

武汉理工大学

(申请硕士学位论文)

面向行泊一体的智能车运动规划方法

培养单位：交通与物流工程学院

学科专业：交通运输工程

研究 生：杜宇豪

指导教师：邱志军 教授

2025 年 5 月

Intelligent vehicle motion planning method for integrated driving and parking

Dissertation Submitted to
Wuhan University of Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

by
Yuhao Du
(Transportation Engineering)

Dissertation Supervisor:
Tony Qiu.z
Affiliation:
School of Transportation and Logistics Engineering

May,2025

分类号 _____
UDC _____

密 级 _____
学校代码 _____ 10497

武汉理工大学

学位论文

题目 面向行泊一体的智能车运动规划方法

英文题目 Intelligent vehicle motion planning method for integrated driving and parking

研究生姓名 杜宇豪

指导教师 姓名 邱志军 职称 教授 学位 博士

单位名称 武汉理工大学 邮编 430000

副指导教师 姓名 贺宜 职称 教授 学位 博士

单位名称 武汉理工大学 邮编 430000

申请学位级别 硕士 学科专业名称 交通运输工程

论文提交日期 2025 年 5 月 论文答辩日期 2025 年 5 月

学位授予单位 武汉理工大学 学位授予日期 2025 年 6 月

答辩委员会主席 评阅人 X

2025 年 5 月

独创性声明

本人声明，所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得武汉理工大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名: _____ 日期: _____

学位论文使用授权书

本人完全了解武汉理工大学有关保留、使用学位论文的规定，即学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权武汉理工大学可以将本学位论文的全部内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存或汇编本学位论文。同时授权经武汉理工大学认可的国家有关机构或论文数据库使用或收录本学位论文，并向社会公众提供信息服务。

(保密的论文在解密后应遵守此规定)

研究生（签名）: _____ 导师（签名）: _____ 日期: _____

摘要

汽车工业的蓬勃发展在极大提升民众出行便利的同时，也导致城市交通系统复杂度呈指数级增长，衍生出交通拥堵、事故频发及停车位稀缺等系统性难题。自动驾驶汽车作为智能交通系统的核心载体，被寄予提高出行效率、缓解交通压力及增强道路安全的厚望。在技术演进过程中，行车域与泊车域这两大传统分离的功能模块，得益于人工智能与高算力芯片的突破，正加速向行泊一体架构演进。然而，行泊一体系统的实际落地仍面临双重技术壁垒：动态行车场景中，交通环境的高度动态性和不确定性；狭窄泊车场景中，狭窄空间的精确建模与复杂运动约束的非凸性，使得问题的复杂度指数级上升。现有研究虽在局部问题上取得进展，但面对行泊一体复合场景，安全行车规划问题的高效求解，仍是亟待解决的技术难题。

自动驾驶系统是一个高度复杂的系统，包含多个相互独立但又联系紧密的算法单元。运动规划是其中的一个核心模块，负责生成平滑可行的轨迹。本文针对行车和泊车两大使用场景，进一步将运动规划问题划分为极端狭窄空间和复杂动态环境下的两个子问题，并展开相关研究。

狭窄环境中自由空间极度受限、非线性约束耦合复杂、轨迹精度与计算效率难以平衡。为实现可行轨迹的快速求解，本文提出一种变精度改进混合 A* 算法，利用变精度采样替代均匀采样，探索步长随着转向幅度的增加而减少，提升搜索效率的同时，增加了输出轨迹中直线段的比例，减少了无效大曲率转向。为进一步提升轨迹质量，将轨迹优化问题描述为最优控制问题，并利用增量式矩形通行走廊将复杂非线性碰撞约束转换为线性 Box 约束，降低了问题的复杂度，为提升问题求解成功率，将运动学硬约束转换为软约束，并利用迭代求解框架，实现了狭窄空间可行最优轨迹的高效快速求解。

结构化环境下道路拓扑复杂，兼具交互性与不确定性。针对交互式行车问题，利用 OpenDrive 格式地图中的路段离散邻接特性，建立了路段邻接矩阵，使用 dijkstra 算法实现了车道级全局路径规划算法。引入 Frenet 坐标系，将车辆的横纵向运动解耦，将笛卡尔坐标系下的时空耦合问题转化为 SL 坐标系下的路径规划问题和 ST 坐标系下的速度规划问题。建立 ST 图和 SL 图，完成可行凸空间的开辟。将车辆的横纵向规划问题描述为可行凸空间下的分段平滑问题，利用曲率约束和曲线导数约束替代车辆的运动性能约束，使其能够满足车辆的性能需求。

为有效提升车辆在复杂环境下的跟踪稳定性。分析并建立了多种车辆跟踪模型，介绍了 LQR 问题的一般形式并对其求解过程进行了详细推导。然后将车

辆的轨迹跟踪问题描述为最优控制问题，构建了基于 PID 算法的纵向控制器和基于 LQR 算法的横向控制器。为解决横向误差模型下稳态误差稳定存在的问题，引入曲率前馈，提升了算法的鲁棒性。

为了进一步体现本文算法的有效性和鲁棒性。利用虚实融合自动驾驶实车测试工具链，构建了 11 个测试场景，对本文算法进行了实车测试。实验结果表明，本文所提出算法在 11 个测试场景中均表现良好，能够在保证安全性和舒适性的前提下完成紧密跟驰，无信号通过，换道通行和泊车等任务。

关键词：运动规划；计算最优控制；自动驾驶；非线性规划；轨迹跟踪

Abstract

The booming development of automobile industry not only brings great convenience to people's travel, but also increases the complexity of urban transportation system exponentially, and brings a series of traffic problems such as traffic congestion, frequent accidents and rare parking spaces.g. As an important part of intelligent transportation system, self-driving vehicle is considered as one of the important and feasible ways to improve travel efficiency, ease traffic congestion and improve travel safety. As two important parts of the automatic driving system, the driving area and parking area have been gradually separated and unified with the development of artificial intelligence and chip manufacturing technology in recent years. Narrow feasible space parking and high-speed interactive driving are two important use scenarios of the driving and parking integrated automatic driving system. Due to the complexity of the environment and space, they are considered as technical difficulties to be solved urgently.

The autopilot system is a highly complex system consisting of multiple independent but closely linked algorithm units. Motion Planning is one of the core modules that generates a smooth and feasible trajectory. The motion planning problem is further divided into two subprojects: static confined space and complex dynamic environment, and the relevant research is carried out.

In order to achieve the fast solution of feasible trajectory in narrow space, a variable precision improved hybrid A * algorithm is proposed, which uses variable precision sampling to replace uniform sampling. The exploration step decreases with the increase of steering amplitude, improves the search efficiency, increases the proportion of straight line segments in the output trajectory, and reduces the invalid large curvature steering. To further improve the trajectory quality, the trajectory optimization problem is described as the optimal control problem, and the complex nonlinear collision constraint is converted into the linear Box constraint by using the incremental rectangular passage corridor, which reduces the complexity of the problem. In order to improve the success rate of the problem solving, the kinetic hard constraint is converted into the soft constraint, and the efficient and fast solution of feasible optimal trajectory in a narrow space is realized by using the iterative solution framework.

To solve the interactive traffic problem under the structured road, the dis-

crete adjacency characteristics of road sections in OpenDrive map are used to establish the road section adjacency matrix, and the dijkstra algorithm is used to realize the lane-level global path planning algorithm. The Freet coordinate system is introduced to decouple the transverse and longitudinal motion of the vehicle, and the time-space coupling problem in the Cartesian coordinate system is converted into the path planning problem in the SL coordinate system and the speed planning problem in the ST coordinate system. Establish ST map and SL map to complete the development of feasible convex space. The transverse and longitudinal planning problem of a vehicle is described as a piecewise smoothing problem in a feasible convex space, and the motion performance constraint of the vehicle is replaced by the curvature constraint and the curve derivative constraint to meet the performance requirements of the vehicle.

Various vehicle tracking models are analyzed and established. The general form of LQR problem is introduced and its solution process is deduced in detail. Then, the vehicle trajectory tracking problem is described as the optimal control problem, and the longitudinal controller based on PID algorithm and the lateral controller based on LQR algorithm are constructed. To solve the problem of steady state error stability in the transverse error model, curvature feedforward is introduced to improve the robustness of the algorithm.

Based on the test tool chain of virtual and real automatic driving, 11 test scenes are constructed, and the algorithm in this paper is tested in real vehicle. The experimental results show that the algorithm proposed in this paper performs well in 11 test scenarios, and can complete such tasks as close following, no signal passing, lane change passing and parking under the premise of guaranteeing safety and comfort.

Key words: Motion planning; Calculate optimal control; Automatic driving; Non-linear programming; Path Tracking

目录

摘要	ii
Abstract	iv
第 1 章 绪论	1
1.1 研究背景与意义	1
1.2 国内外研究现状	3
1.2.1 运动规划方法研究现状	3
1.2.2 轨迹跟踪方法研究现状	9
1.3 主要研究内容及技术路线	12
1.3.1 课题来源	12
1.3.2 主要研究内容	13
第 2 章 面向极端狭窄空间的运动规划方法	15
2.1 基于改进混合 A* 的可行解搜索方法	15
2.1.1 A* 算法原理	16
2.1.2 混合 A* 算法原理	16
2.1.3 改进混合 A* 算法	18
2.2 基于增量矩形拓展的安全走廊构建方法	20
2.3 基于安全走廊的后端轨迹优化方法	25
2.3.1 轨迹优化问题	25
2.3.2 轨迹优化问题构建与求解	27
2.4 仿真验证	30
第 3 章 面向动态交互环境下的运动规划方法	35
3.1 动态交互环境下运动规划问题分析	35
3.1.1 动态交互环境定义	35
3.1.2 交互环境下运动规划问题分析	36
3.2 全局路径规划与局部坐标系构建	37
3.2.1 基于 dijkstra 算法的全局路径规划算法	37
3.2.2 Frenet 坐标系及坐标系转换	38
3.3 SL/ST 图构建与可行凸空间开辟	40

3.3.1	SL 图构建与可行边界提取	40
3.3.2	ST 图与可行空间构建	42
3.4	基于分段优化的轨迹规划方法	44
3.4.1	分段优化问题的构建	44
3.4.2	基于分段优化的路径规划方法	46
3.4.3	基于分段优化的速度规划方法	48
3.5	仿真验证	49
第 4 章	面向行泊一体的轨迹跟踪控制器	55
4.1	车辆运动模型建立	55
4.1.1	单车运动学模型	55
4.1.2	运动学跟踪误差模型	57
4.1.3	动力学跟踪误差模型	58
4.2	基于 PID 的纵向控制器	60
4.3	基于前馈 LQR 的横向控制器	61
4.4	实车轨迹跟踪验证	65
第 5 章	虚实融合实车验证	70
5.1	虚实融合测试	70
5.2	实车实验验证	70
5.2.1	低速车辆占道通过测试	72
5.2.2	无信号合流汇入测试	74
5.2.3	环岛汇入汇出测试	76
5.2.4	泊车测试	77
第 6 章	结论与展望	80
6.1	全文总结	80
6.2	未来展望	81
参考文献		82
致谢		88
攻读学位期间获得与学位论文相关的科研成果		89

第 1 章 绪论

1.1 研究背景与意义

自动驾驶汽车作为一个新兴且快速发展的领域，在提高驾驶安全性和交通运输系统效率方面展现出巨大潜力。公安部数据显示，截至 2024 年 6 月，我国机动车保有量达 4.4 亿辆，其中汽车保有量达到 3.45 亿辆，且呈逐年上升趋势^[1]。车辆数量的增长也增加了交通事故发生的可能性。2024 年我国共发生交通事故 25.5 万起，其中机动车引发交通事故 21.2 万起^[2]。有观点指出，约有 70% 以上道路交通事故的发生都与驾驶员的操作直接相关^[3]。自动驾驶汽车的应用有可能大幅减少由于驾驶员错误操作或疏忽而造成的车辆碰撞。由中国汽车技术研究中心，同济大学和百度 Apollo 自动驾驶研发团队所共同撰写《自动驾驶汽车交通安全白皮书》中指出，人是交通安全中最不稳定的因素并提出了“自动驾驶比人类驾驶更安全，但非零事故”的观点^[4]。与人类相比，自动驾驶汽车拥有更加广阔的感知范围，更加持久的驾驶能力且行驶过程中严格遵守交通规则。有调查显示，如果将路面行驶车辆的九成换成自动驾驶汽车，交通事故的发生和因此导致的人员伤亡将会显著减少^[5]。同时，自动驾驶的发展有利于减少温室气体排放和提高交通效率^[6]。毋庸置疑，自动驾驶汽车已经成为未来智能交通运输系统的重要发展方向之一。

近年来，随着人工智能、深度学习等高新技术的发展，自动驾驶技术得到了广泛的关注和研究。与此同时，学术界和工业界都加大了对自动驾驶技术的研发力度，推动其不断发展和完善^[7]。目前，自动驾驶与辅助驾驶技术已经取得了长足的进步，在政策的促进下，开始迈入落地及应用阶段，工信部数据显示，2024 年 1 至 6 月，中国乘用车 L2 级新车渗透率达 55.7%^[8]，其中具备领航辅助驾驶功能的新车渗透率达到 11.0%，大量 L2 甚至 L3 级别的高阶先进辅助驾驶系统（Advanced Driver Assistance System, ADAS），如自动紧急制动系统（Autonomous Emergency Braking, AEB）和车道偏离预警系统（Lane Departure Warning, LDW），已在量产车辆中广泛应用，成功降低了人为失误引起的交通事故发生率^[9]。

尽管自动驾驶与辅助驾驶技术已经进入落地应用阶段，但是其开发和使用成本仍然居高不下，主要原因之一是，车辆的行车系统和泊车系统通常被作为两个独立的模块进行开发，并且两者所能调用的传感器和算力相互独立，存在较大的冗余。为了进一步降低高阶自动驾驶系统的落地成本，提升智能汽车算

力和传感器的利用率，行车域和泊车域作为两个独立开发的功能也逐步走向融合发展，行泊一体自动驾驶/辅助驾驶系统应运而生^[10]。行泊一体系统是将高阶行车辅助系统和自动泊车系统两个独立的系统模块，共同封装于一个自动驾驶域控制器中，使用一套硬件系统完成高速行车与低速泊车等智能驾驶功能，实现算力资源与感知资源的高效互联互通。行泊一体化设计，通过精简系统架构，打通数据链路，不仅能够有效降低自动驾驶系统的硬件成本，凭借其灵活的软件集成开发架构，使其能够轻松适应不同场景的需求。

典型的行泊一体自动驾驶系统架构如图1-1所示，可以大致概括为三个层次，环境感知层，运动规划层和控制执行层。环境感知层的主要任务为利用车载激光雷达，视觉传感器，毫米波雷达等传感器获取车辆周围的环境信息；依托组合导航和轮速计对车辆的实时状态进行估计，最后融合上述环境与状态信息，完成对车辆状态估计与感知融合。运动规划层需要根据环境感知与状态信息，对环境中存在的障碍物和交通参与者进行意图识别并进行态势评估，完成合理合规的交互式行车决策，输出安全可行，平稳舒适的行车轨迹。控制执行层则需要将运动规划层输出的轨迹信息，转换成具体的控制指令传达给底层的控制器，控制车辆完成加速，制动和转向等任务，使车辆能够按照上层输出预期轨迹行驶。

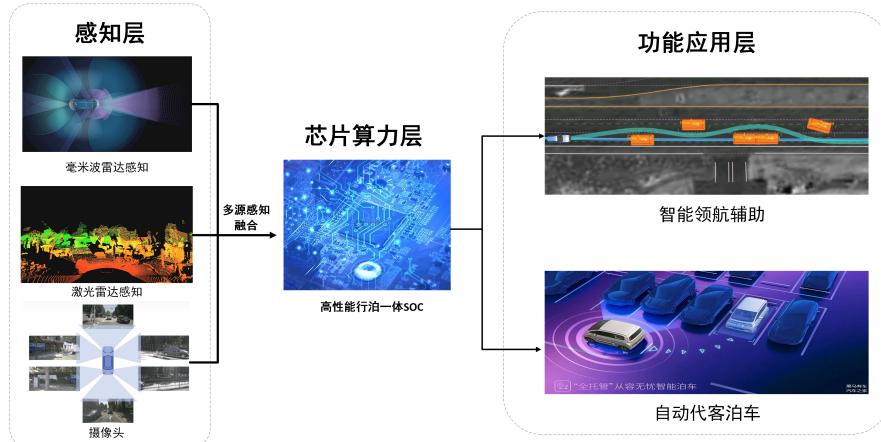


图 1-1 行泊一体自动驾驶系统架构图

由于自动驾驶技术的复杂性，目前行泊一体自动驾驶系统仍然处于初期发展阶段，距离大规模的商业化普及仍有一段距离^[11]。其主要原因有以下几点：首先，行车环境的复杂性和障碍物的多样性，动态性对车辆的环境感知系统提出了较高要求。并且，受限于感知系统的有效感知范围，定位精度，环境遮挡及交通参与者的行行为不确定性，完成实时准确的环境感知任务的难度进一步上升。其次，自动驾驶系统规划过程中需要充分考虑环境中交通参与者意图的不确定

性，感知的不确定性和约束的动态变化性，对道路环境的交通态势做出正确的研判，以一定的频率稳定输出合理的驾驶行为和可行的类人行车轨迹。不仅如此，由于行泊一体自动驾驶系统中多个功能集成于一个域控制器上，如何根据既定行车任务，合理地分配计算资源，在瞬息万变的交通环境中，实现实时感知与规划，同样充满挑战性。

运动规划模块是行泊一体自动驾驶系统的核心模块之一，其主要目标为根据车辆的运动状态和周围的交通环境，完成对应的决策和规划任务，输出安全高效的类人行车轨迹，完成在复杂交通环境中的行车任务。行车轨迹的质量将直接影响行车过程中的安全性，舒适性和稳定性。尽管智能车的运动规划算法在近年来有所发展，但是大多数算法的适用场景单一，在行泊一体系统中，由于行车域和泊车域的归一化，行车算法需要与泊车算法实现有机结合，在有限的算力空间下，不仅需要满足动态环境下的交互式高效行车规划的需求，而且能够完成极端狭窄空间的安全通过与泊车任务，完成点到车位的通行。因此，研究面向行泊一体的运动规划方法对自动驾驶系统提升运行效率，扩大应用范围，降低开发成本具有重要意义。

1.2 国内外研究现状

1.2.1 运动规划方法研究现状

运动规划方法可以大致分为五大类，分别是基于搜索的方法，基于采样的方法，基于数值优化的方法，基于曲线插值的方法和基于机器学习的方法^[12]。运动规划算法的质量好坏可以利用完备性和最优化来对算法进行描述，完备性是描述的是算法在理论上是否能够找到解（如果解存在），以及在什么条件下能够找到解，最优化描述则是算法找到的解是否是最优的。因此，一个优秀的运动规划算法应该同时具备完备性和最优化，以最大程度地满足实际需求^[13]。

(1) 基于图搜索的方法

基于图搜索的路径规划方法通过投影或栅格化将环境信息映射到图中，将地图表示为离散顶点的集合，顶点之间通过边连接形成可行路径，从而实现在图中搜索从起点到目标点的最优或可行路径。该类方法包括两个主要步骤，分别是图的构建和可行路径搜索。

经典的图构造方法主要包括 Voronoi 图法、栅格分解法等。Voronoi 图通过将空间划分为若干区域，每个区域包含一个生成点，区域内任意一点到该生成

点的距离比到其他生成点更近，能够反映空间特征（如障碍物分布）的网络。栅格分解法通过将连续的场景，离散化为多个栅格，根据栅格内是否包含障碍物判断可行状态。通过对环境的离散化栅格表示，一方面可以降低地图的信息密度，简化地图表示；另一方面，可以加速碰撞检测过程，提升运动规划效率^[14]。

在图搜索领域，深度优先搜索（DFS）和广度优先搜索（BFS）是两种最为经典的图搜索方法，通过对地图的大范围无向搜索来获得可行路径。在可行解存在的情况下，可以保证完成对求解空间的完备搜索，获得可行解，但是由于缺乏搜索方向，需要花费大量时间搜索，且由于缺少解的修正过程，路径同样也不能保证最优性^[15]。为保证在离散空间下的求解最优性，1959年Prim等人^[16]提出了dijkstra算法，可以计算拓扑地图中的单一源的最短路径，但是为保证最优性需要对所有节点进行搜索，随着地图栅格数量的增加，求解时间也将显著增加。Hart等^[17]提出了A*算法，在代价函数中加入了启发式代价，能够提升算法在搜索过程中的方向性，减少无序搜索，提升算法效率。但A*算法只能是解决静态环境下的规划问题。为了适应动态环境变化，实现连续的路径规划过程，Likhache^[18]提出了Dynamic A star (D star) 算法，当环境发生变化时，通过增量式更新方法，可以实现动态环境下路径的高效重规划，满足动态避障需要。

基于栅格地图搜索的方法的搜索效率受制于地图的大小和分辨率。当栅格过小，分辨率过大时，尽管算法可以保证解的完备性和最优性，但是算法的搜索效率将会显著降低，难以满足实时性的要求。当栅格过大、分辨率过低时，对障碍物、机器人形状的表示会非常粗糙，使得搜索出的路径质量显著降低，甚至无法完成求解任务，因此，这样一种方法，多适用于低速场景和离线的全局地图搜索^[19]。

(2) 基于采样的方法

基于采样的方法。基于采样的方法是一种流行的运动规划方法，尤其适用于高维空间和复杂环境。该方法是一种概率性算法，其核心思想为通过对求解空间进行随机采样或确定性采样以获得局部最优解，实现可行路径的求解。^[20]。随机采样方法具有概率完备性，即如果可行解存在，在采样时间充分的情况下一定可以找到该解。快速随机搜索树（Rapidly—exploring Random Tree,RRT）和概率路标图（Probabilistic Roadmap Method,PRM）算法是该类算法的两个重要组成部分。

概率路线图（Probabilistic Roadmap, PRM）是一种基于采样的路径规划算法，用于解决机器人在复杂环境中的路径规划问题^[21]。它的主要思想是通过随机采样方式在环境中建立路径网络图，将连续空间转换成离散空间，再利用搜

索算法(如 dijkstra, A* 等)在路线图上寻找路径。该类算法包括两个步骤,分别是采样学习和离散搜索。该类算法是概率完备的,只要学习的时间足够长且可行解存在,就一定能够找到可行解。通过密集的障碍物或狭窄通道时,PRM 方法的效率可能会降低。由于最终搜索的轨迹与采样点质量以及选取的连接距离有关,因此该算法也不具备最优性^[22]。

快速随机搜索树(RRT)算法在可达空间中随机生成采样点,并以起始点作为根节点,逐步向这些采样点扩展新的节点,从而构建出一棵随机树,当树中的某个节点与目标点的距离小于设定的阈值时,算法便认为找到了一条可行的路径。其核心优势在于其能够快速有效地搜索高维空间,并通过随机采样点将搜索导向空白区域,从而避免了对空间的精确建模带来的大计算量。这使得 RRT 算法能够解决高维空间和复杂约束的路径规划问题。但是由于随机采样的特性,在狭窄空间的突破过程中,需要花费较大的时间,也一定程度上影响来算法的效率,并且规划结果为离散的树节点连接构成,并不满足运动学约束,难以满足约束复杂的机器人的需要^[23]。为了解决这样一种问题, RRT 的一些改进算法被提出。JJK^[24]提出了 RRT-connect 算法,引入了双树拓展机制,以起点和终点为根节点分别构建两颗搜索树,显著提高了搜索速度,并减少了空白区域的无用搜索,应对狭窄走廊也有良好的表现,但是由于随机采样方法的局限性,所获得的解仍然是局部最优解。Karaman 等人^[25]提出了 RRT* 算法,引入了父节点选择策略,采样节点可以在邻域范围内选择最合适的节点,提升了求解轨迹的最有性,但是由于随机采样的特性,难以满足部分机器人的非完整性约束。Webb 等^[26]提出了 Kineodynamic RRT 算法,通过引入动力学约束,使得规划的路径满足机器人的动力学约束,但也使得算法的计算复杂度显著增加,难以满足实时性的要求。Gammell 等^[27]提出了 Informed RRT*, 在找到可行解后,算法并不会停止搜索,而是用椭圆包围路径,并在椭圆范围内继续采样,不断提升解的质量。近年来, RRT 算法的改进不断出现,如 neural RRT*^[28], anytime RRT^[29], KD-RRT^[30]等。

尽管基于采样的方法是概率完备的,但是该类方法在处理移动机器人的非完整性约束过程中就会显得力不从心,因为该类方法的搜索过程是基于采样点的,无法保证采样点的连续性。且由于随机采样的盲目性,难以获得符合复杂约束的高质量精确解。此外,由于采样点的随机性,容易使得前后两个相邻规划周期内的解差异较大,导致规划结果的不稳定性,影响机器人的实际表现。

(3) 基于曲线插值的方法

得益于计算机辅助技术和计算几何技术的发展,有学者开始将辅助几何制图领域的散点平滑方法应用于路径规划领域中,这样一种方法可以在给定一定

离散路径点的情况下，通过样条曲线的生成规则，对离散散点的位置进行平滑调整，完成平滑避障及导航路径的生成^[31]。常见的曲线插值方法包括多项式插值法，Dubins 插值法，Reeds-Shepp 插值法，B 样条插值法，贝塞尔插值法等。

最早用于运动规划的曲线是直线和圆弧，由于过渡平滑的特点，被用于表示一些简单场景下的车辆轨迹。但是，该组合存在连接过渡不平滑、曲率不连续的问题。^[32]。多项式曲线在运动规划过程由于其构造的便捷性和高阶连续性，在运动规划领域得到了较为广泛的应用，王明强^[33]提出了一种利用三阶多项式生成平滑避障轨迹的方法；WERLING^[34]综合运用五次多项式和四次多项式表示车辆的横向轨迹和纵向轨迹，完成了时空耦合条件下的优质可行轨迹的求解。Clothoid 曲线（回旋曲线），具有曲率线性平滑的特性，可以用来连接两个不同曲率的路径段，使其在道路设计、车辆和机器人路径规划、游戏开发等领域都有广泛的应用^[35]。Dubins 曲线是一种用于描述最短路径的几何方法，其核心思想是通过组合圆弧和直线段来构造从起点到目标点的最短可行路径，适用于满足非完整性约束的运动规划中，在轮式机器人/底盘规划中应用广泛。Reeds-Shepp 曲线是 Dubins 曲线的变种曲线，引入了对称求解的思想和允许前进后退的机制，被广泛应用于泊车和非结构化道路的运动规划当中，但是 Dubins 曲线和 Reeds-Shepp 曲线求解的为最短路，并不包含对障碍物的考虑，多与其他算法结合使用。类似地，贝塞尔曲线和 B 样条曲线也是参数化曲线，由于其高阶连续和凸特性，在运动规划中同样应用广泛。

（4）基于学习的方法

近年来，随着以机器学习与深度学习为代表的人工智能技术的发展，有学者开始运用这一类智能方法用于自动驾驶规划控制当中。该类运动规划方法由 NVIDIA 公司首先提出，通过设计深度卷积神经网络（Convolutional Neural Network, CNN），从摄像头获取的图像中提取图像特征，并完成图像与操作行为的映射，直接输出车辆的操控数据。由于不存在传统的感知-决策过程，因此该类方法也被称作端到端的运动规划方法。目前主流的智能规划方法主要包括基于深度模仿学习的方法和基于强化学习的方法两种^[36]。

基于模仿学习的方法通过模仿人类或其他智能体的运动行为，使机器人能够自主地规划出有效的运动路径和策略，这一类方法包含三种构造形式，包括行为克隆（Behavior Clone, BC），策略梯度法（Direct Policy Learning, DPL）和逆强化学习（Inverse Reinforcement Learning, IRL）^[37]。Bojarski 通过使用前视灰度相机来计算方向盘转角得到控制指令的方法搭建了 BC 框架，但是这种方法忽略了纵向控制信息，导致了应用的局限性^[38]。Codevilla^[39]提出了著名的条

件模仿学习 (Conditional Imitation Learning, CIL), 这种方法可以同时输出横向纵向控制信息，并丰富了比如灰度图像、自车速度、高阶决策等输入信息。CIL 的方法极大地丰富了 CL 这类端到端模型的发展前景。Wang^[40]在 CIL 的基础上丰富了更多的输入信息，比如全局规划、定位、点云信息等等。但是这样一种方法对于专家数据的质量提出了较高的要求，并且这样一种方法可能受限于专家样本数据的限制，使得算法的泛化性差，在陌生环境中表现不佳，并且由于行为克隆算法的局限性，这一过程仅仅在模仿专家的行为动作，缺乏自主学习和创新理解的能力。

基于深度强化学习的运动规划方法充分利用了深度学习的特征提取和函数逼近能力，以及强化学习的决策优化能力，通过深度神经网络，对环境进行感知和理解，提取出有用的特征信息，做出相应的抉择，通过与环境的高频交互，通过不断在环境中试错，使得智能体学习高维复杂状态空间与动作空间的映射关系，学习如何在复杂交通环境中做出正确的决策行为。Wolf 等人^[41]将 Q-learning 算法引入到自动驾驶领域，并且针对四种驾驶行为在 Gazebo 中完成了仿真实验，证明了该类算法在自动驾驶运动规划方法中的有效性。为进一步改善 Q-learning 算法在自动驾驶领域的表现，Alizadeh 等人^[42]提出将 DQN 于 DNN 结合，使得车辆的安全性和灵活性可以在驾驶中得到平衡。Ronecker 等人^[43]通过结合控制理论的 DQN 网络，提出了一种在高速场景下的更安全的方法，这表明了基于价值的 RL 可以在高速公路交通场景下产生高效和安全的驾驶行为。Hu 等人^[44]提出了 UniAD 一种以规划方向为引导的深度强化学习方法，重新对规划过程中的子任务的优先级进行排序和任务分割，使得所有子任务都能够服务于最终的运动规划过程。

(5) 基于数值优化的方法

依托对机器人系统的精准建模，运动规划问题可以被进一步描述为一个具有复杂约束的多目标优化问题。在问题的构建过程中可以综合考虑运动过程中的各种复杂约束和多样化的目标函数。相比与其他方法，不仅能够考虑车辆的运动学可行性，还能够优化运动轨迹的各项参数特征，具备更加出色的轨迹性能，并且获得的解是局部最优解甚至全局最优解，近年来应用广泛^[45]。一般来说，基于数值优化的规划算法可以归结为两类，一类是将运动规划问题构造为最优控制问题 (Optimal Control Problem,OCP) 求解，另一种则是将其构造为形态更为一般的非线性规划问题 (Nonlinear Programming Problem,NLP)，并利用数值求解。

