

分类号: TP242.6

密 级: 公开

学校代码: 10363

学 号: 2220320175



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题目 基于 ACO-DWA 融合算法的
智能车路径规划研究

论文作者 汤俊杨

校内导师 王正刚（副教授）

校外导师 毛德平

专业学位类别 电子信息（085400）

研究方向（领域） 人工智能

论文提交日期: 2025 年 5 月 16 日

分类号: TP242.6

单位代码: 10363

密 级: 公开

学 号: 2220320175

题 目 基于 ACO-DWA 融合算法的智能车路
径规划研究

题 目 Research on Path Planning of Intelligent
Vehicles Based on the Fusion Algorithm
of Ant Colony Optimization (ACO) and
Dynamic Window Approach (DWA)

论文作者: 汤俊杨

校内导师: 王正刚

校外导师: 毛德平

专 业: 电子信息

研究方向: 人工智能

论文答辩日期: 2025 年 5 月 12 日

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：



日期：2025 年 5 月 16 日

安徽工程大学学位论文版权使用授权书

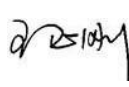
学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

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： 

指导教师 

日期： 2025 年 5 月 16 日

日期： 2025 年 5 月 16 日

基于 ACO-DWA 融合算法的智能车路径规划研究

摘 要

在现代科技蓬勃发展的大背景下，智能车在工业、物流等领域的应用愈发广泛，其路径规划能力成为决定任务执行效率的核心要素。传统路径规划算法在复杂室内环境中存在易陷入局部最优、实时响应效率低等问题，严重制约了智能车的性能。随着智能车应用场景的不断拓展和环境复杂度的持续增加，对高效路径规划算法的需求更为迫切。针对这些问题，本文深入研究智能车路径规划算法，旨在提升其在复杂环境下的路径规划能力与工作效率，具体研究内容如下：

（1）在全局路径规划方面，针对传统蚁群算法（Ant Colony Optimization, ACO）存在收敛速度慢、易陷入局部最优解的问题，提出了一系列改进措施。本文设计了融合风险评估与历史频率因素的势场启发函数，使蚂蚁在路径选择时能有效避开高风险区域，加快算法收敛速度的同时提高了路径安全性；引入信息素差分机制，避免算法过早陷入局部最优；采用挥发系数自适应调整机制，平衡全局与局部搜索能力；运用剪枝策略，去除冗余转折点，优化全局路径。在 MATLAB 环境下的仿真实验表明，改进后的蚁群算法在路径长度、迭代次数等关键指标上表现优异，显著优于传统算法和其他相关算法。

（2）在局部路径规划方面，针对传统动态窗口法（Dynamic Window Approach, DWA）在局部路径规划中的不足，如缺乏全局路径引导、对动态障碍物适应性差等问题，本文同样提出了有效的改进

策略。引入全局偏差函数，综合考虑机器人当前位置与全局路径点的距离和方向夹角，使智能车运动轨迹更好地贴合全局路径；改良障碍物评价函数，运用运动学公式预测未知移动障碍物位置，提升对动态障碍物的应对能力，实现动态避障；采用局部目标点自适应插入法，在障碍物附近智能增设局部目标点，引导机器人回归全局路径，防止路径偏离；提出多智能车优先避障策略，综合考虑路径长度和转折点数量对智能车进行优先级排序，实现多智能车高效协调避障。

（3）为验证改进算法的实际有效性，本文基于 ROS(Robot Operating System)平台设计了智能车自主导航实验。搭建了包含 Gazebo 仿真软件和 Rviz 可视化工具的 ROS 智能车仿真实验平台，利用 gmapping 算法构建全局地图，通过 AMCL(Adaptive Monte Carlo Localization)定位算法实现智能车自主导航。在静态和动态环境下的实验结果显示，改进后的 ACO-DWA 融合算法能够有效帮助智能车在未知环境中实现协同避障，可以实现全局最优，并且具有高安全性，显著提升了复杂场景下的路径规划能力和工作效率。

本文通过对 ACO 和 DWA 算法的改进与融合，为智能车路径规划提供了更有效的解决方案。未来，可进一步优化算法性能，增强其对复杂环境的适应性，拓展多智能车协作能力，并融合新兴技术，推动智能车路径规划领域的持续发展。

关键词：智能车，路径规划，蚁群算法，动态窗口法，ROS

RESEARCH ON PATH PLANNING OF INTELLIGENT VEHICLES BASED ON ACO-DWA FUSION ALGORITHM

ABSTRACT

In the context of the vigorous development of modern technology, intelligent vehicles are being increasingly applied in industries such as industry and logistics. Their path planning ability has become a core factor determining the efficiency of task execution. Traditional path planning algorithms have issues like being prone to getting trapped in local optima and having low real-time response efficiency in complex indoor environments, which severely limit the performance of intelligent vehicles. With the continuous expansion of the application scenarios of intelligent vehicles and the continuous increase in environmental complexity, there is an even more urgent need for efficient path planning algorithms. To address these issues, this paper conducts an in-depth study of the path planning algorithms for intelligent vehicles, aiming to enhance their path planning ability and work efficiency in complex environments. The specific research contents are as follows:

(1) In terms of global path planning, aiming at the problems of slow convergence speed and easy trapping in local optimal solutions of the traditional Ant Colony Optimization (ACO) algorithm, a series of

improvement measures have been proposed. In this paper, a potential field heuristic function that integrates risk assessment and historical frequency factors is designed, enabling ants to effectively avoid high-risk areas during path selection, accelerating the convergence speed of the algorithm and improving the path safety at the same time. A pheromone difference mechanism is introduced to prevent the algorithm from falling into local optimality prematurely. An adaptive adjustment mechanism for the evaporation coefficient is adopted to balance the global and local search capabilities. A pruning strategy is applied to remove redundant turning points and optimize the global path. Simulation experiments in the MATLAB environment show that the improved ant colony algorithm performs excellently in key indicators such as path length and the number of iterations, significantly outperforming the traditional algorithm and other related algorithms.

(2) In terms of local path planning, aiming at the deficiencies of the traditional Dynamic Window Approach (DWA) in local path planning, such as the lack of global path guidance and poor adaptability to dynamic obstacles, this paper also proposes effective improvement strategies. A global deviation function is introduced, comprehensively considering the distance and direction angle between the current position of the robot and the global path points, enabling the movement trajectory of the intelligent vehicle to better conform to the global path. The obstacle evaluation

function is improved, and kinematic formulas are used to predict the positions of unknown moving obstacles, enhancing the ability to deal with dynamic obstacles and achieving dynamic obstacle avoidance. A self-adaptive local target point insertion method is adopted, intelligently adding local target points near obstacles to guide the robot back to the global path and prevent path deviation. A priority obstacle avoidance strategy for multiple intelligent vehicles is proposed, comprehensively considering the path length and the number of turning points to sort the intelligent vehicles by priority, achieving efficient coordinated obstacle avoidance for multiple intelligent vehicles.

(3) To verify the practical effectiveness of the improved algorithm, this paper designs an autonomous navigation experiment for intelligent vehicles based on the ROS (Robot Operating System) platform. An ROS intelligent vehicle simulation experiment platform including the Gazebo simulation software and the Rviz visualization tool is built. The gmapping algorithm is used to construct a global map, and the AMCL (Adaptive Monte Carlo Localization) algorithm is used to achieve autonomous navigation of the intelligent vehicle. The experimental results in both static and dynamic environments show that the improved ACO-DWA fusion algorithm can effectively help intelligent vehicles achieve collaborative obstacle avoidance in unknown environments, can achieve a global optimum, and has high safety, significantly improving the path planning

ability and work efficiency in complex scenarios.

This paper provides a more effective solution for the path planning of intelligent vehicles through the improvement and integration of the ACO and DWA algorithms. In the future, the performance of the algorithm can be further optimized, its adaptability to complex environments can be enhanced, the collaborative capabilities of multiple intelligent vehicles can be expanded, and emerging technologies can be integrated to promote the continuous development of the field of path planning for intelligent vehicles.

KEY WORDS: intelligent vehicle, path planning, ant colony optimization algorithm, dynamic window approach, ROS

目 录

摘 要.....	III
ABSTRACT.....	V
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	2
1.2 国内外研究现状分析.....	3
1.2.1 智能车研究现状.....	3
1.2.2 路径规划算法研究现状.....	5
1.2.3 多智能车场景研究现状.....	8
1.3 主要研究内容和结构安排.....	9
第 2 章 智能车仿真环境与运动学分析.....	11
2.1 引言.....	11
2.2 阿克曼底盘运动学分析.....	11
2.3 环境地图建模.....	16
2.3.1 拓扑地图法.....	17
2.3.2 几何特征地图法.....	18
2.3.3 栅格地图法.....	18
2.4 智能车仿真软件.....	20
2.4.1 ROS 操作系统介绍.....	20
2.4.2 ROS 系统架构.....	20
2.4.3 ROS 话题通信机制.....	23
2.5 本章小结.....	24
第 3 章 基于改进蚁群算法的全局路径规划研究.....	25
3.1 引言.....	25
3.2 传统蚁群算法.....	25
3.3 改进的蚁群算法.....	26
3.3.1 启发函数的改进.....	27
3.3.2 信息素初始值差异化.....	28
3.3.3 信息素差分机制.....	29
3.3.4 自适应调整机制.....	30
3.3.5 剪枝策略.....	31
3.4 改进蚁群算法仿真实验与分析.....	33
3.5 本章小结.....	38
第 4 章 基于改进 DWA 算法的局部路径规划研究.....	39
4.1 引言.....	39
4.2 传统 DWA 算法.....	39
4.2.1 机器人运动模型.....	39
4.2.2 机器人速度采样.....	40
4.2.3 评估函数.....	41

4.3 改进的 DWA 算法	41
4.3.1 局部最优问题的解决	42
4.3.2 全局偏差函数	42
4.3.3 改进的 $dist(v, w)$ 函数	43
4.3.4 局部目标点的自适应插入法	45
4.3.5 多智能车的优先避障策略	46
4.3.6 总体改进算法的流程	47
4.4 IACO-IDWA 仿真实验与分析	48
4.5 多智能车优先级策略	52
4.6 本章小结	55
第 5 章 基于 ROS 平台的智能车自主导航	56
5.1 引言	56
5.2 系统基本介绍	56
5.3 搭建 ROS 仿真实验平台	58
5.4 基于激光雷达建图及导航实验	59
5.4.1 智能车模型建立	60
5.4.2 仿真环境地图的构建	61
5.4.3 基于 ROS 的自主导航仿真验证	62
5.5 本章小结	66
第 6 章 总结与展望	67
6.1 工作总结	67
6.2 未来展望	68
参考文献	69
攻读硕士学位期间取得的成果	75
致谢	76

第1章 绪论

1.1 研究背景及意义

1.1.1 研究背景

在科技蓬勃发展的时代，智能车作为多学科交叉融合的产物，正逐渐改变着工业生产、物流运输等众多领域的运作模式。智能车技术的发展依托于多个关键领域的进步，环境感知技术借助激光雷达、摄像头等多种传感器，如同智能车的“眼睛”，使其能够精准捕捉周围环境信息，实时感知行人、车辆等交通参与者的位置与状态；高精度定位技术依靠 GPS (Global Positioning System)、IMU (Inertial Measurement Unit) 以及高精度地图数据，为智能车在复杂环境中确定精确的位置和姿态，犹如为其指明方向的“指南针”；规划决策技术则是智能车的“大脑”，它基于感知系统收集的数据，进行快速分析与处理，制定出安全、高效的行驶决策，规划最优路径并计算理想控制指令，同时还能有效躲避障碍物，降低事故风险。

在实际应用场景里，智能车于工业与物流两大领域所发挥的作用日益关键。在工业生产环节，智能车能按照预先设定的程序与路径，有条不紊地执行物料搬运、设备巡检等工作。这不仅极大提升了生产效率，还显著削减了人力成本投入。就拿汽车制造工厂来说，智能车可精准地把零部件运送至既定位置，有力保障了生产线能够高效顺畅地运行。而在物流行业范畴，智能配送车的投入使用，切实攻克了长期以来困扰行业的“最后一公里”配送难题，使得配送效率与准确性得以大幅提升。像京东所推行的无人配送服务，配送机器人依靠先进的感知技术以及智能决策技术，能够自行规划最优路径，灵活避开各类障碍物，最终把货物安全无误地送到客户手中。

然而，智能车要实现高效、安全的运行，路径规划是核心关键。传统的路径规划算法在面对复杂多变的室内环境时，存在诸多局限性。复杂室内环境中往往存在大量形状各异、分布不规则的障碍物，且环境动态变化频繁，如人员走动、设备临时摆放等。传统算法在处理这类环境时，容易陷入局部最优解，导致规划

出的路径并非全局最优，使智能车行驶距离增加，能耗增大，工作效率降低。而且，传统算法的实时响应能力较差，当环境出现动态变化时，无法及时调整路径，可能导致智能车与障碍物发生碰撞，影响其运行安全和任务执行。例如，在仓储物流中心，当货架位置发生变动或出现临时障碍物时，传统算法可能无法及时为智能车规划出安全有效的路径。因此，为了满足智能车在复杂环境下的高效运行需求，对路径规划算法进行优化改进迫在眉睫。

1.1.2 研究意义

理论意义：本文融合蚁群算法与动态窗口法，应用于智能车路径规划领域，具重要理论价值。蚁群算法属群体智能启发式算法，模拟蚂蚁觅食行为，借信息素积累与挥发引导搜索路径，全局搜索和自适应性强。但在动态环境下，因信息素更新，收敛速度慢，实时性欠佳。动态窗口法基于机器人运动学模型，通过搜索速度空间的可行速度组合，能快速应对动态障碍与环境变化，实现实时运动控制，不过全局路径规划能力弱，易陷局部最优。融合二者可优势互补，为智能车路径规划提供更有效方案。

实际意义：从实际应用的角度来看，改进后的算法具有显著的应用价值。在工业生产领域，智能车借助优化后的路径规划算法，能够更加高效地完成物料运输任务，减少生产过程中的等待时间和能源消耗，进而提高整个生产线的运行效率，降低生产成本。例如，在电子制造工厂中，智能车可以快速、准确地将原材料运输到各个生产工位，避免因物料供应不及时导致的生产停滞，提高生产效率的同时，降低了能源成本。在物流配送方面，改进算法能够使智能配送车更合理地规划配送路线，避开拥堵路段和障碍物，实现货物的快速、安全送达，提升物流服务质量，优化物流资源的配置。此外，在军事和救援等特殊领域，智能车在执行任务时往往面临着复杂危险的环境，改进后的算法能够使其在复杂地形和多变环境中更高效地规划路径，减少人员伤亡风险，为保障人民生命财产安全和社会稳定发展提供有力支持。比如在地震、火灾等灾难救援场景中，智能车可以快速穿越危险区域，为救援人员提供关键信息，协助完成救援任务，提高救援效率，减少人员伤亡。

1.2 国内外研究现状分析

1.2.1 智能车研究现状

“机器人”这一术语诞生于 20 世纪。1920 年，捷克剧作家卡佩克（Capek）在其作品《罗萨姆万能机器人公司》中，首次创造了“robot”一词来命名机器人。

“robot”原意是强制劳动，这个词的出现，映射出人类长久以来渴望制造出一种类似人类、既能思考又能劳作的机器，用以替代自身工作的愿景^[1]。智能车做为机器人的典型应用，整合了多个核心技术范畴，重点包含环境感知、高精度定位、规划决策以及智能车线控技术。在环境感知技术方面，其依靠激光雷达、摄像头等各类传感器，对车辆周围环境数据进行收集与分析，从而能够实时监测周边行人、车辆等交通参与者的相关信息^[2]；高精度定位技术依靠 GPS、IMU 或高精度地图数据^[3]，精准估算智能车空间位置与姿态；决策规划技术作为自动驾驶核心，分析处理感知系统数据，实现快速安全驾驶决策，包括规划最优路径^[4]、计算理想控制指令、躲避障碍物及应对紧急状况降低事故风险。

随着全球科技持续迅猛发展，众多科研机构、企业以及国内外高校纷纷投身于智能车研究领域并深入探索。早在 20 世纪 70 年代初，美国斯坦福研究院的科学家便研制出名为 Shakey 的智能车^[5]，尽管只能在较为复杂的环境中解决一些简单的感知问题，完成简单的路径规划和控制效果，但已经将人工智能技术的实用化放入了机器人中，如图 1-1 所示。日本早稻田大学成功发明并研制出人形机器人。该机器人能够模拟人类行走，可在存在障碍物或凹凸不平的地面上执行移动作业，展现出卓越的机动性能与灵活性^[6]。

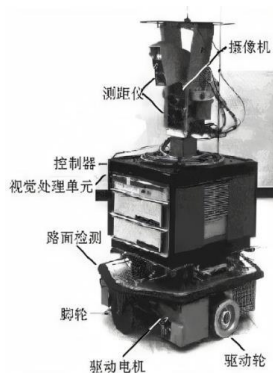


图 1-1 Shakey 智能车

自第一台机器人诞生后，德国率先提出将机器人应用于工业生产的构想，然而受当时机器人技术发展水平的限制，这一构想未能大规模推广。1973 年，日本学者加藤一郎成功研制出全球首台两足机器人 WABOT-1^[7]，以此为开端，机器人技术在日本迎来快速发展期。期间，Honda 公司研发的仿生机器人 ASIMO 以及 Sony 公司于 2000 年末推出的机器狗 AIBO，都产生了较大影响。

美国在机器人研发领域同样投入大量资金。进入 21 世纪，伴随计算机技术与传感器技术的迅猛发展，所研制的机器人功能愈发多样。2001 年 911 事件后研制出的战略机器人 PackBot^[8]可以帮助人们在复杂的场景中搜救被困人员。2006-2010 年，Roomba 扫地机器人实用化，DARPA 赛事推动机器人军事应用，工业 4.0 带动工业机器人智能化及协作机器人兴起。2011-2015 年，IBM 的 Watson 机器人获胜推动智能化，波士顿动力的机器人借助深度学习实现复杂动作；物流电商中亚马逊引入 Kiva 机器人，医疗领域美敦力的脊柱手术机器人获批临床。

加拿大 Clearpath Robotics 公司研发的 JACKAL 智能车，采用 ROS 机器人操作系统，功能包丰富。其外壳防水、底盘合金加固，动力强劲，配备激光雷达、GPS、摄像头等多类传感器，能胜任复杂工作，广泛用于工业制造、科研教育、物流运输领域^[9]，如图 1-2 所示。



图 1-2 JACKAL 智能车

我国于 20 世纪 70 年代开启机器人研究征程。彼时，国内工业基础薄弱，硬件制造工艺落后，计算机性能欠佳，且在运动控制、智能算法、传感器融合等关键技术上近乎空白，既缺理论积淀又无技术可鉴，致使机器人技术起步艰难，初期发展迟缓^[10]。从 20 世纪 80 年代末期起，国家日益重视机器人领域的发展。在科研人员的不懈努力下，中国机器人技术取得了丰硕成果。众多高校积极投身研发，陆续推出多功能机器人。1994 年，清华大学成功研制出 THMR-V 智能车^[11]，2000 年，国防科技大学研发出具备简易语音功能的机器人“先行者”^[12]。

2013 年，“玉兔号”月球车成功登陆月球。它配备全景相机、测月雷达等多

种先进传感器，出色完成月球表面探测、地球勘察任务，为科研提供关键数据。同年，中科院研制出基于激光雷达融合 3D 摄像头技术的自主导航智能小车，利用二者优势精准感知环境，实现自主路径规划，在多领域具应用潜力^[13]。

2017 年，京东于中国人民大学开展全球首次机器人无人配送服务实验并获成功。配送机器人配备激光雷达、全景相机、超声波传感器与 GPS 定位技术，具备强大感知和决策计算能力，能避障、规划最优配送路线，实现货物快速安全送达^[14]，如图 1-3(a)所示。

2021 年，菜鸟智能车入驻大学校园。学生在软件选定楼号，它就能送货到楼下。行驶中，智能车可在人车密集处自主规划路线、灵活调整。若遇学生故意挡路，还会语音提醒让路，如图 1-3(b)所示。

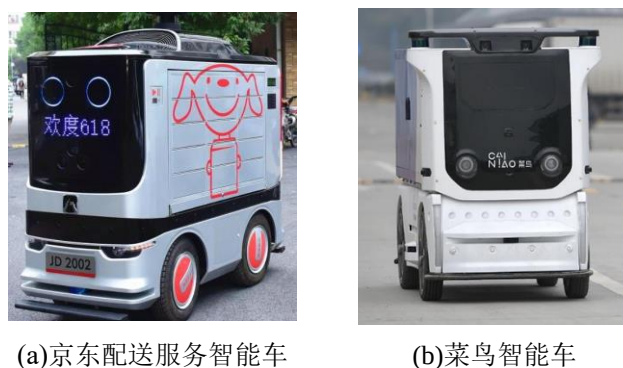


图 1-3 智能车应用

1.2.2 路径规划算法研究现状

移动机器人技术发展迅猛，在生活服务、工业等众多领域广泛应用^[15]。路径规划与避障技术作为智能车关键研究领域，直接决定了移动机器人的应用前景。路径规划的核心目的是依据环境信息，寻找从起始点到目标点的最优路径，同时可以有效的避开障碍物^[16]。路径规划依环境信息掌握度分两类。全局路径规划基于完整环境信息，像地图、障碍物位置等，从全局规划最优路线，追求最短路径、最少耗时等，给出宏观行动方案。局部路径规划针对未知或局部已知环境，靠传感器实时感知，动态生成路线，应对突发状况，确保在复杂环境安全前行^[17-18]。

常用的全局路径规划算法有 Dijkstra 算法^[19]、A*算法^[20]、遗传算法^[21]、粒子群算法^[22]、蚁群算法^[23-25]等。Dijkstra 算法作为一种贪心算法，是由 Johnson 在 1973 年提出^[26]。它以起始点为中心，将相邻节点作为搜索域，持续计算各节点

到目标点距离,按当前最短路径扩展,直至找到最短路径。该算法计算量大、耗时久,不适合实时性要求高的场景,如自动驾驶避障、实时游戏寻路。1968年, Peter E. Hart^[27]提出 A 算法,它在 Dijkstra 算法基础上引入启发式函数。Dijkstra 算法需计算所有节点,计算量大, A 算法则利用启发式函数,只聚焦可能在最短路径上的节点,大幅缩短计算时间,在机器人导航、游戏寻路等领域广泛应用^[28]。但 A 算法规划的路径拐点多。因为其选择节点时侧重启发函数与已走路径代价,忽视路径平滑。这导致移动主体行驶时要频繁转向,既增加能耗与机械磨损,又影响运行效率与安全。针对该算法的不足,槐创锋等人^[29]则扩大传统 A*算法搜索邻域、优化搜索角度,让路径平滑度显著提升,增强了算法在实际场景中的实用性。

蚁群算法融合分布式计算、正反馈和贪婪式搜索。众多蚂蚁并行探索,依信息素与局部最优选择路径,通过正反馈强化较优解搜索。但初始时信息素分布均匀,蚂蚁主要靠启发式信息盲目搜索,无效探索多,收敛到较优解耗时久,处理复杂问题难满足实时性^[30-31]。针对这些问题,王晓燕等人^[33]通过在启发式函数构造中引入人工势场,提出初始信息素不平衡原理,设置迭代阈值,并自适应调整信息素波动系数,提高了蚁群算法的全局搜索效率。刘瀛^[34]把改进的人工势场法和蚁群算法结合。人工势场将目标设为引力源、障碍物设为斥力源,引导蚁群避开障碍物,降低了算法陷入局部最优的概率。但该方法让路径拐角过多,移动主体行驶时需频繁转向,增加能耗与机械磨损。Miao 等人^[35]引入角度引导,让蚂蚁选节点时注重方向连贯,减少急转弯;提前移除部分影响小的障碍物,简化搜索空间;优化转移概率公式,综合多种因素,提升实时性与安全性。信息素更新时,采用自适应启发式,依算法阶段与路径质量调整启发信息权重,并用波动因子引入随机性,平衡算法收敛性与全局搜索能力,使规划路径更优。Zhai 等人^[36]采用结合人工势场和量子进化理论,使用网格法对环境进行建模,并通过量子信息素更新机制提高了蚁群算法的性能。

局部路径规划在智能移动体的运行过程中起着至关重要的作用,它致力于依据当下瞬息万变的实时环境状况,为移动体规划出一条从其当前所在位置直达目标位置的最优路径。相较于全局路径规划侧重于在已知的完整环境地图上进行宏观的路径构建,局部路径规划更聚焦于应对移动过程中随时出现的未知因素与动

态变化, 具有更强的实时性与灵活性。目前, 典型的局部路径规划算法有时间弹性带法(Teb)^[37]、动态窗口法(DWA)^[38-39]、人工势场法(APF)^[40]等。

时间弹性带算法(Timed Elastic Band, Teb)的核心思想是将路径表示为一系列具有时间戳的路点序列, 通过对这些路点的位置和时间进行优化, 以生成一条满足移动体动力学约束且适应环境变化的最优路径。它能够根据移动体的当前速度、加速度等动态参数以及周围环境中障碍物的实时位置信息, 动态调整路径的形状与行进时间。Wu 等^[41]针对 Teb 算法其过弯规划差, 智能小车迂回浪费资源。改进算法让小车沿墙过弯, 结合墙体约束优化轨迹。Rviz 平台实验显示, 改进后机器人能在复杂障碍环境平稳避障, 缩短规划时间、降低能耗。陈奕梅等^[42]指出, 动态环境中 Teb 算法响应不足, 轨迹与速度控制不佳。为此提出 TEB-VO 算法, 融合 VO 算法判断碰撞风险的优势, 通过自适应模块动态调整轨迹与速度。实验表明, 该算法使机器人轨迹平滑度提升约 30%, 速度波动降约 40%, 显著增强动态避障与运行稳定性。

