

分类号: TN249

密 级: 公开

学校代码: 10363

学 号: 2220110158



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题目 无人叉车自主定位与导航
的 3D 激光 SLAM 算法研究

论文作者 毛伟

校内导师 程军（副教授）

校外导师 李公文

专业学位类别 机械

研究方向（领域） 机器人导航

论文提交日期: 2025 年 5 月 20 日

分类号：TN249

学校代码：10363

密 级：公开

学 号：2220110158

无人叉车自主定位与导航的 3D 激光 SLAM 算法研究

RESEARCH ON AUTONOMOUS POSITIONING AND
NAVIGATION OF INTELLIGENT UNMANNED
FORKLIFTS USING 3D LASER SLAM

学生姓名：_____毛伟_____

校内导师：_____程军（副教授）_____

校外导师：_____李公文_____

专 业：_____机械_____

研究方向：_____机器人导航_____

论文答辩日期：2025 年 5 月 9 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： 年 月 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：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

无人叉车自主定位与导航的 3D 激光 SLAM 算法研究

摘 要

叉车式 AGV(Automated Guided Vehicle)是智能物流系统中执行货物堆垛和取放任务的关键设备,高精度的自主定位与导航技术是叉车式 AGV 实现这些功能的核心。由于工业环境的复杂多变性以及软硬件性能的高要求,叉车式 AGV 的进一步发展面临一定限制。因此,如何在通用平台下提升叉车式 AGV 的自主定位与导航精度和稳定性,成为智能物流领域的重要研究方向。本文通过以单舵轮驱动的堆垛式无人叉车为研究对象,针对单舵轮驱动的堆垛式无人叉车在运动过程中产生的定位误差以及现有激光 SLAM(Simultaneous Localization and Mapping)算法在复杂环境中出现的定位偏差问题,展开了相关研究,旨在提高其自主定位与导航的精度与稳定性。为此,提出了一种基于 EKF 的 DLO 信息融合定位建图算法,实现了单舵轮驱动的堆垛式无人叉车的精准定位与建图;在此基础上,进一步提出了基于 DLO-HDL 的信息融合定位建图算法,旨在提升单舵轮驱动的堆垛式无人叉车自主定位与导航的精度和稳定性。实验结果表明,该算法成功实现了单舵轮驱动的堆垛式无人叉车在不同工作环境下的高精度与高稳定性定位建图。本文的主要研究内容如下:

(1) 为实现单舵轮驱动的堆垛式无人叉车基于 3D 激光 SLAM 的同时定位与建图,本文建立了单舵轮驱动的堆垛式无人叉车的运动

模型和导航系统。通过推导激光雷达和 IMU 的测量模型，以及单舵轮驱动的堆垛式无人叉车的运动学模型，为所提出算法的误差机理分析提供了理论支持；此外，本文还推导了数据融合模型，为后续算法中的数据融合提供了具体方案；然后通过 MATLAB 进行了运动学仿真，验证了推导模型的准确性，并通过 MATLAB/SIMULINK 建立了数据融合系统仿真模型，验证了数据融合系统的有效性。最后，设计了单舵轮驱动的堆垛式无人叉车的导航系统，为实验验证搭建了实验平台。

(2) 针对单一激光雷达传感器在单舵轮驱动的堆垛式无人叉车定位建图中存在的误差较大、复杂动态环境下建图效果差以及传统图优化算法在处理 3D 激光 SLAM 时面临的高计算量和高存储需求等问题，本文提出了一种基于 EKF 的 DLO 信息融合定位建图算法。该方法通过扩展卡尔曼滤波（EKF）将 IMU（惯性测量单元）和 LiDAR（激光雷达）传感器的数据进行融合实现了 IMU 与 LiDAR 的前端紧耦合，构建了一个轻量化 SLAM 框架。实验结果表明该方法可以有效降低传统 3D SLAM 对计算机的算力需求，初步实现单舵轮驱动的堆垛式无人叉车的实时定位与建图。

(3) 为了进一步解决基于 EKF 的 DLO 信息融合定位建图算法这一前端融合算法存在的误差累积问题，在基于 EKF 的 DLO 信息融合定位建图算法的基础上引入后端优化机制，提出了一种基于 DLO-HDL 的信息融合定位建图算法。该方法将 HDL-Graph-SLAM 后端的

回环检测与位姿图优化模块嵌入前端融合系统，形成了“前端 EKF-DLO 融合+加后端图优化”的完整架构，从而实现了单舵轮驱动的堆垛式无人叉车的高精度、高稳定性定位与建图。通过对比实验验证，该方法使定位精度提升约 60.9%，同时稳定性也显著提升，能够满足单舵轮驱动的堆垛式无人叉车的高精度、高稳定性定位与建图要求。

关键词：堆垛式叉车AGV；环境感知；地图漂移；回环检测；同时定位与建图

RESEARCH ON AUTONOMOUS POSITIONING AND NAVIGATION OF INTELLIGENT UNMANNED FORKLIFTS USING 3D LASER SLAM

ABSTRACT

Forklift AGV (Automated Guided Vehicle) is a key piece of equipment in intelligent logistics systems, responsible for stacking and retrieving goods. High-precision autonomous localization and navigation technology are at the core of these functions. However, due to the complexity and variability of industrial environments, as well as the high performance requirements for both hardware and software, the further development of forklift AGVs faces certain limitations. Therefore, how to enhance the localization accuracy and navigation stability of forklift AGVs on a general platform has become an important research direction in the field of intelligent logistics. This paper takes a single-steering-wheel-driven stacker-type unmanned forklift as the research object and investigates the positioning errors generated during its movement, as well as the localization deviations of existing LiDAR SLAM (Simultaneous Localization and Mapping) algorithms in complex environments. The research aims to improve the accuracy and stability of autonomous localization and navigation. To achieve this, an EKF-based DLO information fusion localization and mapping algorithm is proposed, enabling high-precision localization and mapping of the single-steering-wheel-driven stacker-type unmanned forklift. Furthermore, a DLO-HDL-based information fusion localization and mapping algorithm is introduced

to further enhance the accuracy and stability of autonomous localization and navigation. Experimental results demonstrate that the proposed algorithm successfully achieves high-precision and high-stability localization and mapping of the single-steering-wheel-driven stacker-type unmanned forklift in various working environments. The main research contents of this paper are as follows:

(1) To achieve simultaneous localization and mapping (SLAM) based on 3D LiDAR for a single-steering-wheel-driven stacker-type unmanned forklift, this paper establishes the motion model and navigation system of the forklift. By deriving the measurement models of the LiDAR and IMU, as well as the kinematic model of the single-steering-wheel-driven stacker-type unmanned forklift, theoretical support is provided for the error mechanism analysis of the proposed algorithm. Additionally, a data fusion model is derived, offering a concrete framework for data fusion in the subsequent algorithm. MATLAB is then used to perform kinematic simulations, verifying the accuracy of the derived model. A data fusion system simulation model is further developed using MATLAB/SIMULINK to validate the effectiveness of the data fusion system. Finally, a navigation system for the single-steering-wheel-driven stacker-type unmanned forklift is designed, and an experimental platform is constructed for experimental validation.

(2) To address issues such as significant errors in positioning and mapping for single-steering-wheel-driven stacker unmanned forklifts using a single LiDAR sensor, poor mapping performance in complex dynamic environments, and the high computational and storage demands of traditional graph optimization algorithms when handling 3D LiDAR SLAM, this paper proposes a DLO (Distributed LiDAR Odometry) information fusion positioning and mapping algorithm based on EKF

(Extended Kalman Filter). This method achieves tight coupling between IMU (Inertial Measurement Unit) and LiDAR at the front end by fusing data from IMU and LiDAR sensors using EKF, constructing a lightweight SLAM framework. Experimental results demonstrate that this approach can effectively reduce the computational power requirements of traditional 3D SLAM on computers and preliminarily achieve real-time positioning and mapping for single-steering-wheel-driven stacker unmanned forklifts.

(3) To further address the issue of error accumulation in the front-end fusion algorithm based on the EKF-DLO (Extended Kalman Filter - Distributed LiDAR Odometry) information fusion positioning and mapping method, this study introduces a back-end optimization mechanism and proposes a DLO-HDL (Distributed LiDAR Odometry - Hierarchical Dynamic Loop Closure) information fusion positioning and mapping algorithm. This approach integrates the loop closure detection and pose graph optimization modules of HDL-Graph-SLAM into the front-end fusion system, forming a complete architecture of "front-end EKF-DLO fusion + back-end graph optimization." As a result, it achieves high-precision and highly stable positioning and mapping for single-steering-wheel-driven stacker unmanned forklifts. Comparative experiments demonstrate that this method improves positioning accuracy by approximately 60.9%, while also significantly enhancing stability. It effectively meets the requirements for high-precision and highly stable positioning and mapping in single-steering-wheel-driven stacker unmanned forklifts.

KEY WORDS: stacker-type forklift AGV, environmental perception, map drift, loop closure detection, Simultaneous Localization and Mapping

目录

摘 要.....	I
ABSTRACT.....	IV
目录.....	VII
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	4
1.2.1 叉车式 AGV 国内外研究现状.....	4
1.2.2 多传感器信息融合国内外研究现状.....	6
1.2.3 激光 SLAM 国内外研究现状.....	7
1.3 论文主要工作及章节安排.....	9
1.4 本章小结.....	11
第 2 章 激光 SLAM 导航原理及导航系统设计.....	12
2.1 引言.....	12
2.2 激光 SLAM 的基本原理.....	12
2.3 激光 SLAM 核心方法.....	14
2.3.1 基于滤波的 SLAM 方法.....	14
2.3.2 基于图优化的 SLAM 方法.....	18
2.3.3 基于扫描匹配的 SLAM 方法.....	19
2.4 导航系统设计.....	20
2.4.1 系统硬件结构设计.....	20
2.4.2 系统软件结构设计.....	22
2.5 本章小结.....	23
第 3 章 堆垛式无人叉车运动模型及仿真分析.....	24
3.1 引言.....	24
3.2 堆垛式无人叉车的坐标系统模型.....	24
3.3 堆垛式无人叉车的运动模型.....	26

3.3.1 激光雷达测量模型.....	26
3.3.2 IMU 测量模型.....	28
3.3.3 堆垛式无人叉车运动学模型.....	32
3.4 数据融合模型	35
3.5 堆垛式无人叉车运动学仿真分析	37
3.5.1 基于运动学模型的仿真分析.....	37
3.5.2 引入数据融合模型的运动学仿真分析.....	40
3.5.3 数据融合系统仿真分析.....	41
3.6 本章小结.....	45
第 4 章 基于 EKF 的 DLO 信息融合定位建图算法.....	46
4.1 引言	46
4.2 堆垛式无人叉车的定位误差机理分析	46
4.2.1 堆垛式无人叉车运动模型误差机理分析.....	46
4.2.2 堆垛式无人叉车传感器模型误差机理分析.....	48
4.3 基于 EKF 的 DLO 信息融合定位建图算法	49
4.3.1 激光雷达与 IMU 数据预处理及数据融合	49
4.3.2 基于 GICP 扫描匹配	52
4.3.3 先验估计.....	54
4.4 基于图优化的 Cartographer 定位建图算法	55
4.5 实验分析	57
4.5.1 地图构建及性能分析.....	58
4.5.2 轨迹误差分析.....	61
4.5.3 稳定性分析.....	65
4.6 本章小结.....	66
第 5 章 基于 DLO-HDL 的信息融合定位建图算法.....	67
5.1 引言	67
5.2 基于 EKF 的 DLO 信息融合定位建图误差机理分析	67
5.3 基于 DLO-HDL 的信息融合定位建图算法系统设计	68

5.3.1 系统框架.....	68
5.3.2 算法流程.....	69
5.4 后端位姿图优化及闭合回环检测	71
5.4.1 IMU 预积分	71
5.4.2 闭合回环检测.....	73
5.4.3 因子图优化.....	74
5.4.4 地图重建.....	75
5.5 实验分析	75
5.5.1 地图构建及性能分析.....	76
5.5.2 轨迹误差分析.....	79
5.5.3 稳定性分析.....	86
5.6 本章小结	87
第 6 章 总结与展望.....	88
6.1 论文总结	88
6.2 研究展望	89
参考文献.....	90
附录 1 直线运动学仿真 MATLAB 代码.....	96
附录 2 曲线运动学仿真 MATLAB 代码.....	97
附录 3 不同线速度及转向角运动学仿真 MATLAB 代码.....	98
附录 4 转向角随时间变化运动学仿真 MATLAB 代码.....	101
附录 5 理想状态下数据融合运动学仿真 MATLAB 代码.....	103
附录 6 噪声误差下数据融合运动学仿真 MATLAB 代码.....	105
附录 7 Simulink 仿真子模块	107
攻读硕士专业学位期间学术成果.....	109
致谢.....	110

第 1 章 绪论

1.1 研究背景及意义

随着《关于推动未来产业创新发展的实施意见》及《工业和信息化部等七部门关于加快推动制造业绿色化发展的指导意见》等国家战略的发布与实施，制造业正在向自动化和智能化方向迅速发展^[1]。新时代下，国家政策鼓励制造业向智能制造和工业互联网方向转型升级，通过引入先进的信息技术，如大数据、人工智能、物联网等，来提高制造业的智能化水平和生产效率，实现生产过程的数字化、网络化和智能化，促进关键核心技术的突破和应用^[2]。自动导引车(Automated Guided Vehicle, AGV)以其自主导航、运载能力强、灵活性高等特点近年来已经成为工业制造和物流管理中不可或缺的重要工具，被广泛应用于解决物流、制造和高风险劳动等各种工程问题^{[3][4]}。如图 1-1 所示，AGV 已被广泛应用于仓储、工业制造、医药、港口运输等领域^[5]。



(a) AGV 用于仓储



(b) AGV 用于工业制造



(c) AGV 用于医疗

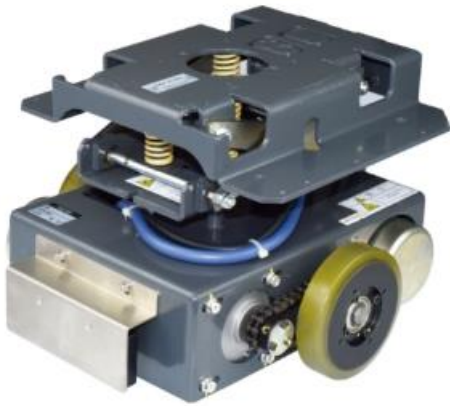


(d) AGV 用于港口运输

图 1-1 AGV 应用于不同领域

对于几十吨甚至上百吨的重型物料，普通的 AGV 难以满足其搬运需求，在

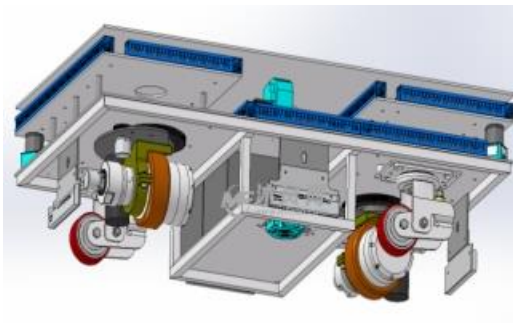
目前的国内外重工制造业领域，通常依赖于叉车、起重机、龙门吊等设备结合人工操作来处理重型材料的搬运工作，这种传统方式存在着生产效率低、安全风险高以及人工成本昂贵的问题^[6]。相比之下，叉车式 AGV 有着负载能力强，自动化程度高，灵活性等优势^[8]。在这一背景下，叉车式 AGV 作为实现智能物流的关键技术之一，其研究和应用显得尤为重要^[8]。



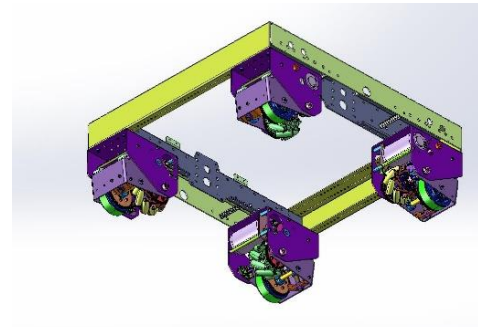
(a) 差速驱动 AGV



(b) 单舵轮式 AGV



(c) 双舵轮式 AGV



(d) 四舵轮式 AGV

图 1-2 不同驱动方式的 AGV

传统的 AGV 通常采用的差速驱动或者轮式驱动，受限于其驱动方式，其在狭窄空间或需要频繁转弯的场景下存在一定的局限性^[9]。而叉车式 AGV 通过采用全向舵轮驱动，使其能够往任意方向移动，具备较高的机动性和灵活性^[10]。采用舵轮驱动的 AGV 又可分为单舵轮式 AGV、两舵轮加两万向轮式 AGV 以及四舵轮式 AGV，如图 1-2 所示。四舵轮式移动更加灵活，但是其成本较高，每个轮子都需要独立的驱动和控制系统，因而其设计与调试会更加复杂^[11]。双舵轮加两万向轮的设计具有结构简单、稳定性高、维护成本低和适用性广等优点^[12]。然而，双舵轮的叉车式 AGV 在导航定位和运动控制上面临更大挑战，因为在承载大负

载的同时，还需要确保导航精度和运动的稳定性^[13]。相比之下，叉车式 AGV 通常采用单舵轮驱动，因其具有更高的灵活性、更小的转弯半径、结构更简化、成本更低，并便于实现精准的运动控制，能够在狭小空间内高效完成搬运任务。

当前，叉车式 AGV 广泛采用多种导航方式包括磁导航、激光导航、视觉导航和惯性导航等，如表 1-1 所示^{[14][15]}。相比之下，激光 SLAM 在室内环境中能够提供更精确的定位，因此经常被应用于工业领域^[19]。激光导航的定位精度主要受传感器性能的影响，而采用高精度激光雷达通常会增加成本。此外，当出现机器人绑架问题时，可能导致机器人失去定位，影响导航效果^[19]。

表 1-1 AGV 导航方法分类

导航方式	硬件设备	优点	缺点
磁导航	磁条、磁感应传感器	成本低、安全、定位精度较高	灵活性差、安装复杂、受外部磁场干扰
激光导航	激光雷达	定位精度高、灵活性强、适应性广	成本高、维护复杂
视觉导航	视觉相机	通用性强，成本较低、信息丰富	受环境影响大，定位精度不稳定
惯性导航	陀螺仪、加速度计	实时性强、环境适应性强	成本高、累积误差大、校准和算法复杂

实现高精度定位导航是叉车式 AGV 的核心研究课题，也是当前行业的研究热点^{[16]~[19]}。基于激光 SLAM 的叉车式 AGV 自主定位与导航研究，对提升叉车式 AGV 的自主性和智能性具有重要意义。首先，研究如何实现叉车式 AGV 的高精度、高稳定性导航技术，有助于提高生产效率和安全性。其次，叉车式 AGV 在承载大量货物时，缺乏精确的导航定位可能带来安全隐患。通过优化 SLAM 算法以提升定位精度，能有效减少运输过程中出现的问题，确保货物和设备的安全。

综上所述，叉车式 AGV 的自主定位与导航研究不仅是对现有 AGV 技术的延伸与提升，更是推动未来工业智能化发展的必然趋势。深入探索叉车式 AGV 自主定位与导航的理论与方法，将为实现工厂自动化、提升物流效率、降低生产成本提供重要支持，并助力工业领域迈向更高水平的智能化与高效化。

1.2 国内外研究现状

1.2.1 叉车式 AGV 国内外研究现状

AGV, 又称无人搬运车、自动装卸车或搬运机器人, 是一种配备电磁、光学或激光雷达等自动导引系统的运输设备, 能够沿设定路线行驶, 并具备安全防护及多种转向功能^[21]。叉车式 AGV 作为 AGV 的重要类型, 是无人化车间物料搬运的关键设备, 显著提升仓储效率, 已成为数字化车间的重要组成部分^[21]。目前, 叉车式 AGV 已广泛应用于物流运输、家电制造、汽车零部件等行业。随着应用场景的不断拓展, 其市场需求和数量持续增长, 吸引了众多智能科技企业加大研发投入。其中, 杭叉集团、衡阳合力、沈阳新松、昆船等知名企业在这一领域表现突出, 推动了叉车式 AGV 技术的不断进步和应用的深化。

国外 AGV 技术发展迅速。早在 1913 年, 美国福特公司便采用沿导轨运行的 AGV 运输汽车配件, 大幅缩短装配时间, 提高运输效率, 充分展现其独特优势^[22]。20 世纪 50 年代, 美国 Barret 电子公司研制出首台真正意义上的 AGV^[22]。该 AGV 由农业拖拉机改造而成, 沿地面导引线路运行, 将物料置于车斗中, 实现无人化运输, 开创了 AGV 技术先河^[23]。叉车的最早雏形由 CLARK 公司于 1917 年研发。1923 年, 耶鲁大学首次开展电动叉车研究, 并成功推出 K 型电动叉车^[24]。然而, 早期叉车缺乏自主移动功能, 仍需人工操作。直到 1970 年, NYK 力至优公司成功制造出首辆叉车式 AGV, 这一突破标志着叉车自主移动研究的正式开端^[25]。1988 年, Rodriguez F J 等人在早期叉车式 AGV 的基础上进行优化改进, 成功设计出一款基于视觉引导的叉车式 AGV, 使其导航能力进一步提升^[26]。2010 年, Teller S 和 Walter M R 等人基于丰田叉车开发新一代叉车式 AGV, 在环境感知与路径规划方面取得了更大的进步^[27]。进入 21 世纪, 叉车式 AGV 的研究取得显著突破, 并逐步向智能化、灵活化方向发展。英杰明公司设计的叉车式 AGV 不仅能够适应多场景、多姿态的货物取放操作, 还新增夹抱式功能, 实现对非托盘货物的精准灵活处理^[28]。Vecna Robotics 公司则研发了自主平衡式叉车式 AGV, 具备平移搬运与垂直堆垛功能, 同时具备更强的承载能力, 进一步提升了叉车式 AGV 的实用性和可靠性, 为智能物流发展提供了更高效的解决方案^[28]。



(a) Vecna Robotics 公司的叉车式 AGV

(b) 林德公司的叉车式 AGV

图 1-3 国外叉车式 AGV

国外知名叉车厂商包括德国的林德叉车（Linde）和日本的丰田叉车（TOYOTA）等。其中，林德叉车作为全球领先的叉车制造企业，凭借卓越的技术实力和产品质量，已为全球 100 多个国家提供高效的物流搬运解决方案。1993 年，林德叉车在中国厦门设立分公司，并建立研发基地，专注于平衡重叉车的设计与创新，不断提升产品性能^[28]。其先进的导航技术支持柔性路径配置，适应复杂多变的作业环境，同时具备维护便捷、定位精度高等优势，在中国市场占据重要地位，并深受行业认可^[22]。

国外在叉车式 AGV 领域的研究一直处于领先地位，至 20 世纪 80 年代末，其技术已趋于成熟。相比之下，我国起步较晚，长期依赖进口，最早引进的叉车来自前苏联，并于 1958 年成功仿制，从此开启了国产叉车制造的历程^[29]。1991 年，中科院沈阳自动化研究所与新松机器人自动化股份有限公司合作，研发出应用于汽车生产装配线的运输叉车式 AGV^[30]。2017 年，杭叉在传统叉车基础上融合 AGV 技术，成功推出国内首款叉车式 AGV，并已应用于华晨宝马的生产车间^[31]。随着叉车向智能化、自动化方向发展，越来越多传统叉车制造商开始布局智能叉车研发^[31]。沈阳新松、海康机器人、昆明昆船等专业厂商也在加大对叉车智能技术的投入，推动行业向更高水平迈进。



(a) 杭叉集团的叉车式 AGV



(b) 合力叉车的叉车式 AGV

图 1-4 国内叉车式 AGV

国内叉车行业的龙头企业包括杭叉集团和合力叉车。杭叉集团作为全球知名的叉车制造商，专注于叉车、物料搬运设备及牵引车的生产，在国内市场占有率达 30%^[32]。2018 年，杭叉集团基于激光导航技术成功研制堆垛式叉车 AGV，路径跟踪控制误差控制在 5mm 以内，并针对复杂工况额外配备机械臂^[32]。近年来，随着 5G 技术的快速发展，杭叉集团积极探索 5G 在叉车领域的应用，实现了多台 AGV 在工厂环境下的高效协同作业^[33]。合力叉车是国内规模领先、产业链完善、经济效益突出的叉车研发及出口供应商。2014 年，合力叉车在上海设立分公司，并成功开发采用 NDC 导航的叉车 AGV^[34]。2018 年，其叉车 AGV 智能运输线实现完整部署^[34]。2019 年，合力叉车将 5G 通信技术融入 AGV 系统，显著提升通信速率，进一步优化作业效率^[34]。

1.2.2 多传感器信息融合国内外研究现状

为了完成地图构建、路径规划和导航等任务，移动机器人首先需要获取更精准的环境数据。然而，单一传感器在环境信息采集方面存在一定局限，难以完全满足复杂场景的需求，因此，多传感器信息融合技术逐渐成为研究重点^[35]。该技术起源于 20 世纪 80 年代，融合了多个学科的信息处理方法，通过整合不同传感器对环境特征的感知，实现统一的信息表达与处理^[35]。目前，多传感器融合技术已广泛应用于军事战略、医疗健康、工业制造、智能物流及智慧生活等多个领域。

随着移动机器人技术的发展,多传感器融合技术的研究受到广泛关注。2016年,Google 实验室开源了 Cartographer 室内激光 SLAM 库,该库支持 ROS 环境,并兼容 2D 和 3D 激光雷达输入^[36]。Cartographer 算法融合激光雷达和 IMU,实现高精度定位与栅格地图构建^[36]。其中,2D-SLAM 依赖 2D 栅格地图,适用于机器人导航,而 3D-SLAM 采用 hybridGrid 技术构建混合概率地图,虽精度更高但使用较复杂。2017 年,Tong Qin 提出 VINS-MONO 算法,该算法结合 IMU 与相机数据,具有高度鲁棒的初始化、基于优化的紧耦合单目视觉惯性里程计(VIO)、在线重定位及 4 自由度(DOF)全局位姿图优化能力,并支持位姿图重用,提高系统的鲁棒性与可扩展性^[37]。2018 年,TiXiao Shan 基于 LOAM 优化提出 Lego-LOAM 算法,通过 LiDAR 与 IMU 数据融合,实现高精度定位^[38]。该算法采用 GTSAM 因子图优化库替代 CERES 优化库,以提升优化效率与精度^[39]。此外,Lego-LOAM 新增回环检测功能,分为回环帧检测和优化两个阶段,以增强全局一致性^[38]。2020 年,TiXiao Shan 提出 LIO-SAM 算法,在 Lego-LOAM 基础上引入 IMU 预积分因子与 GPS 因子,实现更紧密的多传感器融合^{[40][41]}。同年,浙江大学 Jiajun Lv 提出 LIDAR IMU CALIB 算法,采用 B 样条曲线拟合实现 IMU 与 LiDAR 精准标定^[42]。Weichen Dai 则针对动态环境中运动物体干扰,提出一种 SLAM 方法^[43]。该方法分析地图点相关性,将静态点与动态点分类,并运用 Delaunay 三角剖分算法去除运动物体干扰,确保定位与建图精度^[43]。2021 年,Tixiao Shan 推出 LVI-SAM 算法,相较 LIO-SAM 增加视觉里程计与视觉回环检测模块,即使激光雷达或视觉里程计失效,系统仍能稳定运行^[44]。同年,Shibo Zhao 提出 Super Odometry 算法,融合视觉、激光雷达与 IMU 数据,构建 IMU 里程计、视觉惯性里程计及激光惯性里程计三大模块,实现多源传感器的高效融合^[45]。

1.2.3 激光 SLAM 国内外研究现状

自主移动机器人面临的核心挑战是对外部环境的精准感知,而 SLAM 技术作为关键支撑,能够有效助力机器人在未知环境中的自主感知与导航^[29]。经过三十余年的发展,SLAM 技术已趋于成熟。近年来,随着无人驾驶、虚拟现实等领域的快速发展,该技术受到了广泛关注。众多科研团队致力于 SLAM 研究,期

望通过持续创新，为自主移动机器人的智能化发展提供新动力^[29]。

激光 SLAM 主要依赖激光雷达，通过稳定的点云数据实现位姿估计，并利用点云配准技术进行地图构建。相比于视觉 SLAM，其数据处理更简洁，在理论研究和实际应用中也更为成熟，具备较高的定位可靠性^[46]。根据激光雷达配置的不同，激光 SLAM 可分为 2D 和 3D 两种类型。2D 激光 SLAM 通常采用单线激光雷达，例如基于反光板定位的激光叉车 AGV；而 3D 激光 SLAM 依托多线激光雷达，包括 8 线、16 线、32 线和 64 线等，在无人驾驶领域得到广泛应用。此外，依据优化方法的不同，激光 SLAM 可分为基于粒子滤波和基于图优化两类，为技术发展提供了多种方案^[46]。

Smith R. C.和 Cheeseman P.最早提出 SLAM 问题，创新性地将机器人定位与地图构建相结合，并采用卡尔曼滤波进行实现，为后续 SLAM 技术的发展奠定了基础^[47]。Choi J 等人提出了一种基于 Rao-Blackwellized 粒子滤波器(RBPF)的混合 SLAM 方法，该方法结合特征地图与网格地图进行环境表示，并利用 RBPF 技术确保机器人在两种地图中的位姿精度，从而增强了 SLAM 系统的可靠性和环境描述的丰富性^[48]。Gmapping 算法（基于网格的快速 SLAM，Grid-based FastSLAM）由 Grisetti 等人首次提出，其核心思想是将机器人在环境中的运动建模为贝叶斯滤波问题，并通过递归算法不断优化机器人位姿和地图，实现同步定位与地图构建^[49]。安赫等人基于改进粒子群优化提出了增强版 Gmapping 算法，通过优化粒子分布，有效降低了陷入局部最优的风险，从而提升了全局搜索能力和定位精度^[50]。

基于滤波的 SLAM 方法已取得诸多研究进展，主要通过减少滤波和重采样次数来降低计算负担并提升运行效率。然而，随着地图规模的扩大，该方法的更新效率逐步降低，同时因缺乏回环检测，适应性较弱，尤其在大范围、多回环及长距离环境中问题更为突出。

Gutmann J. S.和 Konolige K.最早提出了基于图优化的 SLAM 算法，并将其划分为前端和后端两部分^[51]。其中，前端主要负责地图构建，而后端则用于优化机器人位姿及地图匹配。尽管这种架构提高了精度，但也带来了较高的计算量，影响了实际应用^[51]。郑川川等人提出了一种融合激光与惯导里程计的建图方法，

相较于传统图优化算法，该方法在定位精度和建图质量方面均有所提升^[52]。谷歌推出的 Cartographer 算法（Google Cartographer SLAM）作为基于图优化的 SLAM 代表性方案，创新性地引入了子地图概念，并采用分支定界方法加速 scan-to-submap 匹配约束计算，从而提升实时闭环检测能力^[36]。沈欣等人通过优化滤波效率，并结合速度积分改进位姿融合策略，增强了点云匹配的准确性，提高了建图精度并减少轨迹误差^[53]。丛佩超等人进一步优化了建图算法，通过降低二值优质地图间的匹配误差，使建图精度得到了进一步提升^[54]。

基于图优化的 SLAM 算法将建图划分为前端与后端两部分，以降低计算负担。谷歌推出的 Cartographer 算法结合多年图优化研究经验，通过引入子地图概念与分支定界方法，提高了计算效率，实现了实时闭环检测，从而确保了地图精度。

1.3 论文主要工作及章节安排

本文以叉车式 AGV 为研究对象，深入探讨了叉车式 AGV 在自主定位与导航过程中所面临的主要问题，提出了针对性的解决方案，并设计了具体实验进行验证。全文共分为六章，各章节的具体研究内容如下：

第 1 章 绪论。介绍了叉车式 AGV 自主定位与导航的研究背景与意义，阐述了课题的来源，并对叉车式 AGV、信息融合以及激光 SLAM 的研究现状进行了分析。同时，探讨了现阶段叉车式 AGV 自主定位与导航研究中存在的一些不足之处，为后续章节的安排奠定了基础。

第 2 章 激光 SLAM 导航原理及导航系统设计。主要阐述了激光 SLAM 导航的原理及方法，介绍了相关技术的背景与基本原理，包括 2D 激光 SLAM 技术、3D 激光 SLAM 技术。最后，设计了单舵轮驱动的堆垛式无人叉车的导航系统，详细阐述了系统硬件和软件结构，确保导航系统在复杂环境中能够稳定、高效地运行，为后续章节的研究提供了重要的理论基础。

第 3 章 单舵轮驱动的堆垛式无人叉车模型及仿真分析。本文对单舵轮驱动的堆垛式无人叉车进行了运动学分析，建立了运动学模型，并在 MATLAB 中进行了仿真分析。通过单舵轮驱动的堆垛式无人叉车的运动模型、激光雷达和 IMU 测量模型以及数据融合模型，为后续章节的研究奠定了系统的理论框架。

第 4 章 基于 EKF 的 DLO 信息融合定位建图算法研究。通过分析单舵轮驱动的堆垛式无人叉车的定位误差机理,提出了基于 EKF 的 DLO 信息融合定位算法。首先,分析了单舵轮驱动的堆垛式无人叉车运动模型和传感器模型中的误差来源,为算法设计提供了理论依据。接着,针对激光雷达和 IMU 数据进行预处理,并采用 EKF 方法将两者数据进行融合,利用融合后的数据作为 DLO SLAM 的输入,结合扫描匹配和先验估计进行定位建图,从而优化定位精度和建图效果。然后,介绍了基于图优化的 Cartographer 定位建图算法。最终,通过实验对比验证了所提出算法的有效性,并探索了其不足之处,为单舵轮驱动的堆垛式无人叉车在实际环境中的定位与建图提供了新的解决方案。

第 5 章 基于 DLO-HDL 的信息融合定位建图算法研究。首先分析了基于 EKF 的 DLO 信息融合定位建图的误差机理,并探讨了系统框架和算法流程的设计。接着,重点研究了后端位姿图优化与闭合回环检测,具体包括 IMU 预积分、位姿图优化、闭合回环检测和地图重建等技术,旨在提高定位精度与地图的构建质量。通过实验分析对所提出的算法进行了验证,并与其他方法进行了对比,评估其在实际场景中的效果。

第 6 章 总结及展望。对所进行研究的内容和取得的成果进行总结,思考尚未解决的问题和当前研究的不足之处,并提供一些未来研究方向的参考,以便后续研究更深入地探讨 3D 激光 SLAM 的智能无人叉车自主定位与导航问题。

本文的主体结构如图 1-5 所示。第三章首先构建了单舵轮驱动的堆垛式无人叉车的运动模型和导航系统。第四章基于第三章中的运动模型和导航系统,分析了单舵轮驱动的堆垛式无人叉车运动控制过程中定位误差的产生机理;随后,针对该误差机理,提出了一种基于 EKF 的 DLO 信息融合定位建图算法,以优化无人叉车的定位与建图性能;最后,通过与基于图优化的 Cartographer 定位建图算法进行对比实验,验证了所提出算法的有效性。第五章首先通过第四章中的实验和算法实现过程,分析了基于 EKF 的 DLO 信息融合定位建图算法的误差机理;然后,在此基础上,提出了一种基于 DLO-HDL 的信息融合定位建图算法,对第四章的算法进行了优化;最后,通过地图构建、性能分析、轨迹误差分析及稳定性分析等实验,验证了该优化算法的有效性。

