
南京理工大学泰州科技学院

毕业设计说明书(论文)

作 者：张富高 学 号：1901050146

学院(系)：电子电气工程学院

专 业：电气工程及其自动化

题 目：基于 ROS 的机器人路径规划与导航

系统设计

指导者：周军 副教授

评阅者：张顺 高级工程师

2023 年 5 月

毕业设计说明书（论文）中文摘要

今天和技术全球化的大背景下，人工智能迅速发展。其中轮式机器人的研究与应用已广泛渗透各行各业。在该领域中，即时定位与地图构建（SLAM），路径规划和导航成为移动机器人三大核心问题。同时，移动机器人普遍存在一些问题，例如：如何使用低成本激光雷达构建精确的二维栅格地图；如何迅速地根据地图规划出准确的路径；如何提高导航精度，准确到达目的地等等。

针对以上问题，通过调研国内外研究现状，本文设计了一种基于麦克纳姆轮底盘的机器人硬件平台，并根据移动机器人的运动学模型建立相应的数学模型。在 ROS 中搭建软件框架，利用 EKF(扩展卡尔曼滤波算法)对单线激光雷达和 IMU（惯性测量单元）进行数据融合实现 SLAM 技术；使用机器人构建的全局地图，通过应用 AMCL 自适应蒙特卡罗粒子滤波定位算法、A*全局路径规划算法以及 DWA 和 TEB 局部路径规划算法进行研究。

最后，在实验室的复杂环境中，通过多次训练和实验，移动机器人成功地利用 Gmapping 算法建立了一幅地图，并通过 Rviz 实现了指定目标点导航的可视化图像。此外，设计还验证了移动机器人在定位、路径规划和避障导航方面的能力。通过 SLAM、路径规划和导航关键点的研究，成功地搭建了一个具有实际应用价值的自主路径规划和导航系统。该系统具有成本低、准确性高等诸多优点。

关键词 移动机器人 ROS 路径规划 导航

Title The Design of Robot Path Planning and Navigation

System Based on ROS

Abstract

Today, in the context of global technological globalization, artificial intelligence has rapidly developed, and the research and application of wheeled robots have been widely infiltrating various industries. In this field, simultaneous localization and mapping (SLAM), path planning and navigation have become the three core issues of mobile robots. At the same time, there are some common problems for mobile robots, such as how to use low-cost LIDAR to construct accurate 2D grid maps, how to plan accurate paths quickly based on the maps, and how to improve navigation accuracy to reach the destination accurately.

To address these issues, this paper designs a robot hardware platform based on Mecanum wheel chassis, and establishes a corresponding mathematical model according to the mobile robot's kinematic model. In ROS, a software framework is built to implement SLAM technology by using EKF (extended Kalman filter algorithm) to fuse data from single-line LIDAR and IMU (inertial measurement unit). The research then utilized the mobile robot's constructed global map to conduct further research using AMCL adaptive Monte Carlo particle filtering localization algorithms, A* global path planning algorithms, and DWA and TEB local path planning algorithms.

Finally, in the complex environment of the laboratory, the mobile robot successfully used Gmapping algorithm to build a map, and implemented the visualization of the navigation of designated target points using

Rviz after multiple training and experiments. Further, the research also validated the mobile robot's ability in localization, path planning and obstacle avoidance navigation. By studying SLAM, path planning and navigation key points, the research team successfully built an autonomous path planning and navigation system with practical value, which has many advantages such as low cost and high accuracy.

Keywords Mobile robot ROS Path planning Navigation

目 录

1 引言	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.3 本章小结	2
2 硬件系统设计	3
2.1 引言	3
2.2 机器人硬件设计	4
2.3 传感器设计	5
2.4 本章小结	6
3 机器人即时定位和地图构建算法研究	7
3.1 引言	7
3.2 常用的地图表示法	7
3.3 SLAM 原理	8
3.4 本章小结	9
4 机器人自主导航算法研究	10
4.1 引言	10
4.2 已知环境下的机器人定位	10
4.3 代价地图	10
4.4 路径规划	11
4.5 本章小结	15
5 实验验证	16
5.1 实验环境介绍	16
5.2 SLAM 实验	17
5.3 自主导航实验	18
5.4 本章小结	19
结束语	20
致谢	21
参考文献	22
附录 A 源程序（摘要）	23
附录 B 在学期间取得的成果	27

1 引言

智能移动机器人受到市场的认可，主要原因在于它能够提供更高效和便捷的服务，自动完成重复性任务，节省人力成本，减少人员接触并在突发事件如疫情中提供有效服务，从而满足人们对高效、便捷和安全的需求。

1.1 研究背景及意义

移动机器人现以具有自主感知、传感技术、动态决策、行动控制和执行功能的形式呈现。运动方面，基于麦克纳姆轮的移动机器人具有更强的狭窄空间内位置调整能力，更加精确的定位，可以在平面内进行无约束的全向移动，优于传统的轮式差速移动机器人^[1]。

1.2 国内外研究现状

本节从移动机器人的发展现状，SLAM 技术研究现状和路径规划技术现状三方面入手，详细介绍了现阶段机器人路径规划与自主导航系统发展的现状。

1.2.1 移动机器人发展现状

在国内，中国政府高度重视机器人的研究和发展。2021 年，中国政府在“十四五”规划和 2035 年远景目标纲要提出《机器人产业发展规划（2021-2025 年）》^[2]，明确了机器人产业的目标和措施，对推动机器人产业发展作出了明确部署。在这份规划中，对于移动机器人领域特别提出了推广应用的重点，并明确了在工业、物流、农业、服务等领域推广移动机器人应用的目标。在国际上，移动机器人的研究和发展也受到了高度重视。美国是全球机器人研究和发展的领导者之一，其移动机器人的研究主要集中在医疗、物流和服务军事等领域。例如美国的 Fetch Robotics 公司曾在 2018 年推出了一款名为“Fetch”的自动搬运机器人，用于在仓库和物流中心内进行货物搬运。

1.2.2 SLAM 技术研究现状

SLAM (Simultaneous Localization and Mapping) 技术是机器人导航、自主导航、虚拟现实等领域的重要研究方向。目前，SLAM 技术已经取得了长足的进展，在环境映射、精确定位、机器人路径规划等方面都取得了显著的突破。激光 SLAM 是目前主流的算法。

激光 SLAM 技术的发展始于 20 世纪 90 年代末，当时的研究主要集中在如何使用激光雷达来建立环境的地图，并同时实现机器人的定位。目前，激光雷达的类型主要有两种：单线程和多线程。前者常用于室内移动机器人，而后者则被广泛应用于无人车。随着机器人技术和传感技术的不断提升，激光 SLAM 在自主导航、环境感知、机器人规划等领域越来越普遍。然而，由于环境光线强度的波动、干扰因素等问题，激光 SLAM 的精度和稳定性仍需提高。因此，该技术仍在积极发展和完善中。

1.2.3 路径规划技术现状

路径规划技术，它用于确定从一个位置到另一个位置的最短或最优路径^[3]。根据机器人对环境信息掌握程度的不同，路径规划算法可分为全局路径规划算法与局部路径规划算法。目前国内外在这种技术都普遍面临一些问题，例如：复杂的环境对路径规划带来的困难，例如在体育场等复杂环境下寻找最优路径较为困难；一些环境下环境信息收集不够充分，导致路径规划算法无法得到足够的数据，从而降低了算法的准确性，如何在路径规划过程中考虑到人类因素，以便机器人能够更好地与人类协作等等^[4]。