动态窗口法(Dynamic Window Approach, DWA)基于移动体当前的速度和加速度限制, 在速度空间中定义一个动态窗口。在这个窗口内, 对移动体可能采取的不同速度组合进行模拟和评估, 计算每个速度组合下移动体在未来一段时间内的轨迹。通过考量轨迹与障碍物的碰撞风险、与目标位置的接近程度等因素, 选择出最优的速度组合, 进而确定移动体下一步的行动方向和速度。比如在一个拥挤的室内环境中, 机器人利用 DWA 算法能够实时感知周围行人的移动和障碍物的分布, 快速调整自身的速度和方向, 在避免碰撞的同时向着目标地点前进。Chang L 等^[43]提出基于 Q 学习的改进 DWA 算法, 对原评价函数进行拓展, 纳入障碍物密度、机器人能量等新因素; 明确状态空间(含位置、朝向、障碍分布)、动作空间(不同速度、转向组合)及奖励函数(前进避障给正奖, 反之给负奖)。借助 Q 学习自适应特性, 机器人在环境中试错、更新 Q 值表, 训练为能灵活应对的智能体。测试显示, 在复杂未知环境, 该算法导航效率较传统提升约 40%, 成功率从 60%左右跃升至 90%以上。

人工势场法(Artificial Potential Field approach, APF)原理是将目标点视为引力源, 对移动体产生吸引作用; 把障碍物看作斥力源, 对移动体施加排斥力。移动体在这两种力的合力作用下, 朝着目标点移动并避开障碍物。在规划路径时,

APF 算法根据移动体与目标点以及障碍物之间的相对位置和距离，计算出合力的大小和方向，从而引导移动体沿着合力方向移动。Sun 等^[44]提出在排斥力函数中引入无人机对目标点的吸引力。靠近障碍物时，此吸引力助无人机摆脱局部最优陷阱，避免被不合理合力误导；接近目标点阶段，它能抵消部分干扰斥力，让无人机顺利抵达目标，解决了传统方法的两大难题。

1.2.3 多智能车场景研究现状

多智能车（Automated Guided Vehicle，自动导引车）系统研究正处于蓬勃发展且成果丰硕的阶段，同时也面临诸多挑战。

在调度策略方面，优化算法层出不穷。浙江工业大学鲁建厦教授团队提出的考虑分配规则的多台智能车调度方法(MASA)，运用基于拥挤的分配规则(CAR)和基于开放时间的分配规则(OTAR)来灵活调整智能车工作路线，显著缓解工作站堵塞情况^[45]。该方法通过仿真实验，在特定场景下将智能车平均等待时间缩短了约 20%。同时，多目标调度成为热门研究方向，不再局限于传统的完成时间最短、路径最短等目标，而是兼顾低碳、能耗等。如《一种基于 DQN 的 Memetic 算法求解集成有限智能车的低碳作业车间调度问题》一文中，通过构建融合深度 Q 网络与 Memetic 算法的模型，致力于同时降低最大完工时间与总能耗^[46]。实验结果表明，相较于传统算法，该方法在能耗降低方面有显著提升，最多可降低 15% 左右的能耗。

路径规划领域，改进冲突消解策略成果显著。江苏大学汽车工程研究院提出的基于改进冲突搜索的路径规划模型(ICBS - PRI)，通过设计冲突检测子模块和冲突解决子模块，制定基于备用区域和旁路规划的策略，极大提升智能车库存取车效率^[47]。在某智能仓储模拟环境中，采用该模型后，出入库效率提高了约 30%。随着人工智能兴起，深度强化学习等先进技术被引入路径规划。如相关研究利用深度强化学习算法，使智能车能够在复杂动态环境中实时感知并规划路径，有效避开动态障碍物，增强了路径规划的适应性与实时性^[48]。

在系统建模与优化方面，Petri 网广泛应用于描述智能车系统状态变化与资源分配，助力分析死锁、冲突问题。如浙江大学利用 Petri 网研究自动化码头多

智能车系统死锁问题，提出了有效的死锁预防策略^[49]。仿真技术也不断发展，为虚拟建模、模拟场景、验证优化方案提供有力支持。例如，借助 Flexsim 软件对自动化立体仓库智能车调度系统建模与仿真，可直观呈现系统运行状况，提前发现潜在问题并优化参数^[50]，帮助企业在实际部署前预估系统性能，减少试错成本。

硬件与通信技术层面，智能车硬件设备全面升级，驱动、导航、传感器等性能提升。港口智能车有效载荷可达 70 吨，能满足重载作业需求^[51]。在导航精度上，部分先进智能车借助激光导航与视觉导航融合技术，定位精度可达 $\pm 5\text{mm}$ 。通信技术也借 5G 实现高速稳定传输，降低延迟，提升协同效率^[52]。在某大型工厂应用案例中，引入 5G 通信后，智能车之间数据传输延迟降低了 80%以上，系统响应速度大幅提升，整体生产效率提高约 10%。

不过，多智能车系统仍面临复杂环境适应性难题，如动态障碍物、光线与地面变化等干扰。面对复杂光照环境，智能车的视觉传感器易出现误判，导致导航偏差。大规模系统管理也颇具挑战，当智能车数量超过一定规模，调度算法复杂度呈指数级增长，需避免混乱与死锁以保障性能。此外，与其他系统的融合问题，如和仓储、生产管理系统的无缝对接，也亟待攻克。不同系统间数据格式、接口标准不一致，增加了系统集成难度。

1.3 主要研究内容和结构安排

首先，本文剖析课题研究的背景与意义，梳理智能车和路径规划领域国内外研究现状，明确融合改进蚁群和 DWA 算法用于 ROS 智能车路径规划的优势。随后，深入研究路径规划算法，将改进的蚁群算法与 DWA 算法融合实现导航避障。最后，对改进的算法展开实验验证与结果分析。本论文一共分为六个章节，各章节主要内容如下：

第 1 章：绪论。介绍研究背景与意义，涵盖智能车在多领域的应用及面临的挑战，以及本文的理论和实际价值。梳理国内外研究现状，明确融合改进算法的优势。介绍研究内容与章节安排。

第 2 章：聚焦智能车平台。基于阿克曼转向几何理论构建运动学模型，分析其运动规律。介绍环境地图建模方法，包括拓扑地图法、几何特征地图法和栅格

地图法。详细阐述智能车仿真软件 ROS 的操作系统、系统架构和话题通信机制。

第 3 章：基于改进蚁群算法的全局路径规划。介绍传统蚁群算法原理，用控制变量法获取较优参数。针对传统算法收敛慢等问题，改进启发函数、信息素差分机制、信息素更新规则及采用关键节点剪枝策略。通过仿真验证改进算法有效性。

第 4 章：基于改进 DWA 算法的局部路径规划研究。阐述传统 DWA 算法原理，针对其不足提出改进策略，包括解决局部最优问题、引入全局偏差函数、改进障碍物评价函数函数、采用局部目标点自适应插入法和多智能车优先避障策略。通过仿真实验验证改进 DWA 算法与改进蚁群算法融合后的性能提升。

第 5 章：基于 ROS 平台的智能车自主导航。介绍 ROS 平台的特点与系统框架，基于 Ubuntu 操作系统搭建 ROS 仿真实验平台。利用 Gazebo 搭建仿真环境，采用 gmapping 算法构建环境地图，在静态和动态环境下进行基于 ROS 的自主导航仿真验证，证明改进蚁群融合 DWA 算法的有效性。

第 6 章：总结与展望。总结针对传统蚁群算法和 DWA 算法提出的改进措施以及实验验证结果。展望未来研究方向，如优化算法性能、增强环境适应性、拓展多机器人协作、融合新兴技术及拓展应用等。

第2章 智能车仿真环境与运动学分析

2.1 引言

随着现代工业和服务行业的繁荣发展，智能车得到了越来越广泛的应用，其设计与运动学分析也备受关注^[53]。本章节围绕智能车系统展开深入探讨，内容涉及设计方法、软件架构等多个方面。特别基于阿克曼转向几何理论构建运动学模型，详细分析智能车的运动规律、位置关系和速度矢量变化，为精准导航与路径规划提供关键理论基础^[54]。同时，借助模块化设计的 ROS 框架，实现了软件系统的数据高效处理与任务合理管理。在这个框架下，各个功能模块相互协作，使得智能车能够快速、准确地处理各类信息，并做出合理的决策。上述一系列设计和分析工作，有效提升了智能车在复杂环境中的自主性和适应性，为智能车技术的进一步发展和更广泛应用奠定了坚实的理论与实践基础。

2.2 阿克曼底盘运动学分析

在智能车领域，阿克曼结构凭借其独特优势成为主流选择。该结构为四轮驱动转向的智能车提供了一种几何学解决方案，有效解决了内外轮径转向中心不一致的问题，极大地优化了机器人的转向运动性能。

从运动学角度出发，建立智能车的运动学模型是深入理解其运动特性的关键^[55]。基于阿克曼转角理论，本文对机器人的运动规律、物体间的位置关系以及速度矢量随时间的变化规律展开详细分析。这一分析方法能够精准揭示智能车在转弯过程中的关键规律，为其在路径规划时进行合理转向决策提供重要依据。

在建模过程中，为简化分析，做了一系列合理假设。通常情况下，仅考虑机器人在水平方向的运动，忽略竖直（ z 轴）方向的运动；在假设车轮转角一致性的前提下，只分析单侧车轮的转向响应，不考虑另一侧车轮的转向；同时，假设机器人的载荷分布均匀，且在转向过程中载荷保持稳定，不发生位移，并且车体和车架均为刚性结构。

在合理假设下，构建出阿克曼底盘的全驱动运动学模型，有三个输入和三个

输出。阿克曼结构后两轮与两轮差分底盘类似，负责提供动力和维持稳定。前两轮由舵机驱动，舵机转角值决定车辆行驶方向，与众多运动参数紧密相关。通过严谨数学推导，综合车辆前后轴距、后两轮间距、前两轮打角值、左右后轮线速度、车辆线速度和角速度等参数，运用运动学理论与数学工具，得出阿克曼结构运动学模型公式。该公式揭示了各参数内在联系，如已知部分参数，能算出转弯半径、角速度等，为底盘优化设计与精准控制提供理论支撑。

在惯性坐标系下，以变量下标区分前后位置，进一步确定前轴轴心坐标、后轴轴心坐标、前后轴中心速度、车辆前后两轴轴距、车辆姿态航向角、车辆前轮偏向角等参数。假设转向过程中车辆质心侧偏角保持不变，即车辆瞬时转向半径与道路曲率半径相同，在此基础上对车辆前后轴轴心处速度进行合成分析。

通过对车辆前后轮分别进行运动学分析，依据前后轮几何关系进行推导，最终整合得到车辆运动学模型。该模型精确描述了智能车在阿克曼底盘驱动下的运动状态，为后续的路径规划和运动控制提供了坚实的理论基础，有助于提升智能车在复杂环境中的运动规划能力和控制精度。

综合上述的假设对智能车设计运动学模型，阿克曼底盘的运动学模型如下图 2-1 所示，它是一个全驱动模型，有三个输入(v_L 、 v_R 、 θ)和三个输出(X 、 Y 、 θ)。

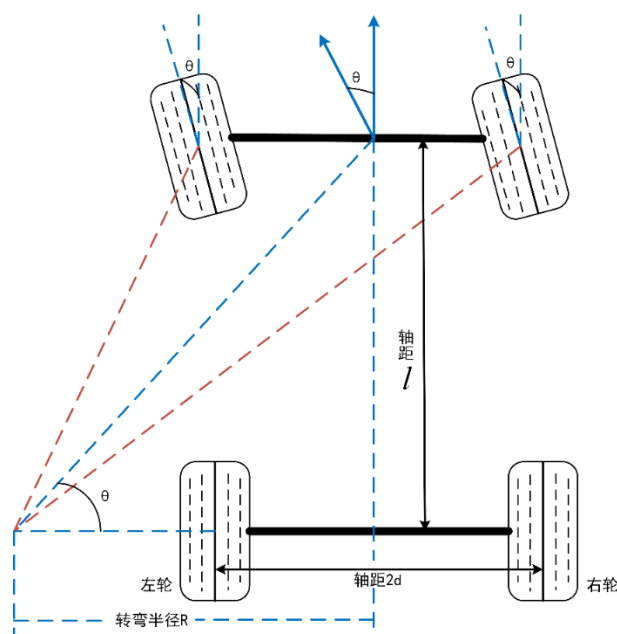


图 2-1 阿克曼底盘模型

阿克曼底盘转弯时，后侧两轮速度有差异，外侧快于内侧，确保平稳转向。结合简化运动学模型与物理知识，能推导出角速度和转弯半径相关信息，助力掌握车辆运动特性，利于运动控制与设计优化。

$$\omega = \frac{v}{R} = \frac{v_L}{R-d} = \frac{v_R}{R+d} \quad (2-1)$$

$$R = \frac{v_R + v_L}{v_R - v_L} d \quad (2-2)$$

由得到的车辆角速度公式和转弯半径公式可以推导出车辆角速度和车辆线速度与车辆左侧后轮线速度、车辆右侧后轮线速度和车辆后两轮之间距离的关系：

$$\omega = \frac{v_R}{R+d} = \frac{v_R - v_L}{2d} \quad (2-3)$$

$$v = \omega R = \frac{v_R + v_L}{2} \quad (2-4)$$

根据三角函数公式可以推导出前轮转角值与车辆转弯半径和车辆前后轴之间距离的关系：

$$\tan \theta = \frac{l}{R} \quad (2-5)$$

$$R = \frac{v}{\omega} \quad (2-6)$$

$$\theta = \arctan l \frac{\omega}{v} \quad (2-7)$$

由此得到阿克曼结构的运动学模型如下：

$$\begin{cases} \omega = \frac{v_R - v_L}{2d} \\ v = \frac{v_R + v_L}{2} \\ \tan \theta = l \frac{\omega}{v} \end{cases} \quad (2-8)$$

$$\begin{cases} v_R = v + \omega d \\ v_L = v - \omega d \\ \theta = \arctan l \frac{\omega}{v} \end{cases} \quad (2-9)$$

式中， l ——车辆前后轴之间距离（cm）；

$2d$ ——车辆后两轮之间距离（cm）；

θ ——车辆前两轮打角值($^{\circ}$);

v_L ——车辆左侧后轮线速度(m/s);

v_R ——车辆右侧后轮线速度(m/s);

v ——车辆线速度(m/s);

ω ——车辆角速度(m/s)。

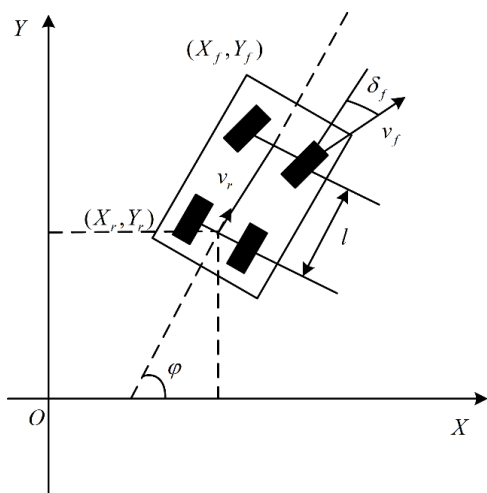


图 2-2 智能车运动模型

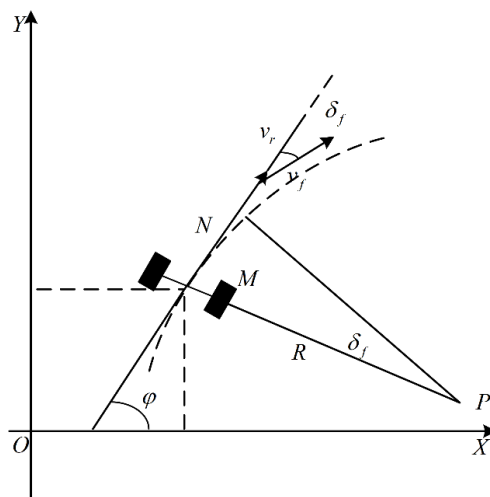


图 2-3 智能车转向模型

如图 2-2 和图 2-3 所示,在惯性坐标系 xOy 下,以变量下标 f 表示 *front* 前,以下标 r 表示 *rear* 后:

(X_f, Y_f) ——前轴轴心坐标;

(X_r, Y_r) ——后轴轴心坐标;

v_f ——前轴中心速度 (m/s);

v_r ——后轴中心速度(m/s);

l ——车辆前后两轴轴距 (cm)

φ ——车辆姿态航向角($^{\circ}$);

δ_f ——车辆前轮偏向角($^{\circ}$);

N ——车辆前轴轴心;

M ——车辆后轴轴心;

R ——车辆后轮转弯半径 (cm);

P ——车辆瞬时转动中心。

假设转向过程中车辆质心侧偏角保持不变，即车辆瞬时转向半径与道路曲率半径相同。

对车辆前后轴轴心处速度进行合成，如图 2-4 所示：

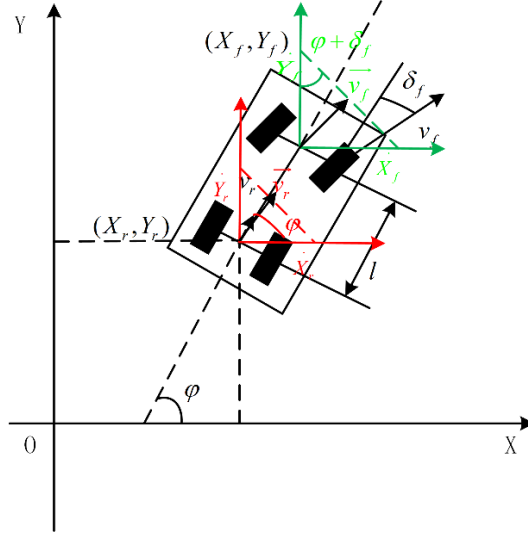


图 2-4 前后轮速度合成图

其中图 2-4 中：

$\dot{X}_f$ ——局部坐标系车辆前轴中心的横坐标；

$\dot{Y}_f$ ——局部坐标系车辆前轴中心的纵坐标；

$\dot{X}_r$ ——局部坐标系车辆后轴中心的横坐标；

$\dot{Y}_r$ ——局部坐标系车辆前轴中心的纵坐标；

$\dot{v}_f$ ——局部坐标系车辆前轴中心的速度；

$\dot{v}_r$ ——局部坐标系车辆后轴中心的速度；

$\varphi + \delta_f$ ——局部坐标系前轮转向角的车辆航向角。

车辆后轮轴心处的速度为合成后沿平行于车辆速度方向分量：

$$\dot{X}_r \sin \varphi = \dot{Y}_r \cos \varphi \quad (2-10)$$

而其运动学约束为合成后沿垂直于车辆速度方向分量：

$$\dot{X}_r \sin \varphi = \dot{Y}_r \cos \varphi \quad (2-11)$$

带入上式，可得：

$$\begin{cases} \dot{X}_r = v_r \cos \varphi \\ \dot{Y}_r = v_r \sin \varphi \end{cases} \quad (2-12)$$

车辆前两轮的运动学约束同样为合成后沿垂直于车辆速度方向分量：

$$\dot{X}_f \sin(\varphi + \delta_f) = \dot{Y}_r \cos(\varphi + \delta_f) \quad (2-13)$$

根据前后轮几何关系，可以得到：

$$\begin{cases} X_f = X_r + l \cos \varphi \\ Y_f = Y_r + l \sin \varphi \end{cases} \quad (2-14)$$

对等式两边进行求导：

$$\begin{cases} \dot{X}_f = \dot{X}_r - l \sin \varphi \cdot \omega = v_r \cos \varphi - l \sin \varphi \cdot \omega \\ \dot{Y}_f = \dot{Y}_r + l \cos \varphi \cdot \omega = v_r \sin \varphi + l \cos \varphi \cdot \omega \end{cases} \quad (2-15)$$

将其带入运动学约束等式中展开并化简：

$$v_r \sin \delta_f = l \cos \delta_f \cdot \omega \quad (2-16)$$

同时得到转弯半径和前轮偏向角：

$$\begin{cases} R = \frac{v_r}{\omega} \\ \delta_f = \arctan \frac{1}{R} \end{cases} \quad (2-17)$$

根据上面所有公式，进行化简整合，最终可得到车辆位姿与车辆速度和转向角度的运动学模型数学公式：

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ \frac{\tan \delta_f}{l} \end{bmatrix} v_r \quad (2-18)$$

2.3 环境地图建模

在智能车的运行体系里，环境地图建模是其实现自主导航和高效路径规划的核心要素。它就像智能车的“虚拟眼睛”，让智能车能够“看清”所处环境的布

局与特征。精准的环境地图不仅能帮助智能车提前规划合理路径，还能在行驶过程中及时避开障碍物，确保运行的安全性与高效性。不同的建模方法有着各自独特的优势和适用场景，对智能车的性能影响显著。在移动机器人研究领域，环境地图建模方法众多，本节主要对拓扑地图法^[56-58]、几何特征地图法^[59]和栅格地图法^[59]的优劣性做了主要分析介绍。

2.3.1 拓扑地图法

拓扑地图法是机器人学中地图的表示方法之一，将环境表示为带结点和相关连接线的拓扑结构图^[60]。拓扑地图法基于拓扑学，将环境抽象为由节点和边构成的网络。如图 2-5 所示中 V 代表节点，节点代表关键位置，像建筑出入口、道路交叉点，边则对应节点间连接，比如走廊、街道。它不侧重物理坐标与尺寸，而是凸显位置关联和可达性。构建方式有基于传感器数据，利用激光雷达、相机信息转化；也有基于视觉感知，借分析图像场景特征确定节点及连接。其特点为简洁、灵活且具有较好的鲁棒性。该方法在机器人导航、自动驾驶、虚拟现实和增强现实等领域应用广泛，能助力机器人规划路径、为车辆提供全局路网信息、营造真实空间体验。拓扑地图法优点突出，结构简单，存储便捷，抽象层次高，助力机器人理解环境结构以规划路径。更新维护容易，可扩展性强，在动态环境优势尽显，鲁棒性好，能容忍环境变化与不精确传感器数据，利于多机器人协作。但它也有短板，环境存在相似区域时，拓扑节点识别困难，信息量有限，常需搭配其他传感器使用。

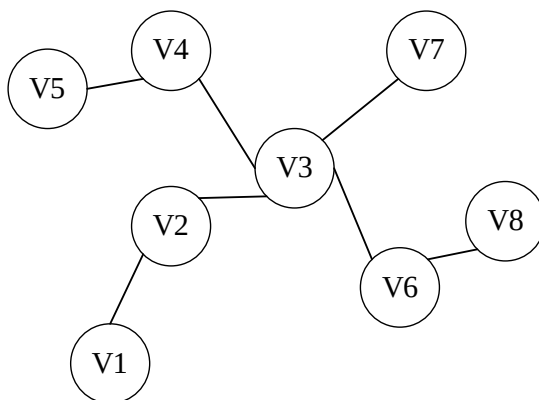


图 2-5 拓扑地图

2.3.2 几何特征地图法

几何特征地图法是基于特征信息的地图表示方法，主要依靠移动机器人从探测的环境信息中提取具有几何特征的数据来构建地图^[61]，如图 2-6 所示。

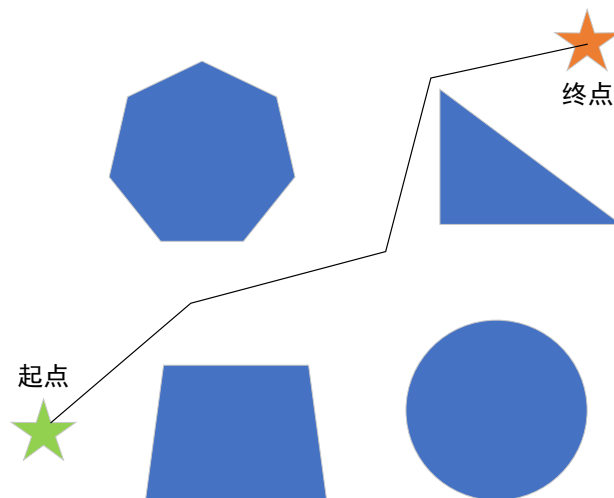


图 2-6 几何特征地图

几何特征地图法是机器人借传感器收集环境感知信息，提取点、线、多边形等几何特征来描绘环境的方法。其优势突出，信息紧凑，能大幅节省存储与计算资源，助力机器人精准进行位置估计与目标识别。在局部区域运用卡尔曼滤波，可高效获取高精度结果，计算量也不大。例如室内作业机器人，靠提取墙壁直线、角落点等特征，就能完成环境构建、实现自身定位。

不过该方法存在局限。在复杂自然环境，如茂密森林，或非结构化环境，像震后废墟中，环境杂乱无章，提取几何特征难度剧增，常需对感知信息额外去噪、增强处理，且需大量数据支撑，过程繁琐耗时。在广域环境，如沙漠、草原，缺少标志性特征，机器人移动易累积定位误差，很难维持精确坐标信息，限制了其在大型场景的应用。所以，使用几何特征地图法通常需要对提取到的特征进行预处理，以减少传感器噪声的影响，并收集足够的感知数据来获得最终结果^[62]。