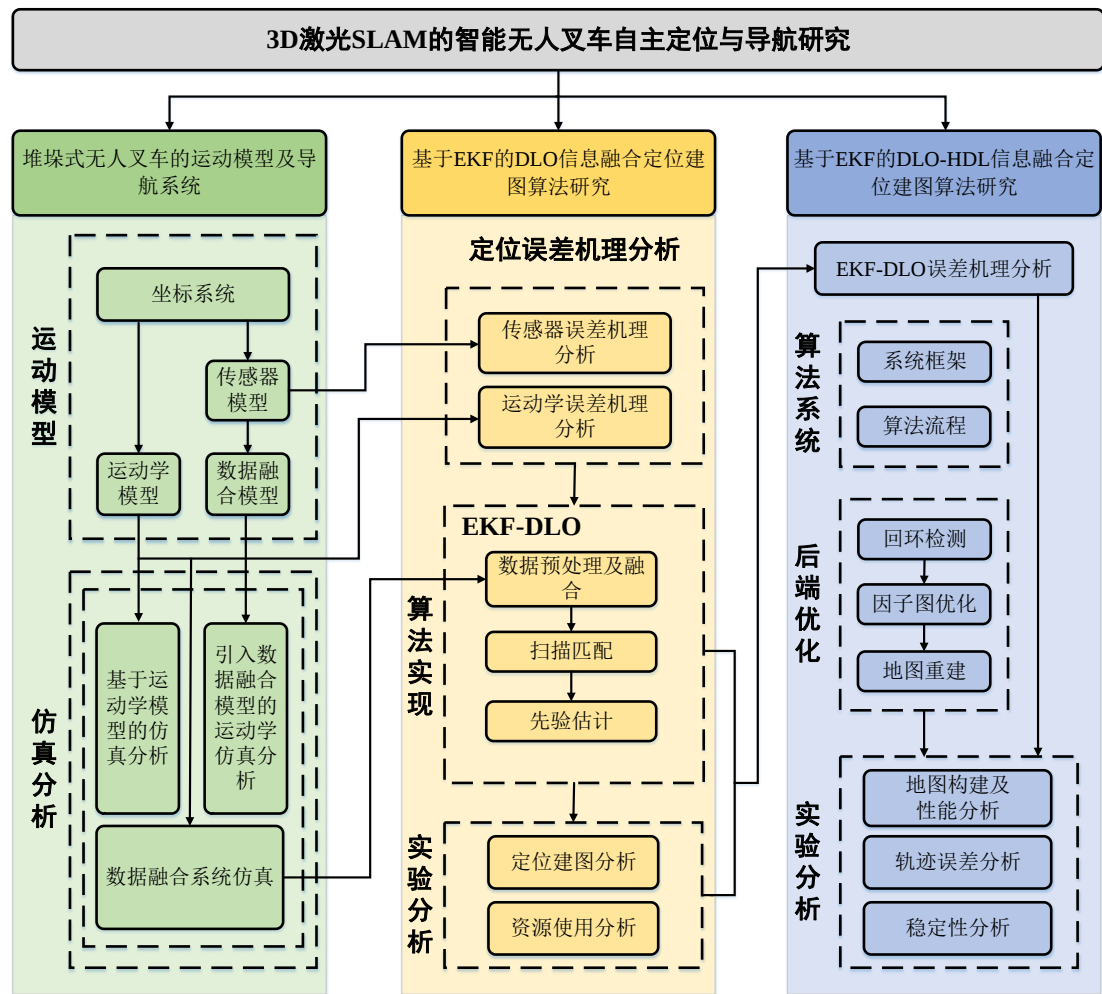


图 1-5 论文主体结构图

1.4 本章小结

本章系统地介绍了论文的研究背景、国内外研究现状、研究内容以及章节安排。首先，阐述了堆垛式无人叉车在自动化物流系统中的关键作用及其在实际应用中的需求，分析了提升叉车式 AGV 系统定位与导航精度所面临的技术挑战，并明确了本研究的意义和目标。接着，回顾了国内外在叉车式 AGV、多传感器信息融合和激光 SLAM 等领域的研究进展，深入探讨了相关技术的发展历程、已有成果及其局限性，指出了现有研究中的不足之处和潜在的改进空间。基于此，本章明确了本文的研究重点，提出了基于 EKF 的 DLO 信息融合定位建图算法，旨在实现单舵轮驱动的堆垛式无人叉车的精准定位与建图；在此基础上，进一步提出了基于 EKF 的 DLO-HDL 信息融合定位建图算法，旨在解决现有定位技术在精度和鲁棒性方面的挑战。最后，简要介绍了论文的结构安排，概述了各章节的研究内容与主要贡献，为后续的研究与算法实现提供了理论支持和指导方向。

第2章 激光 SLAM 导航原理及导航系统设计

2.1 引言

随着自动化技术的迅猛发展，激光 SLAM（Simultaneous Localization and Mapping）作为核心的环境感知与自主定位技术，在移动机器人、自动驾驶、无人机等领域得到了广泛应用，成为应对复杂环境下导航与定位挑战的重要解决方案。激光 SLAM 通过激光雷达获取环境点云数据，并利用 SLAM 算法实现机器人在未知环境中的实时定位与地图构建。由于激光雷达具备高精度、高分辨率以及较强的环境适应性，激光 SLAM 在复杂环境中的应用具有重要意义。随着多传感器融合技术的不断发展，激光 SLAM 逐渐与 IMU、视觉传感器等其他传感器结合，显著提升了系统的精度和鲁棒性。本章将重点介绍激光 SLAM 技术的基本原理与核心方法，并设计本文的导航系统，旨在为后续算法设计与系统实现提供坚实的理论基础和技术支持。

2.2 激光 SLAM 的基本原理

激光 SLAM 技术是一种基于激光雷达进行定位与建图的技术，广泛应用于室内导航、移动机器人、自动引导车（AGV）等领域。激光 SLAM 技术主要分为 2D 激光 SLAM 和 3D 激光 SLAM，2D 激光 SLAM 主要侧重于二维平面内的定位与建图，处理的数据为二维平面上的激光扫描数据，3D 激光 SLAM 主要侧重于三维空间内的定位与建图，其处理的数据为三维空间内的激光扫描数据^[55]。

激光 SLAM 是一种利用机器人搭载的 2D 或 3D 激光雷达，实时获取周围环境信息，从而实现自主定位与地图构建的技术。其技术组成包括激光雷达数据获取与转换、特征地图构建、定位和导航等模块。在这些组成部分中，激光雷达数据获取与转换以及特征地图构建是为定位任务做准备，而导航则依赖于定位结果，因此定位在导航系统中起着至关重要的作用^[55]。与单目、双目相机或深度相机等视觉传感器相比，激光雷达具有更强的抗光照变化能力、更高的测距精度、更强的环境适应性、更稳定的实时性能以及出色的三维建图能力^[56]。这些优点使激光雷达在复杂的动态环境中，或光照条件变化较大的场景下，能够高效完成定位与

建图任务。因此，激光雷达在自主移动机器人领域的应用日益广泛。本章将介绍激光 SLAM 的相关理论与方法。激光 SLAM 系统的技术组成与执行流程图如图 2-1 所示。

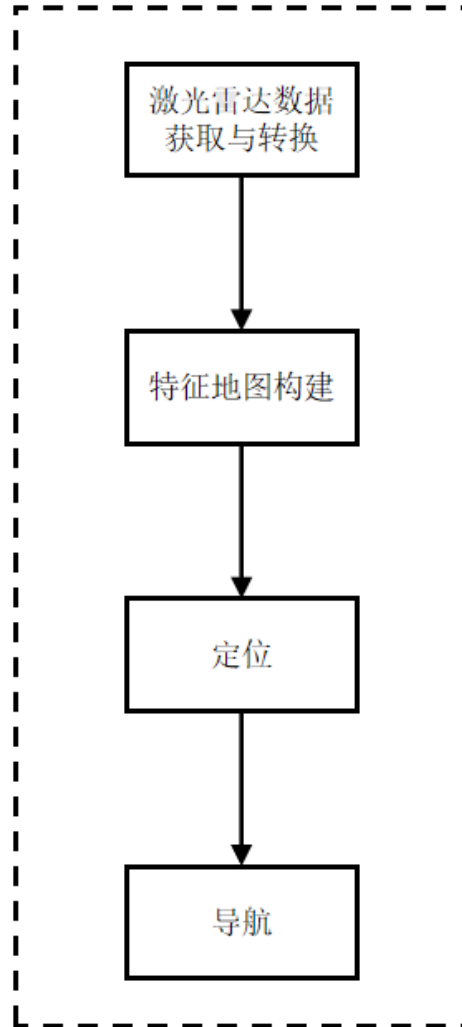


图 2-1 激光 SLAM 系统执行流程

激光 SLAM 的核心目标是通过激光传感器获取环境信息，并基于这些信息实现机器人定位（即确定机器人在地图中的位置）和环境建图（即构建环境的地图）。在此过程中，机器人通过激光扫描获取周围障碍物的距离信息，并对当前激光数据与历史数据进行匹配，以完成自我定位与地图更新^[57]。以下是激光 SLAM 的关键步骤：

1) 激光扫描与数据获取：激光雷达在每次扫描中发射激光束，激光束与环境中的物体相碰后反射回来，激光雷达多次扫描，激光雷达获取到一个二维的环境点云数据。

2) 机器人运动模型构建与定位：通过运动学模型（如里程计、速度控制器等），估算机器人在平面中的移动轨迹。每次扫描时，机器人会根据其估计的当前位置（即里程计数据）更新激光扫描数据的位置。

3) 地图构建与定位修正：激光雷达每次扫描得到的点云数据被用来生成当前的环境地图。SLAM 算法会根据这些数据不断更新已有地图，并修正机器人的位姿与历史扫描数据进行匹配，计算机器人当前的位置和旋转角度。

4) 闭环检测：如果机器人回到先前访问过的地方，SLAM 系统会通过闭环检测来识别并修正机器人位置中的误差。通常通过比对当前扫描数据与历史地图数据进行匹配，从而修正累计的定位误差。

5) 图优化：利用图优化算法融合所有扫描数据与位置信息，对机器人轨迹与地图进行优化。该过程通常依托非线性优化技术，以降低定位误差。

2.3 激光 SLAM 核心方法

2.3.1 基于滤波的 SLAM 方法

基于滤波的 SLAM 方法是一种利用滤波器来估计机器人状态和地图的技术。滤波方法在 SLAM 中通常用于估计机器人位置、姿态以及环境地图的状态，并且这些估计值是随着时间推移不断更新的。滤波算法通过处理传感器数据，来进行状态估计，并且能够处理来自不同传感器的噪声，典型的算法有扩展卡尔曼滤波（EKF-SLAM）和粒子滤波（Fast SLAM）^[58]。

1) 扩展卡尔曼滤波

扩展卡尔曼滤波（Extended Kalman Filter, EKF）是卡尔曼滤波的一种扩展形式，用于处理非线性系统中的状态估计问题^[59]。相比于基于线性模型的传统卡尔曼滤波，EKF 通过对非线性系统进行线性化，使其能够适用于更复杂的动态环境^[59]。扩展卡尔曼滤波的核心思想是将非线性模型通过泰勒展开线性化，从而能够应用卡尔曼滤波的框架进行状态估计^[59]。EKF 的执行流程包括预测和更新两个阶段，即根据前一最优状态预测当前值，并结合观测数据进行修正，以获得更准确的估计结果。以机器人匀加速直线运动为例（如图 2-2 所示）。

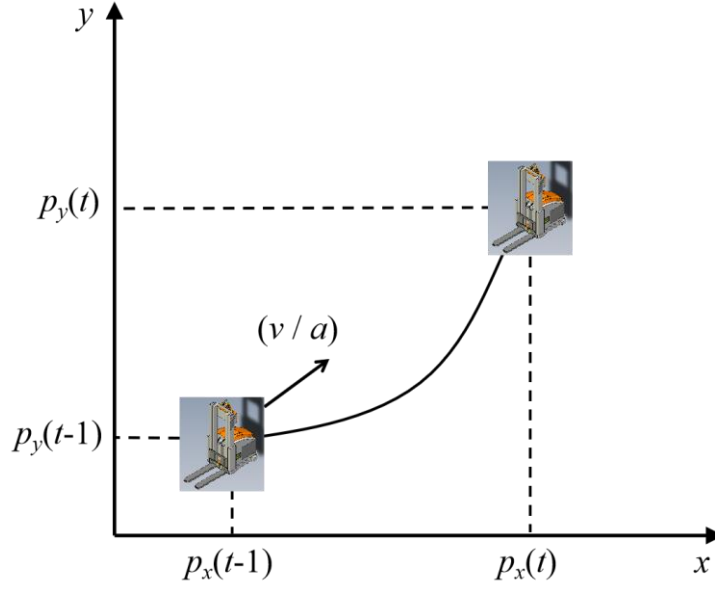


图 2-2 扩展卡尔曼滤波实例模型

图 2-2 中的小车 t 时刻的状态为 $\mathbf{x}_t$:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix} \quad (2-1)$$

根据小车的运动模型得到预测模型:

$$\begin{bmatrix} \mathbf{p}_t \\ \mathbf{v}_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \mathbf{a}_{t-1} \quad (2-2)$$

即预测模型的先验估计见式 2-3 所示。

$$\bar{\mathbf{x}}_t = \mathbf{F} \bar{\mathbf{x}}_{t-1} + \mathbf{B} \mathbf{u}_{t-1} \quad (2-3)$$

其中, $\bar{\mathbf{x}}_t$ 是第 t 时刻的预测状态; $\mathbf{F}$ 是状态转移矩阵; $\bar{\mathbf{x}}_{t-1}$ 是上一时刻的状态估计; $\mathbf{B}$ 是控制矩阵; $\mathbf{u}_{t-1}$ 是控制输入。

预测模型的先验估计的协方差矩阵见式 2-4 所示。

$$\begin{aligned} \mathbf{P}_t^- &= \text{cov}(\bar{\mathbf{x}}_t, \bar{\mathbf{x}}_t) \\ &= \text{cov}(\mathbf{F} \bar{\mathbf{x}}_{t-1} + \mathbf{B} \mathbf{u}_{t-1}, \mathbf{F} \bar{\mathbf{x}}_{t-1} + \mathbf{B} \mathbf{u}_{t-1}) \\ &= \mathbf{F} \text{cov}(\bar{\mathbf{x}}_{t-1}, \bar{\mathbf{x}}_{t-1}) \mathbf{F}^T + \text{cov}(\mathbf{B} \mathbf{u}_{t-1}, \mathbf{B} \mathbf{u}_{t-1}) \\ &= \mathbf{F} \mathbf{P}_{t-1}^- \mathbf{F}^T + \mathbf{Q} \end{aligned} \quad (2-4)$$

其中, $\mathbf{P}_t^-$ 是第 t 时刻预测的协方差矩阵; $\mathbf{P}_{t-1}^-$ 是上一时刻预测的协方差矩阵; $\mathbf{Q}$ 是过程噪声协方差矩阵。

在更新步骤中，通过测量值来更新预测的状态估计。测量值通过观测模型与预测状态进行比较，并更新状态估计。

小车的测量状态为 $\mathbf{z}_t$:

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{z}_p \\ \mathbf{z}_v \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_t \\ \mathbf{v}_t \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{p}_t \\ \Delta \mathbf{v}_t \end{bmatrix} \quad (2-5)$$

即测量方程见式 2-6 所示。

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v} \quad (2-6)$$

更新过程具体见式 2-7, 2-8, 2-9 所示。

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_t^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (2-7)$$

其中， $\mathbf{K}_t$ 是 t 时刻的卡尔曼增益， $\mathbf{H}$ 是观测模型的雅可比矩阵， $\mathbf{R}$ 是观测噪声协方差矩阵。

$$\mathbf{x}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H} \hat{\mathbf{x}}_t^-) \quad (2-8)$$

其中， $\hat{\mathbf{x}}_t$ 是更新后的状态估计， $\mathbf{z}_t$ 是第 t 时刻的测量值。

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^- \quad (2-9)$$

其中， $\mathbf{P}_t$ 是第 t 时刻更新后的协方差矩阵， $\mathbf{I}$ 是单位矩阵。

扩展卡尔曼滤波 (EKF) 是一种高效的滤波方法，在非线形系统状态估计中具有重要应用价值，它通过对非线性模型进行一阶线性化处理，能够在保持较低计算复杂度的情况下，进行实时的状态估计^[60]。尽管 EKF 在一些情况下存在精度下降和计算瓶颈等问题，但它仍然是许多实际应用中的首选方法，尤其是在实时处理和小规模系统中。

2) 粒子滤波

粒子滤波，也叫蒙特卡洛滤波 (Monte Carlo Filter)，是一种基于贝叶斯理论的递归估计方法，与卡尔曼滤波不同，粒子滤波适用于更复杂、更高维度的非线性、非高斯系统^[58]。粒子滤波利用蒙特卡洛方法，通过在状态空间中生成多个粒子并对其进行加权，来进行状态估计。粒子滤波通过利用一组粒子（即样本）来逼近目标分布的状态；在每一时刻，粒子代表着系统状态的一个假设，每个粒子都有一个与之相关的权重，表示其“可信度”；随着时间的推移，这些粒子会根据

系统的运动模型和观测模型不断更新^[61]。其基本思想是通过生成一组粒子，近似表示系统的概率分布，然后利用这些粒子来进行状态估计，每个粒子的权重反映了它与当前观测数据的匹配程度^[62]。粒子滤波的工作流程主要包括：初始化、预测、更新和重采样。

粒子滤波首先通过生成一组粒子来初始化系统状态。每个粒子对应于状态空间中的一个可能状态，这些粒子的初始位置通常是基于先验分布或假设的，初始粒子集为：

$$\{\mathbf{x}_0^{(i)}\}_{i=1}^N \quad (2-10)$$

其中， N 是粒子数量， $\mathbf{x}_0^{(i)}$ 是第 i 个粒子状态。

在预测步骤中，粒子根据系统的动态模型进行状态更新。对于每个粒子，使用控制输入和系统的运动模型来预测下一时刻的状态。

$$\mathbf{x}_t^{(i)} = f(\mathbf{x}_{t-1}^{(i)}, \mathbf{u}_t) + \boldsymbol{\omega}_t^{(i)} \quad (2-11)$$

其中， $f(\cdot)$ 是系统的运动模型， $\mathbf{u}_t$ 是控制输入， $\boldsymbol{\omega}_t^{(i)}$ 是过程噪声。

在更新步骤中，粒子的权重根据测量数据和观测模型进行调整。权重反映了粒子与当前观测数据的匹配程度。通常，权重由观测模型和粒子预测的状态给出。

$$\varpi_t^{(i)} = p(\mathbf{z}_t | \mathbf{x}_t^{(i)}) \quad (2-12)$$

其中， $\varpi_t^{(i)}$ 是 t 时刻第 i 个粒子的权重， $\mathbf{z}_t$ 是 t 时刻的测量值， $p(\mathbf{z}_t | \mathbf{x}_t^{(i)})$ 是观测模型，表示粒子 $\mathbf{x}_t^{(i)}$ 给定观测值 $\mathbf{z}_t$ 的概率。

粒子滤波的关键问题之一是权重的“退化”现象：即很多粒子的权重会变得非常小，这意味着大部分粒子对最终估计的贡献微乎其微，为了避免这种情况，粒子滤波使用重采样步骤来重新选择粒子^[63]。通过对粒子进行重采样，使得高权重的粒子被选中的概率较大，而低权重的粒子被丢弃^[63]。

粒子滤波是一种强大的递归估计算法，适用于高维、非线性和非高斯系统^[64]。它通过生成一组粒子并根据观测更新粒子的权重，能够灵活处理各种复杂系统。尽管粒子滤波在计算效率和重采样误差方面存在一定的挑战，但它的适用性和灵活性使得它在机器人、目标追踪、SLAM 等领域得到了广泛的应用。

2.3.2 基于图优化的 SLAM 方法

基于图优化的 SLAM 方法是一种常用的 SLAM 算法框架，广泛应用于机器人、自动驾驶、无人机等领域。其基本思想是将 SLAM 问题转化为一个图优化问题，通过构建图模型来表示机器人位置与环境特征的约束关系，并通过优化算法求解最佳的状态估计^[65]。

图优化 SLAM 的基本思想是将 SLAM 中的状态估计问题表示为图中的节点和边，节点表示机器人或特征的状态（如位姿、地图特征位置等），边表示机器人之间或者机器人与环境之间的约束（如位移约束、观测约束等）^[66]。通过对图进行优化，求解每个节点的最佳状态，从而得到最优的轨迹估计和地图构建。具体来说，图优化方法将 SLAM 问题描述为一个最小化问题，目标是 minimized 机器人位姿和环境特征之间的误差。

图优化的核心是构建一个包含多个节点和约束的图。这个图的基本构成有节点、边和误差模型。节点包括机器人位姿节点和地图特征节点。机器人位姿节点通常表示机器人在每个时刻的位置和姿态 (x, y, θ) ，也可能包含更高维度的信息（如速度、加速度等）；地图特征节点通常表示地图中的特征（地标或特征点），通常包括特征的空间位置。边包括运动约束边和观测约束边。运动约束边表示机器人连续两次位姿之间的关系，通常由里程计、惯性测量单元（IMU）或视觉里程计等传感器提供，这些约束通常来源于机器人在两次位姿之间的相对运动；观测约束边表示机器人与环境中地标或特征之间的观测关系，通常来自传感器（如激光雷达、相机等），这类约束描述了机器人观测到的地图特征的相对位置。每一条边都有一个与之相关的误差模型。误差模型描述了约束的“不确定性”，即实际测量和估计值之间的差异。图优化的优化目标是通过调整节点，使边的约束误差最小化，即求解一个最优的状态估计，通常是最小化一个误差函数^[67]。这个误差函数表示了所有节点和约束之间的“拟合”误差。假设我们有 n 个节点，误差函数可以表示为：

$$E(\{\mathbf{x}_i\}, \{\mathbf{z}_{ij}\}) = \sum_i \sum_j \left\| \mathbf{h}(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{z}_{ij} \right\|_{\Sigma_{ij}}^2 \quad (2-13)$$

其中， $\{\mathbf{x}_i\}$ 表示所有节点的状态（机器人位姿、地图特征位置等）， $\mathbf{h}(\mathbf{x}_i, \mathbf{x}_j)$ 是

观测模型，表示在状态 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 下的预测值， $\mathbf{z}_{ij}$ 表示第 i 个节点和第 j 个节点之间的观测， Σ_{ij} 是约束的协方差矩阵。

图优化问题通常通过求解最小二乘问题来进行优化，最常见的优化算法是高斯-牛顿法和 Levenberg-Marquardt 法。这些方法通过迭代更新节点的状态，直到误差函数收敛^[68]。

基于图优化的 SLAM 方法通过将 SLAM 问题转化为图优化问题，能够提供精确的状态估计和高质量的地图构建。它能够通过全局优化方法来减少机器人定位中的累积误差，并能有效处理复杂的非线性约束。尽管图优化方法计算复杂度较高，但通过优化算法和数据结构的改进，它在实际应用中已经取得了广泛的成功。

2.3.3 基于扫描匹配的 SLAM 方法

基于扫描匹配的 SLAM (Scan Matching SLAM) 方法是一种常见且重要的 SLAM 技术。它主要依赖于从传感器（如激光雷达）获取的环境扫描数据，并通过比对不同时间点的扫描数据来估计机器人或车辆的运动轨迹，同时构建环境地图^[69]。扫描匹配的核心思想是通过对比连续两次激光扫描数据（或点云数据），计算机器人在这段时间内的相对位姿变化（包括平移和旋转）。主要包括：激光扫描数据获取、扫描数据配准、姿态估计、地图更新^[69]。激光扫描数据获取通常使用激光雷达 (LiDAR) 等传感器来扫描环境，得到一组在二维或三维空间中的点云数据。每个扫描点表示了传感器到物体表面的距离，以及其相应的角度（二维激光扫描）或空间位置（在三维激光扫描中）^[69]。扫描匹配就是将当前时刻的激光扫描数据与上一时刻的扫描数据进行比对，寻找最合适的变换（通常是平移和旋转），从而确定机器人在这段时间内的相对位置变化；其次通过扫描匹配计算出的相对变换（平移和旋转）可以用来更新机器人在世界坐标系中的位姿（位置和方向）；这样，机器人就可以不断地推算出当前位置，同时避免了累积误差；最后在估计出当前位姿后，扫描数据被用于更新地图，地图通常包括栅格地图，特征地图和点云地图^[70]。

2.4 导航系统设计

2.4.1 系统硬件结构设计

本文实验研究所用的无人叉车平台采用了实验室自主研发的单舵轮驱动的堆垛式无人叉车，该平台配备了研华 EI-52 工控机作为上位机，STM32 嵌入式控制板作为下位机，并搭载了速腾 RS-LiDAR-16 激光雷达和维特 HWT905 九轴 IMU 等传感器。传感器与主控板之间通过 USB 接口进行通信，而上、下位机之间则通过 UDP 协议进行数据传输，以实现信息交换。单舵轮驱动的堆垛式无人叉车硬件实物图及硬件模型图如图 2-3 所示。

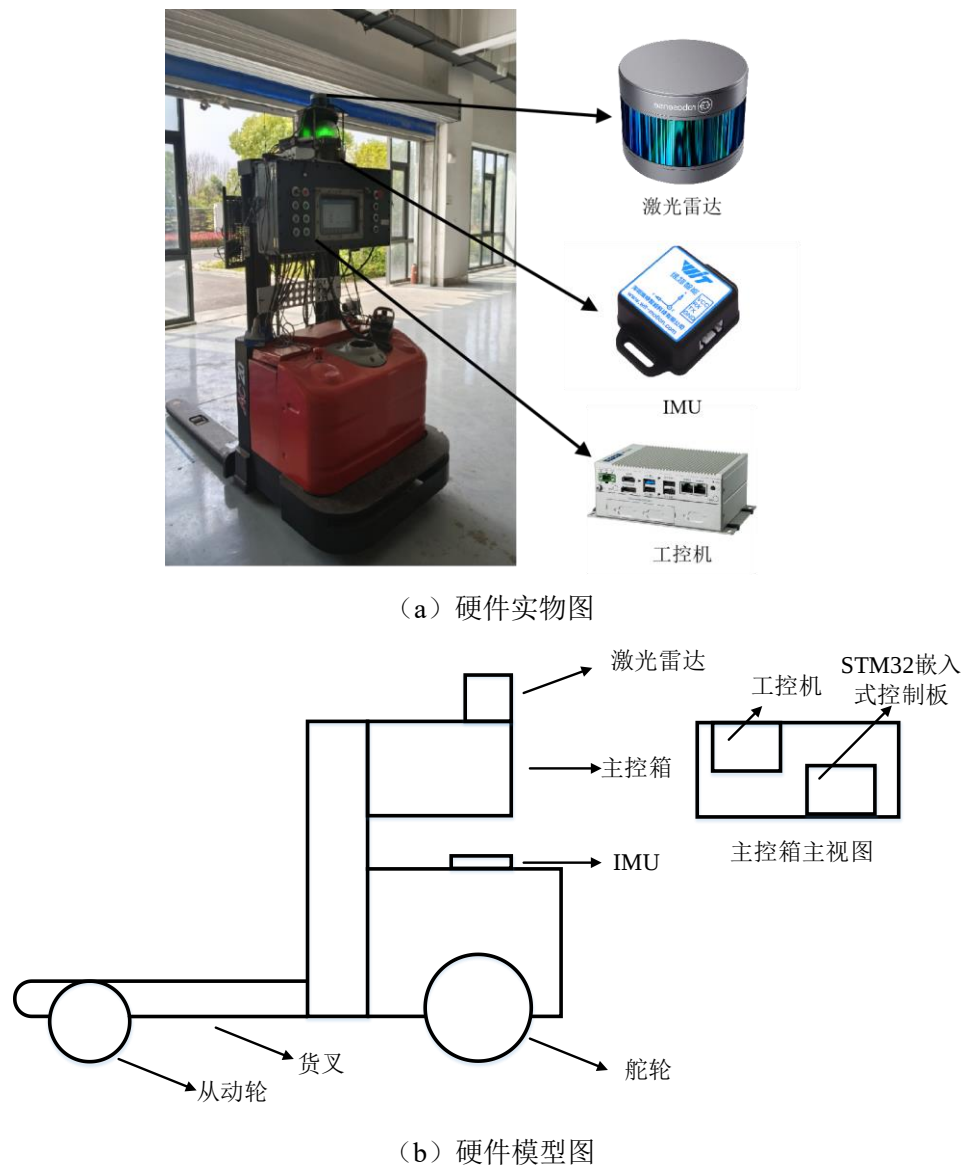


图 2-3 实验硬件图

激光雷达是移动机器人重要的组成部分之一，同时也是应用较为广泛的传感

器，主要用于环境感知与地图构建。本文采用的激光雷达是速腾 RS-LiDAR-16，其测量范围为 40 cm 至 150 m，扫描角度为 360°，分辨率为 0.4°，最高采样频率可达到 10 Hz。其技术参数详见表 2-1 所示。

表 2-1 速腾 RS-LiDAR-16 激光雷达技术参数

项目	最小值	典型值	最大值	单位
测距	0.4	0.4~150	150	m
扫描范围	0	0~360	360	°
视角	-15	-15~15	15	°
分辨率	5	5~20	20	Hz
精度	\	±2	\	cm
转速	5	5~20	20	Hz
采样频率	1	5.5	10	Hz
波长	\	905	\	nm

本文采用的惯性测量单元 IMU 是维特的 HWT905 九轴 IMU，其姿态测量精度为 0.05°，通信距离可达 300 m，最高输出频率为 200 Hz。其技术参数详见表 2-2 所示。

表 2-2 维特 HWT905 九轴 IMU 技术参数

项目	最小值	典型值	最大值	单位
加速度计量程	\	±8	\	g
加速度计分辨率	\	0.001	\	g
加速度计精度	\	±0.03	\	g
陀螺仪量程	\	±500	\	° / s
陀螺仪分辨率	\	0.01	\	° / s
陀螺仪计精度	\	±0.02	\	° / s
磁力计量程	\	±500	\	μT
磁力计分辨率	\	0.1	\	μT
磁力计精度	\	±0.5	\	μT

2.4.2 系统软件结构设计

软件平台是机器人系统的核心，负责控制与管理，涵盖传感器数据处理、控制算法执行、地图构建及导航等功能^[70]。同时它还作为机器人与外部设备及系统通信的桥梁。

机器人操作系统（Robot Operating System, ROS）是一种专为机器人应用设计的软件平台，能够高效连接机器人与系统，并精准执行指令以完成任务^[71]。ROS 提供丰富的软件库和工具，支持程序的移植、编译及跨平台运行。其分布式架构将系统组件拆分为独立进程，并由 ROS master 协调进程间通信，通信方式包括同步、异步以及服务器数据存储^[71]。

本文采用的机器人操作系统是基于 Ubuntu 18.04 的 ROS Melodic，根据实现的不同功能将其划分为多个模块。本文将分别对各个模块进行研究，并最终将它们整合，以实现预定目标功能。本文设计的软件系统如图 2-4 所示。

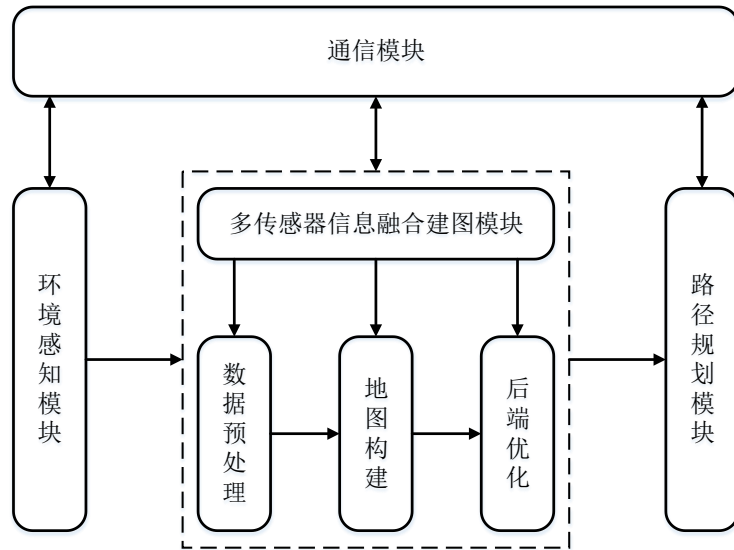


图 2-4 软件系统方案

软件系统主要包括以下几个模块：

- （1）环境感知模块：利用激光雷达传感器和 IMU 传感器收集环境数据。
- （2）多传感器信息融合建图模块：对传感器数据进行预处理，并建立坐标转换关系，将处理后的数据进行融合以构建环境地图，最后对环境地图进行后端优化。
- （3）路径规划模块：基于多传感器信息融合构建的环境地图以及无人叉车的当前位姿，系统能够进行全局路径规划，并计算出从起点到终点的最佳路径。

(4) 通信模块：通过配置无人叉车与上位机的参数，实现两者之间的数据交互。在 ROS 操作系统下，通过发布和订阅话题，不同的节点协同工作，完成数据交换和监测功能。

2.5 本章小结

本章全面介绍了激光 SLAM 技术，并设计了本文堆垛式无人叉车的导航系统。首先，阐述了激光 SLAM 的基本原理，包括激光雷达的工作原理、数据处理流程以及 SLAM 中的定位与建图任务。接着，详细讲解了滤波算法、图优化算法和扫描匹配等核心方法。最后，设计了单舵轮驱动的堆垛式无人叉车的导航系统，深入分析了系统的硬件和软件结构，确保导航系统在复杂环境中能够稳定、高效地运行。本章为激光 SLAM 与多传感器信息融合的研究提供了坚实的理论支持，并为实际导航系统的设计与优化提供了宝贵的指导。

第3章 堆垛式无人叉车运动模型及仿真分析

3.1 引言

堆垛式无人叉车作为自动化物流系统中的重要组成部分，广泛应用于仓储和物流等领域。为了使无人叉车在复杂环境中高效完成任务，精确的运动建模至关重要。本章首先介绍了单舵轮驱动的堆垛式无人叉车的坐标系统模型，明确了定位与导航的参考框架。接着，深入探讨了单舵轮驱动的堆垛式无人叉车的运动模型，包括激光雷达、IMU（惯性测量单元）测量模型和运动学模型，为后续的导航与路径规划提供了理论支持。结合多传感器信息融合技术，本文构建了一个多传感器数据融合模型，有效提升了定位精度和系统的鲁棒性。最后，在 MATLAB 中进行运动学仿真分析以验证模型的正确性。

3.2 堆垛式无人叉车的坐标系统模型

坐标系统用于描述机器人及其末端执行器的位置、姿态，以及它们之间的相对位置和方向。建立不同的坐标系是机器人运动学中的关键步骤，这些坐标系能够描述机器人各个部件的状态，并通过坐标系之间的相对转换，计算机器人的运动学特性，如末端执行器的空间状态^[71]。常见的坐标系包括全局坐标系、局部坐标系和传感器坐标系。

本文中的全局坐标系表示的是单舵轮驱动的堆垛式无人叉车在实验环境中的位姿，记作 $O-XYZ$ (W 系)；局部坐标系包括无人叉车坐标系、激光雷达坐标系、IMU 坐标系；无人叉车坐标系以无人叉车本体的中心为基准，记作 $o-xyz$ (R 系)，无人叉车的位姿用 $(x, y, z, \phi, \theta, \varphi)$ 表示，其中 (x, y, z) 是无人叉车在三维空间中的位置坐标， ϕ 是翻滚角，表示无人叉车绕 X 轴的旋转角度， θ 是俯仰角，表示无人叉车绕 Y 轴的旋转角度， φ 是偏航角，表示无人叉车绕 Z 轴的旋转角度；为了方便坐标转换，激光雷达坐标系的 X 轴、 Y 轴和 Z 轴方向与无人叉车坐标系保持一致， X 轴方向上存在 1.5 个偏移量， Z 轴方向上存在一个偏移量，记作 $o_1-x_1y_1z_1$ (L 系)；IMU 坐标系的 X 轴、 Y 轴和 Z 轴方向与无人叉车坐标系也保

持一致，但 X 轴上存在 1.5 个偏移量，记作 $o_2-x_2y_2z_2$ (B 系)。如图 3-1 所示。

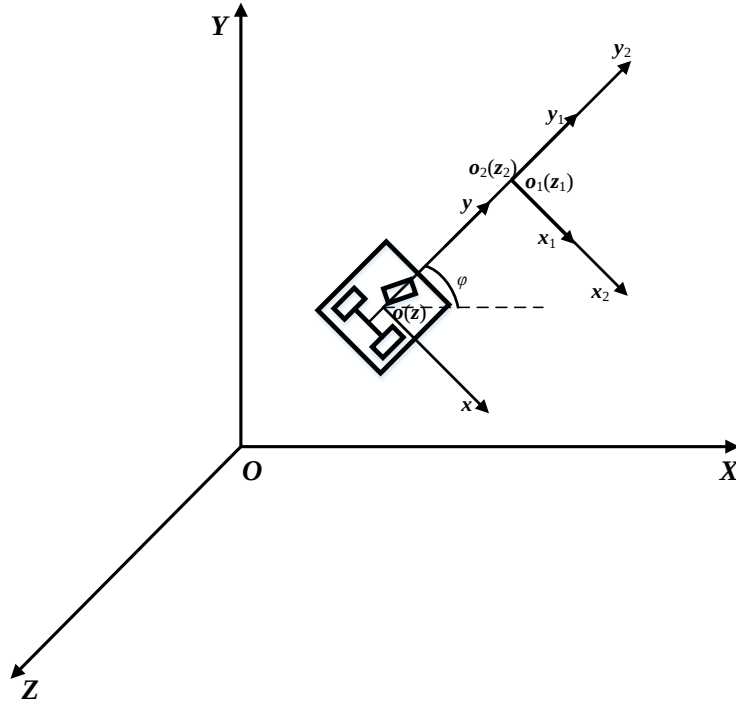


图 3-1 空间坐标系表示

在某一时刻，无人叉车坐标系相对于世界坐标系的平移和旋转变换可以通过相应的旋转矩阵表示，如式 3-1、3-2、3-3 和 3-4 所示。

绕 X 轴的旋转矩阵 $R_x(\phi)$:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (3-1)$$

绕 Y 轴的旋转矩阵 $R_y(\theta)$:

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3-2)$$

绕 Z 轴的旋转矩阵 $R_z(\varphi)$:

$$R_z(\varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-3)$$

旋转顺序通常为 Z - Y - X ，即先绕 Z 轴旋转（偏航角 φ ），再绕 Y 轴旋转（俯

仰角 θ), 最后绕 X 轴旋转 (翻滚角 ϕ)。组合旋转矩阵 $\mathbf{R}$ 为将这三个旋转矩阵相乘, 即可得到总的旋转矩阵 $\mathbf{R}$:

$$\mathbf{R} = \mathbf{R}_z(\varphi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi)$$

$$= \begin{bmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi \\ \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (3-4)$$

