



吉林大学

JILIN UNIVERSITY

本科生毕业论文（设计）

中文题目 基于多项式曲线优化的汽车轨迹规划
方法

英文题目 Vehicle trajectory method planning based
on polynomial curve optimization

学生姓名 王成烨 班级 152005 学号 18200624

学 号 18200624

学 院 汽车工程学院

专 业 车辆工程

指导教师 庄晔 职称 教授

合作导师 秦也辰 北京理工大学 职称 副教授

2024 年 5 月

吉林大学学士学位论文承诺书

本人郑重承诺：所呈交的学士学位毕业论文，是本人在指导教师的指导下，独立进行实验、设计、调研等工作基础上取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的作品成果。对本人实验或设计中做出重要贡献的个人或集体，均已在文中以明确的方式注明。本人完全意识到本承诺书的法律结果由本人承担。

承诺人：王成烨

2024年5月20日

吉林大学本科毕业论文（设计）版权使用授权书

本毕业论文（设计）作者同意学校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。本人授权吉林大学可以将本毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业论文（设计）。

保密☐，在____年解密后适用本授权书。

本论文属于

不保密☒.

（请在以上方框内打“√”）

作者签名：王成辉

指导教师签名：Fe 呀

日期：2024 年 5 月 27 日

日期：2024 年 5 月 27 日

摘要

伴随着计算机和互联网产业的快速发展，汽车的智能化已逐渐成为社会发展的重要趋势之一。轨迹规划算法作为汽车智能化技术的重要一环，是确保主动安全的关键因素，其质量直接影响了自动驾驶系统的成败。因此，研究轨迹规划算法对于实现安全高效的自动驾驶技术具有重要意义。然而复杂多变的交通场景对轨迹规划的平滑性、实时性以及规划成功率等提出了严苛的要求。为此，本研究在 **Frenet** 坐标系下，采用基于采样的路径规划算法进行路径规划，利用高次多项式曲线拟合平滑路径曲线。针对采样方法暴露出的实际缺陷，进而采用基于优化的轨迹规划算法进行改善。本研究设置了三种实际场景对上述算法进行了仿真分析。实验结果表明，基于优化的轨迹规划算法能有效改善规划性能，为未来自动驾驶轨迹规划技术的发展提供了有力支撑。

关键词:

无人驾驶；轨迹规划；高次多项式曲线；优化；采样

Abstract

With the rapid development of computer and Internet industry, the intelligent car has gradually become one of the important trends of social development. As an important part of intelligent vehicle technology, trajectory planning algorithm is a key factor to ensure active safety, and its quality directly affects the success or failure of autonomous driving system. Therefore, it is of great significance to study the trajectory planning algorithm for realizing safe and efficient autonomous driving technology. However, the complex and changeable traffic scenes put forward strict requirements for the smoothness, real-time performance and success rate of trajectory planning. Therefore, in this study, a sampling-based path planning algorithm is used for path planning in the Frenet coordinate system, and a higher degree polynomial curve is used to fit the smooth path curve. In view of the actual defects exposed by the sampling method, the trajectory planning algorithm based on optimization is used to improve it. In this study, three actual scenarios are set to simulate and analyze the above algorithms. The experimental results show that the trajectory planning algorithm based on optimization can effectively improve the planning performance, which provides strong support for the development of future autonomous driving trajectory planning technology.

Keyword:

Autonomous vehicle; Trajectory planning; Higher degree polynomial curve; Optimization; Sample

目录

摘要.....	I
Abstract.....	III
第一章 绪论.....	1
1.1 课题研究背景及意义.....	1
1.2 国内外研究现状.....	4
1.2.1 无人驾驶国内外研究现状.....	4
1.2.2 轨迹规划国内外研究现状.....	5
1.3 本文主要研究内容.....	8
第二章 基于多项式曲线的路径规划方法.....	11
2.1 Frenet 坐标系的介绍.....	11
2.2 Frenet 坐标系和笛卡尔坐标系的转换.....	12
2.2.1 笛卡尔坐标系参数转化为 Frenet 坐标系参数.....	12
2.2.2 Frenet 坐标系下参数转化为笛卡尔坐标系下参数.....	14
2.3 多项式曲线轨迹的描述方法.....	15
2.3.1 多项式曲线的一般形式.....	16
2.3.2 多项式曲线次数的确定.....	16
2.3.3 多项式曲线系数的确定.....	16
2.4 多项式曲线轨迹规划场景分析.....	18
2.4.1 定速巡航场景.....	18

2.4.2 变道场景.....	20
2.4.3 停车场景.....	23
2.5 本章小结	24
第三章 基于采样的多项式轨迹规划方法	25
3.1 基于采样的多项式轨迹规划的基本流程.....	25
3.2 定速巡航场景下的轨迹规划	27
3.2.1 定速巡航纵向轨迹采样	27
3.2.2 定速巡航横向轨迹采样	28
3.2.3 定速巡航循环采样.....	29
3.2.4 定速巡航横纵向轨迹合成及轨迹检测	30
3.3 变道场景下的轨迹规划	31
3.3.1 变道纵向轨迹采样.....	32
3.3.2 变道横向轨迹采样.....	32
3.3.3 变道循环采样.....	33
3.3.4 横纵向轨迹合成及轨迹检测	33
3.4 停车场景下的轨迹规划	34
3.4.1 停车纵向轨迹采样.....	35
3.4.2 停车循环采样.....	35
3.5 基于采样的轨迹规划的不足	36
3.6 本章小节	36

第四章 基于优化的多项式曲线规划方法	39
4.1 基于优化的多项式曲线轨迹规划基本流程	39
4.2 基于优化的多项式曲线一般形式	40
4.3 二次规划问题的目标函数	41
4.3.1 偏离目标运动程度的评价指标	41
4.3.2 舒适性评价指标	43
4.3.3 轨迹安全性评价指标	44
4.3.4 目标函数	45
4.4 二次规划问题的约束条件	46
4.4.1 初始点连续性约束	46
4.4.2 车辆动力学约束	46
4.4.4 约束矩阵的计算	47
4.5 二次规划问题的求解方法	47
4.5.1 等式约束二次规划问题	48
4.5.2 不等式约束二次规划问题	48
4.5.3 有效集法求解不等式约束二次规划问题算法研究	50
4.6 优化算法实际场景的应用	52
4.6.1 定速巡航场景	52
4.6.2 停车场景	53
4.7 本章小结	54

第五章 基于不同场景下的算法对比分析	55
5.1 基于场景下的参数设置	55
5.1.1 定速巡航场景	55
5.1.2 停车场景	56
5.1.3 变道场景	57
5.2 基于算法的对比分析	57
5.2.1 实时性分析	58
5.2.2 规划成功率及合理性分析	61
5.3 综合对比分析	63
5.4 问题分析	64
5.5 本章小结	65
第六章 工程与社会	67
第七章 总结与展望	69
7.1 全文研究总结	69
7.2 研究意义和价值	70
7.3 不足和展望	70
参考文献	73
致谢	77

第一章 绪论

本章重点从本研究方向的背景和意义、国内外无人驾驶领域的应用研究及国内外路径规划算法的研究、本文主要研究内容三部分论述。本章重点说明无人驾驶领域特别是轨迹规划模块对于社会的意义，总结了目前主流的无人驾驶落地成果和轨迹规划算法研究成果，最后介绍了本次研究的主要内容。

1.1 课题研究背景及意义

现如今，汽车已经成为日常生活出行必备的交通工具。2011 到 2021 年中国乘用车产量如图 1.1 所示。伴随着计算机及人工智能技术的发展，汽车产业逐步与其他行业相关联形成包括智能网联汽车、新能源汽车等众多新兴产业。

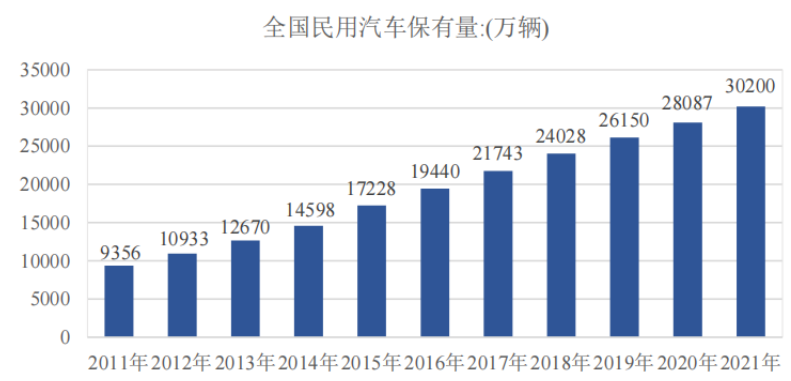


图 1.1 2011-2021 民用汽车保有量

汽车行业的蓬勃发展在方便日常生活的同时也带来了巨大的交通压力：据国家统计局数据显示^[1]2021 年我国全年汽车交通事故共发生 171941 起，同时人们花在交通工具上的时间越来越多将极大降低了人们对日常生活的体验。据相关部门统计，驾驶员主观因素是造成交通事故和交通拥堵的主要原因，其中疲劳驾驶、驾驶技术不熟练、醉酒驾驶等是主要影响因素。而自动驾驶技术被认为是解决上述问题的有效解决方案。近年来，无人驾驶汽车发展迅速，国家也相应出台了相关政策大力推动相关产业的落地。国务院中央在 2015 年发布的《中国制造 2025》指出：自动驾驶作为汽车产业未来转型升级的重要战略方向，在 2025 年实现 L4、L5 级别

的自动驾驶，逐步实现自动驾驶技术自主化和标准化^[52]。2020年，我国发布了《智能网联汽车技术路线图 2.0》，明确了以车路协同、车网协同及车车协同的横向技术线及智能出行、智能驾驶的纵向技术线，形成了“三纵两横”的技术路线^[2]。

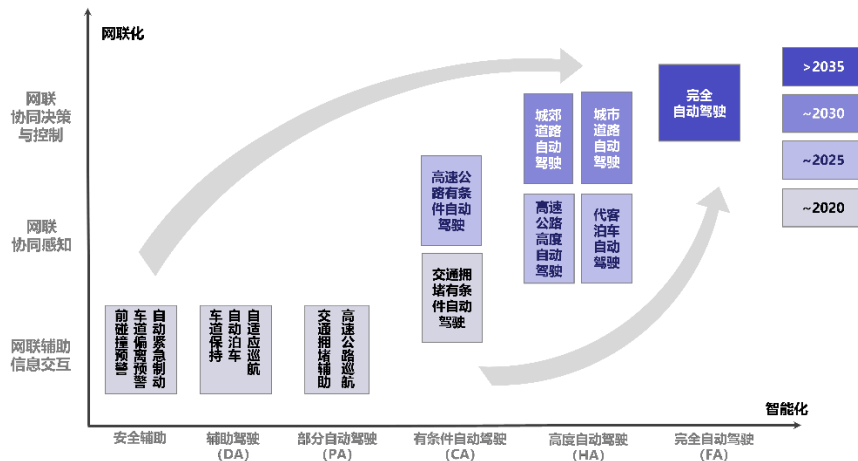


图 1.2 智能网联汽车技术路线图^[2]

目前，国内外涌现出大量无人驾驶产学研平台，公司平台以 Google、Tesla、小鹏汽车及百度 Apollo 为主，高校平台以卡内基梅隆大学、卡尔斯鲁厄理工学院、清华大学等为主。SAE International 在其 J3016 文件中概述了自动驾驶的五个级别^[3]：L0—L5,划分标准参见表 1.1。

表 1.1 自动驾驶分级

等级	定义	转向加减速控制	环境监测	紧急情况应对	应对工况
L0	人工驾驶	驾驶员	驾驶员	驾驶员	-
L1	辅助驾驶	驾驶员+系统	驾驶员	驾驶员	部分
L2	部分自动驾驶	系统	驾驶员	驾驶员	部分
L3	有条件自动驾驶	系统	系统	驾驶员	部分
L4	高度自动驾驶	系统	系统	系统	部分
L5	全自动驾驶	系统	系统	系统	全部

自动驾驶目前较常采用模块化分类，VIAC 项目^[4]、KIT 和戴姆勒等描述了无人驾驶的架构将无人驾驶核心框架分为感知，决策规划和控制^[5]，见图 1.3。感知模块^[6]负责估计系统状态，并使用车载传感器、摄像头、全球定位系统（GPS）、惯性测量单元（IMU）捕获的数据，和道路网络信息、交通信息及汽车动态信息创建一

个综合的环境表达；决策系统^[7]负责在行驶过程中根据感知模块的环境信息做出变道、超车等指令；规划模块^[8]是将自车规划一条从初始位置导航到用户定义的目标位置的路径轨迹并根据环境和车身的动态信息进行局部路径规划；控制模块^[7]控制层负责将汽车规划层输出的轨迹点序列或控制指令转化为具体的控制信号以控制汽车的加速、减速等实际操作。除了模块化分类以外，目前端到端的无人驾驶也是研究热点：通过神经网络等方法将输入的环境信息直接转化为汽车的驾驶行为从而更贴近人类驾驶员的驾驶行为。1988 年，DA Pomerleau^[9]等人就使用反向全连接神经网络使用仿真路面图进行训练，输入单目相机和雷达数据，能够输出方向使得车辆一直沿着道路行驶。2005 年，学术界已产生 end to end 概念，Lecun 等人^[10]实现了野外多种路况下的端到端自动驾驶。利用机器学习等方法，近年端到端无人驾驶功能场景逐步完善^[11]，有望成为实现无人驾驶的新途径。

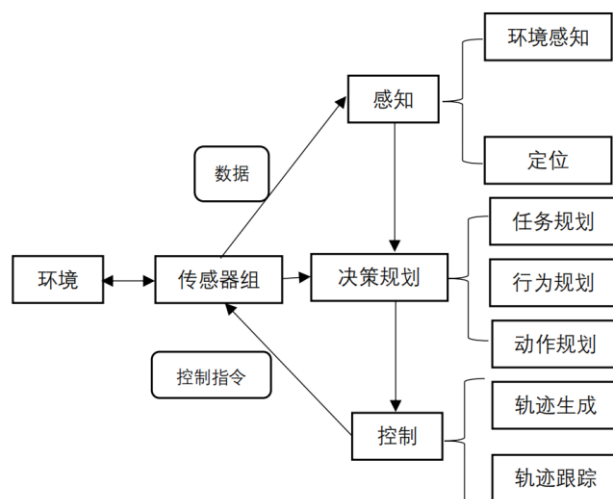


图 1.3 无人驾驶基本框架

相比于人工驾驶，无人驾驶具有下列优势^[12]：1.提高汽车行驶的安全性，避免人为造成的交通事故。2.解放生产力，降低人工成本。对于长距离行驶如客运或者货运，未来可广泛采用无人驾驶模式，这将有效降低昂贵的人工成本。3.缓解交通压力，提高通行效率。通过车联网、车路协同的手段可以及时获取路况信息，选择最优路径避免大面积的拥堵。4.提高舒适性。人类驾驶员的技术参差不齐，对路况缺少合理的判断，无人驾驶技术将致力于改善人类未来的出行体验，将汽车作为人类移动的娱乐场所。

在无人驾驶基本框架中，决策规划模块是汽车的“大脑”，他负责根据变化的环境做出合理的判断并生成安全舒适的轨迹。轨迹规划是无人驾驶汽车决策规划模块的核心内容，是串联感知层和控制层的重要组成，其目的是在获取来自决策层的指令以后生成符合车辆实际场景的路径并继续行驶。轨迹规划算法的质量和规划效率直接影响了自动驾驶系统的成败，它是确保主动安全的关键因素。因此，研究轨迹规划算法对于实现自动驾驶技术具有至关重要的意义。

1.2 国内外研究现状

1.2.1 无人驾驶国内外研究现状

自动驾驶技术首次提出是在 1939 年，通用汽车在世博会搭建的 Futurama 的模拟城市展厅,提出了自动高速公路(Automated Highway System , AHS)的概念。1994 年在美国普罗米修斯的项目中，VAMP 无人驾驶汽车行驶了 1600 公里，其中 95% 是自动驾驶的^[13]；不久 CMU NAVLAB 在无人技术领域取得了进展，并于 1995 年在美国进行了 5000 公里的驾驶，其中 98%是自动驾驶^[14]。2004 年，由美国国防高级研究计划局举办的首届 DARPA 挑战赛旨在鼓励研发无人驾驶汽车；随后几年，陆续有车辆完成 DARPA 挑战赛^[15]并证实了在城市道路实现完全无人驾驶的可能性^[16]。自 DARPA 挑战赛之后，众多无人驾驶测试迭出不穷如：Intelligent Vehicle Future Challenges^[17]（2009—2013）、Hyundai Autonomous Challenge^[18]（2010）、the VisLab Intercontinental Autonomous Challenge^[19]（2010）、the Public Road Urban Driverless Car Test^[20]（2013）、the autonomous drive of the Bertha-Benz historic route^[21]等。戴姆勒(Daimler)和卡尔斯鲁厄理工学院(Karlsruhe Institute of Technology, KIT)的实现，以自动驾驶的方式重现了历史上第一次越野汽车之旅^[22]，不仅如此，工业界也加快了无人驾驶系统的研发如谷歌无人驾驶^[23]和特斯拉无人驾驶测试^[24]等。2018 年，Google 公司的 Waymo 推出首款自动驾驶叫车服务，这标志着自动驾驶汽车正式上路，如图 1.4。在此之后，国内如百度 Apollo 等也相继实现了自动驾驶汽车的上路,如图 1.5。

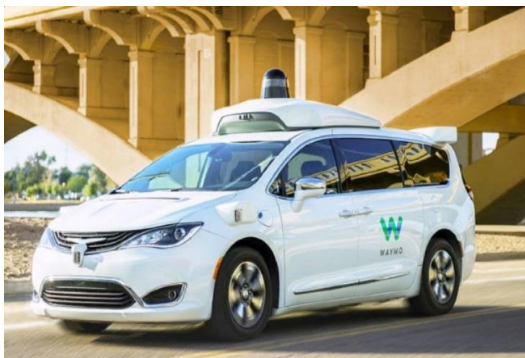


图 1.4 Waymo 出租车



图 1.5 Apollo 出租车

1.2.2 轨迹规划国内外研究现状

自动驾驶路径规划可分为全局路径规划和局部路径规划。在任务规划模块，汽车首先根据 GPS 等高精地图进行最优路径搜索，粗略获得一条到达终点的路线；在行为规划模块，汽车决策执行指令如定速巡航、变道、停车等；在动作规划模块，汽车根据当前环境，参考自身性能指标精确输出一条从当前时刻到达下一时刻的最优轨迹参考路线。

这些规划算法根据实现原理可大致分为图搜索、基于采样、基于曲线插值和数值优化四类^[25]。

1.图搜索算法

图搜索方法^[26]将车辆的构型空间离散为一个图，其中顶点表示车辆构型的有限集合，边表示顶点之间的过渡。通过在这样的图中执行搜索最小代价路径来找到所需的路径。算法包括 Dijkstra 算法和 A*算法及延伸。Dijkstra 算法是一种以路径代价为基础的搜索算法^[11]，Ben Franklin 车队和 VictorTango 车队^[27]中在参加 DARPA 城市挑战赛中实现。A*算法是在其基础上通过定义节点到达终点的代价启发式搜索趋向终点的图搜索算法。除此之外，在以上算法基础上改进的混合 A*算法、D*算法等可以更好地处理动态环境也获得了良好的实车效果^[28]。

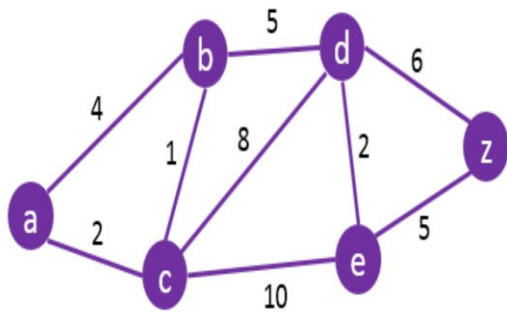


图 1.6 Dijkstra 算法规划

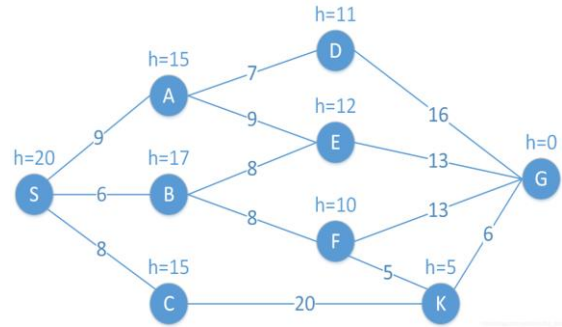
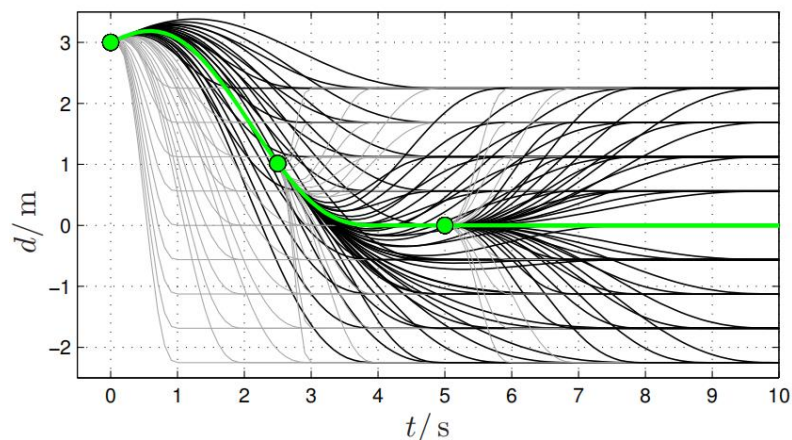


图 1.7 A*算法规划

2. 基于采样的路径规划算法

该算法允许在半结构化空间中快速规划^[29]，通过在导航区域中执行随机搜索。主要算法包括 RRT 快速搜索随机树算法、PRM 概率路线图算法等。RRT 算法通过在地图上随机撒点快速生成随机树路径，具有显著的效率。在 RRT 算法基础上，MIT 团队在 DARPA 城市挑战赛使用并获得了良好的效果，针对 RRT 算法路径非最优、曲率不连续的缺点，Karaman^[30]等提出了 RRT*算法进行了有效改进。RRT 算法通常用于全局路径规划，在局部路径中主要分为状态空间采样和控制空间采样两种。状态空间采样通过给定初始和目标状态，反向求解中间过程得到轨迹，2010 年 Werling M 等人提出了基于 Frenet 坐标系下的状态空间采样方法引起了广泛的关注^[51]如图 1.9 所示。控制空间采样则是固定输入的控制量和前向积分时间，由此前向积分得到一系列从初始位置到达目标位置的状态集合如状态栅格法，但该方法缺少目的性，无法精确采样到某一精准状态如图 1.10。


 图 1.9 状态空间采样^[51]

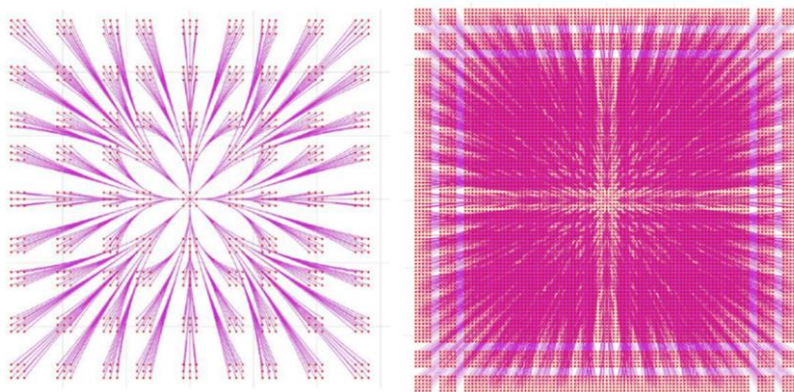


图 1.10 状态栅格算法

3. 基于曲线插值的路径规划算法^[31]

曲线插值法的核心思想就是基于预先选择的曲线类型，对汽车的前进方向的状态进行合理采样，采样点的信息包括自车的位置、速度、加速度等。使用车辆所需达到的状态作为边界条件（比如要求车辆到达某点的速度和加速度为期望值），将其代入曲线模型方程进行求解，获得曲线的相关系数见图 1.11。主要曲线包括：样条曲线、贝塞尔曲线及多项式曲线等。对于轨迹的曲线拟合，Kelly^[32]和 Nag^[33]在早年提出了利用高阶多项式来描述曲线曲率，高次多项式轨迹规划是一种兼具连续性和平滑性的规划方式，常见的高次多项式规划有三次多项式规划、五次多项式规划^[34]以及七次多项式规划。以三次多项式为例，这被认为是类车机器人的理想原始轨迹，高次多项式在转向机构中转矩是连续的，其产生的轨迹容易被车辆跟踪。五次多项式曲线则是在一段时间 T 内，使得函数 $\int_0^T f(\ddot{t})dt$ 最小的曲线。同时采用多项式轨迹还有平滑性好、可控性高等优点^[35]。