1.3 本章小结

本章介绍了本研究的背景、意义，以及国内外相关研究现状。针对机器人路径规划和导航在实际应用中的重要性进行了详细探讨，并总结了当前最新的 SLAM 技术和路径规划技术研究进展与优劣。本章将为论文研究提供理论基础、研究思路和方法的启示，并为未来相关领域的研究提供重要参考。

2 硬件系统设计

2.1 引言

全向移动机器人是目前机器人研究领域的焦点，它因其卓越的运动性能和精确的运动精度而在室内场景中被广泛使用。在本章中，我们将介绍全向移动机器人自主导航的建模，构建系统框架，详细介绍系统中的硬件部分。系统整体设计框架如图 2.1。

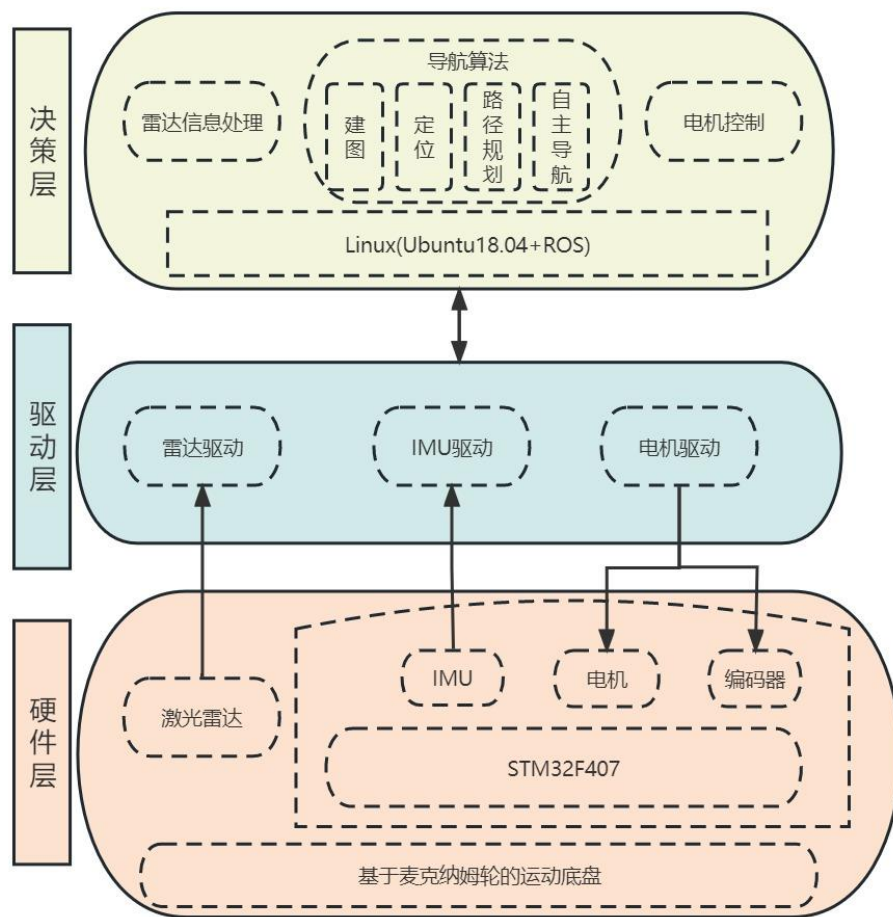


图 2.1 系统整体设计框架

硬件层的构成主要包括机器人的机械平台、控制单元和外围模块。机械平台使用了麦克纳姆轮，赋予机器人全向移动的能力。底盘上装有 NVIDIA Jetson nano 和 STM32F407 两个控制单元，它们之间通过串行通信进行数据交换。Jetson nano 运行 Linux 系统，在 ROS 操作系统下收集并处理激光雷达数据，而 STM32F407 则是控制机器人运动的核心部分，接收上位机发送的决策信号，驱动平台的运动。决策层主要负责传感器信息采集、导航算法的运行和电机控制。其中导航算法包括建图算法、定位算法和规划算法三个部分。这些算法在 Jetson nano 的 ROS 系统中运行，基

于 ROS 导航框架对导航算法进行研究和改进。整个硬件层的驱动层则负责管理机器人的各个外接硬件设备的运作，具体包括激光雷达、IMU 和电机的驱动等，传感器和相应的驱动接口之间则通过串口通信进行通讯。

2.2 机器人硬件设计

2.2.1 底盘核心控制板

机器人底盘核心控制板采用大疆推出的一款移动机器人开发板 RoboMaster 开发板，配备了 STM32F407 芯片，拥有 168MHz 的主频，192KB 的 RAM 和 1MB 的 Flash 存储；MPU6050 六轴陀螺仪加速度计，可以对机器人进行精确的姿态检测。还配备了丰富的接口，如串口、I2C、SPI、CAN 等，支持各种外部设备的接入和数据交互。搭载 STM32F407 的 RoboMaster 开发板如图 2.2 所示。

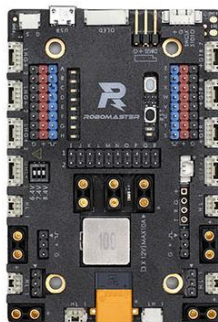


图 2.2 搭载 STM32F407 的 RoboMaster 开发板

2.2.2 底盘运动学模型

麦克纳姆轮（Mecanum wheel），也称艾隆轮（Ilon wheel），是一种全向轮，由瑞典麦克纳姆公司的工程师 Bengt Erland Ilon 发明而得名^[5]。麦克纳姆轮是将排列成 45 度角的滚轮安装于车轮外侧，使其在工作时向轴心方向产生 45 度的反推力，这种力分为垂直和水平方向。一辆麦克纳姆轮车由两对照片对称的车轮组成，每个车轮都独立产生相应的推力，这些推力的综合决定车辆的动态状态^[6]。通过调整每个车轮的独立转向和转速，车辆可以实现前进、横移、斜行、旋转等各种运动方式，甚至组合运动，图 2.3 所示为实物图。



图 2.3 麦克纳姆轮

2.2.3 计算机处理模块

机器人采用 NVIDIA 公司推出的 Jetson Nano 作为上位机。NVIDIA Jetson Nano 是一款由 NVIDIA 推出的嵌入式计算机，常用于边缘计算和机器人控制等领域。它具有高性能的四核 Arm Cortex-A57 CPU，128 核 Nvidia Maxwell GPU，4GB 的 LPDDR4 内存等优秀特点。图 2.4 为 Jetson Nano 实物图。



图 2.4 Jetson Nano

2.3 传感器设计

2.3.1 IMU

设计采用 MPU6050 六轴陀螺仪加速度计。IMU 是指惯性测量单元，用于测量物体的线性速度和角速度。IMU 通常由加速度计和陀螺仪组成，其中加速度计测量物体的线性加速度，陀螺仪则测量物体在旋转轴上的角速度。加速度计利用牛顿第二定律来测量物体的加速度，陀螺仪则利用柯氏定理来测量旋转轴上的角速度^[7]。

2.3.2 激光雷达

设计采用思岚 A1 雷达，实物图如图 2.5 所示。激光雷达通过激光束发射并接收反射回来的光信号，利用时间差法或调频连续波法计算物体的距离和速度，从而得到物体在三维空间中的位置信息。激光雷达有高精度、高鲁棒性、适用于各种复杂环境等优点。通常应用于机器人导航、智慧交通、三维建模、环境检测和避障等领域^[8]。此外，激光雷达还可以与相机、GPS 和 IMU 等传感器相结合，构建高精度的感知系统。



