

分类号: U461.6

学校代码: 10363

密 级: 公开

学 号: 2220110108



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题目 功能型无人车路径规划
与跟踪控制算法研究

论文作者 许柳柳

校内导师 杨爱喜(教授)

校外导师 马露杰(博士)

专业学位类别 机械

研究方向(领域) 无人驾驶

论文提交日期: 2025 年 5 月 20 日

分类号: U461.6

单位代码: 10363

密 级: 公开

学 号: 2220110108

功能型无人车路径规划与跟踪控制算法研究

RESEARCH ON PATH PLANNING AND TRACKING CONTROL ALGORITHM OF FUNCTIONAL UNMANNED VEHICLE

学 生 姓 名: 许柳柳

指 导 教 师: 杨爱喜 (教授)

校 外 导 师: 马露杰 (博士)

专 业: 机械

研 究 方 向: 无人驾驶

论文提交日期: 2025 年 5 月 10 号

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：许柳柳

日期：2025 年 5 月 20 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，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，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文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学位论文。

保密 ☐，在 年解密后适用本版权书。

本学位论文属于

不保密 ☐。

学位论文作者签名： 许柳柳

指导教师签名： 杨松

日期： 2025 年 5 月 20 日

日期： 2025 年 5 月 20 日

功能型无人车路径规划与跟踪控制算法研究

摘 要

随着全球城市化进程的加速，交通拥堵、电商爆发式增长产生的“最后一公里”配送的自动化需求等问题的出现，功能型无人车可以有效解决这些问题。在汽车智能化领域的前沿探索中，自动驾驶技术展现出多维度的应用价值。其中路径规划和轨迹跟踪控制是无人车导航的核心，针对城市化进程中交通拥堵与物流末端配送自动化需求，本研究围绕功能型无人车的路径规划与轨迹跟踪控制展开系统性研究，主要工作包括：

(1) 首先针对全局路径规划 A*算法遍历节点多、转折点多、搜索效率低和无法躲避动态障碍物等问题，通过优化 A*算法的代价函数，优化后的算法遍历节点明显减少，缩短了规划路径的长度，提高了寻路效率。通过设置安全距离，路径转折点减少，规划路径更加平滑，使得无人车在行驶过程中更加安全平稳地躲避障碍物，顺利到达目标位置。而局部路径规划 DWA 算法适用于动态环境，可以有效躲避动态障碍物，算法的局限性是容易陷入局部最优，无人车可能无法顺利到达指定位置而寻路失败，考虑到两种算法的优势和局限性，将两种算法进行融合。融合算法解决了 A*算法无法躲避动态障碍物和 DWA 算法易陷入局部最优的问题，同时又提高了无人车的行驶稳定性和安全性。

(2) 建立车辆三自由度模型，构建了基于 LTV-MPC 轨迹跟踪控制器。由于 LTV-MPC 控制器在车速和路径曲率等变化时，轨迹跟踪的精确性和稳定性降低，为了提高无人车的跟踪精度和行驶稳定性。提出一种基于 LTV-MPC 的轨迹跟踪优化策略，以车辆前轮转角作为控制输入，设计融合 T-S 模糊推理的动态权重调节机制。通过实时采集横向位移与横摆角误差参数作为模糊系统输入变量，构建具有在线自适应能力的权重矩阵优化框架。实现了目标函数中轨迹跟踪精度与车辆横向稳定性指标的动态平衡，有效提升了控制系统的工况适应能力。在 Matlab/Simulink 搭建模型中加入本文设计的 T-S 模糊控制器，通过双移线工况仿真，对比分析 LTV-MPC 控制和 T-S FMPC 控制，仿真结果可知，T-S FMPC 控制器有效提高了轨迹跟踪精度和稳定性。

(3) 通过线控底盘对本文规划和控制算法进行实车试验，首先介绍了实车试验的软硬件，通过激光雷达对试验场地建立点云地图，在 Autoware 中加载点云地图和其他配置进行实车试验，针对本文所涉及的路径规划算法与轨迹跟踪控制算法的可行性进行验证，试验结果表明，本文所设计的规划算法能够躲避障碍物顺利到达目标位置，并且跟踪控制算法能够较好地跟踪规划轨迹。

关键词： 功能型无人车，路径规划，轨迹跟踪，T-S 模糊控制，模型预测控制，实车试验

RESEARCH ON PATH PLANNING AND TRACKING CONTROL ALGORITHM OF FUNCTIONAL UNMANNED VEHICLE

ABSTRACT

With the acceleration of global urbanization, issues such as traffic congestion and the automated “last-mile” delivery demand generated by the explosive growth of e-commerce have emerged. Functional unmanned vehicles can effectively address these challenges. In the frontier exploration of automotive intelligence, autonomous driving technology demonstrates multi-dimensional application values. Among them, path planning and trajectory tracking control are the core of unmanned vehicle navigation. Aiming at the traffic congestion in urbanization and the automated demand for logistics terminal distribution, this study systematically investigates the path planning and trajectory tracking control of functional unmanned vehicles. The main work includes:

(1) Firstly, addressing issues such as excessive node traversal, numerous turning points, low search efficiency, and inability to avoid dynamic obstacles in the global path planning A* algorithm, the cost

function of the A* algorithm is optimized. The optimized algorithm significantly reduces traversed nodes, shortens the planned path length, and improves pathfinding efficiency. By setting a safety distance, the number of path turning points is reduced, resulting in a smoother planned path, enabling the unmanned vehicle to avoid obstacles more safely and smoothly during driving and reach the target location successfully. On the other hand, the local path planning DWA algorithm is suitable for dynamic environments and can effectively avoid dynamic obstacles. However, its limitation is that it is prone to falling into local optima, and the unmanned vehicle may fail to reach the designated location. Considering the advantages and limitations of both algorithms, they are fused. The fusion algorithm solves the problem that the A* algorithm cannot avoid dynamic obstacles and the DWA algorithm is prone to falling into local optima, while also improving the driving stability and safety of the unmanned vehicle.

(2) A three-degree-of-freedom vehicle model is established, and a trajectory tracking controller based on LTV-MPC is constructed. Since the accuracy and stability of trajectory tracking by the LTV-MPC controller decrease with changes in vehicle speed and path curvature, a trajectory tracking optimization strategy based on LTV-MPC is proposed to improve the tracking accuracy and driving stability of the unmanned vehicle. Taking

the front wheel angle of the vehicle as the control input, a dynamic weight adjustment mechanism integrating T-S fuzzy inference is designed. By collecting lateral displacement and yaw angle error parameters in real-time as input variables of the fuzzy system, an optimization framework for the weight matrix with online adaptive capability is constructed. This achieves a dynamic balance between trajectory tracking accuracy and vehicle lateral stability indicators in the objective function, effectively enhancing the adaptability of the control system to operating conditions. By adding the T-S fuzzy controller designed in this paper to the model built in Matlab/Simulink and conducting double lane change simulations, a comparative analysis of LTV-MPC control and T-S FMPC control shows that the T-S FMPC controller effectively improves trajectory tracking accuracy and stability.

(3) Real vehicle tests are conducted on the planning and control algorithms using a wire-controlled chassis. The software and hardware for the real vehicle tests are first introduced. A point cloud map of the test site is created using lidar, and the map and other configurations are loaded in Autoware for real vehicle tests. The feasibility of the path planning algorithm and trajectory tracking control algorithm involved in this paper is verified. The test results show that the designed planning algorithm can avoid obstacles and reach the target location smoothly, and the tracking

control algorithm can track the planned trajectory well.

KEY WORDS: functional unmanned vehicle, path planning, trajectory tracking, T-S fuzzy control, model predictive control, real vehicle test

目录

摘 要.....	I
ABSTRACT.....	III
目录.....	VII
第 1 章 绪论.....	1
1.1 研究背景和意义.....	1
1.2 功能型无人车国内外研究现状.....	2
1.2.1 国外研究现状.....	2
1.2.2 国内研究现状.....	4
1.3 路径规划和轨迹跟踪算法研究现状.....	6
1.3.1 路径规划算法研究现状.....	6
1.3.2 轨迹跟踪算法研究现状.....	9
1.4 研究内容及章节安排.....	12
第 2 章 无人车模型分析及仿真平台建立.....	14
2.1 车辆运动学模型.....	14
2.2 车辆动力学模型.....	15
2.3 联合仿真平台.....	18
2.3.1 CarSim 仿真环境	18
2.3.2 Matlab/Simulink 控制模块	19
2.4 本章小结.....	20
第 3 章 基于改进 DWA 算法无人车局部路径规划	22
3.1 改进 A*算法的全局路径规划.....	22
3.1.1 传统 A*算法.....	22
3.1.2 改进 A*算法.....	23
3.1.3 改进 A*算法仿真分析.....	24
3.2 改进融合算法的局部路径规划.....	26

3.2.1 DWA 算法	26
3.2.2 改进 DWA 算法	27
3.2.3 融合算法仿真分析	29
3.3 本章小结	30
第 4 章 基于改进 LTV-MPC 的无人车轨迹跟踪控制	32
4.1 LTV-MPC 轨迹跟踪控制器设计	32
4.1.1 模型预测控制原理	32
4.1.2 车辆非线性系统线性化和离散化	33
4.1.3 输出模型构建和目标函数设计	35
4.1.4 LTV-MPC 轨迹跟踪控制算法仿真	37
4.2 改进 LTV-MPC 轨迹跟踪控制器设计	41
4.2.1 T-S 模糊控制器设计	41
4.2.2 改进 LTV-MPC 轨迹跟踪仿真分析	44
4.3 本章小结	49
第 5 章 无人车实车试验	50
5.1 实车试验平台搭建	50
5.1.1 实车硬件平台搭建	50
5.1.2 实车软件系统架构	54
5.2 实车试验结果与分析	55
5.2.1 点云地图建立	55
5.2.2 试验场景搭建与结果分析	58
5.3 本章小结	60
第 6 章 总结与展望	62
6.1 全文总结	62
6.2 研究展望	63
参考文献	64
攻读学位期间发表的学术论文目录	70
致谢	71

第 1 章 绪论

1.1 研究背景和意义

随着全球城市化进程的加速，交通拥堵问题变得越来越严重，电商爆发式增长产生的“最后一公里”配送的自动化需求^[1]以及极端环境或重复性劳动需要无人化作业等问题的出现，其中功能型无人车(如物流配送、农业作业、矿区运输、清洁卫生等领域的专用无人驾驶车辆)可以有效解决这些问题。中国政府也发布相关政策，推进无人车行业发展。其中科技部等六大部门联合出台的《关于支持建设新一代人工智能示范应用场景的通知》，明确了开展自动驾驶场景示范应用，以引导无人驾驶汽车产业的发展。在政策、技术进步和社会需求等多方面因素的推动下，中国无人驾驶汽车的应用将日益广泛，消费者对无人驾驶服务的需求也在不断增加。根据最新行业分析，中国无人驾驶汽车产业正处于快速发展阶段，2023 年市场规模已达 118.5 亿元，预计到 2029 年将实现显著增长，突破 1206.8 亿元。

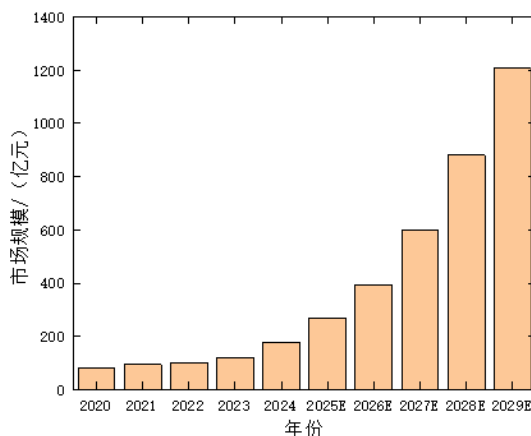


图 1-1 2020-2029 年中国无人驾驶汽车市场规模预测

随着人工智能、机器学习和传感器等关键技术的不断突破，无人驾驶技术已经取得了显著进展^[2]。功能型无人车的商业化应用场景在物流配送、环卫清扫、安防巡逻、移动零售、救援侦察和移动充电等领域逐渐落地，市场需求不断增加，推动了产业的迅速发展。它们不仅是技术的体现，更是新时代城市智慧化、现代化发展的重要推动力。其中，无人物流车的运营调度系统相当于建立了一个城市级的自动导引车（Automated Guided Vehicle, AGV）搬运系统，在夜间道路空闲

时,全自动化的物流车如“城市传送带”一样将各分拣中心的货物自动传送到各驿站的场景,从而减少能源消耗和交通拥堵^[3,4]。无人观光车的出现,增强了旅游景区的吸引力,游客不仅能深入体验当地风情,也能享受科技带来的全新旅游体验。无人环卫车能够通过无人驾驶技术提高清扫效率,特别是高铁站以及机场等清扫面积大的区域,无人清扫车可以实现全天不间断作业,缓解环卫工人的作业压力。随着新能源汽车数量的增加,无人移动充电车有效解决了高峰时段充电难、找桩困难和充电速度慢等问题,提升了充电桩的利用率。随着城市化进程的加速,传统巡逻方式难以满足日益复杂的安全需求。人工巡逻存在巡逻盲区和安全风险,而无人巡逻车能够全天候、不受环境因素影响地进行巡逻,大幅提高了巡逻效率和覆盖范围,保障了城市的安全。

综上所述,功能型无人车不仅在提高交通安全、出行效率、共享出行等方面具有重要意义,同时也在推动技术创新和行业进步方面发挥着深远的作用。无人车的感知技术、决策规划技术、跟踪控制技术等核心技术是其能够自动驾驶的关键^[5,6]。路径规划与轨迹跟踪算法的性能直接决定了规划路径的优化程度和跟踪精度,进而对无人车的整体性能、行驶稳定性和道路安全性产生深远影响。本文旨在通过改进 A*算法与动态窗口算法相结合的路径规划方法,使无人车能够高效地避开未知动态障碍物,并确保其能够沿着规划路径安全平稳地抵达目标位置。为了进一步提升无人车的行驶稳定性和轨迹跟踪精度,本文提出通过 T-S(Takagi-Sugeno)模糊控制方法对模型预测控制目标函数的权重进行在线优化。此方法可以动态调整权重矩阵,以有效协调轨迹跟踪精度与行驶稳定性之间的关系,提升控制算法在不同驾驶工况下的自适应能力,从而满足无人车在复杂道路环境中的稳定性需求。

1.2 功能型无人车国内外研究现状

1.2.1 国外研究现状

功能型无人车作为自动驾驶技术的重要应用分支,近年来在国内外均呈现快速发展态势。丰田 e-Palette 的实车于 2019 年在东京车展上亮相,在 2021 年“东京 2020 年奥运会及残奥会”期间作为巡回巴士在选手村内运行,并得到了稳步发展。丰田 e-Palette 与普通的汽车不同,e-Palette 是一款可以满足不同需求和应用场景的多功能无人车,其功能设计与无人驾驶平台结合,可以提供清洁、互联、

安全和节能的出行体验，如图 1-2 所示。

Nuro 是一家成立于 2016 年的硅谷自动驾驶公司。Nuro 的第三代无人配送车是专为小型货物（如生鲜和外卖等）配送设计，结合了先进的无人驾驶技术和传感器，能够在复杂城市环境中自主导航，避障、识别交通信号并精准送达。该车体积小，适合城市街道，特别适用于生鲜食品和包裹等物品的快速配送。同时无人车内部不设驾驶舱，结构简洁轻便，进一步提高了安全性与效率，如图 1-3 所示。



图 1-2 丰田 e-Palette 多功能无人车

图 1-3 Nuro Third generation 无人配送车

PIX Moving 成立于美国硅谷，专注于基于滑板底盘的模块化智能车辆研发。自 2017 年启动以来，公司核心技术为可定制尺寸的无人驾驶超级底盘，能够灵活适配各种应用场景。公司已成功推出包括 Robobus 和无人清扫车在内的多款创新产品。Robobus 是一款无人驾驶观光车，专为城市接驳和游客观光设计。其模块化设计支持不同的乘客需求，能够容纳多人并提供舒适安全的体验。PIX Moving 的无人清扫机器人是一款结合自动驾驶和清洁技术的设备，主要用于公共场所如街道、商场和机场的清洁，配备先进的传感器和 AI 系统，能智能识别需要特别清扫的区域，提升清洁效率，降低人工成本，推动智能化公共设施管理，如图 1-4 和 1-5 所示。



图 1-4 无人观光车 Robobus

图 1-5 无人清扫机器人

1.2.2 国内研究现状

功能型无人车在物流、农业、公共交通等领域加速落地，例如菜鸟、京东和顺丰等企业的无人配送车队已进入商业化试点阶段。政府推动“车路云一体化”试点项目，2024 年发布的《自动驾驶汽车运输安全服务指南》明确了功能型无人车的测试与运营规范。同时《智能网联汽车自动驾驶数据记录系统》等国家标准将于 2026 年实施，强化数据安全与功能安全要求。

菜鸟无人车是专为端到端打造的智能物流机器人，致力于实现物流行业的数字化、智能化转型和探索 L4 级自动驾驶技术。菜鸟无人车是菜鸟网络在智能物流领域的一项重要创新，它运用了自动驾驶、大数据分析等先进技术，提高物流效率，降低运营成本，提供更灵活、快速的配送服务。整车结合数类端到端大模型，链接上下游模块，最终实现物流全链路自动化，如图 1-6 所示。

由浙江省智能网联汽车创新中心研发的移动自卸无人充电车如图 1-7，利用 GPS 和惯性导航系统等高精度定位技术，实现充电车的精确定位，结合内置的地图数据和导航算法，决策出最佳行进路径。在云端监测平台的数据支撑之下，智能决策系统能根据客户的充电需求及距离位置的远近，通过智能决策系统能够快速匹配出合适的电池柜体，并召唤就近的线控底盘将电池柜体拉往客户车边进行充电。云控平台监测到电池 SOC 值过低时，将调度无人车前往将电池柜搬运至基站进行补电，有效解决了充电难和找桩难等问题。



图 1-6 菜鸟无人车 GT Pro



图 1-7 移动自卸无人充电车

新石器无人车是一家专注于研发、制造和提供服务的载物型无人车公司。新石器无人车 X3 是一款配备高度自动化的高性能无人配送车，能够在复杂城市环境中自主行驶和进行配送，如图 1-8 所示。新石器无人车 X3 Plus 是 X3 的升级

版，具备更强的载货能力和更高效的自动驾驶性能，如图 1-9 所示。它支持多种配送需求，配备先进的 AI 算法和自主决策能力，适用于大型商业区、物流中心、校园和医疗等场景，推动无人车在智慧城市中的广泛应用。



图 1-8 新石器无人车 X3



图 1-9 新石器无人车 X3 Plus

文远知行致力于自动驾驶技术的研发和应用，特别是在智能出行领域。打造出全球首个全无人、可大规模量产的无人驾驶小巴，如图 1-10 所示。无人驾驶小巴是城市中新的出行方式，采用电动驱动系统，不仅减少了燃油消耗，还能够降低排放，不仅可以应用于景区也能够全天候在城市开放道路上行驶，为乘客提供便捷的交通服务和新的出行体验。



图 1-10 文远知行无人驾驶小巴



图 1-11 仙途智能 Autowise V3

由仙途智能研发的自动驾驶清扫车 V3，专为智能环卫量身打造。通过采用多种传感器和高精度定位技术，仙途智能的 V3 扫路机具有极高的安全性和可靠性，能够精准导航并高效完成清扫任务，是智能环卫领域的重要创新。该车具备全天候作业能力，无论白天还是夜间，都能在复杂的城市环境中稳定运行。V3 不仅提升了环卫作业的效率，还大大减少了人工成本，并推动城市公共环境管理的智能化，如图 1-11 所示。

功能型无人车的应用越来越广泛，主要是提高了道路安全、缓解交通压力、提高出行效率和体验、环保节能、提高物流与配送效率以及智能城市建设等方面。由于无人车数量越来越多，目前关于功能型无人车的管理和监测，通过智能网联

监测平台可以获取车辆的实时数据以及在城市道路中的行驶情况,实现了对无人车的高效管理和安全监测。功能型无人车国内的政策支持和市场潜力等方面具有明显优势,而国外则在核心技术积累和商业化经验上更为成熟。随着相关法规的不断完善以及技术成本的逐步降低,未来功能型无人车将在物流、农业和城市交通等领域得到更加广泛的应用,推动智慧城市建设和绿色经济的转型升级。

1.3 路径规划和轨迹跟踪算法研究现状

功能型无人车数量和应用场景越来越多,保证无人车安全行驶至关重要,而路径规划和轨迹跟踪算法是自动驾驶系统中不可或缺的部分。路径规划确保了从起点到终点的全局导航,帮助车辆选择最佳路径。轨迹跟踪则确保车辆沿着规划好的路径精确行驶,保障了行驶的安全性与稳定性。两者的协同作用使得自动驾驶车辆能够在复杂的交通环境中实现智能驾驶。

1.3.1 路径规划算法研究现状

功能型无人车的路径规划算法可以分为经典算法、基于采样的路径规划算法和智能仿生算法的路径规划算法^[7]。经典算法包括 A*、Dijkstra 算法^[8],基于采样的路径规划算法包括概率路图法(Probabilistic Road Map, PRM)^[9]与快速随机扩展树算法(Rapidly-exploring Random Tree, RRT)^[10],机器学习方法包括深度强化学习算法等,智能优化算法包括蚁群算法和遗传算法等。

(1) 经典算法

Dijkstra 算法是一种求解图中单源最短路径问题的经典方法,它采用逐步构建最短路径树的策略,来寻找从起始节点到图中其他所有节点的最短路径。该算法的优点是能确保找到最优路径,但缺点是扩展节点较多,效率较低。在移动机器人路径规划领域,GONG H^[11]等人提出了一种创新性方法,该方法将实际环境数据与几何拓扑模型相结合,构建了基于 Dijkstra 算法的路径优化框架。通过引入环境约束条件,实现了路径的平滑性和可行性。根据应用场景得到连续映射,连续映射离散化后随机生成离散点阵,并计算点间的欧几里得距离。该方法能够有效减少累积转弯角度,提高最小平均转弯角度,增加路径的平滑度,从而缩短机器人运动时间并提高工作效率。

A*算法于 1968 年提出,算法的核心是可以规划出一条最短路径,通过优化代价函数减少遍历节点数,从而提高寻路效率。A*算法在无人车的应用中也存在

冗余节点多、实时性差以及搜索效率低等问题。Xing 等^[12]提出通过替换八方向邻域为无障碍矩形边界，并从起点和目标点双向探索，减少了搜索节点数量和路径振荡，通过使代价函数自适应提升了路径的平滑性和安全性。针对传统路径搜索范围受限的问题，杨芳清^[13]等人进行了算法改进，其研究将搜索节点数量从 8 个扩展至 32 个，显著提升了路径探索的广度，但改进算法存在一定局限性，在复杂环境中，由于未充分考虑障碍物分布特征，可能导致规划路径穿越障碍区域，影响实际应用效果。

（2）基于采样的路径规划算法

RRT 算法通过随机生成扩展树，能够迅速扩展大部分空间，因此适用于高维空间，尤其适合用于动态环境和复杂约束条件下的路径规划。Zhou 等^[14]研究提出了一种增强型 RRT 路径规划算法，通过多维度优化策略提升算法性能。在传统 RRT 框架中，融入了智能采样机制与节点优化技术，显著改善了路径规划的寻路效率。算法核心改进包括三个方面，采用智能导向机制优化节点扩展过程，通过动态调整采样区域范围，有效提升搜索效率；引入自适应采样因子，根据环境复杂度实时调整采样密度，确保规划过程的灵活性；通过节点优化技术，在保证路径安全性的前提下，最大限度地缩短路径长度。孙馨宇等^[15]提出了一种改进的 Informed-RRT* 算法，该算法通过全局采样优化和自适应步长提升节点利用率。采用概率偏置的双向搜索及重选父节点的方法找到初始路径，为后续的迭代优化提供较好的初始值。在进行椭圆迭代时加入贪心策略以减少无用节点，最后对路径回溯优化减少无用节点提升路径的平直度。

（3）机器学习方法

深度强化学习算法是一种结合了深度学习和强化学习的技術，通过神经网络模型在高维和复杂任务中进行自我学习和优化，借助环境反馈优化决策，帮助机器人在复杂未知环境中实现更高水平的自主性和智能化。该算法适用于动态复杂环境，但是对硬件要求较高，参数调节比较困难。Tan 等^[16]提出了一种基于 Q 学习的完全覆盖路径规划算法，解决了传统生物启发神经网络算法中的局部最优路径问题。该算法通过强化学习引入全局环境信息，并在可达路径点变化处使用 Q 学习优化路径规划策略。