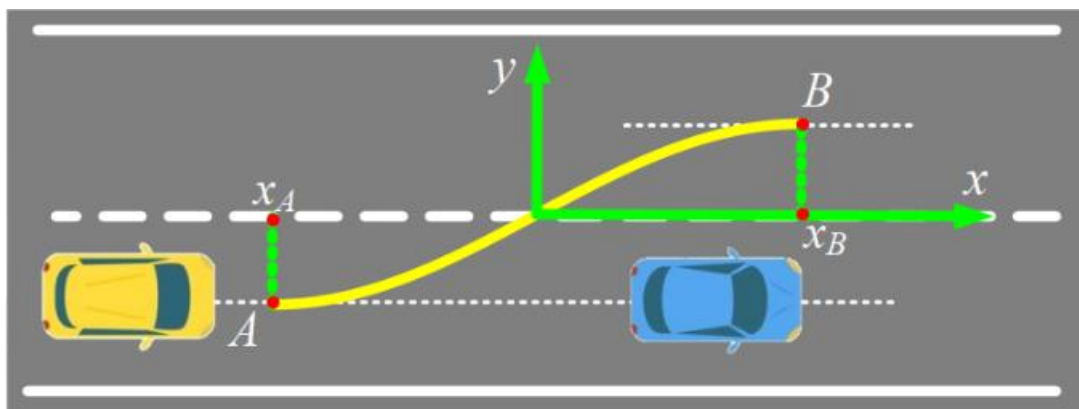


图 1.11 汽车变道超车场景的曲线插值拟合

4. 基于数值优化的路径规划算法

该算法旨在最小化受约束方程约束的目标函数，这一方法的原理在于将目标函数和约束条件转换为凸优化问题的形式，然后利用二次规划（Quadratic Program, QP）或其他数值优化迭代器来计算其最优解。Dolgov^[36]等人利用数值优化办法实现了已有轨迹的平滑处理，Xu 等人^[37]利用相似的思路在已有轨迹上对路径、速度迭代优化从而显著降低了规划时间并提升了规划质量；Ziegler 等人^[38]将运动约束转化为优化函数问题设计了最优的轨迹路线；MIT-Corne^[39]通过将多种参数如安全性、舒适性及未来行为的预测等作为函数优化的目标以解决赛事碰撞的问题。

上述几类路径规划算法中，图搜索算法在大型地图上的复杂度高、不适合应用于动态场景下的路径规划，采样的方法虽然一定程度提高了效率但是具有“局部最优”问题。基于曲线插值的方法通过对状态转移过程参数的插值可以获得平滑连续的轨迹曲线，但在目标状态附近需要进行大量采样后进行插值拟合。基于数值优化的算法则通过二次规划等优化方法迭代轨迹保证了实时性。

1.3 本文主要研究内容

轨迹规划方法正朝着混合算法发展，即发掘不同算法的优点并整合利用。目前，基于 Frenet 坐标系的高次多项式曲线轨迹规划方法获得了大量应用，其规划轨迹有平滑性好、可控性高和容易被控制系统追踪等优点。基于曲线插值的基本思想，本研究首先设计了基于采样的多项式曲线轨迹规划，随后通过将数值优化和曲线插值方法的优点相结合，针对基于采样的轨迹规划方法暴露出的不足采用数值优化的方法进行改善，最后设置场景对两种轨迹规划算法进行比较，分析了改善前后的进步和不足。

全文内容共六章，主要研究内容和章节安排如下：

第一章：绪论。本章重点介绍了本次研究的研究背景和研究现状，归纳了各种主流的轨迹规划算法及其特点，并最后概述了本次研究的研究内容。

第二章：基于多项式曲线的路径规划方法。本章重点研究了在结构化的道路应用广泛的 Frenet 坐标系，并推导了 Frenet 坐标系和笛卡尔坐标系的转化关系。分析

了基于高次多项式曲线拟合的路径规划算法的应用和流程。本节为轨迹曲线拟合确定了曲线类型并为后续的路径规划提供了理论指导。

第三章：基于采样的多项式轨迹规划方法。本章基于曲线插值的基本思想，在 Frenet 坐标系下，采用横纵向轨迹规划解耦的方法，分别针对特定场景下的横向和纵向目标状态进行采样插值多项式曲线，随后针对采样空间进行运动学约束检测和碰撞检测获得符合要求的轨迹组成备选轨迹，通过设定合理的评价函数选取备选轨迹中最优轨迹输出并执行。本章设计了一种基于采样的多项式曲线路径规划方法并实际测试，其实际暴露的弊端为后文设计方案解决提供了工作方向。

第四章：基于优化的多项式轨迹规划方法。本章针对前文基于采样的轨迹规划方法在实际应用中的缺陷，采用优化的方式加以改进。将冗杂的采样空间插值多项式转化为二次规划高效求解多项式曲线系数，以及将碰撞检测环节转化为优化问题中的目标函数减少耗时高的坐标变换从而提高实时性，将采样空间的有限解空间转化为优化问题的无限解空间来提高实际场景下的规划成功率。

第五章：基于不同场景下的算法对比分析。本节针对前文设计的两种轨迹规划算法，设置标准场景，对基于采样的轨迹规划算法耗时高、规划成功率低的原因进行了分析，并对比分析了相同场景下基于优化的轨迹规划算法对改善基于采样的轨迹规划算法中的缺陷的效果。最后综合分析了两种算法的优劣。

第六章：总结与展望。本章重点总结了本次研究的研究成果，并针对上述研究成果提出了研究的意义和价值，最后针对研究的不足进行了分析，并对接下来的完善工作进行了展望。

本文主要研究路线如图 1.12 所示。

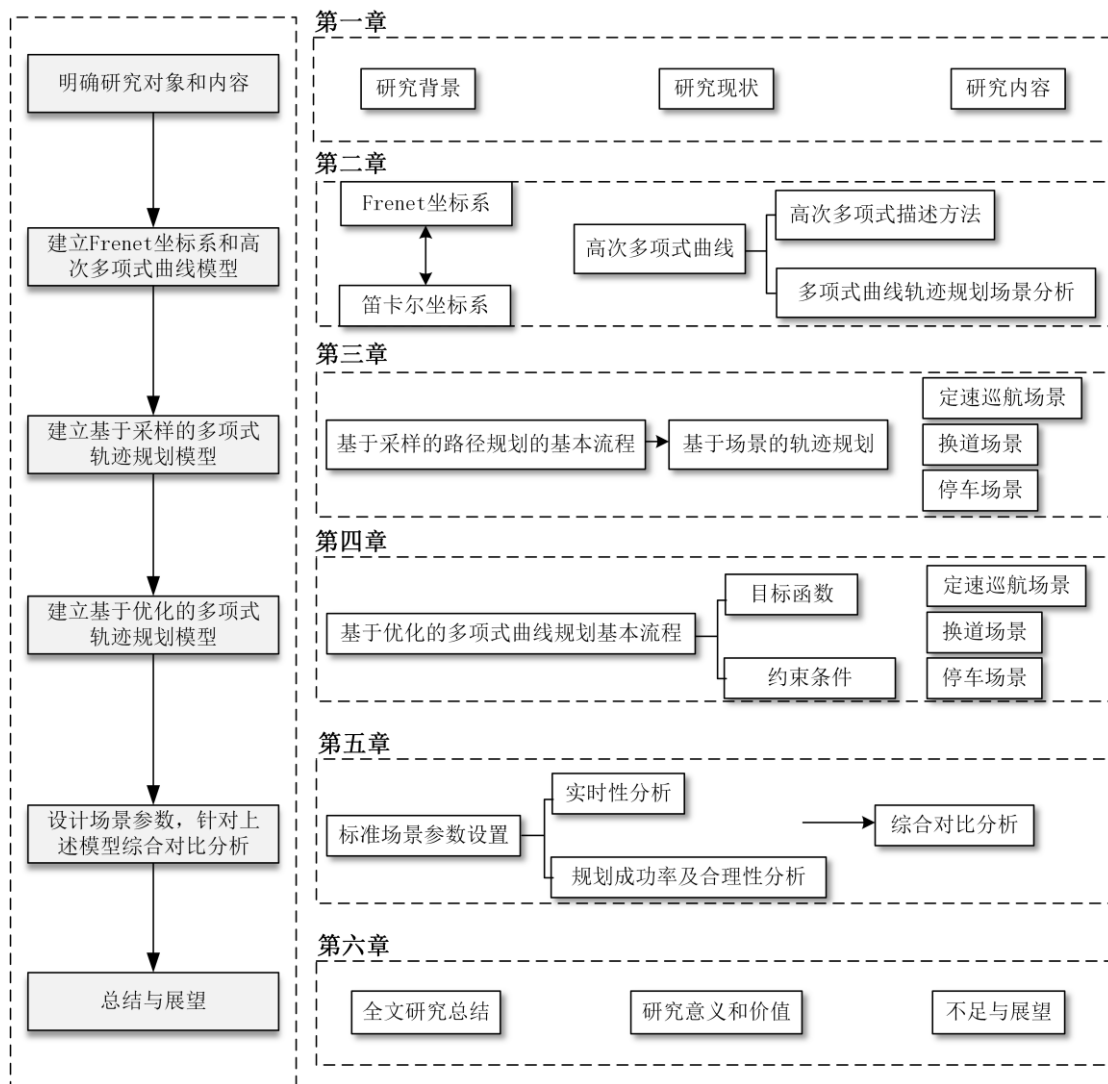


图 1.12 研究路线图

第二章 基于多项式曲线的路径规划方法

在轨迹规划中，运动坐标系和平滑轨迹曲线的选择是自车轨迹规划的第一步。针对轨迹规划的坐标系，本章重点研究在结构化的道路应用广泛的 Frenet 坐标系，并推导了 Frenet 坐标系和笛卡尔坐标系的转化关系。针对平滑轨迹曲线的拟合，本章提出了基于高次多项式曲线拟合的路径曲线规划算法的应用和流程。最后本章提出了定速巡航、变道和停车三种实际场景，针对这以上场景进行了多项式轨迹拟合算法的初步应用。

2.1 Frenet 坐标系的介绍

传统笛卡尔坐标系难以良好表达汽车与道路之间的相对关系^[40]，这一问题是在自动驾驶路径规划技术的难点。笛卡尔坐标系中的 $((x, y))$ 无法直接提供道路位置或车辆偏离道路中心线的具体信息，这些信息需要进行额外的计算和转换。然而，在自动驾驶系统中，对实时性的要求非常严格，因此对笛卡尔坐标的额外运算会影响系统的性能表现。同时，笛卡尔坐标系通常表达的是二维运动，这在实际规划场景问题中更复杂。在 DARPA (Defense Advanced Research Projects Agency) 汽车挑战赛期间，斯坦福大学提出的路径规划算法将横向偏移定义为相对于参考线的垂直距离^[41]，纵向偏移为该点在参考线上的投影点相对其上初始点的曲线长度，这一定义以道路中心线作为参考线，使道路与车辆之间的关系更直观。

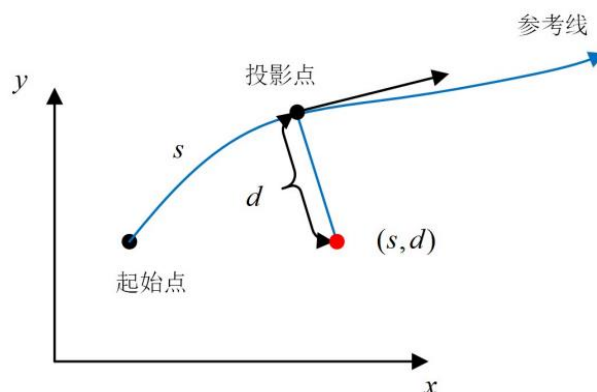


图 2.1 笛卡尔坐标系和 Frenet 坐标系之间的关系

如图 2.1，不同于笛卡尔坐标系，Frenet 坐标系使用纵向坐标(s)和横向坐标(d)来描述二维平面质点的位置。对于 Frenet 坐标，我们使用变量 s 和 d 来描述车辆在道路或参考路径上的位置，(d)表示质点到参考线上投影点的横向位移，而(s)则表示从参考线初始点到投影点的弧线长度。在自动驾驶系统中的轨迹规划问题中，通常将目标车道的中心线作为参考线。这样，Frenet 坐标系中的(s, d)可以直接提供车道位置以及车辆偏离车道中心线的距离等信息。这种方式减轻了轨迹规划模块和跟踪控制模块的计算负担，将笛卡尔坐标系下的运动的二维问题转化为横向和纵向两个方向运动的一维问题求解，使得求解效率更高，因此成为结构化道路轨迹规划中常用的坐标系。

2.2 Frenet 坐标系和笛卡尔坐标系的转换

车辆进行路径规划时，路径曲线是利用 Frenet 坐标系描述的，但是感知层接收环境信息以及针对规划轨迹的碰撞检测都是在笛卡尔坐标下进行的，在规划完成时轨迹的输出也是最终在笛卡尔坐标系下表达，因此需要笛卡尔坐标系和 Frenet 坐标系的相互转换。

2.2.1 笛卡尔坐标系参数转化为 Frenet 坐标系参数

如图 2.2 所示，车辆坐标在 Frenet 坐标系和笛卡尔坐标系中的表示有较大的差别，因此表 2.1 针对两种坐标系下的常用数学符号进行了规定和定义。

表 2.1 Frenet 坐标系和笛卡尔坐标系部分参数定义

符号	含义	符号	含义
s	自车在参考线上投影点相对原点的纵向坐标， Frenet 坐标系	x_x	x 坐标， 笛卡尔坐标系
\dot{s}	纵向坐标的一阶导	y_x	y 坐标，

	数，纵向速度		笛卡尔坐标系
\ddot{s}	纵向坐标的二阶导数，纵向加速度	θ_x	朝向角， 笛卡尔坐标系
d	自车相对于参考线投影点的横向距离， Frenet 坐标系下	v_x	线速度， 笛卡尔坐标系
\dot{d}	横向坐标的一阶导数，横向速度	a_x	加速度， 笛卡尔坐标系
\ddot{d}	横向坐标的二阶导数， 横向加速度	k_x	曲线的曲率
\vec{t}_r	参考线投影点的单位切向量	\vec{t}_x	轨迹上点的单位切向量
\vec{n}_r	参考线投影点的单位法向量	\vec{n}_x	轨迹上点的单位法向量
θ_r	参考线投影点的方位角	θ_x	轨迹上点的方位角
x_r	轨迹点在参考线上的投影 x 坐标	y_r	轨迹点在参考线上的投影 y 坐标

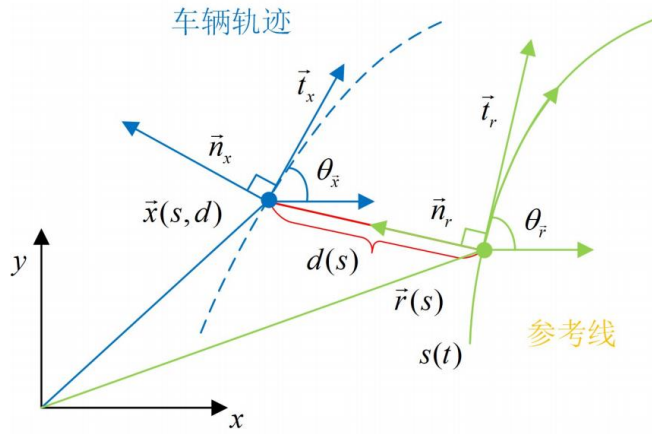


图 2.2 车辆坐标参数在 Frenet 坐标系和笛卡尔坐标系的表示

本研究重点计算推导 Frenet 坐标下 s 和 d 。

根据定义可知，投影点 (x_r, y_r) 对应的 s 就是点 (x_x, y_x) 在 Frenet 坐标系下对应的 s 。

由数学几何知识可得

$$\vec{x}(s, d) = \vec{r}(s) + d(s)\vec{n}_r \quad (2.1)$$

其中 $\vec{x}(s, d)$ 在笛卡尔坐标系下的坐标是 (x_x, y_x) ， $\vec{r}(s)$ 在笛卡尔坐标系下的坐标是 (x_r, y_r) ，由两点间距离公式可得

$$d = \begin{cases} +\sqrt{(x_r - x_x)^2 + (y_r - y_x)^2}, & \text{if } (y_x - y_r)\cos\theta_r - (x_x - x_r)\sin\theta_r > 0 \\ -\sqrt{(x_r - x_x)^2 + (y_r - y_x)^2}, & \text{if } (y_x - y_r)\cos\theta_r - (x_x - x_r)\sin\theta_r \leq 0 \end{cases} \quad (2.2)$$

本研究坐标变化重点应用与碰撞障碍物检测环节和最终轨迹输出环节，因此重点研究笛卡尔坐标系和 Frenet 坐标下的位置坐标变换，对二者速度和加速度的相关参数推导省略。

2.2.2 Frenet 坐标系下参数转化为笛卡尔坐标系下参数

碰撞检测模块的原理是将在 Frenet 坐标系规划得到的坐标转换为笛卡尔坐标系下并与障碍物的距离进行比较来判断是否安全，基于这一原理，本研究需推导 Frenet 坐标系下参数和笛卡尔坐标系下参数的转换关系。

设轨迹点的笛卡尔坐标是 (x_x, y_x) , 参考线上投影点的坐标是 (x_r, y_r) , 根据数学几何关系可推导如下。

$$x_x = x_r - d \sin \theta_r \quad (2.3)$$

$$y_x = y_r + d \cos \theta_r \quad (2.4)$$

2.3 多项式曲线轨迹的描述方法

多项式插值拟合是在全局路径轨迹完成的基础上进行的。在全局路径生成的插值点上, 这些曲线用于满足它们插值点所需的约束如拟合位置, 角度和曲率约束等, 开始段和结束段的期望值或约束将决定曲线的系数^[41]。高次多项式在转向机构中转矩是连续的, 其产生的轨迹容易被车辆跟踪。同时采用多项式轨迹还有平滑性好、可控性高等优点^[50]。

Glaser^[43]等人采用四次多项式拟合纵向约束条件、五次多项式拟合横向约束条件; 王沙晶等人^[44]在此基础上, 在 Frenet 坐标系下构建五次多项式分别建立自动驾驶车辆的横、纵向轨迹规划模型, 针对多场景建立横向轨迹质量评估函数和纵向轨迹质量评估函数, 最终将横纵向最优轨迹合成形成局部最优轨迹。针对智能车辆变道轨迹中存在的侧向加速度过大或轨迹曲率不连续的问题, 李胜琴等人^[45]提出了一种五次多项式变道方法, 兼顾了变道的安全性和效率; Petrov 等人^[46]采用三次多项式形成了超车轨迹。多项式轨迹规划还广泛用于其他机器的行动轨迹: 马宇豪等人^[47]实现了六次多项式的机械臂轨迹规划, 姚梦飞等人^[48]利用多项式轨迹解决了机场航空器滑行轨迹的规划和预测。

基于上述研究的成果, 本次研究在 Frenet 坐标系下, 基于多项式曲线的轨迹规划方法将整个轨迹规划过程分解为纵向轨迹规划和横向轨迹规划两个独立的阶段。各自的规划完成后并逐帧合成横向和纵向轨迹, 从而得到完整的车辆轨迹。将纵向轨迹和横向轨迹解耦的优势在于将笛卡尔坐标系下的二维运动问题转化为横纵向的一维问题求解, 降低了轨迹规划的复杂性, 减少了规划所需的时间, 从而提高了轨迹规划的实时性。

2.3.1 多项式曲线的一般形式

高次多项式曲线基本形式如下。

$$f(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + \cdots + a_nt^n \quad (2.5)$$

本研究分别设计了横向和纵向轨迹，其两个方向上的速度、加速度和加速度变化率分别如下所示。

$$\begin{cases} f'(t) = a_1 + 2a_2t + 3a_3t^2 + \cdots + na_nt^{n-1} \\ f''(t) = 2a_2 + 6a_3t + \cdots + n(n-1)a_nt^{n-2} \\ f'''(t) = 6a_3 + \cdots + n(n-1)(n-2)a_nt^{n-3} \end{cases} \quad (2.6)$$

其中，高次多项式曲线的系数 a_n 和最高次数 n 的确定是多项式轨迹曲线设计的重点。求解方法是根据实际场景要求获得，具体方法将在下文详述。

2.3.2 多项式曲线次数的确定

多项式的次数由初始和终点处的条件决定。 n 次多项式有 $n+1$ 个系数，需要 $n+1$ 个方程求解，因此对于初始和终点状态为 $n+1$ 的轨迹规划问题的解应为 n 次多项式。例如，初始和终点处的已知条件为6个，那么多项式轨迹曲线的最高次数为5次。

2.3.3 多项式曲线系数的确定

基于多项式曲线的轨迹规划本质上是根据插值的原理确定，即基于预先构造的曲线类型，根据车辆已知和期望达到的状态，将此期望值作为边界条件代入曲线类型进行方程求解，获得曲线的相关系数。

本研究工作基于高次多项式插值方法的原理，对高次多项式如3次、4次和5次多项式进行了一般性的分析，并利用Python实现了插值算法的编写，为后文特定场景下的多项式轨迹规划奠定了理论基础。

（1）三次多项式（cubic spline）插值

三次多项式具有连续可导的性质，可以保证规划轨迹在加速度曲线上的连续性，被认为是类车机器人理想的原始轨迹。本研究重点基于三次多项式曲线模拟结

构化道路中车道中心线（参考线）的参考线拟合。设定插值坐标点 (x,y) 依次为 $(-2.5,0.7)$ 、 $(0.0,-6)$ 、 $(2.5,5)$ 、 $(5.0,6.5)$ 、 $(7.5,0.0)$ 、 $(3.0,3.0)$ 、 $(-1.0,-2.0)$ 。如图 2.3。

（2）四次多项式（quartic polynomials）插值

四次多项式插值用于边界条件个数为 5 个的场景，具有高次多项式平滑、连续的优点。本次插值点设定为三个，每相邻两个插值点的边界条件个数为 5 进行插值，起点的边界条件是函数值、一阶导、二阶导已知，终点处的边界条件是一阶导、二阶导已知。见图 2.4。

（3）五次多项式（quintic polynomials）插值

五次多项式插值用于边界函数为 6 个的场景，是多项式曲线轨迹规划最常使用的高次多项式类型。其始末端点的条件分别是函数值、一阶导和二阶导已知。对于自动驾驶路径规划场景来说，通常由起点和终点的位置、速度、加速度三个参数构成边界函数。在所有已知始末端点状态的曲线拟合中，五次多项式是能保证其在全过程中 jerk 最小的一种方法。五次多项式插值如图 2.5。

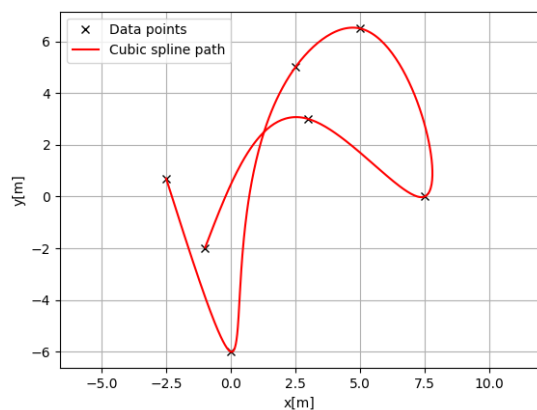


图 2.3 三次多项式曲线插值

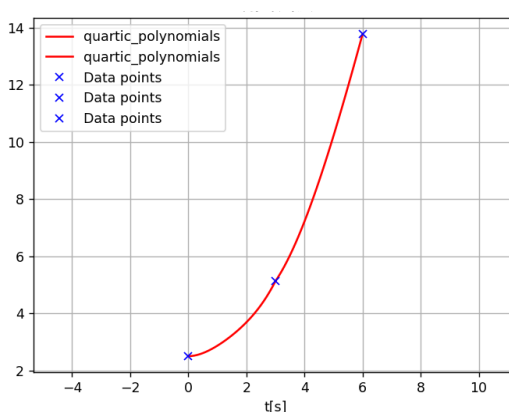


图 2.4 四次多项式曲线插值

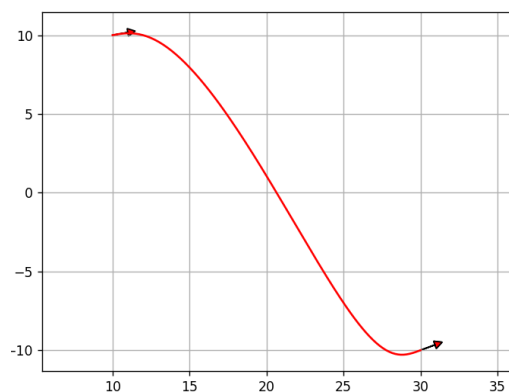


图 2.5 五次多项式曲线插值

2.4 多项式曲线轨迹规划场景分析

不同场景下的多项式轨迹的横纵向约束略有不同，因此需根据特定场景进行轨迹规划。本节重点选取三个场景：定速巡航、变道和停车，详细说明了求解特定场景下轨迹多项式的方法和流程。