最优控制问题的经典方法包括模型预测控制 (Model Predictive Control,MPC) 和线性二次调节器 (Linear Quadratic Regulator,LQR)，但是原始方法通常难以

直接运用于运动规划问题的求解。早在上世纪九十年代就有相关学者就利用最优控制求解运动轨迹展开了相关研究。Patten 等人^[46]针对平行泊车问题，构造数值优化问题并利用庞特里亚金极小值原理实现了最短时间的运动轨迹的解析式求解。为了解决 LQR 算法只能处理线性定长系统的问题，Todorov^[47]提出了 ILQR 算法，通过对系统的状态转移方程和代价函数进行一阶和二阶泰勒展开，将非线性系统在每次迭代中局部线性化，并利用二次近似来逐步优化控制策略，从而能够有效处理非线性系统的运动规划和控制问题，但是算法难以解决系统的约束条件。Chen^[48]提出了 CILQR 算法，将系统的约束条件线性化之后以控制障碍函数的方式加入了代价函数中，从而实现的 ILQR 的约束优化并在自动驾驶车辆中实现应用。Howell^[49]提出了一种结合增广拉格朗日方法 (Augmented Lagrangian Method, ALM) 和微分动态规划 (Differential Dynamic Planning, DDP) 的约束优化框架，以罚函数的方式将约束条件加入目标函数中，并利用 ALM 和 DDP 的双层迭代框架，不断更新惩罚因子，并求解当前的最优轨迹，实现了快速且鲁棒的非线性轨迹优化。Jun Ma^[50]在此基础之上将交替方向乘子法 (Alternating Directions Method of Multiplier, ADMM) 和 ILQR 相结合，利用分而治之的思想，通过一个复杂的约束优化问题分解为多个不同的子问题，分别求解，再通过 ADMM 的方式将子问题的解进行整合，实现了非线性约束下的高效运动轨迹求解。

模型预测控制 (model predictive control, MPC) 作为一种最优控制的数值解法，由于其基于模型的特性，也被应用于路径规划当中。该类方法的核心是在问题的构建过程中增加运动学可行约束和车辆微分模型的滚动优化过程，通过对最优控制问题的求解，可以获得典型求解空间内的最优路径。当求解空间是非凸的情况下，这将对问题的求解带来巨大的困难。学者 Zhang 等^[51]就这一问题提出了运动规划约束优化问题的松弛求解方法，通过舍弃部分硬约束条件，并将该部分约束添加仅目标函数当中，可以使得原始非凸优化问题满足 KKT 条件，保证解的可行性与局部最优性。He^[52]进一步引入了变步长机制和热启动方法，并将终点约束转换为损失函数的形式，实现了同一问题的高效求解。但是这样一种问题构型在障碍物广泛存在的开放空间内，仍然难以满足规划的实时性要求因此 Li 等^[53]，提出了一种通行走廊方法，通过对障碍物的膨胀和车辆的质点化处理，可以将车辆与障碍物之间的碰撞约束进一步转换为在通行走廊内的约束，由此，所有的约束条件均可以转换为线性约束条件，实现了开放空间下的松弛求解。但是这种方法也会减少一部分求解空间。随着计算机性能和凸优化理论的发展，非线性模型预测控制也越来越多地被运用于运动规划当中，这类方法能够直接处理非线性碰撞约束，保证了求解空间的完备性和最优性。

与基于最优控制的规划方法有所区别的是，基于非线性规划的规划方法通常将规划问题构造为非线性优化问题，然后根据问题的形式选择对应的数值求解方法完成对运动规划问题的高效求解。Zhang^[54]为处理结构化道路下的运动规划问题，考虑车辆的运动特性，引入了横纵向解耦规划思想和 Frenet 坐标系，将三维空间下的非凸优化问题转换为两个而为空间下的曲线平滑问题，实现了可行轨迹的快速求解。类似地，Wang^[55]将安全势场引入运动规划当中，将碰撞约束转换为势场带来的损失，通过求解非线性优化问题，实现了车辆的平滑无碰撞路径生成。进一步地，Lim 针对结构化道路环境，提出一种混合规划方法，利用采样结果作为非线性优化问题热启动解，能够解决复杂环境下的运动规划问题。Li^[56]为解决横纵向解耦规划算法在大曲率弯道场景表现不佳的情况，提出了一种在笛卡尔坐标系下的 NLP 运动规划方法。Han^[57]利用车辆的微分平坦特性，将高维状态空间和非线性动力学问题转化为低维平坦输出空间的规划问题，并利用轨迹参数化方法，有效降低了问题的维度和复杂度，实现了运动规划问题的实时求解。

1.2.2 轨迹跟踪方法研究现状

轨迹跟踪的主要目标是使机器人在运动过程中能够准确、稳定地按照上层规划出的参考轨迹进行运动，是自动驾驶系统中的关键模块之一，其性能直接决定了自动驾驶汽车的运行稳定性、安全性和乘坐舒适性。目前，轨迹跟踪方法根据所采用模型的不同，主要可以分为以下三类：无模型跟踪方法、基于几何模型的方法和基于车辆模型的方法。

(1) 无模型跟踪方法

无模型跟踪方法是一类不依赖系统精确数学模型的轨迹跟踪技术，其核心思想是通过反馈控制或启发式规则实现目标轨迹的跟踪。这类方法通常具有实现简单、计算效率高和适应性强的特点，广泛应用于低速、动态环境简单的场景中。

PID（比例-积分-微分）控制是最经典的无模型跟踪方法之一。其通过调节比例、积分和微分三个参数，实现对系统误差的快速响应和稳定控制。PID 控制在工业控制和机器人领域得到了广泛应用，但是固定参数的 PID 控制器难以应对复杂动力学系统和非线性系统的控制需求，且对参数的选择较为敏感，容易出现震荡和不稳定现象。

Macadam 和 Guo 提出预瞄点的概念，通过根据车辆运动状态和预瞄点的设置，推测出车辆在预瞄点处和目标位置的偏差，并通过最小化预瞄点处的误

差来降低路径跟踪误差。Farag^[58]提出了一种基于实验和错误的 PID 参数调节方法，并利用交叉跟踪误差（CTE）作为轨迹跟踪 PID 控制器的输入，通过实验表明，该算法可以实现复杂条件下的轨迹跟踪。Ali^[59]将 PID 控制器与遗传算法 (Gene Algorithm,GA)，实现了 PID 控制器三参数的动态调整，有效提升了轨迹跟踪的效果，Hao 等人基于 PID 控制，并设计了基于当前车速和路径曲率的预瞄控制算法，在较大曲率处可以显著降低其横向控制偏差，提高了控制精度。

基于无模型自适应控制 (Model-Free Adaptive Control, MFAC) 的估计跟踪方法是一类不依赖系统精确数学模型的控制策略，通过在线学习和自适应调整控制参数，实现对目标轨迹的精确跟踪。这类方法在处理非线性、时变和不确定性系统时表现出显著优势。田涛涛等^[60]将轨迹跟踪问题转化为预瞄偏差角跟踪问题，利用动态线性化技术设计无模型自适应控制算法、伪梯度估计算法和伪梯度重置算法并在“中国智能车未来挑战赛”中证明了其有效性。段建民^[61]通过动态线性化技术将非线性系统转化为线性系统，并利用滚动优化策略在线计算控制输入，结合了无模型自适应控制 (MFAC) 和预测控制的优点，实现了高效的轨迹跟踪为了解决无人履带车辆在外部环境不确定性情况下对精确动力学模型的依赖，该研究提出了 MPC-MFAC 双层控制算法，实现了模型失配和环境不确定性情况下高效、鲁棒的轨迹跟踪方法。Liu^[62]提出了基于预瞄偏差偏航角的路径跟踪方法，利用部分形式动态线性化方法将车辆动态模型转换为具有时变伪梯度 (PG) 的离散时间数据模型，基于该模型设计了 PFDL-EMFAC 无模型自适应控制方法并在数学上证明了其收敛性。

基于强化学习 (Reinforcement Learning, RL) 的轨迹跟踪方法是一种数据驱动的控制策略，通过智能体与环境的交互学习最优控制策略，而无需依赖系统的精确数学模型。这类方法在处理复杂、非线性系统以及动态环境时表现出显著优势，近年来在自动驾驶、机器人控制等领域得到了广泛关注。

贺伊琳等人^[63]将车辆的横向控制问题描述为马尔科夫决策过程 (MDP)，利用 DDPG 算法结合深度神经网络和强化学习，能够在连续动作空间中优化控制策略并在仿真环境中验证了其有效性 Liu 等人^[64]针对城市道路环境中的轨迹跟踪问题，结合注意力机制和多任务密集奖励函数，设计了基于强化学习的轨迹跟踪方法，实现了传感器原始数据为输入，控制指令为输出的端到端轨迹跟踪。Zhao 等人^[65]综合利用多任务密集级奖励函数和自注意力机制，提出了一种基于强化学习的端到端轨迹跟踪方法，并在仿真环境中实现了高效轨迹跟踪。Wang 等人^[66]为提升强化学习模型在现实世界的可用性，提出了 SRL-TR2 算法，一种考虑安全约束的演员-评论家强化学习框架，并设计了轻量级控制指令和动作空间对齐方法实现了方针环境与现实世界之间的联系，在实车环境中实现了高效

轨迹跟踪。

(2) 基于几何模型的方法

基于几何模型的轨迹跟踪方法主要包括纯跟踪算法 (Pure Pursuit) 和 Stanley 算法。这两种方法均基于车辆与目标轨迹之间的几何关系以及车辆的阿克曼转向特性，通过计算车辆的位置、航向角与目标轨迹的相对关系，推算出车辆的方向盘转角，从而实现轨迹跟踪。

纯跟踪算法是最基础的几何模型跟踪方法之一，其核心思想是假设车辆在跟踪过程中，其后轴中心会与预瞄点相切，随后利用阿克曼转向几何关系反推出车辆的方向盘转角，从而使车辆沿着目标轨迹行驶^[67]。由于其实现简单、计算效率高，纯跟踪算法被广泛应用于低速场景下的轮式机器人中。然而，该方法对预瞄点的选择较为敏感，且在高速或复杂动力学场景中表现有限。

Stanley 算法是纯跟踪算法的一种改进方法，最早应用于 2005 年 DARPA 自动驾驶挑战赛。该算法通过综合考虑车辆当前位置与目标轨迹之间的横向偏差和航向角偏差，并结合车辆速度和前视距离，动态计算前轮转向角，从而实现更精确和稳定的轨迹跟踪^[68]。与纯跟踪算法相比，Stanley 算法在低速场景下表现更为鲁棒，且能够更好地适应曲线路径的跟踪需求。

基于几何模型的轨迹跟踪方法具有计算效率高、模型简单和易于实现等优点，特别适用于低速场景下的自动驾驶系统。然而，在高速或动力学复杂的场景中，由于忽略了车辆的动力学特性（如惯性、轮胎力等），这类方法的性能可能显著下降，难以满足高精度和高稳定性的控制需求。

(3) 基于车辆模型的方法

基于几何模型的轨迹跟踪方法通常基于车辆刚体假设和无侧滑假设，这类假设在低速场景下能够较好地满足实际需求，但在高速或复杂动力学场景中，由于轮胎侧滑、惯性效应等因素的影响，几何模型的假设不再成立，从而导致跟踪性能下降。因此，基于动力学模型的轨迹跟踪方法逐渐受到广泛关注。这类方法通过建立车辆的动力学模型，综合考虑车辆的惯性、轮胎力、摩擦力以及悬架特性等动力学因素，将轨迹跟踪问题描述为最优控制问题，并通过求解最优控制指令，实现对目标轨迹的精确跟踪。常用的基于车辆动力学模型的最优控制算法包括模型预测控制 (MPC)、线性二次调节器 (LQR) 和滑模控制 (SMC) 等。这些方法通过优化控制输入，能够在高速、高动态场景中实现更鲁棒和精确的轨迹跟踪。

Raffo 等人^[69]为提升控制器的在复杂环境中的适应性，提出了一种结合运动学和动力学模型预测控制器，实现了对车辆的高精度控制。倪兰青等人^[70]将预瞄误差模型与二自由度车辆动力学模型相结合，构建了基于 LQR 的 4WS 轨迹跟踪控制器，在 4 轮独立转向平台实现了精准轨迹跟踪。梁忠超等人^[71]提出一种自适应 MPC 跟踪方法，通过更新增益系数矩阵及控制器状态补偿车辆的非线性以及观测噪声带来的影响，提升了轨迹跟踪器的稳定性。陈亮等人^[72]构建了基于“前馈 + 反馈”的 LQR 控制器对期望前轮侧偏力进行实时求解并利用刷子轮胎模型将控制量转化为期望前轮转角，提升了大曲率高速工况下的横向稳定性和跟踪精度。张雷等人^[73]针对四轮轮毂电机独立驱动平台的轨迹跟踪问题，提出一种分层架构的轨迹跟踪控制器，上层通过 MPC 求解最优控制前轮转角和横摆力矩，下层进行轮毂电机最优化分配，提升了极限工况下的跟踪精度和稳定性。Luan 等人^[74]考虑随机网络延迟引起的转向角振荡，建立了车辆线性不确定时滞系统，提出了不确定模型自适应模型预测控制算法 (UM-MPC)，并在硬件在环仿真中验证了随机网络延迟环境下的稳定性。Cao 等人^[75]对车辆动力学模型和轮胎模型进行一定的简化，提出了一种考虑转弯特性的线性时变模型预测控制器 (LTV-MPC)，提升了车辆在弯道段跟踪的稳定性。Chu 等人^[76]为解决传统 MPC 算法模型过渡简化带来的稳态误差，利用人工势场获得跟踪轨迹并在 MPC 控制器中增加了 PID 反馈，在跟踪精度和转向平顺性方面的性能有所提高。Wang 等人^[77]利用微分平坦特性将复杂的动力学车辆模型转化为三个具有相同结构的全驱动子系统并基于李雅普诺夫框架和状态稳定性理论设计了一种抗观测扰动轨迹跟踪控制器并在仿真环境中得到了验证。

相比于基于几何模型的方法，基于动力学模型的轨迹跟踪方法能够更精准地考虑车辆的动力学特性、性能极限以及道路环境的影响，从而显著提升车辆的行驶稳定性、跟踪精度和安全性。

1.3 主要研究内容及技术路线

1.3.1 课题来源

本文受国家自然科学基金项目：重载车辆安全运行控制（52322217）资助。

1.3.2 主要研究内容

本文以面向行泊一体的自动驾驶运动规划问题为主要研究内容，以提升自动驾驶在结构化道路上的行车舒适性，安全性和非结构化道路上的通过性，稳定性为目标。行泊一体自动驾驶系统的细分使用场景可以概括为结构化道路下的动态交互式行车规划和非结构化道路下的极端狭窄空间通过和泊车规划。本文以上述两个主要应用场景为研究方向，展开相关研究。

具体研究内容和相关章节安排如下：

第一章 绪论。对行泊一体自动驾驶系统的出现契机和国内外自动驾驶系统发展现状进行系统性的阐述。对现今各类智能车运动规划方法研究进展进行分析总结，概括各方法的优劣，分析行泊一体自动驾驶系统的需要，提出本文的研究重点。

第二章 面向极端狭窄空间的运动规划方法。针对狭窄空间下混合 A* 算法搜索效率低，灵活性差的问题，提出自适应变精度搜索策略，提升算法在狭窄空间下的搜索速度和搜索精度。为进一步提升混合 A* 算法所获得的原始可行轨迹质量，本文将极端狭窄空间下的运动规划问题描述为一个最优控制问题。针对狭窄空间下障碍物多，碰撞约束复杂的特点，构造增量式矩形通行走廊将非凸非线性的碰撞约束简化为线性约束，使最优控制问题的复杂度独立于环境的复杂度。综合考虑运动过程中的乘坐舒适性，行车安全性和稳定性，构造最优控制问题，以混合 A* 算法所获得的轨迹作为原始可行解，使用内点法求解器 (IPOPT) 对该最优控制问题进行数值求解，获得通行走廊内部的局部最优可行解，并通过仿真验证本文算法多种不同场景下的表现，证明其有效性。

第三章 面向动态交互式场景的运动规划方法。为满足结构化道路下的行车导航需要，基于 Opendrive 车道级高精地图，构建车道级邻接矩阵，并使用 dijkstra 算法，实现车道级全局路径规划。针对结构化道路环境下速度快，交互强，道路平坦和过渡平滑的特点，利用 Frenet 坐标系对车辆横纵向运动解耦进行规划，在离散空间下构建 ST 图和 SL 图，完成安全可行凸空间提取，将横纵向运动规划问题构造成分段平滑的二次规划问题，综合考虑运动过程中的舒适性，稳定性和安全性，完成平滑可行的轨迹求解。为进一步提升行车过程中的灵活性，设计了考虑通行效率的换道决策方法。并搭建仿真平台，验证了算法的有效性。

第四章 面向行泊一体的轨迹跟踪控制器。对车辆的单轨模型，运动学误差模型和动力学误差模型进行了描述和推导。将轨迹跟踪问题描述为最优控制问题，详细介绍线性二次调节器 (Linear Quadratic Regulator,LQR) 算法的推导和求解过程。为解决基于 LQR 的轨迹跟踪算法在横向误差模型在高速运动过程

中稳态误差稳定存在的问题，通过公式推导，完成稳态误差的解析计算并通过增加曲率前馈，提升轨迹跟踪算法的稳定性，通过实车实验验证了算法的有效性。

第五章 虚实融合实车实验。为进一步体现本文所提出算法的有效性，利用虚实融合自动驾驶测试工具链在国内某实车测试场构建了 11 个测试场景（包含 10 个行车场景和 1 个停车场景）进行了相关测试，本文算法在上述场景下均表现良好，有效体现了本文算法的有效性和稳定性。

第六章 总结与展望。对全文的研究内容进行总结归纳，分析本文算法的优缺点，并对未来的工作进行展望。

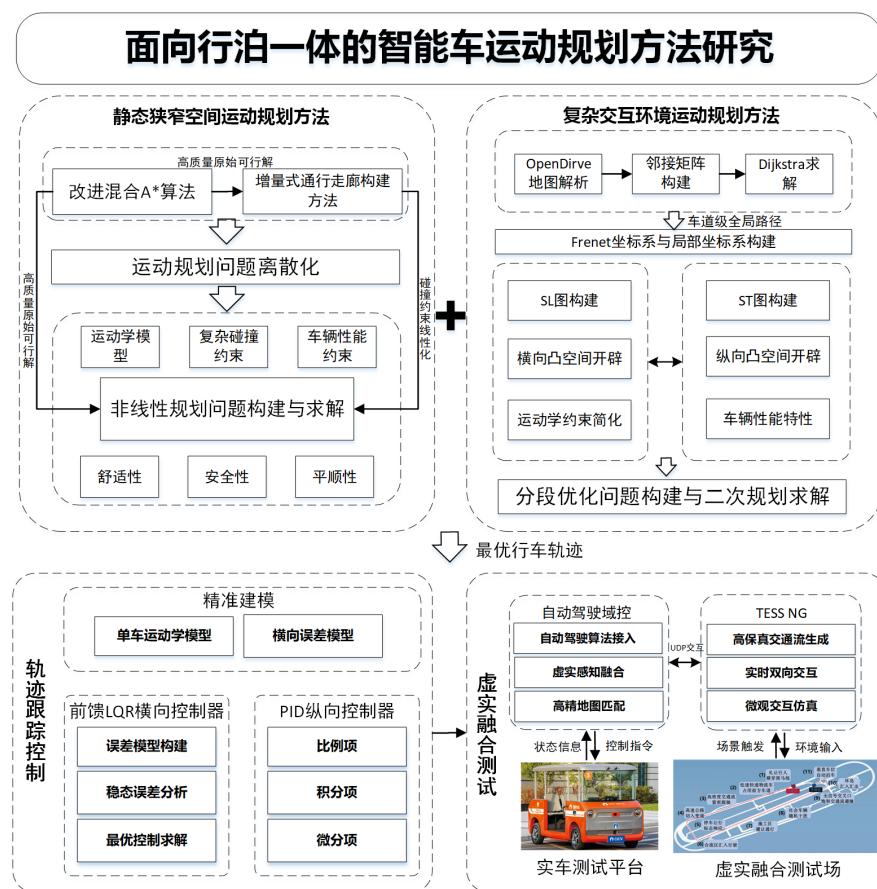


图 1-2 本文技术路线图

第 2 章 面向极端狭窄空间的运动规划方法

极端狭窄空间指环境中可供移动物体（如车辆、机器人等）通行的自由空间宽度接近或小于其物理尺寸的最小安全裕度的区域，通常表现为通道宽度仅略大于移动物体本身的宽度。以泊车为代表的典型极端狭窄空间下的运动规划问题的主要目标是在狭窄空间的可行驶区域中规划出一条平滑，高效且运动学可行的轨迹，引导车辆从起点，通过狭窄区域并安全地到达预设的终点。即搜索一组符合运动学约束的动作序列 $u(t)$ ，通过执行该控制序列能够控制车辆安全且平稳地通过狭窄空间，到达终点。

由于车辆运动学约束和碰撞约束具有较强非线性甚至非凸性，使得问题的复杂度相较于开阔环境下的运动规划问题大幅提升，呈现出求解难度大，计算耗时高等特点。为解决上述问题，本文提出了一种基于通行走廊的分层运动规划算法，该算法首先利用改进混合 A* 算法，完成在极端狭窄空间内的原始可行解计算，然后利用增量矩形拓展方法，构建车辆从起点至终点的通行走廊，将非凸非线性的碰撞约束转换为线性的 Box 约束，最后将狭窄空间的运动规划问题构建为最优控制问题，综合考虑车辆的运动学约束，起终点约束和安全约束。完成过程平顺，运动学可行，途中无碰撞的轨迹求解。

2.1 基于改进混合 A* 的可行解搜索方法

运动规划问题通常会被精心构造成一个包含众多复杂非线性约束的数值优化问题。这类问题的目标是在满足一系列物理、环境及安全约束的条件下，找到一条从起始点到目标点的最优路径。然而，这些约束条件往往交织在一起，形成非凸的优化空间，导致问题的解空间呈现出高度的不规则性和多模态性。因此，数值优化算法在求解此类问题时，常常会陷入局部最优解的困境，难以跳出当前搜索区域去探索可能存在的更优解^[13]。

在这种情况下，找到更高质量的原始可行解就显得尤为重要。高质量的初始解不仅能够为后续的优化过程提供一个良好的起点，还有助于引导搜索算法在解空间中更有效地探索，从而增加发现全局最优解的可能性。这是因为，在复杂的非凸优化问题中，局部最优解可能遍布整个解空间，而全局最优解则隐藏在这些局部最优解的“海洋”之中。通过提供一个接近全局最优的初始解，可以显著减少搜索算法陷入较差局部最优解的风险，同时加快收敛速度，提高求解效率。

2.1.1 A* 算法原理

A* 算法，作为一种经典的启发式路径规划算法，自 1968 年由 Hart 提出以来，便以其高效性和灵活性在栅格地图和有向图中的单源路径规划问题中占据了举足轻重的地位。该算法的核心在于其启发式搜索策略，它结合了广度优先搜索和深度优先搜索的优点，通过引入一个启发式函数来评估从当前节点到目标节点的最优路径的代价估计，从而指导搜索过程沿着最有希望的方向进行的同时减少了无效搜索的产生。

在栅格地图的路径规划问题中，A* 算法通常能够较快地找到最优解，且相比与 dijkstra 算法，无用节点拓展量大幅减少。经典 A* 算法的伪代码如算法 1 所示，包含横向，竖向和斜向，共三个搜索方向，8 个搜索栅格。尽管 A* 算法简单高效，但是由于如下几个缺陷，使得 A* 算法难以应用于智能车的运动规划问题当中：

1. 规划路径为栅格的中心点，仅包含 x, y 两个维度，缺少 yaw 信息，难以满足车辆的精准位姿控制。
2. 未考虑车辆的档位信息，无法满足车辆前进-后退档位切换的需求。
3. 未考虑车辆的运动学约束，使得轨迹难以被车辆跟踪。
4. 栅格搜索策略中，横向搜索和斜向搜索难以完全满足车辆的运动学约束。

2.1.2 混合 A* 算法原理

为解决 A* 算法难以满足运动学约束的问题，提升栅格搜索算法的适应性和稳定性。混合 A* 算法于 2007 年在 Drapa 自动驾驶挑战赛期间被提出，通过引入额外的状态空间和栅格内搜索迭代过程，使得算法能够在满足复杂约束条件的前提下，高效地找到高质量的路径解^[78]。其搜索策略如图 2-2 所示，将 A* 算法的搜索状态空间由 (x, y) 增广至 (x, y, yaw) ，加入了关于 yaw 的约束。且相比与 A* 算法仅能在栅格处搜索，混合 A* 算法能够在栅格内部进一步搜索，路径可以由栅格中任意点的位置组成，这极大地提高了路径的平滑性。在节点拓展过程中，通过模拟车辆的前进/后退档位切换和转向行为，来计算下一个待探索节点的位置，具体的节点拓展公式如式2-1所示：

$$\begin{cases} x_n = x_c + d_{step} \cos(yaw_c + yaw_n)g \\ y_n = y_c + d_{step} \sin(yaw_c + yaw_n)g \end{cases} \quad (2-1)$$

式中， x_c, y_c, yaw_c 为当前节点状态， x_n, y_n 为待拓展节点的位置， yaw_n 为航向角变化量， d_{step} 为搜索步长， g 为档位。混合 A* 算法模拟车辆前进后退及转

向的思想，弥补了 A* 算法不满足运动学非完整性约束的缺陷，极大提升了栅格搜索算法的适应性，使得该算法能够广泛应用于轮式机器人的运动规划中。

混合 A* 算法与经典 A* 算法的代价函数基本保持一致，如式2-2所示：

$$f(n) = g(n) + h(n) \quad (2-2)$$

式中， $g(n)$ 为历史代价， $h(n)$ 为启发式代价。 $g(n)$ 代表由起点出发，到达当前探索节点所需要花费的代价， $h(n)$ 代表由当前探索节点到终点的预估代价。A* 算法的历史代价通常仅包含距离代价一项，但是与 A* 算法所不一样的是，混合 A* 算法的目标并不仅仅是找到从起点到终点的路径，我们仍然希望找到的这样一条路径能够符合正常车辆的行驶习惯，如减少前后档位切换，减少转向行为，尽可能保持前进档位。基于上述考虑，在设计历史代价的过程中，加入了档位切换，转向行为，后退档位行驶的代价，完整的历史代价函数可以写作2-3式：

$$g(n) = \sum g_{direction} + g_{gear} + g_{turning} + g_{distance} \quad (2-3)$$

式中， $g_{direction}$ 为行驶方向代价， g_{gear} 为档位切换代价， $g_{turning}$ 为转向代价， $g_{distance}$ 为行驶距离代价。启发式代价 $h(n)$ 通常表示从当前探索节点出发，到达终点可能需要花费的代价，主要用于评估不同路径的优劣，从而指导搜索算法选择最优路径进行扩展。由于估计过程中难以直接考虑障碍物对实际路径所带来的影响，随着 $h(n)$ 愈发逼近实际路径代价，算法所需探索的搜索空间显著缩减，具体表现为搜索节点数量的减少以及整体搜索时间的缩短。鉴于这样一种特征，为提高算法的启发性效率，增大启发函数在算法决策过程中的权重，可以引导算法更加侧重于启发式信息的利用，从而优化搜索路径并加速求解过程。

一般来说，在混合 A* 算法中，启发式代价可以分为两种，一种是考虑非完整性约束的启发式代价 $h_{nonholonomics}$ ，另一种考考虑避障的避障的启发式代价 $h_{collision}$ 。完整的启发式函数可以写作2-4式：

$$h(n) = \max(h_{collision}, h_{nonholonomics}) \quad (2-4)$$

式中， $h_{nonholonomics}$ 为考虑非完整性约束的启发式代价，负责记录仅考虑车辆的运动学约束，而不考虑运动过程中的安全约束的距离代价，通常使用 Reeds Shepp 曲线的最小长度来计算。 $h_{collision}$ 为考虑运动过程中安全的启发式代价，通常使用栅格搜索的方法（如 A* 算法）来估算该值。算法最终的启发式代价为 $h_{collision}$ 和 $h_{nonholonomics}$ 中的较大值。

混合 A* 算法作为一种基于栅格搜索的路径规划算法，在地图分辨率较高或可行空间狭窄的环境中，难以在有限时间内完成高效可行路径的搜索。此外，由于栅格搜索算法的特性，栅格搜索的终点难以完全和设定的终点完全对应。为了解决这一问题，最经典的做法为在搜索过程中尝试使用 Reed Shepp 曲线直接连接当前探索节点和终点，实现探索节点与终点之间点对点的连接，弥补栅格搜索无法完全到达终点的缺陷的同时加快搜索过程。

Reed Shepp 曲线是 Dubins 曲线的变种曲线，由 J.A.Reeds 和 L.A.Shepp 于 1990 年提出，是一种用于描述机器人或车辆在二维平面上运动轨迹的曲线，尤其适用于需要考虑前进和后退情况下，找到一条连接起始状态和目标状态且满足曲率约束的最优路径。Reeds Shepp 曲线在 Dubins 曲线的基础上，允许切换车辆行驶方向，这在某些情况下能够找到比 Dubins 曲线更短的路径。Dubins 曲线有 6 种组合方式，而 Reeds Shepp 曲线为每一个运动基元都增加了前进 (+) 和后退 (-)，因此组合方式大幅增加。Reeds Shepp 曲线总共有 48 种组合方式，详细的组合方式如表2-1所示，但是不同组合曲线之间存在镜像、翻转的关系，因此并不需要对每一种组合都进行计算，可以通过时间变换、反射变换和逆向变换等技巧来快速计算剩下的情况。

