

分类号_____

学校代码 10487

学号 M201672165

密级_____

华中科技大学

硕士学位论文

适用于扫地机器人多传感器融合 SLAM 算法的研究与设计

学位申请人：代强

学 科 专 业：集成电路工程

指 导 教 师：雷鑑铭 副教授

答 辩 日 期：2018.5.21

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering**

**Research and design of SLAM algorithm for sweeping
robot with multi-sensor**

Candidate : Dai Qiang

Major : Integrated Circuit Engineering

Supervisor : Associate Prof. Jianming Lei

Huazhong University of Science & Technology

Wuhan 430074, P.R.China

May, 2018

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文版权使用授权书

本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，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技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 _____ 年解密后适用本授权书。
本论文属于 不保密 ☐。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

摘要

扫地机器人的自主移动,需要借助传感器对周围环境和空间进行感知进行定位和规划路径,这个过程叫即时定位和地图构建(SLAM, Simultaneous localization and mapping)。在早期的扫地机器人中常利用里程计和惯性定位与超声波传感器配合进行定位和地图构建,这种方案效果较差,无法满足需求,而常用的基于里程计的SLAM定位方法精度不高,运算量太大而限制了其应用。本文提出了一种基于多传感器的SLAM算法,通过提高预测模型的精度来提高SLAM算法的定位精度,以满足本系统的应用需求。

本文首先从SLAM算法的理论入手,分析其运动模型和观测模型,得出多传感器融合SLAM算法的理论基础是通过滤波算法对运动模型和观测模型进行融合,进而比较分析并系统仿真了两种常用SLAM算法:EKFSLAM算法和粒子滤波SLAM算法,确定本系统选择具有更高定位精度的粒子滤波算法。为了提高预测模型的精度,有效降低预测位姿的累积误差,本文提出了一种基于里程计、陀螺仪、加速度计和电子罗盘的融合位姿预测方案。然后,采用粒子滤波算法对预测位姿和观测数据进行融合,得到更高精度的位姿估计。最后,采用占据栅格地图对激光雷达观测的数据进行地图构建,能以较小资源实现地图构建,降低计算量,提高路径规划效率。

最后为了验证算法的有效性,本文基于ROS仿真平台对基于里程计SLAM和融合SLAM算法进行仿真对比,结果显示基于里程计SLAM算法构建的地图出现混叠,而融合SLAM算法构建的地图具有较高精度。在此基础上,对扫地机器人软硬件平台进行搭建,并在室内对扫地机器人进行融合SLAM算法实验,得到点云图的最大相对误差为5.7cm,相对误差百分比低于8.86%,点云图与室内布局一致性良好。通过对扫地机器人进行几次定位测量实验,得到误差的平均值为5.8cm,最大值为9cm。

关键词: SLAM 室内定位 扫地机器人

Abstract

As for simultaneous localization and mapping (SLAM), it's necessary for sweeping robots to locate and plan the path for autonomous movement by sensing the surrounding environment and space by the sensor. In the last decades, the location and map construction of previous cleaning robot are widely applied in combination with the odometer and the inertial positioning and ultrasonic sensors. It's a less effective scheme which can not meet the needs, but the method of SLAM positioning based on odometer is often accompanied by low precision and large amount of computation, which limits its application. This paper presents a multisensor-based SLAM algorithm, which improves the accuracy of the SLAM algorithm by improving the accuracy of the prediction model and meets the application requirements of the system.

This thesis begins with the theory of SLAM algorithm, analyzes its motion model and observation model, and concludes that the theoretical foundation of the multi-sensor fusion SLAM algorithm is to fuse the motion model and the observation model through a filtering algorithm, and then to compare and analyze and systematically simulate the two commonly used methods. SLAM algorithm: EKFSLAM algorithm and particle filter SLAM algorithm, determine that this system chooses particle filter algorithm with higher positioning accuracy. In order to improve the accuracy of the prediction model and effectively reduce the cumulative error of the predicted pose, this paper presents a fusion pose prediction scheme based on odometer, gyroscope, accelerometer and electronic compass. Then, the particle filter algorithm is used to fuse the predicted pose and observation data to get higher accuracy pose estimation. Finally, maps constructed using lidar maps for the data observed by Lidar can be used to build maps with smaller resources, reduce the amount of calculations, and improve the efficiency of route planning.

Finally, in order to verify the effectiveness of the algorithm, the ROS-based simulation platform was used to compare the SLAM algorithm based on the odometer and the fusion

SLAM algorithm. The results showed that the map constructed based on the odometer SLAM algorithm appeared aliased, while the map constructed by the fusion SLAM algorithm had a higher map. Accuracy. On this basis, the software and hardware platform of the sweeping robot was built, and the SLAM algorithm experiment was performed on the sweeping robot in the room. The maximum relative error of the point cloud diagram was 5.7cm, the relative error percentage was less than 8.86%, and the point cloud diagram and indoor layout Consistency is good. Through several positioning measurement experiments on the sweeping robot, the average error is 5.8cm and the maximum value is 9cm.

Key Words: SLAM Indoor positioning Sweeping robot

目 录

摘 要.....	I
Abstract.....	II
1 绪 论	
1.1 课题研究的背景和意义	(1)
1.2 国内外发展现状	(2)
1.3 论文的研究内容和组织结构	(6)
2 SLAM 算法基本原理和算法分析	
2.1 引言.....	(7)
2.2 EKF SLAM 算法.....	(11)
2.3 粒子滤波 SLAM 原理	(15)
2.4 MATLAB 仿真测试.....	(16)
2.5 本章小结.....	(18)
3 多传感器融合 SLAM 算法系统设计	
3.1 多传感器融合 SLAM 系统流程	(19)
3.2 传感器独立模型及误差分析	(20)
3.3 里程计和 IMU 融合位姿预测.....	(22)
3.4 基于融合位姿预测的激光雷达数据处理流程	(24)
3.5 本章小结.....	(30)

华中科技大学硕士学位论文

4	扫地机器人平台搭建及仿真测试	
4.1	扫地机器人实验平台介绍	(32)
4.2	软件开发平台	(34)
4.3	系统仿真及测试	(36)
4.3	本章小结.....	(40)
5	总结与展望	
5.1	总结.....	(42)
5.2	展望未来.....	(43)
	致 谢.....	(44)
	参考文献.....	(45)

1 绪论

1.1 课题研究的背景和意义

随着科技的高速发展，扫地机器人在家居、工业、商场等方面的应用越来越多。其高度集成化和智能化，受到了大家广泛欢迎。与此同时，其也存在诸多问题，严重影响了使用者的体验。其中最主要的问题是如何实现较高精度室内导航方案。对导航的定义可以是引导人、机器人或运输工具沿一定路线从一个地点运动到另一个地点的方法，对导航比较重要的三个元素为：定位、地图和路径规划^{[1][2]}。在扫地机器人室内导航中，有许多比较成熟的路径规划方案可以采用，但定位与地图的构建中还存在很多问题需要解决^[3]。

在不少扫地机器人中，能够看到其有很明确的路径规划功能，比如沿边模式、弓字形路线、螺旋形重点清扫路线，但因为缺乏地图和有效的定位，在室内复杂环境下工作时，容易碰到各种问题：到处乱撞；某些地方重复的清扫而有些角落却覆盖不了；有时会卡在一个角落里出不来；有时也会把自己缠绕的动不了，需要人为协助^{[4][5]}。

针对此问题提出了采用同步定位与地图构建又叫并发建图与定位的方法，SLAM最早在1988年由Smith、Self和Cheeseman提出。在SLAM中，将机器人放在一个未知的位子，机器人根据视觉或距离传感器感知周围环境，进行定位和地图构建^{[6][7]}。

SLAM有基于里程计和超声波测距，陀螺仪和超声波测距，视觉传感器和里程计，激光雷达和里程计等多种方式的^[8]。前两个在低端产品中有应用，属于比较早期的算法从严格意义上来讲并不算真正的SLAM，其虽然计算量小，算法简单，但累积误差很大。视觉SLAM中，虽然图像中含有丰富的纹理信息，但特征检测难度大，计算量大，难以在嵌入式设备中实现^[9]。在经常使用的激光雷达和里程计模型中，因为里程计的噪声和累积误差比较大，为了提高精度常常需要提高雷达扫描的频率，对硬件性能也提出了较高要求^[10]。

1.2 国内外发展现状

扫地机器人的出现不到 25 年。1996 年，第一台走向量产的扫地机器人原型才面世，叫做“三叶虫”(Trilobite)，是由瑞典家电巨头伊莱克斯(Electrolux)制造，直到 2001 年，Trilobite 才开始走向市场，并成为史上第一款量产的扫地机器人^[11]。Trilobite 通过超声波传感器躲避障碍物，同时帮助它进行地图构建。由于超声波传感器盲区比较多，第一代并不能很好的避开障碍物^[12]。当在第二代版本的 Trilobite 中加入了红外传感器后，才使它的避障能力有比较大的提升。为保证安全和只在特定房间工作，可以通过在房间门口和楼梯口放置磁条作为虚拟墙壁来阻止机器人的进入。Trilobite 的电池支持其工作 60 分钟，三种工作模式可以选择：正常、快速和点清理，吸尘器在充满垃圾时会发出提示警告，其较低的高度可以对桌子和床下进行清扫，其完全自动化的清扫，使用非常方便^[12]。“三叶虫”虽然具有较好的避障能力，但其定位精度差，构建的地图精度差，并不能对室内进行有效的全局规划，致使其效率较低。

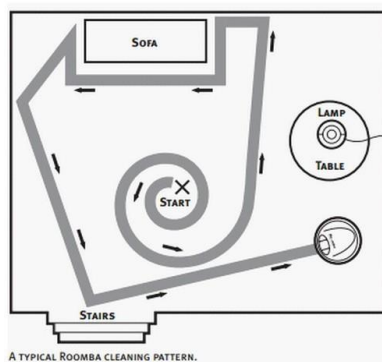


图 1-1 Roomba 工作路线图

2002 年 iRobot 公司推出的 Roomba 是扫地机器人发展史上的里程碑式事件。Roomba 适用于木质地板、防水地毯、瓷砖和中短毛地毯，其有三种工作模式，分别对应较小面积、中等面积、较大面积三种区域的清扫。如图 1-1，它在开启后会先沿着螺旋线向外运动，直到碰到墙壁或其它障碍物时，它会利用距离传感器贴着墙壁运动，然后转弯 90 度对房间的另一端进行清洁，通过交叉和重复螺旋运行的路线清洁。由于较强的路径规划能力和便宜的价格，Roomba 相对算得上是一款比较成功的产品^{[13][14]}。Roomba 更多的是借助其沿边传感器和红外测距传感器进行路径规划，没有全

局规划，清扫重复率较高。

Neato 扫地机器人于 2010 年推出，其通过 360 度旋转的激光测距仪来对周围的环境进行扫描，然后进行即时定位与环境地图构建（SLAM），在此基础上进行合理规划清扫路线。Neato 具有断点恢复清扫功能，可以返回上次的断点继续进行清扫，它还具有自动寻找充电桩充电的功能。Neato 的缺点是其所采用的方案成本较高导致其售价偏高，所以市场渗透率不高^{[15][16]}。



图 1-2 BL001A 机器人

国内扫地机器人行业起步相对较晚，目前仍处于快速发展阶段，有不少产品陆续推出，其中不乏优秀的产品。2004 年，重庆大学能科学技术研究室与宁波波朗电器公司合作开发扫地机器人^[17]，如图 1-2 其开发的 BL001A 扫地机器人在前机身前半周侧面安装有红外传感器，可以用来探测障碍物和贴着障碍物边沿行走，并根据这些传感器的信息计算房间大小及障碍物分布信息，并利用随机控制规划遍历策略进行更加高效的清扫路径规划，自动清扫房间地面的每一个角落，其运行轨迹为遇到障碍物后退 10-20cm 左转 45° 继续前行，它的粉尘净化率达到 98%，而清扫面积率高达 99.9%。它具备高效、灵活、安全、节能等特点，清扫结束后可自动寻找充电桩进行充电。其缺点是对房间的信息只能大体感知，不具有定位和地图构建功能，行走清扫路线比较随机，重复率高，效率低。

科沃斯的 DT85G，是一款广受好评的产品。其前半周采用 10 组红外避障，可以达到前半周无死角检测障碍物，同时利用编码器和惯性定位单元，其可以对室内进行地图构建。DT85G 具有螺旋线、弓字形、沿边模式等多种清扫模式，其自动回充采

用的是由充电桩发射红外线,扫地机器人接收的方式对准,随意移动充电桩人可以自动返回充电。DT85G 引入了定位和地图构建功能,这使其在局部表现比较智能,但其定位采用的是里程计和惯性定位,定位累积误差大,无法得到可靠的地图进行路径规划,整体扫地效率不高。

传统定位一般基于里程计或惯性传感器,存在不可避免的累积误差。定位与地图构建问题一直是移动机器人研究的重点之一,传统定位在已知地图或位置的情况有比较好的解决办法,但在面临地图和位置同时未知的情况下就显得束手无策了。针对此问题提出了同步定位与地图构建又叫并发建图与定位,SLAM 其最早由 Smith、Self 和 Cheeseman 在 1988 年提出,其重要的研究和使用价值受到大家的广泛关注,并被认为是实现自主机器人的关键所在^[18]。

