条曲线进行路径平滑优化，并在优化过程施加安全走廊约束和最大曲率约束。所提出的算法在 200 米的路径规划中平均规划时间为 17.71 毫秒，能够有效满足路径规划的实时性需求。

其次，针对于局部轨迹规划，提出一种基于安全走廊的铰接车轨迹规划方法。算法在规划时充分考虑铰接车前桥后桥的全尺寸，以保证轨迹的安全性。其次，算法前端使用针对铰接车改进的 Hybrid A* 算法搜索路径，后端建立一个最优控制问题，并且利用惩罚函数、微分同胚等方法处理约束。与传统算法相比，所提出的算法在狭窄避障和泊车等场景中表现优异，在 30 米以内的规划距离中平均规划时间为 78.2 毫秒。

最后，本文将所提出的算法进行了仿真集成，从安全性、平滑性、换向次数、舒适度和实时性方面分别做出仿真分析。

关键词：铰接车；运动规划；Minco 样条；RRT-Connect；Hybrid A*；安全走廊

分类号：TM464

Research on Global and Local Planning Methods for Unmanned Articulated Vehicles Based on Safety Corridors

Abstract: Articulated vehicles, known for their flexible steering, are extensively used in tasks like road cleaning and mining. Efficient operation during material handling and obstacle avoidance relies heavily on motion planning, addressed here in two stages: global path planning for a general route and local trajectory planning for precise obstacle avoidance.

Firstly, kinematic Analysis and Optimal Control: The kinematic model under non-slip conditions is analyzed, and the trajectory planning problem is framed within an optimal control context.

Subsequently, global Path Planning: A differential flatness-based method combined with RRT-Connect for waypoint search and Minco spline curves for path smoothing is proposed. This approach averages 17.71 milliseconds over 200 meters, ensuring real-time performance.

Furthermore, Local Trajectory Planning: A safety corridor-based method is introduced, incorporating full vehicle dimensions and using a modified Hybrid A* algorithm for pathfinding, followed by optimal control for refinement. It performs well in tight spaces, averaging 78.2 milliseconds for up to 30 meters.

Finally, Simulations demonstrate the algorithms' effectiveness across various metrics including safety, smoothness, maneuver count, comfort, and real-time feasibility.

Keywords: articulated vehicle; motion planning; Minco spline; RRT-Connect; Hybrid A*; safety corridor

Classification: 621.3

目 次

摘要.....	I
Abstract.....	II
第1章 绪论	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.....	5
1.2.4 基于优化的运动规划方法	7
1.3 铰接车运动规划技术国内外研究现状	7
1.4 本文研究的主要内容	9
1.5 本文结构安排	10
第2章 基于最优控制的铰接车运动规划问题建模.....	11
2.1 铰接车运动学模型建立	11
2.2 基于最优控制的运动规划问题建模	13
2.3 最优控制问题的解法	14
2.3.1 极小值原理.....	14
2.3.2 基于配点法的数值求解.....	14
2.4 非线性规划问题的求解	15
2.4.1 罚函数法	15
2.4.2 非线性规划问题的初始解	16
2.5 本章小结	17
第3章 基于微分平坦的无人铰接车全局轨迹规划方法	18
3.1 铰接模型微分平坦性	19
3.2 轨迹的参数化表达形式	20
3.2.1 Minco 样条曲线基本原理	20
3.2.2 起点和终点约束方程	22
3.2.3 航点的连续性约束方程	23
3.2.4 轨迹的闭式解	24
3.3 基于安全走廊的安全约束分析	25
3.4 点到点全局路径规划	27
3.4.1 铰接车的最大曲率.....	27

3.4.2 优化问题的建立	28
3.4.3 起点和终点航向角保证	30
3.4.4 优化问题求解	30
3.5 多站点航行全局路径规划	31
3.5.1 多站点航行规划的问题描述	32
3.5.2 站点的换向处理	32
3.6 本章小结	33
第4章 基于安全走廊的无人铰接车局部规划方法	35
4.1 铰接车的几何结构构建模	35
4.2 铰接车 Hybrid A* 算法的改进	36
4.2.1 节点拓展策略改进	37
4.2.2 代价函数设计	38
4.2.3 碰撞检测算法设计	39
4.3 基于 NMPC 和安全走廊的轨迹规划	40
4.3.1 NMPC 问题构建	40
4.3.2 运动学约束	42
4.3.3 安全走廊的防碰撞约束处理	43
4.3.4 变量的边界约束	45
4.3.5 时间正则化约束	46
4.3.6 优化问题求解	46
4.4 本章小结	47
第5章 无人铰接车运动规划系统仿真	48
5.1 仿真环境及运动规划软件架构设计	48
5.2 全局规划算法仿真	52
5.2.1 点到点路径规划仿真	52
5.2.2 多站点路径规划仿真	55
5.3 局部规划算法仿真	56
5.3.1 自主泊车场景	58
5.3.2 侧方位停车场景	61
5.3.3 局部规划算法的性能对比	63
5.4 本章小结	64
第6章 结论	66
参考文献	68
攻读硕士学位期间取得的成果	73
致谢	74

第1章 绪论

1.1 研究背景与意义

作为人工智能与机器人技术的交叉领域，无人驾驶相关技术凭借其高效、精准的特点，正在成为智能化转型的核心支柱之一。针对复杂环境下的自主导航与决策问题，无人驾驶技术通过创新的算法与系统设计，提供了切实可行的解决方案^[1]。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^[2]。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^[3]。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^[4]。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^[5]。两个规划层使得系统既能适应大范围导航，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

运动规划是自主导航任务中必不可少组成部分之一，主要分为：基于图搜索的、基于概率采样的、基于曲线插值的和基于优化的规划算法。

1.2.1 基于图搜索的运动规划方法

基于图搜索的规划算法通常将路径点抽象为图中的节点，而路径段则被抽象为连接这些节点的边。通过这种方式，可以构建出一个拓扑图结构来代表可行的路径选项。图1-1是几种搜索规划算法的仿真效果图。Dijkstra 算法^[6] 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很高，难以应用于大规模场景。A* 算法^[7] 创新性地引入启发式函数，将搜索方向引导至目标节点，在保证路径最优性的前提下降低复杂度，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。D* 算法^[8] 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 $O(k)$ ，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^[9] 提出的 LPA* (Lifelong Planning A*) 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^[10] 提出 ARA* (Anytime Repairing A*) 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。

LRTA* (Learning in real-time A*) 算法^[11] 通过剪枝与动态调整搜索深度，将启发式搜索的延迟控制在毫秒级，但其剪枝策略可能丢失潜在最优路径。JPS 算法^[12] 针对栅格地图提出“跳跃点”，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。RTAA* (Real-time adaptive A*) 算法^[13] 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。

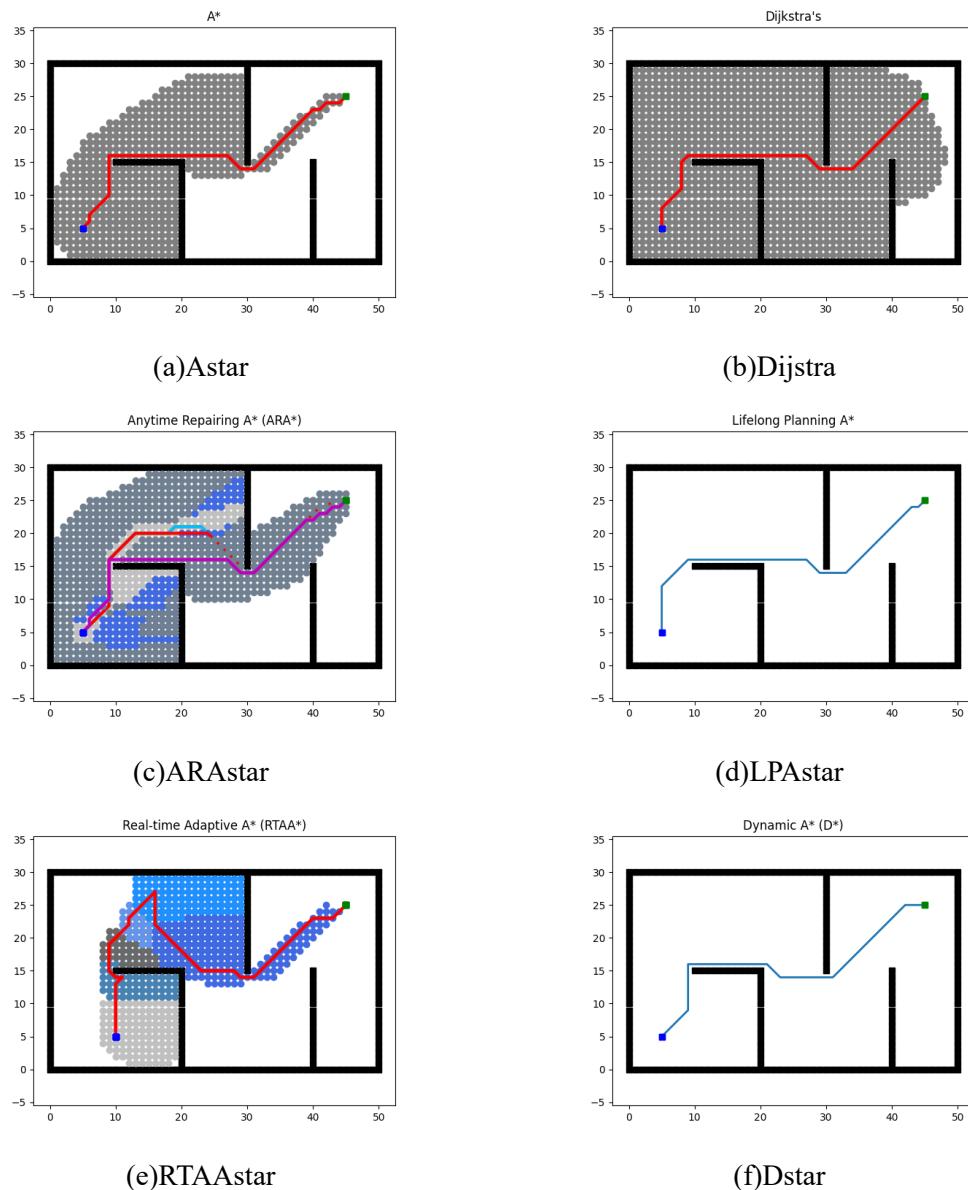


图 1-1 基于搜索的路径规划算法效果对比图

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通过在地图中随机生成采样点来探索可能的路径。这些采样点被生成在整个可行空间内，任何落在障碍物区域内的点都会被排除在外。剩余的采样点随后被相互连接，形成一条从起点到终点的连续路径。与基于搜索图结构的传统方法不同，基于采样的方法特别适合解决高维空间中的路径规划问题，因为它们不需要对整个空间进行详尽的离散化。这类采样的算法在理论上具有概率完备性，这意味着只要存在一个解，在足够长的时间内，通过不断采样，算法几乎可以

确定地找到至少一个可行解。这种特性使得基于采样的路径规划方法在处理复杂和动态环境时显得尤为有效。此外，随着计算时间的增加，找到最优或接近最优解的概率也会相应提高。图1-2是几种RRT系列采样规划算法的仿真效果图。

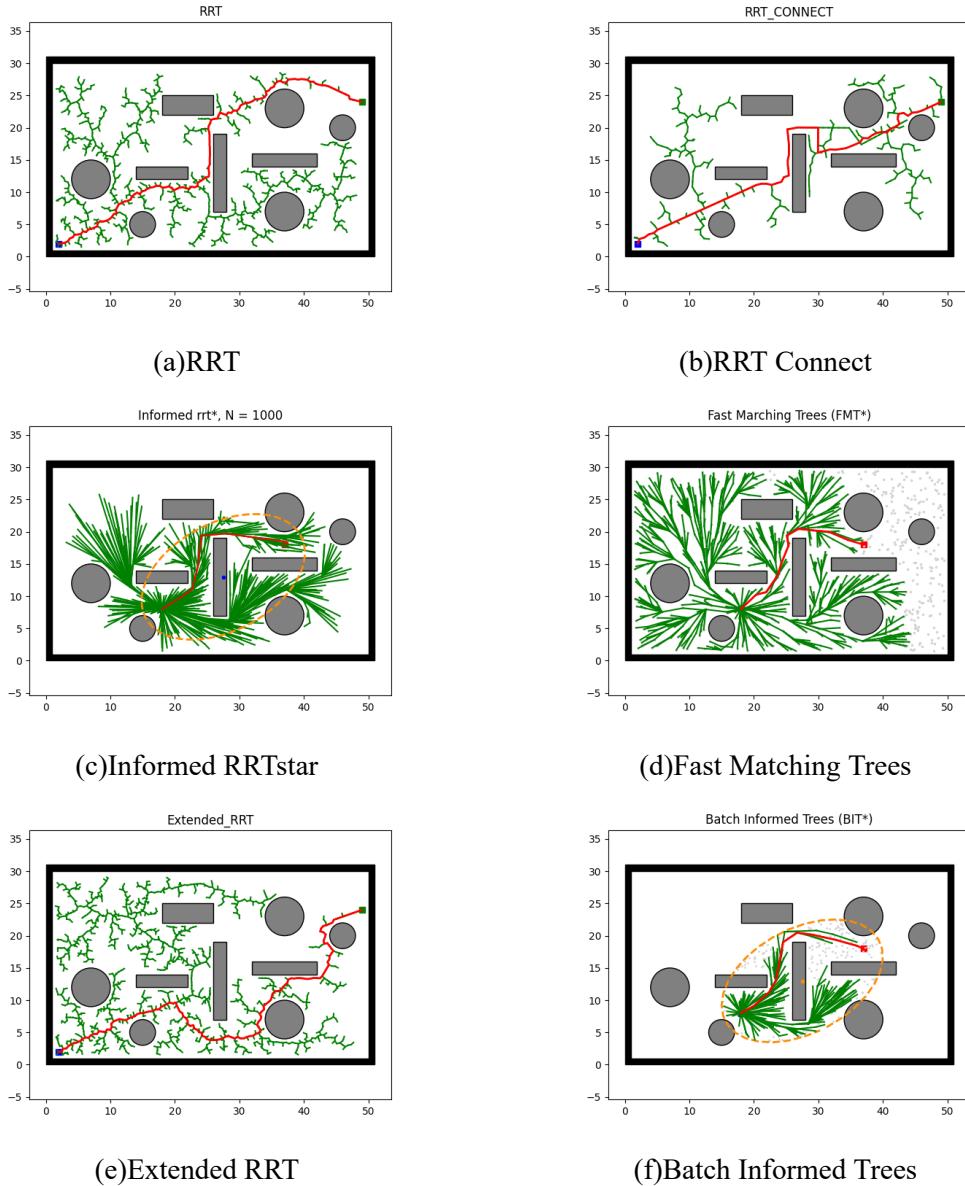


图 1-2 基于采样的路径规划效果对比图

RRT (Rapidly-exploring Random Trees) 算法^[14]开创性地结合了随机采样与树形结构，通过倾向于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。其概率完备性的特性为后续的采样算法奠定了理论基础，但该算法生成路径的质量并不保证最优。为了提高查询效率，RRT-Connect 算法^[15]引入双向树增长与贪

心连接策略，显著加快了在复杂环境中的收敛速度，成为实时机器人导航的关键技术之一。然而，这类算法主要集中在寻找可行解，未能解决路径最优性的问题。针对动态环境下的实时规划需求，ERRT（Execution-extended RRT）算法^[16] 通过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重复计算的问题，实现了首次实时在线规划。尽管如此，ERRT 的路径优化依赖于启发式规则，缺乏全局最优性保障。进一步的发展包括 RRT* 算法^[17]，它不仅能够在复杂环境中高效找到可行路径，还能优化路径以达到时间或距离上的最优化，特别是在高维空间中表现优异。Informed RRT^[18] 则通过椭圆启发式采样，将搜索限制在可能改进当前解的区域内，提升了收敛速度和最终路径质量，并首次从理论上证明了采样算法可渐进趋近最优解。这揭示了传统 RRT 因全局均匀采样导致的效率瓶颈，为后续最优性导向算法提供了理论依据。最后，BIT*（Batch Informed Trees）算法^[19] 融合了图搜索（基于 A* 启发式）与批处理采样的方法，在隐式随机几何图上执行增量搜索，解决了高维空间中传统算法收敛缓慢的问题。这些进展标志着从单纯追求“可行解”向确保“最优解”的转变，但在面对高维场景时，仍需克服计算复杂度的挑战。

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方法以其路径平滑性的显著优势，区别于图搜索和采样规划方法。图1-3展示了多种曲线插值技术的效果，突显了其在路径规划中的应用潜力。

早期的研究主要集中在静态环境下的几何路径求解，Reeds-Shepp 曲线^[20] 作为这一领域的先驱，为车辆运动规划奠定了理论基础。它在 Dubins 曲线^[21] 的基础上引入倒车机制，证明了包含至多两个尖点的 CCSCC 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。然而，该方法局限于二维平面几何推导，并未考虑动态障碍物和实时计算需求，在复杂场景下其枚举式搜索效率较低。随着自动驾驶技术的发展，百度 Apollo 团队于 2017 年提出的 EM Motion Planner 算法^[22] 标志着向动态环境适应性和多约束协同优化的重大进步。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，实现了交通规则、障碍物决策与运动平滑性的多目标协同。其创新之处在于将决策问题转化为可行轨迹生成，通过

构建凸可行空间确保了解的存在性。Gao 等人^[23]利用安全飞行走廊和贝塞尔曲线^[24]提出了一种优化方法以构建时间最优轨迹，推动了无人机领域实时规划的发展。该方法巧妙地结合快速行进法与 ESDF（截断符号距离场），构建速度自适应的时空索引路径，解决了传统方法中时间分配不合理的问题。通过利用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。实验结果表明，该方法能够在未知环境中实时生成安全轨迹，但在狭窄空间中的飞行走廊膨胀策略可能过于保守，且未能有效处理动态障碍物的预测与避让。

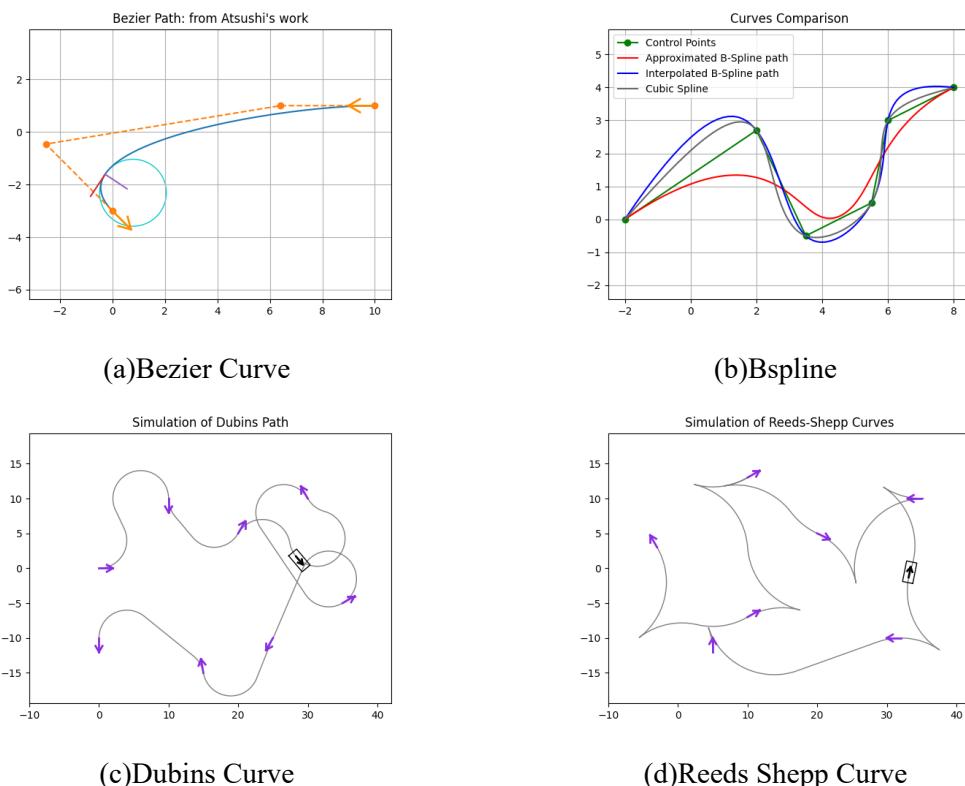


图 1-3 基于曲线插值的路径规划效果对比图

Nguyen 团队^[25]提出了多面体导航工具箱，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，使得线性约束控制点即可保障全局安全性。该工作建立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。然而，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待提升。