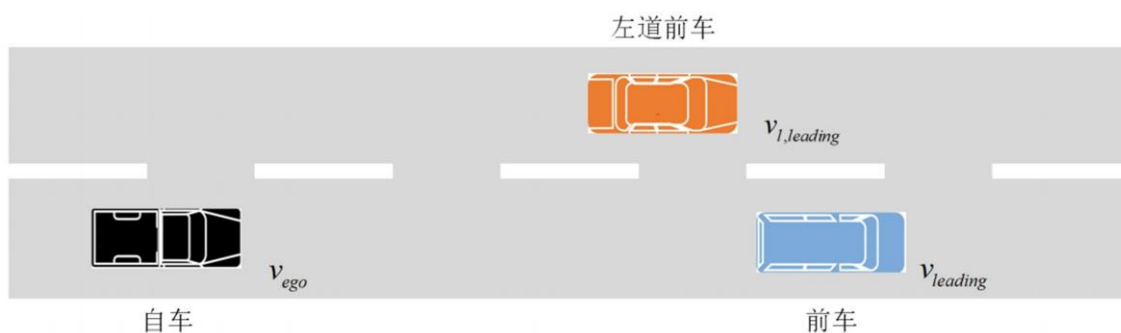


图 2.6 特定场景下的轨迹规划问题

2.4.1 定速巡航场景

定速巡航是一种汽车辅助驾驶技术，也称为巡航控制，这项技术允许驾驶员设定车辆在一定速度范围内保持恒定速度，而不需要持续踩油门，在众多高速路段，若上层决策判断周围环境良好，汽车将开启定速巡航模式，这将有效降低驾驶员的疲劳和紧张感。如图 2.7 所示，当自车所在车道前方车辆 $v_{leading}$ 大于或等于自车巡航车速时，自车不会与前车产生安全问题，因此定速巡航场景的规划目标是保持该车道并在规划终点时达到巡航速度 v_{cruise} 。下面进行分别对横纵向轨迹进行分析。

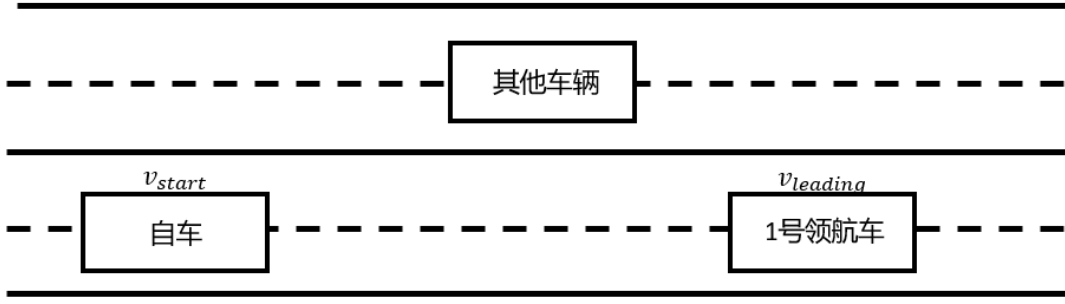


图 2.7 定速巡航场景

对于定速巡航纵向轨迹规划，为保证运动的连续性，规划起点状态需要和当前车辆的状态相同，即位置、速度和加速度相同；对于末端点的状态，应保证最终自车达到目标巡航车速并稳定，即 $v = v_{cruise}$ ，加速度为零。综上分析，起点和终点的已知条件为 5 个，纵向轨迹规划可以构建 4 次多项式曲线。

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \quad (2.7)$$

根据 $s(t)$ 曲线可以得到速度和加速度曲线

$$\begin{cases} \dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 \\ \ddot{s}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 \end{cases} \quad (2.8)$$

初始时刻为 $t = 0$ ，终点时刻 $t = T$ ，可得下述方程

$$\begin{cases} a_0 = s_0 \\ a_1 = \dot{s}_0 \\ 2a_2 = \ddot{s}_0 \\ a_1 + 2a_2 T + 3a_3 T^2 + 4a_4 T^3 = v_{cruise} \\ 2a_2 + 6a_3 T + 12a_4 T^2 = 0 \end{cases} \quad (2.9)$$

上式中， s_0 、 \dot{s}_0 、 \ddot{s}_0 分别表示自车的纵向位置、速度和加速度， T 代表一个规划周期的时长。将其转换为矩阵形式如下。

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 & 4T^3 \\ 0 & 0 & 2 & 6T & 12T^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} s_0 \\ \dot{s}_0 \\ \ddot{s}_0 \\ v_{cruise} \\ 0 \end{bmatrix} \quad (2.10)$$

令 $X_{cruise, long} = [a_0 \ a_1 \ a_2 \ a_3 \ a_4]^T$ ，上述矩阵可简写为

$$K_{cruise, long} X_{cruise, long} = C_{cruise, long} \quad (2.11)$$

可求解出纵向多项式轨迹的系数

$$X_{cruise,long} = K_{cruise,long}^{-1} C_{cruise,long} \quad (2.12)$$

对于定速巡航场景横向轨迹规划，为保证运动的连续性，规划起点状态需要和当前车辆的状态相同，即位置、速度和加速度相同；对于末端点的状态，自车应在该车道行驶，横向位置确定，由于汽车已经稳定，横向速度和加速度均为 0。综上分析，横向轨迹规划的始末端点约束条件为 6 个，可构建五次多项式曲线。

假设横向轨迹曲线多项式为

$$d(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \quad (2.13)$$

根据 $d(t)$ 曲线可以得到速度和加速度曲线

$$\begin{cases} \dot{d}(t) = b_1 + 2b_2 t + 3b_3 t^2 + 4b_4 t^3 + 5b_5 t^4 \\ \ddot{d}(t) = 2b_2 + 6b_3 t + 12b_4 t^2 + 20b_5 t^3 \end{cases} \quad (2.14)$$

初始时刻为 $t = 0$ ，终点时刻 $t = T$ ，可得下述方程

$$\begin{cases} b_0 = d_0 \\ b_1 = \dot{d}_0 \\ 2b_2 = \ddot{d}_0 \\ b_0 + b_1 T + b_2 T^2 + b_3 T^3 + b_4 T^4 + b_5 T^5 = 0 \\ b_1 + 2b_2 T + 3b_3 T^2 + 4b_4 T^3 + 5b_5 T^4 = 0 \\ 2b_2 + 6b_3 T + 12b_4 T^2 + 20b_5 T^3 = 0 \end{cases} \quad (2.15)$$

上式中， d_0 、 \dot{d}_0 、 \ddot{d}_0 为车辆目前的纵向位置、速度和加速度。 T 代表一个规划周期的时长。将其转换为矩阵形式如下。

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \begin{bmatrix} d_0 \\ \dot{d}_0 \\ \ddot{d}_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.16)$$

令 $X_{cruise,lat} = [b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5]^T$ ，上述矩阵可简写为。

$$K_{cruise,lat} X_{cruise,lat} = C_{cruise,lat} \quad (2.17)$$

可求解出横向多项式轨迹的系数。

$$X_{cruise,lat} = K_{cruise,lat}^{-1} C_{cruise,lat} \quad (2.18)$$

2.4.2 变道场景

当自车所在车道前车速度较低时，自车为实现更快的速度会进行变道行为。若变道后自车前方车辆行驶速度 $v_{leading}$ 低于自车定速巡航车速 v_{cruise} ，自车处于变道并加速的行为，根据国际交通法^[49]要求并与前车保持一定的安全距离行驶。该场景既可用于拥堵路段的自适应巡航，也可以用于作为提升交通效率的解决办法，相比定速巡航功能具有更加广阔的应用前景。在变道场景下，自车目标车速为 v_{target} ，横向运动位置为变道的车道中心位置。

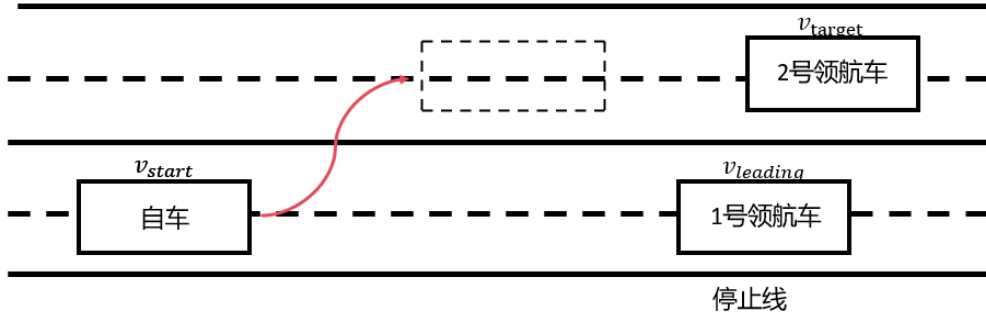


图 2.8 停车场景

对于变道的纵向轨迹规划，为保证轨迹规划的连续性，应保证规划起点出的状态和自车当前状态相同，即纵向位置、速度和加速度已知；终点处达到前车速度 v_{target} 并保持稳定，故加速度为 0。综上分析，始末端点已知条件为 5 个，可构建四次多项式轨迹。

纵向轨迹曲线多项式为

$$s(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 \quad (2.19)$$

根据 $s(t)$ 曲线可以得到速度和加速度曲线

$$\begin{cases} \dot{s}(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 \\ \ddot{s}(t) = 2a_2 + 6a_3t + 12a_4t^2 \end{cases} \quad (2.20)$$

初始时刻为 $t = 0$ ，终点时刻 $t = T$ ，可得下述方程

$$\begin{cases} a_0 = s_0 \\ a_1 = \dot{s}_0 \\ 2a_2 = \ddot{s}_0 \\ a_1 + 2a_2T + 3a_3T^2 + 4a_4T^3 = v_{l,leading} \\ 2a_2 + 6a_3T + 12a_4T^2 = 0 \end{cases} \quad (2.21)$$

将其转换为矩阵形式,其中 $v_{l,leading}$ 为车速。

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 & 4T^3 \\ 0 & 0 & 2 & 6T & 12T^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} s_0 \\ \dot{s}_0 \\ \ddot{s}_0 \\ v_{l,leading} \\ 0 \end{bmatrix} \quad (2.22)$$

令 $X_{LF,long} = [a_0 \ a_1 \ a_2 \ a_3 \ a_4]^T$, 上述矩阵可简写为

$$K_{LF,long} X_{LF,long} = C_{LF,long} \quad (2.23)$$

可求解出纵向多项式轨迹的系数

$$X_{LF,long} = K_{LF,long}^{-1} C_{LF,long} \quad (2.24)$$

对于变道横向轨迹规划，自车横向初始位置已知，横向速度和加速度初始状态已知；末端点的横向位置已知，横向速度为 0，横向加速度为 0。综上分析，始末端点已知条件为 6 个，可构建五次多项式轨迹。

横向轨迹曲线多项式为

$$d(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \quad (2.25)$$

根据 $d(t)$ 曲线可以得到速度和加速度曲线

$$\begin{cases} \dot{d}(t) = b_1 + 2b_2 t + 3b_3 t^2 + 4b_4 t^3 + 5b_5 t^4 \\ \ddot{d}(t) = 2b_2 + 6b_3 t + 12b_4 t^2 + 20b_5 t^3 \end{cases} \quad (2.26)$$

初始时刻为 $t = 0$ ，终点时刻 $t = T$ ，可得下述方程

$$\begin{cases} b_0 = d_0 \\ b_1 = \dot{d}_0 \\ 2b_2 = \ddot{d}_0 \\ b_0 + b_1 T + b_2 T^2 + b_3 T^3 + b_4 T^4 + b_5 T^5 = d_{target} \\ b_1 + 2b_2 T + 3b_3 T^2 + 4b_4 T^3 + 5b_5 T^4 = 0 \\ 2b_2 + 6b_3 T + 12b_4 T^2 + 20b_5 T^3 = 0 \end{cases} \quad (2.27)$$

将其转化为矩阵形式，其中 d_{target} 为自车横向运动的目标位置。

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \begin{bmatrix} d_0 \\ \dot{d}_0 \\ \ddot{d}_0 \\ d_{target} \\ 0 \\ 0 \end{bmatrix} \quad (2.28)$$

令 $X_{LF,lat} = [b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5]^T$

$$K_{LF,lat} X_{LF,lat} = C_{LF,lat} \quad (2.29)$$

可求解出横向多项式轨迹曲线系数

$$X_{LF,lat} = K_{LF,lat}^{-1} C_{LF,lat} \quad (2.30)$$

2.4.3 停车场景

停车场景要求自车能在前方指定位置处将车速降至零实现定距离停车。在该场景中，汽车纵向目标车速为 0，目标纵向位移距离为 s_{target} ，横向运动目标仍是沿该车道中心线行驶。

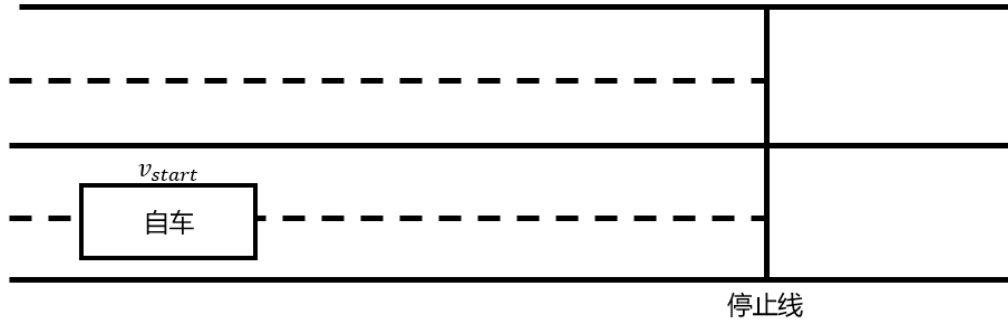


图 2.9 停车场景

与定速巡航场景不同的是该场景下的纵向轨迹规划需额外考虑位移参数，即多项式曲线的拟合边界条件为初始和目标状态的位置、速度和加速度共 6 个参数，因此该场景下的纵向轨迹曲线为五次多项式曲线。

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (2.31)$$

根据 $s(t)$ 曲线可以得到速度和加速度曲线

$$\begin{aligned} \dot{s}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ \ddot{s}(t) &= 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{aligned} \quad (2.32)$$

初始时刻为 $t = 0$ ，终点时刻 $t = T$ ，可得下述方程

$$\begin{cases} a_0 = s_0 \\ a_1 = \dot{s}_0 \\ 2a_2 = \ddot{s}_0 \\ a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 + a_5 T^5 = s_{target} \\ a_1 + 2a_2 T + 3a_3 T^2 + 4a_4 T^3 + 5a_5 T^4 = 0 \\ 2a_2 + 6a_3 T + 12a_4 T^2 + 20a_5 T^3 = 0 \end{cases} \quad (2.33)$$

将其转换为矩阵形式,其中 s_{target} 为刹车距离。

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} s_0 \\ \dot{s}_0 \\ \ddot{s}_0 \\ s_{target} \\ 0 \\ 0 \end{bmatrix} \quad (2.34)$$

令 $X_{break,long} = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5]^T$, 上述矩阵可简写为。

$$K_{break,long} X_{break,long} = C_{break,long} \quad (2.35)$$

可求解出纵向多项式轨迹的系数。

$$X_{break,long} = K_{break,long}^{-1} C_{break,long} \quad (2.36)$$

在停车场景中，横向轨迹的规划目标是保持车道中心线行驶，这和定速巡航横向轨迹的规划目标相同，因此可参照定速巡航场景的横向轨迹规划。

2.5 本章小结

本章重点提出了一种基于结构化路面的轨迹规划坐标：**Frenet** 坐标系并详细推导了该坐标系和笛卡尔坐标系之间的变换关系。随后提出了一种基于始末端点进行插值拟合的高次多项式的轨迹曲线描述方法，先后搭建了三次、四次和五次高次多项式插值算法，并分析了基于场景目标下的高次多项式轨迹次数的确定和系数的求解。最后，提出了本次研究的三个基本场景，并利用上述推导的算法原理实际应用于具体场景中。后续研究将基于上述研究成果，重点研究在特定场景中基于采样的多项式曲线路径规划算法。

第三章 基于采样的多项式轨迹规划方法

本研究分析了基于 Frenet 坐标的多项式曲线拟合在无人驾驶路径规划领域的重要应用，但是根据始末已知条件只规划一条路径是难以满足实际应用要求的。

Werling 等人^[51]提出了一种 semi-reactive 的轨迹生成办法，该方法将状态空间采样的思想和实际场景相结合。本章提出了一种基于采样的多项式轨迹规划方法：将横纵向轨迹规划问题解耦，针对实际场景需求在末端点目标状态附近采样从而生成由大量多项式曲线组成的采样空间，针对采样空间进行运动学约束检测和碰撞检测获得其中符合要求的轨迹组成备选轨迹，通过设定合理的评价函数选取备选轨迹中最优轨迹输出并执行。

3.1 基于采样的多项式轨迹规划的基本流程

为保证轨迹规划的舒适性和安全性，基于采样的轨迹规划方法需经过一系列处理得到最优路径。本节重点对基于采样的多项式轨迹规划的基本流程进行介绍。见图 3.1

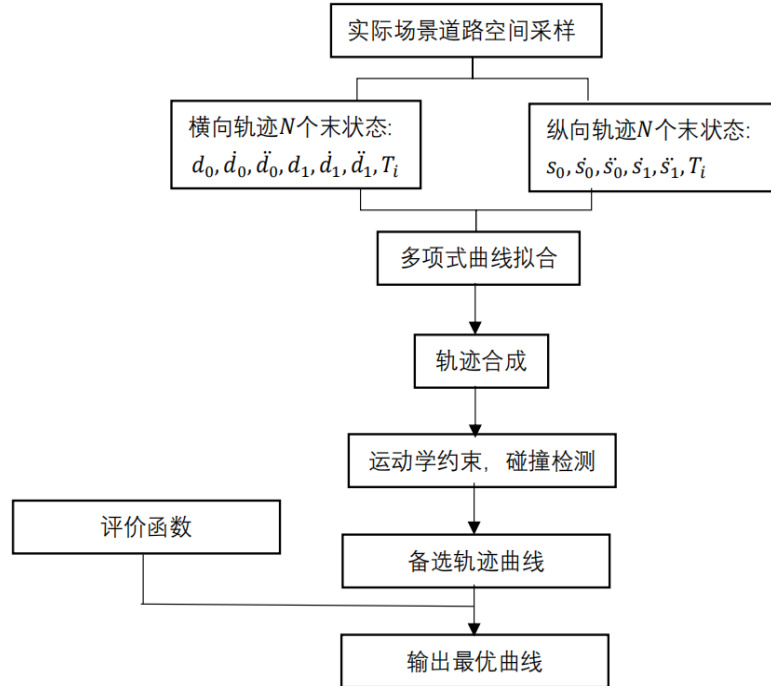


图 3.1 基于采样的多项式轨迹规划流程

（1）末端点状态采样

针对上述定速巡航场景，对于定速巡航纵向轨迹规划，需对末状态的速度和时间在一定范围内进行采样。定速巡航场景下，汽车纵向的目标速度 v_{cruise} 已知，在决策进行定速巡航之后，汽车需要加速或者减速至 v_{cruise} ，若汽车当前速度 v 和 v_{cruise} 相差过大时，难以在一个规划时长 T 中实现，因此应允许汽车在 T 内达到 v_{cruise} 上下范围内的速度，同时规划时长 T 的确定将影响汽车运动的舒适性，因此也应对规划时长 T 进行采样。对于定速巡航横向轨迹规划，需要对末端点的横向偏移 d_{target} 和规划时长在一定范围内进行采样。定速巡航场景下，汽车虽然在该车道行驶，但是应具有对于该车道附近障碍物的躲避能力，处于安全性的考虑应具有一定的横向偏移能力；同时规划时长 T 的确定将影响汽车横向运动的舒适性，因此也需要进行采样。

针对变道场景，对于变道纵向轨迹规划，和定速巡航场景相同，需对末状态的速度和时间在一定范围内进行采样，旁车道的目标速度 $v_{cruise2}$ 已知，汽车需要加速或者减速至 $v_{cruise2}$ ，这里的 $v_{cruise2}$ 是相比上述定速巡航 v_{cruise} 更高的速度，用以提高自车的通行效率。对于变道横向轨迹规划，需要对需要对末端点的横向偏移 d_{target} 和规划时长在一定范围内进行采样，横向偏移 d_{target} 为目标车道的位置。此时参考线选择变道后的车道。

针对停车场景，对于停车纵向轨迹规划，纵向目标速度 $v_{target} = 0m/s$ 是场景需求，因此不进行采样。刹车距离 s_{target} 将影响刹车场景下的舒适性和耗时性，因此可进行小范围采样以保证轨迹规划的合理性。同时，若在一个规划周期 T 中，自车无法在符合动力学约束的条件下减速至 $0m/s$ ，同时规划周期 T 的长短也会影响刹车过程中的舒适性，因此需要对规划周期 T 进行采样。针对刹车横向轨迹规划，同定速巡航场景相同，为保证对于该车道附近障碍物的躲避能力，处于安全性的考虑应具有一定的横向偏移能力，因此横向轨迹规划中要对末端点的横向偏移 d_{target} 进行采样。同时规划时长 T 的确定将影响汽车横向运动的舒适性，因此也需要进行采样。

（2）轨迹的合成

上述轨迹规划是在横纵向分别进行的，因此实际运动应为二者的合成方向来获得完整的轨迹。合成的方案是将在 Frenet 坐标系下规划的轨迹曲线进行坐标变化至笛卡尔坐标系下进行轨迹输出。

(3) 运动学约束筛选

实际场景下的汽车运动通常在动力学约束下进行，因此需要对不符合动力学约束的备选轨迹进行排除。常见的动力学约束有最大速度 v_{max} ，最大加速度 a_{max} ，最大曲率 c_{max} 等。轨迹规划时，需要根据实际的运动学约束对每个点进行检验。

(4) 碰撞检测

这一环节是用来检测备选轨迹中是否含有与障碍物周围相接处的情况，如检测到可能发生碰撞则将该轨迹排除。这一环节将有效保证无人驾驶轨迹的安全性。

(5) 轨迹质量评价

通过上述几个环节的检测，备选轨迹可能有多条，为确保选取最优轨迹，应设定评价函数进行评定，通常评价函数由舒适性、耗时性和对目标状态的偏离程度三部分构成。舒适性通常利用加速度的变化率 $jerk$ 评价， $jerk$ 越大意味着行驶过程越不平缓，舒适性越差；耗时性由汽车的轨迹规划时长 T 决定。实际场景下根据要求，需对上述三项进行加权处理，评价函数的选择是相对而言的，针对不同目标的要求评价函数不尽相同。评价函数的确定通常需要实车标定，本次研究仅做大致规定不做重点研究。

3.2 定速巡航场景下的轨迹规划

本节针对定速巡航场景，分别研究了基于采样的横纵方向的轨迹规划问题，首先分别研究了横纵向轨迹的采样方法，然后研究了横纵向曲线的评价函数，最后将横纵向曲线合并后设定循环次数、参考车道和障碍物实现了该场景的模拟。

3.2.1 定速巡航纵向轨迹采样

如上一章分析，该轨迹曲线采用四次多项式拟合，已知条件包括起点的位置、速度、加速度和终点的速度、加速度，并对规划时长 T 和巡航速度 v_{cruise} 进行采样。本例选取最小时间 $T_{min}=4s$ ，最大时间 $T_{max}=5s$ ，时间采样间隔为 $0.1s$ ，为，目标速度 $v_{target}=8m/s$ ，速度采样区间为 $[8 - 4 \times ds, 8 + 4 \times ds]$ ， $ds=3km/s$ 。初始速度（ c_speed ）、初始加速度（ c_accel ）和结果如图 3.2。