2.3.3 栅格地图法

在移动机器人环境建模领域，栅格地图法是应用最为广泛的建模方法^[63-65]，其示意如图 2-7 所示。

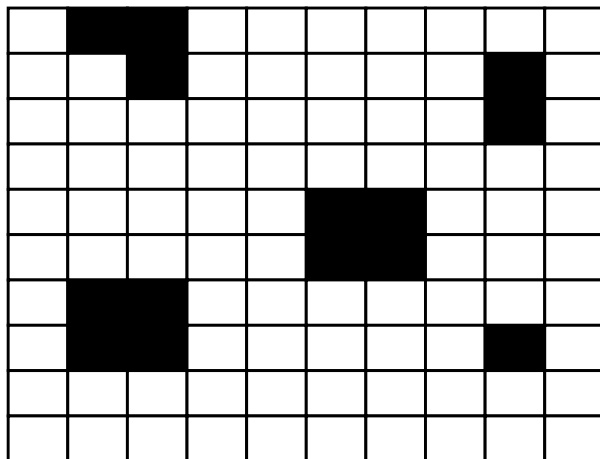


图 2-7 栅格地图

栅格地图法核心在于对环境信息实施数字化处理。它把整个环境地图拆解成一系列离散的栅格，针对每个栅格单元赋予特定值，以此定量呈现其状态。在实际运用中，常采用简洁明了的赋值规则，像以“1”代表该栅格被占用，即表明存在障碍物，移动机器人无法从此处通行；以“0”代表空闲状态，意味着移动机器人能够在该区域自由活动。而且，每个栅格单元都被赋予唯一坐标，如同在地图上有了精准定位。基于这样的栅格地图构建模式，移动机器人原本复杂的路径规划难题，就巧妙地转化为从起始栅格单元朝着目标栅格单元的规划问题。这一转化极大地简化了路径规划的思路，使得机器人能够依据栅格状态和坐标信息，通过特定算法筛选出从起点到终点的最优或可行路径，进而实现高效的自主移动。

在二维环境中，将栅格地图合理划分为具有 0 和 1 二值信息的区域，以实现最佳的存储、分辨率和规划效率。在这个二值信息中，0 代表自由栅格，1 表示障碍物栅格。当相邻栅格没有障碍物时，机器人可移动的方向有 8 个。在 $M \times M$ 的栅格环境中，栅格序号 i 和坐标 (x_i, y_i) 的关系可以用公式表示：

$$\begin{cases} x_i = a[\text{mod}(i, M) - a/2] \\ y_i = a[M + a/2 - \text{ceil}(i/M)] \end{cases} \quad (2-19)$$

其中： a 为栅格的边长，这里 $a=1$ ， mod 为取余运算， ceil 表示向上取整运算。通常的路径规划算法会搜索自由栅格之间的最优路径，以确保机器人可以安全地穿越栅格地图，达到目标位置。

2.4 智能车仿真软件

2.4.1 ROS 操作系统介绍

ROS^[66-68]是专为智能车开发设计的软件框架，在智能车的软件开发中具有不可替代的重要作用。它通过集成丰富的工具和代码库，极大地简化了智能车编程的复杂度，开发者无需从头编写大量基础代码，可直接调用 ROS 提供的功能模块，从而显著缩短开发周期。例如，在构建机器人的导航功能时，开发者可以利用 ROS 中已有的导航相关功能包，快速实现机器人的路径规划和定位功能。

ROS 支持并行执行和进程间通信，这使得多个任务可以在同一时间内高效运行，不同的功能模块之间能够顺畅地进行数据交换和协同工作。其模块化和代码重用理念促进了全球范围内的代码库共享，开发者可以方便地获取和使用其他开发者贡献的代码资源，加速项目开发进程。同时，这种理念也优化了组件之间的交互方式，使得系统具有更强的灵活性和扩展性。当需要为机器人添加新功能时，只需在 ROS 框架中添加相应的模块，并配置好与其他模块的接口，即可轻松实现功能扩展。

2.4.2 ROS 系统架构

ROS 的架构主要分为文件系统、组织管理程序文件、计算图层三个部分^[69]，如图 2-8 所示。

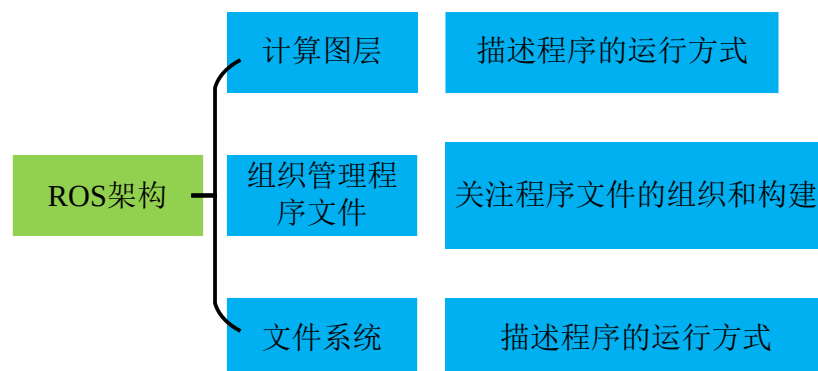


图 2-8 ROS 架构图

（1）文件系统

如图 2-9 所示，文件系统主要包括六个部分：工作空间（workspace）、源文件空间（src）、功能包（package）、编译空间（build）、开发空间（devel）。这些部分共同构成了一个系统，用于支持智能车的功能实现。具体来说，在 ROS 架构中，功能包集提供了一系列接口，供智能车调用，而每个功能包可能包含数个可执行的程序单元。功能包集清单详细记录了包含在该集中的各个功能包的信息。相互之间的依赖关系和使用的编程语言则由功能包清单所指定^[70]。服务文件制定了特定的格式，用于定义服务请求和相应数据结构的规范。

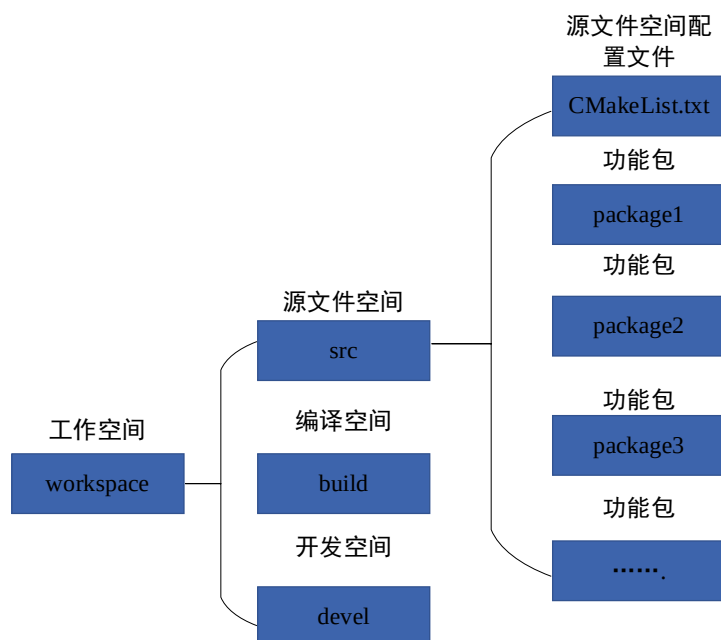


图 2-9 ROS 文件系统层结构图

（2）ROS 计算图层

如图 2-10 所示，主要涉及节点间的通信机制，通过网络实现信息交换。在 ROS 架构中，计算图层通过定义节点、消息、话题及服务等机制，促进了模块化与分布式处理。节点，作为执行特定任务的进程，通过发布和接收消息机制进行通信，其中“话题”充当消息的传输管道，实现了进程间的信息流动。这种机制不仅增强了运行时进程关系的可视化，还支持了多节点协同，通过明确的发布者与接收者角色，实现了结构化且高效的通信。因此，ROS 为智能车系统中组件间的数据交换和指令传输提供了一套灵活且有效的方法，支持复杂功能集成与任务并行处理，为感知、决策和行动提供了强大的软件支撑。以此循环，发布者和接收

者就实现了节点之间的通讯^[70]，如图 2-11 所示。

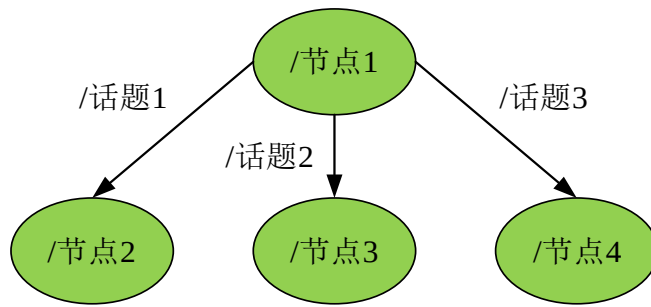


图 2-10 ROS 中的节点关系图

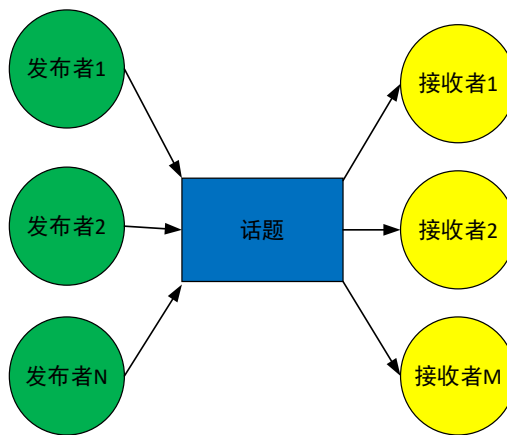


图 2-11 ROS 发布订阅话题示意图

（3）ROS 开源社区级

ROS 开源社区通过资源共享，加速了机器人技术的发展。ROS 开源社区层资源结构如图 2-12 所示。

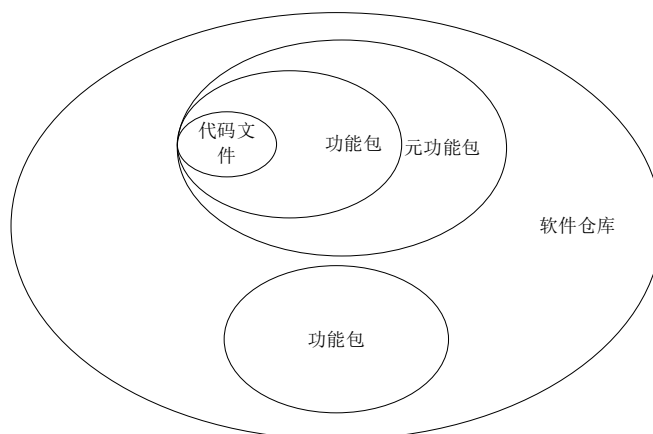


图 2-12 ROS 开源社区层资源结构

ROS 构建了一个全面的生态系统，囊括了发行版、软件库、ROS 维基、邮件列表和 Bug 提交系统等关键元素，共同为用户提供一个完整的支持和交流平台^[72]。发行版简化了安装流程并满足多样化需求，软件库促进了开源代码的存储与分享，ROS 维基作为信息交流平台便利用户贡献资源。邮件列表和 Bug 提交系统分别为交流反馈和问题报告提供渠道，增强社区互动与系统稳定性。

2.4.3 ROS 话题通信机制

在 ROS 的分布式通信架构中，信息流的管理采取了一套细致的注册与连接机制，保证了消息发布（Talker）与消息订阅（Listener）之间的高效协同。此过程体现了 ROS Master 在节点间建立连接过程中的调度作用，尽管其不直接参与最终的数据传输，但通过有效的注册信息管理和地址匹配服务，其确保了系统中的消息能够准确、高效地在各节点间流动^[72]。这一机制不仅提高了系统的可扩展性和灵活性，而且也促进了在复杂网络环境下的稳定通信，是 ROS 分布式架构中不可或缺的一环。ROS 系统具体进行话题通信的示意图如图 2-13 所示。

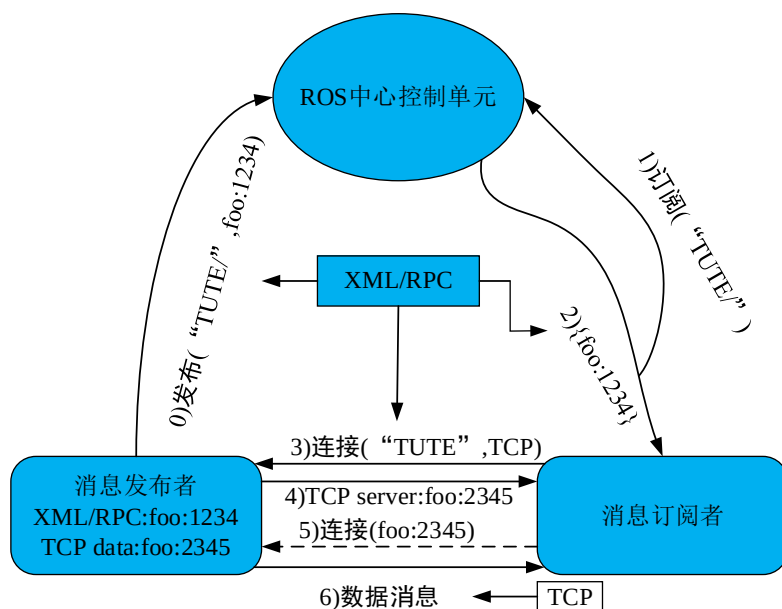


图 2-13 ROS 话题通信机制流程图

具体流程如下：

1.初始化消息发布节点：当 Talker 节点启动时，它通过特定的 RPC 端口（例如 1234 端口）向 ROS 中心控制单元（即 ROS Master）提交自己的注册信息，这

些信息包括但不限于其将要发布的消息的话题名称。ROS Master 随后将这些信息纳入其维护的注册数据库中。

2.初始化消息订阅节点：与此同时，Listener 节点为了接收特定话题的消息，也会通过 RPC 机制向 ROS Master 提交其订阅意图，明确指出所需的话题名称。

3.信息匹配与地址传递：ROS Master 在收到订阅请求后，将基于已注册的发布节点信息进行匹配搜索。若初步匹配成功，ROS Master 则利用 RPC 机制将发布节点的地址信息传递给订阅节点；若匹配不成功，则等待新的发布节点注册。

4.建立节点间直接连接：获得发布节点地址后，订阅节点向该地址发起直接连接请求，此请求包含话题名称、消息类型及期望的通信协议（TCP 或 UDP）。发布节点在接收到请求后，通过 RPC 确认连接，并提供建立数据传输所需的 TCP 连接信息。

5.数据传输：在双方节点确认连接信息后，它们之间便建立起直接的 TCP 网络连接，发布节点随即开始向订阅节点传送具体的话题消息数据。

2.5 本章小结

本章聚焦智能车平台展开多方面研究。在运动学分析上，借助阿克曼转向几何理论构建模型，明晰车辆运动参数关系，为路径规划和控制提供理论支撑。环境地图建模方面，剖析拓扑、几何特征和栅格地图法的优劣，为实际应用选择合适建模方式提供依据。智能车仿真软件部分，着重介绍 ROS 操作系统，其丰富工具、模块化设计和高效通信机制，简化编程复杂度，提升开发效率。总之，本章研究成果为智能车在复杂环境下的运行、算法应用及进一步发展筑牢了理论和实践根基。

第3章 基于改进蚁群算法的全局路径规划研究

3.1 引言

在智能车应用日益广泛的背景下，其路径规划算法性能备受关注，尤其是全局路径规划，对机器人高效执行任务意义重大，像工业生产、物流配送等场景都依赖它寻找最优路径，以提升效率和降低能耗。传统蚁群算法虽有自适应性和鲁棒性，可用于复杂环境路径规划，但收敛慢、易陷入局部最优的问题突出，在大规模复杂环境中，会导致机器人路径规划效果差，影响任务执行效率。为此，本章致力于研究改进蚁群算法在全局路径规划中的应用，通过创新优化算法关键要素，提升其搜索与寻优能力，使其能更好适应复杂环境，为机器人提供优质全局路径规划方案，也为后续算法融合及系统性能提升筑牢根基。

3.2 传统蚁群算法

在智能车的全局路径规划领域，蚁群算法凭借其独特的原理和特性占据着重要地位。它模拟蚂蚁群体在觅食过程中通过信息素交流来寻找最优路径的行为，为解决路径规划问题提供了一种有效的思路。该算法具有分布式计算、正反馈机制和贪婪式搜索等特点，能够在复杂的环境中逐步搜索出较优路径，在诸多领域得到了广泛应用。然而，随着应用场景日益复杂，传统蚁群算法在实际应用中暴露出一些问题，这些问题严重影响了算法的性能和智能车的路径规划效果，亟待改进。接下来将详细介绍传统蚁群算法的具体内容，以便更好地理解后续改进措施的针对性和必要性。

(1)概率选择：在时刻 t ，蚂蚁 k 位于节点 i ，以概率 $q_1(q_1 \in [0,1])$ 转移到信息素浓度和启发信息乘积最大的节点。蚂蚁的状态转移公式如下：

$$j = \begin{cases} \arg_{i \in a_k} \max[\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)], & q \leq q_1 \\ p_{ij}^k(t), & q > q_1 \end{cases} \quad (3-1)$$

(2)轮盘赌选择: 若概率大于 q_l , 则按照轮盘赌方式选择下一个节点。蚂蚁计算每个邻居节点的概率, 然后蚂蚁根据这些概率选择下一个节点。

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in a_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, & j \in a_k \\ 0, & otherwise \end{cases} \quad (3-2)$$

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (3-3)$$

式中: a_k 为邻域候选可移动栅格集合, α 为信息素启发因子, β 为距离启发因子, d_{ij} 为待选节点 i 到下一节点 j 的欧式距离。

(3)信息素更新: 任意 t 时刻, 蚂蚁 k 根据选择规则从节点 i 移动到节点 j 后, 将实时更新从节点 i 到节点 j 路径上的信息素浓度。全局信息素更新的公式如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (3-4)$$

其中, ρ 为全局信息素挥发率, 表示全局信息素在时间上的衰减程度, 其中 $0 < \rho < 1$ 。信息素浓度增量的计算公式如下:

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{1}{L_k}, & tour(i, j) \in p_k \\ 0, & else \end{cases} \quad (3-5)$$

其中, L_k 为蚂蚁 k 搜索到的全局最优路径长度; p_k 是在该迭代中到达目的地所有蚂蚁路径点的集合。

3.3 改进的蚁群算法

传统蚁群算法在智能车全局路径规划中虽有一定应用, 但存在收敛慢、易陷入局部最优等问题, 严重制约其在复杂环境下的路径规划效果。为提升算法性能, 满足智能车在复杂环境中高效路径规划的需求, 对蚁群算法进行改进十分必要。通过改进算法中的关键要素, 如启发函数、信息素更新机制等, 有望克服传统算法的缺陷, 增强其搜索与寻优能力, 使算法能更好地适应复杂多变的环境, 为智能车提供更优质的全局路径规划方案。

3.3.1 启发函数的改进

在传统蚁群算法中,启发式函数通常使用欧几里得距离公式来评估节点的吸引力,导致蚂蚁的搜索全局性大大下降。改进算法中,通过计算点与目标点之间的距离,并结合势场常数概念来评估每个点的吸引力。

计算势场力:假设节点 i 的坐标为 p_i , 目标节点 E 的坐标为 p_E 。

$$\sigma(k) = \frac{\sigma_0}{1 + ak} \quad (3-6)$$

$$f_a = \sigma(k) \cdot d_{iE}^2 \quad (3-7)$$

式中, a 是势场常数调节因子, d_{iE} 为节点 i 与目标节点 E 的欧几里得距离, σ_0 表示势场常数初始值, $\sigma(k)$ 通常表示势场常数。用来控制蚂蚁在选择下一个节点时对信息素浓度和启发信息的相对重视程度。它的取值会影响蚂蚁的搜索行为和算法的收敛性, $\sigma(k)$ 的值越大, 蚂蚁在选择下一个节点时越倾向于选择信息素浓度较高的路径; 而 $\sigma(k)$ 的值越小, 蚂蚁在选择下一个节点时越倾向于选择启发信息更强的路径。

$$\eta_{ij}(t) = \begin{cases} \frac{\mu^{fa^*(1+\cos\theta_{iE})}}{d_{ij} + d_{jE}} * (1 - R(ij)) * (1 - H(ij)), & \text{if } i \neq E \\ 100, & \text{if } i = E \end{cases} \quad (3-8)$$

其中: μ 为势场启发系数, θ_{iE} 为蚂蚁选择下一个节点的方向和势场力方向的夹角。通常情况下, 目标节点具有较高的启发值, 表示蚂蚁更倾向于选择目标节点作为下一步。 $R(ij)$ 表示从节点 i 到 j 的风险评估值, $H(ij)$ 表示从节点 i 到 j 的历史访问频率。

通过引入风险评估系数, 让蚂蚁在路径选择时能有效避开高风险区域, 全面提升了路径的安全性。其次, 历史频率的加入充分利用过去经验数据, 使蚂蚁倾向于选择性能更好、适应性更强的路线, 极大增强了路径的可靠性。

如图 3-1 所示, iE 为势场力的方向, 当 ij 与 jE 的角度 γ 越大时, θ 越小, 则启发函数值越大, 此时蚂蚁更倾向于选择该节点; 当角度 γ 达到 γ_3 时, i, j_3, E 三点共线, 表示当前节点与下一节点没有拐角, 此时是最适合蚂蚁转移的下一个节点。

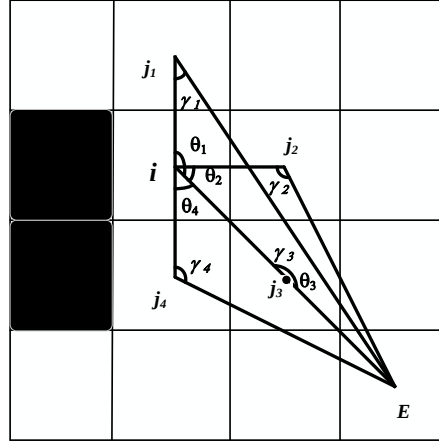


图 3-1 蚂蚁运动方向示意图

在初始搜索阶段中，蚂蚁主要依靠势场的吸引力，加快了蚂蚁的前期收敛速度，同时，整个搜索过程中，势场的吸引力持续对蚂蚁产生影响并动态减小，从而增强了算法的全局性能；在势场力为 0 时，蚁群启发信息主要靠 $1/(d_{ij}+d_{jE})$ 作用，避免蚂蚁搜索过程中陷入“死锁”情况。

3.3.2 信息素初始值差异化

在蚁群算法中，信息素的初始值对算法的性能和收敛速度有着重要的影响。通过差异化初始化信息素，运行 Dijkstra 算法生成一条相对较优路径，并将该路径上的信息素设为初始值，使用 Dijkstra 算法计算从起点到每个顶点的最短路径长度，得到最短路径长度数组 $D=[d_1, d_2, \dots, d_n]$ ，其中 d_i 表示起点到顶点 i 的最短路径长度。

对于所选择的路径，计算每条边 e_{ij} 的信息素浓度。使用一个基础值 τ_0 表示初始信息素浓度，然后根据障碍物个数和路径长度因素进行调整。

$$\tau_{ij}^0 = \tau_0 \times e^{-\alpha \cdot O_{ij}} \times \left(\frac{1}{L_{ij}} \right)^\beta \quad (3-9)$$

其中： τ_{ij}^0 是路径 i 到 j 的初始信息素水平； τ_0 是全局的初始信息素水平； O_{ij} 是路径 i 到 j 上的障碍物个数； L_{ij} 是路径 i 到 j 的长度。 α 和 β 是调节参数，分别用于平衡障碍物个数和路径长度的影响。

使用计算得到的初始信息素值更新蚁群的信息素矩阵：

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij} \quad (3-10)$$

其中, $\Delta\tau_{ij}$ 是从上一步计算得到的初始信息素值。更新的目的是引导蚁群在初始阶段更多地利用已知路径, 提高算法的收敛速度。

3.3.3 信息素差分机制

传统的蚁群算法全局信息素只对最优路径遗留信息素, 使算法收敛性极强, 但是容易陷入局部最优而无法跳出。在改进的蚁群算法中, 信息素的更新方法经过重新设计, 通过信息素差分机制, 针对不同质量的路径进行信息素的更新, 从而在全局搜索和局部优化方面都取得更好的效果, 以实现高质量节点的更快收敛。

在改进的部分中, 根据最优和最差路径上蚂蚁的数量来调整信息素更新量。改进后的信息素更新公式为:

$$\tau_{ij}^{(k+1)} = \begin{cases} (1-\rho)*\tau_{ij}^{(k)} + f*(\frac{Q}{L_{km}} + \frac{b*Q}{L_{min}}), bestpath \\ (1-\rho)*\tau_{ij}^{(k)} + f*(\frac{Q}{L_{km}} - \frac{Q}{w*L_{max}}), worstpath \\ (1-\rho)*\tau_{ij}^{(k)} + f*\frac{Q}{L_{km}}, else \end{cases} \quad (3-11)$$

$$f = \begin{cases} 1.5 & \text{if } PL_{km} < 0.8 * \min kl \\ 1.2 & \text{if } 0.8 * \min kl \leq PL_{km} < \min kl \\ 1 & \text{otherwise} \end{cases} \quad (3-12)$$

$$\tau_{ij}(k) = \begin{cases} \tau_{max} & \text{if } \tau_{ij}(k) > \tau_{max} \\ \tau_{ij}(k) & \text{else} \\ \tau_{min} & \text{if } \tau_{ij}(k) < \tau_{min} \end{cases} \quad (3-13)$$

f 是一个因素参数, 可以调节; Q 是一个常数, 代表信息素沉积的强度; M 是蚂蚁的数量; L_{km} 表示第 k 次循环中蚂蚁 m 所走路径的长度; b 是在当前循环中找到最短路径的蚂蚁数量; L_{min} 表示当前循环中最短路径的长度; w 表示在当前循环找到最长路径的蚂蚁数量; L_{max} 表示当前循环中最长路径的长度;

其中： τ_{min} 为最小信息素浓度， τ_{max} 为最大信息素浓度。在更新信息素时，通过限制信息素的取值范围来保持信息素的多样性，防止信息素过早地收敛到局部最优解，从而能够更好地避免陷入局部最优解。

3.3.4 自适应调整机制

相较于传统的蚁群算法中的 ρ 是一个固定的常数，改进的信息素挥发系数通过引入自适应调整机制，将信息素挥发系数从静态的值改变为根据迭代次数而动态调整的值。改进后的信息素挥发系数公式为：

$$\rho(k) = A * (B + \frac{1}{1 + e^{-\frac{k}{K}}}) \quad (3-14)$$

$$\rho = 0.3 + 0.4 * (\frac{\rho - \rho_{min}}{\rho_{max} - \rho_{min}}) \quad (3-15)$$

