

电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS



论文题目 基于激光雷达的室内服务机器人
 导航系统的设计与实现

学 科 专 业 计算机应用技术

学 号 201521060511

作者姓名 陈莹

指导教师 葛树志 教授

分类号 _____ 密级 _____

UDC ^{注 1} _____

学 位 论 文

基于激光雷达的室内服务机器人导航系统 的设计与实现

(题名和副题名)

陈 莹

(作者姓名)

指导教师

葛树志

教 授

电子科技大学

成 都

(姓名、职称、单位名称)

申请学位级别 **硕士**

学科专业 **计算机应用技术**

提交论文日期 **2018.03.22**

论文答辩日期 **2018.05.18**

学位授予单位和日期

电子科技大学

2018 年 06 月

答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类

Design and Implementation of a Lidar-based Indoor Service Robot Navigation System

**A Master Thesis Submitted to
University of Electronic Science and Technology of China**

Discipline: Computer Application Technology

Author: Ying Chen

Supervisor: Prof. Shuzhi Sam Ge

School: School of Computer Science & Engineering

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名： 陈莹 日期： 2018 年 6 月 15 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名： 陈莹 导师签名： 葛树志

日期： 2018 年 6 月 15 日

摘要

本论文采用电子科技大学财务机器人 FinaRobot 作为研发平台，FinaRobot 以拟人的人机交互方式协助财务处员工为师生提供财务问题的在线咨询，为保证其自主服务不受到电量过低导致自动关机的影响，本文为它设计了一套自动充电系统，完成远程导航和近程充电桩对接两部分工作，实现自动充电。另外，针对用户的地点引导需求，设计了一套融合语音交互的智能导航系统，在该系统中机器人需要使用语音与用户交流，同时能够规划一条合理的导航路径，并且实现避障功能。因此本文将重点围绕移动机器人建图、定位和导航技术开展研究和实验。

本文所完成的详细工作和贡献说明如下：

（一）针对本文对导航系统的功能需求，对财务机器人 FinaRobot 的总体架构（包含软件系统和硬件系统）进行构思和设计，其中最主要的软件系统主要包括三大模块：环境地图创建、语音导航和自动充电，同时选用 Socket 作为两大导航系统中主要的网络通信方式。

（二）介绍了 SLAM 和激光雷达的基本原理。针对传统 RBPF-SLAM 存在的计算复杂度高以及粒子退化问题。提出了一种改进的 RBPF-SLAM 算法，通过改进的粒子重采样策略，降低粒子退化的速度，减少粒子个数，进一步提高建图精度和效率。基于上述改进算法，进行代码编写，选取两个不同的环境验证该算法的优化效果。最后对如何在 ROS 下面创建环境地图的工作流程进行了详细介绍。

（三）提出了一种融合语音交互和自然语言处理的导航方式，在获取用户的语音导航需求后，通过语音识别技术转化为文本信息，接着通过自然语言处理将标注目的地点的环境地图反馈给用户，最后进入导航模式。将用户引导到目的地后，通过语音合成技术给予用户反馈信息。最终完成了基于语音交互的智能导航系统的设计与实现。

（四）提出了一种基于突变点检测的充电桩对准算法，可以仅靠现有的激光雷达传感器实现较高精度的充电桩对准。同时设计了一套较高误差容忍度的充电头和充电座，充分提高机器人自动充电的精准对接的成功率。最后完成了自动充电系统的设计与实现。

（五）最后，在已有的硬件平台上搭建和整合导航系统软件模块，针对完整的机器人导航系统做了一系列测试实验用于验证各个软件模块的合理性和有效性。

关键词：移动机器人，粒子滤波，RBPF-SLAM，自然语言处理，突变点检测

ABSTRACT

This thesis uses FinaRobot, the financial robot of the University of Electronic Science and Technology of China, as an R&D platform. FinaRobot assists Finance Department staff to provide teachers and students with online consultation on financial issues through anthropomorphic human-computer interaction to ensure that their autonomous service is not affected by the automatic shutdown caused by low battery. This article has designed a set of automatic charging system for it to complete the remote navigation and short-range charge docking two parts of the work to achieve automatic charging. In addition, according to the user's location guidance requirements, a set of intelligent navigation system that integrates voice interaction is designed. In this system, the robot needs to use voice to communicate with users, and at the same time, it can plan a reasonable navigation path and achieve obstacle avoidance. Therefore, this article will focus on research and experimentation on the construction, positioning and navigation of mobile robots.

The detailed work and contributions completed in this thesis are described as follows:

(1) Concerning the functional requirements of the navigation system in this thesis, the overall architecture (including the software system and the hardware system) of the financial robot FinaRobot is conceived and designed. The most important software system mainly includes three modules: Environment map creation, voice navigation With automatic charging, Socket communication is selected as the network communication method in the two navigation systems.

(2) Introduced the basic principles of SLAM and Lidar. The traditional RBPF-SLAM has the problems of high computational complexity and particle degradation. An improved RBPF-SLAM algorithm is proposed, which uses an improved particle resampling strategy to reduce the speed of particle degradation, maintain particle diversity, reduce the number of particles, and further reduce computational complexity. Based on the above improved algorithm, code writing, and finally select two different environments to verify the optimization effect of the algorithm. Finally, the workflow of how to create an environment map under ROS is described in detail.

(3) It proposes a navigation method that integrates speech interaction and natural language processing. After obtaining the user's speech navigation requirements, it is

converted into textual information through speech recognition technology, and then the environmental map of the annotation destination point is fed back to the natural language processing. The user finally enters the navigation mode. After guiding the user to the destination, feedback information is given to the user through speech synthesis technology. Finally, the design and implementation of a smart navigation system based on voice interaction was completed.

(4) A charging pile alignment algorithm based on mutation point detection was proposed. It can rely on existing lidar sensors to achieve higher accuracy of charging pile alignment. At the same time, a set of charging heads and charging bases with a relatively high tolerance for error was designed. Fully increase the success rate of accurate docking for automatic robot charging. At the same time, the design and implementation of an automatic charging system have been completed.

(5) Finally, build and integrate the navigation system software module on the existing hardware platform, and made a series of test experiments for the complete robot navigation system to verify the correctness and effectiveness of each software module.

Keywords: Mobile robot, particle filter, RBPF-SLAM, natural language processing, mutation point detection

目录

第一章 绪论	1
1.1 研究背景	1
1.2 室内服务机器人研究概况	2
1.3 机器人导航技术的研究概况	4
1.4 课题研究意义	5
1.5 论文主要研究内容与结构安排	5
1.5.1 论文研究内容及创新点	5
1.5.2 论文结构安排	6
第二章 移动机器人导航系统总体设计	8
2.1 移动机器人导航系统总体架构	8
2.2 移动机器人硬件系统结构	10
2.3 移动机器人导航系统业务流程	12
2.4 系统通信模块	13
2.5 移动机器人导航系统的软硬件开发环境	14
2.6 本章小结	15
第三章 移动机器人定位与建图	16
3.1 SLAM 简介	16
3.1.1 SLAM 系统结构	17
3.1.2 SLAM 的概率模型	18
3.2 激光雷达模型	19
3.2.1 激光雷达概述	19
3.2.2 激光雷达的原理及坐标转化	20
3.3 基于 Rao-Blackwellized 的 FastSLAM 算法	21
3.3.1 传统的 RBPF-SLAM 算法	21
3.3.2 改进重采样策略的 RBPF-SLAM	23
3.3.3 传统 RBPF-SLAM 和改进的 RBPF-SLAM 实验对比	25
3.4 环境地图创建系统结构	29
3.4.1 消息传递机制	30
3.4.2 Gmapping 建图	31
3.5 本章小结	33

第四章 基于语音交互的移动机器人智能导航系统	34
4.1 语音交互导航系统关键技术	34
4.1.1 自然语言处理	34
4.1.2 基于 ROS 的移动机器人导航技术	35
4.2 语音交互导航系统总体设计	41
4.2.1 在线问答系统	42
4.2.2 基于 ROS 的导航系统	44
4.3 本章小结	47
第五章 基于突变点检测的自动充电系统	48
5.1 自动充电关键技术	48
5.2 基于突变点检测的充电桩识别算法	49
5.3 自动充电系统整体设计	51
5.3.1 电量检测和反馈	51
5.3.2 自动充电远程导航	52
5.3.3 充电桩识别和对接	52
5.4 本章小结	55
第六章 移动机器人导航系统实验与测试	56
6.1 机器人环境地图创建实验与测试	56
6.2 机器人语音交互导航实验与测试	62
6.3 机器人自动充电技术实验与测试	63
6.4 本章小结	65
第七章 全文总结与展望	66
7.1 全文工作总结	66
7.2 后期工作展望	67
致 谢	68
参考文献	69

第一章 绪论

1.1 研究背景

现如今科学技术迅猛发展，关于机器人的探索至今为止也已经大约有 60 年的历史了。最开始关于机器人的研发方向为探索并研制一款机器，该机器的作用是用来取代目前采用人工方式所进行的反复性高的劳力活。到目前为止，许多发达国家的机器人研发水准已然处于较为成熟的平稳期，所研制的机器人在功能和发挥的作用上呈现出了较高的平稳性，并且随着技术的越来越纯熟，机器人研制所需要的成本也逐渐减低，因此大批量生产机器人逐渐成为现实并且已经应用至各类行业里^[1]。特别要指出的是日本发那科公司，其充分发挥了机器人的优势，采用工业机器人来制造其他领域所需要的机器人，很大程度上使得机器人的制造变得高效并且也有利于机器人的推广使用。

随着工业机器人的推广使用，目前许多原本依靠人工来进行的繁杂劳力活都已经被工业机器人所承担，因此关于机器人的研发方向已经慢慢从工业上转向平时生活相关的领域，比如服务方面。如果说原本的工业机器人都是在进行着傻瓜操作，无需多余的思考，那么服务机器人就相对需要智能许多，因为其所服务的对象是人类，需要满足人类各式各样的生活需求，无论从种类还是复杂度上考虑都与之前的工业机器人不在同一个级别的，同时还需要考虑人与人之间的差异性，单一操作的机器人完全无法满足服务要求，并且应用场景的多样性使得对机器人功能要求的增加。

尽管在服务机器人的研发上需要面临着许多困境，但是由于人类需求的日益增长与高要求，研发服务机器人迫在眉睫^[2]。特别是由于生活和医护水平的提高，人类越来越长寿，这也就意味着老龄化问题愈发显著。这种现象的产生导致整个社会各方面的生产力大大下降，同时大大增加了负担，因此需要一些较为智能的机器人来帮助人类的日常生活，减轻人类生活的负担。并且随着科学技术的快速发展，人们的生活品质得到了明显的改善，但是在有利的同时也造成了许多难以避免的问题，比如每天都在发生的交通事故，这也就导致了伤残人士大大增加，也就意味着增添了家人的负担。同时，人类未必能够完全掌握日益发展的科技，有许多方面实际上不适合人类去操作的，所以在关注人身安全的角度上有必要采用服务机器人来取代人工劳作。除此之外，随着人类生活质量的不断提升，人类所处环境的舒适程度和自我精神满足便得到更多的关注，而日常生活的繁杂以及工作中的压力严重阻碍了更高品质生活的实现^[3]。

关于上述所提到的一些现象以及目前存在的问题和需求，许多发达国家已经

开始着手考虑缓解这类状况，同时逐渐把研制服务机器人置于长期发展战略目标上。考虑到我国的独特状况，人口数量巨大并且人口老龄化也日渐严重，同时人们日益希望提高生活品质，因此非常有必要开展服务机器人的探究与研制工作，将其列入未来的推动社会发展的工作战略上^[4]。关于这一点，也明显体现在了国家“十三五”发展规划战略中，在该规划里直接表明需要将研制各类机器人的工作摆在发展工作的重要位置，并且要将其推广至各类的不同应用领域中。这也就意味着有了政府的支持，服务机器人将更加具有非常可观的发展远景。

1.2 室内服务机器人研究概况

室内机器人属于服务机器人的一个分支。在与其相关的探索科研方面国外要早先于国内，并且这些研究无论在理论还是实际研制实物中都呈现出优异的结果。早在 2010 年 Siddharthas.Srinivasa 等^[5]一些研究学者就研制出一款室内服务机器人，并将其命名为 Herb（如下图 1-1（a））。该室内服务机器人的作用是结合视觉系统来实现精确的定位与导航，能够完成对所处环境中具体物品的精准识别，同时安装了智能自动化机械手来实行规定好的动作。Thomas Breuer 等^[6]一些科研人员立足于目前已有的研究成果，把实际环境物体的识别技术应用于 SLAM 创建语义地图，该应用由一款名为 Johnny（图 1-1（b））的室内服务机器人来完成的。其能够区别不同的操作目标，在已有的地图上实现物体的精准定位并进行合理的指引与导航，这样可以很大程度上根据已有的语义上的信息来更好的顺应复杂的周遭环境。StefanSchiffer 等人^[7]为了给老人的日常生活提供便利，研制了一款室内服务机器人并将其命名为 Casar（图 1-1（c）），该机器人能够依靠激光传感器来同时实现定位、建立地图和导航技术。在设计中首先采用 A*算法来完成全局范围内的最优路径的选择，接着对局部范围内的躲避障碍物进行了处理，同时还可以识别手势的编号并进行语音上的交流，大致上能够达到导航的需求。

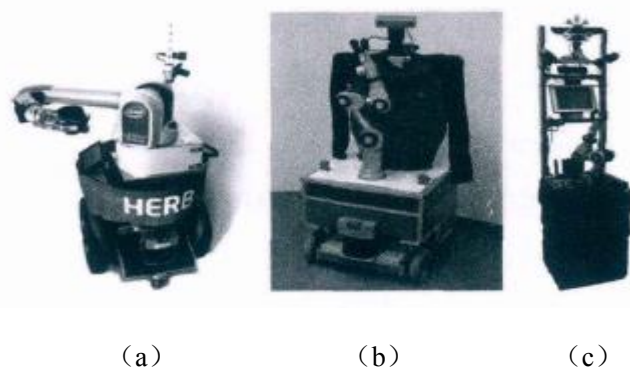


图 1-1 国外学术室内机器人 (a) Herb; (b) Johnny; (c) Carsar

除此之外，外国不少的高新企业在服务机器人的探索和研制上也有着许多的功劳。比如日本本田公司在 2011 年第一次向大众介绍了其研发的双足室内服务机器人，并将其命名为 Asimo，该机器人可以综合利用超声波以及视觉系统的传感器来自动研究人类行走的路径特点并实现智能自动选择局部范围的路径方向；日本软银集团以及法国 Aldebaran Robotics 在 2014 年向大众展现了世界上第一款能够拥有和人类情绪相似的外形酷似人类的服务机器人，其名字为 Pepper(图 1-2(b))。该机器人的设计思路是采用多个传感器来捕捉人类多变的表情，同时结合采用智能的语音识别系统来进行人类语言音调以及一些较为剧烈的感叹词的识别与分类；美国 iRobot 公司在 2015 年向大众介绍了一种较为新颖的智能服务机器人，其名字为 Roomba980。该机器人的主要任务就是代替传统的人工扫地实现智能扫地，减轻人类的日常家务负担，在其相关的设计中首次采用了以单目视觉系统为基础的 SLAM，该方法的引入使得相关机器人的设计不再需要价格高昂的激光传感器。美国波士顿动力公司在 2016 年向大众展现了一款新颖的服务机器人 Spotmini，外形酷似一条狗，主要用于进行一些机械操作，所以也叫机械狗。该机器人是采用立体视觉系统来实现物体的识别，由惯性系统来维持整体的稳定，同时结合一些传感器来实现洗完以及一些其他较为简易操作的家务事。

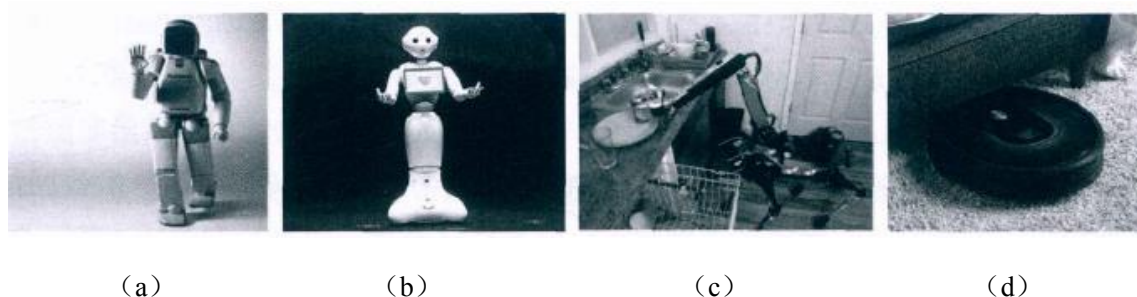


图 1-2 国外企业室内机器人 (a) Asimo; (b) Pepper; (c) Spotmini; (d) Roomba980

与外国的研究状况进行对比，我国关于室内服务机器人的探索和研制开始的比较迟一些，不过随着科技的进步以及科研人员的苦心钻研目前也拥有的一些较为不错的成就。比如清华大学的相关科学学着开发了一款能够行动的护理机器人，该机器人的服务对象主要为因为意外等情况而高位截瘫的病人，能够帮助他们实现拿药饮水等一些较为简单的日常生活需要，而不需要他人的帮助；中科大所研制的一款名为“可佳”的服务机器人(图 1-3 (a))，其外形上与正常人类无疑，可以实现自动定位与导航，同时其拥有合理的逻辑思考能力，可以进行有条理的交流；哈工大开发出一款室内服务机器人(图 1-3 (b))，该机器人能够运用到生活中的一些家务或者娱乐相关的场景中，例如打水倒水、舞蹈^[8]。上交的相关研究

人员开发了一款室内服务机器人(图 1-3 (c)),其主要运用场景为食堂、餐厅等,实际设计的原理是采用双目视觉系统来实现某一物体的精准定位,并且完成对应所需要的拿取送等动作^[9]。南大的研究人员发明了一款用于对图书馆各类书籍进行盘查清点的服务机器人(图 1-3 (d)),该机器人采用 RFID 并结合 SLAM 等方法来完成书籍的智能盘查与清点^[10]。

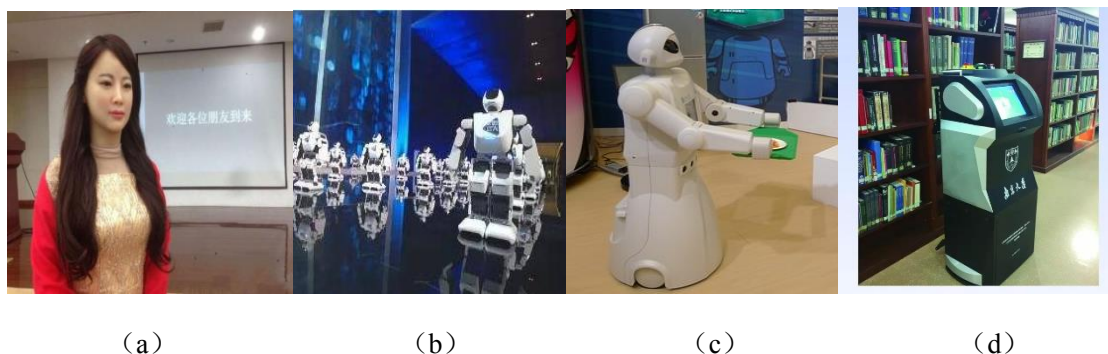


图 1-3 国内室内机器人 (a) 可佳; (b) 跳舞机器人; (c) 餐厅机器人; (d) 图书馆机器人

1.3 机器人导航技术的研究概况

现如今室内服务机器人的发展已处于较为成熟并且兴盛的阶段,不过科技的发展是没有停歇的,在研发服务机器人的过程中需要考虑生产率是否高效以及所生产的服务机器人是否能够给人类带来便捷,这也就意味着有必要着力于高质量地提高所研发的室内服务机器人的自主能力以及实时顺应周遭环境的技能。特别是自主移动机器人,如果其要实现一项工作那么实现的基础即为可以处于多种繁杂环境中实现自主定位并且对目标实行导航。

导航技术作为移动机器人的研究核心,其概念上讲是指移动机器人通过测量传感器感知环境信息和自身位姿,实现从一个位姿到另一个位姿的自主移动,同时在移动过程中合理规避障碍物。导航主要解决三方面的问题:

(1) 采用某种有效的检测方式来获得机器人所处坐标点的相关信息(包括方向,位姿等)和周遭环境情况;

(2) 设计合适的算法来分析处理(1)中得到的相关数据以及信息同时建立相对应的数学模型;

(3) 求解环境中的最优路径解,如果没有最优解则寻求逼近最优的解,并且该解需要满足机器人在运行过程中不会发生碰撞或者冲突。

移动机器人实现定位以及导航所基于的方法并非单一,例如基于惯性^[11,12]、基于视觉^[13-16]、基于传感器^[17-20],基于卫星^[21-23]等一些其他方法。这些各式各样的方法有着自己独特的特性,因此其所应用的场景也是有所差别的,例如应用于室