1.2.4 基于优化的运动规划方法

Li 等^[26]提出了一种基于时间最优控制 (Optimal Control) 的动态优化框架, 该框架将平行、垂直等规则停车场景与不规则障碍物场景统一建模。通过严格描述车辆运动学、机械约束及碰撞条件, 并采用内点法求解非线性优化问题, 该方法构建了一个开放框架, 允许用户自定义约束和优化目标。然而, 由于其计算复杂度较高, 难以满足实时性需求。Kondak 等^[27]则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, 并引入人工势场来处理障碍物碰撞, 支持全维车辆模型。尽管这种方法在处理障碍物时表现出一定的灵活性, 但其势场函数设计依赖于经验参数, 且未解决非凸优化对初值敏感的问题。Arantes 等^[28]针对离散时间步间的“跳跃碰撞”问题, 提出了基于混合整数线性规划 (MILP) 的连续时间碰撞避免编码方法。通过确保相邻状态位于障碍物同侧, 保证了轨迹段的安全性, 并引入机会约束至风险分配框架, 平衡了路径的安全性和保守性。虽然该方法显著提升了非凸环境下的路径安全性, 但由于 MILP 的 NP-hard 特性, 限制了其在大规模障碍物场景中的应用扩展性。Zhang 与 Borrelli 等^[29]提出的 OBCA (Optimization-Based Collision Avoidance) 框架进一步创新, 通过强对偶定理重构平滑碰撞约束, 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。符号距离函数用于量化侵入程度, 实现了“最小侵入”轨迹生成。然而, 此方法要求障碍物分解为凸集, 导致预处理成本较高。Zhang 等^[30]提出的分层优化框架 H-OBCA (Hierarchical-OBCA), 结合了 Hybrid A 算法^[31]的全局搜索能力与优化碰撞避免 (OBCA) 的局部精细化调整。Hybrid A 基于简化模型快速生成粗路径, OBCA 则利用全维模型优化路径的平滑性和动态可行性, 解决了非凸优化依赖初值的瓶颈。实验表明, 该方法在狭窄场景中能够实现实时规划, 且生成的轨迹可以被底层控制器精准跟踪。然而, 其性能受限于 Hybrid A* 的路径质量, 在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。图1-4为几种常见的铰接车类型。早期的代表性工作如 Choi 与 Huhtala 提出基于 Bezier 曲线的分层规划框架^[32], 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。Sarata 等人^[33] 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。Nayl 等^[34] 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 $\pm 0.15\text{m}$ 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^[35] 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^[36] 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^[37] 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。



(a) 轮式装载机



(b) 四驱铰接式叉车



(c) 沃尔沃 A30F 铰接式卡车



(d) 沃尔沃 A35F 铰接式卡车

图 1-4 几种常见的铰接车: 铰接式叉车、轮式装载机、铰接式四驱卡车

Zhao 等^[38] 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 等^[39] 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很值得研究的方向之一。

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。

首先，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一组平坦变量的多阶导数表示，以简化轨迹规划问题，并加强求解的数值稳定性。其次，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，本文引入 Minco 样条曲线 (Minimum Control Effort Spline)，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。最后，设计合适的曲率约束松弛函数，将路径规划问题转化为无约束优化问题，并且利用 L-BFGS 算法来达到高效求解的目的，并且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。

首先，将轨迹规划问题建模为非线性模型预测控制 (NMPC) 问题。其次，引入安全走廊策略，保证轨迹的安全性。同时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。最后，设计合适的外罚函数策略和微分同胚变换，在不改变原问题凸性的条件下，将问题转换为无约束的优化，以增强求解速

度。同时避免由于大量的约束和优化变量产生的维度灾难问题。

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，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），设计换向点处理机制，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 (Robot Operating System) 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 基于最优控制的铰接车运动规划问题建模

本章节将首先介绍运动规划系统框架，它包括有全局规划和局部规划两个部分。然后介绍铰接式车辆的运动学模型，并基于该模型构建最优控制问题。最后介绍外点法和内点法在优化问题中的使用。

2.1 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2-1所示。

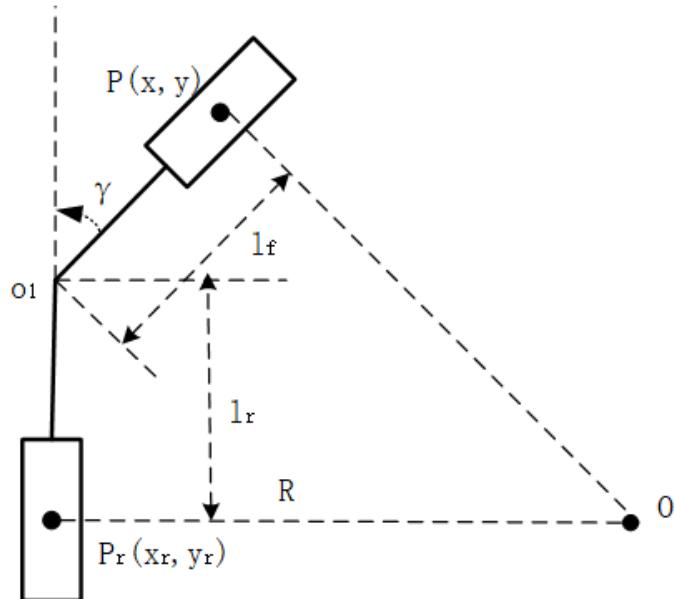


图 2-1 铰接车简易模型

图中，前轴中点 $P(x, y)$ 为前轴中点位置， $P_r(x_r, y_r)$ 为后轴中点位置。也是整车参考点。 θ 为前轴的航向角， θ_r 为后轴航向角。 O 点为瞬时转向圆心， o_1 为铰接点。 l_f 是前轴长度， l_r 是后轴长度， γ 是铰接角， R 为车体转弯半径。考虑到铰接车工作在小范围场景，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。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，从而可以得到：

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{x} \sin \theta - \dot{y} \cos \theta &= 0 \\ \dot{x}_r \sin \theta_r - \dot{y}_r \cos \theta_r &= 0\end{aligned}\tag{2-1}$$

其中， v 为车体前桥的参考点速度， \dot{x} 和 \dot{y} 分别为铰接车位置 x 和 y 对时间的导数。

由于前桥和后桥通过铰接角刚性连接，可以推导出前桥参考点和后桥参考点的坐标变换见式 (2-2)：

$$\begin{aligned}x &= x_r + l_r \cos \theta_r + l_f \cos \theta \\ y &= y_r + l_r \sin \theta_r + l_f \sin \theta \\ \theta_r &= \theta - \gamma\end{aligned}\tag{2-2}$$

在本文的研究中铰接车的前桥和后桥相等，即 $L = l_f = l_r$ 。联合式 (2-1) 和式 (2-2) 可以得到铰接车的运动学模型，见式 (2-3)。

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta \\ \gamma \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ v \tan(\gamma/2)/L + \omega/(\cos \gamma + 1) \\ \omega \end{bmatrix} \tag{2-3}$$

其中， v, ω 是输入量，分别代表这前轴速度和铰接角速度。详细的运动学模型推导过程可以参考文献 [40]。

接着可以推导出铰接车的转弯半径为：

$$R = \frac{v}{\dot{\theta}} = v \cdot L \frac{1 + \cos \gamma}{v \sin \gamma + L\omega} \tag{2-4}$$

特别的，如果忽略铰接车的动态转向，只保留静态转向，即 $\omega = 0$ ，式 (2-4) 可以重写为：

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

2.2 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。

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，性能代价取最小值，此刻满足条件的控制称为最优控制 $u^*(t)$ ，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^[41]。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：

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

其中， $\Phi[x(t_f)]$ 为终点 $x(t_f)$ 的代价， $L[x(t), u(t), t]$ 为时间 t 时刻的过程代价。 $\dot{x}(t) = f(x(t), u(t))$ 为车辆的运动学方程， x_{min} 和 x_{max} 分别为状态变量的下边界和上边界。 u_{min} 和 u_{max} 分别为输入变量的下边界和上边界。 $x(t_0) = x_{start}, x(t_f) = x_{goal}$ 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。

对于离散系统，一般代价泛函可以表示为：

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

其中， $\Phi[x(N)]$ 为终点 $x(N)$ 的代价函数， $L[x(k), u(k), k]$ 为第 k 步的过程代价函数。 $x(k)$ 和 $u(k)$ 表示第 k 步的状态量和控制量， N 是控制区间长度， J 是最优控制的目标函数。

2.3 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，即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。本文使用数值解法，因为后续提出的策略可以有效解决高维度下求解难的问题。

2.3.1 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^[42] 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。

2.3.2 基于配点法的数值求解

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^[43]。由于直接将状态量纳入到优化变量，得到的最优解就是所求轨迹，所以配点法适合用来自做轨迹规划。这种方法在配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插值。当配点较为密集时，配点之间也可以近似用直线连接。

本文中使用配点法求解，可以同时求解状态量和控制量。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：

$$\begin{aligned}
 & \min_{x(k), u(k)} J = \Phi[x(N)] + \sum_{k=0}^{N-1} L[x(k), u(k), k] \\
 & s.t. \quad x(k+1) = x(k) + \Delta t \cdot f(x(k), u(k)) \\
 & \quad x_{min} \leq x(k) \leq x_{max}, u_{min} \leq u(k) \leq u_{max}, k \in 0, N-1 \\
 & \quad x(0) = x_{start}, x(N) = x_{goal} \\
 & \quad h(x(k)) \cap C_{occ} = \emptyset
 \end{aligned} \tag{2-8}$$

其中， N 为离散化的区间数目。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）。

针对全局规划和局部规划对式（2-8）设计各自的具体目标函数和约束形式会在第三章和第四章中详细介绍，并使用非线性优化求解器进行求解。非线性优化求解器将在下一节中介绍。

2.4 非线性规划问题的求解

上一节中把最优控制离散化为了一个非线性规划（NLP）问题，这个问题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）^[44]、交叉方向乘子法（ADMM）^[45] 和罚函数法^[46] 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数值稳定性很高，下文设计中将使用罚函数。

2.4.1 罚函数法

对于带约束的优化问题一般可以使用罚函数把它化为无约束优化问题。无约束问题的求解有梯度法和牛顿法，两类方法都可以有效求解。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^[47] 和 G2O^[48]。考虑如下一般的优化问题：

$$\begin{aligned}
 & \min_x J(x) \\
 & s.t. g(x) < 0 \\
 & \quad h(x) = 0
 \end{aligned} \tag{2-9}$$

其中, $g(x)$ 为不等式约束函数, $h(x)$ 为等式约束函数。把这两种约束使用罚函数加到代价函数中得到:

$$\min_x J(x) + Pen1(g(x)) + Pen2(h(x)) \quad (2-10)$$

其中, 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。针对罚函数 $Pen1$ 和 $Pen2$ 的具体形式, 会在后续文中进行合理设计。

2.4.2 非线性规划问题的初始解

罚函数是将带约束的问题转化为了无约束优化问题, 求解无约束优化问题需要一个迭代初始解。在凸优化中, 初始解的质量至关重要, 问题的凸特性会使得初始解向最优解迭代, 初始解的不同只是影响收敛快慢而已。但是在本文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可以使得最终的求解结果接近最优解。图2-2为不同初始轨迹与最后收敛的轨迹比较图。图中灰色区域是设置的障碍物, 虚线表示初始解, 实线为优化后的局部最优轨迹。

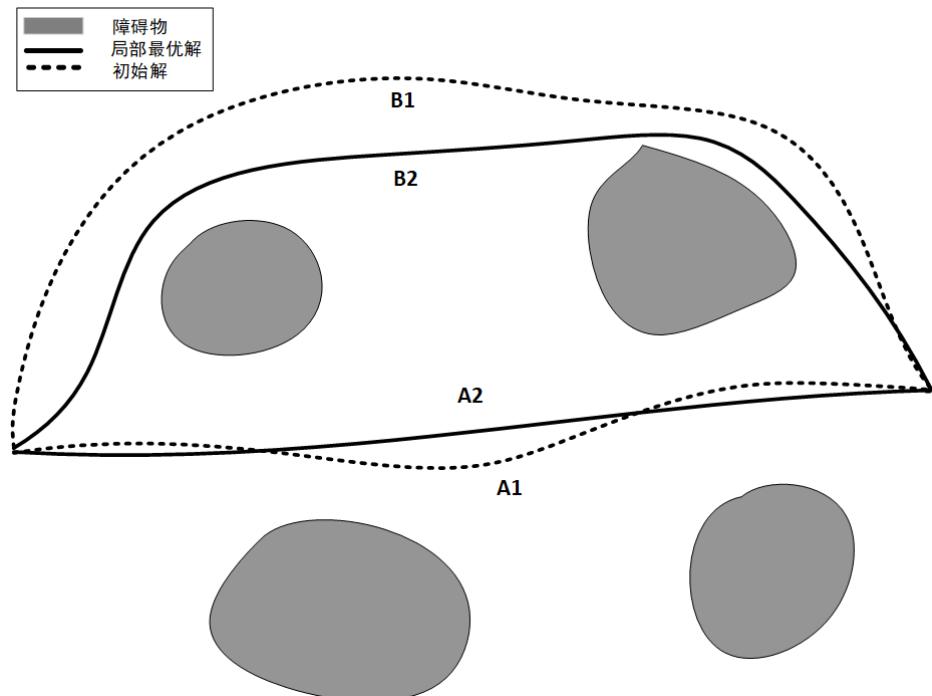


图 2-2 不同初始轨迹与最后收敛的轨迹比较

可以看出，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。

2.5 本章小结

本章系统构建了铰接车运动规划的理论框架与求解体系，并通过全局-局部分层规划架构实现任务分解。

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，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 基于微分平坦的无人铰接车全局轨迹规划方法

在第二章介绍了铰接车的运动学模型，并将轨迹规划统一表述为了最优控制问题。在传统的全局规划问题中，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。这些全局规划算法针对全向或者差速机器人会非常有效，然而，它们并不会考虑车身的朝向和曲率等信息。因此，面对具有非完整性的机器人来说并不适合。如图3-1，是期望路径和铰接车实际轨迹的对比图。

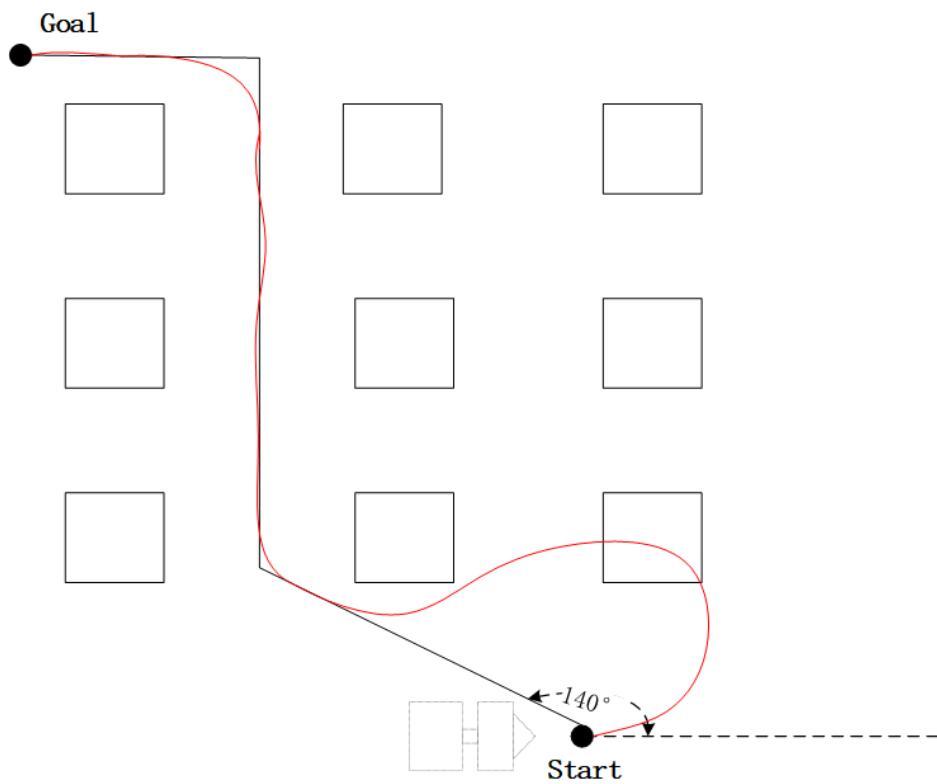


图 3-1 期望路径和实际轨迹对比图

考虑如图3-1的情况，即需要规划起点 Start 到终点 Goal 的路径。黑色路径代表了 A* 算法所规划的理想路径，此路径并没有考虑车辆的初始姿态。实际上，车体的初始朝向与理想路径的起始切线方向之间存在 140 度的偏差。当车辆尝试沿着这条预设路径行进时，由于初始角度的巨大差异，车辆行驶出了红色的轨迹。最终导致追踪效果极不理想，撞击了障碍物。

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

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。微分平坦系统具有如下特征^[49]：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，则这个系统为微分平坦系统。

对于铰接车而言，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可以使用简化的运动学模型，把需要精细调整的部分交给局部规划来处理，以节省计算资源。在这里忽略动态转向，只保留静态转向模型。令 v 和 γ 为系统输入，运动学方程（2-3）可以简化为：

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

对于上述的铰接车的运动模型，选择 x 和 y 作为平坦输出，这在轨迹规划中可以带来很好的性质。 x 和 y 也是需要的轨迹，具有实际的物理意义。下面给出铰接车的微分平坦系统：

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

其中， x 和 y 是系统的平坦输出变量， \dot{x} 和 \dot{y} 是平坦输出变量关于时间的一阶导数， \ddot{x} 和 \ddot{y} 是平坦输出变量关于时间的二阶导数。 (v, θ, γ, k) 是选取的状态变量。 k 为轨迹的曲率，也是铰接角的函数。 $n \in \{-1, 1\}$ 是车辆运动的方向，分别代表后退和前进两种情况。

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^[50]。这一小节将介绍 minco 曲线的原理，并且运用它处理铰接车的运动规划问题。图3-2为 Minco 样条曲线的可视化图。图中 q_0, \dots, q_N 为轨迹航点， $\beta(t_i)$ 为第 i 段多项式轨迹。 $N + 1$ 个航点一共可以生成 N 段多项式轨迹。