表 2-1 Redds Shepp 曲线组合方式

基础组合类型	组合顺序
$C C C$	$(L^+R^-L^+)(L^-R^+L^-)(R^+L^-R^+)(R^-L^+R^-)$
$CC C$	$(L^+R^+L^-)(L^-R^-L^+)(R^+L^+R^-)(R^-L^-R^+)$
$C CC$	$(L^+R^-L^-)(L^-R^+L^+)(R^+L^-R^-)(R^-L^+R^+)$
CSC	$(L^+S^+L^+)(L^-S^-L^-)(R^+S^+R^+)(R^-S^-R^-)$ $(L^+S^+R^+)(L^-S^-R^-)(R^+S^+L^+)(R^-S^-L^-)$
$CC_3 C_3C$	$(L^+R_3^-L_3^-R^-)(L^-R_3^+L_3^+R^+)(R^+L_3^-R_3^-L^-)(R^-L_3^+R_3^+L^+)$
$C C_{\pi/2}SC$	$(L^+R_{\pi/2}^-S^-R^-)(L^-R_{\pi/2}^+S^+R^+)(R^+L_{\pi/2}^-S^-)(R^-L_{\pi/2}^+S^+L^+)$ $(L^+R_{\pi/2}^-S^-L^-)(L^-R_{\pi/2}^+S^+L^+)(R^+L_{\pi/2}^-S^-R^-)(R^-L_{\pi/2}^+S^+R^+)$
$CSC_{\pi/2} C$	$(L^+S^+L_{\pi/2}^+R^-)(L^-S^-L_{\pi/2}^-R^+)(R^+S^+R_{\pi/2}^+L^-)(R^-S^-R_{\pi/2}^-L^+)$ $(R^+S^+L_{\pi/2}^+R^-)(R^-S^-L_{\pi/2}^-R^+)(L^+S^+R_{\pi/2}^+L^-)(L^-S^-R_{\pi/2}^-L^+)$
$C C_{\pi/2}SC_{\pi/2} C$	$(L^+R_{\pi/2}^-S^-L_{\pi/2}^-R^+)(L^-R_{\pi/2}^+S^+L_{\pi/2}^+R^-)$ $(R^+L_{\pi/2}^-S^-R_{\pi/2}^-L^+)(R^-L_{\pi/2}^+S^+R_{\pi/2}^+L^-)$

2.1.3 改进混合 A* 算法

相比于传统 A* 算法，前文中所介绍的混合 A* 算法已经能够在开放空间内完成平滑无碰撞的行车轨迹搜索，且得益于 RS 曲线的解析拓展策略，可以解决

在复杂稀疏栅格地图中，起终点难以适配的问题，显著提升算法的求解成功率。但是经典混合 A* 仍然存在几点明显的不足之处，使得算法的效率无法达到最大化。具体包括如下几点：

1. 固定的采样步长将会使得算法的灵活性下降，在部分极端环境难以求解。
2. 算法缺乏对行车规律的考虑，无效转向较多。
3. 固定间隔拓展缺乏拓展成功可能性的考虑，降低算法的拓展效率。

为了解决上述问题，本文将分别从变精度节点搜索和变间隔解析拓展两个方面对经典混合 A* 算法进行改进。

考虑车辆在行车过程中更倾向于直线行驶和小幅度转向行驶，因此，在搜索过程中，应当适当增加直线采样的距离，此外，大幅度的转向会使得行车的舒适性明显下降，但是适当减少大幅度转向的搜索步长可以适当降低带来的影响。基于上述考虑，本文提出一种变精度节点搜索方法，其主要实现过程如下。

节点拓展过程中，单侧拓展的子节点拓展个数 N_{expand} 公式2-5进行计算：

$$N_{expand} = \text{int}\left(\frac{\theta_{max}}{\Delta\theta}\right) \quad (2-5)$$

式中， θ_{max} 为最大拓展转向角， $\Delta\theta$ 为转向拓展间隔。在节点拓展过程中，如果栅格过大，搜索步长过小，容易使得搜索节点难以跳出栅格，降低节点的拓展效率，同时，过长的搜索步长容易降低搜索精度，降低可行解的最优性。因此在进行变精度拓展的过程中，应当同时考虑最小搜索长度和最大搜索长度。具体搜索步长 L 可以通过公式2-6进行计算：

$$L = L_{max} - \frac{L_{max} - L_{min}}{N_{expand}} \left(\left| \frac{\theta_{expand}}{\Delta\theta} \right| \right) \quad (2-6)$$

式中， L_{max}, L_{min} 分别为最大，最小节点拓展步长， θ_{expand} 为节点搜索角度。变精度拓展的搜索和经典混合 A* 算法的搜索示例如图2-1所示。经典混合 A* 的单次搜索呈现均匀分布的扇形，而变精度混合 A* 算法呈现出转向采样更短，直线采样更长的特点，更加贴合行车实际情况，能够更快地进行直线段地探索，同时也尽可能减少了转向节点的行驶长度。

经典的解析拓展策略通常会每个一定的间隔，尝试直接连接当前节点与目标节点，但是以给定间隔直接连接忽略了当前节点与目标节点之间的距离以及连接路径中的障碍物分布情况，考虑狭窄通行环境，频繁地进行解析拓展和碰撞检测尝试会大量消耗算力，影响算法的实时性。通常来说，当前节点距离节点越近，解析拓展的成功率越高，因此在算法的搜索前期，应当尝试更多地向目标节点搜索，减少不必要的解析拓展，以提升算法前期的搜索效率；在靠近目标

节点之后，在搜索的过程中更多地进行解析拓展，以提升算法后期的求解效率。

为了更好地指导算法进行高效地解析拓展，本文提出一种变步长解析拓展策略。在算法搜索初期，更多地进行栅格搜索工作，减少不必要的解析拓展。随着节点向终点靠近，增加解析拓展的次数，提升算法后期的求解成功率。具体的解析拓展间隔 N_{RS} 可以被表示为公式2-7：

$$N_{RS} = \max(1, \text{floor}(k_{rs} \cdot \frac{h(n_{cur})}{h(n_{start})})) \quad (2-7)$$

式中， k_{rs} 为缩放因子， $h(n_{cur})$ 为当前节点的启发式代价， $h(n_{start})$ 为起点的启发式代价

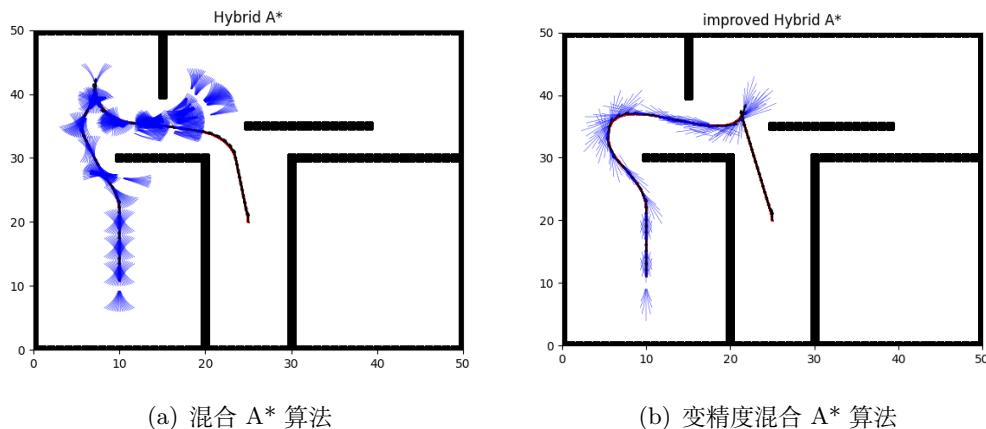


图 2-1 混合 A* 算法与变精度混合 A* 算法比较

2.2 基于增量矩形拓展的安全走廊构建方法

在轨迹优化问题中，碰撞约束相比于其他约束，由于障碍物外形的非凸性和碰撞约束强非线性等特点，往往更难以处理。为了对环境进行高效、精确的建模，给出车辆与障碍物之间的明确的相对关系，许多研究集中于利用符号距离、可微分碰撞检测算法和分层优化解决这一问题^[79]。但是对于一般的轨迹优化问题中，为了保证完整运动过程的安全性，优化问题中需要加入车辆与每个障碍物之间的无碰撞约束，随着障碍物的增加，优化问题的规模也将进一步增加，为实时求解安全运行轨迹带来了较大的冗余计算量，降低了规划问题的求解实时性。

如图2-2所示，存在两条可行路径引导车辆从起点行驶至终点，但是从这两条路径出发，并不是所有障碍物都需要被考虑。若沿路线 1 行驶，给行车安全

算法 2-1: 改进混合 A* 算法

Input: 二维地图信息 map , 起始位姿 (x_s, y_s, yaw_s) 和终止位姿 (x_g, y_g, yaw_g)

Output: 连接始末状态的可行路径 $Path$

```

1: 初始化 OpenList=∅, CloseList=∅, iter=0;
2: while OpenList!=∅ do
3:   iter ← iter + 1;
4:   从 OpenList 中取出 f 值最小的节点, 并记为当前节点  $Node_{cur}$ ;
5:   计算解析拓展间隔, 记为  $N_{rs}$ 
6:   if iter - iterRS > Nrs then
7:     使用 RS 曲线连接当前节点  $Node_{cur}$  和目标节点  $Node_{goal}$ 
8:     if PathRS 可行 then
9:       从当前节点  $Node_{cur}$  开始回溯获取到起点的路径  $Path_{search}$ 
10:      将  $Path_{search}$  和  $Path_{RS}$  拼接成完成的可行路径  $Path_{output}$ 
11:      iterRS = iter
12:      break
13:    end if
14:  end if
15:  计算子节点拓展长度, 进行模拟拓展, 获得子节点  $Node_{child}$ , 并保存至集合  $\Gamma$  中
16:  for  $Node_{child} \in \Gamma$  do
17:    if  $Node_{child} \in CloseSet$  then
18:      Continue;
19:    else
20:      if  $Path_{cur \rightarrow child}$  Collision then
21:         $Node_{child}.is\_Collision = 1$ 
22:        Continue
23:      else
24:        设置  $Node_{child}$  的父节点为  $Node_{cur}$ 
25:        计算节点  $Node_{child}$  的代价  $f = g(Node_{child}) + h(Node_{child})$ 
26:        if  $Node_{child} \in OpenList$  then
27:          if  $g_{new}(Node_{child}) < g_{old}(Node_{child})$  then
28:            更新节点  $Node_{child}$ , 并对 OpenList 重新排序
29:          end if
30:        else
31:          将  $Node_{child}$  加入 OpenList
32:        end if
33:      end if
34:    end if
35:  end for
36: end while
37: if OpenList == ∅ then
38:   无可行解
39: else
40:   从终点回溯获得  $Path_{output}$ 
41:   return  $Path_{output}$ 
42: end if

```

带来威胁的障碍物仅包括 $O_1 - O_5$ ，如果将车辆与障碍物 $O_5 - O_7$ 之间的碰撞约束加入优化问题中，则会显著提升轨迹优化问题的复杂度。找到一种简化碰撞约束的方法，减少不必要的碰撞约束，是降低轨迹优化问题复杂度，提升求解效率的可行途径。

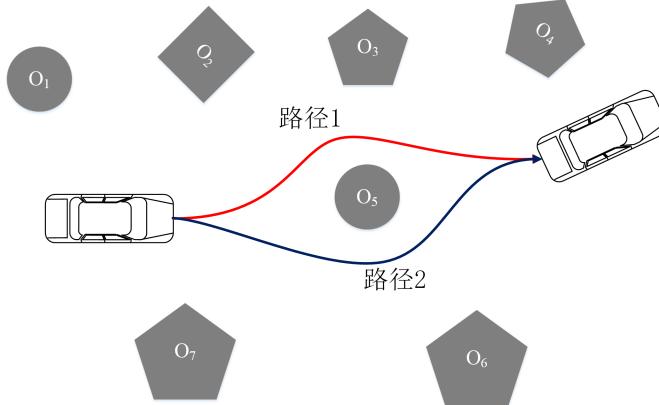


图 2-2 车辆在复杂环境下的多种可行路径

本文将安全走廊 (Safe Corridor) 这一碰撞约束简化方法引入车辆轨迹优化问题中。这一概念在解决无人机的大规模非凸轨迹优化问题中被首次提出，其本质是对安全的自由空间进行凸分解，将轨迹限制在这些凸多边形内，从而实现避障约束的简化^[80]。如图2-3所示，该方法通过构建一系列的连续矩形通行走廊将无人机的可行驶区域与危险碰撞区域区分开来，将非线性非凸的碰撞约束转换为满足 KKT(Karush-Kuhn-Tucker) 条件的在凸多边形内部的约束，使得轨迹优化问题的计算复杂度独立于环境的复杂度，不会随着障碍物数量的增加而上升，能够有效保证算法的实时性和复杂度。

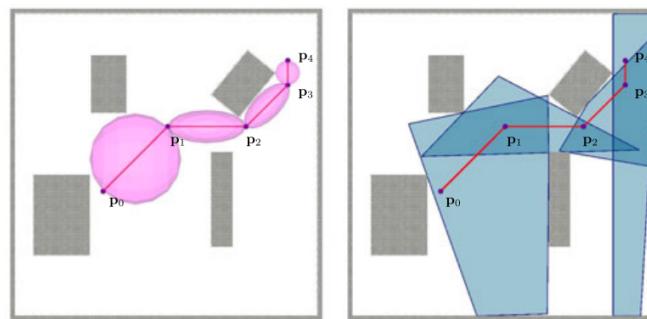


图 2-3 无人机飞行走廊

四旋翼无人机可以近似地看作椭球体，在对环境中的障碍物以一定的膨胀半径膨胀之后，可以近似地看作质点。但是车辆通常为矩形，以单个圆对车辆

进行包络，会使得车辆侧面的避碰行为过于保守，在部分狭窄空间难以完成可行轨迹求解。为了解决这一问题，本文选择使用双圆包络模型，对车辆的形状进行近似。如此操作之后，车辆与障碍物之间的碰撞条件由车辆与凸多边形不相交转变为，车辆的前轴中心和后轴中心位于相应的通行走廊内，其对应的数学描述可以写作公式2-8：

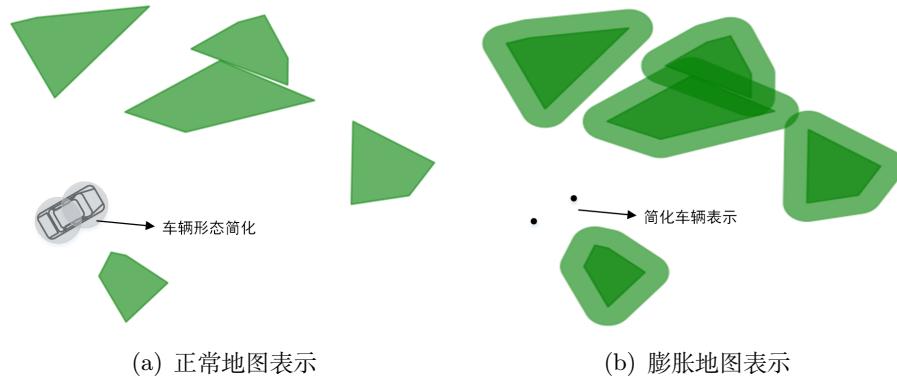


图 2-4 膨胀地图与车辆形态简化示意图

$$\mathbb{E}(x_k) \cap \mathbb{O} = \emptyset \Rightarrow \begin{bmatrix} x_{min}^f \\ y_{min}^f \\ x_{min}^r \\ y_{min}^r \end{bmatrix} \leq \begin{bmatrix} x^f \\ y^f \\ x^r \\ y^r \end{bmatrix} \leq \begin{bmatrix} x_{max}^f \\ y_{max}^f \\ x_{max}^r \\ y_{max}^r \end{bmatrix} \quad (2-8)$$

包络圆的大小和位置应当与车辆的长度和宽度呈现一定的关系，在本文中，车辆的前后两个包络圆的圆心位置 $P_f = (x_f, y_f)$ 和 $P_r = (x_r, y_r)$ 可以通过公式2-9确定：

$$\begin{aligned} x_f &= x + \frac{1}{4}(3L_W + 3L_F - L_R) \cdot \cos(\theta) \\ y_f &= y + \frac{1}{4}(3L_W + 3L_F - L_R) \cdot \sin(\theta) \\ x_r &= x + \frac{1}{4}(L_W + L_F - 3L_R) \cdot \cos(\theta) \\ y_r &= y + \frac{1}{4}(L_W + L_F - 3L_R) \cdot \sin(\theta) \end{aligned} \quad (2-9)$$

式中， (x_f, y_f) 为前包络圆中心， (x_r, y_r) 为后包络圆中心， L_W 为车辆轴距， L_F 为前轴至车辆前端的距离， L_R 为车辆后轴中心至后端的距离。完成车辆形状的简化和膨胀地图构建之后，下一步就是在膨胀地图上构建通行走廊。迄今为止，通行走廊都由凸多边形组成，相较于其他凸多边形和圆，矩形由于在栅格

地图上构建的便利性和碰撞约束的强线性，具备更好的可操作性和可行性，因此本文选择使用增量式矩形拓展方法来构建通行走廊。

构建通行走廊，首先需要一段包含 N 个路径点可行的粗轨迹，该轨迹可以通过前文2.1中的混合 A* 算法获得。局部通行走廊的具体构建方法如算法2-2所示：从某一可行路径点出发，初始化四个拓展方向的拓展宽度为极小值，即初始走廊初始化为一个变长为极小值的矩形，以 Δs 为拓展长度向外侧进行拓展，沿顺时针对走廊进行拓展，如果矩形与障碍物无碰撞，则接受该拓展结果，反之则撤销此次拓展（不结束拓展），并标记该拓展方向为碰撞拓展方向，后续拓展过程中将跳过该方向，直至拓展次数超过最大推展次数 N_{max}^{expand} 或四个拓展方向均发生碰撞。

完整的局部走廊拓展过程如图2-6所示，以笛卡尔坐标系 y 轴正半轴为拓展起始拓展方向，在顺利完成三次拓展之后，在第 4,5,9 次拓展过程中，局部走廊与膨胀之后的障碍物产生了重叠，这表明车辆的关键点如果落在该区域，存在一定的碰撞风险，因此选择放弃这几次拓展的结果并终止该方向的局部走廊拓展；直至到达最大拓展次数，最终形成完整的局部走廊，输出沿路径四个方向的拓展距离大小 $(x_{min}, x_{max}, y_{min}, y_{max})$ 。

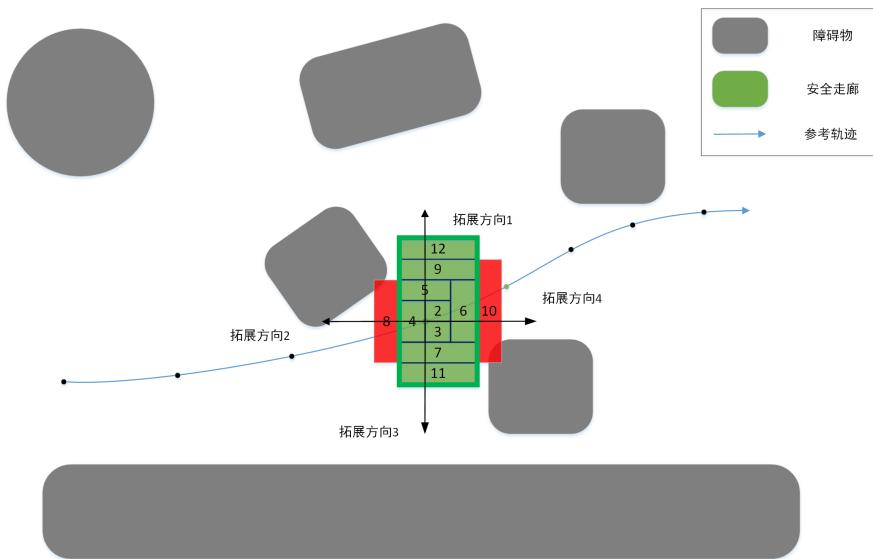


图 2-5 局部通行走廊的构建过程

完整的通行走廊构建算法如算法2-3所示，对一段包含 N 个路径点的轨迹，前轴包络圆中心 P_f 和后轴包络圆中心 P_r 的局部通行走廊进行单独求解，构建共计 $2 * N$ 个局部通行走廊，即可完成整体的通行走廊构建。对一段可行解的每个离散时刻进行局部走廊的构建，即可获得一段连续轨迹的通行走廊，图2-6为一段完整泊车轨迹所构建的通行走廊，其中红色矩形为后轴包络圆的所构建的

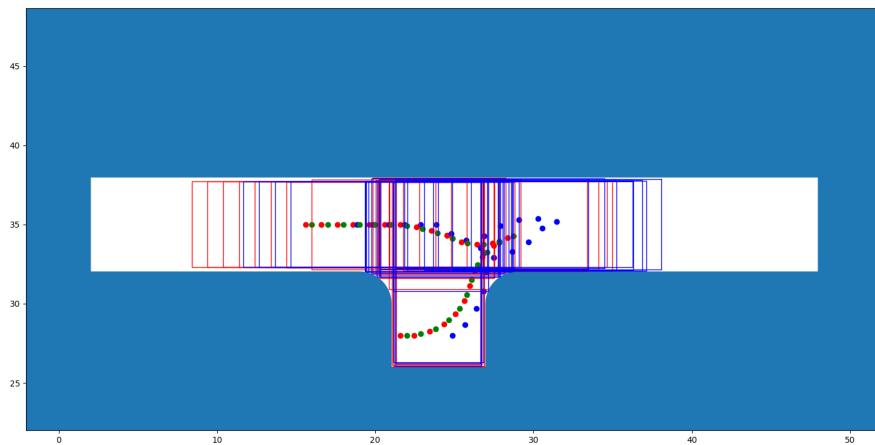


图 2-6 狹窄环境下通行走廊的构建过程

通行走廊，蓝色矩形为以前轴包络圆所构建的通行走廊，使用本文算法所构建的通行走廊可以完整覆盖车辆的完整运行轨迹，为之后的轨迹优化打下良好的基础。

2.3 基于安全走廊的后端轨迹优化方法

尽管混合 A* 算法所获得的路径在运动学上是切实可行的，但是由于算法本质上仍然是基于栅格搜索的规划算法，所获得的路径是由一段一段的栅格之间的平滑路径所构成，常常包含一些不必要的转向行为，且不包含沿路径的速度信息。为了获取更加平滑，高效的运动轨迹，本文选择将狭窄空间的轨迹优化问题描述为最优控制问题，综合考虑车辆的性能限制和碰撞约束，利用数值优化方法，完成安全，平滑且舒适的轨迹求解。

2.3.1 轨迹优化问题

车辆的轨迹优化问题是指，在已经获取起始状态和目标状态之间的可行轨迹之后，对可行轨迹进行进一步的平滑，获得一条满足车辆性能限制，安全高效且无碰撞的轨迹。该问题的核心在于求解一个性能指标最优的轨迹。这个性能指标可以是时间、能量、轨迹平滑性等。而最优控制问题中的目标函数正是用来衡量系统性能的指标，通过优化这个指标可以找到最优的控制策略。因此，本文选择将智能车的轨迹优化问题建模为最优控制问题。

最优控制问题的分类方式多种多样，车辆的轨迹优化问题多被描述为连续 Bolza 形式的最优控制问题，该类问题通常具备如下特征：

算法 2-2: 局部走廊生成算法

Input: 障碍物信息, 车辆位置信息

Output: 局部安全走廊

```

1: 初始化拓展顺序  $\Omega_{ID} \leftarrow \{1, 2, 3, 4\}$ 
2: 初始化拓展角度顺序  $\Omega_{direction} \leftarrow \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}, 0\}$ 
3: 初始化起始拓展点  $\gamma \leftarrow (x_c, y_c)$ 
4: while  $\Omega_{ID}! = \emptyset$  do
5:   for  $i \in \Omega_{id}$  do
6:      $\gamma_{expand} \leftarrow$  拓展局部走廊 ( $\gamma_{approved}, \Omega_{direction}[i]$ )
7:     if 拓展走廊  $\gamma_{expand}$  无碰撞 then
8:        $\gamma_{length}[i] \leftarrow \gamma_{length}[i] + ds$ 
9:        $\gamma_{approved} \leftarrow \gamma_{expand} \cup \gamma_{approved}$ 
10:      if  $\Omega_{length}[i] > L_{max}$  then
11:        从拓展顺序  $\Omega_{ID}$  中删除第 i 个值
12:      end if
13:    else
14:      从拓展顺序  $\Omega_{ID}$  中删除第 i 个值
15:    end if
16:  end for
17: end while
18:  $x_{min} \leftarrow x_c - \Omega_{length}[3]$ 
19:  $x_{max} \leftarrow x_c + \Omega_{length}[1]$ 
20:  $y_{min} \leftarrow y_c - \Omega_{length}[2]$ 
21:  $y_{max} \leftarrow y_c + \Omega_{length}[4]$ 
22: return 局部可行走廊  $[x_{min}, x_{max}, y_{min}, y_{max}]$ 

```

1. 控制输入是连续的时间函数, 是一个连续控制系统。
2. 目标函数包含多个组成部分, 具备符合性能指标。
3. 包含终末状态约束, 但末状态时间不确定。
4. 系统状态方程是随时间变化的非线性方程, 是时变非线性系统。

该问题的目标为求解容许控制序列 $u(t)$, 并确定到达终点的目标时刻 t_f , 使得被控对象状态 $x(t)$ 在控制序列 $u(t)$ 下, 目标函数 $J(x, u)$ 取得局部最小值。对这样一个最优控制问题, 目标函数和约束条件包括:

$$\begin{aligned}
J(x(t), u(t)) &= \sum x(t)^T Q x(t) + u(t)^T R u(t) \\
\frac{\partial dx(t)}{\partial t} &= f(x(t), u(t), t) \\
\varphi(t_0, t_f, x(t_0), x(t_f), u(t_0), u(t_f)) &= 0 \\
C(x(t), u(t), t) &\leq 0, t \in [t_0, t_f]
\end{aligned} \tag{2-10}$$

算法 2-3: 全局通行走廊生成算法

Input: 障碍物信息, 原始可行轨迹 $Path$
Output: 完整安全走廊集合 Γ

```

1:  $[p_f, p_r] \leftarrow CoarseTraj(Path)$ 
2:  $\Gamma \leftarrow \emptyset$ 
3: for  $k = 1, 2, \dots, N$  do
4:    $[xr_{min}^k, xr_{max}^k, yr_{min}^k, yr_{max}^k] \leftarrow$  局部走廊生成  $(p_r.x, p_r.y)$ 
5:    $\Gamma \leftarrow \Gamma \cup [xr_{min}^k, xr_{max}^k, yr_{min}^k, yr_{max}^k]$ 
6:    $[xf_{min}^k, xf_{max}^k, yf_{min}^k, yf_{max}^k] \leftarrow$  局部走廊生成  $(p_f.x, p_f.y)$ 
7:    $\Gamma \leftarrow \Gamma \cup [xf_{min}^k, xf_{max}^k, yf_{min}^k, yf_{max}^k]$ 
8: end for
9: return  $\Gamma$ 
```

其中系统状态方程约束, 用于在离散空间下车辆的状态更新规律和物理特性, 即车辆在给定控制量 u 和离散时间间隔 t 的前提下, 车辆状态量 x 的变化规律; 边值约束通常指在运动规划过程中, 规划的起始时刻 t_0 和目标时刻 t_f 的状态与起始状态和目标状态一致, 保证规划问题的构建与实际问题一致; 性能约束在车辆运动规划中起着至关重要的作用, 它主要受到车辆性能的限制, 如速度、加速度等需要约束在一定范围内, 以保证车辆在运动过程中的安全性和稳定性。同时, 为避免与障碍物发生碰撞, 还需要考虑路径约束和障碍物约束。

2.3.2 轨迹优化问题构建与求解

在2.3.1节提到了轨迹优化问题的一般形式, 即公式2-10, 但是该问题中的目标函数和约束条件并没有进一步给出, 接下来对该问题的具体构型展开进一步的分析。

为了更好地利用问题中的梯度信息, 该问题的目标函数使用平方误差函数进行构建。对狭窄空间的安全行驶问题, 主要考虑以下几个方面:

1. 行驶的舒适性。根据国际化标准组织 (International Organization for Standardization, ISO) 发布的相关标准研究显示, 行车过程中, 对舒适性影响最大的是横纵向的加速度。参考该标准, 对应控制量为车辆的前轮转角 δ_f 和加速度 a , 舒适性的损失可以进一步被表示为公式2-11:

$$cost_c = \sum w_1 a_i^2 + w_2 \delta_f^2 \quad (2-11)$$

式中: w_1 为加速度权重系数, w_2 为前轮转角权重系数

2. 时间的最优性。本文期待完整的行驶过程能够尽快完成, 所以在目标函

数加入关于行驶时间的惩罚项，具体下作公式2-12:

$$cost_t = \sum w_3 t_i^2 \quad (2-12)$$

式中: w_3 为时间权重系数, t_i 为第 i 个步长的时间长度。

3. 路径长度最短。由于混合 A* 搜索出来的路径包含大量的多余转向行为, 该方法虽然确保了路径的可行性, 但是不能保证路径是足够短且光滑的, 这样一段轨迹仍然存在较大的提升空间, 因此本文在目标函数中加入关于轨迹长度的损失项, 具体写作下式2-13:

$$cost_d = \sum w_4 ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) \quad (2-13)$$

式中, w_4 为距离权重系数, $(x_i, y_i), (x_{i-1}, y_{i-1})$ 分别为车辆在 i 时刻和 $i+1$ 时刻的位置。

4. 运动学约束违背最少。对于 Bolza 形式的最优控制问题来说, 如果直接以硬约束的行驶将运动学约束加入问题当中, 将会显著增加这一问题的复杂度, 使得问题的求解的实时性降低。因此, 本文选择软化该硬约束, 以损失函数的行驶, 最小化轨迹的运动学约束违反程度, 具体可以写作公式2-14:

$$cost_g = \sum w_5 \|f(x(t), u(t))\|^2 \quad (2-14)$$

式中, w_5 为运动学约束权重系数, $f(x(t), u(t))$ 为车辆运动学约束。

构造完成目标函数之后, 需要进一步地明确问题的约束条件, 对于智能车的运动规划问题, 主要考虑如下几个方面:

1. 车辆运动学约束。在车辆运动过程中, 主要考虑车辆运动学约束。本文使用二自由度自行车模型来表述车辆的行驶过程, 考虑将完成的行驶时间分解为 k 个离散时间段, 在惯性坐标系下, 车辆的运动过程受到如下约束:

$$\frac{d}{dt} \begin{bmatrix} x_t \\ y_t \\ v_t \\ \phi_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} v_t \cos \theta \\ v_t \sin \theta \\ a_t \\ \phi_t \\ v_t \tan \phi_t / L_W \end{bmatrix} \quad (2-15)$$

2. 车辆性能约束。受到车辆自身的运动性能限制, 其动力输出和转向输出

需要约束在一定的范围内，具体写作下式：

$$\begin{bmatrix} v_{\min} \\ a_{\min} \\ -\delta_{\max} \\ -\omega_{\max} \end{bmatrix} \leq \begin{bmatrix} v_k \\ a_k \\ \delta_k \\ \frac{\delta_{k+1}-\delta_k}{\tau_k} \end{bmatrix} \leq \begin{bmatrix} v_{\max} \\ a_{\max} \\ \delta_{\max} \\ \omega_{\max} \end{bmatrix} \quad (2-16)$$