（4）智能优化算法

蚁群算法通过模拟蚂蚁觅食过程中的信息素释放、感知和更新机制，实现了一种分布式、自组织的优化搜索方法。但搜索比较盲目，搜索较慢。张浩等^[17]提出一种基于 Sigmoid 统计迭代的蚁群算法。采用 Sigmoid 激活函数分布策略，增加起点到目标点路线上信息素的初始浓度，降低算法前期搜索的盲目性；再引入自适应因子动态调节启发函数，增加蚂蚁选择全局最优节点的期望程度，降低算法的收敛时间。江明^[18]提出了一种优化的蚁群算法，通过在全局优势区域提升初始信息素浓度，显著增强了初始搜索阶段的效率。此外，该算法引入了避障机制，有效规避了盲目搜索的问题。薛翔等^[19]提出将改进的蚁群算法通过引入趋向启发函数和柯西分布函数，增强了全局搜索能力，避免了陷入局部最优。又优化了距离启发函数，通过迭代次数调整信息素挥发因子，加快算法的收敛速度。

局部轨迹规划的目标是根据无人车当前位置、速度、环境的实时信息以及运动学约束，实时地生成一条平滑、可行的轨迹。可以实时调整无人车的行驶路径，从而躲避动态障碍物或复杂的动态环境，保证了无人车在寻路过程中的安全性。本文主要介绍人工势场法和动态窗口法。

（1）人工势场法

人工势场法（Artificial Potential Field, APF）是由 Khatib 于 1985 年提出，该方法通过模拟物体间的相互作用力来为机器人创建运动轨迹，并引导机器人避开障碍物，朝着目标位置前进。该方法通常能够生成平滑且安全的路径，属于一种反馈控制策略，具有较好的鲁棒性。人工势场法存在局部最优解问题，可能使机器人难以接近或到达目标点。Fu 等^[20]提出了一种结合 RRT 算法和 APF 方法的优化策略，引入了一种基于 APF 的启发式策略，这有助于引导路径搜索过程更快地更接近目标点。其次，在路径生成过程中，使用 APF 方法对路径进行实时平滑处理，以降低路径的复杂性和执行过程中的能耗，提高了 RRT 算法的路径搜索效率，并结合 APF 的目标导向和避障能力，实现更高效、更可靠的机器人路径规划。

（2）动态窗口法

动态窗口法（Dynamic Window Approach, DWA）是一种广泛应用于移动机器人路径规划的局部避障技术，尤其在实时避障和动态环境路径规划方面表现出色^[21]。DWA 算法的核心思想是基于机器人的当前状态，计算出一个动态的控制

空间（动态窗口），然后在这个控制空间内通过评估多个候选控制命令，选择最优的命令来实现局部避障和目标导向的路径规划。它通过实时调整路径、避开障碍物并确保平滑性，适应了动态环境中的变化。蒋华扬等^[22]提出了一种基于速度采样策略的融合方向信息和距离信息的移动机器人路径规划算法。在传统 DWA 算法评价函数中增加子目标点与机器人运动方向的角偏差评估项，机器人局部路径末端点到目标点和全局路径的距离评估项，以及机器人与未知障碍物之间的距离评估项，来优化移动机器人运动路径。董晓东等^[23]把改进 A*算法和 DWA 算法融合，首先通过对栅格地图中的栅格进行量化处理，调整 A*算法中的启发函数和节点选择，已达到全局最优并及时躲避障碍物，但是未包含动态障碍物的设置，仍存在一定的不足。程传奇等^[24]通过采用对关键点选取策略，提出了路径中的多余节点并与 DWA 算法进行融合，规划路径更加平滑同时具备局部避障能力。但是所规划路径仍然有斜穿障碍物的问题，不利于无人车安全行驶。

综上所述，A*算法具有搜索效率高和规划路径短等优点，但是 A*算法在无人车的应用中也存在冗余节点多、实时性差以及搜索效率低等问题，不利于无人车的运行。DWA 算法具有躲避未知障碍物的功能，但传统 DWA 算法在较复杂的环境中距离目标点过远时，易陷入局部最优导致寻路失败，面对复杂环境下可能出现的随机障碍物，仅依靠全局最优路径规划是不够的，无人车还需要具备应对随机障碍的能力^[25]。本文将改进后的 A*算法与 DWA 算法融合，算法融合后使得无人车在躲避动态障碍物的同时顺利到达指定位置。

1.3.2 轨迹跟踪算法研究现状

轨迹跟踪控制算法是无人车实现自主导航的核心技术之一，其研究是为了确保系统能够快速、精确和鲁棒地跟踪期望轨迹。算法分为经典控制方法和智能控制方法。根据控制方法和系统模型的不同，常用的轨迹跟踪算法包括 PID 控制^[26]、纯跟踪控制^[27]、模型预测控制^[28]、线性二次调节器^[29]、神经网络控制算法等。

（1）PID 控制算法

PID 控制通过计算目标值与实际值之间的误差，并根据这个误差调整系统的控制输入，使实际输出值更接近目标值。通过调整 PID 控制器中的比例、积分和微分增益，可以优化轨迹跟踪性能，其优点是不需要建立系统模型，易于理解和

实现。斯坦福大学的研究团队^[30]提出了一种创新的横向控制系统，该系统结合了前馈与反馈控制策略。前馈控制模块利用道路曲率信息进行预判性调节，有效降低了系统稳态误差。同时，PID 反馈控制模块通过实时监测横向偏差，动态调整控制参数，实现了系统误差的快速收敛。章仁燮等^[31]提出一种基于条件积分算法的鲁棒控制策略，该方法通过基于条件积分算法设计运动学和动力学控制器，确保无人驾驶车辆在跟踪给定轨迹时的全局渐进稳定。运动学控制器解决质心位置误差和横摆角速度上界问题，而动力学控制器则应对车辆参数不确定性和执行器约束，实现横摆角速度误差的渐进稳定。Huang 等^[32]提出了一种基于传统 PID 控制器与模糊 PID 控制器关系的设计方法，并通过序列二次规划（Sequential Quadratic Programming, SQP）算法调优模糊 PID 控制器的参数，应用于线性和非线性控制系统中。

（2）纯跟踪控制算法

纯跟踪控制器是自动驾驶中常用的一种轨迹跟踪算法，原理是通过调整转向角，使车辆沿圆弧轨迹到达前方预瞄点^[33]。由于纯跟踪控制器通过几何关系实现路径跟踪，更适用于低速场景，其性能依赖于前视距离的选择和预瞄点的精确计算，实际应用中需结合自适应策略和参数调优以提升鲁棒性，不适用于高速或者复杂动力学场景^[34]。Shan 等^[35]提出了一种基于纯追踪方法的改进型 CF 追踪方法，通过使用克洛索伊德 C1 曲线替代圆形曲线来减少拟合误差，该方法采用模糊系统根据路径曲率选择前瞻距离，提高了鲁棒性和稳定性。

（3）线性二次调节器

线性二次调节器（Linear Quadratic Regulator, LQR）是一种用于设计最优状态反馈控制器的方法，特别适用于线性系统。它通过最小化一个包含状态误差和控制代价的二次型代价函数来实现控制目标，广泛应用于控制工程领域。LQR 假设系统是线性的，这对于很多非线性系统来说不适用。周卫琪等^[36]提出了一种改进的 LQR 路径跟踪横向控制策略，解决了智能车辆在弯道或狭窄区域中难以兼顾控制精度和安全性的问题。通过建立车辆系统动力学模型、引入前馈控制和车体尺寸约束，提升了车辆的路径跟踪精度。陈亮等^[37]提出了一种创新的 LQR 路径跟踪控制方法，通过优化前轮侧偏力并结合前馈与反馈控制策略，有效提升了车辆横向稳定性。该方法采用刷子轮胎模型替代传统线性模型，保留了车辆与

轮胎系统的非线性特性，显著降低了轨迹跟踪误差。

（4）神经网络控制算法

神经网络控制算法是一种利用神经网络在学习和非线性映射方面来设计和实现控制系统的方法，能够有效地控制复杂的非线性系统。它是基于神经网络的拟合能力，能够自动地从数据中学习控制策略，尤其适用于复杂、非线性或难以建模的系统。算法的核心思想是通过神经网络学习系统的动态特性，并生成控制输入使系统的输出趋于目标值。与传统控制方法不同，神经网络控制算法不依赖于精确的数学建模，而是通过大量的数据学习、识别系统的行为，并自动调整控制策略，但也存在一些挑战，如对数据的依赖性、计算资源的要求以及稳定性问题。Hamid 等^[38]提出了一种基于多约束非线性预测控制（NMPC）的自动驾驶车辆路径跟踪控制器，旨在提高瞬态性能并考虑防翻车标准。通过引入神经网络自回归模型、Frenet-Serret 几何和垂直运动模式，提出的方法有效地增强了偏航稳定性和路径跟踪性能。Sung 等^[39]提出了一种采用自递归小波神经网络的广义预测控制策略，实现移动机器人在路径跟踪过程中的稳定性，通过自适应学习率优化训练过程，所提出的控制策略通过离散李雅普诺夫稳定性定理保证系统收敛。

（5）模型预测控制算法

模型预测控制算法（Model Predictive Control, MPC）的核心在于滚动优化机制，即在每个控制周期重新计算最优控制量，仅执行序列中的第一个控制输入，随后更新系统状态并重复优化过程。该控制方法利用系统动态模型预测未来行为，并通过优化算法求解最优控制序列，仅执行序列中的第一个控制输入，随后更新系统状态并重复优化过程。这种滚动优化策略能够实时应对系统变化和不确定性，提高了控制的鲁棒性和适应性。MPC 可以显著提高系统的控制性能，尤其在复杂、非线性、时变系统中，同时能有效地处理系统的多种约束，适用于多种应用场景，包括线性 and 非线性系统。吴西涛等^[40]提出了一种基于 MPC 的轨迹跟踪控制器，结合横摆稳定性判据和贝叶斯优化方法优化控制器参数，提升了轨迹跟踪性能。Erkan 等^[41]基于轨迹跟踪误差设计，并构建了一个融合了前馈控制与鲁棒特性的线性模型预测控制器，所设计的控制器能够在建模误差和干扰的情况下，以合理的精度控制系统。陈梓宁等^[42]本文提出了一种基于模糊自适应预测时域参数的轨迹跟踪算法。首先建立了车辆运动学模型与模型预测控制器，再将实时

的车速和航向角偏差作为模糊系统的输入，通过模糊逻辑在线调整 MPC 的预测时域参数，以实现优化控制。张修伟等^[43]提出了一种当路面附着系数与车速变化时的模糊自适应轨迹跟踪控制策略。构建了七个自由度的车辆动力学模型，并基于轮胎模型和整车动力学模型，设计了无迹卡尔曼滤波估计器。又设计了以路面附着系数和车速为输入的自适应模糊控制器，用于实时调整 MPC 控制器的参数。

综上所述，MPC 是一种利用系统模型来预测未来系统行为并进行优化控制的先进方法。其核心思想是通过对系统未来状态的预测，结合优化算法来实时计算控制输入，并在有限时域内求解最优控制序列，仅实施当前时刻的控制输入，随后更新系统状态并重复优化过程。因此本文将 MPC 控制算法用于轨迹跟踪控制，实现对规划路径的跟踪，提高无人车在行驶过程中的稳定性。

1.4 研究内容及章节安排

本文主要研究功能型无人车的路径规划和轨迹跟踪控制，图 1-12 为本文研究框架图。相关的内容包括：建立了车辆的运动学和动力学模型、对传统 A* 算法进行改进并与 DWA 算法融合、基于模型预测控制算法对无人车的轨迹进行跟踪控制并提出了改进方法。本文章节安排如下：

第 1 章：本章概述了功能型无人车的研究背景与意义，分析了功能型无人车的国内外研究现状，并探讨了无人车领域常用的规划算法和控制算法的基本原理与应用。

第 2 章：本章主要是对车辆进行动力学和运动学建模，对本文需要用到的联合仿真平台进行介绍，包括 CarSim 仿真环境介绍和 Matlab/Simulink 控制模块。根据功能型无人车的尺寸大小在 CarSim 仿真平台中的车辆模型进行了配置，然后在 Simulink 中搭建模型并对各模块输入输出接口进行介绍。

第 3 章：本章主要是优化 A* 算法的启发函数，提高了搜索效率，并设置安全距离，安全距离的设置既解决了无人车斜穿障碍物的问题，又消除了路径中的冗余点，确保无人车可以安全平稳的到达目标点。首先在不同大小的栅格地图对改进前后的算法进行仿真，通过仿真数据分析对比改进后算法在遍历节点、转折角度等方面的优势，然后将改进后的 A* 算法与 DWA 算法进行融合，最后将传统 DWA 算法、融合算法和蚁群算法的仿真结果进行对比分析。由仿真结果可知，融合算法不仅可以提高搜索效率，同时在行驶过程中可与障碍物保持一定的安全

距离，不会出现无法到达目标位置的情况，保证无人车可以安全到达目标位置。

第 4 章：建立车辆三自由度模型，构建了基于 LTV-MPC 轨迹跟踪控制器。通过联合仿真平台 CarSim 和 Matlab/Simulink 对改进前的 LTV-MPC 算法进行仿真分析，总结该算法的优势和局限性并提出改进后的控制算法。对于改进 LTV-MPC 轨迹跟踪控制器设计，提出 T-S 模糊控制器优化控制算法的目标函数的权重系数，最后在联合仿真平台对改进后的 LTV-MPC 进行仿真分析，结果表明改进后的控制算法提高了跟踪精确性和稳定性，同时改善自适应能力，有效解决了传统 MPC 对于无人车的行驶稳定性会随着路面附着系数和车速的不同有着不同的影响结果的局限性。

第 5 章：本章主要通过实车平台对上文中的算法进行验证，首先，介绍了试验平台的软硬件配置，利用激光雷达构建厂区点云地图，并基于该地图实现车辆定位与障碍物检测。在厂区环境中构建测试场景，验证所提出的路径规划算法与轨迹跟踪方法的有效性和可行性。试验结果表明，本文算法能够生成最优轨迹，实现障碍物避让，并有效跟踪规划路径。

第 6 章：本章首先总结了所设计并改进的路径规划及跟踪控制算法在实际应用中的有效性和实用性，再对所提出算法的局限性以及实验过程中存在的不足之处进行深入的分析与探讨，最后给出未来应该如何解决并完善不足之处。

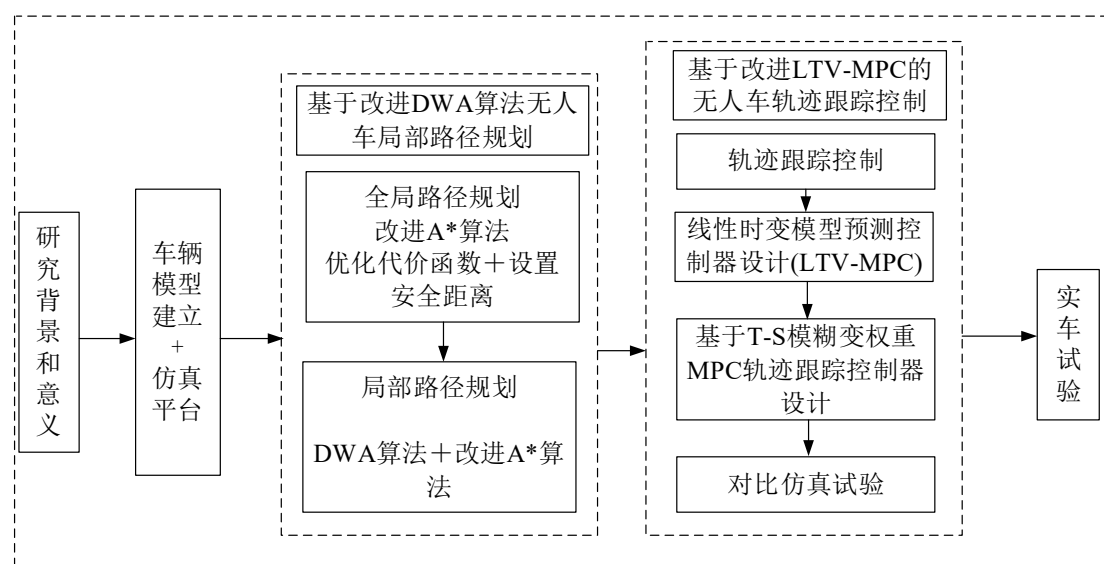


图 1-12 论文研究框架图

第 2 章 无人车模型分析及仿真平台建立

功能型无人车的路径规划与跟踪控制依赖于精确的车辆模型，以确保行驶的安全性和高效性。车辆通过简化车辆的运动约束，忽略了动力学因素，能够快速计算出可行的路径，特别适用于复杂的环境下的实时路径生成。其主要优势在于计算速度快，能够有效应对环境变化。第 3 章介绍的局部路径规划算法采用了运动学模型，以确保局部路径规划能够满足实时性的需求。车辆动力学模型可以考虑到车辆的转向、加速度、速度限制、轮胎摩擦等因素，在第 4 章的轨迹跟踪控制算法中，通过采用精确的动力学模型，可以进一步提升控制器的轨迹跟踪精度，从而更全面地理解和深入分析车辆在行驶过程中的动态表现和性能特征^[44,45]。本章通过建立运动学模型和动力学模型，在路径规划和轨迹跟踪过程中达到了效率与精度的平衡，从而提升了无人驾驶车辆的整体性能和可靠性。

2.1 车辆运动学模型

车辆运动学模型主要研究车辆几何关系和运动状态，采用车辆的前进方向、速度、航向角等信息来描述车辆的位置和姿态。该模型采用两个坐标系：惯性坐标系和车体坐标系。惯性坐标系用于惯性导航系统，车辆运动状态通过车体坐标系进行描述，其 X、Y、Z 轴分别指向东、北和向上。为了减少复杂性，本文采用单车模型如图 2-1 所示，并做出以下假设：（1）车辆的运动由前轮的转角和车辆的当前速度控制；（2）假设车轮是刚性的，且忽略了轮胎的侧滑效应；（3）车辆模型通常使用的是单车模型，将车辆视为一个简化的模型，只考虑车辆前后两组车轮的运动。其中 A、B 分别为前后轮的中心点； δ_f 和 δ_r 分别是前轮和后轮转角；a 和 b 分别是车辆质心 C 到前后轮的距离，轴距是 L；车辆的轨迹半径 R；V 是车辆质心的速度， β 为车辆质心侧偏角；其中X、Y分别代表车辆质心的横、纵向位置， ϕ 为车身横摆角。

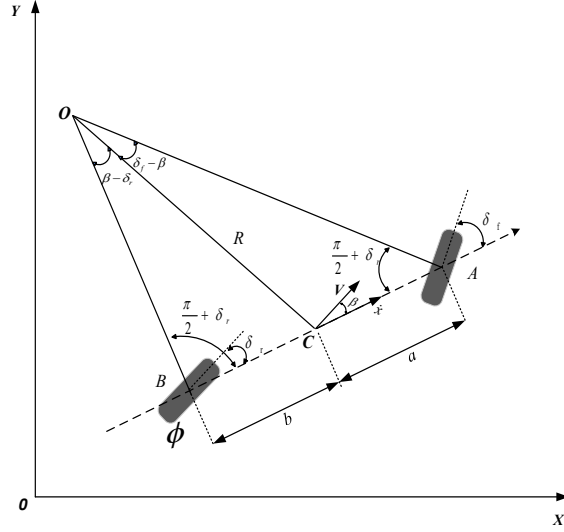


图 2-1 单车模型图

在图 2-1 的三角形 OAC、OBC 中，由根据正弦定理得：

$$\frac{\sin(\delta_f - \beta)}{a} = \frac{\sin(\frac{\pi}{2} - b)}{R} \quad (2-1)$$

$$\frac{\sin(\beta - \delta_r)}{b} = \frac{\sin(\frac{\pi}{2} - \delta_r)}{R} \quad (2-2)$$

由几何知识及条件假设，化简后的姿态信息：

$$\dot{\phi} = \frac{V \cos(\beta)}{a+b} \{ \tan(\delta_f) - \tan(\delta_r) \} \quad (2-3)$$

由于轴距 $L=a+b$ 远远小于轨迹半径 R ， β 接近于 0。一般车辆模型为前轮转向，故 δ_r 为 0，所以等式 (2-3) 可以近似为：

$$\dot{\phi} = \frac{V}{L} \tan(\delta_f) \quad (2-4)$$

又根据单车模型的位置信息得：

$$\begin{cases} \dot{X} = V \cos(\beta + \phi) \\ \dot{Y} = V \sin(\beta + \phi) \end{cases} \quad (2-5)$$

假设车辆重心点位于后轴，则 $\beta \approx 0$ ，则由上式 (2-1)、式 (2-2)、式 (2-4) 和式 (2-5) 可得如下其微分方程形式：

$$\begin{cases} \dot{X} = V \cos(\phi) \\ \dot{Y} = V \sin(\phi) \\ \dot{\phi} = \frac{V \tan(\delta_f)}{L} \end{cases} \quad (2-6)$$

将车辆运动学模型应用于第 3 章的局部路径规划算法，作为 DWA 算法的模型来生成可行的速度空间，并预测轨迹。

2.2 车辆动力学模型

基于 MPC 轨迹跟踪研究车辆的操控稳定性,核心在于深入分析车辆的纵向、横向及横摆动力学特性。由于运动学模型仅关注车辆的位置和速度等因素,而动力学模型更精确的模拟复杂的动力学特性,对于第 4 章的轨迹跟踪控制算法需要保证车辆的跟踪精度和行驶稳定性。高自由度的系统可能更容易受到外部扰动或不确定性的影响,难以维持精确的控制,尤其是在动态环境中。因此本文选择车辆三自由度动力学模型。为了使模型更适用于实际应用,作出以下五点假设:(1)假设轮胎的纵向力和横向力彼此独立,不考虑轮胎在不同方向上的耦合效应;(2)不考虑车辆的车身侧向倾斜和上下垂直振动;(3)假设悬架系统是刚性的,忽略车身悬架特性;(4)忽略车身的横摆惯性;(5)忽略空气阻力和外部扰动。理想化假设后的车辆的动力学模型如图 2-2 所示。 a 和 b 分别是车辆质心 O 到前后轮的距离; F_{cf} 、 F_{cr} 分别为前、后轮胎受到的横向力; F_{lf} 、 F_{lr} 分别为前、后轮胎受到的纵向力。

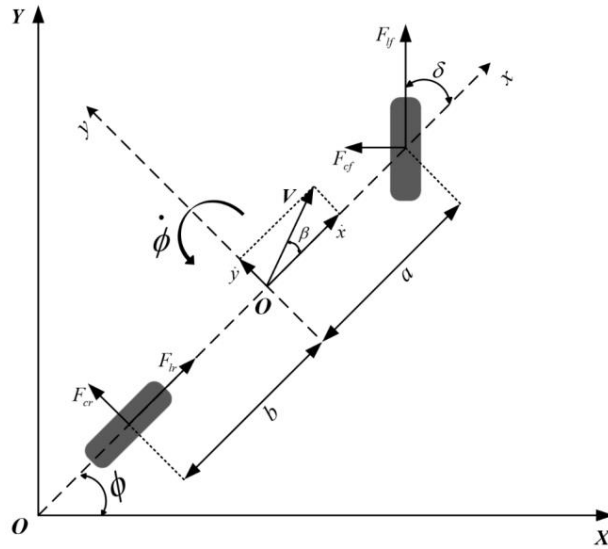


图 2-2 车辆动力学模型

基于牛顿第二定律,分别为车身沿 X 轴、Y 轴的受力平衡和绕 Z 轴的转矩平衡建立方程:

$$\begin{cases} m\ddot{x} = m\dot{y}\dot{\phi} + 2(F_{lr} + F_{lf}\cos\delta_f - F_{cf}\sin\delta_f) \\ m\ddot{y} = -m\dot{x}\dot{\phi} + 2(F_{cr} + F_{lf}\sin\delta_f + F_{cf}\cos\delta_f) \\ I_z\ddot{\phi} = 2a(F_{lf}\sin\delta_f + F_{cf}\cos\delta_f) - 2bF_{cr} \end{cases} \quad (2-7)$$

式(2-7)中： m 为车辆总质量； I_z 为车身绕 Z 轴转动时的转动惯量； δ_f 为车辆前轮转角； ϕ 为车辆的横摆角； $\ddot{\phi}$ 为车辆的横摆加速度； $\ddot{x}$ 、 $\ddot{y}$ 分别为车辆纵向和横向的加速度。

当轮胎的侧偏角较小时，前后轮胎所受的横向力，纵向力与侧偏角的变化关系可近似描述为线性函数。前后轮的横向力分别为：

$$\begin{cases} F_{cf} = C_{cf} \left(\delta_f - \frac{\dot{y} + a\dot{\phi}}{\dot{x}} \right) \\ F_{cr} = C_{cr} \frac{b\dot{\phi} - \dot{y}}{\dot{x}} \end{cases} \quad (2-8)$$

式(2-8)中： C_{cf} 为前轮侧偏刚度、 C_{cr} 为后轮侧偏刚度。

车辆前轮、后轮的纵向力分别为：

$$\begin{cases} F_{lf} = C_{lf} S_f \\ F_{lr} = C_{lr} S_r \end{cases} \quad (2-9)$$