而 SLAM 的关键问题在于位姿估算与地图表示。位姿估算算法大致分为以下几种:拓展卡尔曼滤波(EKF)、粒子滤波器(RBPF)、GraphSLAM、平滑技术、稀疏的拓展信息滤波器、最大似然法等,其中最常使用的为拓展卡尔曼滤波和粒子滤波器^[19]。

在 1999 年的国际机器人学研讨会中,Thrun 等人首次提出基于 EKF(拓展卡尔曼率)的 EKFSLAM 算法,并证明了该算法在概率学上具有收敛性^[20]。随后基于 EKF 的 SLAM 得到迅速发展,基于不同理论的 SLAM 算法也不断被提出,SLAM 算法的理论与应用得到迅速发展。拓展卡尔曼滤波是基于卡尔曼滤波而来的,卡尔曼滤波对于线性高斯模型系统可以通过迭代得到最小方差估计的系统状态,其对系统的线性要求制约在 SLAM 的应用范围。而拓展卡尔曼滤波实际上是将非线性系统通过泰勒级数展开的方式转换成线性滤波问题,从而将标准卡尔曼滤波应用到非线性系统中^[21]。在 EKFSLAM 中,通常使用一个包含机器人位姿信息和环境特征的高维向量来表示,其系统可以理解成一个马尔科夫过程,机器人初始状态 t_0 ,系统首先通过 t_0 到 $t-1$ 的观测向量和得到的位姿估计来估算 t 时刻的预测位姿,然后通过测模型得到 t 时刻观测位姿,结合预测位姿可以得到预测观测位姿。此时若从传感器获得 t 时刻真实值,此时可以通过协方差得到他们的权值,通过权值计算最终位姿估计,然后更新地图。EKFSLAM 首先应用到 SLAM 中,并在很长一段时间时间是解决 SLAM 问题

的主要手段，但也存在很多不足。EKFSLAM 在每次计算中都需要计算所有特征的协方差矩阵，而在复杂环境下就会导致协方差矩阵维数较高，计算量较大，因此其灵活性较差。

粒子滤波（Particle Filter）的定义为，通过寻找一组在状态空间中传播的随机样本来近似的表示概率密度函数，然后通过样本的均值来代替积分运算，从而获得系统状态的最小方差估计的过程，这些样本被形象的称为“粒子”，故而叫粒子滤波^[22]。粒子滤波基于序贯重要性采样(SIS)思想，相较于 EKFSLAM 基于高斯模型和卡尔曼滤波的泰勒展开，粒子滤波是采用粒子集来表示概率分布的，能够用在任何形式的空间状态模型上，能更好的适应非高斯、非线性系统。值得一提的是，Murphy 等人通过对粒子滤波算法计算的复杂性分析，提出基于分解理念的 Rao-Blackwellized 粒子滤波器（RBPF），极大地简化了计算量^[23]。粒子滤波器的缺点是，为了更加接近系统的后验概率密度，需要足够多的样本，不同环境下所需样本数量也不一致，如何自适应不同环境有效减少样本数量是研究的问题所在。

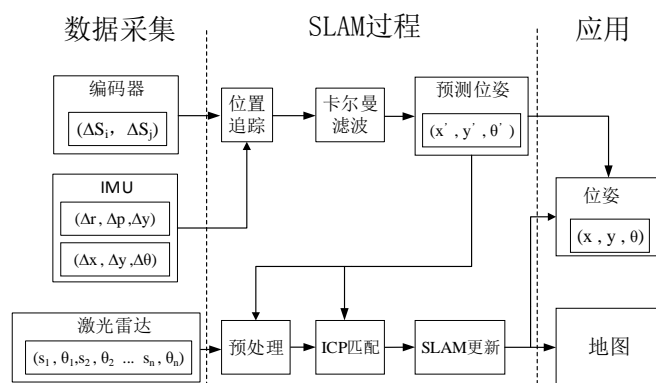


图 1-3 多传感器融合 SLAM 流程图

基于此本文提出一种基于多种传感器融合的 SLAM 算法，其整体流程如图 1-3 所示，并在扫地机器人上的实现，具有即时定位和地图构建能力，其通过激光雷达扫描周围环境，融合里程计、陀螺仪、加速度计等传感器的数据进行定位和地图构建，在室内环境下可以不断的完善地图，来优化路径，其优势在于融合多传感器，提高了精度，降低了对激光雷达扫描频率和采样率的要求，降低了对硬件计算性能的要求。

1.3 论文的研究内容和组织结构

本文研究了一种适用于扫地机器人的 SLAM 算法，传统扫地机器人存在导航精度差和 SLAM 应用成本高的问题，本文采用里程计与 IMU 联合优化预测模型，采用激光雷达进行位姿估计，通过尝试降低 SLAM 更新频率的方法降低系统计算量从而达到降低对硬件的要求。

第一章，绪论。首先是本系统研究的背景，本系统研究的背景是基于导师项目，开发一种成本较低的具备较高定位功能的扫地机器人，以解决现在扫地机中存在的导航问题。然后总结国内外扫地机器人的发展现状，针对其中采用的一些具有代表性的定位方案进行了介绍，指出其中存在的一些问题；随后对 SLAM 算法的发展现状进行了介绍。

第二章 SLAM 算法模型和两种常用的算法分析及仿真。首先分析 SLAM 算法的模型，针对模型介绍了其中经典的两种算法，分析其优点和缺点，然后对算法进行仿真对比，提出系统初步设计的方法。

第三章基于融合 SLAM 算法的扫地机器人的系统设计。第三章首先分析了本系统的 SLAM 框架，从整体对定位过程进行分析；然后对里程计、IMU、激光雷达的三种模型进行分析，通过对模型进行分析，指出其中数据的特点和误差来源；然后对基于里程计和 IMU 模型的位姿预测进行介绍；随后针对激光雷达数据一般处理流程进行设计，从数据滤波到 ICP 匹配到粒子滤波过程最后建立栅格地图。

第四章机器人平台设计介绍和系统的测试。首先介绍机器人的硬件系统，对硬件进行展示和介绍，对其中部分相关电路进行介绍；然后对系统软件开发进行介绍，STM32 基于 keil 平台开发，机器人采用 ROS 进行控制；最后对系统进行仿真测试。

第五章总结与展望。对本文的工作进行总结，分析本系统存在的不足，还有许多地方需要研究与测试，对未来的工作方向进行展望，以期做出更出色的系统。

2 SLAM 算法基本原理和算法分析

2.1 引言

2.1.1 自主导航一般性问题

在机器人自主导航时，有三个核心问题需要解决：（1）我在哪？（2）这是哪？（3）怎么走？在常用的导航系统中，都是基于已知地图进行定位，进而进行路线的规划和选择。但当进入一个完全未知的环境中时，这些问题就变得十分复杂。第一个问题可以认为是对自身定位的过程，第二个问题可以认为是通过观测周围环境并建立地图的过程，在未知环境中，第一个问题和第二个问题一般是同时进行相互协助的：基于多种传感器数据进行自身定位，为地图的构建提供位置信息；再基于观测通过算法生产周围环境地图，辅助机器人进行定位。第三个问题可以理解为路径规划，在解决了定位和地图构建问题后，我们的最终目的是要实现“怎么走？”，结合自身位置通过对地图进行规划，找到合适的路线到达目的地。

本文所要研究问题为第一个问题和第二个问题，可以引申为即时定位与地图构建，具体指：机器人在未知环境中，通过传感器对周围环境进行观测，来完成定位和地图构建。

2.1.2 SLAM 概念及建模

SLAM 是指同步定位与地图构建又叫并发建图与定位的方法，SLAM 最早在 1988 年由 Smith、Self 和 Cheeseman 提出，并将同步定位与地图构建引申为概率学问题，通过拓展卡尔曼滤波（EKF, Extended Kalman Filter）来消除运动噪声和观测噪声，基于 EKF 的 SLAM 算法在早期的同步定位与地图构建研究中具有十分重要的地位。随后又出现了粒子滤波等算法，对移动机器人自主导航的需求，使得更多人投入到 SLAM 的研究中。以下为 SLAM 模型建立和分析。

在建立 SLAM 运动模型前，需要先对机器人定位和建图的整个过程进行分析。如图 2-1 所示，用 x_k 表示 k 时刻机器人， w_k 表示 k 时刻系统噪声， u_k 表示 k 时刻系

统对机器人的运动控制量，用 m_k 表示在 k 时刻，通过观测信息 z_k 记录的环境路标信息。通过动态贝叶斯网络的 SLAM 模型，可以分析出，机器人在 k 时刻的位置与姿态只与 $k-1$ 时刻的位置与姿态 x_{k-1} 和控制量 u_{k-1} 有关，在 k 时刻所观测到的环境路标也只和本时刻位置与姿态 x_k 有关，即 SLAM 运动过程与马尔科夫过程相似^[24]。

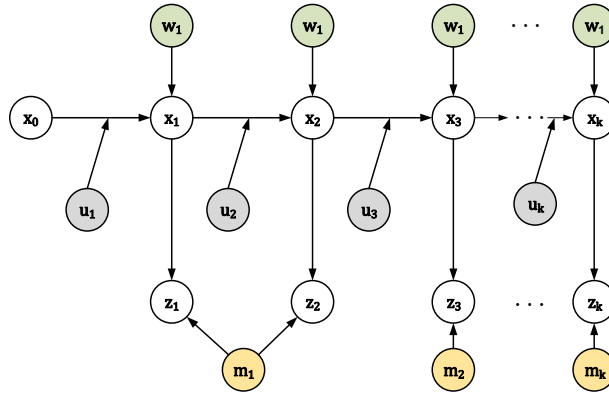


图 2-1 基于动态贝叶斯网络的 SLAM 模型

若用 x_0 表示机器人初始时刻的位置， M 表示 m_1 到 m_n 的集合， W_k 表示 w_1 到 w_k 的集合， Z_k 表示 z_1 到 z_k 的集合， U_k 表示 u_1 到 u_k 的集合，则可以将机器人概率密度表示为：

$$P(x_k, M | U_k, Z_k, W_k, x_0) \quad (2-1)$$

上述介绍的是整个 SLAM 过程，可以将其分别建立运动模型和观测模型。

1) 运动模型

所谓运动模型指的是机器人在运动过程中，其位置和姿态随时间变化的关系。机器人的运动状态及位置和姿态，是受系统控制的，在任何时刻只与前一时刻的位姿 x_{k-1} 和前一时刻系统的控制量 u_k 有关，因此，其运动模型概率分布可以用机器人本时刻的位姿、上一时刻的位姿、系统的控制信号及控制噪声表示^[25]：

$$P(x_k | u_k, x_{k-1}, w_k) \quad (2-2)$$

其中， x_k 为 k 时刻机器人的位姿， u_k 是 $k-1$ 到 k 时刻系统控制量， x_{k-1} 为 $k-1$ 时刻机器人的位姿， w_k 为系统控制机器人运动过程的噪声量。

根据其运动过程分析，系统并不满足线性要求，则系统应该为随机非线性时不变的，由 2-2 可以得到其系统方程为：

$$x_k = f(u_k, x_{k-1}) + w_k \quad (2-3)$$

其中, x_k 为 k 时刻机器人的位姿, u_k 是 $k-1$ 到 k 时刻系统控制量, x_{k-1} 为 $k-1$ 时刻机器人的位姿, w_k 为系统控制机器人运动过程的噪声量。

2) 观测模型

在机器人运动过程中, 要通过传感器不断获取周围环境信息, 在 SLAM 中, 机器人需要从不断获取到的环境信息中获取当前位置信息和建立地图。虽然机器人的运动模型包含位姿信息, 但其存在误差, 而且这种误差会累积。这时, 就需要建立合适的观测模型来协助机器人完成定位和地图构建。在观测环境时, 传感器会接收到大量环境信息, 比如本系统采用的 RPLINDAR A2 激光雷达, 其旋转频率为 5Hz-15Hz, 每秒测距次数可达 8K, 那么如果所有信息都加入地图中, 必然也会使得系统计算效率下降, 同时由于传感器观测噪声的存在, 这样也会引入更多的噪声。

本系统所采用的激光雷达测距传感器, 其对外部环境的观测值 z 一般包含环境路标与传感器的距离 r 和角度 θ , 可以表示为^[26]:

$$z = \begin{bmatrix} r \\ \theta \end{bmatrix} \quad (2-4)$$

机器人在运动时, 所观测的环境信息与本时刻的位置和姿态有关, 和之前时刻观测的信息无关, 即在 k 时刻所观测到的环境路标只和本时刻位置与姿态 x_k 有关。观测模型的概率分布可以用机器人本时刻的位姿、上一时刻的位姿、系统的控制信号及控制噪声表示:

$$P(z_k | x_k, v_k) \quad (2-5)$$

其中, x_k 为 k 时刻机器人的位姿, z_k 是 k 时刻环境观测路标, v_k 为传感器观测噪声量。

激光雷达传感器的观测模型与距离和角度有关, 不同传感器其观测量也不相同, 可以将观测模型用一般表达式表示为:

$$z_k = h(x_k) + v_k \quad (2-6)$$

其中, x_k 为 k 时刻机器人的位姿, z_k 是 k 时刻环境观测路标, v_k 为传感器观测噪声量。

为进一步探究其模型，将 2-3 代入 2-5 可得：

$$z_k = h(x_k) + v_k = \begin{bmatrix} \sqrt{(x_i - x_k^v)^2 + (y_i - y_k^v)^2} \\ \tan^{-1} \frac{y_i - y_k^v}{x_i - x_k^v} - \Phi_k^v \end{bmatrix} + v_k \quad (2-7)$$