3. 安全无碰撞约束。为保证车辆在行驶过程中的安全无碰撞，利用前文中2.2节中的通行走廊，可以将强非线性的无碰撞约束转换为线性的 Box 约束，可以写作下式：

$$\begin{bmatrix} x_{\min}^f \\ x_{\min}^r \\ y_{\min}^f \\ y_{\min}^r \end{bmatrix} \leq \begin{bmatrix} x^f \\ x^r \\ y^f \\ y^r \end{bmatrix} \leq \begin{bmatrix} x_{\max}^f \\ x_{\max}^r \\ y_{\max}^f \\ y_{\max}^r \end{bmatrix} \quad (2-17)$$

4. 起终点边值约束。轨迹优化问题中，初始时刻的车辆位置通常为车辆的当前位置，终点为期望达到的终点位置。为保证优化问题与实际问题的一致性，需要保证车辆的规划起始时刻 t_0 与结束时刻 t_f 的状态与起点状态和目标状态一致。

$$\begin{aligned} [x_{t_0}, y_{t_0}, \theta_{t_0}, v_{t_0}, a_{t_0}, \delta_{t_0}] &= [x_{start}, y_{start}, \theta_{start}, v_{start}, a_{start}, \delta_{start}] \\ [x_{t_f}, y_{t_f}, \theta_{t_f}, v_{t_f}, a_{t_f}, \delta_{t_f}] &= [x_{goal}, y_{goal}, \theta_{goal}, 0, 0, 0] \end{aligned} \quad (2-18)$$

上述最优控制问题，可以使用非线性数值优化求解器完成求解，但是由于通行走廊约束的存在，使得车辆会被约束在构建的通行走廊中，使得该非线性最优控制完成求解后，虽然是可行且光滑的，但是仍然不是局部最优解。为了进一步提升解的质量，使其进一步向局部最优解靠近，本文使用迭代求解的方案，在完成最优控制问题求解之后，保存当前的可行解，然后重新生成通行走廊，构建最优控制问题并求解，直至相邻两次的求解结果目标函数之差小于收敛阈值或到达最大迭代次数，完成问题的求解。

完整的求解过程如下，首先通过混合 A* 算法获取原始可行解，初始化问题的参数；然后构建计算膨胀半径，完成膨胀地图的构建；接着进入迭代求解循环中，以当前的可行解为路径点，构建通行走廊，并构造最优控制问题，并对问题进行迭代求解，直至问题收敛，最优提取可行解为最优轨迹。

2.4 仿真验证

为进一步验证本文所提出的运动规划方法的有效性，本节通过在不同的场景下展开仿真实验，对算法的稳定性和适应性进行验证和分析。考虑典型狭窄空间通行工况，包含泊车和狭窄空间行驶。仿真实验在编程环境 Python 3.7，硬件平台为搭载 32GB RAM 的 intel-i2100F cpu 上运行，求解器选择原始对偶内点求解器 IPOPT 进行求解，相关参数如下表2-2所示：

表 2-2 仿真实验参数设置表

参数名称	参数描述	设置
N	采样点个数	50
Δs	通行走廊单次拓展宽度	0.3m
S_{max}	通行走廊最大拓展宽度	7m
w_1	目标函数舒适性权重	10
w_2	目标函数舒适性权重	100
$iter_{max}$	最大迭代次数	5
$WheelBase$	车辆轴距	3.5m
$Length$	车辆长度	5.5m
v_{max}	最大速度	10km/h
a_{max}	最大加速度	1m/s ²
ϕ_{max}	最大前轮转角	0.6rad
ω_{max}	最大前轮转角速度	0.2rad/s

工况一：狭窄车位停车工况

OnSite 自动驾驶泊车测试平台是国内多个科研平台基于严格的场地和场景设计所联合开发设计的泊车仿真测试环境，支持垂直、平行、倾斜车位等多种泊车场景的需求^[81]。本文在选择 OnSite 自动驾驶泊车测试环境下验证本文算法在狭窄车位泊车环境下的表现。如图2-7所示，OnSite 泊车测试环境共包四张不同类型的停车场地图，共包含 159 个停车位，覆盖了垂直泊入，垂直进入，水平泊入，斜向泊入等多种常见车位类型，可以有效测试自动驾驶泊车算法的性能表现。

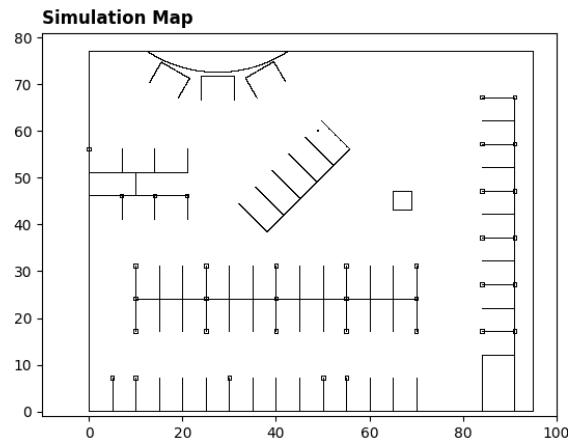


图 2-7 OnSite 泊车泊车测试环境

本文算法在四种最为常见的车位表现如图2-8所示，红色轨迹为混合 A* 算法的规划结果，绿色轨迹为本文算法的优化结果，经过本文算法的优化，泊车轨迹不仅变得更加的平滑且简洁，前后档位切换次数也有效减少，不必要的转向行为消失，行车轨迹更加拟人。

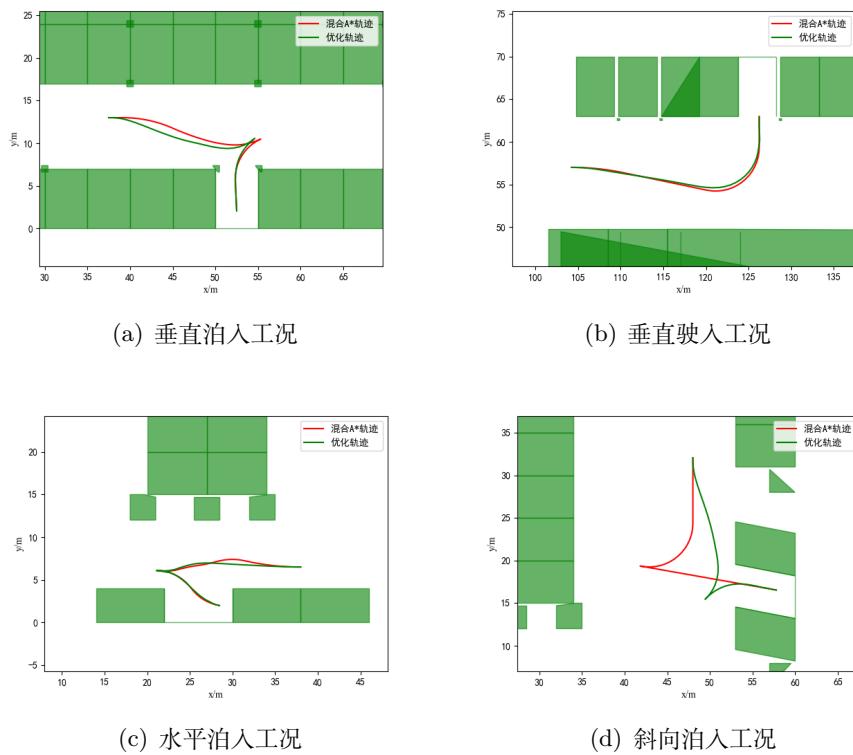


图 2-8 泊车入位规划结果

如图2-8中所示，经过本文算法的优化，多个不同场景下的泊车轨迹均得到了不同程度的平滑优化，其中斜向入位泊车尤为显著，运动轨迹长度被有效缩短，原前后挡位切换点由车位的正前方转变为车位的左前侧。此外，混合 A* 算法所得到的路径在进入车位的过程中，为了在空间内进行更加完备的探索，所以进行了多次的转向模拟，这一操作也直接导致了冗余转向次数的增加，使得乘坐的舒适性明显下降，如水平泊入工况，经过本文算法进一步优化后，无效转向减少，整个泊车过程变得更加简洁且高效。

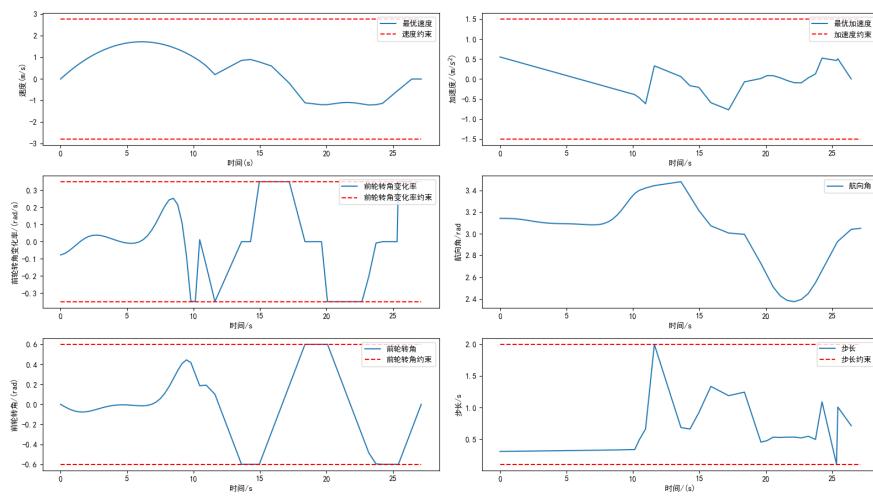


图 2-9 水平泊入过程运动参数

以水平泊车入位为例，规划起点为 $(30, 6.5, \pi)$ ，规划终点为 $(28, 2, \pi)$ ，优化求解后的相关指标如图所示。车辆的各项运动参数均在约束范围内，整体运动过程中，状态过渡平滑，运动过程中最大加速度为 $0.6 m/s^2$ ，为急加速和急减速行为，保证了车辆纵向的稳定性。运动过程中，考虑了车辆的车轮转角极限及车辆的角速度约束，减少了运动过程中的大幅度及转向行为，提升了算法在横向稳定性方面的表现。此外，由于通行走廊约束的存在，车辆也并未出现紧贴障碍物行驶的情况，也进一步保证了运动过程中的安全性。

工况二：障碍物随机布置通行工况

为进一步体现本文算法在通行随机布置通行工况下的表现，本文选择构建多个狭窄空间通行场景对本文算法进行有效性测试。本文算法在对应的四个障碍物随机布置场景下的表现如图2-10所示，相较于混合 A* 所给出的原始可行解，经过本文算法的优化之后，均可以得到不错的表现。

以 case2 为例，规划起始位姿为 $(10, 10, \frac{\pi}{2})$ ，规划终点位姿为 $(25, 10, \frac{\pi}{2})$ ，图2.11(a)展示了本文算法所构建的通行走廊，其中红色为后轴包络圆的通行走

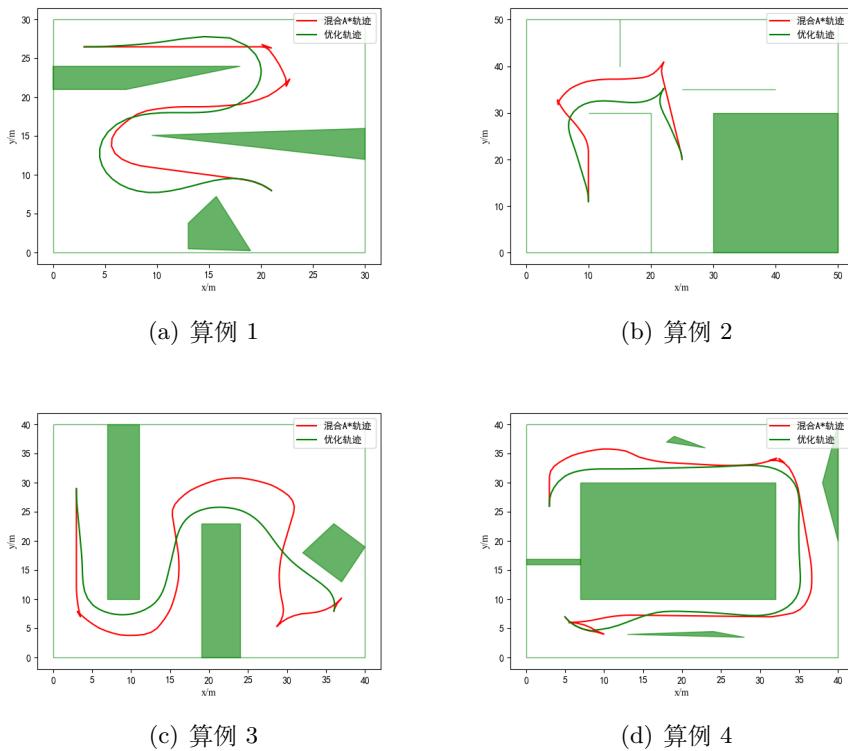


图 2-10 本文算法在障碍物随机布置情况下的表现

廊，蓝色走廊为前轴包络圆的通行走廊，本文算法所构建的通行走廊能够完整覆盖混合 A* 算法的规划结果每一时刻的位姿。混合 A* 算法的原始可行轨迹长度为 66.24 米，而经过优化后，轨迹长度减少到了 52.10 米，本文算法算法成功地将轨迹长度缩短了 21.3%，揉库次数从 3 次减少为 1 次

图5-7展示了以混合 A* 算法为原始可行解优化之后的轨迹，本文算法的优化求解结果能够较好地满足运动规划问题地各项约束，速度 v ，加速度 a ，前轮转角 ϕ ，前轮转角速度 ω 均在车辆的运动学约束范围内，且整体过渡平滑，有效地保证了车辆在运动过程中的运行稳定性和安全性。

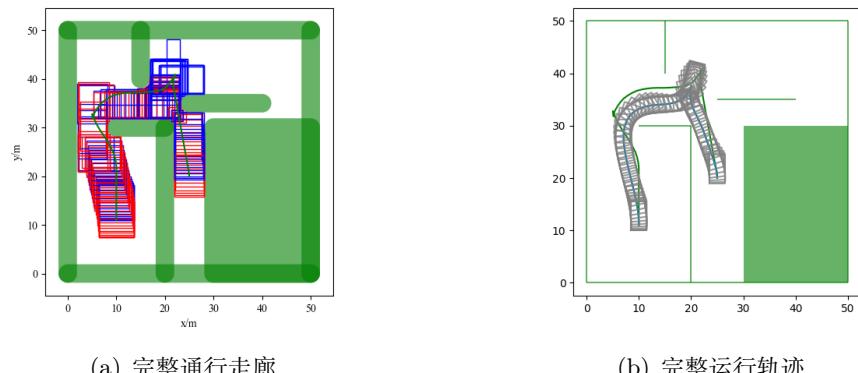


图 2-11 算例 2 规划结果

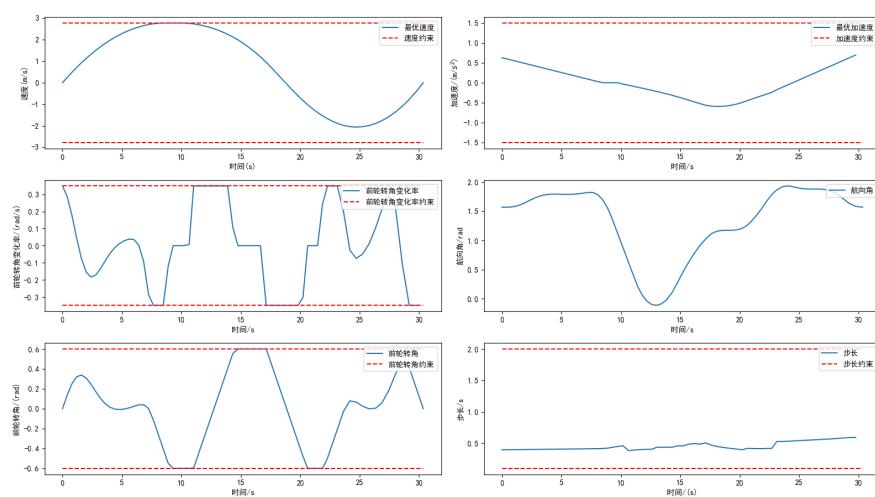


图 2-12 本文算法在障碍物随机布置情况下的优化结果

第3章 面向动态交互环境下的运动规划方法

在本文第二章中，已实现了极端狭窄空间下的运动规划求解，但因其未充分考虑动态交通参与者的交互行为，难以适应城市道路等高动态、强交互场景的规划需求。相较于停车场等非结构化狭窄空间，结构化道路环境具有标志标线清晰、通行规则明确、动态风险源多样等特征。在此类复杂场景中，车辆运动规划不仅需要满足动力学约束和行驶安全要求，同时需兼顾交通法规的合规性约束，这对规划算法的多维度约束处理能力提出了更高要求。

为提升车辆在动态交互环境下的适应性和求解实时性，保证车辆的安全行驶的同时提升通行效率，本节提出一种基于优化的 Frenet 坐标系下的横纵向解耦的运动规划方法和考虑通行效率的换道决策方法。

3.1 动态交互环境下运动规划问题分析

3.1.1 动态交互环境定义

动态交互环境通常指以城市道路和高速公路为代表的附着条件良好，标志标线清晰，障碍物车辆运动规律的结构化道路环境，通常具备如下特点：

1. 道路规则清晰：结构化道路通常指车道边沿规则、路面平坦的公路，如城市主干道、高速公路等。这些道路的设计使各断面组成的线性要素之间保持相互均衡，并且对于同一个设计路段，设计标准尽量保持一致。
2. 标志线明显：具有鲜明的车道线和其他道路标志，如道路指示牌、交通信号灯等。这些标志线为驾驶员提供了明确的行车指引，有助于保持行车方向和速度的稳定。
3. 几何特征明显：几何特征明显，如道路宽度、曲率、坡度等参数都符合设计规范。这些特征使得道路在视觉上更加清晰，也便于驾驶员进行驾驶。
4. 背景环境单一：结构化道路的背景环境相对单一。道路两旁通常没有复杂的障碍物或干扰物，这使得驾驶员可以更加专注于道路前方的行车情况。
5. 行车条件理想：单车道路宽不低于一定标准、道路两旁的侧向余宽也符合要求。此外，行车视野开阔、纵断面坡度平缓、平面线性缓和路况良好等特点也使得行车更加安全舒适。

鉴于结构化道路的上述特点，相比与狭窄空间，在该环境下智能车将会拥有多样化的行车选择，决策空间和交通规则，如换道，超车，汇入等，因此结构

化道路的运动规划不仅仅要考虑车辆的通行效率和安全行驶，还需要加入对交通合规性以及行车决策的考量。

3.1.2 交互环境下运动规划问题分析

针对3.1.1节中动态交互环境下的特点，车辆在行驶过程中确实需要综合考虑多个方面以确保安全、高效和合规。以下是这四个主要方面的详细解释：

1. 求解实时性。在动态交互环境中，车辆需要快速响应周围的变化，包括其他车辆的突然变道、行人的横穿马路、交通信号的快速切换等。因此，运动规划算法必须能够在极短的时间内计算出可行的行驶路径，以满足实时性的要求。
2. 行驶安全性。安全是车辆行驶的首要考虑因素。在复杂动态环境中，车辆需要能够预测其他交通参与者的行，并避免潜在的碰撞风险。这包括车辆需要采用先进的传感器技术（如雷达、摄像头、激光雷达等）来实时感知周围环境，识别并响应其他车辆的行驶轨迹、行人的行走路径以及静态障碍物的位置，并结合高精度的地图数据进行路径规划和避障。
3. 交通合规性。为保证车辆在复杂交互环境中的合理合规行驶，如遵守交通信号灯、保持车道行驶、礼让行人等。这要求规划算法需要具备一定对交通规则的理解和遵守能力，以确保与其他交通参与者之间完成合规交互。
4. 运动可行性。车辆的运动规划需要考虑车辆自身的动力学特性和运动学约束。这包括车辆的加速、减速、转向等能力，以及车辆的尺寸、重量和悬挂系统等对行驶路径的影响。为了确保运动可行性，在规划过程中，需要综合考虑车辆的动力学特性和环境约束，生成符合车辆能力的行驶轨迹。

鉴于上述多维度考量，动态交互环境下的运动规划问题构成了一项综合性研究挑战，它要求全面兼顾多个关键要素，以确保车辆能够安全、高效且合规地行驶。为解决此问题于交互环境中实现安全、合规及高效行驶的目标，本文采取了一种策略性分解方法，将该问题细化为全局路径规划问题和考虑车辆交互与避障的轨迹规划问题。全局路径规划问题聚焦于在宏观层面为车辆设计一条从起点至终点的最优或次优路径，轨迹规划问题侧重于在微观层面根据实时交通状况及与其他交通参与者的动态交互，规划出既能避免碰撞又能保持行驶流畅性的车辆轨迹。通过上述分解，可以将复杂问题转换为两个相对简单的子问题，降低问题的复杂度。本文将在后续章节对这两个问题进行分析并给出相应的求解方法。

3.2 全局路径规划与局部坐标系构建

3.2.1 基于 dijkstra 算法的全局路径规划算法

Dijkstra 算法，又称迪杰斯特拉算法，是由荷兰计算机科学家 Edsger W. Dijkstra 于 1956 年提出的一种图搜索算法，该算法旨在从图中一个顶点出发，找到该顶点到图中其他所有顶点的最短路径。它主要应用于有权图，即图中各边具有权重的场景。

Dijkstra 算法的核心思想是贪心策略，即每一步都选择当前距离起点最近的未确定最短路径的顶点，并更新其相邻顶点的最短路径长度。算法通过迭代的方式逐步逼近最终的最短路径解。该算法的详细计算过程共包括三步，分别是节点初始化，节点迭代计算更新和终止迭代过程。

Dijkstra 算法的主要应用范围是有权图，而当前地图尚未被处理成有权图的格式。为了实现过程，还需要对地图进行进一步处理，完成离散路段地图向有向权邻接矩阵的转换。幸运的是，OpenDrive 地图中，道路 (road) 和车道 (<lane>) 中都包含道路之间的邻接关系，利用这一特点，可以将离散的路段，拼接成有向的道路邻接矩阵。为了保证规划过程的最优性，需要在有向道路邻接矩阵中加入权重，对离散的路段而言，能够最直接车辆在路段上的行驶成本的，即道路的长度。因此对前后邻接的道路，行驶成本被赋值为当前道路的长度，对相邻车道而言，行驶成本可以被赋值为车道的宽度。具体伪代码如3-1所示。

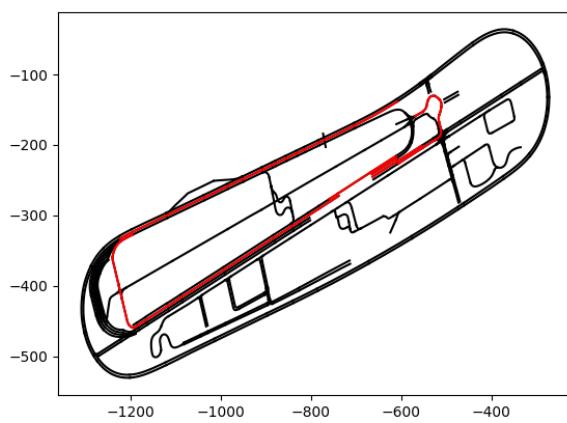


图 3-1 dijkstra 全局路径规划结果

算法 3-1: Dijkstra Algorithm

Input: Opendrive 地图信息, 起点 *Start*, 终点 *Goal*
Output: 可行全局路径 *Path*

- 1: 以 OpenDrive 中的离散路段为节点构建邻接矩阵, 链接节点间的代价为路段长度 *L*, 相邻道路之间的代价为道路宽度 *W*
- 2: 寻找起点和终点所在的道路节点 *Node_{start}* 和 *Node_{goal}*
- 3: 初始化距离数组 *D*, 所有值设为 ∞ , 起点 *Node_{start}* 的值设为 0
- 4: 初始化一个空集合 *S*, 用于存储已确定最短路径的顶点
- 5: 将起点加入优先队列 *Q*
- 6: **while** 优先队列 *Q* 不为空 **do**
- 7: 从 *Q* 中取出距离最小的顶点 *u*
- 8: **if** *u* 不是目标点 **then**
- 9: 对 *u* 的每个邻接点 *v*
- 10: **if** 通过 *u* 到达 *v* 的距离小于当前记录的距离 **then**
- 11: 更新 *v* 的距离
- 12: **end if**
- 13: **else**
- 14: **break**
- 15: **end if**
- 16: 将 *u* 加入集合 *S*
- 17: **end while**
- 18: 回溯获取可行路径 *Path*
- 19: **return** 全局可行路径 *Path*

3.2.2 Frenet 坐标系及坐标系转换

Frenet 是一种描述曲线在二维或三维空间中的运动状态表示方法, Frenet 坐标系的建立依赖于一个给定的参考线 (通常为车道中心线), 以参考线和其正交曲线构建形成一个正交坐标系。假定沿参考线方向为纵轴 (*s*), 垂直于参考线方向为横轴 (*d*), 由于参考线相互垂直, 能够忽略道路曲率给车辆运动状态估计所带来的影响, 相比于笛卡尔坐标系, 能够更加简洁、直观地表示自动驾驶车辆沿车道线的行驶距离和偏离车道中心线的程度。图3-2 描述了笛卡尔坐标系下的道路和 Frenet 坐标系下的道路。车辆由笛卡尔坐标系转换至 Frenet 坐标系下之后, 曲线参考线被拉直, 如此操作可以降低问题的复杂度, 更好地描述车道之间的关系。除此之外, 得益于 Frenet 坐标系沿车道线构建的特性, 可以将车辆的运动信息解耦, 对横纵向运动进行解耦规划。但是经过规划的横纵向运动信息, 仍然需要将其重新映射到笛卡尔坐标系, 供底层控制模块跟踪。

在全局坐标系下, 车辆的运动状态可以表示为 $[x, y, yaw, \kappa, v, a]$, 其中, *x*, *y* 为车辆在全局坐标系下的位置; 为 *yaw* 车辆在全局坐标系下的朝向角; κ 为车辆当前轨迹的曲率; *v* 为车辆的速度, *a* 为车辆的加速度.

在 Frenet 坐标系下, 车辆的运动状态可以利用车辆与参考线的相对关系进

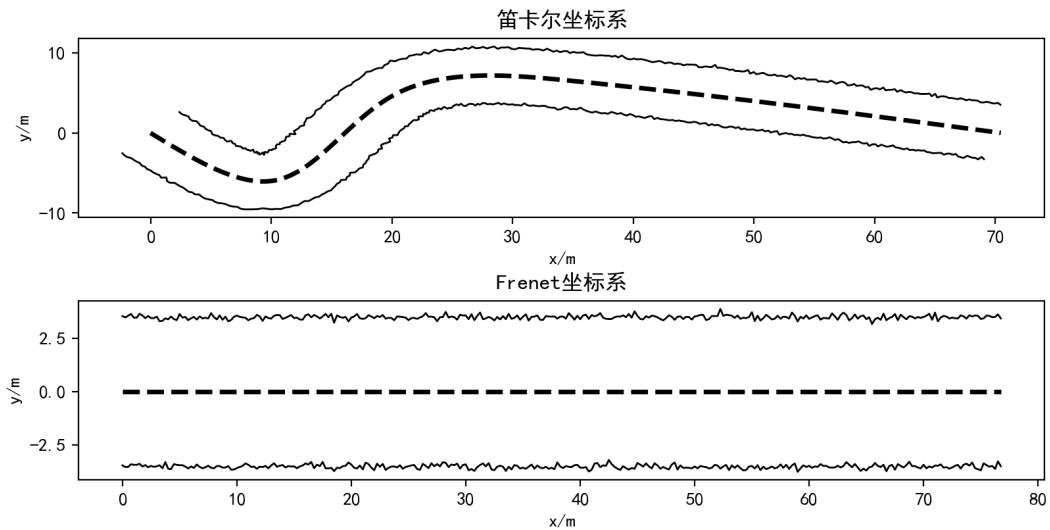


图 3-2 Frenet 坐标系

行表示, 具体可以写作 $[s, \dot{s}, \ddot{s}, d, \dot{d}, \ddot{d}, d', d'']$, 其中, s, d 为车辆在 Frenet 坐标系下的位置, 分别表示车辆沿参考线方向上的位移和垂直参考线方向上的位移, \dot{s}, \dot{d} 分别为 s, d 的一阶导数, 表示车辆沿参考线方向的速度和垂直参考线方向的速度。 \ddot{s}, \ddot{d} 分别为 s, d 的二阶导数, 表示车辆沿参考线方向的加速度和垂直参考线方向的加速度, d', d'' 分别为 Frenet 坐标系下横向坐标对纵向坐标的一阶和二阶导数。

$$\left\{ \begin{array}{l} s = s_r \\ \dot{s} = \frac{v_x \cos \theta}{(1 - r_r d)} \\ \ddot{s} = \frac{a_x \cos \theta - \hat{s}^2 [d' (k_x \frac{1-k_r d}{\cos \theta} - k_r) - (k'_r d + k_r d')]}{1 - k_r d} \\ d = \text{sign}((y - y_r) \cos(\theta_r) - (x - x_r) \sin(\theta_r)) \sqrt{(x - x_r)^2 + (y - y_r)^2} \\ \dot{d} = v_x \sin \theta \\ \ddot{d} = a_x \sin \theta \\ d' = (1 - k_r d) \tan \theta \\ d'' = - (k'_r d + k_r d') \tan \theta + \frac{1 - k_r d}{\cos^2 \theta} \left(k_x \frac{1 - k_r d}{\cos \theta} - k_r \right) \end{array} \right. \quad (3-1)$$

$$\begin{cases} x = x_r - d \sin(\theta_r) \\ y = y_r + d \cos(\theta_r) \\ \theta_x = \arctan\left(\frac{d'}{1 - k_r \cdot d}\right) + \theta_r \\ v = \sqrt{[s(1 - k_r d)]^2 + (sd')^2} \\ a = \ddot{s} \frac{1 - k_r d}{\cos \Delta\theta} + \frac{s^2}{\cos \Delta\theta} \left[d' \left(k_x \frac{1 - k_r d}{\cos \Delta\theta} - k_r \right) - (k'_r d + k_r d') \right] \\ k = \left((d'' + (k'_r d + k_r d') \tan \Delta\theta) \frac{\cos^2 \Delta\theta}{1 - k_r d} + k_r \right) \frac{\cos \Delta\theta}{1 - k_r d} \end{cases} \quad (3-2)$$

通过公式3-1和3-2, 可以实现笛卡尔坐标系下的运动状态和 Frenet 坐标系下运动状态的相互转换, 式中 $\Delta\theta$ 为车辆当前航向角 θ_c 与参考线上最近的投影点 θ_r 之间的差值, κ_r 为参考线上投影点处的曲率,