图 2.5 激光雷达

2.4 本章小结

本章主要介绍了机器人硬件设计，包括机器人硬件结构和动力学模型以及传感器设计。在机器人硬件设计中，我们详细介绍了麦克纳姆轮机器人的运动学模型。而在传感器设计中，我们详细介绍了各种传感器的特点和原理。本章的内容是硬件系统设计的基础，对于更好地了解和掌握机器人系统是非常必要的。

3 机器人即时定位和地图构建算法研究

3.1 引言

本章节介绍机器人即时定位和地图构建的相关技术，主要涵盖了常见地图表示法和 SLAM 原理。SLAM 技术是机器人即时定位的重要方法，它是通过利用传感器获取的环境数据建立地图并计算机器人位置的过程。SLAM 包含多种算法，本章将详细介绍几种常见的算法，并综合对照各种算法的优缺点，根据本课题的需求选择出最适合的算法。

3.2 常用的地图表示法

在机器人领域，有几种表示地图的方法，包括概率占位图（POMs）以及栅格地图^[9]等。

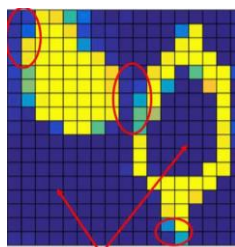
（1）概率占位图

概率占位网格图是机器人学中常见的地图表示方法。它们是一个单元格，每个单元格代表它被障碍物占据的概率。这种表示方法的优点是计算效率高且易于使用，因为它可以直接用于常见的机器人规划算法。它还允许在同一网格内有多个概率，从而可以表示地图中的不确定性。这种表示法的主要缺点是，它在表示环境中的复杂形状和结构方面能力有限，而且随着不确定性的增加，地图会变得不精确。如图 3.1（1）所示。

（2）栅格地图

栅格地图以单元划分的方式来描述环境，不需要依靠物体的三维形状，机器人的通行能力则取决于激光雷达扫描到的平面图栅格是否被障碍物占据。栅格地图具有对平面中细小物体的表示具有高精度，容易被系统识别的优点。缺点是存在误差累积，对于一些复杂的环境可能难以准确表示。如图 3.1（2）所示。

综合对比以上两种地图表示法的优缺点，结合本设计常用场景为封闭的平面室内场所，因此选择栅格地图作为 SLAM 的地图表示法。



（1）概率占位图



（2）栅格地图

图 3.1 常用的地图表示法

3.3 SLAM 原理

SLAM (Simultaneous Localization and Mapping) 是机器人移动时同时获取定位信息和地图信息的一类技术。它的工作原理是机器人通过获取多种数据（如激光雷达，IMU，摄像头等）进行定位与地图构建^[10]。目前有几种流行的激光雷达 SLAM 算法，包括 Gmapping 和 Cartographer 等。

(1)Gmapping

Gmapping 算法是一种可以在机器人导航和自动驾驶任务中广泛应用的概率式 SLAM 算法。其实现主要分为三个步骤：地图建立、机器人位置估计和数据关联。

在第一阶段中，机器人搭载的激光雷达设备会扫描环境，并基于扫描数据生成概率地图。这个地图可以分为很多个小区域，每个区域都被称为二元栅格，栅格内部包含了关于区域内物体占用情况的概率信息。

在第二阶段中，采用了粒子滤波算法来估计机器人在环境中的位置。换言之，通过大量的假设来近似目标状态的分布，从而实现在未知环境中的机器人位置估计。

第三阶段中，数据关联算法被用于将机器人激光雷达扫描数据与建立的概率地图进行匹配和比对。该算法的核心是基于格子匹配和激光雷达扫描序列匹配。同时，通过匹配得分对匹配结果进行排序，找到最优匹配，并更新机器人当前位置和建立的概率地图。

Gmapping 适用于具有丰富结构和纹理的环境，其中激光测距数据可以提供足够的信息来建立一个地图。它有一个相对较快的收敛速度，并能处理动态环境。然而，该算法在无特征的环境中可能效果不佳，例如那些具有平坦表面和均匀纹理的环境，因为它严重依赖激光测距数据。此外，Gmapping 可能无法处理大比例尺的地图，而且可能会随着时间的推移受到累积漂移的影响。如图 3.2 所示为 Gmapping 算法流程图。

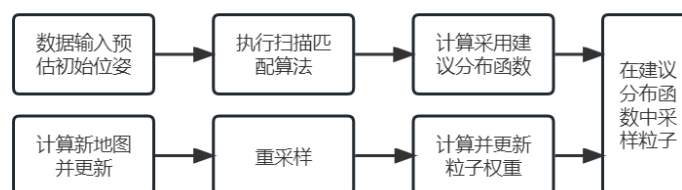


图 3.2 Gmapping 算法流程图

(2)Cartographer

Cartographer 是一个实时同步定位和绘图（SLAM）库，能够使用来自不同来源

的数据，包括激光测距仪、IMU 和车轮测距。Cartographer 遵循 SLAM 算法的基本结构，它使用基于图形的优化方法来估计机器人的姿势并建立环境地图，有前端处理和后端优化^[11]。前端负责数据关联和循环闭合检测。数据关联与各帧之间的输入传感器信息相匹配，以获得各帧之间的相对姿态，解决各帧之间的匹配和姿态估计问题^[14]。循环闭合检测的主要目的是检测机器人的当前姿势与之前访问过的区域的姿势之间的匹配问题。SLAM 的后端主要解决来自前端的误差，优化前端建立的姿态图，以获得最佳的姿态估计，或最佳的姿态序列。

Cartographer 的优点包括它能够与多种类型的传感器一起工作，使用高效的算法进行快速和准确的绘图和定位，以及处理大规模复杂环境的能力。Cartographer 的一个局限的点在于它需要大量的计算资源，使它不太适合在资源有限的设备上使用。此外，该库的配置和设置可能具有挑战性，特别是对于不太熟悉 SLAM 或基于图的优化的用户。图 3.3 所示为 Cartographer 示例。



图 3.3 Cartographer

综合对比各算法的优缺点，本文选择 **Gmapping** 作为机器人的 SLAM 算法。首先，**Gmapping** 有一个简单的实现，易于使用。其次，它已经在许多研究中被广泛使用，并被证明在各种环境下都很稳定。此外，**Gmapping** 在计算效率和映射精度之间提供了良好的平衡，使其适合在资源有限的平台上实时实施。

3.4 本章小结

本章介绍了与机器人实时定位和绘图有关的技术，涵盖了常见的地图表示方法和 SLAM 的原理。SLAM 是机器人实时定位的重要方法，它涉及使用从传感器获得的数据建立地图并计算机器人的位置，这对本系统机器人路径规划和自主导航的实现起着基础性的作用。

4 机器人自主导航算法研究

4.1 引言

本设计将机器人的自主导航系统分为两个主要部分：定位和路径规划。本章将介绍包括机器人在已知环境中的定位、代价地图和路径规划的关键概念。关于代价地图的部分将深入探讨以自主导航的目的创建成本图的问题。然后，路径规划部分将被分成两部分，即全局路径规划和局部路径规划，以全面了解自主导航中使用的不同技术和方法。