其中， (x_i, y_i) 为观测到的第 i 个环境特征的坐标。

对其求导可得观测模型的雅克比矩阵^[27]：

$$H_k = \frac{\partial h}{\partial x} \Big|_{x_{k,k-1}} = \begin{bmatrix} \frac{\Delta x}{\Delta} & \frac{\Delta y}{\Delta} & 0 & 0 & \cdots & -\frac{\Delta x}{\Delta} & -\frac{\Delta y}{\Delta} & \cdots & 0 \\ -\frac{\Delta y}{\Delta^2} & -\frac{\Delta x}{\Delta^2} & 0 & 0 & \cdots & -\frac{\Delta x}{\Delta^2} & -\frac{\Delta y}{\Delta^2} & \cdots & 0 \end{bmatrix} \quad (2-8)$$

其中， $\Delta x = (x_i - x_k^v)$ ； $\Delta y = (y_i - y_k^v)$ ； $\Delta = \sqrt{(\Delta x)^2 + (\Delta y)^2}$ 。

2.1.3 融合 SLAM 算法引出

在 SLAM 中，有基于视觉的 SLAM 和基于激光的 SLAM，视觉常采用的传感器有单目摄像头、双目摄像头、深景摄像头等，激光 SLAM 常采用传感器为激光雷达传感器。视觉 SLAM 信息丰富，但信息冗余度高，计算量大，易受光线干扰。基于激光雷达的 SLAM 算法抗干扰能力较强，计算量较小，易于实现，所以本文选择激光雷达作为主要传感器。

一般基于激光雷达的 SLAM 算法常采用的是基于里程计模型进行航迹推算进而通过滤波算法将推算结果与观测数据融合，而这种方法存在的问题。一般里程计采用的是编码器进行里程计算，编码器是利用码盘将光电信号转化为脉冲信号进行读取的，然后通过左右两轮编码器的读数来进行航迹推算。在实际使用中由于编码器自身存在的误差、机器人速度过快引起的漂移、路面不平整导致的两轮位移距离差与模型有出入、机器人单轮或两轮出现腾空等问题使得编码器存在误差，而这种误差会不断累积使得定位精度和地图构建精度逐渐下降，出现如图 2-2 所示的混叠、重复乱做一团，无法进行导航。



图 2-2 基于里程计的 SLAM 点云图

基于此，本文提出一种利用里程计、陀螺仪、加速度计、电子罗盘多种传感器融合的方案来替代里程计进行航迹推算，然后将推算结果与观测数据融合和得到最优位姿估计的方法。

2.2 EKF SLAM 算法

拓展卡尔曼滤波是卡尔曼滤波的改进和优化。卡尔曼滤波是一种适用于线性、离散和有限维系统的算法，对于一个系统，如果有外部变量并且可以转化成状态空间表示，基本都能用卡尔曼滤波来进行计算。卡尔曼滤波需要系统数据和状态服从高斯分布，卡尔曼滤波给出了系统状态的条件概率密度的更新过程线性最小方差估计，也就是最小方差估计。

2.2.1 拓展卡尔曼滤波原理

卡尔曼的对系统的线性要求制约了其使用范围，从 SLAM 运动模型和观测模型来看，其位置和观测到的地图特征量不是线性分布的，但可以从卡尔曼滤波的五个基本公式逐步推导出其改进模型。

系统的状态方程和观测方程：

$$\begin{cases} x_k = Ax_{k-1} + Bu_k + w_k \\ z_k = Hx_k + v_k \end{cases} \quad (2-9)$$

其中， x_k 为 k 时刻系统状态，A 和 B 为系统参数， z_k 是 k 时刻测量值， u_k 为 k 时刻对系统的控制量，H 为测量系统参数， w_k 、 v_k 为过程和测量的噪声量,服从高斯分

布^[28]。

1) 依据上一状态的最优值预测 k 时刻状态:

$$x_{k|k-1} = Ax_{k-1|k-1} + Bu_k \quad (2-10)$$

其中, $x_{k|k-1}$ 为基于 k-1 时刻预测的 k 时刻的系统状态, $x_{k-1|k-1}$ 为 k-1 时刻系统状态最优值, A 和 B 为系统参数, u_k 为 k 时刻对系统的控制量。

2) 得到系统状态预测值后, 需要更新协方差:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_k \quad (2-11)$$

其中, $P_{k|k-1}$ 为 $x_{k|k-1}$ 对应的协方差, $P_{k-1|k-1}$ 为 $x_{k-1|k-1}$ 对应的协方差, A^T 为系统参数 A 的转置矩阵, Q_k 为系统过程噪声 w_k 的协方差。

3) 在得到系统状态预测值后, 结合传感器测量值, 可以得到系统最优估计:

$$x_{k|k} = x_{k|k-1} + Kg_k(z_k - Hx_{k|k-1}) \quad (2-12)$$

其中, $x_{k|k}$ 为 k 时刻的系统状态最优估计, $x_{k|k-1}$ 为基于 k-1 时刻预测的 k 时刻的系统状态, Kg_k 为 k 时刻的卡尔曼增益, z_k 是 k 时刻测量值, H 为测量系统参数。

4) 计算卡尔曼增益:

$$Kg_k = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R_k} \quad (2-13)$$

其中, Kg_k 为 k 时刻的卡尔曼增益, $P_{k|k-1}$ 为 $x_{k|k-1}$ 对应的协方差, H^T 为测量系统参数 H 的转置矩阵, R_k 为系统过程噪声 v_k 的协方差

5) 更新系统协方差:

$$P_{k|k} = (1 - Kg_kH)P_{k|k-1} \quad (2-14)$$

其中, $P_{k|k}$ 为 $x_{k|k}$ 对应的协方差, $P_{k|k-1}$ 为 $x_{k|k-1}$ 对应的协方差, H 为测量系统参数, Kg_k 为 k 时刻的卡尔曼增益。

由公式 2-9 到 2-13 可以看出, 对于线性高斯系统离散序列, 很容易就能计算出其卡尔曼滤波后的结果。公式 2-8 描述的是一个线性高斯系统, 而公式 2-13 为 SLAM 整个过程的描述, 可以看出其并不是线性系统,

$$\begin{cases} x_k = f(u_k, x_{k-1}) + w_k \\ z_k = h(x_k) + v_k \end{cases} \quad (2-15)$$

在扩展卡尔曼滤波中, 系统状态的预测与观测值的预测是由非线性函数计算出的, 在卡尔曼滤波中的表示状态转移的矩阵 A 和观测的矩阵 H , 则可以分别由其 f 和 h 函数的雅克比矩阵代替^[29]。

1) 依据上一状态的最优值预测 k 时刻状态:

$$x_{k|k-1} = f(u_k, x_{k-1|k-1}) + w_k \quad (2-16)$$

2) 得到系统状态预测值后, 需要更新协方差:

$$P_{k|k-1} = \alpha_k P_{k-1|k-1} \alpha_k^T + Q_k \quad (2-17)$$

3) 在得到系统状态预测值后, 结合传感器测量值, 可以得到系统最优估计:

$$x_{k|k} = x_{k|k-1} + Kg_k(z_k - h(x_{k|k-1})) \quad (2-18)$$

4) 计算卡尔曼增益:

$$Kg_k = \frac{\alpha_{k|k-1} \beta^T}{\beta_k P_{k|k-1} \beta_k^T + R_k} \quad (2-19)$$

5) 更新系统协方差:

$$P_{k|k} = (1 - Kg_k \beta_k) P_{k|k-1} \quad (2-20)$$

其中, α_k 、 β_k 为 f 和 h 的雅克比矩阵, 若其状态有 n 维, 则^[30]:

$$\alpha_k = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (2-21)$$

$$\beta_k = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \quad (2-22)$$

2.2.2 拓展卡尔曼滤波的 SLAM 应用

在拓展卡尔曼滤波中，用 x_k 表示系统的状态向量，在 SLAM 算法构建的系统坐标系中， x_k 所包含的信息有：机器人的位姿信息和观测到的路标信息。在本系统中激光雷达扫描的为一个平面即 2D 雷达，则其所建立的坐标系也是二维的，可以将系统状态向量表示为：

$$x_k = \left[\underbrace{x_k, y_k, \theta_k}_{\text{位姿}}, \underbrace{m_1, m_2, \cdots, m_n}_{\text{路标}} \right] \quad (2-23)$$

本系统采用的是激光雷达来进行周围环境信息采集，其观测到的环境特征数据包含两个信息：距离 d 和相对传感器的角度 ϕ_1 。所观测的数据可以表示成如下表格：

表 2-1，传感器观测值

d_1	ϕ_1	d_2	ϕ_2	\cdots	d_n	ϕ_n
-------	----------	-------	----------	----------	-------	----------

基于上一状态估测本时刻的预测值时，一般是利用对系统的控制量 u_k ，在移动

机器人中，常采用里程计或陀螺仪对 u_k 进行描述，那么其描述一般用 Δx 、 Δy 、 $\Delta \theta$ 示：

$$x_{k|k-1} = x_{k-1|k-1} + [\Delta x, \Delta y, \Delta \theta, 0, \dots, 0]^T + w_k \quad (2-24)$$

对状态预测和更新所要使用的 f 和 h 的雅克比矩阵：

$$\alpha_k = \frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & -\frac{\Delta d_r + \Delta d_l}{2} \sin\left(\theta + \frac{\Delta d_r + \Delta d_l}{2b}\right) \\ 0 & 1 & \frac{\Delta d_r + \Delta d_l}{2} \cos\left(\theta + \frac{\Delta d_r + \Delta d_l}{2b}\right) \\ 0 & 0 & 1 \end{bmatrix} \quad (2-25)$$

$$\beta_k = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{x_i - x_k}{d} & \frac{y_i - y_k}{d} \\ -\frac{y_i - y_k}{d^2} & -\frac{x_i - x_k}{d^2} \end{bmatrix} \quad (2-26)$$

2.3 粒子滤波 SLAM 原理

粒子滤波的前身是蒙特卡洛算法。在基于蒙特卡洛的递归贝叶斯算法中，利用已知目标当前观测信息 Y_k ，来推算目标状态矢量 X_k 的后验概率密度 $p(X_k|Y_{1:k})$ ，在得到目标状态矢量 X_k 的后验概率密度后，就可以采用多种估计算法来估计目标状态。在蒙特卡洛近似估计中，会将一些目标状态的离散点进行加权处理，随着采样数量 N 的增大其估计的准确性会提高。

粒子滤波是基于序贯重要性采样，整个过程可以分为：采样、计算权值、重采样等几个部分，采样 N 个粒子一般表示为：

$$X_k^{(i)} \sim p(X_k|Y_{1:k}) \quad (2-27)$$

根据重要性采样，目标后验概率密度函数可由 N 个独立的粒子表示的数值采样分布近似估计^[30]。

$$p_N(X_k|Y_{1:k}) = \frac{1}{N} \sum_{i=1}^N \omega_k^{(i)} \delta(X_k - X_k^{(i)}) = \left\{ X_k^{(i)}, \omega_k^{(i)} \right\}_{i=1:N} \quad (2-28)$$

其中， $\omega_k^{(i)}$ 为第 i 个粒子 k 时刻的权重， $\delta(X)$ 为狄拉克函数。

在粒子滤波中，从目标当前观测信息 Y_k 中抽取样本是比较困难的，在重要性采样中，我们引入了粒子权重的概念，通过计算粒子的权值，可以通过去除权值较小的粒子，分裂权值较大的粒子保证采样的准确性，防止粒子退化，其非归一化权值表达式一般为：

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \frac{p(Y_k | X_k^{(i)}) p(X_k^{(i)} | X_{k-1}^{(i)})}{q(X_k^{(i)} | X_{k-1}^{(i)}, Y_k)} \xrightarrow{\text{目标先验状态转移}} = \omega_{k-1}^{(i)} p(Y_k | X_k^{(i)}) \quad (2-29)$$

其中， $p(X)$ 为目标分布， $q(X)$ 为建议分布。

其归一化表达为：

$$\tilde{\omega}_k^{(i)} = \frac{\omega_k^{(i)}}{\sum_{i=1}^N \omega_k^{(i)}} \quad (2-30)$$

最后一步为重采样，在计算完权值后，去掉权值较小粒子，以提高计算效率和降低权值对系统精度的影响。

2.4 MATLAB 仿真测试

在对算法进行仿真时，所采用的仿真平台为 MATLAB。首先对 EKFSLAM 算法进行仿真，仿真实验先选取一块区域，机器人由中间开始移动，围绕场地运动一圈，记录下路径，设置的路路标数目为 15 个，时间为 72 秒，每秒进行 5 次位姿采样，位姿预测采用编码器，图 2-3 为仿真实验结果。