式(2-9)中： C_{lf} 分别为前轮纵向刚度， C_{lr} 分别为后轮纵向刚度； S_r 为前轮的滑移率； S_f 为后轮的滑移率。

然后将车身坐标系与惯性坐标系之间进行转换：

$$\begin{cases} \dot{Y} = \dot{x} \sin \phi + \dot{y} \cos \phi \\ \dot{X} = \dot{x} \cos \phi - \dot{y} \sin \phi \end{cases} \quad (2-10)$$

式(2-10)中： $\dot{X}$ 、 $\dot{Y}$ 为车辆在惯性坐标系下的纵向速度和横向速度。

联合上述式并建立式(2-11)：

$$\begin{cases} m\dot{y} = -m\dot{x}\phi + 2 \left[C_{cf} \left(\delta_f - \frac{\dot{y} + a\dot{\phi}}{\dot{x}} \right) + C_{cr} \left(\frac{b\dot{\phi} - \dot{y}}{\dot{x}} \right) \right] \\ m\ddot{x} = m\dot{y}\phi + 2 \left[C_{lf} S_f - C_{cf} \left(\delta_f - \frac{\dot{y} + a\dot{\phi}}{\dot{x}} \right) \delta_f + C_{lr} S_r \right] \\ I_z \ddot{\phi} = 2 \left[a C_{cf} \left(\delta_f - \frac{\dot{y} + a\dot{\phi}}{\dot{x}} \right) - b C_{cr} \left(\frac{b\dot{\phi} - \dot{y}}{\dot{x}} \right) \right] \\ \dot{Y} = \dot{x} \sin \phi + \dot{y} \cos \phi \\ \dot{X} = \dot{x} \cos \phi - \dot{y} \sin \phi \end{cases} \quad (2-11)$$

将式(2-11)转换成控制系统状态空间方程的形式：

$$\begin{cases} \dot{\xi}_{(t)} = f(\xi_{(t)}, u_{(t)}) \\ \eta_{(t)} = C \xi_{(t)} \end{cases} \quad (2-12)$$

式(2-12)中系统状态量 $\xi_{(t)} = [\dot{y}, \dot{x}, \phi, \dot{\phi}, Y, X]$ ；系统控制量 $u_{(t)} = [\delta_f]$ ；输出矩阵

$\eta_{(t)} = [\phi, Y]$ ；输出矩阵 $C = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$ 。

在车辆行驶过程中，车辆的动力学特性会影响轨迹的跟踪精度，将车辆三自由度动力学模型应用于第 4 章的跟踪控制算法 MPC 中，通过模型预测系统未来一段时间内的状态，并通过优化问题得到最优的控制输入，保证车辆的跟踪精度。

2.3 联合仿真平台

2.3.1 CarSim 仿真环境

CarSim 是一款专业的车辆动力学仿真软件，通过高精度建模和实时仿真能力，分析车辆在各种场景下的动态响应。CarSim 利用高度精确的物理模型来模拟车辆的动态表现，能够模拟不同类型的道路环境，同时还能够考虑各种路面条件及天气变化，为提供了真实且可靠的仿真结果。CarSim 可与 Matlab/Simulink 进行集成，支持对控制系统和自动驾驶算法的开发与测试。还提供了直观的图形化界面和详细的输出数据，可以轻松地设置仿真场景、调整参数并实时查看仿真结果。主界面如图 2-3 所示，分为三部分：左侧区域用于设置车辆的基本参数；中间区域配置仿真参数，并与控制模型进行接口对接；右侧区域用于查看仿真结果、动画及曲线。

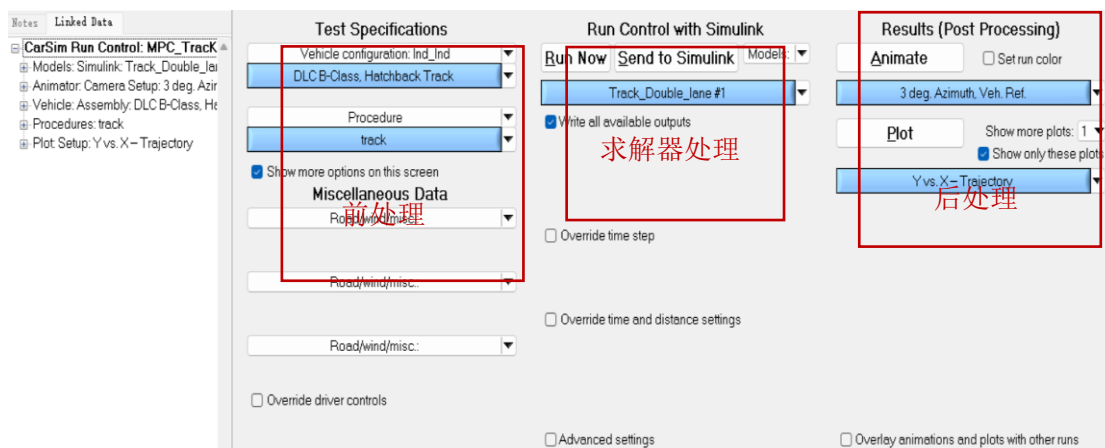


图 2-3 CarSim 主界面

本文研究的是功能型无人车，考虑到车辆的尺寸大小，参照日本 Deriro 物流机器人的外形尺寸参数，对 CarSim 仿真平台中的车辆模型参数进行了配置，如图 2-4 所示。通过调整轴距、轮距和质心高度等关键参数，构建了符合功能型无人车的车辆参数，如表 2-1 所示。

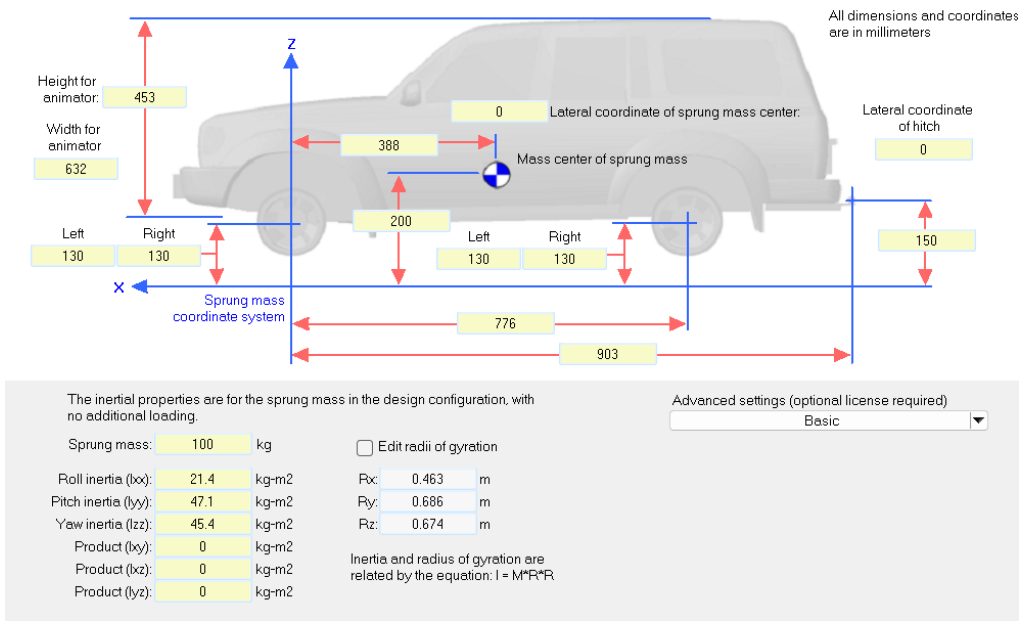


图 2-4 无人车参数设置界面

表 2-1 车辆建模具体参数

车辆参数	数值
车辆轴距	776/1100 mm
前轴距离质心长度	388 mm
后轴距离质心长度	388 mm
质心高度	120 mm
整车质量	160 kg
横摆转动惯量	45.4 kg·m ²

2.3.2 Matlab/Simulink 控制模块

Simulink 是一个图形化建模与仿真工具，可以通过拖拽模块并连接信号，支持对系统进行深入的性能分析，从而更高效地进行建模与优化。本文在 Simulink 搭建的模型如图 2-5 所示，首先建立 Double_line 和模糊控制模块，Double_line 控制模块需要接收来自 CarSim 的车辆状态参数，包括车辆的位置、速度、角度等。Double_line 的控制模块的轨迹跟踪算法进行计算期望的前轮转角，并将该转角传递给 CarSim 中的车辆模型，以实现轨迹跟踪控制。各模块的具体功能说明见表 2-2。



编号	名称	输入接口	输出接口
①	基于 MPC 的 Double_line 模块 模糊控制模块	车辆状态参数 模糊输出变量	前轮转角 模糊输出变量
②	CarSim 输出模块	CarSim 配置的输出接口	车辆状态参数
③	参考横向位移、横摆角计算 模块	车辆位置	参考横向位移 参考横摆角 参考曲率 横向位移轨迹 横向跟踪误差
④	输出参数可视化模块	参考横向位移 参考横摆角 参考曲率	车辆质心侧偏角 前轮转角 车辆横摆角速度 横摆角

本章主要围绕车辆三自由度的运动学和动力学模型展开,运动学模型将主要应用于第3章的路径规划算法,而动力学模型则为第4章的轨迹跟踪控制算法提供支持。此外,本章还介绍联合仿真平台 CarSim 和 Matlab/Simulink 控制模块。

首先, 根据功能型无人车的尺寸大小在 CarSim 仿真平台中的车辆模型进行了配置并导入模型, 然后在 Simulink 中搭建模型并对各模块输入输出接口进行介绍, 最后通过联合仿真 CarSim 和 Matlab/Simulink 为本文提出的轨迹跟踪控制算法进行验证。

第3章 基于改进 DWA 算法无人车局部路径规划

本章研究的路径规划是功能型无人车完成导航功能的核心，是解决功能型无人车能否沿着最优路径顺利到达目标点的问题。路径规划主要分为全局路径规划和局部路径规划两类^[46,47]，全局路径规划是在完整的环境信息基础上，利用已知的地图条件，生成一条从起点至终点的无碰撞路径，这属于静态规划的范畴，适用于静态环境^[48]，局部轨迹规划根据无人车当前位置、速度、环境的实时信息以及运动学约束，实时地生成一条平滑、可行的轨迹。动态调整路径以避免未知障碍物，属于动态规划，适用于动态环境^[49,50]。目前，应用于全局路径规划的主要算法有 A*算法，局部路径规划中常用的方法包括动态窗口法（DWA）以及人工势场法^[51]。A*算法搜索效率高，但无法躲避未知障碍物不利于无人车的运行。DWA 算法是一种基于速度采样的局部路径规划算法，能够实时调整机器人速度以避免未知障碍物。面对复杂环境下可能出现的随机障碍物，仅依靠全局最优路径规划是不够的，无人车还需要具备应对随机障碍的能力。基于第2章建立的运动学模型，提出了融合改进 A*和 DWA 算法的路径规划算法，融合算法解决了 A*算法无法躲避动态障碍物和 DWA 算法易陷入局部最优的问题，同时又提高了无人车的行驶稳定性和安全性。

3.1 改进 A*算法的全局路径规划

3.1.1 传统 A*算法

A*算法是结合 Dijkstra 算法和广度优先搜索算法的一种图搜索算法，算法的核心是可以规划出一条最短路径，从起点到目标点的移动代价估计函数为：

$$F(n) = G(n) + H(n) \quad (3-1)$$

式（3-1）中 n 为当前节点， $F(n)$ 是从起点到目标点的估计总代价， $G(n)$ 表示从起点到当前节点 n 的实际代价， $H(n)$ 是从当前节点 n 到目标点的估计代价。

在传统 A*算法中，启发函数通常基于对角线距离、曼哈顿距离或欧几里得距离来定义^[52]，本文选用欧几里得距离作为定义启发函数的方式，欧几里得距离公式：

$$S = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3-2)$$

式 (3-2) 中 x_1 、 x_2 和 y_1 、 y_2 分别代表任意两节点的横坐标和纵坐标, S 表示任意两节点在欧几里得空间中的直线距离。

3.1.2 改进 A* 算法

(1) 优化启发函数

在传统 A* 算法中, 路径遍历节点多, 寻路效率低。因此需要对评价函数进行优化。其中启发函数 $H(n)$ 和实际代价函数 $G(n)$ 之间的大小关系对算法的搜索效率和路径优化程度至关重要。当启发函数 $H(n)$ 被低估时, 探索的节点更多是确保找到最佳路径; 当 $H(n)$ 被准确估计时, A* 算法能够有效地引导搜索向目标前进, 同时避免不必要的扩展; 而当 $H(n)$ 被高估时, 搜索的节点数量会减少, 搜索效率提高, 但很难找到最优路径^[53]。因此, 在实际应用中, 需要根据具体情况调整 $H(n)$ 的权重。通过优化代价函数解决上述算法的不足, 当前节点距离目标点较远时, 通过增大代价函数中 $H(n)$ 的权重提高无人车寻路效率, 遍历节点减少。同理, 当前节点距离目标点较近时, 无人车降低寻路速度以确保顺利到达目标点。改进后的启发函数具体公式为:

$$\begin{cases} F(n) = G(n) + \left(1 + \frac{r}{R} - \log P\right) H(n) \\ P = \frac{M}{N} \end{cases} \quad (3-3)$$

式 (3-3) 中: P 表示起始点与目标点之间的障碍率, M 为障碍物的数量, N 为栅格总数, r 为当前节点到目标点的距离, R 为起点到目标点的距离。

(2) 设置安全距离

设置由 A* 算法探索的路径与最近的障碍物顶点保持安全距离 D 。如图 2 所示, 设 $D=1$, Start 为起点, Target 为目标点, N1、N2、N3 为中间节点, 蓝色区域为障碍物。假设 N1 是当前节点, 如果线段 L1 与障碍物顶点之间的距离小于 D , 则删除节点 N2 并重新搜索下一个节点。当线段 L1、L2 与障碍物顶点的距离大于 D 时, 则 N2 为冗余节点, 节点 N2 也被删除。如此重复, 直到所有节点都满足上述条件并到达目标点。安全距离的设置既解决了无人车斜穿障碍物的问题, 又消除了路径中的冗余点, 确保无人车可以安全平稳地到达目标点, 安全距离如图 3-1 所示。

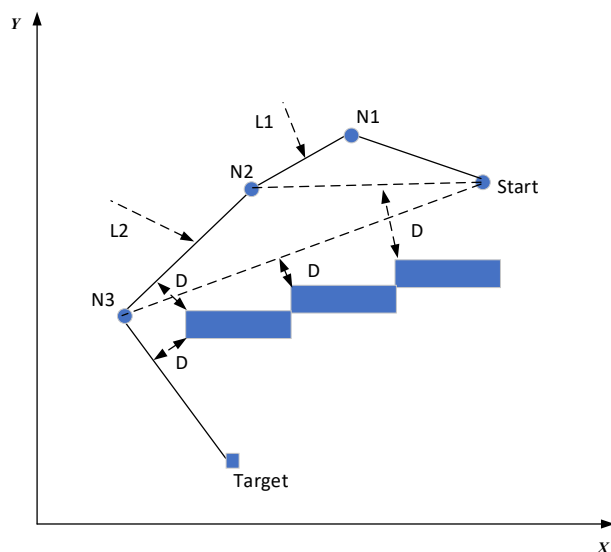


图 3-1 安全距离示意图

3.1.3 改进 A*算法仿真分析

本文分别使用传统 A*算法和改进 A*算法进行对比分析,通过建立不同大小和障碍物数量的栅格地图作为模拟环境验证改进 A*算法的有效性,仿真结果如图 3-2、图 3-3 和图 3-4 所示。

(1) 30×30 栅格地图的算法对比

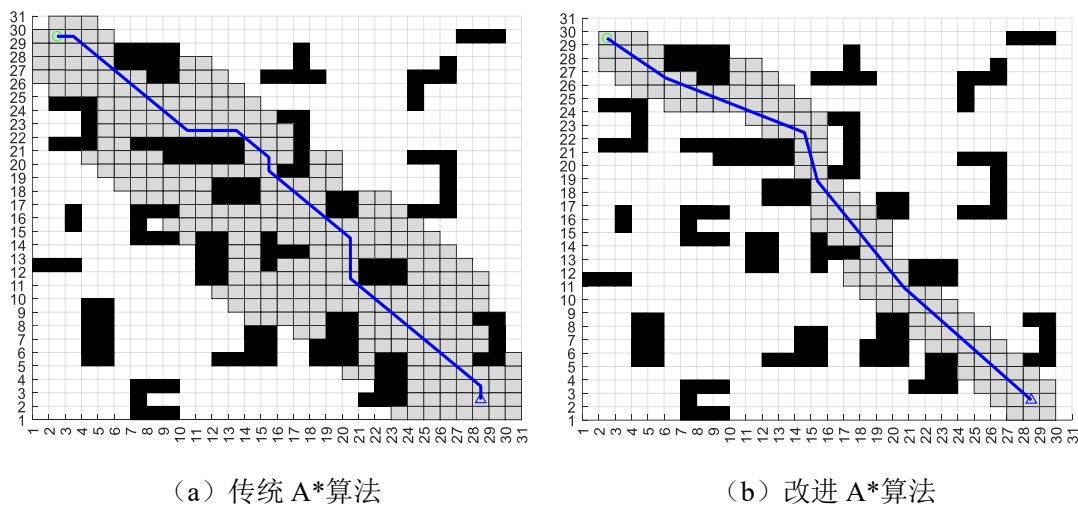


图 3-2 30×30 栅格地图的算法对比图

表 3-1 30×30 栅格地图下的算法性能对比

	路径长度/m	遍历节点数/个	转折次数/次	转折角度/°	搜索时间/ms
传统 A*算法	40.1127	301	8	360.000	25.163
改进 A*算法	38.8076	128	4	96.3199	17.446

(2) 40×40 栅格地图的算法对比

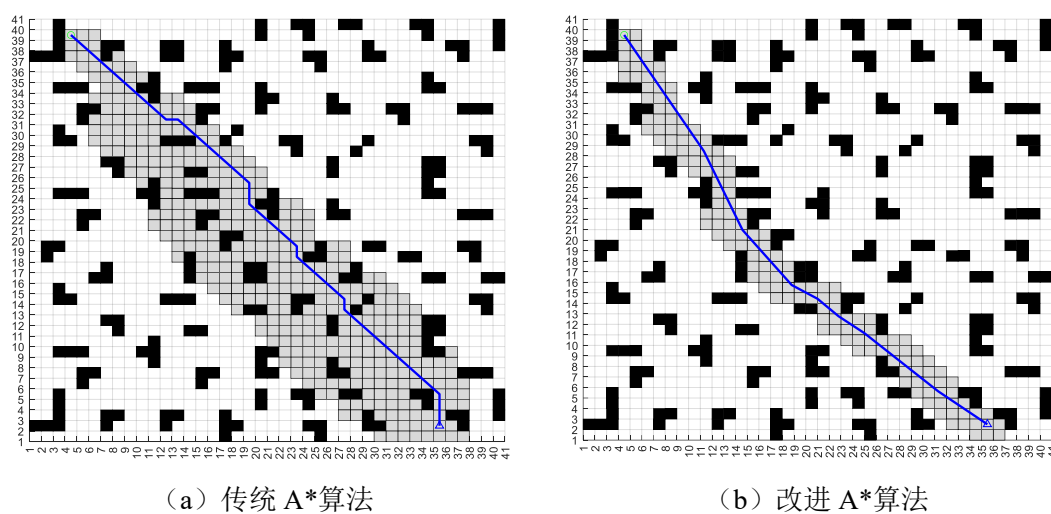


图 3-3 40×40 栅格地图的算法对比图

表 3-2 40×40 栅格地图下的算法性能对比

	路径长度/m	遍历节点数/个	转折次数/次	转折角度/°	搜索时间/ms
传统 A*算法	50.4264	382	9	405.000	15.722
改进 A*算法	49.2160	156	7	71.8006	5.9060

(3) 50×50 栅格地图的算法对比

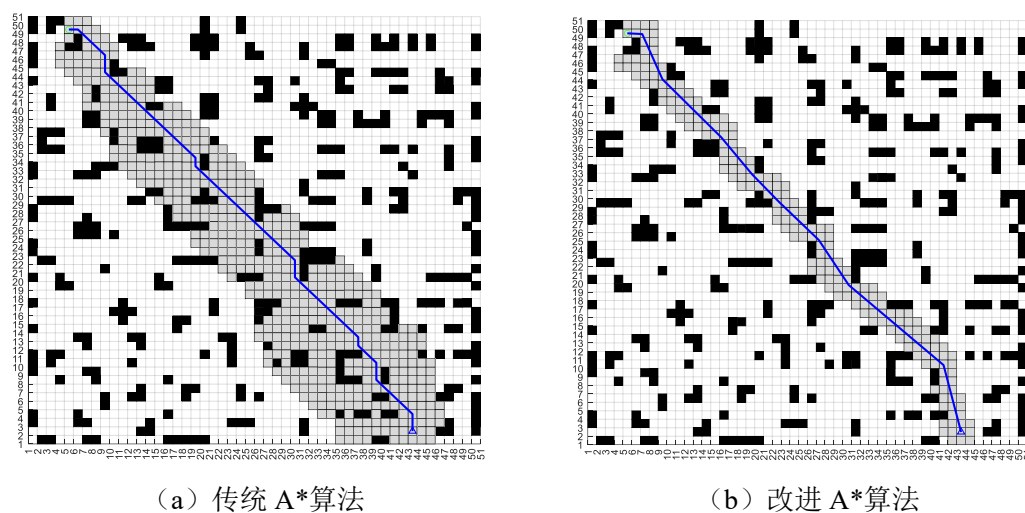


图 3-4 50×50 栅格地图的算法对比图

表 3-3 50×50 栅格地图下的算法性能对比

	路径长度/m	遍历节点数/个	转折次数/次	转折角度/°	搜索时间/ms
传统 A*算法	63.3259	546	12	540.000	26.663
改进 A*算法	62.3140	206	8	160.2678	5.1390

从图中可以看出,改进后的 A*算法所规划的路径遍历节点明显减少,转折次

数减少路径更加平滑，路径长度也有一定程度的缩短，设置安全距离后，无人车不会有沿着障碍物行驶的风险。在 30×30 的地图环境下，改进后的 A* 算法遍历节点数和转折角度明显减少，遍历节点数减少了 56.81%，改进后的 A* 算法的转折角度与传统 A* 算法相比减少了 73.24%，搜索效率也提升了 30.66%，提升了无人车行驶的安全和稳定性，路径长度也有一定程度的缩短。在 40×40 栅格地图中，改进后的 A* 算法遍历节点数效率提升 59.16%，转折角度减少了 82.27%，搜索效率提升了 62.43%。在 50×50 栅格地图中，改进后的 A* 算法遍历节点数效率提升 62.27%，转折角度减少了 70.32%，搜索效率提升了 80.73%。由此可见改进 A* 算法不仅大大提高了效率，而且路径长度也有一定程度的缩短，路径也因为转折角度和转折点的减少更加平滑，减少了搜索时间。

3.2 改进融合算法的局部路径规划

3.2.1 DWA 算法

DWA 算法是一种针对未知环境设计的局部动态路径规划算法，它能够在保证安全性的同时有效地避开障碍物^[54]。算法的核心思想是通过将无人驾驶车辆的位置变化转换为线速度和角速度的控制指令，将原本复杂的随机避障问题简化为在速度空间中的约束优化问题。通过计算无人车在这些速度情况下移动一定时间内的预测轨迹，并通过评价函数对该轨迹进行评价^[55,56]，结合评价函数的输出结果，选择得分最高的轨迹作为无人驾驶车辆的行驶路径。驱动无人车行驶。因此该过程无须考虑无人车的受力情况，故选择用单车模型作为本文的运动学模型。根据运动学模型，设无人车在时刻 t 的线速度为 $v(t)$ ，角速度为 ω ，则相邻两时刻的位姿变化可通过以下公式描述：

$$\begin{cases} x(t+1) = x(t) + v_t \Delta t \cos \phi(t) \\ y(t+1) = x(t) + v_t \Delta t \sin \phi(t) \\ \phi(t+1) = \phi(t) + \omega \Delta t \end{cases} \quad (3-4)$$

式 (3-4) 中： $x(t+1)$ 、 $y(t+1)$ 、 $\phi(t+1)$ 为 $t+1$ 时刻的位姿信息， Δt 为两时刻的时间间隔。

(1) 约束条件

在速度空间中，无人车的行驶速度由多个因素共同决定，包括其当前速度、电机性能以及安全制动距离等。这些因素相互作用，影响无人车的速度规划和动态调整能力^[57]。该约束主要分为三类：

速度约束:

$$V_m = \{v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \quad (3-5)$$

式 (3-5) 中: V_m 表示无人车自身速度大小的限制; v_{max} 、 ω_{max} 为最大速度和最大角速度; v_{min} 、 ω_{min} 为最小速度和最小角速度。