世界坐标系到无人叉车本体坐标系的转换只需要对总的旋转矩阵 $\mathbf{R}$ 求逆矩阵即可, 见式 3-5 所示。

$$\mathbf{R}^{-1} = \mathbf{R}^T$$

$$= \begin{bmatrix} \cos \varphi \cos \theta & \sin \varphi \cos \theta & -\sin \theta \\ \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \cos \theta \sin \phi \\ \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (3-5)$$

无人叉车的齐次变换矩阵 $\mathbf{T}$ 是将旋转矩阵和平移向量 $\mathbf{t}_{WR}(x, y, z)$ 结合在一起如式 3-6 所示。

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t}_{WR} \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi & x \\ \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi & y \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-6)$$

3.3 堆垛式无人叉车的运动模型

单舵轮驱动的堆垛式无人叉车的运动学模型在环境建图中起着至关重要的作用。只有深入了解其运动学特性, 才能准确确定无人叉车在实验环境中的位置, 从而生成可靠的环境地图。因此, 建立堆垛式无人叉车的运动学模型是实现环境建图和自主导航的关键步骤。

3.3.1 激光雷达测量模型

3D 激光雷达 (Light Detection and Ranging, LiDAR) 是一种通过发射激光束并接收其反射波来测量目标物体距离的技术, 激光雷达通过发射激光脉冲并接收其反射波, 从而测量与物体的距离, 最终生成三维点云数据集^[72]。与毫米波雷达相比, 3D 激光雷达具有更高的激光发射频率和更短的波长, 能够提供远高于毫

米波雷达的分辨率和探测距离，同时也能更清晰地捕捉目标物体的具体轮廓^[72]。

本文以速腾 RS-LiDAR-16 激光雷达进行分析，以激光雷达中心为坐标原点建立激光雷达坐标系，激光雷达模型如图 3-2 所示，通过激光雷达测量数据读取被测点返回的坐标值，其中 d 代表被测点距离雷达中心的距离， β 表示激光光束与平面 $x_1 o_1 y_1$ 之间的夹角， α 表示激光光束与平面 $x_1 o_1 z_1$ 之间的夹角。

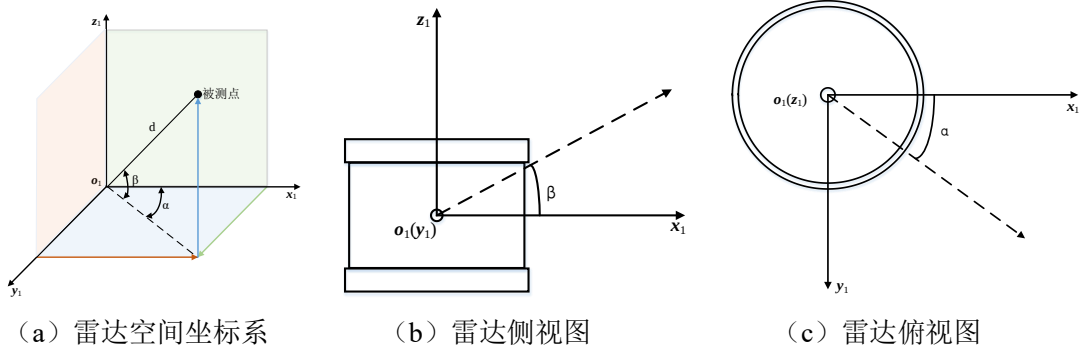


图 3-2 激光雷达测量原理

由于激光雷达坐标系的 X 轴、 Y 轴和 Z 轴方向与无人叉车坐标系保持一致， X 轴方向上存在一个偏移量， Z 轴方向上存在一个偏移量，因此激光雷达坐标系相对于无人叉车坐标系的旋转矩阵 R_{LR} 为一个单位矩阵。

$$R_{LR} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-7)$$

激光雷达坐标系相对于世界坐标系的旋转矩阵 R_{LW} 见式 3-8 所示。

$$\begin{aligned} R_{LW} &= R \cdot R_{LR} \\ &= \begin{bmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi \\ \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \end{aligned} \quad (3-8)$$

世界坐标系到激光雷达坐标系的转换只需要旋转矩阵 R_{LW} 求逆矩阵即可，见式 3-9 所示。

$$\begin{aligned} R_{LW}^{-1} &= R_{LW}^T \\ &= \begin{bmatrix} \cos \varphi \cos \theta & \sin \varphi \cos \theta & -\sin \theta \\ \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \cos \theta \sin \phi \\ \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi & \cos \theta \cos \phi \end{bmatrix} \end{aligned} \quad (3-9)$$

因为激光雷达坐标系相对于无人叉车坐标系在 X 轴方向上存在 1.5 个偏移

量， Z 轴方向上存在一个偏移量，所以激光雷达坐标系相对于无人叉车坐标系的平移向量 $\mathbf{t}_{LR}=(1.5, 0, 1)$ ，无人叉车在世界坐标系下的位姿是 $\mathbf{t}=(x, y, z)$ ，所以激光雷达坐标系相对于世界坐标系的平移向量 $\mathbf{t}_{LW}$ 见式 3-10 所示。

$$\mathbf{t}_{LW} = \mathbf{t}_{LR} + \mathbf{t} = \begin{pmatrix} x+1.5 \\ y \\ z+1 \end{pmatrix} \quad (3-10)$$

激光雷达的齐次变换矩阵 $\mathbf{T}_{LW}$ 见式 3-11 所示。

$$\begin{aligned} \mathbf{T}_{LW} &= \begin{bmatrix} \mathbf{R}_{LW} & \mathbf{t}_{LW} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi & x+1.5 \\ \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi & y \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi & z+1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3-11)$$

激光雷达被测物体与传感器之间的距离计算公式见式 3-12 所示。

$$d = \frac{c \cdot \Delta t}{2} \quad (3-12)$$

其中， c 是光速， $c=3 \times 10^8 \text{m/s}$ ； Δt 是激光束从发射到返回的时间差。

所以对于被测物体在激光雷达坐标系下的坐标由极坐标转换成笛卡尔坐标解算过程如下：

$$\begin{aligned} x_1 &= d \cdot \cos \beta \cdot \cos \alpha \\ y_1 &= d \cdot \cos \beta \cdot \sin \alpha \\ z_1 &= d \cdot \sin \alpha \end{aligned} \quad (3-13)$$

3.3.2 IMU 测量模型

惯性测量单元（Inertial Measurement Unit, IMU）是一种用于测量物体在三维空间中加速度、角速度（以及有时磁场强度）变化的传感器设备。IMU 的工作原理基于惯性原理和传感器技术，能够实时监测物体的运动状态（如位置、速度、姿态等）^[73]。本文采用的 IMU 为维特 HWT905 九轴 IMU，包含加速度计、陀螺仪和磁力计三个主要组件。通过这三部分，IMU 能够分别检测三轴加速度、三轴角速度以及三轴偏航角。其工结构如图 3-3 所示。

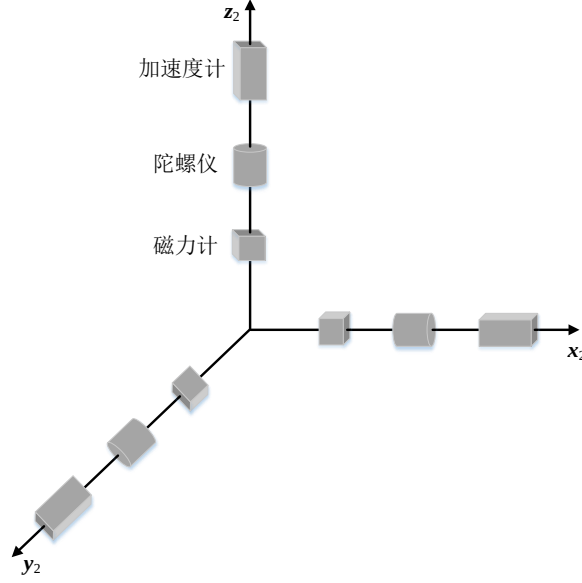


图 3-3 IMU 模型

因为 IMU 坐标系相对于无人叉车坐标系在 X 轴方向上存在两个偏移量，所以 IMU 坐标系相对于无人叉车坐标系的平移向量 $t_{IR}=(1.5, 0, 0)$ ，无人叉车在世界坐标系下的位姿是 $t=(x, y, z)$ ，所以 IMU 坐标系相对于世界坐标系的平移向量 t_{IW} 见式 3-14 所示。

$$t_{IW} = t_{IR} + t = \begin{pmatrix} x+1.5 \\ y \\ z \end{pmatrix} \quad (3-14)$$

IMU 坐标系相对于无人叉车坐标系的旋转矩阵 R_{IR} 为一个单位矩阵。

$$R_{IR} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-15)$$

IMU 坐标系相对于世界坐标系的旋转矩阵 R_{IW} 见式 3-16 所示，齐次变换矩阵 T_{IW} 见式 3-17 所示。

$$\begin{aligned} R_{IW} &= R \cdot R_{IR} \\ &= \begin{bmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi \\ \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \end{aligned} \quad (3-16)$$

$$\begin{aligned}
 T_{IW} &= \begin{bmatrix} \mathbf{R}_{IW} & \mathbf{t}_{IW} \\ 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi & x+1.5 \\ \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi & y \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi & z \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3-17}$$

然而我们使用 IMU 的陀螺仪来估算物体的滚转角、俯仰角和偏航角，通常通过四元数来描述旋转，四元数可以避免万向节死锁问题。

陀螺仪测量的角速度 ω 见式 3-18 所示。

$$\omega = \begin{bmatrix} \omega_{x_2} \\ \omega_{y_2} \\ \omega_{z_2} \end{bmatrix} \tag{3-18}$$

通过四元数来描述旋转，则四元数的更新见式 3-19 所示。

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \cdot \boldsymbol{\Omega}(\omega) \tag{3-19}$$

其中 $\mathbf{q}=[q_1, q_2, q_3, q_4]$ 是无人叉车四元数的表示；角速度的四元数矩阵 $\boldsymbol{\Omega}(\omega)$ 定义见式 3-20 所示。

$$\boldsymbol{\Omega}(\omega) = \begin{bmatrix} 0 & -\omega_{x_2} & -\omega_{y_2} & -\omega_{z_2} \\ \omega_{x_2} & 0 & \omega_{z_2} & -\omega_{y_2} \\ \omega_{y_2} & -\omega_{z_2} & 0 & \omega_{x_2} \\ \omega_{z_2} & \omega_{y_2} & -\omega_{x_2} & 0 \end{bmatrix} \tag{3-20}$$

通过四元数可以转换为旋转矩阵 $\mathbf{R}_{IW}(t)$ ，进而得到无人叉车的姿态。

$$\mathbf{R}_{IW}(t) = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{3-21}$$

加速度计会测量无人叉车的总加速度 $\mathbf{a}$ ，其中包含了由运动和重力引起的加速度。我们需要将加速度中的重力项去除，得到无人叉车的运动加速度。根据无人叉车的姿态 $\mathbf{R}_I(t)$ 可以得到重力向量在世界坐标系中的表示 $\mathbf{g}=[0 \ 0 \ 9.81]$ ，通过加速度计测量无人叉车的加速度 $\mathbf{a}_I$ 由式 3-22 计算得到。

$$\mathbf{a}_I = \mathbf{R}_{Iw}(t)(\mathbf{a} - \mathbf{g}) \quad (3-22)$$

运用欧拉法通过加速度积分来更新无人叉车速度见式 3-23 所示。

$$\mathbf{v}(t) = \mathbf{v}(t-1) + \mathbf{a}_I(t) \cdot \Delta t \quad (3-23)$$

再通过速度积分来更新无人叉车位置见式 3-24 所示。

$$\mathbf{p}(t) = \mathbf{p}(t-1) + \mathbf{v}(t) \cdot \Delta t \quad (3-24)$$

因为磁力计能够提供物体的磁场方向，因此可以用于估计偏航角。磁力计测量的是地球磁场的三维分量，一般表示为 $\vec{M} = [\mathbf{M}_{x_i} \quad \mathbf{M}_{y_i} \quad \mathbf{M}_{z_i}]$ ，其中 $\mathbf{M}_{x_i}$ 是沿着 x 轴方向的磁场分量， $\mathbf{M}_{y_i}$ 是沿着 y 轴方向的磁场分量， $\mathbf{M}_{z_i}$ 是沿着 z 轴方向的磁场分量。偏航角 φ_m 是无人叉车相对于地磁北极的角度，计算公式见式 3-25 所示。

$$\varphi_m = \text{atan2}(\mathbf{M}_{y_i}, \mathbf{M}_{x_i}) \quad (3-25)$$

其次，再通过磁力计与加速度计与陀螺仪结合修正偏航角。加速度计测量无人叉车的加速度 $\mathbf{a}_I$ 记作 $\mathbf{a}_I = [a_{Ix} \quad a_{Iy} \quad a_{Iz}]$ ，通过 $\mathbf{a}_I$ 计算无人叉车的俯仰角 θ_I 和翻滚角 ϕ_I 。

$$\begin{aligned} \theta_I &= \arctan 2(-a_{Ix}, \sqrt{a_{Iy}^2 + a_{Iz}^2}) \\ \phi_I &= \arctan 2(a_{Iy}, a_{Iz}) \end{aligned} \quad (3-26)$$

将磁力计的 $\mathbf{M}_{x_i}$ 和 $\mathbf{M}_{y_i}$ 分量进行旋转变换：

$$\begin{aligned} \mathbf{M}'_{x_i} &= \mathbf{M}_{x_i} \cos \theta + \mathbf{M}_{z_i} \sin \theta \\ \mathbf{M}'_{y_i} &= \mathbf{M}_{x_i} \sin \phi \sin \theta + \mathbf{M}_{y_i} \cos \phi - \mathbf{M}_{z_i} \sin \theta \cos \phi \end{aligned} \quad (3-27)$$

因此，通过磁力计与加速度计结合修正的偏航角 φ'_m 计算公式见 3-28 所示。

$$\varphi'_m = \arctan 2(\mathbf{M}'_{y_i}, \mathbf{M}'_{x_i}) \quad (3-28)$$

在动态运动时，陀螺仪提供的角速度积分可以帮助估算无人叉车的实时偏航角 φ_g ，通过互补滤波计算得到更精确的偏航角修正。

$$\varphi_I(t) = 0.98(\varphi_g(t) + \omega_{z_2} \cdot \Delta t) + 0.02\varphi'_m(t) \quad (3-29)$$

3.3.3 堆垛式无人叉车运动学模型

由于堆垛式无人叉车在物流运输中的运行速度较慢，侧向力较小，因此可以假设在运行过程中不会发生打滑现象。因此，我们只需建立堆垛式无人叉车的运动学模型来描述其运动特性。该模型是导航与控制算法设计的基础，能够准确描述堆垛式无人叉车的运动状态和运动规律。

我们采用非完整约束系统的方法进行运动学建模^[74]。该单舵轮驱动的堆垛式无人叉车采用三轮结构，由一个主舵轮和两个从动轮组成^[74]。此外，为了提高稳定性和辅助行走，无人叉车的底盘四周还配备了万象辅助轮^[75]。其底盘模型如图 3-4 所示。

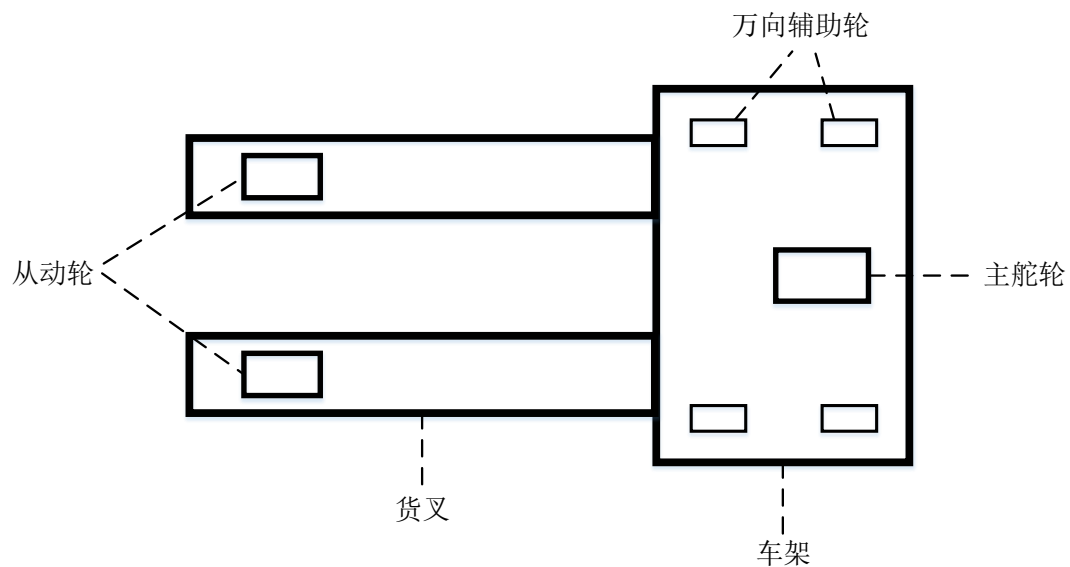


图 3-4 单舵轮驱动的堆垛式无人叉车底盘模型

如图 3-4 所示，单舵轮驱动的堆垛式无人叉车具有出色的驱动力，转弯灵活，控制性能优越，广泛应用于复杂的车间环境中。为了更好地分析单舵轮驱动的堆垛式无人叉车的运动学模型，我们对图 3-4 所示的单舵轮驱动的堆垛式无人叉车底盘模型进行了简化，简化后的模型如图 3-5 所示，呈现了单舵轮驱动的堆垛式无人叉车的运动学特性。

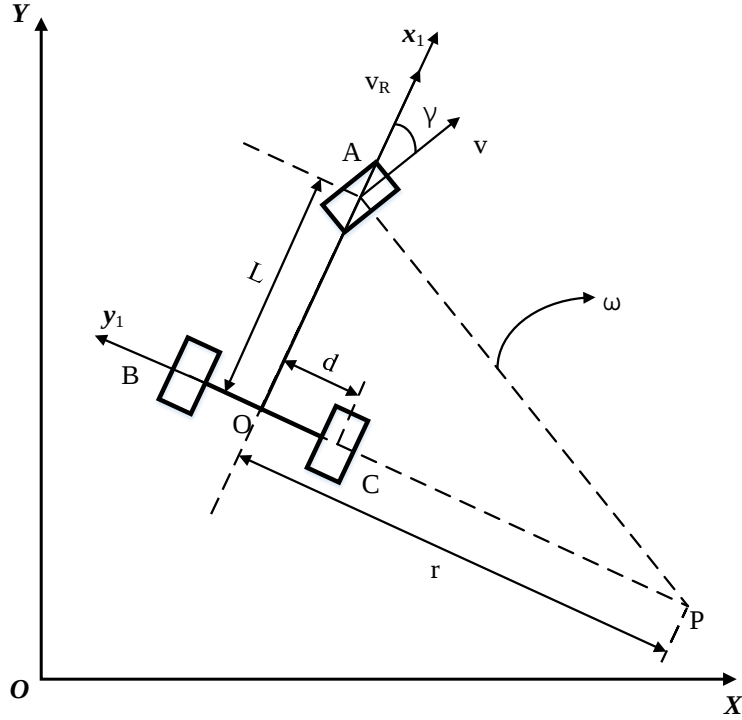


图 3-5 单舵轮驱动的堆垛式无人叉车运动学模型

如图 3-5 所示, A 为单舵轮驱动的堆垛式无人叉车的主舵轮, B 和 C 为从动轮, L 为无人叉车的轴距, 即主舵轮到从动轮之间的距离, d 为半轮距, r 为无人叉车在运动过程中的旋转半径。假设无人叉车在运动过程中的瞬时角速度为 ω , 瞬时线速度为 v_R ; 无人叉车运动过程中舵轮的瞬时转动速度为 v , 瞬时偏转角为 γ 。根据几何学原理, 可以推导出无人叉车的线速度、角速度及旋转半径。

$$v_R = v \cos \gamma \quad (3-30)$$

$$r = \begin{cases} \frac{L}{\tan \gamma}, (\gamma > 0) \\ \frac{-L}{\tan \gamma}, (\gamma < 0) \end{cases} \quad (3-31)$$

$$\omega = \frac{v_R}{r} = \begin{cases} \frac{v \sin \gamma}{L}, (\gamma > 0) \\ \frac{-v \sin \gamma}{L}, (\gamma < 0) \end{cases} \quad (3-32)$$

假设无人叉车在 Δt 时间内由 M 点运动到 N 点, 运动示意图如图 3-6 所示。

在世界坐标系 XOY 下, 无人叉车车体的偏航角为 φ 。

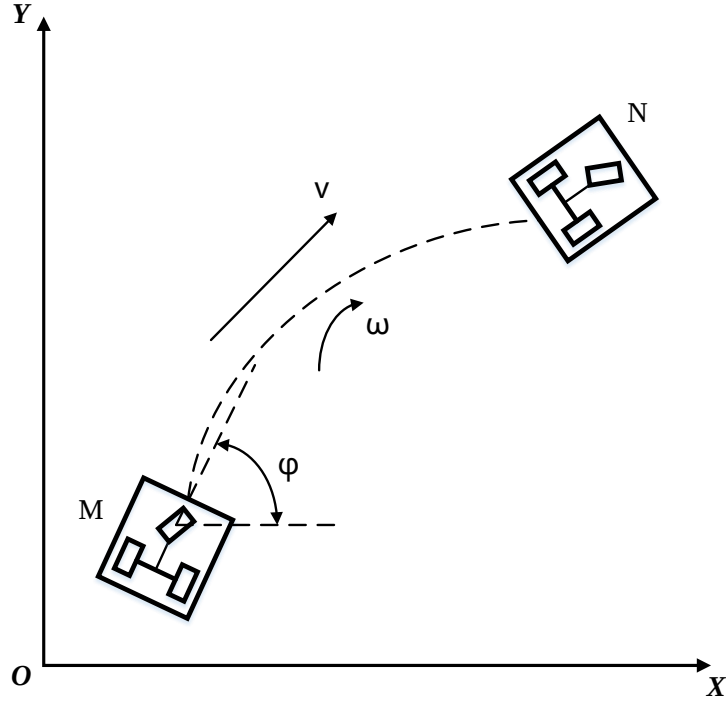


图 3-6 单舵轮驱动的堆垛式无人叉车运动示意图

由图 3-6 可知,若已知单舵轮驱动的堆垛式无人叉车的起始点 M 的位置(x_M , y_M),则可以通过 M 点的坐标计算出在时间间隔 Δt 内到达目标点 N 的位置坐标,具体推算步骤如下:

在时间 Δt 内,无人叉车车体偏航角为 φ 会发生变化,变化量 $\Delta\varphi$ 见式 3-33。

$$\Delta\varphi = \omega \cdot \Delta t = \frac{v \cdot \Delta t \cdot \sin \gamma}{L} \quad (3-33)$$

在时间 Δt 内,无人叉车车体的 X 轴坐标会发生变化,变化量 Δx 见式 3-34。

$$\Delta x = -\frac{L \cdot \sin \varphi}{\tan \gamma} + \frac{L \cdot \sin(\varphi + \frac{v \cdot \Delta t \cdot \sin \gamma}{L})}{\tan \gamma} \quad (3-34)$$

在时间 Δt 内,无人叉车车体的 Y 轴坐标会发生变化,变化量 Δy 见式 3-35。

$$\Delta y = \frac{L \cdot \cos \varphi}{\tan \gamma} - \frac{L \cdot \cos(\varphi + \frac{v \cdot \Delta t \cdot \sin \gamma}{L})}{\tan \gamma} \quad (3-35)$$

由式 3-34 和式 3-35 可得 N 点坐标 (x_N , y_N) 的值, 见式 3-36 和式 3-37。

$$x_N = x_M + \Delta x = x_M - \frac{L \cdot \sin \varphi}{\tan \gamma} + \frac{L \cdot \sin(\varphi + \frac{v \cdot \Delta t \cdot \sin \gamma}{L})}{\tan \gamma} \quad (3-36)$$

$$y_N = y_M + \Delta y = y_M + \frac{L \cdot \cos \varphi}{\tan \gamma} - \frac{L \cdot \cos(\varphi + \frac{v \cdot \Delta t \cdot \sin \gamma}{L})}{\tan \gamma} \quad (3-37)$$

由式 3-33 可得 N 点偏航角 φ_N 的值，见式 3-38。

$$\varphi_N = \varphi + \Delta \varphi = \varphi + \frac{v \cdot \Delta t \cdot \sin \gamma}{L} \quad (3-38)$$

3.4 数据融合模型

本文研究的单舵轮驱动的堆垛式无人叉车主要用于工业车间复杂环境中的货物搬运和堆垛，因此对定位精度有着极高的要求。然而，仅依靠单一激光雷达传感器进行无人叉车的定位导航，其精度可能无法满足需求。因此，本研究采用激光雷达与 IMU 数据融合的方法，以提高定位导航的精度。

我们采用 EKF 的方法进行数据融合模型的构建。使用 EKF 建模首先定义无人叉车系统的状态向量 $\mathbf{x}(t)$ 。

$$\mathbf{x}(t) = [x(t) \ y(t) \ z(t) \ v_x(t) \ v_y(t) \ v_z(t) \ \phi(t) \ \theta(t) \ \varphi(t) \ \omega_x(t) \ \omega_y(t) \ \omega_z(t)]^T \quad (3-39)$$

EKF 使用 IMU 的数据来预测系统的状态。IMU 提供的是加速度和角速度信息。根据 IMU 测量数据，得到预测方程 $\mathbf{u}(t)$ 。

$$\mathbf{u}(t) = [a_x(t) \ a_y(t) \ a_z(t) \ \omega_x(t) \ \omega_y(t) \ \omega_z(t)]^T \quad (3-40)$$

通过加速度 $\mathbf{a}=[a_x(t) \ a_y(t) \ a_z(t)]^T$ ，更新速度见式 3-41 所示，再通过速度更新位置见式 3-42 所示。

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \mathbf{a}(t) \cdot \Delta t \quad (3-41)$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t) \cdot \Delta t \quad (3-42)$$

通过 IMU 提供的角速度 $\boldsymbol{\omega}=[\omega_x(t) \ \omega_y(t) \ \omega_z(t)]^T$ 进行姿态更新。

$$\begin{cases} \phi(t + \Delta t) = \phi(t) + \omega_x(t) \cdot \Delta t \\ \theta(t + \Delta t) = \theta(t) + \omega_y(t) \cdot \Delta t \\ \varphi(t + \Delta t) = \varphi(t) + \omega_z(t) \cdot \Delta t \end{cases} \quad (3-43)$$

激光雷达（LiDAR）提供的是距离（点云数据）。通过激光雷达测量的距离和角度，可以估算物体的相对位置。对于 RS-LiDAR-16，其提供的是 360° 的点云数据，可以提取出激光雷达的数据来用于修正姿态估计。激光雷达的观测模型见式 3-44 所示。

$$\mathbf{z}_L(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{V}_L(t) \quad (3-44)$$

使用 EKF 的预测和修正步骤更新状态估计。

在预测步骤中，状态向量 $\mathbf{x}(t)$ 的更新是通过 IMU 数据来完成的，预测公式见 3-45 所示。

$$\mathbf{x}_{\text{pred}}(t + \Delta t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \begin{bmatrix} x(t + \Delta t) = x(t) + v_x(t) \cdot \Delta t + \frac{1}{2} a_x(t) \Delta t^2 \\ y(t + \Delta t) = y(t) + v_y(t) \cdot \Delta t + \frac{1}{2} a_y(t) \Delta t^2 \\ z(t + \Delta t) = z(t) + v_z(t) \cdot \Delta t + \frac{1}{2} a_z(t) \Delta t^2 \\ v_x(t + \Delta t) = v_x(t) + a_x(t) \cdot \Delta t \\ v_y(t + \Delta t) = v_y(t) + a_y(t) \cdot \Delta t \\ v_z(t + \Delta t) = v_z(t) + a_z(t) \cdot \Delta t \\ \phi(t + \Delta t) = \phi(t) + \omega_x(t) \cdot \Delta t \\ \theta(t + \Delta t) = \theta(t) + \omega_y(t) \cdot \Delta t \\ \varphi(t + \Delta t) = \varphi(t) + \omega_z(t) \cdot \Delta t \\ \omega_x(t + \Delta t) = \omega_x(t) \cdot \Delta t \\ \omega_y(t + \Delta t) = \omega_y(t) \cdot \Delta t \\ \omega_z(t + \Delta t) = \omega_z(t) \cdot \Delta t \end{bmatrix} \quad (3-45)$$

在修正步骤中，我们根据激光雷达的数据来修正预测的状态。首先计算激光雷达的观测值。

$$\mathbf{z}_L(t) = \mathbf{h}(\mathbf{x}_{\text{pred}}(t)) + \mathbf{V}_L(t) \quad (3-46)$$

使用卡尔曼增益来修正预测的状态。

$$\mathbf{K}(t) = \mathbf{P}_{\text{pred}}(t) \mathbf{H}^T (\mathbf{H} \mathbf{P}_{\text{pred}}(t) + \mathbf{R})^{-1} \quad (3-47)$$

其中， $\mathbf{P}_{\text{pred}}(t)$ 是预测的协方差矩阵见式 3-48 所示， $\mathbf{H}$ 是观测模型的雅可比

矩阵， R 是观测噪声协方差矩阵。

$$\mathbf{P}_{\text{pred}}(t + \Delta t) = \mathbf{F}_{t+\Delta t} \mathbf{P}_{\text{pred}}(t) \mathbf{F}_{t+\Delta t}^T + \mathbf{Q}_{t+\Delta t} \quad (3-48)$$

其中， $\mathbf{F}_{t+\Delta t}$ 是状态转移矩阵见式 3-49 所示， $\mathbf{Q}_{t+\Delta t}$ 是过程噪声协方差矩阵。

$$\mathbf{F}_{t+\Delta t} = \frac{\partial \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))}{\partial \mathbf{x}(t)} \quad (3-49)$$

由卡尔曼增益来更新状态估计（见式 3-50 所示），以及更新状态的协方差矩阵（见式 3-51 所示）。

$$\mathbf{x}(t) = \mathbf{x}_{\text{pred}}(t) + \mathbf{K}(t)(\mathbf{z}_L(t) - \mathbf{h}(\mathbf{x}_{\text{pred}}(t))) \quad (3-50)$$

$$\mathbf{P}(t) = (\mathbf{I} - \mathbf{K}(t)\mathbf{H})\mathbf{P}_{\text{pred}}(t) \quad (3-51)$$

结合激光雷达和九轴 IMU 数据来构建扩展卡尔曼滤波器(EKF)模型中 IMU 提供了加速度和角速度数据，用于预测车辆的状态，而激光雷达提供了点云数据用于修正这些预测。EKF 通过卡尔曼增益来融合两者的数据，从而获得更精确的状态估计。

3.5 堆垛式无人叉车运动学仿真分析

3.5.1 基于运动学模型的仿真分析

该运动学仿真分析的核心目的是验证前文所建立的单舵轮驱动的堆垛式无人叉车运动学模型的正确性，即验证状态转移方程是否能够准确描述单舵轮驱动的堆垛式无人叉车的运动行为。根据前文中给出的运动学方程和基本符号定义，在 MATLAB 中实现了以下几种运动学仿真：直线运动仿真、曲线运动仿真、不同线速度条件下的运动仿真、不同转向角条件下的运动仿真，以及转向角 γ 随时间变化的运动仿真。

在直线运动学仿真中，设置初始驱动轮线速度为 1 m/s，初始转向角度为 0 rad，仿真时间为 10 s，单舵轮驱动的堆垛式无人叉车保持直线运动状态其理论运动轨迹为一条直线；通过式 3-36，3-37，3-38 计算单舵轮驱动的堆垛式无人叉车运动学模型运动轨迹点坐标，并绘制其实际运动轨迹。结果显示，轨迹跟踪均方根误差为 8.4763e-15，几乎为 0，表明单舵轮驱动的堆垛式无人叉车的实际轨迹

与目标轨迹几乎完全重合。在曲线运动学仿真中,设置初始驱动轮线速度为 1 m/s ,初始转向角度设置为 0.2 rad ,仿真时间为 10 s ,单舵轮驱动的堆垛式无人叉车保持曲线运动状态其理论运动轨迹为一段圆弧,通过式 3-36, 3-37, 3-38 计算单舵轮驱动的堆垛式无人叉车运动学模型运动轨迹点坐标,并绘制其实际运动轨迹。计算结果显示,轨迹跟踪均方根误差为 0.02858 ,接近 0 ,同样表明单舵轮驱动的堆垛式无人叉车的实际轨迹与目标轨迹几乎重合。直线运动与曲线运动的轨迹图如图 3-7 所示, MATLAB 仿真代码见附录 1, 附录 2 所示,进一步验证了单舵轮驱动的堆垛式无人叉车的实际轨迹与目标轨迹的高度一致性。这表明,运动学模型的推导与实现是正确的。

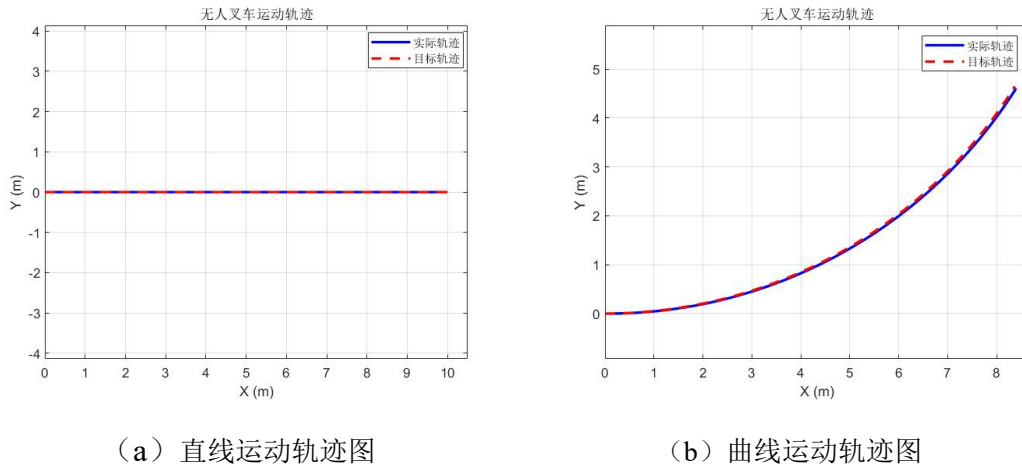


图 3-7 直线与曲线运动的运动学仿真图

在不同线速度条件下的运动仿真中,设置转向角度为 0.2 rad 固定不变,分别将速度设置为 0.5 m/s 、 1 m/s 、 1.5 m/s ,仿真时间为 10 s ,通过式 3-36, 3-37, 3-38 模拟单舵轮驱动的堆垛式无人叉车的曲线运动,并绘制了其运动轨迹。在不同转向角度条件下的运动仿真中,设置速度为 1 m/s 固定不变,分别将转向角度设置为 0.1 rad 、 0.3 rad 、 0.5 rad ,仿真时间为 10 s ,通过式 3-36, 3-37, 3-38 模拟单舵轮驱动的堆垛式无人叉车的曲线运动,并绘制了其运动轨迹。计算结果表明,理论曲率为 0.10136 ,仿真偏航角变化率为 0.10136 ,理论偏航角变化率为 0.10136 ,三者完全一致,进一步验证了运动学模型的推导和实现是正确的。不同线速度条件下单舵轮驱动的堆垛式无人叉车的运动轨迹与不同转向角条件下的运动轨迹图如图 3-8 所示,而不同线速度条件下单舵轮驱动的堆垛式无人叉车的运动偏航角变化则如图 3-9 所示, MATLAB 仿真代码见附录 3 所示。