其中： k 为当前迭代次数， K 为最大迭代次数，将迭代次数 K 分为 x 个阶段，通过调整 A 和 B 的大小可以控制不同阶段信息素挥发系数的范围， ρ_{min} 和 ρ_{max} 是挥发系数的最小值和最大值，通过公式对信息素挥发系数进行归一化处理，以确保其在一定范围内平稳变化。如图 3-2 所示是改进后 $\rho(k)$ 的变化曲线图。

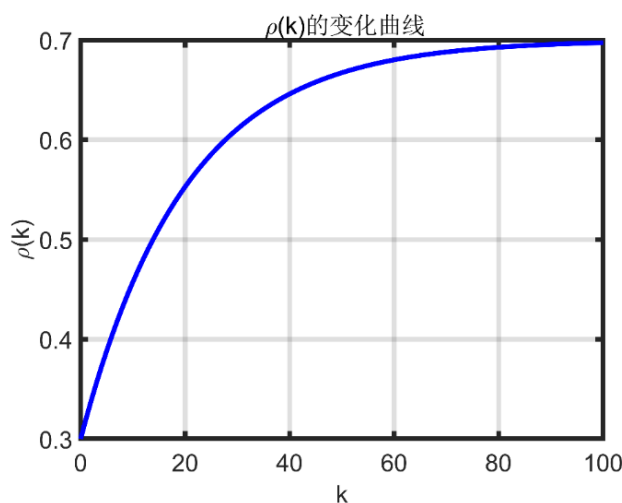


图 3-2 $\rho(k)$ 的变化曲线图

通过自适应调整信息素挥发系数，使其在迭代初期较小并逐渐增加，促使蚂蚁更多进行探索；在迭代后期，随着迭代次数的增加，路径上信息素的含量逐渐

增加,使得信息素浓度对蚂蚁的引导作用增强,减少蚂蚁的搜索时间。既可以保证信息素挥发系数动态变化,也可以保持在一定的范围内,信息素挥发系数 ρ 在算法的不同阶段灵活调整,更好地平衡了全局搜索和局部搜索。

3.3.5 剪枝策略

蚁群算法是根据蚁群的觅食行为推导出的八自由度搜索概率算法,蚁群算法步长选择简单,限制了智能车路径规划的效率。为了解决该问题,通过剪枝策略消除冗余转折点,同时提高算法步长的多样性,具体流程如图 3-3 所示。

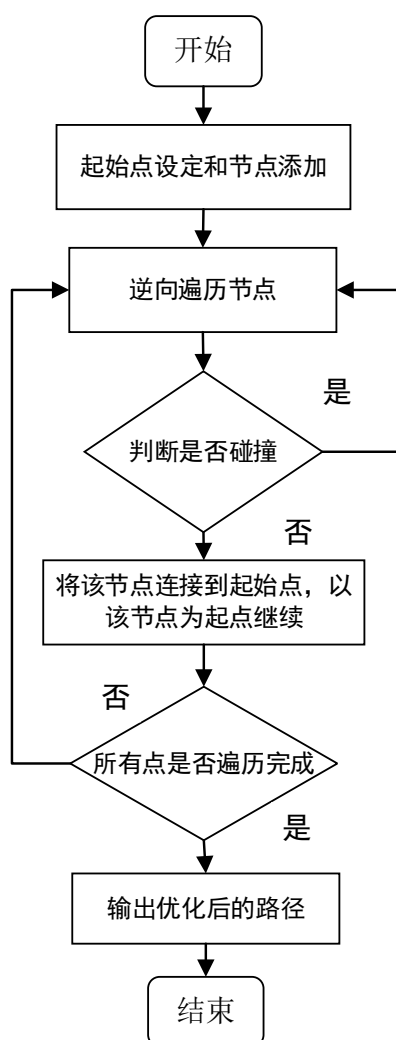


图 3-3 剪枝策略流程图

如图 3-4 所示，检测到 A 与 G 可以直接相连但是两点连接的直线路径与障碍物相交，说明该路径不可行，需要重新对前一个点进行判断，以此类推选择连通的相邻点。

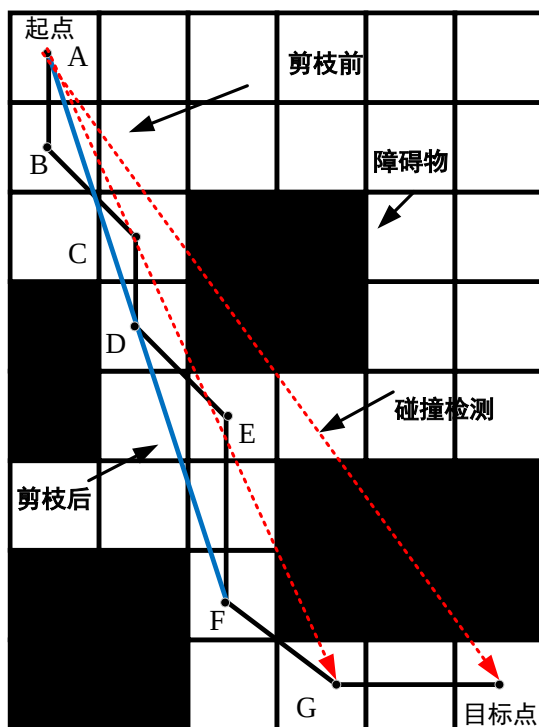


图 3-4 剪枝策略原理图

使用叉积公式判断路径线段 (X_1, Y_1) 到 (X_2, Y_2) 是否与障碍物边界相交时。

$$cross = (X_2 - X_1) \cdot (Y_{max} - Y_1) - (Y_2 - Y_1) \cdot (X_{max} - X_1) \quad (3-16)$$

其中， (X_{max}, Y_{max}) 是障碍物边界的最大坐标点。如果叉积 $cross$ 为正值，表示路径线段与障碍物边界相交，需要避免这条路径。如果叉积 $cross$ 为零或负值，表示路径与障碍物没有相交，路径是可行的。

如图 3-4 所示：首先判断起始点 A 与目标点 G 是否可以相连，然后再对前一个点 F 进行判断，以此类推，直到找到可以连接并且连线没有与障碍物相交的点，进行连接。原始路径为 ABCDEFG，经过优化后为 AEFG，减少了转折点个数，同时路径长度也得以减少。

3.4 改进蚁群算法仿真实验与分析

仿真实验环境:在 *windows10 64bits* 环境下运行,仿真平台为 *MATLAB 2021b*,计算机处理器 *Intel(R) Core(TM) i5-7400 CPU*,主频为 *3.00 GHz*,内存 *8GB*。

首先,实验设置了不同势场常数取值下的仿真数据,并比较了不同势场常数取值下最优路径长度的大小和迭代次数。如表 3-1 所示,在 $\sigma=0.03$ 时,路径长度达到最短,结合蚂蚁搜索路线图 3-5 所示,可以看出蚂蚁极易陷入局部最优解,所以当 $\sigma=0.03$ 时效果最好。

表 3-1 势场常数 σ 不同取值

势场常数 σ	最优路径长度/m	迭代次数/次
0.001	32.7264	不稳定
0.005	31.9706	86
0.01	30.0416	52
0.03	29.2132	18
0.05	29.2132	8
0.1	29.2132	2

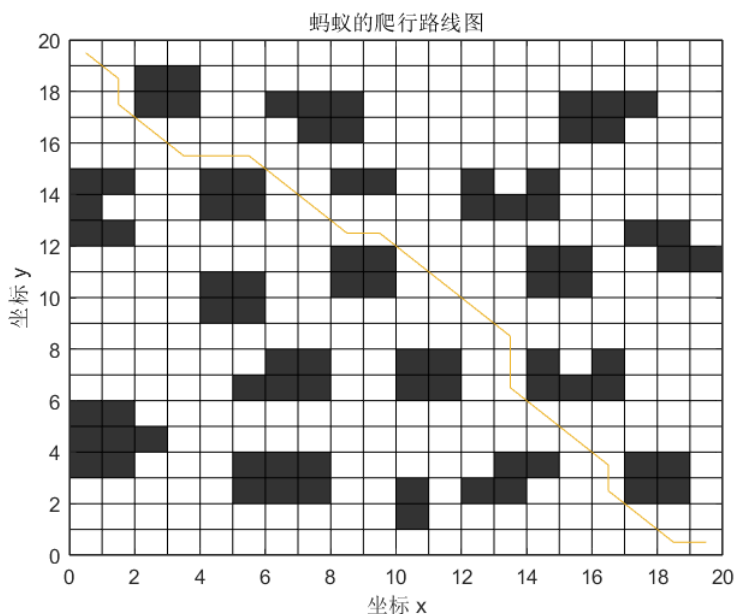


图 3-5 势场常数为 0.1 时蚂蚁的搜索路线图

然后，确定势场启发式系数 μ 的值。如图 3-6 所示：在 $\mu=0.3$ 时，迭代次数最少，此时路径长度也最短，因此取 $\mu=0.3$ 时最优。

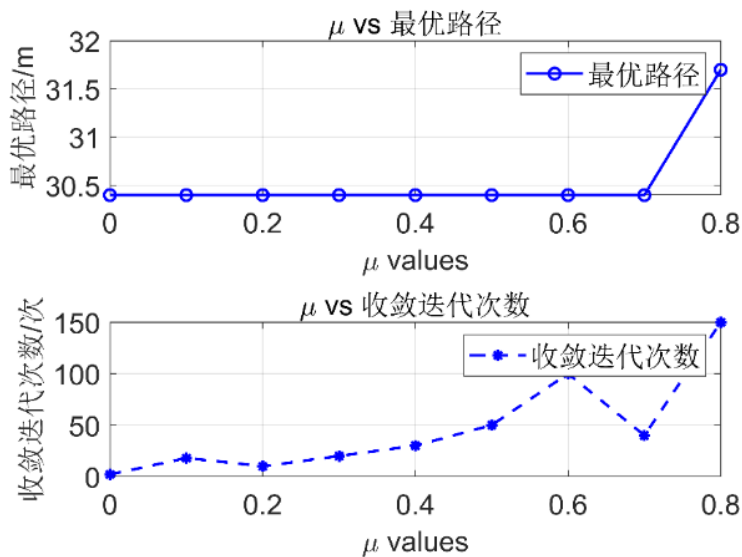


图 3-6 势场启发系数 μ 取值变化图

确定完参数的取值后进行仿真实验对比，实验中其他各相关参数设置如表 3-2 所示。

表 3-2 公共参数

参数	参数描述	参数值
M	蚂蚁数量	50
K	最大迭代次数	100
α	信息启发因子	1
β	距离启发因子	3.5
Q	信息素加强系数	10
ρ	信息素挥发系数	0.65
τ_0	信息素初始值	3

为了验证改进的蚁群算法在栅格地图环境中路径规划的有效性、优越性，分别使用传统蚁群算法、文献[23]算法、文献[24]算法和改进的蚁群算法在 20×20 的栅格环境下进行仿真模拟对比。

如图 3-7 所示是四种对比算法的路径规划图，改进优化后的路径长度减少，转折点个数减少，更加适合智能车的运动。如图 3-8 所示是进一步对改进的蚁群

算法路径进行剪枝，使得路径长度更短，转折点数量更少，对于环境更复杂的，转折点更多的路径，剪枝策略优化的效果更为明显。

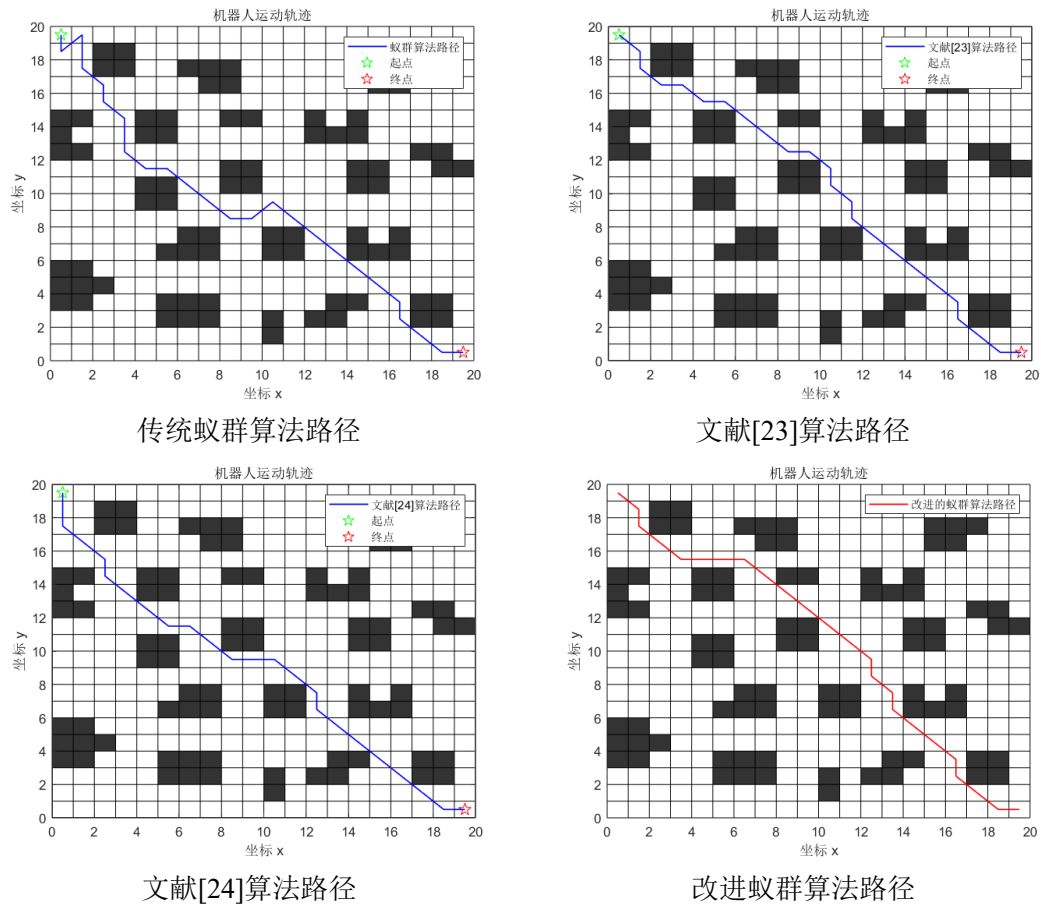
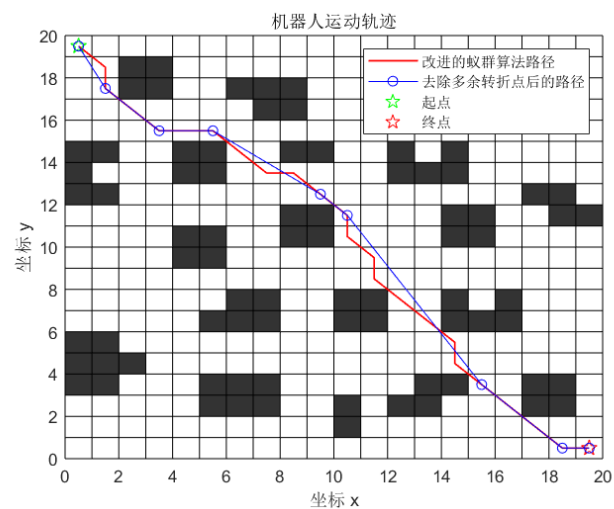


图 3-7 各算法的路径规划图



如图 3-9(a)所示：传统的蚁群算法搜索范围大，消耗了很多时间，而如图 3-9(b)所示：蚂蚁搜索有引导性，能够使蚂蚁更快地接近目标，减少了搜索过程中的无效移动，提高了蚁群算法的搜索效率。

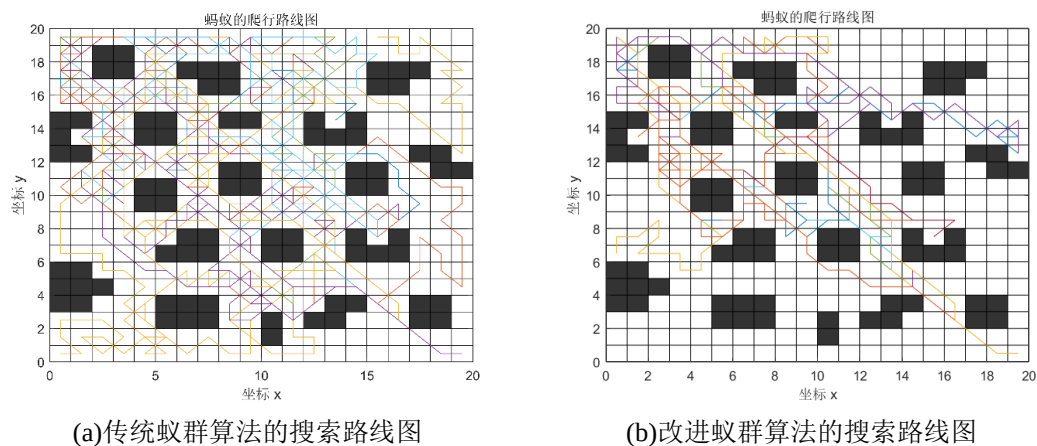


图 3-9 两种算法的搜索路线图

如图 3-10 所示是四种算法的收敛曲线图。改进的算法相比其他三种算法收敛速度显著提高。

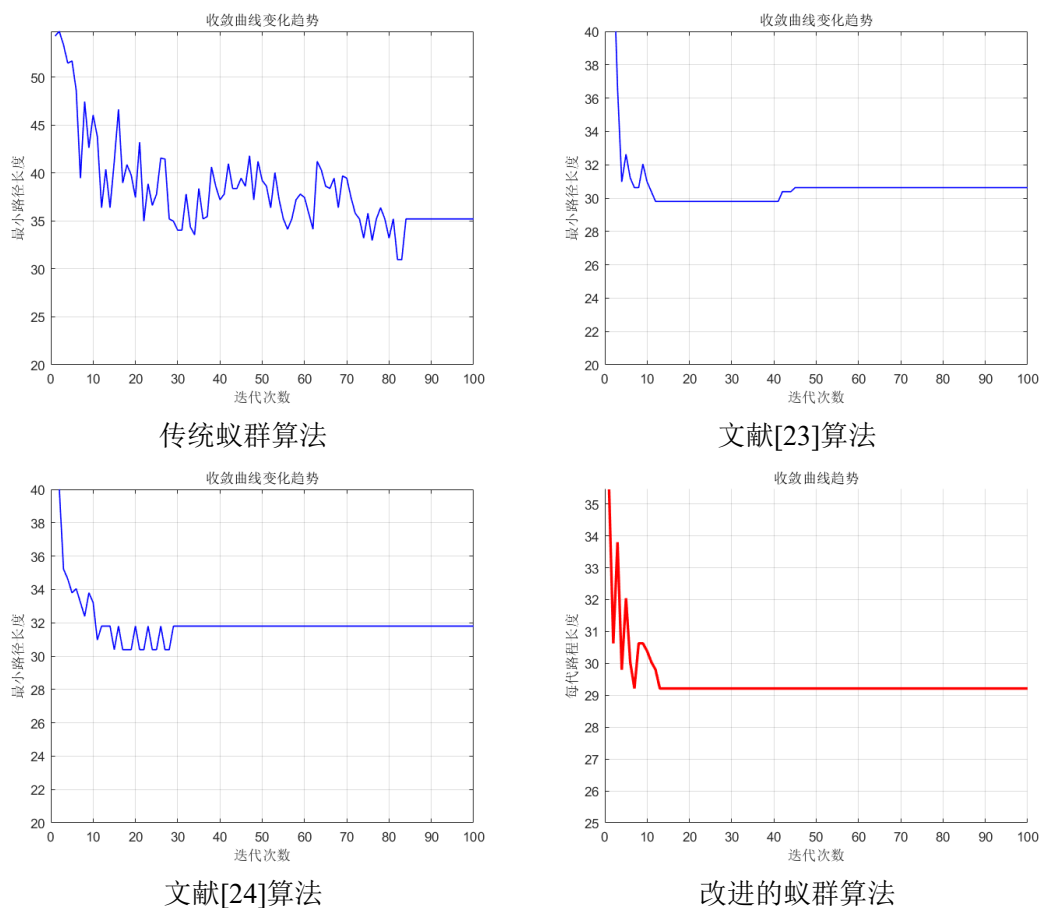


图 3-10 各算法的收敛图

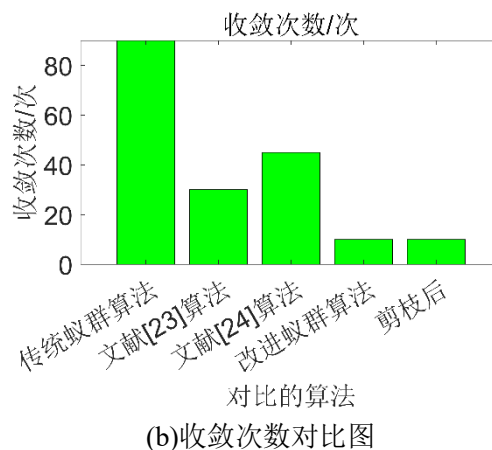
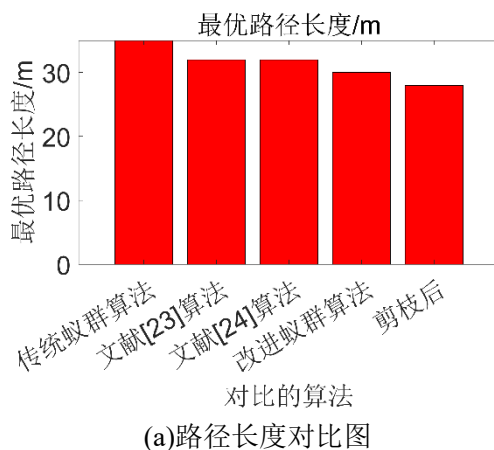
如表 3-3 和图 3-11 所示,当与其他三种算法相比,改进的算法在多个方面呈现了优势。首先,与其他算法相比,改进的蚁群算法在最优路径长度上取得了明显的降低,达到了 29.2132,相较于传统蚁群算法有了约 17.03%的降低,在去除多余转折点后进一步减短路径的长度。

其次,通过势场引导并结合历史访问频率,蚂蚁在搜索过程中更加迅速地聚焦于最优的路径,减少了不必要的搜索盲目性。平均迭代次数从传统蚁群算法的 83 降低到了 12,收敛时间平均仅为 6.25,相较于传统蚁群算法有了约 23.68%的提升。

此外,相对于传统蚁群算法的 15 个转折点,改进的算法只有 11 个转折点,而经过进一步采用剪枝优化的算法只有 7 个转折点,有了约 26.67%至 53.33%的降低。

表 3-3 仿真实验数据对比

算法	最优路径 长度/m	平均迭代 次数/次	平均搜索 时间/s	路径转折 点个数/个
传统蚁群算法	35.2132	83	8.32	15
文献[23]算法	31.7990	29	6.90	15
文献[24]算法	30.9701	42	6.55	9
改进蚁群算法	29.2132	12	6.25	11
剪枝后	27.9415	12	6.21	7



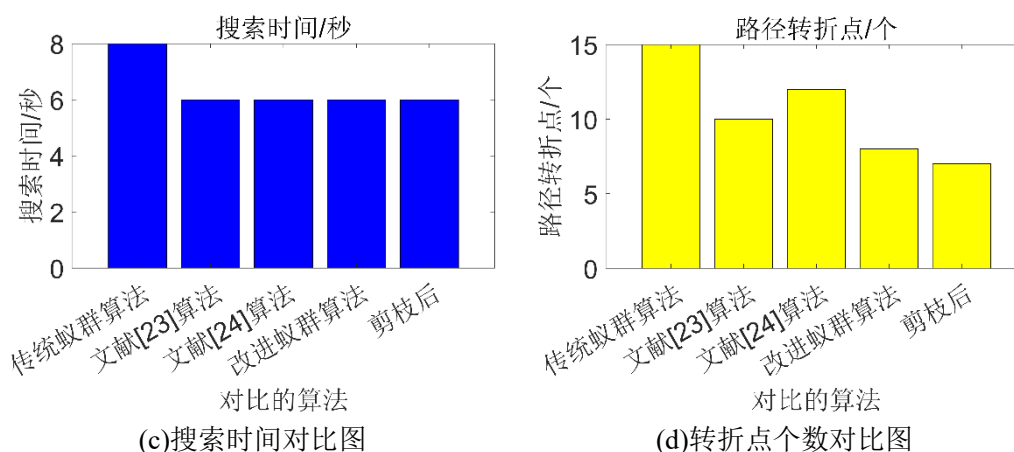


图 3-11 各算法仿真数据对比图

3.5 本章小结

本章针对传统蚁群算法于全局路径规划时存在的收敛速度慢、易陷入局部最优解等问题，提出了改进的蚁群算法。通过设计新的势场启发函数，增强了路径搜索的引导性，提高了算法的收敛速度，降低了整体路径的风险；利用 Dijkstra 算法对初始信息素予以差异化处置，同时引入信息素差分机制，有力规避了算法过早陷入局部最优。自适应调整机制使信息素挥发系数根据迭代次数动态变化，提升算法运算效率；采用剪枝策略减少了路径转折点，提升了路径质量。

仿真实验表明，改进蚁群算法在路径长度、迭代次数、搜索时间和转折点个数等性能指标上相比传统蚁群算法和其他相关算法具有明显优势，验证了改进算法在全局路径规划中的有效性和优越性，为后续与改进 DWA 算法融合实现多机器人高效路径规划奠定了坚实基础。

第4章 基于改进 DWA 算法的局部路径规划研究

4.1 引言

在多机器人路径规划中，局部路径规划对于机器人在动态环境中的实时避障和运动控制至关重要。DWA 算法作为一种常用的局部路径规划方法，具有实时性强等优点，但也存在一些局限性。例如，传统 DWA 算法缺乏全局路径引导，容易陷入局部最优解，且在面对复杂环境和高度移动障碍物时适应性不足。为了提高多机器人在复杂环境中的路径规划性能，实现高效、安全的运动，本章将对传统 DWA 算法进行改进，并将其与改进蚁群算法（IACO）融合，以克服各自的缺点，充分发挥两种算法的优势。