3.3 SL/ST 图构建与可行凸空间开辟

SL(Stretch-Lateral) 图和 ST(Stretch-Time) 图分别为里程-偏移图, 里程-时间图。在自动驾驶领域, SL 图常用于描述物体沿特定路径的运动状态, 包括位置、速度和加速度等。这种表示方法有助于将复杂的路径规划问题简化为在二维平面上的几何问题, 从而便于进行算法设计和优化。ST 图常用于将速度规划、障碍物预测轨迹等信息投影到二维平面上, 以便进行决策和优化。在 ST 图中, 斜率表示速度, 斜率的导数 (即二阶导数) 表示加速度, 通过构建 ST 图, 可以直观地展示物体的运动状态、预测轨迹以及潜在的危险区域等信息, 从而为自动驾驶系统提供决策依据。本节中将详细介绍 SL 与 ST 图的构建与可行凸空间的开辟方法。

3.3.1 SL 图构建与可行边界提取

本节以车辆所在车道 (lane) 的道路中心线为参考线, 即 Frenet 坐标系的 S 轴, 以垂直参考线方向为 L 轴, 构建 Frenet 坐标系, 考虑将前方规划范围内的静态障碍物的关键点转换至 SL 图中, 即可完成 SL 图的构建。在完成 SL 图的构建之后, 为了更好地服务于轨迹规划, 需要考虑规划范围内的障碍物进行可行边界提取。

为了简化边界提取问题的同时降低优化问题中的优化变量，需要将 SL 图进行离散化，本文选择优化间隔 ds 为 1m，规划范围 80m，如图3-3所示。

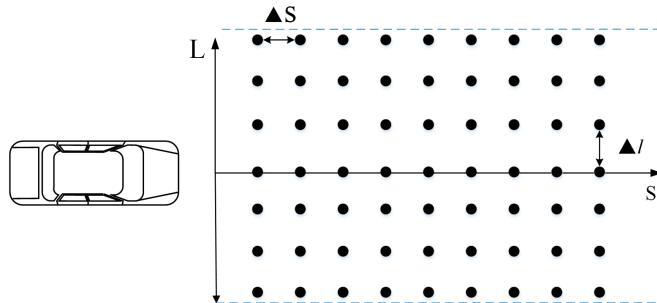


图 3-3 SL 图

对一段完整的可行凸空间，作出如下定义：

$$\text{ConvexSpace} = \{(s_0, l_{min}^0, l_{max}^0), (s_1, l_{min}^1, l_{max}^1) \dots (s_n, l_{min}^n, l_{max}^n)\} \quad (3-3)$$

由于障碍物形状各异，并且在坐标系转换过程中其集合特征将变化，对障碍物的精细化处理将使得算法的复杂度进一步上升。因此，本文考虑障碍物在 Frenet 坐标系两个不同坐标轴的占用情况，对障碍物的形状进行简化，统一处理成如下形式，其中， s_i 表示障碍物中心点在 Frenet 坐标系下的位置， S_i 为障碍物在 Frenet 坐标系下的 S 轴占用长度， L_i 为障碍物在 Frenet 坐标系下 L 轴的占用长度。

$$\text{Obs} = \{(s_0, S_0, L_0), (s_1, S_1, L_1), \dots, (s_n, S_n, L_n)\} \quad (3-4)$$

凸空间开辟算法的伪代码如算法3-2所示。为提取完整的可行边界，首先需要初始化规划的边界和参考行驶路径（本文中为 $0 * n$ ），然后对每一个障碍物进行遍历，计算障碍物在 S 轴上的占用范围，接着寻找障碍物左右两侧空间更大的绕行方向，并根据绕行方向，重新调整凸空间的上下界（若向右绕行，则缩小 l_{max} ，否则增大 l_{min} ）。完成对障碍物的遍历之后，可行凸空间的雏形以及完成构建，为进一步保证该凸空间的可行性，需要进一步对凸空间进行可行性检查，即 $l_{max}^i > l_{min}^i$ ，若满足该条件，则输出开辟成功的凸空间，若不满足，则重置冲突范围 (s_i, s_{i+n}) ，并分别尝试以最左侧和最右侧为绕行方向，开辟可行凸空间，若开辟成功，则输出开辟凸空间，否则空间开辟失败。

算法 3-2: 横向凸空间开辟方法

Input: 参考路径 Path, Frenet 坐标系下障碍物纵向位置集合 s_{obs} , 障碍物位置集合 l_{obs} ,
长度信息集合 $Length_{obs}$, 自车横向位置 l_{ego}

Output: 凸空间上界 l_{upper} , 凸空间下界 l_{lower} , 求解成功状态 $conflict$

```

1: 初始化  $l_{upper}$ ,  $l_{lower}$  和  $conflict$ 
2: for  $s$  in  $s_{obs}$  do
3:   计算障碍物的占用区间  $s_{start}, s_{end}$ 
4:   for  $s$  in  $(s_{start}, s_{end})$  do
5:     if  $l_{ego} < l_{obs}$  then
6:       更新凸空间上界  $l_{upper}$ 
7:     else
8:       更新凸空间下界  $l_{lower}$ 
9:     end if
10:   end for
11: end for
12: if  $\forall(l_{upper} < l_{lower})$  then
13:   找到冲突位置  $s_{start}^{conflict}, s_{end}^{conflict}$ 
14:   计算左右侧可行空间  $Space_{left}, Space_{right}$ 
15:   if  $Space_{left} > Space_{right}$  then
16:     更新  $l_{lower}$ , 修改向左侧绕行
17:   else
18:     更新  $l_{upper}$ , 修改向右侧绕行
19:   end if
20: end if
21: return  $l_{lower}, l_{upper}$ 

```

3.3.2 ST 图与可行空间构建

在本节中, 以规划出的路径为 S 轴, 以时间为 T 轴, 并进一步地将环境中的动态障碍物中投射至 ST 图的规划空间中, 即可完成 ST 图的构建。但是规划过程中, 如果所有障碍物都投射至 ST 图中, 势必对规划的实时性带来一定的影响, 为车辆的实时安全规划带来潜在的威胁。

为提升规划效率, 减少不必要的算力损失, 本文在将障碍物投射至 ST 图之前, 加入一定的筛选规则, 获得对正常行车带来影响的车辆。详细的过滤规则如下:

1. 当环境车距离规划路径的距离大于安全阈值 d_{safe} 时, 不做投影。
2. 当环境车的纵向位置 s 大于最大规划距离 s_{max} 时, 不做投影。
3. 当环境车在规划的横向缓冲区间 d_{buffer} 内, 且横向速度 v_d 小于横向安全速度阈值 v_d^{safe} , 不做投影。

障碍物向 ST 图投射, 需要计算背景车进入和离开路径交互区间的纵向坐标 s_{in} 和 s_{out} , 以及进入时间和离开时间 t_{in} 和 t_{out} , 详细的计算方法如公式3-5所

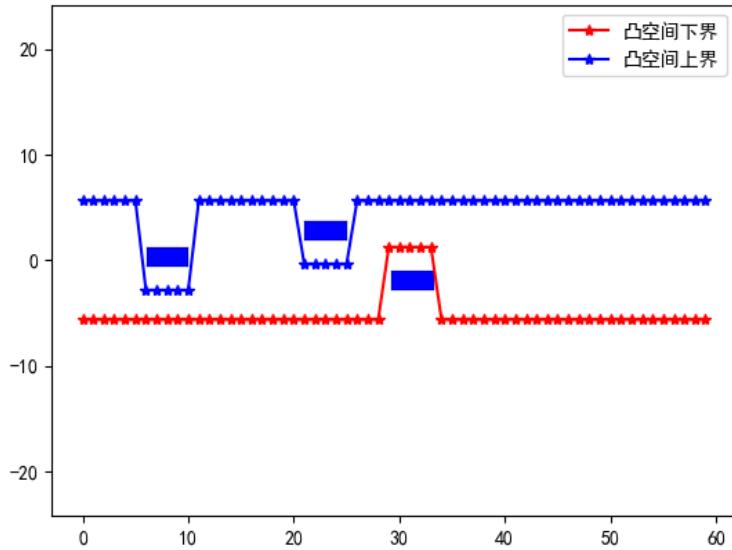


图 3-4 边界提取结果

示:

$$\begin{cases} t_{in} = \frac{|d| - d_{buffer}}{\dot{d}} \\ t_{out} = \frac{|d| + d_{buffer}}{\dot{d}} \\ s_{in} = s + t_{in}\dot{s} \\ s_{out} = s + t_{out}\dot{s} \end{cases} \quad (3-5)$$

式中, (s, d, \dot{s}, \dot{d}) 为以规划路径为参考线所构建的 Frenet 坐标系下的运动状态, 分别代表纵向位置, 横向位置, 沿参考向方向的速度, 垂直参考线方向的速度。 t_{in} 为进入交互区间的时间, t_{out} 离开交互区的时间, s_{in} 为进入交互区间的纵向位置, s_{out} 为离开交互区间的纵向位置。对环境中所有对本车正常行驶可能存在干扰的车辆重复 ST 图投射这一过程, 即可完成 ST 图的构建。

图3-5为投射完成之后所构建的 ST 图, s_0, t_0 分别为规划的位置起点, 起始时间, s_{end}, t_{end} 分别为最大规划长度和最大规划周期, 两条蓝色的带状区域表明环境中存在两辆对本车正常行驶存在干扰的车辆, 其位置表示在规划时域内对路径的占用情况, 假定障碍车名称分别为 $Obs1$ 和 $Obs2$, 带状区域的斜率障碍车代表沿路径方向的速度。 $Obs1$ 从规划起始时刻 t_0 便出现在规划空间内, 在 t_{out} 时刻超出纵向规划区域 s_{end} , $Obs2$ 在 ST 图中的占用区间为 $t_{in1} - t_{out1}$ 。对 $Obs2$, 在 $t_{in1} - t_{out1}$ 区间内存在交互可能。

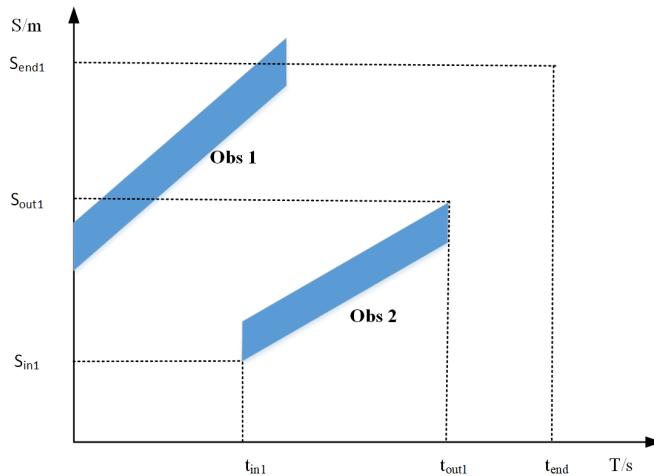


图 3-5 ST 图

ST 图的构建是为了服务于速度规划，在复杂交互场景下，ST 图中的障碍物数量增加，完整解空间中的可行解不再唯一。对于这样一个存在多个局部最优解的问题，如何构造一个凸的求解空间，使得原始问题的解能够收敛到一个稳定且合适的数值显得至关重要。

3.4 基于分段优化的轨迹规划方法

得益于 Frenet 坐标系对运动规划问题的简化，使得车辆的运动规划问题可以拆解成横向的路径规划问题和纵向的速度规划问题，降低了问题的复杂度。本文沿用这一思路，对车辆的运动规划问题进行解耦求解。

3.4.1 分段优化问题的构建

在3.3节中，已经完成了在 ST 图和 SL 图中的构建和凸空间开辟，智能车车辆在三维空间下的非线性运动规划问题被拆解成了两个二维空间下的凸优化问题。即横纵向运动规划问题均转换为可行空间下的平滑分段优化问题。分段优化的核心思想在于将一段连续的曲线，拆解为多个连接相邻离散状态的曲线，将一个复杂问题拆解为了多个简单的子问题，进而降低问题的复杂度，提升问题的求解效率。任意一段有限时域内的横/纵向轨迹，都可以用一段曲线来描述，因此车辆的运动规划问题可以用分段优化的思想进行求解。

对连续曲线中某一节点，其状态可以定义为 x, \dot{x}, \ddot{x} ，分别为曲线上点 x 处的值，曲线上点 x 处的一阶导，曲线上点 x 处的二阶导。对应的，在运动规划

问题中，其物理意义分别为位置，速度，加速度。

为了实现完整曲线的分段优化，首先需要对一段连续曲线离散为 n 段相邻的分段曲线，然后根据相邻曲线之间的连续性约束构建对应的优化问题。在这样一个优化问题中，优化变量为 $\{x_1 \dot{x}_1, \ddot{x}_1\}, \{x_2, \dot{x}_2, \ddot{x}_2\}, \dots \{x_n, \dot{x}_n, \ddot{x}_n\}$ ，优化变量个数为 $3n$ 。对一个运动规划问题，其优化目标通常需要从舒适性，参考偏离，连续性，安全性等多个方面进行考虑。因此，对一个运动规划中的分段优化问题，损失函数和约束条件应当覆盖上述几个方面：

1. 舒适性。车辆在行驶过程中需要尽可能保证舒适性，对行车舒适性影响最大的即加速度和加加速度，在运动曲线中对应的体现为 \dot{x}, \ddot{x} 。为减少行驶过程中的不必要加减速，并尽可能保证运动过程平滑。有关舒适性的损失函数可以以如下行驶描述：

$$cost_{comfort} = w_{acc}\ddot{x} + w_{jerk}\ddot{\dot{x}} \quad (3-6)$$

2. 参考状态偏离。参考状态偏离用于描述车辆与参考状态的偏离程度，规划轨迹与参考轨迹需要尽可能贴近，以保证轨迹靠近参考状态。有关参考偏离的损失函数可以写作如下形式：

$$cost_{ref} = w_{x_{ref}}(x - x_{ref}) + w_{\dot{x}_{ref}}(\dot{x} - \dot{x}_{ref}) \quad (3-7)$$

3. 安全性。由于分段曲线规划过程中难以直接考虑安全性，但是幸运的是，本文在3.3中进行了 SL 坐标系下的可行空间提取，通过该操作可以将非凸非线性的安全约束条件修改为路径点位于可行凸空间内的约束，可以写作如下形式：

$$x_{min} < x < x_{max} \quad (3-8)$$

4. 连续性。与安全性相似，曲线的连续性也难以直接在目标函数中体现，且对分段优化问题来说，曲线的连续性是必须要满足的条件。对一段曲线的连续性约束，可以通过泰勒展开式的形式给出：

$$\begin{aligned} x_{i+1} &= x_i + \dot{x}_i dt + \frac{1}{2} \ddot{x}_i dt^2 + \frac{1}{6} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt^3 \\ \dot{x}_{i+1} &= \dot{x}_i + \ddot{x}_i dt + \frac{1}{2} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt^2 \\ \ddot{x}_{i+1} &= \ddot{x}_i + \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt \end{aligned} \quad (3-9)$$

除此之外，为了保证规划轨迹的可行性，规划起始时刻的状态需要与车辆

与状态一致，具体写作公式3-10：

$$\begin{cases} x_0 = x_{start} \\ \dot{x}_0 = \dot{x}_{start} \\ \ddot{x}_0 = \ddot{x}_{start} \end{cases} \quad (3-10)$$

由此，上述完整的分段优化问题可以被描述如下二次规划问题：

$$\begin{aligned} \min J &= w_{velo} \dot{x}_i^2 + w_{acc} \ddot{x}_i^2 + w_{jerk} \left(\frac{\ddot{x}_{i+1} - \ddot{x}_i}{ds} \right)^2 + w_{x_{ref}} (x_i - x_{ref}^i)^2 + w_{\dot{x}_{ref}} (\dot{x}_i - \dot{x}_{ref}^i)^2 \\ s.t. \quad &\begin{cases} x_{i+1} = x_i + \dot{x}_i dt + \frac{1}{2} \ddot{x}_i dt^2 + \frac{1}{6} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt^3 \\ \dot{x}_{i+1} = \dot{x}_i + \ddot{x}_i dt + \frac{1}{2} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt^2 \\ \ddot{x}_{i+1} = \ddot{x}_i + \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt \\ x_0 = x_{start} \\ \dot{x}_0 = \dot{x}_{start} \\ \ddot{x}_0 = \ddot{x}_{start} \end{cases} \end{aligned} \quad (3-11)$$

3.4.2 基于分段优化的路径规划方法

在3.4.1节，已经完成了分段优化问题的构建，但是这样一个问题构型，显然难以直接用于求解路径规划问题。本节将分段优化问题进行修改调整，使其能够完成路径规划问题的求解。对一个路径规划问题，前文中的优化变量可以具体表示为 (d, \dot{d}, \ddot{d}) ，即横向位移，横向速度和横向加速度。

使用3.4.1节中的问题构型，可以将路径规划问题写成分段优化问题的一般形式。但是在路径规划过程中，为了尽可能保证规划路径的平稳性，预期横向速度，加速度，加加速度均为 0。在3.4.1节的构型的基础上，对问题的损失函数简化，如公式3-12所示：

$$J = w_{velo} \dot{d}_i^2 + w_{acc} \ddot{d}_i^2 + w_{jerk} \left(\frac{\ddot{d}_{i+1} - \ddot{d}_i}{ds} \right)^2 + w_{x_{ref}} (x_i - x_{ref}^i)^2 \quad (3-12)$$

通过求解这个分段优化问题，一定能够获得一条在安全凸空间内的连续路径。这样一条路径考虑了路径的连续性，平稳性和安全性，但是由于规划坐标系是 Frenet 坐标系而不是笛卡尔坐标系，所以难以保证路径在转换至笛卡尔坐标系之后的运动可行性。并且，由于 Frenet 坐标系与笛卡尔坐标系的转换公式

复杂，直接在分段优化问题中加入运动学可行约束是难以直接实现的。

最能表现运动学约束的曲线性能指标是曲率 (curvature)，曲率的计算公式可以通过公式3-13完成计算：

$$\kappa = \frac{\left(\frac{((d'' + (\kappa_r d + \kappa_r d') \tan \Delta\theta) \cos^2 \Delta\theta)}{1 - \kappa_r d} + \kappa_r \right) \cos \Delta\theta}{1 - \kappa_r d} \quad (3-13)$$

式中, k_r 为参考曲率, d, d', d'' 分别为横向偏移和对应的一阶和二阶导数, $\Delta\theta$ 为车辆的航向角与参考线的航向角的差值。对这样一个约束条件，显然是非线性非凸的，直接在问题中加入该约束条件势必显著增加问题的复杂度，影响规划的实时性。为了简化松弛这一约束，本文做出以下几点假设：

1. 车辆在行驶过程中几乎平行于参考线，因此车辆的航向角与参考线对应点位的航向角几乎一致，因此有 $\Delta\theta = 0$ 。
2. 考虑规划出的轨迹中，横向加速度 \ddot{d} 被作为损失项加入目标函数中，规划出来的结果应当在数值上尽可能小，因此也考虑该项为 0。

基于上述两点假设，公式3-13可以被简化为公式3-14：

$$\kappa \simeq \frac{k_r}{1 - k_r \times 1} \quad (3-14)$$

考虑二自由度运动学模型，在给定最大车轮转角 α_{max} 的情况下，所能接受的路径最大曲率可以通过公式3-15完成计算：

$$\kappa_{max} = \frac{\tan(\alpha_{max})}{L} \quad (3-15)$$

通过公式3-14和公式3-15，可以将非线性的曲率约束（公式3-13）简化为线性的约束，如公式3-16：

$$\tan(\alpha_{max})\kappa_r l - \tan(\alpha_{max}) + |\kappa_r|L \leq 0 \quad (3-16)$$

完整的规划问题可以写作如公式3-17的形式，通过对这样一个问题求解，即可获得结构化道路下的可行平滑轨迹：

$$\begin{aligned}
 \min J &= w_{velo} \ddot{d}_i^2 + w_{acc} \ddot{\dot{d}}_i^2 + w_{jerk} \left(\frac{\ddot{d}_{i+1} - \ddot{d}_i}{ds} \right)^2 + w_{x_{ref}} (x_i - x_{ref}^i)^2 \\
 \text{s.t. } &\left\{ \begin{array}{l} x_{i+1} = x_i + \dot{x}_i dt + \frac{1}{2} \ddot{x}_i dt^2 + \frac{1}{6} \left(\frac{\ddot{x}_{i+1} - \ddot{x}_i}{dt} \right) dt^3 \\ \dot{x}_{i+1} = \dot{x}_i + \ddot{x}_i dt + \frac{1}{2} \left(\frac{\ddot{x}_{i+1} - \ddot{x}_i}{dt} \right) dt^2 \\ \ddot{x}_{i+1} = \ddot{x}_i + \left(\frac{\ddot{x}_{i+1} - \ddot{x}_i}{dt} \right) dt \\ x_0 = x_{start} \\ \dot{x}_0 = \dot{x}_{start} \\ \ddot{x}_0 = \ddot{x}_{start} \\ \tan(\alpha_{max}) \kappa_r l - \tan(\alpha_{max}) + |\kappa_r| L \leq 0 \end{array} \right. \quad (3-17)
 \end{aligned}$$

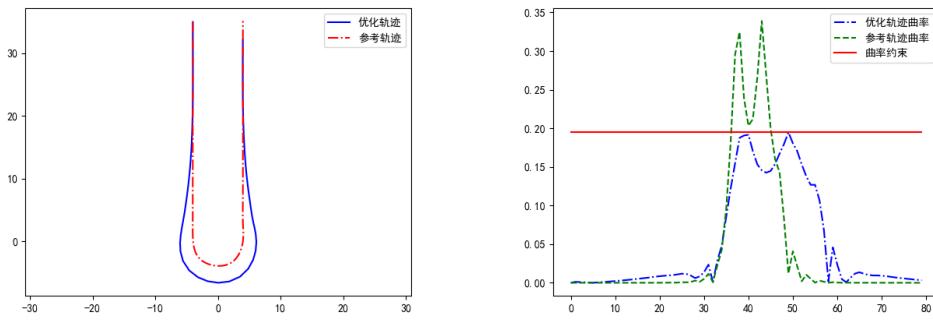


图 3-6 基于分段平滑的路径优化结果

3.4.3 基于分段优化的速度规划方法

速度规划的目标是在确保安全、合规和舒适的前提下，通过优化车辆的速度曲线来提高行驶效率、减少能源消耗、缩短行程时间。在本节中，将对基于分段优化的速度规划进行问题构造，参考分段优化问题的一般构造形式，但是考虑速度规划问题中，不考虑速度 \dot{s} 的损失和参考状态 s 的损失，因此，速度规划的目标函数中， w_{velo} 和 $w_{x_{ref}}$ 应当为 0，对应的目标函数可以简化为公式3-18形式：

$$J = w_{acc} \ddot{s}_i^2 + w_{jerk} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right)^2 + w_{\dot{s}_{ref}} (\dot{s}_i - \dot{s}_{ref}^i)^2 \quad (3-18)$$

但是在这种问题构型下，仅考虑了车辆在运动过程中加速与减速的平滑性，

并没有考虑车辆的动力学性能，求解出的纵向轨迹容易超过车辆的运动性能极限，并且考虑交通规则，车辆在公开道路行驶时，不允许进行倒车行为。因此需要在3.4.1节的问题构型下加入对车辆纵向性能的约束，可以写作公式3-19：

$$\begin{aligned} 0 &\leq \dot{s}_i \leq v_{max} \\ a_{min} &\leq \ddot{s}_i \leq a_{max} \\ jerk_{min} &\leq \frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \leq jerk_{max} \end{aligned} \quad (3-19)$$

3-19式中的约束，实现了对车辆的纵向性能的考虑，但是由于本文算法利用Frenet坐标系实现了横纵向解耦规划，在纵向规划的过程中，并没有考虑车辆的横向运动，使得横纵向轨迹融合后出现横摆角速度过大，显著影响车辆的行驶稳定性和侧向安全性。为了在纵向规划过程中，考虑车辆的行驶稳定性，对车辆的速度约束进行改写，加入对路径曲率 κ 的考虑，可以改写为3-20式：

$$0 \leq \dot{s}_i \leq \min\left(\sqrt{\frac{a_{lateral}}{\kappa}}, v_{max}\right) \quad (3-20)$$

至此，可以完成速度规划问题的构建，完整的速度规划问题构建，完成问题构型如公式3-21所示：

$$\begin{aligned} \min J = & w_{acc} \ddot{s}_i^2 + w_{jerk} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right)^2 + w_{\dot{s}_{ref}} (\dot{s}_i - \dot{s}_{ref})^2 \\ s.t. \quad & \left\{ \begin{array}{l} x_{i+1} = x_i + \dot{x}_i dt + \frac{1}{2} \ddot{x}_i dt^2 + \frac{1}{6} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt^3 \\ \dot{x}_{i+1} = \dot{x}_i + \ddot{x}_i dt + \frac{1}{2} \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt^2 \\ \ddot{x}_{i+1} = \ddot{x}_i + \left(\frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \right) dt \\ x_0 = x_{start} \\ \dot{x}_0 = \dot{x}_{start} \\ \ddot{x}_0 = \ddot{x}_{start} \\ 0 \leq \dot{s}_i \leq \min\left(\sqrt{\frac{a_{lateral}}{\kappa}}, v_{max}\right) \\ a_{min} \leq \ddot{s}_i \leq a_{max} \\ jerk_{min} \leq \frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} \leq jerk_{max} \end{array} \right. \end{aligned} \quad (3-21)$$

3.5 仿真验证

为了进一步验证本文算法的有效性，本文基于Python3.8平台搭建了车辆轨迹规划仿真环境，仿真环境中包含了一条长1163m的三车道公路和多个动态

障碍物和静态障碍物，具体图3-7所示。我们在仿真环境中设置多个测试场景，如静态障碍物避让，换道超车，跟驰行驶等。在仿真环境中，被控车辆统一为蓝色车辆，静态障碍车为红色，动态障碍物为绿色车辆，颜色越浅说明该状态与当前状态越远。在该环境中，由于被测车辆和背景车辆的状态都可以直接读取，因此，不考虑规划过程中的误差存在。车辆在不同测试场景中的具体表现如下所示。

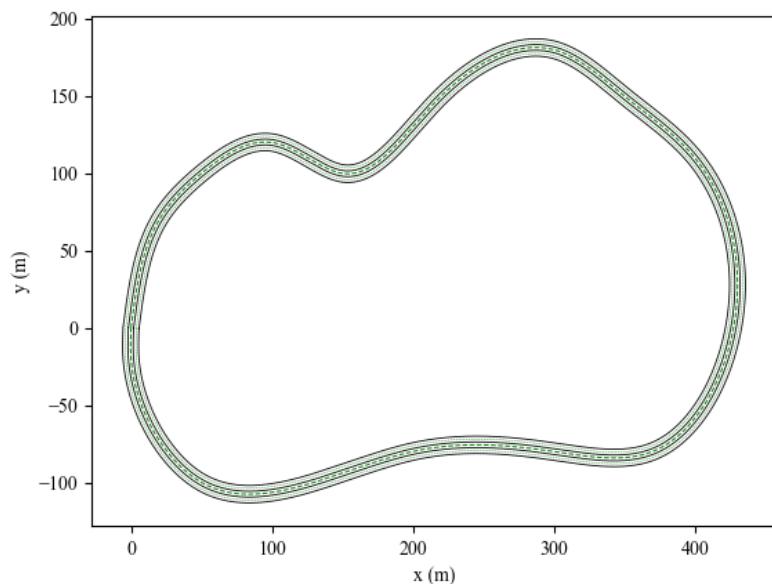


图 3-7 仿真测试环境

静态障碍物避让是运动规划器中最重要的一项功能，要求算法实时为车辆规划出一条平滑，平稳，运动学可行且无碰撞的安全轨迹。为了检测算法在该方面的功能，我们设置了静态障碍物避让场景以测试算法在该方面的性能。在该测试场景中，被测车辆初始位置为道路中心车道，初始速度为 0，道路期望速度为 40km/h，环境中包含三个静态障碍车辆，分别位于两条道路和道路交界处。

车辆在该测试环境下的效果如图3-8所示。面对静态障碍物，本文算法规划出了一条从足够光滑的路径，实现从三个障碍物的中间穿插通过，整体轨迹平滑，无碰撞发生。图3.9(d)展示了规划路径的曲率，本文算法在该场景下的避障路径最大曲率仅为 0.045，有效保证了运动过程中的可行性和稳定性。图3-9展示了对应的速度规划结果，本文生成速度曲线足够平滑，最大加速度为 $2.14m/s^2$ ，能够保证车辆在纵向运动中的平顺性和舒适性。

自主换道是自动驾驶车辆必不可少的一项功能，它赋予车辆在结构化道路环境和交通合规性的前提下更加自由合规行驶的能力，使其能够完成换道超车，

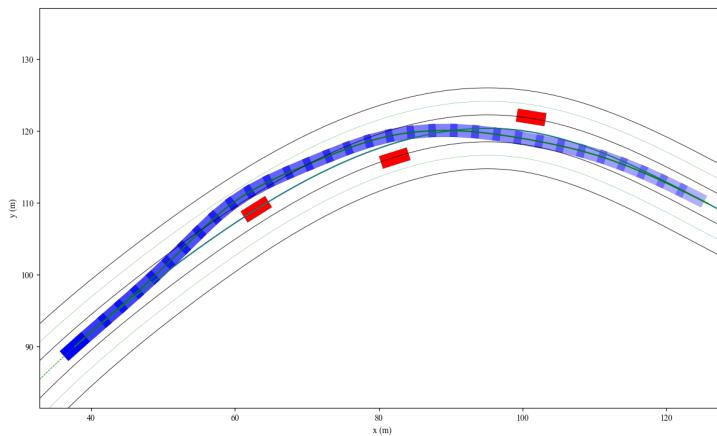


图 3-8 静态障碍物避让

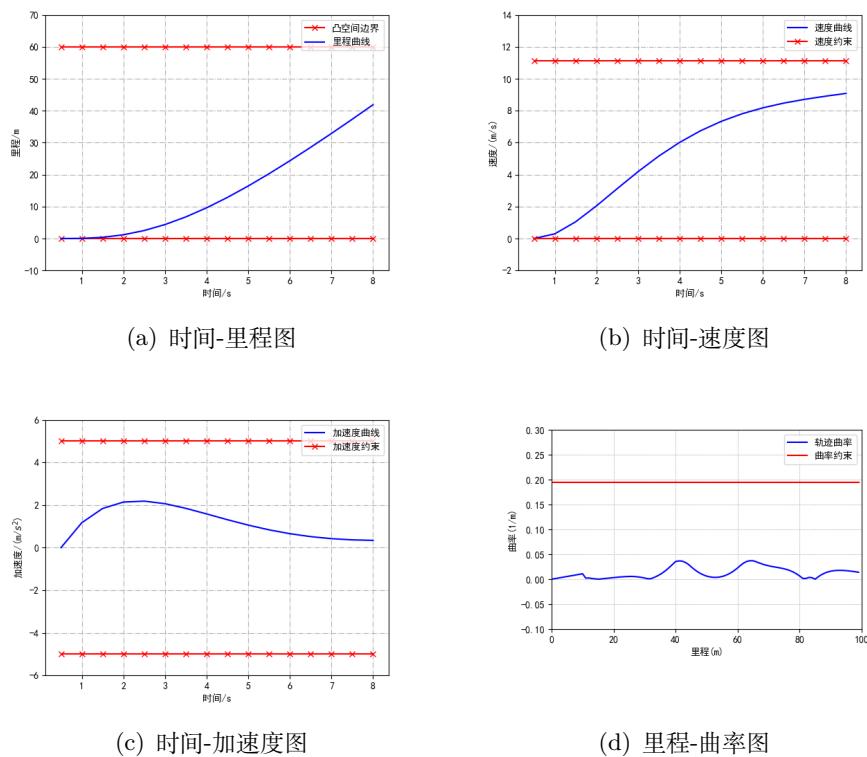


图 3-9 静态障碍物避让

路径切换等驾驶行为，获得更多提升交通效率的方法和能力。本文选择在自车前方设置一个低速环行车辆来触发车辆的换道行为。该环境的具体设置如下，自车行车速度为 15km/h，在自车正前方 20m 处存在一辆低速缓行车辆，速度为