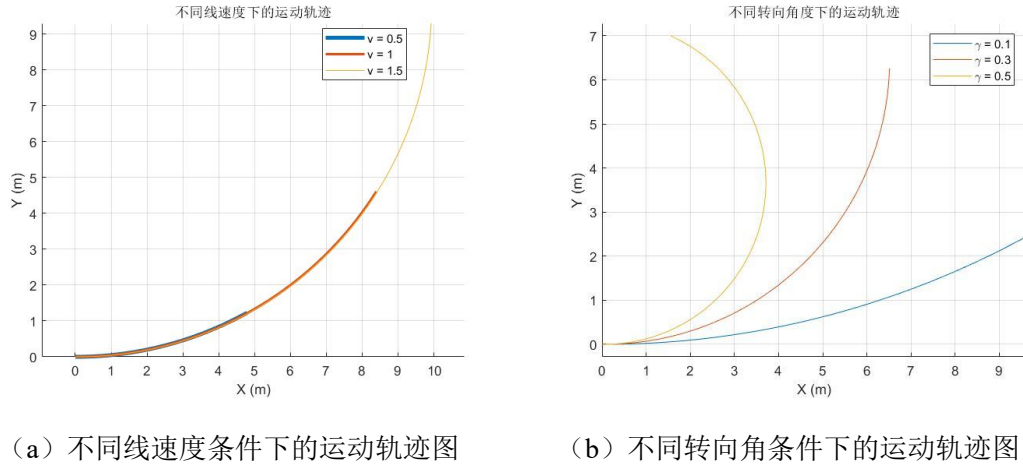


图 3-8 不同线速度条件下与不同转向角条件下的运动学仿真图

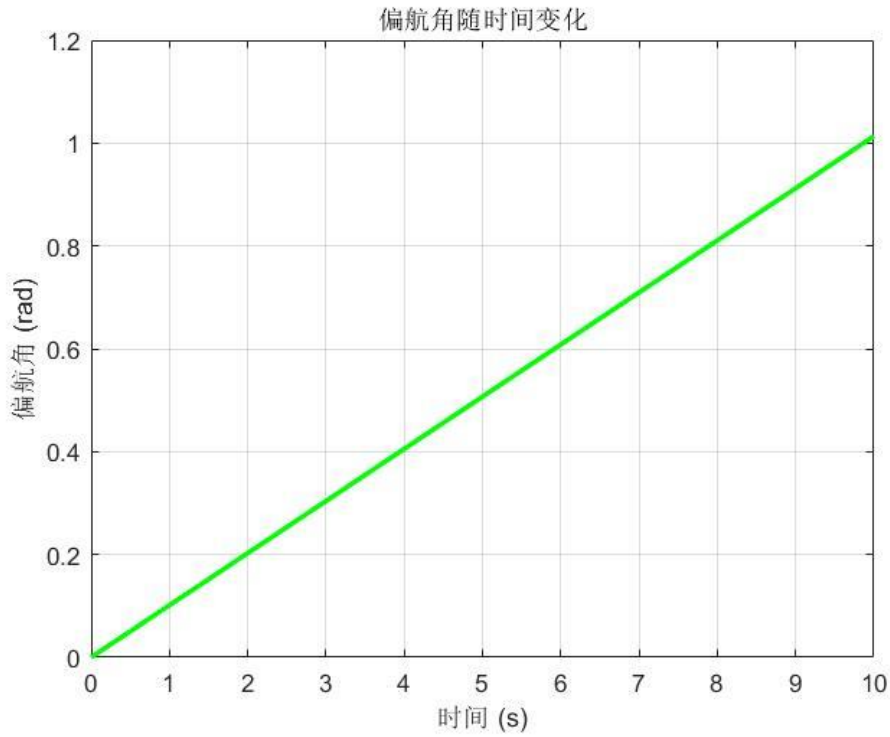
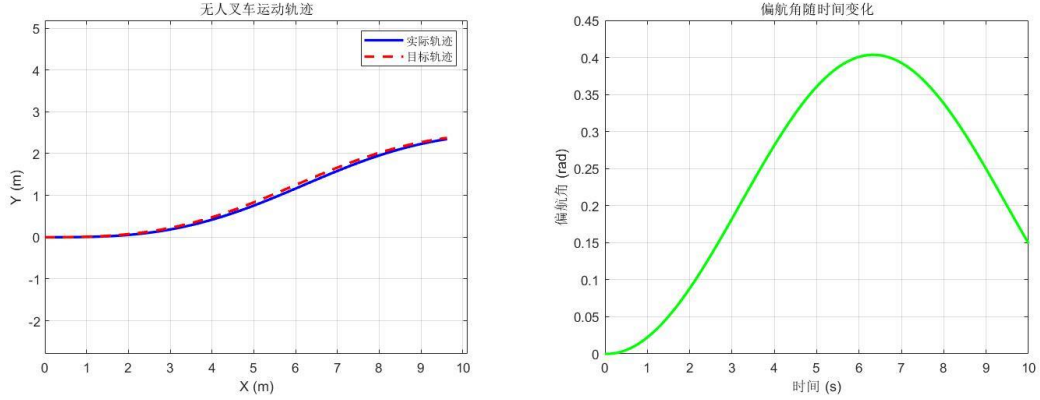


图 3-9 不同线速度条件下的运动偏航角变化图

在转向角 γ 随时间变化的运动仿真中，设置速度为 1 m/s，仿真时间为 10 s，转向角度随时间的正弦频率变化，模拟单舵轮驱动的堆垛式无人叉车的曲线运动状态，并绘制了其运动轨迹。如图 3-10 所示，单舵轮驱动的堆垛式无人叉车的实际轨迹与目标轨迹几乎完全重合，进一步验证了运动学模型的推导和实现是正确的。MATLAB 仿真代码见附录 4 所示。



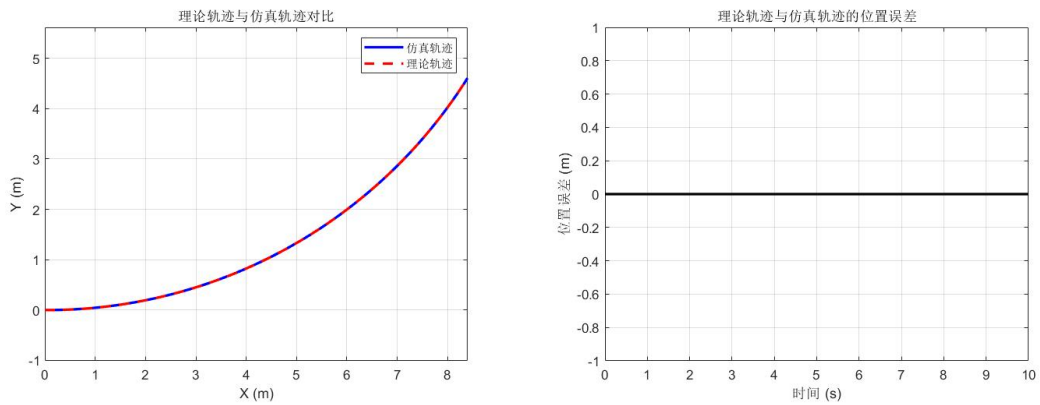
(a) 转向角随时间变化下的运动轨迹图 (b) 转向角随时间变化下的运动偏航角变化图

图 3-10 转向角 γ 随时间变化下的运动学仿真图

根据前文给出的运动学方程和基本符号定义，在 MATLAB 中实现的直线运动仿真、曲线运动仿真、不同线速度条件下的运动仿真、不同转向角条件下的运动仿真以及转向角 γ 随时间变化的运动仿真（MATLAB 代码见附录 1-4）结果均表明，3.3 节中所推导的运动学模型及其实现是正确的。

3.5.2 引入数据融合模型的运动学仿真分析

该仿真分析的核心目的是验证前文所建立的单舵轮驱动的堆垛式无人叉车数据融合模型的正确性，因此需要引入激光雷达模型和 IMU 模型。根据 3.4 节中给出的数据融合方程及基本符号定义，在 MATLAB 中实现了引入数据融合模型的运动学仿真。MATLAB 代码见附录 5。



(a) 理想状态下的运动轨迹图

(b) 理想状态下的轨迹误差图

图 3-11 理想状态下引入数据融合模型的运动学仿真图

该仿真主要实现了在理想状态下引入数据融合模型的运动学仿真，以及在引入噪声误差的情况下进行数据融合模型的运动学仿真。理想状态下引入数据融合模型的运动学仿真，设置初始驱动轮线速度为 1 m/s ，初始转向角度设置为 0.2 rad ，仿真时间为 10 s ，结果如图 3-11 所示，单舵轮驱动的堆垛式无人叉车的实际轨迹与目标轨迹几乎完全重合，轨迹误差始终为 0，表明数据融合模型和运动学模型的推导与实现是正确的。

引入噪声误差后的数据融合模型仿真设置初始驱动轮线速度为 1 m/s ，初始转向角度设置为 0.2 rad ，仿真时间为 10 s ，结果如图 3-12 所示，单舵轮驱动的堆垛式无人叉车的实际轨迹与目标轨迹相似，但存在一定误差，轨迹误差呈波动状态，且数值较小。这表明，虽然数据融合模型和运动学模型的推导与实现是正确的，但在实际过程中仍然存在噪声误差的影响。MATLAB 代码见附录 6。

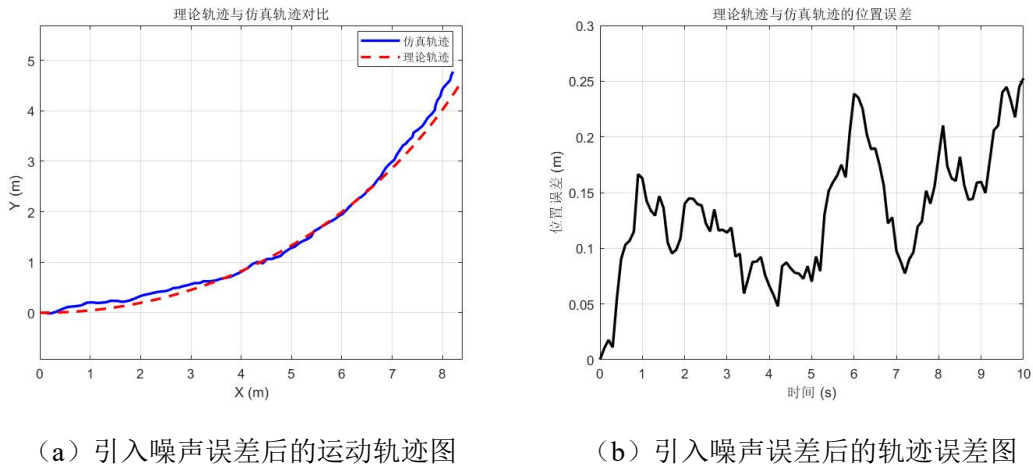


图 3-12 引入噪声误差后的数据融合模型的运动学仿真图

根据前文给出的数据融合方程及基本符号定义，在 MATLAB 中实现的引入数据融合模型的运动学仿真结果表明，3.3 节中所推导的运动学模型以及 3.4 节中所推导的数据融合模型及其实现是正确的。然而，在实际过程中，噪声误差的影响仍然存在，这为后续研究提供了理论基础。

3.5.3 数据融合系统仿真分析

该仿真的核心目标是验证单舵轮无人叉车的运动学模型是否能够准确刻画叉车的运动行为，并检验运动学方程在位置、速度和航向角更新方面的正确性。与此同时，仿真还旨在实现多传感器数据融合，为无人叉车的状态估计、控制与路径规划提供可靠的理论支持和技术基础。

基于 MATLAB 平台分析该数据融合系统的性能，根据式 3-30 到式 3-51 的数学模型表达式建立 Simulink 模型进行验证，初始实验参数如表 3-1 所示。

表 3-1 Simulink 仿真初始实验参数

参数名称	参数数值	单位
舵轮速度 v_{wheel}	1	m/s
舵轮转角 γ	0.2	rad
Z 轴加速度 a_z	0	m ² /s
X 轴角速度 ω_x	0	rad/s
Y 轴角速度 ω_y	0	rad/s
重力加速度 g	9.81	m ² /s
仿真时间 t	10	s

将表 3-1 中的数值作为 Simulink 仿真的初始输入条件，结合式 3-30 到式 3-51 的数学模型表达式建立 Simulink 模型进行数据融合系统仿真分析，Simulink 模型如图 3-13 所示。

根据图 3-13 所示的 Simulink 仿真模型，计算基于单舵轮驱动的堆垛式无人叉车运动学模型解算得到的位姿结果，如图 3-14 所示。同时，利用该模型推算出基于 EKF 信息融合更新后的单舵轮驱动堆垛式无人叉车状态估计，如图 3-15 所示。

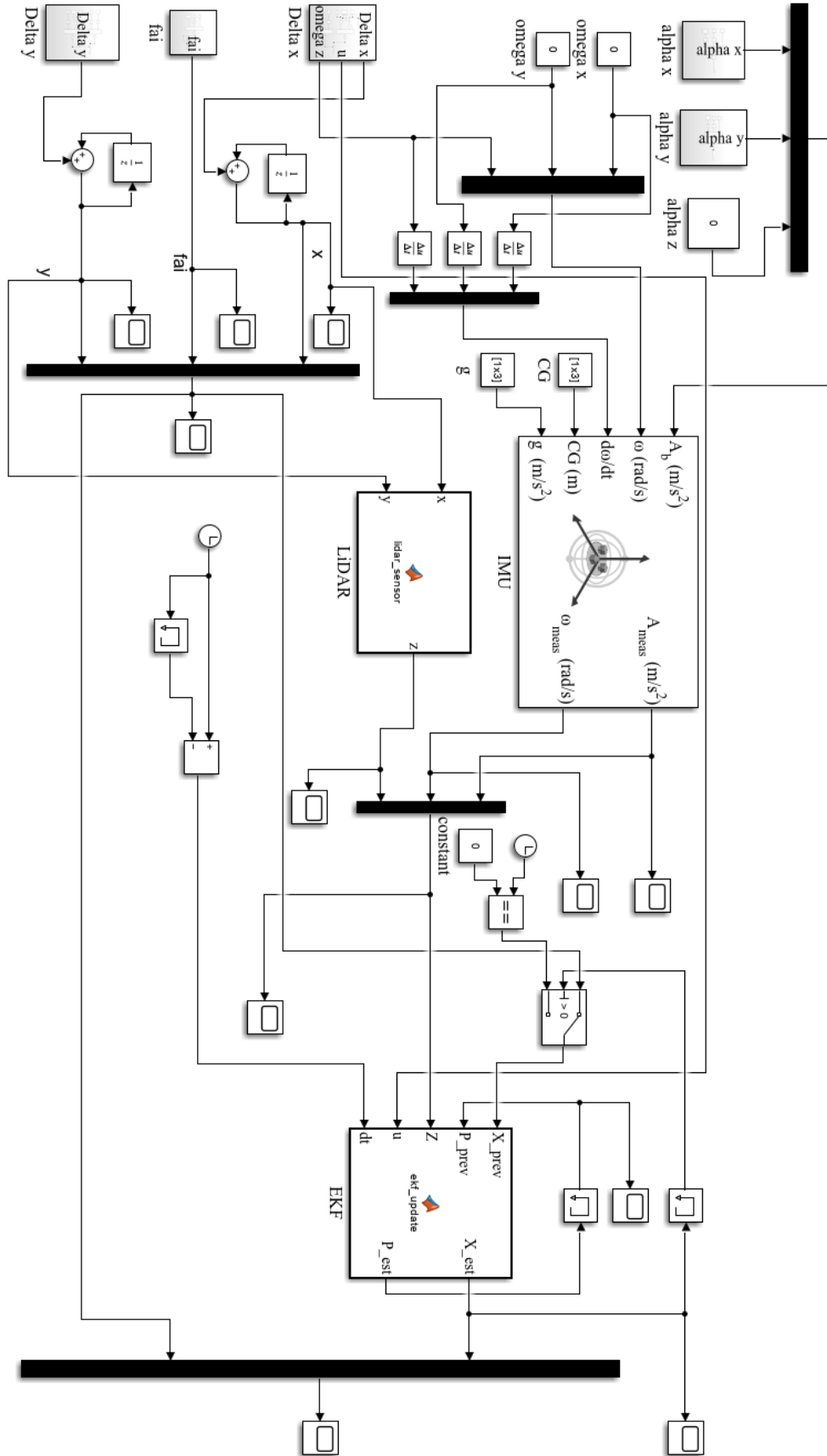


图 3-13 数据融合系统 Simulink 仿真模型图

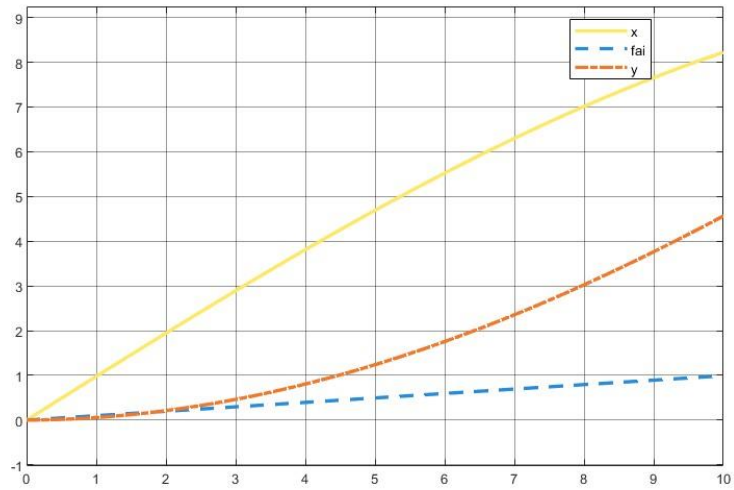


图 3-14 单舵轮驱动的堆垛式无人叉车运动学模型解算位姿图

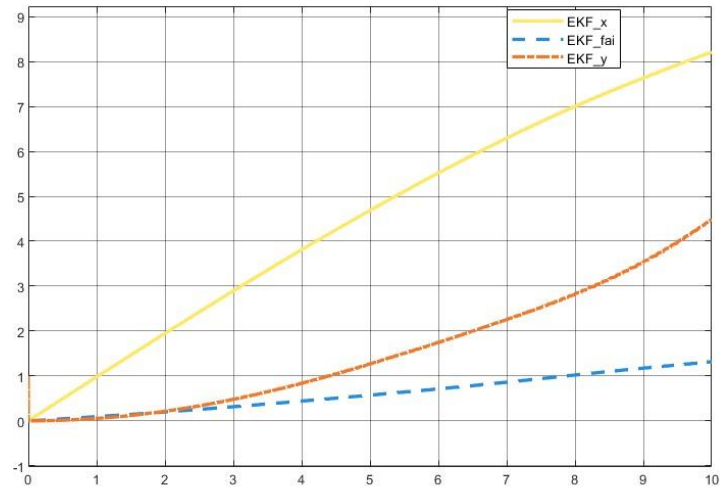


图 3-15 EKF 信息融合更新后的单舵轮驱动堆垛式无人叉车状态估计图

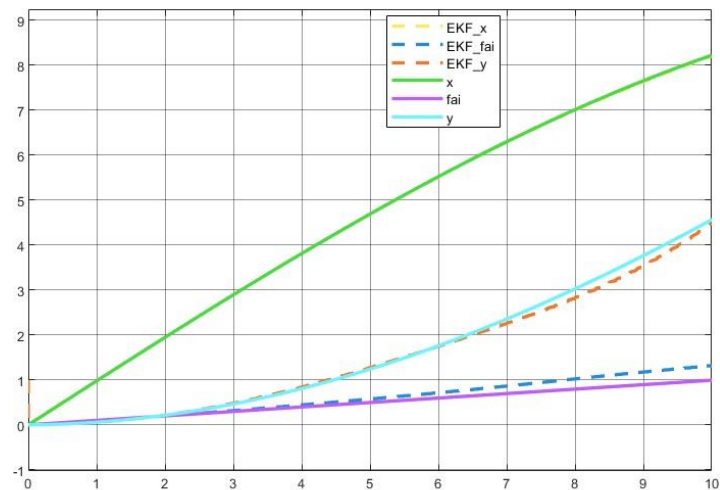


图 3-16 位姿对比图

最后, 通过对比该模型基于运动学方程解算得到的单舵轮驱动堆垛式无人叉

车位姿与基于 EKF 信息融合更新后的状态估计，评估数据融合模型的性能。如图 3-16 所示，融合后的状态估计值与真实值高度吻合，EKF 信息融合更新后的状态估计结果几乎与运动学模型解算的位姿重合，这一结果验证了系统的正确性和数据融合模型的有效性。

3.6 本章小结

本章对单舵轮驱动的堆垛式无人叉车运动模型进行了系统的建模与仿真分析。首先，明确了单舵轮驱动的堆垛式无人叉车的坐标系统，为后续的运动模型和导航设计提供了统一的参考框架。接着，深入分析了单舵轮驱动的堆垛式无人叉车的运动模型，涵盖了激光雷达测量模型、IMU 测量模型以及运动学模型，全面描述了其运动学特性。随后，结合多传感器数据融合技术，提出了一种数据融合模型，显著提升了定位精度和系统的鲁棒性。最后，通过在 MATLAB 中进行运动学仿真分析，详细验证了所推导的运动学模型、激光雷达测量模型、IMU 测量模型以及数据融合模型的正确性，并通过 Simulink 搭建了数据融合系统模型分析了该数据融合模型的性能，为后续算法的实现和实验分析奠定了坚实的基础。

第4章 基于EKF的DLO信息融合定位建图算法

4.1 引言

随着自动化技术的发展,堆垛式无人叉车在物流仓储中的应用日益广泛。为了提高其在复杂环境中的定位精度和稳定性,基于多传感器数据的信息融合成为当前研究的热点。本章针对单舵轮驱动的堆垛式无人叉车运动过程中产生的定位误差以及现有激光SLAM在复杂环境中的偏差问题,提出了一种基于EKF的DLO信息融合定位建图算法(EKF-DLO SLAM),融合激光雷达与IMU数据,并将EKF状态估计作为DLO SLAM输入,解决了单舵轮驱动的堆垛式无人叉车运动过程中产生的定位误差以及现有激光SLAM在复杂环境中的偏差问题,实现了精准定位与建图。此外,本文还实现了基于图优化的Cartographer算法,并通过实验验证其有效性。然而,DLO SLAM因缺乏后端优化,仍存在误差累积问题,需进一步改进。本研究为单舵轮驱动叉车的高精度定位与建图提供理论与技术支持。

4.2 堆垛式无人叉车的定位误差机理分析

4.2.1 堆垛式无人叉车运动模型误差机理分析

舵轮是无人叉车中常见的驱动和转向机构,其主要功能是通过旋转舵轮来控制叉车的转向。然而,舵轮的变形误差通常由舵轮的物理变形、摩擦、弹性变形等因素引起,这些误差会导致运动不准确,从而影响运动学模型的精度,进而影响定位和导航的准确性。因此,分析舵轮变形误差的机理对于优化运动模型和提高定位精度具有重要意义。

舵轮变形误差主要来源于以下几个方面:负载变化引起的机械结构弹性变形、轮胎磨损导致的摩擦力分布不均匀以及地面不平整造成的轮胎局部形变。在第三章中,单舵轮驱动的堆垛式无人叉车的运动模型是在理想状态下构建的,未考虑负载变化、轮胎磨损以及地面不平整等因素。然而,这些因素会导致舵轮产生弹性形变,从而影响定位精度。具体而言,弹性形变会导致IMU里程计的测量值与实际值之间产生误差,并且这一误差会随着叉车行驶的持续而逐步积累,最终

降低定位精度。因此，可以通过等效弹簧模型来表示舵轮的变形。图 4-1 展示了无人叉车行驶过程中舵轮形变的示意图。

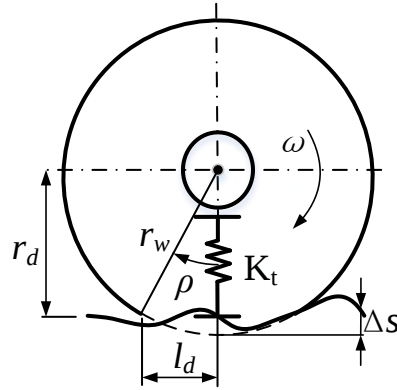


图 4-1 舵轮形变示意图

在图 4-1 中， l_d 为舵轮与地面接触纵向长度的一半； ρ 为舵轮中垂线与接触印记之间的夹角； r_w 为未发生弹性变形舵轮半径； r_d 为舵轮静态半径； Δs 为舵轮垂向变形量； ω 为舵轮的角速度； K_t 为等效弹簧的刚度。根据图 4-1 中的几何关系可以得出下列公式：

$$\begin{cases} r_d = r_w \cos \rho \\ l_d = r_w \sin \rho \\ F_t = K_t \Delta s \end{cases} \quad (4-1)$$

其中， F_t 为等效弹簧产生变形收到的合力。

舵轮的线速度与角速度见式 4-2 所示，其中 r_ε 为舵轮在行驶过程中的实际有效半径， t 是堆垛式无人叉车行驶时间。

$$\begin{cases} v = r_\varepsilon \omega = \frac{l_d}{t} \\ \omega = \frac{\rho}{t} \end{cases} \quad (4-2)$$

舵轮的静态半径 r_d 与未发生弹性变形的舵轮半径 r_w 之间的关系见式 4-3 所示。

$$r_d = r_w - \Delta s \quad (4-3)$$

由式 (4-1~4-3) 以及图 4-1 中的几何关系可以推导出舵轮在行驶过程中的实

实际有效半径 r_e 的表达式，见式 4-4 所示。

$$r_e = \frac{r_w \sin \rho}{\rho} = \frac{r_w \sin \left[\arcsin \left(1 - \frac{\Delta s}{r_w} \right) \right]}{\arcsin \left(1 - \frac{\Delta s}{r_w} \right)} \quad (4-4)$$

4.2.2 堆垛式无人叉车传感器模型误差机理分析

在本文的单舵轮驱动的堆垛式无人叉车自主定位与导航研究中，核心传感器为速腾 RS-LiDAR-16 激光雷达和维特 HWT605 九轴 IMU。由于传感器硬件和环境因素的影响，系统中不可避免地存在误差。因此，分析传感器误差的来源及其机理，对于提高 SLAM 系统的鲁棒性和精度至关重要。

1) 激光雷达误差机理分析

速腾 RS-LiDAR-16 是一个 16 线激光雷达，具有高精度的距离测量能力，并能够生成高密度的 3D 点云数据。其误差来源主要包括测距误差和角度误差两个方面，误差模型图如图 4-2 所示。

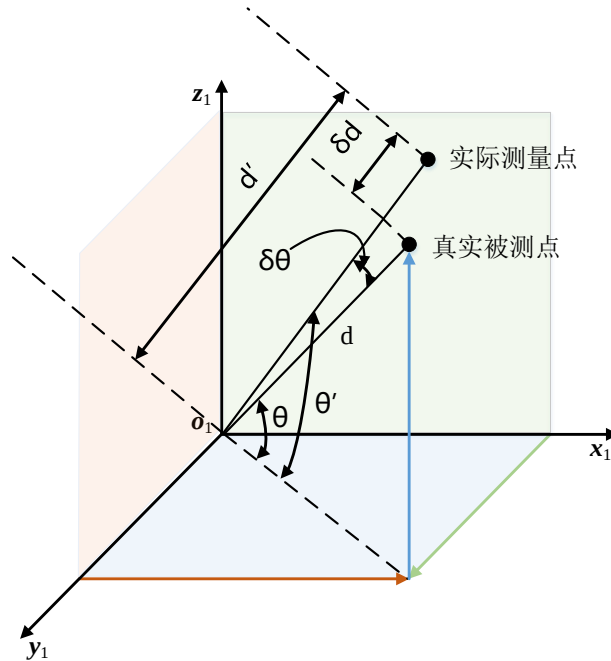


图 4-2 激光雷达误差模型

在图 4-2 中 d 为被测目标点距离激光雷达真实距离； d' 为实际被测距离； δd 为测距误差； θ 为被测目标点的真实扫描角度； θ' 为实际扫描角度； $\delta \theta$ 为角度

误差。其中测距误差和角度误差均可建模为高斯噪声。根据图 4-2 中的几何关系可以得出下列公式：

$$\begin{cases} d' = d + \delta d, \delta d \sim N(0, \sigma_d^2) \\ \theta' = \theta + \delta \theta, \delta \theta \sim N(0, \sigma_\theta^2) \end{cases} \quad (4-5)$$

2) IMU 误差机理分析

维特 HWT605 IMU 包括三轴加速度计、三轴陀螺仪和三轴磁力计。IMU 的误差模型主要由加速度计的零偏误差和噪声误差、陀螺仪的零偏误差和噪声误差，以及地磁场变化引起的磁力计误差所组成。

4.3 基于 EKF 的 DLO 信息融合定位建图算法

基于 EKF 的 DLO 信息融合定位建图算法的基本思路是直接利用激光雷达采集的点云数据以及惯性测量单元 (IMU) 提供的运动模型来进行机器人的位姿估计和环境建图，而不依赖于传统的特征提取与匹配方法。与传统 SLAM 方法（如基于视觉或基于特征的激光 SLAM）不同，基于 EKF 的 DLO 信息融合定位建图算法通过直接处理原始的激光雷达数据与 IMU 数据来提高精度和鲁棒性。基于 EKF 的 DLO 信息融合定位建图算法特别适合在复杂、动态、纹理不丰富的环境中使用。其系统框架如图 4-3 所示。

4.3.1 激光雷达与 IMU 数据预处理及数据融合

激光雷达点云在采集过程中受传感器采样延迟、噪声等因素影响，容易出现漂移、噪点和运动畸变等问题^[76]。为降低这些影响并提升建图效果，需要对点云数据进行预处理，包括运动去畸变和点云过滤等方法，以保证点云质量和精度满足复杂应用需求。经过预处理后，数据准确性得以提升，噪声干扰得到抑制，从而为后续精确的建图与定位提供可靠保障。

激光雷达依靠旋转扫描机制工作，需经历完整的扫描周期才能生成一帧完整点云。在运动过程中，不同点的测量位置不断变化，从而导致点云数据产生运动畸变^[77]。

在激光雷达进行扫描时，若雷达保持静止，其测得的物理距离应与实际距离相符。然而，在实际应用中，无人叉车通常处于运动过程中。本文所使用的速腾 RS-LiDAR-16 激光雷达的测量频率为 10 Hz，扫描时间间隔为 0.05 s 到 0.1 s。假

设无人叉车的运动速度为 0.3 m/s ，那么在扫描的开始和结束时，叉车的位姿差异可达到 3 cm 。在户外实验中，尤其是在无人叉车转弯或特征不明显的情况下，如果不对激光雷达扫描数据进行补偿，几乎每次扫描都会产生畸变。因此，针对这种情况，需要采取相应的补偿措施以保证数据的准确性。

本文的去畸变处理主要针对激光雷达自身运动引起的畸变效应，未考虑物体运动导致的失真。为补偿雷达运动畸变，需要将扫描点的位姿转换至正确的雷达坐标系（ L 系）。随后，结合 IMU 里程计数据，对雷达点云进行运动补偿，其中 IMU 里程计数据来源于第 3.4 节所述的数据融合模型。运动去畸变的示意图如图 4-3 所示，其中（a）展示了去畸变前的状态，（b）则为去畸变后的效果。图中，红色虚线代表激光雷达获取的点云数据，黑色实线则表示被测物体的轮廓信息。

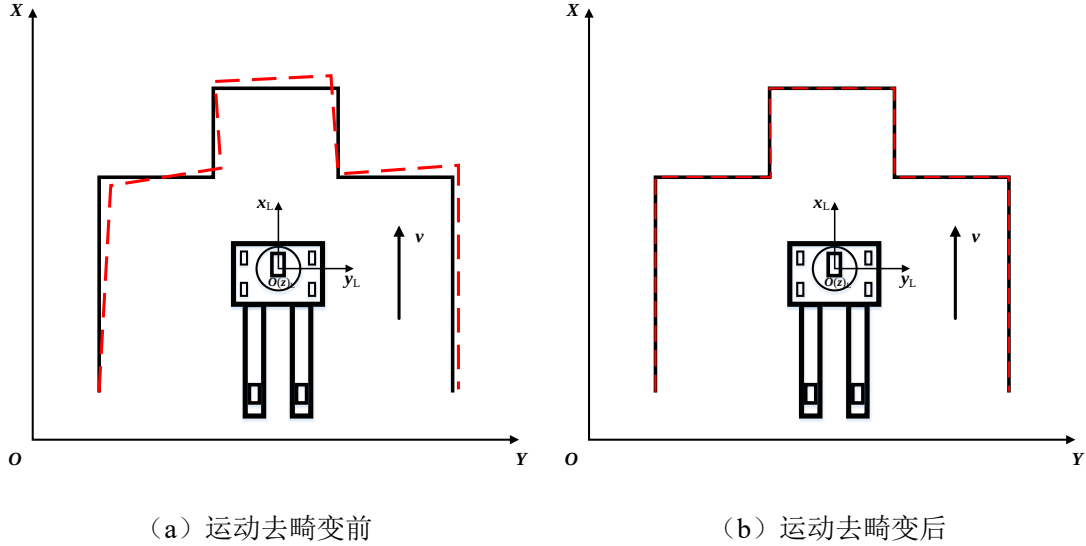


图 4-3 激光雷达点云运动去畸变示意图

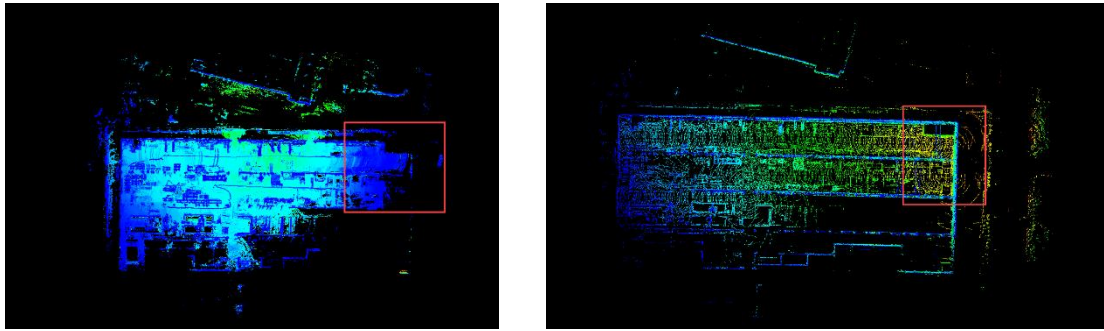
假设激光雷达的单次扫描时间为 t_s ，且激光雷达的初始扫描周期为 $P_t(x_{t_s}, y_{t_s}, z_{t_s})$ ，可以通过对数据融合模型中的位姿进行插值，得到每个雷达扫描点对应时间的测量位姿。以世界坐标系（ W 系）为点云转换的基准， t 时刻对应的位姿为 T_t^{WL} ，测量结束时刻的位姿为 T_{end}^{LW} ，IMU 与激光雷达的外参为 T_{LL} 。因此，激光雷达运动畸变补偿公式可以表示为：

$$P_{\text{output}} = T_{LL} T_{\text{end}}^{LW} T_t^{WL} T_{LL} P_t \quad (4-6)$$

式 4-6 中 P_t 为激光雷达输入点云； P_{output} 为激光雷达运动去畸变后的输出点

云。

激光雷达点云的运动畸变去除可分为以下几个步骤：首先，确定每个激光点在当前扫描周期内的具体测量时间；其次，计算激光雷达在本次扫描结束时的位姿；然后，利用插值方法推算扫描过程中雷达的运动轨迹；最后，将扫描点转换到世界坐标系下进行修正。补偿前后的效果如图 4-4 所示。在拐弯区域，由于特征点较少，理想情况下雷达点云应准确描绘出直角墙面的拐角。然而，图（a）中，最右侧的点云数据存在明显畸变，未能准确反映墙体特征。经过运动补偿后，图（b）中的拐角形态得到恢复，较好地还原了真实结构。



（a）去畸变前的点云地图

（b）去畸变后的点云地图

图 4-4 实际激光点云地图运动去畸变示意图

在对激光雷达点云数据进行运动去畸变后，还需进行点云过滤，以去除无效或冗余数据，降低点云密度，从而减少计算开销。结合实际应用需求与算法复杂性，本文采用体素滤波对原始点云数据进行优化处理^[78]。体素滤波通过将点云划分为三维体素网格，并在每个体素内选取一个代表点来代替该区域的数据，实现点云的降采样。三维体素网格划分的体素索引按照式 4-7 计算。

$$i = \left\lfloor \frac{x}{L_x} \right\rfloor, j = \left\lfloor \frac{y}{L_y} \right\rfloor, k = \left\lfloor \frac{z}{L_z} \right\rfloor \quad (4-7)$$

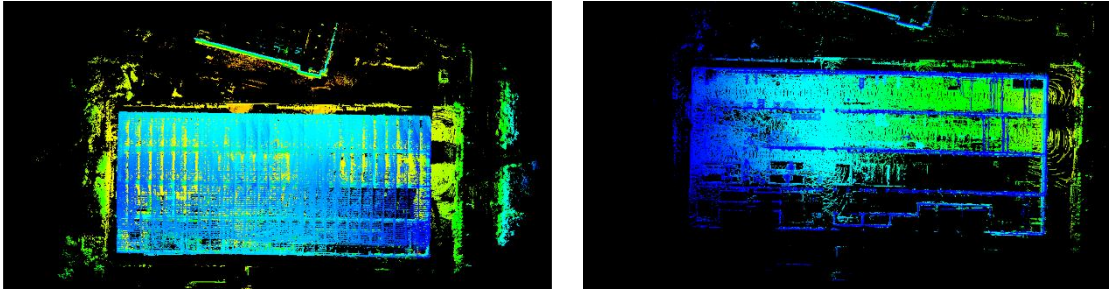
其中 (L_x, L_y, L_z) 为每个体素大小， (x, y, z) 为点云中的每个点， (i, j, k) 为每个点所对应的体素索引。

通过体素索引，将所有点分配到对应的体素中，因此每个体素内会包含多个点。接着，计算每个体素内所有点的质心，并将该质心作为该体素的代表点，用来代替该体素中的所有点，从而获得降采样后的点云。体素内所有点的质心按照式 4-8 进行计算。

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n, \bar{y} = \frac{1}{N} \sum_{n=1}^N y_n, \bar{z} = \frac{1}{N} \sum_{n=1}^N z_n \quad (4-8)$$

其中, N 为该体素内点的数量, (x_n, y_n, z_n) 是第 n 个点的坐标 p_n , $(\bar{x}, \bar{y}, \bar{z})$ 为质心坐标 $\bar{p}$ 。