基于上述采样轨迹，进一步设定评价函数 $cost_s$ 模拟筛选最优轨迹见图 3.3。

$$cost_s = k_j \times Js + k_T \times Ti + k_d \times ds \quad (3.1)$$

$Js = \int_0^T s(\ddot{t})^2 \times dt$ ，为汽车的舒适性指标。

Ti 汽车的规划时间，为汽车耗时性指标。

$ds = (v_{target} - s(\ddot{T}))^2$ 汽车末状态和规划目标状态的差别。

k_j 、 k_T 、 k_d 为三者的权值。

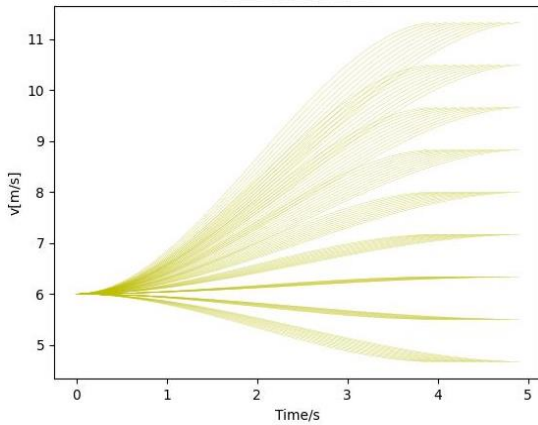


图 3.2 定速巡航纵向速度曲线采样 ($c_speed =$

$6m/s$ 、 $c_accel = 0m/s^2$)

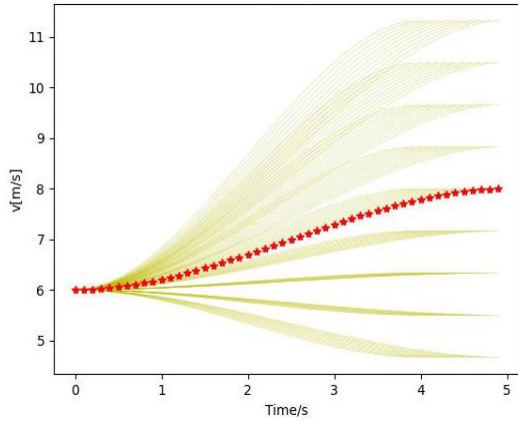


图 3.3 定速巡航纵向速度最优曲线 ($c_speed =$

$6m/s$ 、 $c_accel = 0m/s^2$)

该最优曲线最终收敛于 $s(\ddot{T}) = 8m/s = v_{target}$ ，同时为保证舒适性选取了尽量平滑的曲线导致耗时性相对增加。

3.2.2 定速巡航横向轨迹采样

如上一章分析，该轨迹曲线采用五次多项式拟合，已知条件包括起点和终点的位置、速度、加速度，并对规划时长 T 和横向位置 d 进行采样，本例选取本例选取最

小时间 $T_{min}=4s$ ，最大时间 $T_{max}=5s$ ，时间采样间隔为 $0.1s$ ，横向位置 d 在 $[0.6, +0.6]m$ 。初始横向速度（ c_d ）、初始横向加速度（ c_{d_d} ）和结果如图 3.4。

基于上述采样轨迹，进一步设定评价函数 $cost_d$ 模拟筛选最优轨迹见图 3.5。

$$cost_d = k_j \times Js + k_T \times Ti + k_d \times d1^2 \quad (3.2)$$

$Js = \int_0^T d''(t)^2 \times dt$ ，为汽车的舒适性指标。

Ti 汽车的规划时间，为汽车耗时性指标。

$d1 = d(T)$ 汽车末状态和规划目标状态的差别。

k_j 、 k_T 、 k_d 为三者的权值。

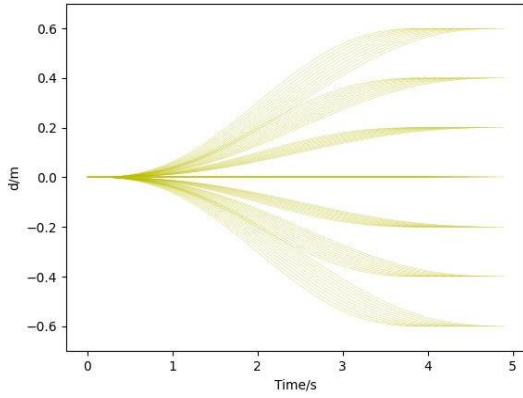


图 3.4 定速巡航横向位移曲线($c_d = 0m, c_{target}=0m$)

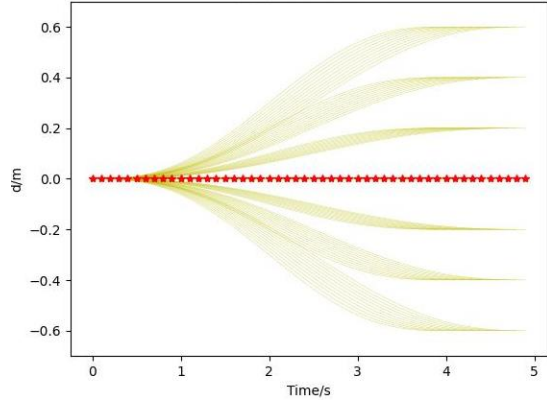


图 3.5 定速巡航横向位移最优曲线($c_d = 0m, c_{target}=0m$)

该曲线最终收敛于 $d=0m$ ，这表明在没有障碍物的情况下，汽车会沿该车道中央直线行驶。

3.2.3 定速巡航循环采样

实际场景下，为应对动态场景，车辆的路径规划轨迹并非在一个规划时长 T 结束后再进行下一个规划时长。通常车辆的轨迹规划路径是实时更新的，汽车轨迹规划时，会以 f 的频率更新状态，每个时间窗口 t 规划完成时获取该时刻最优的汽车状态作为下一时间窗口下的初始状态进行新的轨迹规划。见图 3.6

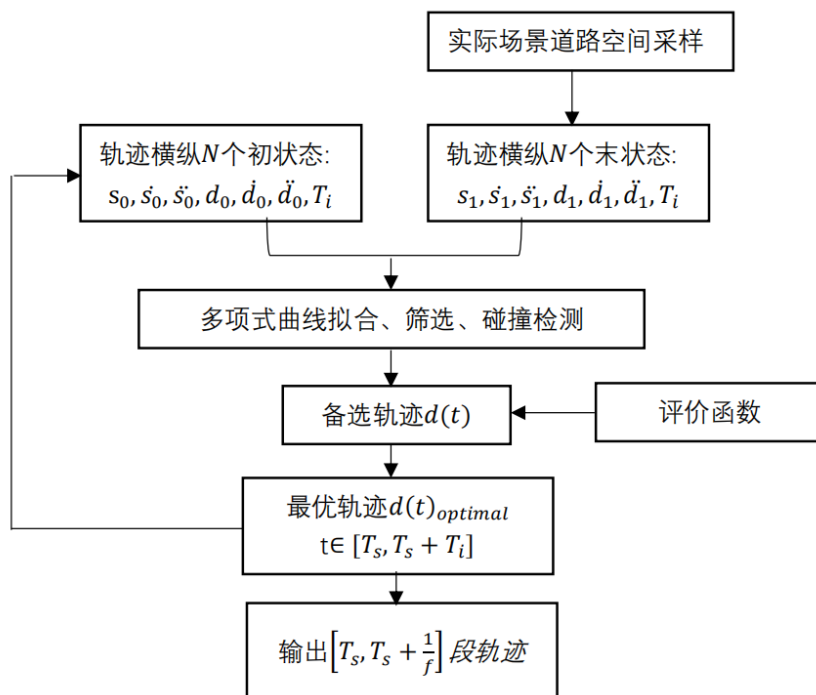


图 3.6 循环采样原理图

本节利用 $t = 1s$ 来模拟这个过程如图 3.7、图 3.8 所示

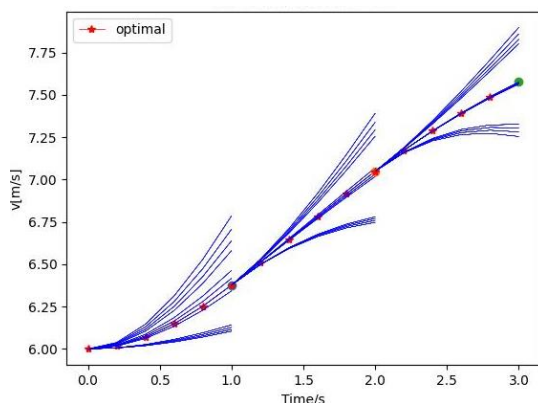


图 3.7 定速巡航纵向速度最优曲线循环采样

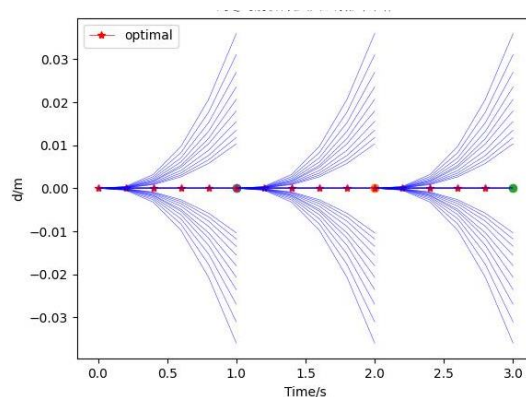


图 3.8 定速巡航横向位移最优曲线循环采样

3.2.4 定速巡航横纵向轨迹合成及轨迹检测

本次研究首先通过三次多项式模拟了一条参考路径，并随机定义了该参考道路的障碍物，根据 Frenet 坐标系和笛卡尔坐标系转换的关系求取每个规划点的全局坐标并绘制出车辆的规划轨迹，求取每条备用轨迹的速度、加速度和曲率，排除不符合运动学约束的曲线轨迹（本研究选择 $max_{speed} = 50km/s$ ， $max_{curvature} = 11/m$ ， $max_{accel} = 2m/s^2$ ）；定义车辆的半径 $R=2m$ ，若 $R^2 \leq l_{ob}^2$ ，则认为发生碰撞

并排除该轨迹。将横向轨迹评价函数 $cost_d$ 和纵向轨迹评价函数 $cost_s$ 相加作为轨迹的评价函数 $cost$ ，选取备选轨迹中 $cost$ 最小的作为下一个规划窗口的初始值并循环该过程，流程图见图 3.6。最终获得最优速度曲线见图 3.8。通过设置一条全局路径和障碍物，本研究展示了该算法在单车道上轨迹规划的实际应用能力，图 3.7 展示了在一定时间内的局部规划效果。

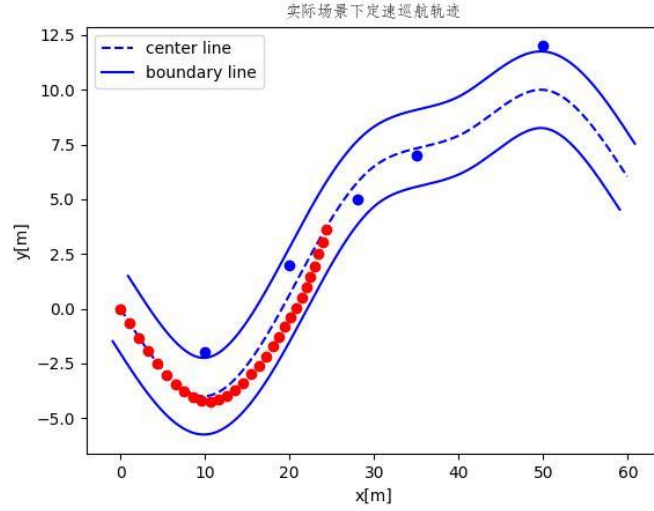


图 3.7 实际场景下定速巡航轨迹 ($c_speed = 6m/s, v_{target} = 8m/s$)

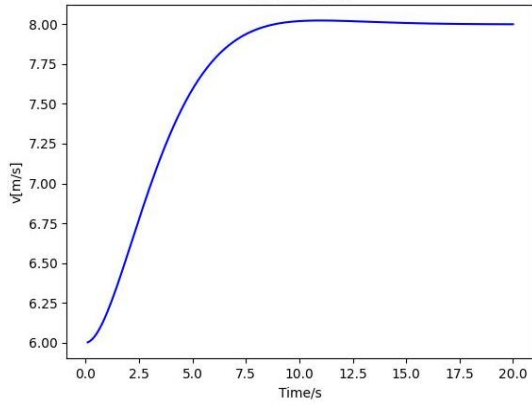


图 3.8 定速巡航速度曲线

($c_speed = 6m/s, v_{target} = 8m/s$)

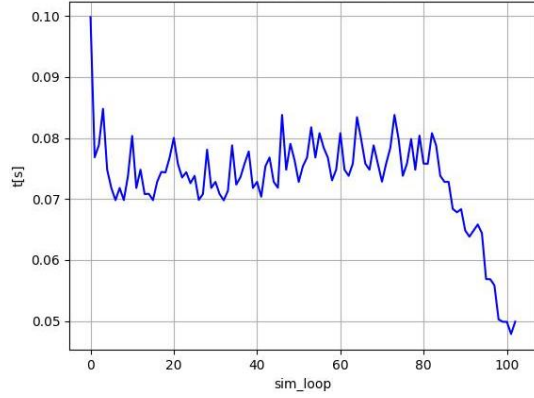


图 3.9 实际场景下定速巡航轨迹循环时间

($c_speed = 6m/s, v_{target} = 8m/s$)

通过仿真发现，该算法对于路上的障碍物有较好的躲避能力，同时轨迹会尽量贴合参考线路径，这保证了实际场景下的车辆安全性。

3.3 变道场景下的轨迹规划

本节针对变道场景，分别研究了基于采样的横纵方向的轨迹规划问题，首先分别研究了横纵向轨迹的采样方法，然后研究了横纵向曲线的评价函数，最后将横纵向曲线合并后设定循环次数、参考车道和障碍物实现了该场景的模拟。

3.3.1 变道纵向轨迹采样

该场景下纵向轨迹规划与定速巡航场景相似

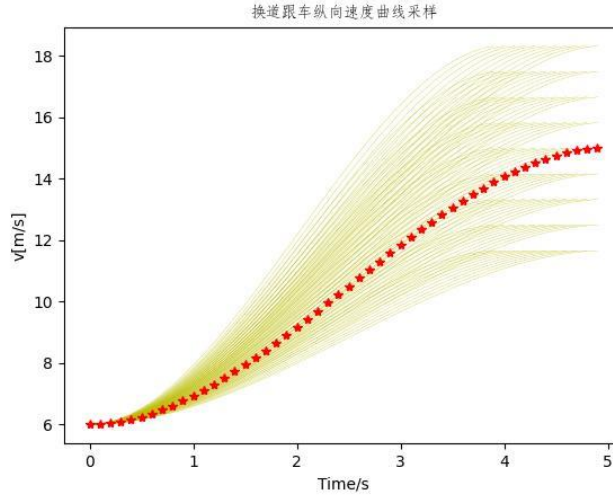


图 3.10 实际场景下变道纵向最优轨迹 ($c_{speed} = 6\text{m/s}$, $v_{target} = 15\text{m/s}$)

该结果表明，最优轨迹最终收敛于 $v = 15\text{m/s}$ ，同时采取较为缓和的加速形式保证自车行驶的舒适性。

3.3.2 变道横向轨迹采样

同定速巡航相同，该场景采用五次多项式，并对规划时长 T 和横向位置 d 进行采样。标准车道宽度为 3.5m ，本例选取本例选取最小时间 $T_{min}=4\text{s}$ ，最大时间 $T_{max}=5\text{s}$ ，时间采样间隔为 0.1s ，横向位置 d 在 $[-3, +3]\text{m}$ ，

评价函数和上述定速巡航场景相同：

$$cost_d = k_j \times Js + k_T \times Ti + k_d \times d^2 \quad (3.3)$$

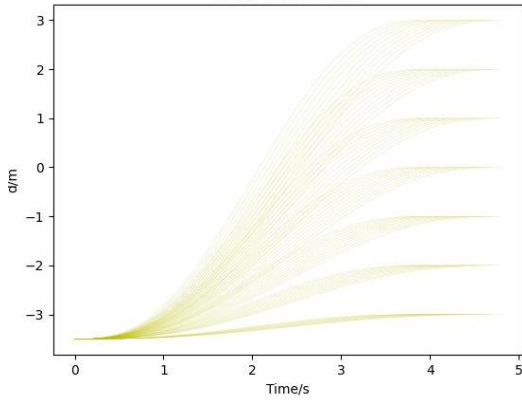


图 3.11 实际场景下变道横向轨迹采样

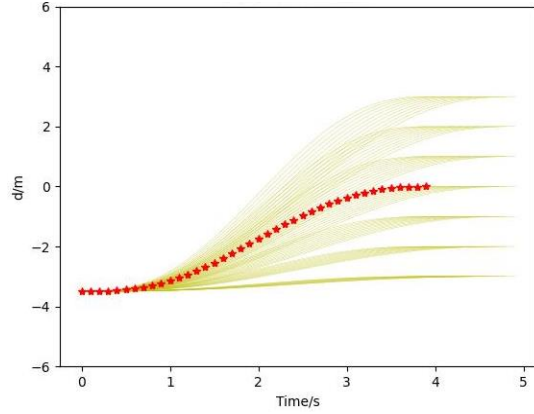


图 3.12 实际场景下变道横向最优轨迹

$$(c_d = -3.5m, c_{target} = 0m)$$

$$(c_d = -3.5m, c_{target} = 0m)$$

该场景下，最优轨迹为终点状态为 $d=0m$ ，表明通常情况下该轨迹会最终收敛到目标车道中心线。

3.3.3 变道循环采样

同定速巡航场景，本研究采用 $t=1s$ 模拟循环采样过程见图 3.13、图 3.14。

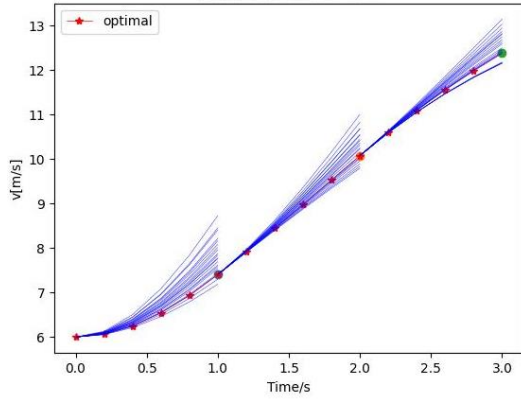


图 3.13 变道纵向速度循环采样

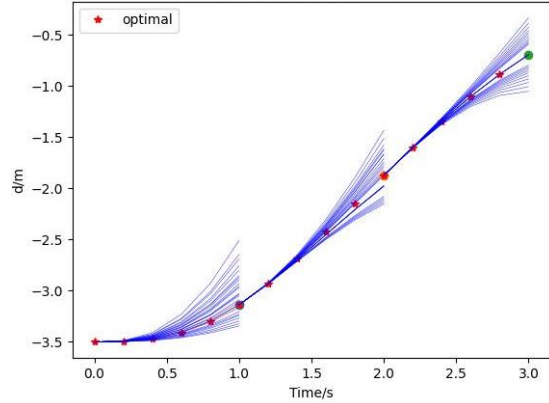


图 3.14 变道横向位置循环采样

3.3.4 横纵向轨迹合成及轨迹检测

本节，通过三次多项式曲线模拟两条临近车道，两条车道之间的距离为 $3.5m$ ，汽车从原车道切换至目标车道并实现目标速度，本研究选择 $max_{speed} = 80km/s$ ， $max_{curvature} = 11/m$ ， $max_{accel} = 3m/s^2$ 作为动力学约束，其余条件和

定速巡航场景相同。实际场景效果见图 3.15。

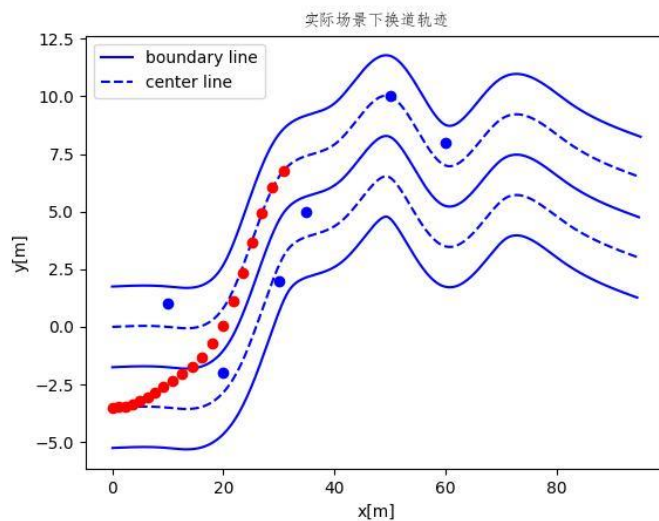


图 3.15. 实际场景下变道轨迹 ($c_speed = 6m/s, v_{target} = 13m/s$)

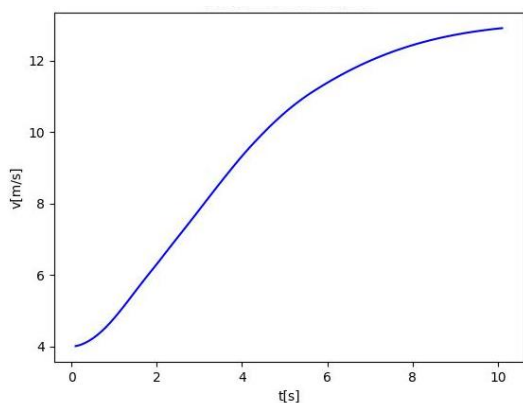


图 3.16 实际场景下变道速度曲线

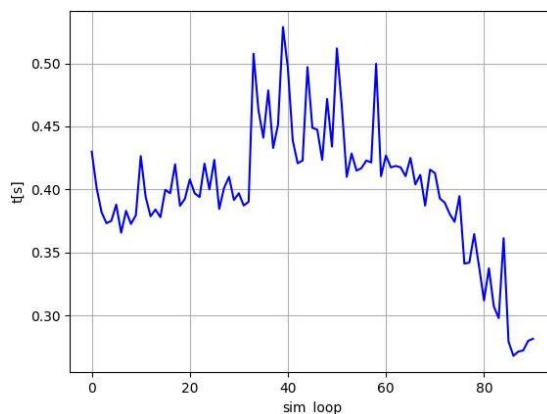


图 3.17 实际场景下变道轨迹循环时间

($c_speed = 6m/s, v_{target} = 13m/s$)

($c_speed = 6m/s, v_{target} = 13m/s$)

通过实验结果发现，该算法可以较为良好地模拟变道并提速的过程，但是从图 3.17 的循环时间可以看出，该采样的时间耗费较大，每次循环时间较长，这在实际应用中有较大的弊端。

3.4 停车场景下的轨迹规划

本研究下的停车场景规定如下：自车在预定车道以指定速度定速巡航行驶，在指定距离处实现减速至停止。本节针对该场景对于轨迹规划的要求，分别设定了横

向和纵向轨迹规划方法，并设定循环次数实现停车场景的路径规划。

3.4.1 停车纵向轨迹采样

纵向轨迹采样同前面两种场景有所不同。定速巡航场景和变道场景纵向轨迹规划通常采用四次多项式轨迹规划，即已知初始状态下的位置、速度及加速度，末端点状态的速度及加速度。停车场景要求自车在指定地点停车，因此已知末状态的位置，即需构建包含六个未知数的五次多项式轨迹。本研究对停车终点距离 $s_{target} = 20m$ ，并以 $ds = 2m$ 在 s_{target} 附近进行采样，求解结果如下，见图 3.18。停车横向轨迹采样同定速巡航场景，在此不再赘述。