4.2 已知环境下的机器人定位

在已知环境下进行机器人定位的基本条件是拥有准确的地图信息和关于机器人姿态的估计。在这种情况下，机器人定位可以被看作是一种坐标转换，其中将机器人的位置与地图的坐标进行对齐。其中，Adaptive Monte Carlo Localization (AMCL)对MCL 算法进行了优化，借助激光雷达的扫描数据，实时获取无人驾驶车的位姿，并用粒子不断地进行概率估计，且可视化成箭头标识^[12]。它是一种概率算法，使用粒子过滤器来表示机器人的可能状态。AMCL 的基本思想是在整个地图上对一组粒子进行采样，并估计每个粒子是机器人真实状态的概率。如图 4.1 所示为 AMCL 的实际应用。

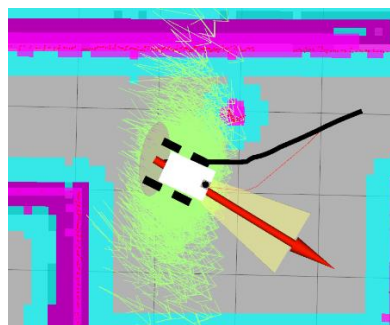


图 4.1 AMCL 算法匹配图

4.3 代价地图

在现实世界的场景中，障碍物会改变它们的位置，这就是为什么通过纳入激光雷达等传感器的实时反馈，拥有一个最新的代价地图是至关重要的。

代价地图（Costmap）是多个地图层的组合，代表了机器人周围的环境，包括静态障碍物、动态障碍物和机器人的膨胀足迹，所有这些都用来为机器人规划安全和高效的路径这可以确保机器人在运动过程中能够做出明智的决定并避开障碍物。

根据 Move_Base 包的 Yaml 文件中定义的数值范围，代价地图被划为障碍物、安全和未知区域。如图 4.2 所示为代价地图。粉红色区域表示静态障碍物。为了防止机器人与障碍物碰撞，需要将机器人的垂直投影所在的栅格与成本图中的黑色区域做区分，机器人的垂直投影不能和黑色区域相交。此外，为了提高安全性，需要对机器人的垂直投影所在的栅格进行膨胀操作，得到膨胀区域，机器人的运动路径不能与膨胀区域相交。因此，黑色区域和膨胀区域的设置都需要考虑机器人的几何形状和实际尺寸，以保证机器人在栅格地图中的运动安全。

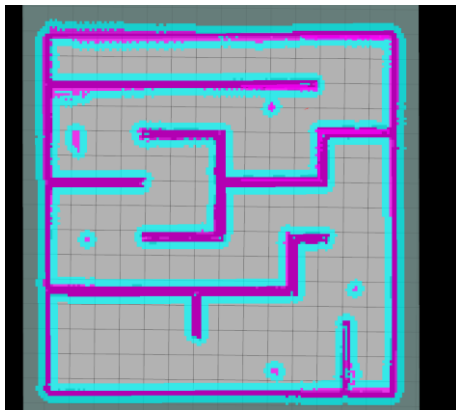


图 4.2 代价地图

4.4 路径规划

路径规划的重点在于如何在已知地图上找到最优路径，并在遇到移动中的障碍物时能够及时规避。这通常包括全局路径规划和局部路径规划两个方面^[13]。全局路径规划需要考虑地图上所有障碍物和路径约束条件，并在计算中尽量优化路径长度和安全性。而局部路径规划则需要考虑移动机器人周围环境的实时变化，并能够快速做出规避移动中障碍物的决策。

4.4.1 全局路径规划

全局路径规划是指在机器人的整个导航区域中，寻找一条能够连接起点和终点的最优路径的过程，该路径是在范围限制的环境中使机器人从一个初始位置到达一个预定目标位置的路径。全局路径规划通常采用启发式算法，可以在未来相对较长的时间内计算出一条最优路径，从而使机器人能够较高效地完成任务。目前主流的算法有 Dijkstra 算法和 A* 算法。

(1) Dijkstra 算法

Dijkstra 算法是求单源最短路径的算法，它的基本思想是贪心策略，每次选取距离源点最近的点，并从该点出发更新与之相邻的点的最短距离。执行过程如下：

首先，选定源点并将其到源点的距离设为 0，然后将源点加入已经确定最短路径的点的集合中。然后，从图中所有未确定最短路径的点中选取一个距离源点最近的点，将其加入已确定最短路径的点的集合中，并更新从该点出发到其他点的最短距离。具体地，对于每个确定最短路径的点 u ，遍历图中所有与 u 相邻的未确定最短路径的点 v ，更新 v 的最短距离，即：

$$d(v) = \min(d(v), d(u) + w(u, v)) \quad (4-1)$$

其中 $d(v)$ 表示 v 到源点的最短距离， $w(u, v)$ 表示边 (u, v) 的权重。

(2) A* 算法

A* 算法采用启发式搜索的方式来缩短搜索时间，并通过评估函数 $f(n)$ 计算最小代价以找到最佳路径。公式为

$$f(n) = g(n) + h(n) \quad (4-2)$$

其中 $g(n)$ 表示起点到 n 点的实际代价， $h(n)$ 表示启发式预估代价。搜寻时以 $f(n)$ 为评价标准，每次选择 $f(n)$ 最小的点进行扩展，而 $h(n)$ 的选择会导向搜索方向，这直接影响到是否能够找到最优解。如果 $h(n) = 0$ ，则 A* 算法就变成了 Dijkstra 算法，而 $h(n)$ 越小，那么算法效率越高，但不一定能找到最优解。如此选择 $h(n)$ 时需要在搜索速度和搜索精度之间进行权衡，选择适当的 $h(n)$ 可以达到更优秀的搜索效果。

4.4.2 局部路径规划

局部路径规划是指在环境的有限部分内为机器人生成路径的过程，通常是靠近其当前位置。它的目的是找到一条安全有效的轨迹，避开障碍物，同时满足各种约束条件，如最大速度和最小间隙。目前主流的局部路径规划算法有 DWA 和 TEB 两种。

(1) DWA

DWA 是 Dynamic Window Approach（动态窗口方法）的缩写，是一种局部路径规划算法。其原理是选取几组速度组合，在速度空间 (v, ω) 满足一定约束条件的前提下，对每组速度组合对应的机器人运动轨迹进行评估，通过评价函数选择最佳轨迹对应的速度方案作为此时机器人的运动速度。

首先，设机器人的速度最大值 v_{max} 和最低值 v_{min} ，采样空间应该满足式（4-3）：

$$V_1 = \{(v, \omega) | v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \quad (4-3)$$

然后，设机器人的最大加速度为 a ，最大减速度为 a' ，速度组合为 (v_0, ω_0) ，采样空间还应该满足式（4-4）：

$$V_2 = \{(v, \omega) | v \in [v_0 - a\Delta t, v_0 + a\Delta t], \omega \in [\omega_0 - a\Delta t, \omega_0 + a\Delta t]\} \quad (4-4)$$

最后，为了避免机器人发生碰撞，其刹车速度 V_3 需满足式（4-5）：

$$V_3 = \{(v, \omega) | v \leq \sqrt{2 \cdot \text{dist}(v, \omega) a}, \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) a}\} \quad (4-5)$$

其中， $\text{dist}(v, \omega)$ 是在速度空间 (v, ω) 下，模拟轨迹距离障碍物最近的距离。

综合式(4-3)、(4-4)和(4-5)，将机器人的速度限制在 $V_1 \cap V_2$ 的集合内， V_3 为 $V_1 \cap V_2$ 的附加条件。

通过模拟真实环境和机器人的参数，获取机器人的采样速度，进而模拟出各组采样速度对应的移动轨迹。接下来，使用轨迹评价函数评估计算生成的多条移动轨迹，以选出最优方案。

然而，DWA 也有一些局限性，例如：DWA 对初始条件很敏感，如机器人的起始位置和方向，这可能导致次优或不可行的路径；DWA 不能保证全局最优，这意味着它可能无法为一个给定的问题找到全局最佳的解决方案；DWA 只考虑狭义的环境，可能会忽略一些重要的特征，如障碍物或地标，这可能导致路径不可行。