4.2 传统 DWA 算法

4.2.1 机器人运动模型

DWA 算法可以在窗口内对智能车的线速度 $v(t)$ 和角速度 $\omega(t)$ 进行采样，建立了两轮差速智能车的运动学模型，如图 4-1 所示。

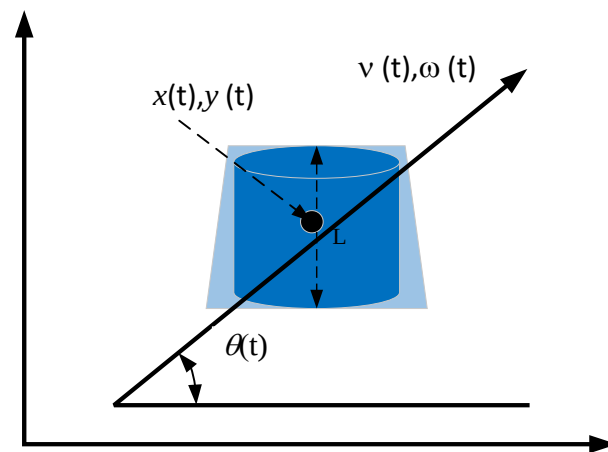


图 4-1 运动学模型图

假设机器人轨迹可以被分解为多个时间片，并且在每个时间片 Δt 内，机器人进行均匀的线性运动，此时机器人的运动学模型可以表示如下：

$$\begin{cases} x_{t+1} = x_t + v\Delta t \cos(\theta_t) \\ y_{t+1} = y_t + v\Delta t \sin(\theta_t) \\ \theta_{t+1} = \theta_t + \omega\Delta t \end{cases} \quad (4-1)$$

其中, $x(t)$ 、 $y(t)$ 和 $\theta(t)$ 分别表示时间 t 时机器人在 x 和 y 方向上的位置和角度; 线速度 $v(t)$ 的变化范围取决于离障碍物最近的距离和最大线减速度; 角速度 $\omega(t)$ 变化的内容由距离障碍物最近的距离和最大角减速度决定。

4.2.2 机器人速度采样

在速度空间中, DWA 算法将机器人的避障问题描述为一个具有约束的优化问题, 包括微分机器人的不完全运动约束、机器人结构的动力学约束和环境约束, 如图 4-2 所示。

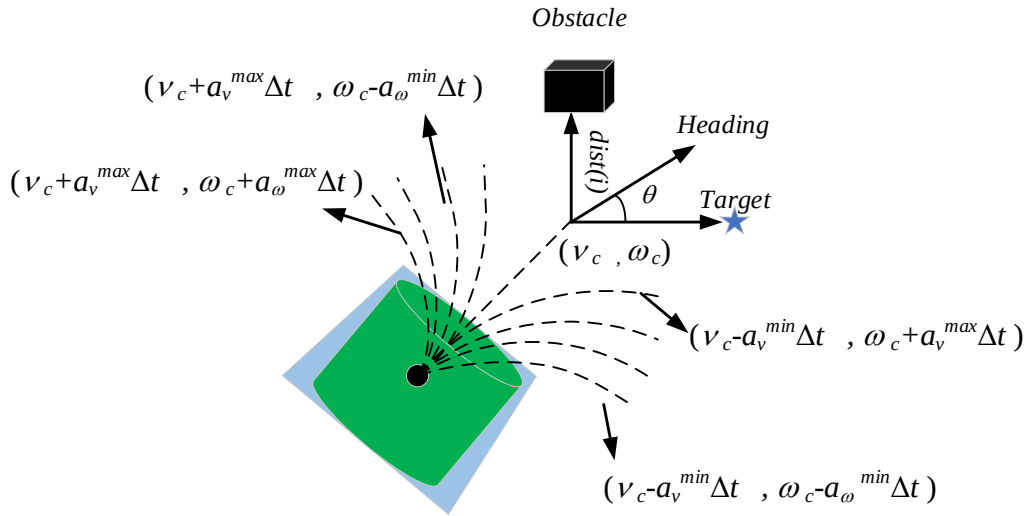


图 4-2 机器人约束图

(1) 机器人的最大最小约束如下:

$$v_s = \{(v, w) | v \in [v_{min}, v_{max}], w \in [w_{min}, w_{max}]\} \quad (4-2)$$

(2) 电机加减速约束如下:

$$v_d = \{(v, \omega) | v \in [v_c - a_v^{min} \Delta t, v_c + a_v^{max} \Delta t], \omega \in [\omega_c - a_\omega^{min} \Delta t, \omega_c + a_\omega^{max} \Delta t]\} \quad (4-3)$$

其中, v_c 和 w_c 分别是机器人在当前时刻的线速度和角速度; a_v^{min} 和 a_w^{min} 分别是最小线性减速度和最小角减速度; a_v^{max} 和 a_w^{max} 分别是最大线性加速度和最大角加速度; Δt 为采样时间。

(3) 制动距离约束如下:

$$v_a = \left\{ (v, \omega) \mid v \leq \sqrt{2 \text{dist}(v, \omega) a_v^{\min}}, \omega \leq \sqrt{2 \text{dist}(v, \omega) a_\omega^{\min}} \right\} \quad (4-4)$$

其中 $\text{dist}(v, w)$ 是模拟轨迹与最近障碍物之间的距离。根据表达式 $0 - v_a^2 = -2 \text{dist}(v, w) a_v^{\min}$ 和 $0 - \omega_a^2 = -2 \text{dist}(v, w) a_\omega^{\min}$, 模拟速度必须满足方程的条件, 才能在更大程度上确保机器人运动的安全性。

4.2.3 评估函数

动态窗口方法通过评估函数评估预测轨迹, 并输出满足约束的最优速度。传统的评估函数表示如下:

$$G(v, w) = \sigma[\delta \cdot \text{heading}(v, \omega) + \eta \cdot \text{vel}(v, \omega) + \gamma \cdot \text{dist}(v, \omega)] \quad (4-5)$$

θ 是机器人运动方向与目标节点之间的夹角, $\text{vel}(v, w)$ 是速度评估函数, 一般指机器人的线速度, $\text{dist}(v, w)$ 是障碍物距离评估函数, 表示预测轨迹末端到障碍物的最短距离, δ 、 η 和 γ 是每个评估函数的权重系数; σ 是表示评估函数被归一化的评估函数。

4.3 改进的 DWA 算法

传统的 DWA 算法存在以下问题:

(1) 缺乏全局路径引导: 传统的 DWA 算法在规划机器人路径时缺乏全局路径引导, 容易陷入局部最优解。

(2) 环境适应性低: 传统的机器人导航方式依赖于全局路径的拐点, 然而, 当未知的静态障碍物较多时, 机器人的跟踪精度会降低。

(3) 对高度移动障碍物的无效性: 传统的 DWA 算法在静态环境中表现出良好的鲁棒性和避障性, 但在运动空间中出现高度移动的障碍物时, 可能变得无效。

针对以上问题, 提出了以下改进策略, 使智能车能够在复杂的静态和动态环境中实现高效、安全的运动。

4.3.1 局部最优问题的解决

针对传统的 DWA 算法规划路径缺乏全局规划的指导这一缺陷，将改进蚁群算法规划的全局路径信息与机器人运动相结合，保证局部动作基本遵循全局路径。如图 4-3 所示：从图中可以看出，融合算法下的机器人具有更好的安全性和平滑性，最终可以顺利到达目标点。

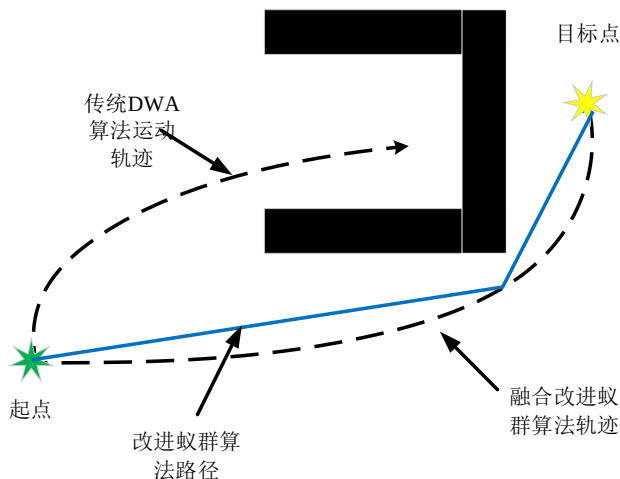


图 4-3 融合算法前后对比图

4.3.2 全局偏差函数

大多数机器人在运动过程中会在一定程度上偏离全局路径，假设有一个机器人正在沿着全局路径移动，当机器人靠近一个转折点时，局部路径规划可能导致机器人稍微偏离全局路径，这可能是由于局部环境的复杂性或者避免障碍物等原因。

如图 4-4 所示：假设全局路径由一系列的路径点 P_i 组成，机器人当前的位置为 (x, y) 。定义一个全局偏差项 $global(v, w)$ ，并引入方向偏差信息，考虑机器人当前方向与全局路径点方向之间的夹角。

$$global(v, w) = \min[d((x, y), P_i), \text{angle}((x, y), P_i)] \quad (4-6)$$

其中， (x, y) 是机器人当前位置坐标， θ 是机器人当前方向， P_i 是全局路径上的路径点。 $d((x, y), P_i)$ 表示机器人当前位置与全局最近路径点 P_i 之间的距离。

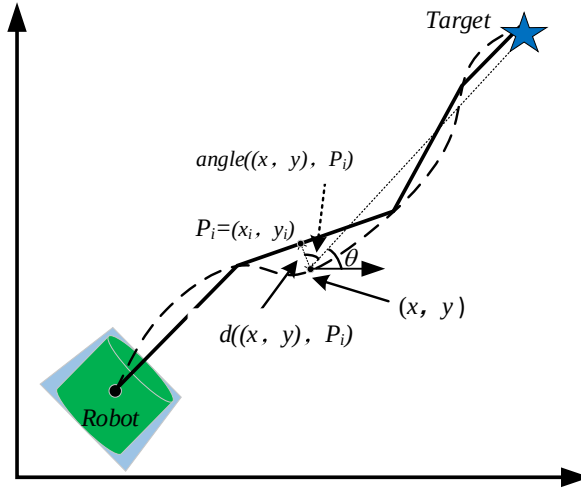


图 4-4 全局偏差函数示意图

全局偏差函数考虑了机器人当前位置与最近全局路径点的距离，以及机器人当前方向与全局路径点方向的夹角，目的是在局部路径选择时考虑全局路径，以便机器人尽可能地贴近全局路径。

4.3.3 改进的 $dist(v, w)$ 函数

传统的 $dist(v, w)$ 函数评估的是机器人到已知静态障碍物的距离，缺乏对未知动态障碍物的考虑。因此，对 $dist(v, w)$ 函数进行改进，优化未知障碍物信息以随时间动态更新。

首先利用运动学公式来预测未知移动障碍物在未来的位置，假设移动障碍物在当前时刻的位置为 (x_{ob}, y_{ob}) ，速度为 v_{ob} ，加速度为 a_{ob} ，则可以使用以下运动学公式来预测未来时刻的位置。

$$\begin{aligned} x_{\Delta t} &= x_{ob} + v_{ob} \cdot \Delta t + \frac{1}{2} a_{ob} \cdot (\Delta t)^2 \\ y_{\Delta t} &= y_{ob} + v_{ob} \cdot \Delta t + \frac{1}{2} a_{ob} \cdot (\Delta t)^2 \end{aligned} \quad (4-7)$$

其中， Δt 是未来时刻与当前时刻的时间间隔。然后，使用机器人当前位置 (x, y) 和预测的未来位置 $(x_{\Delta t}, y_{\Delta t})$ 之间的距离作为动态距离进行机器人的动态避障。

如图 4-5 所示：机器人需要提前估计 Δt 秒后移动障碍物的状态信息，在速度空间中模拟出一条能够避开障碍物的路径。

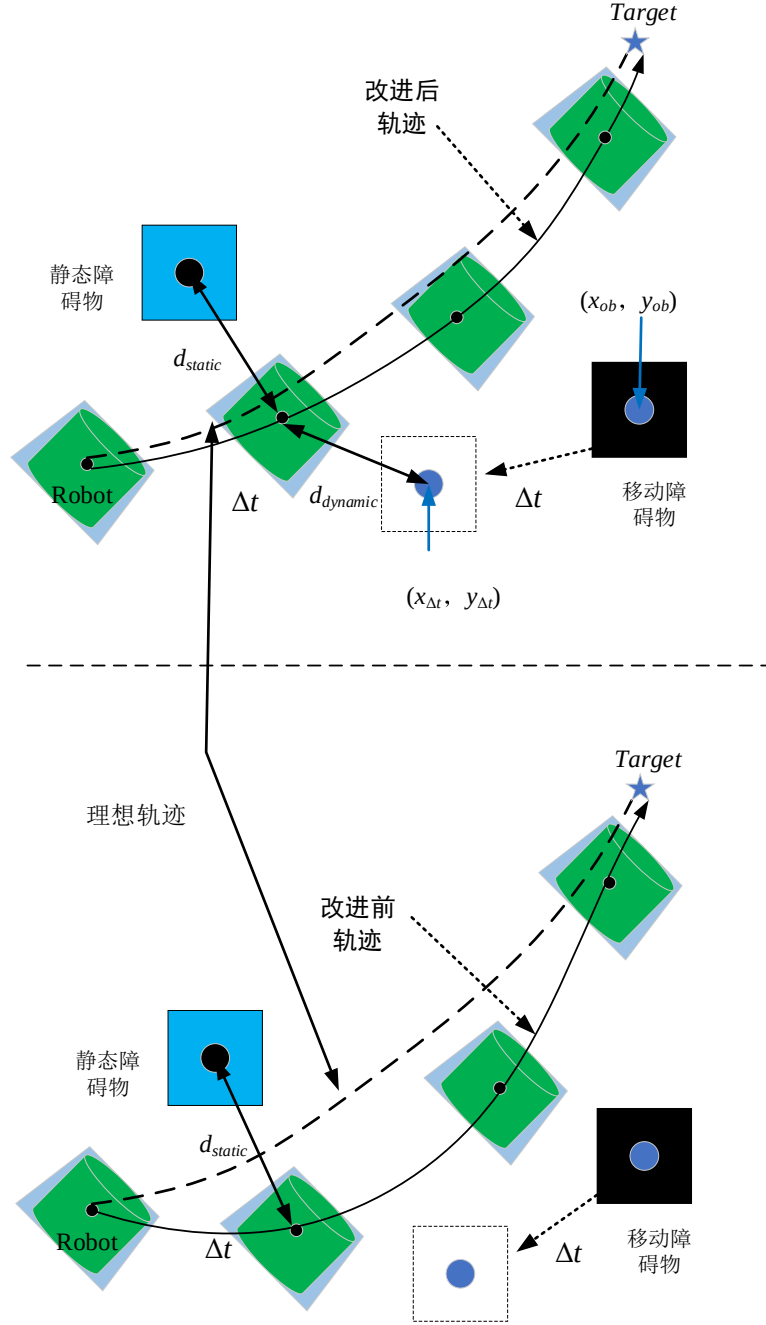


图 4-5 改进的 DWA 算法前后动态避障对比图

基于移动障碍物的线速度 v_{ob} 和机器人当前速度 v ，定义预测时间为：

$$\Delta t = \varphi \cdot \left(\frac{d((x, y), (x_{\Delta t}, y_{\Delta t})) * v}{v_{ob}} \right) \quad (4-8)$$

其中： v_{ob} 是机器人在运动空间中对移动障碍物的最小感知速度。 v 是机器人当前速度。 φ 是调整因子，用于控制感知时间的计算。

改进后的 $dist(v, w)$ 函数将计算从当前模拟轨迹点到移动障碍物在 Δt 时

间后的预测移动的距离，并丢弃那些与障碍物的最短距离小于或等于机器人的安全半径 R_{safe} 的轨迹，如式 4-9 所示。

$$dist'(v, \omega) = \begin{cases} discard, d_{min} \leq R_{safe} \\ d_{min}(r, ob(t + \Delta t)), R_{safe} \leq d_{min} \leq R_{max} \\ R_{max}, d_{min} \geq R_{max} \end{cases} \quad (4-9)$$

其中： $d_{min}(r, obs(t+\Delta t))$ 表示机器人模拟轨迹 r 到预测时间 Δt 后所有障碍物的最短距离， R_{max} 为最大安全距离。通过引入障碍物速度和加速度信息，该函数能更全面地考虑障碍物的运动状态，从而更好地指导机器人在动态环境中避障。

4.3.4 局部目标点的自适应插入法

在传统的 DWA 局部路径规划中，机器人可能只关注全局路径上的转折点作为局部目标点，对于障碍物密集、未知因素复杂的环境，单纯地将转折点作为局部导航点是不适用的，不能充分利用全局路径信息，当通过障碍物附近时，机器人容易偏离并失去正确的方向导致机器人无法到达。

通过在障碍物附近添加局部目标点，引领机器人靠近全局路径，局部目标点的自适应插入法如图 4-6 所示：首先检查全局路径上是否经过障碍物，如果是，则将局部目标点设置在距离障碍物最近栅格的全局路径上，引领机器人靠近全局路径。有助于提高在密集障碍物环境中的路径规划效果，防止机器人偏离正确方向。

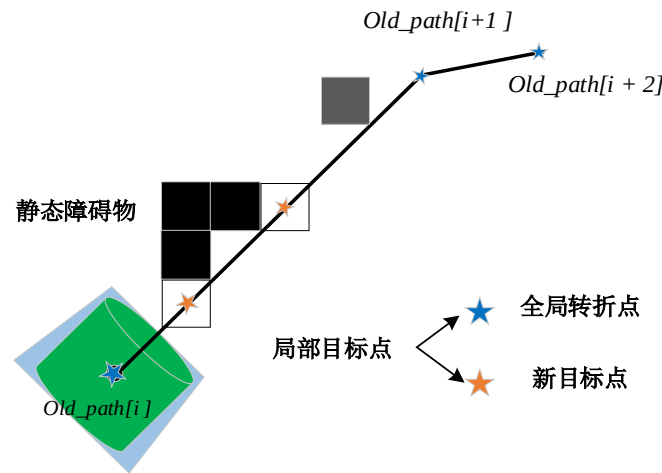


图 4-6 局部目标点自适应插入法

最终得到改进后的机器人避障的评价函数为:

$$G(v, w) = \sigma[\delta \cdot \text{heading}(v, w) + \eta \cdot \text{vel}(v, w) + \gamma \cdot \text{dist}(v, w) + \lambda \cdot \text{global}(v, w)] \quad (4-10)$$

其中: λ 为全局偏差评价函数 $\text{global}(v, w)$ 的权值。

4.3.5 多智能车的优先避障策略

为解决多智能车碰撞问题, 实验研究了智能车对动态障碍物的避障策略, 通过计算每个智能车的路径长度和转折点个数, 采用综合路径长度和转折点个数的综合优先级排序机制, 实现了智能车的协调避障。

$$\text{Priority}(\text{AGV}_i) = w_1 * \text{Path_Length}(\text{AGV}_i) + w_2 * \text{Turn_Points}(\text{AGV}_i) \quad (4-11)$$

其中: w_1 和 w_2 是两个影响因素的权重因子, 在进行局部路径规划时, 为每个智能车在网格环境下预先规划全局最优路径, 根据全局路径可以得出智能车的优先级。假设 AGV1 优先级最高, 其次是 AGV2, 优先级最低是 AGV3。

如图 4-7 所示: 全局路径交于一点, 当三个 AGV 发生碰撞冲突时, 优先级较低的 AGV2 和 AGV3 减速为零。当 AGV1 离开冲突区域后, AGV3 和 AGV2 之间仍然存在冲突。AGV3 停止等待, 待 AGV2 离开后恢复运动, 冲突解除。

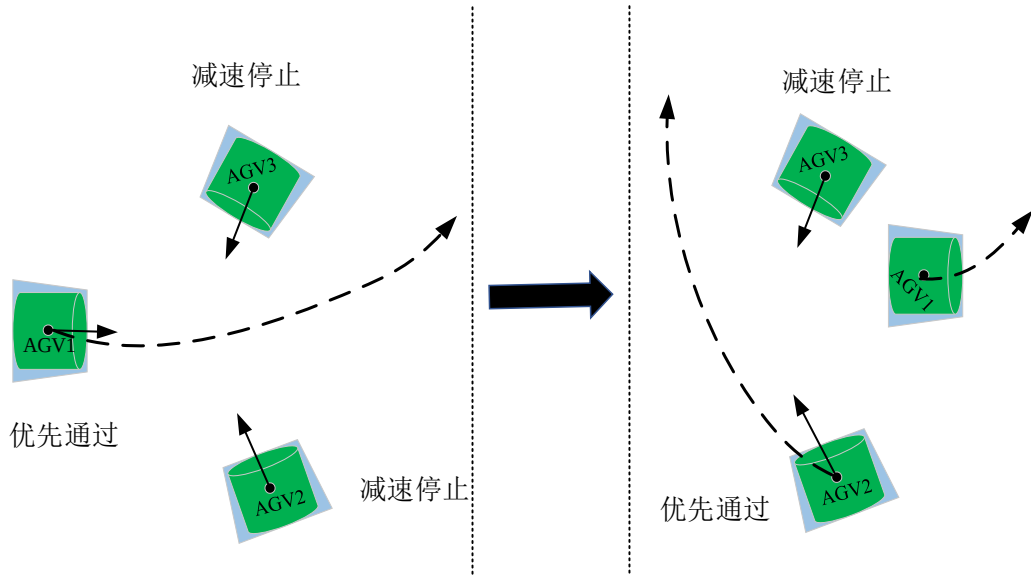


图 4-7 优先级避障示意图

4.3.6 总体改进算法的流程

针对传统 ACO 和 DWA 算法在大规模复杂环境中存在的问题，综合考虑这两种算法的优缺点，将改进的 ACO 和改进的 DWA 融合，使用 IACO 获得智能车的全局导航路径，并使用 IDWA 构建智能车的运动学模型进行动态路径规划。改进蚁群算法与改进的 DWA 融合可按以下步骤进行。融合算法的流程如图 4-8 所示。

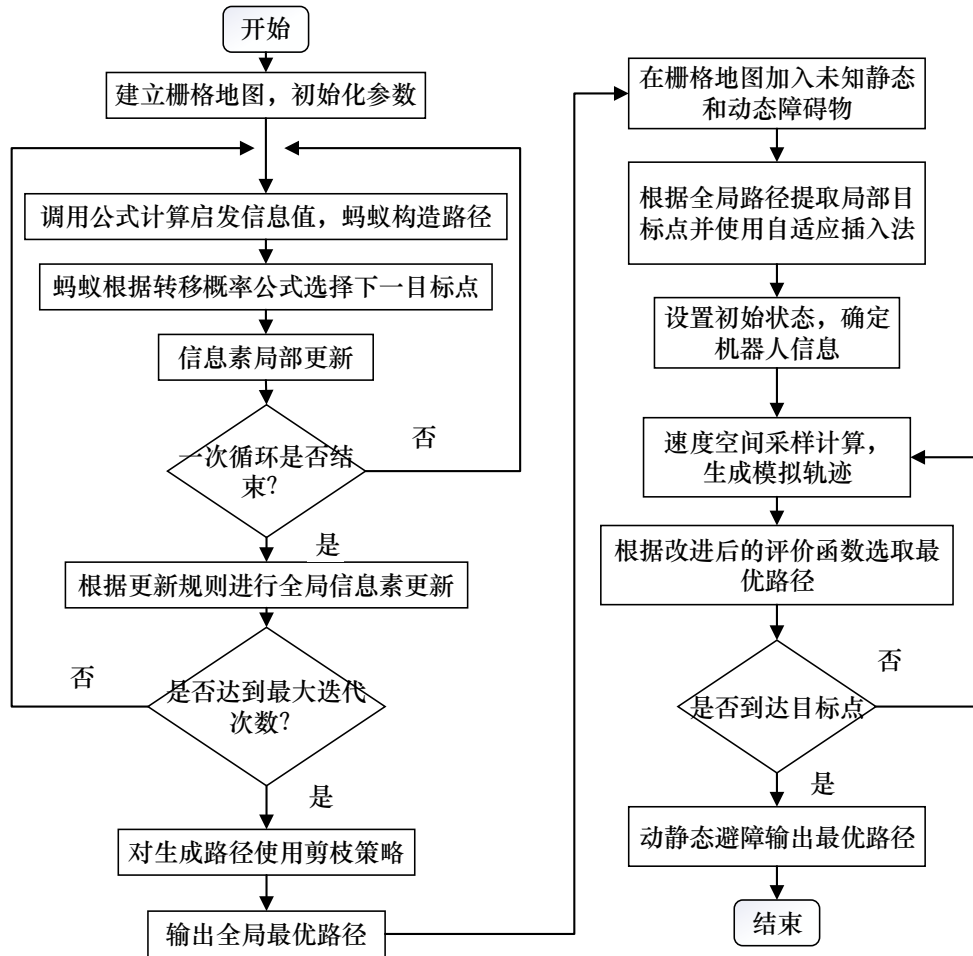


图 4-8 改进蚁群融合 DWA 算法的流程图

首先，对蚁群算法和 DWA 算法分别改进并确定环境模型。接着，通过蚁群算法生成全局路径并将信息传递给 DWA 算法用于初始化。然后进入分层规划阶段，全局规划层中蚁群算法通过蚂蚁搜索和信息素更新不断优化全局路径，局部规划层中改进的 DWA 算法生成动态窗口，计算评价函数选择最优速度。在动态调整与反馈环节，DWA 算法发现更好路径或遇障碍时反馈给蚁群算法调整信息

素分布和全局路径，蚁群算法也可根据 DWA 算法反馈优化全局路径并调整启发式函数。最后，不断重复迭代过程，智能车根据 DWA 算法规划的局部路径行动，持续优化路径规划。