内或者室外、应用环境是否结构化等。

至今为止，在移动机器人导航方向的探索和科研上的投入，科研人员已经收获了不菲的成效，不过仍旧存在不少的理论相关的重要细节以及技术瓶颈需要后续科研人员进行探索和改善。

1.4 课题研究意义

关于移动机器人的相关研发里，导航技术在相关科研中为一个非常关键的技术点。这也就意味着可以将导航技术视为研制移动机器人相关技术的重中之重，从原理上分析导航技术也是使得移动机器人智能化以及达到自主运动的最为重要的一个环节。

在移动机器人协助人类进行导航和作业的相关工作中，机器人在完成精确的导航任务之外，还有一个非常值得考虑的问题，那就是如何提高该导航过程的智能化和交互感，操作简便性，提交过程的趣味度。例如，用户只需使用自然语言与机器人进行沟通，表明目的地，机器人便会引导用户前往，并且在到达目的地后使用语音反馈。

除此之外，目前的移动智能机器人通常情况下采用机载蓄电池来作为运行所需用电的供给者，该方法存在一定的局限性，因为机载蓄电池所持续供电用于移动的时间较短，一般只有几个小时，因此务必要处理持续供电的问题，研发能够实现自动充电功能的机器人。

为了让机器人能够更好地服务于人类，需要在现有移动机器人在导航技术的基础上，向着智能化和拟人化的方向发展。因此本文可以具有十分深远的研究意义和价值。

1.5 论文主要研究内容与结构安排

1.5.1 论文研究内容及创新点

本文旨在开发一台具有自主定位与导航功能，能完成语音交互智能引导和低电量状态下自动充电的财务咨询机器人。作为一个完整的财务咨询机器人生态系统，FinaRobot 主要包含环境地图创建、语音导航和自动充电三大重要模块。

本文完成了基于激光雷达的室内移动服务机器人导航系统的设计，在机器人总计结构和业务设计的基础上，针对每个模块都进行了详细的设计。主要涉及的技术有：基于 SLAM 的定位与环境地图创建、基于自然语言处理的问答系统、导航中的全局路径规划和局部路径规划技术以及充电桩识别技术。财务机器人

FinaRobot 通过激光雷达和里程计数据进行地图创建和自定位，同时融合其他信息完成智能语音导航和自动充电。基于以上技术，提出如下创新点：

(1) 在环境地图创建模块中，基于传统 RBPF-SLAM 计算复杂度高以及粒子退化的问题，提出一种改进重采样策略的 RBPF-SLAM 算法，通过该改进的算法，减少重采样运算时间，降低粒子退化的速度，保持粒子多样性，从而减少粒子个数，提高建图的精度和效率。

(2) 在智能导航系统中，提出一种融合语音交互和自然语言处理的导航方式，在获取用户的语音导航需求后，通过语音识别技术转化为文本信息，接着通过自然语言处理将标注目的地点的环境地图反馈给用户，最后进入导航模式。将用户引导到目的地后，通过语音合成技术将反馈信息合成自然语言进行反馈。

(3) 在自动充电系统中，提出一种基于突变点检测的充电桩对准算法，可以仅靠现有的激光雷达传感器实现较高精度的充电桩对准。同时设计一套对误差有较高容忍度的充电头和充电桩。充分提高机器人自动充电的精准对接的成功率。

1.5.2 论文结构安排

本文一共分为七章，其章节安排如下：

第一章：绪论部分，本章主要介绍当前移动机器人的研究背景，对室内移动机器人国内外的研究状况进行对比和阐述，同时针对机器人研究领域非常重要的导航技术阐述其研究现状和主要研究内容。

第二章：移动机器人导航系统总体设计。主要针对本文对导航系统的功能需求，对财务机器人 FinaRobot 的总体架构（包含软件系统和硬件系统）进行构思和设计，其中最主要的软件系统主要包括三大模块：环境地图创建、语音导航和自动充电

第三章：移动机器人地图创建和定位算法，本章主要对 SLAM 和激光雷达的原理进行简单介绍。基于传统 RBPF-SLAM 计算复杂度高以及粒子退化的问题，提出一种改进重采样策略的 RBPF-SLAM 算法，通过改进的粒子重采样策略，解决上述两个问题。基于改进算法，进行代码编写，最终选取两个不同的环境验证该算法的优化效果。最后对如何在 ROS 系统下面创建环境地图的工作流程进行详细介绍。

第四章：基于语音交互的机器人智能导航系统的设计与实现，本章将设计一套基于语音交互的智能导航系统，系统采用语音识别技术将输入查询地点问题转化为文本，通过自然语言处理和余弦相似度匹配最终在答案库中匹配到该地点的平面地图，接着便开启导航系统，将用户引导到目的地点，同时客户端通过语音合

成技术向用户进行反馈。

第五章：自动充电系统的设计与实现，本章将提出一种基于突变点检测的充电桩对准算法，同时设计一套较高误差容忍度的充电头和充电座。基于已有的远程导航系统设计一套能实现精准对接的自动充电系统。

第六章：移动机器人导航系统实验与测试，本章主要在已有的硬件平台上搭建和整合导航系统软件模块，将针对本文设计的机器人导航系统做一系列测试和实验，验证各个软件模块的合理性和有效性。

第七章：全文总结与展望，最后对论文的工作内容进行总结，对于目前还未解决的若干问题提出后期研究方向以及展望。

第二章 移动机器人导航系统总体设计

现如今室内服务机器人的发展已处于较为成熟并且兴盛的阶段，不过科技的发展是没有停歇的，在研发服务机器人的过程中需要考虑生产率是否高效以及所生产的服务机器人是否能够给人类带来便捷，这也就意味着有必要着力于高质量地提高所研发的室内服务机器人的自主能力以及实时顺应周遭环境的技能。特别是自主移动机器人，如果其要实现一项工作那么实现的基础即为可以处于多种繁杂环境中实现自主定位并且对目标实行导航。

从前面的分析中能够得出本课题所研究的导航系统的三个非常核心的技术：定位、建图以及导航。在这三者中，可以把定位以及建图看作一体，现实操作中，周围环境以及机器人所处的位置是无法预知的，同时建图与定位之间的关系是互相契合并且依附于彼此，由此产生了移动机器人同时建图与定位技术（SLAM, Simultaneous Localization and Mapping）。机器人导航技术主要包含定位和路径规划两部分，路径规划包含全局路径规划和局部路径规划，在此基础上融合语音交互，使得整个导航过程为用户提供更多的便利和趣味性。

本论文采用电子科技大学财务机器人 FinaRobot 作为研发平台，FinaRobot 以拟人的人机交互方式协助财务处员工为师生提供财务问题的在线咨询，咨询内容包括财务办公室地点的查询和引导，针对该功能设计了一套融合语音的自动导航和引导系统。另外，为保证其自主服务不受到电量过低导致自动关机的影响，本文为它设计了一套自动充电系统。在该系统中机器人需要感知自身位姿，明确充电桩所在位姿，同时能够规划一条合理的导航路径，并且实现避障功能，最后采用基于突变点检测的充电桩对接技术实现精准对接。另外，本论文还设计了客户端的财务在线咨询系统和人机交互系统。

2.1 移动机器人导航系统总体架构

本课题实验平台为财务处引导机器人 FinaRobot，其中重点研究方向为机器人环境地图创建、融合语音的移动机器人智能导航系统和基于突变点检测的自动充电系统，除此之外，本课题还包含财务在线问答系统、员工人脸签到、机器人运动控制等功能，但是这些功能不作为本课题的重点研究方向，所以不在本章中做介绍。移动机器人系统的总体框架结构如图 2-1 所示。

根据引导机器人的功能与执行逻辑，将财务机器人软件分为三层，分别是接入层，算法层与应用层。下面简要介绍各层次及之间的关系：

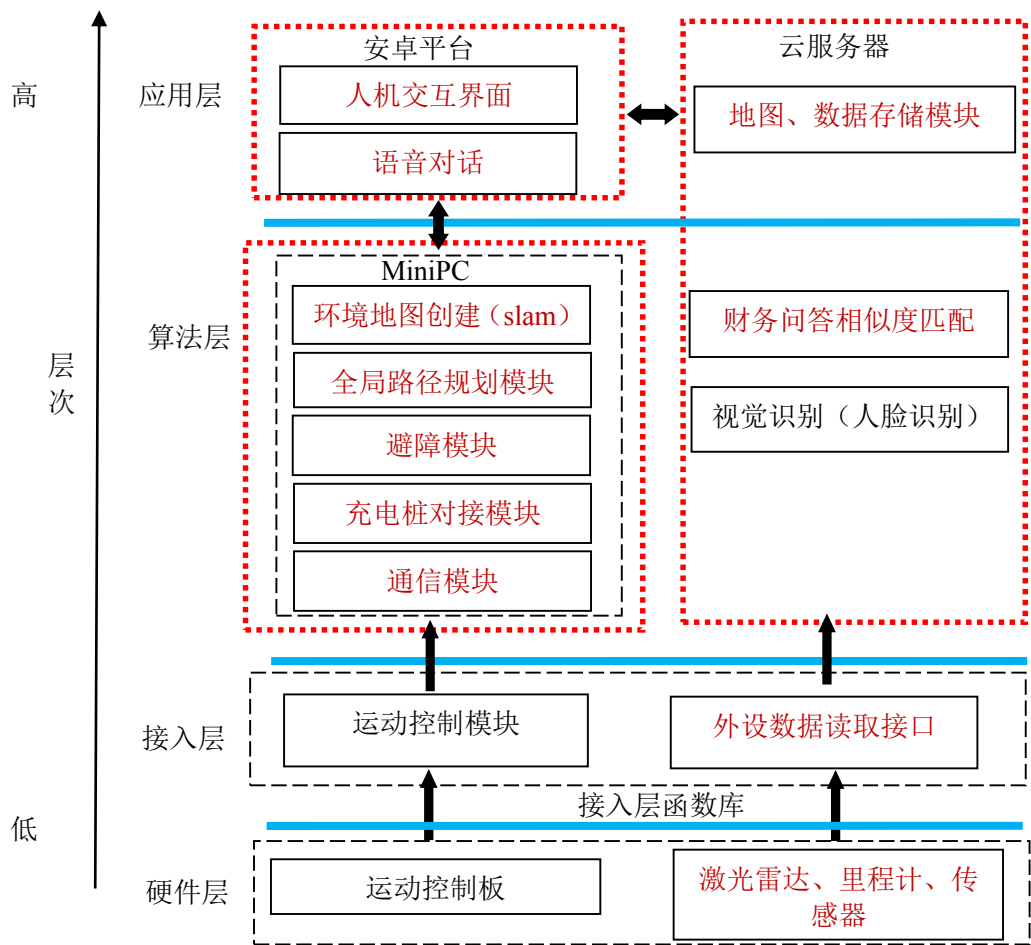


图 2-1 室内移动机器人系统总体框架图

接入层：接入层的功能是提供机器人底层硬件信息，包括时间戳对应的激光雷达、里程计和传感器数据。接入层包含两个模块，即运动控制模块与外设数据读取模块，其中，外设数据读取模块的输出将被算法层调用，而运动控制模块将间接被运行于安卓板的逻辑层调用，包括语音导航、机器人跳舞等。

算法层：算法层主要提供环境地图创建（SLAM 算法）、路径规划算法、基于突变点检测的自动充电算法和图像处理算法。图像处理算法主要用于财务处员工的脸部签到；SLAM 算法和路径规划算法主要用于机器人的导航；充电桩对接算法用于自动充电；财务问答相似度匹配算法主要应用于财务在线问答系统。另外，还包含一个用于实现与客户端和底层硬件通信的通信模块。

应用层：主要包含人机交互界面的展示、语音识别与合成、云服务器端的数据查询与存储和传感器数据的展示。

另外，根据平台也可以划分成三部分：硬件平台、嵌入式计算平台和应用客户端，详细模块内容见表 2-1。

硬件平台：机器人运动底盘、激光雷达、编码器、充电感应头、电池传感器等。

嵌入式计算平台：机器人定位与建图模块、机器人路径规划模块、机器人充电桩识别模块、机器人电量反馈模块、通信模块等。

应用客户端：语音导航和人机交互模块。

表 2-1 系统模块功能清单

平台	模块说明	
硬件平台	机器人运动底盘	完成整个机器人平台的运动控制功能
	激光雷达	采集激光扫描数据，用于建图和导航。
	编码器	采集里程计数据，用于建图和导航。
	充电感应头	结合软件完成充电功能。
	电池电压传感器	识别电量。
嵌入式计算平台	定位与建图模块	利用 SLAM 进行定位和建图。
	路径规划模块	在已有地图的基础上实现全局路径规划和局部避障。
	充电桩对接模块	在小范围内精确定位充电桩，并实现精准对接。
	通信模块	1、实现与硬件平台的通信，通信数据包含传感器信息等。 2、实现与客户端的通信，通信内容包括传感器信息、地点信息等。
应用客户端	语音导航	用户语音设定目的地，导航过程中机器人语音播报。
	人机交互模块	人机交互界面展示、人脸签到、传感器数据展示等。

2.2 移动机器人硬件系统结构

移动机器人硬件系统在整个机器人生态中扮演着举足轻重的角色，如果没有硬件系统的支撑，所有的软件部分都无法实施，但是硬件部分不属于本论文的主要研究对象，所以本节只对硬件的基本组成部分和对应功能做相应的阐述。硬件部分主要包括机器人运动底盘、MiniPC、激光雷达、编码器等，具体内容见上表。

本文研究所搭建的硬件部分为电子科技大学财务机器人（FinaRobot），其具体外观如下图 2-2（a）所示。在财务机器人的相关设计中，其底盘的车轮采用的是

全方向的车轮，在底座上安装了三个这种车轮^[32]，具体外观如下图 2-2 (b) 所示。所谓全方向车轮即为车轮的边缘均为各种小圆柱，并且这些小圆柱能够在车轮表面的垂直方向上转动，因此这样生成的自由度可以驱使车轮沿着车轮边缘的小圆柱转动的方向运动。底座的外形结构表现为一个圆盘，该圆盘的直径大致是 30cm，同时上述提到的三个全方向的车轮以 120° 的间距安装于底座上，车轮的半径大致是 8cm。采用固定的电机来驱使底座上的车轮转动，并且车轮在互相协作的情况下，根据速度合成理论，可以完成该移动机器人朝任何一个方向的稳定运动。

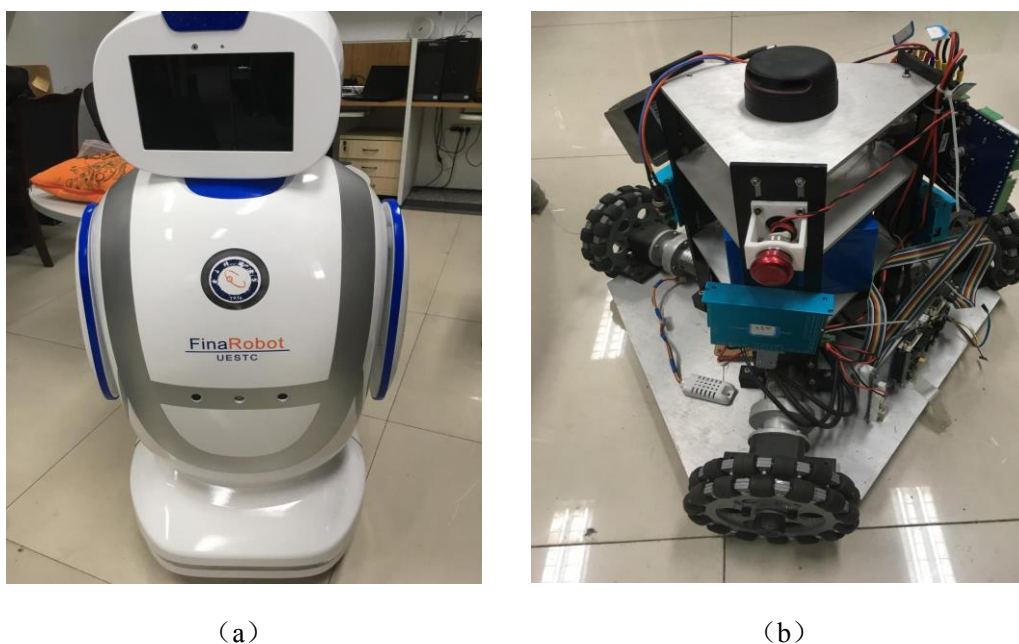


图 2-2 FinaRobot 及底盘外观图 (a) FinaRobot 外观; (b) FinaRobot 底盘外观

电子科技大学财务机器人 (FinaRobot) 在运动控制方面使用的是分布式控制构造，该构造的具体构想根据图 2-3 来实现，整体的设计可以分成上中下三部分。其中最上层的操控部分结合其运算中心以及交互平台来构成，运算中心的具体实现为 MiniPC，交互平台的具体实现为安卓平板；中间部分的操控主要结合底座运动的操作机械以及实现躯干运动的操作机械来构成；最下层部分的操控主要结合驱使车轮转动的电机和传感器部分来构成。

依照之前对于系统构造的简单介绍，运动操控部分的各个模块的操控功能的设计按照下列来实现：

(1) MiniPC 主要实现各种繁杂的算法运行和设置运动操作指令。MiniPC 能够联合激光雷达以及底座上的传感器以及电机相关的信息，根据所设计好的合理的算法来实现行动操作指令的计算，同时把操作指令的相关数据采用 Socket 协议送达到底座的操作控制机械上；

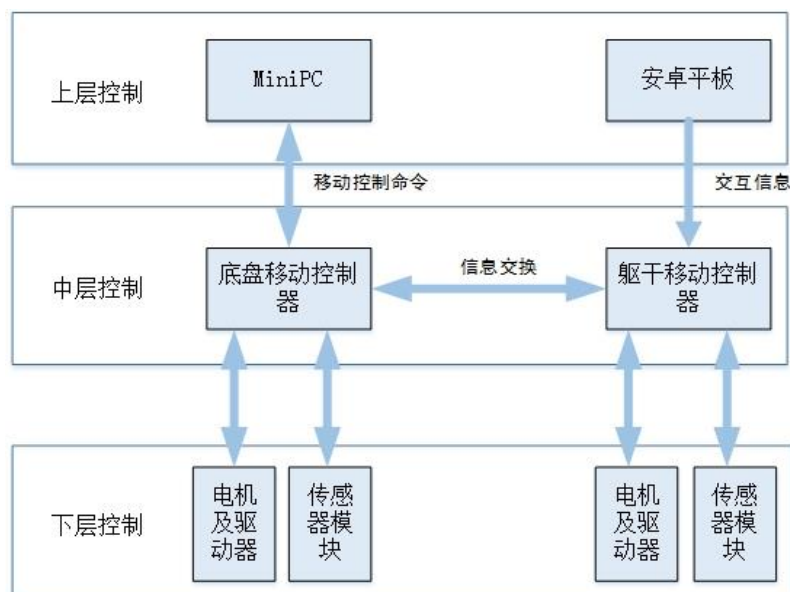


图 2-3 FinaRobot 硬件构架示意图

(2) 交互平台采用安卓平板来充当，该平台能够把安卓平板上交互的相关数据传递到躯干运动的机械中，比如机器人跳舞的整套指令，躯干机械充当数据传递的媒介把该数据传输至 MiniPC 供其执行后续的计算与研究；

(3) 底座运动操作机械用于完成 FinaRobot 底座运动的操纵。并接受 MiniPC 的操作指令相关数据，操作底座的运动以及把躯干运动操作指令传输至躯干操作机械部分。并且把传感器相关的数据和电机运动的相关数据传输至 MiniPC；

(4) 躯干运动操作机械用于接受底座操作机械所传输的躯干操作指令，用来实现操控躯干，完成 FinaRobot 上躯干的运动。并且把躯干传感器相关数据和电机相关数据传输至底座操作机械；

(5) 系统最下部分的传感器相关部分、电机和发动装置为 FinaRobot 运动操作的控制中心以及数据来源中心。其中，传感器部分囊括了许多小部分，比如激光雷达传感器、编码器、电压传感器等。

2.3 移动机器人导航系统业务流程

机器人 FinaRobot 软件系统主要分为三个模块：环境地图创建、融合语音的移动机器人智能导航系统和基于突变点检测的自动充电系统。这三部分在实验阶段相互分离、可并行操作，但在机器人整体导航系统中，这三大模块是相辅相成、缺一不可的。三大模块的总体业务架构如图 2-4 所示。