(2)TEB

TEB（Timed Elastic Band）是一种用于机器人局部路径规划和优化的算法。它通过用弹性带状轨迹连接一系列的航点来生成路径，同时考虑到机器人的运动学和动力学约束。优化过程使穿越路径所需的总行程时间和精力最小化，同时避开障碍物并保持在一定的速度和加速度限制之内^[15]。TEB 可以处理非人体工程学的约束条件，以及纳入有关环境和机器人运动的先验信息，使其成为动态和不可预测的环境中实时运动规划的合适选择。TEB 定义机器人位姿为：

$$X_i = (x_i, y_i, \beta_i)^T \quad (4-6)$$

X_i 表示机器人在三维坐标系下的第 i 个位姿。

TEB 算法采用加权多目标优化方法来获得最优位姿，它的目标函数是：

$$f(B) = \sum_k y_k f_k(B) \quad (4-7)$$

$$B^* = \arg \min_B f(B) \quad (4-8)$$

其中， $f(B)$ 表示综合考虑各种约束的目标函数， $f_k(B)$ 表示任一目标函数， y_k 表示各项权重， B^* 表示最优路径。

TEB 算法常用的约束函数主要有是障碍物约束函数，该函数的具体表示形式为

$$e_r(x, x_r, \varepsilon, S, n) \approx \begin{cases} \left(\frac{(x - (x_r - \varepsilon))}{S}\right)^n & x > x_r - \varepsilon \\ 0 & \text{其它} \end{cases} \quad (4-9)$$

$$f_{path} = e_r(d_{min}, r_{path}, \varepsilon, S, n) \quad (4-10)$$

$$f_{ab} = e_r(-d_{min}, -r_{path}, \varepsilon, S, n) \quad (4-11)$$

其中， f_{path} 和 f_{ab} 均表示惩罚函数， x_r 表示边界， S 表示缩放比例， n 表示阶数，

d_{min} 表示路劲点和障碍物直接的最短距离， r_{path} 表示轨迹偏离路径点的距离。

TEB 规划算法流程如图 4.3 所示。首先执行初始化，设定机器人的起点、终点以及障碍物信息，并预先生成一条初始规划轨迹。然后通过预测机器人的速度和加速度，以及考虑到轨迹离散间隔，估计机器人的状态。接着利用图优化的思想，将机器人相邻的时间间隔和状态作为图优化节点，速度、加速度和机器人非完整约束作为图优化的边，同时考虑障碍物信息和相邻几个时间、空间序列的约束。按照三个目标的重要性，分配权值，将到达目标的速度、避开障碍的安全性和运动过程中的稳定性作为目标函数。最后使用 G2O 求解器对目标函数进行求解，得到最优的轨迹。



图 4.3 TEB 执行流程

4.4.3 Move_Base

Move_Base 是一个 ROS 功能包，提供了一种基于动作的导航系统实现。图 4.4 显示了 Move_Base 节点的高层视图以及它与其他组件的互动。它通过结合全局地图、Odometry 和传感器数据等多种信息，使机器人能够在复杂环境中规划和执行无

碰撞的路径。Move_Base 还可以处理动态障碍物、传感器的反馈以及对全局地图的实时更新，因此是一个适用于在变化的环境中进行自主导航的强大工具。

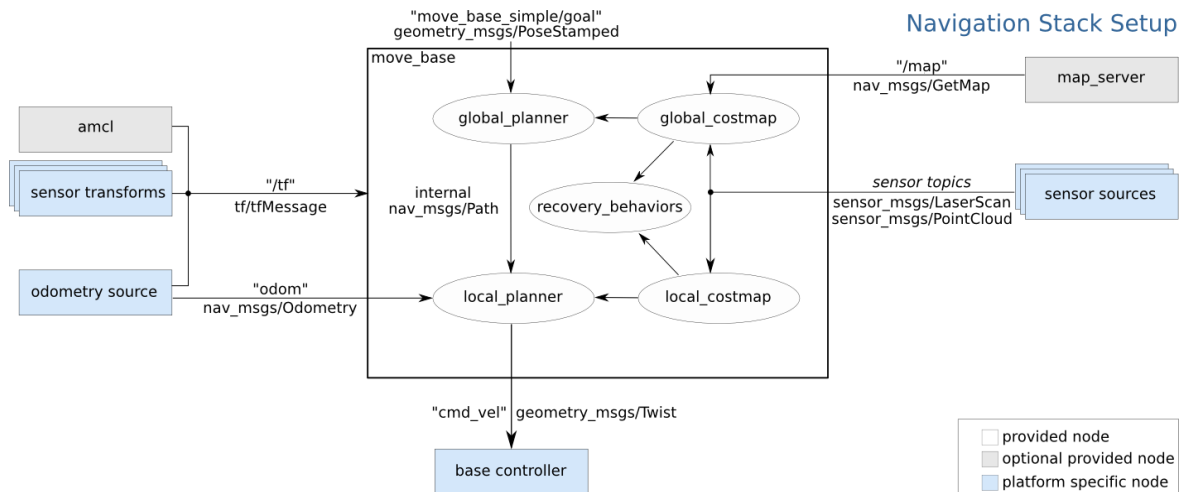


图 4.4 Move_Base 节点的通信

4.5 本章小结

本章的内容围绕着机器人的自主导航算法相关原理展开。重点介绍了路径规划，包括全局路径规划，局部路径规划以及 ROS 中一个关键的功能包 Move_Base。在全局路径规划研究中详细对比分析了 D*算法和A*算法的优劣，结合设计目标最终选用 A*算法进行导航。在全局路径规划研究中对 DWA 算法和 TEB 算法的原理以及工作流程做了分析结合设计目标最终选用 TEB 算法进行局部路径规划导航。

5 实验验证

前几章深入地介绍了基于 ROS 的麦克纳姆轮全向移动机器人路径规划和自主导航的理论方法和具体实现流程。实验通过在第二章所搭建的硬件平台上，首先构建室内环境地图，然后确定起始点和目标点，并通过 A* 和 TEB 算法规划出从起始点到目标点的搜索路径。自主导航设计框架如图 5.1 所示。

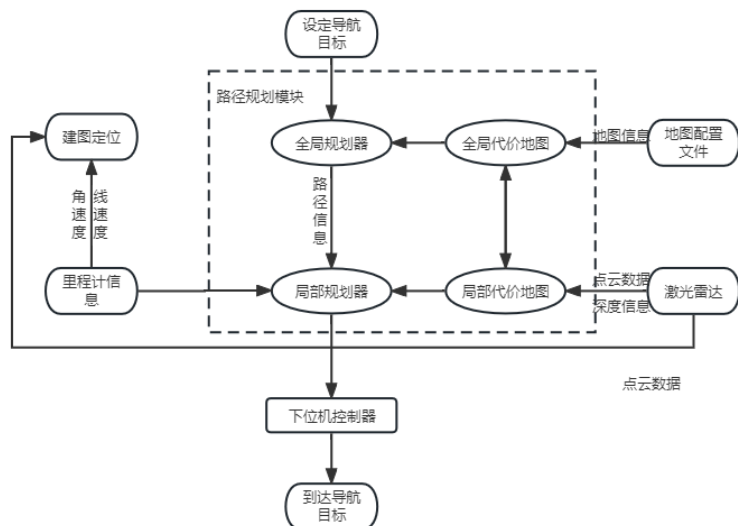


图 5.1 自主导航设计框架

5.1 实验环境介绍

5.1.1 实验平台构建

本章的实验是基于一个麦克纳姆轮的移动机器人平台，其中包含了激光雷达、内置传感器以及便携式 PC。机器人三视图如图 5.2 所示。所有实验都是在这个麦克纳姆轮移动底盘上进行设计，并在一个构建的室内环境中进行。移动机器人通过 WIFI 局域网连接到智能移动端，通过车载的 Jetson Nano 开发板进行控制。Jetson Nano 运行着 Ubuntu 18.04 系统，并使用 ROS 操作系统控制移动机器人。