4.4 IACO-IDWA 仿真实验与分析

为了验证改进的 ACO 融合改进的 DWA 的性能，实验选择传统 DWA 算法、文献[74]融合算法以及本文算法在未知环境中进行算法比较分析。文献[74]使用改进的 A*算法来规划高度安全的全局路径，并结合传统的 DWA 算法进行局部路径规划。DWA 算法的标准参数如表 4-1 所示：

表 4-1 DWA 算法参数设置

参数	参数描述	参数值
v_{max}	最大线速度	1.5m/s
ω_{max}	最大角速度	20°/s
v_a	线加速度	0.2m/s ²
ω_a	角加速度	50°/s ²
δ	/	0.05
η	/	0.2
γ	/	0.3
λ	/	0.15
/	速度分辨率	0.02m/s
/	角速度分辨率	1°/s
R_{safe}	安全距离	0.43m

首先验证其有效性，如图 4-9 所示是算法的对比实验图，通过引入全局偏差函数，本文算法相较于文献[74]算法，局部路径规划更加贴合全局路径。如表 4-2 所示：相较于传统 DWA 算法，本文算法的路径更为平滑，路径长度也有效的减少路径长度减少了 2%。本文算法在路径规划方面相对于传统 DWA 算法和文献[74]中提到的算法具有更好的性能，即在保证路径安全性的前提下，能够实现更短、更平滑的路径规划。

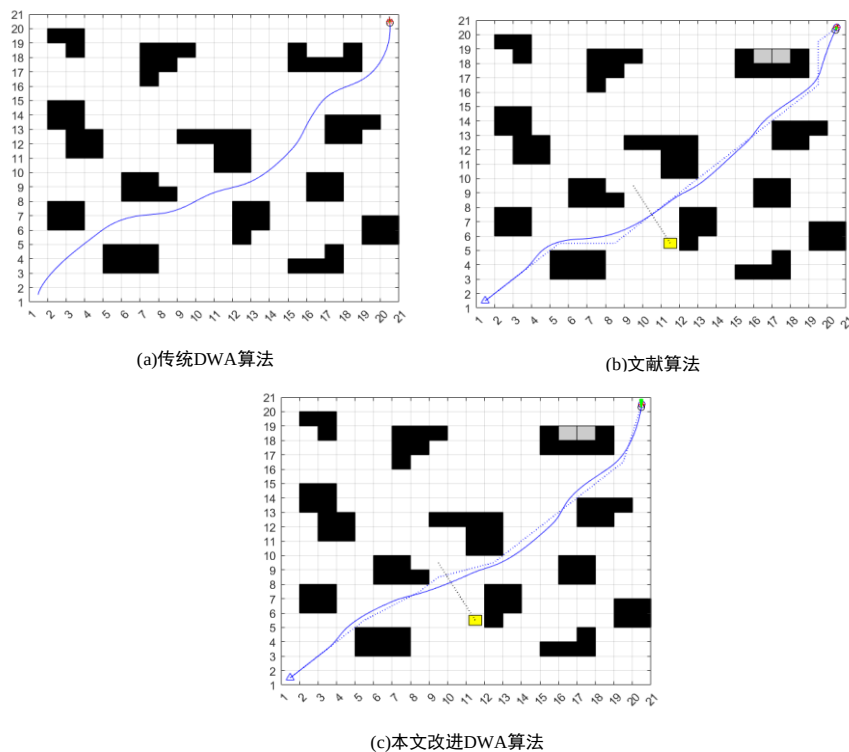


图 4-9 各 DWA 算法路径对比图

表 4-2 数据对比

算法	路径长度/米	全局路径转折点/个	路径是否安全
传统 DWA 算法	28.5170	—	是
文献[74]算法	28.0609	4	是
本文算法	27.9580	3	是

通过建立更加复杂的未知环境验证本文算法在复杂环境中的避障可行性。如图 4-10 所示为智能车避开单个移动障碍物的路径规划图。首先在 30×30 网格环境中建立智能车的静态环境，并基于 IACO 规划的最优全局路径。其中：(a)中粉色线是基于改进蚁群算法的全局路径规划，蓝色线是进一步去除多余转折点的路径，(b)是向环境中继续添加未知障碍物图，(c)是使用局部目标点自适应插入法插入新的局部目标点，(d)是融合算法的局部路径规划。由图可以看出智能车沿着全局路径移动，实时判断周围环境，并根据障碍物的情况避开障碍物，完成整体路径规划。

如图 4-11 所示是智能车运行过程中线速度角速度变化图，由于本文对 DWA 算法中 $dist(v, w)$ 函数的改进可以保证智能车提前预测移动障碍物的位置，以至

于智能车线速度整个过程平稳,没有减速停止的状态。由于本文提出的局部目标点的自适应插入法,当智能车经过障碍物附近时时,插入新的导航点,这避免了智能车在经过障碍物时偏离全局路径,可以更好地跟踪全局路径。如图 4-12 所示是智能车运行过程中姿态角度变化图,可以看出整个过程较为平稳。

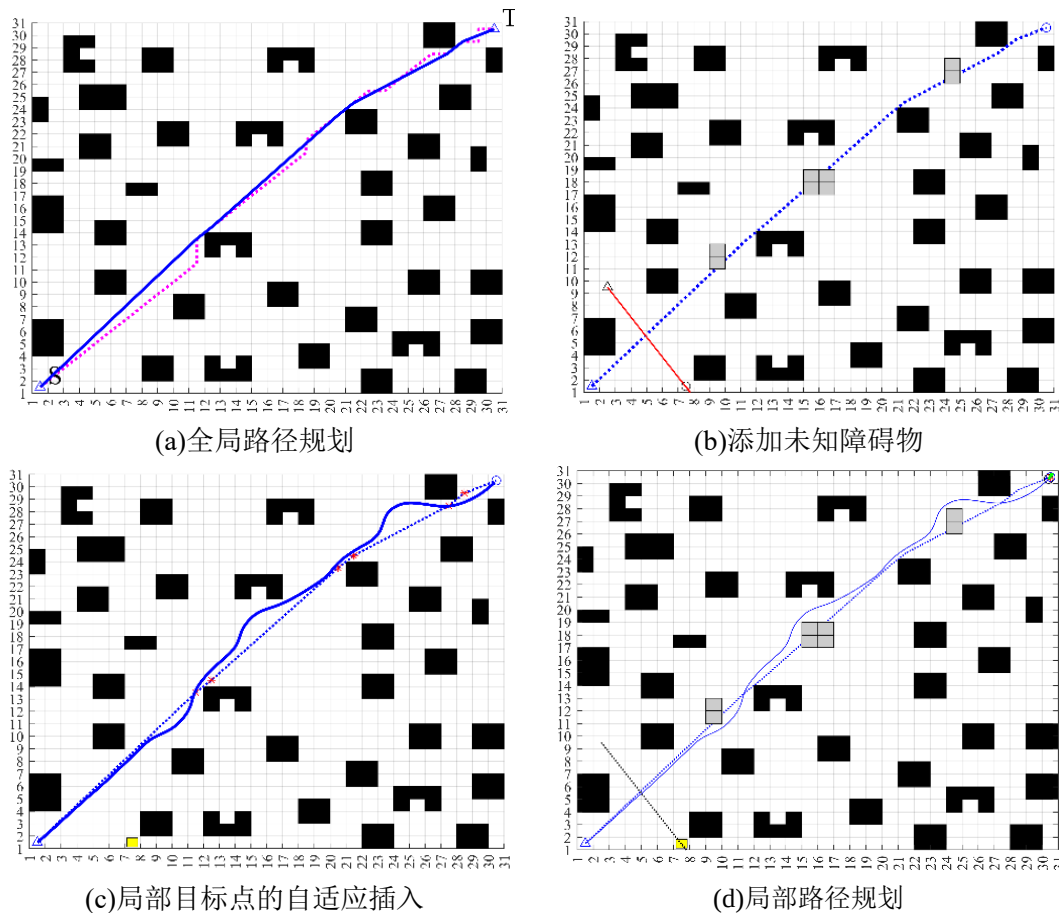
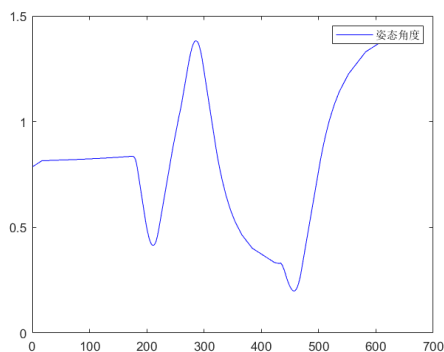
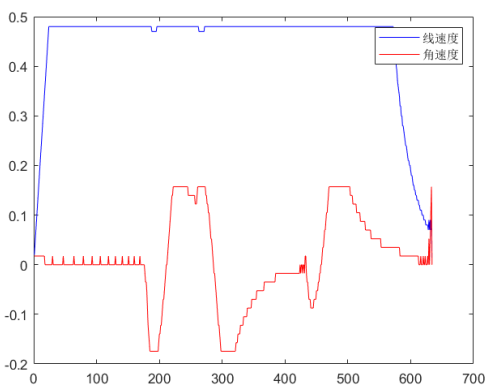


图 4-10 智能车避开单个移动障碍物的路径规划图



为了进一步体现本文融合算法的避障能力,如图 4-13(c)所示:在实验中,继续向地图中添加更多更大的未知障碍物和更多的移动障碍物,使其在智能车运动的方向经过,其中:移动障碍物起移动速度为 0.16m/s ;并且轨迹用红色实线标记。在智能车运动过程中,智能车通过传感器发现以 0.16m/s 的速度向自己移动的移动障碍物,执行避障策略。

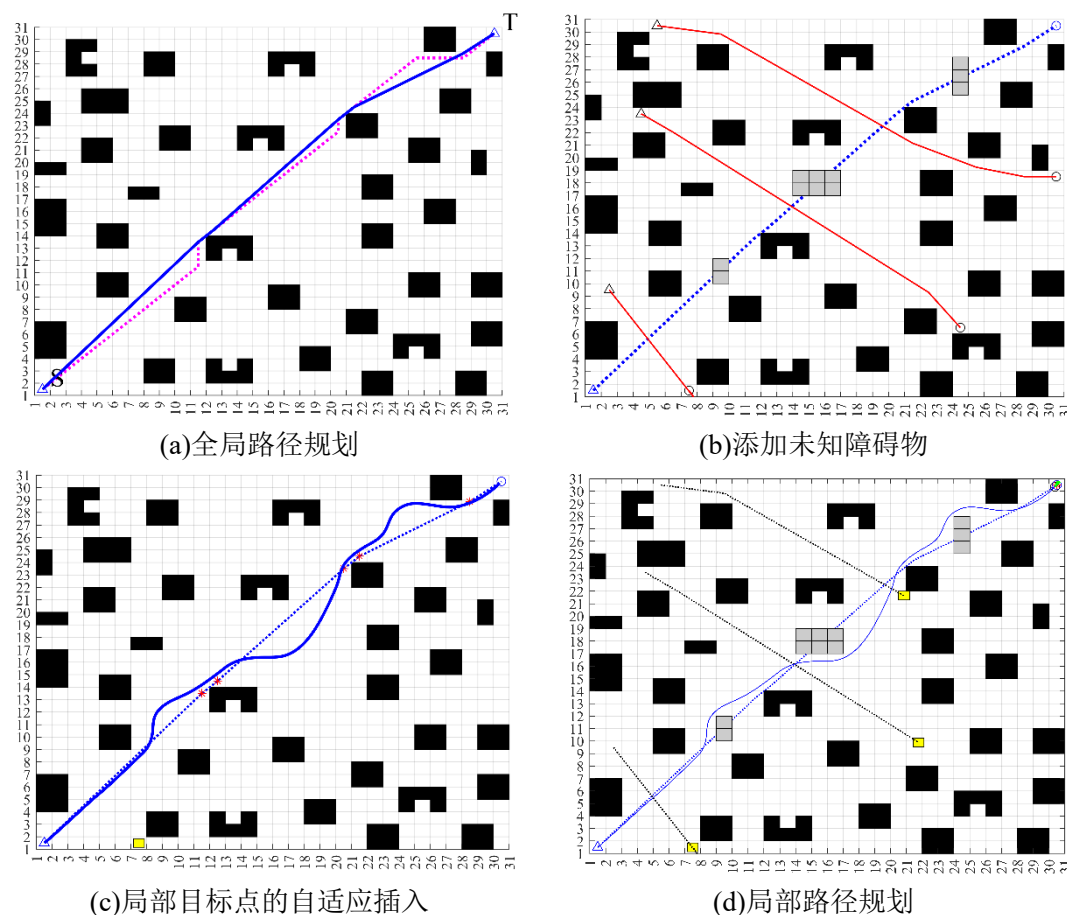


图 4-13 智能车避开多个移动障碍物的路径规划图

仿真实验中智能车的线速度、角速度和姿态角的变化记录在图 4-14 和 4-15 中。从图中数据可知,在改进评估函数后,即便智能车面临更大未知障碍物以及更多移动障碍物的复杂环境,其运动总体依旧保持稳定。在整个运动过程中,线速度能够保持匀速状态,未出现剧烈波动,同时角速度的变化相对平稳,未产生频繁且大幅度的转向调整,体现出优化策略对智能车动态控制性能的显著提升。

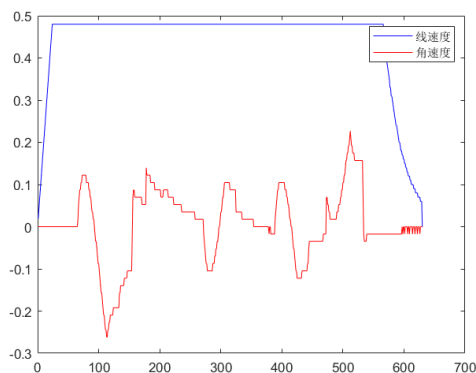


图 4-14 智能车线速度角速度变化图

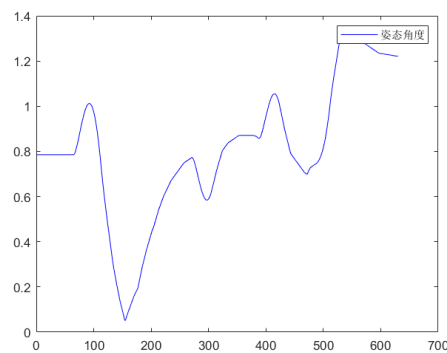


图 4-15 智能车姿态角变化图

在整个仿真实验中，智能车沿着全局路径进行实时路径规划，路径与原始路径基本相同。IDWA 的跟踪能力再次得到验证。智能车在这条路径下安全通过，没有碰撞任何障碍物，安全到达目标点。

4.5 多智能车优先级策略

首先验证多级智能车在未知环境避障的有效性，然后验证本文提出的综合路径长度和转折点个数的综合优先级排序机制在解决多智能车路径规划中的冲突回避问题的有效性，在考虑运动学模型约束和障碍物环境约束的情况下，设计了三个智能车，利用改进的 DWA 构建多智能车运动学模型，进行了以下实验。

如图 4-16 所示：本文改进的蚁群算法为每一个智能车规划了一条全局最优路径，其中蓝色、粉色和红色线分别是智能车 1、智能车 2 和智能车 3 的全局路径。由全局规划路线根据转折点和路径长度对智能车进行优先级排序。智能车的初始位姿的角度分别为-0.7854、0.8270 和 0.3332。三个智能车的路径长度分别为 30.4440m、35.5700m 和 28.4960m。如图 4-16 所示实时显示了智能车在规划过程中的变化，其中：(a)为全局路径规划，(b)为局部路径规划，(c)为三个智能车运行过程中线速度变化图，(d)为三个智能车运行过程中角速度变化图，(e)为三个智能车运行过程中姿态角度变化图。

从图中可以看出智能车 1 优先级最高，其次是智能车 2，最后是智能车 3，从图 29(c)可以看出当低优先级的智能车 2 遇到高优先级的智能车 1 时，智能车减速等待，当智能车 1 通过后智能车 2 再通过。

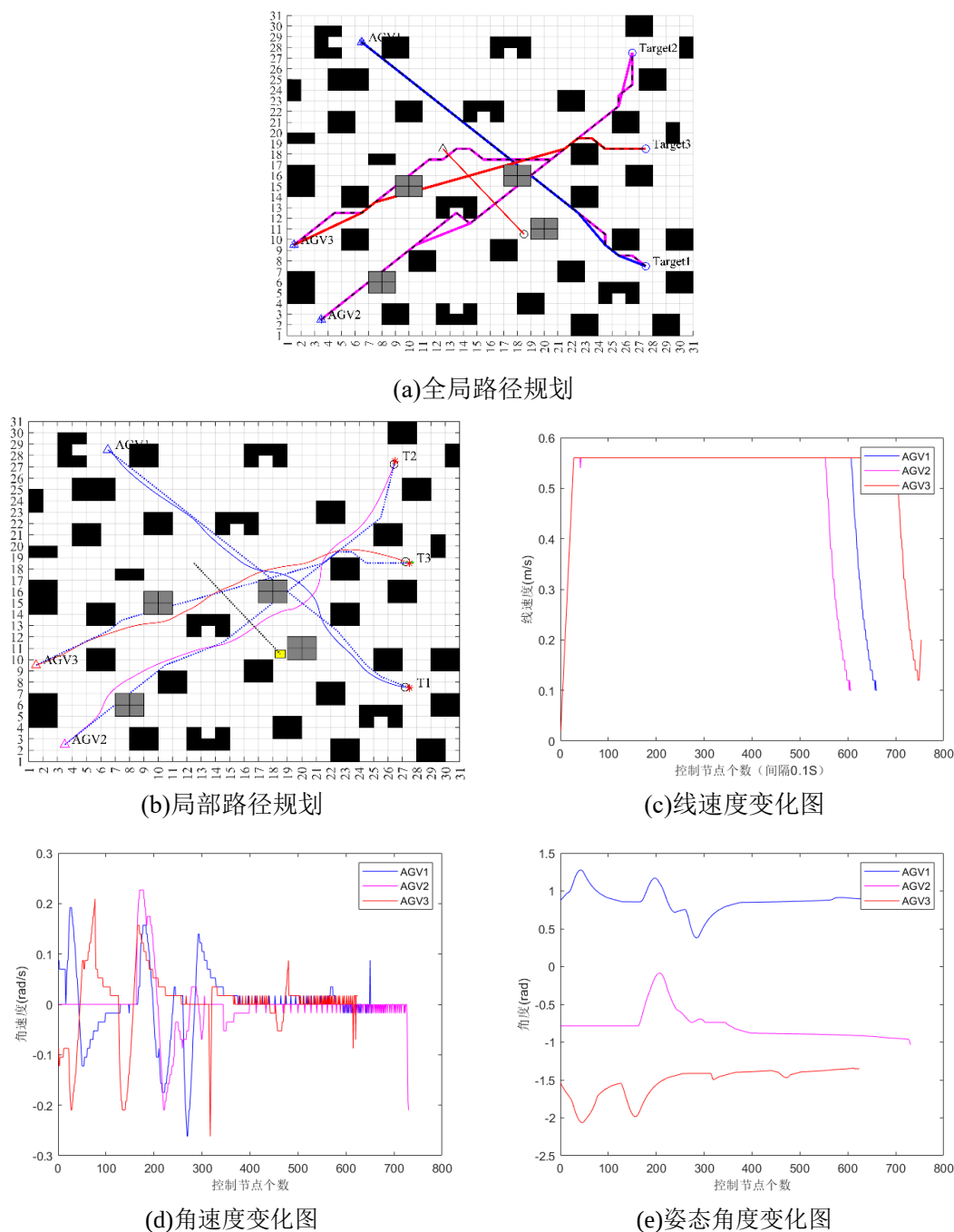


图 4-16 多智能车动态路径规划图

该实验仿真结果证明了多智能车优先避障策略在具有随机静态和未知动态障碍物的环境中的有效性。由图 4-16(c)可以看出智能车 3 由于遇到未知动态障碍物在 20-30 控制节点有一个减速过程，其他两个智能车整体过程平稳。

以上实验证明了多智能车避障的有效性，但是由于智能车并未都相遇，所有以下实验通过设计使得智能车全部相遇从而验证提出的优先级碰撞协调机制的有效性。智能车的初始方向角分别为 0.6747 、 -0.6202 和 -0.0624 。三个智能车的

额路径长度分别为 19.7040m、16.9740m 和 16.0040m。如图 4-17 所示实时显示了智能车在规划过程中的变化。

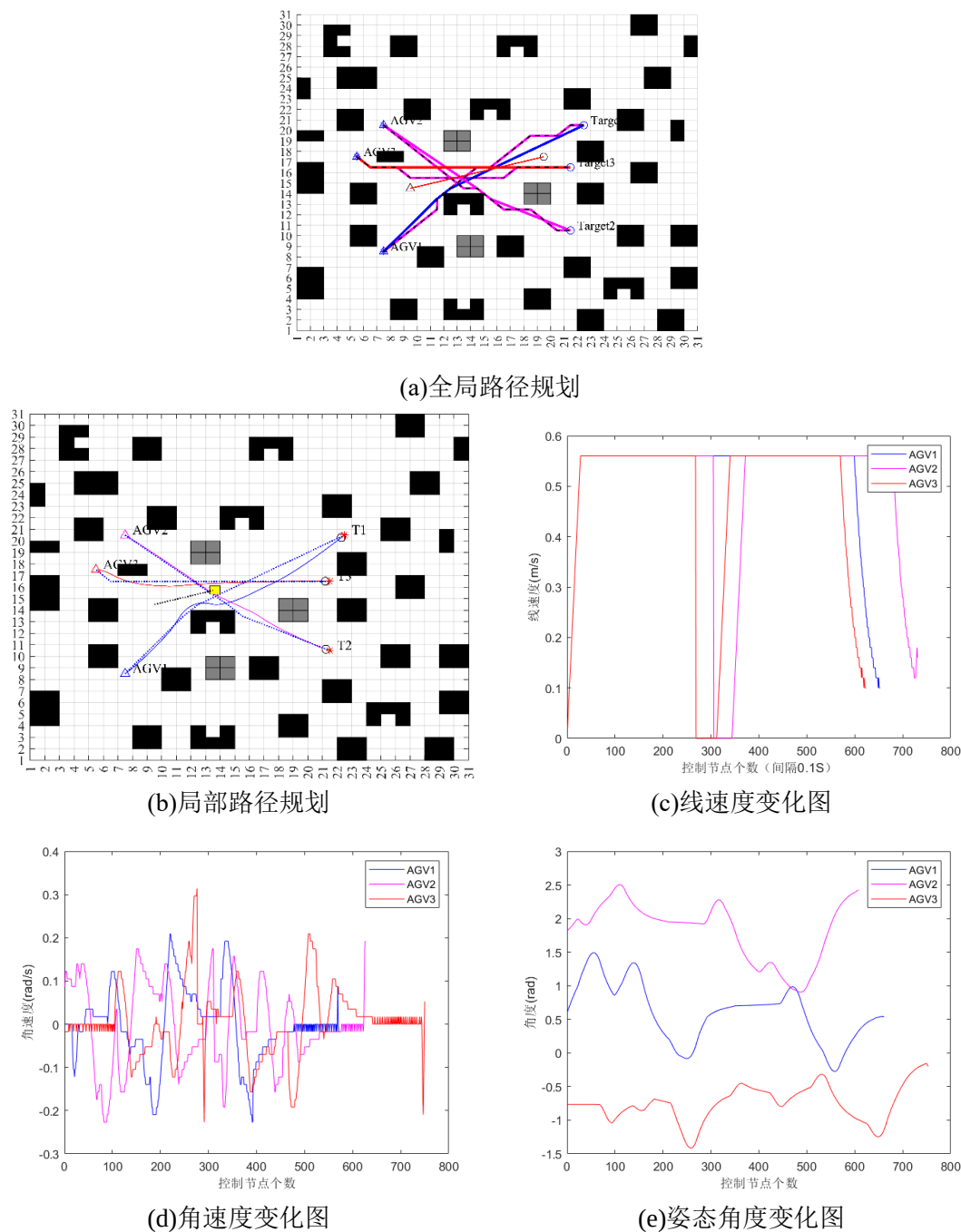


图 4-17 多智能车优先级路径规划图

从图 4-17(c)可以看出，当最低优先级智能车 3 和高优先级智能车 2 之间存在运动冲突时，智能车 3 在第 270 个运动控制节点开始减速，在第 280 个运动控制节点完全停止，并在第 300 个运动控制节点恢复运动，冲突解除。然后智能车 2 由于最高优先级的智能车 1 相遇，智能车 2 开始从第 310 控制节点减速到完全

停止，并且在第 300 处冲突解决的情况下恢复运动。整个过程中，在机器之间相遇时，高优先级智能车将低优先级智能车视为未知静态障碍物，根据改进的优先级避障策略进行合理的避障。

在最低优先级智能车 3 的避障过程中，智能车 3 比较高优先级的智能车 2 和智能车 1 执行更多的避障过程，这会产生更明显的角速度变化和全局路径偏差，但线速度在相遇之后都正常恢复运动，整个避障顺利完成。

4.6 本章小结

本章针对传统 DWA 算法在局部路径规划中的不足，提出了一系列改进措施。通过引入全局偏差函数，使智能车运动轨迹更好地贴合全局路径；改进的障碍物评价函数提高了对动态障碍物的避障能力；局部目标点的自适应插入法有效解决了智能车在障碍物附近的路径偏离问题；多智能车的优先避障策略实现了智能车之间的协调避障。

仿真实验表明，改进的蚁群算法与改进的 DWA 算法融合后，在路径规划性能上有显著提升。与传统 DWA 算法和其他相关算法相比，该融合算法规划的路径更短、更平滑，且在复杂环境中具有良好的避障能力和稳定性，有效验证了改进算法在局部路径规划中的有效性和实用性，为多智能车在复杂动态环境中的路径规划提供了可靠的解决方案。