3km/h，两车均位于最右侧车道。于此同时，车辆的侧后方与侧前方均存在正常行驶车辆，行车速度为 20km/h。

车辆的自主换道运动过程如图3-10所示，由于换道安全空间足够，车辆在起始时刻便开始进行换道，由于前方车辆缓行，为保证行车过程中的安全，车辆在换道过程中并未采取急加速行为，而是采取缓慢加速的方式进行换道，换道过程中与前车保持了适当的安全距离，完成换道后再进一步加速。车辆的纵向运动曲线如图3-11所示，车辆的运动曲线平滑，加速度变化平稳，并未出现大幅波动，运动过程中最大加速度为 $0.5m/s^2$ ，运动过程中最大速度为 20km/h，到达目标运动速度。换道轨迹的曲率如图3.11(d)所示，整体路线的最大曲率出现在 14m 处，为换道过程的中间路段，最大值为 0.055，整体运动过程平滑，在车辆的运动学约束范围内，确保了换道轨迹的可行性

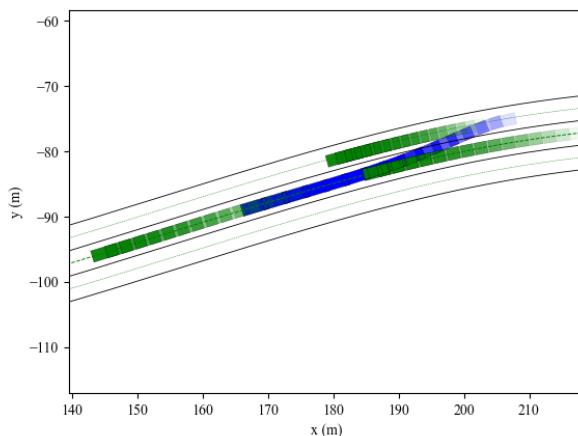


图 3-10 换道汇入

自主刹停是保障自动驾驶汽车行驶安全的重要功能，赋予自动驾驶车辆到达终点后平稳刹停，或前方车辆紧急刹车时快速反应的能力，能够极大程度保证车辆的安全行驶和行程结束时刻的平稳性。为了测试算法在该方面的性能，本文在高速行驶的车辆前方，设置一个忽然出现的缓慢行驶车辆以模拟前方车辆紧急刹车的情况。测试环境的具体设置如下，自车位置位于中间车道，行车速度为 30km/h，前方车辆缓行，速度为 0km/h，左右侧车道均有侧向来车，无可行换道空间，具体运动过程如图3-12所示。

被控车辆在发现前方低速车辆制动之后，立刻进行了刹车制动行为，并快速刹停，并未与前方车辆发生碰撞。运动过程中的参数如图3-13所示，完整制动过程中，为保证刹车过程中的平稳性和舒适性，以及制动过程中的动力学特性。被控车辆在制动起始时刻，速度为 $30km/h$ ，加速度为 $-2.3m/s^2$ ，随后缓慢

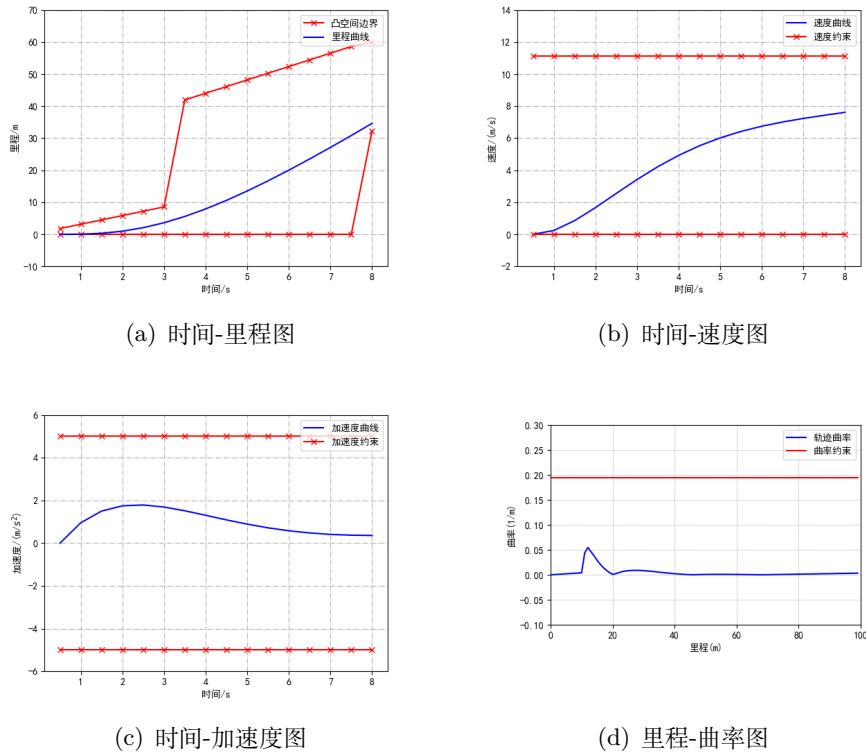


图 3-11 换道汇入仿真结果

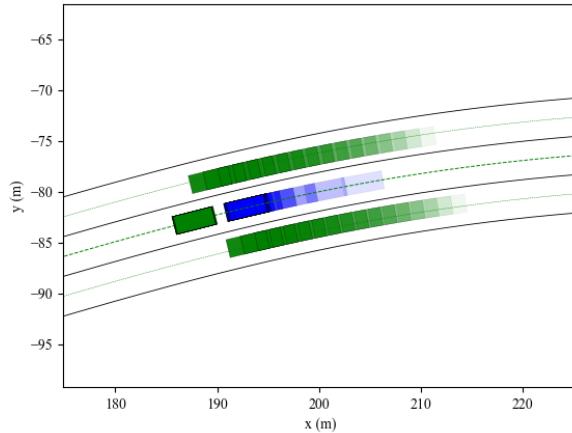


图 3-12 自主刹停

增长，在 2.0s 时刻增长至最大值 $-4m/s^2$ ，随后逐渐缓慢降低至 $0m/s^2$ ，车辆速度在 4.5s 时完成刹停，该过程变化平稳，无剧烈波动，有效保证了刹停过程中的稳定性和舒适性。

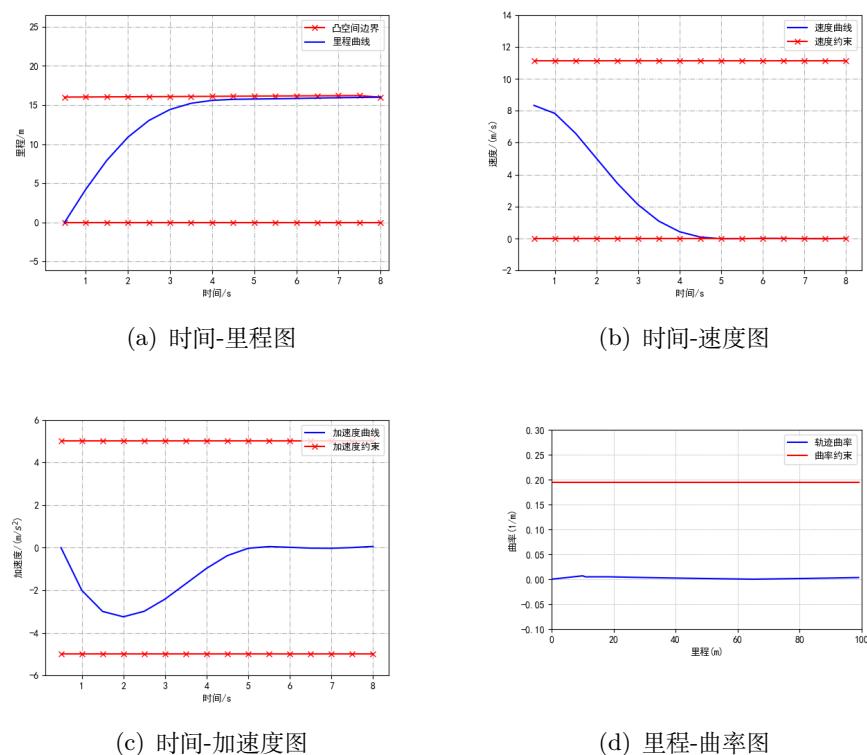


图 3-13 自主刹停运动参数变化曲线

第4章 面向行泊一体的轨迹跟踪控制器

为了确保车辆能够安全、准确地执行预设的轨迹，横纵向轨迹跟踪控制器的建立至关重要。面向行泊一体的横纵向轨迹跟踪控制器需要同步满足行车与泊车场景的协同控制需求，是实现车辆精确控制的核心，负责根据预设的轨迹和车辆的实际状态，实时调整车辆的横向和纵向运动，实现对轨迹的精确跟踪。为了确保规划出的轨迹能被底层的控制器正确执行并满足行泊一体场景的需求，本节设计了一种利用不同模型，满足不同工况的轨迹跟踪算法。

4.1 车辆运动模型建立

运动学模型建立是自动驾驶技术中的关键步骤，它涉及到从几何学的角度研究车辆的运动规律，包括车辆在空间的位置、速度、加速度等随时间的变化。对一个一般的运动学模型，可以都可以写成状态方程的模型，如公式4-1所示：

$$x_{t+1} = Ax_t + Bu_t \quad (4-1)$$

4.1.1 单车运动学模型

单车运动学模型一般指二自由度自行车模型，该模型是经典简化车辆模型，因其简洁性和足够的准确性而被广泛用于车辆动态分析和控制算法的开发工作中。

该模型的三项基本假设如下所示：

1. 简化假设：二自由度自行车模型假设车辆是一个刚体，忽略轮胎侧偏、弹性变形等复杂因素。它只考虑车辆在垂直于车辆前进方向的两个自由度：横向（侧向）和纵向（前进方向）。
 2. 两个自由度：模型中的“二自由度”指的是车辆的横向加速度和纵向加速度。车辆的垂直运动和绕垂直轴的旋转被忽略。
 3. 几何关系：模型中车辆的运动由前轮的转向角和车辆的纵向速度来描述。车辆的横向运动受到转向角的影响，而纵向运动则受到车辆速度的控制。
- 车辆的二自由度自行车模型如图4-1所示，车辆行驶于某一点时，其后轴中

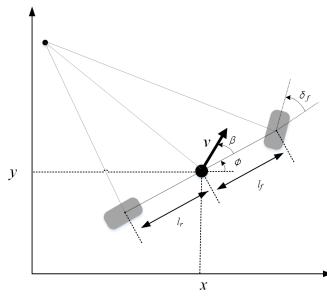


图 4-1 二自由度自行车模型

心 (x_r, y_r) 处速度可以通过公式4-2进行表示：

$$\begin{aligned}\dot{x}_r &= v \cos \theta \\ \dot{y}_r &= v \sin \theta\end{aligned}\quad (4-2)$$

车辆前后轴的运动学约束为：

$$\begin{aligned}\dot{x}_f \sin(\varphi + \delta_f) - \dot{y}_f \cos(\varphi + \delta_f) &= 0 \\ \dot{x}_r \sin \varphi - \dot{y}_r \cos \varphi &= 0\end{aligned}\quad (4-3)$$

联合公式4-2和4-3，可以建立 x 和 y 方向速度的相关微分方程：

$$\begin{aligned}\dot{x} &= v \cos(\delta_f + \beta) \\ \dot{y} &= v \sin(\delta_f + \beta)\end{aligned}\quad (4-4)$$

前轴中心位置 (x_f, y_f) 可以通过几何关系计算得到：

$$\begin{aligned}x_f &= x_r + l \cos \theta \\ y_f &= y_r + l \sin \theta\end{aligned}\quad (4-5)$$

基于前文中对车辆进行的刚体假设，车辆的角速度可以通过速度与转向半径的比值完成计算，如公式4-6：

$$\omega = \frac{v}{R} \quad (4-6)$$

车辆的前轮转角 δ_f 可以通过公式4-7进行计算：

$$\delta_f = \arctan \frac{l}{R} \quad (4-7)$$

完整的车辆运动学模型可以通过公式4-8进行描述：

$$\begin{cases} \dot{x}_r &= v \cos \theta \\ \dot{y}_r &= v \sin \theta \\ \dot{v} &= a \\ \dot{\theta} &= \frac{v}{l} \tan \delta_f \end{cases} \quad (4-8)$$

4.1.2 运动学跟踪误差模型

轨迹跟踪的核心是保持车辆在目标轨迹上行驶，而不仅仅是最小化车辆与目标点之间的距离。横向误差和航向误差可以准确捕捉到车辆的相对位置和方向，使控制器不仅能够纠正车辆的位置偏差，还能调整航向，使其对准轨迹方向。跟踪误差模型在轨迹跟踪问题中同样常用的一种模型，如图4-2所示，该模型是用来描述车辆实际轨迹与期望轨迹之间偏差的数学模型。因此跟踪误差模型所对应的状态量如公式4-9所示：

$$x = [e_{d_t}, \dot{e}_{d_t}, e_{\theta_t}, \dot{e}_{\theta_t}] \quad (4-9)$$

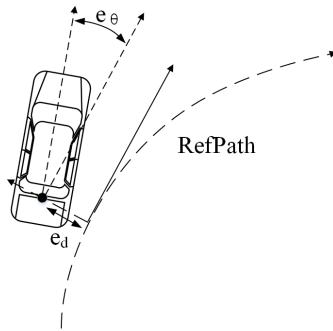


图 4-2 跟踪误差模型

式中， e_t, \dot{e}_t 分别为 t 时刻跟踪的横向误差和横向误差变化率， $\theta_t, \dot{\theta}_t$ 分别为 t 时刻跟踪的角度误差和角度误差变化率。 e_d, e_θ 具体计算公式如公式4-10所示：

$$\begin{aligned} e_d &= \sqrt{(x_r - x_{ref})^2 + (y_r - y_{ref})^2} \\ e_\theta &= \theta_{veh} - \theta_{ref} \end{aligned} \quad (4-10)$$

式中 (x_r, y_r) 为车辆的后轴中心位置， (x_{ref}, y_{ref}) 为参考轨迹上最近的投影点的坐标， $\theta_{veh}, \theta_{ref}$ 分别为车辆的航向角和投影点处的参考航向角。 e_t, \dot{e}_t 的估

计更新规则如公式4-11所示：

$$\begin{aligned} e_{d_t} &= e_{t-1} + \dot{e}_t dt \\ e_{\theta_t} &= e_{\theta_{t-1}} + \dot{e}_{\theta_t} dt \end{aligned} \quad (4-11)$$

横向误差 e_t 可以通过同时4-12完成计算：

$$\dot{e}_{dt} = v \sin(e_{\theta_{t-1}}) \quad (4-12)$$

由于车辆在行驶过程中，默认沿道路行驶，那么就可以假设角度误差 θ 足够小， $\sin(\theta) = \theta$ 成立，公式4-12可以被进一步简化为公式4-13：

$$\dot{e}_t = v e_{\theta_{t-1}} \quad (4-13)$$

车辆的角度误差 $\dot{\theta}_t$ 可以通过公式4-14完成计算，式中 δ_f 为前轮转角， l 为车辆轴距。

$$e_{\dot{\theta}_t} = \frac{v}{l} \tan \delta_f \quad (4-14)$$

当前轮转角 δ_f 足够小时， $\tan(\delta_f) = \delta_f$ 这一假设成立，公式4-14可以被简化为公式4-15：

$$e_{\dot{\theta}_t} = \frac{v}{l} \delta_f \quad (4-15)$$

通过公式 (4-9-4-15)，以及实现了完整运动学误差跟踪模型的推导，完整的模型可以写作如下形式。

$$\begin{cases} e_t = e_{t-1} + \dot{e}_{t-1} dt \\ \theta_t = \theta_{t-1} + \dot{\theta}_{t-1} dt \\ \dot{e}_t = v_t \sin(\theta_{t-1}) \\ \dot{\theta}_t = \frac{v_t}{L} \tan(\delta_f) \end{cases} \quad (4-16)$$

4.1.3 动力学跟踪误差模型

4.1.2节中完成了对运动学模型下的跟踪误差模型推导，但是运动学模型的推导过程中对车辆进行了刚体假设，忽略了车辆的侧偏和变形等因素，这种假设在低速状态下是成立的，因为在这个速度范围区间内，车辆运动受到的侧偏力和变形对车辆运动的影响较小，可以忽略不计。然而，随着行车速度的增加，这些因素开始变得显著，对车辆的行驶稳定性带来重要影响。

在高速行驶时，车辆受到的的侧偏力会随着速度的增加而增加，这会影响

车辆的横向加速度和航向角速度。侧偏力的增加可能会导致车辆的稳定性降低，增加失控的风险。此外，车辆的变形，如轮胎的弹性变形，也会对车辆的响应和稳定性产生影响。在高速行驶时，轮胎的变形可能会导致车辆的实际转向响应与预期的转向响应之间存在差异，从而影响轨迹跟踪的准确性。为了解决上述问题，提升车辆高速运动过程中的稳定性，本节将对跟踪误差模型的动力学形式进行推导。

二自由度车辆动力学模型的微分方程形式如公式4-17所示：

$$\begin{cases} m(\dot{v}_x + v_x \dot{\theta}) = F_{yf} \cos \delta_f + F_{yr} \\ I_z \ddot{\theta} = a \cdot F_{yf} \cos \delta_f - b \cdot F_{yr} \end{cases} \quad (4-17)$$

式中 m 为车辆的质量， v_x 为车辆的纵向速度， \dot{v}_y 为车辆的横向加速度， δ_f 为车辆的前轮转角， a, b 分别为车辆质心到前轴和质心到后轴的距离， $\dot{\theta}$ 为车辆的横摆角速度， I_z 为车辆质心绕 z 轴的转动惯量。上述公式是一个非线性的模型，并不适合直接用于控制器的设计，因此，考虑对模型进一步的线性化，简化模型的复杂度。对前轮转角进行小角度假设 ($\delta_f < 5^\circ$)，可以得到 $\cos \delta_f = 1$ ，轮胎模型可以被简化为轮胎侧偏力和侧偏角的一次线性关系，如公式4-18所示：

$$\begin{cases} F_{yf} = C_{af} a_f \\ F_{yr} = C_{ar} a_r \end{cases} \quad (4-18)$$

式中， a_f, a_r 为车辆的前、后轮侧偏角， C_{af}, C_{ar} 为前后轮的侧偏刚度。在小角度假设下，可以获得前后轮的侧偏角可以通过公式4-19完成计算：

$$\begin{cases} a_f = \frac{v_y + l_f \dot{\theta}}{v_x} - \delta_f \\ a_r = \frac{v_y - l_r}{v_x} \end{cases} \quad (4-19)$$

联立公式4-17 至 4-19，可以将动力学模型写成矩阵形式，如公式4-20：

$$\begin{bmatrix} \dot{v}_y \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{C_{af} + C_{ar}}{mv_x} & \frac{aC_{af} - l_r C_{ar}}{mv_x} - v_x \\ \frac{aC_{af} - bC_{ar}}{I_z v_x} & \frac{a^2 C_{af} + b^2 C_{ar}}{I_z v_x} \end{bmatrix} \cdot \begin{bmatrix} v_y \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} -\frac{C_{af}}{m} \\ -\frac{aC_{af}}{I_z} \end{bmatrix} \delta_f \quad (4-20)$$

横向误差模型的状态量 $[e_d, \dot{e}_d, e_\theta, \dot{e}_\theta]$, 可以通过如下公式计算

$$\begin{cases} e_d = v_x e_\theta + v_y \\ \dot{e}_d = v_x \dot{e}_\theta + \dot{v}_y \\ e_\theta = \theta_{veh} - \theta_{ref} \\ \dot{e}_\theta = \dot{\theta} \end{cases} \quad (4-21)$$

将公式4-21代入4-20中, 即可获得动力学横向误差模型, 如图4-22所示。

$$\begin{cases} e = e, \\ \dot{e}_d = \left(\frac{C_{af}+C_{ar}}{mv_x} \right) \dot{e}_d + \left(-\frac{C_{af}+C_{ar}}{m} \right) e_\theta + \left(\frac{aC_{af}-bC_{ar}}{mv_x} \right) e_q + \left(\frac{aC_{af}-bC_{ar}}{mv_x} - v_x \right) \dot{\theta}_r + \left(-\frac{C_{af}}{m} \right) \delta \\ e_\theta = \dot{e}_q \\ \dot{e}_\theta = \left(\frac{aC_{af}-bC_{ar}}{I_z v_x} \right) \dot{e}_d + \left(-\frac{aC_{af}-bC_{ar}}{I_z} \right) \dot{e}_\theta + \left(\frac{a^2 C_{af} + b^2 C_{ar}}{I_z v_x} \right) \dot{e}_q + \left(\frac{a^2 C_{af} + b^2 C_{ar}}{I_z v_x} \right) \dot{\theta}_r + \left(-\frac{aC_{af}}{I_z} \right) \delta, \end{cases} \quad (4-22)$$

4.2 基于 PID 的纵向控制器

PID (比例-积分-微分) 控制算法是结合比例、积分和微分三种环节于一体的经典控制算法。自 20 世纪 30 年代由美国工程师泰勒 (Taylor Instruments) 和俄国学者尼古拉斯·米诺尔斯斯基 (Nicolas Minorsky) 提出雏形以来, 经过近一个世纪的发展与完善, 已成为连续控制系统中技术最成熟、适应性最强的控制策略。其核心优势在于无需精确掌握被控对象的数学模型, 仅通过误差反馈即可实现稳定控制, 因此在工业过程控制、机器人运动控制、航空航天等复杂系统中得到广泛应用。

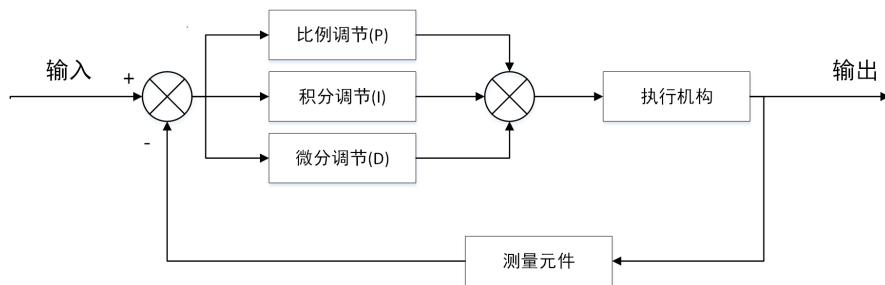


图 4-3 PID 控制算法

如图4-3所示, PID 算法的核心架构包含三个并行作用的控制环节: 比例环

节提供即时响应，积分环节消除稳态误差，微分环节预测未来趋势。该算法通过实时计算被控系统输出与目标值的偏差 $e(t)$ ，对偏差信号进行时域内的比例放大、历史累积和趋势预测，最终将三路信号线性叠加生成控制量 $u(t)$ 。其数学描述如公式4-23所示：

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p \tau \frac{de(t)}{dt} \quad (4-23)$$

其中， K_p 为比例系数， T_i 为积分时间常数， τ 为微分时间常数， $e(t)$ 为状态偏差， $u(t)$ 为控制输出。三项协同工作的物理意义可进一步阐释为：比例项计算被控系统当前状态与目标状态之间的偏差值，所以其目标是使系统尽可能快速达到目标状态，即消除系统当前时刻的偏差值；积分项所计算的为被控系统在控制过程中所产生的累积误差，其目标为使被控系统尽可能达到目标状态，消除被控系统的稳态误差；微分项计算的为误差的变化趋势，其目标为尽可能消除误差变化的趋势，使系统趋于稳定。

由此可见，比例控制是 PID 控制的核心，用于消除被控系统当前时刻的误差，积分控制的主要辅助比例控制，在系统在趋于稳定时，尽可能减少系统的稳态误差，而微分控制主要是为了增强系统的鲁棒性，增加系统阻尼程度，减少 PI 控制所带来的超调而存在。

进一步地，对车辆的纵向控制来说，状态偏差可以被描述为车辆当前速度 $v(t)$ 与目标速度 $v_{ref}(t)$ 时刻的差值，即公式4-24

$$e(t) = v(t) - v_{ref}(t) \quad (4-24)$$

4.3 基于前馈 LQR 的横向控制器

横向控制器是自动驾驶系统的核心组成部分之一，其主要功能为通过控制车辆转向系统使车辆按照期望的参考路径行驶的同时保证车辆的安全性和稳定性。LQR(线性二次调节器)作为一种成熟且先进的控制策略，因其在优化控制性能和鲁棒性方面的优势而被广泛应用。本节将探讨基于 LQR 的前馈横向控制器的构建及其实现。

LQR (Linear Quadratic Regulator, 线性二次调节器) 是一种经典的最优控制算法，能够用于解决线性系统的最优控制问题。该算法的目标在于找到一条控制律，使得整个被控系统从某一状态出发，能够以最小代价达到稳定的平衡状态或跟踪某一给定的轨迹。

一个离散时域下的 LQR 问题，可以构造成如下形式：

- 状态方程：

$$x_{k+1} = Ax_k + Bu_k \quad (4-25)$$

- 目标函数：

$$J = \frac{1}{2}x_n^T Q x_n + \frac{1}{2} \int (x_k^T Q x_k + u_k^T R u_k) \quad (4-26)$$

式中， x 为系统的状态矩阵， u 为系统的控制矩阵， Q 和 R 是半正定的对角权重矩阵。对于这样一个二次最优问题，通过求解 Hamilton-Jacobi-Bellman (HJB) 方程，可得 LQR 的最优控制律为状态反馈形式：

$$\begin{aligned} u_k^* &= -K x_k \\ K &= R^{-1} B^T P \end{aligned} \quad (4-27)$$

其中，矩阵 P 为代数 Riccati 方程 ARE(Algebraic Riccati Equation) 的唯一半正定解，可以通过公式4-28完成迭代求解。

$$P = A^T P A - (A^T P B)(R + B^T P B)^{-1}(B^T P A) + Q \quad (4-28)$$

自动驾驶汽车的横向控制问题可以被进一步描述为最优控制问题。其核心在于通过设计合理的控制策略，使车辆能够精确地沿着预设路径行驶，同时最小化横向偏差和航向角偏差。这一过程需要综合考虑车辆的动力学特性、道路环境以及控制输入的约束条件。具体而言，横向控制的目标是通过调整车辆的前轮转角，实时修正车辆的横向位置偏差和航向角偏差，从而确保车辆在复杂道路条件下保持稳定、安全的行驶姿态。对自动驾驶汽车的轨迹跟踪问题，其误差动力学模型可以写作公式4-29：

$$\dot{x}_k = Ax_k + Bu_k + C\kappa_k \quad (4-29)$$

式中， $x = [e_d, \dot{e}_d, e_\theta, \dot{e}_\theta]$ 为车辆的状态量， $u = [\delta_f]$ 为车辆的控制量，前轮转角， κ 为道路曲率。由于 $C\kappa$ 是一个与状态量 x 和控制量 u 都无关的曲率常数项，因此在建模过程中以及目标函数的构建过程中往往可以被忽略，因此，动力学模型可以被简化为公式4-25。

对一个轨迹跟踪问题，其优化目标为最小化车辆运行过程中的跟踪误差和航向角误差，减少车辆的侧向速度，并减少使整个系统到达稳定状态过程中所需

要的控制量消耗。对这样一个多目标优化问题，其目标函数可以写作公式4-30。

$$J = \sum_{k=0}^N (x_k^T Q x_k + u_k^T R u_k) \quad (4-30)$$

式中， Q 为状态量权重矩阵， R 为控制量权重矩阵， Q, R 通常为严格的对角矩阵。式中第一项 $\sum x_k^T Q x_k$ 用于描述完整状态转移过程中与期望状态之间的离散误差累积和， $\sum u_k^T R u_k$ 描述状态转移过程中所施加的控制量总和。

上述多目标优化问题，可以被进一步转换为 bolza 形式的最优控制问题，求解最优控制量 u^* （在本问题中为车辆的前轮转角 δ_f ），使得价值函数 J 取得最小值，即在保证车辆稳定的前提下，使用最小控制量使得车辆朝目标状态靠近。

由于该最优控制问题的目标函数仅包含二次项，且仅仅受到连续两个状态转移之间的运动学约束，这样一个问题是典型的 LQR 问题，该问题的解可以通过公式4-28中的黎卡提方程，完成迭代求解，当前时刻的最优控制量 u^* 可以通过 $u = -Kx$ 完成计算， K 为前馈矩阵， $K = [K_1, K_2, K_3, K_4]$ 。

但是，对 $u = -Kx$ 这一最优反馈控制量，是满足简化动力学约束 $x_{k+1} = Ax_k + Bu_k$ 的最优控制量，而非实际的最优控制量。因为求解过程中，由于对状态量和控制量的微分操作，曲率常数项 κ 被省略，但是由于这一项的省略，使得最优控制量 u^* 的求解过程中忽略了道路曲率的变化，使得该控制量并非实际的最优控制量 u^* 。

将通过黎卡提方程求解 LQR 问题所获得的最优控制量 $u^* = -Kx$ 代入完整的误差动力学模型中，可以得到公式4-31：

$$\dot{x}_k = (A - BK)x_k + C\kappa_k \quad (4-31)$$

由于误差动力学模型当中，所有状态量均为误差项，当系统趋于稳定时，状态量 $x = 0$ 和状态变化量 $\dot{x} = 0$ 成立。因此，令状态量变化 $\dot{x} = 0$ ，公式4-31可以被进一步转换为公式4-33：

$$x_k = -(A - BK)^{-1}C\kappa_k \quad (4-32)$$

在 κ_k 不为 0 的情况下，无论 K 取任何值， $x_k \neq 0$ 。也就是说，无论 K 取任何值， x_k 和 \dot{x}_k 无法同时为 0，整个系统的稳态误差存在。可以通过公式4-33被显式地完成计算。