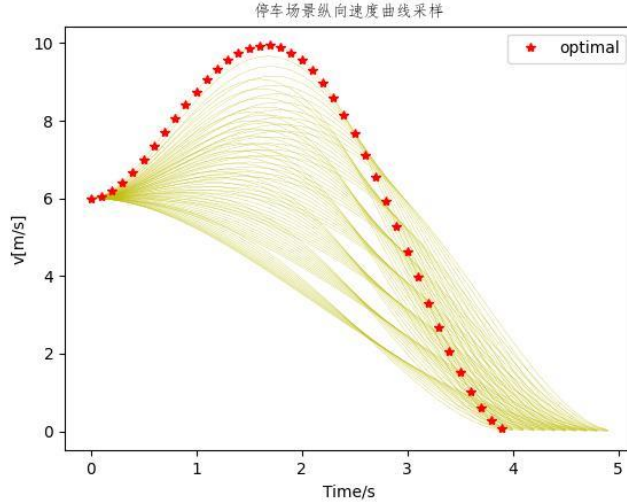


图 3.18 停车场景纵向速度曲线采样

3.4.2 停车循环采样

通过设置循环次数进行轨迹的迭代循环，并在迭代中设置循环终止条件，即车速降低至 $0m/s$ 附近。得到如图 3.19 所示效果

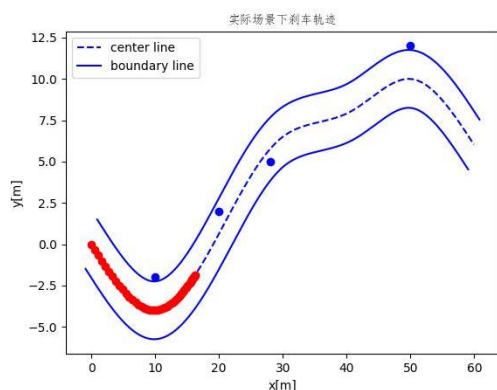


图 3.19 停车场景循环采样

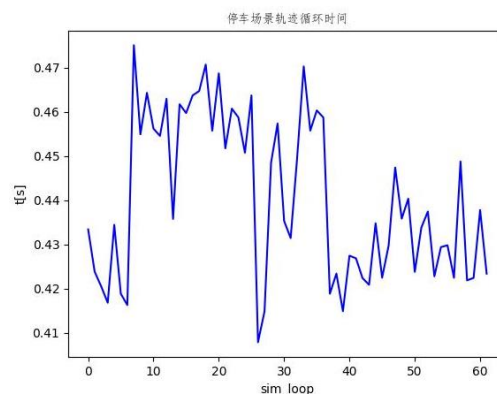


图 3.20 停车场景循环采样时间

通过上述算法测试发现，在停车场景下可以较为良好的实现在指定位置停车，但同样面临着耗时长、实时性差的缺点。

3.5 基于采样的轨迹规划的不足

实验证明，基于采样的多项式轨迹规划方法可以实现定速巡航、变道及停车等大多数场景的路径规划，但是该方法存在较为明显的缺陷：首先是实时性问题，通常允许的规划周期在 100ms ，通过上述研究发现基于采样的算法在时间上不符合要求。基于采样的路径规划往往需要大量的采样空间以保证规划路径的最优性，这极大增加了计算的负担，同时每条路径的碰撞检测都涉及大量的笛卡尔坐标和 Frenet 坐标间的变换，这一步骤耗时大。同时该方法面临容易规划失败的困境。基于采样的方法路径同类化明显、备选解的个数有限，在场景参数变化时这些解可能都会被排除导致规划失败，这在实际场景中是不被允许的。

3.6 本章小节

本章基于前面章节的研究工作进行了进一步拓展，将状态空间采样的思想引入到 Frenet 坐标系下的多项式曲线拟合模型中。本研究首先针对定速巡航、变道和停车场景，分别分析了场景目标并设置了采样空间。随后编写并整合了运动学约束模块和碰撞检测模块，对轨迹曲线的曲率、最大速度和最大加速度等动力学参数进行了约束；为自行车设置了避免障碍物碰撞的安全范围。最后本章提出了由舒适性指

标、耗时性指标及目标性指标组成的评价函数并设置循环算法实现最优规划轨迹的迭代更新。基于上述测试结果，本章分析了该方法在轨迹规划中的实时性不高、规划解集有限等弊端，后文将重点根据本章研究体现的弊端采用基于优化的方法进行改进和完善。

第四章 基于优化的多项式曲线规划方法

前文研究发现，基于采样的多项式曲线轨迹规划方法具有耗时高、规划成功率不高等缺点。为解决上述问题，本章提出了基于优化的多项式轨迹规划方法。本章设计了求解高效的二次规划解法，将上述场景下的轨迹规划问题转化为优化问题进行求解。该方法将采样中坐标变换频繁的碰撞检测环节转化为障碍物到车辆距离的目标函数，只需要在规划完成后进行坐标变换；提出了基于优化的方式求解多项式曲线系数获取轨迹曲线而非根据始末状态建立方程组，将基于采样的方法中有限的解转化为无限解集。

4.1 基于优化的多项式曲线轨迹规划基本流程

基于优化的多项式曲线轨迹规划本质是将求解特定场景下的多项式轨迹曲线系数转化为优化问题中求解在约束条件下的最小化目标函数的解。

本节重点对基于优化的多项式曲线轨迹规划流程进行了基本的分析并给出了该算法的基本框架。基本流程图见图 4.1

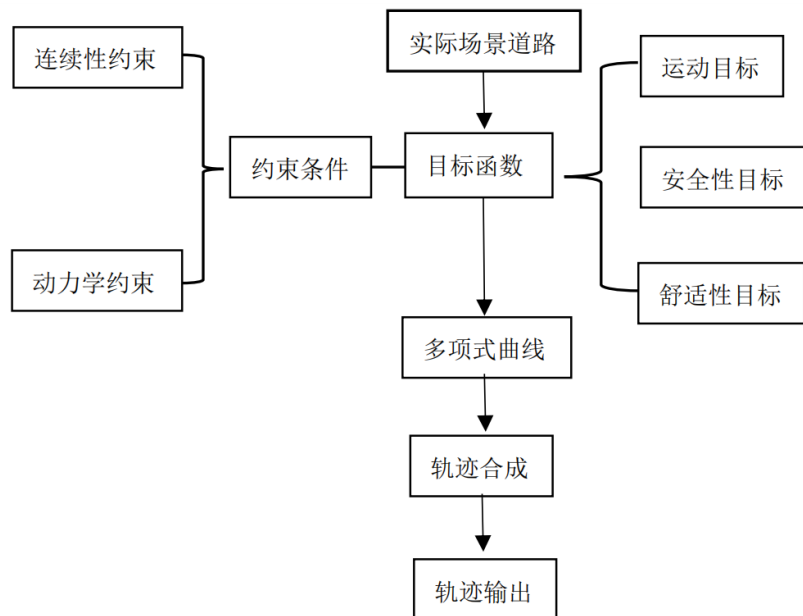


图 4.1 基于优化的轨迹规划算法流程图

(1) 设置场景目标函数

根据场景特定要求，设置和场景相关的目标函数。在定速巡航场景下，运动目标主要包括目标速度及横向目标位置、速度和加速度。在变道场景下，运动目标和定速巡航场景类似。在停车场景下，运动目标主要包括纵向目标速度和位置及横向目标位置、速度和加速度。除运动目标之外，本研究设置了躲避障碍物的安全性目标和乘坐体验相关的舒适性目标。

（2）设置约束条件

根据基本规划要求和自车性能等，设置了以初始点连续性约束为主的等式约束和运动学约束为主的不等式约束。

（3）求解多项式曲线

根据目标函数和约束条件组成的优化问题利用二次规划解法求解得到横向和纵向多项式曲线系数。

（4）轨迹合成和输出

将横纵向多项式轨迹合成并转换为笛卡尔坐标系下的坐标，输出最终规划的轨迹曲线。

4.2 基于优化的多项式曲线一般形式

基于采样的多项式曲线系数求解基于始末状态，本章则将轨迹规划曲线系数求解转化为优化问题，通过设置约束条件和目标函数，求出在约束条件下使目标函数最小化的系数作为多项式曲线的系数。本节将给出多项式曲线的一般形式，为后文二次优化问题求解奠定基础。

本节采用横纵向轨迹解耦，分别规定了纵向轨迹 $s(t)$ 和横向轨迹 $d(t)$ 的多项式形式，本节采用五次多项式，格式如下

$$s(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (4.1)$$

$$d(t) = b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4 + b_5t^5 \quad (4.2)$$

横向多项式曲线的速度曲线 $\dot{d}(t)$ ，加速度曲线 $\ddot{d}(t)$ ，加速度变化率曲线 $\dddot{d}(t)$ 如下，

$$\begin{cases} d(\dot{t}) = b_1 + 2b_2t + 3b_3t^2 + 4b_4t^3 + 5b_5t^4 \\ d(\ddot{t}) = 2b_2 + 6b_3t + 12b_4t^2 + 20b_5t^3 \\ d(\dddot{t}) = 6b_3 + 24b_4t + 60b_5t^2 \end{cases} \quad (4.3)$$

纵向多项式曲线的速度曲线 $s(\dot{t})$ ，加速度曲线 $s(\ddot{t})$ ，加速度变化率曲线 $s(\dddot{t})$ 同上。

为求解多项式参数的系数，本章将所有未知数设置为 x ，

$$x = [a_0, a_1, a_2, a_3, a_4, a_5, b_0, b_1, b_2, b_3, b_4, b_5]^T \quad (4.4)$$

求解优化问题常采用二次规划解决，其具有求解效率的优点，可以有效解决前文路径规划耗时长的缺点。二次规划的一般形式如下。

$$\min \left(\frac{1}{2} x^T Q x + q^T x \right) \quad (4.5)$$

$$s. t. Gx \leq h$$

$$Ax = b$$

上述 Q 是二次目标参数， q 是线性目标参数， G 是不等式约束矩阵， h 是不等式约束向量， A 是等式约束向量， b 是等式约束向量。

通过将 Frenet 坐标系下的轨迹规划问题转化为上述的二次规划问题，下文将详细分析该优化问题的目标函数，约束条件和求解方法。

4.3 二次规划问题的目标函数

该目标函数与前文质量评估函数类似，用来评价轨迹质量。二次规划问题中，通过不断优化迭代，从而求解出使目标函数最小化的解，从而求解出多项式轨迹。

本研究设计目标函数 J 由偏离目标运动的程度 J_{target} 、舒适性评价指标 $J_{comfort}$ 、防止碰撞的轨迹安全指标 J_{safety} 三部分组成。

$$J = J_{target} + J_{comfort} + J_{safety} \quad (4.6)$$

以上的代价值越小，说明该轨迹质量越高，通过求解该优化问题可以获得最优路径，下文将重点介绍三个评价指标的推导过程。

4.3.1 偏离目标运动程度的评价指标

目标运动是自车分析周围环境后做出的未来一段时间内轨迹规划的终点状态，本节将运动目标聚焦于纵向目标位置、纵向目标速度、横向目标位置和横向目标速度，以该指标来衡量上述目标的实现程度。

纵向目标位置为 s_{target} ，纵向目标速度为 v_{target} ，评价运动目标位置指标选择轨迹点纵向位置和纵向目标位置的差的平方和；纵向目标速度评价指标选择轨迹点纵向速度和纵向目标速度的差的平方和；横向目标位置和横向目标速度设为0，因此横向目标位置的评价指标为横向目标位置的平方，横向目标速度的评价指标为横向目标速度的平方，故与偏离目标运动程度的评价指标 J_{target} 可表示为

$$J_{target} = \sum_{i=0}^{i=N} \{w_{longPos}(s(t_i) - s_{target})^2 + w_{longSpeed}(\dot{s}(t_i) - v_{target})^2 + w_{latPos}d^2(t_i) + w_{latSpeed}\dot{d}^2(t_i)\} \quad (4.7)$$

其中 N 为轨迹点， $N = T/\Delta t$ ， T 为规划周期， Δt 为循环间隔。

1.纵向目标位置的评价指标计算

对第一项展开：

$$w_{longPos}(s(t_i) - s_{target})^2 = w_{longPos}(s^2(t_i) - 2s_{target}s(t_i) + s_{target}^2) \quad (4.8)$$

其中 $w_{longPos}s^2(t_i) = x^T Q_{longPos} x$

$$Q_{longPos,t_i} = w_{longPos} \begin{bmatrix} Q_{longPos,t_i}^{11} & 0 \\ 0 & 0 \end{bmatrix}_{12 \times 12} \quad (4.9)$$

$$Q_{longPos,t_i}^{11} = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 & t_i^6 \\ t_i^2 & t_i^3 & t_i^4 & t_i^5 & t_i^6 & t_i^7 \\ t_i^3 & t_i^4 & t_i^5 & t_i^6 & t_i^7 & t_i^8 \\ t_i^4 & t_i^5 & t_i^6 & t_i^7 & t_i^8 & t_i^9 \\ t_i^5 & t_i^6 & t_i^7 & t_i^8 & t_i^9 & t_i^{10} \end{bmatrix} \quad (4.10)$$

其中 $-2w_{longPos}s_{target}s(t_i) = -2w_{longPos}s_{target}(a_0 + a_1t_i + a_2t_i^2 + a_3t_i^3 + a_4t_i^4 + a_5t_i^5) = q_{longPos,t_i}^T x$

$$q_{longPos,t_i} = -2w_{longPos}s_{target} [q_{longPos,t_i}^{11} \quad 0]^T_{1 \times 12} \quad (4.11)$$

$$q_{longPos,t_i}^{11} = [1 \quad t_i \quad t_i^2 \quad t_i^3 \quad t_i^4 \quad t_i^5] \quad (4.12)$$

2.纵向目标速度的评价指标计算

$$w_{longSpeed}(\dot{s}(t_i) - v_{target})^2 = w_{longPos}(\dot{s}^2(t_i) - 2v_{target}\dot{s}(t_i) + v_{target}^2)$$

$$w_{longPos}\dot{s}^2(t_i) = x^T Q_{longSpeed,t_i} x \quad (4.13)$$

$$\text{其中 } Q_{longSpeed,t_i} = w_{longSpeed} \begin{bmatrix} Q_{longSpeed,t_i}^{11} & 0 \\ 0 & 0 \end{bmatrix}_{12 \times 12}$$

$$Q_{longSpeed,t_i}^{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 2t_i & 4t_i^2 & 6t_i^3 & 8t_i^4 & 10t_i^5 \\ 0 & 3t_i^2 & 6t_i^3 & 9t_i^4 & 12t_i^5 & 15t_i^6 \\ 0 & 4t_i^3 & 8t_i^4 & 12t_i^5 & 16t_i^6 & 20t_i^7 \\ 0 & 5t_i^4 & 10t_i^5 & 15t_i^6 & 20t_i^7 & 25t_i^8 \end{bmatrix} \quad (4.14)$$

$$-2w_{longSpeed}v_{target}\dot{s}(t_i) = q_{longSpeed,t_i}^T x \quad (4.15)$$

$$\text{其中 } q_{longSpeed,t_i} = -2w_{longSpeed}v_{target} [q_{longSpeed,t_i}^{11} \quad 0]^T_{1 \times 12}$$

$$q_{longSpeed,t_i}^{11} = [0 \quad 1 \quad 2t_i \quad 3t_i^2 \quad 4t_i^3 \quad 5t_i^4] \quad (4.16)$$

3. 横向目标位置的评价指标计算

$$w_{latPos}d^2(t_i) = x^T Q_{latPos,t_i} x \quad (4.17)$$

$$\text{其中 } Q_{latPos,t_i} = w_{latPos} \begin{bmatrix} 0 & 0 \\ 0 & Q_{longPos,t_i} \end{bmatrix}_{12 \times 12}$$

4. 横向目标速度的评价指标计算

$$w_{latSpeed}\dot{d}(t_i) = x^T Q_{latSpeed,t_i} x \quad (4.18)$$

$$\text{其中 } Q_{latSpeed,t_i} = w_{latSpeed} \begin{bmatrix} 0 & 0 \\ 0 & Q_{longSpeed,t_i} \end{bmatrix}_{12 \times 12}$$

$$\text{综上所述, } J_{target} = \sum_{i=0}^{i=N} \{x^T (Q_{longPos,t_i} + Q_{longSpeed,t_i} + Q_{latPos,t_i} + Q_{latSpeed,t_i})x + (q_{longPos,t_i}^T + q_{longSpeed,t_i}^T)x\}$$

4.3.2 舒适性评价指标

舒适性是自动驾驶轨迹规划的重要组成部分，它直接决定了乘客的体验和载货的安全性。本节采用和采样同样的评价方法，利用加速度的变化率（jerk）来量化舒适性。舒适性的评价指标采用横纵两项的 jerk 的平方和，公式如下

$$J_{comfort} = \sum_{i=0}^{i=N} (w_{longjerk}s(\ddot{t}_i)^2 + w_{latjerk}d(\ddot{t}_i)^2) \quad (4.19)$$

$$w_{longjerk}s(\ddot{t}_i)^2 = x^T Q_{longjerk,t_i} x \quad (4.20)$$

$$w_{latjerk}d(\ddot{t}_i)^2 = x^T Q_{latjerk,t_i} x \quad (4.21)$$

其中

$$Q_{longjerk,t_i} = w_{longjerk} \begin{bmatrix} Q_{longjerk,t_i}^{11} & 0 \\ 0 & 0 \end{bmatrix}_{12 \times 12} \quad (4.22)$$

$$Q_{latjerk,t_i} = w_{latjerk} \begin{bmatrix} 0 & 0 \\ 0 & Q_{longjerk,t_i}^{11} \end{bmatrix}_{12 \times 12} \quad (4.23)$$

$$Q_{longjerk,t_i}^{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 36 & 144t_i & 360t_i^2 \\ 0 & 0 & 0 & 144t_i & 576t_i^2 & 1440t_i^3 \\ 0 & 0 & 0 & 360t_i^2 & 1440t_i^3 & 3600t_i^4 \end{bmatrix} \quad (4.24)$$

$$\text{舒适性目标函数 } J_{target} = \sum_{i=0}^N \{x^T (Q_{longjerk,t_i} + Q_{latjerk,t_i}) x\}$$

4.3.3 轨迹安全性评价指标

安全性是智能汽车轨迹规划最重要的部分，也是无人驾驶汽车能实际应用并推广的必备条件。前文研究中基于采样的方法通常在每次迭代规划开始时对障碍物进行检测，频繁的坐标变换极大增加了运算负担，使规划实时性降低。本节将障碍物的横向和纵向距离转换为目标函数的一部分，从而最大化与障碍物之间的距离，这一改变将有效提高轨迹规划算法的实时性。

由于二次规划格式的限制，该目标函数通常求取最小值，因此对该部分目标函数格式进行改进，转而求解最小化平方和的相反数。假设 m 个障碍物，其在 Frenet 坐标系 t_i 时刻，第 j 个障碍物纵向和横向距离分别是 s_{ij} 和 d_{ij} ，因此评价指标可写为如下形式

$$J_{safety} = - \sum_{i=0}^N \sum_{j=0}^m \{w_{obsLongPos}(s(t_i) - s_{ij})^2 + w_{obsLatPos}(d(t_i) - d_{ij})^2\} \quad (4.25)$$

展开上式：

$$w_{obsLongPos}(s(t_i) - s_{ij})^2 = w_{obsLongPos}(s^2(t_i) - s_{ij}s(t_i) + s_{ij}^2) \quad (4.26)$$

$$w_{obsLatPos}(d(t_i) - d_{ij})^2 = w_{obsLatPos}(d^2(t_i) - d_{ij}d(t_i) + d_{ij}^2) \quad (4.27)$$

其中 $w_{obsLongPos}s^2(t_i) = x^T Q_{obsLongPos,t_i} x$, $w_{obsLatPos}d^2(t_i) = x^T Q_{obsLatPos,t_i} x$

其矩阵形式为

$$Q_{obsLongPos,t_i} = w_{obsLongPos} \begin{bmatrix} Q_{obsLongPos,t_i}^{11} & 0 \\ 0 & 0 \end{bmatrix}_{12 \times 12} \quad (4.28)$$

$$Q_{obsLatPos,t_i} = w_{obsLatPos} \begin{bmatrix} 0 & 0 \\ 0 & Q_{obsLongPos,t_i}^{11} \end{bmatrix}_{12 \times 12} \quad (4.29)$$

$$Q_{obsLongPos,t_i}^{11} = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 & t_i^6 \\ t_i^2 & t_i^3 & t_i^4 & t_i^5 & t_i^6 & t_i^7 \\ t_i^3 & t_i^4 & t_i^5 & t_i^6 & t_i^7 & t_i^8 \\ t_i^4 & t_i^5 & t_i^6 & t_i^7 & t_i^8 & t_i^9 \\ t_i^5 & t_i^6 & t_i^7 & t_i^8 & t_i^9 & t_i^{10} \end{bmatrix} \quad (4.30)$$

又有:

$$-2w_{obsLongPos}s_{ij}s(t_i) = q_{obsLongPos,t_i}^T x \quad (4.31)$$

$$-2w_{obsLatPos}d_{ij}d(t_i) = q_{obsLatPos,t_i}^T x \quad (4.32)$$

其中:

$$q_{obsLongPos,t_i} = -2w_{obsLongPos}s_{ij}[q_{obsLongPos,t_i}^{11} \quad 0]^T_{1 \times 12} \quad (4.33)$$

$$q_{obsLatPos,t_i} = -2w_{obsLatPos}d_{ij}[0 \quad q_{obsLongPos,t_i}^{11}]^T_{1 \times 12} \quad (4.34)$$

$$q_{obsLongPos,t_i}^{11} = [1 \quad t_i \quad t_i^2 \quad t_i^3 \quad t_i^4 \quad t_i^5] \quad (4.35)$$

则舒适性目标函数可写成:

$$J_{safety} = -\sum_{i=0}^{i=N} \sum_{j=0}^{j=m} \{x^T (Q_{obsLongPos,t_i} + Q_{obsLatPos,t_i})x + (q_{obsLongPos,t_i}^T + q_{obsLatPos,t_i}^T)x\} \quad (4.36)$$

4.3.4 目标函数

总的目标函数为

$$J = J_{target} + J_{comfort} + J_{safety} \quad (4.37)$$

上式中 J_{target} 、 $J_{comfort}$ 和 J_{safety} 依次带入上面推导结果。

4.4 二次规划问题的约束条件

由典型二次规划问题的标准形式可知，该算法需设定等式和不等式约束条件，对比前文采样算法的研究成果，主要约束条件包括初始点连续性约束及车辆动力学约束。

4.4.1 初始点连续性约束

在每个规划周期及相邻周期，自动驾驶车辆轨迹应保持位置、速度、加速度的连续性。汽车规划的初始状态应和汽车当前状态相同，约束条件如下

$$\begin{cases} s(0) = s_0 \\ \dot{s}(0) = \dot{s}_0 \\ \ddot{s}(0) = \ddot{s}_0 \\ d(0) = d_0 \\ \dot{d}(0) = \dot{d}_0 \\ \ddot{d}(0) = \ddot{d}_0 \end{cases} \quad (4.38)$$

将上式转化为二次规划等式约束

$$A_{start}x = b_{start} \quad (4.39)$$

其中

$$A_{start} = \begin{bmatrix} A_{start}^{11} & 0 \\ 0 & A_{start}^{11} \end{bmatrix}_{6 \times 12} \quad (4.40)$$

$$b_{start} = [s_0 \quad \dot{s}_0 \quad \ddot{s}_0 \quad d_0 \quad \dot{d}_0 \quad \ddot{d}_0] \quad (4.41)$$

$$A_{start}^{11} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \quad (4.42)$$

4.4.2 车辆动力学约束

动力学约束方面的限制主要是速度 v_{max} 和加速度 a_{max} 。假设纵向最高车速 $v_{longmax}$ ，则汽车在所有时刻的速度应满足

$$\dot{s}(t_i) \leq v_{longmax} \quad (4.43)$$

$$-\dot{s}(t_i) \leq 0 \quad (4.44)$$

规定汽车的最大加速和减速的加速度

$$s(\ddot{t}_i) \leq a_{longmax} \quad (4.45)$$

$$s(\ddot{t}_i) \geq a_{longmin} \quad \text{即} \quad -s(\ddot{t}_i) \leq -a_{longmin}$$

将上述不等式转化为不等式约束问题

$$G_{longKine}x \leq h_{longKine} \quad (4.46)$$

其中

$$G_{longKine} = \begin{bmatrix} G_{longKine}^{t_0} & 0 \\ G_{longKine}^{t_1} & 0 \\ \vdots & \vdots \\ G_{longKine}^{t_i} & 0 \\ \vdots & \vdots \\ G_{longKine}^{t_N} & 0 \end{bmatrix}_{4(N+1) \times 12} \quad (4.47)$$

$$G_{longKine}^{t_i} = \begin{bmatrix} 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & -1 & -2t_i & -3t_i^2 & -4t_i^3 & -5t_i^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 0 & 0 & -2 & -6t_i & -12t_i^2 & -20t_i^3 \end{bmatrix} \quad (4.48)$$

$$h_{longKine} = [h_{longKine,t_0}^T \quad h_{longKine,t_1}^T \quad \cdots \quad h_{longKine,t_N}^T]^T_{1 \times 4(N+1)} \quad (4.49)$$

$$h_{longKine,t_i} = [v_{longmax} \quad 0 \quad a_{longMax} \quad -a_{longMax}]^T \quad (4.50)$$