如图 4-5 所示, 经过体素滤波处理后的点云地图相比于原始全采样数据, 冗余信息得到了有效削减, 同时噪声点也被大幅压缩。经过优化后的点云数据更加精简, 有助于提升后续点云匹配与建图的效率和精度。



(a) 全采样的点云地图

(b) 降采样后的点云地图

图 4-5 体素滤波降采样

IMU 数据预处理的具体过程详见第 5.4.1 节。在完成激光雷达和 IMU 数据的预处理后, 利用第 3.4 节所述的数据融合模型对两类传感器数据进行融合, 得到融合后的状态估计值, 并将其作为 DLO SLAM 算法的输入, 用于单舵轮驱动的堆垛式无人叉车工业环境下的定位与建图任务。

4.3.2 基于 GICP 扫描匹配

基于 LiDAR 的里程计可以看作是通过连续获取并匹配环境点云来估计机器人相对位移和旋转, 从而解决机器人自我运动的过程。该过程通常分为两个阶段: 首先, 通过连续扫描匹配来估计短时间内的相对位姿; 其次, 通过将当前扫描与全局地图进行匹配, 以修正位置并提高全局一致性。

1) Scan-to-Scan: 在第一阶段, scan-to-scan 的匹配目标是计算在雷达坐标系 L 中 k 时刻的位姿 p_k 和 $k-1$ 时刻的位姿 p_{k-1} 之间的相对变换 X_k^L :

$$X_k^L = \arg \min_{X_k^L} \varepsilon(X_k^L p_k, p_{k-1}) \quad (4-9)$$

其中 GICP (广义最小二乘法, generalized iterative closest point) 的残差 ε 定

义为:

$$\varepsilon(\mathbf{X}_k^L \mathbf{p}_k, \mathbf{p}_{k-1}) = \sum_i^N \mathbf{d}_i^T \left(\mathbf{C}_{k-1,i} + \mathbf{X}_k^L \mathbf{C}_{k,i} (\mathbf{X}_k^L)^T \right)^{-1} \mathbf{d}_i \quad (4-10)$$

定义协方差矩阵 $\mathbf{C}_{k,i}$:

$$\mathbf{C}_{k,i} = \sum_{i=1}^N \mathbf{p}'_{k,i} (\mathbf{p}'_{k,i})^T \quad (4-11)$$

这一阶段的最终目标函数是:

$$\mathbf{X}_k^L = \arg \min_{\mathbf{X}_k^L} \sum_i^N \mathbf{d}_i^T \left(\sum_{i=1}^N \mathbf{p}'_{k-1,i} (\mathbf{p}'_{k-1,i})^T + \mathbf{X}_k^L \sum_{i=1}^N \mathbf{p}'_{k,i} (\mathbf{p}'_{k,i})^T (\mathbf{X}_k^L)^T \right)^{-1} \mathbf{d}_i \quad (4-12)$$

其中, $\mathbf{d}_i$ 表示平移向量 $\mathbf{d}_i = \bar{\mathbf{p}}_{k-1,i} - \mathbf{X}_k^L \bar{\mathbf{p}}_{k,i}$; $\mathbf{p}'_{k,i}$ 表示去质心的点 $\mathbf{p}'_{k,i} = \mathbf{p}_{k,i} - \bar{\mathbf{p}}_{k,i}$ 。在第 4.3.3 节中进一步分析了用外部传感器 IMU 提供的先验信息初始化上述目标函数, 即通过 IMU 预积分可以得到一个先验的 $\mathbf{X}_k^R$, 可以设置一个初始设想 $\mathbf{X}_k^L = \mathbf{X}_k^R$ 来建立一个松耦合系统; 倘若先验信息不可用, 则系统仅依赖点云进行配准。

2) Scan-to-Map: 在推导出机器人的初始位姿估计后执行 scan-to-map 的第二阶段, 执行过程与 scan-to-scan 相类似。然而, 此时的目标并不是计算两个瞬时点云之间的相对变换, 而是通过与局部子图匹配的方式, 进一步改进前一步的运动估计, 使其更加全局一致。这里的任务即计算当前坐标在世界坐标系 W 下的相对位姿 $\mathbf{X}_k^W$:

$$\mathbf{X}_k^W = \arg \min_{\mathbf{X}_k^W} \varepsilon(\mathbf{X}_k^W \mathbf{p}_k, \mathbf{p}_{k-1}) \quad (4-13)$$

对 GICP 的残差定义与式 4-10 相似, 则这一阶段的总体目标函数是:

$$\mathbf{X}_k^W = \arg \min_{\mathbf{X}_k^W} \sum_j^N \mathbf{d}_j^T \left(\sum_{j=1}^N \mathbf{p}'_{k-1,j} (\mathbf{p}'_{k-1,j})^T + \mathbf{X}_k^W \sum_{j=1}^N \mathbf{p}'_{k,j} (\mathbf{p}'_{k,j})^T (\mathbf{X}_k^W)^T \right)^{-1} \mathbf{d}_j \quad (4-14)$$

更新位姿 $\mathbf{X}_{k+1}^W$:

$$\mathbf{X}_{k+1}^W = \mathbf{X}_k^W + \Delta \mathbf{X} \quad (4-15)$$

计算位姿增量 ΔX :

$$J^T J \Delta X = -J^T e_i \quad (4-16)$$

这一阶段最终的目标函数是:

$$X_{k+1}^W = \arg \min_{X_k^W} \sum_j^N d_j^T \left(\sum_{j=1}^N p'_{k-1,j} (p'_{k-1,j})^T + X_k^L \sum_{j=1}^N p'_{k,j} (p'_{k,j})^T (X_k^W)^T \right)^{-1} d_j + \Delta X \quad (4-17)$$

其中, e_i 为误差向量 $e_i = X_k^W p_{k,i} s_i - m_i$, s_i 为扫描中的第 i 个点, m_i 为地图中

与 s_i 相对应的点; J 为雅可比矩阵 $J = \frac{\partial e_i}{\partial X_k^W p_{k,i}}$ 。

4.3.3 先验估计

式 4-9 通过 GICP 解决了 scan-to-scan 的非线性优化问题, 在这个过程中可以使用 IMU 数据进行先验估计以减少局部最优解问题。该先验估计通过 IMU 提供的加速度和角速度来故居机器人短时间内的运动状态从而作为 SLAM 过程中的先验信息。

利用 IMU 数据进行先验估计在数据预处理之后首先对数据进行检测判定数据是否可用于先验估计, 此过程主要分为两个阶段: 第一阶段为数据质量检查, 这一阶段主要检查加速度、角速度和噪声水平, 通过设定加速度和角速度的范围阈值检测每个分量是否在预定范围内, 设定噪声水平阈值, 计算加速度和角速度分量的标准差判断是否在阈值范围内; 第二阶段为数据一致性检查, 这一阶段主要检查轨迹平滑性, 判断 IMU 数据轨迹是否平滑, 避免异常跳变, 通过轨迹变化率和平滑度阈值来判定结果。当 IMU 数据完成这两个阶段的检测则判断可用于先验估计。

先验估计的匹配目标是将 IMU 预积分结果作为先验估计结合 scan-to-scan 的位置预测采用因子图优化获取一个新位姿预测结果 X_k^R :

$$X_k^R = \arg \min_{X_k^R} \sum_{i,j} \|r_{ij}\|^2 \quad (4-18)$$

定义因子图优化的最小化残差 r_{ij} :

$$\mathbf{r}_{k,ij}^R = \mathbf{h}\left(\mathbf{X}_k^L, \mathbf{u}_{k,ij}^R\right) - \mathbf{h}\left(\mathbf{X}_k^L\right) \quad (4-19)$$

定义 IMU 的预积分结果 $\mathbf{u}_{ij}$:

$$\mathbf{u}_{k,ij}^R = \left[\Delta \mathbf{s}_{k,ij}^R, \Delta \mathbf{v}_{k,ij}^R, \Delta \mathbf{R}_{k,ij}^R \right] \quad (4-20)$$

其中 $\mathbf{h}(\cdot)$ 表示预期观测模型； $\Delta \mathbf{s}_{k,ij}^R$ 是 IMU 数据位移预积分； $\Delta \mathbf{v}_{k,ij}^R$ 是 IMU 数据速度预积分； $\Delta \mathbf{R}_{k,ij}^R$ 是 IMU 数据姿态预积分。 $\Delta \mathbf{s}_{k,ij}^R$ ， $\Delta \mathbf{v}_{k,ij}^R$ ， $\Delta \mathbf{R}_{k,ij}^R$ 的计算公式见式 4-21 所示。

$$\begin{cases} \Delta \mathbf{s}_{k,ij}^R = \sum_{k=i}^{j-1} \mathbf{v}_k^R \Delta t_k^R + \frac{1}{2} \mathbf{g} \left(\sum_{k=i}^{j-1} \Delta t_k^R \right)^2 \\ \Delta \mathbf{v}_{k,ij}^R = \sum_{k=i}^{j-1} \mathbf{R}_k^R \left(\mathbf{a}_k^R \Delta t_k^R \right) + \mathbf{g} \sum_{k=i}^{j-1} \Delta t_k^R \\ \Delta \mathbf{R}_{k,ij}^R = \prod_{k=i}^{j-1} \exp\left(\boldsymbol{\omega}_k^R \Delta t_k^R\right) \end{cases} \quad (4-21)$$

这一阶段的最终目标函数 $\mathbf{X}_k^R$:

$$\mathbf{X}_k^R = \arg \min_{\mathbf{X}_k^R} \sum_{i,j} \left\| \mathbf{h}\left(\mathbf{X}_k^L, \left[\Delta \mathbf{s}_{k,ij}^R, \Delta \mathbf{v}_{k,ij}^R, \Delta \mathbf{R}_{k,ij}^R \right] \right) - \mathbf{h}\left(\mathbf{X}_k^L\right) \right\|^2 \quad (4-22)$$

4.4 基于图优化的 Cartographer 定位建图算法

Cartographer 算法采用了前端局部建图与后端闭环检测优化全局地图的架构。前端通过局部子图（submap）逐步构建完整地图（如图 4-6 所示），多个激光雷达扫描帧（scan）被整合为一个子图，所有子图最终形成全局地图。当某个子图不再接收新的雷达帧时，意味着其构建完成，系统随即生成新的子图，最终所有局部子图共同构成完整的全局地图^[79]。

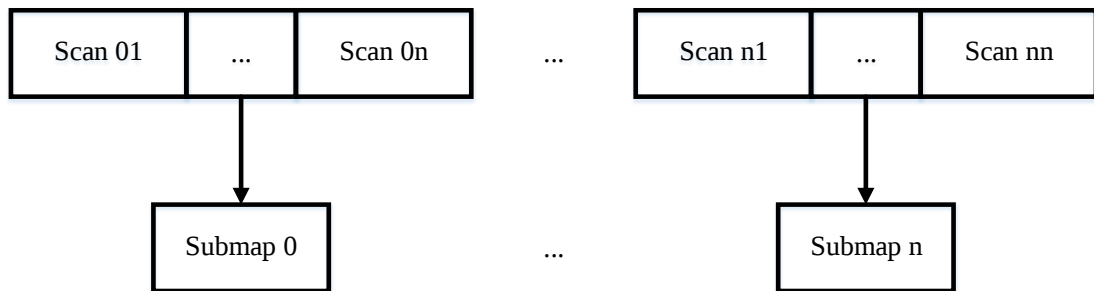


图 4-6 Cartographer 建图概述

每个雷达扫描帧都包含全局地图坐标系下的全局坐标信息，同时也包含局部子图坐标系下的局部坐标信息。在构建局部子图时，需要进行多个坐标系之间的坐标变换。无人叉车的位姿坐标表示为 $\xi=[x_t \ y_t \ z_t \ \theta_t \ \phi_t \ \psi_t]$ ，激光雷达观测到的点云 $\{p_i\}$ 是世界坐标系中的点，无人叉车通过变换得到这些点相对于自身坐标系的坐标。对于一个特定的点 p_i （在世界坐标系中的位置），如果无人叉车位于 ξ 位置，并且姿态为 $[\theta_t \ \phi_t \ \psi_t]$ ，那么这个点相对于无人叉车坐标系的位置是：

$$p_i = R_t^{-1} p_i^W + t_t \quad (4-23)$$

其中， R 是式 3-4 中从世界坐标系到机器人坐标系到世界坐标系的旋转矩阵， t_t 是无人叉车位姿的平移部分。

测量的距离 d_i 就是从机器人到点云中的点 p_i 的距离：

$$d_i = \|p_i^R - \xi_t\| \quad (4-24)$$

其中， p_i^R 是在无人叉车坐标系下的测量点。为了消除误差，假设测量存在一定的噪声 ε_i ，因此激光雷达的测量可以表示为：

$$z_i = h(x_t, p_i^W) + \varepsilon_i \quad (4-25)$$

这里， $h(x_t, p_i^W)$ 是激光雷达的观测函数， $\varepsilon_i \sim N(0, \Sigma)$ 是一个零均值高斯噪声，且协方差矩阵为 Σ 。

为了估计机器人的位姿，Cartographer 使用基于 ICP (Iterative Closest Point) 算法来进行位姿的局部优化。假设当前时刻的点云 $P=\{p_i\}$ 和参考帧的点云 $Q=\{q_i\}$ 。ICP 的目标是最小化两个点云之间的距离：

$$E(T) = \sum_{i=1}^N \|T \cdot p_i - q_i\|^2 \quad (4-26)$$

其中， T 是从当前点云到参考点云的变换矩阵， p_i 是当前点云的一个点， q_i 是参考点云的一个点， $\|T \cdot p_i - q_i\|^2$ 是欧几里得距离。最小化这个误差函数可以获得最佳的变换矩阵 T 。

Cartographer 的后端部分使用图优化方法来全局优化机器人在整个路径中的

位姿。在位姿图中，节点表示无人叉车的位姿 $\mathbf{x}_i$ ，每一条边表示两个相邻节点之间的相对约束。假设图中有 N 个节点，表示 N 个时刻的无人叉车位姿。每一条边 $\mathbf{x}_{ij}$ 表示位姿 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 间的约束。位姿图中的每一条边可以用以下误差模型表示：

$$\mathbf{e}_{ij} = \mathbf{x}_j - \mathbf{T}_{ij}(\mathbf{x}_i) \quad (4-27)$$

图优化的目标是最小化所有边的误差，具体目标函数为：

$$\min_{(\mathbf{x}_i)} \sum_{(i,j) \in E} \|\mathbf{e}_{ij}\|_{\Sigma_{ij}^{-1}}^2 \quad (4-28)$$

其中， E 表示所有边的集合， Σ_{ij}^{-1} 是边的协方差矩阵，代表了每条边的权重。

根据式 4-27 的误差模型构建雅可比矩阵 $\mathbf{J}$ ，并更新位姿 $\mathbf{x}_i$ ：

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} - (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{e} \quad (4-29)$$

最后进行回环检测，当检测到回环时，Cartographer 会更新位姿图中的边，并通过图优化算法调整所有位姿节点，以最小化回环误差。回环误差的优化目标是 minimize 回环检测引入的误差：

$$\min_{(\mathbf{x}_i)} \sum_{(i,j) \in E_{\text{loop}}} \|\mathbf{e}_{ij}\|_{\Sigma_{ij}^{-1}}^2 \quad (4-30)$$

其中， E_{loop} 表示所有回环检测的边。

4.5 实验分析

本实验旨在通过单舵轮驱动的堆垛式无人叉车在复杂工业环境中的建图定位实验，评估基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO）和基于图优化的 Cartographer 定位建图算法的地图构建性能。为确保实验结果的公平性，首先需要对基于 EKF 的 DLO 信息融合定位建图算法和基于图优化的 Cartographer 定位建图算法的功能包进行必要的参数设置，确保两种算法所需的传感器数据规格相同，并能够顺利录入。接着，启动相关包，并通过键盘操作无人叉车在室内环境中进行运动。最后，确保实验平台的一致性。

实验设备基于实验室自研的单舵轮驱动的堆垛式无人叉车平台（如图 4-7 所示），主控系统采用研华 EI-52 工控机，配备 Intel Core i5-1135G7 和 Celeron 6305E 处理器，具备低功耗与高性能特点，四核八线程，最大睿频可达 4.2 GHz，运行 ROS 机器人操作系统。工业场地环境数据以及无人叉车坐标信息通过 16 线激光

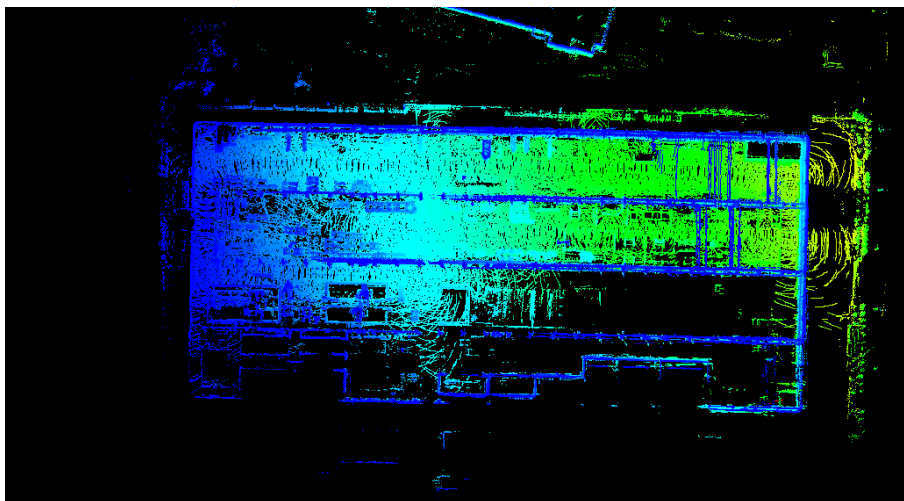
雷达和九轴 IMU 进行采集。激光雷达选用了稳定性较高的速腾 RS-LiDAR-16，其测量范围为 40 cm 至 150 m，扫描范围为 360°，分辨率为 0.4°，最高采样频率可达 10 Hz。惯性测量单元 IMU 采用维特 HWT605 九轴 IMU，其姿态测量精度为 0.05°，通信距离可达 300 m，最高输出频率为 200 Hz。



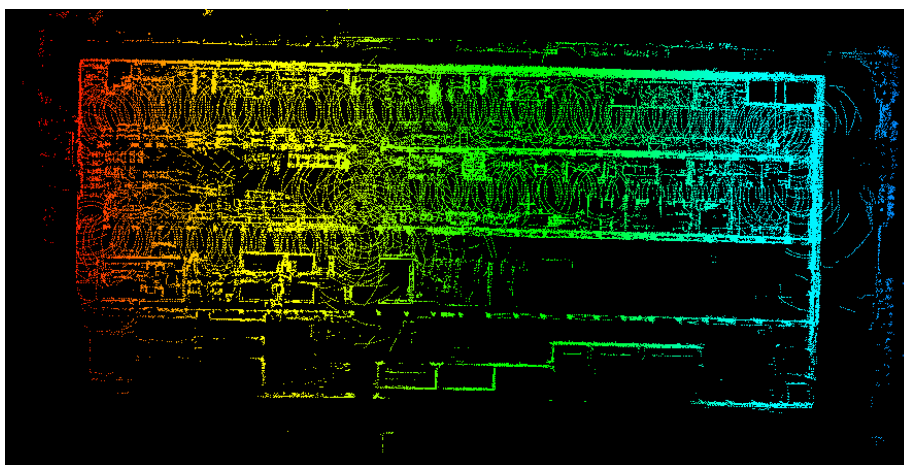
图 4-7 实验场景及设备

4.5.1 地图构建及性能分析

在实验室场景中，分别使用单舵轮驱动的堆垛式无人叉车采用基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO）和基于图优化的 Cartographer 定位建图算法进行实验场景的地图构建。实验结果表明，基于 EKF 的 DLO 信息融合定位建图算法构建的地图与基于图优化的 Cartographer 定位建图算法构建的地图差异较小，如图 4-8 所示。通过单舵轮驱动的堆垛式无人叉车在室内简单场景中应用基于 EKF 的 DLO 信息融合定位建图算法构建的环境地图，结果显示其清晰度良好，与实际环境差距较小，且环境中的障碍物能够有效显示，如图 4-9（a）所示。然而，坐标偏移问题较为明显，实际轨迹呈环形，而实验轨迹与实际轨迹（红线）并未完全重合，偏差较大，如图 4-9（b）所示。

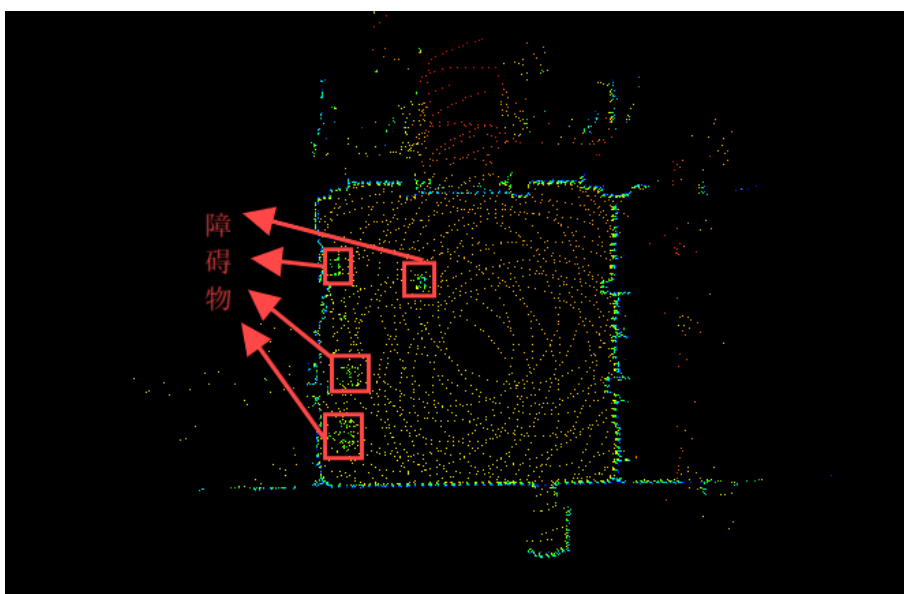


(a) 基于 EKF 的 DLO 信息融合定位建图算法的建图效果

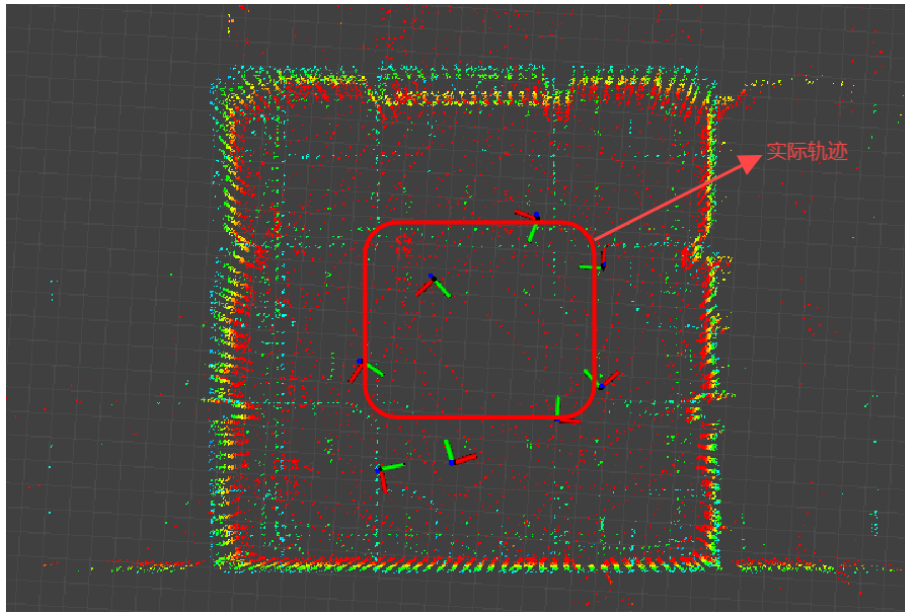


(b) 基于图优化的 Cartographer 定位建图算法的建图效果

图 4-8 两种算法的建图效果



(a) 基于 EKF 的 DLO 信息融合定位建图算法的室内简单场景建图效果



(b) 基于 EKF 的 DLO 信息融合定位建图算法建立室内简单环境地图的坐标偏移

图 4-9 基于 EKF 的 DLO 信息融合定位建图算法的建图效果

在使用基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO）进行单舵轮驱动的堆垛式无人叉车的定位建图时，计算机资源的使用率明显低于基于图优化的 Cartographer 定位建图算法，如图 4-10 所示。由于基于图优化的 Cartographer 算法对计算机资源的需求较高，导致建图过程中可视化出现较大延迟，从而影响实验人员对细节的实时观察。而基于 EKF 的 DLO 信息融合定位建图算法则有效避免了这一问题，确保了实验过程更加顺畅，如图 4-11 所示。

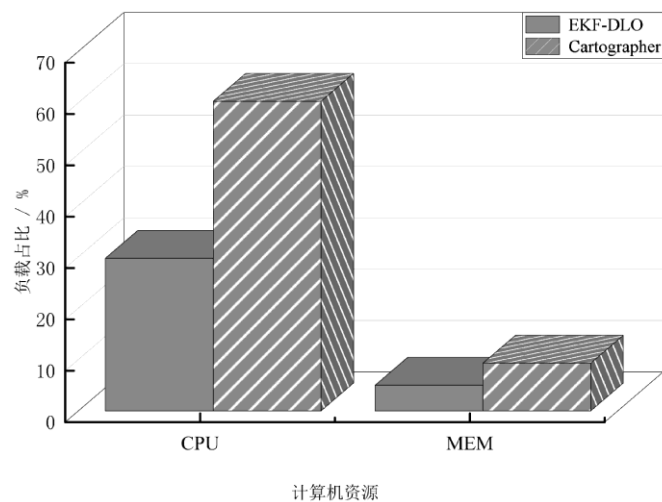
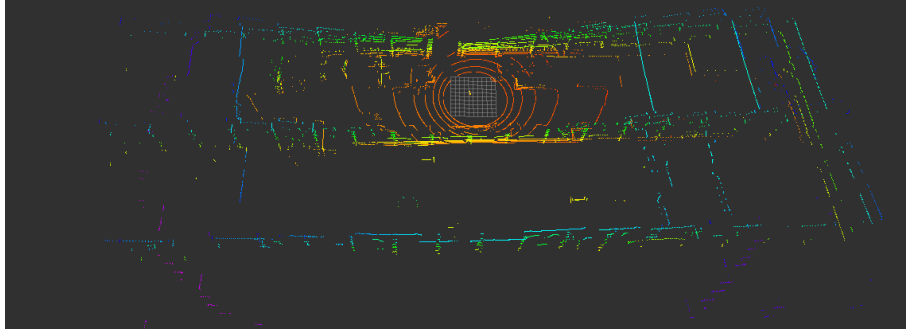
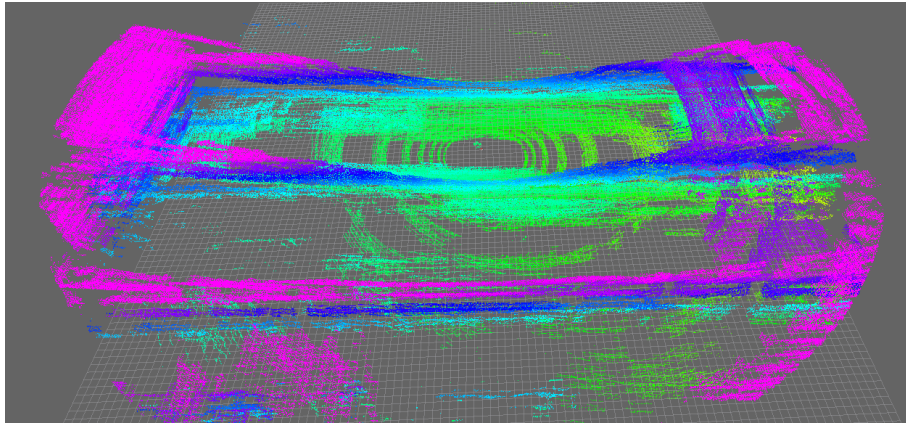


图 4-10 两种算法的计算机资源使用率



(a) 基于 EKF 的 DLO 信息融合定位建图算法启动效果

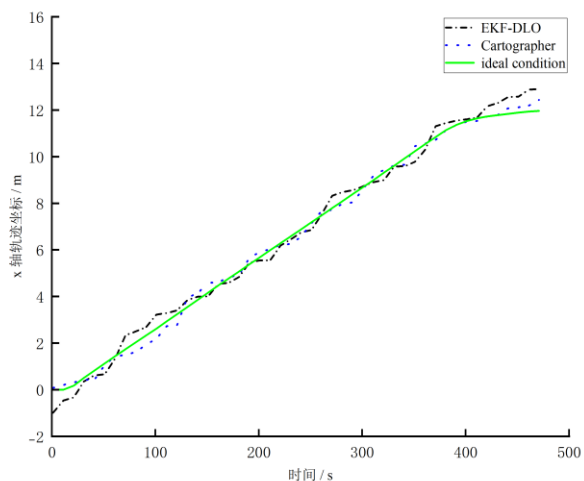


(b) 基于图优化的 Cartographer 定位建图算法启动效果

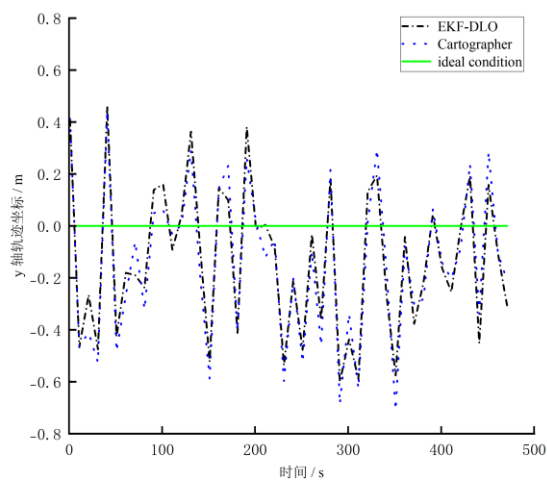
图 4-11 两种算法的启动效果

4.5.2 轨迹误差分析

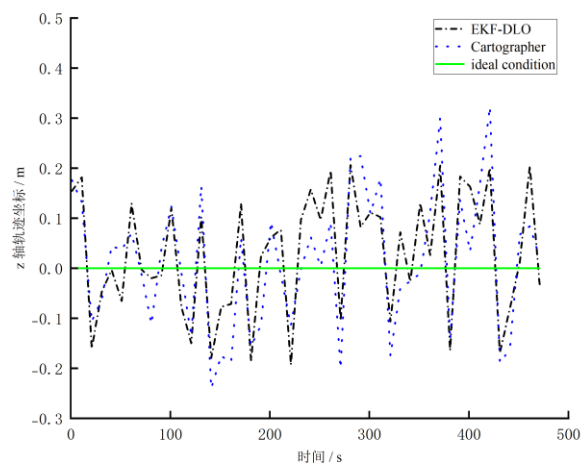
在室内简单环境下，分别使用单舵轮驱动的堆垛式无人叉车采用基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO）和基于图优化的 Cartographer 定位建图算法进行建图，并完成实际直线轨迹运动规划对比两种算法在实际运动过程中的轨迹误差。结果表明，基于 EKF 的 DLO 信息融合定位建图算法与基于图优化的 Cartographer 定位建图算法的轨迹误差差异不大，但仍可以观察到，基于 EKF 的 DLO 信息融合定位建图算法在 x 轴、y 轴和 z 轴方向的运动误差相较于基于图优化的 Cartographer 定位建图算法有所改善。如图 4-12 所示。



(a) x 轴轨迹误差



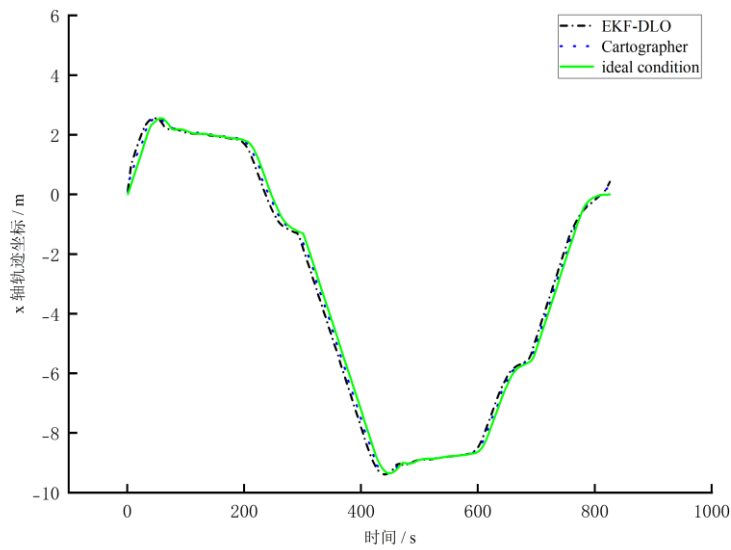
(b) y 轴轨迹误差



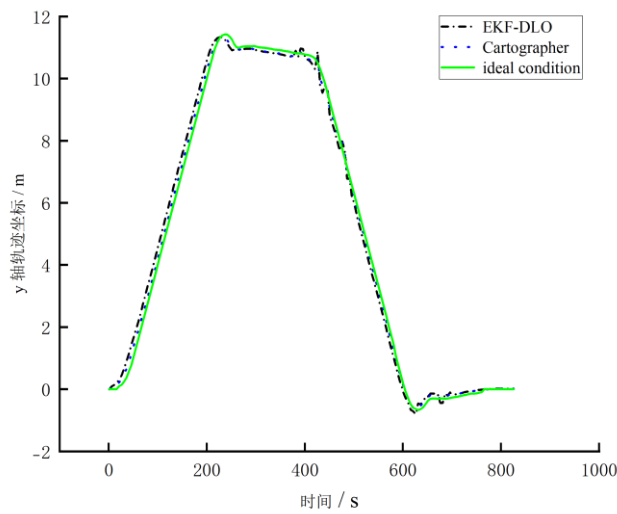
(c) z 轴轨迹误差

图 4-12 室内简单环境下两种算法直线行驶的轨迹误差对比图

在室内简单环境下，分别使用单舵轮驱动的堆垛式无人叉车采用基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO）和基于图优化的 Cartographer 定位建图算法进行建图，并完成实际环形轨迹运动规划对比两种算法在实际运动过程中的轨迹误差。结果表明，基于 EKF 的 DLO 信息融合定位建图算法与基于图优化的 Cartographer 定位建图算法的轨迹误差差异不大，但仍可以观察到，基于 EKF 的 DLO 信息融合定位建图算法在 x 轴、 y 轴和 z 轴方向的运动误差相较于基于图优化的 Cartographer 定位建图算法有所改善。如图 4-13 所示。



(a) x 轴轨迹误差



(b) y 轴轨迹误差

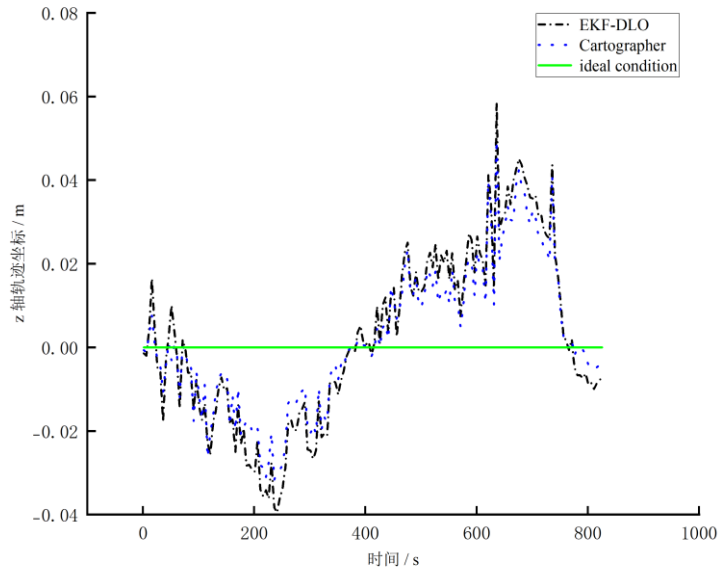
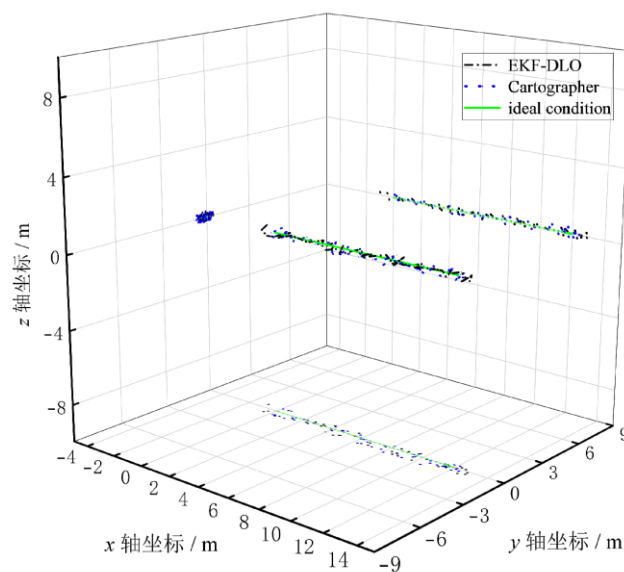
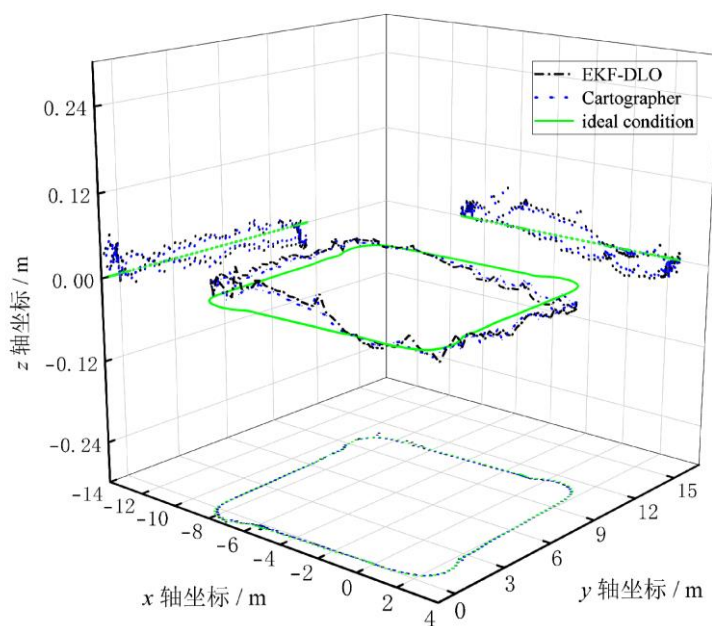
(c) z 轴轨迹误差