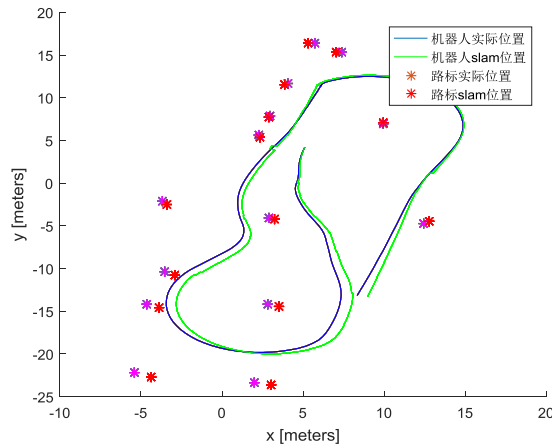


图 2-3 EKFSLAM 算法仿真

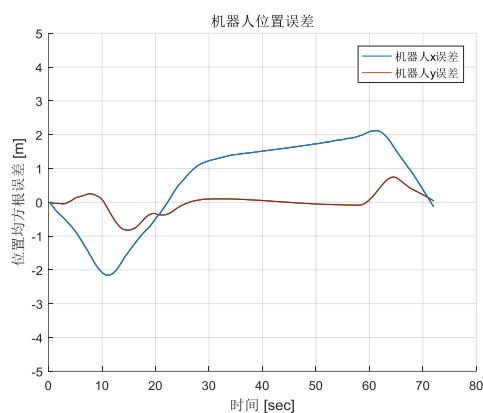
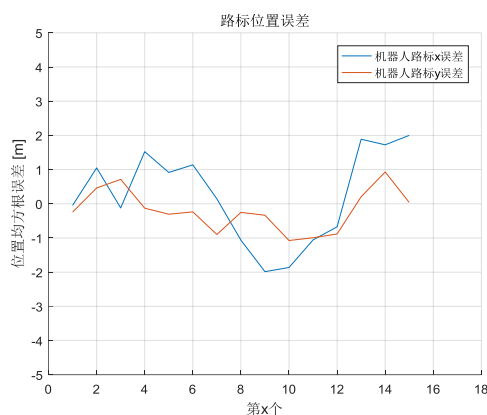


图 2-4 机器人位置误差



2-5 路标测量误差

由结果可以看出，在使用 **EKFSLAM** 算法进行定位算法时，由于实验采用的是里程计模型，其误差还是比较大的，可以看出，在初始时刻误差相对较小，但误差随着时间逐渐变大，并且在变化趋势也在增大。

对粒子滤波 **SLAM**，实验条件选取与上面保持一致，机器人同样由中间开始移动，围绕场地运动一圈，记录下路径，设置的路路标数目为 15 个，时间为 72 秒，每秒进行 5 次位姿采样，位姿预测采用编码器，图 2-6 为仿真实验结果，图 2-7 和图 2-8 为误差。

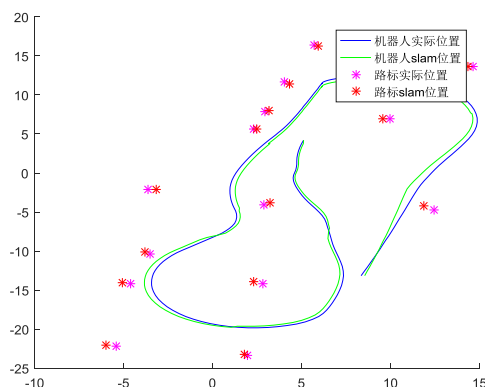


图 2-6 粒子滤波 SLAM 算法仿真

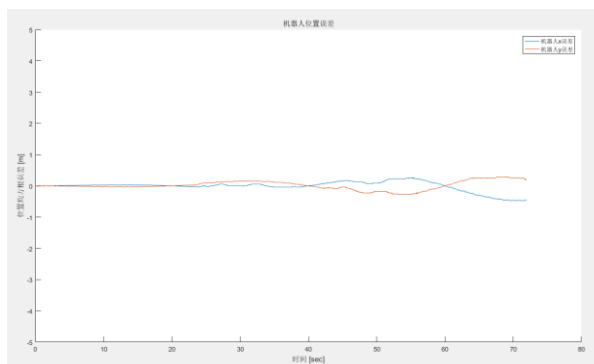
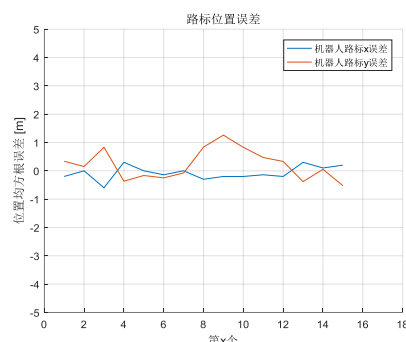


图 2-7 机器人位置误差



2-8 路标测量误差

由结果可以看出，与 EKFSLAM 相比，在使用粒子滤波 SLAM 算法进行定位算法时，定位的误差要明显较小，但随着时间的变化，误差也在累积，精度有所下降，综上，本系统采用粒子滤波来进行定位和地图构建。

2.5 本章小结

本章主要介绍了 SLAM 常用的两种滤波算法：基于拓展卡尔曼滤波的 SLAM 算法和基于粒子滤波的 SLAM 算法。针对拓展卡尔曼滤波，先对基本的卡尔曼滤波进行分析，解释其原理，分析其应用范围和局限性，然后基于卡尔曼滤波推导出其适用于非线性高斯系统的拓展卡尔曼滤波，并给出了其基于里程计模型的雅克比矩阵。对粒子滤波，简单分析了其整个算法流程。下一章将结合 SLAM 算法的运用，介绍整个扫地机系统设计。

3 多传感器融合 SLAM 算法系统设计

3.1 多传感器融合 SLAM 系统流程

可以将整个系统分为三块，如图 3-1，包括信号采集部分，SLAM 过程和应用三部分。

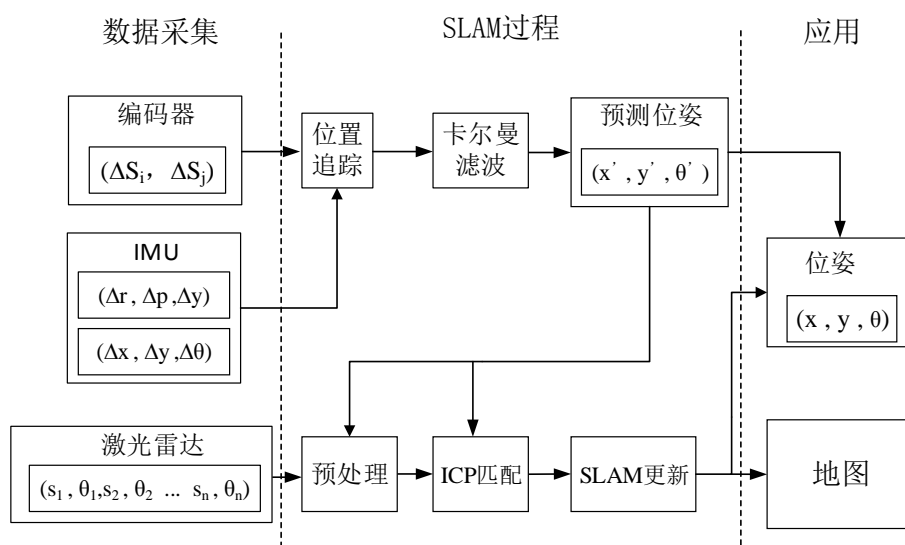


图 3-1 SLAM 系统流程图

采用编码器、陀螺仪和激光雷达对运动及环境数据进行采集，编码器为光电增量式编码器，其特点是能够实时显示轮子移动的速度和角度变化，经过测试发现，编码器在速度较低且无碰撞时，角度精度在短时间内还是比较高的，而陀螺仪因为存在零点漂移在匀速或较低速度时误差还是比较大的，在对 MPU9250 测试时，放置不动的陀螺仪仍然有角加速度的输出，故采用编码器对陀螺仪的测量误差进行估计，然后通过卡尔曼滤波融合加速度计角度，从而保证在一定时间内具有较高精度的预测位姿。激光雷达采集的数据需要先预处理滤除干扰粒子，然后与前一时刻点云匹配得到位姿变化，可用匹配位姿代替预测位姿，然后采用粒子滤波 RBPF 进行最优位姿估测和跟新地图。

3.2 传感器独立模型及误差分析

3.2.1 里程计模型及误差分析

扫地机采用的是光电增量式编码器，码盘的安装方式如图 3-2 所示，采用红外对管进行码盘空格的检测，用过定时器和计数器进行转速检测，通过系数修正可以找到电机转动与轮子前进距离 S 对应的关系。



图 3-2 电机及码盘安装

机器人在左右轮子分别安装有一个编码器，那么机器人的运动可以看成沿着圆的边沿运动，此时两轮相当于沿着同心圆圆弧的差动运动。如图 3-3 所示，假设机器人的内侧轮子运动距离 S_a ，外侧侧轮子运动距离 S_b ，两轮之间距离为 L ，则机器人的行驶距离为 ΔS ，行驶角度的变化 $\Delta\theta$ 可以表示为

$$\begin{cases} \Delta S = \frac{S_a + S_b}{2} \\ \Delta\theta = \frac{S_a - S_b}{L} \end{cases} \quad (3-1)$$

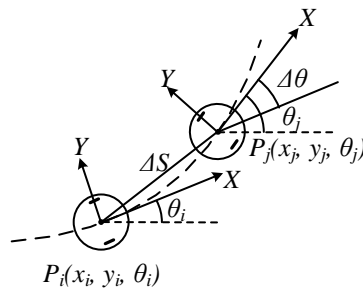


图 3-3，里程计运动模型

此时，可以将 j 点的位姿由 i 点位姿表示为

$$P_j = \begin{bmatrix} x_j \\ y_j \\ \theta_j \end{bmatrix} = \begin{bmatrix} x_i + \Delta S \cos(\theta_i + \Delta\theta) \\ y_i + \Delta S \sin(\theta_i + \Delta\theta) \\ \theta_i \end{bmatrix} \quad (3-2)$$

根据其模型可以看出，里程计模型的误差源主要有三点：机器人速度过快引起的漂移、路面不平整导致的两轮位移距离差与模型有出入、机器人单轮或两轮出现腾空。

3.2.2 IMU 模型及误差分析

惯性测量单元（Inertial measurement unit, IMU），是测量物体三轴加速度及三轴姿态角的传感器，一般包括三轴加速度计和三轴角速度计，有些芯片内部也可能集成了三轴磁力计。其输出信号一般为三轴的加速度和三轴的角速度，而在实际运用中常使用的是俯仰角（Pitch）、偏航角（Yaw）、横滚角（Roll）和三个轴上的

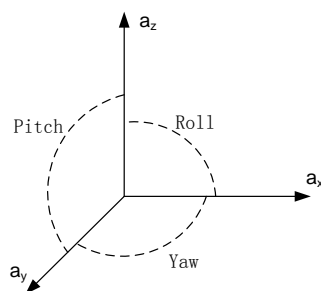


图 3-4 IMU 模型

位移 S_x 、 S_y 、 S_z 。陀螺仪和加速度计的数据是相互独立的，一般加速度计的精度是比较高的，而陀螺仪的精度比较差，所以常使用加速度计的数据通过卡尔曼滤波对角度进行修正。

3.2.3 激光雷达模型及误差分析

激光雷达测距常用的方式有 TOF 和三角测距，TOF 是通过测量激光传播的时间来测量距离，需要极高时钟来提高测量精度一般价格比较贵，国内低端激光雷达一般采用几何方法测距，噪声较大，但成本较低。本系统采用的是国产 RPLIDAR A2

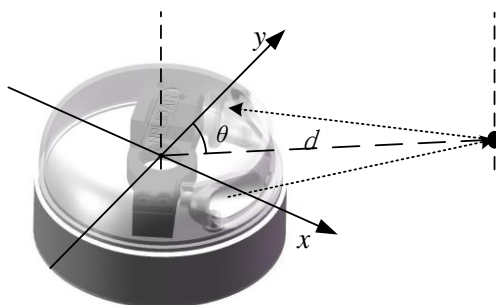


图 3-5 激光雷达测距原理

激光雷达，如图 3-5 所示，其所依据的就是三角测距，其有自己的坐标系，其坐标系为过中点的直角坐标系，每次采集的点均计算其基于传感器坐标系的夹角 θ 和距离 d 。图 3-6 是一种室内环境使用下的示意图。

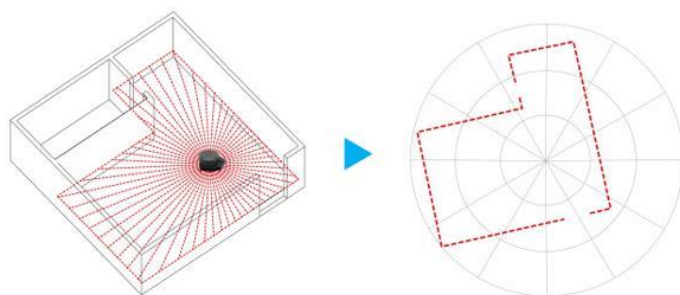


图 3-6 RPLIDAR 扫描所得环境示意图

RPLIDAR A2 激光传感器所采用的三角测距原理使其很难将量程做到很远，并且会有很多噪声，其所宣称的 12m 也是在算法优化下的结果，一般 6m 内的精度还是可以保证的，相较 TOF 激光雷达可对信号进行编码，其所采用的相机模组与透镜组合抗干扰能力较差，在大噪声背景下很难使用，在使用时需要注意这些干扰。