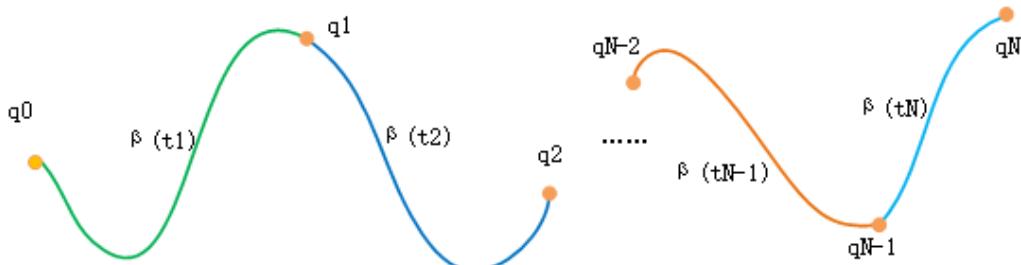


图 3-2 Minco 样条曲线

3.2.1 Minco 样条曲线基本原理

在针对 minimumJerk (最小化加加速度) 形式的优化问题中，其最优解通常表现为一个五次多项式。这种轨迹可以通过多个多项式曲线拼接而成，在自动驾驶技术领域被广泛应用。根据文献 [50] 所述，对于微分平坦的系统而言，在平坦输出条件下求得的最优轨迹会表现出更高阶次的连续性。

例如，minimumJerk 问题的最优解体现为一个五次多项式。特别地，当系统的运动学模型呈现出微分平坦特性，并使用平坦变量来构建 minimumJerk 问题，Minco 理论进一步证实了平坦变量在 snap (即加速度的变化率) 这一阶次上具有连续性。由此，基于给定的起点、终点构建边值约束，再确保中间航点满足至 snap 阶的连续性条件，可以得出轨迹的闭式解。这种特定的闭式解被称为 Minco 样条曲线。

接下来，将详细探讨在平坦输出下的两种轨迹优化方法——边值问题 (BVP, Boundary Value Problem) 和边界与中间值问题 (BIVP, Boundary and Intermediate Value)。

(1) BVP

在多项式规划中一个典型的 BVP 问题可以表示为：

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

其中， $z(t)$ 是样条曲线，它可以是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。 \mathbf{W} 是输入代价的权重矩阵。 \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^[50]。

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

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

其中， $\mathbf{c} = [c_0 \ c_1 \ c_2 \ c_3 \ c_4 \ c_5]^T$ 是多项式的系数向量，由各阶系数 (c_0, \dots, c_5) 组成。 $\boldsymbol{\lambda}(t) = [1 \ t \ t^2 \ t^3 \ t^4 \ t^5]^T$ 。

假设轨迹的时间段是 T ，起点和终点带来的约束可以写为：

$$\begin{bmatrix} x_s \\ v_s \\ a_s \\ x_e \\ v_e \\ a_e \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix}}_{\mathbf{A}_F(t)} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}}_c \quad (3-5)$$

其中， 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 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：

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

其中， t_i 为时间变量， t_0 是起点时刻， t_N 是终点时刻。 \mathbf{W} 是输入代价的权重矩阵。 N 为 BIVP 的区间数目，也就是多项式轨迹的段数。其他变量和 BVP 中描述的一致。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。例如，在处理 minimumJerk (s 等于 3) 的轨迹时，这个轨迹具有 snap 阶 (s 等于 4) 阶的连续性^[50]。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 $N + 1$ 个航点可以生成 N 段五次多项式轨迹，每个多项式有 6 个系数，一共有 $6N$ 个系数。这意味着需要 $6N$ 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

假设共有 $N+1$ 个航点 \mathbf{q}_0 至 \mathbf{q}_N ，其中起点 $\mathbf{q}_0 = [x_0, y_0, v_{0x}, v_{0y}, a_{0x}, a_{0y}]$ 和终点 $\mathbf{q}_N = [x_N, y_N, v_{Nx}, v_{Ny}, a_{Nx}, a_{Ny}]$ 完全已知。 $\mathbf{q}_1 = [x_1, y_1]$ ，一直到 $\mathbf{q}_{N-1} = [x_{N-1}, y_{N-1}]$ 。

由此，一共可以得到 6 个约束方程，起点的约束方程为：

$$\underbrace{\begin{bmatrix} x_0 & y_0 \\ v_{0x} & v_{0y} \\ a_{0x} & a_{0y} \end{bmatrix}}_{\alpha_0} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}}_{F_0} \underbrace{\begin{bmatrix} c_{0x}^1 & c_{0y}^1 \\ c_{1x}^1 & c_{1y}^1 \\ c_{2x}^1 & c_{2y}^1 \\ c_{3x}^1 & c_{3y}^1 \\ c_{4x}^1 & c_{4y}^1 \\ c_{5x}^1 & c_{5y}^1 \end{bmatrix}}_{c_1} \quad (3-7)$$

其中， c_{jx}^1 为 x 方向上第一段多项式轨迹的第 j 个系数， c_{jy}^1 为 y 方向上第一段多项式轨迹的第 j 个系数。

终点的约束方程为：

$$\underbrace{\begin{bmatrix} x_N & y_N \\ v_{Nx} & v_{Ny} \\ a_{Nx} & a_{Ny} \end{bmatrix}}_{\alpha_N} = \underbrace{\begin{bmatrix} 1 & T_N & T_N^2 & T_N^3 & T_N^4 & T_N^5 \\ 0 & 1 & 2T_N & 3T_N^2 & 4T_N^3 & 5T_N^4 \\ 0 & 0 & 2 & 6T_N & 12T_N^2 & 20T_N^3 \end{bmatrix}}_{F_N} \underbrace{\begin{bmatrix} c_{0x}^N & c_{0y}^N \\ c_{1x}^N & c_{1y}^N \\ c_{2x}^N & c_{2y}^N \\ c_{3x}^N & c_{3y}^N \\ c_{4x}^N & c_{4y}^N \\ c_{5x}^N & c_{5y}^N \end{bmatrix}}_{c_N} \quad (3-8)$$

其中， c_{jx}^N 为 x 方向上第 N 段多项式轨迹的第 j 个系数， c_{jy}^N 为 y 方向上第 N 段多项式轨迹的第 j 个系数。

3.2.3 航点的连续性约束方程

除了已知起点和终点之外，还必须保证 BIVP 问题 $N - 1$ 个中间航点的连续性。根据 Minco 的理论，minimumJerk 问题 (s 等于 3) 具有 snap 阶的连续性。对于每一

段多项式轨迹，可以求出它的各阶导数为：

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

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：

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

其中 T_i 为第 i 段多项式轨迹的时间区间， c_i 为第 i 段多项式轨迹的系数向量。由于一共有 $N-1$ 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 $5N-5$ 个。

3.2.4 轨迹的闭式解

令中间航点为 $\mathbf{q}_1(x_1, y_1), \mathbf{q}_2(x_2, y_2), \dots, \mathbf{q}_{N-1}(x_{N-1}, y_{N-1})$ 。如果中间航点是已知的，则可以带来 $N-1$ 个航点的位置约束：

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

现在将中间航点位置约束和连续性约束写在一起：

$$\begin{bmatrix} \mathbf{E}_i & \mathbf{F}_i \end{bmatrix} = \begin{bmatrix} \beta(T_i) & G(T_i) \\ 0 & -G(0) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_i \\ 0 \end{bmatrix} \quad (3-12)$$

综合起点、终点和航点的约束可以得到 $6N$ 个约束方程：

$$\underbrace{\begin{bmatrix} \mathbf{F}_0 & 0 & 0 & \cdots & 0 \\ \mathbf{E}_1 & \mathbf{F}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{E}_2 & \mathbf{F}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{F}_{N-1} \\ 0 & 0 & 0 & \cdots & \mathbf{E}_N \end{bmatrix}}_M \underbrace{\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \vdots \\ \mathbf{c}_{N-1} \\ \mathbf{c}_N \end{bmatrix}}_c = \underbrace{\begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{q}_1 \\ 0 \\ \vdots \\ 0 \\ \boldsymbol{\alpha}_N \end{bmatrix}}_{\boldsymbol{\alpha}} \quad (3-13)$$

其中， \mathbf{M} 为维度为 $2N \times 6N$ 的系数方阵，由 \mathbf{M} 和 $\boldsymbol{\alpha}$ 可以唯一确定一组多项式系数 \mathbf{c} ，即轨迹的参数形式。

值得一提的是，在 \mathbf{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 \mathbf{M} 就是完全已知的矩阵了。由于本章聚焦于路径规划，所以把时间参数 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需要解决，就是障碍物躲避。

鉴于施工现场环境复杂多变，地面上可能会出现各种不可预知的障碍物。为了充分利用栅格地图快速索引的优势并增强方案的通用性，本文选用二维空间下的占据栅格地图（Occupancy Grid Map）作为主要的地图类型进行路径规划。为实现高效的避障，本文采用了一种基于安全走廊的避障方法方法。

安全走廊由一个个凸包构成，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。在本文中，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^[23] 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。Li 等

人^[51]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。Deits 等人^[52]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。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^[53]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。

本文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 [53]。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。图3-3是矩形走廊和不规则凸包走廊的对比图。图3-4描述了凸包的线性约束表示方法。

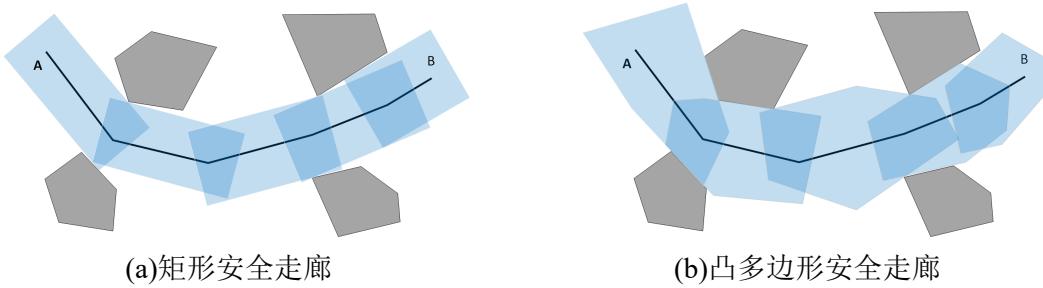


图 3-3 两种常见的安全走廊比较图

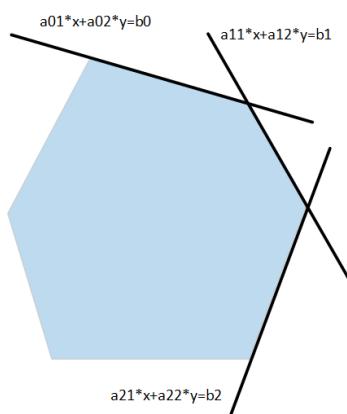


图 3-4 凸包与半空间约束关系图

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，在凸包内的点必然满足以下不等式约束：

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

$$\mathbf{A} = [\mathbf{a}_0, \dots, \mathbf{a}_z, \dots, \mathbf{a}_{n_z-1}]_{(n_z-1) \times 2}^T, \mathbf{b} = [b_0, \dots, b_z, \dots, b_{n_z-1}]_{(n_z-1) \times 1}^T$$

其中， \mathbf{p} 为凸包内的点， \mathbf{a}_i 是第 i 个半空间的法向量， \mathbf{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式 (3-14) 带来的不等式约束。

3.4 点到点全局路径规划

在 3.2 节中，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，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生成的路径常常包含尖锐拐角，这对具有转向半径限制的车辆而言难以准确跟踪。因此，在进行轨迹规划时，必须充分考虑转向半径的约束条件。这样做不仅避免了路径中出现车辆无法执行的急转弯，还提升了路径的整体实用性和车辆运行的安全性，确保所规划的路径符合车辆的实际物理特性。

对于铰接式车辆来说，其最大曲率主要由静态转向模型决定，这意味着车辆能够达到的最大转向角度是可以精确计算出来的。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：

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

其中， R_{min} 为最小转弯半径， γ_{max} 为最大铰接角。本文铰接车模型的最大转向角度为 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 $\mathbf{p} = [x, y]^T$, 则:

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

$$\begin{aligned} \frac{\partial k}{\partial \ddot{\mathbf{p}}} &= \frac{\mathbf{B}^T \ddot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|_2^3} - 3 \frac{\ddot{\mathbf{p}}^T \mathbf{B} \dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|_2^5} \dot{\mathbf{p}} \\ \frac{\partial k}{\partial \dot{\mathbf{p}}} &= \frac{\mathbf{B} \dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|_2^3} \end{aligned} \quad (3-18)$$

3.4.2 优化问题的建立

根据式 (3-13), 轨迹的参数可以由两个要素唯一确定: 航点向量 α 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 α 直接做为优化变量是很自然的选择。同时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:

$$\begin{aligned} &\min_{q_1, \dots, q_{N-1}} J(\mathbf{q}) \\ s.t. \quad &-k_m - k_{ij} \leq 0 \\ &k_{ij} - k_m \leq 0 \\ &\mathbf{A}_i \mathbf{p}_{ij} - \mathbf{b}_i < 0, \quad i \in [1, N], j \in [1, m_i] \end{aligned} \quad (3-19)$$

其中, k_m 是最大曲率, k_{ij} 的计算方法见公式3-17。 N 是轨迹的段数, m_i 是第 i 段轨迹的离散点数。 $\mathbf{p}_{ij} = \beta^T (\frac{T_i}{m_i} \cdot j) \cdot \mathbf{c}_i$, 是第 i 段多项式轨迹的第 j 个点, \mathbf{c}_i 通过 $\mathbf{c} = M^{-1} \alpha$ 计算得到。

针对于曲率和安全这两个不等式约束, 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:

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

其中, a 是一个已知参数, 一般选取为优化问题迭代的停止条件值, 本文设置的 a 为 0.05。 $S'(x)$ 是 $S(x)$ 的导数。如图3-5, 是松弛函数和其导数图。

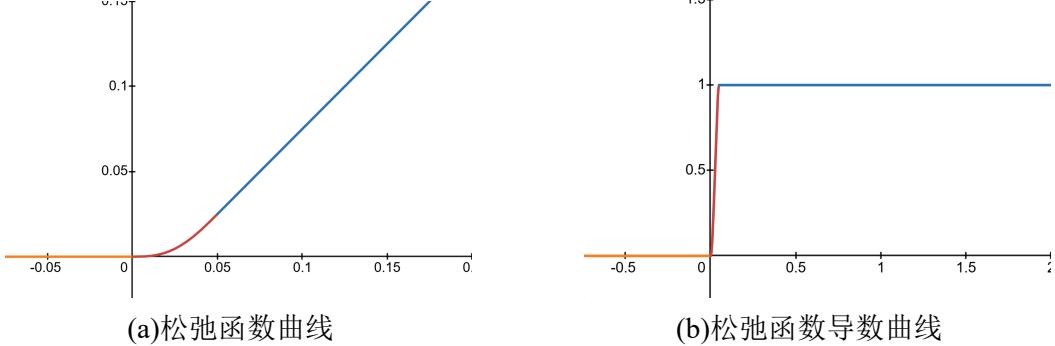


图 3-5 松弛函数和其导数图

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:

$$\min_{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{N-1}} J(\mathbf{q}) = \sum_{i=1}^N \sum_{j=1}^{m_i} w_k \cdot S(-k_m - k_{ij}) + w_k \cdot S(-k_m + k_{ij}) + w_c \cdot S(\mathbf{A}_i \mathbf{p}_{ij} - \mathbf{b}_i) \quad (3-21)$$

其中, w_k 为曲率约束违反的惩罚权重, w_c 为走廊约束违反的惩罚权重。

利用矩阵求导的链式法则可以求得各个部分的梯度信息, 注意本文矩阵求导法则遵循分母布局。

$$\begin{aligned} \frac{\partial S(-k_m - k_{ij})}{\partial \mathbf{c}_i} &= -\left(\frac{\partial \dot{\mathbf{p}}_{ij}}{\partial \mathbf{c}_i} \frac{\partial k_{ij}}{\partial \mathbf{p}_{ij}} + \frac{\partial \ddot{\mathbf{p}}_{ij}}{\partial \mathbf{c}_i} \frac{\partial k_{ij}}{\partial \ddot{\mathbf{p}}_{ij}}\right) S'(-k_m - k_{ij}) \\ \frac{\partial S(-k_m + k_{ij})}{\partial \mathbf{c}_i} &= \left(\frac{\partial \dot{\mathbf{p}}_{ij}}{\partial \mathbf{c}_i} \frac{\partial k_{ij}}{\partial \mathbf{p}_{ij}} + \frac{\partial \ddot{\mathbf{p}}_{ij}}{\partial \mathbf{c}_i} \frac{\partial k_{ij}}{\partial \ddot{\mathbf{p}}_{ij}}\right) S'(-k_m + k_{ij}) \\ \frac{\partial S(\mathbf{A}_i \mathbf{p}_{ij} - \mathbf{b}_i)}{\partial \mathbf{c}_i} &= \frac{\partial \mathbf{p}_{ij}}{\partial \mathbf{c}_i} \mathbf{A}_i^T S'(\mathbf{A}_i \mathbf{p}_{ij} - \mathbf{b}_i) \end{aligned} \quad (3-22)$$

其中 $\dot{\mathbf{p}}$ 和 $\ddot{\mathbf{p}}$ 分别为 \mathbf{p} 对时间的一阶、二阶导数。利用上式求出了 J 关于 \mathbf{c} 的梯度, 于是由 $\alpha = M\mathbf{c}$, 可以推导出: $\frac{\partial \alpha}{\partial \mathbf{q}} M^{-T} = \frac{\partial \mathbf{c}}{\partial \mathbf{q}}$, 因此 J 关于 \mathbf{q} 的梯度为:

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

3.4.3 起点和终点航向角保证

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 $\mathbf{q}_0 = [x_0, y_0, v_{0x}, v_{0y}, a_{0x}, a_{0y}]$ 和终点 $\mathbf{q}_N = [x_N, y_N, v_{Nx}, v_{Ny}, a_{Nx}, a_{Ny}]$ 。其中, 加速度的分解值令其为 0, 这在路径规划中是合理的。但是对于速度的分解还需要详细讨论, 因为其涉及到航向角 θ 。

由运动学模型可以求得 x 和 y 方向上的速度分量:

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

其中 θ_s 是铰接车的起始航向角。对于初始速度 v_s 不为零的情况, v_{0x}, v_{0y} 都不等于零, 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 $v_{0x} = 0, v_{0y} = 0$ 。此时用平坦模型计算 θ_s :

$$\theta_s = \arctan 2(v_{sy}, v_{sx}) = \arctan 2(0, 0) \quad (3-25)$$

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, 需要给 v_s 加上一个微小正量 ϵ 保证它的有效性:

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

这样就可以保证车辆的起点朝向施加在轨迹上, 同样由于 ϵ 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^[54]。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。综合众多因素本文选择使用 LBFGS 优化算法最为本优化问题的优化求解算法。

由于使用优化的方法生成轨迹，所以需要一个快速生成航点的路径规划算法，为算法提供软起动初值。对于这个路径规划算法一般并不会去关心是否最优，但是它需要以较低的时间复杂度运行，因为优化过程可以带来足够的最优性。在本文中使用 RRT-Connect 生成路径。区别于传统的 RRT 算法，RRT-Connect 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。局部规划算法的步骤如下：