加速度受电机性能约束:

$$V_d = \{(v, \omega) | v \in [v_c - v_b \Delta t, v_c + v_a \Delta t], \omega \in [\omega_c - \omega_b \Delta t, \omega_c + \omega_a \Delta t]\} \quad (3-6)$$

式 (3-6) 中: V_d 表示无人车能够达到的速度空间, v_c 、 ω_c 为无人车的当前速度, v_a 、 ω_a 为最大加速度和最大角加速度, v_b 、 ω_b 为最小加速度和最小角加速度。

制动距离约束, 局部规划还需要有动态实时的避障功能:

$$V_a = \{(v, \omega) | v \leq \sqrt{2dist(v, \omega)v_b}, \omega \leq \sqrt{2dist(v, \omega)\omega_b}\} \quad (3-7)$$

采样空间内的速度是以上 3 个约束的交集, 具体形式如下:

$$V_r \in V_m \cap V_d \cap V_a \quad (3-8)$$

式 (3-7) 和式 (3-8) 中: V_a 是满足制动距离的速度空间, $dist(v, \omega)$ 为 (v, ω) 对应的轨迹上离障碍物最近的距离, V_r 是三个约束条件的交集。

(2) 评价函数

DWA 算法的评价体系通常包括方位角、行进速度以及与目标点的距离这三个关键要素。用于评估路径的安全性、目标导向性和效率。在进行局部路径规划时, 系统需要依赖于多种传感器来收集周围环境的实时信息。传感器数据的非连续性可能导致评价结果出现显著差异, 会降低路径规划的效率和安全性, 因此需要对每个指标进行归一化处理^[58], 归一化处理后得到的轨迹评价函数公式如下:

$$G(v, \omega) = \alpha Head(v, \omega) + \beta Vel(v, \omega) + \gamma Dist(v, \omega) \quad (3-9)$$

式 (3-9) 中, α 、 β 、 γ 分别是方位角、速度和距离的加权系数; $Head(v, \omega)$ 为无人车当前朝向与目标点方向的夹角; $Vel(v, \omega)$ 是用来评价无人车行驶时的速度大小的; $Dist(v, \omega)$ 是指轨迹上的点与障碍物的最短距离; $G(v, \omega)$ 是轨迹总的评分。

3.2.2 改进 DWA 算法

传统 A* 算法的优点是能够规划出一条最短路径, 但是规划的路径冗余点较多, 实时性差, 容易与未知障碍物发生碰撞。DWA 算法的优点在于能够局部有效避开未知障碍物, 但 DWA 算法只考虑了下一步的规划, 没有考虑到全局的路

径规划,使得无人车在面对复杂环境时无法很好地进行避障,同时也导致无人车选择了一条非全局最优的路径。针对上述两种算法的不足,本文融合改进 A*算法和 DWA 算法^[59,60],确保无人车能够实时避障,安全到达目标点。融合算法不仅使规划的路径搜索节点减少,而且路径平滑性和安全性也有所提高,融合算法流程图如图 3-5 所示。

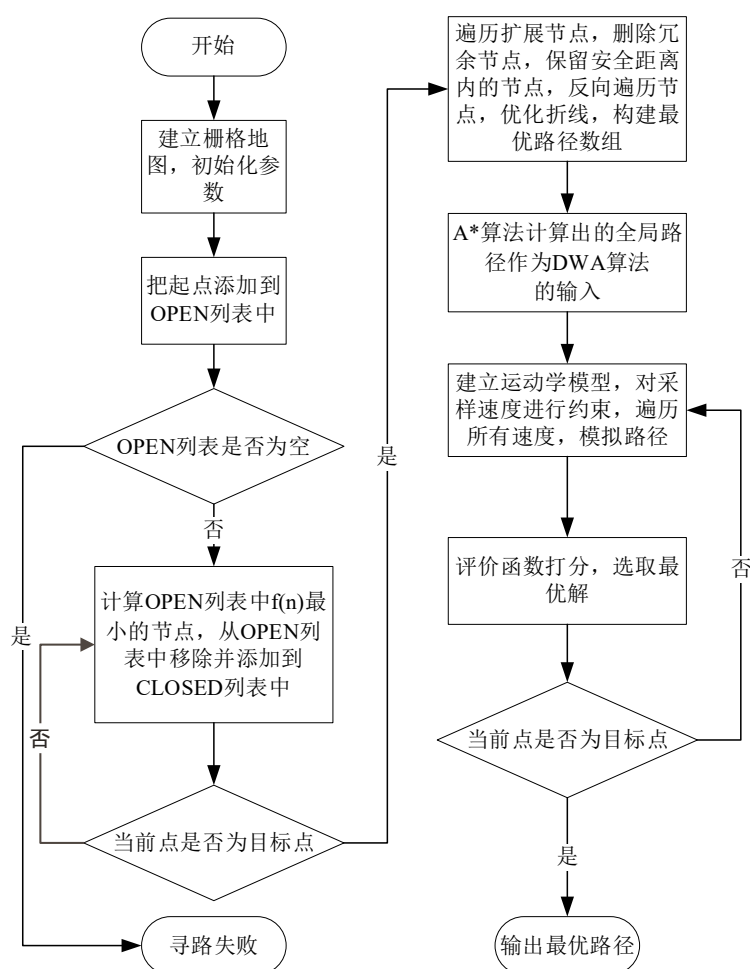


图 3-5 融合算法流程图

针对传统 A*和 DWA 算法的不足,首先对 A*算法进行改进,改进后的算法再与 DWA 算法进行融合,具体改进步骤如下:

步骤 1: 首先建立栅格地图,并初始化参数。同时建立 OPEN 和 CLOSED 列表,将起始点加入至 OPEN 列表中,并检查该列表是否为空。若为空,则表明路径搜索失败;若非空,则需在 OPEN 列表中找出具有最小代价的节点,将其移除并加入 CLOSED 列表。再判断当前节点是否为目标点,若不是,则重复刚才的操作,若是则进行步骤 2。

步骤 2: 若当前节点是目标点,遍历扩展节点,删除冗余节点,保留在安全

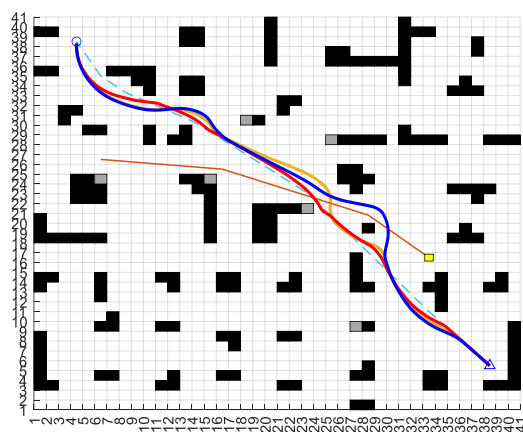
距离内的节点。反向遍历保留的节点，优化折线路径，并生成最优路径数组，返回路径距离和 OPEN 列表中的节点数。将改进后的 A* 算法规划出的全局路径作为 DWA 算法的输入进行融合。

步骤 3：改进 A* 算法规划好全局初始路径后，首先随机设置移动障碍物和静止障碍物，通过建立好的运动学模型进行多组速度和方向角度采样，并在一定时间内生成不同的运动轨迹；再通过评价函数进行评分，选出最优轨迹作为无人车下一时刻的指令，使无人车沿着最优路径行驶；最后判断当前点是否为目标点，若是，则寻路成功，若不是，则重复步骤 3 的操作。

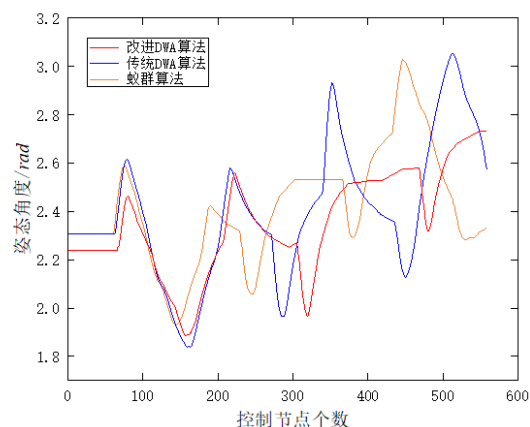
步骤 4：输出最优路径，包括改进 A* 算法所规划的全局路径，以及经过融合算法处理后的最优路径。以及移动障碍物路径，同时还包含无人车的姿态角度、线速度和角速度的变化曲线。

3.2.3 融合算法仿真分析

仿真结果如图 3-6 所示，图 (a) 三种算法对比图，其中蓝色虚线是改进后的 A* 算法规划的全局路径，红色实线表示本文改进 DWA 算法规划的路径，蓝色实线表示传统 DWA 算法规划的路径，黄色实线是蚁群算法规划路径，橙色实线是移动障碍物规划路径，黄色矩形方块表示移动障碍物，灰色方块表示随机静态障碍物。图 (b) 是无人车行驶过程中的姿态角度对比，图 (c) 和 (d) 是无人车行驶过程中线、角速度随着路径节点的变化曲线对比。



(a) 三种算法对比图



(b) 姿态角度对比图

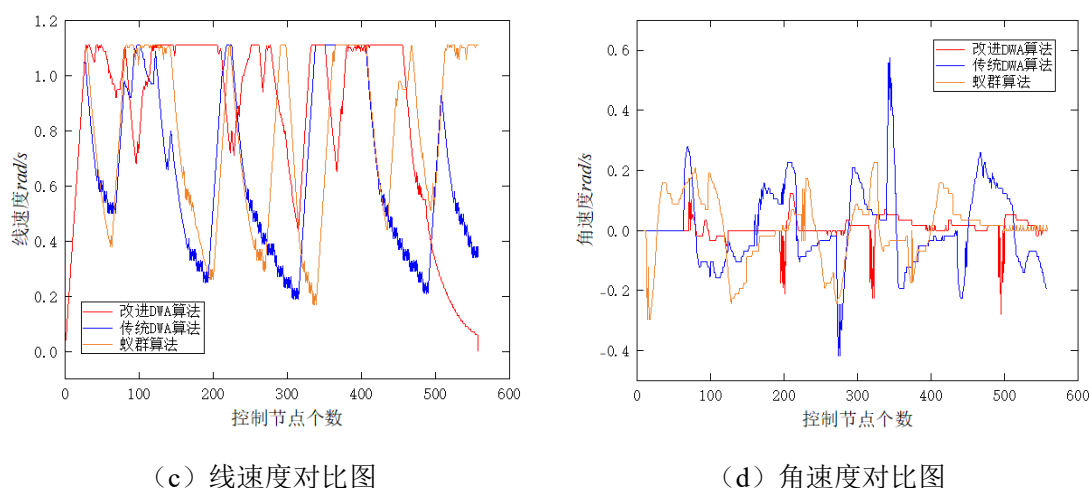


图 3-6 算法动态避障结果对比图

表 3-4 三种算法路径规划性能对比

	遍历节点数/个	路径长度/m	搜索时间/s
传统 DWA 算法	708	51.9305	89.1274
改进 DWA 算法	554	48.5357	52.8725
蚁群算法	640	49.4140	78.0763

由图 3-6 仿真结果可知，针对传统 DWA 算法容易陷入局部最优无法到达目标点的问题，通过改进 A*算法规划出全局最优路径，局部路径规划使用改进 A*和 DWA 融合算法，无人车不仅可以有效地避开动态障碍物，而且不陷入局部最优。本文融合算法不仅可以实现随机避障，而且规划的路径不会陷入局部最优，路径长度也有一定程度的减少，与传统算法和蚁群算法相比搜索效率提升了 40.68%和 32.28%，图 (b)、(c) 和 (d) 分别对比了无人车行驶过程中的姿态角度、线速度和角速度的变换对比，可以从图中看出改进 DWA 算法的无人车姿态角度更加平稳，线速度和角速度变化幅度更小，因此本文算法在保证不斜穿障碍物的情况下以贴合全局最优路径行驶，安全平稳的到达目标点，达到了本文的预期目的。

3.3 本章小结

传统 A*算法在路径规划中虽然具有高效性和完备性，但其存在搜索效率低、冗余节点多等问题，尤其是在复杂环境中表现更为明显，优化后的算法冗余节点减少，无人车不会贴合障碍物行驶，转折点也有所减少，提高了无人车的行驶稳定性和安全性。针对 DWA 算法只考虑了下一步的规划，没有考虑到全局的路径

规划,容易陷入局部最优的情况,为解决局部最优问题,提出融合改进 A*与 DWA 算法的无人车随机避障方法,将改进后的 A*算法所生成的路径节点作为 DWA 算法的输入信息,用于局部路径规划,从而使无人驾驶车辆能够有效规避未知障碍物,确保安全且平稳地抵达预定目标位置。通过实验验证,得到结论如下:

(1) 本文将改进 A*算法和传统 A*算法进行对比,并分别在不同障碍物数量和地图环境下进行验证。通过实验结果可知,改进后 A*算法在转折点个数和路径长度方面都有所减少,有效提高了 A*算法搜索效率,同时对于更大更复杂的情景,可以规划出安全且平稳的路径,说明该算法对于复杂场景具有一定的实用性。

(2) 融合改进 A 算法与 DWA 算法能够有效提升无人车的路径规划能力。改进 A 算法负责全局路径规划,通过优化启发函数和减少冗余节点,确保路径的全局最优性。DWA 算法则专注于局部避障,利用动态窗口实时调整速度和方向,避开未知障碍物。融合算法不仅解决了传统 DWA 算法易陷入局部最优无法顺利到达目标位置的问题,还实现了路径的平滑性和安全性,使无人车能够在复杂环境中高效、稳定地到达目标点,通过与传统 DWA 算法和蚁群算法作对比,融合算法规划的路径效率更高且更加符合无人车的运动特性,该算法具有一定的实用性和有效性。

第 4 章 基于改进 LTV-MPC 的无人车轨迹跟踪控制

基于第 3 章的路径规划算法的研究,本文优化的路径规划算法具有较好的安全性和实用性,但仅研究规划算法并不能很好的保证车辆在行驶过程中的平稳性。因此本章将对功能型无人车轨迹跟踪控制算法进行研究,轨迹跟踪的目的是使车辆尽可能的沿着参考轨迹行驶,同时兼顾跟踪精度和操控稳定性^[61]。轨迹跟踪控制作为连接上层规划与底层执行的关键环节,对功能型无人车的安全行驶至关重要。MPC 是在每一个控制时刻,利用系统的动态模型进行优化预测,并根据预测结果求解控制输入,可以灵活处理状态、输入和输出的各种约束条件^[62,63]。基于车辆三自由度动力学模型,本章将构建线性时变模型预测控制(LTV-MPC),LTV-MPC 能够在线处理状态约束并预测未来状态,特别适用于智能汽车的轨迹跟踪控制问题,因此被选为本文的路径跟踪控制策略。为了提高本文设计的控制器的自适应能力,提出了一种 T-S 模糊控制对 LTV-MPC 中目标函数的权重系数进行自适应。最后在第 2 章搭建的联合仿真平台 CarSim/Simulink 中对本文设计的控制器进行验证。

4.1 LTV-MPC 轨迹跟踪控制器设计

4.1.1 模型预测控制原理

模型预测控制的基本原理如 4-1 图所示,在控制过程中利用系统的预测模型,预测未来一段时间内的状态,并基于这些预测结果进行优化。通过不断调整控制输入,MPC 可以在满足约束条件的情况下,使系统的行为尽量接近期望目标。 k 为当前时刻,结合预测模型,预测系统在 $[k, k + Np]$ 时域内的输出,通过对目标函数进行求解得到最优控制序列,同时系统模型预测在这样的控制序列下状态值的变化。在完成控制量的计算和模型预测之后,只对系统施加最优控制序列中的第一个元素。到达时刻 $k+1$ 时,重复此过程,预测控制随着时间的前进不断反复地在线运行,在每一个新的时刻都需要计算与分析预测区间内发生的情况,在每一次优化开始,系统的状态变量会作为初始条件用于系统的模型预测,也可以理解为系统的反馈项。

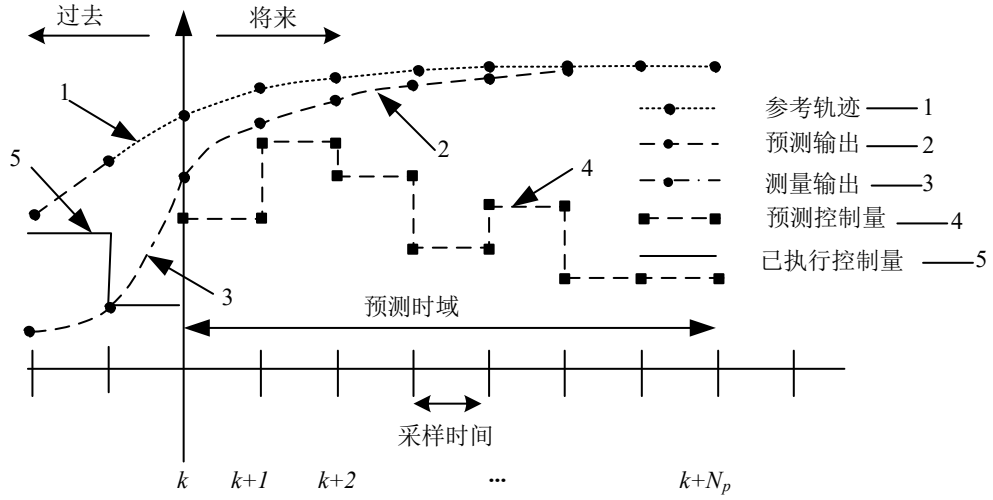


图 4-1 模型预测控制原理示意图

4.1.2 车辆非线性系统线性化和离散化

由于车辆非线性控制方法因计算复杂度高,难以满足实时性和稳定性的需求。第 2 章建立的动力学模型是非线性的,因此需要将车辆非线性系统线性化^[64,65]。线性化处理后的 MPC 轨迹跟踪控制器通常能够简化求解过程,加速控制输入的响应,同时确保系统的稳定性。

将式 (2-11) 建立的状态空间方程进行线性化处理,设系统的任意工作点位为 $[\xi_{(0)}, u_{(0)}]$,在施加控制量 $u_{(0)}$ 后,系统的状态量将发生变化,经过线性化处理后,得到新的状态量表达式:

$$\dot{\xi}_{(0)} = f(\xi_{(0)}, u_{(0)}) \quad (4-1)$$

在 $[\xi_{(0)}, u_{(0)}]$ 处进行保留一阶项的泰勒展开可得:

$$\dot{\xi}_{(t)} = f(\xi_{(0)}, u_{(0)}) + \frac{\partial f}{\partial \xi} \Big|_{\xi=\xi_{(0)}} (\xi_{(t)} - \xi_{(0)}) + \frac{\partial f}{\partial u} \Big|_{u=u_{(0)}} (u_{(t)} - u_{(0)}) \quad (4-2)$$

式 (4-2) 中:

$$\frac{\partial f}{\partial \xi} \Big|_{\xi=\xi_{(0)}} = \begin{bmatrix} \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial \dot{y}} & 0 & \frac{\partial \ddot{x}}{\partial \dot{\phi}} & 0 & 0 \\ \frac{\partial \ddot{y}}{\partial \dot{x}} & \frac{\partial \ddot{y}}{\partial \dot{y}} & 0 & \frac{\partial \ddot{y}}{\partial \dot{\phi}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\partial \ddot{\phi}}{\partial \dot{x}} & \frac{\partial \ddot{\phi}}{\partial \dot{y}} & 0 & \frac{\partial \ddot{\phi}}{\partial \dot{\phi}} & 0 & 0 \\ \cos\phi & -\sin\phi & \frac{\partial f_{\dot{y}}}{\partial \phi} & 0 & 0 & 0 \\ \sin\phi & \cos\phi & \frac{\partial f_{\dot{x}}}{\partial \phi} & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial f}{\partial u} \Big|_{\substack{\xi=\xi_{(0)} \\ u=u_{(0)}}} = \left[\frac{2C_{cf}}{m}, \frac{2C_{cf}}{m} (2\delta_{f,t-1} - \frac{\dot{y}_t + a\dot{\phi}_t}{\dot{x}_t}), 0, \frac{2aC_{cf}}{I_z}, 0, 0 \right]^T$$

将式 (4-2) 和 (4-1) 式进行做差得:

$$\dot{\tilde{\xi}}_{(t)} = A_{(t)}\tilde{\xi}_{(t)} - B_{(t)}\tilde{u}_{(t)} \quad (4-3)$$

式 (4-3) 中:

$$\dot{\tilde{\xi}}_{(t)} = \dot{\xi}_{(t)} - \dot{\xi}_{(0)}; \tilde{\xi}_{(t)} = \xi_{(t)} - \xi_{(0)}; \tilde{u}_{(t)} = u_{(t)} - u_0; \tilde{A}_{(t)} = I + TA_{(t)}; \tilde{B}_{(t)} = TB_{(t)}$$

本采用向前欧拉法对式 (4-3) 进行离散化, 得:

$$\begin{cases} \tilde{\xi}_{(k+1|t)} = \tilde{A}_{(k|t)}\tilde{\xi}_{(k|t)} + \tilde{B}_{(k|t)}\tilde{u}_{(k|t)} + d_{(k|t)} \\ d_{(k|t)} = \bar{\xi}_{(k+1|t)} - \tilde{A}_{(k|t)}\tilde{\xi}_{(k|t)} - \tilde{B}_{(k|t)}\tilde{u}_{(k|t)} \end{cases} \quad (4-4)$$

式 (4-4) 中:

$$\tilde{A}_{(k|t)} = \begin{bmatrix} 1 + \frac{2\delta_r C_{rf}(\dot{y} + a\dot{\phi})}{m\ddot{x}^2} T & \left(\phi - \frac{2\delta_f C_{cf}}{m\dot{x}}\right) T & 0 & \left(\dot{y} - \frac{2a\delta_r C_{rf}}{m\dot{x}}\right) T & 0 & 0 \\ \left(-\phi + \frac{2[C_{cf}(\dot{y} + a\dot{\phi}) + C_{cr}(\dot{y} - b\dot{\phi})]}{m\ddot{x}^2}\right) T & 1 - \frac{2C_c + 2C_{cr}}{m\dot{x}} T & 0 & \left(-\dot{x} + \frac{2(aC_{ef} - bC_{er})}{m\dot{x}}\right) T & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ \frac{2aC_{cf}(\dot{y} + a\dot{\phi}) + 2bC_{cr}(\dot{y} - b\dot{\phi})}{I_z\ddot{x}^2} T & \frac{-2aC_{cf} + 2bC_{cr}}{I_z\dot{x}} T & 0 & 1 + \frac{-2(a^2C_{ef} + 2b^2C_{er})}{I_z\dot{x}} T & 0 & 0 \\ T\cos\phi & -T\sin\phi & -T(\dot{x}\sin\phi + \dot{y}\cos\phi) & 0 & 1 & 0 \\ T\sin\phi & T\cos\phi & T(\dot{x}\cos\phi - \dot{y}\sin\phi) & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \tilde{B}_{(k|t)} &= \left[\frac{2C_{cf}}{m}, \frac{2C_{cf}}{m} (2\delta_f - \frac{\dot{y}_t + a\dot{\phi}_t}{\dot{x}_t}), 0, \frac{2aC_{cf}}{I_z}, 0, 0 \right]^T \\ \begin{cases} \bar{\xi}_{(k+1|t)} = f(\bar{\xi}_{(k|t)}, \bar{u}_{(k|t)}), k = t, t+1, \dots, t+N_p-1 \\ \bar{\xi}_{(t|t)} = \tilde{\xi}_{(t)} \\ \bar{u}_{(k|t)} = u_{(t-1)} \\ \tilde{A}_{(k|t)} = (I + TA_{(t)}) \\ \tilde{B}_{(k|t)} = TB_{(t)} \end{cases} \end{aligned}$$

为降低控制器的计算复杂度, 做式 (4-5) 假设:

$$\begin{cases} \tilde{A}_{(k|t)} = \tilde{A}_{(t|t)}, k = t, t+1, \dots, t+N_p-1 \\ \tilde{B}_{(k|t)} = \tilde{B}_{(t|t)}, k = t, t+1, \dots, t+N_p-1 \end{cases} \quad (4-5)$$

完成车辆系统的线性离散化处理后, 需对所得状态空间方程进行处理, 以确保控制系统在调节过程中, 控制增量的幅度保持在合理范围内。将线性离散化后的方程转化为描述车辆控制增量的状态空间方程:

$$\tilde{\xi}_{(k|t)} = \begin{pmatrix} \tilde{\xi}_{(k|t)} \\ \tilde{u}_{(k-1|t)} \end{pmatrix} \quad (4-6)$$

$$\begin{cases} \check{\xi}_{(k+1|t)} = \check{A}_{(k|t)}\check{\xi}_{(k|t)} + \check{B}_{(k|t)}\Delta u_{(k|t)} + \check{d}_{(k|t)} \\ \check{\eta}_{(k|t)} = \check{C} * \check{\xi}_{(k|t)} \end{cases} \quad (4-7)$$