3.3 里程计和 IMU 融合位姿预测

从上述里程计定位模型和 IMU 定位模型可以发现对位姿中角度的预测精度一般

都不是很高，在低速平整的地面时，编码器精度比较高，但此时陀螺仪由于静态漂移的存在，其误差是比较大的，因此，先采用编码器对陀螺仪的测量误差 w_g 进行估计，然后通过卡尔曼滤波融合角速度传感器数据。

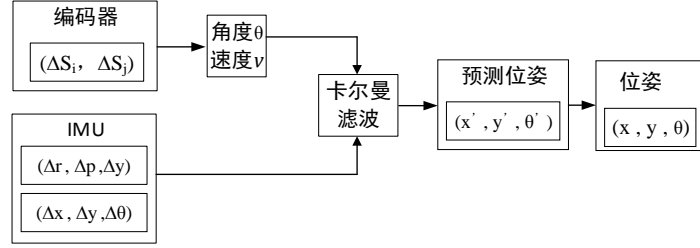


图 3-7 基于里程计和 IMU 的位姿预测框图

首先建立系统的运动方程和观测方程：

$$\begin{cases} x_k = \begin{bmatrix} 1 & -Ts \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} Ts \\ 0 \end{bmatrix} (\omega_{gyro})_{k-1} - \begin{bmatrix} (w_g)_k Ts \\ 0 \end{bmatrix} \\ (\varphi_{acce})_k = [1 \quad 0] x_k + (w_a)_k \end{cases} \quad (3-3)$$

其中， φ 为系统角度变化， b 为陀螺仪静态漂移， ω_{gyro} 为陀螺仪输出的角速度， w_g 为陀螺仪测量噪声， φ_{acce} 为加速度计计算得到的角度， w_a 为加速度计计算角度的误差。令采样周期为 Ts 则：

1) 根据陀螺仪建立系统状态的先验估计：

$$x_k = Ax_{k-1|k-1} + BU_k \quad (3-4)$$

其中， $A = \begin{bmatrix} 1 & -Ts \\ 0 & 0 \end{bmatrix}$ ， $B = \begin{bmatrix} Ts \\ 0 \end{bmatrix}$ ， $U_k = (\omega_{gyro})_{k-1} - (w_g)_k$

2) 系统过程噪声协方差阵为 Q ，测量误差的协方差矩阵为 R ，计算系统协方差矩阵：

$$\begin{cases} Q = \begin{bmatrix} q_{acce} & 0 \\ 0 & q_{gyro} \end{bmatrix} \\ R = [r_{acce}] \end{cases} \quad (3-5)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (3-6)$$

3) 更新系统状态最优估计值：

$$x_{k|k} = x_{k|k-1} + (K_g)_k (Z_k - Hx_{k|k-1}) \quad (3-7)$$

其中， K_g 为卡尔曼增益， $H = [1 \ 0]$

4) 计算卡尔曼增益：

$$(K_g)_k = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R} \quad (3-8)$$

5) 更新误差协方差矩阵：

$$P_{k|k-1} = (I - (K_g)_k H)P_{k-1|k-1} \quad (3-9)$$

此时得到的位姿(x' , y' , θ')为预测位姿，其精度随着误差的逐渐积累而变差，并不能保障长时间的精度，但短时间内其可以作为系统的位姿参与地图的遍历和路径规划，使系统沿着规划好的路径行进。有效的提高此时定位的精度可以降低系统对SLAM 更新频率的要求，从而降低对硬件性能的要求，使其能更好更广泛地引入不同应用平台。

3.4 基于融合位姿预测的激光雷达数据处理流程

3.4.1 采集数据滤波

对激光雷达采集到的点云需要先进行预处理。如图3-8是在室内采集的一组数据，

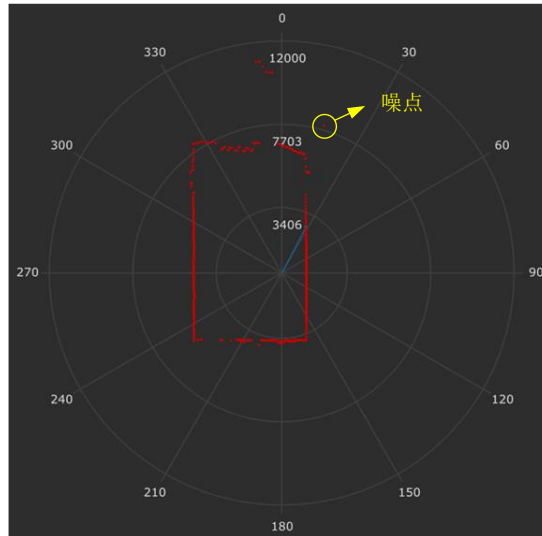


图 3-8 RPLIDAR 扫描室内环境噪点展示

其中有一处明确的噪点，激光雷达扫描到的点云具有以下特点：

- 1) 离散盲目性：由于障碍物表面不平整，反射率不同，激光雷达扫描到的数据在障碍物附近呈现出离散分布的，对障碍物扫描的点也具有盲目性，通过在障碍物上随机扫描的一些点来表示其位子。
- 2) 密度分布不均：激光雷达的扫描是 360 度匀速扫描，但扫描到的障碍物斑点密度不同，较近的物体一般密度比较高，还会受到反射率的影响，垂直的墙壁的密度比倾斜墙壁的密度高一些，雷达在移动速度高的时候也会使点云密度降低。
- 3) 扫描区空白：在点云中超出量程的障碍物分布是无法扫描到的，存在倾角的镜面一般也无法扫描到或密度很低，此外，激光照到水中也会部分吸收掉。

数据重新采样是为了滤除噪声点和使点云的分布比较均匀，然后作为后续滤波算法的输入。重采样的方法一般采用虚拟网格的方法，如图 3-9 所示，网格的距离

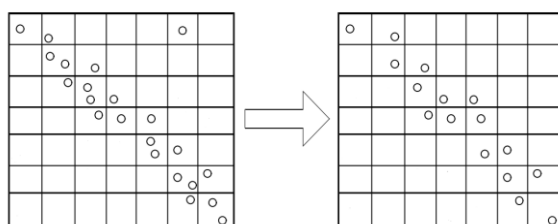


图 3-9 虚拟网格抽取数据

一般采用略小于点与点之间的平均距离的，如果平均距离是 7cm 那么栅格距离就采用 5cm。每个网格可能有多个点，点越多证明其可靠性越高，对孤立的点，如图 3-9 所示，可以直接将其去掉，判断的依据是，先以其为中心做出 3×3 的宫格，判断其余的 8 个格子中是否有点，若没有即可将其作为噪点去除。

3.4.2 激光雷达点云匹配定位

激光雷达在运动时，对周围的环境不断的扫描，每扫描一周的数据可以称为一帧，点云匹配的方法是指通过下一帧数据对上一帧通过匹配得到这两帧数据采集时刻的位移差和角度差，通过累加得到相对初始时刻的位置。

将雷达在 k 时刻扫描到的一帧数据记为 p_k ，激光雷达扫描的数据是通过极坐标表示的数据传输的（参考图 3-5），则可以表示其中的一个点为 (d_k, α_k) ，转化到直

角坐标系可以表示为:

$$\begin{cases} x_k = d_k \cos \alpha_k \\ y_k = d_k \sin \alpha_k \end{cases} \quad (3-10)$$

假设在 $k+1$ 时刻扫描到的一帧数据为 p_{k+1} ，那么可以假设与 k 时刻点 (d_k, α_k) 对应点的极坐标为 (d_{k+1}, α_{k+1}) 和直角坐标为 (x_{k+1}, y_{k+1}) ，如图 3-10 所示，对同一位置，前一帧数据对应的点云，到下一帧数据的点云，具有相似性，将两帧数据进行

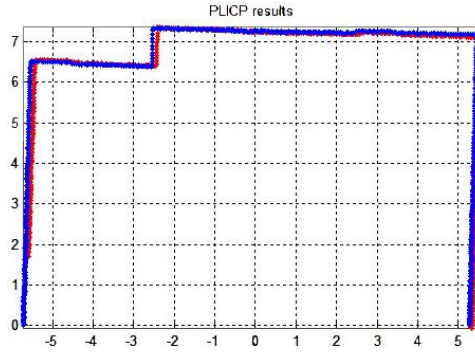


图 3-10 点云匹配示意图

匹配可以得到角度变化 $\Delta\theta$ 和位置变化 $(\Delta x, \Delta y)$ ，那么，其对应关系可以表示为:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{bmatrix}}_{\text{旋转矩阵}} \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} + \underbrace{\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}}_{\text{平移矩阵}} \quad (3-10)$$

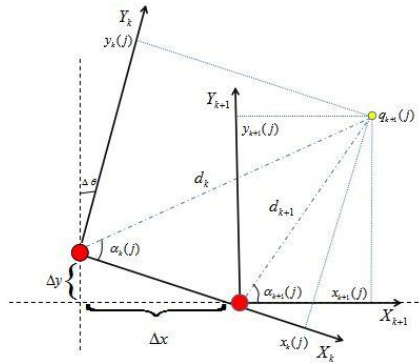


图 3-11 点云位移与角度变化示意图

常用的点云匹配算法有迭代最近点 (ICP, Iterative Closest Point)，ICP 算法可

以将不同坐标系的点云数据合并到同一坐标系，匹配过程就是要找到两个坐标系之间的变换规则，其本质是基于最小二乘法的优化配准方案，其不断的重复的选择两个点云之间的匹配关系，直到匹配关系达到所设置的收敛精度要求。具体操作步骤如下：

- 1) 计算点云 p_k 与点云 p_{k+1} 对应关系的最近点。
- 2) 对上述的对应最近点求最小平均距离的转换关系，得到位置变化和角度变化关系。
- 3) 将点云 p_{k+1} 根据上一步中得到的变换关系进行平移和旋转得到新的点云 q_n 。
- 4) 计算新的点云 q_n 与点云 p_k 对应关系的平均距离，如果小于设置的阈值，则停止迭代，否则用 q_n 代替 p_{k+1} 继续进行迭代，直到满足上述阈值。

3.4.3 RBPF 粒子滤波算法流程

在 RBPF-SLAM 中，首先是基于里程计和陀螺仪数据融合或点云匹配得到预测位姿，然后与激光雷达数据结合运动预测方程得到最优位姿估计，并将新的特征点添加到地图中去。

在上述的预测位姿模型中，本文采用的方法融入了多种传感器的数据来提高预测位姿的精度，在 RBPF-SLAM 中，需要这些数据作为运动预测方程的输入以得到重要性密度函数。

可以将 RBPF-SLAM 用以下公式表达：

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) = p(m|x_{1:t}, z_{1:t})p(x_{1:t}|z_{1:t}, u_{1:t}) \quad (3-11)$$

其中， $x_{1:t}$ 表示机器人的位姿， m 表示运动过程中构建的地图， $z_{1:t}$ 表示激光雷达观测到特征序列， $u_{1:t}$ 表示系统预测运动过程的输入，对于本系统就是基于里程计、IMU 和点云匹配对运动过程的描述。

$p(x_{1:t}, m|z_{1:t}, u_{1:t})$ 表现的是整个 SLAM 过程，机器人通过系统控制模型和观测模型不断完成对自身的定位和构建全局地图。 $p(m|x_{1:t}, z_{1:t})$ 和 $p(x_{1:t}|z_{1:t}, u_{1:t})$ 则分别表示对构建地图的更新和机器人自身定位。

可以采用以下 4 步对整个 RBPF-SLAM 的流程进行总结：

1) 在初始时刻指定机器人的位姿为 (x_0, y_0, θ_0) ，基于初始位置开始构建全局地图，构建的过程为采用激光雷达扫描周围环境，对点云进行滤波建立局部地图，并将此地图作为初始时刻的全局地图不断进行完善。初始化样本为 $x_0^{1:N} \sim q(x_0)$ ， N 表示初始时刻粒子数目，初始粒子权值 $\omega_0^{(i)} = \frac{1}{N}$ 。

2) 在时间 t 内计算陀螺仪、里程计和点云匹配数据 u_t ，通过重要密度函数 $q(x_t | x_{t-1}^{(i)}, z_t)$ 来抽取并得到新的样本粒子 $x_t^{(i)}$ ，每个粒子对系统状态的描述并不是对等的，需要对粒子的权值进行计算以确定粒子的可靠性，权值较大的粒子其可靠性也较高。

3) 经过多次迭代后，很多粒子的权重就会变得很小，几乎可以忽略掉，而有少数的粒子的权重却会变得很大，并且随着迭代，有效粒子的数目会更少，此时需要根据有效粒子的数目来衡量粒子退化程度，并决定是否进行重采样。