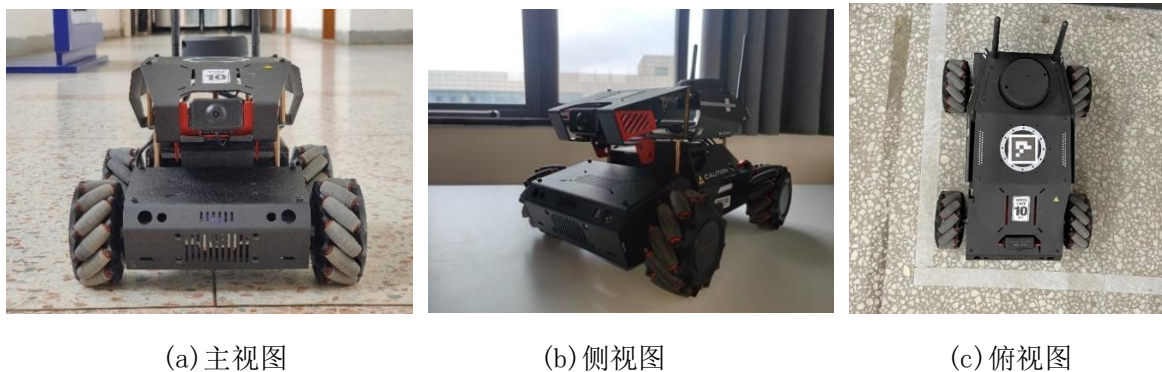


图 5.2 机器人实物图

5.1.2 实验环境

本设计选取的实验场景为室内，地点在笃行楼 501 教室。实验环境是由硬纸板模拟搭建的复杂地图，场景中包含随机障碍物，狭窄路段等等具有挑战性的部分。实验环境如图 5.3。

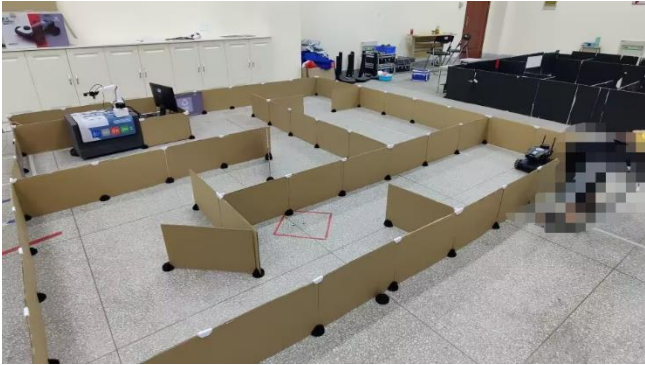


图 5.3 实际场地测试图

5.2 SLAM 实验

根据第三章对机器人即时定位和地图构建算法研究，结合 ROS 进行地图构建，然后保存构建的地图，为接下来的机器人自主导航实验提供环境模型。SLAM 实验流程如图 5.4 所示。

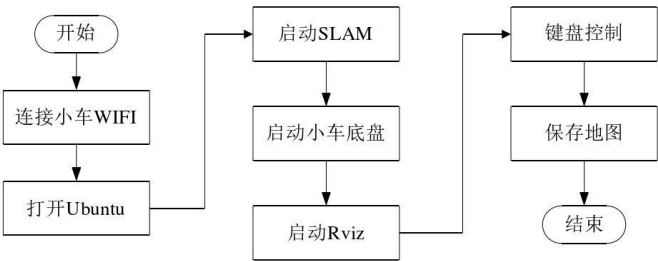


图 5.4 SLAM 实验流程

在建图前接通移动机器人电源系统开关，将上位机接入 WIFI，并通过远程软件 Todesk 在远程电脑中可视化车机电脑，实时观察移动机器人的建图效果，如图 5.5 所示为建图实际效果。

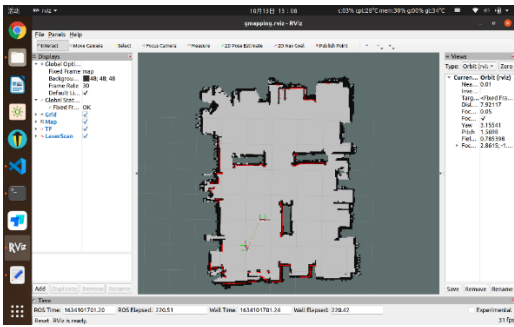


图 5.5 Gmapping 建图

通过启动 Launch 文件运行机器人驱动程序、激光雷达驱动程序和 Gmapping。同时启动 Rviz 可视化工具实时监控机器人的运动状态。机器人可以由键盘或手柄控制，指导其运动，构建二维网格地图。一旦地图构建完成，先前打开的窗口将被保留，地图将通过 Map_Server 节点保存。最后，关闭所有窗口，完成自主导航地图的构建。图 5.6 为实际构建的栅格静态地图。

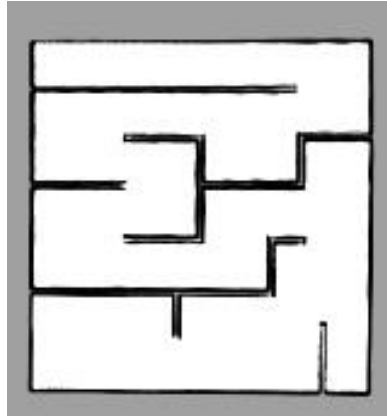


图 5.6 静态地图

5.3 自主导航实验

5.3.1 实验规划

上一节通过 Gmapping 算法完成了 SLAM，构建了自主导航所需的基本实验地图。图 5.7 为自主导航实验流程。

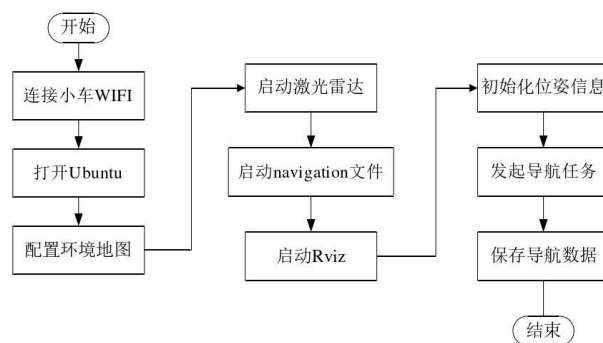


图 5.7 自主导航实验流程

首先，基于 PC 的 Ubuntu18.04 系统连接到移动机器人的 WIFI 进行远程登录，并访问移动机器人上的 NVIDIA Jetson nano。在启动导航节点之前，先对导航参数进行了配置，确认文件中的地图就是所需的导航地图后运行导航的 Launch 文件。打开 Rviz 可视化界面，配置好导航目标，然后关闭所有节点。手动将移动机器人放置在地图的起点处，或者通过 Rviz 界面中的二维姿势估计来设置起点和目标点。最后，