第5章 基于 ROS 平台的智能车自主导航

5.1 引言

前面的第三章和第四章中,提出的融合改进蚁群和 DWA 算法仅在 MATLAB 中进行了仿真实验。这种仿真实验虽然能够初步验证算法的一些特性,但并不能充分证明改进算法在实际的智能车应用中也能取得良好效果。因此,本章将把改进算法实际应用到 ROS 系统中集成的 Rviz 可视化平台和 Gazebo 物理仿真平台,对智能车的导航功能进行仿真研究,以此进一步验证改进算法的有效性。

具体而言,首先会在 Gazebo 中搭建仿真环境,选择与智能车对应的模型,通过键盘操作智能车进行地图构建。在获取其栅格地图后,对论文所提融合算法的有效性展开验证。然后,在真实环境下利用智能车的激光雷达进行扫描建图,得到真实环境对应的栅格地图,再在真实环境中利用智能车验证融合改进蚁群和 DWA 算法的可行性。

5.2 系统基本介绍

ROS 是专为机器人软件开发打造的开源平台。它采用分布式架构,各节点可在不同主机运行,支持 Python、C++ 等多语言开发,扩展性强且源码开放。其核心由负责特定功能并通过消息通信的节点、(ROS1 中的)提供登记查找服务的主节点、存储静态值的参数服务器、基于发布/订阅机制的主题、作为通信载体的消息,以及实现请求 - 响应式通信的服务构成。同时,它还拥有如管控通信的 roscore、用于开发的 roscpp/rospy 库、3D 可视化的 Rviz、进行仿真的 Gazebo、集成开发环境 rqt、构建管理功能包的 catkin、记录回放消息的 rosbag,以及启动多节点并配置参数的 roslaunch 等丰富开发工具。凭借这些特性与工具,ROS 广泛应用于工业自动化的机器人装配搬运、农业的智能耕种收割、医疗的手术康复辅助、家庭服务的清洁陪伴、娱乐的机器人游戏,以及学术研究的算法开发测试等众多领域,有力推动机器人技术不断创新与发展。

当前,在智能车导航领域,Rviz 可视化平台与 Gazebo 物理仿真平台是两款

应用最为广泛的仿真软件。在本研究中，同样借助这两款软件对智能车的导航功能展开仿真探究。

(1) Rviz

Rviz 是 ROS 生态中极为重要的三维可视化平台，基于插件架构设计，功能丰富且扩展性强。它能将激光雷达测距、三维点云、相机图像、里程计等各类传感器数据，以直观的三维图形形式呈现，并支持用户通过点击、拖拽等交互操作筛选数据、调整视角、设置目标点。其界面布局合理，涵盖用于展示内容的 3D 视图显示区、包含多种功能按钮的工具栏、可设置显示项目及参数的显示项目区、提供时间信息的时间显示区域，以及能设置观测视角的对应区域。凭借这些特性，Rviz 在机器人研发与调试、运行监控管理、教育演示，乃至自动驾驶技术的研发测试等诸多场景中，都发挥着助力直观理解复杂信息、高效定位并解决问题、辅助优化算法策略的关键作用。Rviz 可视化平台的界面如图 5-1 所示。

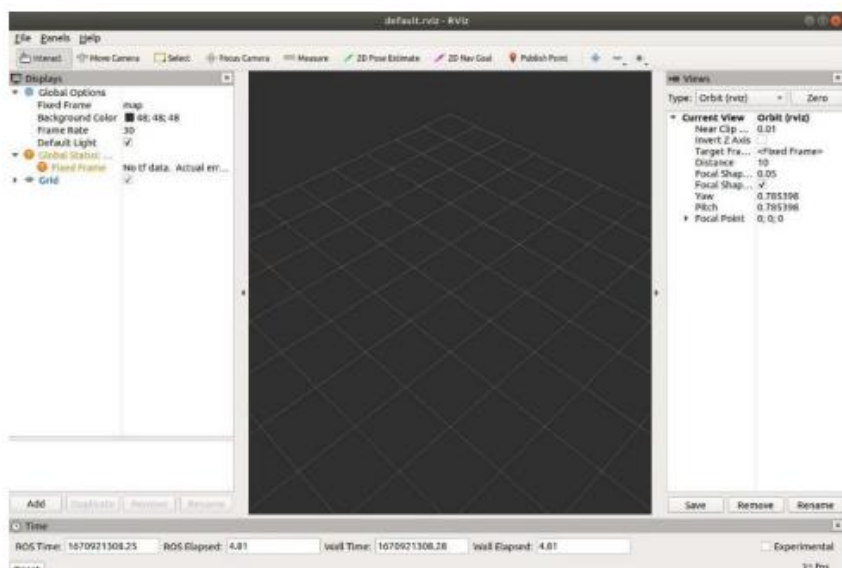


图 5.1 Rviz 可视化平台界面图

(2) Gazebo

Gazebo 是由开源机器人基金会 (OSRF) 开发维护的强大开源机器人仿真器。它具备突出的物理仿真能力，借由 ODE、Bullet 等多种高性能物理引擎，逼真模拟重力、碰撞等物理现象，精准呈现机器人在虚拟环境中的运动效果。在传感器模拟方面，激光雷达、摄像头等各类传感器均可模拟，且能添加噪声生成近似真实的数据。环境建模功能丰富，既有大量现成环境模型，又支持自定义创建复杂

室内外场景。凭借插件扩展特性，开发者可编写自定义插件，便捷拓展功能。其界面布局合理，场景区展示仿真模型，左右面板用于操作与交互，工具栏包含各类常用操作及模拟控制按钮，菜单栏提供更多选项。基于这些特性，Gazebo 在机器人研发中助力算法与系统调试，为教育培训搭建学习实践平台，还广泛应用于算法验证及多机器人协作研究等领域，推动机器人技术不断进步。如图 5-2 所示为 Gazebo 的图形界面。

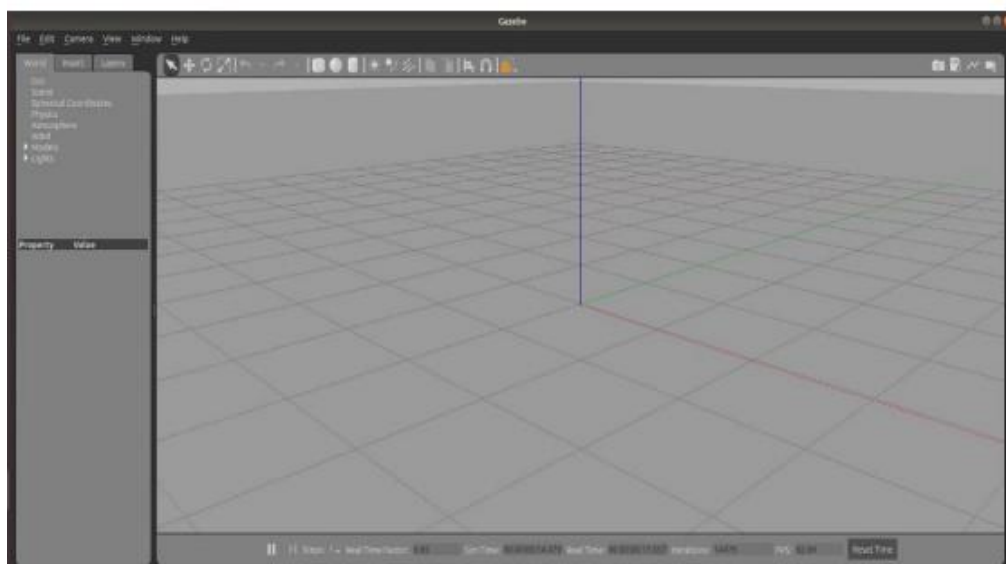


图 5-2 Gazebo 平台的图形界面图

5.3 搭建 ROS 仿真实验平台

首先，需要准备合适的硬件设备和操作系统。本文选用 Ubuntu 18.04 系统作为基础平台，该系统在智能车开发领域具有良好的兼容性和稳定性，能够为 ROS 软件的运行提供可靠的环境支持。在硬件方面，确保计算机具备足够的处理能力、内存以及存储容量，以应对复杂的仿真计算和数据存储需求。

接下来进行 ROS 软件的安装。根据 Ubuntu 系统的版本和架构，从 ROS 官方网站下载相应的安装包。在安装过程中，按照官方文档提供的详细步骤逐步进行操作，确保安装过程的完整性和正确性。安装完成后，需要进行一系列的配置工作，包括设置环境变量、初始化 ROS 工作空间等。通过修改系统的 `.bashrc` 文件，将 ROS 的相关路径添加到环境变量中，使得系统能够正确识别和调用 ROS 命令及工具。创建 ROS 工作空间是组织和管理项目代码的重要环节，在工作空

间中创建 Src 文件夹，并在其中放置项目相关的功能包源代码。

然后是功能包的创建与编译。根据实验需求，使用 `catkin_create_pkg` 命令创建自定义的功能包，在创建过程中指定功能包所依赖的其他 ROS 功能包，确保功能包能够正确引用所需的库和工具。例如，对于涉及智能车导航和路径规划的功能包，可能需要依赖于导航相关的功能包（如 `move_base`、`AMCL` 等）以及传感器数据处理的功能包。创建完成后，进入工作空间的根目录，使用 `catkin_make` 命令对功能包进行编译，该命令会自动检测功能包的依赖关系，并生成可执行文件和库文件。在编译过程中，需要注意查看终端输出的信息，及时解决可能出现的编译错误，如依赖项缺失、代码语法错误等。

此外，还需要对 Gazebo 和 Rviz 软件进行配置和集成。在 Gazebo 中，根据实验场景的需求导入合适的智能车模型和环境模型。对于本文中的智能车路径规划实验，导入智能车模型，并搭建相应的室内仿真环境，设置环境的尺寸、障碍物分布等参数。在 Rviz 中，配置相关的显示选项，使其能够准确地显示智能车的位姿、传感器数据以及规划路径等信息。通过调整 Rviz 的可视化参数，如视角、颜色、显示类型等，以获得清晰、直观的可视化效果，方便实验过程中的观察和分析。

通过以上步骤，成功搭建了基于 ROS 的仿真实验平台，为后续的智能车路径规划实验提供了坚实的基础。在搭建过程中，每一个步骤都需要仔细操作和验证，确保平台的稳定性和功能性，以满足实验的严格要求。

5.4 基于激光雷达建图及导航实验

为探究改进蚁群算法和动态窗口法的融合算法在智能车导航中的效能，本研究选用 Ubuntu 18.04 系统，依托 ROS Melodic 框架，借助 Gazebo 与 Rviz 两款软件推进验证。

实验前，利用 Gazebo 搭建仿真环境。其自定义建模工具能依据实验需求，从环境规模、地形到障碍物分布等，构建更加真实的场景。Rviz 在验证流程中负责展示数据，将智能车轨迹、传感器信息、规划路径等以三维图形呈现，助力研究人员把握实验情况。

仿真分三步：首先，在 Gazebo 搭建环境并导入定制智能车模型，搭建时考虑场地尺寸、地面材质、障碍物布局等，对模型设定车身规格、质量分布、动力及传感器参数，让模拟环境和模型接近真实。其次，联合 Rviz，用 Slam 中的 gmapping 算法构建栅格图。Gmapping 算法靠激光雷达数据计算地图，Rviz 实时显示构建过程。最后，基于栅格图开展路径规划自主导航仿真。融合算法依环境信息和智能车状态规划路径，动态窗口法兼顾车辆运动学约束与环境动态变化，确保路径高效且安全。Rviz 全程显示车的轨迹、规划路径及环境交互，方便研究人员评估算法性能，为优化算法提供参考，提升智能车导航技术。

5.4.1 智能车模型建立

ROS 系统为智能车建模提供了支持 XML 规范的 URDF 智能车描述格式，可描述智能车各部件的尺寸、连接状态、碰撞参数及颜色等，但单独的 URDF 文件无法显示智能车模型，需结合 Rviz 可视化平台和 Gazebo 物理仿真平台。智能车在 Rviz 中的仿真模型如图 5-3 所示。

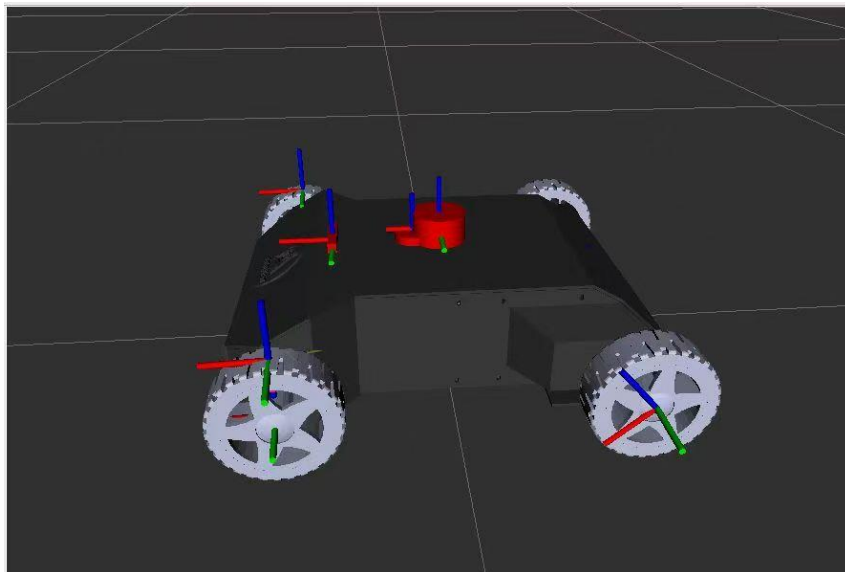


图 5-3 智能车 Rviz 模型图

在智能车研发中，仿真环境搭建对算法验证至关重要，主要借助 Gazebo 物理仿真平台完成。该平台有房间、树木、家具等丰富插件，可依需求搭建各类现实场景。

为验证改进算法可行性，设置两类实验环境。静态环境在原有地图上加静态

障碍物，用于验证智能车面对静态障碍时算法的有效性，可观察算法能否依传感器反馈快速规划避障路径，并评估规划精准度。动态环境则在原有地图基础上添加未知静态和移动障碍物，移动障碍物随机移动，增加环境复杂性。借此验证智能车面对未知障碍时算法的实时性与安全性，观察智能车能否实时感知环境变化并合理决策。这两种环境设置可多维度验证改进算法，为智能车实际应用奠定技术基础。

5.4.2 仿真环境地图的构建

智能车导航功能主要包括定位与路径规划，在 ROS 系统中分别由 AMCL 与 move_base 功能包实现。move_base 可进行全局和局部路径规划，要应用本文改进的算法，需编写对应插件。具体是在 move_base 功能包的全局和局部路径规划基础上修改，得到 Better ACO 全局和 Better DWA 局部路径规划插件，然后对插件注册、调整参数文件并添加到 ROS 包系统。使用时，在 Rviz 中把全局和局部路径规划算法分别替换为这两个插件，智能车即可借助它们实现改进算法的路径规划。

完成全局路径和局部路径的插件配置后，利用 ROS 导航软件包中的 Slam 算法，通过 Slam_gmapping 节点构建地图。Slam_gmapping 节点是 ROS gmapping 应用软件包的核心，它能获取激光数据信息和 tf 数据信息，并发布占用栅格地图数据信息。智能车启动后，在主控终端加载雷达参数，接着运行虚拟机并打开可视化工具 Rviz，在 Rviz 界面启动 Slam.Rviz。此时，智能车利用 AMCL 功能包实现自身定位，开始创建地图。

如图 5-4 为整个建图过程。当 Gazebo 和 Rviz 成功启动后，在终端输入 `roslaunch teleop_twist_keyboard teleop_twist_keyboard.py` 运行键盘控制节点，便可以实时控制移动智能车。随着智能车在室内环境中不断移动，Rviz 中的地图会实时更新。当智能车在室内运动一圈后，Rviz 中的地图创建完成。完成激光雷达扫描地图后，保存地图数据，最后使用 `map_server` 保存地图并提供给智能车，生成一个 .yaml 元数据文件和 .pgm 图片文件。其中，.yaml 文件主要描述地图信息，包括分辨率、图片路径和占用阈值等。

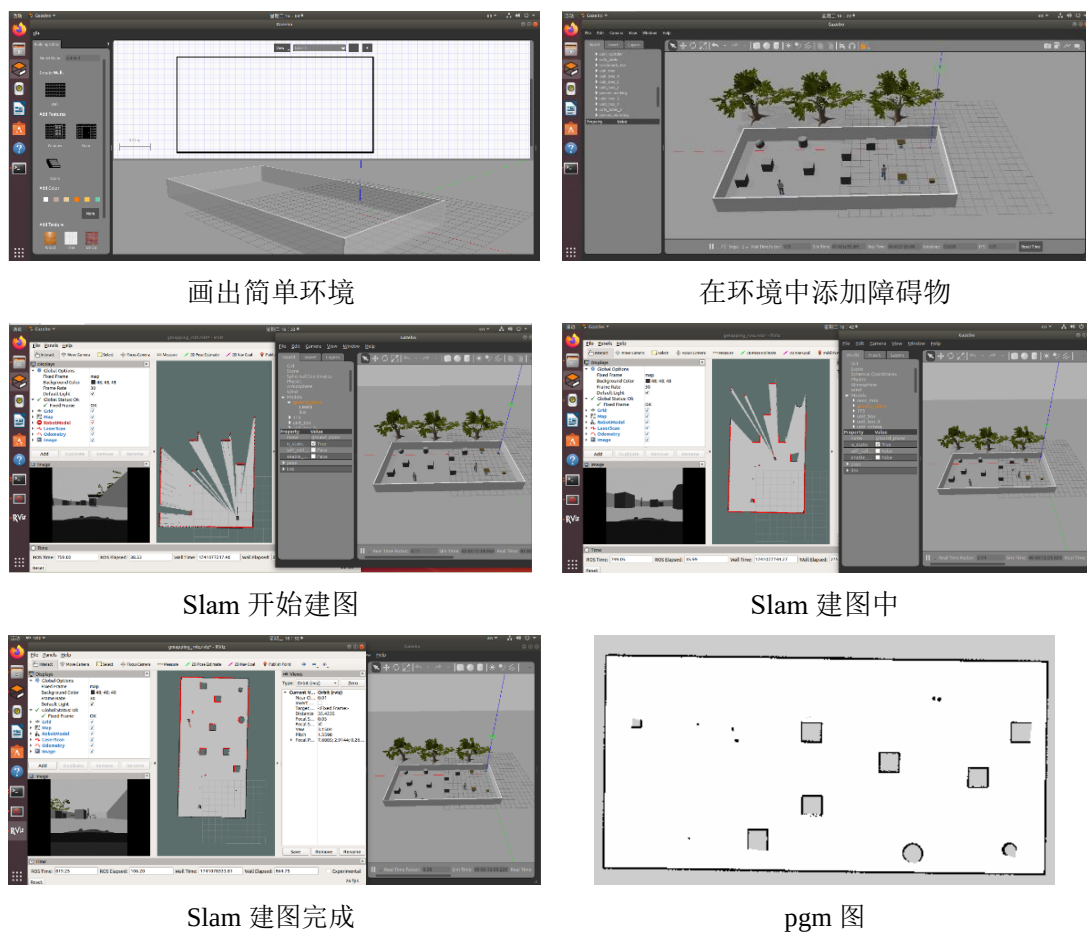


图 5-4 Slam gmapping 建图过程

5.4.3 基于 ROS 的自主导航仿真验证

智能车进行路径规划和导航，ROS 提供了完备的智能车导航功能模块，如图 5-5 展示了智能车导航模块的工作架构。目标点设置后，全局路径规划器结合坐标变换传感器信息、里程计信息，在全局代价地图辅助下（数据来自地图服务）进行路径规划。生成的全局路径传递给局部路径规划器，局部路径规划器再参考局部代价地图，结合里程计信息，进一步规划更细致路径。AMCL 定位提供定位数据支持。最终规划好的路径用于运动控制，使智能车按规划路径驶向目标点，整个过程通过各模块协同与信息交互实现智能导航。

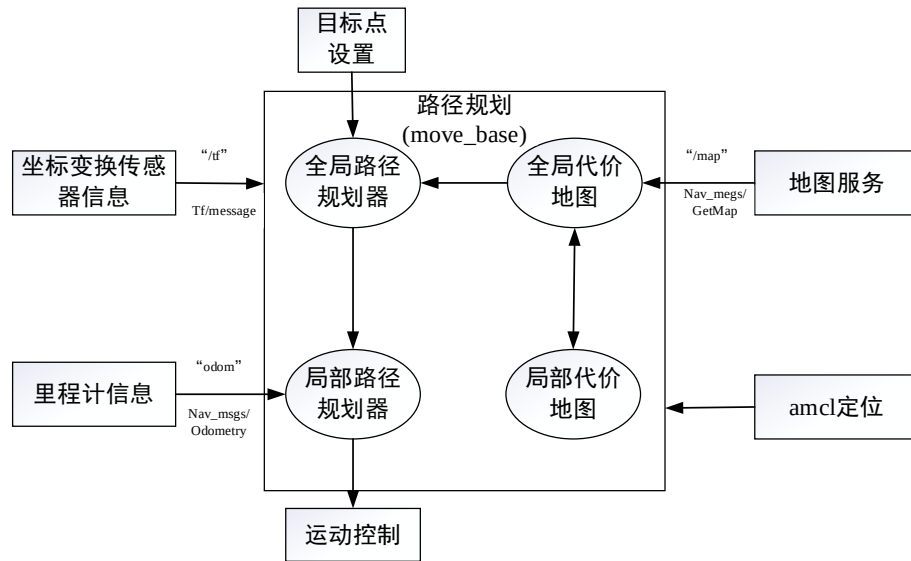


图 5-5 智能车导航模块图

(1) 静态环境

为验证融合算法的避障功能，在实验环境中加入各种障碍物，完成地图构建的准备工作后，通过 2D Nav Goal 确定需要到达的目标点，Rviz 中会显示出规划出来的路径，选择不同的终点进行多组对照实验确保实验的准确性。

小车依据全局路径规划算法向目标点移动，移动的过程中雷达不断扫描当前环境，当靠近障碍物时，依靠路径规划算法避开障碍物，实现避障。绿色方格是智能车，红色箭头表示小车移动的轨迹，箭头的方向表示小车在移动过程中实际的朝向。绿色实线表示小车规划的路径。如图 5-6 所示是小车运动过程图。通过实验观察发现小车能够成功绕开障碍物实现避障功能且能根据规划的路径安全到达终点。

图 5-6 呈现的是改进蚁群融合 DWA 算法的融径规划导航实验。从结果中可以看出：在 0s 时，改进算法下的小车同样位于起点，处于待出发状态；36s 时，小车开始移动到障碍物附近，开始准备避障；44s 时，小车在面对障碍物时，开始避障；54s 时，小车完成避障；77s 时，小车开始识别到站立的人；81s 时，小车开始避开站立的人；87s 时，小车很好的避开站立的人，108s 时，小车顺利到达目标点，且整个行驶路径平滑，它能够有效减少冗余转折节点，使智能车在避开障碍物的同时，保持与障碍物的安全距离，大大提升了智能车在复杂环境中的导航性能和安全性，为智能车在实际应用中的高效运行提供了有力的技术支持。

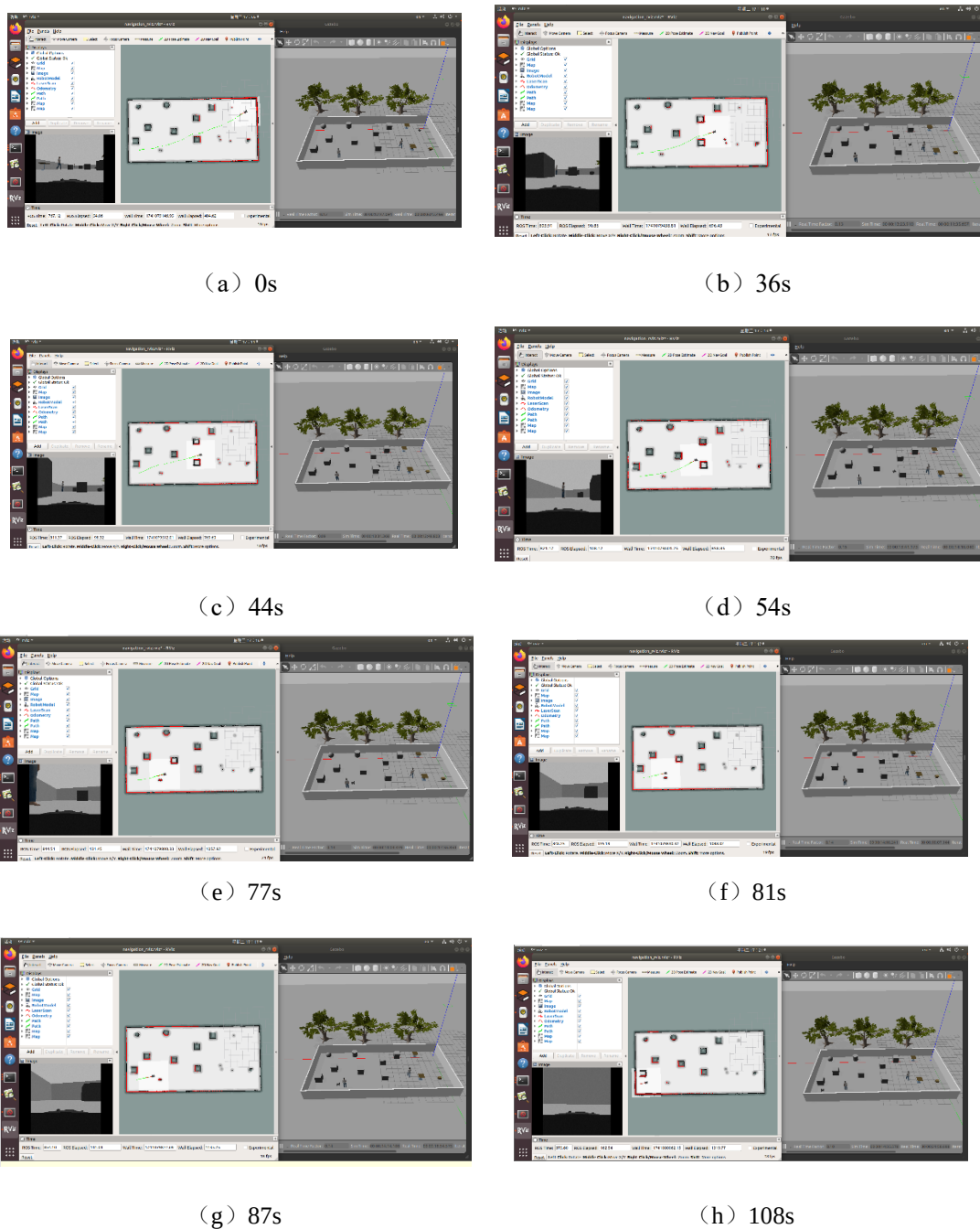


图 5-6 改进蚁群融合 DWA 算法的智能车静态环境导航实验

(2) 动态环境

在复杂的实际应用场景中，智能车常常会面临充满动态变化的环境，如物流仓库中频繁移动的人员与货物、工业车间里位置不定的设备等。为了验证改进蚁群融合 DWA 算法在这类复杂动态环境下的可靠性与高效性，设置了如下实验。在该动态环境实验中，模拟了多种复杂情况，通过观察智能车在不同场景下的运行表现，来检验改进算法是否能精准感知环境变化、及时规划合理路径并实现安