在式 (4-7) 中:

$$\check{A}_{(k|t)} = \begin{bmatrix} \check{A}_{(k|t)} & \check{B}_{(k|t)} \\ 0_{m \times n} & I_m \end{bmatrix}; \quad \check{B}_{(k|t)} = \begin{bmatrix} \check{B}_{(k|t)} \\ I_m \end{bmatrix}; \quad \Delta u_{(k|t)} = u_{(k|t)} - u_{(k-1|t)}; \quad \check{d}_{(k|t)} = \begin{bmatrix} d_{(k|t)} \\ 0 \end{bmatrix}; \quad \check{C} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}; \quad m, n \text{ 分别表示系统控制量和状态量的个数。}$$

4.1.3 输出模型构建和目标函数设计

定义 k 时刻系统输出和输入分别为:

$$\begin{cases} Y_{(k+1|t)} = [\check{\eta}_{(k+1|t)}, \check{\eta}_{(k+2|t)}, \dots, \check{\eta}_{(k+N_p|t)}]^T \\ \Delta U_{(k|t)} = [\Delta u_{(k|t)}, \Delta u_{(k+1|t)}, \dots, \Delta u_{(k+N_c|t)}]^T \end{cases} \quad (4-8)$$

在式 (4-8) 中: N_p 、 N_c 分别为系统的预测时域和控制时域, 且 $N_p > N_c$ 。

为了得到系统未来时刻 k 的预测方程矩阵表达式, 需要在整个预测时域内进行迭代计算, 并通过递推过程逐步求解。预测方程矩阵表达式为:

$$Y_{(k|t)} = \psi_{(k|t)}\check{\xi}_{(k|t)} + \theta_{(k|t)}\Delta U_{(k|t)} + \Gamma_{(k|t)}\phi_{(k|t)} \quad (4-9)$$

式 (4-9) 中:

$$\begin{aligned} \psi_{(k|t)} &= \begin{bmatrix} \check{C}\check{A}_{(k|t)} \\ \check{C}\check{A}_{(k|t)}^2 \\ \vdots \\ \check{C}\check{A}_{(k|t)}^{N_p} \end{bmatrix} \\ \theta_{(k|t)} &= \begin{bmatrix} \check{C}\check{B}_{(k|t)} & 0 & \dots & 0 \\ \check{C}\check{A}_{(k|t)}\check{B}_{(k|t)} & \check{C}\check{B}_{(k|t)} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \check{C}\check{A}_{(k|t)}^{N_p-1}\check{B}_{(k|t)} & \check{C}\check{A}_{(k|t)}^{N_p-2}\check{B}_{(k|t)} & \dots & \check{C}\check{A}_{(k|t)}^{N_p-N_c}\check{B}_{(k|t)} \end{bmatrix} \\ \Gamma_{(k|t)} &= \begin{bmatrix} \check{C} & 0 & 0 & 0 \\ \check{C}\check{A}_{(k|t)} & \check{C} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \check{C}\check{A}_{(k|t)}^{N_p-1} & \check{C}\check{A}_{(k|t)}^{N_p-2} & \dots & \check{C}\check{A}_{(k|t)}^0 \end{bmatrix} \end{aligned}$$

$$\Delta U_{(k|t)} = \begin{bmatrix} \Delta u_{(k|t)} \\ \Delta u_{(k+1|t)} \\ \vdots \\ \Delta u_{(k+N_c-1|t)} \end{bmatrix}$$

$$\phi_{(k|t)} = \begin{bmatrix} \check{d}_{(k|t)} \\ \check{d}_{(k+1|t)} \\ \vdots \\ \check{d}_{(k+N_p-1|t)} \end{bmatrix}$$

目标函数的作用为确保车辆能够稳定且精确地跟踪预定的参考轨迹，在设计目标函数时，需兼顾对系统状态量误差以及控制量增量的优化考量。以保证车辆在跟踪过程中能够尽可能减少与期望轨迹之间的偏差，引入松弛因子防止优化问题无解，设计的目标函数的表达式如下所示：

$$J(\xi_{(k|t)}, u_{(k-1|t)}, \Delta U_{(k|t)}) = \sum_{i=1}^{N_p} \|\check{\eta}_{(k+i|t)} - \eta_{ref(k+i|t)}\|_Q^2 + \sum_{i=0}^{N_c-1} \|\Delta U_{(k+i|t)}\|_R^2 + \rho \varepsilon^2 \quad (4-10)$$

式（4-10）中： Q 和 R 权重矩阵； ρ 为松弛因子； η_{ref} 为输出量的参考值。

$$Q = \begin{bmatrix} Q_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & Q_{N_c} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & Q_{N_p} \end{bmatrix}, Q_i = \begin{bmatrix} Q_\varphi & 0 \\ 0 & Q_Y \end{bmatrix}, i = 1, 2, \dots, N_p \quad (4-11)$$

$$R = \begin{bmatrix} R_0 & 0 & \cdots & 0 \\ 0 & R_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{N_c-1} \end{bmatrix}, R_i = [R_{\Delta\delta}], i = 1, 2, \dots, N_c-1 \quad (4-12)$$

在式（4-11）和（4-12）中： Q_φ 为横摆跟踪目标权重， Q_Y 为横向位移跟踪目标权重， $R_{\Delta\delta}$ 为输出增量目标权重。

控制器条件约束部分包括控制量约束、控制增量约束和输出量约束。

控制量约束：

$$U_{min} \leq U_t + A_I \times \Delta U \leq U_{max} \quad (4-13)$$

控制增量约束：

$$lb \leq X \leq ub \quad (4-14)$$

输出量约束：

$$Y_{min} \leq Y(k) \leq Y_{max} \quad (4-15)$$

在上式（4-13）、式（4-14）和式（4-15）中： U_{max} 、 U_{min} 为控制量的上限与

下限值； lb 、 ub 则代表状态量的最大值与最小值范围； ΔU_{max} 、 ΔU_{min} 为控制增量的最大值、最小值。

对式（4-10）进行求解计算，可得到控制增量序列：

$$\Delta U_{(k|t)}^* = [\Delta u_{(k|t)}^*, \Delta u_{(k+1|t)}^*, \dots, \Delta u_{(k+N_c-1|t)}^*]^T \quad (4-16)$$

$$u_{(k|t)} = u_{(k-1|t)} + \Delta u_t^* \quad (4-17)$$

通过 MPC 算法结构图 4-2 可知，MPC 算法通过优化目标函数，得到一个控制增量序列，将式（4-16）得到的控制增量序列中第一个项作为被控对象的输入，再通过被控对象的输出的状态量进行反馈校正给 MPC 预测下一时刻的输出，利用式（4-17）计算出前轮转角输入值后，系统持续进行预测与优化的循环更新流程，实现对参考轨迹的有效追踪。

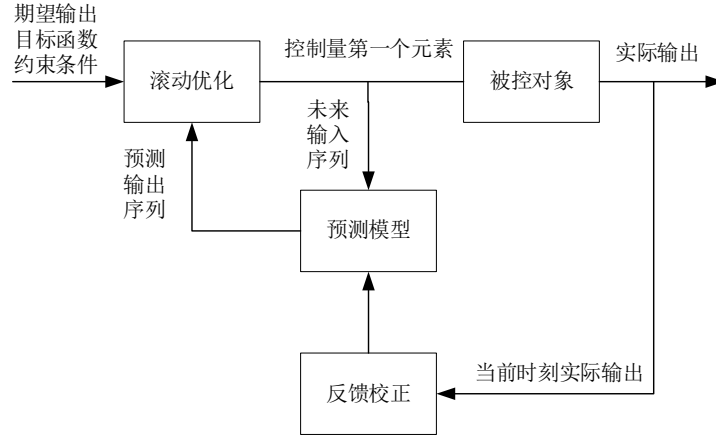


图 4-2 模型预测算法结构图

4.1.4 LTV-MPC 轨迹跟踪控制算法仿真

本文在 Carsim/Simulink 联合仿真平台上进行了轨迹跟踪控制仿真实验，为了验证本文设计的控制器的有效性。选取双移线轨迹作为参考轨迹，通过对比不同车速下的跟踪效果，评估控制器的性能。实验过程中，LTV-MPC 的目标函数权重系数保持恒定以确保实验的一致性。

（1）仿真工况 1:16 km/h

在车速 16 km/h 的双移线工况下，仿真结果如图 4-3 所示，LTV-MPC 轨迹跟踪控制器能够有效实现无人车的精确路径跟踪。横向位移和横摆角的误差较小，表明车辆能够紧密跟随参考轨迹，确保了跟踪精度和稳定性。车辆在行驶过程中，横摆角速度和质心侧偏角均呈现出较小的变化幅度，这一表现证明了车辆始终维持着良好的行驶稳定性。此外，前轮转角输出值在 -3° 至 3° 之间，处于合理范围，

保证了车辆的低速稳定行驶。综上所述，LTV-MPC 控制器在低速工况下表现出色，能够满足无人车对参考路径的跟踪需求。

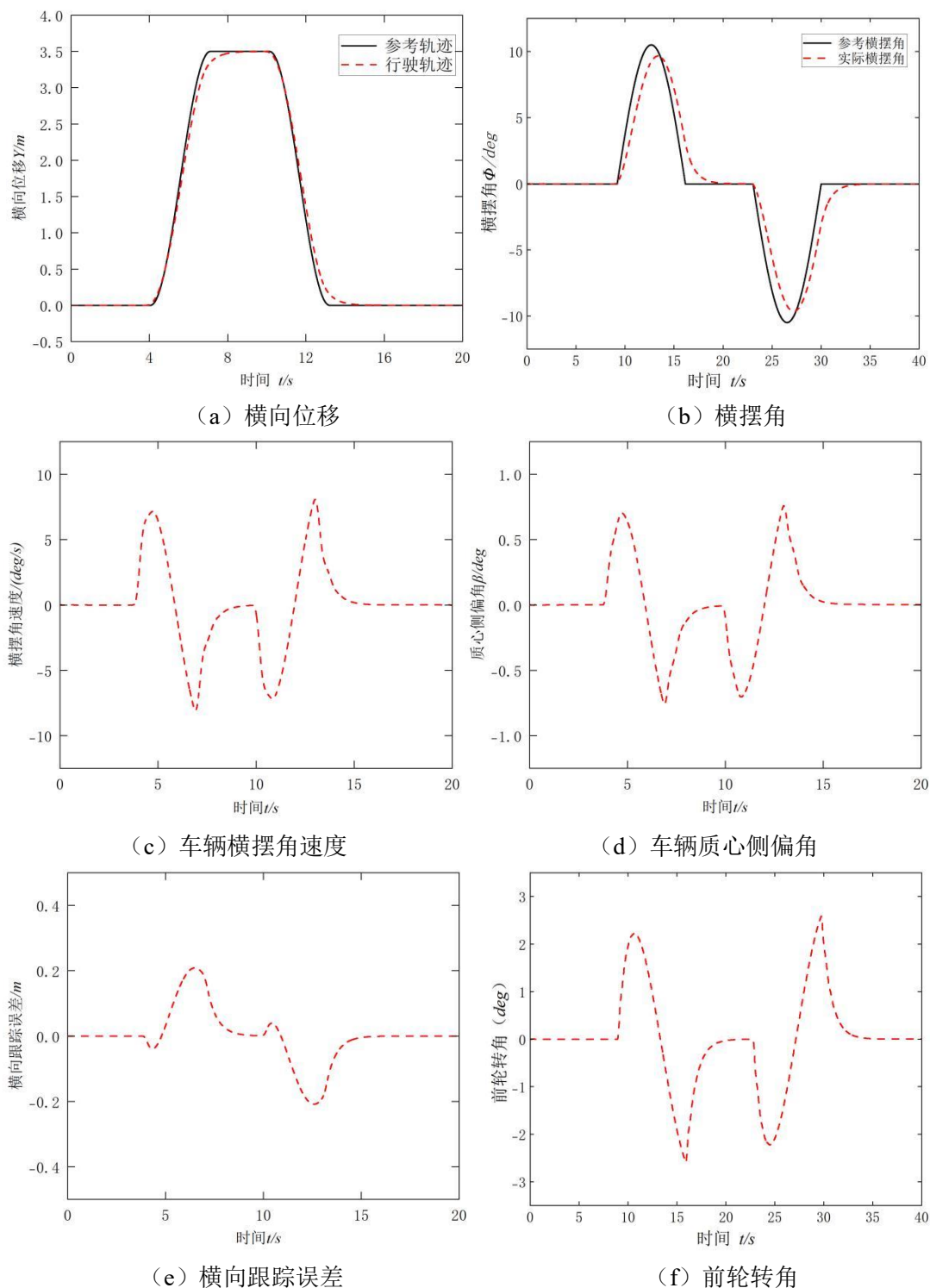


图 4-3 车速 16 km/h 双移线仿真结果

(2) 仿真工况 2:32 km/h

在车速 32 km/h 的双移线工况下，仿真结果如图 4-4 所示。从图表 (a) 与 (b) 观察得知，相较 16km/h 工况，横向位移误差增大但在可接受范围内，MPC

轨迹追踪系统展现良好稳定性与追踪精度。图(c)、(d)及(f)显示,当参考路径从弯道转变为直线段时,横摆角速度和质心侧偏角出现变化,这主要因为路径曲率快速变化且此时横向误差较大,增加了跟踪难度,无人驾驶车辆为确保准确跟踪精度,需持续调节前轮转向角的大小。综上所述,能够满足无人车对参考路径的跟踪需求。

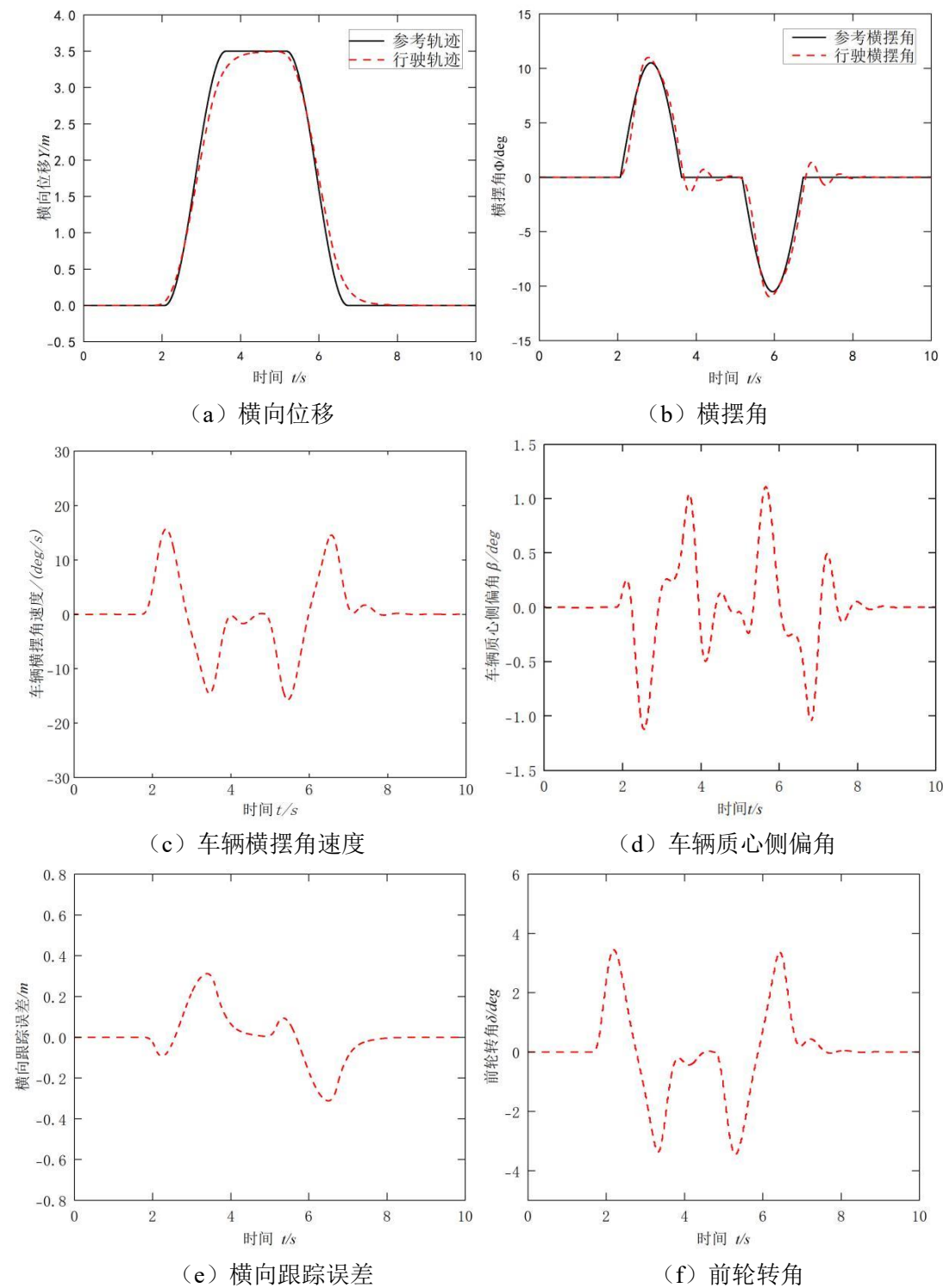


图 4-4 车速 32 km/h 双移线仿真结果

(3) 仿真工况 3: 48 km/h

在车速 48 km/h 的双移线工况下, 仿真结果如图 4-5 所示。图表 (a)、(b) 分析表明, 随着车辆行驶速度的提升, 实际行驶路径与参考轨迹间的横向偏差逐渐增大。车辆的横摆角在路径曲率变化时也出现波动。

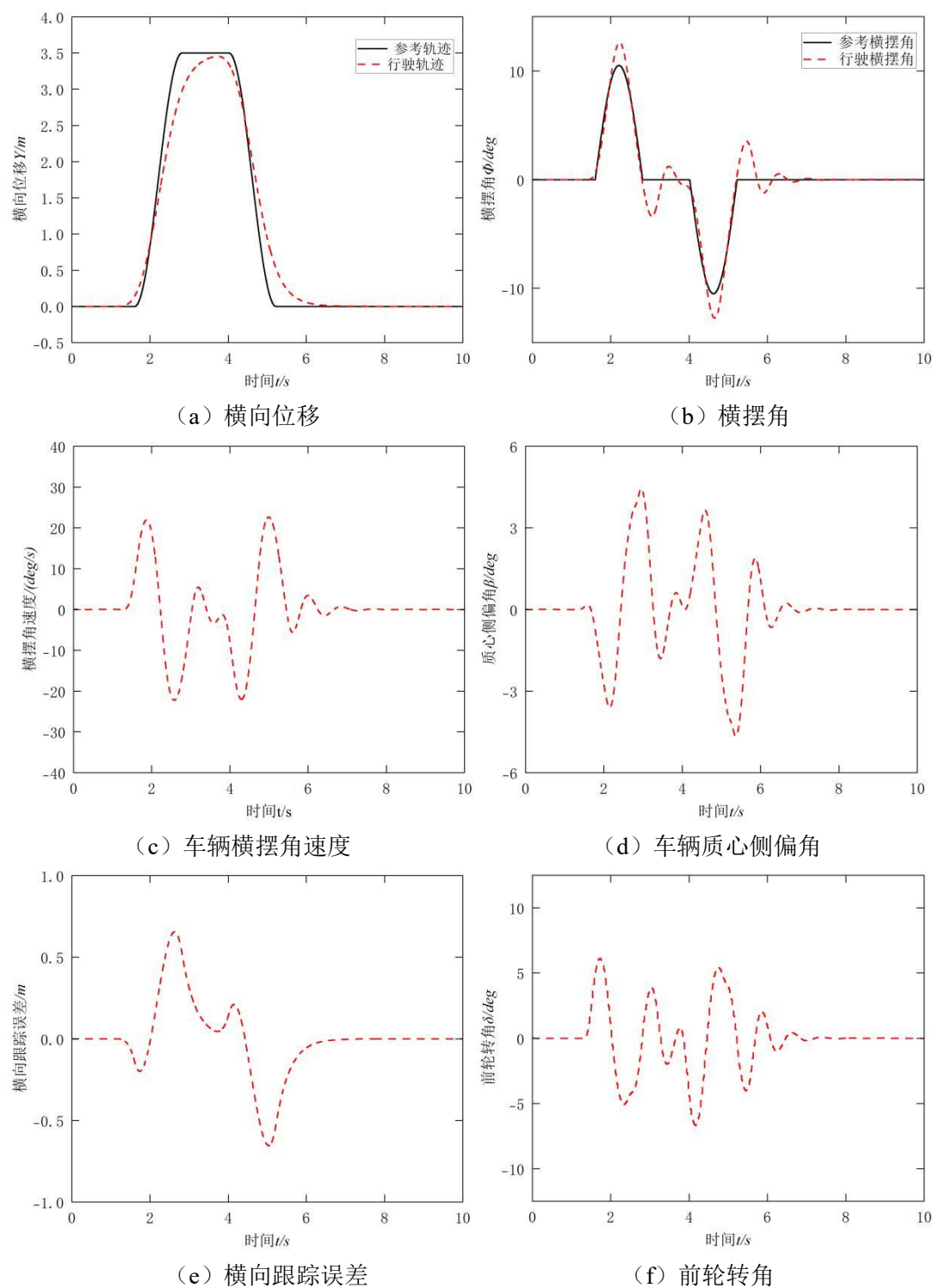


图 4-5 车速 48 km/h 双移线仿真结果

由图 (c)、(d) 和 (f) 可知, 高速状态下横摆角速度及质心侧偏角未引发车辆失稳, 虽波动幅度扩大, 车辆并未发生失稳情况。综上所述, 该系统虽能满足无人驾驶车辆对参考路径的基本跟踪需求, 但在跟踪精度与稳定性方面仍有待提升。

在本文设计的路径跟踪控制器中, 当目标函数的权重系数被设定为固定值时, 从仿真结果中可以看出, 随着车速的增加, 横摆角速度和质心侧偏角的波动范围明显增大, 横向跟踪误差变大, 稳定性和跟踪精确性降低。

4.2 改进 LTV-MPC 轨迹跟踪控制器设计

为了提高 LTV-MPC 控制的跟踪精确性和稳定性, 改善自适应能力^[66,67], 采用 T-S 模糊控制在线优化在 LTV-MPC 控制, LTV-MPC 对于无人车的行驶稳定性会随着路面附着系数和车速的不同有着不同的影响结果。随着车速提升与路径曲率的变化, 自动驾驶车辆稳定性降低, 需调高稳定性权重确保车辆沿规划路径行驶。为确保路径追踪精准度与行驶稳定性, 适应多种工况环境, 研究采用前轮转角作为控制变量构建 LTV-MPC 控制架构, 结合 T-S 模糊控制技术实时调整 MPC 目标函数权重系数, 将侧向偏移误差与横摆角偏差作为模糊系统输入参数。

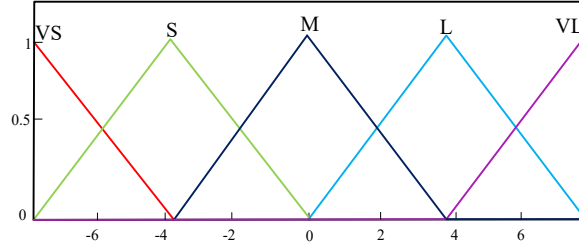
4.2.1 T-S 模糊控制器设计

基于双移线工况仿真分析结果可见知, 车辆在各种速度条件下, 设计的 LTV-MPC 轨迹追踪控制器均能有效引导车辆完成轨迹跟踪。观察发现, 参考路径曲率较大区域, 车辆跟踪精准度下降, 偏移量增加, 为降低误差提高安全系数, 需要频繁调整前轮转向角, 影响行驶平顺性; 当路径从弯道过渡至直线时, 车辆为保证跟踪精度, 实际轨迹出现振荡, 稳定性降低。研究表明, 车辆无法同时兼顾跟踪精度与行驶稳定性。

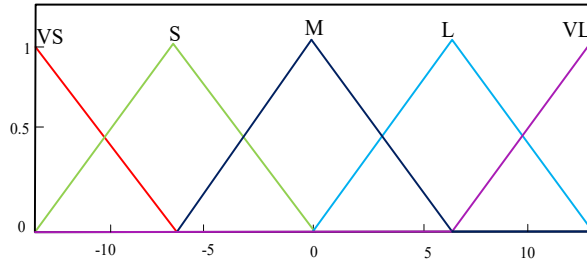
研究表明, LTV-MPC 目标函数权重参数对车辆跟踪性能产生显著影响。权重系数反映了精确追踪与稳定行驶的优先级划分, 车辆与目标路径横向偏移较大时应优先保证跟踪精度, 侧向偏移较小时则应注重车辆行驶稳定性。通过针对车辆实际运行状态动态调整权重参数, 能有效避免跟踪过程中产生过大横向偏差, 同时保持车辆良好稳定性能。