第一，初始解的构造。使用 RRT-Connect 算法搜索出从起点到终点的一条路径 $\hat{\mathbf{path}}$ ，其中包含 $N + 1$ 个路径点。

$$\hat{\mathbf{path}} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \dots \\ x_N & y_N \end{bmatrix} \quad (3-27)$$

由此可以构造初始解 $\bar{\mathbf{q}}_{init} = [\mathbf{q}_1 \ \mathbf{q}_2 \ \dots \ \mathbf{q}_{N-1}]^T$ 。

第二，构造系数矩阵 \mathbf{M} 。将时间变量 T_i 归一化，则 \mathbf{M} 矩阵是完全已知的。对于系 α 中的 α_0 和 α_N 可以通过起点 \mathbf{q}_0 和终点 \mathbf{q}_N 构建。

第三，构造安全走廊。沿着路径 $\hat{\mathbf{path}}$ 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 \mathbf{q}_{opt} 。

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

第六，得到路径最优表达式。 $\mathbf{path}(t_i) = \beta(t_i)\mathbf{c}_i$ ，其中 t_i 是第 i 条轨迹多项式中的时刻， $0 \leq t_i \leq 1$ 。通过这个公式采样可以得到离散的轨迹。

3.5 多站点航行全局路径规划

在上一节中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。这一节将针对码头搬运场景设计一个周期性的规划流程，也称多站点航行全局路径规划。图3-6展示的是一个多站点搬运示意图。

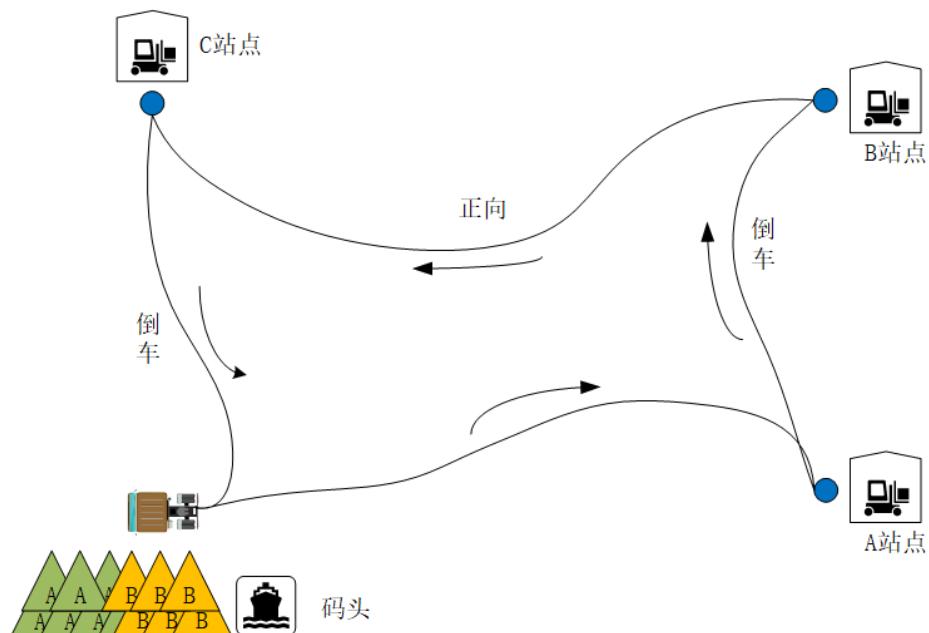


图 3-6 多站点场景下全局规划示意图

3.5.1 多站点航行规划的问题描述

考虑图3-6的搬运问题，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 点、B 点和 C 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

如图3-7和图3-7，在中间站点处的是否换向包含两种情况。

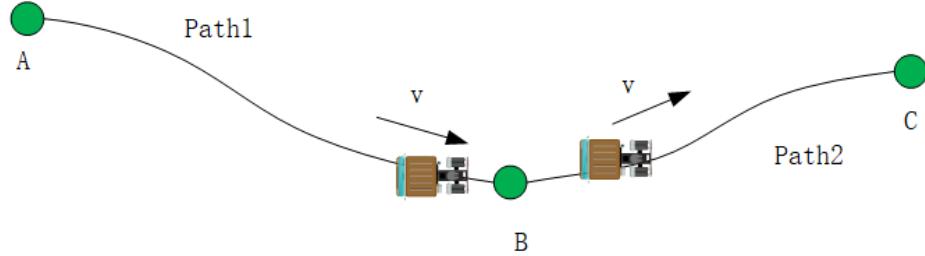


图 3-7 B 点不需要换向

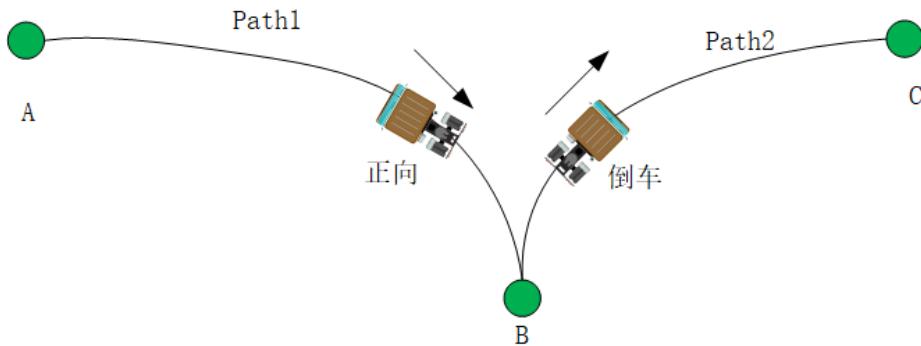


图 3-8 B 点完成换向

如图3-7和图3-8，一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。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 本章小结

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，设计基于安全走廊的生成机制，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

在第三章中介绍了铰接车的全局轨迹规划方法，在本章中将提出一种基于安全走廊的时空联合局部轨迹规划方法，全局和局部规划形成一个完整的运动规划系统。首先，将对铰接车的几何结构进行建模，可以有效加强碰撞检测的速度。其次，针对铰接车提出一种改进的 Hybrid A* 算法，它将用于局部轨迹规划算法的热启动。接着，对规划问题中涉及到的所有约束并进行处理。最后，针对最终的优化问题进行求解。

4.1 铰接车的几何结构建模

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，从而将铰接车的避障问题转化为多边形几何碰撞问题。如图4-1所示，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 O₁。

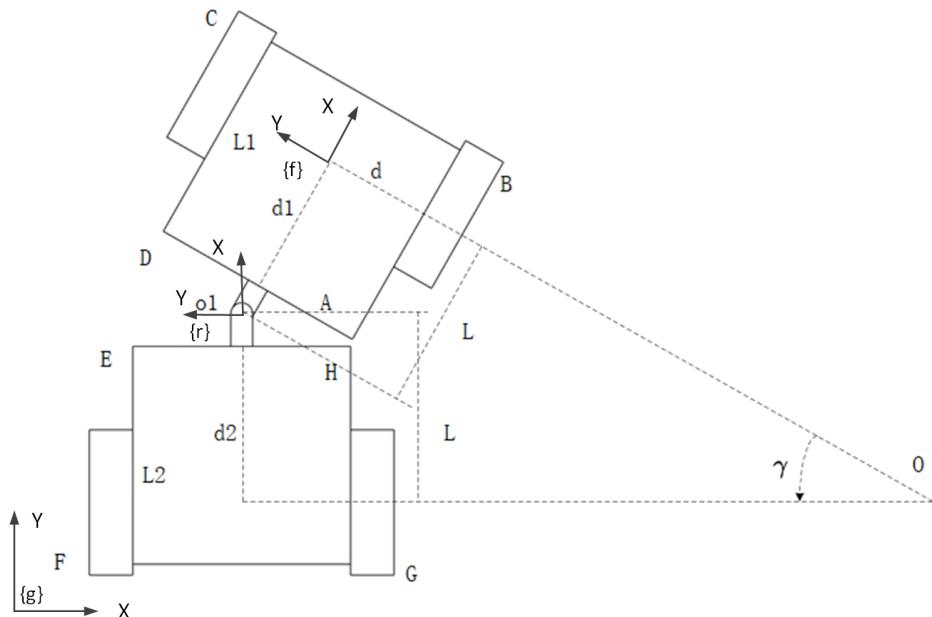


图 4-1 铰接式车辆结构建模

图中 $\{f\}$ 、 $\{r\}$ 和 $\{g\}$ 分别为前桥坐标系、后桥坐标系和世界坐标系。与动学模型相一致，选取前桥中点作为参考点 (x, y) ，沿着车体方向作为 x 轴正方向。则 A、B、C、D、O₁ 相对于前桥坐标系 $\{f\}$ 和 E、F、G、H 的相对后桥坐标系 $\{r\}$ 的坐标

可以表示为：

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

其中， $width$ 为 BC 的长度。令 \mathbf{R}_θ 和 $\boldsymbol{\sigma}$ 为世界坐标系到车辆坐标系的旋转矩阵和平移向量，根据4-1，则前桥顶点在世界坐标系下表示为：

$$\varepsilon_f = \{\mathbf{p}_f \in \mathbb{R}^2 \mid \mathbf{p}_f = \mathbf{R}_\theta l_f + \boldsymbol{\sigma}, f = A, B, C, D\}, \tag{4-2}$$

其中， $\boldsymbol{\sigma}$ 也是前轴参考点坐标，表示为 $\boldsymbol{\sigma} = (x, y)$ 。

l_E, l_F, l_G, l_H 是相对于后桥坐标系下的坐标。在从世界坐标系到后桥坐标系的变换中，还涉及到前桥坐标系，需要应用旋转矩阵 \mathbf{R}_θ 和 \mathbf{R}_γ 。平移向量由世界坐标系中的铰接点给出，记为 $\boldsymbol{\sigma}_1$ 。可以得到后桥的坐标变换为：

$$\begin{aligned}
 \mathbf{p}_r &= \mathbf{R}_\gamma \mathbf{R}_\theta l_b + \boldsymbol{\sigma}_1 \\
 \boldsymbol{\sigma}_1 &= \mathbf{R}_\theta l_{O_1} + \boldsymbol{\sigma}
 \end{aligned} \tag{4-3}$$

则后桥的顶点集表示如下：

$$\varepsilon_r = \{\mathbf{p}_r \in \mathbb{R}^2 \mid \mathbf{p}_r = \mathbf{R}_\gamma \mathbf{R}_\theta l_r + \mathbf{R}_\theta l_{O_1} + \boldsymbol{\sigma}, r = E, F, G, H\}. \tag{4-4}$$

其中， $\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$, \mathbf{R}_γ 同理。

4.2 铰接车 Hybrid A* 算法的改进

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。

为了能够适应铰接车的路径规划问题，本节将提出一种改进 Hybrid A* 算法。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，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

假设铰接车当前的状态为 $x_0(x, y, \theta, \gamma)$ ，一般 Hybrid A* 算法将控制区间 v_{min} 至 v_{max} 和 ω_{min} 至 ω_{max} 进行离散化。并从当前状态 x_0 开始以输入 v 和 ω 在 Δ_t 时间内进行运动学推演得到下一个状态，运动学推演见式 (2-3)。图4-2是原始 Hybrid A* 算法运动学采样策略和改进后的采样策略比较。

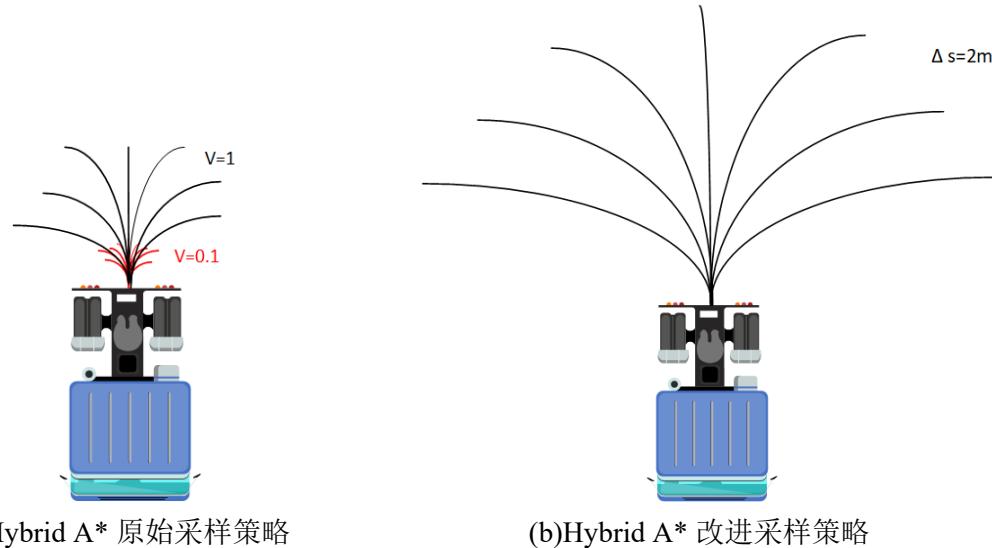


图 4-2 Hybrid A* 原始采样策略和改进采样策略

如图4-2(a)，列出了速度为 0.1 和 1 采样点。可以看出这种采样方式在 $|v|$ 很小时，不管 ω 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。由于系统的输入是二维 (v 和 ω)，所以算法的时间复杂度是 $O(N^2)$ 。下面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。

首先为了降低计算的复杂度，使用铰接车的静态转向模型，系统的输入为 v 和 γ_u 。不再以固定的时间 Δt 进行采样，以固定的行驶路程 Δs 采样，于是系统的输入为 γ_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， Δs 是采样的路径步长， x 、 y 、 θ 和 γ 是铰接车的位形空间状态变量。

图4-2(b)为 Δs 等于 2 m 的采样过程。可以看出相对于原始推演策略来说这种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 g 和启发式函数代价 h 两个部分，见式 (4-6)。

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

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。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。

采样策略的输入变量是 γ_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。所以在代价函数中必须含有 $\Delta\gamma$ 项使得铰接角变化率尽量小，称之为转向惩罚。因此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的距离代价，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，使用无穷范数作为避障启发式代价，见式 (4-8)。

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

4.2.3 碰撞检测算法设计

在碰撞检测算法的设计中，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，采用 Bresenham 算法^[55] 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(a)为铰接车没有和障碍物发生碰撞的场景，图4-3(b)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。

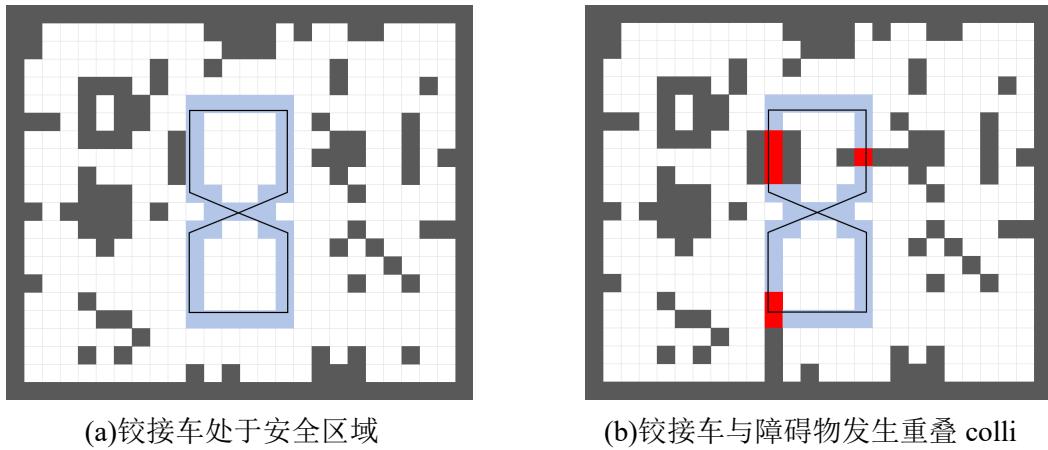


图 4-3 铰接车在环境中的碰撞示意图

4.3 基于 NMPC 和安全走廊的轨迹规划

在 4.2 节中针对铰接车对 Hybrid A* 算法进行了改进，它将作为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：

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

其中， $\dot{x}(t) - f(x(t), u(t)) = 0$ 表示运动方程约束，为了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。将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。定义如下形式的解向量形式：

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

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

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，和第三章中式(3-14)一样，避障约束是不等式的形式。假设 \mathbf{g}_f 为铰接车前桥的不等式约束函数， \mathbf{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：

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

其中， $\hat{\xi} = [\xi_\gamma^T, \xi_v^T, \xi_a^T, \xi_\omega^T, \xi_{jerk}^T]^T$ ， \mathbf{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

在第二章中介绍了铰接车的运动学模型，模型的状态量为 x 、 y 、 θ 和 γ ，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 ω ，它们并没有受到微分方程的约束，这意味着 v 和 ω 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，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 ω （铰接角速度）。运动学模型可以进一步写为 $\dot{x} = f(x, u)$ ：

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta \\ \gamma \\ v \\ a \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ v \tan(\gamma/2)/L + \omega/(\cos \gamma + 1) \\ \omega \\ a \\ jerk \end{bmatrix} \quad (4-13)$$

其中，令 $\mathbf{x} = [x, y, \theta, \gamma, v, a]^T$ 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 $\mathbf{u} = [jerk, \omega]^T$ 给出，其中 *jerk* 为前轴中点加加速度（舒适度）， ω 为铰接角速度。

对于上述运动学方程需要对其进行离散化，采用一阶离散运动学模型，可表示为：

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

令 $\mathbf{g}_{kin} = [\mathbf{g}_x^T \mathbf{g}_y^T \mathbf{g}_\theta^T \mathbf{g}_\gamma^T \mathbf{g}_v^T \mathbf{g}_a^T]^T$ ，其中 \mathbf{g}_{kin} 的每一个元素对应 $m = N - 1$ 个等式约束。例如：

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

使用惩罚函数将运动学约束纳入代价函数，表达式如下：

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

梯度为：

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

其中， w_{kin} 为运动学约束权重。 $\frac{\partial \mathbf{g}_{kin}^T}{\partial \xi}$ 表示所有运动约束向量的雅可比矩阵，表示为：

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

其中 $\psi_E = [\mathbf{E}_{n \times n}, \mathbf{0}_n] - [\mathbf{0}_n, \mathbf{E}_{n \times n}]$ ，其中 \mathbf{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，将探讨避障约束在笛卡尔坐标系中的表述方法。仍然采用安全走廊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，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。

如图4-4所示，灰色区域是设置的障碍物示意，黑色的是运动轨迹。轨迹上设置有多个离散点，蓝色的区域是凸多边形安全区域。每一个离散的路径点都生成一个凸多边形，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。