为了解决稳态误差存在的问题，引入前馈控制量 δ ，最优控制量 $u^* = -Kx$

可以被改写为 $u^* = -Kx + \delta$, 稳态误差可以被进一步改写为公式4-33:

$$x = -(A - BK)^{-1}(Bu + C\kappa) \quad (4-33)$$

该问题可以被进一步改写为, 如何选取合适的前馈量, 使得稳态误差尽可能接近 0。借助一些软件计算, 可以获得最优的前馈控制量 δ 。在给定运动学误差模型下, 车辆的稳态误差可以表示为

$$x = \begin{bmatrix} e_d \\ \dot{e}_d \\ e_\theta \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} -\frac{l\kappa_{ref} - \tan \delta_f}{K_1 v} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4-34)$$

稳态误差仅留下 e_d 一项, 为了使 e_d 的稳态误差为 0, 需要公式4-35成立

$$l\kappa_{ref} - \tan \delta_f = 0 \quad (4-35)$$

最优控制前馈可以被计算为

$$\delta = \arctan l\kappa_{ref} \quad (4-36)$$

对动力学模型, 其前馈控制量可以以相同的方式完成计算, 完整系统的稳态误差为

$$x = \begin{bmatrix} \frac{1}{K_1} \left\{ \delta_f - \frac{\dot{\theta}_{ref}}{v_x} [a + b - bK_3 - \frac{mN_x^2}{a+b} (\frac{b}{C_f} + \frac{a}{C_r} K_3 - \frac{a}{C_r})] \right\} \\ 0 \\ -\frac{\dot{\theta}_r}{v_x} (b + \frac{a}{a+b} \frac{mN_x^2}{C_{ar}}) \\ 0 \end{bmatrix} \quad (4-37)$$

因此, 动力学模型下的前馈可以被计算为

$$\delta = \frac{\dot{\theta}_r}{v_x} \left[a + b - bK_3 - \frac{mv_x}{a+b} \left(\frac{b}{C_{af}} + \frac{a}{C_{ar}} K_3 - \frac{a}{C_{ar}} \right) \right] \quad (4-38)$$

完整的控制输出可以被写作反馈输出和前馈输出之和, 如公式4-39

$$u = -Kx + \delta \quad (4-39)$$

4.4 实车轨迹跟踪验证

为了验证本文所提出的跟踪算法的有效性，本文利用某国产线控底盘，在国内某实车测试场开展了实车实验，对车辆的跟踪性能展开了进一步的验证。该试验场的俯视图与高精地图分别如图4.4(a),4.4(b)所示：

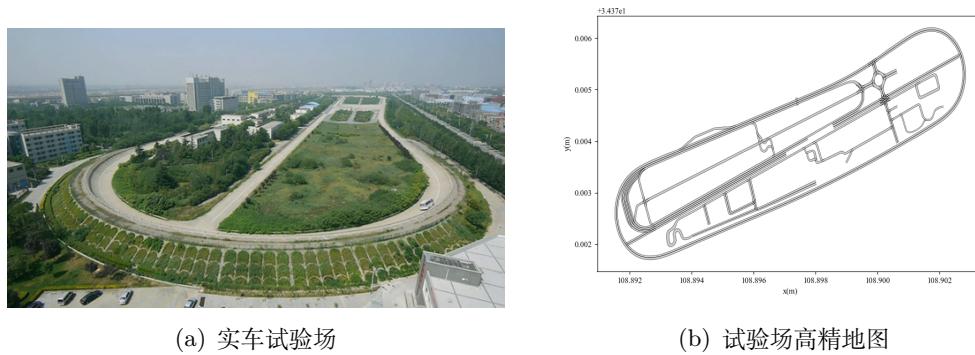


图 4-4 实车实验测试场地

车辆的其他动力学参数如下表4-1所示：

表 4-1 高级别自动驾驶多功能车技术参数

车辆参数名称	具体数值
整车尺寸 (长 * 宽 * 高 mm)	3600x1605x1950
轴距 (mm)	2490
最大设计时速 (km/h)	60
续航里程 (km)	>100
轮胎规格	165/65R14
轮辋材质	铝合金-钢轮辋 (加轮罩)
最小离地间隙 (mm)	150
爬坡度	>15%
前轮距 (mm)	1415
后轮距 (mm)	1415
整备重量 (kg)	950
满载重量 (kg)	1325
最小转弯半径 (mm)	<6500

完整的跟踪效果如图4-6所示，本文算法在实车测试过程中表现良好，能够较好地跟随所设定的跟踪路线，跟踪过程平滑，不存在较大偏离的情况。跟踪过程中，车辆在直线段和小曲率弯道的跟踪效果较好，在大曲率弯道及环岛通

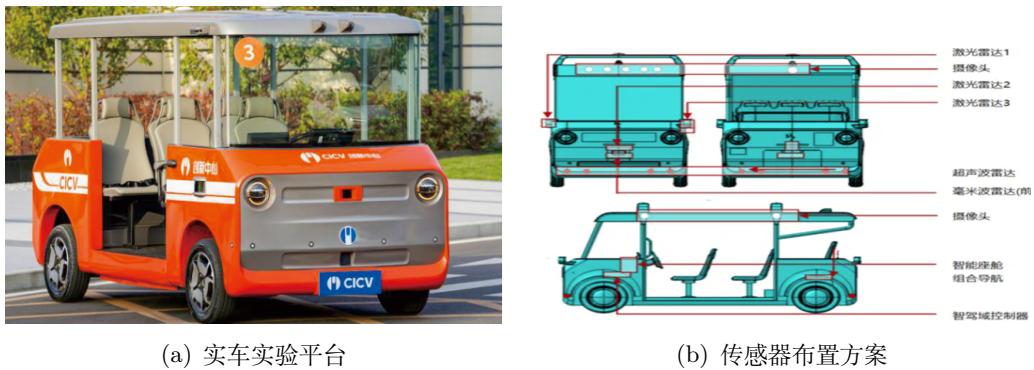


图 4-5 实车测试平台

行过程中会出现一定程度的震荡现象，因此，接下来对过弯及环岛通行场景展开进一步的分析。

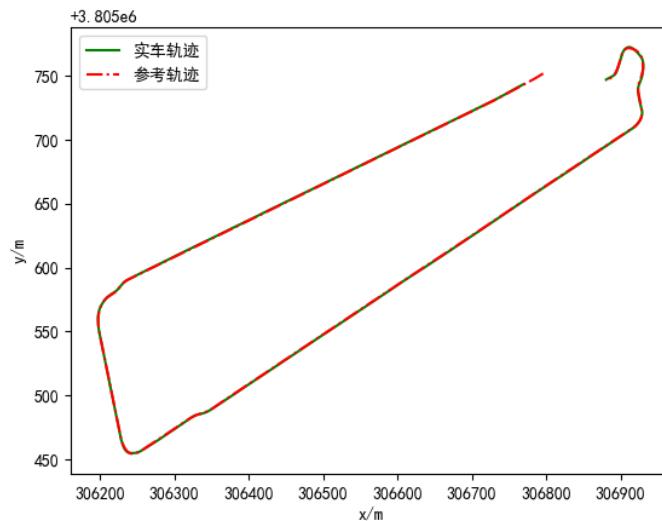


图 4-6 整圈跟踪效果

换道过弯段的跟踪效果和该过程中的具体运动参数如图4-7和图4-8所示。车辆在过弯和换道的过程中，跟踪效果良好，并不存在明显的超调现象，整体轨迹与期望轨迹几乎一致。跟踪过程中最大误差出现在换道过程中，最大横向误差为 0.24m，最大角度误差为 0.02rad，整体在可接受范围内。在转向和换道路段，前轮转角和角速度变化均匀，在该过程中，车辆的最大前轮转角为 0.25rad，距离车辆的转向极限值 0.6rad，仍有一定安全冗余；最大角速度为 0.25rad/s，出现在过弯过程中，仍在可控的舒适范围内，且整体运动过程中角速度无突变，保证了跟踪过程中的运动舒适性和稳定性。

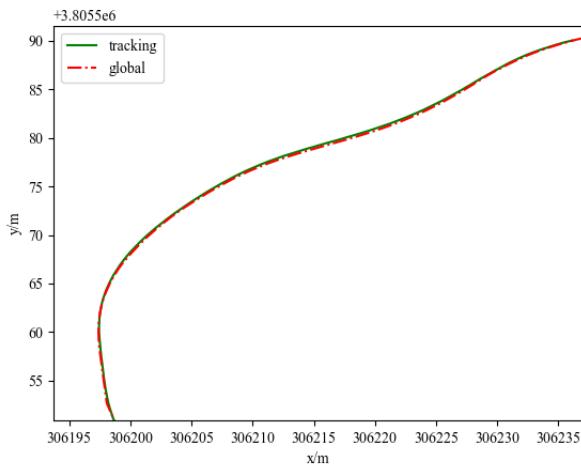


图 4-7 换道过弯段跟踪效果

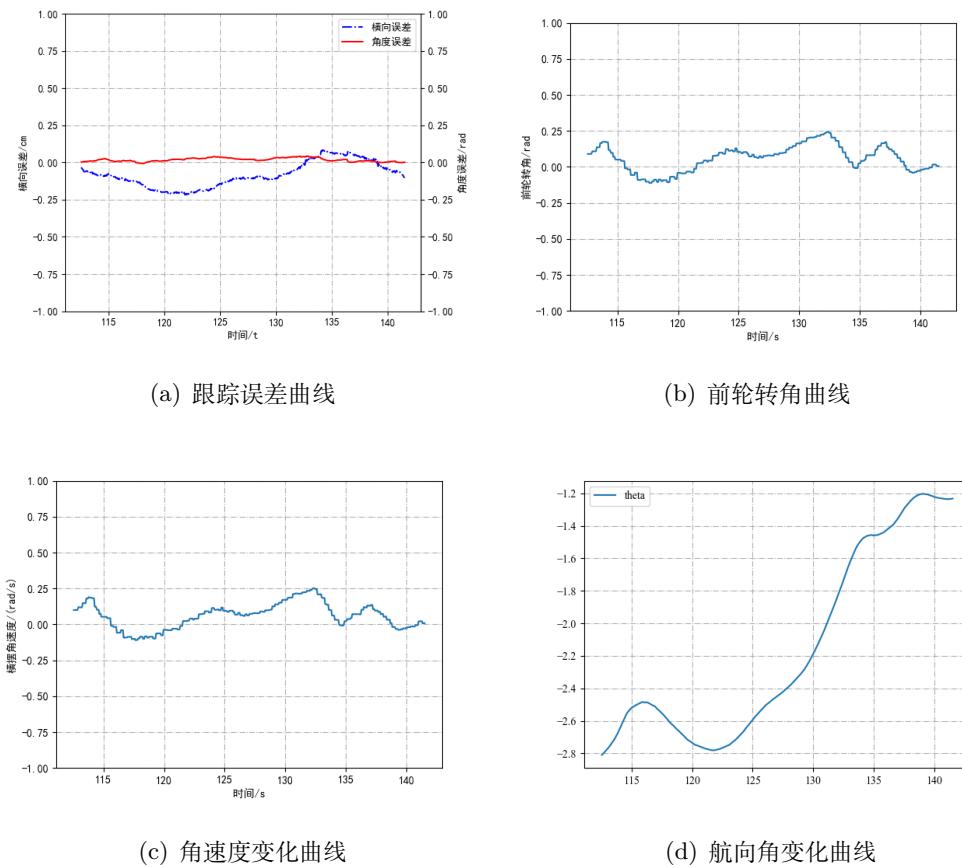


图 4-8 换道过弯跟踪运动参数

环岛通行路段的跟踪效果如图4-9所示，跟踪效果良好，但是相较于换道转

向路段，跟踪效果出现了一定程度的下降。在环岛通行过程中，面对连续的弯道，算法的横向误差和横摆角误差出现了一定程度的振荡现象，最大横向误差为0.28m，最大横摆角误差为0.28rad。车辆在该过程中的最大前轮转角为0.37rad，整体变化过程相对平滑，无急转向情况；在进入换道的过程中达到最大角速度，为0.46rad/s，在车辆稳定行驶和驾驶员的舒适范围内，且不存在角速度剧烈变化的情况。

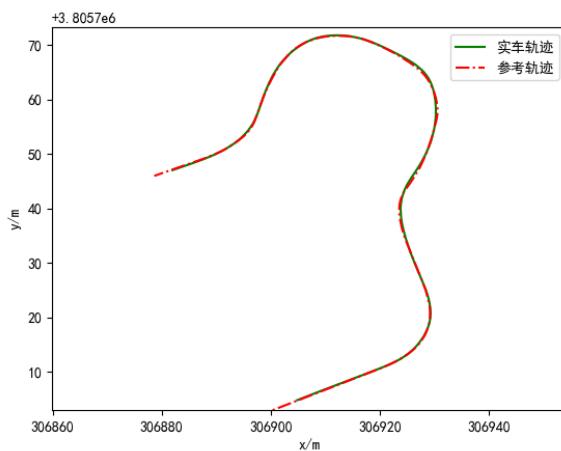


图 4-9 环岛跟踪效果

如前文所述，本文所构建的面向行泊一体的轨迹跟踪控制器能够满足跟踪需求，能够叫好地跟踪目标轨迹，能够很好地完成不同场景下地运动控制和轨迹跟踪任务。

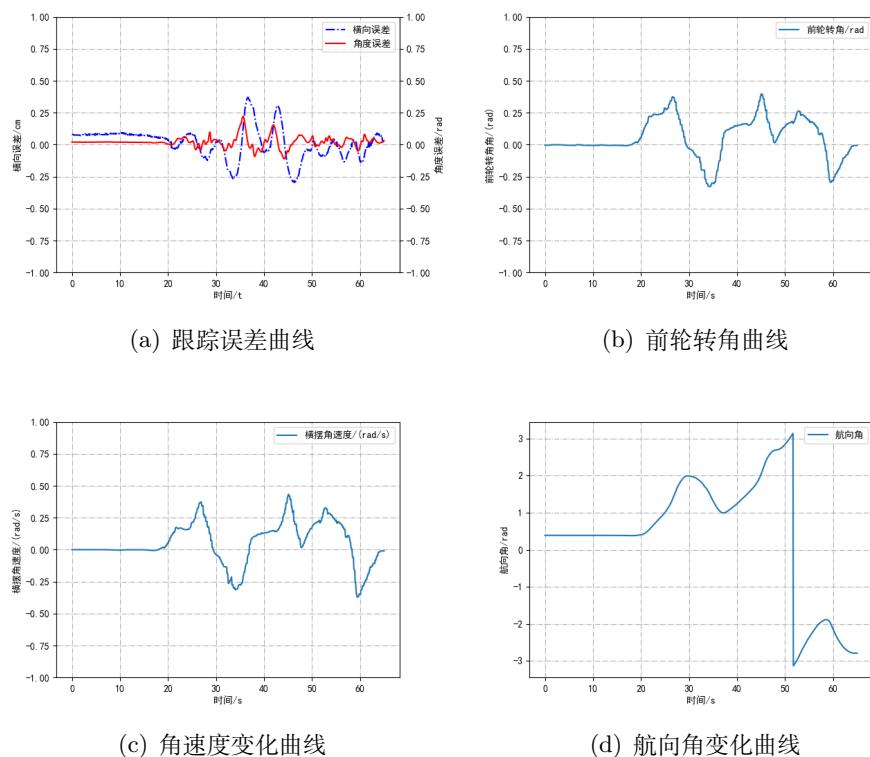


图 4-10 环岛跟踪运动参数

第 5 章 虚实融合实车验证

在前文 2-4 章中，已经介绍了针对动态交互环境和极端狭窄空间的运动规划算法及行泊一体轨迹跟踪控制器。为了进一步验证本文提出算法的有效性，本章节将利用虚实融合实车验证平台，对不同工况下算法的表现进行测试及验证，并对不同场景下的测试结果进行分析。

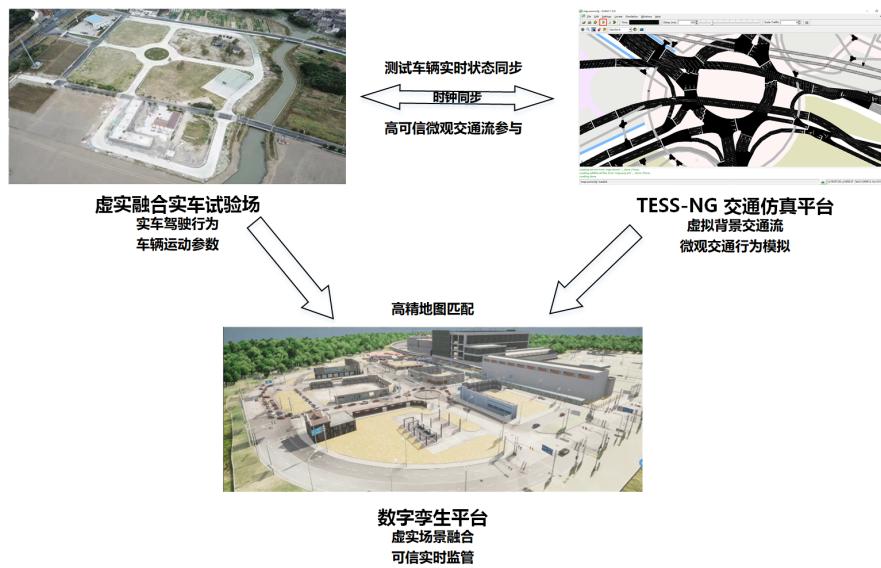
5.1 虚实融合测试

虚实融合实车测试平台，是由长安大学、同济大学等国内顶尖研究机构共同研发的新一代自动驾驶测试工具链。该平台旨在解决当前自动驾驶仿真测试工具链中存在的仿真真实性不足、长尾问题频发、传感器仿真失真及测试效率低下等核心挑战。

如图5-1虚实融合测试系统主要由三大核心部分组成：具备高精度感知与可靠通信能力的封闭试验场、支持状态数据上传与虚拟场景注入的测试车辆、以及拥有实时数字孪生与动态仿真能力的云端平台。这三大部分通过多模式互联互通进行信息交互与协同工作，共同构成了完整的测试体系^[82]。通过云端构建真实试验场的数字孪生平台，利用多模异构车-云-场通信网络，实现了云平台与测试车辆间的双向低时延、高可靠性交互。通过路侧与车载数据采集设备，车辆的运动状态能够实时映射至云平台的虚拟仿真环境中；同时，云平台动态生成的虚拟测试场景也能实时注入给待测的自动驾驶汽车，确保车-云-场三者能够一体化同步运行，从而大幅提升自动驾驶汽车的测试效率。该系统主要面向自动驾驶整车级平台，通过融合真实测试场地环境与云端数字孪生仿真系统，能够执行加速测试、大规模测试及强化测试等多种测试任务。它是弥补国际“三支柱”测试方法（即软件仿真测试、公共道路测试、封闭场地测试）在测试准确度、测试效率及测试安全性等方面不足的有效解决方案。

5.2 实车实验验证

为了充分验证本文所提出的规划及跟踪控制算法的有效性，本文依托某虚实融合测试场和虚实融合自动驾驶工具链对本文的算法展开相应的实车测试（实验车与测试场与4.4节中一致）。该试验场集成 LTE-V，5G 等多种无线网络，



且具备城市，高速，乡村公路等多种路段，可以满足多场景自动驾驶测试的需要。

本文在整圈测试中设置了 11 个具体的测试场景，如图5-3所示：分别为礼让行人横穿斑马线，低速物流车占道，高密度交通流紧密跟驰，高速公路切入变道，停车让行标志响应，合流区汇入行驶，施工区避让通行，社会车辆随机干扰，无信号交叉口饱和交通流通行，环岛汇入汇出以及垂直车位自动泊车，涵盖了行车过程中的大部分场景。

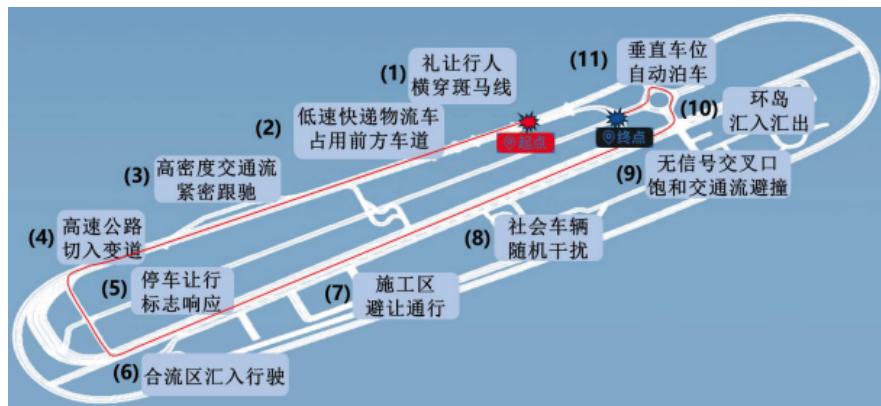


图 5-3 虚实融合测试场景

经测试，本文算法能够顺利通过上述 11 个场景，完成整圈虚实融合测试。考虑到上述 11 个场景中，部分场景之间存在一定的重复性，本文将上述 11 个场景中选取五个代表性场景展开分析。分别为礼让行人，紧密跟驰，无信号合



图 5-2 虚实融合仿真测试平台

流汇入，换道合流，垂直泊车。

为了更好地对测试场景进行描述，本文对图中的元素作出一定的约定，所有场景图中，被控车辆的颜色统一为蓝色，背景车辆的速度越慢，颜色越接近红色，速度越快，速度越接近绿色，在车辆的连续运动过程中，透明度越高，表明该时刻越靠近起点，颜色越深，距离起始时刻越远。

5.2.1 低速车辆占道通过测试

低速车辆占用前方道路场景设置如下，测试道路为三车道长直道，蓝色被控车辆位于中间车道，行车速度为 21km/h，正前方存在一辆缓行车辆，行车速度为 3km/h，后方存在一跟驰车辆，右侧车道存在四辆缓行车辆，均位于车辆的前方，左侧车道存在连续交通流，但车辆之间间隙较大，存在一定的换道空间。车辆在该场景下的虚实融合结果如下图5.4(a)所示。

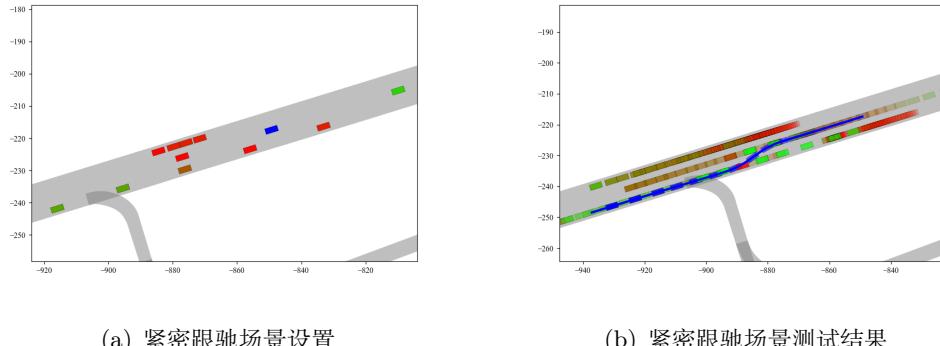


图 5-4 紧密跟驰场景

在测试过程中随着车辆的靠近，测试场景被触发，背景车辆开始加速，其颜色由红色逐渐向绿色过渡，但是车辆在中间车道的行驶过程中，由于前方缓行车辆的遮挡，无法满足车辆在通行效率方面的期望。因此，车辆在跟驰一段时间后，发现前方车辆无高速欲望，左侧车道行驶效率较快且存在可行换道空间之后，车辆的换道欲望积累到一定阈值之后，选择向左侧车道换道，并加速超越前方占道车辆。

被控车辆在该测试场景下的具体参数如图所示。车辆在换道超越过程中的运动轨迹，整体过渡平滑且不存在明显的震荡现象，平顺性和舒适性俱佳。被控车辆在 70s 时开始规划换道行为，在 75.2s 时开始进行换道，在 81.3s 时完成整个换道行为，整个换道过程持续约 6.1s。图 5.5(a)和 5.5(b)展示了该场景中车辆的速度和加速度变化曲线，面对前方的低速行驶车辆，被控车辆首先从 24km/h 减速至 1km/h，整个过程持续了约 6s，在刹车过程中最大减速度为 $-1.8m^2$ ，整体刹车过程平滑，无急刹行为。图 5.5(c)展示了车辆的前轮转角变化曲线，车辆在直行过程中，车辆的车轮转角保持在 ± 0.5 度之内，考虑车道并不完全笔直，虽然存在一定的曲率变化，但是仍然在可控范围内；在换道过程中，车辆的前轮转角呈现均匀的 S 型变化，最大车轮转角仍被控制在 ± 8 度内，在完成换道过程之后，前轮转角被控制在 ± 0.5 度之内，无明显超调，有效保证了车辆运行过程中的横向稳定性。

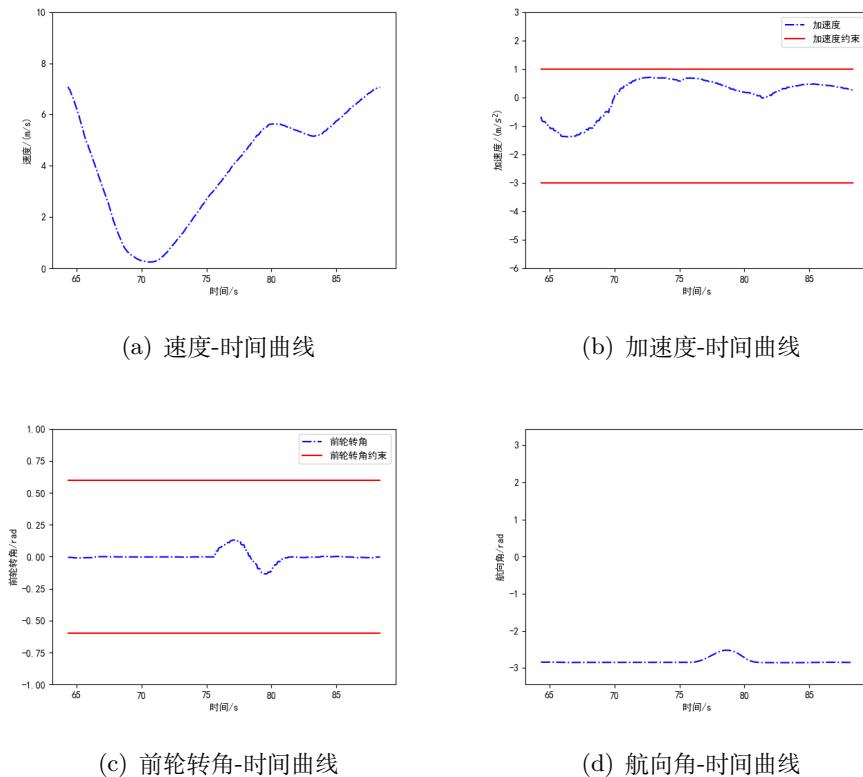


图 5-5 低速车辆占道通过测试结果

5.2.2 无信号合流汇入测试

匝道合流汇入场景设置如下，自车位于汇入匝道中，运行速度约 15km/h，自车后侧存在跟驰车辆，匝道主路存在连续行驶的交通流，且由于交通流之间的跟驰交互行为，位于交通流头部的车辆运动速度明显高于交通流尾部的车辆，且越靠近尾部的车辆，运行速度越慢，所以背景交通流呈现出头部车辆为绿色，尾部车辆为红色的情况。被控车辆需要从主路匝道，逐步向主路合流。

车辆在该场景下的运动过程如图5.6(b)所示。测试过程中，被控车辆在汇入段之前，始终保持道路中间行驶，在进入汇入合流段之后，车辆开始尝试将车头汇入主路，第二辆背景车在感知到被控车辆的匝道合流行驶之后进行了刹停等待让行行为，本文算法发现了可行轨迹之后，立刻向主路靠拢，超越主路第二辆背景车，安全地完成了匝道合流。

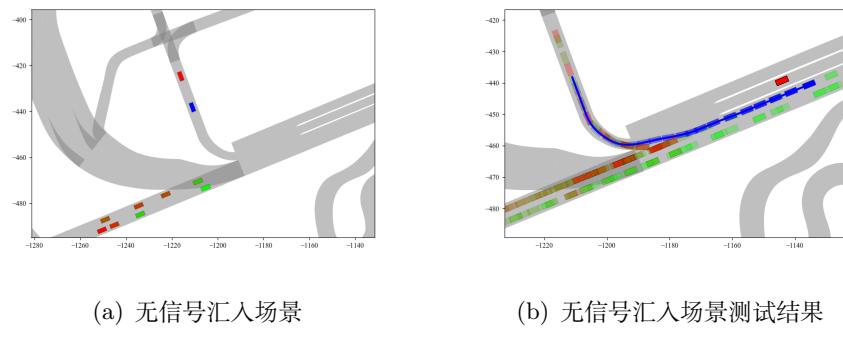


图 5-6 无信号汇入场景

车辆在该场景下的具体运动参数如图5.7所示。图5.7(a)展示了车辆在该场景下的速度变化曲线，车辆在进入匝道段的过程中，车辆的行车速度逐步从15km/h降低至11km/h，由于匝道段是一段曲率较大的弯道段，本文速度规划算法考虑了运动过程中的横向稳定性，在过弯过程中(200s-210s)进行了小幅度的减速行为，在210s-213s时，车辆开始减速并向主路靠拢，察觉背景车刹停让行后，车辆立即采取了过段的加速行为，汇入主路，但是在汇入过程的末端，由于最右侧车辆的干扰，考虑行车过程中的安全性，采取了一定程度的减速，在实际成熟后，果断平稳加速，完成主路的汇入，整个纵向运动过程中，最大加速度控制在 $\pm 0.8m^2$ 范围内，且过渡平滑，无明显急刹。车辆的前轮转角变化曲线如图5.7(c)所示，在车辆的横向运动过程中，车辆的前轮转角整体过渡平滑，无急转向行为，前轮在匝道段的前半段和到达最大值，约16.2度，与最大前轮转角仍然有一定的差距；在到达汇入区域之后，前轮转角快速由正变负，完成换道汇入操作后回正方向盘，实现了车辆的合流汇入。