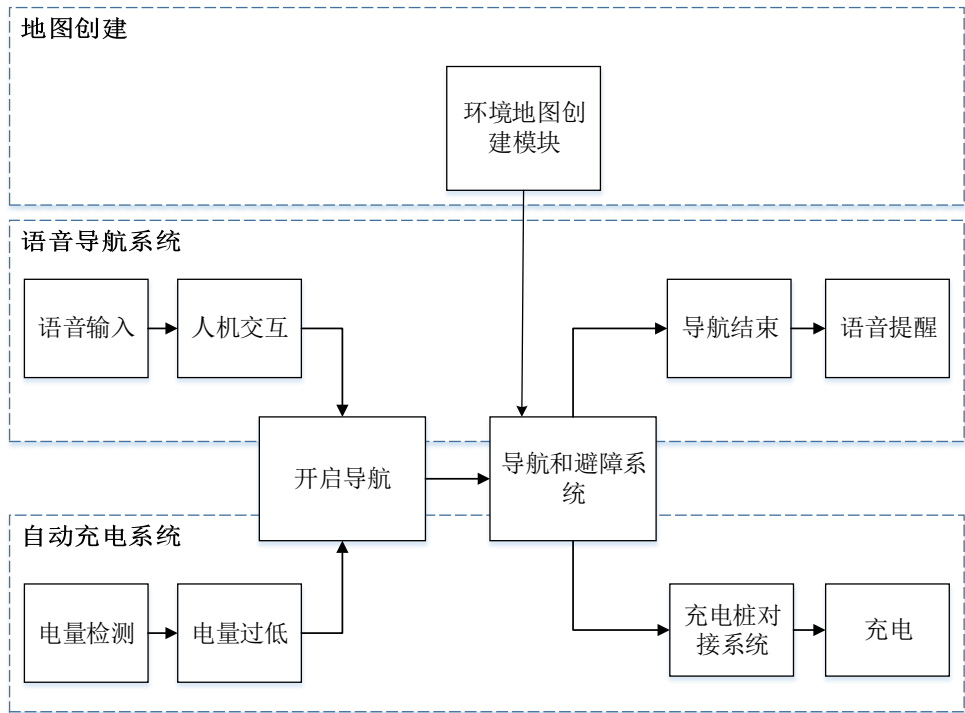


图 2-4 移动机器人导航系统业务架构

地图创建模块为后续的两个导航模块提供已知的环境地图，可以认为是语音导航系统和自动充电系统的基础。语音导航经过语音识别、自然语言处理、答案相似度匹配等过程后开启导航模块，自动充电系统在检测到电量过低之后也会开启导航系统，这两部分共用一个导航系统。完成导航后，它们会分别继续执行后续的操作。

2.4 系统通信模块

在语音智能导航系统和自动充电系统中下位机和客户端之间的通信的作用可谓举足轻重。在语音智能导航系统中，导航开始之前，需要将客户端中语音输入并且经过语义分析提取出的目的地点发送到算法层的导航模块中；当导航结束后，需要将导航结束信息发送给客户端，语音提醒用户目的地已到。在自动充电系统中，客户端会实时接收 MiniPC 发送的传感器数据报（包括电量信息），并显示，当电量低于阈值时，客户端会提醒用户该电量不足以此给用户提供更好的体验。

本文采用的是 socket 通信方式来实现安卓平台和 MiniPC 间的通信。一个完整的通信过程分为 Request 请求和 Response 应答。在双边通信过程中，请求方发起 Request，应答方发送相应的 Response。Socket 通信模型见图 2-5。

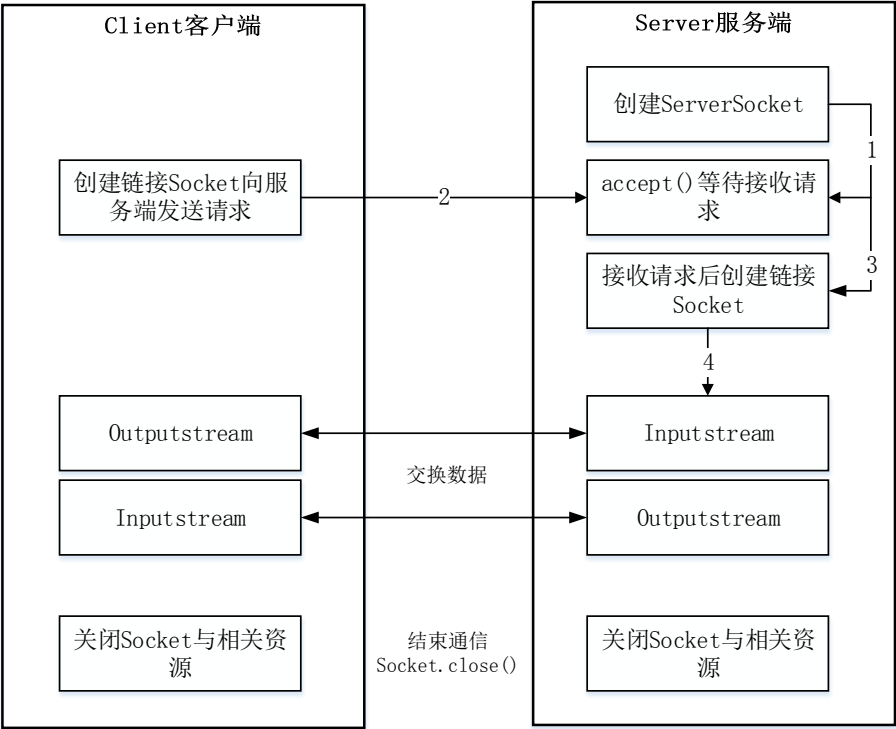


图 2-5 Socket 通信模型

2.5 移动机器人导航系统的软硬件开发环境

FinaRobot 开发过程中所搭建的软硬件开发环境表 2-2 所示：

表 2-2 FinaRobot 软硬件开发环境

名称	备注
主机	主机型号： Intel Windows NUC5PPYH CPU： 奔腾四核 内存 16G、256G 固态硬盘
激光雷达	RPLIDAR，型号为A2M4，最大测距范围8m
服务器	阿里云服务器（Windows server 2008 企业版）
操作系统	Windows7 64位、Ubuntu 14.04 LTS 64位
编程语言	C++/JAVA
PC端编译器	Visual Studio 2012 或者gcc，g++编译器
安卓端编译器	Android Studio2.2

2.6 本章小结

本章在电子科技大学财务机器人(FinaRobot)项目的研发基础上,对 FinaRobot 的系统总体架构进行了阐述,按照系统功能、执行逻辑或平台对整个系统进行分层,并简单阐述各个层级模块的功能。接着就硬件系统的主要组成部分、对应功能和移动控制设计做了简单的说明。然后针对移动机器人 Finarobot 软件系统的三大组成部分:环境地图创建、融合语音的智能导航系统和自动充电系统进行整体的业务流程阐述。最后对各个系统内部的通信协议和整个系统的软硬件开发环境做了相应介绍。通过本章对机器人导航系统的总体架构描述,可以帮助我们更加清晰地了解机器人 FinaRobot 导航系统的整体设计思路,同时也为后三章的分系统详细设计奠定了基础。

第三章 移动机器人定位与建图

SLAM 通常也被称为同时建图和定位问题或者 CML^[27]。当机器人不能访问当前所在的环境地图同时也不能获得自己的位姿时，SLAM 问题就此浮现出来^[24]。此时，我们只能得到两个变量：测量量 $z_{t,i}$ 和控制量 $u_{t,i}$ 。“同时定位和建图”描述了由此产生的问题：将机器人放置在一个陌生环境下，最初它对自身的位姿也无从判断，从这个未知位姿开始，通过控制量 $u_{t,i}$ 来估计下一时刻位姿，接着通过机器人观测传感器获取环境信息，对上述位姿进行估计，同时创建环境地图，并利用观测量和已构建的环境地图对当前位姿进行判断。

本章将针对 SLAM、激光雷达模型和进行简单介绍。同时基于现有的 FastSLAM2.0 算法，提出一种粒子重采样算法，降低粒子匮乏的速度，保持粒子多样性，从而减少建图过程中所需的粒子，提高建图效率。

3.1 SLAM 简介

SLAM 问题是当前机器人研究领域中最困难的问题^[25,26]，由于最初环境地图信息不可知，导致机器人在定位过程中没有可参考的信息，只能依靠估算，所以 SLAM 问题比单纯在地图基础上的定位要难得多；建图过程需要明确机器人的位姿，但由于位姿不明，导致环境地图创建也相当困难。

由上述介绍可知，SLAM 的工作过程可以简单描述为一个闭环的过程：“估计”——“观测”——“矫正”。它的系统状态转化图如下图 3-1 所示：

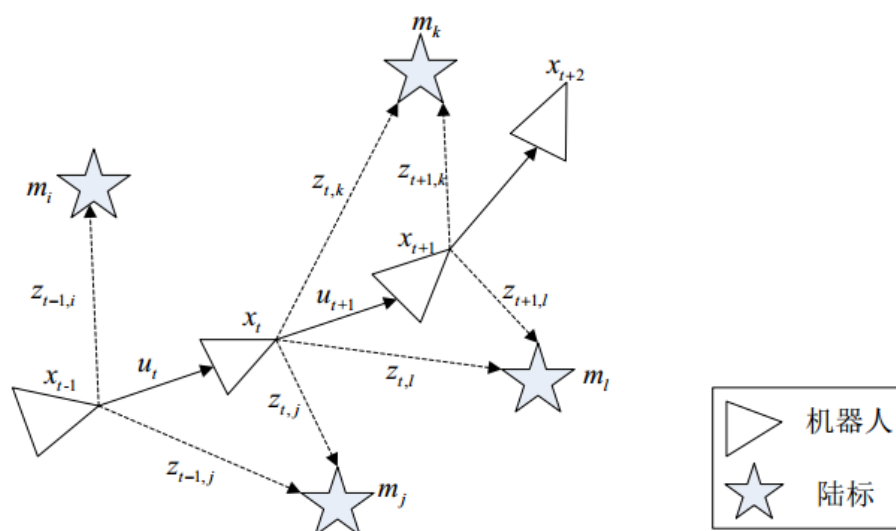


图 3-1 SLAM 系统状态转化图

其中测量传感器的测量量表示为 z_t ，机器人运动控制量表示为 u_t ，环境特征（即环境地图）表示为 m_j ，机器人当前时刻的位姿状态表示为 x_t 。机器人 $t-1$ 时刻的位姿 x_{t-1} 和 t 时刻输入的控制量以及 t 时刻的测量量共同决定了其当前的位姿。

3.1.1 SLAM 系统结构

移动机器人一般通过外部运动控制传感器，如编码器等，以此来对当前的位姿做出估计，但是传感器一般都会存在误差，所以需要一定的辅助信息来帮助矫正估计位姿，这时候就需要传感器识别环境特征得到观测数据，用当前观测测量来判断当前机器人在环境地图中的位姿，利用该位姿来对利用控制量估计出的位姿进行估计，反复执行上述步骤。机器人的位姿全部可以由环境中的特征估算得到。

SLAM 的执行步骤如下：

- （1）已知机器人 $t-1$ 时间点的位姿 x_{t-1} 和 t 时间点的输入控制量 u_t 对机器人 t 时间点的位姿 x_t 和该位姿进行估计，并预测当前位姿的环境特征。
- （2）得到观测传感器数据（如激光数据），进行信息提取，并将该信息关联到环境地图的特征中。
- （3）在环境地图中加入新的特征，再得到测量信息。
- （4）根据测 t 时刻测量信息计算的位姿对机器人当前位姿进行估计，进一步更新地图。

SLAM 的系统流程如下图 3-2 所示：

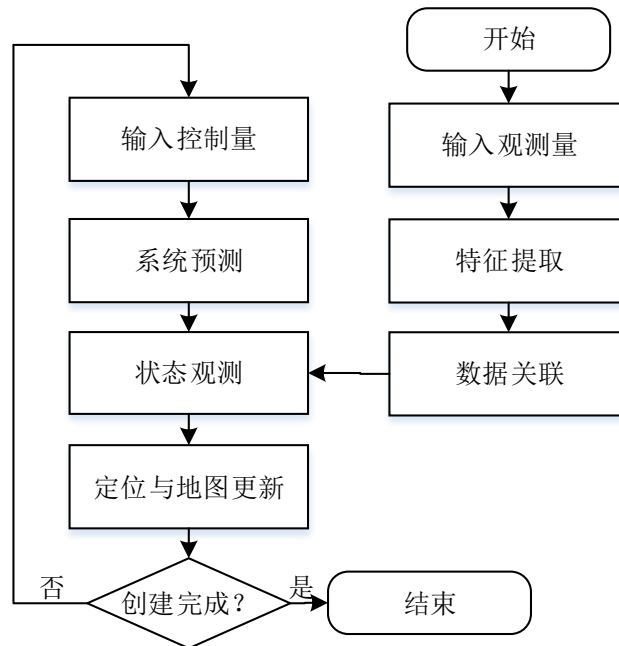


图 3-2 SLAM 系统流程图

SLAM 流程中最主要的就是通过测量量来估计机器人位姿进而对通过控制量估计出来的位姿进行矫正，所以观测信息在 SLAM 中至关重要，因此观测传感器的选取也就非常重要。当前 SLAM 中常用的观测传感器包括声纳传感器、摄像头、激光雷达等。早期 SLAM 中测量数据主要采用激光雷达和声纳传感器获取^[28-30]。由于机器视觉和图像处理技术的逐渐成熟，摄像头等视觉传感器也逐渐被研究人员注意到，同时由于摄像头可以提取出丰富的测量信息并且价格低廉，使得其广泛应用在机器人研究领域^[31]。但是从设备的稳定性、测量精度以及测量范围广度上来看激光雷达还是更胜一筹，虽然价格相对高昂，但是也无法否认它在 SLAM 问题中的适用性。另外，现在国内也推出很多低成本的激光雷达，可以在最大限度的精度保留条件下，降低成本，也不失为很好的选择。

3.1.2 SLAM 的概率模型

移动机器人结合输入控制量以及上一个时刻的位姿进行当前位姿估计，接着利用观测传感器的观测数据确定机器人当前在环境地图中的位姿，继而矫正上述估计位姿，同时依靠定位信息和观测信息迭代地构建环境地图。

用 s_k 描述机器人在 k 时间点的位姿，用 z_k 描述测量传感器 k 时间点的测量量，用 $\Theta = \{\theta_1, \dots, \theta_N\}$ 描述环境地图中 N 个环境特征的集合，用 n_k 描述 k 时刻测量量和环境特征的关联关系，如果在 k 时间点测量量对应的就是特征 θ_i ，那么 $n_k = i$ 。上述信息量的集合表示如下所示：

(1) k 时间点及之前的所有位姿集合 $s^k = \{s_1, \dots, s_k\}$ ，该集合可以表示为机器人的运动路径。

(2) 环境地图中 N 个环境特征值： $\Theta = \{\theta_1, \dots, \theta_N\}$

(3) k 时间点之前所有测量量的集合： $z^k = \{z_1, \dots, z_k\}$

(4) k 时间点前检测到的环境特征关联关系： $n^k = \{n_1, \dots, n_k\}$

SLAM 算法归根结底要解决的问题就是：在已知 k 时间点之前的控制量集合、测量量集合和最原始的位姿信息的前提下，对机器人当前的位姿和环境地图（ N 个环境特征的集合）进行估计。

SLAM 算法可以认为是对机器人位姿集合（路径）和 N 个环境特征集合的特殊后验估计，可以由下式 3-1 表示：

$$P(s^k, \Theta | z^k, u^k, n^k) \quad (3-1)$$

根据贝叶斯概率模型在已知 $k-1$ 时间点的位姿及环境特征集合后验估计概率基础上，计算 k 时刻位姿及环境特征集合的后验估计概率，即：

$$P(s^k, \Theta | z^k, u^k, n^k) = \eta P(z_k | s_k, \Theta, n_k) \int P(s_k | s_{k-1}, u_k) P(s_{k-1}, \Theta | z^{k-1}, u^{k-1}, n^{k-1}) ds_{k-1} \quad (3-2)$$

其中 η 表示常数， $P(z_k | s_k, \Theta, n_k)$ 表示系统观测模型， $P(s_k | s_{k-1}, u_k)$ 表示系统过程模型。

3.2 激光雷达模型

激光雷达一般基于 TOF 原理进行设计，即根据激光发射和遇到障碍物返回经过的时间按照激光在空气中的传播速度折算得到当前位置到障碍物的距离。它具有原理结构简单、扫描频率高、测量距离信息精确度高、数据信息丰富和受光线影响小等优点，故被广泛应用于机器人研究领域。激光雷达从测量维度上还分为单线（2D）和多线（3D）的激光雷达，但是多线的激光雷达一般体积大、价格高昂、信息获取速度慢，故相对单线激光雷达在室内机器人领域中很少使用。单线的激光雷达由于结构简单、价格合适、精度较高等优点，故被普遍选用于室内移动机器人的 SLAM 研究、导航技术、障碍物检测中。

3.2.1 激光雷达概述

本文中使用的 360 度激光雷达是由思岚科技研发的 RPLIDAR，它是一种低价的二维激光雷达。本文使用的激光雷达型号为 A2M4，如图 3-3 所示。它的角度测量范围为 $[0^\circ, 360^\circ)$ ，距离测量范围为 0.15-8m，其应用范围非常广，主要包括：机器人全局导航、SLAM、障碍物检测等。

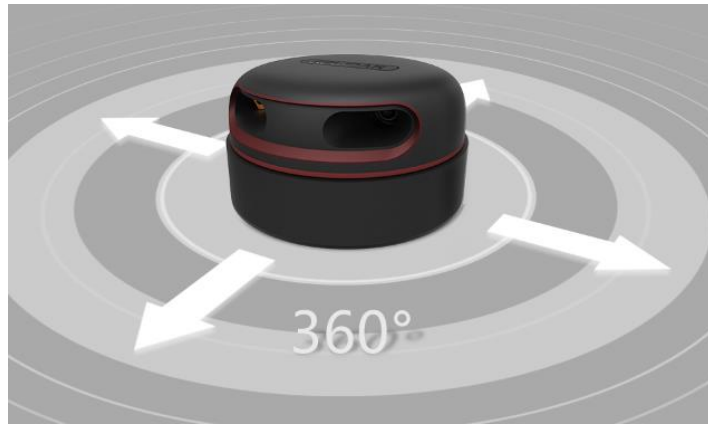


图 3-3 RPLIDAR 外形图

与本文相关的参数包括测距范围、扫描角度和测距分辨率，具体的参数值如表 3-1 所示：

表 3-1 RPLIDAR 性能参数

项目	单位	典型值	备注
测距范围	米	0.15-8/16	基于白色高反光物体测得
扫描角度	度	0-360	全方位测距
测距分辨率	毫米	<0.5	测量物体在 1.5m以内
		<实际距离的 1%	全部量程范围内

3.2.2 激光雷达的原理及坐标转化

如图 3-4 所示，激光雷达的原理和声纳的原理非常相似，主要是通过发射器向外发射红外光线，该光束在接触到障碍物之后经过物理反射返回，由接收器接收后发送给激光雷达内部的数字信号处理芯片，根据往返时间、激光传输速度、该时间内的角度变化，运用三角函数，计算得到障碍物与激光雷达的距离以及当前处于激光雷达角度范围内的角度信息。最终通过 USB 接口输出。

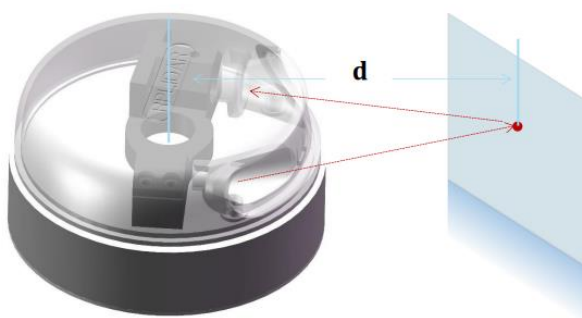


图 3-4 激光雷达工作原理

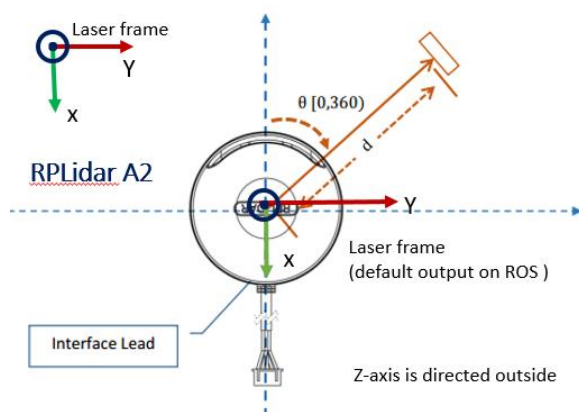


图 3-5 RPLIDAR 扫描数据坐标系定义

RPLIDAR A2 的坐标定义遵循左手法则，雷达引线的方向定义为坐标系的 x 轴，按照左手法则，以雷达中心为原点，旋转 90° 得到的方向作为 y 轴方向，具体的坐标系定义如图 3-5 所示。

3.3 基于 Rao-Blackwellized 的 FastSLAM 算法

利用 Rao-Blackwellization 思想实现的传统的 RBPF-SLAM 算法，包括 FastSLAM1.0 和 FastSLAM2.0，下面针对 FastSLAM2.0 介绍其主要原理和实现步骤，并针对当前 RBPF-SLAM 算法中不合理的重采样方式提出一种改进的重采样策略。