4.4.4 约束矩阵的计算

综上所述，该优化问题的约束条件为：

$$A_{start}x = b_{start} \quad (4.51)$$

$$G_{longKine}x \leq h_{longKine} \quad (4.52)$$

4.5 二次规划问题的求解方法

本节重点分析一种基于有效集法求解二次规划问题的方法，该方法具有原理明晰。计算高效的优点，可以作为轨迹规划问题的求解方法。有效集法大致步骤是从一个初始可行解开始，通过求解初始有效约束集下 QP 问题的最优解，得到下一个

可行解和有效约束集，逐步迭代，最终找到满足所有约束条件的 QP 问题的最优解。本节将重点介绍有效集法求解二次规划问题的理论基础和步骤，并给出求解算法。

4.5.1 等式约束二次规划问题

首先给出等式约束二次规划的基本形式

$$\min \frac{1}{2} x^T Q x + q^T x \quad (4.53)$$

$$s. t. a_i^T x - b_i = 0, i \in E_m$$

m 为等式约束条件的个数。

求解 QP(等式约束)问题通常利用 KKT 方程求解，KKT 方程的解即为该等式约束方程的最优解。上述等式约束有 m 个，则 KKT 条件为

$$\begin{cases} Qx + q + A^T \lambda = 0 \\ Ax = b \end{cases} \quad (4.54)$$

转化为矩阵形式

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} \quad (4.55)$$

对所有 λ ，如果有 $\lambda \geq 0$ 成立，则 KKT 方程的解就是原 QP 问题的最优解

4.5.2 不等式约束二次规划问题

对于不等式约束下的二次规划问题，格式如下

$$\min \frac{1}{2} x^T Q x + q^T x$$

$$s. t. a_i^T x - b_i = 0, i \in E_m \quad (4.56)$$

$$a_i^T x - b_i \leq 0, i \in I_l$$

该格式表明有 m 个等式约束，1 个不等式约束，下面将详细叙述该问题求解方法。

1. 首先假设一个可行解 x^0 ，该可行解可使部分不等式约束取到等式，则该约束问题变为

$$\min \frac{1}{2} x^T Q x + q^T x \quad (4.57)$$

$$s.t. a_i^T x - b_i = 0, i \in E_m \cup I(x^0)$$

通过求解该等式约束二次规划问题可获得最优解 \hat{x}^0 ,对应的 λ^0 。

2.若 $\hat{x}^0 = x^0$, 通过判断 λ^0 有

(1) 若所有 $\lambda_i^0 \geq 0, i \in I(x^0)$,则 x^0 为该二次优化问题的最优解, 输出并终止。

(2) 如果存在 $\lambda_j^0 < 0$, 则说明添加该不等式为等式会使效果变差, 因此排除 λ_j^0 所在的等式约束, 即 $x^1 = \hat{x}^0, I(x^1) = I(x^0) \setminus \{j\}$, 转回步骤 1 重新求解。

3.若 $\hat{x}^0 \neq x^0$, 则将 \hat{x}^0 带入原约束条件中验证是否为可行解。

(1) 若 \hat{x}^0 为该优化问题的可行解, 则更新可行解为 $x^1 = \hat{x}^0, I(\hat{x}^0)$, 这里的 $I(\hat{x}^0)$ 是否更新根据 λ 是否为 0, 转回步骤 1 重新求解。

(2) 若 \hat{x}^0 不是该优化问题的可行解, 则定义步长因子 α 和步长 $d = \hat{x}^0 - x^0$, 更新可行解如下

$$\begin{aligned} x^1 &= x^0 + \alpha d \\ \alpha &= \min \left\{ \frac{b_i - a_i x^0}{a_i d} \right\} \\ i &\in I \setminus I(x^0) \end{aligned} \quad (4.58)$$

求解完上述步骤后更新有效集 $I(x^1) = I(x^0) \cup \{P\}$, P 为更新可行解后重新满足的不等式的等式约束, 转回步骤 1 重新求解。有效集法逻辑大致如下: 见图 4.1。

从初始可行解 x^0 开始, 有效约束 I_0 , $k = 0$

Step1.

求解原等式约束集和有效约束集 I_0 组成的 QP 问题, 得 x 和 λ 。

Step2.

if ($x = x^k$)

if ($\lambda \geq 0$)

$x_{optimal} = x$

break

else

$\lambda_p = \min\{\lambda_i\}, i \in I^k$

$x^k = x$

$I^k = I^k - \{p\}$

$k = k + 1$

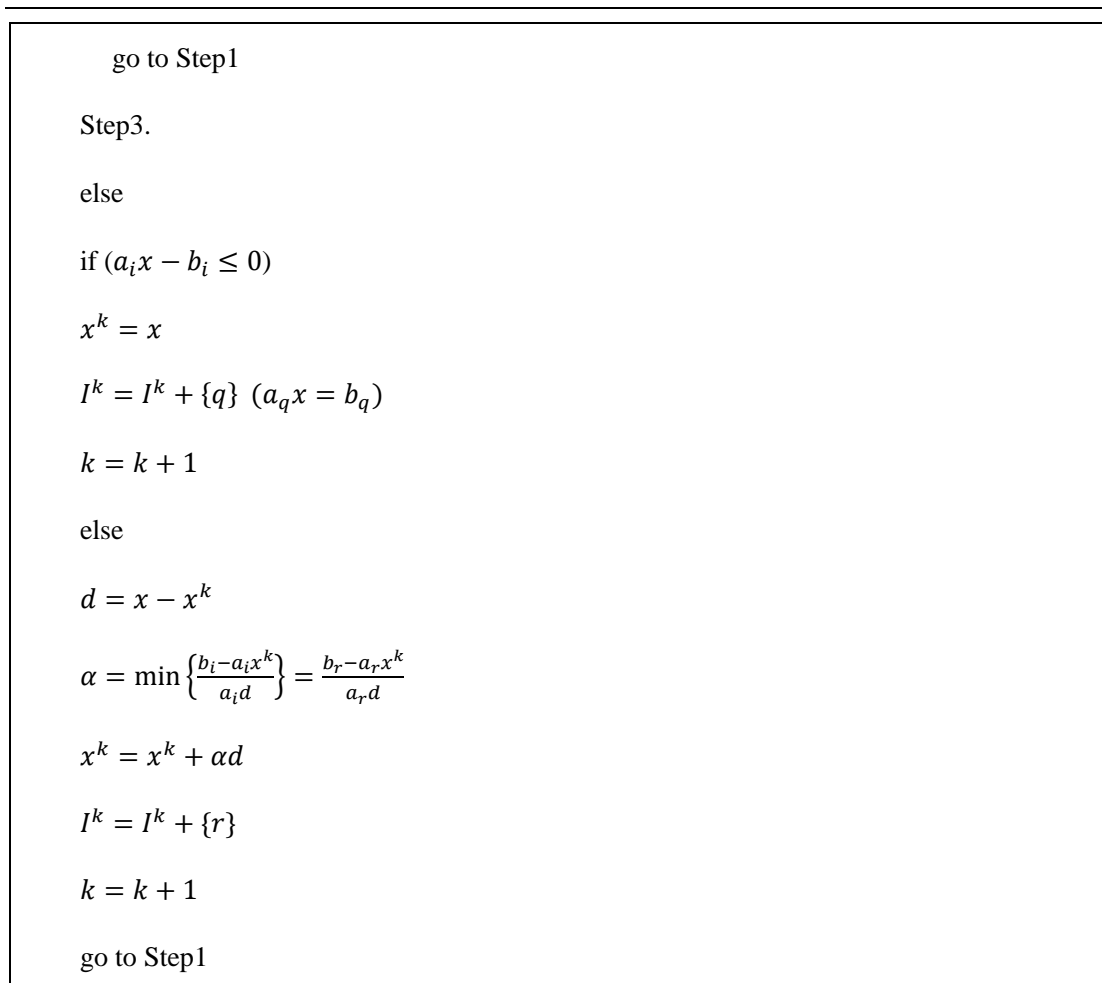


图 4.1 有效集法求解二次规划问题逻辑图

4.5.3 有效集法求解不等式约束二次规划问题算法研究

本小节针对前文对有效集法的分析，利用 Python 编写基于有效集法求解不等式约束二次规划的算法，并可视化该求解过程。

本文取如下例子进行求解

$$\begin{aligned}
 \min q(x) &= (x_1 - 1)^2 + (x_2 - 2.5)^2 \\
 \text{s.t. } x_1 - 2x_2 + 2 &\geq 0 \\
 -x_1 - 2x_2 + 6 &\geq 0 \\
 -x_1 + 2x_2 + 2 &\geq 0 \\
 x_1 &\geq 0 \\
 x_2 &\geq 0
 \end{aligned} \tag{4.59}$$

基于前面的理论推导可得

$$Q = \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 2.0 \end{bmatrix} \quad (4.60)$$

$$q = \begin{bmatrix} -2.0 \\ -5.0 \end{bmatrix} \quad (4.61)$$

$$A = \begin{bmatrix} -1 & 1 & 1 & 1 & -1 \\ 2 & 2 & -2 & 0 & -1 \end{bmatrix}^T \quad (4.62)$$

$$b = [2 \quad 6 \quad 2 \quad 0 \quad 0] \quad (4.63)$$

编写 Python 算法求得最优解为 $x = (1.4, 1.7)^T$

有效集法为初始点从初始位置逐渐迭代到最优位置，将该过程可视化见图 4.2 该问题的约束条件和迭代过程可表示为图 4.3 所示。

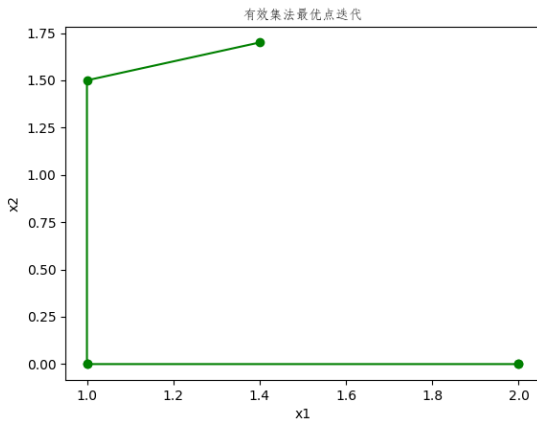


图 4.2 有效集法最优点迭代

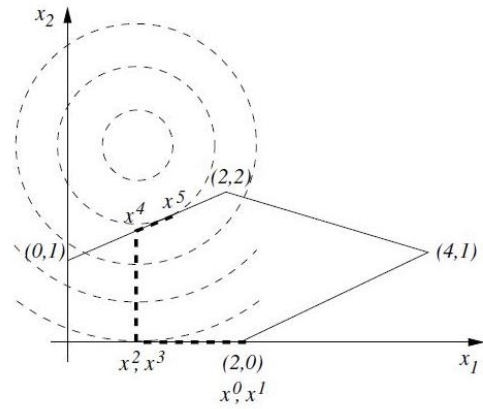


图 4.3 二次规划问题的可视化

图 4.3 中 x^5 为该二次规划问题下的最优解。

同时针对求解该优化问题的普遍性，Python 开发了开源优化软件 CVXOPT，通过输入上述 Q 、 q 、 A 和 b 四个矩阵即可获得该目标函数和约束条件下的最优解。求解结果见图 4.4

```

      pcost      dcost      gap      pres      dres
0: -5.3526e+00 -2.2409e+01  2e+01  3e-17  7e-01
1: -5.9518e+00 -7.3326e+00  1e+00  1e-16  5e-02
2: -6.4265e+00 -6.4758e+00  5e-02  2e-16  2e-04
3: -6.4498e+00 -6.4503e+00  5e-04  7e-17  2e-06
4: -6.4500e+00 -6.4500e+00  5e-06  1e-16  2e-08
Optimal solution found.
[ 1.40e+00]
[ 1.70e+00]
```

图 4.4 CVXOPT 求解结果

4.6 优化算法实际场景的应用

本节根据前文对优化算法的实现和场景的规定，进行针对定速巡航和停车两个场景的模拟分析。

4.6.1 定速巡航场景

本节采用五次多项式拟合定速巡航轨迹曲线，将横纵向曲线分别进行规划后再合并形成最终曲线。本次场景设定为 $T = 5s$, $v_{start} = 6m/s$, $v_{target} = 15m/s$, $a_{target} = 0m/s^2$, $c_{start} = 0m$, $c_{target} = 0m$ 。纵向轨迹速度曲线见图 4.5。横向轨迹位置优化曲线见图 4.6。

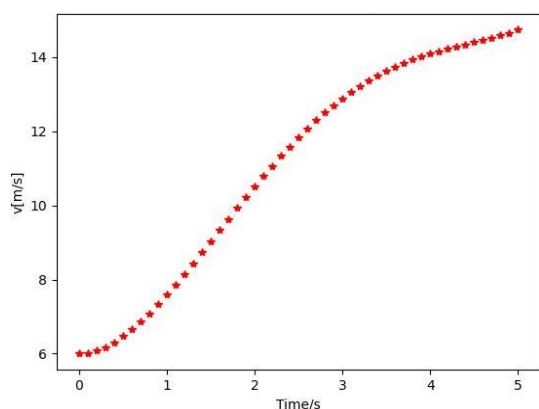


图 4.5 定速巡航纵向速度优化曲线

$$(a_{max} = 3m/s^2, v_{max} = 70km/h)$$

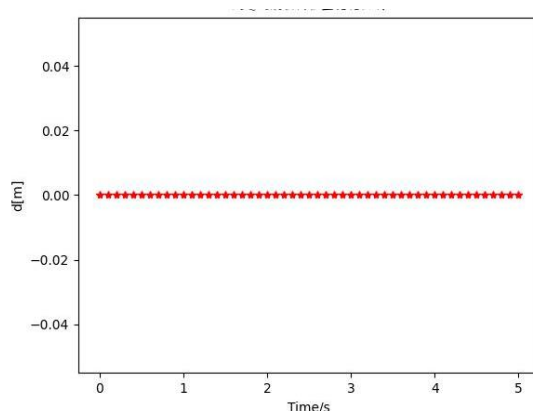


图 4.6 定速巡航横向位置优化曲线

$$(a_{max} = 3m/s^2, v_{max} = 70km/h)$$

由上述模拟结果分析，纵向轨迹速度能良好跟踪目标速度，横向轨迹位移保持参考线中央位置，该规划结果符合场景需求。

通过合并横纵向轨迹曲线获得如图 4.7 所示轨迹，该优化算法在定速巡航场景下能较为理想地实现轨迹规划，同时该场景下的每次循环时间极短，具有实时性好的优点。

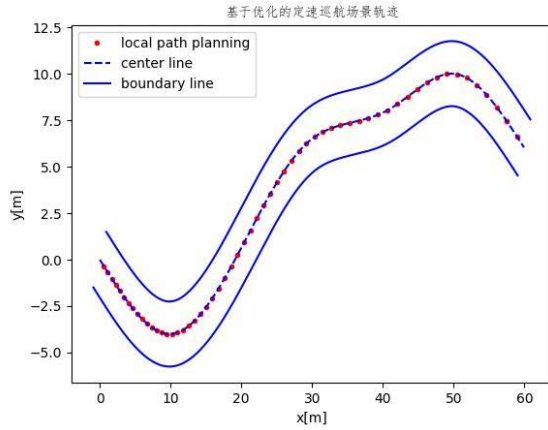


图 4.7 定速巡航循环采样

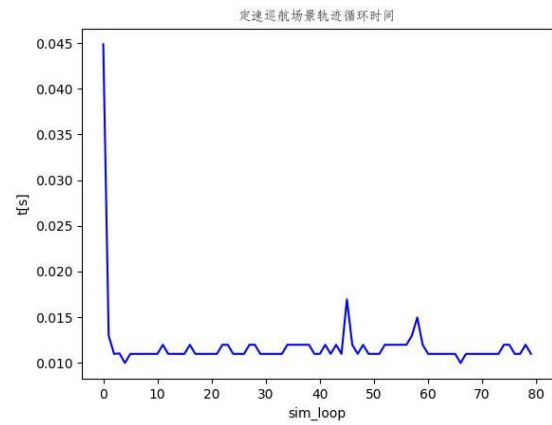


图 4.8 定速巡航轨迹循环时间

在上述成果的基础上，人为添加静态障碍物并设置和碰撞相关的目标函数，获得效果如下图 4.9 所示的。测试证明，在认为添加较为宽松的约束条件时，该路径规划具有良好的避障效果。

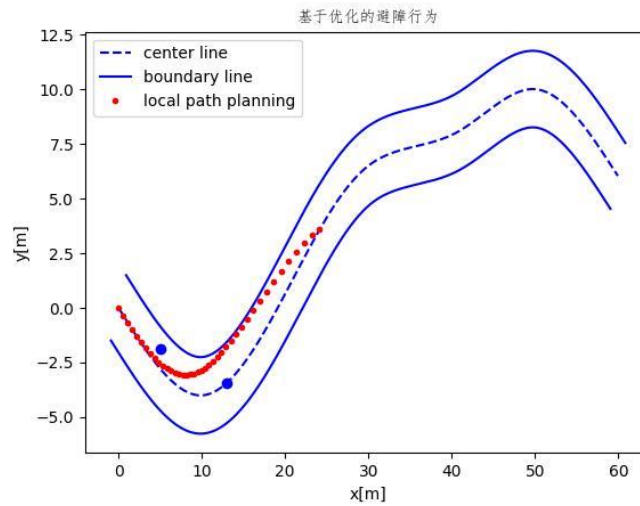


图 4.9 基于优化的避障行为

4.6.2 停车场景

该场景轨迹规划依然采用横向和纵向分别进行轨迹规划，轨迹规划轨迹利用五次多项式拟合。本次场景设定为 $T = 5s$, $v_{start} = 6m/s$, $v_{target} = 0m/s$, $a_{target} = 0m/s$, $c_{start} = 0m$, $c_{target} = 0m$ 。本次停车距离 $s_{target} = 10m$ 。通过编写算法实现了停车轨迹规划。速度曲线见图 4.10。

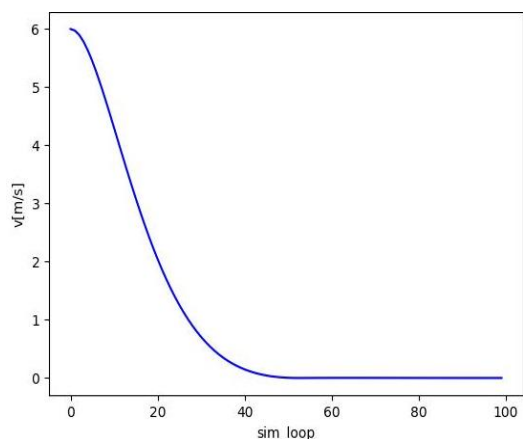


图 4.10 停车场景下的速度曲线

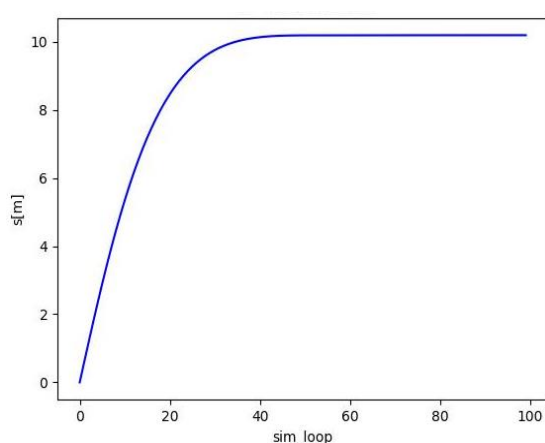


图 4.11 停车场景下的位移曲线

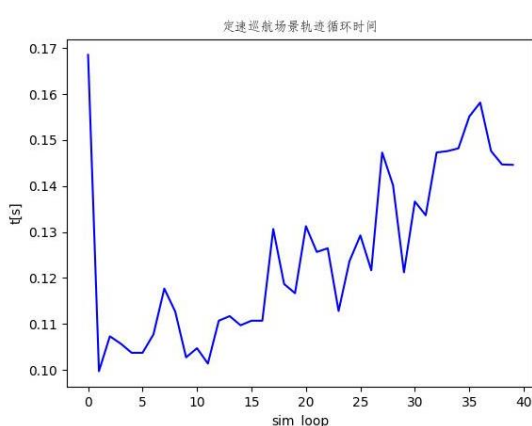


图 4.13 停车场景下的循环时间曲线

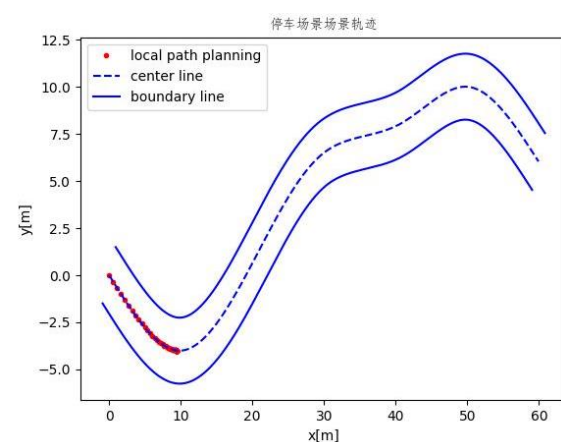


图 4.12 停车场景下的循环曲线

该算法能较为准确的实现停车场景下的目标，即在指定位置减速至指定速度，同时该算法每次循环相比基于采样的实现思路耗时较少，具有实时性好的特点。

4.7 本章小结

本章基于前文基于采样的路径规划算法中存在的实时性不强和规划成功率低的缺点采用基于优化的方法进行了改进。本章首先提出了基于优化的轨迹规划的基本流程，设计了包括舒适性、安全性和目标性在内的目标函数以及车辆动力学约束和连续性约束的约束条件。本章将实际场景下的路径规划问题转化为基于二次规划的优化问题求解，提出了利用有效集法求解二次规划问题的方法。最后搭建了定速巡航、停车等场景对算法进行了测试。通过测试结果初步判定该算法能显著降低路径规划的循环时间，并在场景参数发生改变时仍能保持较高的规划成功率。

第五章 基于不同场景下的算法对比分析

针对前文提出的两种轨迹规划算法，为综合对比改善前后的算法实际效果，本章设置了定速巡航、变道和停车的标准场景参数。在相同场景参数下分别利用两种算法进行轨迹规划并利用 MATLAB 对比分析了两种算法在实际场景下的规划耗时、算法稳定性及合理性等。针对采样算法的实时性不强、规划成功率不高等问题，本章设计了实验方法探究了其原因为。最后针对两种算法的优劣进行了综合总结。

5.1 基于场景下的参数设置

本节在前文研究工作的基础上，设定了统一的场景参数并利用基于采样的路径规划算法和基于优化的路径规划算法分别进行路径规划并对比两者的效果，为后文分析两种算法的优劣进行场景铺垫。

5.1.1 定速巡航场景

该场景下的规划目标当自车周围环境良好时，自车在指定的规划时间内，尽快加速到前方领航车的车速并保持恒定速度行驶，该场景下自车继续保持该车道行驶。效果如图 5.1 所示。

本文设置场景相关参数如表 5.1 所示

表 5.1 定速巡航标准场景参数

	s_{start}	v_{start}	a_{start}	d_{target}	v_{target}	a_{target}	T_{min}	T_{max}
	[m]	[m/s]	[m/s ²]	[m]	[m/s]	[m/s ²]	[s]	[s]
采样	0	6	0	0	15	0	4	5
算法								
优化	0	6	0	0	15	0		5
算法								

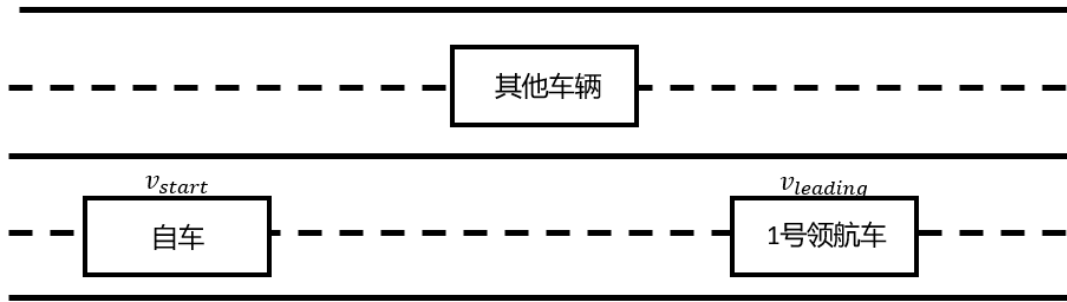


图 5.1 定速巡航场景

5.1.2 停车场景

该场景的规划目标是自车以初始速度 v_{start} 匀速行驶，当感知到前方需要减速停车时，自车需在规定的规划时间内在指定位置处减速并停车。

该场景主要参数规定如下表 5.2 所示

表 5.2 停车标准场景参数

	s_{start}	v_{start}	a_{start}	d	s_{target}	v_{target}	a_{target}	T_{min}	T_{max}
	[m]	[m/s]	[m/s ²]	[m]	[m]	[m/s]	[m/s ²]	[s]	[m]
采样	0	6	0	0	10	0	0	4	5
算法									
优化	0	6	0	0	10	0	0		5
算法									