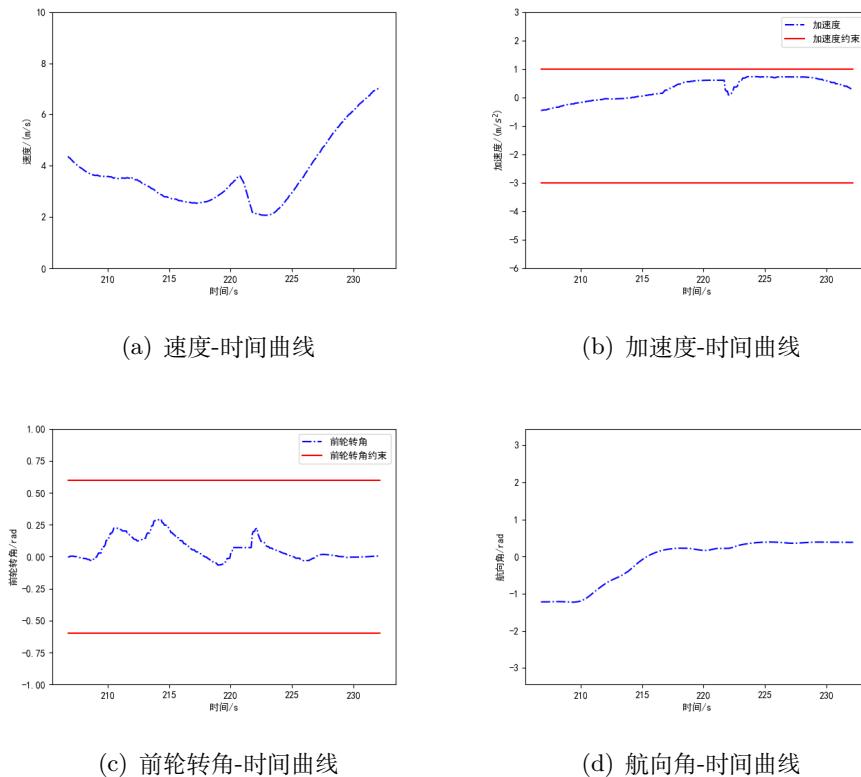


图 5-7 无信号汇入具体测试结果

5.2.3 环岛汇入汇出测试

环道合流测试场景的设置如下。被控车辆位于环岛入口处，运动速度为3.5km/h，环岛中存在正常运动的交通流，环道右侧汇入口中存在三辆等待汇入的车辆，第一辆车会在场景触发时刻汇入环岛，第二辆车会在车辆位于右侧匝道前方时尝试汇入匝道，尝试对车辆的正常运动造成干扰。被控车辆需要从匝道入口进入，让后在匝道中绕行一圈，在环道左侧匝道驶出，完成环岛的安全驶入驶出。

车辆在环道入口处时，由于前方背景车和后方背景车与自车存在一定的安全距离，难以对车辆的正常行驶造成干扰，因此车辆能够正常地进入匝道，再经过第一个匝道口时，由于右侧匝道第二辆背景车尝试汇入主路，算法察觉到前方车辆的汇入操作后，采取了一定的刹车操作，在完成对前方车辆的让行之后，跟驰前方车辆并在环岛左侧出口驶出匝道。

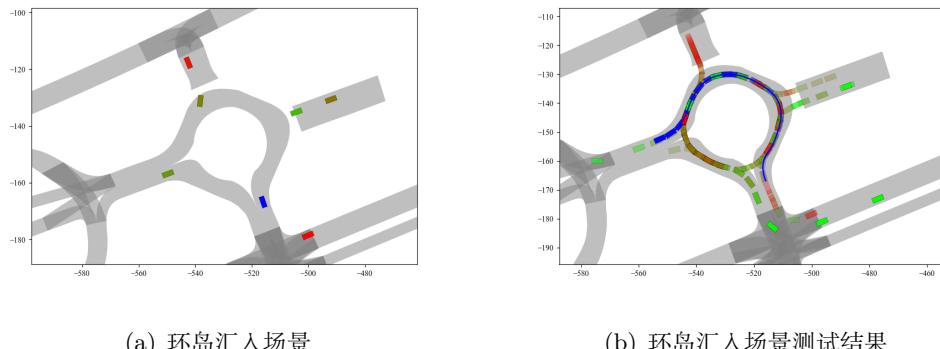


图 5-8 环岛汇入场景

车辆在该场景下的具体运动参数如图5-9所示。图5.9(a)展示了环道通行过程中的速度曲线，在环道通行过程中，整体速度曲线平稳，考虑环道通行过程中的横向稳定性，车辆的纵向运动速度被约束在 14km/h 内，在环道右侧汇入段第二辆车尝试汇入过程中，自车采取减速避让行为，因此速度从 8.3km/h 在 4.4s 内逐步降低至 3.1km/h ，在完成避让之后，逐步恢复正常环道通行速度 12km/h ，整体过程中最大加速度保持在 $\pm 1\text{m/s}^2$ 内，无急刹车和急减速行为，整体纵向运动过程稳定。图5.9(c)展示了车辆的前轮转角变化曲线，在汇入初期，车辆需要向右侧进行转向，完成环岛的汇入，并达到前轮转角最大值 19 度，在完成汇入之后，由于环道路段曲率一直不断变化，所以在环岛运动过程中，车辆的前轮转角一直维持一定的水平，在驶出环岛时，由于运动方向发生变化，前轮转角的方向发生转变，由左转变为右转，完成驶出。图 5.9(d)展示了环岛运动过程中的航向角变化曲线，无突变急转向行为，横向运动过程平稳。

5.2.4 泊车测试

泊车测试场景为虚实融合测试的最后一个场景，具体设置如图5.10(a)所示，车辆离开环岛后进入泊车测试场景，车辆的正前方存在五个停车位，但是车位的姿态与车辆当前不一致，并且五个车位中会存在两个随机停放的车辆作为干扰车辆，对车辆的泊入过程带来一定的干扰。

本文算法在泊车测试过程中的具体运动过程如图5.10(b)所示，本文算法选择泊入了右侧第一个车位，仅用一次转向便完成了直行入围，且泊入轨迹平滑。图5-11展示了运动过程中的具体数据，车辆在泊车入位之前，先进行了减速，所以泊入起始速度为 0.4m/s ，在正向泊入过程中，并不涉及车辆前进/后退挡位切换，在运动过程中车辆先加速 3.7m/s 之后，开始缓慢刹停，直至到达终点，整

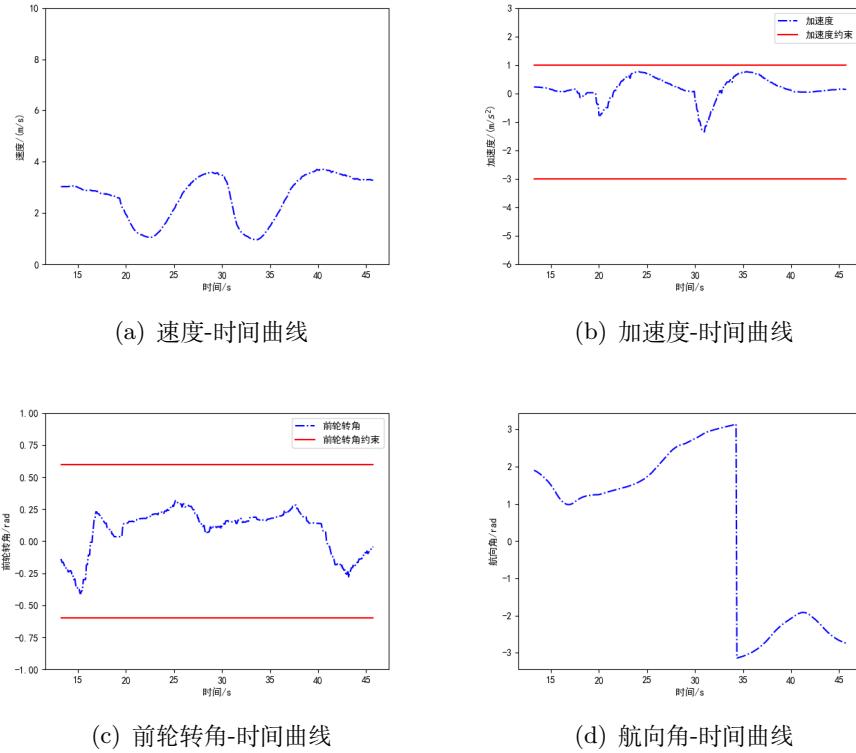


图 5-9 环岛汇入具体测试结果

体纵向轨迹平滑，运动过程中最大加速度为 $0.8m/s^2$ ，最小加速度为 $-0.81m/s^2$ ，无急加速急减速行为，有效保证的运动过程中的舒适性和稳定性。前轮转角在开始转向时，出现了较大变化，考虑到车辆速度较低，仅为 $3.75m/s$ ，因此对车辆的稳定性无较大影响。最大前轮转角为 $-0.263rad$ ，运行过程稳定。

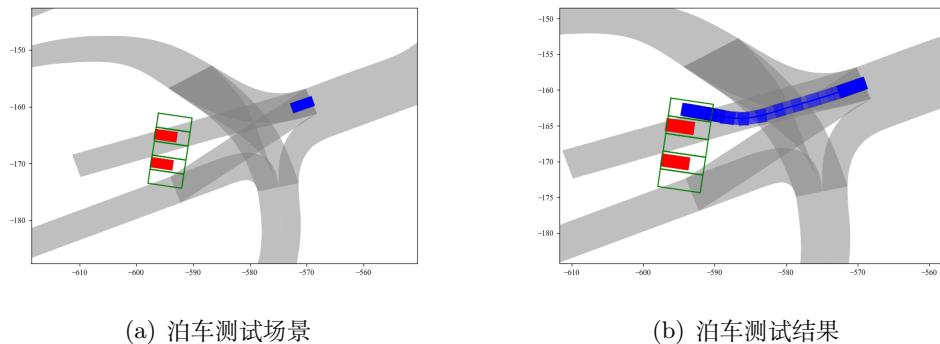


图 5-10 泊车测试场景

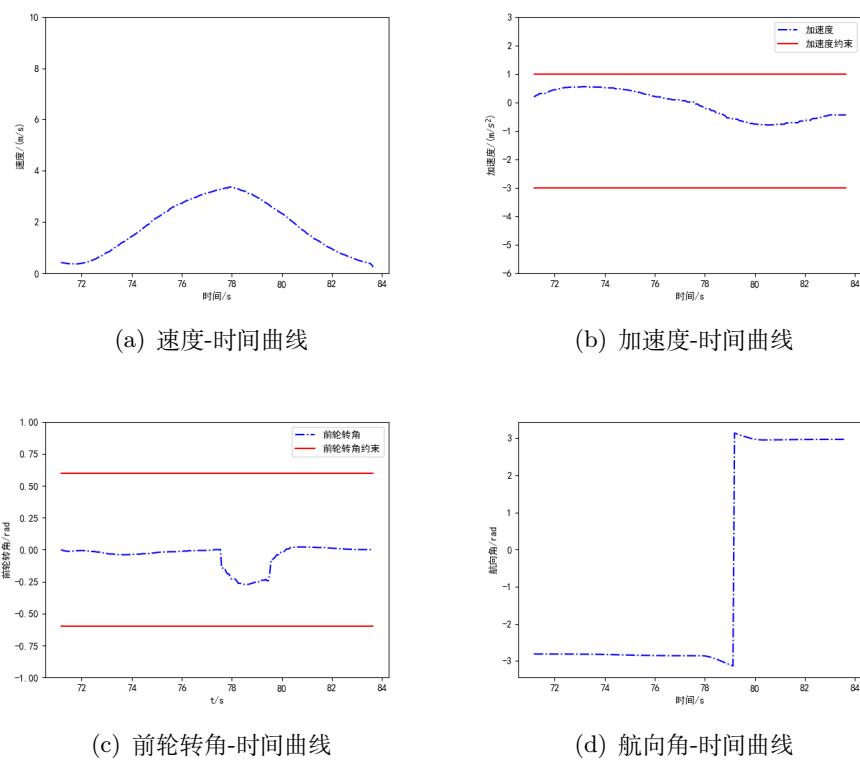


图 5-11 泊车入位具体测试结果

第 6 章 结论与展望

6.1 全文总结

本文面向行泊一体自动驾驶系统的现实需求，对其主要应用场景，静态狭窄通过/泊车场景和复杂动态交互行车场景下的运动规划问题，展开了深入研究。针对两种主要应用场景，考虑其应用场景和环境特性，设计了两种不同的算法，使得车辆能够在不同的场景下高效完成指定的运动规划任务。全文对所用算法进行了详细的公式推导，并进行了实车虚实融合实验，体现了本文算法在行泊一体场景下的有效性。本文的主要工作如下：

1. 介绍了行泊一体自动驾驶系统的发展缘由和出现契机，总结概括了自动驾驶是运动规划方法国内外研究现状，为本文的研究提供了坚实的理论基础和现实意义。总结归纳了现有各类方法的特点，点明出了研究方向。
2. 针对极端狭窄非结构化道路的通行和泊车运动规划问题。提出了一种变精度混合 A* 算法，通过动态调整搜索步长和解析拓展间隔，提升了算法在静态狭窄空间下的求解效率。为进一步提升轨迹质量，提升运动过程中的稳定性和舒适性。本文将运动规划问题描述为最优控制问题。为了加快最优控制问题的求解速度，降低求解问题的复杂度，引入增量式矩形通行走廊，使得最优控制问题的复杂度独立于环境的复杂度，并将非线性非凸碰撞约束转换为简单的线性约束，极大地简化了原始最优控制问题。通过仿真实验，本文算法在泊车场景和狭窄空间通行场景下均能获得良好的表现，能够保证车辆运动过程中的舒适性和安全性。
3. 针对结构化道路下复杂动态交互环境下的行车运动规划问题。依托 OpenDrive 格式的高精地图，构建车道级邻接矩阵，并使用 dijkstra 算法实现了车道级全局路径规划。通过 Frenet 坐标系对车辆的横纵向运动解耦，构建 SL 和 ST 图，并在其中完成可行凸空间的开辟。并进一步将横纵向运动规划问题描述为考虑运动过程中的车辆性能限制和运动舒适性的分段平滑问题，由于该问题是一个被良好构造的二次规划问题，可以保证算法的求解实时性。经仿真验证，本文算法在多个场景下均可以规划出安全无碰撞的运动轨迹，保证求解实时性的同时兼具良好的舒适性和稳定性。
4. 针对行泊一体系统的轨迹跟踪需求。推导了常见的运动学和动力学轨迹跟踪模型，并设计了横向误差模型下的前馈 LQR 横向运动控制器和基于 PID 算法的纵向控制器，并通过实车验证了设计控制器的有效性。

6.2 未来展望

本文面向行泊一体自动驾驶系统的现实需求，针对复杂动态交互行车和极端狭窄通行两种典型工况，设计了基于通行走廊的运动规划算法和基于分段平滑的运动规划算法，受限于自身条件和研究时间。本文仍存在诸多不足之处，未来仍有必要在如下几个方面继续开展相关研究工作：

1. 本文算法的研究重点为车辆的运动规划算法，并在完美感知环境下开展相关研究测试，与实际大范围工程应用仍有一定差距。后续有必要考虑感知不确定的情况下开展相关运动规划研究工作。
2. 本文虽然在动态交互环境下实现了运动规划问题的实时求解，但是缺少对交通参与者运动过程中交互博弈的考虑，后续可以增加对复杂交互博弈和轨迹预测部分的考虑，展开预测博弈场景下的运动规划研究。
3. 本文虽然依靠通行走廊实现了静态狭窄空间下的运动规划问题的快速求解，但是通行走廊在一定程度上也减小了求解空间，使得极端复杂情况下难以求解。并且在规划过程中缺少对动态障碍物的考虑，后续可以修改通行走廊的构建方法，减少走廊构建所带来的解空间的损失，并加入对动态障碍物的考虑。

参考文献

- [1] 新华社. 全国机动车达 4.4 亿辆驾驶人达 5.32 亿人[EB/OL]. 2024. %7Bhttps://www.gov.cn/lianbo/bumen/202407/content_6961935.htm%7D.
- [2] 中国统计年鉴 2024[M]. 北京: 中国统计出版社, 2024.
- [3] 马艳丽. 驾驶员驾驶特性与道路交通安全对策研究[D]. 哈尔滨工业大学, 2007.
- [4] 百度 Apollo. 自动驾驶汽车安全白皮书[Z]. 出版地: 中国汽车技术研究中心, 2021.
- [5] 赵祥模, 承靖钧, 徐志刚, 等. 基于整车在环仿真的自动驾驶汽车室内快速测试平台[J]. 中国公路学报, 2019, 32(6): 124-136.
- [6] 章军辉, 陈大鹏, 李庆. 自动驾驶技术研究现状及发展趋势[J]. 科学技术与工程, 2020, 20(09): 3394-3403.
- [7] 徐志刚, 张宇琴, 王羽, 等. 我国自动驾驶汽车行业的发展现状及存在问题的探讨[J]. 汽车实用技术, 2019, 1: 13-21.
- [8] 腾讯新闻. 上半年乘用车 L2 级新车渗透率超 55%[EB/OL]. 2024. <https://news.qq.com/rain/a/20241018A02ZV200>.
- [9] 崔东, 祁志楠, 张慧. 紧急制动下的驾驶员运动姿态变化与碰撞损伤研究[J]. 汽车工程, 2020, 42(08): 1060-1065.
- [10] 于雷. 深度 | “大势所趋”的行泊一体, 窗口期还有多久? [EB/OL]. 2023. <https://news.qq.com/rain/a/20230218A038ZH00>.
- [11] 李晓华. 自动驾驶的发展现状、挑战与应对[J]. 人民论坛, 2023(18): 68-72.
- [12] 朱晓媛. 基于模型预测控制的智能车辆运动规划算法[D]. 浙江大学, 2021.
- [13] 李柏. 复杂约束下自动驾驶车辆运动规划的计算最优控制方法研究[D]. 浙江大学, 2018.
- [14] 张蓝天, 王光霞, 刘旭, 等. 机器人栅格地图编码与索引方法[J]. 测绘工程, 2022, 31(03): 23-30.
- [15] 杨涛. 基于 Kinect 辅助的服务机器人抓取路径规划研究[D]. 浙江大学, 2017.
- [16] Prim R C. Shortest connection networks and some generalizations[J]. The Bell System Technical Journal, 1957, 36(6): 1389-1401.
- [17] 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.
- [18] Likhachev M, Ferguson D, Gordon G, et al. Anytime Dynamic A*: An Anytime, Replanning Algorithm[C]//Fifteenth International Conference on International Conference on Automated Planning & Scheduling. 2005.

- [19] 朱冰, 贾士政, 赵健, 等. 自动驾驶车辆决策与规划研究综述[J]. 中国公路学报, 2024, 37(01): 215-240.
- [20] Tao J. Research on Modeling and Homing Control of Parafoil Systems in Complex Environments[J]. 2017.
- [21] Wu Y, Amato M, Trinkle J C, et al. An Obstacle-Based Probabilistic Roadmap Method For Path Planning[J]. Texas A & M University, 1996.
- [22] 温乃峰, 苏小红, 马培军, 等. 低空复杂环境下基于采样空间约减的无人机在线航迹规划算法[J]. 自动化学报, 2014, 40(7).
- [23] 姚陈鹏, 石文博, 刘成菊, 等. 移动机器人导航技术综述[J]. 中国科学: 信息科学, 2023(12): 2303-2324.
- [24] Jr J J K, Lavalle S M. RRT-Connect: An Efficient Approach to Single-Query Path Planning[C]//Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA. 2000.
- [25] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. The international journal of robotics research, 2011, 30(7): 846-894.
- [26] Webb D J, van den Berg J. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics[C]//2013 IEEE International Conference on Robotics and Automation. 2013: 5054-5061.
- [27] 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.
- [28] Wang J, Chi W, Li C, et al. Neural RRT*: Learning-based optimal path planning[J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(4): 1748-1758.
- [29] Karaman S, Walter M R, Perez A, et al. Anytime motion planning using the RRT[C] //2011 IEEE international conference on robotics and automation. 2011: 1478-1483.
- [30] Lian J, Cui C, Sun W, et al. KD-RRT: Restricted Random Testing based on K-Dimensional Tree[C]//2021 8th International Conference on Dependable Systems and Their Applications (DSA). 2021: 590-599.
- [31] 苏致远, 李永乐, 徐友春, 等. 具有时空约束的无人车集群构型变换方法[J]. 交通运输系统工程与信息, 2020, 20(2): 7.
- [32] 茅力非. 两轮自平衡移动机器人建模与控制研究[D]. 华中科技大学, 2013.
- [33] 王明强, 王震坡, 张雷. 基于碰撞风险评估的智能汽车局部路径规划方法研究[J]. 机械工程学报, 2021, 57(10): 14.

- [34] Werling M, Ziegler J, Kammel S, et al. Optimal trajectory generation for dynamic street scenarios in a frenet frame[C]//2010 IEEE international conference on robotics and automation. 2010: 987-993.
- [35] 王怿, 祝小平, 周洲. 一种基于 Clothoid 曲线的无人机路径规划算法[J]. 西北工业大学学报, 2012, 30(6): 5.
- [36] 许雅筑, 武辉, 游科友, 等. 强化学习方法在自主水下机器人控制任务中的应用[J]. 中国科学: 信息科学, 2020, 50(12): 1798-1816.
- [37] 龙慧, 朱定局, 田娟. 深度学习在智能机器人中的应用研究综述[J]. 计算机科学, 2018, 45(S2): 43-47+52.
- [38] Mariusz B. End to end learning for self-driving cars[J]. arXiv: 1604.07316, 2016.
- [39] Codevilla F, Müller M, López A, et al. End-to-end driving via conditional imitation learning[C]//2018 IEEE international conference on robotics and automation (ICRA). 2018: 4693-4700.
- [40] Wang Q, Chen L, Tian B, et al. End-to-end autonomous driving: An angle branched network approach[J]. IEEE Transactions on Vehicular Technology, 2019, 68(12): 11599-11610.
- [41] Wolf P, Hubschneider C, Weber M, et al. Learning how to drive in a real world simulation with deep q-networks[C]//2017 IEEE Intelligent Vehicles Symposium (IV). 2017: 244-250.
- [42] Alizadeh A, Moghadam M, Bicer Y, et al. Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment[C] //2019 IEEE intelligent transportation systems conference (ITSC). 2019: 1399-1404.
- [43] Ronecker M P, Zhu Y. Deep Q-network based decision making for autonomous driving [C]//2019 3rd international conference on robotics and automation sciences (ICRAS). 2019: 154-160.
- [44] Hu Y, Yang J, Chen L, et al. Planning-oriented autonomous driving[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 17853-17862.
- [45] 朱战霞, 靖飒, 仲剑飞, 等. 基于碰撞检测的空间冗余机械臂避障路径规划[J]. 西北工业大学学报, 2020, 38(1): 183-190.
- [46] Patten W N, Wu H C, Cai W. Perfect Parallel Parking Via Pontryagin's Principle [J]. Journal of Dynamic Systems, Measurement, and Control, 1994, 116(4): 723-728.
- [47] Todorov E, Li W. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems[C]//Proceedings of the 2005, American Control Conference, 2005. 2005: 300-306 vol. 1.

- [48] Chen J, Zhan W, Tomizuka M. Constrained iterative LQR for on-road autonomous driving motion planning[C]//2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). 2017: 1-7.
- [49] Howell T A, Jackson B E, Manchester Z. ALTRO: A Fast Solver for Constrained Trajectory Optimization[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2019: 7674-7679.
- [50] Ma J, Cheng Z, Zhang X, et al. Alternating Direction Method of Multipliers for Constrained Iterative LQR in Autonomous Driving[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(12): 23031-23042.
- [51] Zhang X, Liniger A, Borrelli F. Optimization-Based Collision Avoidance[J]. IEEE Transactions on Control Systems Technology, 2021, 29(3): 972-983.
- [52] He R, Zhou J, Jiang S, et al. TDR-OBCA: A Reliable Planner for Autonomous Driving in Free-Space Environment[C]//2021 American Control Conference (ACC). 2021: 2927-2934.
- [53] 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, 2021, 23(8): 11970-11981.
- [54] 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). 2020: 978-984.
- [55] Wang M, Zhang L, Zhang Z, et al. A hybrid trajectory planning strategy for intelligent vehicles in on-road dynamic scenarios[J]. IEEE Transactions on Vehicular Technology, 2022, 72(3): 2832-2847.
- [56] Li B, Ouyang Y, Li L, et al. Autonomous Driving on Curvy Roads Without Reliance on Frenet Frame: A Cartesian-Based Trajectory Planning Method[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(9): 15729-15741.
- [57] Han Z, Wu Y, Li T, et al. An Efficient Spatial-Temporal Trajectory Planner for Autonomous Vehicles in Unstructured Environments[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(2): 1797-1814.
- [58] Farag W. Complex trajectory tracking using PID control for autonomous driving[J]. International Journal of Intelligent Transportation Systems Research, 2020, 18(2): 356-366.
- [59] Alouache A, Wu Q. Genetic Algorithms for Trajectory Tracking of Mobile Robot Based on PID Controller[C]//2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP). 2018: 237-241.

- [60] 田涛涛, 侯忠生, 刘世达, 等. 基于无模型自适应控制的无人驾驶汽车横向控制方法[J]. 自动化学报, 2017, 43: 1931.
- [61] 段建民, 马学峥, 柳新. 基于 MFAPC 的无人驾驶汽车路径跟踪方法[J]. 计算机工程, 2019, 45(06): 6-11+20.
- [62] Liu S, Lin G, Ji H, et al. A Novel Enhanced Data-Driven Model-Free Adaptive Control Scheme for Path Tracking of Autonomous Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2025, 26(1): 579-590.
- [63] 贺伊琳, 宋若F, 马建. 基于强化学习 DDPG 的智能车辆轨迹跟踪控制[J]. 中国公路学报, 2021, 34(11): 335-348.
- [64] Liu M, Zhao F, Yin J, et al. Reinforcement-Tracking: An Effective Trajectory Tracking and Navigation Method for Autonomous Urban Driving[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(7): 6991-7007.
- [65] Zhao G, Chen Z, Liao W. Reinforcement-Tracking: An End-to-End Trajectory Tracking Method Based on Self-Attention Mechanism[J]. International Journal of Automotive Technology, 2024, 25(3).
- [66] Wang C, Wang L, Lu Z, et al. SRL-TR2: A Safe Reinforcement Learning Based TRajecotry TRacker Framework[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(6): 5765-5780.
- [67] Coulter C. Implementation of the Pure Pursuit Path Tracking Algorithm[C]//. 1992.
- [68] Thrun S, Montemerlo M, Dahlkamp H, et al. Stanley: The Robot that Won the DARPA Grand Challenge[J]. Journal of Field Robotics, 2006, 23(9): p.661-692.
- [69] Raffo G V, Gomes G K, Normey-Rico J E, et al. A Predictive Controller for Autonomous Vehicle Path Tracking[J]. IEEE Transactions on Intelligent Transportation Systems, 2009, 10(1): 92-102.
- [70] 倪兰青, 林F, 王凯正. 基于预瞄的智能车辆路径跟踪控制研究[J]. 重庆理工大学学报(自然科学), 2017, 31(03): 27-33.
- [71] 梁忠超, 王永富, 张欢, 等. 基于自适应 MPC 的无人驾驶车辆轨迹跟踪控制[J]. 东北大学学报(自然科学版), 2020, 41(6), 835: 835-840.
- [72] 陈亮, 秦兆博, 孔伟伟, 等. 基于最优前轮侧偏力的智能汽车 LQR 横向控制[J]. 清华大学学报(自然科学版), 2021, 61(09): 906-912.
- [73] 张雷, 赵宪华, 王震坡. 四轮轮毂电机独立驱动电动汽车轨迹跟踪与横摆稳定性协调控制研究[J]. 汽车工程, 2020, 42(11): 1513-1521.
- [74] Luan Z, Zhang J, Zhao W, et al. Trajectory Tracking Control of Autonomous Vehicle With Random Network Delay[J]. IEEE Transactions on Vehicular Technology, 2020, 69(8): 8140-8150.

- [75] Cao J, Song C, Peng S, et al. Trajectory Tracking Control Algorithm for Autonomous Vehicle Considering Cornering Characteristics[J]. IEEE Access, 2020, 8: 59470-59484.
- [76] Chu D, Li H, Zhao C, et al. Trajectory Tracking of Autonomous Vehicle Based on Model Predictive Control With PID Feedback[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(2): 2239-2250.
- [77] Wang X, Sun W. Trajectory Tracking of Autonomous Vehicle: A Differential Flatness Approach With Disturbance-Observer-Based Control[J]. IEEE Transactions on Intelligent Vehicles, 2023, 8(2): 1368-1379.
- [78] Dolgov D, Thrun S, Montemerlo M, et al. Practical search techniques in path planning for autonomous driving[J]. Ann Arbor, 2008, 1001(48105): 18-80.
- [79] Wang J, Zhang T, Zhang Q, et al. Implicit swept volume sdf: Enabling continuous collision-free trajectory generation for arbitrary shapes[J]. ACM Transactions on Graphics (TOG), 2024, 43(4): 1-14.
- [80] 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.
- [81] 郑四发. Onsite 自动驾驶泊车测试环境[Z]. <https://www.onsite.com.cn/#/dist/park>.
- [82] OnSite 仿真测试环境. 虚实融合自动驾驶公共测试服务平台[EB/OL]. (2024-12-01 [2024-03-15]. <https://www.onsite.com.cn/dist/customPage?menuId=233>.

致谢

致谢的字体为楷体小四，也可改为宋体小四。

攻读学位期间获得与学位论文相关的科研成果

学术论文

- [1] He S, Du Y, Li J, et al. Second-based queue length estimation with fusing MMW and low penetration rate CAV trajectory data[J]. Transportmetrica B: Transport Dynamics, 2024, 12.
- [2] 贺宜, 杜宇豪, 邱志军, 等。基于层次迭代搜索的半挂牵引车运动规划方法 [J]。(机械工程学报接收)

专利

- [1] 一种基于层次迭代搜索的重载车辆运动规划方法 [P]. 湖北省: CN202410951242.2 (公开实审)

参加的科研项目与获奖情况

- [1] 国家自然科学基金项目: 重载车辆运行安全控制 (52322217)
- [2] 第二届 OnSite 自动驾驶算法挑战赛 (泊车专项赛) 冠军
- [3] 第二届 OnSite 自动驾驶算法挑战赛 (虚实融合实车挑战赛) 冠军
- [4] 第一届 OnSite 自动驾驶算法挑战赛 (汇入汇出专项赛) 第三名
- [5] 第一届 OnSite 自动驾驶算法挑战赛 (高速公路基本段专项赛) 第十名