3.3.1 传统的 RBPF-SLAM 算法

3.3.1.1 FastSLAM 原理

RBPF-SLAM 是在粒子滤波的基础上融合 Rao-Blackwellization 思想^[33,34]来解决 SLAM 问题，Rao-Blackwellization 思想的核心就是将完整的 SLAM 问题划分成两部分分别针对性解决：第一部分是而非高斯后验部分，其主要作用是粒子滤波器来进行机器人位姿感知和位姿确定；第二部分是条件高斯部分，其主要作用是利用卡尔曼滤波器^[35]进行环境地图的膨胀和更新。具体表达式如下：

$$\begin{aligned}
 p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) &= p(m | x_{1:t}, z_{1:t}) p(x_{1:t} | x_{1:t-1}, u_{1:t-1}) \\
 p(s^k, \Theta | z^k, u^k, n^k) &= p(s^k | z^k, u^k, n^k) p(\Theta | s^k, z^k, u^k, n^k) \\
 &= p(s^k | z^k, u^k, n^k) \prod_{i=1}^N p(m_i, s | z^k, u^k, n^k)
 \end{aligned} \tag{3-3}$$

上式 3-3 中， $p(m | x_{1:t}, z_{1:t})$ 对环境地图进行估计，采用的方式是扩展卡尔曼滤波， $p(x_{1:t} | x_{1:t-1}, u_{1:t-1})$ 是对机器人位姿进行估计，采用的方式是粒子滤波。环境地图估计又可以分解为 N 个环境特征的信息集合， N 环境特征之间是没有相关性的，它的测量信息只与机器人的位置有关，所以针对单独的环境特征采用单独的拓展卡尔曼滤波器进行估计。

现在在 SLAM 试验中选取了 M 个粒子，环境地图中有 N 个环境特征点。针对每个粒子都需要 N 个扩展卡尔曼滤波器对该环境地图中的 N 个环境特征进行估计，那么该过程一共需要 MN 个扩展卡尔曼滤波器，所以 FastSLAM 算法总体的时间复杂度为 $O(MN)$ ，利用多叉树对算法进行优化，将时间复杂度优化到 $O(M \log N)$ 。

FastSLAM 中的扩展卡尔曼滤波，它的分布类似于线性高斯函数。故经过 EKF 估计得到的环境特征观测模型是服从高斯分布的。即便是控制量不服从线性分布，

最终得到的环境特征仍然是服从高斯分布的。这就是为何可以使用粒子近似表示机器人位姿。

在 FastSLAM 算法中，机器人在 k 时刻及之前的位姿信息集合可以表示为一条路径，即 $p(s^k | z^k, u^k, n^k)$ ，该路径是用粒子滤波估计得到。对于其中任意一个粒子，都需要 N 个卡尔曼滤波器对该环境中的 N 个环境特征进行估计，环境特征 i 的估计值，即 $p(\theta_i | s^k, z^k, n^k)$ 。那么粒子 i 的表示如下式：

$$S_k^i = \{s_k^i, \mu_{k,1}^i, \Sigma_{k,1}^i, \dots, \mu_{k,N}^i, \Sigma_{k,N}^i\} \quad (3-4)$$

按照上述描述，1 个粒子的表示包含了移动机器人从初始位姿到 k 时刻的位姿估计集合 s_k^i 以及对环境地图的估计（ N 个环境特征），第 m 个环境特征估计的服从高斯分布 $N(\mu_{k,N}^m, \Sigma_{k,N}^m)$ 。

3.3.1.2 FastSLAM 实现步骤

FastSLAM 2.0 [36,37] 的主要思想是：在 $k-1$ 时刻的所有粒子位姿集合 S_{k-1} 的基础上，根据 k 时刻的控制输入量 u_k 对 k 时刻位姿进行估计，接着利用传感器观测数据 z_k 对有误差估计的位姿进行估计和矫正，最后增量式地求解出 k 时刻的所有粒子新的位姿集合 S_k 。FastSLAM 2.0 算法的实现步骤如下：

(1) 初始化阶段。规定合适的粒子数量 N ，指定机器人在 $t=0$ 时刻的位姿，以该位姿作为机器人初始规划区域（局部地图），并在该区域上均匀地撒上粒子，对粒子的分布 $s_0^{(i)} \sim q(s_0)$, $i=1, 2, \dots, N$ 和权重 $w_0^{(i)} = 1/N$ 进行初始化。

(2) 粒子转移阶段。在当前时间读取到控制输入量 u_t （一般由里程计信息转化得到），然后通过状态转移方程对当前时刻的粒子进行位姿更新，得到一个新的预测粒子 $s_t^{(i)}$ 。

(3) 粒子决策阶段。针对 (2) 中通过输入控制量估计得到的粒子位姿，结合观测传感器（激光雷达、摄像头等）测量当前环境特征，判断机器人当前所在的位姿，以此位姿对由控制量估算得到的位姿进行估计。最后由重要性采样公式计算新粒子的权重，计算公式如下：

$$w_t^{(i)} = \frac{p(x_{t:t}^{(i)} | z_{1:t}, u_{1:t-1})}{q(x_{t:t}^{(i)} | z_{1:t}, u_{1:t-1})}, i=1, \dots, N \quad (3-5)$$

(4) 计算 N_{eff} 值，如果 $N_{eff} > N_{th}$ 则暂时不进行重采样，反之，进入重采样步骤。

(5) 重采样。总的来说重采样的目的就是淘汰权重小的粒子，保留权重大的粒子，同时复制大权重的粒子来保持粒子总数。

(6) 地图更新阶段。针对单独的粒子 i ，在已知 t 时刻前的位姿集合（运动路径）和 t 时刻前的测量量 $z_{1:t}$ 的前提下，可以估算出 i 粒子对应的环境地图 $p(m^i | x_{1:t}^i, z_{1:t})$ ，每个粒子都会保存其对应的环境地图。

(7) 在下一个时间段内，循环执行 (2) - (7)。

当机器人在 t 时刻前进到一个新的位姿，在已知控制输入量的基础上完成位姿的估计，同时针对当前光测量对新粒子的权重进行重新计算，随着粒子移动步数的逐渐增多，上述算法的计算量也逐渐增大。因此 Doucet 等人[34]通过满足下式 3-6 假设的重要性密度函数 q ，继而采用得到的递归公式计算粒子权重：

$$q(x_{1:t}^{(i)} | z_{1:t}, u_{1:t-1}) = q(x_t | x_{1:t-1}, z_{1:t}, u_{1:t-1}) q(x_{1:t-1} | z_{1:t-1}, u_{1:t-2}) \quad (3-6)$$

从式(3-6)得到计算粒子权重的递归公式：

$$\begin{aligned} w_t^{(i)} &= \frac{\eta p(z_t | x_{1:t}^{(i)}, z_{1:t-1}) p(x_t^{(i)} | x_{1:t-1}^{(i)}, u_{1:t-1})}{q(x_t^{(i)} | x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1})} \cdot \frac{p(x_{1:t-1}^{(i)} | z_{1:t-1}, u_{1:t-2})}{q(x_{1:t-1}^{(i)} | z_{1:t-1}, u_{1:t-2})} \\ &\propto \frac{p(z_t | m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)} | x_{1:t-1}^{(i)}, u_{1:t-1})}{q(x_t | x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1})} \cdot w_{t-1}^i \end{aligned} \quad (3-7)$$

其中 $\eta = \frac{1}{p(z_t | z_{1:t-1}, u_{1:t-1})}$ 为常数，针对每个粒子它都是一样的。

3.3.2 改进重采样策略的 RBPF-SLAM

在传统粒子滤波中，粒子每进行一次更新就会迭代地计算对应权值，随着更新次数的增多，大部分的权重都收敛于一小部分粒子上，大部分的粒子处于无用的状态。这种情况下，后验概率密度函数已经得不到有效的表达。以上描述的就是粒子退化现象。

缓解退化现象的方式有两种：选择好的重要性采样函数或者进行重采样^[45]。文献[38-44] 基于以上两个方向讲述了多种改进算法，但目前看来，改进效果一般。

重采样策略来缓解粒子退化现象的主要思想便是：保留高权重粒子，剔除低权重粒子，为保持粒子总数不变，复制高权重粒子。但是基于以上策略，经过多次重采样步骤之后，粒子多样性降低，导致滤波发散^[46]。最终除了一个粒子以外所有的粒子权重都为零，这就是所谓的粒子匮乏问题。可以从以下两个方面着手解决粒子匮乏的问题：降低重采样次数和进行合理重采样。

首先针对如何降低重采样次数，引进 N_{eff} 值来估计当前是否需要重采样， N_{eff} 定义如下：

$$N_{eff} = 1 / \sum_{i=1}^N (w^{(i)})^2 \quad (3-8)$$

式中 $w^{(i)}$ 为进行归一化处理后的粒子权重。当 $N_{eff} < N_{th}$ 时，执行重采样。另外，可以对当前状态进行判断，当状态为 $x_t = x_{t-1}$ 即机器人没有任何运动的时候，不进行重采样操作。以上两种方式虽然一定程度上减少了重采样的次数，但是对每个粒子都进行归一化计算或者进行状态判断却降低了算法效率，所以还需要从如何进行合理重采样角度改进我们的算法。

目前，针对改进的采样算法，已有的研究包括：系统重采样^[50]、多项式重采样^[47]、残差重采样^[48]、分层重采样^[49]等。

解决粒子匮乏问题的核心思想：给非主流的粒子一定的机会。针对上述思想，本文提出了一种分区的自适应局部重采样算法，其基本思路是：对粒子按照权值大小进行排序，并且划分为三个权值区域 I、II、III，将中间区域 II 的粒子全部保留，合并 I、III 区域的粒子得到区域 IV，在该区域内采用自适应的粒子复制和剔除方式（利用高权值粒子和低权值粒子的比值采取不同的复制策略）。具体步骤如下：

步骤 1：对粒子集合按照权重从低到高排序，划分区域。通过经验设置合适分区阈值：高权重阈值 $w_H = 2/N$ 和低权重阈值 $w_L = 1/(2N)$ ^[51]。根据上述两个划分界限将粒子划分为三个部分

Part I: 低权重粒子区：

$$\{x_t^{(i)}, w_t^{(i)}\} (w_t^{(i)} < w_L), i \in [1, N_L] \quad (3-9)$$

Part II: 中等权重的粒子区：

$$\{x_t^{(i)}, w_t^{(i)}\} (w_L \leq w_t^{(i)} \leq w_H), i \in (N_L, N - N_H) \quad (3-10)$$

PartIII: 高权重粒子区：

$$\{x_t^{(i)}, w_t^{(i)}\} (w_t^{(i)} > w_H), i \in [N - N_H, N] \quad (3-11)$$

其中 N_L 和 N_H 表示低权重的粒子和高权重的粒子数量。

步骤 2：低权重粒子区与高权重粒子区合并为 PartIV：

$$\{x_t^{(i)}, w_t^{(i)}\} (w_L \leq w_t^{(i)} \leq w_H), i \in [1, N_L] \cup [N - N_H, N] \quad (3-12)$$

令 $N_w = N_L + N_H$ 来表示 PartIV 部分的粒子数目（该集合中的粒子按照权重有序排列）

步骤 3：采用自适应的粒子复制方式。

当 $N_L \leq N_H$ 时，各个区域的粒子被复制的次数 n_i 如下：

$$n_i = \begin{cases} 0, & \sum_{k=1}^i w_k < 1/2N, \text{ 且 } i \in [1, N_L] \\ 2 + \lfloor N_H / N_L \rfloor, & \sum_{k=1}^i w_k \geq 1/2N, \text{ 且 } i \in [1, N_L] \\ 1 + \lfloor N_H / N_L \rfloor, & \text{ 其他} \end{cases} \quad (3-13)$$

当 $N_L > N_H$ 且 $N_L / N_H < 2$ 时, 各个区域的粒子被复制的次数 n_i 如下:

$$n_i = \begin{cases} \lfloor N_L / N_H \rfloor + 2, & i = \lfloor N_w / 2 \rfloor \\ \lfloor N_L / N_H \rfloor + 1, & i \in (\lfloor N_w / 2 \rfloor, N_w] \end{cases} \quad (3-14)$$

当 $N_L > N_H$ 且 $N_L / N_H \geq 2$ 时, 各个区域的粒子被复制的次数 n_i 如下:

$$n_i = \begin{cases} \lfloor N_L / N_H \rfloor + 2, & i \in [\lfloor N_w / 2 \rfloor, N_L] \\ \lfloor N_L / N_H \rfloor + 1, & i \in (N_L, N_w] \end{cases} \quad (3-15)$$

按照上述自适应策略最后统计粒子复制的总数, 若复制总数小于 N_w , 则从 PartIV 被复制的粒子中按照权值从高到低的顺序依次复制直到补齐粒子个数; 若复制总数大于 N_w , 则按照权值从低到高的顺序依次减少, 直到复制粒子数等于 N_w 。

本文提出的改进的重采样策略, 完全保留中等权重的粒子, 同时在低权重粒子较少的情况下给它们一定的机会。这种采样策略需要计算的粒子数目减少, 从而提高了重采样算法的效率。同时自适应的粒子复制和剔除方式能够在很大程度上保持粒子多样性, 缓解粒子匮乏的问题。

3.3.3 传统 RBPF-SLAM 和改进的 RBPF-SLAM 实验对比

本次实验分为仿真实验和真实环境建图实验, 仿真实验是从状态误差和运行时间两个方面来比较改进的自适应分层重采样算法和其他重采样算法的性能差异; 真实环境建图实验通过搭建实验平台分别在不同的实验环境下进行建图, 最终通过建图结果来验证改进算法的可行性。

3.3.3.1 仿真实验对比

本文采用 Matlab 2012b 进行仿真, 硬件处理器为 Intel Core i5, 内存 4GB, 操作系统为 Windows 7。

BOT(Bearings Only Tracking) 模型是在有噪声情况下, 跟踪在二维平面内做匀速运动的目标。该跟踪模型的方程式如下:

$$\begin{cases} X_t = FX_{t-1} + U_t \\ Z_t = HX_t + V_t \end{cases} \quad (3-16)$$

其中 $F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, $X_t = \begin{bmatrix} x_t & y_t & \dot{x}_t & \dot{y}_t \end{bmatrix}^T$ 中的各参数分别对

应 t 时刻目标的 x 坐标位姿、 y 坐标位姿、 x 方向的速度和 y 方向的速度。 $\Delta t=1$ 表示采样时间间隔。 $U_t \sim N(0,5)$, $V_t \sim N(0,5)$ 均是服从高斯分布的白噪声, 分别表示运动噪声和测量噪声。由于是匀速运动, x 和 y 方向的速度由当前时刻目标朝向角度决定, 分别表示为:

$$\dot{x}_t = -d * \cos \theta_t \quad (3-17)$$

$$\dot{y}_t = d * \sin \theta_t \quad (3-18)$$

其中 $d=80/T$ 表示每个时间步长的运动长度, $\theta=\pi/T$ 表示每个时间步长的旋转角度, T 表示总步数。另外, 将 X_0 的初始状态设置为 $[50 \ 20 \ d \ 0]^T$, 表示从坐标(50,20)处开始运动。

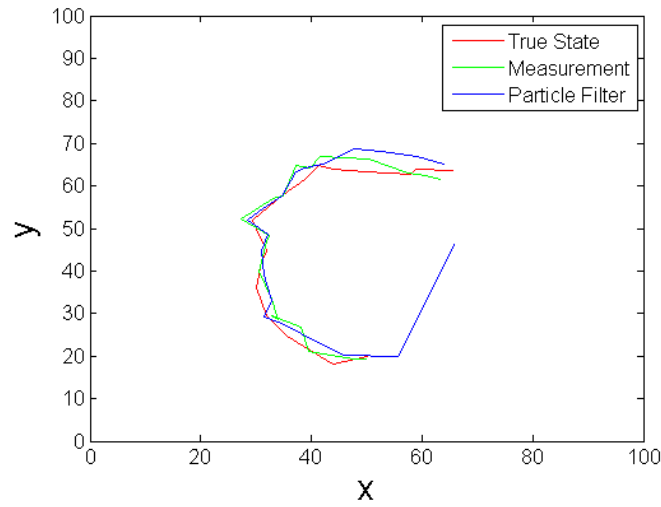


图3-6 系统运动轨迹

首先在100*100的地图上, 设 $N=20$, $T=20$ 时的运动轨迹图如图3-6所示, 其中蓝色曲线表示的是所有粒子的几何中心位置移动轨迹。

实验结果采用粒子几何中心与系统真实状态的误差error作为算法性能的评估标准, 误差越小表示粒子越收敛于系统真实状态, 即算法性能更好。针对无阈值限定重采样、随机重采样、系统重采样、残差重采样、多项式重采样和自适应分层重采样进行如下误差仿真对比:

(1) $N=100$, $T=100$:

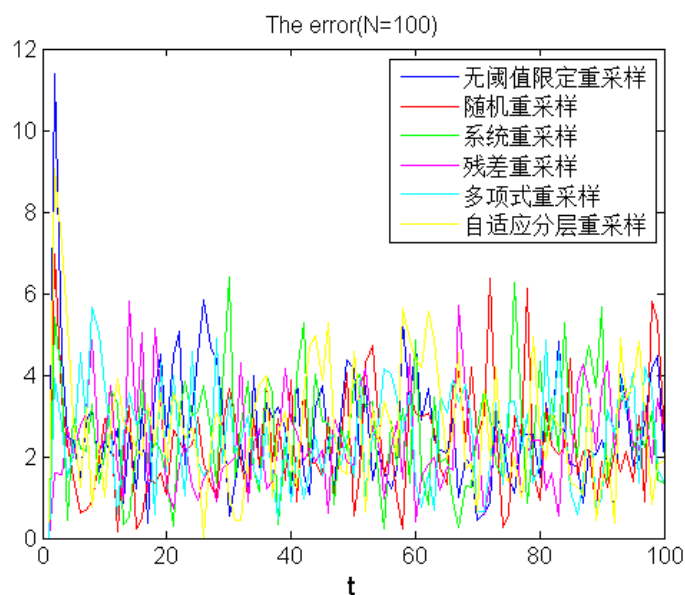


图3-7 $N=100$ 时各重采样算法的误差

(2) $N=20$, $T=100$:

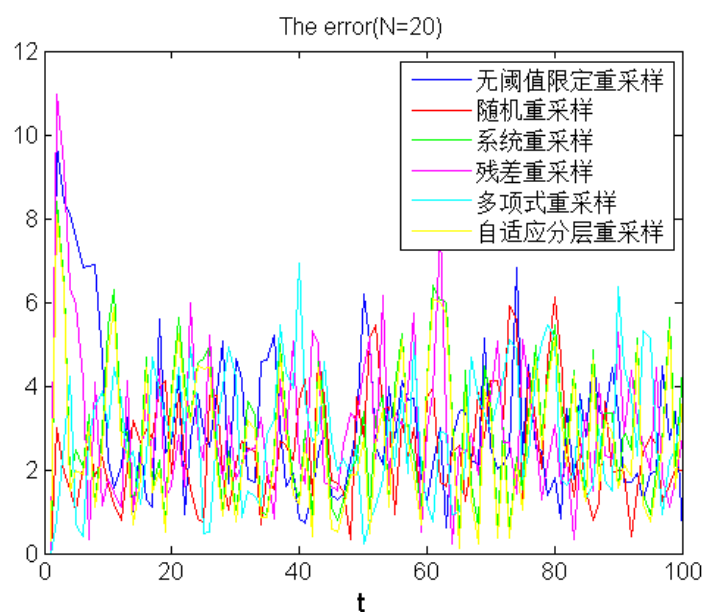


图3-8 $N=20$ 时各重采样算法的误差

表3-2 各个重采样算法在粒子数不同情况下的平均误差

	无阈值限定重采样	随机重采样	系统重采样	残差重采样	多项式重采样	自适应分层重采样
$N=100$	2.6410	2.3366	2.3579	3.0253	2.4961	2.3585
$N=20$	3.0977	2.5320	2.9703	3.0407	2.9000	2.6790

对上述两组数据进行对比发现，当N取值较大时，各个重采样算法的性能无法做出直观的比较。主要原因是，N较大的情况下，即使粒子多样性下降很快，在较小的步数范围内，仍旧能够满足较为精准的收敛追踪。

当N取较小的数值时，各个重采样算法对于保持粒子多样性的性能也就凸显出来。其中无阈值限定重采样的平均误差最大，本文改进的自适应分层重采样算法平均误差比较低，性能表现较为优异。

为验证改进算法在效率方面的优化，分别选取不同的步数和粒子数对六种算法进行10次实验，表3-3比较了六种算法完成追踪的平均消耗时间。

表3-3 各个重采样算法的平均消耗时间

	无阈值限定 重采样	随机 重采样	系统 重采样	残差 重采样	多项式 重采样	自适应分 层重采样
N=1000 T=100	22.6780	22.3610	21.2600	21.2120	21.1400	21.0760
N=100 T=500	73.7800	73.5930	72.7480	72.6000	72.8990	72.4350

由于改进的自适应分层重采样算法在进行重采样过程中只针对部分粒子进行操作，并且有着较低的算法复杂度，所以可以在一定程度上减少粒子滤波的计算时间，提高运行效率。

3.3.3.2 实验平台

本实验平台为电子科技大学财务机器人（FinaRobot），机器人底盘为全向轮三轮轮式底盘，并且在底盘上面安装有 RPLIDAR A2M4 激光雷达。同时采用串口通信的方式连接一台 Ubuntu 14.04 的 Linux 系统的 MiniPC，该 Linux 系统内安装了 indigo 版本的 ROS 系统，机器人的运动通过机器人移动控制程序进行远程控制，该机器人运动的最大速度为 0.4m/s。根据输入的激光雷达数据和编码器数据构建地图，最后可在可视化界面 rviz 中展示。