全避障，进而为智能车在实际动态场景中的应用提供有力支撑。

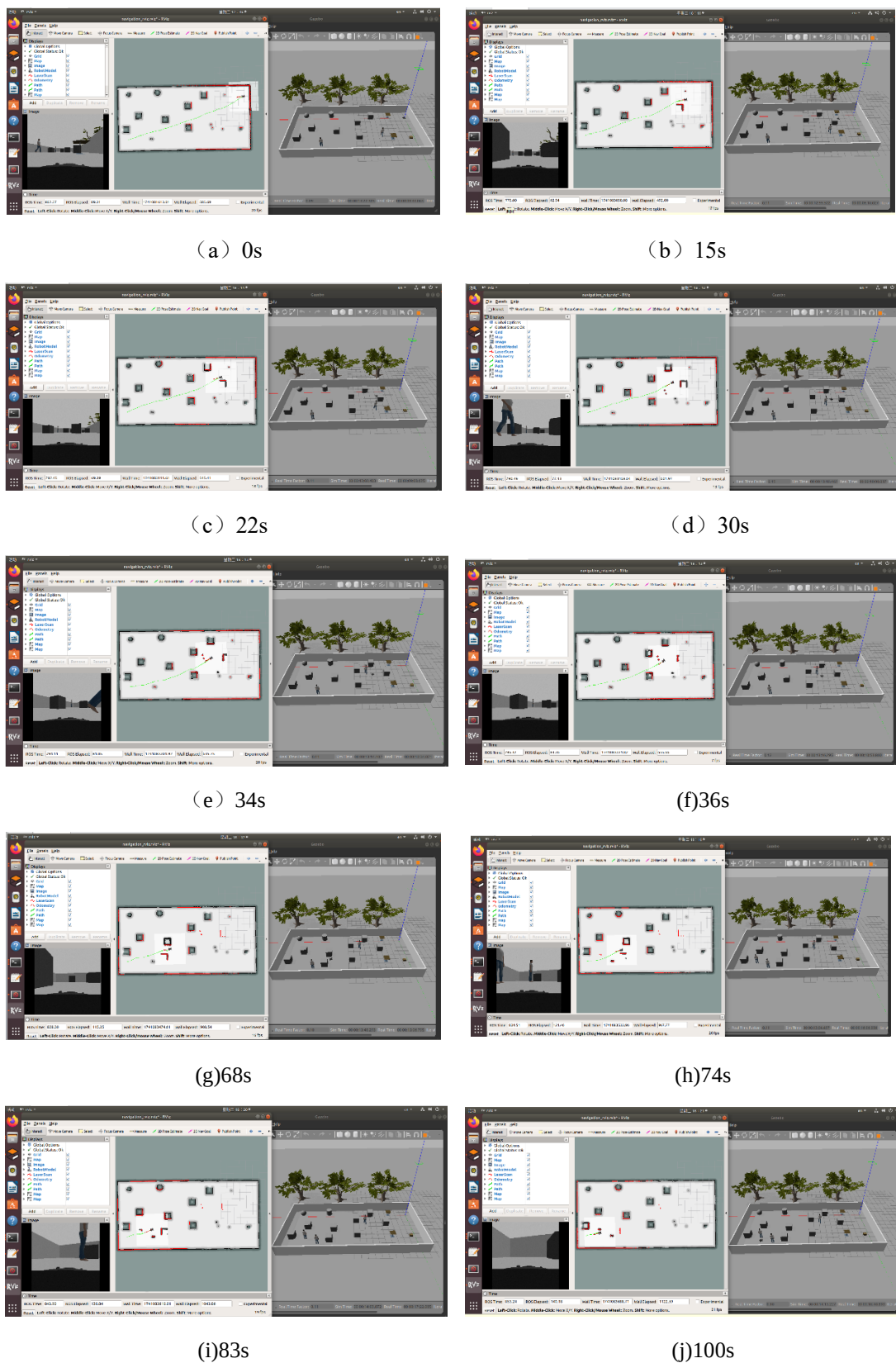


图 5-7 改进蚁群融合 DWA 算法的智能车动态环境导航实验

如图 5-7 所示中智能车导航可以看出，在 0s 时，改进算法下的小车同样位于起点，处于待出发状态；15s 时，对环境中添加未知的静态障碍物，小车识别到障碍物，开始避障；22s 时，小车正在避障；30s 时，小车完成对未知静态障碍物的避障；对环境中继续添加缓慢移动的人；34s 时，小车识别到行人，并开始避开；36s 时，小车正在避开行人；继续对小车行驶轨迹中添加障碍物；68s 时，小车识别到未知障碍物；74s 时，智能车避开障碍物，同时继续添加站立的人，83s 时开始避开站立的人，100s 时，小车顺利避开未知站立的人，向目标点行驶且整个行驶路径平滑。

5.5 本章小结

本章聚焦于 ROS 平台，在融合局部与全局路径规划算法的基础上深入开展相关工作。首先详细介绍了 ROS 的特点与系统框架，充分展现其在智能车开发领域的独特优势，为后续实验奠定理论基础。接着基于 Ubuntu 操作系统，依循特定流程成功搭建了 ROS 仿真实验平台，涵盖相关软件安装、工作空间与功能包创建及编译、环境变量配置等关键步骤，为实验提供了运行环境。

此外，搭建智能车平台，利用 ROS 智能车操作系统控制智能车，采用 gmapping 算法构建环境地图。分别在仿真环境和实际操作平台开展一系列实验，在 Gazebo 中搭建仿真环境并操作智能车建图以验证融合算法有效性，在真实环境下利用激光雷达扫描建图得到栅格地图，进一步验证融合改进算法的可行性。通过上述各项实验工作，有力地证明了在 ROS 平台上改进算法的有效性，为智能车在复杂环境中的路径规划提供了可靠技术支持，同时也为后续的深入研究与广泛应用筑牢了坚实基础。

第6章 总结与展望

6.1 工作总结

本文针对智能车路径规划问题,深入分析了传统蚁群算法和动态窗口算法的不足,提出了一系列改进措施,并通过多方面实验验证了改进算法的有效性。

针对传统蚁群算法在全局路径规划中的不足,本文设计了综合多种因素的势场启发函数,使蚂蚁在路径选择时能更好地避开风险区域,提高路径安全性;利用 Dijkstra 算法处理初始信息素,依据障碍物和路径长度调整其浓度,加快了算法初始阶段的收敛速度;引入信息素差分机制,根据最优和最差路径上蚂蚁数量动态调整信息素更新量,避免算法过早陷入局部最优;采用自适应调整机制,让信息素挥发系数随迭代次数变化,平衡全局和局部搜索能力;运用剪枝策略,消除冗余转折点,优化了路径质量。

传统 DWA 算法在局部路径规划方面有诸多不足,如缺乏全局引导、环境适应性低等。针对这些问题,改进的 DWA 算法引入全局偏差函数,使智能车在局部规划时能更好地贴合全局路径,减少偏离。改进障碍物评价函数,利用运动学公式预测未知移动障碍物位置,提高了动态避障能力。采用局部目标点的自适应插入法,根据障碍物分布和智能车状态在合适位置插入目标点,引导智能车安全绕过障碍物。提出多智能车的优先避障策略,综合计算路径长度和转折点个数来确定优先级,实现了智能车间的协调避障。

在实验验证方面,首先在 MATLAB 中对比改进前后蚁群算法的多维度性能,包括路径长度、迭代次数、搜索时间和转折点个数等。结果显示改进蚁群算法在各方面均有显著提升,验证了其在全局路径规划中的有效性。其次在 ROS 平台搭建仿真环境,利用 Rviz 和 Gazebo 进行大量仿真,在二维静态和动态环境下设置实验,模拟智能车实际工作场景。实验表明,改进蚁群算法与改进 DWA 算法融合的方案能让智能车在未知环境中有效协同避障,提升了智能车在复杂场景下路径规划的工作效率。

6.2 未来展望

在智能车路径规划研究中,当前成果虽有进展,但仍存在优化空间。以下从算法性能、环境适应性、多机器人协作、新兴技术融合四个方面指出待解决问题,并阐述未来研究方向与期望成果,旨在推动智能车路径规划技术进一步发展。

(1) 算法性能优。改进后的 ACO-DWA 融合算法虽在路径长度、迭代次数等方面有提升,但仍有优化空间。参数设置目前依赖经验和试验,难以适应复杂多变的环境。未来可深入研究自适应参数调整机制,利用机器学习技术,让算法根据环境和任务实时自动调整参数。研究更高效的搜索策略,如结合深度学习的强大特征提取和预测能力,引导算法更快找到最优路径,减少计算资源消耗,提高算法效率和寻优能力。

(2) 复杂环境适应性增强。实际应用中,智能车面临的环境极为复杂,现有算法难以完全适应。在复杂光照、电磁干扰等特殊环境下,传感器数据易出现误差,影响路径规划的准确性。未来可加强多传感器融合技术研究,综合激光雷达、摄像头、超声波传感器等多种传感器数据,提高环境感知的准确性和可靠性。针对不同复杂环境,如户外极端天气、室内复杂布局等,开发特定的算法优化策略,提升智能车在各类复杂环境下的路径规划能力。

(3) 多智能车协作能力拓展。目前多智能车的优先避障策略在大规模场景下存在不足,难以满足复杂协作需求。在多智能车协同作业时,任务分配、资源竞争和通信延迟等问题会影响系统效率。未来可研究多智能车协作机制,综合考虑任务优先级、资源分配和通信状况等因素,实现高效协作。开发分布式路径规划算法,让多智能车在无中心控制情况下自主协调路径,提高系统灵活性和鲁棒性,适应大规模复杂任务。

(4) 新兴技术融合。随着科技发展,新兴技术为智能车路径规划带来新机遇,但目前融合应用还不够深入。未来应加强新兴技术与路径规划算法的融合研究,利用深度学习对环境进行更精准感知和预测,提前规划更合理路径。借助 5G 通信技术降低多机器人通信延迟,提高协作效率,推动智能车路径规划技术向智能化、高效化发展。

参考文献

- [1] Zhu Q, Yeh M C, Cheng K T, et al. Fast human detection using a cascade of histograms of oriented gradients[C]//2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06). IEEE, 2006, 2: 1491-1498.
- [2] 许东亮,张学习,熊晓明,等.基于激光雷达云台的救援机器人定位建图系统[J].工业控制计算机,2021,34(06):73-75.
- [3] 张豫宁,史珂,陈裕汉,等.区域连续运行基准站BDS/GPS精密相对定位精度分析[J].工程勘察,2025,53(03):71-77.
- [4] 张军锋.基于人工智能技术的物流路径优化:应用挑战、行业实践与应用策略[J].物流技术,2025,44(01):61-72.
- [5] Goian A, Ashour R, Ahmad U, et al. Victim localization in USAR scenario exploiting multi-layer mapping structure[J]. Remote Sensing, 2019, 11(22): 2704.
- [6] Negri S P, Basile V, Valori M, et al. A modular mobile robotic architecture for defects detection and repair in narrow tunnels of CFRP aeronautic components[J]. Robotics and Computer-Integrated Manufacturing, 2019, 55: 109-128.
- [7] Girshick R, Felzenszwalb P, McAllester D. Object detection with grammar models[J]. Advances in neural information processing systems, 2011, 24.
- [8] Squyres S W, Knoll A H, Arvidson R E, et al. Exploration of Victoria crater by the Mars rover Opportunity[J]. Science, 2009, 324(5930): 1058-1061.
- [9] 叶政韩,吕文涛,郭庆,等.基于树莓派的AI识别智能车控制系统[J].智能计算机与应用,2025,15(02):47-51.
- [10] 高冲.智能移动机器人技术现状以及开发研究[J].信息与电脑(理论版),2019,(06):130-131.
- [11] 王亚飞.THMR-V 平台上的智能交通监控系统设计与实现[J].计算机测量与控制,2017,25(07):106-109.
- [12] Smith R. On the estimation and representation of spatial uncertainty[J]. Int. J. Robotics Res., 1987, 5(4): 113-119.

- [13]胡金涛,元泽怀,陈英子,等.基于激光雷达的智能小车自主导航研究[J].信息与电脑(理论版),2024,36(20):30-32.
- [14]丁蓉,张美娜.“无人配送车”产业迎来快速发展期[N].证券日报,2025-02-10(B02).
- [15]任祥瑞,王正刚,汤俊杨.改进 JPS 算法融合 DWA 的多机器人路径规划[J].计算机应用研究,2024,41(11):3251-3257.
- [16]Ajeil F H, Ibraheem I K, Azar A T, et al. Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments[J]. Sensors, 2020, 20(7): 1880.
- [17]王保剑,胡大裘,蒋玉明.改进A*算法在路径规划中的应用[J].计算机工程与应用,2021,57(12):243-247.
- [18]Song Q, Li S, Yang J, et al. Intelligent Optimization Algorithm-Based Path Planning for a Mobile Robot[J]. Computational intelligence and neuroscience, 2021, 2021(1): 8025730.
- [19]Çakır E, Ulukan Z, Acarman T. Time-dependent Dijkstra's algorithm under bipolar neutrosophic fuzzy environment[J]. Journal of Intelligent & Fuzzy Systems, 2022, 42(1): 227-236.
- [20]张志远,陈海进,章一鸣.基于局部障碍率预获取和双向父节点变更的A*算法优化[J].计算机工程与科学,2023,45(09):1661-1669.
- [21]Abbaspour R A, Samadzadegan F. RETRACTED: Time-dependent personal tour planning and scheduling in metropolises[J]. 2011.
- [22]张钟元,戴炜,李光昱,等.基于改进人工势场和一致性协议的协同避障算法[J].计算机应用,2023,43(08):2644-2650.
- [23]Zang X, Jiang L, Ding B, et al. A hybrid ant colony system algorithm for solving the ring star problem[J]. Applied Intelligence, 2021, 51: 3789-3800.
- [24]Yi G, Feng Z, Mei T, et al. Multi-AGVs path planning based on improved ant colony algorithm[J]. The Journal of Supercomputing, 2019, 75: 5898-5913.
- [25]Yang H, Qi J, Miao Y, et al. A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization[J]. IEEE Transactions on Industrial

- Electronics, 2018, 66(11): 8557-8566.
- [26] Golden B. Shortest-path algorithms: A comparison[J]. Operations Research, 1976, 24(6): 1164-1168.
- [27] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [28] Xu Z, Liu X, Chen Q. Application of improved Astar algorithm in global path planning of unmanned vehicles[C]//2019 Chinese Automation Congress (CAC). IEEE, 2019: 2075-2080.
- [29] 岳高峰,张萌,沈超,等.移动机器人导航规划的双向平滑A-star算法[J].中国科学:技术科学,2021,51(04):459-468.
- [30] 槐创锋,郭龙,贾雪艳,等.改进A*算法与动态窗口法的机器人动态路径规划[J].计算机工程与应用,2021,57(08):244-248.
- [31] 张恒,何丽,袁亮,等.基于改进双层蚁群算法的移动机器人路径规划[J].控制与决策,2022,37(02):303-313.
- [32] Li X, Wang L, Wang L. Application of improved ant colony optimization in mobile robot trajectory planning[J]. Math. Biosci. Eng, 2020, 17(6): 6756-6774.
- [33] 王晓燕,杨乐,张宇,等.基于改进势场蚁群算法的机器人路径规划[J].控制与决策,2018,33(10):1775-1781.
- [34] 刘瀛.基于改进势场与蚁群算法的机器人路径规划法[J].计算机仿真,2021,38(11):355-360.
- [35] Miao C, Chen G, Yan C, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm[J]. Computers & Industrial Engineering, 2021, 156: 107230.
- [36] Zhai L, Feng S. An improved ant colony algorithm based on artificial potential field and quantum evolution theory[J]. Journal of Intelligent & Fuzzy Systems, 2022, 42(6): 5773-5788.
- [37] 王鹏杰,李丹,付金岗,等.融合改进RRT和TEB算法的移动焊接机器人路径规划[J].安徽工业大学学报(自然科学版),2024,41(06):620-626.

- [38]温颖,王冬梅,路玉麟.基于改进A*和改进DWA相融合的AGV路径规划[J].农业装备与车辆工程,2024,62(09):123-127.
- [39]黄金源,陈熙源.基于改进A*与DWA的机器人融合算法研究[J].传感技术学报,2024,37(12):2043-2049.
- [40]衷卫声,闵志豪,权略,等.改进人工势场引导的双向扩展随机树路径规划算法[J].探测与控制学报,2024,46(03):86-93.
- [41]Wu J, Ma X, Peng T, et al. An improved timed elastic band (TEB) algorithm of autonomous ground vehicle (AGV) in complex environment[J]. Sensors, 2021, 21(24): 8312.
- [42]陈奕梅,沈建峰,李柄棋.改进TEB算法的多机器人动态避障策略研究[J].电光与控制,2022,29(05):107-112.
- [43]Chang L, Shan L, Jiang C, et al. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment[J]. Autonomous robots, 2021, 45: 51-76.
- [44]Sun Y, Chen W, Lv J. UAV path planning based on improved artificial potential field method[C]//2022 International Conference on Computer Network, Electronic and Automation (ICCNEA). IEEE, 2022: 95-100.
- [45]王伟,常彦文,余涛,等.基于Lattice算法的多智能车编队运动规划方法[J].探测与控制学报,2022,44(05):105-111.
- [46]崔南方,张成,胡于丹,等.一种基于DQN的Memetic算法求解集成有限智能车的低碳作业车间调度问题[J]. 运筹与管理, 2022, 31 (7): 1-12.
- [47] 段焜.基于多目标的动态环境智能车路径规划算法[J].信息技术,2023,(06):66-70.
- [48]许宏鑫,吴志周,梁韵逸.基于强化学习的自动驾驶汽车路径规划方法研究综述[J].计算机应用研究,2023,40(11):3211-3217.
- [49]Li X, Zhou M C. A Petri net approach for deadlock prevention in automated guided vehicle systems [J]. IEEE Transactions on Automation Science and Engineering, 2008, 5 (2): 315-324.
- [50]张秀艳,王正肖,潘晓弘.基于Flexsim的自动化立体仓库智能车调度系统建模

- 与仿真[J]. 系统仿真学报, 2007, 19 (17): 3951-3954.
- [51] 武晓博, 龚兴涛, 杨皓元. 基于车联网技术的高速公路智能车路协同系统设计[J]. 自动化技术与应用, 2024, 43(12): 20-24.
- [52] 卓辉. 试论 5G时代人工智能技术在工业生产中的应用[J]. 广西通信技术, 2019, (02): 45-47.
- [53] 钟佩思, 曹泉虎, 刘梅, 等. 基于改进Hybrid A*算法的阿克曼移动机器人路径规划[J]. 组合机床与自动化加工技术, 2023, (08): 122-126.
- [54] 桂鹏, 金英连, 廖根兴, 等. 基于改进TEB算法的阿克曼移动机器人[J]. 中国计量大学学报, 2024, 35(03): 432-442.
- [55] 景丽. 基于势场蚁群算法的智能车路径规划研究[D]. 兰州交通大学, 2023.
- [56] Li M, Yang Z, Hao L, et al. A hierarchical planning algorithm based on network topology analysis for minimal Steiner tree problems [C]//Proceedings of the 2013 International Conference on Advanced Mechatronic Systems. IEEE, 2013: 683-686.
- [57] Zobel J, Becker B, Kundel R, et al. Topology-aware path planning for in-transit coverage of aerial post-disaster communication assistance systems [C]//2020 IEEE 45th LCN Symposium on Emerging Topics in Networking (LCN Symposium). IEEE, 2020: 88-98.
- [58] Wang Z, Wang Q, Moran B, et al. Optimal submarine cable path planning and trunk-and-branch tree network topology design [J]. IEEE/ACM Transactions on Networking, 2020, 28 (4): 1562-1572.
- [59] 刘朋, 任工昌. 特征地图的室内机器人路径规划融合算法[J]. 计算机科学与探索, 2023, 17(11): 2755-2766.
- [60] 郝琨, 邓晁硕, 赵璐, 等. 基于区域搜索粒子群算法的机器人路径规划[J]. 电子测量与仪器学报, 2022, 36(12): 126-135.
- [61] 薛午阳, 马焕. 基于非一致拓扑度量地图的视觉导航方法[J]. 电脑与电信, 2024, (09): 41-49.
- [62] 刘峰, 郭阳, 郑辛, 等. 基于道路几何特征的地图匹配方法研究[J]. 导航定位与授时, 2020, 7(01): 67-72.

- [63]李永真,黄学功,张志安.基于融合蚁群-A*算法的多目标路径规划方法[J].计算技术与自动化,2024,43(04):66-72.
- [64]Harris B, Hunt R. TCP/IP security threats and attack methods[J]. Computer communications, 1999, 22(10): 885-897.
- [65]Xilong Q, Shengzong L, Sha F, et al. Design and implementation of wireless environment monitoring system based on STM32[J]. Scientific programming, 2021, 2021(1): 6070664.
- [66]Zhang J, Wang X, Vikash V, et al. ROS and ROS-mediated cellular signaling[J]. Oxidative medicine and cellular longevity, 2016, 2016(1): 4350965.
- [67]Cheung E C, Vousden K H. The role of ROS in tumour development and progression[J]. Nature Reviews Cancer, 2022, 22(5): 280-297.
- [68]Robot Operating System (ROS)[M]. Cham, Switzerland: Springer, 2017.
- [69]Macenski S, Moore T, Lu D V, et al. From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2[J]. Robotics and Autonomous Systems, 2023, 168: 104493.
- [70]于子珺,刁立强,曹镇宇,等.基于树莓派和ROS系统的机器人研究[J].物联网技术,2024,14(12):112-114+118.
- [71]邓洋,陈泽君,李科逸,等.基于ROS的全向轮自主移动底盘平台研制[J].工业控制计算机,2024,37(08):64-66+69.
- [72]冒建亮,曹豪杰,周昕,等.复合移动机器人自主导航系统实验平台设计[J].实验室研究与探索,2024,43(07):1-5+16.
- [73]王杰.基于 ROS 的智能车系统设计与路径规划研究[D].天津职业技术师范大学,2024.
- [74]Chi X, Li H, Fei J Y. Research on robot random obstacle avoidance method based on fusion of improved A* algorithm and dynamic window method[J]. Chinese journal of scientific instrument, 2021, 42(3): 132-140.

攻读硕士学位期间取得的成果

发表论文:

- [1] Wang Z, **Tang J**, Yi F, Ren X, Wang K (2025) Research on path planning of robotic arms based on DAPF- RRT algorithm. PLoS One 20(5): e0323734.
- [2] **Tang J**, Wang Z, Ren X. Improved ant colony algorithm path planning incorporating difference mechanism and dynamic adjustment strategies[C]//International Conference on Automation Control, Algorithm, and Intelligent Bionics (ACAIB 2024). SPIE, 2024, 13259: 765-776.
- [3]任祥瑞,王正刚,**汤俊杨**.改进 JPS 算法融合 DWA 的多机器人路径规划[J].计算机应用研究,2024,41(11):3251-3257.
- [4] Ren X, Wang Z, **Tang J**. Mobile robot path planning based on improved A* algorithm fused with JPS[C]//Proceedings of the 2024 3rd International Symposium on Control Engineering and Robotics. 2024: 478-485.

致谢

行文至此，硕士生涯即将落幕。在这段充满挑战与成长的旅程中，我收获了无数支持与帮助，感恩之情溢于言表，在此汇聚成这篇致谢。

我要衷心感谢我的导师王正刚老师。从论文选题到最终定稿，每个环节都离不开您的悉心指导。您深厚的学术造诣、严谨的治学态度，让我在研究中少走许多弯路。每当我陷入困境，您总是耐心答疑，用智慧为我指明方向。您不仅传授做学问的方法，更教会我为人处世之道，您的言传身教是我人生路上的宝贵财富。同时，也要感谢学院的授课老师们，你们的精彩课程为我打开智能车路径规划领域的大门，扎实了我的专业知识，为研究奠定基础。

还要感谢我的同学们。同门的我们共同探讨问题、分享想法，那些为难题争论又因找到方案而欢呼的日子，是硕士生活最珍贵的回忆。实验遇阻时，你们给予技术支持与精神鼓励。而室友们在生活中对我关怀备至，论文奋战的日夜，有你们默默陪伴。家人更是我坚实的后盾，求学路上，你们给予无条件的支持与鼓励，让我能心无旁骛追逐梦想。

最后，感谢参与论文评审和答辩的专家们，你们的宝贵意见助我完善论文，提升学术水平。感恩这段时光陪伴、帮助过我的每一个人，未来我将带着这份感恩，在智能车路径规划领域努力前行，不负大家的期望。