使用二维导航目标工具来设置目的地的位置和方向。

5.3.2 实验参设置与结果分析

本实验通过构建笃行楼 501 教室的场地地图，使用麦克纳轮移动机器人进行自主导航。使用 ROS 的 Rviz 可视化工具检查全局和局部代价地图、显示路径和障碍物，其中障碍物的膨胀半径为 0.22 米。移动机器人的速度最大限制为 1.5 米/秒，转弯角速度最大限制为 1.3 弧度/秒。图 5.8 所示为由 ROS 机器人通过 Move_Base 框架规划出来的路径，黑色代表全局路径，红色代表局部路径。

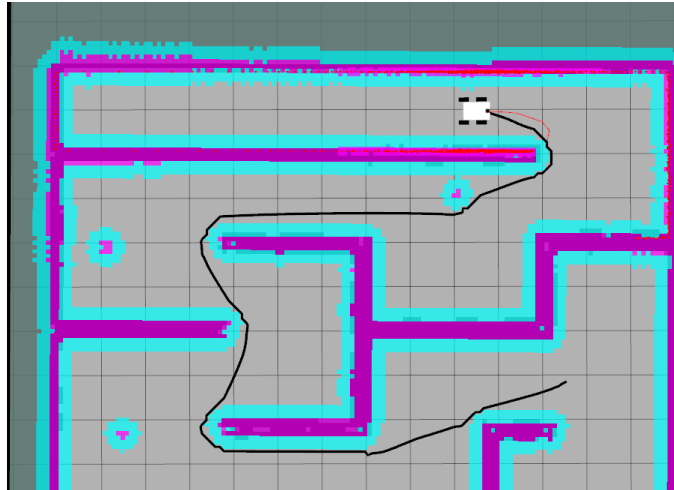


图 5.8 路径规划

本次导航实验结果表明移动机器人能够对障碍物快速作出反应，进行有效的路径规划，并最终成功到达目标点。同时验证了本系统基于 ROS 的麦克纳姆轮机器人平台的可行性。

5.4 本章小结

本章结合前面对 ROS 移动机器人的分析和导航算法研究，说明了移动机器人在自主导航实验中的硬件和软件支持，并建立了实验平台。其次，通过建立的实验平台验证了 Gmapping 算法，结果表明，该算法在室内环境中能有效地构建环境地图。最后，进行了室内自主导航避障实验，验证了该系统的有效性与可行性。

结束语

随着移动机器人的快速发展，自主导航技术在家庭服务、军事救援、物流运输等行业得到了广泛的应用，提高了生产力，节省了人力资源，促进了智能机器人的发展。然而，室内移动机器人在灵活性和适用性方面仍有局限性。因此，本文针对全向移动机器人自主导航和避障的缺陷，研究移动机器人的 SLAM 技术和路径规划。设计了一种基于麦克纳姆轮的移动机器人，依靠机器人携带的激光雷达感知环境信息，建立环境地图，结合 A* 和 TEB 算法的路径规划算法，实现了移动机器人的自主导航和动态避障。但在实际应用中仍存在一些不足，主要表现在以下几个方面。

(1) 本文仅采用单一激光雷达构建环境图，测绘精度较好，但对环境信息的采集仅限于平面。因此，今后的研究可以采用三维激光雷达或视觉与激光雷达相结合的技术。

(2) 全向移动机器人在室内环境中移动缓慢，使得它在高速移动过程中无法快速避开障碍物。因此，未来的研究应该集中在高速动态环境下全向移动机器人的避障问题上。

(3) 近年来，移动机器人的应用场景越来越复杂，单个移动机器人无法准确完成导航任务。因此，可以进一步研究多机协同导航和避障的问题。

致 谢

时光匆匆，光阴如幻，四年大学已经接近尾声。回首过去三年，我有过努力，奋进，也经历过迷茫，失落，但更多的是衷心的感谢。

大一的一次优秀学长经验介绍会上，我听到张镇墅学长向我们介绍院机器人实验室，听完他的讲述后我就暗下决心一定要认真学习编程，因为我从高中时就发现自己对单片机，机器人方面感兴趣，我想主动去学习。大二上学期我加入了院机器人实验室，在这里我遇到了我的恩师周军老师。在实验室期间，周老师给予我很多专业知识方面的帮助，也给予我很多竞赛的机会。每一次学习，每一次竞赛都是一次锻炼的机会，我的实践能力，视野，胆识都得到了提高。离开实验室后，周老师还鼓励我积极备战考研，为我排忧解难。在此由衷地感谢周军老师。此外，我还要感谢电子电气工程学院的众多老师，刘小军老师，刘增元老师，牛绿原老师，孙正凤老师等等，这些老师在各自的领域都有深入的研究，他（她）们教给我的知识在我的学习生活中都有很大的帮助，非常感谢各位老师。

感谢我的同窗王龙飞，陈梓标，宋涛，陈熙文等的关心和帮助，与他们在实验室朝夕相处的日子是我本科生涯一道靓丽的风景线。

特别感谢我的女朋友张明惠，在考研备战途中，她一直给予我学习上和精神上的帮助，最后一起成功上岸；还有我亲爱的家人们，在背后默默支持我，关系我，鼓励我，做我坚强的后盾。

最后，衷心感谢百忙之中抽出时间阅读此论文的各位老师！

参 考 文 献

- [1] 李振富,石学文,徐雍倡,等. ROS 机器人的移动导航设计[J]. 自动仪表, 2021.
- [2] USLU, ERKAN, CAKMAK, FURKAN, ALTUNTAS, NIHAL, et al. An architecture for multi-robot localization and mapping in the Gazebo/Robot Operating System simulation environment[J]. Simulation: Journal of the Society for Computer Simulation, 2017.
- [3] 陈博翁, 范传康, 贺骥. 基于麦克纳姆轮的全方位移动平台关键技术研究[J]. 东方电气评论, 2013.
- [4] 郑仁辉. 麦克纳姆轮全向机器人移动平台的设计[D]. 黑龙江:哈尔滨工程大学, 2017.
- [5] 张禹, 朱光召. 基于麦克纳姆轮的全方位移动平台技术研究[J]. 现代制造技术与装备, 2017.
- [6] 李恒, 杨亮, 曾碧, 等. 基于 ROS 的移动机器人仿真实验平台设计与实现[J]. 电子设计工程, 2022.
- [7] KAMARUDIN, KAMARULZAMAN, MAMDUH, SYED MUHAMMAD, SHAKAFF, ALI YEON MD, et al. Performance Analysis of the Microsoft Kinect Sensor for 2D Simultaneous Localization and Mapping (SLAM) Techniques[J]. Sensors, 2014.
- [8] 温熙, 郭杭. 基于 Kinect 和惯导的组合室内定位[J]. 计算机工程与设计, 2016.
- [9] 危双丰, 庞帆, 刘振彬, 等. 基于激光雷达的同时定位与地图构建方法综述[J]. 计算机应用研究, 2020.
- [10] 刘畅. 基于扩展卡尔曼滤波的同步定位与地图构建 (SLAM) 算法研究进展[J]. 装备制造技术, 2017.
- [11] 杨涵, 万游, 蔡洁萱, 等. 基于步态分类辅助的虚拟 IMU 的行人导航方法[J]. 计算机科学, 2022.
- [12] VALTER COSTA, ROSALDO ROSSETTI, ARMANDO SOUSA. Simulator for Teaching Robotics, ROS and Autonomous Driving in a Competitive Mindset[J]. International journal of technology and human interaction, 2017.
- [13] Research from University of Salerno Yields New Data on Robotics (Robot-Agnostic Interaction Controllers Based on ROS) [J]. 2022.
- [14] 漆保凌. 室内环境下的移动机器人视觉 SLAM 导航研究及系统实现[D]. 重庆:重庆邮电大学, 2019.
- [15] 机甲大师 ROBOMASTER S1[J]. 工业设计, 2019.