3.3.3.3 实验环境和结果

本实验的两个实验环境，分别为龙湖 1501 室和走廊。针对以上两个环境分别采用传统的 RBPF-SLAM 算法和改进的 RBPF-SLAM 算法进行实验。

龙湖 1501 室的面积近似为 $15*8m^2$ ，建图节点配置的栅格大小为 $5cm*5cm$ ，分别采用两种算法进行地图构建，图 3-9（a）展示的是传统 RBPF-SLAM 采用 10 个粒子所建出的环境栅格地图，粒子匮乏严重，累积误差逐渐增大，导致最终建立的地图误差较大；图 3-9（b）展示的是改进算法采用 10 个粒子所建出的高精度

的环境栅格地图。另外，最终得到效果大致相同的地图时（传统 RBPF-SLAM 采用 30 个粒子），改进算法在这个过程中消耗的时间要低于传统 RBPF-SLAM 算法。

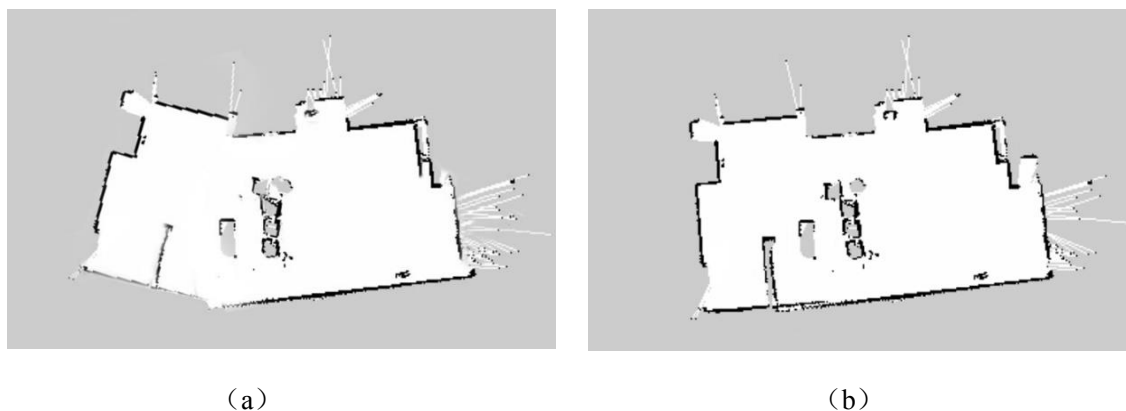


图 3-9 实验结果 (a) 传统 RBPF-SLAM; (b) 改进的 RBPF-SLAM

走廊环境的面积近似 $10 \times 2 \text{ m}^2$ ，图 3-10 展示的是改进的 SLAM 算法，仅适用 5 个粒子便构建出较为精确的栅格地图。

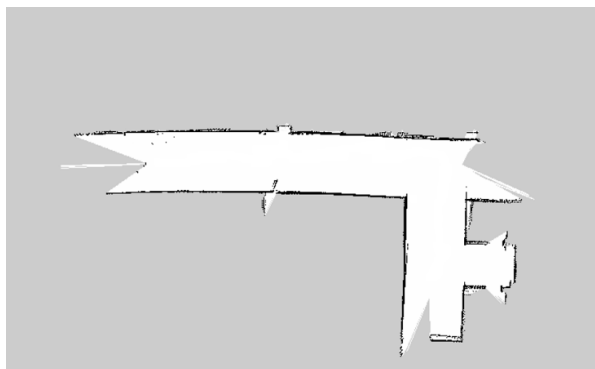


图 3-10 走廊环境地图构建实验结果

3.3.3.4 实验结果分析

本算法采用分区的自适应局部重采样策略来替换原本的重采样方案，可以有效地保持粒子多样性，缓解粒子匮乏问题，使其不同粒子数明显多于 RBPF-SLAM 算法，这也是改进算法需要较少粒子数的重要原因。同时算法运行的时间也随着粒子数目的减少以及算法复杂度的降低而减少，提高算法效率。

3.4 环境地图创建系统结构

环境地图创建系统结构框图如图 3-11 所示，该系统主要包含两大部分：消息传递机制和 Gmapping 建图，Gmapping 在接收到激光雷达数据和里程计数据之后，结合机器人运动控制，最后成功创建环境地图。

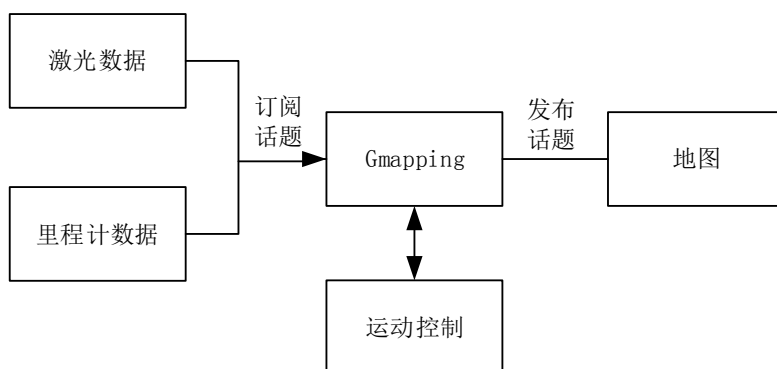


图 3-11 环境地图创建系统结构框图

3.4.1 消息传递机制

ROS 各个节点之间是通过一种消息传递机制来进行通信的，ROS 将消息存放在各个话题中，如果一个节点想要向外传递该节点的信息，那么就需要向某话题（可以一个或者多个）发布（publish）消息。通过上述描述，一个节点如果要想从外界获取消息自然不能坐等对方发送，它需要去订阅消息对应的话题，当话题内有内容发布过来时，该节点的回调函数便能收到该消息。另外 ROS 中的节点管理器的并不参与消息的转交，它只负责统一协调各个节点能通过上述的消息传递机制找到对方，完成通信。在本节的建图环节中，需要订阅和发布的话题如下：

订阅话题：

（1）tf (tf/tfMessage)。tf 转换机制，将 base_frame、laser_frame、odom_frame 和 map_frame 的坐标系连接起来，因为各个坐标系原点不会在同一位置，比如 laser 的坐标系往往和机器人运动坐标系不重叠，所以坐标系转化就尤为重要，各个坐标系的连接情况见图 3-12。只需要将几个坐标系的位置关系发布给 tf，它便能代替我们处理转换关系。

（2）scan (sensor_msgs/LaserScan)。建图过程中首要的数据--激光数据。

发布话题：

（1）map_metadata (nav_msgs/MapMetaData)。向 map_metadata 话题发布地图消息，同时定期更新。

（2）map (nav_msgs/OccupancyGrid)。向 map 话题发布地图消息，同时定期更新。

（3）~entropy (std_msgs/Float64)。根据机器人的位姿来估计分布熵（熵越大表示不确定性越大）。

服务：

（1）dynamic_map (nav_msgs/GetMap)。向 dynamic_map 话题发布地图数据

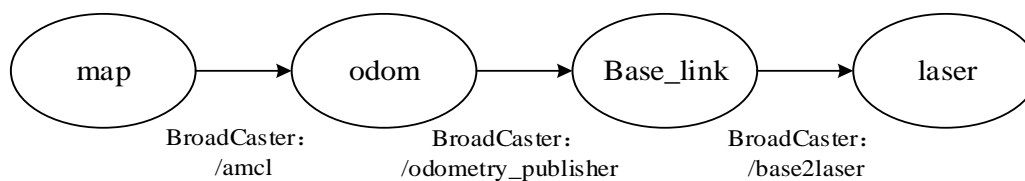


图 3-12 ROS 中各坐标系的转换关系

3.4.2 Gmapping 建图

Gmapping^[52]的程序框架按照应用实现过程可以分成两个大部分（gmapping 和 openslam_gmapping），两者的调用关系见下图 3-13。

其中核心算法部分都在 openslam_gmapping 包中，实现流程如下图 3-14 所示。

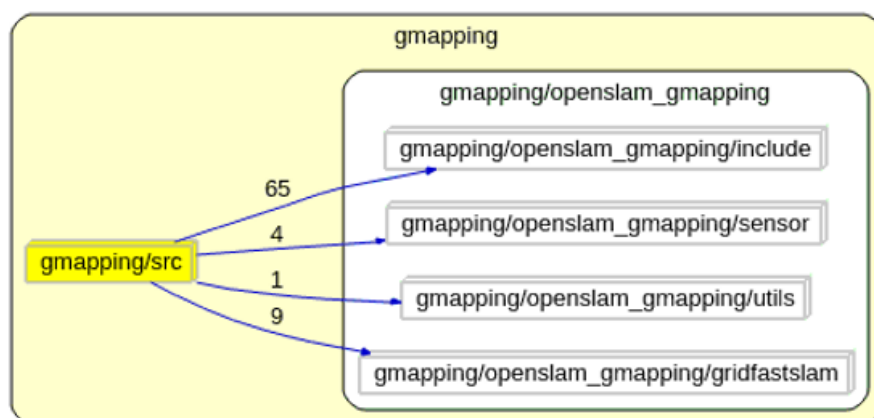


图 3-13 gmapping 和 openslam_gmapping 的调用关系

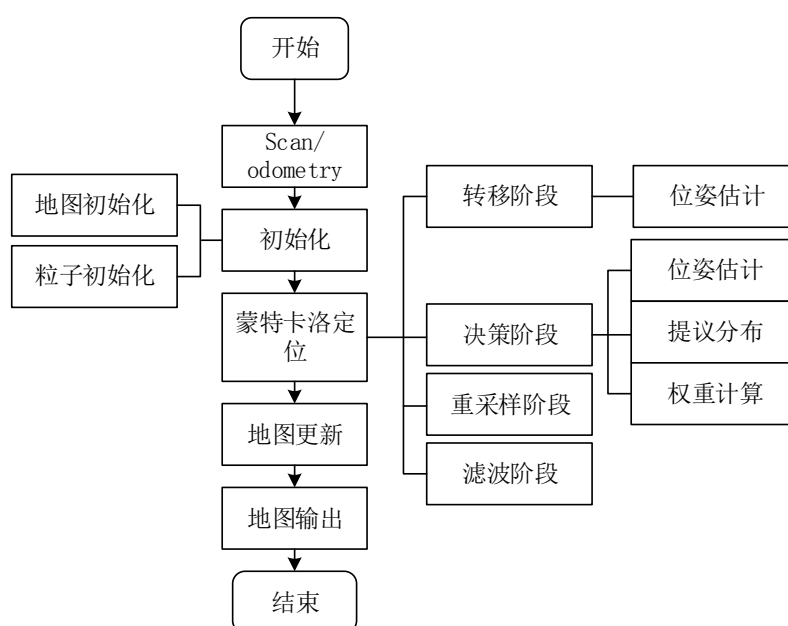


图 3-14 openslam_gmapping 的实现流程

(1) 数据输入

该部分在 `slam_gmapping.cpp` 中实现。

创建 `slam_gmapping` 节点，订阅 `LaserScan` 话题和监听 `tf` 树广播实时获取激光数据和里程计数据，获取到数据后的主要操作在 `laserCallback` 函数里面。在该函数里面，根据 `got_first_scan` 参数判断是否是第一次接收到激光数据，若是则初始化地图。若否，则直接进入蒙特卡洛定位算法中。

(2) 初始化

初始化在 `slam_gmapping.cpp` 的 `initMapper()` 函数中调用，内部调用类 `GridSlamProcessor` 中的 `init()` 方法，执行地图和粒子的初始化同时设置运动模型和扫描匹配等过程的参数。

地图初始化：初始化地图的坐标系、坐标原点、栅格大小等参数。

粒子初始化：对每个粒子赋初始位置、维护的初始地图、运行轨迹、权重等参数。

(3) 蒙特卡洛定位算法

蒙特卡洛定位跟踪的主线代码集中在 `gridslamprocessor.cpp` 中的 `processScan` 方法中，按照阶段划分可以分成以下三个主要阶段：转移阶段、决策阶段和重采样阶段。按照算法流程可以划分成以下五个主要模块：位置估计、扫描匹配、权重更新、重采样和下阶段准备。具体流程以及其对应的模块方法如下图 3-15 所示：

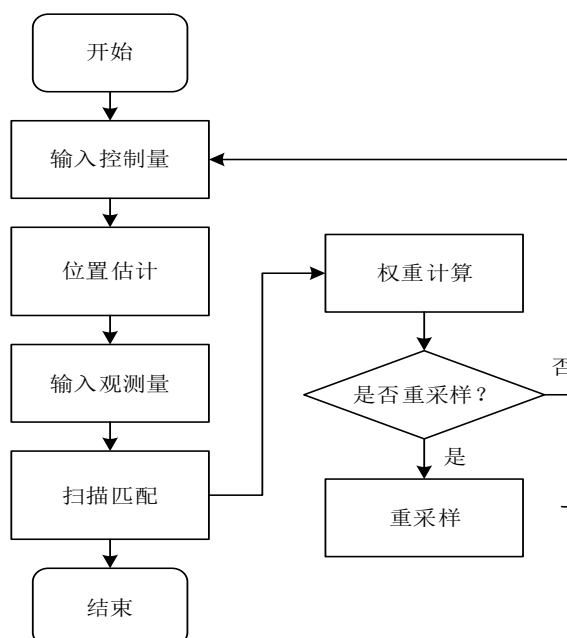


图 3-15 蒙特卡洛定位算法的实现流程

程序在接收到激光数据之后（非第一个激光数据），将该条激光数据和对应的里程计数据封装成一条 `RangeReading` 数据。对所有粒子进行迭代，首先调用 `drawFromMotion()` 函数和 `onOdometryUpdate()` 函数得到当前时刻粒子所在的位置估计，即转移阶段；接着，根据激光测量数据调用 `scanMatch()` 函数和 `onScanmatchUpdate()` 对当前的估计位置进行评分，之后调用 `updateTreeWeight()` 函数来计算并更新每个粒子的权重，即决策阶段；最后根据重采样阈值判断是否进行重采样，若是则进入重采样阶段则调用 `resample()` 函数进行重采样操作，之后再调用 `updateTreeWeight()` 函数进行粒子权值更新，否则，更新粒子维护的轨迹树、位姿等数据，为下一次定位跟踪做准备。

（4）地图更新

在完成上述操作后要实现粒子所携带的地图的更新，在 `gridslamprocessor.cpp` 中调用 `ScanMatcher` 类的 `registerScan()` 方法执行地图的更新。在 `slam_gmapping.cpp` 中 `updateMap()` 方法实现地图更新（地图膨胀）。

3.5 本章小结

本章首先对 SLAM 算法基本原理进行简单介绍，其次激光雷达模型进行介绍，并且对激光雷达的原理和坐标系定义也做了相应说明。

接着基于现有的 FastSLAM2.0 算法，提出一种分区的自适应局部重采样算法来解决粒子退化问题，保持粒子多样性，从而减少建图过程中所需的粒子，进一步降低计算复杂度，提高建图效率。同时对该改进的算法，完成代码编写，选取两个种不同复杂程度的室内环境进行实验，验证这种改进 RBPF-SLAM 的优化效果。

最后介绍整个环境地图创建系统的业务流程，通过整体流程的介绍对上文中介绍的原理有了更清晰的认识，同时也为论文实验阶段奠定基础。

第四章 基于语音交互的移动机器人智能导航系统

语音识别^[53]技术的发明旨在让机器能够明白人类语音话术的真实意义，并且通过机器自身的处理最终将语音翻译成同等意思的本文信息。机器人想要反馈的信息通过语音合成技术转化为人能理解的语音。语音识别和语音合成技术的实现需要很多相关学科的支撑，如数字信号处理（DSP）、语言学（linguistics）、人工智能（AI）。通俗一些讲，语音识别和语音合成技术就是机器人通过各个学科的复杂交互将一段语音信号理解透彻后翻译成同等意思的本文信息，也能将一段文本通过复杂处理后合成语音。归根结底，该技术的出现就是能让机器和人采用人类易懂的自然语言进行沟通^[54]。

导航是指在机器人能够在已知或者位置环境地图的情况下，实现自身位姿的判断，并且能够在实现从起点到终点规划出一条合理路径，在沿着该路径前往终点的过程中遇到障碍物进行合理规避，最后完成自主移动。另外，导航问题仍然是个很复杂的问题，它涉及了感知、规划、结构、执行、问题求解和计算效率等一系列人工智能的问题，所以还是具有很多的研究价值。

本章将融合语音对话和机器人导航技术设计一套智能的、交互良好的室内移动服务机器人智能引路系统。

4.1 语音交互导航系统关键技术

语音交互导航系统涉及的关键技术包括语音识别与语音合成(Speech Recognition, SR 和 Text To Speech ,TTS)、自然语言处理(natural language processing, NLP)(这里的自然语言处理指的是狭义的自然语言处理,不包含语音识别与合成)和基于 ROS 的导航技术。由于语音识别采用的是科大讯飞 SDK,故不作介绍。下面就另外两项关键技术进行说明。

4.1.1 自然语言处理

作为语言学和人工智能的分支学科，自然语言处理技术（natural language processing, NLP）^[55]可以概括性地描述为：使用不同的方法处理自然语言并且将处理过的结论进行合理的利用。自动分词算法、语法分析、词性标注、文本分类等技术都是 NLP 研究的方向。

本章我们将基于自然语言处理设计一套财务问答系统，其中也包含协助带领去某个地点的问题设定，该部分将会融合导航技术形成一个完整的移动机器人智

能引路系统。

财务问答系统从用户问题的提出到答案的反馈应该经过以下步骤：

步骤 1：分词和近义词替换

获取问题的文本格式，采用 `jieba` 分词组件完成分词。完成分词后需要对近义词进行替换，以便在步骤 4 近似匹配过程中能有更好的匹配结果。所以对一些相近意义的词进行统一替换。

步骤 2：词性标注

对分词后的每个词都进行词性标注，中文中很多词汇都有大于一种的词性，在不同的语境下面都会表现出不一样的词性，同一个词的不同词性我们在进行余弦近似的时候可以当做是两个不同的词进行匹配。

步骤 3：删除停用词

停用词是一些完全没有意义的词汇，比如类似“的”、“啊”、“要”“把”这样的助词和语气词等词汇。一般会引入一个停用词表，将包含在该表内的词都剔除掉。

步骤 4：余弦近似

完成上述步骤后，得到最终用于匹配的关键词组，将该关键词组发送到后台服务器和数据库中的问答库中的答案关键词组进行余弦相似度匹配。

步骤 5：答案反馈

考虑到一个问题可能会存在多个符合阈值要求的答案，所以对于所有可行的答案按照余弦相似度从大到小进行排序。将最终答案的集合返回客户端。

4.1.2 基于 ROS 的移动机器人导航技术

基于 ROS 的导航技术，需要用到 ROS 的三个与导航相关的代码包：

(1) `gmapping`：根据激光数据（或者视觉点云数据模拟的激光数据）以及机器人编码器数据来创建地图，该包在环境地图创建模块中使用。

(2) `amcl`：在已知环境下实现自我感知，即定位。

(3) `move_base`：根据起始点位姿、自身定位以及里程计数据进行全局路径规划使移动机器人到达指定的位置，同时在前往途中实现合理避障。

`Gmapping` 和 `amcl` 在本文第三章已经做了详细的说明，在本章中将不再赘述，接下来我们将针对 `move_base` 包中最重要的两个导航算法：全局路径规划算法（A*）和局部路径规划算法（DWA 动态窗口算法）进行简单介绍。

4.1.2.1 全局路径规划算法（A*算法）

A*算法^[56]，是一种在包含若干带有自权重的节点以及路径的图形平面上，求

出最小代价前提下的通过路径。A*算法除了应用于最短路径求解，在其他方面也有着非常多的应用，如搜索问题、游戏中 NPC 的移动计算等。

A*算法融合和广度优先遍历算法（BFS）和迪杰斯特拉最短路径规划算法（Dijkstra）的优点，同时引入启发式搜索以提高计算效率，基于评价函数最终可以确定规划出一条最优路径。它的求解目标就是从源点经过 n 个点到达目标点的距离估计，对应的距离估计函数 $f(n)$ 如下式 4-4 所示：

$$f(n) = g(n) + h(n) \quad (4-1)$$

其中 $g(n)$ 表示的是在图形平面上从初始节点到节点 n 的实际距离， $h(n)$ 表示的是节点 n 到目标节点的最优路径的估计距离。

为了保证可以在保证找到最优路径的前提下实现算法效率的提升，距离估计函数 $f(n)$ 的选取十分重要，换句话说，选择一个合适的 $h(n)$ 函数至关重要。

由于 A*算法是 BFS 算法和 Dijkstra 算法的融合入，选取不同大小的 $h(n)$ 函数最终在算法形态上会导致向其中某一个算法偏离，以下大致分为三种情况：