模糊控制基于丰富实操经验或大量实际数据构建控制规则并通过计算机自动执行^[68,69,70]。由 4.1.4 仿真结果可知, LTV-MPC 随着车速的增加横摆角速度和

质心侧偏角的波动范围较大,横向跟踪误差变大,稳定性和跟踪精确性降低。LTV-MPC 中加入 T-S 模糊控制能有效平衡轨迹追踪精度与稳定性,提升系统自适应能力。将横向偏差和横摆角偏差作为模糊控制的输入变量,其隶属度函数采用三角形形式。图 4-6 为两个输入变量的三角形隶属度函数。



(a) 横向位移误差



(b) 横摆角误差

图 4-6 隶属度函数图

在 T-S 模糊系统中,输入变量 ΔY 的论域被划分为 $[-7, 7]$,而输入变量 $\Delta \phi$ 的论域为 $[-13, 13]$ 。分为 5 个模糊子集: VS (极小)、S (小)、M (中)、L (大) 和 VL (极大),每个子集均采用三角形隶属度函数进行描述。T-S 模糊理论的核心在于输出变量与输入变量之间的线性关系,这种关系通常通过线性函数表示,如式 (4-18) 所示。具体而言, T-S 模糊规则的形式为: 如果输入变量属于某个模糊子集,则输出变量是输入变量的线性组合。这种方法通过局部线性化实现了对全局非线性系统的建模,具有较强的适应性和灵活性:

$$\begin{aligned} R_i: & \text{if } \Delta Y \text{ is } A^i \text{ and } \Delta \phi \text{ is } A^i \\ & \text{then } y_i = p_i \end{aligned} \quad (4-18)$$

式 (4-18) 中, R_i 为第 i 条模糊规则, 其中 $i=1,2,\dots,n$; y_i 为第 i 个输出变量; A^i 为第 i 条模糊规则中的某一模糊子集; p_i 为常数。

输入变量 ΔY 、 $\Delta \phi$ 与各输出变量 ΔQ_Y 、 ΔQ_ϕ 、 $\Delta R_{\Delta \delta}$ 之间对应的输出值如下表所示。模糊控制输出变量的论域均设置为 $[0, 1]$ 。式(4-19)表示权重系数 Q_Y 、 Q_ϕ 、 $R_{\Delta \delta}$ 和输出变量的关系。

$$Q_Y = Q_{Y0} \times \Delta Q_Y$$

$$Q_\phi = Q_{\phi0} \times \Delta Q_\phi \quad (4-19)$$

$$R_{\Delta\delta} = R_{\Delta\delta0} \times \Delta R_{\Delta\delta}$$

式(4-19)中 Q_{Y0} 、 $Q_{\phi0}$ 、 $R_{\Delta\delta0}$ 为MPC控制器权重的初始值。

表 4-1 输出变量 ΔQ_Y

$\Delta\phi$	ΔY				
	VS	S	M	L	VL
VS	0	0.25	0.51	0.25	0
S	0	0.25	0.51	0.25	0
M	0.25	0.51	0.87	0.51	0.25
L	0	0.25	0.51	0.25	0
VL	0	0.25	0.51	0.25	0

表 4-2 输出变量 ΔQ_ϕ

$\Delta\phi$	ΔY				
	VS	S	M	L	VL
VS	0.89	0.71	0.35	0.35	0.71
S	0.71	0.35	0.12	0.35	0.12
M	0.35	0.12	0.12	0.71	0.35
L	0.71	0.35	0.12	0.35	0.12
VL	0.89	0.71	0.35	0.35	0.71

表 4-3 输出变量 $\Delta R_{\Delta\delta}$

$\Delta\phi$	ΔY				
	VS	S	M	L	VL
VS	0.51	0.35	0.21	0.35	0.35
S	0.95	0.51	0.35	0.35	0.51
M	0.95	0.51	0.35	0.71	0.51
L	0.95	0.51	0.35	0.35	0.51
VL	0.51	0.35	0.21	0.35	0.35

由三维图 4-7 (a) 可知, 当横向位移误差和横摆角误差较小时, 可以增大横向位移误差目标权重, 此时车辆的稳定性较好, 可以提高跟踪精度; 当横向位移误差和横摆角误差较大时, 由图 4-7 (b)、(c) 可知, 此时应该增大车辆的期望横摆角和前轮转角增量的目标权重, 先保证车辆的行驶稳定性, 再保证跟踪精度。通过本文设计改进的控制器实现了对权重系数的自适应控制。

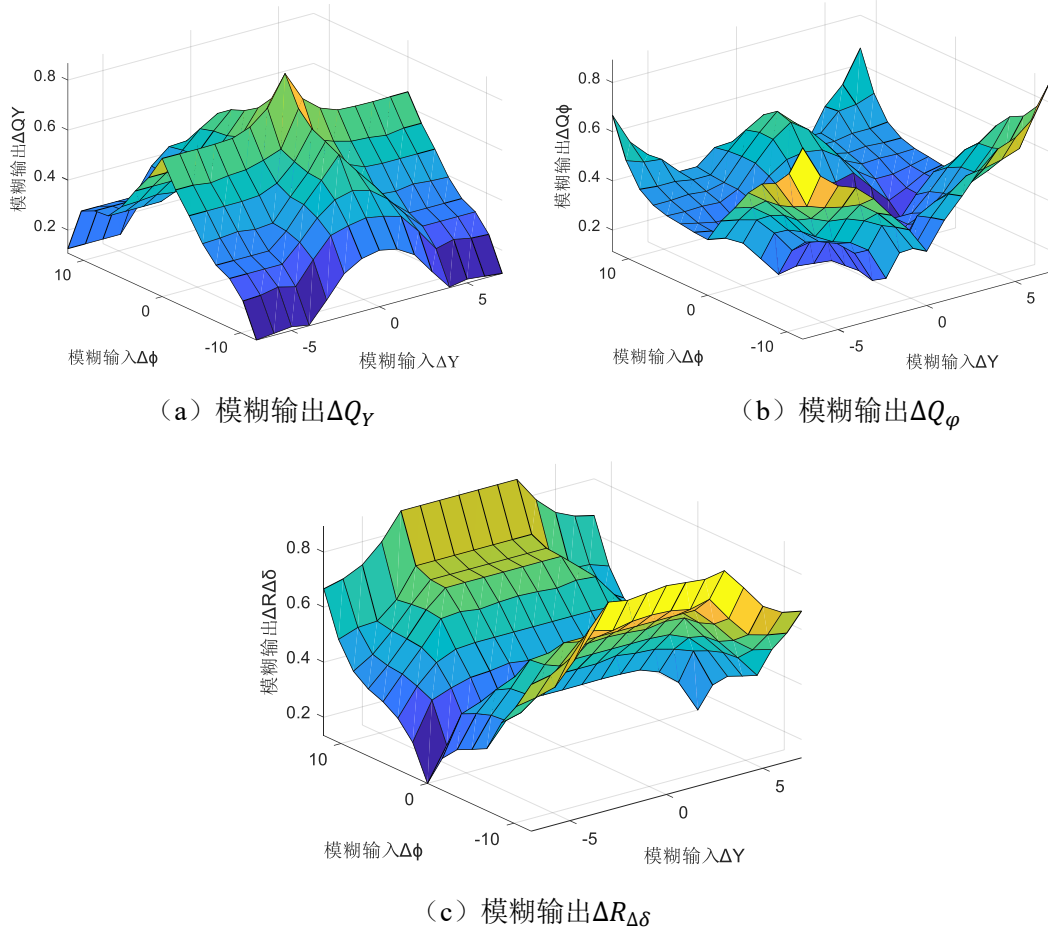


图 4-7 模糊输出三维图

4.2.2 改进 LTV-MPC 轨迹跟踪仿真分析

在 Simulink/CarSim 联合仿真平台中搭建 T-S FMPC 控制模型，选取双移线工况，并设定了三种车速条件：16 km/h、32 km/h、48 km/h，路面附着系数为 0.8。

(1) 仿真工况 1: 16 km/h

在车速 16 km/h 的双移线工况下，由仿真结果图 4-8 和表 4-4 中的数据可知 T-S FMPC 控制器相较于 LTV-MPC 控制器，在轨迹追踪的精确度以及行驶平稳性两方面均展现出了更优的性能。具体而言，T-S FMPC 的最大横向误差为 0.0404 m，较 LTV-MPC 降低了 80.6%，显著提升了跟踪精度。此外，T-S FMPC 的最大横摆角和前轮转角分别较 LTV-MPC 降低了 58.2%和 22.1%，且前轮转角曲线更为平滑，表明其控制输出更加稳定。在车辆动力学性能方面，相较于 LTV-MPC，T-S FMPC 在最大质心侧偏角和横摆角速度方面的降低幅度分别达到了 19.1%和 22%，进一步验证了其在低速工况下的优越性。尽管两种控制器均具备在低速环境下实现轨迹追踪的能力，但 T-S FMPC 通过其先进的控制策略自适应调整目标函数权重，有效优化了控制性能。此外，T-S FMPC 的平滑控制输出减少了车辆

行驶过程中的抖动，提升了驾驶舒适性。综上所述，T-S FMPC 控制器在低速双移线工况下表现出更高的跟踪精度和稳定性，为无人车轨迹跟踪控制提供了更优的解决方案。

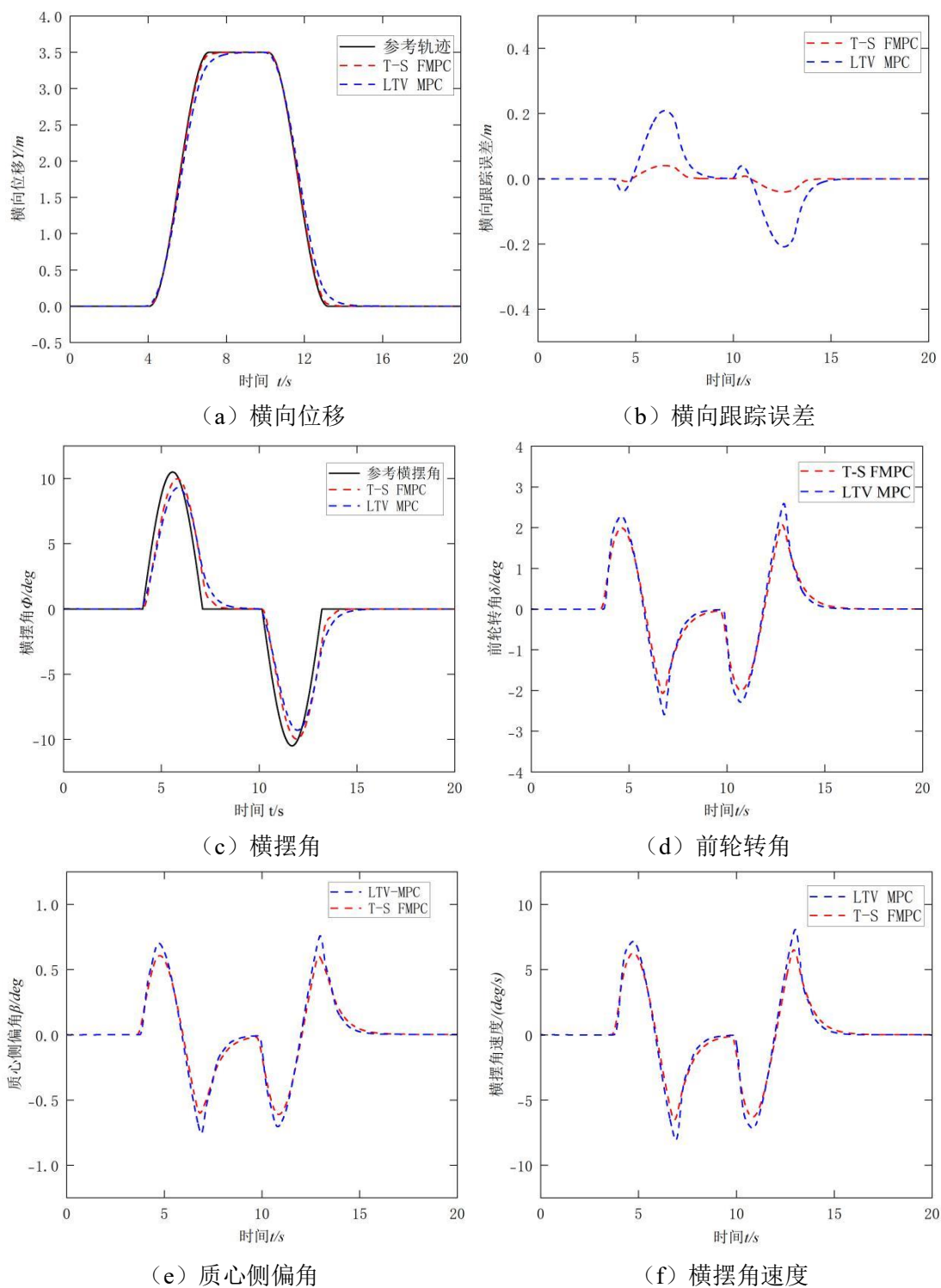


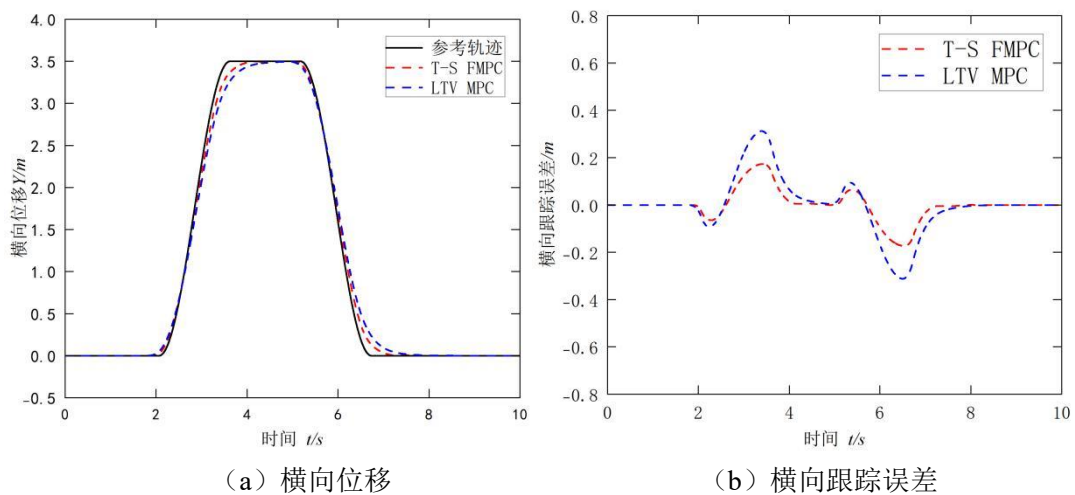
图 4-8 车速 16 km/h 双移线仿真结果

表 4-4 改进前后轨迹跟踪效果对比

	最大横向误差/(m)	最大横摆角误差/(°)	最大质心侧偏角/(°)	最大横摆角速度/(°/s)
LTV MPC	0.2087	1.1740	0.7550	8.013
T-S FMPC	0.0404	0.4900	0.6106	6.250
优化效果	80.6%	58.2%	19.1%	22.0%

(2) 仿真工况 2: 32 km/h

在车速 32 km/h 的双移线工况下，仿真结果图 4-9 和表 4-5 中的数据可知，T-S FMPC 控制器在轨迹追踪精度上相较于 LTV-MPC 控制器具有显著优势。具体而言，T-S FMPC 的最大横向误差为 0.1727 m，较 LTV-MPC 降低了 44.4%，展现了更高的跟踪精度。这一改进主要得益于 T-S FMPC 通过自适应调整目标函数权重，能够根据实时行驶工况优化控制策略，从而更有效地缩小轨迹偏差。由图 (c) 和 (d) 的对比分析可知，T-S FMPC 控制器在横摆角跟踪中表现出更优的动态特性，其轨迹平滑度显著优于 LTV-MPC。具体而言，LTV MPC 在 4s 和 7s 时间节点分别出现了 2° 的幅值振荡，直至 8s 后才逐渐趋于稳定。量化分析显示，T-S FMPC 在控制精度方面实现显著提升：最大横摆角跟踪误差较 LTV-MPC 降幅达 32.3%，同时其前轮转角控制量的峰值水平降低了 26.7%。观察图 (e) 和 (f) 可知，LTV-MPC 的质心侧偏角呈现出波动状态，T-S FMPC 的最大质心侧偏角较 LTV-MPC 减少了 53.5%，同时 T-S FMPC 的横摆角速度也相较于 LTV-MPC 降低了 35.8%。说明 T-S FMPC 能够显著提升轨迹追踪的稳定性。



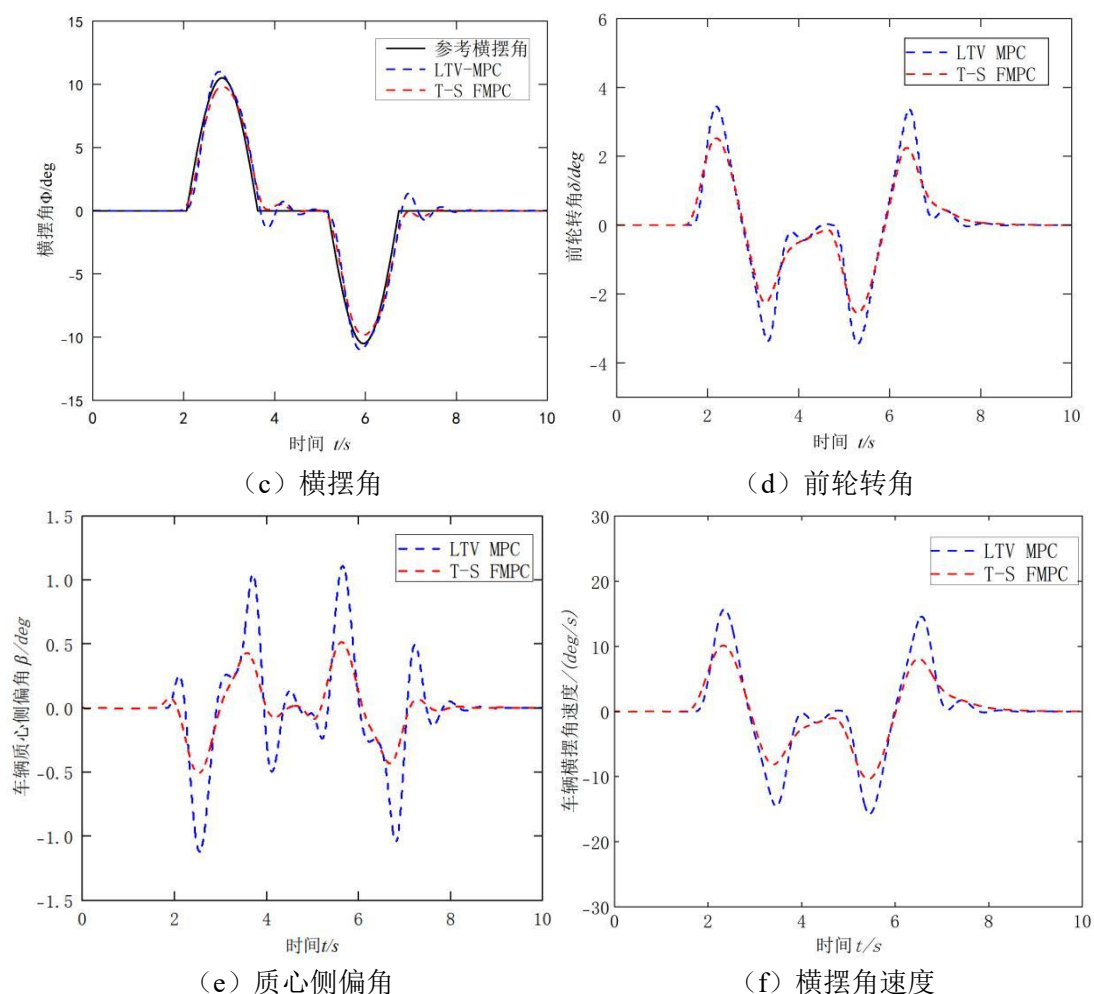


图 4-9 车速 32 km/h 双移线仿真结果

表 4-5 改进前后轨迹跟踪效果对比

	最大横向误差/(m)	最大横摆角误差/(°)	最大质心侧偏角/(°)	最大横摆角速度/(°/s)
LTV MPC	0.3110	0.7480	1.1040	15.65
T-S FMPC	0.1727	0.5060	0.5132	10.04
优化效果	44.4%	32.3%	53.5%	35.8%

(3) 仿真工况 3: 48 km/h

在车速 48 km/h 双移线工况下，由仿真得到的结果如图 4-10 和表 4-6 中数据可知。T-S FMPC 控制器在轨迹跟踪精确性和行驶稳定性方面均优于传统 LTV-MPC 控制器。具体而言，T-S FMPC 的最大横向误差为 0.3412m，较 LTV-MPC 降低了 47.6%，显著提升了跟踪精度。由图（c）和（d）的对比分析可知，T-S FMPC 控制器在横摆角跟踪中表现出更优的动态特性，其轨迹平滑度显著优于 LTV-MPC。具体而言，LTV MPC 在 3s 和 6s 时间节点分别出现了 4°的幅值振荡，直至 7s 后才逐渐趋于稳定。量化分析显示，T-S FMPC 在控制精度方面实现显著

提升：最大横摆角跟踪误差较 LTV-MPC 降幅达 85.3%，同时其前轮转角控制量的峰值水平降低了 30.3%。值得注意的是，LTV-MPC 在整个控制过程中各参数极值始终高于 T-S FMPC 控制器的对应指标。由图 (e) 和 (f) 可以明显看出 LTV-MPC 的质心侧偏角存在波动现象，而 T-S FMPC 的最大质心侧偏角相比 LTV-MPC 则降低了 42.6%，T-S FMPC 横摆角速度相比于 LTV-MPC 降低了 36.1%，说明此时 T-S FMPC 能够改善轨迹跟踪稳定性。

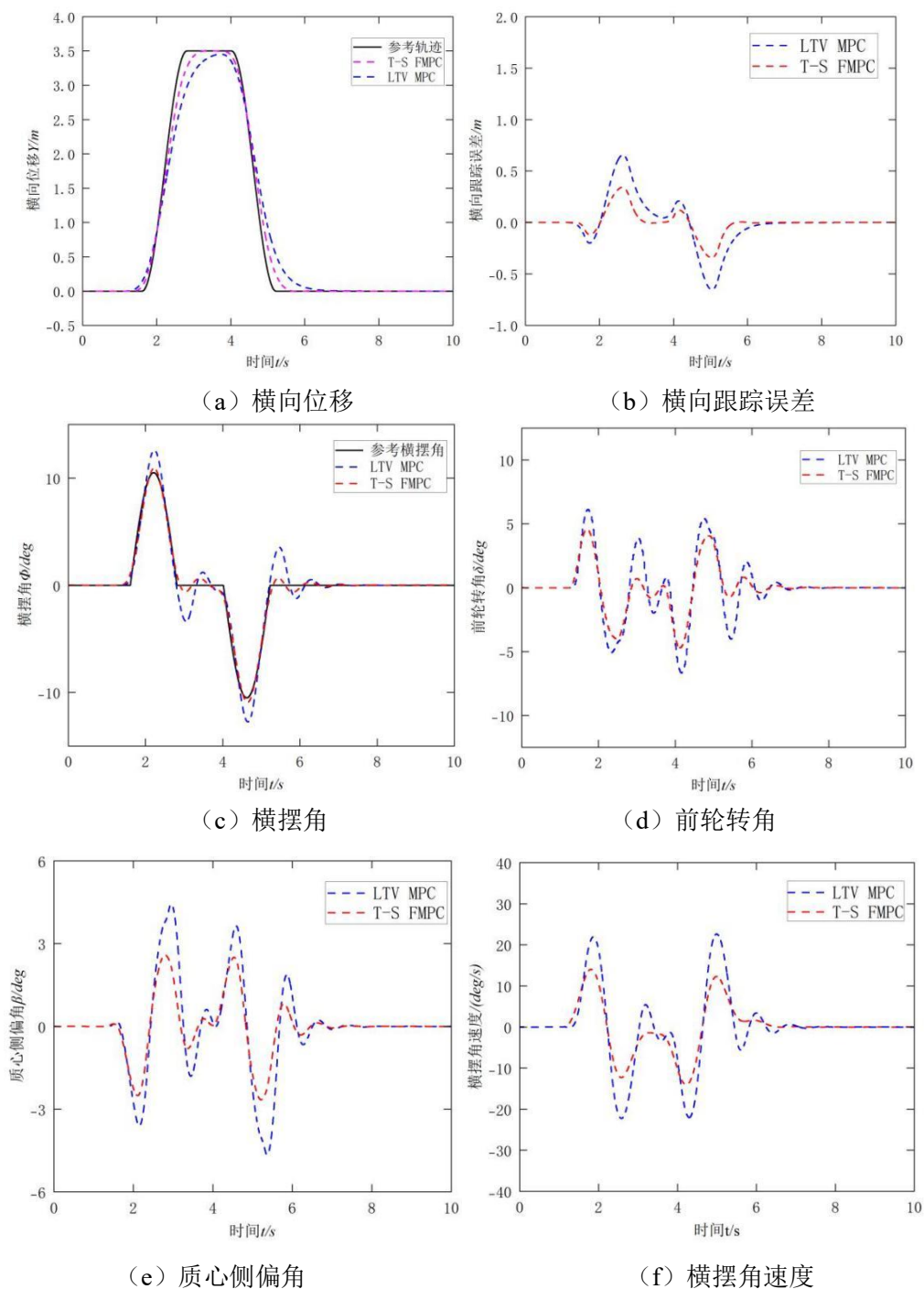


图 4-10 车速 48 km/h 双移线仿真结果

表 4-6 三种算法路径规划性能对比

	最大横向误差/(m)	最大横摆角误差/(°)	最大质心侧偏角/(°)	最大横摆角速度/(°/s)
LTV MPC	0.6522	2.120	4.468	21.89
T-S FMPC	0.3412	0.310	2.561	13.97
优化效果	47.6%	85.3%	42.6%	36.1%