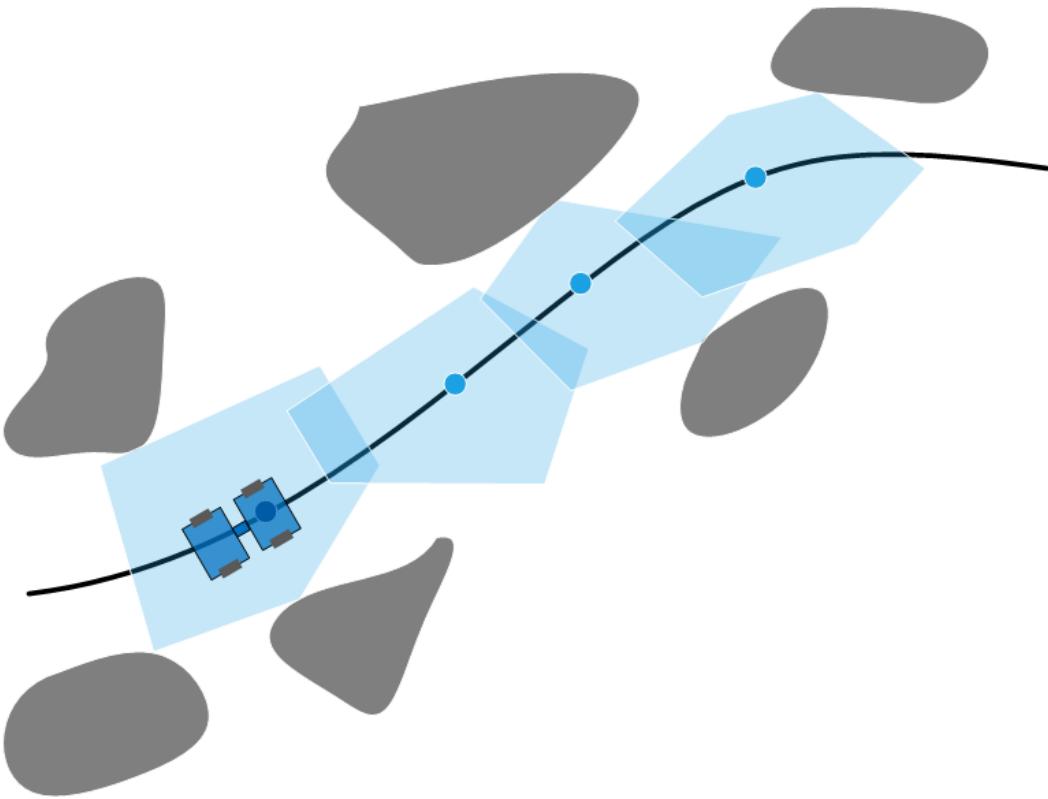


图 4-4 安全走廊示意图

铰接车辆的前桥由四个顶点构成，这四个顶点围成一个凸多边形，即为前桥的轮廓。同样地，后桥也具有类似的四个顶点，形成其自身的凸多边形轮廓。基于凸集的性质，如果将车辆的所有顶点都包含在一个大的凸包内，则可以确定整个车辆都将位于这个凸包之内。例如，在判断车辆是否能够安全通过狭窄通道或避开障碍物时，可以通过检查车辆的凸包是否与障碍物或其他限制发生重叠来进行决策。进一步，考虑到铰接式车辆在转向过程中前后桥的相对位置会发生变化，采用动态更新的凸包能更准确地反映车辆的实际形态和运动轨迹。可以得到如下安全约束：

$$\begin{cases} \mathbf{A}(\mathbf{R}_\theta \mathbf{l}_f + \boldsymbol{\sigma}) - \mathbf{b} < 0, & f = A, B, C, D \\ \mathbf{A}(\mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{l}_r + \mathbf{R}_\theta \mathbf{l}_{O_1} + \boldsymbol{\sigma}) - \mathbf{b} < 0, & r = E, F, G, H \end{cases} \quad (4-19)$$

由于铰接角的运动范围有限，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，令

$$\begin{aligned} g_f(\boldsymbol{\sigma}, \theta, \gamma) &= \mathbf{A}(\mathbf{R}_\theta \mathbf{l}_f + \boldsymbol{\sigma}) - \mathbf{b} \\ g_r(\boldsymbol{\sigma}, \theta, \gamma) &= \mathbf{A}(\mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{l}_r + \mathbf{R}_\theta \mathbf{l}_{O_1} + \boldsymbol{\sigma}) - \mathbf{b} \end{aligned} \quad (4-20)$$

其中, $\mathbf{R}_{\theta+\gamma} = \mathbf{R}_\theta \mathbf{R}_\gamma$, $\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ 。路径上第 k 个点处的走廊约束关于 σ_k, θ_k 和 γ_k 的梯度由下式给出:

$$\begin{aligned} \frac{\partial g_f^T(k)}{\partial \sigma_k} &= \mathbf{A}_k^T, \quad \frac{\partial g_f^T(k)}{\partial \gamma_k} = 0, \quad \frac{\partial g_f^T(k)}{\partial \theta_k} = \mathbf{l}_f^T \frac{\partial \mathbf{R}_\theta^T}{\partial \theta_k} \mathbf{A}_k^T \\ \frac{\partial g_r^T(k)}{\partial \sigma_k} &= \mathbf{A}_k^T, \quad \frac{\partial g_r^T(k)}{\partial \gamma_k} = \mathbf{l}_r^T \frac{\partial \mathbf{R}_{\theta+\gamma}^T}{\partial \gamma_k} \mathbf{A}_k^T \\ \frac{\partial g_r^T(k)}{\partial \theta_k} &= \mathbf{l}_r^T \frac{\partial \mathbf{R}_{\theta+\gamma}^T}{\partial \theta_k} \mathbf{A}_k^T + \mathbf{l}_{o1}^T \frac{\partial \mathbf{R}_\theta^T}{\partial \theta_k} \mathbf{A}_k^T \end{aligned} \quad (4-21)$$

对于上述走廊的不等式约束, 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, 罚函数使用式 (3-20)。为了方便表达, 定义 [] 为逐元素运算符, 例如 [·] 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:

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

其中 d 为矩阵 \mathbf{A}_k 的行数。对式 (4-22) 中的各项分别进行求导得到走廊的惩罚项的梯度:

$$\begin{aligned} \frac{\partial \mathcal{G}_{f,r}}{\partial \sigma(k)} &= \sum_{i=0}^{d-1} \mathbf{A}_k^T S' [\mathbf{g}_{f,r}(k)] = \mathbf{A}_k^T S' [\mathbf{g}_{f,r}(k)] \mathbf{I}_d, \\ \frac{\partial \mathcal{G}_f}{\partial \xi_\theta(k)} &= \mathbf{l}_f^T \frac{\partial \mathbf{R}_\theta^T(k)}{\partial \theta(k)} \mathbf{A}_k^T S' [\mathbf{g}_f(k)] \mathbf{I}_d, \\ \frac{\partial \mathcal{G}_r}{\partial \xi_\gamma(k)} &= \mathbf{l}_r^T \frac{\partial \mathbf{R}_{\theta+\gamma}^T(k)}{\partial \gamma(k)} \mathbf{A}_k^T S', \\ \frac{\partial \mathcal{G}_r}{\partial \xi_\theta(k)} &= \left(\mathbf{l}_r^T \frac{\partial \mathbf{R}_{\theta+\gamma}^T(k)}{\partial \theta(k)} + \mathbf{l}_{o1}^T \frac{\partial \mathbf{R}_\theta^T(k)}{\partial \theta(k)} \right) \mathbf{A}_k^T S' [\mathbf{g}_r(k)] \mathbf{I}_d, \\ \boldsymbol{\sigma}(k) &= [\xi_x(k), \xi_y(k)]^T \cdot [\mathbf{g}_r(k)] \mathbf{I}_d \end{aligned} \quad (4-23)$$

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, 所以需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, 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。为了表示方便后序将省略下标, 变量边界约束对应的罚函数项和其

梯度如下：

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

其中， w_b 是边界约束的权重， \mathbf{I}_{N-2} 表示元素全为 1 的 $N-2$ 维向量。

4.3.5 时间正则化约束

在本文的轨迹优化问题中，每两个轨迹点之间的时间 Δt 是灵活变化且独立的，纳入了优化变量当中。时间正则化约束和走廊约束一样属于不等式约束，与之不同的是，时间项 Δt 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 $\Delta t \leq 0$ ，这并不符合物理规律。

为了解决以上问题，本文使用一个 sigmoid 变换将 Δt 的取值范围区间映射到映射到负无穷到正无穷，变换后的变量记为 τ 。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^[50]。

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

其中， t_{\max} 为时间变量的上界。 $-\infty < \tau < +\infty$ ， $0 < \Delta t < \Delta t_{\max}$ 。

4.3.6 优化问题求解

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，对于时间变量被映射到了整个实数轴，最后所有约束都被处理，问题被化为一个无约束优化的形式

由于时间变量用微分同胚进行了转换，需要重新表示优化变量。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 ξ_τ 和最初优化变量 ξ 中的 ξ_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：

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \xi_{jerk}} &= 2w_{uj} \cdot \xi_{jerk} \\ \frac{\partial \mathcal{J}}{\partial \xi_\omega} &= 2w_{u\omega} \cdot \xi_\omega \\ \frac{\partial \mathcal{J}}{\partial \xi_\tau} &= \Delta t [\xi_\tau]\end{aligned}\quad (4-27)$$

在式 (4-22) 推导出了运动学约束的雅可比矩阵。现在把对时间变量的导数部分进行替换，得到：

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

其中， $\mathcal{G}_{kin}, \mathcal{G}_c, \mathcal{G}_b$ 由式 (4-16)、式 (4-22) 和式 (4-24) 提供。综上所述，优化问题被转化为了一个无约束的形式，可以使用 LBFGS 算法求解。

4.4 本章小结

在本章中，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，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。

在轨迹优化方面，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。

通过本章的研究，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

在前面章节中，本文全面介绍了全局路径规划和局部轨迹规划两个算法，并分别对两个算法进行了严格的理论分析。在这一章节中将把这两个算法集成到自主导航中，组合成一个完整的运动规划系统。并通过大量的仿真分别对这两个算法进行详细的分析和验证，展示整个运动规划系统的能力。

表5-1给出了车体的仿真参数设置，仿真中都将使用栅格地图最为环境信息，地图的分辨率是 0.3 m/pix。

表 5-1 System Parameters

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

5.1 仿真环境及运动规划软件架构设计

本文的实验仿真环境搭载 AMD-Ryzen9-7940HX 处理器，基于 Ubuntu20.04LTS 操作系统构建 ROS-Noetic 机器人系统框架。仿真可视化界面使用 RVIZ，其仿真界面示意图见5-1。核心算法采用 C++14 标准和 Eigen 矩阵库实现，并通过 Catkin 构建系统实现节点编译与依赖管理。

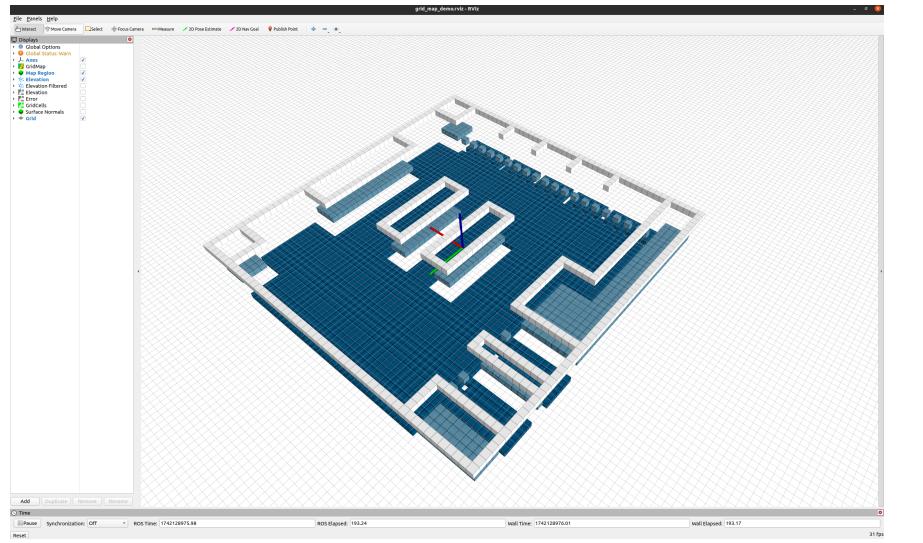


图 5-1 RVIZ 仿真可视化示意图

受开源机器人框架 ROS 的启发，本文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。下图 5-2 为运动规划系统的算法架构图：

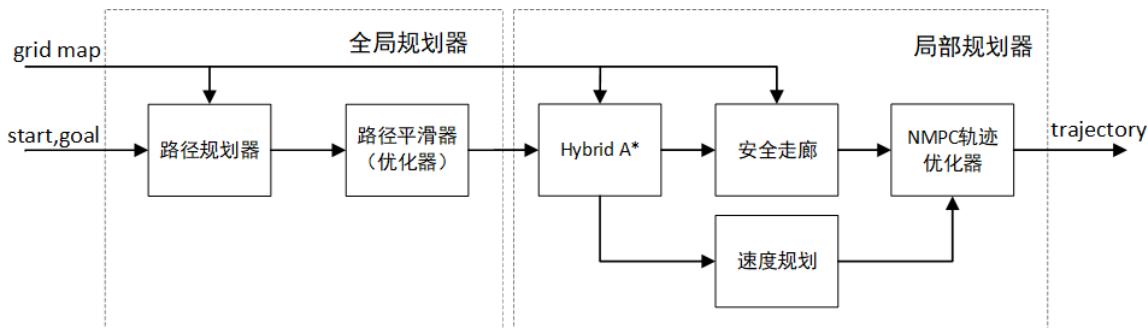


图 5-2 运动规划算法架构图

在 ROS 中，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。它通过自适应蒙特卡洛定位 (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。全局代价地图和局部代价地图更新环境信息，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。图5-3为 navigation 的规划软件架构。如图5-4是本文改进的规划框架图。

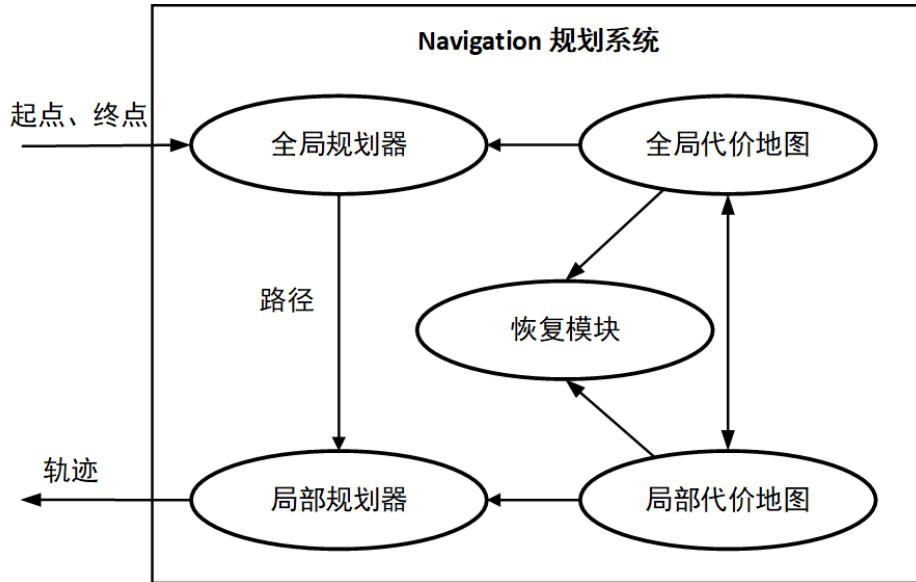


图 5-3 navigation 规划软件架构图

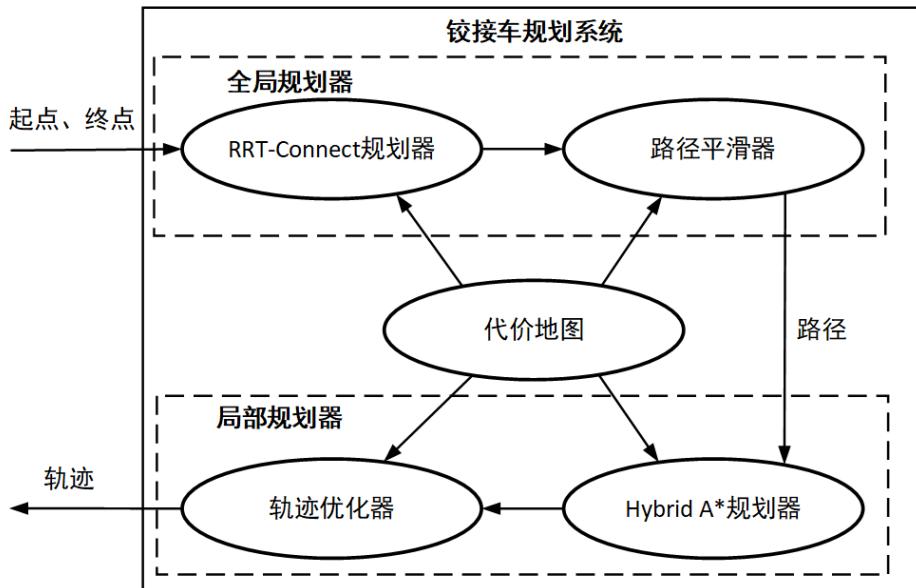


图 5-4 改进的铰接车软件架构图

在 ROS 中 navigation 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。受 navigation 规划框架的启发，本文采用流

水线的设计模式设计了有效的规划框架，如图5-4。图5-5为后续仿真使用到的 URDF 铰接式底盘模型。

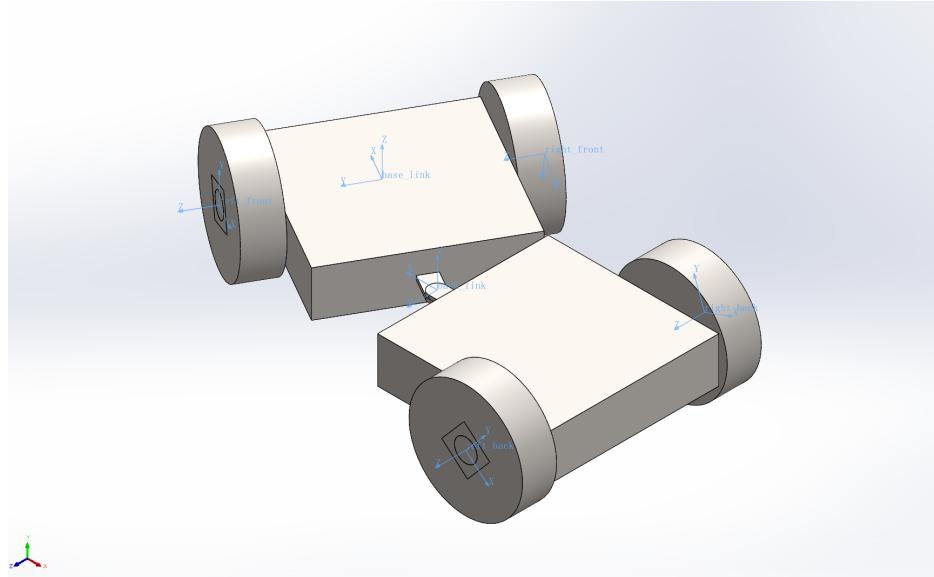


图 5-5 URDF 铰接式底盘模型

相较于 navigation 的规划架构本文做了几个变化：首先，移除了恢复模块。在标准的 navigation 系统中，恢复模块的主要功能是在机器人遇到困境时（例如陷入狭窄空间或被障碍物包围），提供必要的恢复行为，帮助机器人脱离困境。然而，在本文的仿真环境中，由于地图是预先设定且静态不变的，并没有动态变化的情况，因此恢复模块是没必要的。通过去除这一模块，不仅简化了系统的复杂性，还提高了整体框架的效率和稳定性。