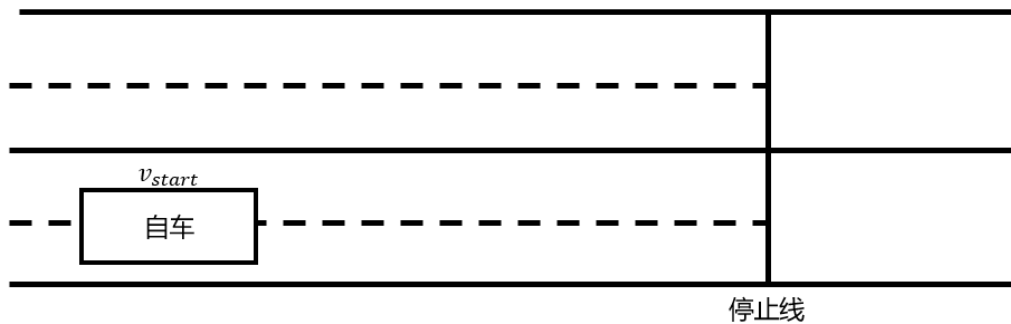


图 5.2 停车场景

5.1.3 变道场景

该场景的规划目标是自车在指定时间内变道到相邻车道，并加速至前方领航车的车速 v_{target} 。

该场景主要参数规定如下表 5.3 所示。

表 5.3 变道标准场景参数

	d_{start}	v_{start}	a_{start}	d_{target}	v_{target}	a_{target}	T_{min}	T_{max}
	[m]	[m/s]	[m/s ²]	[m]	[km/h]	[m/s ²]	[s]	[s]
采样	0	6	0	3.5	50	0	4	5
算法								
优化	0	6	0	3.5	50	0		5
算法								

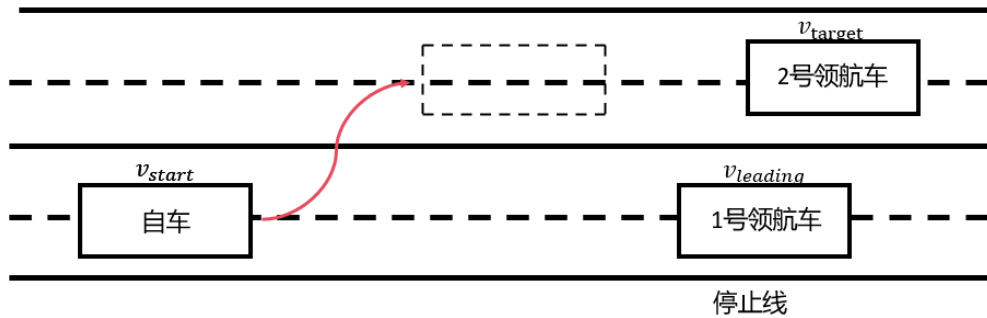


图 5.3 变道场景

5.2 基于算法的对比分析

本节重点根据上述的场景参数设置，利用 Python 语言，在保证相同硬件环境的基础上对以上两个场景下不同算法的轨迹规划效果进行对比分析。本节重点研究轨迹规划的两个重要评价指标：实时性和规划成功率。实时性是评价轨迹规划算法优劣的重要指标，本文通过改变基于采样的路径规划算法的采样空间和障碍物个数来观察对基于采样的路径规划算法的影响，同时在相同场景参数下对比基于采样和基于优化的两种算法的实时性；此外，针对采样算法规划成功率不高的缺陷，通过改

变部分场景参数对比基于优化和基于采样的算法输出解的稳定性。最后根据真实的仿真情况做出评价。

5.2.1 实时性分析

轨迹规划算法的实时性是一个关键的性能指标。在实际场景中，周围环境瞬息万变，为避免发生安全事故，智能汽车需要不断更新自身状态并规划轨迹。在尽量短的时间内规划出合理、安全和舒适并满足车辆动力学要求的路径是进行动态场景应用的关键，因此实时性高的算法是轨迹规划算法评价的重要指标之一。为了评估两种轨迹规划算法的实时性，本研究对比了相同条件下的两种算法的耗时，通过改变采样轨迹数量和障碍物个数比较了不同采样空间下轨迹规划循环时间。

首先，我们对比了一种在前文研究中耗时较高的场景：停车场景。本研究利用 Python 编写算法分别对相同场景参数下的场景进行轨迹规划并利用 MATLAB 进行数据的分析和对比。该场景下对两种算法分别进行 100 次循环并获得每次循环下路径规划的时间，如下图 5.4 所示。

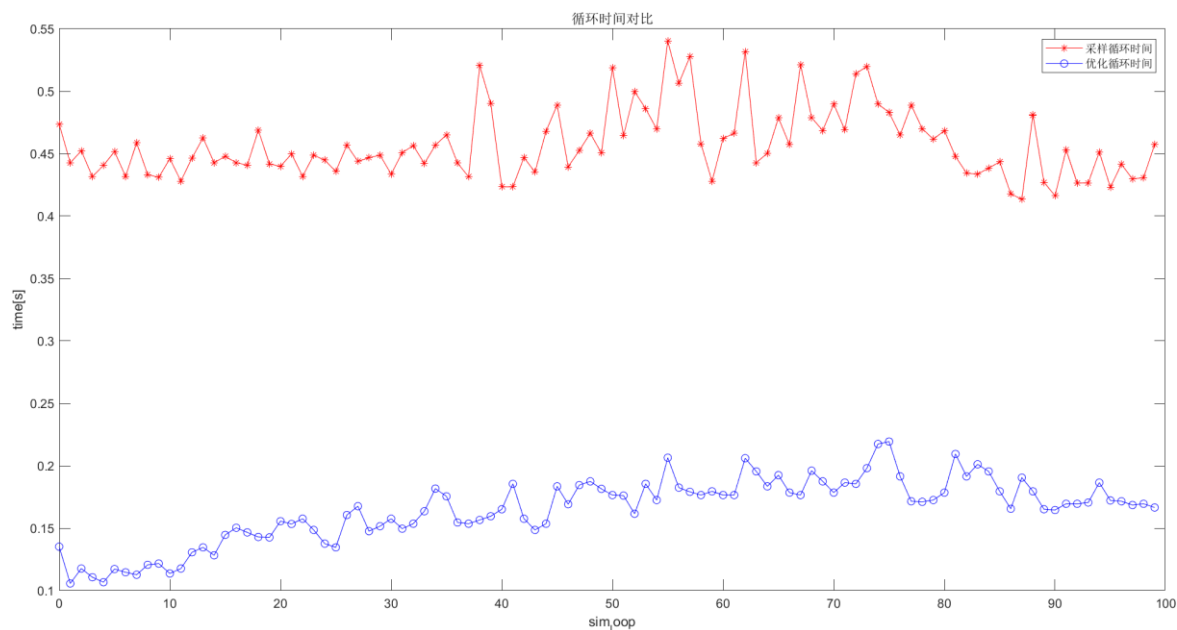


图 5.4 算法循环时间对比

利用 MATLAB 求取所有循环下的规划时间均值，其中基于采样的算法平均循环时间 0.4572s，基于优化的算法平均循环时间 0.1648s，相同场景下基于优化的方

法比基于采样的方法耗时节约超过 60%，这极大地提高了轨迹规划的实时性。同时，本研究探索了采样算法的耗时大的原因，认为采样空间的复杂和障碍物检测是造成路径规划耗时的原因。下面选取规划较为复杂的变道场景，通过分别改变采样空间备选轨迹条数和变道过程的障碍物个数，对比分析轨迹规划的循环时间

首先是对基于采样的轨迹规划方法中采样空间对实时性影响的研究。通过改变采样的范围，本研究先后在相同场景参数下通过改变纵向目标速度采样数量先后模拟了数目为 1600、1200 和 800 采样空间的轨迹规划，并利用 MATLAB 可视化了三种采样范围下轨迹规划的循环时间，如图 5.5 所示。

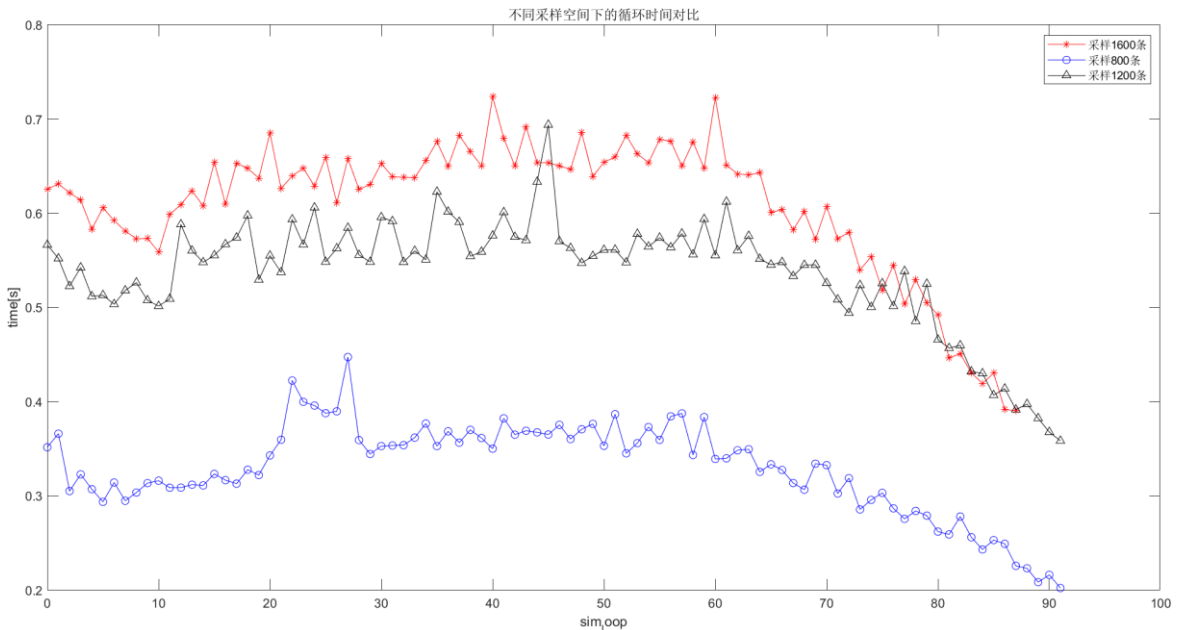


图 5.5 不同采样空间下的循环时间对比

表 5.4 变道场景下不同采样空间的平均规划耗时

采样空间	1600 条	比例	1200 条	比例	800 条	比例
采样耗时	0.6098	100%	0.5378	88.2%	0.3296	54.1%
优化耗时	0.20	100%	0.20	100%	0.20	100%

如表 5.4 所示，实际模拟发现基于采样的轨迹规划算法在采样空间增多的情况下，平均规划耗时水平升高较为明显。以 1600 条备选轨迹组成的采样空间作为标准，1200 条备选轨迹组成的采样空间下的平均规划耗时下降 12%，800 条备选轨迹组成的采样空间下的平均规划耗时下降约 45%。这表明采样空间越多，在实际轨迹

规划时算法需要评价的轨迹就越多，这一过程是造成耗时增加的重要因素。

同时，本研究还通过改变了实际场景下的障碍物个数，探求障碍物个数和轨迹规划时间的关系。基于采样的算法在障碍物检测原理是将 **Frenet** 坐标系下的备选路径转化为笛卡尔坐标系下的坐标进行碰撞检测，这一过程频繁涉及两个坐标系间的变换，因此推测为采样算法轨迹耗时的原因。本研究通过模拟前文预设的变道场景，分别在该场景设定 0 个、3 个和 8 个障碍物，在其余条件相同的情况下分别计算循环时间并对比，如图 5.6

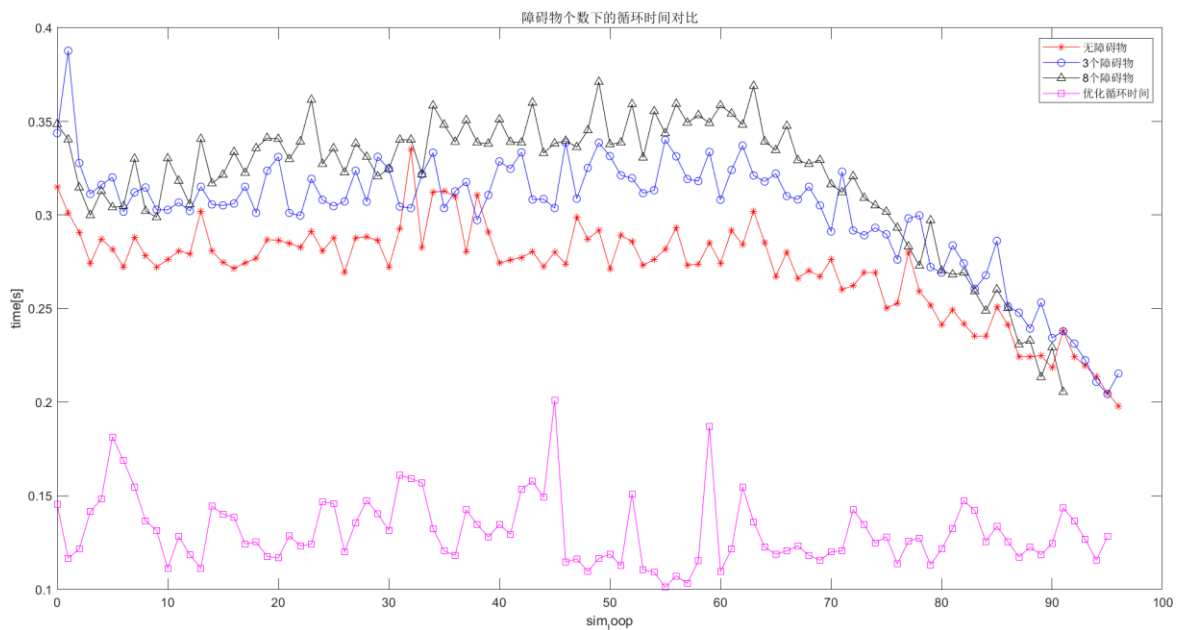


图 5.6 不同障碍物个数下的循环时间对比

表 5.5 变道场景下不同障碍物个数下的平均规划耗时

障碍物个数	8 个障碍物	比例	3 个障碍物	比例	无障碍物	比例
采样耗时	0.3205	100%	0.3024	94%	0.2725	85%
优化耗时	0.1311	100%	0.1311	100%	0.1311	100%

实验结果表明，随着障碍物个数的增加，基于采样算法的循环时间乘上升趋势，这是由于障碍物检测过程中频繁的坐标表换造成的计算负担。以 8 个障碍物为标准，3 个障碍物下的基于采样的路径规划算法循环时间下降约 6%，无障碍物下的基于采样的路径规划算法循环时间下降约 15%。以规划时间 $T = 5s$ ，采样频率为 $10Hz$ ，采样空间为 800 条轨迹为例，一个障碍物在一次循环中的障碍物检测次数为

40000 次，因此障碍物的检测环节是影响路径规划实时性的重要因素。而基于优化的方法在在障碍物个数增加的过程中平均规划耗时变化不明显，这是由于在设置目标函数时将自车和障碍物之间的距离作为目标函数的一部分，在轨迹规划过程中自车轨迹会尽量保持和障碍物的距离，从而避免了障碍物检测的环节，因此基于优化的路径规划算法平均规划耗时随障碍物数目的变化不明显。

通过上面对于基于采样的路径规划算法和基于优化的路径规划算法对比，本研究发现，基于优化的路径规划算法在实时性方面要明显优于基于采样的方法且在场景参数变化时平均规划耗时较为稳定。针对基于采样的路径规划算法实时性差的原因，本研究设计实验发现且障碍物检测环节和采样空间的增加是造成基于采样的路径规划算法实时性差的重要因素。

5.2.2 规划成功率及合理性分析

路径规划算法的规划成功率指的是该算法对于现实中多样化的场景环境能否始终输出规划路径的能力。在实际场景中，车辆会遇到各种形式的场景参数变化，如果规划成功率不高往往意味着自车行驶中无法始终保证规划路线生成，给自车带来潜在的危险。而稳定性强的算法能保证自车快速适应环境、避免碰撞，从而提高路径规划的安全性和可靠性。本节针对上述场景，通过改变场景参数来模拟实际场景的变化并重点分析了基于采样的算法和基于优化的算法在变化场景下的表现的合理性。

首先针对停车场景，我们选择上述标准场景进行分析。通过对上述停车场景的模拟，本研究重点分析了两种算法在该场景下自车速度的变化和刹车距离变化下的对比。首先是自车刹车速度的变化，见图 5.7 所示。测试发现，基于采样的路径规划算法在在停车场景中出现了“刹车过度”的现象，即速度降至 0m/s 以下，形成“倒车”情况，如图 5.8 所示，这在实际规划中显然是不合理的，而基于优化的路径规划算法能够较为良好地符合实际场景应用，同时在同一场景下，基于优化的算法可以通过更少的迭代次数实现目标状态。基于采样的路径规划算法速度规划不合理的原因在于在路径规划中的运动学检测环节基于轨迹的速度绝对值进行检测，规

划节点的速度范围为 $[-v_{max}, v_{max}]$ ，未进行速度为负的检测。而基于优化的算法在路径规划中设置了动力学约束条件，将速度始终限制在 $(0, v_{max})$ 之间，从而保证了在停车场景路径规划的全过程能够进行合理规划。同时，基于采样的方法是针对规划始末端点进行插值多项式，未对之间的多项式轨迹速度进行限制，这可能导致局部曲线出现不合理的情况。

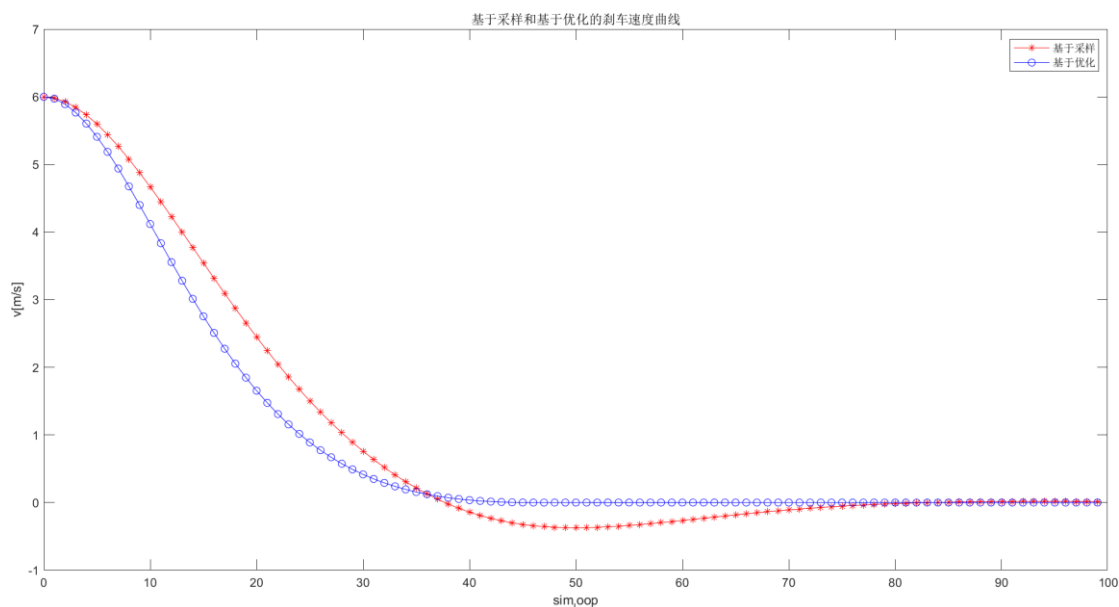


图 5.7 基于采样和基于优化的刹车速度变化对比

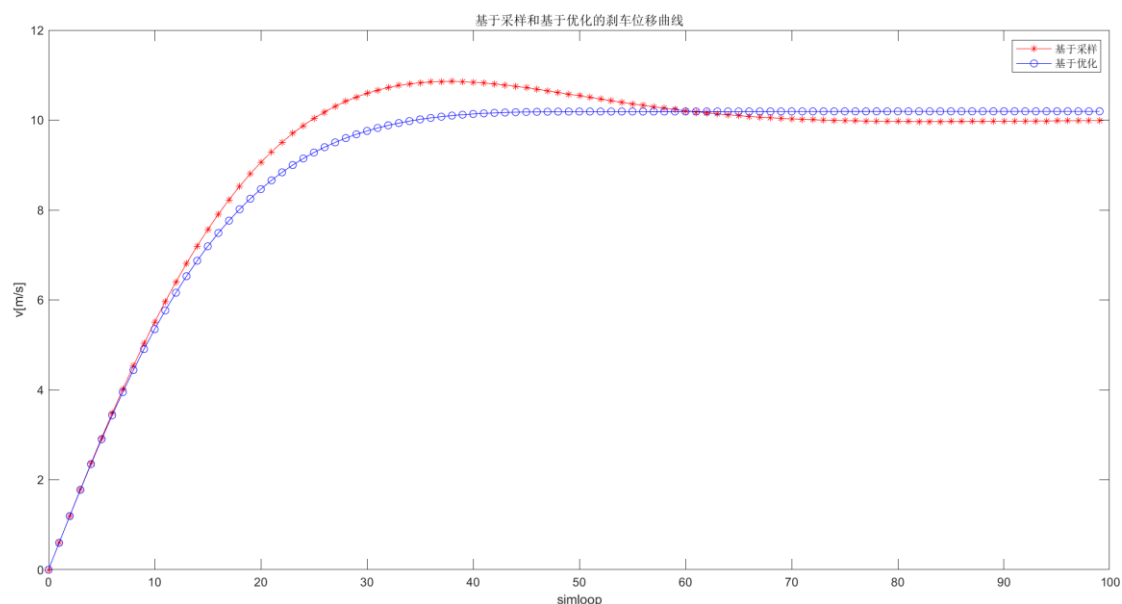


图 5.8 基于采样和基于优化的刹车位移变化对比

同时，由于基于采样的算法由于是在目标状态周围采样，备选轨迹空间有限因此场景变化时可能导致备选空间解集中所有解都不符合实际场景要求，这表明该

算法在该场景下会出现规划失败的结果。基于优化的方法由于是在“无限”解空间下的路径规划，其对于场景变化下的适应性较好，可以在满足要求的情况下尽可能输出路径规划解。以变道场景为例，若将 a_{max} 设置为 $2m/s^2$ 并计算采样空间所有轨迹的加速度曲线如下图 5.9 所示，该场景参数下，所有采样空间下的最小加速度均超过 a_{max} ，因此在有限解集下没有输出解并导致规划失败而基于优化的方法仍能寻找到可行解输出。由前文对轨迹规划实时性的分析，在采样空间加大时，基于采样的路径规划算法平均规划时间大大增加，因此为提高轨迹规划的实时性，通常解空间不能过大，因此在较小解空间下找到可行解的概率会有所降低。相比之下，基于优化的路径规划算法由于其无限的解空间使得轨迹规划失败的概率大大降低，因此在场景参数变化时的稳定性更优。