附录 A 源程序（摘要）

A.1 机器人基本启动文件

A.1.1 底盘控制部分

```
<launch>

  <node pkg="ucar_control" type="robot_base" name="robot_base" output="screen">
    <rosparam file="$(find ucar_control)/config/base_driver.yaml" command=" load"
  />

</node>

</launch>
```

A.1.2 激光雷达

```
<launch>

  <!-- Run Rplidar -->

  <node name="rplidarNode" pkg="rplidar_ros" type="rplidarNode" output="screen">
    <param name="serial_port"          type="string" value="/dev/rplidar"/>
    <param name="serial_baudrate"      type="int"    value="115200"/>
    <param name="frame_id"             type="string" value="laser"/>
    <param name="inverted"              type="bool"   value="false"/>
    <param name="angle_compensate"      type="bool"   value="true"/>
    <param name="scan_mode"            type="string" value="Sensitivity"/>
    <remap from="scan" to="scan_raw"/>
  </node>

  <node pkg="base_driver" type="lidar" name="lidar"/>

</launch>
```

A.2 机器人导航启动文件

<launch>


```
<node name="map_server" pkg="map_server" type="map_server" args="$(find
ucar_nav)/maps/map.yaml"/>

<node pkg="move_base" type="move_base" respawn="false" name="move_base"
output="screen">

  <param name="base_local_planner" value="dwa_local_planner / DWA Planner
ROS " />

  <rosparam file="$(find ucar_nav)/config/costmap_common_params.yaml"
command="load" ns="global_costmap" />

  <rosparam file="$(find ucar_nav)/config/costmap_common_params.yaml"
command="load" ns="local_costmap" />

  <rosparam file="$(find ucar_nav)/config/local_costmap_params.yaml"
command="load" />

  <rosparam file="$(find ucar_nav)/config/global_costmap_params.yaml"
command="load" />

  <rosparam file="$(find ucar_nav)/config/move_base_params.yaml" command
="load" />

  <rosparam file="$(find ucar_nav)/config/dwa_local_planner_params.yaml"
command="load" />

  <node pkg="rviz" type="rviz" name="rviz" args="-d $(find ucar_nav)
/rviz/nav.rviz"/>

</launch>
```

A.3 机器人源程序

A.3.1 主程序

```
#include"string.h"
#include"stdlib.h"
#include"ros/ros.h"
#include <iostream>
#include<std_msgs/Bool.h>
#include<std_msgs/Int8.h>
#include<std_msgs/String.h>
#include<dynamic_reconfigure/client.h>
```

```
#include <move_base_msgs/MoveBaseAction.h>
```

```
#include <actionlib/client/simple_action_client.h>
#include "move_base_msgs/MoveBaseActionFeedback.h"
#include <iostream>
ros::Publisher reconfigure;
std::string reconfigure_topic= "dynamic";
bool get_command,go,if_recognize=1,get_food;
int i,count,voice_play,get_info=99,lock,playone;
int point_goal,food_goal=3,final1;
int flag0,flag1,flag2, flag3, flag4, flag5, flag6, flag7,flag8,flag9,flag10,flag11,flag12,flag13;

void pub_goals(double goal_num[]){
    actionlib::SimpleActionClient<move_base_msgs::MoveBaseAction>
ac("move_base",true);
    ROS_INFO("Waiting for move_base action server...");
    //等待 60 秒以使操作服务器可用
    if(!ac.waitForServer(ros::Duration(30))){
        ROS_INFO("Can't connected to move base server");
        ros::shutdown();}
    ROS_INFO("Connected to move base server");
    ROS_INFO("Starting navigation test");
    move_base_msgs::MoveBaseGoal goal;
    //使用地图框定义目标姿势
    goal.target_pose.header.frame_id = "map";
    //将时间戳设置为“now”
    goal.target_pose.header.stamp = ros::Time::now();
    //将目标姿势设置为第 i 个航点
    goal.target_pose.pose.position.x=goal_num[0];
    goal.target_pose.pose.position.y=goal_num[1];
    goal.target_pose.pose.position.z=0;
    goal.target_pose.pose.orientation.x=0;
    goal.target_pose.pose.orientation.y=0;
    goal.target_pose.pose.orientation.z=goal_num[2];
    goal.target_pose.pose.orientation.w=goal_num[3];
    //让机器人向目标移动
    //将目标姿势发送到 MoveBaseAction 服务器
    ac.sendGoal(goal);
}

void Callback(const move_base_msgs::MoveBaseActionFeedback & msg){
    if(flag1==0){
```

```
doubledistance= sqrt(pow(goal_1[0]
msg.feedback.base_position.pose.position.x,2)+pow(goal_1[1]-
msg.feedback.base_position.pose.position.y,2));
    if(distance<0.6){
        flag1= 1;
        std::cout<<"Goal_1 reached!!!"<<std::endl;
        pub_goals(goal_2);}
int  main(int argc, char* argv[])
{
    ros::init(argc,argv,"go_goals");
    ros::NodeHandle nh;
    /*发布启动动态参数服务器话题*/
    reconfigure = nh.advertise<std_msgs::Int8>(reconfigure_topic, 10);
    /*订阅 move_base 反馈*/
    ros::Subscriber goal_result = nh.subscribe("/move_base/feedback", 10,Callback);
    /*订阅启动指令*/
    ros::Subscriber voice_command = nh.subscribe("/start",10,voice_command_callback);
    /*订阅扫码话题*/
    ros::Subscriber scan_aruco = nh.subscribe("/scan_aruco",10,scan_aruco_callback);
    /*service 请求语言播报
    ros::ServiceClient client = nh.
serviceClient<xf_mic_tts_offline::Play_TTS_srv>("xf_mic_tts_offline_node/play_txt_wav
");
    xf_mic_tts_offline::Play_TTS_srv srv;*/
    system("sh ./test.sh");
while(ros::ok()){
    ros::spinOnce();}
return 0;}
```

附录 B 在学期间取得的成果

专利

[1] 刘小军, 周军, 温宏愿, 牛绿原, 张昕航, 张富高. 一种机器人用抓取机械臂[P], 中国专利: CN202122259698.7 2022-03-04.

奖项

- (1) 2021 年 11 月, 获江苏省机器人大赛冠军（一等奖）一项, 亚军（二等奖）一项
- (2) 2021 年 12 月, 获中国机器人技能大赛冠军（一等奖）一项, 亚军（一等奖）一项
- (3) 2022 年 4 月, 获中国机器人大赛冠军（一等奖）
- (4) 2019 年 6 月, 获恩智浦全国大学生智能汽车竞赛三等奖一项
- (5) 2021 年 4 月, 获第十六届全国大学生智能汽车竞赛三等奖一项
- (6) 2022 年 4 月, 获第十七届全国大学生智能汽车竞赛三等奖一项