（1）如果 $h(n) \gg g(n)$ 或 $g(n) = 0$ ，则不计算初始节点到节点 n 的实际距离，只计算节点 n 到目标节点的估计距离，这种情况下，贪心策略的 BFS 算法占主导作用，即算法效率非常高，但是可能计算不出最优路径。

（2）如果 $h(n) \ll g(n)$ 或 $h(n) = 0$ ，则不计算节点 n 到目标节点的估计距离，只计算初始节点到节点 n 的实际距离，在这种情况下，问题转化为单源最短路径规划问题，所以贪心策略的 Dijkstra 算法占主导作用，此时计算量非常大，算法效率很低，但是可以求出最优路径。

（3）如果在 $h(n)$ 数值合理的情况下，还可以分为下面三种情况：

1) 如果 $h(n)$ 小于节点 n 到目标节点的实际距离时，虽然最终能够求解得到最优路径，但是需要计算的节点数量会随着 $h(n)$ 值得减少而增加，导致计算复杂度增大，从而效率降低。

2) 如果 $h(n)$ 大于节点 n 到目标节点的实际距离时，需要计算的节点数量少，故算法效率高，但是很可能得不到最优路径。

3) 如果 $h(n)$ 等于节点 n 到目标节点的实际距离时，此时是最理想的状况，一定能在效率最高的情况下计算得到最优路径。

为了更直观地理解以上三种情况，下面给出各情况下 A*算法的路径规划仿真结果，其中图 4-1、4-2 和 4-3 分别表示 BFS 算法占主导、Dijkstra 算法占主导和 $h(n)$ 选用对角线距离情况下的路径规划结果。

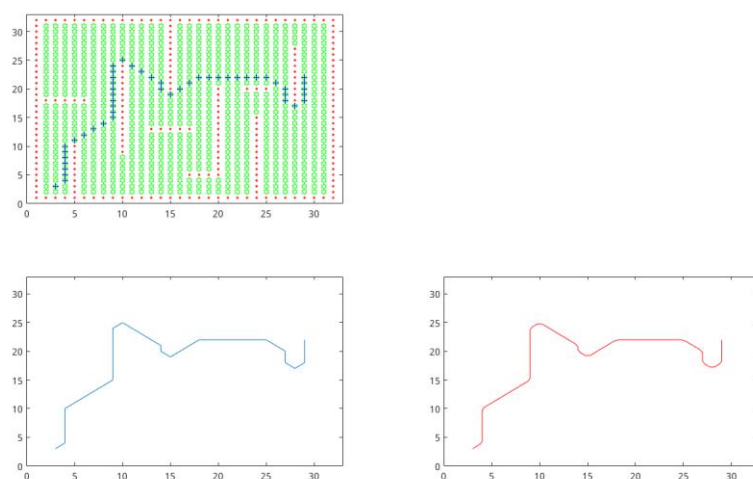


图 4-1 BFS 算法占主导时的路径规划

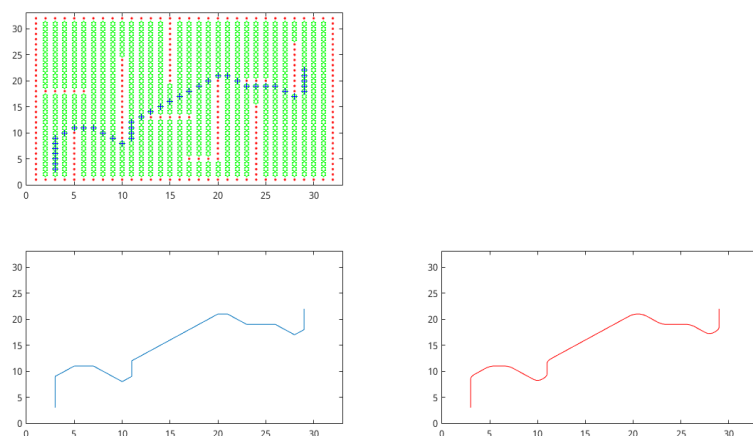


图 4-2 Dijkstra 算法占主导时的路径规划

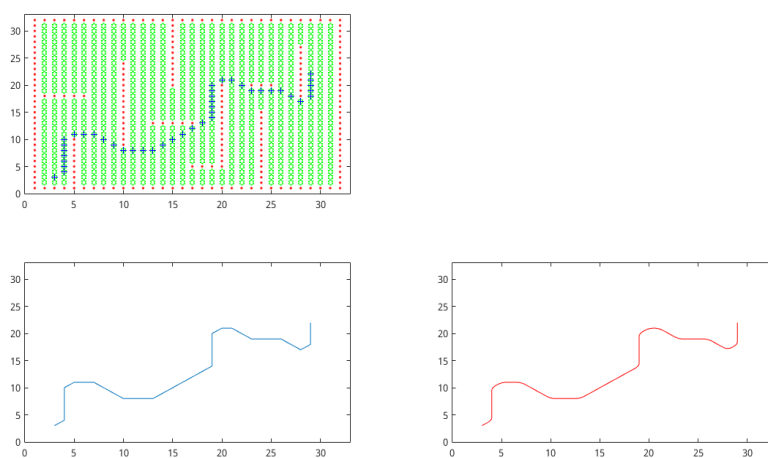


图 4-3 $h(n)$ 选用对角线距离时的路径规划

图中右下角的图是经过平滑处理后的规划路径。BFS 在测试地图中规划 49 步，Dijkstra 算法运行 40 步，A*算法运行 43 步。在运行时间方面，BFS 算法的时间消耗最少，A*算法次之，Dijkstra 算法效率最低，但是 BFS 算法没有得到最优的规划路径，Dijkstra 得到的规划路径最优。综上所述，选取合适的估计函数可以在效率最高的情况下计算得到最优路径。

4.1.2.2 局部路径规划算法（DWA 算法）

本文采用的是 ROS 的 move_base 包下的局部路径规划算法，其内部的实现算法是动态窗口算法（Dynamic Window Approach, DWA）^[57,58]。动态窗口算法的主要思想就是，在速度二维空间 (v, ω) 中，面对无穷多的速度数组，我们可以在一定速度限制的前提下抽取多组速度数组，按照这组速度来预测机器人在 Δt 内的运动轨迹。最后选取合适的评判函数对这些采样速度的轨迹进行评分，其中最合适的轨迹对应的速度数组用于控制机器人进行下一步运动。

步骤 1：机器人运动模型

在取样速度数组以后需要对机器人下一个 Δt 时间内的运动轨迹进行预测，这就需要机器人运动模型来进行计算了。本文移动机器人 FinaRobot 采用的底盘是全向轮运动底盘，机器人本体坐标和世界坐标系转化关系如下图 4-4 所示。

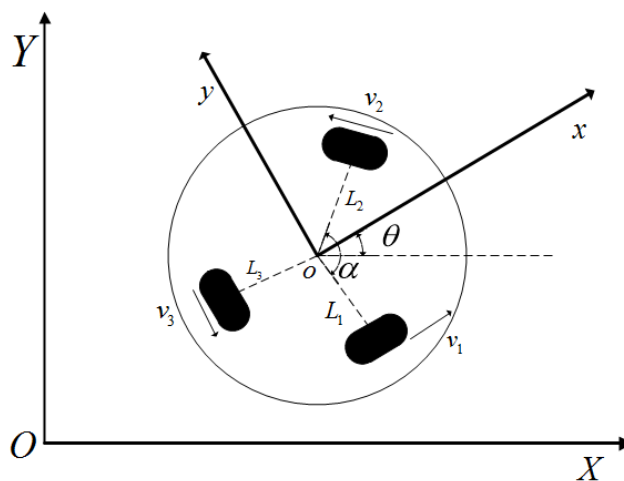


图 4-4 机器人坐标系转化关系

针对机器人不同的运行方式，可以分为以下两种：

- (1) 机器人进行直线运动，即 $\omega=0$

机器人在 Δt 时间内进行直线运动，则只需要将机器人在 x 轴和 y 轴的运动距

离投射到世界坐标系，并与原先位姿进行叠加得到当前位姿，见下式 4-2 至 4-4。

$$x = x + \Delta x = x + v_x \Delta t \cos(\theta) - v_y \Delta t \sin(\theta) \quad (4-2)$$

$$y = y + \Delta y = y + v_x \Delta t \sin(\theta) + v_y \Delta t \cos(\theta) \quad (4-3)$$

$$\theta = \theta + \omega \Delta t \quad (4-4)$$

(2) 机器人运动过程中发生旋转，即 $\omega \neq 0$

在实际情况下，机器人不可能一直按照直线行走，针对速度数组 (v, ω) ，机器人的运动半径可以表示为：

$$r = \frac{v}{\omega} \quad (4-5)$$

机器人位姿表示如式 4-6 至 4-8 所示：

$$x = x - \frac{v}{\omega} \sin(\theta) + \frac{v}{\omega} \cos(\theta + \omega \Delta t) \quad (4-6)$$

$$y = y + \frac{v}{\omega} \cos(\theta) - \frac{v}{\omega} \cos(\theta + \omega \Delta t) \quad (4-7)$$

$$\theta = \theta + \omega \Delta t \quad (4-8)$$

步骤 2：速度采样

在 (v, ω) 的二维向量空间下，可以存在无穷中速度组合方式，显然这种随机采样是不可取的，所以我们根据环境限制和机器人本身的硬件条件限制来缩小速度取样的区域。

1) 机器人本身可以在软件程序内限制其最大或最小运动速度：

$$v_m = \{v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (4-9)$$

2) 机器人的运动速度由电机的功率决定，如果满功率运行或者以最低功率运行，结合机器人自身质量以及电机力矩最终可以得到理想状况下一个可达的速度区间：

$$v_{dyn} = \{(v, \omega) | v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t] \wedge \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t]\} \quad (4-10)$$

3) 由于机器人运行在相对复杂的环境下面，会有一些动态障碍物的出现，这种情况下如果不进行控速，即使有避障措施也有可能因为速度过快而发生碰撞，所以在机器人最大减速基础上，控制机器人在一个合理的速度下运动：

$$v_{safe} = \{(v, \omega) | v \leq \sqrt{2 \cdot dist(v, \omega) \cdot \dot{v}_b} \wedge \omega \leq \sqrt{2 \cdot dist(v, \omega) \cdot \dot{\omega}_b}\} \quad (4-11)$$

$dist(v, \omega)$ 表示在速度数组 (v, ω) 上预测的运动轨迹上避免碰撞障碍物允许的最近距离。

步骤 3: 对采样速度进行评价

根据步骤 2 中得到的采样速度在步骤 1 中机器人运动模型的基础上可以预测出下一时刻机器人的运动轨迹, 该步主要采用评价函数对上述运动轨迹进行评分, 得到最优的轨迹, 即得到下一步的运动速度。一般用于评分的评价函数有如下几种:

(1) 方位角评价函数

在机器人模拟运动到预测轨迹末端时, 机器人的朝向越接近目标位置的朝向, 则该条轨迹的评分越高, 反之, 评分越低。

(2) 空隙评价函数

空隙评价函数可以使用上式 4-11 中的 $dist(v, \omega)$ 函数来充当, 若预测轨迹上面没有障碍物, 则将该函数设为一个常数。

(3) 速度评价函数

该评价函数利用速度变化曲率对预测轨迹进行评估, 设:

$$k = \frac{\omega}{v} \quad (4-12)$$

速度评价函数为:

$$k = \frac{|k_t - k_{last}|}{\sum_{i=0}^n |k_i - k_{last}|} \quad (4-13)$$

其中 n 表示采样速度数量, k_t 为 t 时刻路径的速度变化曲率, k_{last} 表示 $t-1$ 时刻分数最高的轨迹的速度变化曲率。

综合以上三种评估函数, 为其设定依赖参数, 可以得到最终的评估函数为:

$$G(v, \omega) = \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot velocity(v, \omega)) \quad (4-14)$$

如果不对上述评价函数中的三个因子做归一化处理, 可能会导致评价函数不连续, 或者某一个因子在评价函数中占过大的比重。为了公平起见, 对上式 4-14 进行平滑处理, 处理过程如下式 4-15 至 4-17:

$$n_heading(i) = \frac{heading(i)}{\sum_{i=1}^n heading(i)} \quad (4-15)$$

$$n_dist(i) = \frac{dist(i)}{\sum_{i=1}^n dist(i)} \quad (4-16)$$

$$n_velocity(i) = \frac{velocity(i)}{\sum_{i=1}^n velocity(i)} \quad (4-17)$$

其中 n 为采样速度的个数, i 表示第 i 条轨迹。

针对动态窗口算法进行仿真, 如下图 4-5 所示, 在 10×10 的地图范围内, *代表的是起点到终点的路径中出现的障碍物, 动态窗口算法的目的就是要合理地规避这些障碍物到达终点。其中蓝色曲线表示的是行进路线, 绿色小箭头表示的是采样速度样本。

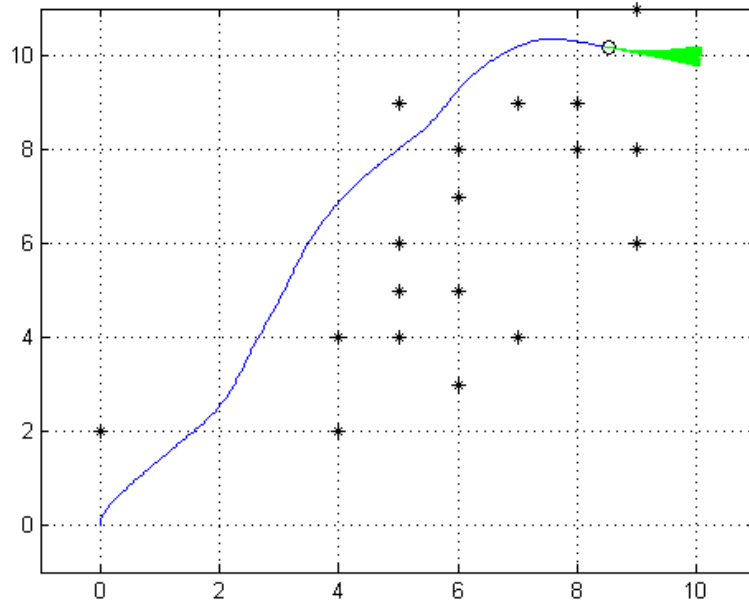


图 4-5 动态窗口算法仿真结果

4.2 语音交互导航系统总体设计

语音交互导航系统主要包含三大部分: 语音识别与合成模块、问答系统和导航系统。

用户在完成语音输入后, 经过语音识别模块将语音翻译为文本格式, 接着以该文本为问题输入问答系统, 经过一系列的自然语言处理步骤最终得到一个合适的答案 (比如国资处的位置是 B2601), 接着由用户开启导航模块, 将要去的位置下发到 MiniPC 上, 转化为在栅格地图上的位姿信息, 以此位姿作为导航系统的目的地。在导航结束到达目的地后, 通信系统会发送指令到客户端, 最后通过语音

合成模块转述给用户。

语音交互导航系统的设计框图如下图 4-6 所示：

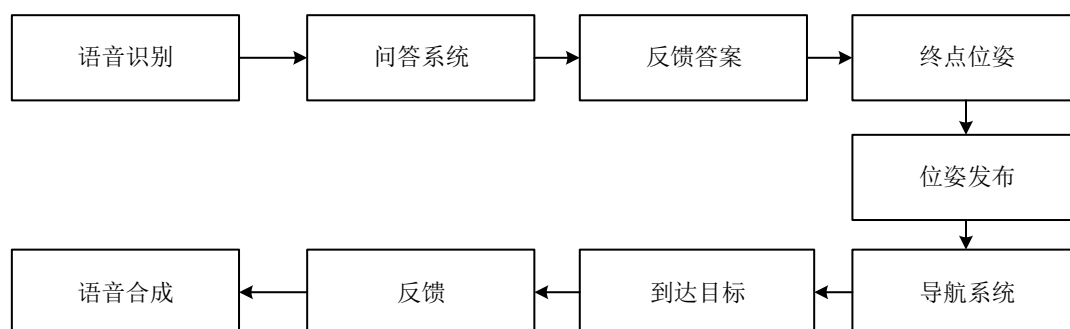


图 4-6 语音交互导航系统设计框图

4.2.1 在线问答系统

在线问答系统运行环境为安卓平板，其采用的是 C/S（客户端/服务器）架构，总体架构如图 4-7 所示：

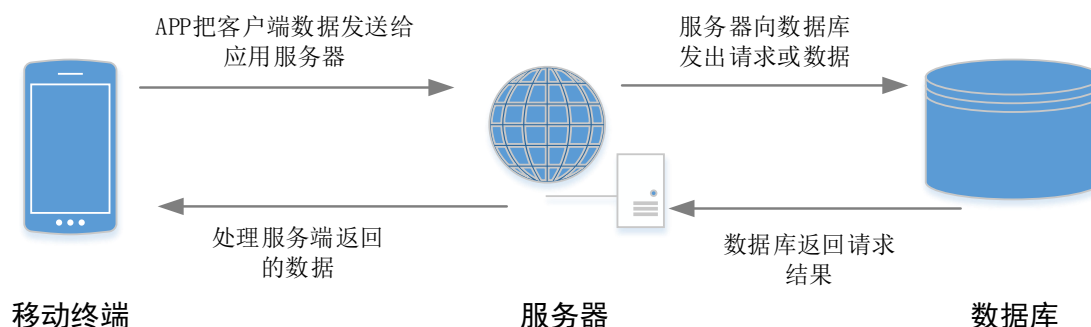


图 4-7 在线问答的 C/S 架构

客户端：向服务器发出数据请求信息，通常是将表单中的信息封装为 JSON 对象，通过 URL 请求提交到服务器。

服务器：本文采用 php 来编写后台程序，处理请求和相关数据，连接数据库，并操作数据库的查询和修改操作，得到返回数据封装成 JSON 格式返回客户端。

数据库：依照服务器的请求信息对数据库中的数据进行对应处理（增删改查）。

FinaRobot 的财务在线问答系统业务流程图如图 4-8 所示。

（1）中央控制单元

该模块主要包括分词、数据库操作、关键词匹配三部分功能

分词：调用 jieba 分词算法、停用词删除和词性标注将主程序接收的字符串分解成可以在数据库中进行匹配查找的关键词组。

数据库操作：主控程序将关键词组传送给服务器，完成关键字匹配后由服务

器返回问题答案，并存储无法解决的问题到数据库中。

关键字匹配：利用余弦相似度匹配，根据关键词组在知识库中匹配出对应的问题答案（可包含多个答案）。

（2）在线问答系统

在线问答系统主要包括三大子模块：财务事务办理、财务地点查找、财务处员工查找。财务处员工查找主要包括显示员工头像、姓名、科室、职务、办公地点、在岗状态、办公电话等。财务地点查找主要包括科室名称、序号以及楼层平面图，最后由机器人带领前往。财务事务办理主要包括对于问题的文字描述、语音回答、办事流程图以及小视频的演示等等。对于无法查询到匹配答案的问题，我们应把该问题放在无效提问表中。答案展示完成后系统会收集用户对该答案的满意度，以此来设定该答案的权重，为后续系统提供精准的答案做好铺垫。

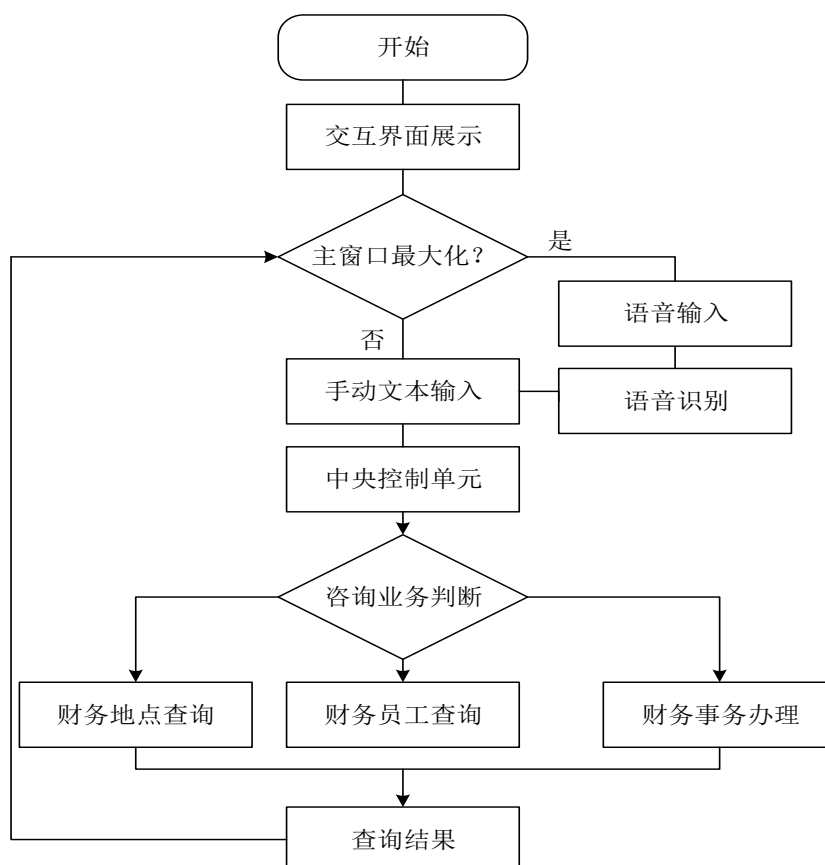


图 4-8 财务在线问答系统业务流程

由于财务在线问答中员工查找和财务事务办理不是本文的重点研究内容，故不做介绍，下面主要针对财务处地点查找做详细说明。下图 4-9 为财务处地点查找的具体流程。

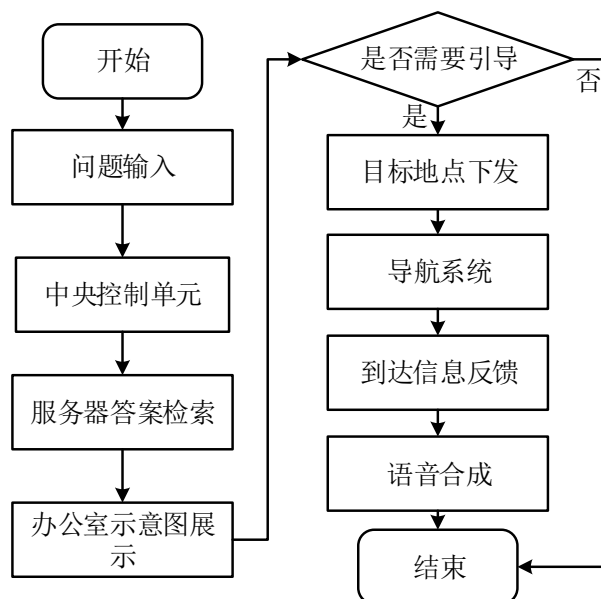


图 4-9 财务地点查找具体流程

在财务在线问答系统中，提问“带我去国资处”，经过自然语言处理系统得到该问题答案，返回客户端，在界面上显示办公室地图，点击“带我去”按钮，可将导航指令以及目的地址发送到 MiniPC。至此客户端的工作已经告一段落。