4) 根据滤波后得到的期望进行位姿估计，并根据新的位姿估计结合激光雷达扫描的数据对全局地图进行更新。

从以上计算过程我们可以看出，整个过程所要解决的问题实际上是对系统控制噪声和观测噪声进行近似采样以得到系统状态的真实分布。

3.4.4 栅格地图的构建

常用的地图有栅格地图、拓扑地图等，栅格地图的优点是容易表示，容易建立，位置具有唯一性，对于较短路径的规划比较方便，在机器人中常采用栅格地图进行导航。

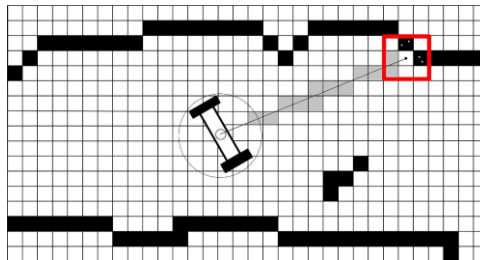


图 3-12 栅格地图建立过程示意图

前面对激光雷达数据进行分析的时候，其中有一点是点云的分布具有随机性和盲目性，比如 RPLIDAR 在对墙壁进行扫描的时候，其扫描到的点云数据沿一条直线两边随机分布，所扫描到的位置也是随机的。那么其对于任意一个栅格就引入了一个占据率的问题，通常采用 $p(s=1)$ 来表示空置自由状态的概率， $p(s=0)$ 来表示被占据状态的概率，为方便表示，一般采用两个的比值作为该栅格的状态：

$$Odd(s) = \frac{p(s=1)}{p(s=0)} \quad (3-11)$$

当新来了一个点时 $z \sim \{0,1\}$ 时，需要更新它的状态，如果更新之前该点状态为 $Odd(s)$ ，那么在 z 条件下更新表达式为：

$$Odd(s|z) = \frac{p(s=1|z)}{p(s=0|z)} \quad (3-12)$$

由贝叶斯公式可以得到：

$$\begin{aligned} Odd(s|z) &= \frac{p(s=1|z)}{p(s=0|z)} = \frac{p(z|s=1)p(s=1)/p(z)}{p(z|s=0)p(s=0)/p(z)} \\ &= \frac{p(z|s=1)}{p(z|s=0)} Odd(s) \end{aligned} \quad (3-13)$$

两边取对数可以得到：

$$\log Odd(s|z) = \log \frac{p(z|s=1)}{p(z|s=0)} + \log Odd(s) \quad (3-14)$$

如果采用 $\log Odd(s)$ 代替 $Odd(s)$ 表示 s 状态 S ：

$$S^+ = S^- + \log \frac{p(z|s=1)}{p(z|s=0)} \quad (3-15)$$

其中， S^+ 表示更新后 s 的状态， S^- 表示更新之前 s 的状态。

对于 $\log \frac{p(z|s=1)}{p(z|s=0)}$ 也只存在两种情况，观测到时为 $\log \frac{p(z=1|s=1)}{p(z=1|s=0)}$ 和未观测到时为 $\log \frac{p(z=0|s=1)}{p(z=0|s=0)}$ ，初始状态下 $\log Odd(s) = \log \frac{p(s=1)}{p(s=0)} = \log \frac{0.5}{0.5} = 0$ ，此时更新一个点的状态就转化为加减运算了。

激光雷达扫描到的点是以极坐标的形式表示的，如图 3-13，假设其中一个点为 $p(d, \alpha)$ ，此时扫地机器人在全局坐标系下的坐标及角度为 (x, y, θ) ，激光雷达的安装位置为过坐标 (x, y) 与 x 轴呈 θ 度角的线上，即在扫地机器人自身坐标系的 y 轴上，距离为 l ，此时障碍物在全局坐标系中的位置 (x_0, y_0) 可以描述为：

$$\begin{cases} x_0 = x + l\cos\theta + d\cos(\theta + \alpha) \\ y_0 = y + l\sin\theta + d\sin(\theta + \alpha) \end{cases} \quad (3-16)$$

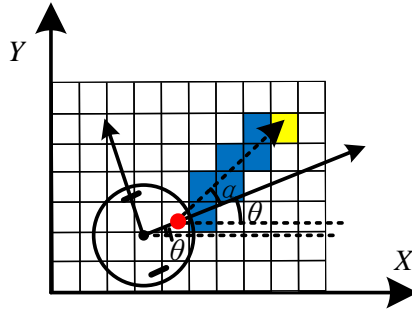


图 3-13 激光雷达扫描点转化全局坐标系示意图

令 $\log \frac{p(z=1|s=1)}{p(z=1|s=0)}=0.9$ ，则 $\log \frac{p(z=0|s=1)}{p(z=0|s=0)}=-0.7$ ，如图 3-14 所示，采用占据率的栅格地图建立过程，值越大证明该点被占据的概率越高。

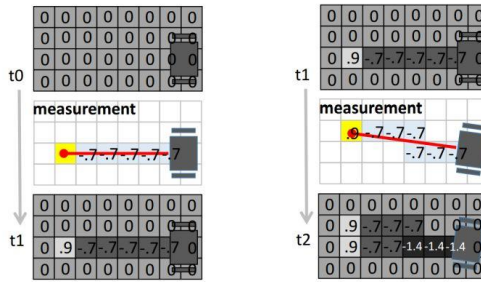


图 3-14 采用占据率的栅格地图建立过程示意图

3.5 本章小结

本章主要介绍系统整个定位和地图构建的设计过程。首先分传感器数据采集、SLAM 过程、应用层三个部分从总体上介绍了扫地机器人整个即时定位与地图构建的算法框架。然后分别从里程计、陀螺仪、激光雷达传感器所采集的数据及其相应定

位模型来分析各传感器数据优缺点和误差来源，为后面应用做铺垫。随后将里程计与 IMU 数据进行融合，得到基于系统控制的位姿预测，将此预测作为扫地机短时间的位姿参与系统控制过程，此位姿预测也作为后面 SLAM 粒子滤波的重要密度函数抽取粒子。最后介绍基于激光雷达的定位与地图构建过程，先对初始扫描数据进行滤波，然后介绍了基于点云匹配的定位过程，通过引入回环检测可以降低累积的误差，提高定位精度，随后通过四步介绍了粒子滤波的一般过程，最后对栅格地图理论进行介绍，并分析了栅格地图建立的整个过程。

4 扫地机器人平台搭建及仿真测试

构建扫地机器人整个系统，首先需要考虑整个系统的功能，整个系统的功能可以划分为以下六个部分：清扫功能、自身安全保护、人机交互、路径规划、电源管理、故障检测，每个功能包含很多子功能。其次要对整个系统功能的实现进行评估，其所需要的资源和计算性能需求，相应的选择系统的核心控制平台和软件的开发平台。最后对众多功能模块进行方案的选择和设计。

对软件开发平台，主处理器选择在 Linux 中运行 ROS，协处理器开发选择 keil 平台进行开发。测试采用在实地进行地图构建、定位测试、地图闭环测试。

开发测试平台如图 4-1 所示，其已经集成包括 IMU、编码器、WiFi 通信、远程调试通信模块、激光雷达等，下面将对其结构仿真和实验进行介绍。

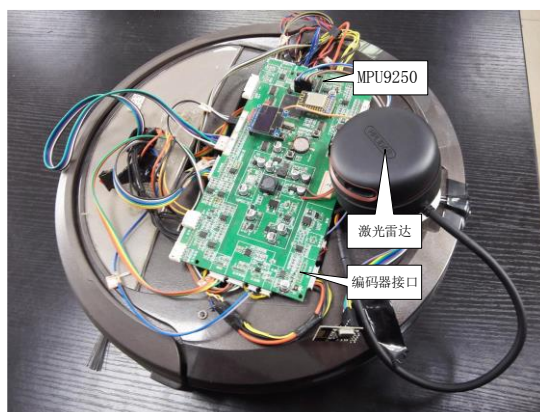


图 4-1 扫地机器人开发样机

4.1 扫地机器人实验平台介绍

扫地机器人整体硬件电路设计框图如图 4-1 所示，整体硬件可以分为清扫模块、安全保护模块、行走控制模块、电源管理模块、通信模块、导航模块、核心控制模块、语音提示等。

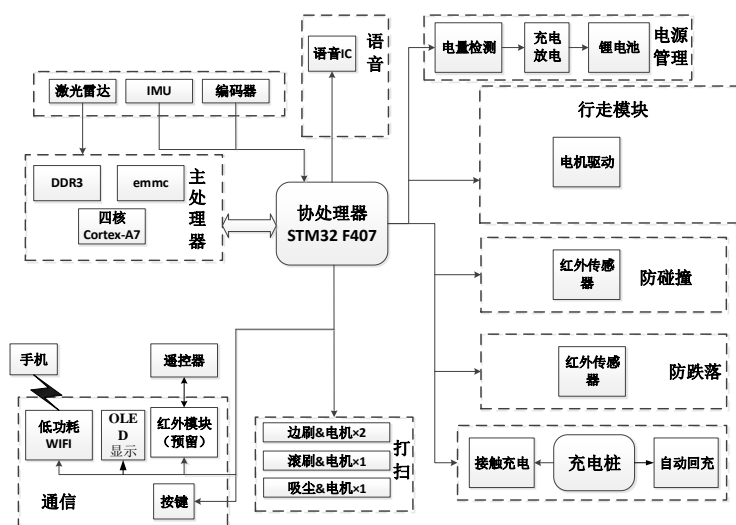


图 4-2 系统硬件设计框图

4.1.1 核心控制系统的选择

本系统所采用的核心控制单元由全志 Allwinner R16 SoC 和 ST 公司的 STM32F407 芯片组成。R16 是由珠海全志科技公司开发的针对智能硬件低成本解决方案，其集成四核 ARM Cortex-A7 处理器，集成 Mali400MP2 GPU，具有非常强大的计算性能，还具有丰富的接口。在软件支持上，其具有基于开源 Linux 的 TinaTM 支持 Android4.4 到 Android6.0，是具有高性价比的解决方案。

经统计，本系统与外部通信和信号采集等使用的 IO 达到八十多个，所以需要—个资源丰富的协处理器。STM32F407ZGT6 是意法半导体公司（ST）开发的，基于 ARM Cortex-M4 内核的单片机。ARM Cortex-M4 内核是 ARM 公司开发的带有 FPU 的 32 位内核，能实现 ART 加速，主频高达 168MHz，具有 MPU（micro processor unit）支持部分 DSP 指令集，能达到 210DMIPS 的处理能力。其外部资源包括，1 MB Flash，192+4KB RAM，17 个 TIM（定时器），3 个 12 位 ADC，2 个 12 位 DAC 转换器，3 个 I2C 接口，4 个 USART，3 个 SPI（42 Mbits/s）以及 SDIO 接口，具有 32 位数据总线的灵活外部存储控制器：SRAM、PSRAM、NOR/NAND，方便拓展外部存储器。相比于传统 MCU，STM32F407ZGT6 加入了硬件 FPU 和 DSP 指令集，其 168MHz 的主频，使其具有较强的浮点运算能力，较强的性能和丰富的

外设满足本系统各功能模块所需的接口。

4.1.2 编码器与 IMU

编码器所采用的原理与图 4-6 类似，采用红外对管将编码器码盘上的空格转化为脉冲信号，通过定时器和计数器对脉冲进行检测，读取轮子的转速。

IMU 采用 TDK 公司的 MPU9250。MPU9250 内部集成三轴陀螺仪、三轴加速度计、三轴磁力计 9 axis 运动传感器 (Motion Tracking)，另外内部还集成了 DMP 运动传感器，其可以直接对运动数据进行解算，在使用时，考虑到内部集成的 DMP 虽然比较方便，但对所有场景下的使用并不一定精度都比较高，故采用 SPI 和 I²C 的 Master 功能来读取数据，在外部通过卡尔曼滤波结合编码器的数据对运动姿态进行解算。

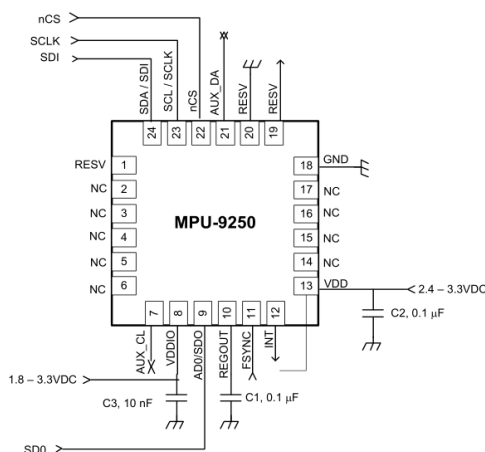


图 4-10 MPU9250 接线方式

4.2 软件开发平台

常用的机器人软件开发测试一般都基于机器人操作系统 (ROS, Robot Operating System)，其起源于斯坦福大学人工智能实验室的 STAIR 项目与 WillowGarage 机器人项目之间的合作，相较于常用的操作系统，ROS 有许多不同的地方，如图 4-11 所示，ROS 是一个针对机器人开发的开源操作系统，其提供了很多方便功能，如硬件