4.3 本章小结

本章主要针对 LTV-MPC 对于无人车的行驶稳定性会随着路径曲率和车速的不同有着不同的影响结果。在车速增加或路径曲率变化时，无人车的稳定性面临挑战，需通过增大稳定性权重以确保其沿参考轨迹行驶，具体内容如下：

(1) 基于第 2 章建立车辆三自由度动力学模型，构建了 LTV-MPC 控制器，通过仿真实验验证了本文设计控制器的有效性，但控制器会受到权重系数的影响，跟踪精度和稳定性下降。

(2) 针对无人车在复杂工况下的轨迹跟踪问题，本文提出了一种基于 T-S 模糊逻辑的变权重 MPC 轨迹跟踪控制策略。该策略的核心在于能够动态地调整目标函数中的权重系数。能够有效应对车速变化和路径曲率变化等工况，从而避免无人车在不同环境下出现稳定性降低和跟踪精度下降的问题。对前轮转角的控制增量进行优化处理，这样做使得控制输出变得更加平稳，从而进一步提升了车辆的动态响应能力和行驶稳定性。

(3) 在 CarSim 中搭建车辆模型，加入模糊控制模块后进行联合仿真验证了该算法的有效性，提升无人车的跟踪精度和稳定性，并实现控制算法对不同工况的自适应性。

第 5 章 无人车实车试验

本章将对第 3 章的改进的路径规划算法和第 4 章的轨迹跟踪控制算法进行实车试验，通过实车试验，可以评估本文规划与控制算法在真实环境中的鲁棒性和适应能力。首先对实验平台的硬件进行介绍，本文使用的是线控底盘进行实车试验，具体介绍了试验中需要用到的传感器及其具体参数。在试验前要先调试各类传感器，以确保实车试验顺利完成。然后对试验的软件方面进行介绍，需要用到的是基于 Linux 系统部署 ROS/Autoware 自动驾驶平台，同时将本文设计的规划和控制算法打包成 ROS 节点并确保成功编译，再与 Autoware 中的规划与控制模块进行集成，试验前还需要安装 PCAN 和激光雷达驱动包。

5.1 实车试验平台搭建

5.1.1 实车硬件平台搭建

(1) 线控底盘

线控底盘是一种用于自动驾驶研究的高度可定制、开源的机器人底盘，如图 5-1 所示。底盘采用了线控技术以提升车辆的灵活性和精确度。底盘支持 ROS 框架，提供丰富的软件库和接口，可以与 ROS 的其他模块进行通信，快速开发和测试自动驾驶。该底盘配备四轮毂电机驱动系统，使得每个车轮可以独立控制，提供更高的动力效率和更精细的操控体验。制动系统采用电液盘式制动，具有更好的反应速度和稳定性。其悬挂系统为前后双叉臂独立悬挂设计，有效提升车辆的舒适性和操控性。车架使用高强度铝合金材料，保证了较轻的重量同时增强了车辆的结构强度。线控底盘、转向电机、轮毂电机的主要参数如下表所示。

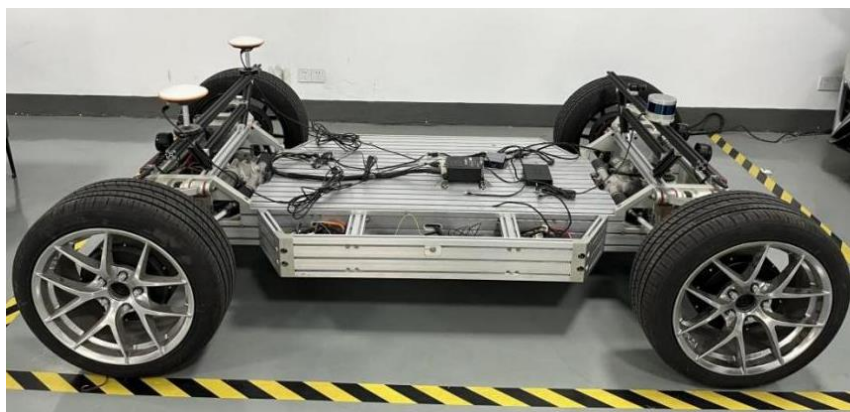


图 5-1 线控底盘

表 5-1 线控底盘主要参数

底盘参数	数值
轮胎规格	205/45R17
轮距	1355mm
最高车速	40km/h
最大负载	1000kg
续航里程	70km
最大爬坡度	20°
最小转弯半径	3m
轮胎最大转弯角度控制	外轮 22° 内轮 32°

表 5-2 线控转向电机主要参数

参数名称	数值
最大输出功率	720W
额定电压	12V DC
额定转速	1050rpm
额定扭矩	3.2N·m
工作温度	-40°C~+85°C
转动范围	±124°C
通信速度	500 k/s

表 5-3 线控轮毂电机主要参数

参数名称	数值
最大输出功率	4424*4W
额定电压	72V
最大转矩电流	58.1A
控速精度	1 rpm/bit
最大转速	811.9rpm
最大转矩	165.3*4N·m
车速范围	±60Km/h
电机尺寸	10 英寸

(2) 激光雷达和相机

Velodyne 公司生产的 VLP-16 线激光雷达配备了 16 个激光束，可以同时扫描并捕捉更广泛的区域。16 线激光雷达能够提供更多的高度信息，提高了点云的密度和精度。具备 360°的水平视场角，可以完整地扫描周围环境，有较强的抗干扰能力，能在多变的复杂场景下维持稳定的工作状态。

相机在无人驾驶中的作用是提供高清晰度的视觉信息，帮助车辆理解周围环境，并做出智能决策。相机通常与激光雷达结合起来使用，提供了更加精确的感知信息。本文选用的是由 FLIR 公司生产的 BFS-U3-04S2C-CS 高性能的单目相

机，这款相机被设计用于需要高精度和高速度的视觉应用。其中激光雷达和相机如图 5-2 和 5-3 所示，具体参数如表 5-4 和 5-5 所示。



图 5-2 VLP-16 线激光雷达



图 5-3 FLIR 单目相机

表 5-4 VLP-16 线激光雷达

参数名称	数值
最大输出功率	4424*4W
额定电压	72V
最大转矩电流	58.1A
控速精度	1 rpm/bit
最大转速	811.9rpm
最大转矩	165.3*4N·m
车速范围	±60Km/h
电机尺寸	10 英寸

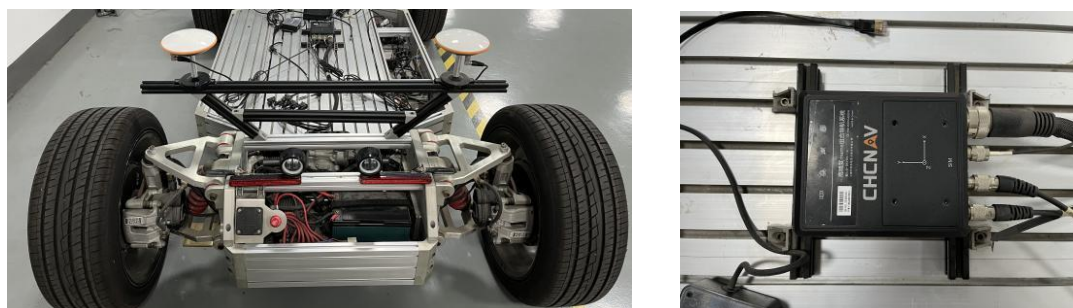
表 5-5 FLIR 单目相机基本参数

单目相机基本参数	数值
尺寸	29×29×30 mm
最大每秒传输帧数	750×540
最大分辨率	522
像元尺寸	6.9 μm
成像缓存	240 MB
数据接口	USB3.1Gen1

(3) 惯性导航系统

华测 CGI-410 组合惯导系统集成了全球导航卫星系统（GNSS）和惯性测量单元（IMU），组合惯导可以在恶劣环境和信号弱的情况下提供高精度的定位、导航和姿态控制服务。首先 CGI-410 集成了 GNSS 接收器，能够接收多种卫星系统如 GPS 等的信号，通过精确的 GNSS 定位，实现高精度的地理位置确认。同时内置高性能的 IMU，能够提供速度、加速度、角速度等数据。这些数据使得 CGI-410 能够在 GNSS 信号丢失或不稳定时，仍然能够提供连续的航位推算和姿

态估计。该设备采用 GNSS 与 IMU 的组合，通过数据融合技术，能够实现比单一传感器更高的精度和可靠性。融合使得系统在信号中断、弱信号环境中仍能够保持稳定运行。图 5-4 为华测 CGI-410 组合惯导，具体参数如表 5-6 所示。



(a)惯导天线

(b)惯导主机

图 5-4 华测 CGI-410 组合惯导 GNSS

表 5-6 华测 CGI-410 组合惯导基本参数

组合惯导基本参数	数值
姿态精度	0.1° (基线长度 $\geq 2\text{m}$)
	单点 L1/L2:1.2m
定位精度	DGPS:0.4m
	RTK:1cm+1ppm
数据更新频率	100HZ
物理尺寸	162 $\times$ 120 $\times$ 53mm
重量	0.5kg
	3 $\times$ RS232 1 $\times$ RS422 1 $\times$ CAN
外部接口	1 $\times$ Micro USB 接口 2 $\times$ GNSS 天线接口 1 $\times$ 4G 天线接口 1 $\times$ 电源接口

(4) CAN 总线

CAN 总线 (Controller Area Network, CAN) 是一种广泛应用于汽车和工业自动化领域的串行通信网络。CAN 网络由含有 CAN 控制器和 CAN 收发器的单元、2 个终端电阻、2 条传输数据线(CAN-H、CAN-L)组成。CAN 控制器用来接收微处理器传来的信息,对这些信息进行处理并传给 CAN 收发器,同时 CAN 控制器也接收来自 CAN 收发器传来的数据,对这些数据进行处理,并传给控制单元的微处理器。CAN 收发器接收控制器送来的数据,并将其发送到 CAN 数据传输总线上,同时 CAN 收发器也接收 CAN 数据总线上的数据,并将其传给 CAN 控制器。终端电阻的作用是防止数据在到达线路终端后像回声一样返回,并因此干扰原始数据,从而保证了数据的正确传送,终端电阻通常安装在控制单元内。传输数据线是两条双向数据线,分为高位(CAN-H)和低位(CAN-L)数据线,主要作用是进

行网络数据传输。

PCAN 是由德国 PEAK-System 公司开发的系列 CAN(Contoller Area Network)硬件和软件产品,旨在实现 CAN 总线的接口和通信,广泛应用于汽车、机器人等多个领域。图 5-5 是一种通过 USB 接口与底盘进行通信的设备。实车试验中需要将设备连接工控机和底盘,获取底盘数据信息。



图 5-5 PCAN

5.1.2 实车软件系统架构

(1) ROS 操作系统

Ubuntu 是基于 Linux 的开源操作系统,ROS 是部署在 Ubuntu 平台上进行开发的软件框架。ROS 是一个功能强大、灵活且易于扩展的机器人开发平台,ROS 采用消息传递机制,实现了模块间的发布-订阅功能,保障了模块间通信的高效性。ROS 支持多种语言,以便支持更广泛的应用开发与移植。ROS 提供了大量的硬件驱动程序,支持多种传感器设备。这些驱动程序封装了硬件与操作系统之间的交互,简化了开发过程。ROS 中还设计集成了多种工具,包括调试工具、可视化工具、数据记录工具、仿真工具等。

(2) Autoware 自动驾驶平台

Autoware 是一个功能全面、开源且基于 ROS 的自动驾驶平台,利用 ROS 的强大功能来实现模块化的自动驾驶系统开发。同时 Autoware 兼容多种传感器和硬件平台,包括激光雷达、摄像头以及各类计算平台,能够与现有的自动驾驶硬件系统进行对接,满足不同应用场景的需求。Autoware 主体框架如图 5-6 所示。

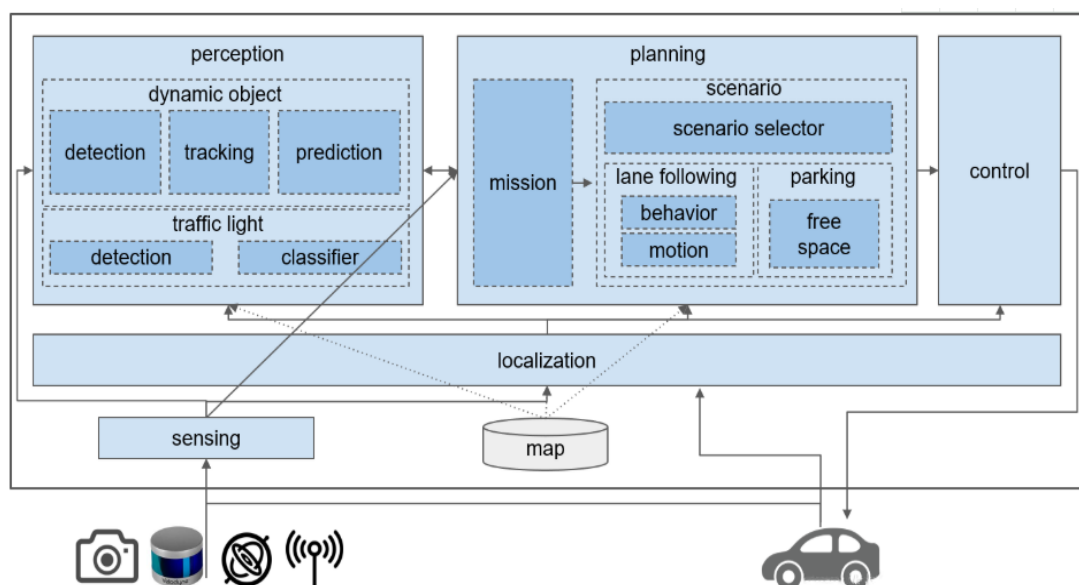


图 5-6 Autoware 主体框架图

Map 模块负责为系统提供自动驾驶车辆行驶中的静态环境信息。主要分为点云地图和向量地图，点云地图提供环境相关的几何信息，点云信息通常由 lidar 来提供。pointcloud_map_loader 节点读取 PCD 格式的点云地图文件，将其解析成 PointCloud2 的数据类型，接收到点云地图的 PointCloud2 消息后，通过 map_tf_generator 对其解析并计算坐标转换。

Sensing 模块通过多个传感器收集环境信息，并将这些信息提供给其他模块。作为硬件与 ROS 系统的接口，Sensing 模块将传感器的原始数据转换为 ROS 消息供系统使用。Localization 模块提供车辆在地图中的实时位置，它通过整合多个传感器的信息来计算车辆的当前位置。

Perception 模块通过传感器数据与地图对比实现定位，无地图时利用 SLAM 构建地图。Detection 模块识别周围动态目标，如车辆和行人。

Prediction 模块预测目标未来行为，保证车辆在未知环境中做出合理的决策。Planning 模块接受感知模块和目标位置的信息规划出路径。Control 模块输出控制信号，确保车辆按照规划路径行驶，同时符合车辆动力学要求。

5.2 实车试验结果与分析

5.2.1 实车试验设计

(1) 建立点云地图

点云地图通过采集车辆周围环境的激光雷达数据,这些点云数据能够为自动驾驶系统提供准确的定位信息。在实车试验中,点云地图能够持续地更新,反映道路环境的变化,因此车辆能够准确感知周围环境中的静态物体和动态物体,从而有助于快速识别和追踪这些物体,提高试验的安全性和避障能力。点云地图提供的车辆周围环境的详细三维信息,使自动驾驶系统可以计算出最优的行驶路径,并对可能的障碍物进行规避。这对于实时决策和路径规划至关重要。因此在实车试验之间先建立厂区内的点云地图,点云地图建立步骤如下。

(1) 在终端中输入命令启动 gazebo 仿真环境和激光雷达驱动,输入 `roslaunch velodyne_pointcloud VLP16_points.launch` 确保激光雷达驱动已正确配置并启动以及激光雷达数据发布到 `/points_raw` 话题。

(2) 点云数据采集,点云数据被实时发布到话题 `/points_raw`,使用 `rostopic echo` 命令来确认点云数据是否正常发布。

(3) 使用 Autoware 进行地图构建,首先先启动 Autoware 在 Autoware 的 `ros/` 目录中执行 `./run` 命令。在 Autoware 中选择激光雷达配置文件,确保点云数据发布到 `/points_raw` 话题,然后启动 `mapping` 模块以构建地图。通过激光雷达数据进行扫描并生成地图。

(4) 通过命令 `roslaunch rviz rviz` 启动 rviz,在 rviz 中选择数据类型 `PointCloud2`,并订阅相关的话题 `/points_raw`。就可以看到实时的点云数据。

(5) 完成地图构建后,输入命令 `roslaunch pcl_ros pointcloud_to_pcd input:=/points_raw`,将 `/points_raw` 话题的数据保存为 PCD 文件,如图 5-7 所示。

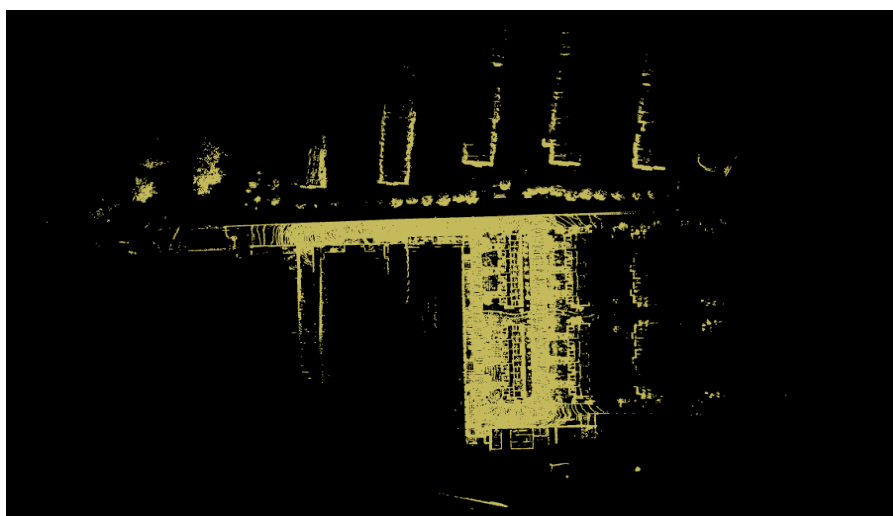


图 5-7 点云地图

(2) 设置参考轨迹

在 ROS 中打开终端启动激光雷达，启动 rviz 修改坐标名称为 velodyne，并确保话题名称为/points_raw 和数据类型 pointcloud2，启动 Autoware 加载点云地图。在 ROS 中创建 A*算法功能包并实现 A*算法接口，修改 Autoware 的启动配置并添加本文设计的全局路径规划算法，将关键坐标进行转换并对参数进行配置后在节点启动时加载参数，最后对算法包进行编译并启动 Autoware，在 Rviz 中设置目标点，观察路径的生成。

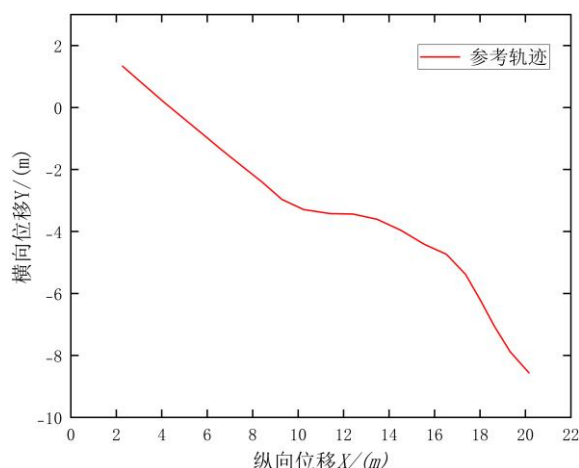


图 5-8 参考轨迹

(3) 试验设计

启动 Autoware 加载点云地图并录制轨迹点，启动 PCAN 驱动与底盘进行通信，然后在 Rviz 中设置好车辆终点后，自动驾驶模式启动，确保车辆顺利避开障碍物。将轨迹点保存，再次打开 Autoware 加载保存的轨迹点，将第 4 章中轨迹跟踪控制算法集成到 Autoware 中，再次启动 PCAN 驱动与底盘进行通信，对轨迹跟踪控制算法进行验证。具体步骤如下：

(1) 打开终端，输入 `source devel/setup.bash` 和 `roslaunch velodyne_piontcloud VLP16_Points.launch` 启动激光雷达。

(2) 打开 Rviz 可视化界面，修改话题名称/points_raw 和数据类型 pointcloud2。

(3) 打开终端，输入 `source install/setup.bash` 和 `roslaunch runtime_manager runtime_manager.launch` 启动 Autoware。

(4) 在 Autoware 的 Setup 模块选择 velodyne 和 TF 坐标，在 Map 模块中加

载点云地图，在 Computing 中勾选 `ndt_matching` 和 `vel_pose_connect`，在 Sensing 中勾选 `voxel_grid_filter` 和 `ring_ground_filter` 优化激光雷达点云数据的处理流程，提升感知效率和准确性。

(5) 在 Autoware 的 Computing 模块中勾选 `waypoint-saver` 录制轨迹点后，选择 `waypoint-loader` 加载参考轨迹点作为控制算法跟踪的期望轨迹。

(6) 跟踪控制算法主要实现在 `mpc_follower` 节点中，该节点接收 `/vehicle_status`、`/vehicle_cmd` 和 `/trajectory` 等消息，配置 `lane_rule`、`lane_stop`、`lane_select`、`velocity_set` 和 `twist_filter`。

(7) 在终端中输入 `source devel/setup.bash` 和 `roslaunch pix_can pcanwrite.launch` 打开 `pcan` 驱动，启动底盘自动驾驶模式，打开 topics 中的 `rosviz` 记录轨迹。录制结束后通过 `Rviz` 查看底盘跟踪路径的过程。



图 5-9 Autoware 计算模块界面

5.2.2 试验场景搭建与结果分析

(1) 试验场景搭建

在浙江亚太机电公司的厂区内，由于测试环境的限制，搭建了一个临时的障碍物避让试验场。场地中放置了 4 个锥桶，模拟障碍物，并将其安置在车辆前方约 20 米的位置，用于测试车辆避障和跟踪的能力。通过遥控切换到自动驾驶模式，为了保证试验过程的安全性，将车速设置为 5m/s。在试验过程中可以通过 ROS 可视化界面观察底盘行驶状态，并将行驶数据保存用于分析本文设计的规划与控制算法的实际应用效果。其中图 5-10 中展示了无人车在避开障碍物时的起点和避障过程。



(a) 车辆开始位置



(b) 车辆避障 1



(c) 车辆避障 2



(d) 车辆避障 3

图 5-10 线控底盘避障过程

(2) 实验结果分析

在实车试验结束后，将 ROS 中保存数据绘制成图 5-9 所示，其中图 5-9 (a) 中红色实线是参考轨迹，蓝色实线是实际行驶轨迹。由 5-9 (a) 可以看出本文的规划算法可以躲避障碍物并顺利到达指定位置，规划完成后，将此路径保存作为轨迹跟踪控制器的参考轨迹，由图中可以看出参考轨迹和实际行驶轨迹的误差很小，表明本文设计的控制算法保证了无人车在行驶过程中的跟踪精度。