当 MiniPC 向客户端发送当前机器人所处位置以及到达终点指令时，客户端将该信息通过语音合成反馈给用户，已达到良好的交互体验。

4.2.2 基于 ROS 的导航系统

本文基于 ROS 环境设计了一套的导航系统，该系统主要包括通信模块、目标位姿自动发布模块和 move_base 导航模块。其总体框架如图 4-10 所示，其中导航系统部分关键代码如图 4-11 所示。

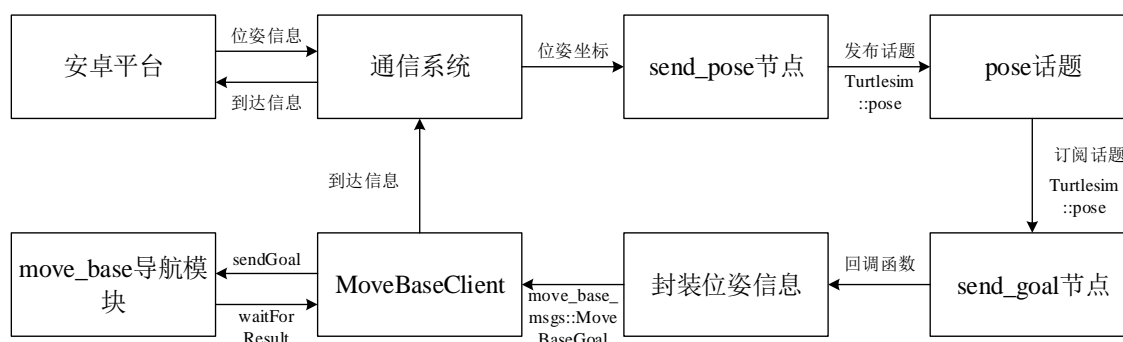


图 4-10 基于 ROS 的导航系统总体框架

```

void sendToMoveBase(const Turtlesim::Pose &pose){
    MoveBaseClient ac("move_base", true);
    ac.waitForServer(ros::Duration(60));

    move_base_msgs::MoveBaseGoal goal;
    goal.target_pose.header.frame_id = "map";
    goal.target_pose.header.stamp = ros::Time::now();
    goal.target_pose.pose.position.x = pose.x;
    goal.target_pose.pose.position.y = pose.y;
    goal.target_pose.pose.orientation.w = pose.theta;

    ac.sendGoal(goal);
    ac.waitForResult();

    if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
        ROS_INFO("You have reached the goal!");
        ReturnToClient();
    else
        ROS_INFO("The base failed for some reason");
}

```

图 4-11 导航系统部分关键代码

在安卓平台上查询得到的位置信息，通过安卓平台和 MiniPC 之间的 Socket 通信系统发送，并将位置信息转化为对应地图下的坐标信息（包含 x、y 和 theta），由 send_pose 节点将该坐标信息发布到 pose 话题。send_goal 节点则订阅了 pose 话题，一旦在 pose 话题中取到位姿信息，便会执行回调函数，在该回调函数中首先会将目标坐标封装成可被 move_base 接受的 move_base_msgs::MoveBaseGoal 类型数据，接着创建 action 客户端，通过该客户端将目标信息发送给 move_base，并且调用 waitForResult 函数阻塞当前程序，直到机器人到达目标地点，返回到达信息。最后通过通信系统将到达信息返回安卓客户端。

move_base 导航模块总体框架如下图 4-12 所示：

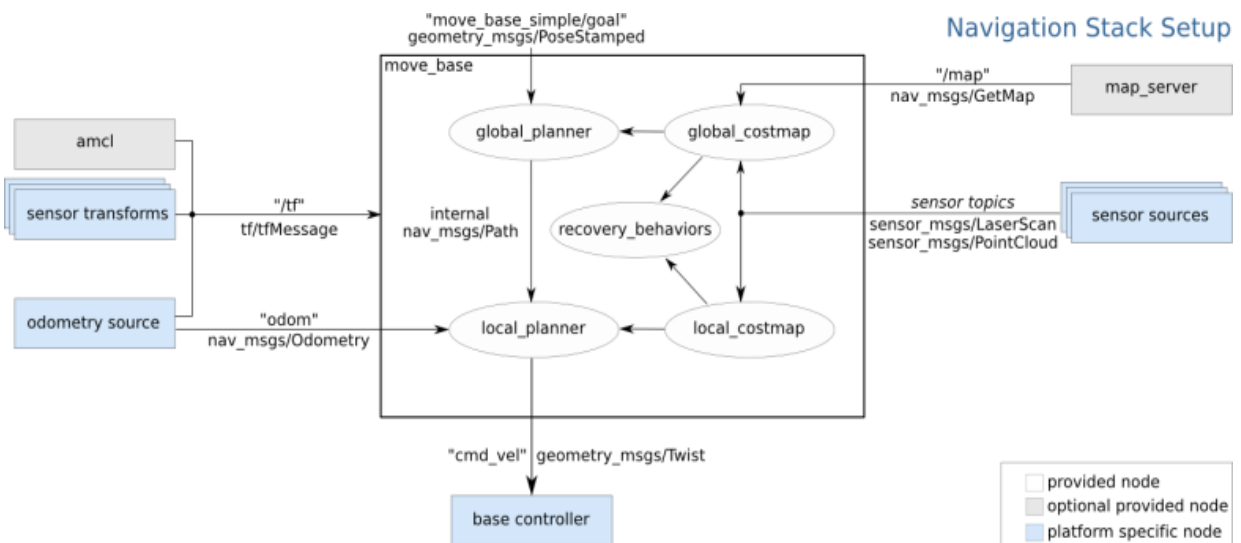


图 4-12 move_base 导航模块的总体框架

`move_base` 包需要外部提供的组件有：坐标转换、传感器信息、里程计信息、定位信息和地图信息，在以上组件基础上确定起点终点进行路径规划，接着控制机器人按照指定速度运动。`move_base` 包中最主要的两个模块为全局路径规划（`global planner`）和局部路径规划（`local planner`）。

4.2.2.1 全局路径规划流程

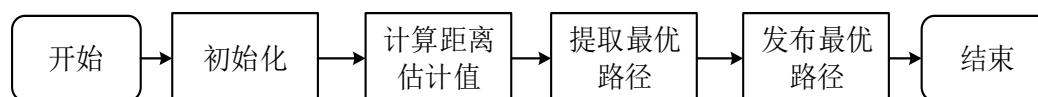


图 4-13 全局路径规划流程

在 `global_planner` 节点函数中，首先加载 `BaseGlobalPlanner` 类，为其创建实例，并进行初始化，接着调用 `setParameter()` 函数进行参数设置。完成上述操作后便开启 `makePlanService` 服务监听路径规划请求。监听到请求之后就调用 `makePlan()` 函数来执行全局路径规划的核心步骤，主要包括计算距离估计值、提取最优路径、发布最优路径。

步骤 1： 计算距离估计值。栅格地图中的每个点相对于一个确定的起点和终点都能得到一个距离估计值，该距离估计值由上式 4-4 计算得到。计算距离估计值的过程如下：

- （1）新建一个队列 `queue`，将起点 `push` 进去；
- （2）将地图上的所有点的距离估计值都设为 `MAX`，其中 `MAX=9999`。将起点的距离估计值设为 0；
- （3）得到当前最小的距离估计值所在的节点，如果该节点指向目标节点，则结束循环。
- （4）针对（3）中得到的最小距离估计值所在的节点，对它上下左右四个节点调用 `add()` 函数，该函数的作用就是利用式 4-4 分别计算最小估计值。并将结果存入 `queue` 中。需要注意的是，邻近的这四个点，如果已经添加则不再重复计算，如果是障碍物则忽略。

（5）如果队列大于 0 且循环次数小于节点总数的两倍。继续执行（3）-（5）。

步骤 2： 提取最优路径，在步骤 1 计算得到所有节点的距离估计值队列的情况下，从中提取出一条最短路径，提取最优路径的步骤如下：

- （1）将目的地点所在位姿加入最优路径中。当前点为目标节点。
- （2）从当前点的上下左右四个节点中找出距离估计值最小的那个节点，加入最优路径中。当前点更新为该节点。

(3) 如果当前点不是起始节点, 继续执行 (2) - (3)

步骤 3: 发布最优路径

根据步骤 (2) 提取得到最优路径, 最终将该路径发布到 ROS 节点中。可以通过 rviz 来查看该路径。

4.2.2.2 局部路径规划流程

ROS 中局部路径规划的具体实现步骤如下:

(1) 初始化。加载 DWAPlannerROS 类, 并创建实例, 调用其初始化函数, 并通过该实例获取机器人当前位姿等信息。同时创建 DWAPlanner 实例, 进行参数配置, 为局部路径规划做好准备。

(2) 进行速度采样。在 move_base 包中经过一系列的函数调用, 最终调用 SimpleTrajectoryGenerator 类的 initialise() 函数完成速度采样。在 $[v_{\min}, v_{\max}]$ 的速度区间内, 从最小速度每次多累加一次步长即为一个速度样本, 直到达到最大速度。获取每个维度的速度样本。最后将速度样本保存到 vel_sample 结构体中。

(3) 给每个预测轨迹打分。对每个速度数组调用 scoreTrajectory 函数进行评分, 最终得到评分最高的速度样本。

(4) 将 (3) 中得到的速度发布给 ROS 节点, 机器人运动控制程序可以订阅该节点得到下一步运动的策略。

4.3 本章小结

本章结合语音识别、自然语言处理和 ROS 导航三大技术设计了一套基于语音交互的移动机器人智能导航系统。首先介绍了上述三大关键技术中的自然语言处理和 ROS 导航技术, 对其原理进行了详细说明。接着介绍了该系统的完整业务流程。最后针对系统中的每个关键技术工程中涉及到的模块进行详细说明。

第五章 基于突变点检测的自动充电系统

当室内移动机器人在电量不足的情况下，需要有一定的低电量保护措施来让其自动转换到低电量模式，防止在后续服务过程中突然断电停机，影响使用感。在低电量保护模式下，机器人会开启自动充电远程导航模块抵达充电区域，接着采用近程对接技术完成精准对接后，开始充电。当机器人电量达到一定阈值后，重新开启远程导航模块，回到开始的地点，继续执行任务。目前看来，市面上大多数的移动机器人的供电系统都由机器人随身携带的蓄电池来提供支持，一般来说这种电池在不间断工作几个小时后便需要再次充电。因此为移动服务机器人设计一套自动充电系统^[59]解决充电问题已经迫在眉睫。

5.1 自动充电关键技术

针对当前机器人自动充电的主要操作步骤和执行流程，有以下几个主要技术点需要得到解决：

（1）通过电量检测技术，判断是否缺电

涉及到的电量检测技术主要包括实时监测电源电压，经过 ADC 芯片转化为电量，或者按照机器人使用时长采用经验来判断，另外可以采用精准物理计算方式得到机器人电能使用数据从而得到电量情况。通过上述检测得到电量剩余情况才能判断机器人是否处缺电。

（2）自动充电远程导航技术

机器人需要借助导航技术从当前点规划一条最短路径回到充电点，同时在复杂环境下，遇到障碍物，需要通过避障技术绕过它。

（3）自动充电近程充电桩对准技术

现有的充电桩对接技术主要是在充电桩上或者周围安装便于传感器进行识别的特殊颜色或者形状的物体。如：便于视觉处理的特殊颜色物体或者光源、便于激光雷达或超声波探测的特殊形状物体等。

（4）机器人充电感应头和充电桩自动连接和断开

目前实现该技术主要采用微调机器人位姿，或者在充电感应头机械设计上可以实现自动伸缩功能。考虑到前面的流程中均存在一定的误差，所以我们设计的充电头和充电桩在对准的过程中必须要有**一定的误差容忍度**。方法，机器人垂直对准，充电桩水平对准，然后开电

（5）检测充电感应头和充电桩的物理连接情况

感应头和充电桩必须先建立物理连接，否则无法在电路上建立连接，从而导

致无法完成充电，所以完成对接后首先要进行物理连接的判断。判断方法有两种，物理测量和电气测量，即利用红外收发设备来检测和利用电压荷载情况来判断。

(6) 误差纠正系统

在非理想的状态下，机器人在经过自动充电远程导航后很有可能根本发现不了识别目标，进而无法进入近程对接过程，或者发现目标识别物之后无法实现费解，基于以上情况，移动机器人本身需要提供一套误差纠正系统来应对相应问题。

上述关键技术（4）、（5）和（6）与本文的导航系统没有太大的相关性。故本章只就前二点关键技术做详细的讨论与设计。

自动充电系统实现远程导航的方案很多，主要包括远距离的红外光引导方案、视觉跟踪方案、基于栅格地图的导航和避障方案。对于近程对接部分现在主流的方法大多数都是在充电座上方或者周边放置容易被传感器(如超声波、激光探测仪或摄像头等)识别到的特殊形状（有规律的四凸物体）或者颜色标记（反射带、光源等）来完成充电桩的识别和精准对接过程。很多自动充电系统都会融合两种以上的传感器来实现自动充电技术。本文将只采用激光雷达测距仪结合机器人自身的里程计信息，完成自整个自动充电过程。

5.2 基于突变点检测的充电桩识别算法

基于突变点检测的充电桩识别算法的基本思想：从集合 R 中筛选出符合设定的突变点变化规律的角度范围，对左右边界角度取平均作为中心角度，以该角度向前行驶，重复上述步骤，直到中心角度的绝对值小于 5° ，结束校正过程，向前行驶对接。具体算法步骤如下：

对于给定输入的激光数据数组 $Rangs[N]$ ，其中 $N=360$ ，即激光雷达绕过一度便产生一个距离信息，激光数据集合表示如下：

$$R = \{r_1, r_2, \dots, r_i\}, i \in [0, 180) \cup [-180, 0) \quad (5-1)$$

其中 r_i 表示的是激光雷达第 i 个角度的距离信息。

步骤 1：数据筛选，将在近程对接范围外或者无效的激光测距数据置零，筛选规则如下：

$$r_i = \begin{cases} 0, & r_i = \text{Inf} \text{ 或 } r_i > 2.1 \\ r_i, & \text{其他} \end{cases} \quad (5-2)$$

其中 Inf 表示无效的激光雷达数据，2.1 米是判断机器人进入近程对接范围的阈值。

步骤 2：数据补全，对于一些由于激光雷达本身误差导致的无效距离，采用人工补齐的方式将其复原，防止硬件误差带来突变点的错误标定，补全规则如下：

$$r_i = \begin{cases} (r_{i+2} + r_{i-1})/2, & r_i=0 \text{ 且 } r_{i+1}=0 \text{ 且 } |r_{i+2} - r_{i-1}| < 0.01 \\ (r_{i+1} - r_{i-1})/2, & r_i=0 \text{ 且 } |r_{i+1} - r_{i-1}| < 0.01 \\ r_i, & \text{其他} \end{cases} \quad (5-3)$$

本文采用的是跨越式补全法，如果无效距离的左右两个距离差的绝对值小于 0.01，则将该角度的距离赋值为左右两个角度距离的平均值。

步骤 3: 突变点标定，标定集合 *record* 定义如下：

$$Record = \{rd_1, rd_2, \dots, rd_i\}, i \in [0, 180) \cup [-180, 0) \quad (5-4)$$

其中 rd_i 表示第 i 个角度的激光雷达测量距离是否发生突变，其初始化值如下：

$$rd_i = 0, i \in [0, 180) \cup [-180, 0) \quad (5-5)$$

扫描激光数据集合，对突变点进行标定，标定结果存入 *Record* 中，具体标定过程如下：

$$rd_i = \begin{cases} 1, & (r_i - r_{i-1}) \geq 0.01 \text{ 且 } (r_i - r_{i-1}) \leq 0.02 \\ -1, & (r_i - r_{i-1}) \geq -0.02 \text{ 且 } (r_i - r_{i-1}) \leq -0.01 \\ 0, & |r_i - r_{i-1}| < 0.01 \end{cases} \quad (5-6)$$

其中 0.01 和 0.02 是根据特殊形状标定物的突起部分和凹陷部分的高度差 0.012 给出的一个突变范围，在该范围内的突变都可以认为是检测到标定物的特征。

步骤 4: 目标位置识别，根据上一步中得到的突变标定集合，判断连续突变点个数符合标定物特征的角度区间，区间左右边界分别表示标定物的左右边界角度。

$$\{m=i, n=j\}, |rd_i, rd_j| = 1 \text{ 且 } count \geq 3 \text{ 且 } count \leq 5 \quad (5-7)$$

其中 *count* 表示如下式 5-8 所示：

$$count = \sum_{k=i}^j |rd_k| \quad (5-8)$$

count 为 $[i, j]$ 区间内突变点的个数，由于标定物一共包含 5 个突变点，其中中间三个突变点比较容易被标定，为了容许一定的误差存在，故若检测到三个突变点以上便可确定该标定物位置。

最后取两个边界角度的均值表示为充电桩的中心位置：

$$target = (m + n) / 2 \quad (5-9)$$

步骤 5: 判断 $|target| < 5^\circ$ ，若成立，则退出识别算法，机器人向前移动完成对接；若不成立，则重复步骤 1-5。

5.3 自动充电系统整体设计

自动充电系统在执行步骤上可以分为二步：第一步，电量检测和反馈。第二步，自动充电远程导航。第二步，充电桩识别和对接。除了电量反馈信息的展示部分运行在安卓客户端，其他部分均在 MiniPC 上实现。

该系统的整体业务流程如下图 5-1 所示。

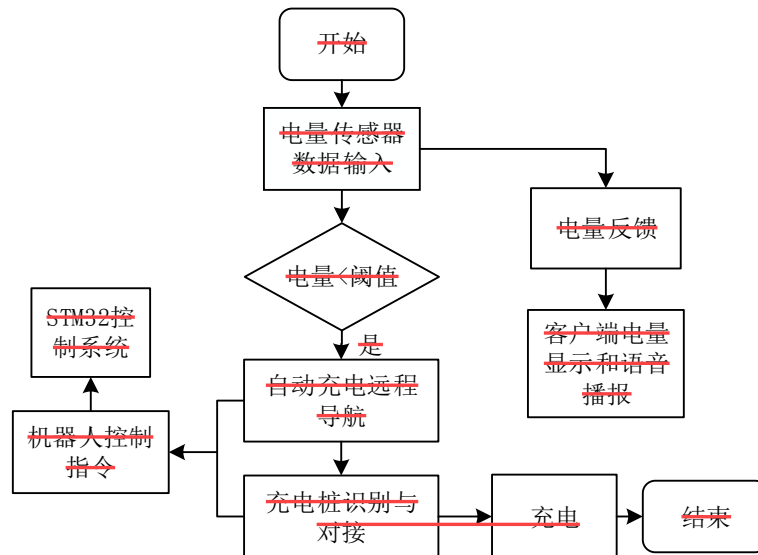


图 5-1 自动充电系统业务流程

5.3.1 电量检测和反馈

本文电量检测模块主要通过检测电池电压，经过 ADC 芯片将模拟信号转化成数字信号，最终得到电池当前的剩余电量。

电量反馈模块则是实时获取检测模块获取的电量，并将该信息和其他传感器信息打包，通过 Socket 通信方式发送给安卓客户端。

安卓客户端在接收到 MiniPC 发送的传感器数据包后，按照本文规定格式解析，得到电量数据，在界面上显示。另外，针对电池电量低于 20% 的情况，客户端会自动提醒用户，尽快完成当前咨询内容，以免机器人进入低电量保护模式。具体流程见下图 5-2。