其次，合并全局代价地图与局部代价地图。通常情况下，全局和局部规划器需要实时更新环境中的静态和动态障碍物信息，这对计算资源提出了挑战。为了应对这个问题，navigation 系统设计了独立工作的全局和局部代价地图，以确保系统能够高效运行。但在本文的仿真场景中，由于所有障碍物都是事先加载且固定不变的，不存在动态障碍物的情况。因此将这两个代价地图合并以提升了系统的响应速度和性能表现。

此外，对全局规划器进行了重构，将其分为 RRT-Connect 规划器和路径平滑器两个主要部分。RRT-Connect 专注于前端快速路径搜索，能够在短时间内找到一条从起点到终点的基本可行路径；而路径平滑器则负责后端处理，对这条路径进行平滑处理，并根据具体需求施加曲率限制，确保生成的路径既安全又符合机器人的运动

特性。这两者紧密结合，共同构成了一个高效的全局路径规划方案。

最后，对局部规划器进行了重构。新的局部规划器由 Hybrid A* 规划器和轨迹优化器组成。Hybrid A* 用于前端快速搜索允许倒车的轨迹，它能在复杂的环境中迅速找到一条初步可行的轨迹；随后，轨迹优化器对这条路径进行进一步优化，确保其满足一系列约束条件，包括运动学约束、安全性要求以及边界限制等。

5.2 全局规划算法仿真

在本节中，将对全局规划算法进行仿真测试。该算法的理论原理已在第三章中详细推导和分析。为了验证提出的算法的有效性，本文分别在单点到点导航场景和多点导航场景下进行全局规划测试。在这两种仿真场景中，所使用的权重参数保持一致，具体参数值见表 5-2。

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

表 5-2 中可仿真设定的最大曲率为 0.1，最小曲率为 -0.1。实际上，本文的铰接车辆允许的最大曲率为 0.2，即最大铰接角可达 30 度。然而，为了给后续的局部轨迹优化留出一定的裕量，设置了较为保守的曲率限制。这样做不仅有助于确保全局路径的可行性，还能为局部规划器提供足够的空间来进行进一步优化，以满足更严格的运动学和安全约束条件。

5.2.1 点到点路径规划仿真

本小节的主要目的是通过仿真验证第三章中点到点场景下的路径规划任务，展示规划算法在穿越不同障碍物下的表现，并评估其生成路径的质量和效率。

图 5-6 是点到点规划的仿真结果图，其中图 5-6(a) 和 5-6(b) 分别是在不同起点和终点下的规划结果。图 5-7 单独展示了一个路径规划仿真例子，用于对局部路径规划算法进行定性分析。

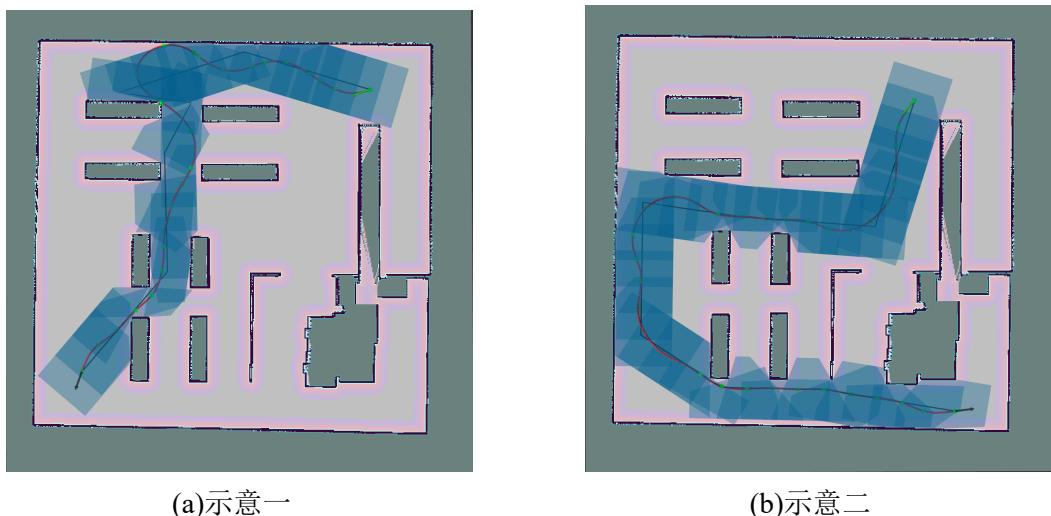


图 5-6 不同起点、终点下的路径规划仿真图

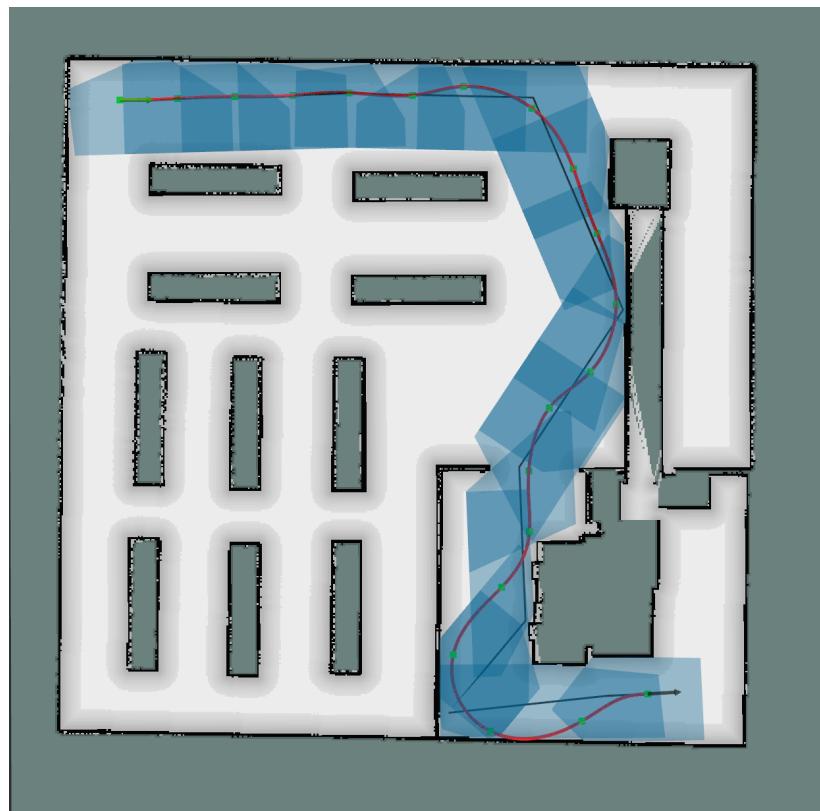


图 5-7 障碍物场景及路径规划结果

在图 5-6 和 5-7 中，黑色细线代表由 RRT-Connect 算法搜索出的初始路径，蓝色区域表示沿着该路径生成的安全走廊，而红色平滑曲线则展示了经优化算法处理后的最终路径。通过对比可以明显看出，优化后的轨迹不仅保证了足够的安全性和平

滑度，而且其曲率（即转弯幅度）保持在一个合理的范围内。此外，起点和终点的方向与车体方向（如图中箭头所示）一致，进一步确保了路径的可用性。

接下来，将简要分析图5-7对应的路径仿真数据。图 5-8 展示了路径的曲率曲线，为了便于详细比较，重新绘制了优化后的全局路径。从右子图可以看出，在坐标点(80, 0)附近，对应于路程 160 米处的曲率达到最大值 0.097。整个曲率曲线的值始终小于 0.1，这意味着曲率半径始终大于 10 米，这与仿真中设置的优化参数相吻合，证明了优化算法的有效性和合理性。

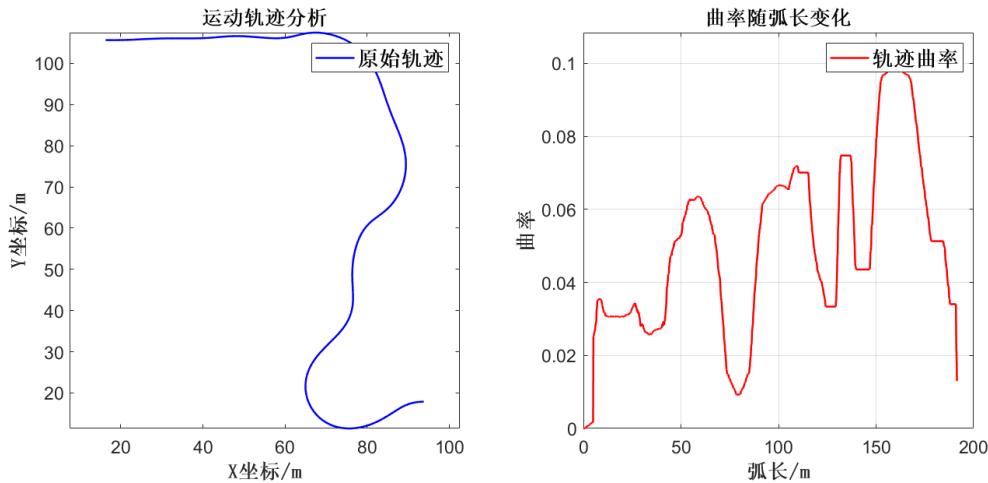


图 5-8 路径及其曲率分析

鉴于本算法分为前端搜索 (RRT-Connect) 和后端路径优化 (Path Optimization) 两大部分，在表 5-3 中记录了各部分的运行时间。实验结果显示，在大约 200 米的平均行程中，后端优化阶段的平均耗时为 10.14 毫秒，充分满足了高实时性的要求。前端的 RRT-Connect 算法借助 OMPL 库实现，平均求解时间为 6.07 毫秒。因此，整个算法的总平均求解时间约为 17.71 毫秒，展现了优异的性能表现。

表 5-3 Solving time of global planning algorithm

Algorithm	Minimum/ms	Maximum/ms	Mean/ms
RRT-Connect	3.04	12.22	6.07
Path Optimization	0.38	21.44	10.14
Total Time	4.32	24.18	17.71

综上所述，通过精心设计的障碍物场景测试、详细的路径曲率分析以及算法运行时间的评估，验证了所提出的路径规划方法在复杂环境中的高效性和可靠性，同

时也证明了其能够满足实际应用中对于安全性和实时性的严格要求

5.2.2 多站点路径规划仿真

在上一节中，对点到点的路径规划算法进行了仿真测试。本节将重点介绍多站点巡航路径规划算法的测试结果。在这种场景下，每一个点都作为一个换向点，允许车辆在这些点上改变行驶方向。值得一提的是，在 RVIZ 可视化界面中，用户可以方便地通过鼠标拖动调整航点的位置，极大地提升了操作的灵活性和便捷性。

图 5-9 展示了的多站点航仿真案例。在这个测试中，手动设定了 6 个航点，分别为 A、B、C、D、E 和 F，其中 A 为起点，F 为终点，其余四个点（B、C、D、E）作为中间的换向点。蓝色区域代表沿前端搜索路径膨胀生成的安全走廊，而红色曲线则展示了经过优化处理后的最终路径。需要注意的是，在每个换向点处，两条路径的切线方向保持一致，这确保了车辆能够在这些关键点顺利完成转向操作，例如从前进切换为倒车。这一现象与第二章中讨论的算法理论完全吻合，进一步验证了算法的有效性和实用性。

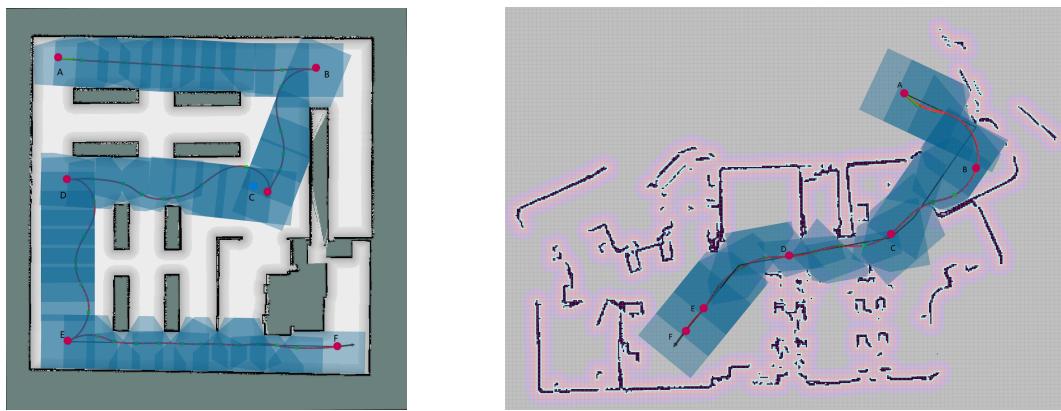


图 5-9 多站点路径规划仿真案例

为了更深入地分析路径的特性，下面对图 5-9(a)进行仿真的数据分析。图 5-10 展示了多航点路径的曲率曲线。绘制了优化后的全局路径，呈现了轨迹曲率随路程的变化情况。从图中可以看出，在路程为 80 米、140 米和 210 米处，曲率接近最大值 0.1，但始终未超出此限值，保证了曲率半径始终大于 10 米。这不仅符合优化参数设置，也证明了算法在复杂环境下的稳定性和可靠性。

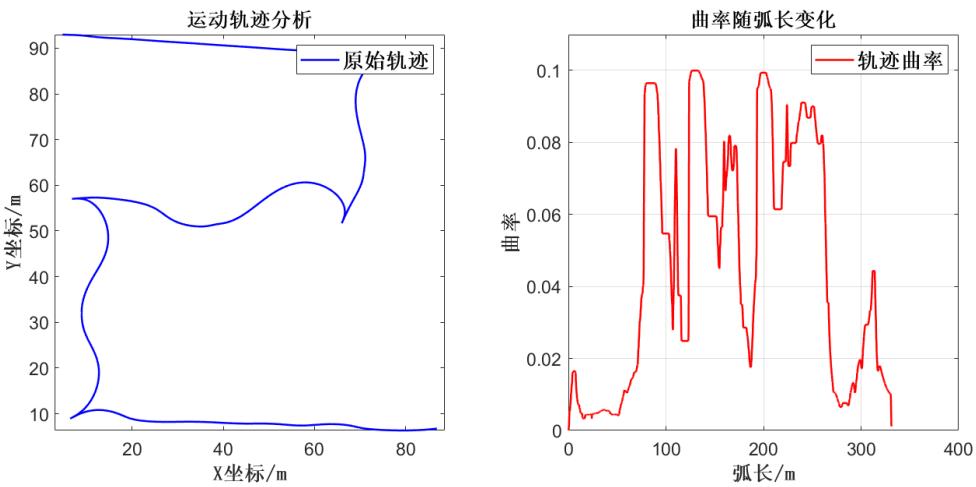


图 5-10 路径和的曲率变化曲线

综上所述，通过本次多点巡航路径规划的仿真测试，不仅验证了算法在处理多个换向点时的有效性，还展示了其在保证路径平滑度和安全性方面的优异表现。此外，通过详细的曲率分析，进一步确认了算法的稳定性，表明它能够满足实际应用中的严格要求。未来的工作将集中在进一步优化算法性能，并探索其在更多复杂场景下的适用性。

5.3 局部规划算法仿真

在本节中，将对局部规划算法进行仿真测试，重点验证其避障能力。详细的算法理论部分已在第四章中进行了详尽的介绍。为了确保仿真的准确性和一致性，使用了通用的车体参数，具体参数值见表 5-1。此外，针对局部规划特有的参数设置，提供了详细的说明，详见表 5-4。从表 5-4 中可以看出，本文在铰接角速度代价、加加

表 5-4 Global Planner Parameters

Parameter	Description	Value
$w_{u\omega}$	Articulation angular velocity penalty weight	1
w_{uj}	Jerk penalty weight	1
w_t	Time penalty weight	1
w_{kin}	Kinematic penalty weight	1000
w_c	Corridor constraint penalty weight	1000
w_b	Boundary constraint penalty weight	1000

速度（舒适度）代价和时间最优代价上设置了相对较小的权重。这主要是因为本次

提出的算法本质上是一个运动规划算法，在输入量方面可以稍作放松，同时也不苛求严格的时间最优性。对于运动学约束（平滑性）、走廊约束（安全性）以及边界约束（可行性），设定了非常高的权重，分别为 1000。

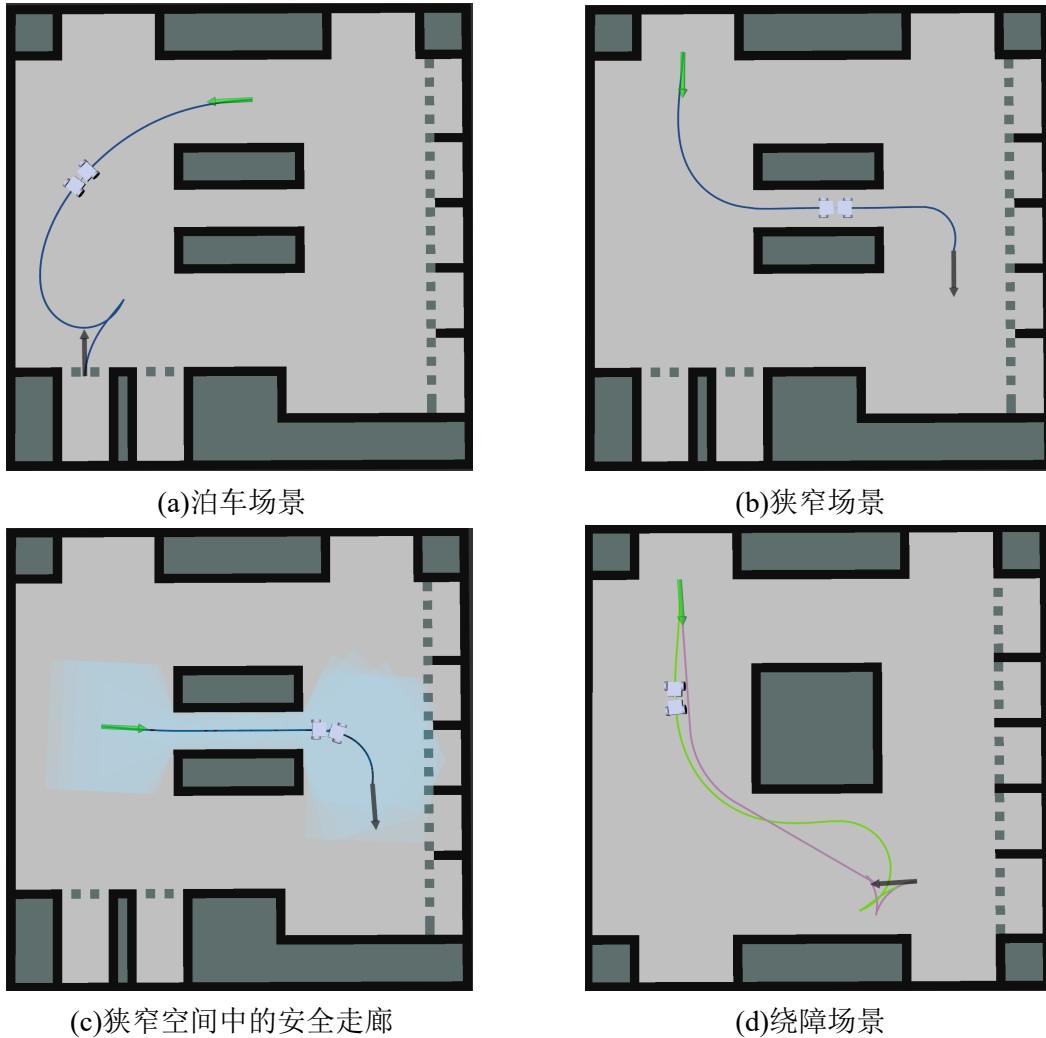


图 5-11 局部规划算法在部分场景中的规划结果展示

为了展示局部规划算法的效果，本文在仿真环境中设置了多种障碍物配置，并记录了算法在不同场景下的表现。如图 5-11 所示，展示了局部规划算法在不同场景下的仿真结果。这些场景包括泊车、穿越狭窄区域以及绕过障碍物等典型情况。在图 5-11(a)、5-11(b) 和 5-11(c) 中，蓝色线条代表由算法生成的最终轨迹，浅蓝色区域则表示安全走廊。这些图展示了算法如何在复杂环境中找到一条既安全又高效的路径。进一步地，在图 5-11(d) 中，紫色线条表示由 Hybrid A* 算法生成的初始解，而绿色线条则是经过优化后的最终轨迹。优化后的轨迹具有更少的换向次数，这不仅