图 5-9 (b)、(c) 和 (d) 是在启动 PCAN 驱动后在底盘行驶时显示在终端的四元数组。其中图 5-9 (b) 是车辆的翻滚角，翻滚角是无人车绕纵轴的旋转角度，主要反映了转弯时车身的侧倾程度，由图中可以看出翻滚角的变化范围在 $-0.025^{\circ} \sim 0.015^{\circ}$ ，角度变化幅度很小，满足无人车在低速行驶时的侧倾角范围。图 5-9 (c) 是车辆的俯仰角，是无人车绕横轴的旋转角度，俯仰角的变化范围在 $-0.015^{\circ} \sim 0.02^{\circ}$ ，由 (b) 和 (c) 可以看出底盘在行驶过程很平稳。

图 5-9 (d) 是车辆偏航角以及偏航角的变化率，偏航角是无人车实际前进方

向与目标方向之间的偏差，由图（d）可以看出底盘在两次转弯时，偏航角有两次变化，变化幅度较小，偏航角的变化率也有明显变化，但都在合理范围内。

综上所述，通过实车试验结果可知，不论是在直线行驶和转弯时，底盘在行驶过程中的翻滚角、俯仰角和偏航角的变化幅度很小并且均处于合理范围之内，表明本文设计的规划和控制算法不仅可以保证车辆在行驶过程中的跟踪精度，还可以保证无人车行驶过程中的安全性和稳定性，因此本文设计算法具有一定的实用性和可行性。

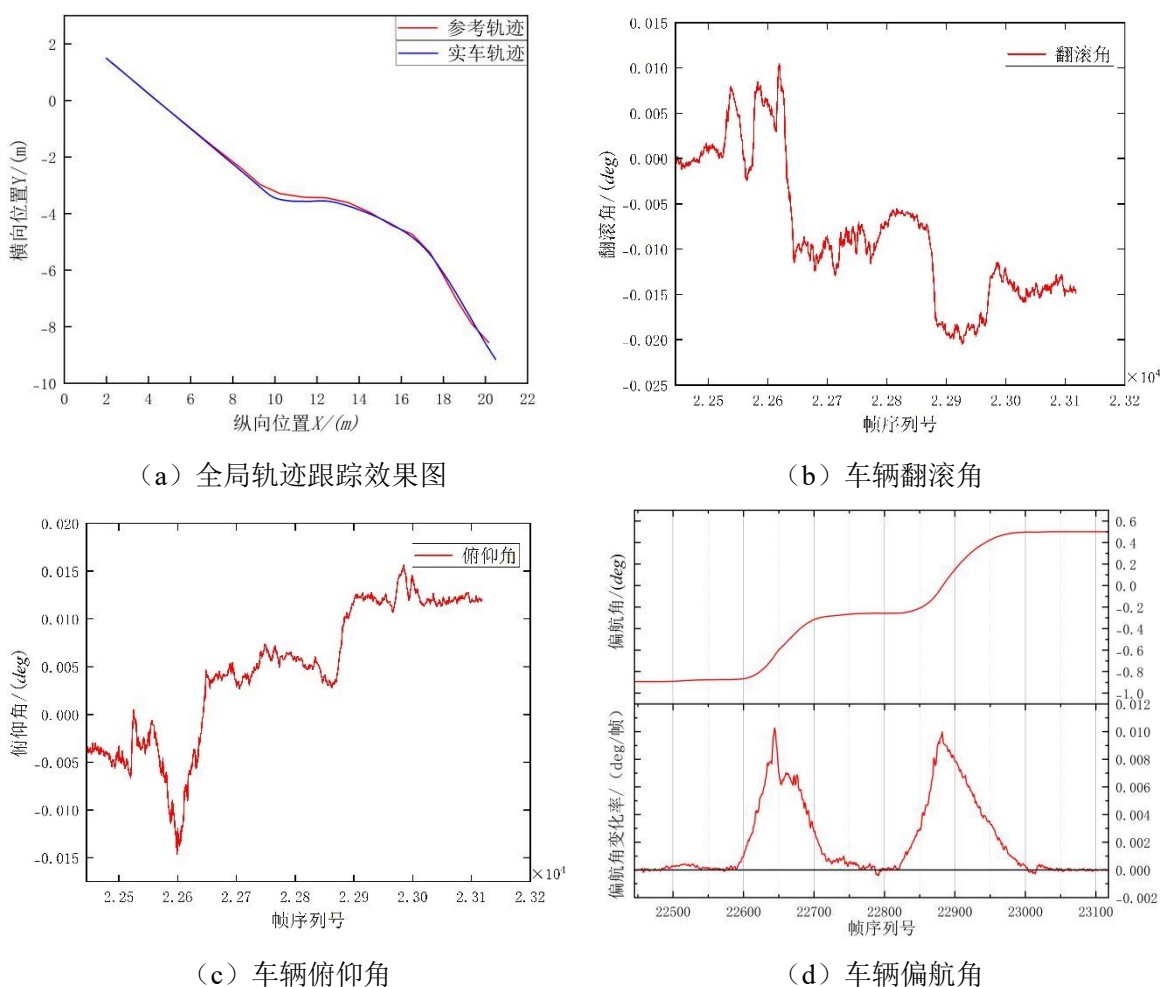


图 5-12 实车试验结果图

5.3 本章小结

本章通过使用线控底盘对本文中的规划与控制算法进行实车验证，并介绍了实车试验中需要用到的软硬件平台。

(1) 搭建实车试验场景后通过激光雷达对试验场地建立点云地图，并将在 ROS 中建立的全局路径规划算法包进行编译并启动 Autoware。并将底盘行驶切

换至自动驾驶模式，将底盘行驶的路径点保存后作为控制算法的参考轨迹。由底盘行驶的路径可以看出，本文设计的全局路径规划算法可以有效避障，保证了无人车在行驶过程中的安全性。

（2）在 **Autoware** 中加载点云地图和参考轨迹，并配置跟踪控制算法，对本文设计的轨迹跟踪控制算法的可行性进行验证。试验结果表明，不论是在直线行驶和转弯时，底盘在行驶过程中的翻滚角、俯仰角和偏航角的变化幅度很小并且均处于合理范围之内，保证了无人车在行驶过程中对参考路径的跟踪精度和行驶稳定性。

第 6 章 总结与展望

6.1 全文总结

无人驾驶车辆技术作为未来交通领域的重要发展方向,本文选取功能型无人车的路径规划和轨迹控制算法作为主要研究方向,路径规划负责在复杂的交通环境中为无人驾驶车辆设计出最优的行驶路线,确保车辆能够有效避开障碍物并顺利到达目的地;而轨迹控制则确保车辆在实际行驶过程中,能够按照规划的路径精确地进行控制,保证驾驶的稳定性与安全性,本文研究内容如下。

(1) 通过优化 A*算法代价函数中 $H(n)$ 的权重,优化后的算法冗余节点减少,转折点也有所减少,解决了传统 A*算法的遍历节点多、转折点多等问题。通过设置安全距离 D,无人车不会贴合障碍物行驶,保证了车辆的安全性。改进后的 A*算法提高了无人车的行驶稳定性和安全性。将改进后的 A*算法与 DWA 算法进行融合,其中路径长度与传统 DWA 算法相比缩短了 6.5%,搜索效率提升了 40.68%,融合算法解决了 A*算法无法躲避动态障碍物和 DWA 算法易陷入局部最优的问题,同时又提高了无人车的行驶稳定性和安全性。

(2) 建立车辆三自由度模型,构建了基于LTV-MPC轨迹跟踪控制器。采用前轮转角为控制量建立LTV-MPC控制系统,采用T-S模糊控制策略,对LTV-MPC的目标函数权重进行在线优化,实时获取的横向位移偏差与横摆角误差数据被用作模糊系统的输入变量。构建具有在线自适应能力的权重矩阵优化框架。实现了目标函数中轨迹跟踪精度与车辆横向稳定性指标的动态平衡有效提升了控制系统的工况适应能力。在Simulink/CarSim联合仿真平台中搭建基于T-S FMPC的控制模型,仿真结果表明,在车速 48 km/h双移线工况下, T-S FMPC的最大横向误差较LTV-MPC降低了 47.6%,最大横摆角跟踪误差较LTV-MPC降低了 85.3%, T-S FMPC的最大质心侧偏角相比LTV-MPC则降低了 42.6%, T-S FMPC横摆角速度相比于LTV-MPC降低了 36.1%,说明本文所设计的控制器显著提升了轨迹跟踪的精确度,并增强了车辆在行驶过程中的稳定性。

(3) 通过线控底盘对规划与控制算法进行实车验证,介绍了实车试验所需的硬件平台搭建和软件系统架构。首先通过激光雷达对试验场地建立点云地图,

在 Autoware 中加载点云地图和其他配置进行实车试验，针对本文所涉及的路径规划算法与轨迹跟踪控制算法的可行性进行验证，试验结果表明，本文所设计的规划算法能够躲避障碍物顺利到达目标位置，并且跟踪控制算法能够较好地跟踪规划轨迹。

6.2 研究展望

本文主要对规划和跟踪控制算法进行研究，但仍存在一定的局限性，需要进一步研究，具体内容如下：

（1）在轨迹跟踪控制算法方面，本文提出基于 T-S 模糊控制使 MPC 中目标函数权重系数自适应，提高了跟踪控制器的跟踪精度和行驶稳定性。但是未来如果需要研究更复杂的系统时，模糊规则的设计也越来越多，计算量会增加，实时性可能就没法保证。进一步提升算法的实时性和稳定性仍是未来研究的关键方向。对于这个问题，后续可以结合神经网络，作为模糊控制器的一部分，在训练过程中优化规则的生成。

（2）在路径规划算法方面，实车试验时由于场景的限制，只能在静态障碍物下进行试验，不能很好的验证本文提出的躲避动态障碍物的情况，由于实车试验为理论研究提供了重要的实践依据，是确保技术可行性、提升系统性能、发现潜在问题以及推动技术应用的关键步骤，未来将进一步在动态环境下进行验证算法的有效性。

参考文献

- [1] 陆淼嘉, 尹钦仪. “最后一公里”无人车配送发展现状及应用前景[J]. 综合运输, 2021, 43(01):117-121.
- [2] 陈慧岩, 熊光明, 龚建伟主编. 无人驾驶车辆理论与设计[M]. 北京: 北京理工大学出版社, 2018: 5-15.
- [3] 江宏. 物流无人化发展状况与趋势展望[J]. 物流技术与应用, 2019, 24(02):90-93.
- [4] 李振中, 许设, 马文博. 无人物流车末端配送研究[J]. 中国汽车, 2021, No.353(08):14-17.
- [5] Mi F, Wang B, Wang C. Research on Logistics Avoidance Strategy of Campus Unmanned Distribution Based on Markov Decision Process[P]. 2022 6th International Workshop on Materials Engineering and Computer Sciences, 2022.
- [6] 孟祥雨, 张成阳, 苏冲. 自动驾驶汽车系统关键技术综述[J]. 时代汽车, 2019, No.320(17):4-5.
- [7] 张曦. 基于模型预测原理的无人车局部路径规划算法研究[D]. 重庆: 重庆理工大学, 2020.
- [8] 蔺博见. 基于改进 A* 算法和人工势场法的搬运机器人路径规划研究[D]. 西安: 长安大学, 2017.
- [9] Kavraki L E, Svestka P, Latombe J C, etc. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces[J]. IEEE Transactions on Robotics and Automation, 1996, 12(4):566-580.
- [10] LAVALLE S M. Motion planning[J]. IEEE Robotics & Automation Magazine, 2011, 18(1): 79.
- [11] GONG H, NI C, WANG P, etc. A smooth path planning method based on Dijkstra algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2024, 50 (2):535-541.
- [12] Xing Xu, Jiazhu Zeng, Yun Zhao, etc. Research on global path planning algorithm for mobile robots based on improved A*[J]. Expert Systems with A

- pplications, Volume 243, 2024, 122922, ISSN 0957-4174.
- [13] 杨芳清, 刘吉成. 融合改进 A* 算法与动态窗口法的移动机器人路径规划[J]. 工业控制计算机, 2021, 34(5): 106-108, 112.
- [14] X. Zhou, X. Luo and Y. Zhang, A Path Planning Algorithm Based on Improved RRT[J]. 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2022, pp. 1135-1140.
- [15] 孙馨宇, 徐家川, 焦学健, 等. 基于改进 Informed-RRT* 的路径规划算法研究[J/O L]. 电子测量技术, 1-10[2025-05-07]. <http://kns.cnki.net/kcms/detail/11.2175.TN.20250221.1609.027.html>.
- [16] Tan X, Han L, Gong H, etc. Biologically Inspired Complete Coverage Path Planning Algorithm Based on Q-Learning[J]. Sensors 2023, 23, 4647.
- [17] 张浩, 刘薇. 基于改进蚁群算法的机器人路径规划[J]. 科学技术与工程, 2025, 25(03): 1142-1149.
- [18] 江明, 王飞, 葛愿, 等. 基于改进蚁群算法的移动机器人路径规划研究[J]. 仪器仪表学报, 2019, 40(02): 113-121.
- [19] 薛翔, 朱其新, 朱永红. 基于改进蚁群算法的机器人路径规划[J]. 西安工程大学学报, 2024, 38(4): 1-8.
- [20] Fu, S., Robot Path Planning Optimization Based on RRT and APF Fusion Algorithm[J]. 2024 8th International Conference on Robotics and Automation Sciences, ICRAS 2024, 32-36.
- [21] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance[J]. IEEE Robotics & Automation Magazine, 1997, 4(1): 23-33.
- [22] 蒋华扬, 吴涛, 任建鑫, 等. 基于改进 DWA 算法的移动机器人局部路径规划[J]. 机械设计与研究, 2024, 40(05): 229-234.
- [23] 董晓东, 李刚, 宗长富, 等. 融合改进 A* 算法和 DWA 算法的全局动态路径规划[J]. 重庆理工大学学报(自然科学), 2024, 38(02): 77-86.
- [24] 程传奇, 郝向阳, 李建胜, 等. 融合改进 A* 算法和动态窗口法的全局动态路径规划[J]. 西安交通大学学报, 2017, 51(11): 137-143.
- [25] Xiu Y, Wei C, Yi Z, etc. "Mobile robot path planning in complex environmen

- t[J].2019 IEEE International Conference on Unmanned Systems (ICUS),Beijing, China, 2019, pp. 426-431.
- [26] KHODAYARI A, GHAFARI A, AMELI S, etc. A historical review on lateral and longitudinal control of autonomous vehicle motions[C]. 2010 International Conference on Mechanical and Electrical Technology, 2010: 421–429.
- [27] 汤新华, 成宇庆, 潘树国, 等.基于时滞补偿的纯追踪控制预瞄距离优化方法[J].中国惯性技术学报, 2023, 31(09):876-882.
- [28] CHENG S, LI L, CHEN X, etc. Model Predictive-Control-Based path tracking controller of autonomous vehicle considering parametric uncertainties and velocity-varying[J]. IEEE Transactions on Industrial Electronics, 2021, 68(9): 8698-8707.
- [29] 张栩源, 李军.基于 LQR 双 PID 的智能电动汽车轨迹跟踪横纵向协同控制[J].汽车安全与节能学报, 2021, 12(03):346-354.
- [30] Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots[C]. IEEE International Conference on Robotics and Automation, 2003:90-98.
- [31] 章仁燮, 熊璐, 余卓平, 等. 基于条件积分算法的无人驾驶车辆轨迹跟踪鲁棒控制方法[J].机械工程学报, 2018, 54(18): 129–139.
- [32] Huang Y, Yasunobu S, A general practical design method for fuzzy PID control from conventional PID control[J]. Ninth IEEE International Conference on Fuzzy Systems, 2000:969-972.
- [33] 何慧玲. 智能车辆局部轨迹规划与跟踪控制算法研究[D].哈尔滨: 哈尔滨工业大学, 2021.
- [34] COULTER R C. Implementation of the pure pursuit path tracking algorithm[R]. Pittsburgh: Carnegie Mellon University Pittsburgh PA Robotics Institute, CMU-RI-TR-92-01, 1992.
- [35] Shan, Y. CF-Pursuit: A Pursuit Method with a Clothoid Fitting and a Fuzzy Controller for Autonomous Vehicles[J]. International Journal of Advanced Robotic Systems, 2015, 12(9):134.
- [36] 周卫琪, 赵羿寒, 刘擎超, 等.基于改进 LQR 的车辆路径跟踪横向控制策略

- [J].华中科技大学学报(自然科学版), 2024, 52(03):135-141.
- [37] 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.
- [38] Taghavifar, H.Neural Network Autoregressive With Exogenous Input Assisted Multi-Constraint Nonlinear Predictive Control of Autonomous Vehicles[J]. IEEE Trans. Veh. Technol. 2019,68(7):6293–6304.
- [39] Yoo, S.J., Choi, Y.H., etc. Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach[J]. IEEE Transactions on Circuit sand Systems I: Regular Papers ,2006,53(6):1381–1394.
- [40] 吴西涛, 魏超, 翟建坤, 等. 考虑横摆稳定性的无人车轨迹跟踪控制优化研究[J]. 机械工程学报, 2021:1-13.
- [41] KAYACAN E, RAMON H, SAEYS W. Robust trajectory tracking error model-based predictive control for unmanned ground vehicles[J]. IEEE/ASME Transactions on Mechatronics, 2016, 21(2): 806–814.
- [42] 陈梓宁, 童亮, 李晓东, 等.自适应时域模型预测控制的轨迹跟踪控制[J].重庆理工大学学报(自然科学), 2024, 38(05):78-85.
- [43] 张修伟, 陈龙, 孙桓五, 等.基于路面附着系数估计的模糊自适应轨迹跟踪控制[J].重庆理工大学学报(自然科学), 2024, 38(06):92-101.
- [44] Rajamani R.Vehicle Dynamics and Control[M]. Springer Science & Business Media, 2011.
- [45] OZGUNER U, UNYELIOGLU K A, HATIPOGLU C. An analytical study of vehicle steering control[C]. Proceedings of International Conference on Control Applications, 1995: 125–130.
- [46] 陈娇, 徐菱, 陈佳, 等.改进 A~*和动态窗口法的移动机器人路径规划[J].计算机集成制造系统, 2022, 28(06):1650-1658.
- [47] 张珂.无人驾驶车辆路径规划算法综述[J].装备制造技术, 2021(06):111-113.
- [48] 孙玉泽. 无人轮式车辆越野路面全局路径规划与轨迹跟踪[D]. 吉林大学, 2020.

- [49] 王海, 徐康俊, 蔡英凤. 面向多工况的智能车换道轨迹规划方法[J]. 江苏大学学报:自然科学版, 2019, 040(003):255-260.
- [50] You F, Zhang R, Lie G. Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system[J]. Expert Systems with Applications An International Journal, 2015, 42(14):5932-5946.
- [51] 张振, 张华良, 邓永胜, 等.融合改进 A*算法与 DWA 算法的机器人实时路径规划[J].无线电工程, 2022, 52(11):1984-1993.
- [52] 刘建娟, 薛礼啟, 张会娟, 等.融合改进 A~*与 DWA 算法的机器人动态路径规划[J].计算机工程与应用, 2021, 57(15):73-81.
- [53] 王海瑶, 安天洋.基于改进 A*算法融合 DWA 算法的机器人路径规划[J].电子制作, 2023, 31(21):48-51.
- [54] 陈冬冬.基于改进 DWA 算法的局部避障路径规划研究[J].汽车实用技术, 2024, 49(04):25-29.
- [55] 张晓熠. 融合动态窗口法与 A*算法的港口 AGV 路径规划方法研究[D]. 北京交通大学, 2019.
- [56] F. Zhang, N. Li, T. Xue, etc.An Improved Dynamic Window Approach Integrated Global Path Planning[J].2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 2019, pp. 2873-2878.
- [57] J. Hu, C. Cheng, C. Wang, etc. An Improved Artificial Potential Field Method Based on DWA and Path Optimization[J].2019 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 2019, pp. 809-814.
- [58] 董翼宁, 曹景胜, 李刚.融合 A-Star 与 DWA 双优化算法的自动引导车路径规划[J].科学技术与工程, 2023, 23(30):12994-13001.
- [59] 董晓东, 李刚, 宗长富, 等.融合改进 A~*算法和 DWA 算法的全局动态路径规划[J].重庆理工大学学报(自然科学), 2024, 38(02):77-86.
- [60] 庞永旭, 袁德成.融合改进 A~*与 DWA 算法的移动机器人路径规划[J].计算机与现代化, 2022(01):103-107.
- [61] 孙银健.基于模型预测控制的无人驾驶车辆轨迹跟踪控制算法研究[D].北京

理工大学, 2015.

- [62] 谢兼明.无人驾驶车辆路径跟踪模型预测方法研究[D].北京理工大学, 2018.
- [63] Stano P, Montanaro U, Tavernini D, etc. Model predictive path tracking control for automated road vehicles: A review[J]. Annual Reviews in Control, 2023, 55(4):194–236.
- [64] 庞辉, 刘楠, 刘敏豪, 等.基于线性时变模型预测控制的自主车辆轨迹跟踪控制器设计与验证[J].机械工程学报, 2022, 58(24):264-274.
- [65] RAFAILA R C, LIVINT G. Nonlinear model predictive control of autonomous vehicle steering[C]. 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), 2015: 466–471.
- [66] 王银, 张灏琦, 孙前来, 等.基于自适应 MPC 算法的轨迹跟踪控制研究[J].计算机工程与应用, 2021, 57(14):251-258.
- [67] 王俊杰.自动驾驶车辆自适应轨迹跟踪控制算法研究[D].南京理工大学, 2021.
- [68] 石辛民等. 模糊控制及其 MATLAB 仿真[M]. 清华大学出版社, 北京交通大学出版社, 2008: 20-45.
- [69] 张丽霞, 田硕, 潘福全, 等.基于 MPC 的智能车辆路径规划与跟踪控制[J].河南科技大学学报(自然科学版), 2024, 45(01):1-11+117.
- [70] 黄正旭, 周坤, 王斌锐.基于模糊自适应模型预测控制的移动机器人路径跟踪控制[J].中国计量大学学报, 2023, 34(03):405-411.

攻读学位期间发表的学术论文目录

一、发表的学术论文

[1] 许柳柳, 杨爱喜, 臧豫徽. 基于改进A*和DWA算法的无人车路径规划[J]. 安徽工程大学学报, 2025, 40(01): 1-7+30.

[2] Yang A, Zang Y, Xu L, Li L, Tan D. A Systematic Review and Future Development of Automotive Chassis Control Technology. Applied Sciences. 2023; 13(21):11859. <https://doi.org/10.3390/app132111859>.

二、参与的科研项目

[1] 无人驾驶自卸充电车, 企业自选研发项目, 项目编号: SZC300010.

[2] 杭州市智能网联汽车安全检测平台, 企业自选研发项目, 项目编号: SZC30008.

三、相关标准的参与建设

[1] DB 3301/T 0491—2025, 智能网联车辆开放道路环境评估与分类规范[S].

四、申请及获得的发明专利

[1] 一种基于相机的目标检测与跟踪方法及系统, CN202211730293.X

[2] 一种基于相机的二维目标检测与跟踪方法及系统, CN202211573310.3

致谢

经过冬，走到夏，等枝桠，成繁花。

2025 年盛夏，完成了硕士阶段的求学生涯。

得遇良师，人生之幸。感谢我的导师杨爱喜教授，感谢您为我提供的实习平台和科研环境，感谢您在论文选题和论文撰写等阶段给予了我极大的鼓励和耐心的指导。还要感谢给予我学术和科研帮助的时培成教授，感谢您在我研一期间的指导和帮助。行将毕业，衷心的祝福导师们工作顺利，身体健康。

焉得萱草，言树之背。感谢我的父母和家人们，见证了我的成长，教会我正直、真诚地对待别人。谢谢你们一路以来默默的陪伴，在我迷茫时做我最坚定的靠山，家永远是我自信勇敢的底气和最重要的支撑。自此落笔，只希望我努力成长也能成为他们的依靠。

人生海海，感恩相遇。很幸运在读研究生涯中结识到许多志同道合的好友，感谢我的师兄臧豫徽和黄瑞钦，对我学习和生活中的悉心指导和帮助。感谢我的师姐江彤，带给我很多笑容和温暖，以及给予我很多帮助的师兄们。感谢实习期间一直陪伴在我身边，给予我无限包容、安慰和鼓励的张绍丹，以及其他一起实习的小伙伴们。感谢课题组的同门周梦如、李帅、潘艺鑫、吴文超、董心龙和杨礼，一路走来，我们互帮互助，留下了很多美好且珍贵的回忆。

愿岁并谢，与友长兮。感谢我的好朋友们周梦如、张胜楠、张立慧、姚雪、王玉茹和侯艳婷一直以来的陪伴，不厌其烦的听我诉说我的喜怒哀乐。我们一起分享着彼此的青春，相互鼓励、共同奋斗着，祝我们的友谊永远看不到尽头，在未来漫长的日子里，继续相伴而行。

凡是过往，皆为序章。最后感谢不曾放弃的自己，希望自己能够继续保持真诚热烈的人生态度，不忘初心，坚定前行。

往事堪堪亦澜澜，前路漫漫亦灿灿。

尚德敏学 唯实惟新