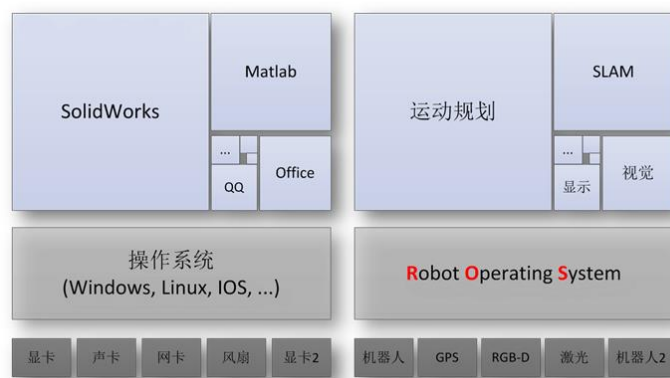


图 4-11 操作系统与 ROS 区别

抽象、底层设备控制、进程间消息传递和程序包管理等操作系统的功能，此外还有许多开发的库和工具来对机器人进行开发。

本系统开发所采用的为基于 Ubuntu 14.04 的机器人操作系统 Indigo，ROS 至今已经有十多个版本了，常使用的版本就是 2014 年 ROS Indigo Igloo，其基于 Linux 系统 Ubuntu 14.04，有许多丰富的应用软件，如图 4-12 是最常用的动力学传感器仿真软件 GAZEBO，图 4-13 为常用的 3D 可视化根据 Rviz，其可以用来显示地图点云模拟，本系统设计就采用 GAZEBO 和 Rviz 进行仿真和测试。

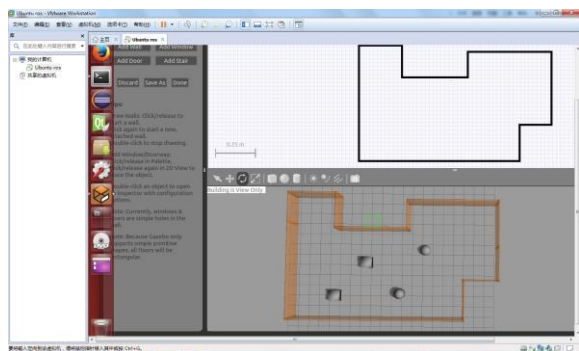


图 4-12 ROS 应用软件 GAZEBO

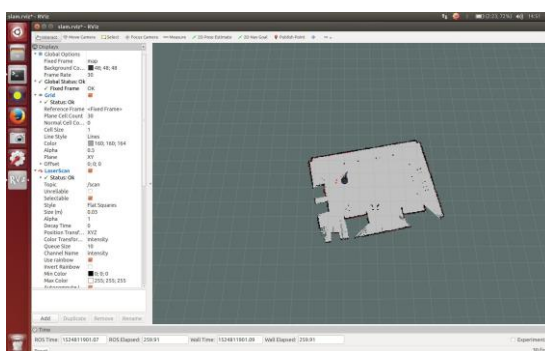


图 4-13 ROS 应用软件 RVIZ

STM32F407 开发采用 Keil 公司的 Keil uVision5，如图 4-14 所示，Keil uVision5 是一款十分方便的开发软件，其有大量的功能库可以采用，在开发时为节省时间一般采用库函数版本进行开发，固件库中封装了大量的便于操作配置底层寄存器的函数。

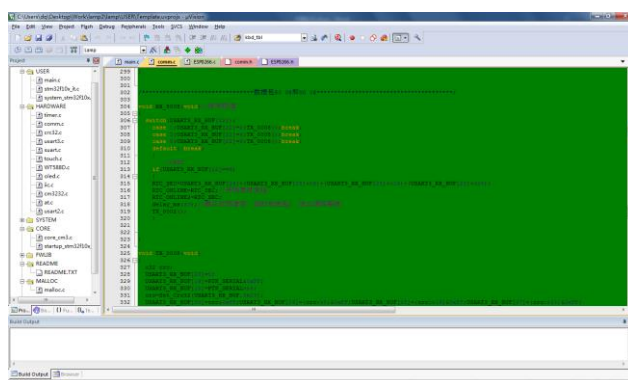


图 4-14 Keil uVision5 软件界面

4.3 系统仿真及测试

4.3.1 系统仿真

仿真是基于软件 Gazebo，Gazebo 是 Indigo 中是一款开源的机器人仿真软件，其可以对机器人的动作和行为进行仿真，还可以对模型添加传感器进行仿真，在本次仿真中，我们加入里程计、IMU 和激光雷达进行仿真。

首先需要建立扫地机器人的模型和搭建仿真环境，机器人模型的搭建一般采用 URDF（Unified Robot Description Format）采用 XML 语言对对机器人进行描述，其可以描述机器人的外观形状、质量、碰撞属性等特点如图 4-15 所示，机器人仿真模型。如图 4-16 所示是实验环境的模拟。

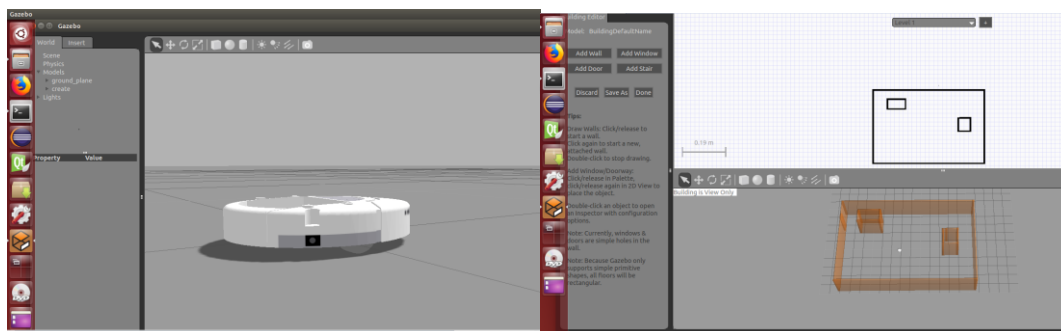


图 4-15 扫地机器人在 Gazebo 的模型

图 4-16 仿真环境的模拟

仿真的步骤分为两个部分进行，首先对基于里程计模型的激光雷达的 SLAM 算法进行仿真；然后对基于里程计和 IMU 的激光雷达的 SLAM 算法进行仿真。下图中，图 4-17，是基于里程计模型的仿真，图 4-18 是基于 IMU 和里程计模型的仿真。

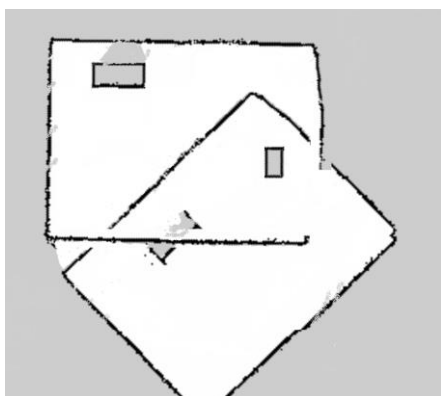


图 4-17 基于里程计模型的仿真

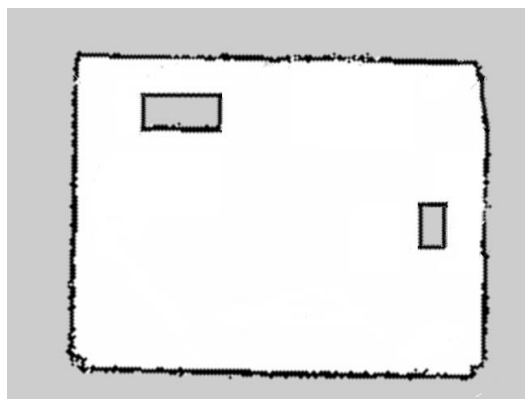


图 4-18 基于 IMU 和里程计模型的仿真

通过对构建的地图分析可以看出，里程计模型的累积误差比较大，随着实验的进行，地图出现了很严重的重叠现象，当加入了 IMU 后整体得到了很大的改善，证明融合 SLAM 算法的定位和构建地图的精度要高于使用里程计模型的 SLAM 算法。

4.3.3 扫地机器人的 SLAM 测试

本次设计的扫地机器人，其主要参数参数如下表所示，

表 4-1 扫地机器人参数

参数	值
长	32.5cm
宽	32.5cm
高	10.2cm
轮距	20.0cm
车轮直径	6.3cm

实验设备如图 4-19 所示，其开发是基于扫地机器人模型重新进行硬件电路及软

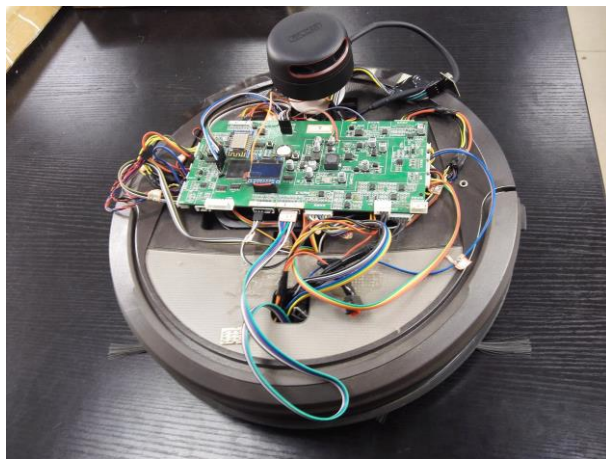


图 4-19 扫地机器人



图 4-20 实验场地



图 4-21 MPU9250 和 RPLIDAR A2

件进行设计，其硬件包含 IMU、编码器、红外测距、RPLIDAR A2 激光雷达及其它控制和传感器电路。激光雷达选用 RPLIDAR A2，如图 4-21 所示，其有效距离为 12 米，在 6 米内精度一般较高，采用无线供电和光通信，精度高，寿命长，设置其扫描频率为 5Hz，采样率为 2K，则每个周期扫描点数为 400。惯性测量单元采用 MPU9250，其内置三轴加速度计、三轴角速度计、三轴磁力传感器，其加速度计一般精度较高，可以通过对磁力计进行校准作为电子罗盘对方向进行纠正。实验时，为了方便调试和获取数据，这里采用 2.4G 无线通信模块将激光雷达数据传到电脑上定位和构建地图。实验场地如图 4-20 所示，在室内周围放置桌子和凳子作为障碍物。

在测试时先将机器人放置一边的墙角，通过控制扫地机器人让其绕室内行进一周然后，进行定位和地图构建工作，室内场景如图 4-20 所示，实验采用 RVIZ 作为控制和地图信息显示软件。对扫地机器人进行两次测试，其结果如下图所示。

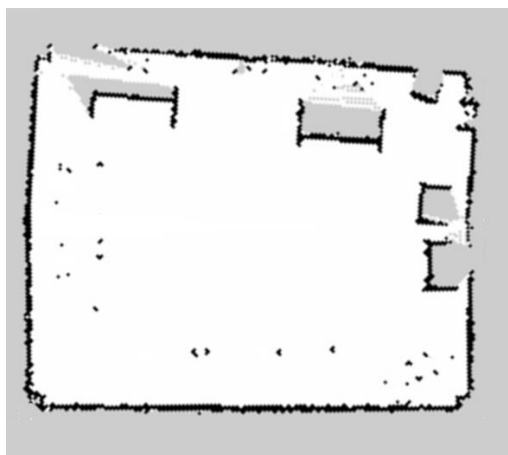


图 4-19 扫地机器人构建结果一

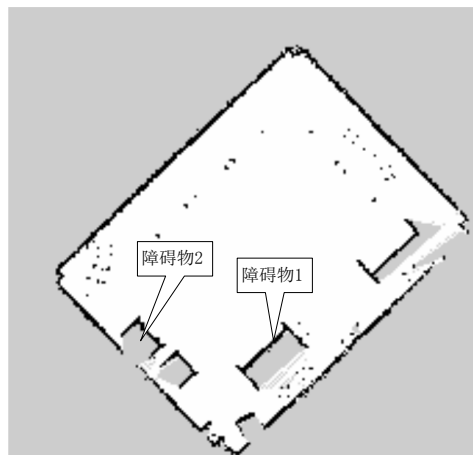


图 4-20 扫地机器人构建结果二

第一次得到的点云出现闭环特性不是很好的问题，所以采用第二次的测试结果进行分析。为了得到实验点云图的精度，对得到的点云中房间尺寸和障碍物尺寸与实验室中的实际尺寸进行对比，结果如表 4-2 所示。

表 4-2 实验室和点云尺寸及误差

实验参数	实际尺寸 (m)	点云尺寸 (m)	误差 (%)
实验室长	6.12	5.87	3.69
实验室宽	4.06	4.23	3.73
障碍物 1 长度	1.35	1.29	4.61
障碍物 1 宽度	0.65	0.68	4.67
障碍物 2 长度	0.85	0.81	4.71
障碍物 2 宽度	0.60	0.63	4.75

由表 4-2 可以计算出，误差的平均值为 4.36%，可以通过障碍物宽度误差计算长度误差来获取点云图的相对误差：

$$\text{相对误差} = \frac{\text{障碍物点云长度} \times \text{障碍物点云宽度误差}}{\text{障碍物实际宽度}} \times 100\%$$

华中科技大学硕士学位论文

计算得到的三个相对误差分别为 5.18%、8.86%、6.1%，则其对应的相对误差分别为 14cm、5.7cm、3.7cm，可以看出点云相对误差并不是很大，其形状比例与室内环境的实际比例相差不大。