提高了路径的平滑度，还增强了车辆行驶的舒适性和安全性。

图 5-12 展示了穿越室内场景的仿真结果。其中，蓝色细线为 Hybrid A* 算法生成的初步路径，而粗线则为经过优化处理后的最终轨迹。优化后的路径更加平滑，并且有效避开了所有障碍物，确保了车辆的安全通行。

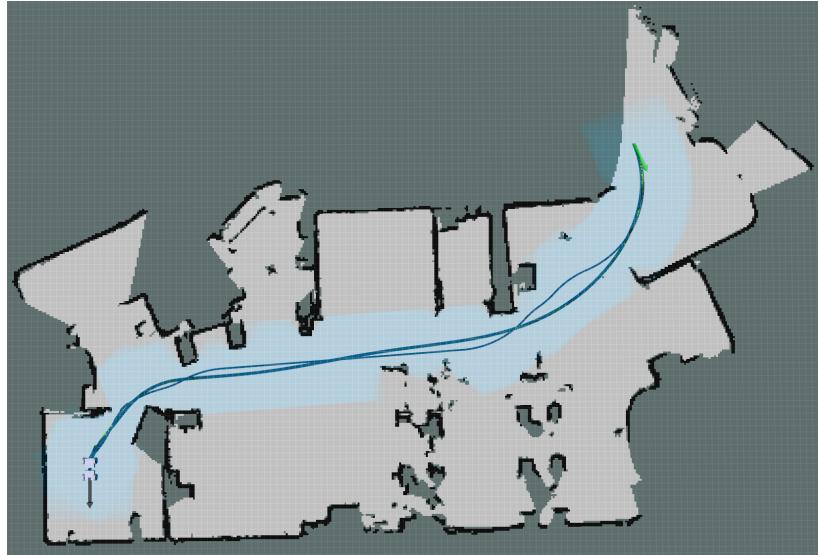


图 5-12 室内栅格地图规划结果展示

下面两小节将主要对两个实用场景进行详细的仿真分析，分别是自主泊车场景和侧方位停车场景。

5.3.1 自主泊车场景

自主泊车类的算法要求车辆能够在前进和后退之间灵活切换，但许多现有的规划算法（如五次多项式等）并未明确区分这两种模式。相比之下，本文提出的算法基于精确的运动学模型构建，天然具备前进与后退的切换能力，即使在泊车这种对精度要求极高的场景下也能表现出色。图 5-13 是在泊车场景中 Hybrid A* 算法与优化后的路径对比图。

铰接车将从绿色箭头处开始移动，并最终停靠在灰色箭头指示的位置。深蓝色细线代表由改进版 Hybrid A* 算法生成的轨迹，在接近终点时需要进行三次换向才能成功完成泊车任务。为了确保安全，在 Hybrid A* 轨迹的基础上创建了一个膨胀的安全区域。只要铰接车保持在这个区域内，就能有效避免碰撞。可以看出，本文提出的局部规划算法生成了更为流畅的绿色轨迹，在接近终点时仅需一次换向即可完

成泊车任务。尽管局部规划算法使用 Hybrid A* 生成的路径作为初始解，但最终得到的轨迹质量远超前者，显著减少了换向次数。这表明，即便初始解的质量并非最佳，本文提出的算法也能够通过迭代优化将轨迹调整至较高水平。

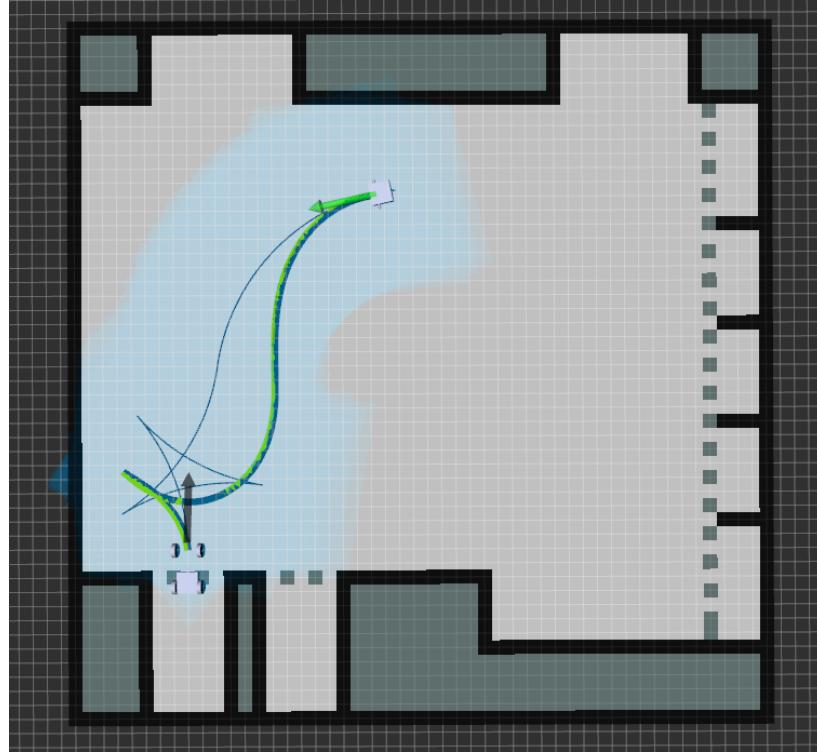


图 5-13 改进的 Hybrid A* 算法与优化后的路径对比

本文提出的算法不仅降低了换向点的数量，提高了泊车过程中的效率和平稳性，还展示了其对初始解质量依赖程度较低的特点。此外，值得注意的是，该算法虽然依赖 Hybrid A* 生成的轨迹作为粗略解，但经过优化处理后轨迹明显大幅度偏离粗略解，并且换向点大大减少。这一结果表明，本文提出的算法具有较强的自我修正能力，能够在初始解质量不高的情况下，仍然生成高质量的最终路径。

接下来，将对仿真数据进行详细分析，以进一步了解算法性能。图 5-14 展示了航向角、铰接角、车体速度和车体加速度随时间的变化情况。从图表中可以看出，航向角和车速曲线非常平滑且连续可微，这主要得益于采用的精确运动学模型。这种平滑度对于确保车辆在行驶过程中的稳定性和安全性至关重要。其次，车体速度在整个过程中保持在-1 m/s 至 3 m/s 的范围内，而加速度则在-0.4 m/s² 至 0.4 m/s² 之间波动，均符合设计要求。然而对于铰接角和加速度的曲线虽然也是连续的，但其斜率有时会出现小突变。这是因为铰接角和加速度与输入量（即铰接角速度和加速度）

度)之间仅存在一阶导数关系。由于输入量可以是不连续的,因此铰接角和加速度并不能保证始终可微。不过,这种现象并不会显著影响车辆的整体表现,因为铰接角范围始终保持在-0.5 rad 至 0.5 rad 之间,满足操作需求。值得注意的是,在 0 秒至 26 秒期间,铰接车的速度为正值,表示车辆处于前进状态。到了 26 秒时,车速降为 0 m/s,标志着车辆到达换向点。此时,结合示意图 5-13 可以看出,铰接角也变为 0 rad,意味着方向盘完全回正。随后,车速转为负值,表明车辆开始倒车并顺利泊入车库。

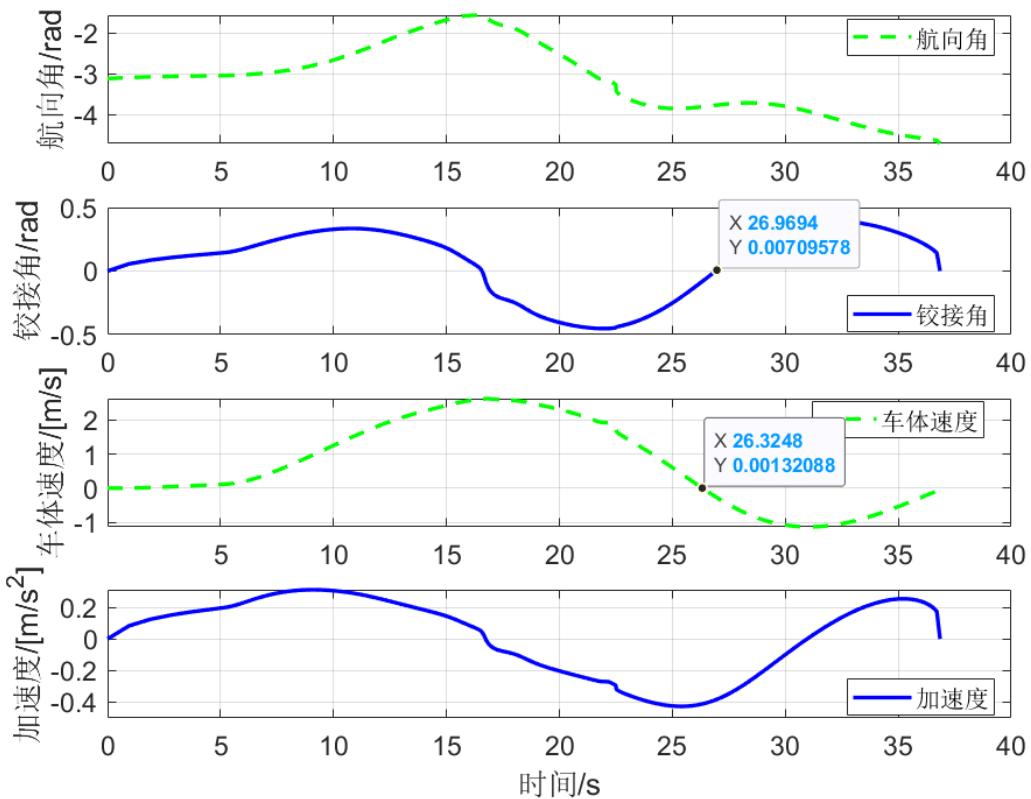


图 5-14 泊车轨迹状态变量曲线

如图5-15,是优化出的轨迹的输入量,可以看到在 5 秒、17 秒和 24 秒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,输入变量和事先定下的边界约束时保持一致的,铰接角速度在-0.2 rad/s 至 0.2 rad/s 之间,加速度在-0.3 m/s³ 至 0.3 m/s³ 之间。

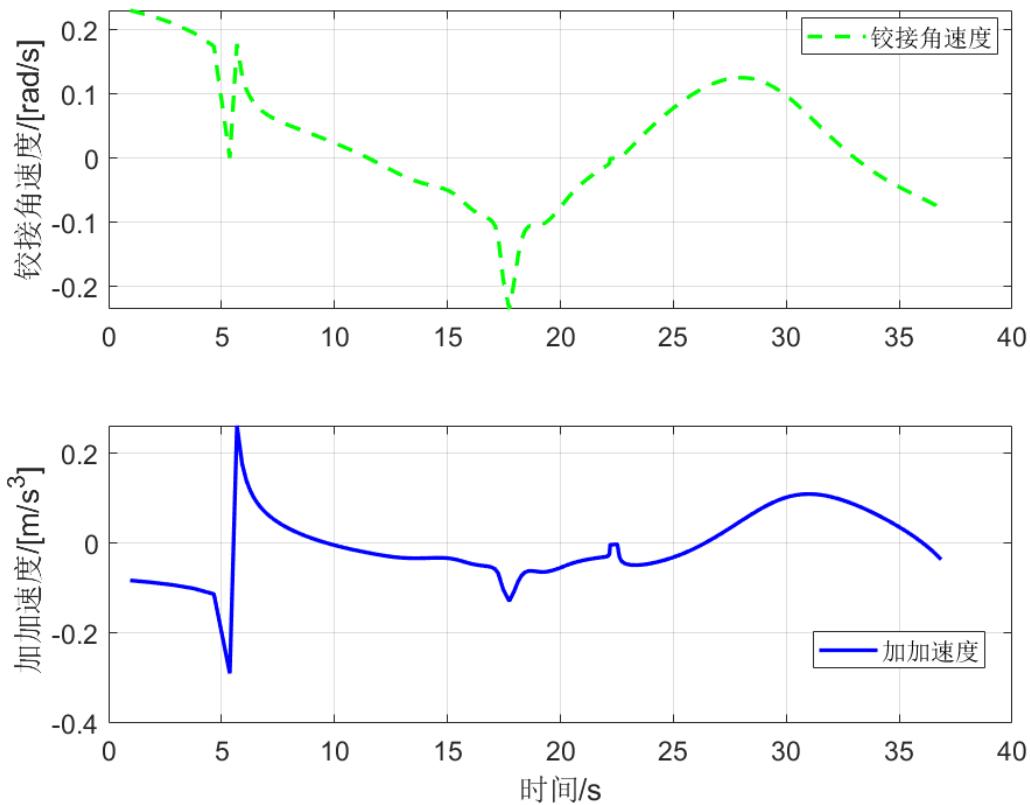


图 5-15 泊车轨迹输入变量曲线

表5-5是局部规划算法的平均求解时间，每个场景测试20次。

表 5-5 Solving time of local planning algorithm

Scenario	Length/m	Runtime/ms
Obstacle Avoidance 1	25	78.2
Obstacle Avoidance 2	35	106.0
Obstacle Avoidance 3	45	147.8
Parking 1	35	207.4
Parking 2	45	515.2

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，这一小节将做侧方位停车的仿真实验。如图5-16是侧边泊车的RVIZ仿真可视化图，图5-17是对应的状态曲线图。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。

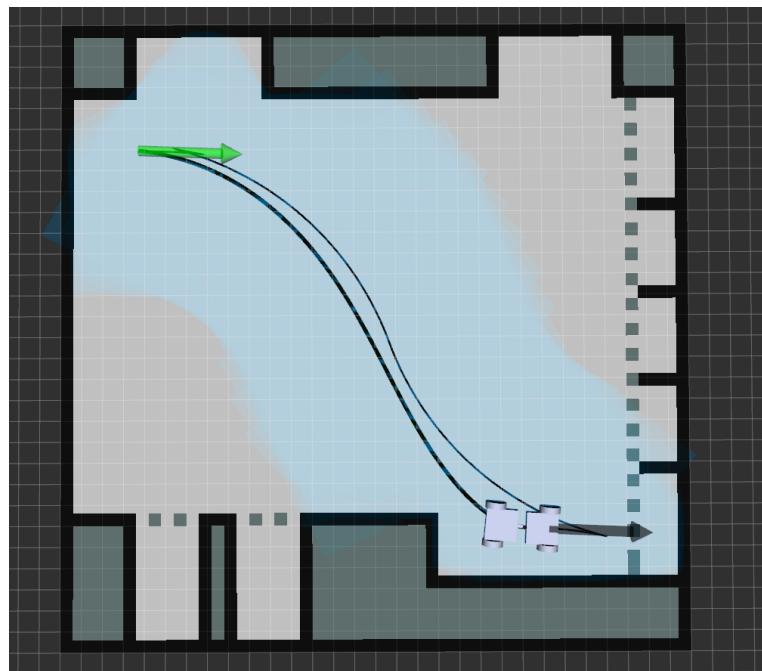


图 5-16 侧边泊车场景

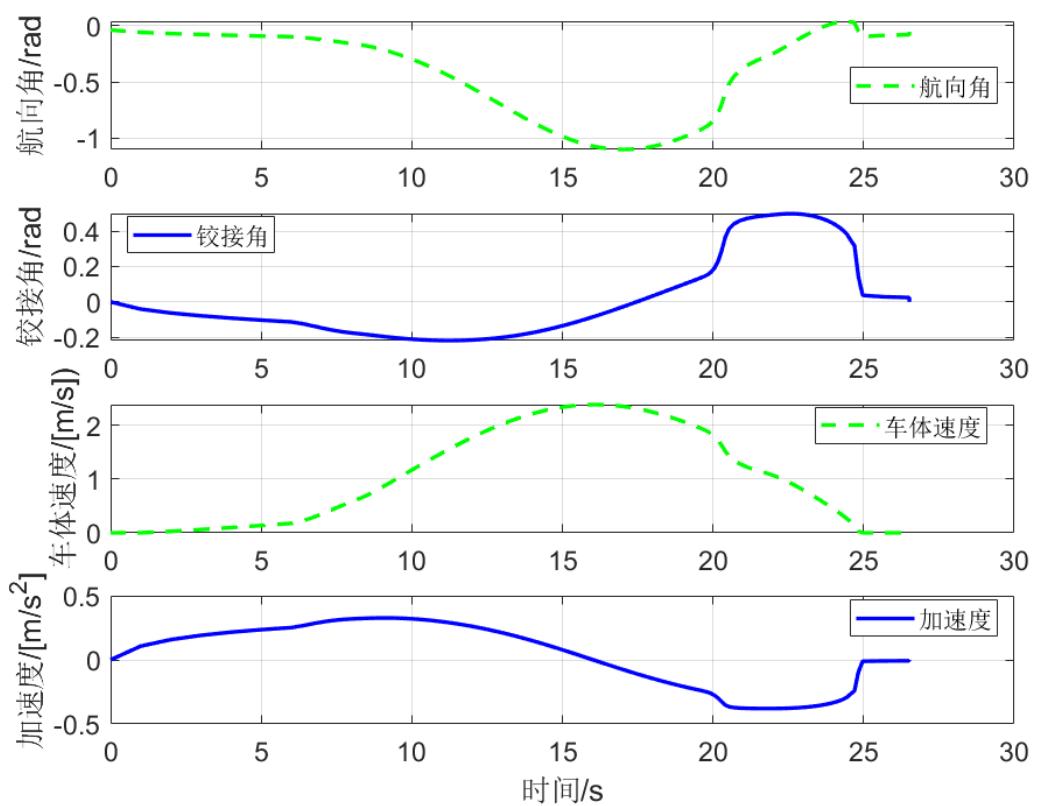


图 5-17 侧边泊车状态变量曲线

图中数据包括航向角、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。蓝色区域是沿着细轨迹膨胀出来的安全走廊区域。较粗的轨迹是算法生成的最终轨迹。可以看出轨迹对于该侧边停车场景并没有发生换向。

5.3.3 局部规划算法的性能对比

为了全面地评估本研究提出的局部轨迹规划算法性能，本文将其与文献??中描述的算法进行了对比分析。图5-18展示了不同算法（Hybrid A*、OBPP 算法以及本文提出的局部规划算法）的规划结果。