图 4-13 室内简单环境下两种算法环形的轨迹误差对比图

利用单舵轮驱动的堆垛式无人叉车分别采用基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO）和基于图优化的 Cartographer 定位建图算法进行建图，并完成实际轨迹运动规划。对应的运动轨迹图如图 4-14 所示。从图中可以看出，基于 EKF 的 DLO 信息融合定位建图算法与基于图优化的 Cartographer 定位建图算法的运动轨迹差异较小，但仍可以观察到，基于 EKF 的 DLO 算法在运动轨迹上相较于 Cartographer 算法有所改善。



(a) 室内简单环境直线运动轨迹图



(b) 室内简单环境环形运动轨迹图

图 4-14 室内简单环境下两种算法的运动轨迹图

4.5.3 稳定性分析

在室内简单实验场景中利用单舵轮驱动的堆垛式无人叉车分别采用基于 EKF 的 DLO 信息融合定位建图算法 (EKF-DLO) 和基于图优化的 Cartographer 定位建图算法进行建图, 并完成实际直线以及环形轨迹运动规划叉车行驶路线上各轨迹点与理想状态下各轨迹点之间距离的方差和标准差如表 4-1 所示, 从表中可以看出, 基于 EKF 的 DLO 信息融合定位建图算法完成实际直线以及环形轨迹运动时的方差和标准差要略小于基于图优化的 Cartographer 定位建图算法完成实际直线以及环形轨迹运动时的方差和标准差, 因此基于 EKF 的 DLO 信息融合定位建图算法的精度和稳定性相对较好。

表 4-1 轨迹点误差的方差和标准差

性能 算法	室内直线轨迹		室内环形轨迹	
	方差/ m^2	标准差/m	方差/ m^2	标准差/m
Cartographer	0.02805	0.16748	0.07297	0.25157
EKF-DLO SLAM	0.02724	0.16505	0.06042	0.24580

4.6 本章小结

本章深入分析了单舵轮驱动的堆垛式无人叉车的定位误差机理,并针对误差来源提出了一种基于 EKF 的 DLO 信息融合定位建图算法。首先,从运动模型和传感器模型两个角度探讨了堆垛式无人叉车定位系统中的误差来源。然后,针对激光雷达和 IMU 的数据进行预处理,并采用 EKF 方法将两者数据进行融合,利用融合后的数据作为 DLO SLAM 的输入,结合扫描匹配和先验估计进行定位建图,从而优化定位精度和建图效果。同时,介绍了基于图优化的 Cartographer 定位建图算法,该算法同样能够实现较为优秀的定位与建图效果。最后,通过实验验证了所提出方法的可行性,并探索了其存在的不足之处。实验结果表明,基于 EKF 的 DLO 信息融合定位建图算法在单舵轮驱动的堆垛式无人叉车的定位与建图中取得了较好的效果,但仍存在一定的不足,值得进一步优化。综上所述,本章提供的定位与建图算法为单舵轮驱动的堆垛式无人叉车的高精度定位提供了有效的技术支持,并为实际应用提供了理论依据和解决方案。

第 5 章 基于 DLO-HDL 的信息融合定位建图算法

5.1 引言

本章在第四章提出的 EKF-DLO SLAM 算法的基础上，引入 HDL-Graph-SLAM 的后端图优化与回环检测技术，提出了一种 DLO-HDL SLAM 算法，以解决 EKF-DLO SLAM 算法由于缺乏后端优化而产生的位姿漂移与误差累积问题。该算法结合了 EKF-DLO 的实时高精度定位与 HDL-Graph-SLAM 的后端图优化和回环检测技术，提升定位精度与建图鲁棒性。EKF-DLO 融合激光雷达与 IMU 数据进行里程计估计，而 HDL-Graph-SLAM 优化路径并消除累计误差，确保地图精度。本章将深入分析该信息融合与后端优化策略，并通过实验验证其在复杂环境中的有效性。

5.2 基于 EKF 的 DLO 信息融合定位建图误差机理分析

本节主要分析了基于 EKF 的 DLO 信息融合定位建图方法中的误差机理。该方法通过 EKF 算法将 IMU（惯性测量单元）和 LiDAR（激光雷达）传感器数据进行融合，但由于传感器噪声、系统漂移以及测量误差的累积，前端建图过程中产生的误差会逐渐放大，从而影响最终的建图精度。本方法采用的数据融合模型参见 3.4 节所示。在系统的状态更新过程中，存在一个过程噪声 ω_k ，而激光雷达的观测模型中也存在观测噪声 $V_L(t)$ 。滤波系统通过最小化这些噪声误差来进行状态估计。由于基于 EKF 的 DLO 信息融合定位建图方法并不包括后端优化过程，因此 IMU 的长期漂移和激光雷达的测量误差会在前端建图过程中逐渐积累，从而影响建图的精度。

假设前端误差为 τ_k ，表示当前估计的位姿和实际位姿之间的差异，随着时间的推移，误差会根据系统模型逐步增大，通常表现为如下关系：

$$\tau_k = \tau_{k-1} + \Delta\epsilon_k \quad (5-1)$$

其中， $\Delta\epsilon_k$ 是在第 k 步更新过程中产生的误差。根据误差传播的理论，当滤

波过程中的噪声（如 IMU 漂移、LiDAR 测量误差等）不断累积时，定位误差 τ_k 会呈现线性或非线性的增长，具体表现为：

$$\tau_k = \sum_{i=1}^k \mathbf{Q}_i \quad (5-2)$$

其中， $\mathbf{Q}_i$ 是第 i 次测量误差或漂移累积项，通常是与传感器精度和滤波算法中的估计误差有关的矩阵。随着时间的推移，误差的累积导致定位精度降低，进而影响建图精度。

随着前端定位误差的积累，建图的精度会下降。考虑到前端的定位误差在地图中的传播，可以用误差传播矩阵来描述该影响。假设误差矩阵为 $\mathbf{E}_k$ ，则随着时间的推移，误差矩阵会不断增长，影响到最终生成的地图的精度。误差传播通常表现为：

$$\mathbf{E}_k = \mathbf{F}_k \mathbf{E}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k \quad (5-3)$$

其中， $\mathbf{F}_k$ 是状态转移矩阵，根据 3.4 节中的式 3-41 计算得到，具体推导见式 5-4 所示； $\mathbf{Q}_k$ 是过程噪声矩阵，描述了每一帧数据对系统精度的贡献。随着误差的积累，最终的地图中可能会出现明显的偏差。

$$\mathbf{F}_k = \frac{\partial f(\mathbf{x}(k-1), \mathbf{u}(k))}{\partial \mathbf{x}(k-1)} \quad (5-4)$$

本节详细分析了基于 EKF 的 DLO 信息融合定位建图中误差的来源和传播过程，探讨了 IMU、LiDAR 以及滤波算法的误差特性及其对建图精度的影响。这为本章提出的基于 DLO-HDL 的信息融合定位建图算法引入后端优化及后续实验分析提供了理论支持。

5.3 基于 DLO-HDL 的信息融合定位建图算法系统设计

5.3.1 系统框架

本系统在第四章提出的基于 EKF 的 DLO 信息融合定位建图算法的基础上，进一步结合 HDL-Graph-SLAM 后端的回环检测与位姿图优化，解决了基于 EKF 的 DLO 信息融合定位建图算法在位姿漂移和误差累积方面的不足。系统整体分为七个部分：1) 数据预处理：对激光雷达原始点云进行特征分割、滤波、降噪

等处理，提取特征点；同时，利用 IMU 预积分得到的相对运动估计对特征点进行畸变校正；2) 激光里程计：将连续时刻的特征点进行配准，估计激光雷达的相对运动，进而估算初始位姿；3) 姿态优化：使用 IMU 数据辅助位姿估计，结合滑动窗口优化方法实时校正当前位姿，提升配准的鲁棒性与准确性；4) 局部地图构建：将经过优化后的当前帧点云与已有的局部地图进行整合，生成更新后的局部环境表示，为后端提供输入；5) 回环检测：通过将当前帧点云与历史点云进行匹配，识别可能的闭环区域，若检测到闭环关系，则生成一个约束因子，表示当前帧与历史帧之间的几何关系；6) 全局点云配准：考虑闭环约束，进行全局优化配准，减小累计误差；7) 因子图优化：构建因子图，将 IMU 信息、闭环因子、位姿估计等作为节点，优化整个位姿图。最终，基于优化后的全局位姿调整点云坐标，生成精确的全局环境地图，并提供准确的位姿信息，支持后续的导航或定位任务。系统框图如图 5-1 所示。

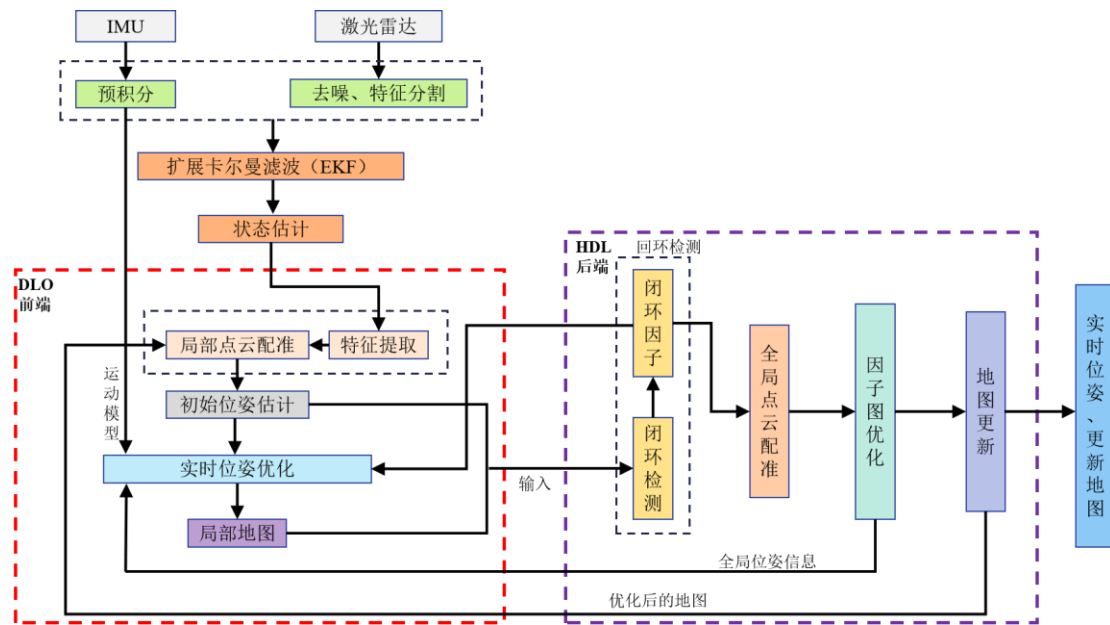


图 5-1 DLO-HDL SLAM 系统框图

5.3.2 算法流程

本章基于 DLO-HDL 的信息融合定位建图算法的流程图如下。其中，点云 p 由笛卡尔坐标 $p_i \in \mathbb{R}^3$ 的点集合 $p \in P$ 组成。将 X_k^w 定义为相邻点云在 k 时刻扫描 p_k 和 p_{k+1} 时估计的单舵轮驱动的堆垛式无人叉车当前位姿； M_k 定义为 k 时刻单舵轮驱动的堆垛式无人叉车重新构建的地图， L 定义为激光雷达的坐标系， B 定

义为堆垛式无人叉车位于 IMU 坐标下的坐标系， $\mathbf{W}$ 定义为与 $\mathbf{B}$ 在初始位置重合的世界坐标系。子映射、协方差和树结构分别定义为 S 、 C 、 T 。坐标系采用 $\mathbf{x}$ 向前， $\mathbf{y}$ 向左， $\mathbf{z}$ 向上的标准定义。

Algorithm Direct LiDAR Odometry with HDL Graph SLAM Backend

```

1: Input:  $\mathbf{P}_k, \mathbf{X}_{k-1}^{\mathbf{W}}$ 
2: Initialize:  $\mathbf{X}_{k-1}^{\mathbf{W}} \leftarrow \mathbf{I}$  or gravity Align()
3: Output:  $\mathbf{X}_k^{\mathbf{W}}, \mathbf{M}_k$  ▷ 预处理
4: while  $\mathbf{P}_k \neq \emptyset$  do
5:    $\mathbf{P}_k \leftarrow \text{preprocessPoints}(\mathbf{P}_k)$ 
6:    $\text{computeAdaptiveParameters}(\mathbf{P}_k)$  ▷ 初始化
7:   if  $k = 0$  then
8:      $\mathbf{T}_{k,1}^L, \mathbf{C}_{k,1}^L \leftarrow \text{NanoGICP1.Build}(\mathbf{P}_k)$ 
9:      $\mathbf{K}_k \leftarrow \text{updateKeyframeDatabase}(\mathbf{X}_k^{\mathbf{W}}, \mathbf{P}_k)$ 
10:    continue
11:  end if ▷ 先验估计
12:  if IMU then
13:     $\mathbf{X}_k^{\mathbf{C}} \leftarrow \mathbf{X}_k^{\mathbf{B}}$ 
14:  else
15:     $\mathbf{X}_k^{\mathbf{C}} \leftarrow \mathbf{I}$ 
16:  end if ▷ 帧-帧
17:   $\mathbf{T}_{k,1}^L, \mathbf{C}_{k,1}^L \leftarrow \text{NanoGICP1.Build}(\mathbf{P}_k)$ 
18:   $\mathbf{X}_k^{\mathbf{C}} \leftarrow \text{NanoGICP1.align}(\mathbf{T}_{k,1}^L, \mathbf{C}_{k,1}^L, \mathbf{X}_{k-1}^{\mathbf{W}}, \mathbf{P}_k)$ 
19:   $\mathbf{X}_k^{\mathbf{W}} \leftarrow \mathbf{X}_{k-1}^{\mathbf{W}} \cdot \mathbf{X}_k^{\mathbf{C}}$  ▷ 帧-图
20:   $\mathbf{Q}_k \leftarrow \text{getKeyframeNeighbors}(\mathbf{X}_{k-1}^{\mathbf{W}}, \mathbf{K}_k)$ 
21:   $\mathbf{H}_k \leftarrow \text{getKeyframeHulls}(\mathbf{X}_{k-1}^{\mathbf{W}}, \mathbf{K}_k)$ 
22:  if  $S_k - 1$  then
23:     $\mathbf{T}_{k,2}^L \leftarrow \mathbf{T}_{k-1}^L$ 
24:     $\mathbf{C}_{k,2}^L \leftarrow \mathbf{C}_{k-1}^L$ 
25:  else
26:     $\mathbf{T}_{k,2}^L, \mathbf{C}_{k,2}^L \leftarrow \text{NanoGICP2.Build}(S_k)$ 
    
```

```

27:   end if
28:    $\mathbf{X}_k^w \leftarrow \text{NanoGICP2.align}(\mathbf{T}_{k,2}^L, \mathbf{C}_{k,2}^L, \mathbf{X}_{k-1}^w, \mathbf{P}_k)$ 
29:    $\mathbf{K}_k \leftarrow \text{updateKeyframeDatabase}(\mathbf{X}_k^w, \mathbf{P}_k)$ 
30:    $\mathbf{M}_k \leftarrow \mathbf{M}_{k-1} \oplus \{ \mathbf{K}_k \}$ 
                                     ▷ HDL Graph SLAM 后端
31:    $\text{updatePoseGraph}(\mathbf{X}_k^w, \mathbf{P}_k)$ 
32:   if detectLoopClosure() then
33:       correctLoopClosure()
34:        $\mathbf{X}_k^w \leftarrow \text{optimizeGraph}()$ 
35:        $\mathbf{M}_k \leftarrow \text{reconstructMap}()$ 
36:   end if
37: end while
    
```

5.4 后端位姿图优化及闭合回环检测

5.4.1 IMU 预积分

因为 IMU 数据测量具有一定的误差，因此对 3.3.2 节中 IMU 测量模型需进行修正再对修正后的模型进行预积分过程。修正后的状态变量测量模型见式 5-5，5-6，5-7 所示。

$$\mathbf{p}(t+\Delta t) = \mathbf{p}(t) + \mathbf{v}(t) \cdot \Delta t + \frac{1}{2} (\mathbf{R}_1(t)(\mathbf{a} - \mathbf{b}_a) + \mathbf{g}) \Delta t^2 \quad (5-5)$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + (\mathbf{R}_1(t)(\mathbf{a} - \mathbf{b}_a) + \mathbf{g}) \Delta t \quad (5-6)$$

$$\mathbf{R}_1(t + \Delta t) = \mathbf{R}_1(t) \text{Exp}((\boldsymbol{\omega} - \mathbf{b}_\omega) \Delta t) \quad (5-7)$$

其中， $\mathbf{b}_a$ 被定义为加速度计的零偏误差； $\mathbf{b}_\omega$ 被定义为陀螺仪的零偏误差。

因状态变量中含有误差项，因此误差状态变量也需要更新。

$$\xi \mathbf{p}(t+\Delta t) = \xi \mathbf{p}(t) + \xi \mathbf{v}(t) \Delta t \quad (5-8)$$

$$\xi \mathbf{v}(t + \Delta t) = \xi \mathbf{v}(t) + \xi (\mathbf{R}_1(t)(\mathbf{a} - \mathbf{b}_a) + \mathbf{g}) \Delta t - \boldsymbol{\eta}_v \quad (5-9)$$

$$\xi \boldsymbol{\theta}(t + \Delta t) = \text{Exp}(-(\boldsymbol{\omega} - \mathbf{b}_\omega) \Delta t) \xi \boldsymbol{\theta} - \xi \mathbf{b}_\omega \Delta t - \boldsymbol{\eta}_\theta \quad (5-10)$$

对状态变量测量模型进行预积分得：

$$\begin{aligned}\Delta \mathbf{p} &= \mathbf{R}_I^T \left(\mathbf{p}(t+\Delta t) - \mathbf{p}(t) - \mathbf{v}(t)\Delta t - \frac{1}{2} \mathbf{g}\Delta t^2 \right) \\ &= \sum_{k=t}^{t+\Delta t-1} \left(\Delta \mathbf{v}_{tk} \Delta t + \frac{1}{2} \Delta \mathbf{R}_{tk} (\mathbf{a}_k - \mathbf{b}_a(k) - \boldsymbol{\eta}_a(k)) \Delta t^2 \right)\end{aligned}\quad (5-11)$$

$$\begin{aligned}\Delta \mathbf{v} &= \mathbf{R}_I^T (\mathbf{v}(t+\Delta t) - \mathbf{v}(t) - \mathbf{g}\Delta t) \\ &= \sum_{k=t}^{t+\Delta t-1} \Delta \mathbf{R}_{tk} (\mathbf{a}_k - \mathbf{b}_a(k) - \boldsymbol{\eta}_a(k)) \Delta t\end{aligned}\quad (5-12)$$

$$\begin{aligned}\Delta \mathbf{R}_I &= \mathbf{R}_I^T(t) \mathbf{R}_I(t+\Delta t) \\ &= \prod_{k=t}^{t+\Delta t-1} \text{Exp}((\boldsymbol{\omega}_k - \mathbf{b}_\omega(k) - \boldsymbol{\eta}_\omega(k)) \Delta t)\end{aligned}\quad (5-13)$$

其中， $\boldsymbol{\eta}$ 表示噪声值。

在 IMU 的实际应用过程中，预积分模型内部通常存在零偏误差，并且该误差会随着叉车运行状态和外部环境的变化而产生随机漂移。为了解决零偏误差与 IMU 预积分之间的相互影响，本文假设在当前帧的预积分过程中，零偏误差保持不变^[80]。因此，针对上述预积分方程，可以将测量的状态变量和噪声分离，得到以下式子：

$$\begin{aligned}\Delta \mathbf{p} &= \Delta \mathbf{p} + \sum_{k=t}^{t+\Delta t-1} \left(-\xi \mathbf{v}_{tk} + \frac{1}{2} \Delta \mathbf{R}_{tk} (\mathbf{a}_k - \mathbf{b}_a(t))^T \xi \phi_{tk} \Delta t^2 - \frac{1}{2} \Delta \mathbf{R}_{tk} \boldsymbol{\eta}_a(k) \Delta t^2 \right) \\ &= \Delta \mathbf{p} - \xi \Delta \mathbf{p}\end{aligned}\quad (5-14)$$

$$\begin{aligned}\Delta \mathbf{v} &= \sum_{k=t}^{t+\Delta t-1} \Delta \mathbf{R}_{tk} (\mathbf{I} - \xi \phi_{tk}) (\mathbf{a}_k - \mathbf{b}_a(k) - \boldsymbol{\eta}_a(k)) \Delta t \\ &= \Delta \hat{\mathbf{v}} - \xi \Delta \mathbf{v}\end{aligned}\quad (5-15)$$

$$\begin{aligned}\Delta \mathbf{R}_I &= \Delta \mathbf{R}_I \prod_{k=t}^{t+\Delta t-1} \text{Exp}(-\Delta \mathbf{R}_I^T(k+1) \mathbf{J}_r(k) \boldsymbol{\eta}_\omega(k) \Delta t) \\ &= \Delta \mathbf{R}_I \text{Exp}(-\xi \phi)\end{aligned}\quad (5-16)$$

其中， $\Delta \mathbf{p}, \Delta \hat{\mathbf{v}}, \Delta \mathbf{R}_I$ 均表示 IMU 的观测值； $\xi \Delta \mathbf{p}, \xi \Delta \mathbf{v}, \xi \Delta \phi$ 均表示误差项； $\mathbf{J}$ 表示雅可比矩阵。

由于误差状态变量的预积分涉及较为复杂的噪声模型，并包含高阶项，本文以激光雷达的关键帧作为基本单位进行预积分^[80]。由于每次预积分的累积时长较短，因此本文在噪声模型中仅保留了一阶系数：

$$\begin{aligned} \xi \Delta \mathbf{p} = & \xi \mathbf{p}_{t,t+\Delta t-1} + \xi \mathbf{v}_{t,t+\Delta t-1} \Delta t - \frac{1}{2} \Delta \mathbf{R}_{t,t+\Delta t-1} \left(\mathbf{a}_{t+\Delta t-1} - \mathbf{b}_a(t) \right)^T \xi \phi_{t,t+\Delta t-1} \Delta t^2 \\ & + \frac{1}{2} \Delta \mathbf{R}_{t,t+\Delta t-1} \boldsymbol{\eta}_a(t) \Delta t^2 \end{aligned} \quad (5-17)$$

$$\begin{aligned} \xi \Delta \mathbf{v} = & \xi \mathbf{v}_{t,t+\Delta t-1} - \Delta \mathbf{R}_{t,t+\Delta t-1} \left(\mathbf{a}_{t+\Delta t-1} - \mathbf{b}_a(t) \right)^T \xi \phi_{t,t+\Delta t-1} \Delta t \\ & + \Delta \mathbf{R}_{t,t+\Delta t-1} \boldsymbol{\eta}_a(t) \Delta t \end{aligned} \quad (5-18)$$

$$\xi \Delta \phi = \Delta \mathbf{R}_{t+\Delta t-1,t+\Delta t}^T \xi \phi_{t,t+\Delta t-1} + \mathbf{J}_{r,t+\Delta t-1} \sum \boldsymbol{\eta}_\omega \mathbf{J}_{r,t+\Delta t-1}^T \Delta t^2 \quad (5-19)$$

所以噪声协方差矩阵：

$$\boldsymbol{\eta}_{t,k} = \begin{bmatrix} \xi \phi_{t,k} & \xi \mathbf{v}_{t,k} & \xi \mathbf{p}_{t,k} \end{bmatrix}^T, \boldsymbol{\eta}_{d,k} = \begin{bmatrix} \boldsymbol{\eta}_\omega(t+\Delta t) & \boldsymbol{\eta}_a(t+\Delta t) \end{bmatrix}^T \quad (5-20)$$

最后预积分状态变量的残差模型，在不同文献中对于预积分的残差定义并不完全一致，相应的雅可比矩阵也会发生变化，比较常见的做法是直接使用定义求导^[80]：

$$r(\Delta \mathbf{p}) = \mathbf{R}_t^T \left(\mathbf{p}(t+\Delta t) - \mathbf{p}(t) - \mathbf{v}(t) \Delta t - \frac{1}{2} \mathbf{g} \Delta t^2 \right) - \Delta \mathbf{p} \quad (5-21)$$

$$r(\Delta \mathbf{v}) = \mathbf{R}_t^T (\mathbf{v}(t+\Delta t) - \mathbf{v}(t) - \mathbf{g} \Delta t) - \Delta \hat{\mathbf{v}} \quad (5-22)$$

$$r(\Delta \mathbf{R}_t) = \log \left(\Delta \mathbf{R}_t^T \left(\mathbf{R}_t^T(t) \mathbf{R}_t(t+\Delta t) \right) \right) \quad (5-23)$$

5.4.2 闭合回环检测

激光 SLAM 前端里程计的误差积累是常见现象。传感器误差（如陀螺仪、加速度计及激光雷达的测量偏差）会随时间逐步累积，导致定位状态估计产生漂移^[81]。这种漂移可能影响系统的长期精度，从而降低其在实际环境中的应用效果。

为减小传感器误差累积的影响，引入了回环检测进行误差校正^[82]。回环检测指机器人在环境中移动时，重新到达先前访问的位置。通过识别这些回环，系统能够建立约束关系，从而调整位姿估计，减少误差的持续积累。

本文采用多分辨率回环检测策略对子地图进行修正，其核心思想是将地图划分为多个分辨率层级，每个子地图包含不同层次的信息。引入多分辨率子地图旨在降低传感器误差累积的影响。不同分辨率层级可提供多层次环境信息，使系统能够在不同精度要求下进行匹配，不仅增强了对传感器误差的鲁棒性，也提升了系统在多种环境条件下的适应性。

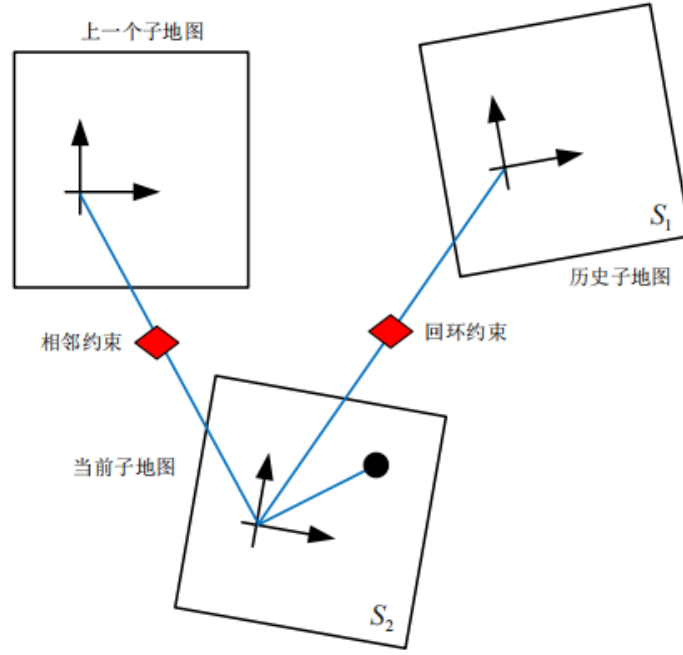


图 5-2 回环约束示意图

回环约束的示意图如图 5-2 所示。在回环优化过程中，每次触发回环约束时，都会增加当前子地图与历史子地图之间的约束条件（见式 5-24 所示）。回环约束来源于相邻位姿之间的相对位姿观测，并与回环检测计算出的两个子地图之间的相对位姿关系相关联。

$$\mathbf{X}_{\text{loop}}^w = \left(\mathbf{X}_{\text{current}}^w \right)^{-1} \mathbf{X}_{\text{previous}}^w \quad (5-24)$$

其中， $\mathbf{X}_{\text{current}}^w$ 为当前帧的位姿； $\mathbf{X}_{\text{previous}}^w$ 为与闭合回环相关的位姿。

5.4.3 因子图优化

传统数据融合优化通常采用滤波方法进行多源传感器融合，但滤波方法仅进行单次迭代，且优化过程中受限于误差的线性化。为进一步增强传感器间的关联性并提升系统整体性能，本文在 SLAM 后端优化中引入紧耦合的图优化方法，该方法在统一的优化框架下协同处理多源传感器数据，同时兼顾地图构建与定位过程，从而提升系统的优化效果^[83]。

因子图优化的目标是最小化所有因子的误差函数 $e_{\min}^w$ ：

$$e_{\min}^w = \min \sum_{ij} \varpi_{ij} \|e_{ij}^w\|^2 \quad (5-25)$$

定义误差函数 e_{ij}^w

$$e_{ij}^w = \mathbf{X}_{ij}^w - \left((\mathbf{X}_i^w)^{-1} \mathbf{X}_j^w \right) \quad (5-26)$$

这一阶段最终目标函数为：

$$e_{\min}^w = \min \sum_{ij} \varpi_{ij} \left\| \mathbf{X}_{ij}^w - \left((\mathbf{X}_i^w)^{-1} \mathbf{X}_j^w \right) \right\|^2 \quad (5-27)$$

其中，其中， ϖ_{ij} 为节点 (i, j) 的权重； $\mathbf{X}_{ij}^w$ 为从帧 i 到帧 j 的真实相对位姿； $\mathbf{X}_i^w$ 和 $\mathbf{X}_j^w$ 由式 4-17 计算得到。

5.4.4 地图重建

地图重建的目标是基于优化后的位姿来生成地图，获取每一帧 k 的点云数据 $\mathbf{P}_k^L$ ，并转换到全局坐标系下：

$$\mathbf{P}_{\text{map},k}^w = \mathbf{X}_k^w \mathbf{P}_k^L \quad (5-28)$$

其中对于点云中具体到每个点 $\mathbf{P}_k^w$ 在全局坐标系下的坐标 $\mathbf{P}_{\text{map},k}^w$ 可由公式 5-29 计算得到。

$$\mathbf{P}_{\text{map},k}^w = \mathbf{R}_k^B \mathbf{X}_k^w \mathbf{P}_k^w + \mathbf{s}_k^B \quad (5-29)$$

这一阶段的最终目标是将所有关键帧的点云数据转换到全局坐标系后，将它们融合在一起，以形成一个新的整体地图达到地图重建的目的：

$$\mathbf{M}_K = \bigcup_{k=1}^N \mathbf{P}_{\text{map},k}^w = \bigcup_{k=1}^N \left(\mathbf{R}_k^B \mathbf{X}_k^w \mathbf{P}_k^w + \mathbf{s}_k^B \right) \quad (5-30)$$

最后，通过第 4.3.1 节中的体素滤波器对新建立地图中的点云进行平均化处理，得到最终的新地图。

5.5 实验分析

基于实验室自研的单舵轮驱动的堆垛式无人叉车平台（如图 5-3 所示），在模拟室内环境、室外光照环境、室外稀疏特征环境以及 HILTI 数据集中室内空旷昏暗大房间数据对基于 EKF 的 DLO-HDL 信息融合定位建图算法的单舵轮驱动的堆垛式无人叉车室内同时定位与建图方法进行了验证实验。设备主控系统采用研华的 EI-52 工控机，配备 Intel Core i5-1135G7 和 Celeron 6305E 处理器，具备低功耗高性能的特点，四核八线程，最大睿频可达 4.2 GHz，搭载 ROS 机器人操

作系统。工业场地环境数据以及无人叉车坐标信息通过 16 线激光雷达和九轴 IMU 采集。激光雷达选用了稳定性较高的速腾 RS-LiDAR-16，测量范围为 40 cm 至 150 m，扫描范围为 360°，分辨率为 0.4°，最高采样频率可达 10 Hz。惯性测量单元 IMU 则选用了维特 HWT905 九轴 IMU，其姿态测量精度为 0.05°，通信距离可达 300 m，最高输出频率为 200 Hz。



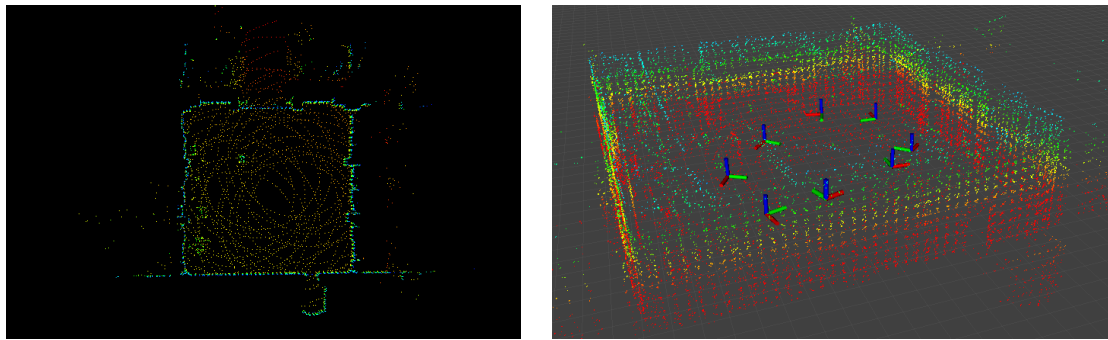
图 5-3 实验场景及设备

5.5.1 地图构建及性能分析

首先,本文采用基于 EKF 的 DLO 信息融合定位建图算法(EKF-DLO SLAM)替代基于图优化的 Cartographer 定位建图算法,用于单舵轮驱动的堆垛式无人叉车的 3D 激光 SLAM。此举旨在解决基于图优化的 Cartographer 算法在计算和存储需求上的高负担,以及其在复杂动态环境中表现不佳、易受动态物体干扰导致定位建图偏差的问题。实验结果表明,基于 EKF 的 DLO 信息融合定位建图算法在计算效率上比 Cartographer 算法有显著提升。然而,在复杂环境中的建图效果仍有待改善,因此急需进一步优化,以提高其在复杂场景下建图的精度和稳定性。针对这一问题,本文进一步提出将 HDL-Graph-SLAM 的后端图优化技术集成到基于 EKF 的 DLO 信息融合定位建图算法中对其进行改进,以有效解决建图过程中出现的问题。

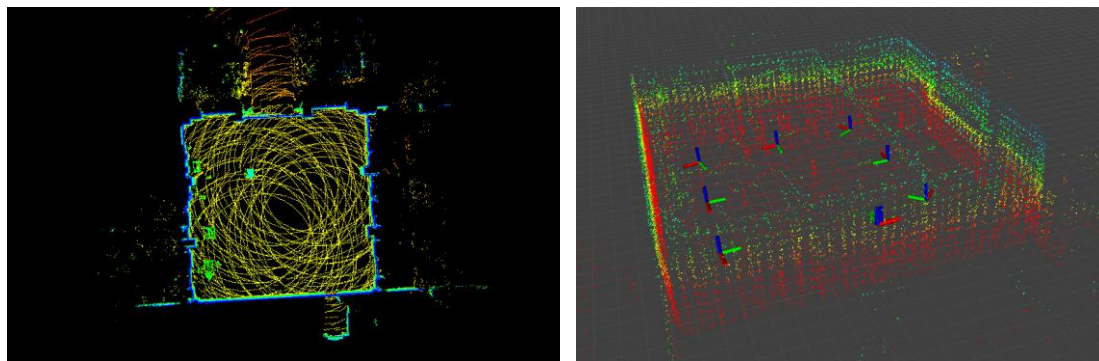
在实验室场景中,分别使用基于 EKF 的 DLO 信息融合定位建图算法(EKF-DLO SLAM)和改进后的基于 DLO-HDL 的信息融合定位建图算法(DLO-HDL SLAM)进行地图构建的实验结果表明,基于 EKF 的 DLO 信息融合定位建图算法构建的地图在清晰度和准确度上存在明显问题,同时坐标系也出现了明显的偏移(如图 5-4 所示)。而采用改进后的基于 DLO-HDL 的信息融合定位建图算法