在文献[40]中提到了一种基于单目相机的 SLAM 算法，其得到的点云最大相对误差为 3.9%，本系统采用的激光雷达传感器，所得的点云相对误差与其相差不大。

将扫地机器人放置房间中进行定位和地图构建实验，在每过一段时间将扫地机器人进行暂停，读出其定位坐标，然后再测量其实际的坐标（实际测量时基于墙壁作为参考，可能存在误差），测得的数据如表 4-3 所示：

表 4-3 定位精度测试

	实际坐标	定位坐标(m)
第一次	(2.43m,1.37m,37°)	(2.46m,1.34m,35.3°)
第二次	(-1.96m,1.53m,184°)	(-2.01m,1.59m,189.6°)
第三次	(-2.57m,-0.79m,246°)	(-2.52m,-0.86m,237.3°)
第四次	(-0.36m,-1.04m,318°)	(-0.43m,-1.12m,309.6°)
第五次	(1.64m,-0.87m,13°)	(1.57m,-0.96m,19.7°)

结果分析：上述测量结果是由扫地机器人围绕室内转一周所测得的，五次定位结果中，定位的平均误差为 5.8cm，定位误差最大为 9cm，随着时间的推移定位精度会有所下降，但总体精度较高。

综上结果可以看出，所构建的点云地图与室内实际地图分布相比相对误差最大为 8.86%，点云地图与室内全局分布的一致性较好，定位精度平均误差为 5.8cm，在室内环境下达到了较高的定位精度。

4.3 本章小结

本章主要介绍了扫地机器人平台搭建及仿真测试。首先介绍整体硬件的设计，通过硬件总图框图介绍整个系统的功能模块，对核心电路和定位相关电路进行的讲解；然后介绍本系统软件开发平台，软件开发平台包括基于 ROS 开发和 Keil 两部

分；随后介绍基于 ROS 中 Gazebo 和 RVIZ 应用软件平台的模拟仿真，通过仿真对比得出融合定位具有较高精度；最后在室内对扫地机器人进行两次实验，验证了本设计的理论。

5 总结与展望

5.1 总结

室内定位和地图构建是室内服务机器人的关键技术之一，也是研究的一大难点。SLAM 算法对硬件计算性能要求比较高也制约其应用，本文提出了一种基于里程计和 IMU 预测模型的激光雷达 SLAM 算法，通过提高预测模型的精度来降低 SLAM 算法的更新频率，以达到降低算法对硬件计算性能的要求，来满足本系统的应用需求。

为提高室内定位与地图构建算法的精度，本文主要从以下几个部分入手：

(1) 通过对室内定位常用的方法及传感器进行分析和比较，确定了以激光雷达为主要传感器的室内定位方法，建立了 SLAM 的运动模型和观测模型，确立了以融合算法为基础的 SLAM 算法。

(2) 针对扫地机器人和激光雷达的使用环境对两种常用的 SLAM 算法进行分析，并通过 MATLAB 对两种滤波算法进行仿真对比，确定了适用范围更广精度更高的 RBPF-SLAM 算法来实现融合算法进行定位与地图构建。

(3) 针对常用的里程计预测模型精度较低的问题，提出了基于里程计、陀螺仪、加速度计、电子罗盘多种传感器融合位姿预测，显著提高运动模型的定位精度，降低对激光雷达频率的采样率的要求。最后根据室内导航面积小距离短的特点和点云存储消耗存储空间大的缺点，采用占据栅格地图对室内地图进行构建，节省存储空间，提高规划效率。

最后对融合位姿预测的 SLAM 算法进行了对比仿真，在室内条件下对扫地机器人进行两次融合 SLAM 算法实验，得到点云图的最大相对误差为 8.86%，点云分布与室内布局一致性良好。对扫地机器人进行几次定位测量实验，得到误差的平均值为 5.8cm，最大值为 9cm，定位精度满足室内导航需求。

5.2 展望未来

融合 SLAM 算法的精度较高，对硬件计算性能的要求降低，但本设计也存在诸多不足的地方。首先由于时间和个人能力问题，对整套 SLAM 算法的优化程度不够，在引入定位算法后，系统常出现故障而死机，然后是整套 SLAM 算法并不完善，长时间使用或较大空间使用时，系统闭环检测特性有缺陷，没有引入点云与地图匹配，机器人被“绑架”后无法恢复，导致定位错乱等。针对上述问题，系统做出如下改进：

（1）为提高在移植平台的兼容性和高效性，需要对程序包进行优化，去掉冗余代码，根据平台底层硬件资源进行合理资源配置，提高代码的执行效率，避免硬件资源的不合理利用。

（2）针对机器人被“绑架”后无法恢复问题，一般有两种方法可以解决。一种是引入点云地图，将激光雷达采集的点云与点云地图进行匹配，重新找回当前的位置；另一种方法是记录关键位置的激光雷达点云数据，然后让机器人对这些关键位置进行寻找和匹配找回自身位置。这两种方法各有优点和缺点，对改善系统性能很有帮助。

致 谢

时光如白驹过隙，转眼两年的研究生学习生涯即将结束，一路走来收获颇丰。彼时，在我踏进华中科技大学，开始研究生生涯时，还只像一个青苹果一样青涩不成熟，经验匮乏，只有少许比赛经验。

这里我首先要感谢我的导师雷鑑铭老师，感谢他的信任与栽培，让我多次参与项目实践，甚至委以组长重任，在他的带领下我们攻克一道又一道科研难题。我也从最初的硬件电路学习、单片机编程到学习操作系统再到学习 ROS，期间经历了很多困难与波折，但收获颇丰。同时也感谢邹老师，其严谨治学态度让我受益颇多。

其次我要感谢我们团队的成员，我们一起学习一起完成项目科研任务，在面对困难时一起面对一起加班一起讨论，很多次在我遇到困难的时候，他们一起帮助我解决困难，非常感谢他们的帮助。非常感谢他们在我论文撰写期间，对我的仿真和实验进行的帮助。也非常感谢实验室所有成员，感谢大家生活中帮助。这里也非常感谢王晓龙同学，经常在我科研遇到困难时，他总能激发我的新思路并解决问题。

最后我要感谢父母。在我成长过程中的每一道难关他们总是能鼓励和引导我一步步走下去，他们在家辛勤的耕耘土地给我机会供我读书，感谢家人对我的支持和陪伴。

参考文献

- [1] 赵旋旋,韩李涛,郑莹,类延辉,吴佳怡. 室内导航模型研究综述[J]. 软件导刊, 2016,15(05): 1-3.
- [2] 史风栋, 刘文皓, 汪鑫, 丁娟, 史屹君, 修春波.室内激光雷达导航系统设计[J]. 红外与激光工程, 2015, 44(12): 3570-3575.
- [3] Yuan L I, Guangyuan F U, Ming W U. A Navigation of an Indoor Autonomous Mobile Robot Based on Fuzzy Logic Method[J]. 2017(aiea).
- [4] 张倩雯, 孙卫卫. 浅谈室内导航技术[J]. 发展, 2015(08): 108.
- [5] 赵航, 刘玉梅, 卜春光, 李艳杰, 刘博. 扫地机器人的发展现状及展望[J]. 信息与电脑(理论版), 2016(12): 167-168.
- [6] Gunathillake A, Savkin A V, Gunathillake A, et al. Mobile robot navigation for emergency source seeking using sensor network topology maps[C]. Chinese Control Conference. 2017: 6027-6030.
- [7] Mutlu L. Control and Navigation of an Autonomous Mobile Robot with Dynamic Obstacle Detection and Adaptive Path Finding Algorithm[J]. 2012, 45(22): 355-360.
- [8] 陈卫东, 张飞. 移动机器人的同步自定位与地图创建研究进展[J]. 控制理论与应用, 2005(03): 455-460.
- [9] 王海军. 未知环境下移动机器人即时定位与地图创建[D]. 上海大学, 2009.
- [10] 郑宏. 移动机器人导航和 SLAM 系统研究[D]. 上海交通大学, 2007.
- [11] Wang C H, Department C. The SLAM Algorithm of Mobile Robots Based on Gaussian Mixture Cubature Kalman Filter[J]. Journal of Inner Mongolia Normal University, 2017.
- [12] 郭毓, 王斌明, 吴益飞, 等. 两级传感器信息融合的移动机器人避障研究[J]. 传感器与微系统, 2008, 27(2): 61-64.
- [13] 武奕晶. 室内智能吸尘机器人全区域覆盖算法和自主充电算法的研究[D]. 上

- 海大学, 2007.
- [14] Jia Shing Sheu, Yi Shiang Wang, Tsong Liang Huang. Implementation of Indoor Wireless Positioning System in an Omni-Wheel Robot Based on Gyroscope[J]. Advanced Materials Research, 2012, 1674(482).
- [15] 孙晓雪, 赵玉山. 扫地机器人的发展现状和趋势研究[J]. 科技资讯, 2017, 15(28): 238-239.
- [16] Neato 新款扫地机器人推出 支持 RPS 绘图定位[J]. 工业设计, 2013(12): 5.
- [17] 赵航, 刘玉梅, 卜春光, 李艳杰, 刘博. 扫地机器人的发展现状及展望[J]. 信息与电脑(理论版), 2016(12): 167-168.
- [18] 高强. 基于激光测距仪的移动机器人同时定位与地图构建研究[D]. 大连海事大学, 2017.
- [19] Agostino Martinelli, Anna Svensson, Nicola Tomatis, Roland Siegwart. Slam based on quantities invariant of the robot's configuration[J]. IFAC Proceedings Volumes, 2004, 37(8).
- [20] Xun Fang, Xiaoyang Fu, Ming Sun. The Improved Locating Algorithm of Particle Filter Based on ROS Robot[J]. IOP Conference Series: Materials Science and Engineering, 2018, 322(5).
- [21] 房芳, 马旭东, 戴先中. 一种基于混合模型的移动机器人 SLAM 方法[C]// 中国智能化会议. 2009.
- [22] TANG, Xian-lun, La-mei, et al. Mobile robot SLAM method based on multi-agent particle swarm optimized particle filter[J]. Journal of China Universities of Posts & Telecommunications, 2014, 21(6): 78-86.
- [23] 李昀泽. 基于激光雷达的室内机器人 SLAM 研究[D]. 华南理工大学, 2016.
- [24] Celik K, Chung S J, Clausman M, et al. Monocular vision SLAM for indoor aerial vehicles[C]. Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE Press, 2009:1566-1573.
- [25] Agostino Martinelli, Viet Nguyen, Nicola Tomatis, Roland Siegwart. A relative map

- approach to SLAM based on shift and rotation invariants[J]. Robotics and Autonomous Systems, 2006,55(1).
- [26] 王消为, 贺利乐, 赵涛. 基于激光雷达与双目视觉的移动机器人 SLAM 研究[J]. 传感技术学报, 2018(3).
- [27] Lim H, Choi J Y, Kwon Y S, et al. SLAM in indoor pipelines with 15mm diameter[C]. IEEE International Conference on Robotics and Automation. IEEE, 2008: 4005-4011.
- [28] TANG, Xian-lun, La-mei, et al. Mobile robot SLAM method based on multi-agent particle swarm optimized particle filter[J]. Journal of China Universities of Posts & Telecommunications, 2014, 21(6): 78-86.
- [29] 徐朝文. 拆除机器人即时定位与地图构建算法研究[D]. 安徽工业大学, 2017.
- [30] CHENPeng, LIJu-hu, NIUKai, et al. Particle filter based automatic frequency control scheme by combining the two-step structure[J]. The Journal of China Universities of Posts and Telecommunications, 2012, 19(2): 9-14.
- [31] Chang H H, Lin S Y, Chen Y C. SLAM for Indoor Environment Using Stereo Vision[C]. Intelligent Systems. IEEE, 2010: 266-269.
- [32] Guerra E, Bolea Y, Grau A, et al. New approach on bearing-only SLAM for indoor environments[J]. 2011, 47(10): 1-4.
- [33] 李洋, 余先涛, 贺勇,等. 基于 Cortex-A53 平台的激光雷达 SLAM 实现[J]. 电子技术应用, 2018(2):105-108.
- [34] Lin R, Wang Y, Yang S. RGBD SLAM for Indoor Environment[M] Foundations and Practical Applications of Cognitive Systems and Information Processing. Springer Berlin Heidelberg, 2014: 161-175.
- [35] 满增光. 基于激光雷达的室内 AGV 地图创建与定位方法研究[D]. 南京航空航天大学, 2014.
- [36] Carlone L, Du J, Ng M K, et al. Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence[J]. Journal of Intelligent & Robotic Systems, 2014, 75(2): 291-311.

- [37] 武二永, 项志宇, 沈敏一, 等. 大规模环境下基于激光雷达的机器人 SLAM 算法[J]. 浙江大学学报(工学版), 2007, 41(12): 1982-1986.
- [38] Gage D W. On-board SLAM for indoor UAV using a laser range finder[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2010, 7692: 769213-769213-11.
- [39] 王依人, 邓国庆, 刘勇, 等. 基于激光雷达传感器的 RBPF-SLAM 系统优化设计[J]. 传感器与微系统, 2017, 36(9):77-80.
- [40] 陈玉. 基于 SLAM 的扫地机器人控制系统研究[D]. 哈尔滨工业大学, 2017.