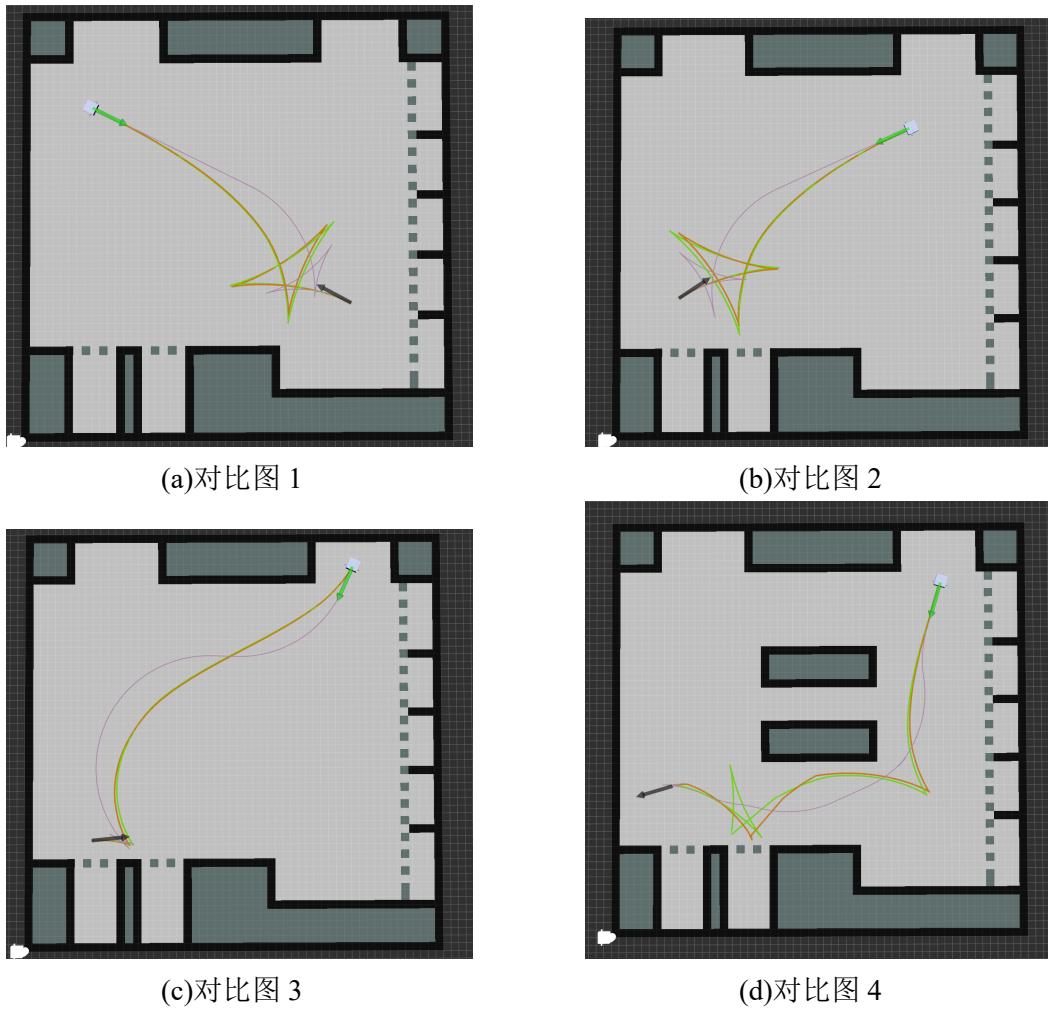


图 5-18 Hybrid A*、OBPP 算法、本文的局部规划算法对比图

为简化叙述过程，本文将该参考文献中的算法称为 OBPP 算法。OBPP 算法通过构建基于最优控制的规划问题，并采用增广拉格朗日方法处理约束条件，这种方法

虽然有效，但由于引入了大量的协态变量，导致计算成本显著增加。避障策略依赖于自车结构和障碍物凸包的距离来实现，这意味着随着环境中障碍物数量的增加，约束条件的数量也会线性增长。

为了确保实验结果的准确性和可靠性，在实现过程中使用了 IPOPT 求解器对本文提出的局部规划算法以及 OBPP 算法进行求解，并利用 CasADi 自动微分工具进行梯度计算。此外，两种算法的初始粗略路径均由 Hybrid A* 算法生成。

图5-18中紫色轨迹代表 Hybrid A* 算法的输出，橙色轨迹对应 OBPP 算法，而绿色轨迹则表示本研究所提出的局部规划算法的结果。从图中可以看出，相较于 OBPP 算法所规划出的较为激进的轨迹，本文算法在转向点处的表现更加自然流畅，提供了更高的舒适度。

表5-6比较了 OBPP 算法与本文算法在 20 米至 30 米路径长度范围内的表现，随机起点、终点共测试了 40 次，测试指标包括规划成功率（Success rate）、平均换向次数（Reversing）、舒适度（Comfort）以及算法求解时间（Runtime）。平均舒适度的计算公式如下：

$$Comfort = \frac{\sum_{i=1}^N jerk_i * \Delta t_i}{\sum_{i=1}^N \Delta t_i} \quad (5-1)$$

其中 $jerk_i$ 为第 i 步系统的输入变量，参考第四章。 Δt_i 为第 i 步系统的离散时间。

表 5-6 20m 至 30m 路径长度时几种算法的性能对比

Algorithm	Success rate	Reversing/times	Comfort/[s ³ /m]	Runtime/ms
This work	100%	1.2	16.13	79.3
OBPP	87.4%	1.2	14.08	21817.1

结果显示，本文提出的算法不仅拥有更高的成功率，而且其求解时间也远低于 OBPP 算法。这是因为 OBPP 算法在应对运动学约束、安全约束及时间约束时采用了增广拉格朗日方法，导致大量协态变量的引入，引发了维度爆炸的问题，增加了求解难度和失败风险。相反，本文的方法通过罚函数和微分同胚策略处理约束条件，避免了额外协态变量的引入，从而提高了计算效率和稳定性。

5.4 本章小结

在本章中，首先介绍了基于 ROS 和 RVIZ 的仿真环境，该环境为后续的研究提供了坚实的基础。接着，设计了一个专门适用于本课题需求的运动规划软件框架，旨

在优化系统的效率和性能。

随后，针对第三章和第四章中提出的全局路径规划算法与局部轨迹规划算法，设计并实施了一系列详细的仿真实验，以全面验证这些算法的有效性和实用性。通过这些实验，不仅展示了算法在复杂环境中的优异表现，还进一步确认了其在实际应用中的可靠性和鲁棒性。

第 6 章 结论

为了解决无人铰接车在复杂非结构化环境中的轨迹规划问题，本文将运动规划任务分解为全局规划和局部规划两个阶段，并分别进行了算法设计，全文工作总结如下：

(1) 建立了铰接式车辆的运动学模型，并基于最优控制问题框架对轨迹规划问题进行建模。本文假设铰接车辆保持低速运动且不发生侧滑，推导出其运动学方程与转弯半径，并将其状态量表示为笛卡尔坐标系下的 (x, y) 坐标及其导数。这为后续的轨迹优化提供了基础。

(2) 对于全局路径规划，本文就铰接式车辆提出了一种基于微分平坦理论与 Minco 样条的全局优化方法。通过选取笛卡尔坐标系作为平坦输出，将非完整约束转化为平坦空间代数关系，并采用 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，设计基于安全走廊算法，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，将铰接角硬约束和走廊安全约束转换到代价函数中，利用 LBFGS 算法实现高效求解。

(3) 针对局部轨迹规划，本文提出了一种基于安全走廊的时空联合轨迹规划方法。首先，对铰接车外观进行通用性建模，将车辆形状简化为两个矩形的连接，将避障问题转化为多边形几何碰撞问题。其次，针对铰接车改进了 Hybrid A* 算法，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。此外，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，构建了基于非线性模型预测控制 (NMPC) 的轨迹规划方法，并加入了动力学约束、运动学约束及避障约束。构建优化问题，通过通过外罚函数和微分同胚等方法去除约束后进行高效求解。

(4) 设计了仿真实验验证了全局规划和局部轨迹规划算法的有效性和鲁棒性。通过在点到点导航场景和多点导航场景下进行全局规划测试，验证了算法在复杂环境中的高效性和可靠性。同时，通过在自主泊车场景和侧方位停车场景下进行局部规划测试，验证了算法的避障能力和路径优化效果。

综上所述，本文所提出的基于全局规划和局部规划的无人铰接车轨迹规划方法能够有效解决无人铰接车在复杂非结构化环境中的路径规划问题，有助于提高无人铰接车整体自主化的水平。

本文提出的解决方案虽然能够在复杂环境中实现无人铰接车的路径规划，但仍存在一些不足之处，有待进一步改进和完善：

(1) 由于实验条件的限制，算法尚未应用于真实的铰接式车辆上，因此无法充分评估其在不确定环境下的鲁棒性。未来的研究应致力于将算法部署到实际车辆中，以验证其在真实世界中的性能表现。

(2) 随着环境复杂度的增加，算法的求解时间也会相应延长，这可能影响其实时性能。为了提高算法的实时性，可以探索更加高效的优化算法或引入并行计算技术。例如，采用先进的启发式搜索方法、分布式计算框架或 GPU 加速等手段，以缩短计算时间并提升响应速度。

(3) 本文提出的全局规划基于样条曲线，要求轨迹的起点和终点航向必须与车辆的实际航向保持一致。尽管这一问题在当前研究中已得到初步解决，但多次仿真的结果显示其鲁棒性仍有待提高。未来的工作需要通过更多的仿真和实验来进一步优化这一方面，确保算法在各种情况下都能稳定可靠地运行。

尽管本文提出的解决方案在复杂环境中的无人铰接车路径规划方面取得了一定进展，但在实际应用、实时性和鲁棒性等方面仍需进一步优化和改进。未来的研究将继续致力于这些方面的提升，以期实现更高效、更可靠的路径规划系统。

参考文献

- [1] YUE C, ZHONG J, NING Q, et al. Development strategy of vehicle-energy-road-cloud collaboration in the new technological situation[J]. Strategic Study of Chinese Academy of Engineering, 2024, 26(1): 45-58.
- [2] GARRETT C R, CHITNIS R, HOLLADAY R, et al. Integrated task and motion planning[J]. Annual review of control, robotics, and autonomous systems, 2021, 4(1): 265-293.
- [3] KWAYU K M, KWIGIZILE V, LEE K, et al. Discovering latent themes in traffic fatal crash narratives using text mining analytics and network topology[J]. Accident Analysis & Prevention, 2021, 150: 105899.
- [4] LI C, HUANG X, DING J, et al. Global path planning based on a bidirectional alternating search a* algorithm for mobile robots[J]. Computers & Industrial Engineering, 2022, 168: 108123.
- [5] YANG Y, XIONG X, YAN Y. Uav formation trajectory planning algorithms: A review[J]. Drones, 2023, 7(1): 62.
- [6] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numer. Math., 1959, 1(1): 269-271.
- [7] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [8] STENTZ A. Optimal and efficient path planning for partially-known environments[C]//Proceedings of the 1994 IEEE international conference on robotics and automation. IEEE, 1994: 3310-3317.
- [9] KOENIG S, LIKHACHEV M, FURCY D. Lifelong planning a□[J]. Artificial Intelligence, 2004, 155(1-2): 93-146.
- [10] LIKHACHEV M, GORDON G J, THRUN S. Ara*: Anytime a* with provable bounds on suboptimality[J]. Advances in neural information processing systems, 2003, 16.
- [11] BULITKO V, LEE G. Learning in real-time search: A unifying framework[J]. Journal of Artificial Intelligence Research, 2006, 25: 119-157.
- [12] HARABOR D, GRASTIEN A. Online graph pruning for pathfinding on grid maps[C]//Proceedings of the AAAI conference on artificial intelligence: Vol. 25. 2011: 1114-1119.
- [13] KOENIG S, LIKHACHEV M. Real-time adaptive a[C]//Proceedings of the fifth international joint

参考文献

- conference on Autonomous agents and multiagent systems. 2006: 281-288.
- [14] LAVALLE S M, KUFFNER J J. Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan[J]. Algorithmic and computational robotics, 2001: 303-307.
- [15] KUFFNER J J, LAVALLE S M. Rrt-connect: An efficient approach to single-query path planning [C]//Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065): Vol. 2. IEEE, 2000: 995-1001.
- [16] BRUCE J, VELOSO M. Real-time randomized path planning for robot navigation[C]//IEEE/RSJ international conference on intelligent robots and systems: Vol. 3. IEEE, 2002: 2383-2388.
- [17] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning[J]. The international journal of robotics research, 2011, 30(7): 846-894.
- [18] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic[C]//2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2014: 2997-3004.
- [19] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). 2015: 3067-3074.
- [20] REEDS J, SHEPP L. Optimal paths for a car that goes both forwards and backwards[J]. Pacific journal of mathematics, 1990, 145(2): 367-393.
- [21] GUPTA MANYAM S, CASBEER D W, VON MOLL A, et al. Shortest dubins paths to intercept a target moving on a circle[J]. Journal of Guidance, Control, and Dynamics, 2022, 45(11): 2107-2120.
- [22] ZHANG Y, SUN H, ZHOU J, et al. Optimal vehicle path planning using quadratic optimization for baidu apollo open platform[C]//2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020: 978-984.
- [23] GAO F, WU W, LIN Y, et al. Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial[J]. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018: 344-351.
- [24] LEE S, KIM Y. Optimal output trajectory shaping using bézier curves[J]. Journal of Guidance, Control, and Dynamics, 2021, 44(5): 1027-1035.

- [25] NGUYEN N T, GANGAVARAPU P T, SAHRHAGE A, et al. Navigation with polytopes and b-spline path planner[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). 2023: 5695-5701.
- [26] LI B, SHAO Z. A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles[J]. Knowledge-Based Systems, 2015, 86: 11-20.
- [27] KONDAK K, HOMMEL G. Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms[C]//Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164): Vol. 3. 2001: 2698-2703 vol.3.
- [28] ARANTES M D S, TOLEDO C F M, WILLIAMS B C, et al. Collision-free encoding for chance-constrained nonconvex path planning[J]. IEEE Transactions on Robotics, 2019, 35(2): 433-448.
- [29] ZHANG X, LINIGER A, BORRELLI F. Optimization-based collision avoidance[J]. IEEE Transactions on Control Systems Technology, 2021, 29(3): 972-983.
- [30] ZHANG X, LINIGER A, SAKAI A, et al. Autonomous parking using optimization-based collision avoidance[C]//2018 IEEE Conference on Decision and Control (CDC). 2018: 4327-4332.
- [31] CHANG T, TIAN G. Hybrid a-star path planning method based on hierarchical clustering and trichotomy[J]. Applied Sciences, 2024, 14(13): 5582.
- [32] CHOI J W, HUHTALA K. Constrained global path optimization for articulated steering vehicles [J]. IEEE Transactions on Vehicular Technology, 2015, 65(4): 1868-1879.
- [33] YOSSAWEE W, TSUBOUCHI T, SARATA S, et al. Path generation for articulated steering type vehicle using symmetrical clothoid[C]//2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT'02.: Vol. 1. IEEE, 2002: 187-192.
- [34] NAYL T, NIKOLAKOPOULOS G, GUSTAFSSON T. On-line path planning for an articulated vehicle based on model predictive control[C]//2013 IEEE international conference on control applications (CCA). IEEE, 2013: 772-777.
- [35] CHEN D, YANG Z, SVENSSON L, et al. Optimization based path planning for a two-body articulated vehicle[C]//2020 IEEE 16th International Conference on Automation Science and Engineering (CASE). 2020: 397-403.
- [36] HONG B, MA X. Path optimization for a wheel loader considering construction site terrain[C]// 2018 IEEE Intelligent Vehicles Symposium (IV). 2018: 2098-2103.
- [37] KAWABE T, TAKEI T, AND E I. Path planning to expedite the complete transfer of distributed

参考文献

- gravel piles with an automated wheel loader[J]. Advanced Robotics, 2021, 35(23): 1418-1437.
- [38] ZHAO M, YIN G, SHEN T, et al. Time optimal trajectory optimization for vertical parking of heavy duty vehicle based on chebyshev pseudospectral method[C]//2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI). IEEE, 2022: 1-6.
- [39] SHI J, SUN D, QIN D, et al. Planning the trajectory of an autonomous wheel loader and tracking its trajectory via adaptive model predictive control[J]. Robotics and autonomous systems, 2020, 131: 103570.
- [40] CORKE P I, RIDLEY P. Steering kinematics for a center-articulated mobile robot[J]. IEEE Transactions on Robotics and Automation, 2001, 17(2): 215-218.
- [41] ANDERSON S J, PETERS S C, PILUTTI T E, et al. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios[J]. International Journal of Vehicle Autonomous Systems, 2010, 8(2-4): 190-216.
- [42] MEHTA P, MEYN S. Q-learning and pontryagin's minimum principle[C]//Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference. IEEE, 2009: 3598-3605.
- [43] ALURU N. A point collocation method based on reproducing kernel approximations[J]. International Journal for Numerical Methods in Engineering, 2000, 47(6): 1083-1121.
- [44] ROCKAFELLAR R T. Augmented lagrange multiplier functions and duality in nonconvex programming[J]. SIAM Journal on Control, 1974, 12(2): 268-285.
- [45] WEI E, OZDAGLAR A. Distributed alternating direction method of multipliers[C]//2012 IEEE 51st IEEE Conference on Decision and Control (CDC). IEEE, 2012: 5445-5450.
- [46] YENIAY Ö. Penalty function methods for constrained optimization with genetic algorithms[J]. Mathematical and computational Applications, 2005, 10(1): 45-56.
- [47] CURTIS F E, SCHENK O, WÄCHTER A. An interior-point algorithm for large-scale nonlinear optimization with inexact step computations[J]. SIAM Journal on Scientific Computing, 2010, 32 (6): 3447-3475.
- [48] KÜMMERLE R, GRISETTI G, STRASDAT H, et al. g 2 o: A general framework for graph optimization[C]//2011 IEEE international conference on robotics and automation. IEEE, 2011: 3607-3613.

- [49] MELLINGER D, KUMAR V. Minimum snap trajectory generation and control for quadrotors[C]// 2011 IEEE international conference on robotics and automation. IEEE, 2011: 2520-2525.
- [50] WANG Z, ZHOU X, XU C, et al. Geometrically constrained trajectory optimization for multicopters [J]. IEEE Transactions on Robotics, 2022, 38(5): 3259-3278.
- [51] LI B, ACARMAN T, ZHANG Y, et al. Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8): 11970-11981.
- [52] DEITS R, TEDRAKE R. Computing large convex regions of obstacle-free space through semidefinite programming[M]. Cham: Springer International Publishing, 2015: 109-124.
- [53] LIU S, WATTERSON M, MOHTA K, et al. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments[J]. IEEE Robotics and Automation Letters, 2017, 2(3): 1688-1695.
- [54] LIU D C, NOCEDAL J. On the limited memory bfgs method for large scale optimization[J]. Mathematical programming, 1989, 45(1): 503-528.
- [55] BRESENHAM J. A linear algorithm for incremental digital display of circular arcs[J]. Communications of the ACM, 1977, 20(2): 100-106.

攻读硕士学位期间取得的成果

1. 发表的学术论文

- [1] YANG Y, LUO X, CHEN K, et al. A spatial-temporal trajectory planner for center-articulated vehicles[C/OL]//2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC). 2024: 442-448. (国际会议论文, 已录用)

致 谢