后，地图的质量得到了显著提升，清晰度和准确度有了明显改善，坐标系偏移问题也得到了有效解决（如图 5-5 所示）。在室外光照影响较大的环境下进行的实验也表明，采用改进后的基于 DLO-HDL 的信息融合定位建图算法所构建的地图质量相较于基于 EKF 的 DLO 信息融合定位建图算法有了显著提升，清晰度和准确度均得到明显改善，坐标系偏移问题同样得到了有效解决（如图 5-6 所示）。



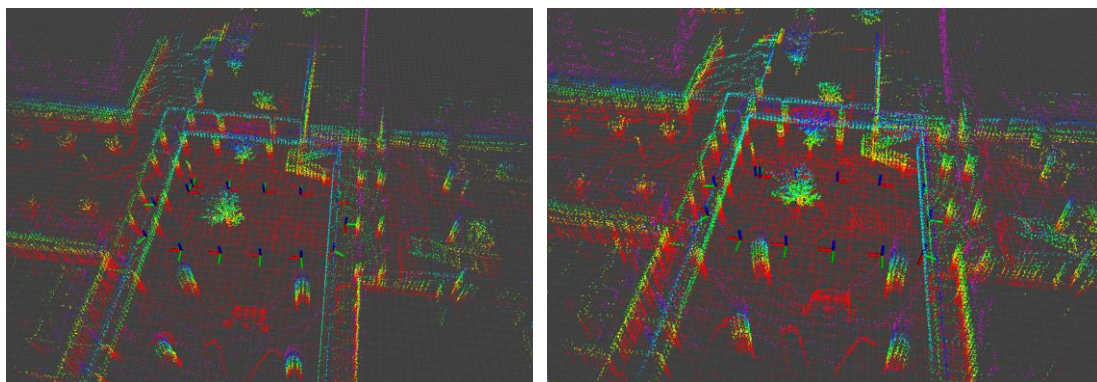
(a) EKF-DLO 建立的室内环境地图 (b) EKF-DLO 建立室内环境地图的坐标偏移

图 5-4 室内环境中基于 EKF 的 DLO 信息融合定位建图算法的建图效果



(a) DLO-HDL 建立的室内地图 (b) DLO-HDL 建立室内地图的坐标偏移

图 5-5 室内环境中基于 DLO-HDL 的信息融合定位建图算法的建图效果



(a) EKF-DLO 室外光照环境坐标偏移 (b) DLO-HDL 室外光照环境坐标偏移

图 5-6 室外光照环境中 EKF-DLO SLAM 与 DLO-HDL SLAM 的建图效果

然而，在室外稀疏特征环境中，采用改进后的基于 DLO-HDL 的信息融合定位建图算法进行地图构建未达到预期效果（如图 5-7 所示）。

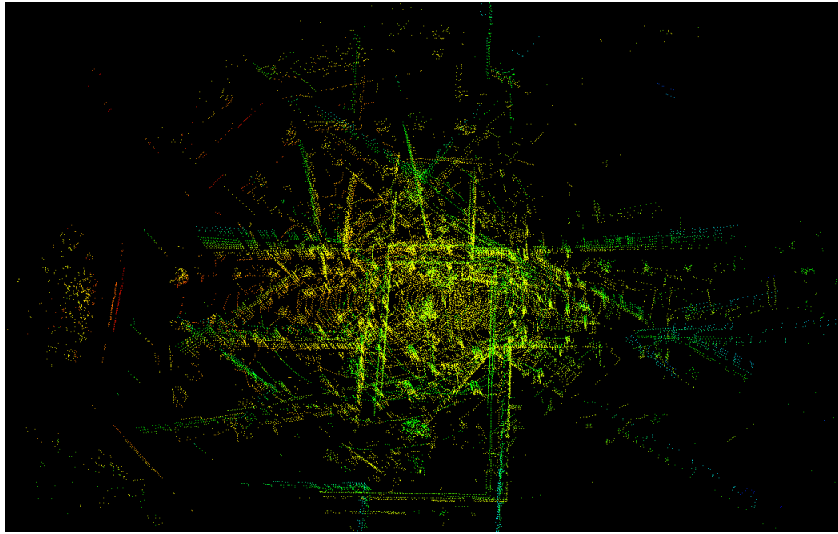
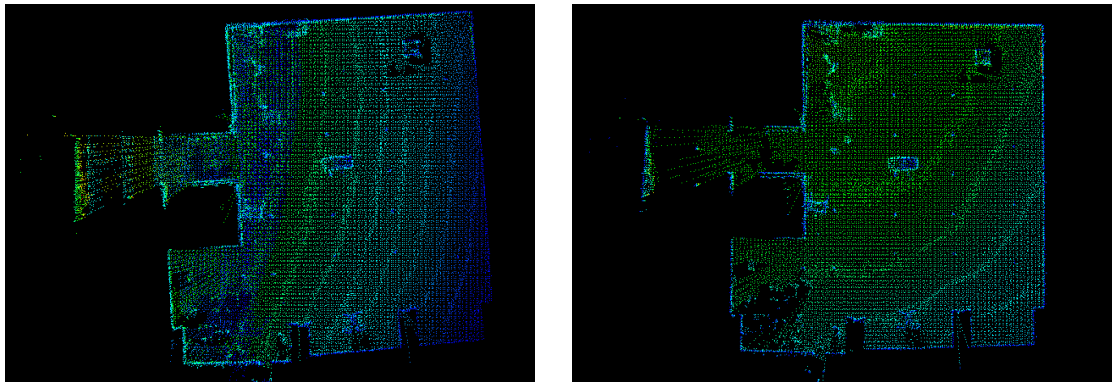


图 5-7 室外具有稀疏特征环境中 DLO-HDL SLAM 的建图效果

在采用 HILTI 数据集的室内空旷昏暗大房间环境中，使用改进后的基于 DLO-HDL 的信息融合定位建图算法则取得了预期效果（如图 5-8 所示）。



(a) HILTI 数据集中 EKF-DLO 的建图效果 (b) HILTI 数据集中 DLO-HDL 的建图效果

图 5-8 HILTI 数据集中 EKF-DLO SLAM 与 DLO-HDL SLAM 的建图效果

实验结果表明，在构建地图时，基于 DLO-HDL 的信息融合定位建图算法（DLO-HDL SLAM）的计算机资源占用率虽然略高于基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO SLAM），但仍显著低于基于图优化的 Cartographer 定位建图算法（如图 5-9 所示）。这一现象表明，本文提出的两种改进算法 EKF-DLO SLAM 和 DLO-HDL SLAM，在计算资源利用效率方面均有明显提升。通过综合分析图 5-4 至图 5-9 的实验数据可以证实，基于 DLO-HDL 的信息融合定位建图算法在保证计算效率的同时，其建图可靠性也得到了有效提升。

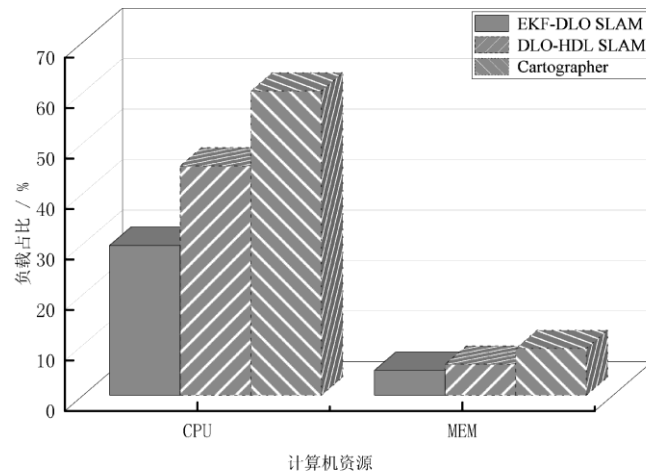
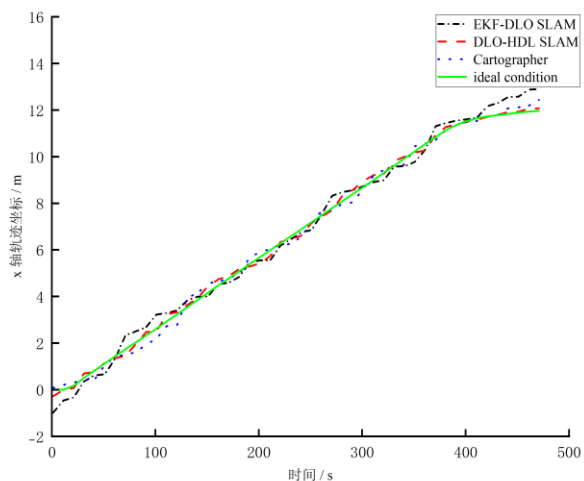


图 5-9 三种算法对计算机资源使用占比

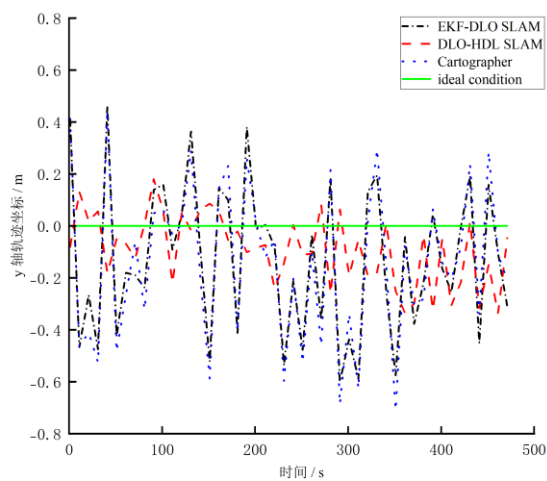
5.5.2 轨迹误差分析

本文将 HDL-Graph-SLAM 的后端集成进基于 EKF 的 DLO 信息融合定位建图算法中对基于 EKF 的 DLO 信息融合定位建图算法进行改进，并猜想单舵轮驱动的堆垛式无人叉车采用改进后的基于 DLO-HDL 的信息融合定位建图算法相对于原算法的轨迹定位精度会有所提高，为了验证猜想，本文设计在室内环境、室外光照环境以及 HILTI 数据集数据下分别利用单舵轮驱动的堆垛式无人叉车采用基于 EKF 的 DLO 信息融合定位建图算法和改进后的基于 DLO-HDL 的信息融合定位建图算法进行建图并完成实际轨迹运动规划对比两种算法在实际运动过程中的轨迹误差以验证猜想。

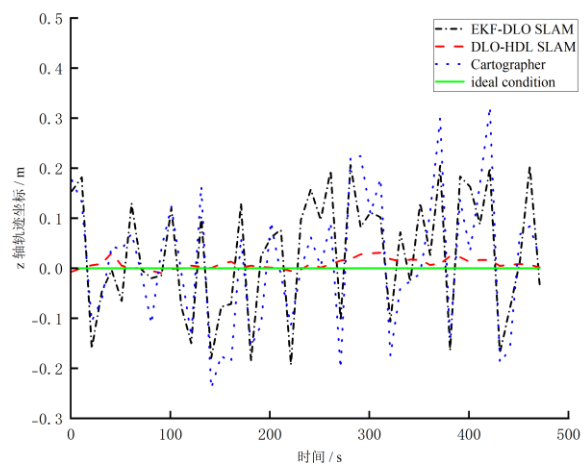
图 5-10 的结果表明，单舵轮驱动的堆垛式无人叉车采用改进后的基于 DLO-HDL 的信息融合定位建图算法（DLO-HDL SLAM）在室内工业环境下进行简单直线轨迹规划时，无论是 x 轴、 y 轴还是 z 轴的运动误差，相较于基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO SLAM）都有显著改善，验证了我们猜想的准确性。



(a) x 轴轨迹误差



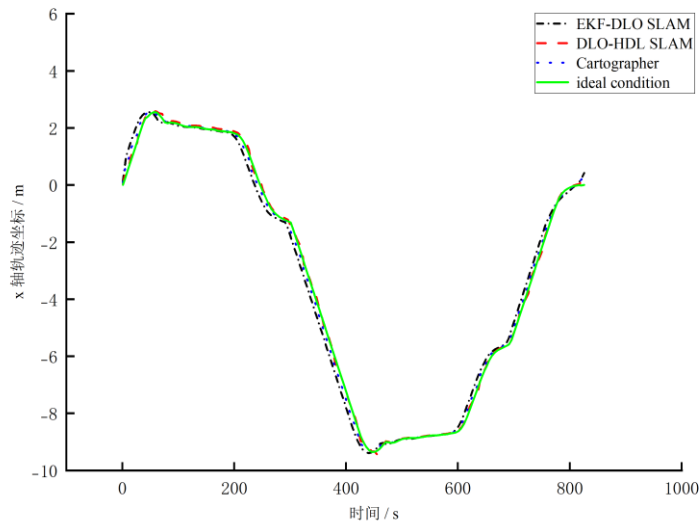
(b) y 轴轨迹误差



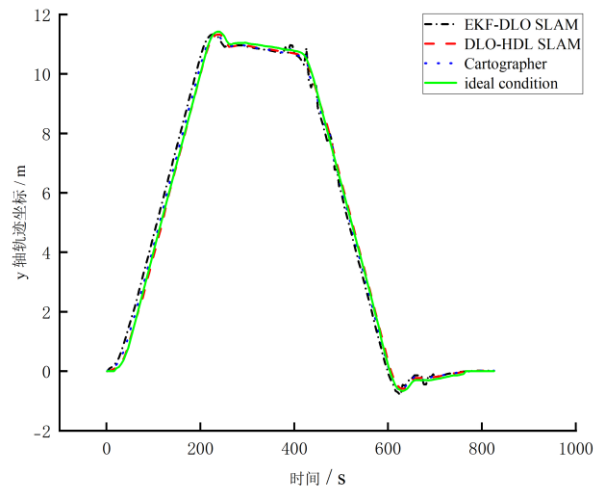
(c) z 轴轨迹误差

图 5-10 室内直线行驶下三种算法的轨迹误差对比图

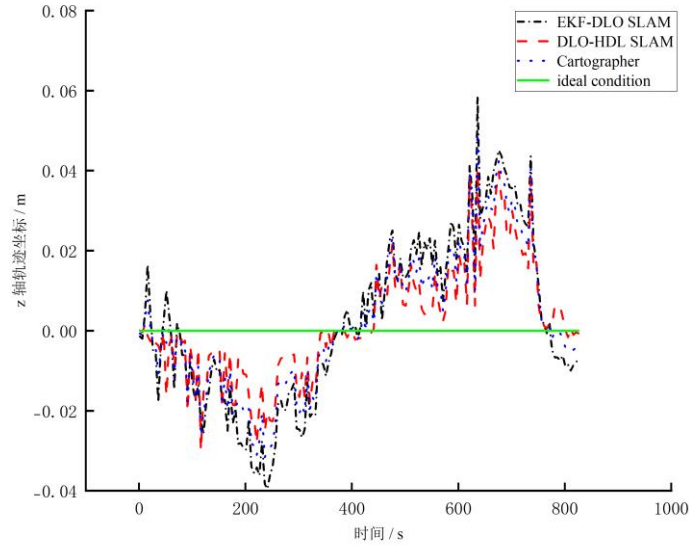
图 5-11 的结果表明,单舵轮驱动的堆垛式无人叉车采用改进后的基于 DLO-HDL 的信息融合定位建图算法 (DLO-HDL SLAM) 在室内工业环境下进行复杂环形轨迹规划时,无论是 x 轴、 y 轴还是 z 轴的运动误差,相较于基于 EKF 的 DLO 信息融合定位建图算法 (EKF-DLO SLAM) 都有显著改善,验证了我们猜想的准确性。



(a) x 轴轨迹误差



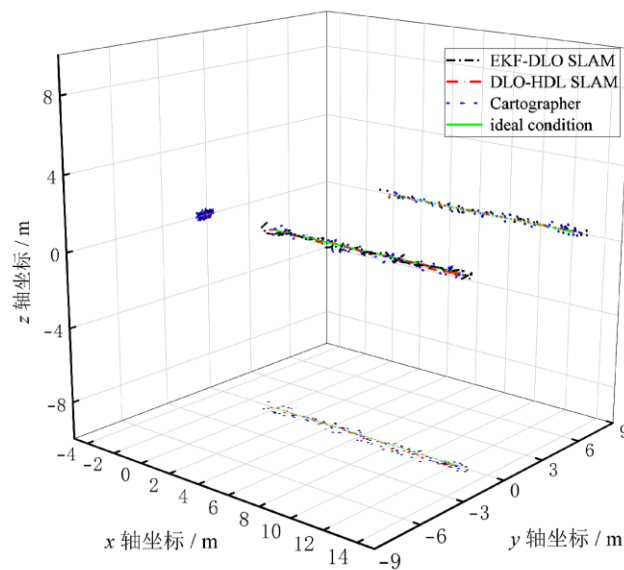
(b) y 轴轨迹误差



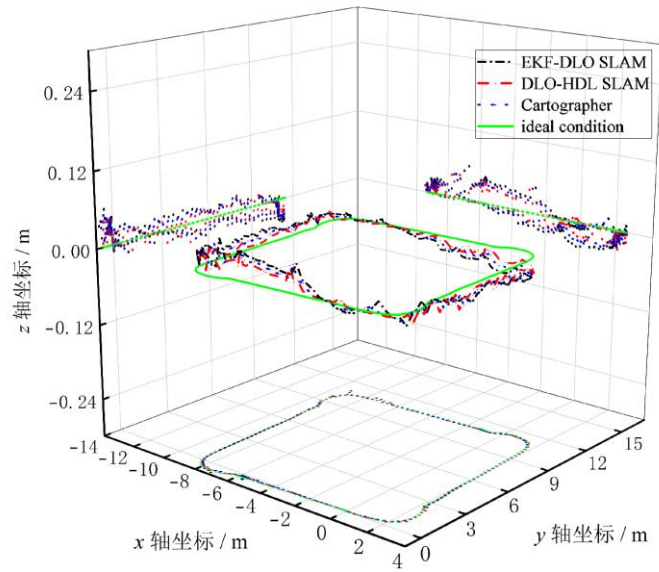
(c) z 轴轨迹误差

图 5-11 室内环形运动下三种算法的轨迹误差对比图

图 5-12 为单舵轮驱动的堆垛式无人叉车在室内工业环境下的运动轨迹图，结果表明，单舵轮驱动的堆垛式无人叉车采用改进后的基于 DLO-HDL 的信息融合定位建图算法（DLO-HDL SLAM）在室内工业环境下进行轨迹规划时，无论是 x 轴、 y 轴还是 z 轴的运动误差，相较于基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO SLAM）都有显著改善，验证了我们猜想的准确性。



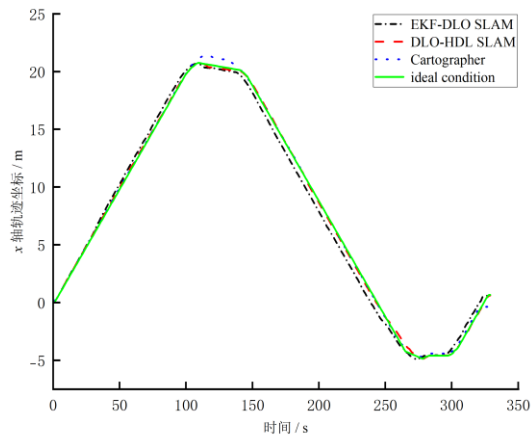
(a) 室内直线运动轨迹图



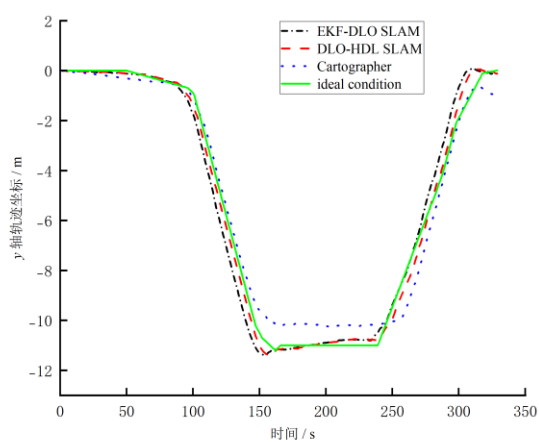
(b) 环形运动轨迹图

图 5-12 室内工业环境下三种算法运动轨迹图

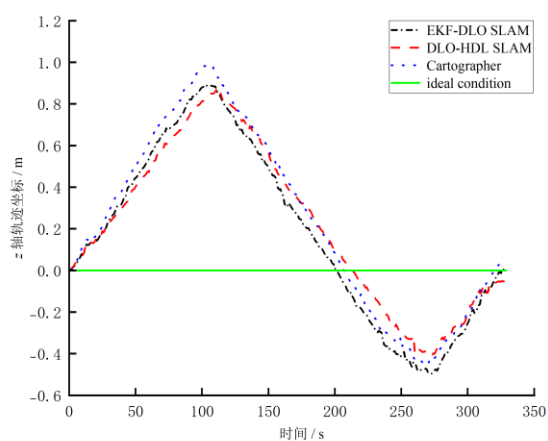
图 5-13、5-14 的结果表明，单舵轮驱动的堆垛式无人叉车采用改进后的基于 DLO-HDL 的信息融合定位建图算法（DLO-HDL SLAM）在室外光照环境下进行轨迹规划时，无论是 x 轴、 y 轴还是 z 轴的运动误差，相较于基于 EKF 的 DLO 信息融合定位建图算法（EKF-DLO SLAM）都有显著改善，验证了我们猜想的准确性。



(a) x 轴轨迹误差



(b) y 轴轨迹误差



(c) z 轴轨迹误差

图 5-13 室外光照环境下三种算法的轨迹误差对比图

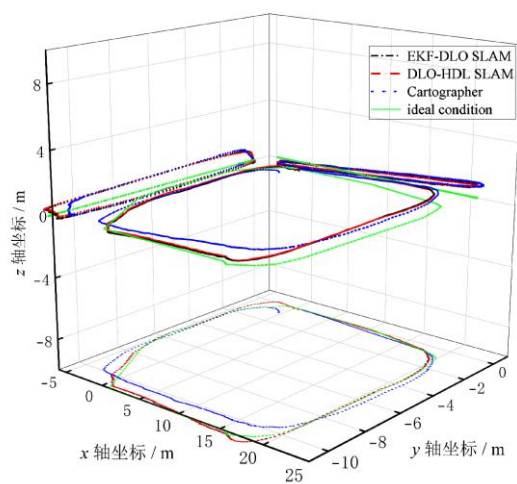
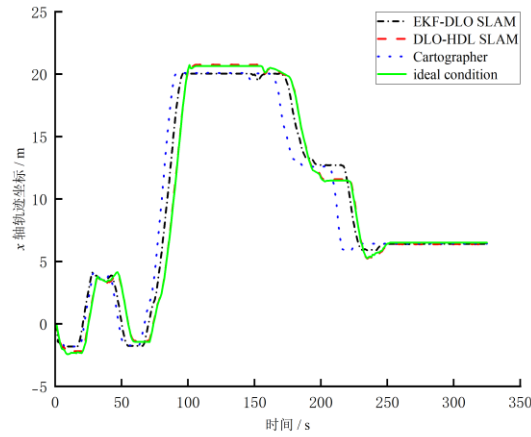
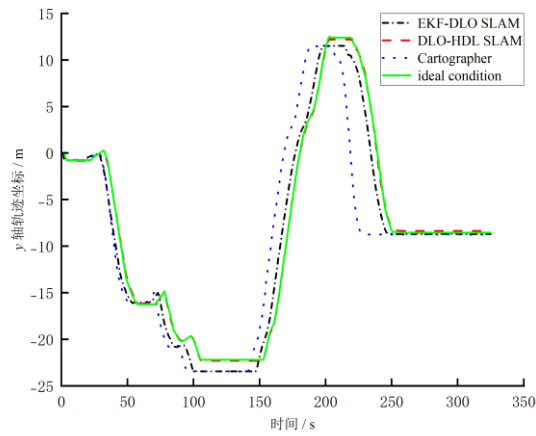


图 5-14 室外光照环境下三种算法运动轨迹图

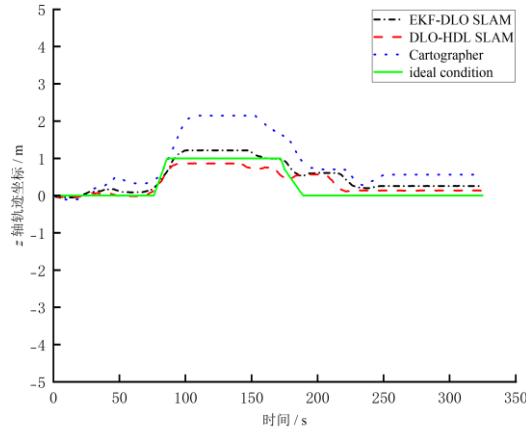
图 5-15、5-16 的结果表明,单舵轮驱动的堆垛式无人叉车采用改进后的基于 DLO-HDL 的信息融合定位建图算法 (DLO-HDL SLAM) 在 HILTI 数据集中室内空旷昏暗大房间下进行轨迹规划时,无论是 x 轴、 y 轴还是 z 轴的运动误差,相较于基于 EKF 的 DLO 信息融合定位建图算法 (EKF-DLO SLAM) 都有显著改善,验证了我们猜想的准确性。



(a) x 轴轨迹误差



(b) y 轴轨迹误差



(c) z 轴轨迹误差

图 5-15 HILTI 数据集下三种算法的轨迹误差对比图

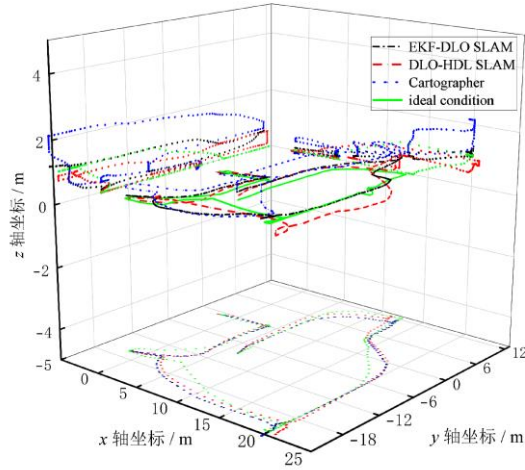


图 5-16 HILTI 数据集下三种算法运动轨迹图

5.5.3 稳定性分析

为了验证本文提出的 DLO-HDL SLAM 算法在精度和稳定性方面的优越性，我们将其与第四章提出的 EKF-DLO SLAM 算法以及基于图优化的 Cartographer 定位建图算法进行了对比实验。实验场景包括：单舵轮驱动的堆垛式无人叉车在实际室内工业环境下的直线轨迹和环形轨迹测试、室外光照环境测试，以及 HILTI 数据集中室内空旷昏暗大房间的场景测试。实验通过比较各算法在叉车行驶轨迹上的定位点与理想轨迹点之间的方差和标准差来评估性能。如表 5-1 和 5-2 所示，DLO-HDL SLAM 算法在所有测试场景下（包括室内工业环境、室外光照环境和 HILTI 数据集场景）的轨迹点方差和标准差均显著小于 EKF-DLO 算法

和 Cartographer 算法。这一结果充分证明了 DLO-HDL SLAM 算法在定位精度和稳定性上的显著优势。具体而言，实验数据显示：DLO-HDL SLAM 算法相较于 EKF-DLO 算法定位精度提升了 60.9%，相较于 Cartographer 算法提升了 56.9%。同时，在系统稳定性方面也取得了显著提升。这些定量结果有力地验证了 DLO-HDL SLAM 算法的优越性能。

表 5-1 室内环境轨迹点误差的方差和标准差

性能 算法	室内直线轨迹		室内环形轨迹	
	方差/m ²	标准差/m	方差/m ²	标准差/m
Cartographer	0.02805	0.16748	0.07297	0.25157
EKF-DLO SLAM	0.02724	0.16505	0.06042	0.24580
DLO-HDL SLAM	0.00205	0.04522	0.00862	0.09285

表 5-2 室外环境与数据集环境轨迹点误差的方差和标准差

性能 算法	室外光照环境		HILTI 数据集	
	方差/m ²	标准差/m	方差/m ²	标准差/m
Cartographer	0.11405	0.33772	19.65863	4.43381
EKF-DLO SLAM	0.13351	0.36539	2.16202	1.47038
DLO-HDL SLAM	0.07107	0.26659	0.07064	0.26577

5.6 本章小结

本章针对第四章提出的基于 EKF 的 DLO 信息融合定位建图算法的误差机理展开深入研究，在第四章的基础上引入后端优化机制，提出了一种 DLO-HDL 信息融合定位建图算法。研究首先系统分析了激光雷达与 IMU 传感器误差对定位建图精度的影响机制，在此基础上构建了 DLO-HDL SLAM 的系统框架，重点优化了后端位姿图与闭环检测等关键技术环节。实验验证表明，该算法在复杂环境下具有显著的性能提升，定位精度和建图质量均优于传统方法。研究成果为无人叉车的高精度定位与建图提供了创新的技术方案，具有重要的工程应用价值。

第 6 章 总结与展望

6.1 论文总结

与传统叉车相比，无人叉车在工业制造、智能仓储和工业运输等众多领域具有显著优势，已广泛应用于各个行业。因此，确保无人叉车具备高精度和高稳定性的定位能力至关重要。3D 激光 SLAM 是一种常见的定位导航方法，通过激光雷达传感器实现移动机器人的定位与建图工作。本文提出了一种基于 EKF 的 DLO 信息融合定位建图算法，研究了单舵轮驱动的堆垛式无人叉车的定位建图问题，并进一步提出了基于 DLO-HDL 的信息融合定位建图算法。该算法实现了单舵轮驱动的堆垛式无人叉车在不同工作环境下的精度和高稳定性的定位建图。本文的主要研究成果如下：

(1) 为实现单舵轮驱动的堆垛式无人叉车基于 3D 激光 SLAM 的同时定位与建图，本文建立了单舵轮驱动的堆垛式无人叉车的运动模型和导航系统。通过推导激光雷达和 IMU 的测量模型，以及单舵轮驱动的堆垛式无人叉车的运动学模型，为所提出算法的误差机理分析提供了理论支持；此外，本文还推导了数据融合模型，为后续算法中的数据融合提供了具体方案，并通过 MATLAB 进行运动学仿真，验证了推导模型的准确性。最后，设计了单舵轮驱动的堆垛式无人叉车的导航系统，为实验验证搭建了实验平台。

(2) 针对单一激光雷达传感器在单舵轮驱动的堆垛式无人叉车定位建图中存在的误差较大、复杂动态环境下建图效果差以及传统图优化算法在处理 3D SLAM 时面临的高计算量和高存储需求等问题，本文提出了一种基于 EKF 的 DLO 信息融合定位建图算法。该方法通过扩展卡尔曼滤波（EKF）将 IMU（惯性测量单元）和 LiDAR（激光雷达）传感器数据融合并利用融合后的数据作为 DLO SLAM 的输入，实现了单舵轮驱动的堆垛式无人叉车的高效定位与建图。实验结果验证了该方法可以基本实现单舵轮驱动的堆垛式无人叉车的精准定位与建图工作，并有效解决了传统图优化算法在处理 3D SLAM 时所面临的计算量过大和存储需求过高的问题。

(3) 针对基于 EKF 的 DLO 信息融合定位建图算法因缺乏后端优化而导致的误差累计问题, 本文提出了基于 DLO-HDL 的信息融合定位建图算法。该方法将 HDL-Graph-SLAM 后端的回环检测与位姿图优化融入到基于 EKF 的 DLO 信息融合定位建图算法中, 从而实现了单舵轮驱动的堆垛式无人叉车的高精度、高稳定性定位与建图。通过对比实验结果表明, 该方法有效解决了基于 EKF 的 DLO 信息融合定位建图算法中的误差累计问题, 并且在精度和鲁棒性方面表现优异。

6.2 研究展望

本文对单舵轮驱动的堆垛式无人叉车基于 3D 激光 SLAM 的同时定位与建图方法进行了深入研究, 但在实际应用中仍需进一步优化, 以提升该方法的实用性和可靠性。以下几个方面仍需要进一步完善:

(1) 极端环境下的高精度与高稳定性同时定位与建图研究。本文提出的算法在动态遮挡、纹理缺乏区域以及特征稀疏等极端环境中的定位建图效果未能达到预期。因此, 如何在这些复杂环境下实现单舵轮驱动的堆垛式无人叉车的高精度、高稳定性同时定位与建图, 仍是未来研究的重要方向。

(2) 多模态感知定位导航系统的研究。本文仅针对基于激光雷达和 IMU 的数据融合进行单舵轮驱动的堆垛式无人叉车的定位导航工作进行研究。在未来的研究中, 可以结合视觉相机、激光雷达、惯性导航等多种传感器的信息, 构建多模态感知系统, 以提升对环境的全面理解和认知能力, 从而使单舵轮驱动的堆垛式无人叉车能够更加可靠地执行各种任务。此外, 借助深度学习技术对传感器数据进行特征提取和融合, 可以更有效地识别和理解环境中的目标物体, 如障碍物和货架, 从而增强单舵轮驱动的堆垛式无人叉车的自主导航和避障能力。

(3) 大负载条件下堆垛式无人叉车同时定位与建图研究。本文未涉及大负载条件下堆单舵轮驱动的堆垛式无人叉车的同时定位与建图分析。然而, 在大负载条件下, 单舵轮驱动的堆垛式无人叉车的车轮变形和滑移现象会更加显著, 从而对定位精度与稳定性产生更大影响。此外, 大负载还会使单舵轮驱动的堆垛式无人叉车的车身转向、加减速操作更加困难。因此, 未来的研究可以聚焦于超大负载条件下单舵轮驱动的堆垛式无人叉车的同时定位与建图, 以应对更为复杂的工作环境和提升系统的性能。

参考文献

- [1] 梁军, 韩冬冬, 盘朝奉, 等. 基于移动机器人的智能车库关键技术综述[J]. 机械工程学报, 2022, 58(03): 1-20.
- [2] 王浩吉, 杨永帅, 赵彦微. 重载AGV的应用现状及发展趋势[J]. 机器人技术与应用, 2019(5): 20-24.
- [3] Jianqi Z, Xu Y, Wei W, et al. Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering[J]. Automation in Construction, 2023, 146.
- [4] 马磊. 基于多传感器融合的AGV定位与自主导航研究[D]. 华南理工大学, 2023.
- [5] 金鹏. 重载AGV结构优化设计与模糊滑模控制[D]. 南京航空航天大学, 2018.
- [6] 王云飞. 电控液驱重载AGV设计研究[D]. 山东大学, 2020.
- [7] 杨建强. 重载AGV运输平台的设计与开发[D]. 山东理工大学, 2021.
- [8] 倪颖臻. 激光导航叉车式自动导引运输车 (AGV) 安全功能设计研究[J]. 价值工程, 2024, 43(30): 83-85.
- [9] 廉胤东, 苟彬, 李超磊, 等. 基于启发式能耗优化的电力仓储分布式多AGV路径规划方法[J]. 自动化与信息工程, 2024, 45(05): 40-46+53.
- [10] 徐建明, 吴小文, 蔡奇正, 等. 基于ROS和全向舵轮驱动的移动机器人系统设计[J]. 高技术通讯, 2021, 31(07): 738-746.
- [11] 张德政. 全向重载AGV多轴运动控制研究[D]. 安徽工程大学, 2022.
- [12] 谭雪, 戴更新. 应用于物流AGV磁导航传感器的研究[J]. 物流科技, 2017, 40(08): 37-39+42.
- [13] Sun Y, Yang J, Zhang Z, et al. An Optimization-Based High-Precision Flexible Online Trajectory Planner for Forklifts[J]. Actuators, 2023, 12(4): 162-183.
- [14] 郑少华. 视觉导航AGV定位与路径规划技术研究[D]. 华南理工大学, 2016.
- [15] 陈伟, 巫帅达, 田子建, 等. 基于ORB-SLAM3视觉与惯导融合的煤矿机器人定位算法研究[J]. 煤炭科学技术, 2024: 1-12.
- [16] S. Su, X. Zeng, S. Song, et al. Positioning Accuracy Improvement of Automated Guided Vehicles Based on a Novel Magnetic Tracking Approach[J]. IEEE Intelligent Transportation

- Systems Magazine, 2020, 12(4): 138-148.
- [17] C. Sun, K. Li. A Positioning Accuracy Improvement Method by Couple RINSs Information Fusion[J]. IEEE Sensors Journal, 2021, 21(17): 19351-19361.
- [18] 朱定赟. 未知环境下基于机器视觉的巡检机器人自主导航控制方法[J]. 自动化与仪器仪表, 2024, (09): 221-224+229.
- [19] Q. Zou, Q. Sun, L. Chen, et al. A Comparative Analysis of LiDAR SLAM-Based Indoor Navigatio for Autonomous Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(7): 6907-6921.
- [20] J.D. Carroll, A.J. Canciani. Terrain-referenced navigation using a steerable-laser measurement sensor[J]. Navigation, 2021, 68(1): 115-134.
- [21] 杨文华. AGV技术发展综述[J]. 物流技术与应用, 2015, 20(11): 93-95.
- [22] 李恒恒. 基于SLAM导航的堆垛式叉车AGV研究与设计[D]. 合肥工业大学, 2022.
- [23] Kuipers B, Feigenbaum E A, Hart P E, et al. Shakey: from conception to history[J]. Ai Magazine, 2017, 38(1): 88-103.
- [24] 贺妍. 基于状态反馈的四轮转向叉车LQR优化控制研究[D]. 合肥工业大学, 2016.
- [25] 张鹏. 托盘式AGV叉车结构设计和稳定性研究[D]. 安徽理工大学, 2019.
- [26] Rodríguez F J, Mazo M, Sotelo M A. Automation of an industrial fork lift truck, guided by artificial vision in open environments[J]. Autonomous Robots, 1998, 5: 215-231.
- [27] Teller S, Walter M R, Antone M, et al. A Voice-Commandable Robotic Forklift Working Alongside Humans in Minimally-Prepared Outdoor Environments[C]. Robotics and Automation (ICRA), 2010 IEEE International Conference on. USA: IEEE, 2010: 526-533.
- [28] 王琛琛. 叉车式AGV路径规划和协同调度算法[D]. 杭州电子科技大学, 2023.
- [29] 刘伟. 动态场景下激光导航叉车式AGV定位与建图应用研究[D]. 西华大学, 2022.
- [30] 褚为杰. AGV叉车结构优化设计[D]. 东南大学, 2020.
- [31] 徐叶帆. 基于标识符视觉定位的AGV导航系统研究[D]. 东南大学, 2016.
- [32] 凡风. 杭叉国产叉车的风向标[J]. 中国储运, 2008, (05): 96-97.
- [33] 胡胜. 杭叉总装一线平衡研究与改进[D]. 浙江工业大学, 2019.
- [34] 王刚, 汪场, 崔丽媛. 合力AGV无人驾驶激光导航[J]. 交通建设与管理, 2017, (09): 28.