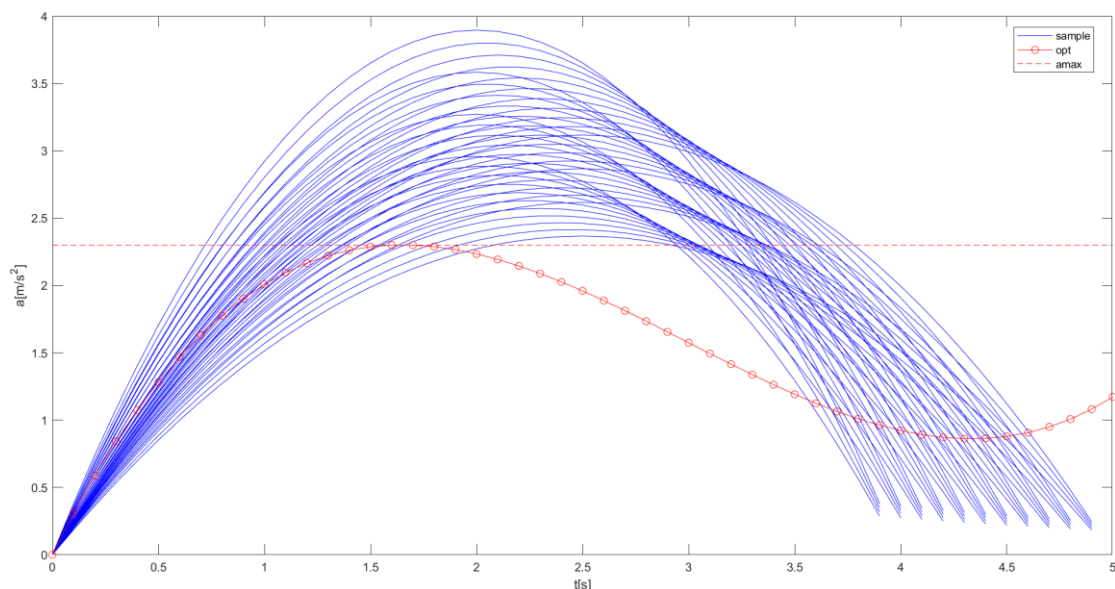


图 5.9 变道场景下自行车加速度的采样空间

5.3 综合对比分析

通过上述对基于优化的轨迹规划算法和基于采样的轨迹规划算法在实时性和规划成功率等方面的表现，本研究发现，基于优化的路径规划算法利用高效的二次规划方法求解极大地提高了算法的求解效率，使得轨迹规划算法的实时性相比基于采样的轨迹规划算法提高很多。本研究还分析了造成基于采样的轨迹规划算法耗时多的原因，将实时性差的原因归结于采样空间的冗杂和障碍物检测环节的耗时。另一

方面，本研究针对两种算法在规划轨迹中的特点，基于采样的算法在轨迹规划中由于曲线插值过程只涉及始末状态的限制且未对端点的运动参数范围限制，因此存在不合理的中间轨迹输出。同时面对场景参数的变化，该算法适应性较差，其主要原因是基于采样的轨迹规划是有限的采样空间，较容易出现规划失败的情况。综上所述，基于优化的轨迹规划算法在一定程度上弥补了基于采样的方法的实时性差和规划成功率不高及算法规划不够合理的缺点，可以作为改善采样算法的一种有效方式。

5.4 问题分析

基于优化的方法目前也存在不足，基于优化的方法虽然简化了轨迹规划的过程，将障碍物躲避也加入到目标函数中，但这增加了目标函数需要权衡的参数，使得目标函数互斥的参数项之间互相博弈从而造成输出轨迹并不能完全贴合使用场景要求。见图 5.10。

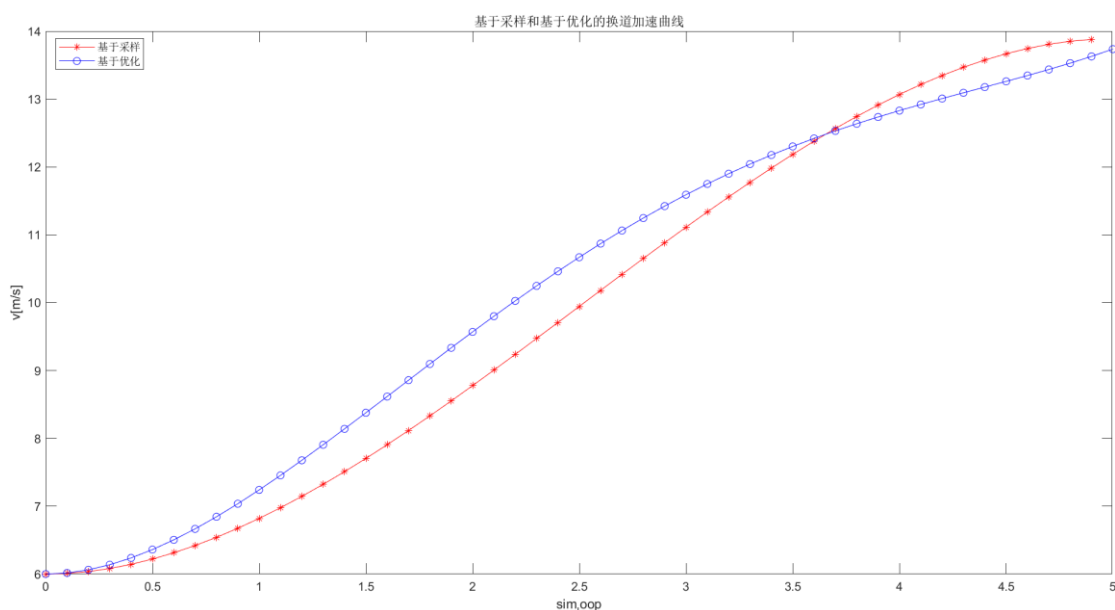


图 5.10 变道场景下车速度的曲线对比

由测试结果可知，在变道并加速这一过程中，基于优化的规划算法速度曲线在规划周期末尾仍保持着一定加速度，即仍具有“过度加速”的趋势。相比之下，基于采样的算法通过在目标状态周围进行采样，输出结果始终保持贴合目标状态需求。这表明在基于优化的轨迹规划中，其他目标项的占比可能在速度接近目标状态

时超过运动目标项，使得目标函数在规划终点附近并不总是趋向最终运动目标而是朝向其他目标规划多项式路径。因此目标函数之间权值的分配和博弈也是场景规划的重点。

同时在有障碍物的场景下，基于优化的路径规划算法的规划成功率有所下降。本研究认为，基于优化的方法通过构建优化方程求解轨迹多项式系数，该过程涉及凸优化的过程，即规划场景必须是一个凸优化问题才能利用二次规划求解轨迹多项式。在避障场景中，基于优化的路径规划往往成为一个非凸优化问题，这是造成障碍物场景下路径规划失败概率增加的重要原因。

5.5 本章小结

本章针对两种算法设置了定速巡航、变道和停车标准场景参数并将算法实际应用于场景中的轨迹规划中进行实时性、规划成功率及合理性的对比分析。研究结果显示，基于优化的算法通过高效的二次规划解法有效降低了轨迹规划的时间，基于优化的轨迹规划算法中无限解空间提高了轨迹规划的成功率。相比采样算法中只对始末端点的动力学参数进行约束，基于优化的轨迹规划算法通过设计约束条件保证了轨迹规划全过程中动力学约束的合理性。针对采样的轨迹规划算法中实时性不高和规划成功率较低的缺陷，本研究发现采样空间的冗杂和障碍物检测环节是造成基于采样的轨迹规划算法实时性下降的主要原因，采样本身造成的解空间的有限是造成基于采样的算法规划成功率不高的主要原因。同时，基于优化的轨迹规划算法由于目标函数的复杂化，因此也存在自车在路径规划上对于目标状态的趋向性不强等缺陷。

第六章 工程与社会

本章针对研究的无人驾驶汽车轨迹规划算法模型，重点分析其对工程应用和社会发展的重大意义。

首先是工程意义：

（1）安全性提升

轨迹规划涉及到多个领域的技术，如机器人学、计算机视觉、人工智能等。推动无人驾驶轨迹规划技术的研发和应用，促进了这些领域的技术创新和进步，为工程领域的发展提供了新的动力和方向。

（2）系统设计优化

轨迹规划算法的优化设计可以提高无人驾驶系统的性能和稳定性，降低能耗和成本。本文重点提出了基于优化的轨迹规划算法模型，通过实际测试可以发现，该轨迹规划算法可以显著降低轨迹规划的耗时，提高轨迹规划成功率，这为未来工程领域的无人驾驶轨迹规划研究提供了新的思路，为工程系统的设计和运营提供了更多可能性和选择。

（3）智能交通建设

轨迹规划技术的应用促进了智能交通系统的建设和发展。这包括了交通信号控制、交通流量优化、智能交通管理等方面，为交通工程的规划、设计和管理提供了智能化的解决方案，提升了交通系统的整体效率和安全性。

该研究的社会意义如下：

（1）提升交通效率

高效安全的轨迹规划是无人驾驶应用落地的重要一环，该研究将致力于改善无人驾驶特别是轨迹规划领域面临的实时性不好、平滑性差及规划成功率不高的缺陷，加速无人驾驶落地。减少交通拥堵和交通事故，提高交通运输效率。这有助于缓解城市交通压力，减少交通排放，改善环境质量，提升城市居民的出行体验。

（2）城市发展推动

无人驾驶技术的发展推动了智慧城市的建设。轨迹规划的实施促进了城市交通

系统的智能化和网络化，为城市的数字化转型提供了重要支撑，推动了城市发展向着更加智能、绿色和可持续的方向发展。

（3）驾驶安全性提升

无人驾驶轨迹规划的实施可以显著提升道路交通的安全性。通过精确计算和预测车辆行驶轨迹，减少了人为驾驶误操作导致的交通事故风险，从而保障了驾乘人员的生命安全。

本章总结了本研究搭建的基于采样的和基于优化的轨迹规划模型在社会和工程领域的应用。未来本研究成果将继续推进，将工程和社会价值与研究价值相结合，为无人驾驶的早日落地贡献自己的力量。

第七章 总结与展望

7.1 全文研究总结

本研究针对无人驾驶汽车规划模块，重点聚焦局部路径规划部分进行了基于优化的多项式曲线轨迹规划方法的研究，改善了目前路径规划算法中存在的实时性较差、规划成功率不高等缺点。基于 Frenet 坐标，本研究采用横纵向轨迹规划解耦的方式并利用高次多项式曲线进行路径曲线的平滑拟合。本研究首先采用了基于采样的路径规划算法，针对该算法暴露出的缺陷采用基于优化的路径规划算法进行改进。实验结果表明，基于优化的路径规划算法在实时性和规划成功率等方面明显优于基于采样的路径规划算法。本文研究成果大致如下：

（1）提出了基于 Frenet 坐标系的高次多项式曲线规划方法

本文提出了基于多项式曲线插值原理的路径曲线拟合办法，有效提高了轨迹的平滑性和连续性。提出了一种基于结构化道路进行高效路径规划的坐标系 Frenet 坐标系，采用将横纵向轨迹解耦规划的方法大大降低了规划的复杂性。

（2）提出了基于采样的多项式路径规划算法

本文针对多项式曲线插值设计了基于采样的路径规划算法，并根据实际场景的规划目标对运动目标状态进行采样，并设计运动学约束和碰撞检测模块确保轨迹安全合理，设计评价函数对备选轨迹进行选择，最终设计实际场景进行算法测试。

（3）提出了基于优化的多项式路径规划算法

针对基于采样的方法在实际测试中暴露出的问题，本研究采用了基于优化的路径规划算法加以改进。通过将路径规划问题转化为求解高效的二次规划问题，本研究设计了实际场景下的目标函数及约束条件，将碰撞检测模块转化为目标函数中远离障碍物的程度，并设计等式约束和不等式约束以满足对实际场景的规划要求。

（4）进行了针对场景的算法对比分析

为综合对比分析在实际场景下两种算法的效果，本研究设计了标准场景参数，将两种算法在实际场景中进行测试。通过实际测试，本研究发掘了导致基于采样的

路径规划算法实时性差和规划成功率不高的原因，并发现基于优化的路径规划算法可以显著提高路径规划的实时性，且在场景参数变化时仍保持稳定。同时，针对基于优化的路径规划算法也存在偏离运动目标状态等缺陷。

7.2 研究意义和价值

本文重点设计了无人驾驶规划模块中的局部路径规划部分。针对局部路径在实际场景下可能存在的规划路径不平顺、可控性不高等缺陷，采用符合运动学规律、更易被控制系统追踪的高次多项式轨迹进行拟合。针对目前采样算法存在的实时性不好、规划成功率不高等问题提出了基于优化的方法进行了合理地解决。本文分析了局部路径轨迹规划的一般流程和涉及的基本原理，包含 Frenet 坐标系下的轨迹曲线插值拟合以及基于采样和优化的轨迹规划算法实现，对算法的基本原理进行了详实地分析和数学推导。这为未来从事路径规划相关研究奠定了良好的基础，也为解决路径规划算法落地问题提供了有力支撑。

7.3 不足和展望

受时间及理论储备限制，本研究仍存在诸多问题尚需要进一步思考和解决。

（1）评价函数及目标函数组成及权重仍需进一步研究。

通过实际测试发现，不同权值下的评价函数或目标函数会使规划轨迹产生不同的倾向性，且不同场景下需根据实际情况选择合适的目标函数。如在障碍物较多的场景，应优先选择安全性而降低对舒适性和运动目标的要求。因此未来应将场景环境感知和目标函数权值相联系，形成根据场景的动态目标函数以适应变化的实际场景要求。

（2）仿真场景搭建和多模块联合仍需进一步学习。

本研究目前的算法设计和测试仍停留在算法层面，未进行真实场景下的仿真测试，未来计划将该算法整合进入仿真平台，搭建模拟真实场景进行更有实际应用价值的测试研究。

(3) 诸多模块细节需贴合实际应用。

本研究重点针对局部路径算法进行了研究，而对碰撞检测模块中的汽车采用为质点设置安全碰撞范围的方式，设置的动力学约束仅限于基本的速度、加速度和曲率范围等，未将汽车运动模型整合进入算法设计。

参考文献

- [1] 周林洁.我国乘用车产业发展与经济增长关系研究[J].西北师大学报(社会科学版),2013,第 50 卷(6): 126-132.
- [2] Sun P, Kretschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: Waymo open dataset[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 2446-2454.
- [3] Shadrin S S, Ivanova A A. Analytical review of standard Sae J3016 «taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles» with latest updates[J]. Avtomobil'. Doroga. Infrastruktura., 2019 (3 (21)): 10.
- [4] Bertozzi M, Bombini L, Broggi A, et al. VIAC: An out of ordinary experiment[C]. 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2011: 175-180.
- [5] 申泽邦,雍宾宾,周庆国,李良,李冠憬编.无人驾驶原理与实践[M].北京:机械工业出版社,2019 .
- [6] Paden B, Čáp M, Yong S Z, et al. A survey of motion planning and control techniques for self-driving urban vehicles[J]. IEEE Transactions on intelligent vehicles, 2016, 1(1): 33-55.
- [7] Badue C, Guidolini R, Carneiro R V, et al. Self-driving cars: A survey[J]. Expert Systems with Applications, 2021, 165: 113816.
- [8] Katrakazas C, Quddus M, Chen W H, et al. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions[J]. Transportation Research Part C: Emerging Technologies, 2015, 60: 416-442.
- [9] Pomerleau D A. Alvin: An autonomous land vehicle in a neural network[J]. Advances in neural information processing systems, 1988, 26(3): 450-463.
- [10] Muller U, Ben J, Cosatto E, et al. Off-road obstacle avoidance through end-to-end learning[J]. Advances in neural information processing systems, 2005, 18: 32-53.
- [11] Marchese F M. Multiple mobile robots path-planning with mca[C].International Conference on Autonomic and Autonomous Systems (ICAS'06). IEEE, 2006,70: 56-56.
- [12] 中国《人工智能标准化白皮书 2018》发布[J].智能建筑,2018,(2): 11.
- [13] Dickmanns, Ernst D. Vehicles capable of dynamic vision: a new breed of technical beings?[J].Artificial Intelligence ,1998,Vol.103: 49-76.
- [14] Phillip Bantz. This ain't no hands across America[J].North Carolina Lawyers Weekly,2017, 17(3): 59-80.
- [15] Buehler M .The 2005 DARPA grand challenge the great robot race[J].The Industrial Robot,2008,Vol.35(6): 48-77.
- [16] Buehler M . The DARPA urban challenge: autonomous vehicles in city traffic[M]. springer, 2009, Vol.56: 44-49.
- [17] Xin J, Wang C, Zhang Z, et al. China future challenge: Beyond the intelligent vehicle[J]. IEEE Intell. Transp. Syst. Soc. Newslett, 2014, 16(2): 8-10.
- [18] Pietro Cerri. Computer vision at the hyundai autonomous challenge[A].2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)[C],2011, 2011(2): 8-10.
- [19] Broggi A, Cerri P, Felisa M, et al. The vislab intercontinental autonomous challenge: an extensive test for a platoon of intelligent vehicles[J]. International Journal of Vehicle Autonomous Systems, 2012, 10(3): 147-164.
- [20] Broggi A, Cerri P, Debattisti S, et al. PROUD—Public road urban driverless-car test[J]. IEEE

- Transactions on Intelligent Transportation Systems, 2015, 16(6): 3508-3519.
- [21] Ziegler J, Bender P, Schreiber M, et al. Making bertha drive—an autonomous journey on a historic route[J]. IEEE Intelligent transportation systems magazine, 2014, 6(2): 8-20.
- [22] Bertozzi M, Bombini L, Broggi A, et al. VIAC: An out of ordinary experiment[C]. 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2011: 175-180.
- [23] Birdsall M. Google and ITE: The road ahead for self-driving cars[J]. Institute of Transportation Engineers. ITE Journal, 2014, 84(5): 36.
- [24] Eisler M N. A Tesla in every garage?[J]. IEEE Spectrum, 2016, 53(2): 34-55.
- [25] González D, Pérez J, Milanés V, et al. A review of motion planning techniques for automated vehicles[J]. IEEE Transactions on intelligent transportation systems, 2015, 17(4): 1135-1145.
- [26] Paden B, Čáp M, Yong S Z, et al. A survey of motion planning and control techniques for self-driving urban vehicles[J]. IEEE Transactions on intelligent vehicles, 2016, 1(1): 33-55.
- [27] Campbell M, Garcia E, Huttenlocher D, et al. Team Cornell: technical review of the DARPA urban challenge vehicle[J]. DARPA Urban Chall. Tech. Pap, 2007, 69:43-45.
- [28] Ferguson D, Howard T M, Likhachev M. Motion planning in urban environments[J]. Journal of Field Robotics, 2008, 25(11 - 12): 939-960.
- [29] LaValle S M, Kuffner Jr J J. Randomized kinodynamic planning[J]. The international journal of robotics research, 2001, 20(5): 378-400.
- [30] Karaman S, Frazzoli E. Optimal kinodynamic motion planning using incremental sampling-based methods[C].49th IEEE conference on decision and control (CDC). IEEE, 2010:, 1(1):7681-7687.
- [31] 胡小平, 彭涛, 左富勇. 一种基于多项式和 Newton 插值法的机械手轨迹规划方法[J]. 中国机械工程, 2012, 23(24): 2946.
- [32] Kelly A, Nagy B. Reactive nonholonomic trajectory generation via parametric optimal control[J]. The International Journal of Robotics Research, 2003, 22(7-8): 583-601.
- [33] Nagy B, Kelly A. Trajectory generation for car-like robots using cubic curvature polynomials[J]. Field and Service Robots, 2001, 11: 479-490.
- [34] 李昊天.点焊机器人轨迹能耗模型及其优化算法研究[J]. 中国机械工程, 2016, 27(01): 14.
- [35] 谭波,罗均,罗雨松,胡春晖,卓俊康,白泽鑫,田金涛.改进 RRT 算法的机器人路径规划[J].重庆大学学报,2023,第 46 卷(9): 13-22.
- [36] 顾川,郭道省,吴彬彬.基于凸优化粒子群算法的在线航迹规划[J].数据采集与处理, 2023, 38(5):1180-1190.
- [37] Xu W, Wei J, Dolan J M, et al. A real-time motion planner with trajectory optimization for autonomous vehicles[C]//2012 IEEE International Conference on Robotics and Automation. IEEE, 2012: 2061-2067.
- [38] 冯招松.基于动态规划和二次规划的无人车局部路径规划和控制研究[D].重庆理工大学,2023.
- [39] 张艺怀.自动驾驶车辆路径规划与跟踪控制算法研究[D].华南理工大学,2022.
- [40] 魏民祥,滕德成,吴树凡.基于 Frenet 坐标系的自动驾驶轨迹规划与优化算法[J].控制与决策,2021,第 36 卷: 815-824.
- [41] Frew E W, Lawrence D A, Morris S. Coordinated standoff tracking of moving targets using Lyapunov guidance vector fields[J]. Journal of guidance, control, and dynamics, 2008, 31(2): 290-306.
- [42] Piazzzi, A.*;Lo Bianco, C.G.*;Bertozzi, M.*;Fascioli, A.*;Broggi, A.*.Quintic G2-Splines for the iterative steering of vision-based autonomous vehicles(Article)[J].IEEE Transactions on

- Intelligent Transportation Systems,2002,Vol.3(1): 27-36.
- [43] Glaser S, Vanholme B, Mammar S, et al. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction[J]. IEEE Transactions on intelligent transportation systems, 2010, 11(3): 589-606.
- [44] 王沙晶.基于 Frenet 坐标系采样的自动驾驶轨迹规划算法研究[D].兰州理工大学,2019.
- [45] 李胜琴,丁雪梅.基于五次多项式的智能车辆轨迹规划[J].江苏大学学报(自然科学版),2023,第 44 卷(4): 392-398.
- [46] Petrov P, Nashashibi F. Modeling and nonlinear adaptive control for autonomous vehicle overtaking[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(4): 1643-1656.
- [47] 马宇豪,梁雁冰.一种基于六次多项式轨迹规划的机械臂避障算法[J].西北工业大学学报,2020,第 38 卷(2): 392-400.
- [48] 姚梦飞.机场航空器场面轨迹预测和路径规划关键技术研究[D].电子科技大学,2018.
- [49] Schwarzenegger A. California driver handbook [J]. Department of Motor Vehicles, 2007, 12(03): 23-33.
- [50] 谭波,罗均,罗雨松,胡春晖,卓俊康,白泽鑫,田金涛.改进 RRT 算法的机器人路径规划[J].重庆大学学报,2023,第 46 卷(9): 13-22.
- [51] 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. IEEE, 2010: 987-993.
- [52] 逯东,池毅.《中国制造 2025》与企业转型升级研究[J].产业经济研究,2019,(5): 77-88.

致谢

写到这里，内心已是百感交集。我兴奋于爬过了人生的一座高山，迫不及待将这种人生感受分享；我伤感于这是我写给大学的最后一章，写完这章，四年的旅途也就画上了句号；我困惑于在这完全属于我的一部分中，想说的很多分别时却总也说不明白了。

首先我要感谢父母及家人。四年离家，父母始终是最耐心的倾听者和最坚实的后盾，他们总能给予我无条件的关心和爱护。每次回家都是一次精神的治愈，让我有勇气再次踏上旅程去面对未知的挑战。

其次，我要感谢吉林大学对我四年的培养。似乎是人生的玩笑，曾经幻想去往温暖湿润的岭南的我意外来到了坐落于北国春城的吉林大学。每一次在寒冷的冬夜走在回宿舍的路上，我都在无数次回想着当时这个意外的相遇。但渐渐的，我感受到了吉大带给我的力量：他给我提供了一个窥见世界的窗口，为我解读了科技的奇妙，更为我提供了广阔的平台，让我在这里收获了丰富多彩的人生体验。

然后，我想感谢在吉大遇到的每一位师长。首先我要感谢我的学业导师许男老师，许老师不仅用其广博的知识去指导我迈入科研的大门，还无私地在我学习生活中提供了宝贵的建议和帮助，许老师博学谦逊亲和的为人永远是我学习的榜样。其次我要感谢秦也辰老师课题组、庄晔老师和吴西涛博士在我毕业设计选题及实施上提供的帮助，经过几个月的科研锻炼，让我体会到了科研生活的酸甜苦辣，这将是我未来研究生生活宝贵的财富。最后，我想感谢在这四年里遇到的每一位帮助我成长的学长学姐，是他们这种不计回报的帮助和不厌其烦的回答让我实现了曾经遥不可及的梦想。

我还想感谢在这四年里遇见和陪伴着我的朋友和同学们。首先要感谢我大学四年的室友们，四年中来自全国各地的我们相互包容也互相感染着，有朋自远方来又各赴远方，希望每一个 109 的伙伴都能在各自的领域发光发热。其次我要感谢大学认识的朋友们，我们亦或是有共同的爱好又或是有共同的目标，在生活中我们欢声笑语，在学习中我们相互激励，这珍贵的情谊是我大学最宝贵的财富。最后我想感

谢从小到大一直在我身边的朋友们，无论是屏幕前的联络还是每次寒暑假回家后的相聚，都让我身处异乡却感受到了家乡的温暖，希望未来我们能不被生活消磨了热情，依然能常回家聚聚。

最后我想感谢长春这座城市，四年在长春的生活让我感受到了这座城市的热情，这份热爱流淌在碧波荡漾的南湖，充斥于热情奔放的南岭体育场，飞舞在冬日漫天的鹅毛大雪。如今我就要告别这座城市，这份对长春的记忆和情感将是我此生不会忘记的人生旅程。

南岭的杏花谢了再开，校园的人们走了又来。我是吉大这辉煌史书中那匆匆几页的经历者。合上这本书，我或许很久不会再去翻看，但我知道那段经历、那份感受一直在那儿。