- [35] 张明路, 戈新良, 唐智强, 等. 多传感器信息融合技术研究现状和发展趋势[J]. 河北工业大学学报, 2003, (02): 30-35.
- [36] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D LIDAR SLAM[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). Stockholm, Sweden: IEEE, 2016: 1271-1278.
- [37] Tong Qin, Peiliang Li, Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual Inertial State Estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004–1020.
- [38] Shan T, Englot B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 4758-4765.
- [39] Agarwal S, Mierle K. The Ceres Solver Team[J]. Ceres Solver. Available online: <http://ceresolver.org> (accessed on 18 September 2022), 2022.
- [40] Shan T, Englot B, Meyers D, et al. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping[C]//2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2020: 5135-5142.
- [41] 徐杭. 复杂场景下多传感器信息融合导航策略研究[D]. 哈尔滨工业大学, 2021.
- [42] Lv J, Xu J, Hu K, et al. Targetless calibration of lidar-imu system based on continuous-time batch estimation[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020: 9968-9975.
- [43] Dai W, Zhang Y, Li P, et al. Rgb-d slam in dynamic environments using point correlations[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(1): 373-389.
- [44] Shan T, Englot B, Ratti C, et al. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping[C]//2021 IEEE international conference on robotics and automation (ICRA). IEEE, 2021: 5692-5698.
- [45] Zhao S, Zhang H, Wang P, et al. Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 8729-8736.
- [46] 冉旭辉. 基于四线激光雷达的可行驶区域与障碍物检测方法研究[D]. 北京工业大学,

2020.

- [47] Smith R C, Cheeseman P. On the Representation and Estimation of Spatial Uncertainty[J]. The International Journal of Robotics Research, 1986, 5(4): 56-68.
- [48] Choi J. Simultaneous localization and mapping based on the local volumetric hybrid map[C]. Intelligent Vehicles Symposium, 2015: 561-566.
- [49] Grisetti G, Stachniss C, Burgard W. Improving Grid-based SLAM with Rao Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling[C]. IEEE International Conference on Robotics & Automation, 2005: 2432-2437.
- [50] 安赫, 崔敏, 张鹏, et al. 基于 IPSO-Gmapping 算法的 SLAM 系统研究[J]. 国外电子测量技术, 2023, 42(03): 110-115.
- [51] Gutmann J S, Kurt Konolige. Incremental mapping of large cyclic environments[C]//Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No.99EX375). IEEE, 1999: 318-325.
- [52] 郑川川, 柯福阳, 汤琴琴. 基于图优化的激光惯导紧耦合 SLAM 研究[J]. 电子测量技术, 2023, 46(01): 35-42.
- [53] 沈欣, 闵华松. 基于速度积分位姿融合的改进 Cartographer 算法[J]. 应用激光, 2021, 41(05): 1063-1069.
- [54] 丛佩超, 吕昆峰, 周加超. 复杂场景下移动机器人的 SLAM 与路径规划问题[J]. 计算机仿真, 2023, 40(02): 443-448+458.
- [55] Zhang J, Pless R. A tutorial on 2D and 3D SLAM: Applications and future directions[J]. Robotics and Autonomous Systems, 2020, 135: 103645.
- [56] Geng W, Xie L. Multi-sensor fusion for autonomous robots using LiDAR and vision: A review[J]. Sensors, 2022, 22(15): 5410.
- [57] 任泉, 王思宇, 张慧, 等. 基于2D激光SLAM的室内移动机器人定位与建图的设计及实现[J]. 信息技术与信息化, 2024, (11): 182-185.
- [58] Zhou Y, Li J, Zhang W. Research on robot navigation technology based on laser SLAM[J]. International Journal of Robotics and Automation, 2023, 38(4): 567-578.
- [59] 庞正辉, 庞正芳, 张品超, 等. 扩展卡尔曼滤波算法在UWB+IMU定位中的应用[J]. 地

- 理空间信息, 2024, 22(11): 95-99.
- [60] 许永跃, 苏金辉, 陈鹏河, 等. 基于多源信息融合的巡检机器人定位系统[J]. 传感器与微系统, 2024, 43(11): 58-62.
- [61] Zhao X, Li X, Wang Y. Multi-sensor fusion and semantic map-based particle filtering for robust indoor localization[J]. *Sensors*, 2024, 24(4): 304-316.
- [62] 尉雪珂, 陈文, 徐启敏. 基于环境辅助粒子滤波的车辆融合定位算法[J]. 仪表技术与传感器, 2024, (12): 120-126.
- [63] Zhou L, Zhang H, Wang X. FAST-LiDAR-SLAM: A robust and real-time factor graph for urban scenarios with unstable GPS signals[J]. *IEEE Transactions on Robotics*, 2023, 39(1): 112-124.
- [64] Li J, Zhang Y, Wang X. Improved FastSLAM algorithm based on improved particle filter and adaptive genetic algorithm[J]. *Journal of Field Robotics*, 2022, 39(7): 1023-1036.
- [65] 谭东旭, 刘鑫宇, 朱杰辉, 等. 基于因子图优化的水下机器人SLAM算法[J]. 现代制造技术与装备, 2024, 60(12): 205-208.
- [66] 相福磊, 彭富明, 方斌, 等. 基于因子图的激光SLAM模型优化算法[J]. 机械制造与自动化, 2024, 53(05): 167-170+208.
- [67] Kaess M, Johannsson H, Roberts R, et al. Efficient graph-based SLAM using hierarchical optimization and incremental smoothing[J]. *IEEE Transactions on Robotics*, 2022, 38(4): 1234-1245.
- [68] Zhang L, Wang Y, Chen X. Robust graph-based SLAM with deep learning for dynamic environments[J]. *Robotics and Autonomous Systems*, 2023, 159: 104567.
- [69] 赵龙云, 伞红军, 陈久朋, 等. 基于ICP算法的移动机器人激光定位研究[J]. 激光与光电子学进展, 2024, 61(08): 99-108.
- [70] Wang J, Mohammed A F, Patterson G. Performance evaluation of 3D point cloud matching with short-range dense 2D range[C]. 2023 IEEE International Conference on Big Data, 2023: 3874-3879.
- [71] 潘运丹, 林艺鑫. 多传感器信息融合的教学移动机器人设计[J]. 物联网技术, 2024, 14(07): 99-103+106.

- [72] 杨丰澧, 赵龙. 基于局部信息融合的激光惯性里程计算法[J/OL]. 导航定位学报, 2025: 1-13.
- [73] 李志宇. 机器学习辅助改进SRCKF的UWB/IMU组合室内定位方法研究[D]. 内蒙古工业大学, 2021.
- [74] 刘志成, 王华龙, 郭威, 等. 单舵轮AGV叉车的高精度导航算法研究与应用[J]. 计算机测量与控制, 2024, 32(09): 307-314+321.
- [75] 佐富兴. 自动导引四向行走叉车设计及关键技术研究[D]. 上海: 上海交通大学, 2021: 15-35.
- [76] Espineira J P, Robinson J, Groenewald J, et al. Realistic LiDAR with noise model for real-time testing of automated vehicles in a virtual environment[J]. IEEE Sensors Journal, 2021, 21(8): 9919-9926.
- [77] Le Gentil C, Vidal-Calleja T, Huang S. 3d lidar-imu calibration based on upsampled preintegrated measurements for motion distortion correction[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 2149-2155.
- [78] Hu D, Lu Y, Cao Y, et al. Laser-Splashed Three-Dimensional Plasmonic Nanovolcanoes for Steganography in Angular Anisotropy[J]. ACS Nano, 2018, 12(9): 9233-9239.
- [79] Jiaqi X, Zhou C, Jie C, et al. A precise localization algorithm for unmanned aerial vehicles integrating visual-internal odometry and cartographer[J]. Journal of Measurements in Engineering, 2024, 12(2): 284-297.
- [80] Forster C, Carlone L, Dellaert F, et al. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation[C]//Robotics: Science and Systems XI. 2015.
- [81] Li Y, Brasch N, Wang Y, et al. Structure-slam: Low-drift monocular slam in indoor environments [J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6583-6590.
- [82] Cattaneo D, Vaghi M, Valada A. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam[J]. IEEE Transactions on Robotics, 2022, 38(4): 2074-2093.
- [83] Min Z, Dunn E. Voldor + slam: For the times when feature-based or direct methods are not good enough[C]//2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021: 13813-13819.

附录 1 直线运动学仿真 MATLAB 代码

```
dt = 0.1; T = 10; t = 0:dt:T;

x0 = 0; y0 = 0; psi0 = 0; X = [x0; y0; psi0];

v = 1.0; delta = 0; L = 2.0;

target_trajectory = [linspace(0, 10, length(t)); linspace(0, 0, length(t))];

x_history = zeros(3, length(t)); x_history(:, 1) = X;

for k = 1:length(t)-1

    X(1) = X(1) + v * cos(X(3)) * dt;

    X(2) = X(2) + v * sin(X(3)) * dt;

    X(3) = X(3) + (v * tan(delta) / L) * dt;

    x_history(:, k+1) = X;

end

error = sqrt((x_history(1, :) - target_trajectory(1, :)).^2 + (x_history(2, :) - target_trajectory(2, :)).^2);

rmse = sqrt(mean(error.^2));

disp(['轨迹跟踪均方根误差 (RMSE): ', num2str(rmse)]);

figure;

plot(x_history(1, :), x_history(2, :), 'b-', 'LineWidth', 2);

hold on;

plot(target_trajectory(1, :), target_trajectory(2, :), 'r--', 'LineWidth', 2);

quiver(x_history(1, :), x_history(2, :), cos(x_history(3, :)), sin(x_history(3, :)), 0.5, 'g');

xlabel('X (m)');

ylabel('Y (m)');

title('无人叉车运动轨迹');

legend('实际轨迹', '目标轨迹');

grid on;

axis equal;
```

附录 2 曲线运动学仿真 MATLAB 代码

```

dt = 0.1; T = 10; t = 0:dt:T

x0 = 0; y0 = 0; psi0 = 0; X = [x0; y0; psi0];

v = 1.0; delta = 0.2; L = 2.0;

R = L / tan(delta); theta = v * t / R;

x_target = R * sin(theta); y_target = R * (1 - cos(theta)); target_trajectory = [x_target; y_target];

x_history = zeros(3, length(t)); x_history(:, 1) = X;

for k = 1:length(t)-1

    X(1) = X(1) + v * cos(X(3)) * dt;

    X(2) = X(2) + v * sin(X(3)) * dt;

    X(3) = X(3) + (v * tan(delta) / L) * dt;

    x_history(:, k+1) = X;

end

error = sqrt((x_history(1, :) - target_trajectory(1, :)).^2 + (x_history(2, :) - target_trajectory(2, :)).^2);

rmse = sqrt(mean(error.^2));

disp(['轨迹跟踪均方根误差 (RMSE): ', num2str(rmse)]);

figure;

plot(x_history(1, :), x_history(2, :), 'b-', 'LineWidth', 2);

hold on;

plot(target_trajectory(1, :), target_trajectory(2, :), 'r--', 'LineWidth', 2);

quiver(x_history(1, :), x_history(2, :), cos(x_history(3, :)), sin(x_history(3, :)), 0.5, 'g');

xlabel('X (m)');

ylabel('Y (m)');

title('无人叉车运动轨迹');

legend('实际轨迹', '目标轨迹');

grid on;

axis equal;

```

附录 3 不同线速度及转向角运动学仿真 MATLAB 代码

```

dt = 0.1; T = 10; t = 0:dt:T;

x0 = 0; y0 = 0; psi0 = 0; X = [x0; y0; psi0];

v = 1.0; delta = 0.2; L = 2.0;

x_history = zeros(3, length(t)); x_history(:, 1) = X;

for k = 1:length(t)-1

    X(1) = X(1) + v * cos(X(3)) * dt;

    X(2) = X(2) + v * sin(X(3)) * dt;

    X(3) = X(3) + (v * tan(delta) / L) * dt;

    x_history(:, k+1) = X;

end

figure;

plot(x_history(1, :), x_history(2, :), 'b-', 'LineWidth', 2);

hold on;

quiver(x_history(1, :), x_history(2, :), cos(x_history(3, :)), sin(x_history(3, :)), 0.5, 'r');

xlabel('X (m)');

ylabel('Y (m)');

title('无人叉车运动轨迹');

grid on;

axis equal;

curvature = abs(tan(delta) / L);

disp(['理论曲率: ', num2str(curvature)]);

figure;

plot(t, x_history(3, :), 'g-', 'LineWidth', 2);

xlabel('时间 (s)');

ylabel('偏航角 (rad)');

title('偏航角随时间变化');

```

```
grid on;

psi_rate = diff(x_history(3, :)) / dt;

theory_psi_rate = v * tan(delta) / L;

disp(['仿真偏航角变化率: ', num2str(mean(psi_rate))]);

disp(['理论偏航角变化率: ', num2str(theory_psi_rate)]);

v_values = [0.5, 1.0, 1.5];

figure;

hold on;

for i = 1:length(v_values)

    v = v_values(i);

    X = [x0; y0; psi0];

    x_history_v = zeros(3, length(t));

    x_history_v(:, 1) = X;

    for k = 1:length(t)-1

        X(1) = X(1) + v * cos(X(3)) * dt;

        X(2) = X(2) + v * sin(X(3)) * dt;

        X(3) = X(3) + (v * tan(delta) / L) * dt;

        x_history_v(:, k+1) = X;

    end

    plot(x_history_v(1, :), x_history_v(2, :), 'DisplayName', ['v = ', num2str(v)]);

end

xlabel('X (m)');

ylabel('Y (m)');

title('不同线速度下的运动轨迹');

legend;

grid on;

axis equal;

hold off;
```

```

delta_values = [0.1, 0.3, 0.5];

v = 1.0;

figure;

hold on;

for i = 1:length(delta_values)

    delta = delta_values(i);

    X = [x0; y0; psi0];

    x_history_delta = zeros(3, length(t));

    x_history_delta(:, 1) = X;

    for k = 1:length(t)-1

        X(1) = X(1) + v * cos(X(3)) * dt;

        X(2) = X(2) + v * sin(X(3)) * dt;

        X(3) = X(3) + (v * tan(delta) / L) * dt;

        x_history_delta(:, k+1) = X;

    end

    plot(x_history_delta(1, :), x_history_delta(2, :), 'DisplayName', ['\delta = ', num2str(delta)]);

end

xlabel('X (m)');

ylabel('Y (m)');

title('不同转向角度下的运动轨迹');

legend;

grid on;

axis equal;

hold off;

target_trajectory = [linspace(0, 10, length(t)); linspace(0, 0, length(t))];

error = sqrt((x_history(1, :) - target_trajectory(1, :)).^2 + (x_history(2, :) - target_trajectory(2, :)).^2);

rmse = sqrt(mean(error.^2));

disp(['轨迹跟踪均方根误差 (RMSE): ', num2str(rmse)]);

```

附录 4 转向角随时间变化运动学仿真 MATLAB 代码

```

dt = 0.1; T = 10; t = 0:dt:T;

x0 = 0; y0 = 0; psi0 = 0; X = [x0; y0; psi0];

v = 1.0; L = 2.0;

delta = @(t) 0.2 * sin(0.5 * t);

x_target = zeros(1, length(t));
y_target = zeros(1, length(t));
psi_target = zeros(1, length(t));

for k = 1:length(t)

    if k == 1

        x_target(k) = x0;

        y_target(k) = y0;

        psi_target(k) = psi0;

    else

        R = L / tan(delta(t(k)));

        dpsi = (v / R) * dt;

        psi_target(k) = psi_target(k-1) + dpsi;

        x_target(k) = x_target(k-1) + v * cos(psi_target(k)) * dt;

        y_target(k) = y_target(k-1) + v * sin(psi_target(k)) * dt;

    end

end

target_trajectory = [x_target; y_target];

x_history = zeros(3, length(t));

x_history(:, 1) = X;

for k = 1:length(t)-1

    delta_k = delta(t(k));

    X(1) = X(1) + v * cos(X(3)) * dt;

```

```
X(2) = X(2) + v * sin(X(3)) * dt;

X(3) = X(3) + (v * tan(delta_k) / L) * dt;

x_history(:, k+1) = X;

end

error = sqrt((x_history(1, :) - target_trajectory(1, :)).^2 + (x_history(2, :) - target_trajectory(2, :)).^2);

rmse = sqrt(mean(error.^2));

disp(['轨迹跟踪均方根误差 (RMSE): ', num2str(rmse)]);

figure;

plot(x_history(1, :), x_history(2, :), 'b-', 'LineWidth', 2);

hold on;

plot(target_trajectory(1, :), target_trajectory(2, :), 'r--', 'LineWidth', 2);

quiver(x_history(1, :), x_history(2, :), cos(x_history(3, :)), sin(x_history(3, :)), 0.5, 'g');

xlabel('X (m)');

ylabel('Y (m)');

title('无人叉车运动轨迹');

legend('实际轨迹', '目标轨迹');

grid on;

axis equal;

figure;

plot(t, x_history(3, :), 'g-', 'LineWidth', 2);

xlabel('时间 (s)');

ylabel('偏航角 (rad)');

title('偏航角随时间变化');

grid on;
```

附录 5 理想状态下数据融合运动学仿真 MATLAB 代码

```

dt = 0.1; T = 10; t = 0:dt:T;

x0 = 0; y0 = 0; psi0 = 0; X = [x0; y0; psi0]; X_est = X;

v = 1.0; delta = 0.2; L = 2.0;

x_history = zeros(3, length(t)); x_history(:, 1) = X;

for k = 1:length(t)-1

    X(1) = X(1) + v * cos(X(3)) * dt;

    X(2) = X(2) + v * sin(X(3)) * dt;

    X(3) = X(3) + (v / L) * tan(delta) * dt;

    p_L = generateLidarData(X);

    [a_meas, omega_meas] = generateIMUData(v, delta, L);

    X_est = ekfFusion(X_est, p_L, a_meas, omega_meas);

    x_history(:, k+1) = X;

end

X_theory = [x0; y0; psi0];

x_theory = zeros(3, length(t));

x_theory(:, 1) = X_theory;

for k = 1:length(t)-1

    X_theory(1) = X_theory(1) + v * cos(X_theory(3)) * dt;

    X_theory(2) = X_theory(2) + v * sin(X_theory(3)) * dt;

    X_theory(3) = X_theory(3) + (v / L) * tan(delta) * dt;

    x_theory(:, k+1) = X_theory;

end

figure;

plot(x_history(1, :), x_history(2, :), 'b-', 'LineWidth', 2, 'DisplayName', '仿真轨迹');

hold on;

plot(x_theory(1, :), x_theory(2, :), 'r--', 'LineWidth', 2, 'DisplayName', '理论轨迹');

```

```

xlabel('X (m)');

ylabel('Y (m)');

title('理论轨迹与仿真轨迹对比');

legend;

grid on;

axis equal;

error_x = x_theory(1, :) - x_history(1, :);

error_y = x_theory(2, :) - x_history(2, :);

error_pos = sqrt(error_x.^2 + error_y.^2);

figure;

plot(t, error_pos, 'k-', 'LineWidth', 2);

xlabel('时间 (s)'); ylabel('位置误差 (m)'); title('理论轨迹与仿真轨迹的位置误差');

grid on;

function p_L = generateLidarData(X)

    obstacle_global = [X(1) + 1*cos(X(3));
                       X(2) + 1*sin(X(3))];

    R = [cos(X(3))  sin(X(3));
         -sin(X(3)) cos(X(3))];

    p_L = R * (obstacle_global - [X(1); X(2)]);

end

function [a_meas, omega_meas] = generateIMUData(v, delta, L)

    noise_level = 0.02;

    a_meas = 0 + noise_level*randn;

    omega_meas = (v / L) * tan(delta) + noise_level*randn;

end

function X_est = ekfFusion(X_est, p_L, a_meas, omega_meas)

    X_est(3) = X_est(3) + omega_meas * 0.1;

end

```

附录 6 噪声误差下数据融合运动学仿真 MATLAB 代码

```

dt = 0.1; T = 10; t = 0:dt:T;

x0 = 0; y0 = 0; psi0 = 0; X = [x0; y0; psi0]; X_est = X;

v = 1.0; phi = 0.2; L = 2.0;

x_history = zeros(3, length(t)); x_history(:, 1) = X;

for k = 1:length(t)-1

    X(1) = X(1) + v * cos(X(3)) * dt;

    X(2) = X(2) + v * sin(X(3)) * dt;

    X(3) = X(3) + (v / L) * tan(phi) * dt;

    p_L = generateLidarData(X);

    [a_meas, omega_meas] = generateIMUData(v, (v / L) * tan(phi));

    X_est = ekfFusion(X_est, p_L, a_meas, omega_meas, dt, v, L, phi);

    x_history(:, k+1) = X_est;

end

X_theory = [x0; y0; psi0]; x_theory = zeros(3, length(t)); x_theory(:, 1) = X_theory;

for k = 1:length(t)-1

    X_theory(1) = X_theory(1) + v * cos(X_theory(3)) * dt;

    X_theory(2) = X_theory(2) + v * sin(X_theory(3)) * dt;

    X_theory(3) = X_theory(3) + (v / L) * tan(phi) * dt;

    x_theory(:, k+1) = X_theory;

end

figure;

plot(x_history(1, :), x_history(2, :), 'b-', 'LineWidth', 2, 'DisplayName', '仿真轨迹');

hold on;

plot(x_theory(1, :), x_theory(2, :), 'r--', 'LineWidth', 2, 'DisplayName', '理论轨迹');

xlabel('X (m)'); ylabel('Y (m)'); title('理论轨迹与仿真轨迹对比');

legend;

```

```

grid on;

axis equal;

error_x = x_theory(1, :) - x_history(1, :);

error_y = x_theory(2, :) - x_history(2, :);

error_pos = sqrt(error_x.^2 + error_y.^2);

figure;

plot(t, error_pos, 'k-', 'LineWidth', 2);

xlabel('时间 (s)'); ylabel('位置误差 (m)'); title('理论轨迹与仿真轨迹的位置误差');

grid on;

function p_L = generateLidarData(X)

    obstacle_global = [X(1) + 1 * cos(X(3));
                       X(2) + 1 * sin(X(3))];

    R = [cos(X(3))  sin(X(3));
         -sin(X(3)) cos(X(3))];

    p_L = R * (obstacle_global - [X(1); X(2)]);

    noise_level = 0.05;

    p_L = p_L + noise_level * randn(2, 1);

end

function [a_meas, omega_meas] = generateIMUData(v, omega)

    noise_level = 0.1;

    a_meas = 0 + noise_level * randn;

    omega_meas = omega + noise_level * randn;

end

function X_est = ekfFusion(X_est, p_L, a_meas, omega_meas, dt, v, L, phi)

    X_est(3) = X_est(3) + omega_meas * dt;

    X_est(1) = X_est(1) + v * cos(X_est(3)) * dt + 0.02 * randn;

    X_est(2) = X_est(2) + v * sin(X_est(3)) * dt + 0.02 * randn;

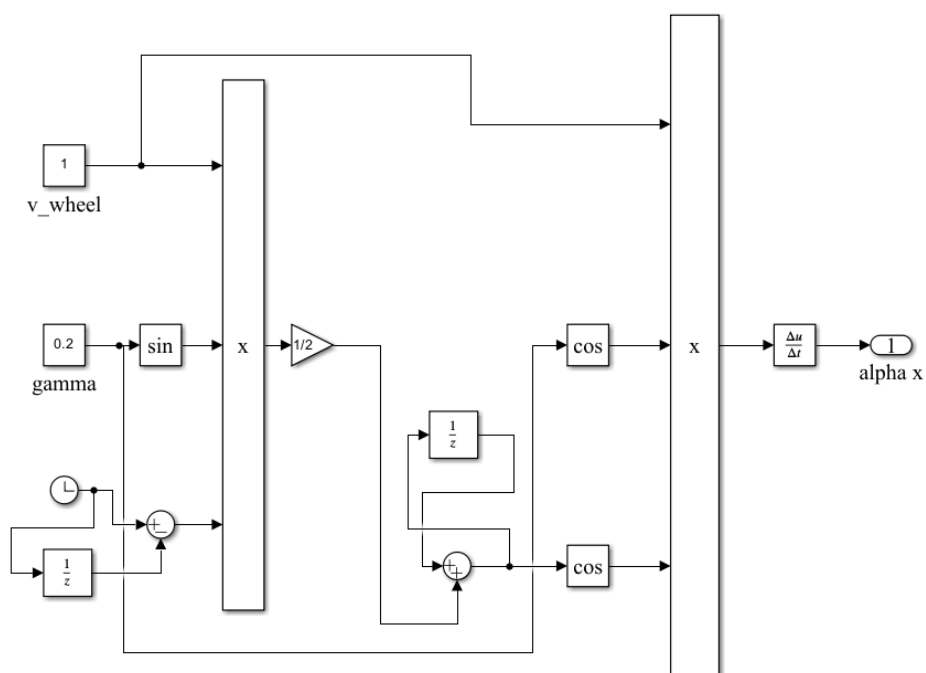
end

```

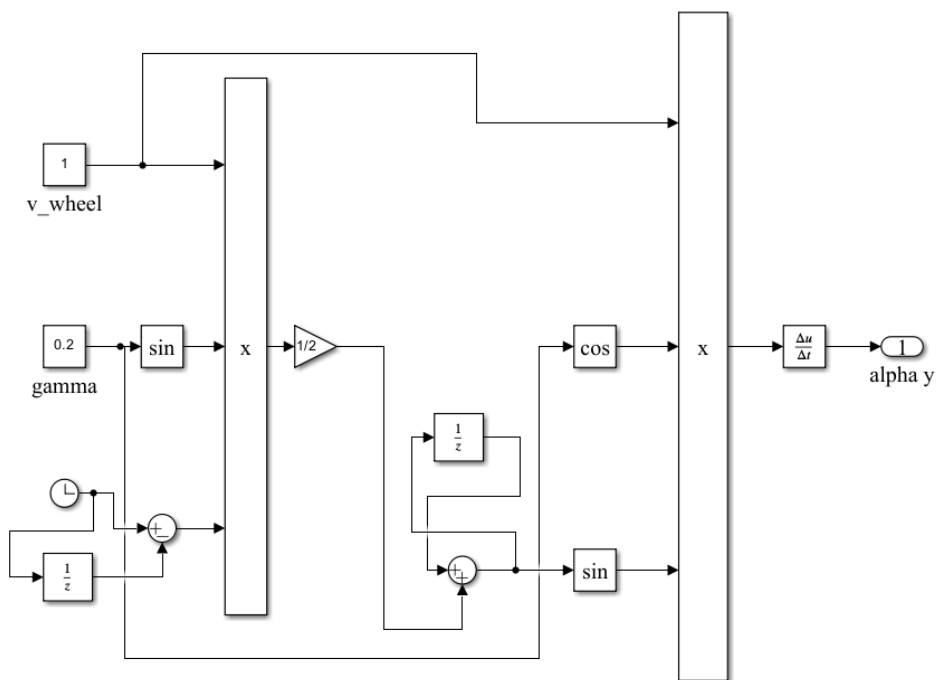
附录 7 Simulink 仿真子模块

3.5.5 节中的数据融合系统 Simulink 仿真模型图中的子模块。

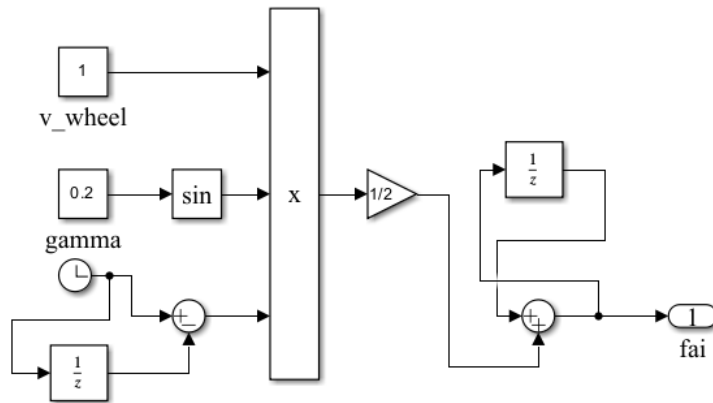
X 轴加速度 α_x :



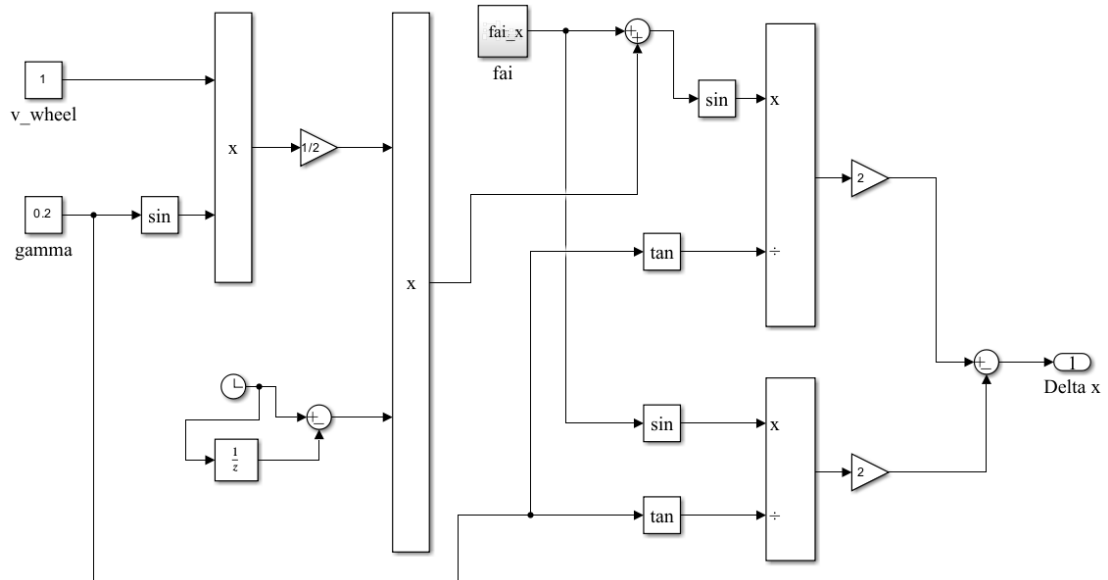
Y 轴加速度 α_y :



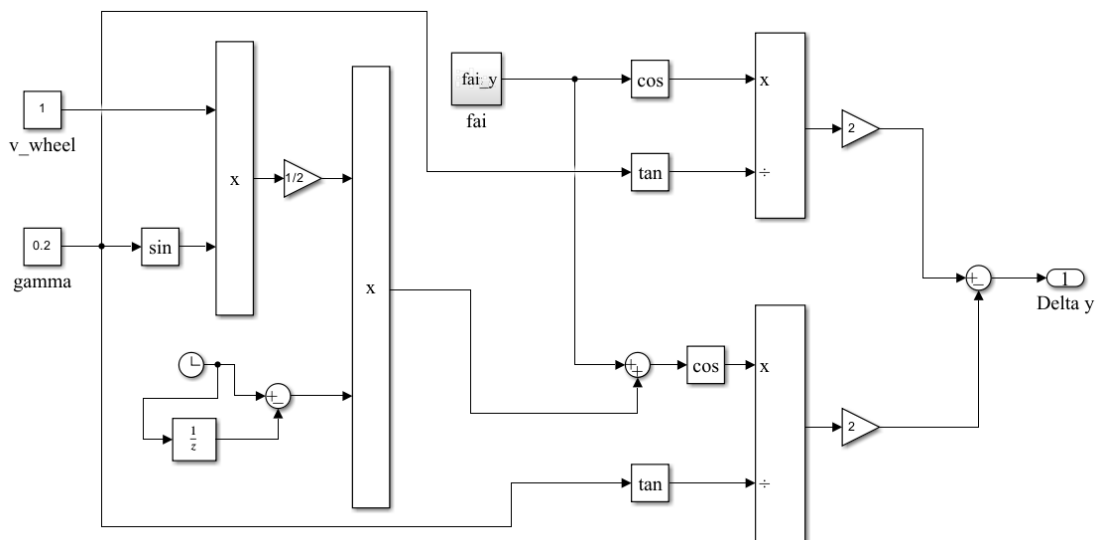
偏航角 φ :



X轴瞬时变化 Delta x:



Y轴瞬时变化 Delta y:



攻读硕士专业学位期间学术成果

论文发表情况：

- [1] 程军, 毛伟, 汪步云, 许德章, 杨秋生. 基于改进 DLO 算法的无人叉车同时定位与建图[J]. 电子测量技术(已见刊, 本人二作)

参与项目：

- [1] 安徽省经信委“揭榜挂帅项目”：“新一代嵌入式无人叉车专用控制器及片上系统研发”(JB22031)。
- [2] 徐州徐工特种工程机械有限公司横向项目：“搬运型 AGV 运动控制及作业调度系统研发”。
- [3] 安徽省重点研发与成果转化项目：“堆垛式叉车 AGV 的 3D 激光导航控制与多车调度系统研发”。

致谢

时光荏苒，三年的研究生时光即将悄然过去。回首往昔，研究生生涯始于壬寅，终于乙巳，千日光阴，转瞬即逝。无论是身边的点滴，还是心中的沉淀，所有的回忆都如潮水般涌来。纵使有万般不舍，心中依旧充满感激。

“令公桃李满天下，何用堂前更种花。”得遇良师，实乃我之幸。首先，我要衷心感谢我的导师程军教授，在我三年的研究生求学过程中，给予了我莫大的帮助与指导。程教授在教学中严谨耐心、充满激情；在科研指导上，不仅传授了我宝贵的知识，更注重培养我独立思考与创新的能力，使我在这段求学旅程中收获颇丰。程教授对学生的指导始终充满耐心与包容，总是细心引导我们在研究过程中保持严谨与细致，敢于探索，勇于质疑，并积极面对失败与挫折。在学习和科研中，程教授为我们提供了良好的学术环境与氛围，使我们能够迅速而全面地投入到科研工作中，并取得了显著的进步。

千金易得，知己难求。接下来，我要感谢我的同门师兄弟们：刘蒙蒙、朱煜龙、张梦茹、周燕、江萌、秦朋，感谢他们在过去三年中在学习上的陪伴与帮助，让我在求学的道路上充满了动力。还要特别感谢我的室友毛徐徐、郑乐、王龙，感谢他们在这三年朝夕相处的日子里给予我的包容与支持。我们一起学习、一起成长，彼此鼓励、共同进步，创造了无数美好的回忆，也建立了深厚的友谊。这段时光将永远珍藏在我心中，成为我宝贵的人生财富。

谁言寸草心，报得三春晖。接下来，我要深深感谢我的父母，感谢他们二十余载的辛勤养育与无私付出。是他们在我求学路上的坚定支持与无条件的爱，让我能够心无旁骛地追求学业与梦想。每当遭遇困难与挫折，是他们的鼓励与信任让我重新振作，坚定步伐，继续前行。无论我身处何方，他们始终是我坚实的后盾。他们教会了我什么是坚韧、责任与感恩，也让我深刻理解了追求卓越、不断向前的力量。感谢他们给予我的一切，这份恩情我将永远铭记在心。

最后，衷心感谢各位评审专家和答辩组的老师们，感谢你们的真诚付出与悉心指导。你们提出的宝贵建议和意见对我的论文完善起到了至关重要的作用，也让我在学术研究的道路上受益匪浅。再次感谢你们的支持与帮助！

尚德敏学 唯实惟新