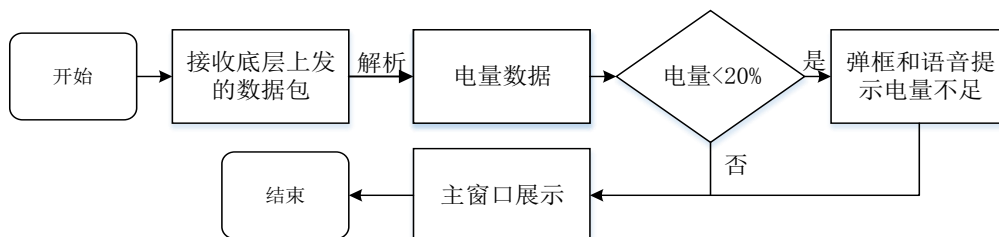


图 5-2 电量检测和反馈流程

5.3.2 自动充电远程导航

充电桩远程导航具体业务流程如图 5-3 所示：

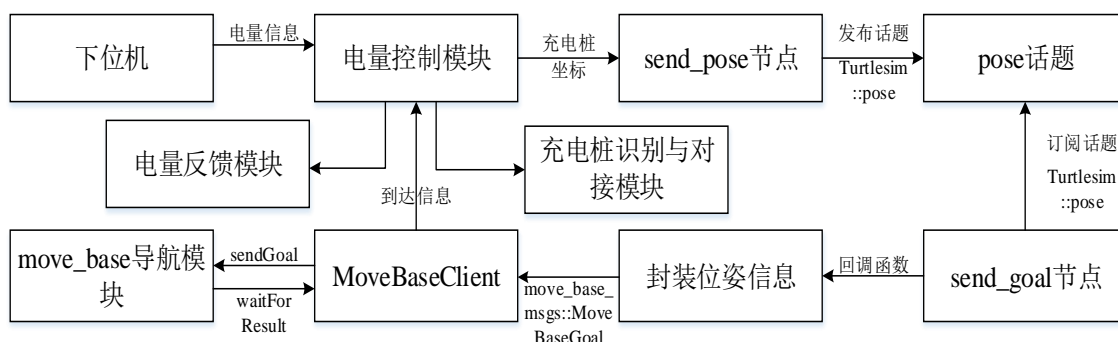


图 5-3 充电桩远程导航业务流程

在机器人开展服务工作前需要先确认该环境地图下充电桩的具体位姿和方向，所以需要使用 `amcl` 包对充电桩的位置和机器人正常充电的方向记录下来。该位姿和方向将作为机器人充电远程导航的目的地点。

电量控制模块每 10 分钟会从下位机获取当前电量信息，并将该信息通过通信系统上传到客户端，同时判断电量是否低于阈值，若是，则将充电桩坐标通过 `send_pose` 节点发送到 `pose` 话题，后续便是与语音导航系统共用的导航技术部分。当远程导航到达目标位姿，返回到达信息到电量控制模块，接着启动充电桩识别与对接模块。

另外，针对该系统还有相应的误差纠正系统。在开启充电桩识别模块之后若在规定范围内未识别到充电桩，则判断为远程导航失败，则返回重新进行远程路径规划。

5.3.3 充电桩识别和对接

充电桩识别过程中，用于传感器识别的特殊形状物体（有规律的凹凸物体）放置在充电桩上方，机器人依据激光雷达测量得到的距离信息来识别充电桩的具体位姿，然后反复调整机器人自身相对充电桩的位置和角度，最后进行对接。

本文采用的是基于突变点检测的充电桩近程对接技术，根据激光测距仪对特殊形状的物体测距会呈现特殊的距离突变规则来判断充电桩的精确位置。特殊形状物体如图 5-4（a）所示。为保证实验过程中能精确地确定充电桩的位置，特殊形状物体的安装位置应与激光雷达在一个水平面上并且在垂直方向上与下方的充电桩中心对齐，具体安装方式如图 5-4（b）所示。

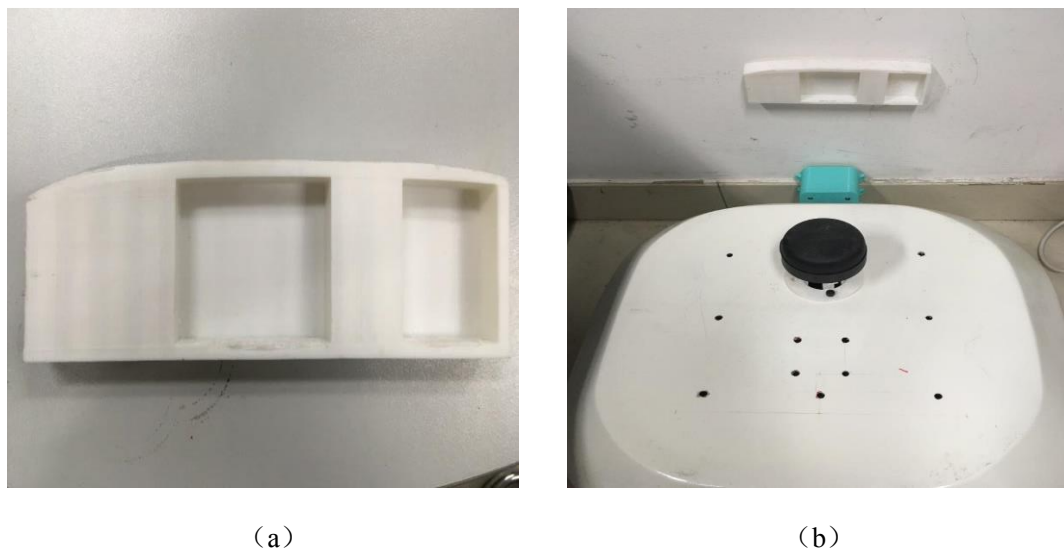
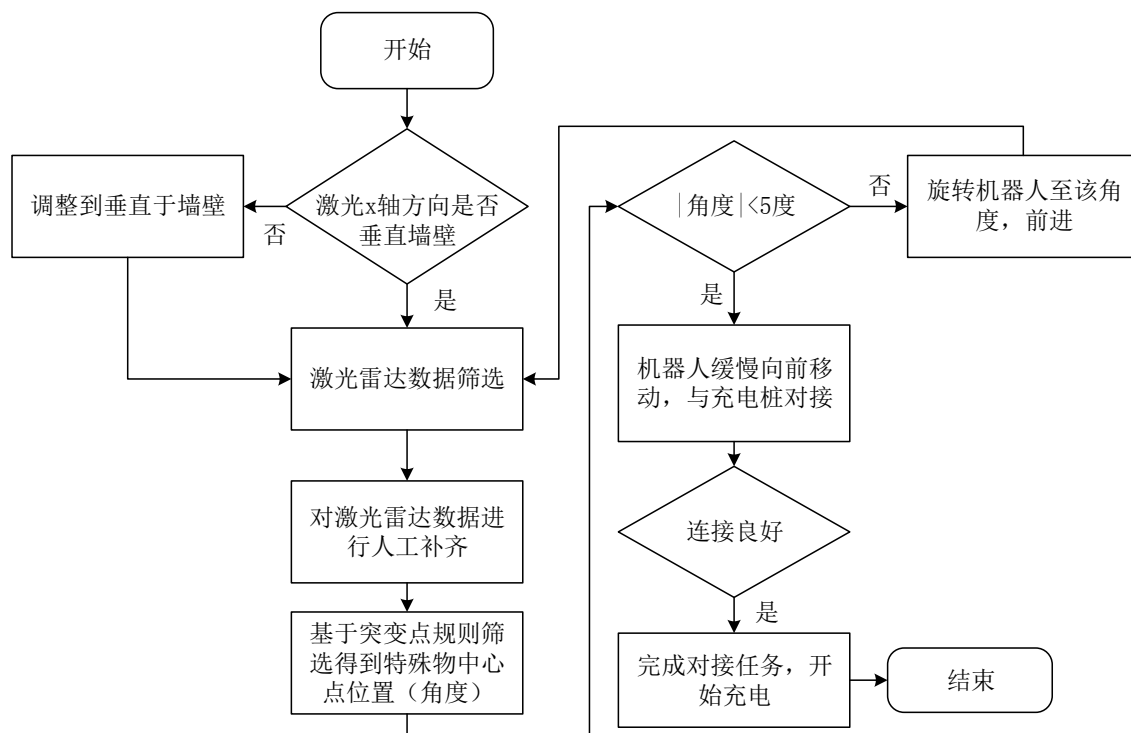


图 5-4 标定物及安装方式 (a) 标定物体; (b) 安装方式

基于突变点检测的充电桩近程对接的具体流程如图 5-5 所示。具体步骤如下:

步骤 1: 判断激光雷达 x 轴是否垂直于墙壁。

在进行自动充电模块的第一步就进行充电桩位置和方向的标定, 该方向就是垂直于墙壁, 由于近程对接过程中对角度的要求较高, 所以在机器人完成远程对接任务后需要判断当前机器人方向是否垂直于墙壁, 以便进行后续操作。



5-5 充电桩检测和对接流程

步骤 2: 激光雷达数据筛选。

从激光开始角度到结束角度的角度范围内，每个角度增量都对应一条距离信息保存在 `Ranges[]` 数组中，该数组中出现的无效距离（小于最小测距范围或者大于最大测距范围）以及大于 2.1m 的距离都将其做相应的剔除。

步骤 3: 对激光雷达的距离数组进行人工补齐。

对于一些由于激光雷达本身误差导致的无效距离，需要采用人工补齐的方式将其复原。本文采用的是跨越式补全法，如果无效距离的左右两个距离差的绝对值小于 0.01，则将该角度的距离赋值为左右两个距离的平均值。

$$ranges[i] = (ranges[i-1] + ranges[i+1]) / 2 \quad (5-10)$$

步骤 4: 基于突变点规则筛选出特殊物中心点位置。

本文中特殊物的突变点为 5 个（包括与墙壁的衔接处），所以只要筛选出五个连续的突变点即可判断出特殊物体的大概位置，此时将左边界和右边界的角度记录下来分别为 `m` 和 `n`，特殊物体中心点的位置粗略的标定为 `target=(m+n)/2`。该角度小于 5 度则表示机器人在这个角度下可以误差允许范围内对接充电桩，无需再做调整即可前进对接；若大于 5 度则继续进行上述调整。如下图 5-6 中（a）、（b）和（c）所示。

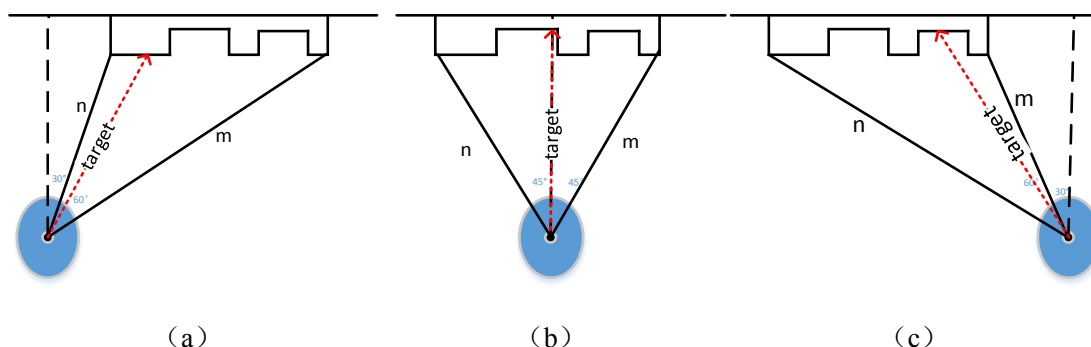


图 5-6 角度调整过程（a）左侧；（b）垂直；（c）右侧

机器人的充电连接有两种方式：接触式和非接触式。本文采用的连接方式是接触式，但是传统的的充电感应头和充电桩的设计使得两方的接触面积非常小，导致需要在对接技术及其精确的条件下才能实现对接，这在非理想状况下显然是不合理的。本文采用的改进的充电感应头可以允许机器人在一个较大的范围内进行充电接入而不用担心定位不太精确导致的无法实现物理连接的问题。这种设计使得两端对接方式相对传统方式简单很多，同时可以接受 $\pm 5^\circ$ 和 $\pm 5\text{cm}$ 的对接误差。

充电感应头具体示意图如图 5-7 所示，在机器人机身前端上下安装两条圆弧状的铜条，铜条长度为 10cm，充电桩上同时安装了两个弹性的金属接触点，如果移

动机器人在对接过程中没有实现控速，那么该设计便可以在对接过程中起到一定的保护作用。

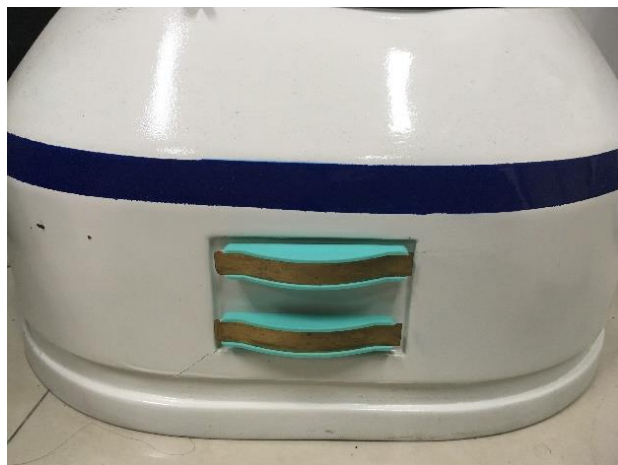


图 5-7 改进充电头安装示意图

5.4 本章小结

本章在开始对自动充电的关键技术做了简单介绍，接着结合本文在远程导航技术上现有的技术成果以及硬件设备（激光雷达）提出了一种基于突变点检测的充电桩识别算法。针对该算法完成了代码的编写。最后设计了一套自动充电系统，并阐述了该系统的详细流程，对于该系统中的三大模块：电量检测和反馈模块、自动充电远程导航模块和充电桩识别和对接模块分别进行了详细的说明。

第六章 移动机器人导航系统实验与测试

本系统的实现目的是为了将财务处的员工从每日重复且繁琐的基本财务问题咨询的处境中解放出来，大大减少学生和教师在咨询或者办理财务事务的时间，同时增加整个过程的趣味性。为实现上述目的，必须为机器人 FinaRobot 设计一套导航系统，方便在机器人处于低电量模式的情况下尽快完成自动充电，以保证机器人每天 8 小时的服务时间内保持电量充足。另外，在师生咨询地点相关问题时能够提供一套交互良好的引导系统，基于该需求，本文设计了一套基于语音交互的智能导航系统。该系统的主要功能如下：环境地图创建、语音导航和自动充电。详细的功能清单如下表 6-1 所示：

表 6-1 机器人功能清单

编号	所属模块	功能描述
1	环境地图创建	在粒子滤波优化基础上，创建一张较理想的实验环境地图
2	自主充电	电池电量实时上报客户端
3		低电量情况下开启远程路径规划，规划回到充电桩的最优路径
4		机器人具备局部避障功能
5		在靠近充电桩附近能实现近程的充电桩对准
6	语音导航	具备与服务器和 MiniPC 进行通信的功能
7		能实现与用户进行语音对话且保留手动文本输入功能
8		能根据用户提出的问题反馈合理的答案
9		能在用户提出带领要求后，带领用户到达准确位置
10		能在到达目的地后语音提醒用户

6.1 机器人环境地图创建实验与测试

在本文第三章中介绍了一种改进重采样策略的 RBPF-SLAM 算法，针对该优化的 SLAM 算法对 Gmapping 算法做了相应的修改，下面完整介绍 SLAM 建图的步骤：

步骤 1：连接激光雷达并获取数据

(1) 首先用一根 Micro-B 转 Type-A 的 USB 线缆将雷达与电脑相连（安卓数

据线)。

(2) 使用 `ls -l /dev |grep ttyUSB` 检查雷达的端口号。

(3) 给雷达的端口号添加写的权限, 如 `sudo chmod 666 /dev/ttyUSB0`。

(4) 通过打开 `rplidar` 的启动文件, 我们可以在 `rviz` 仿真器中直观地看到激光雷达数据。命令: `roslaunch rplidar_ros view_rplidar.launch`。

注意: 在安装激光雷达的时候尽量保持坐标系方向和机器人坐标系方向一致, 以减少在 `tf` 树中的转换。

表 6-2 激光数据 (sensor_msgs/LaserScan) 格式

属性	数据类型	作用
header	Header	记录激光数据坐标系转换、时间戳等信息
angle_min	Float32	激光数据开始角度
angle_max	Float32	激光数据结束角度
angle_increment	Float32	角度增量
time_increment	Float32	时间增量
scan_time	Float32	激光数据记录时长
range_min	Float32	最小测量范围
range_max	Float32	最大测量范围
ranges	Float32[]	Ranges[i]表示第i束激光测量到的距离 Range_min<=values<=range_max
intensities	Float32[]	强度信息

步骤 2: 将激光数据发布到 ROS 话题

下载激光雷达自带的发布包, 放到工作空间下, 编译之后执行 `rplidar.launch` 启动文件即可将激光数据 (sensor_msgs/LaserScan) 发布到 `scan` 话题, 需要的程序只需订阅该话题即可获取激光数据。

通过以上步骤我们可以获取到原始的激光数据, 如果要用这些激光数据建图还需要修改节点文件, 比如, 剔除掉某个角度范围的数据(默认取得是 0-360)。在我们的工程中不对节点文件做修改。

还有个非常重要的步骤是将 `laser_frame` 和 `base_link` 对应起来, 所以在启动文件中需要做些修改。

需要添加一个节点来实现 laser_frame 和 base_link 坐标系的对应。`<node pkg="tf" type="static_transform_publisher" name="base2laser" args="0.37 0.0 0.0 0.0 0.0 0.0 base_link laser 100"/>`，此处参数的转化关系需要依照真实实验环境下的激光雷达的安装位置来决定，比如，我们的激光雷达是安装在机器人本体坐标系 x 轴正方向 0.37m 的位置，两个坐标系的方向一致，所以只需要加上 x 轴的增量即可。在本文的工程中激光雷达所在坐标系与机器人本体坐标系是完全重合的，所以省去了坐标转化这一步。

步骤 3：发布里程计数据

MiniPC 通过串口，以 50ms 的频率从下位机获取里程计数据，数据结构如表 6-3 所示：

表 6-3 里程计数据结构

encoder1	int	轮 1 里程和
encoder2	int	轮 2 里程和
encoder3	int	轮 3 里程和

MiniPC 内创建节点，实时接收下位机发送的里程计数据，通过转化公式将获取的里程计数据转化后分别发送到 odom 节点（消息类型为 navs_msgs/odometry）和 tf 树（实现 odom_frame 和 base_link 坐标系的连接）。

（1）引入两个重要的包：

- 1) `<depend package="tf"/>`
- 2) `<depend package="nav_msgs"/>`

（2）计算当前位姿和当前速度

- 1) 获得三轮距离增量。
- 2) 获得机器人转角增量和当前角度。
- 3) 获取 XY 方向上的位姿增量 deltaX 和 deltaY。
- 4) 得到当前机器人所在的位姿和角度。
- 5) 得到当前机器人 XY 方向的速度和角速度。

（3）发布 tf 消息

更新 geometry_msgs::TransformStamped 消息，消息内容包括机器人当前位姿和角度，并通过 tf 树广播。

（4）发布里程计消息

更新 navs_msgs::Odometry 消息，消息内容包括当前位姿、角度和速度，并发布。

步骤 4: 运行 Gmapping 节点

使用方式很简单，第一种方式采用 `roslaunch` 运行 `slam_gmapping` 节点，但是后面需要带上若干参数，所以我们一般采用第二种方式运行 `slam_gmapping_pr2.launch` 启动文件。启动文件内的各个参数如下表 6-4 所示，这些都是在建图中使用的重要参数。

表 6-4 Gmapping 中的重要参数

名称	类型	默认	备注
<code>base_frame</code>	string	<code>base_link</code>	机器人坐标系
<code>map_frame</code>	string	<code>map</code>	地图坐标系
<code>odom_frame</code>	string	<code>odom</code>	里程计坐标系
<code>map_update_interval</code>	float	5.0	多长时间更新一次地图 (s)
<code>maxUrange</code>	float	8.0	激光雷达实际上能测量的最大距离
<code>maxRange</code>	float	8.0	激光雷达理论上能测量的最大的距离
<code>iterations</code>	int	5	扫描匹配中的迭代数
<code>ogain</code>	float	3.0	估计似然值使用，平滑重采样带来的影响
<code>lskip</code>	int	0	0 表示所有的激光都处理 1 表示处理部分
<code>minimumScore</code>	float	0.0	对激光数据的置信度
<code>linearUpdate</code>	float	1.0	移动linearUpdate距离，进行scanmatch
<code>angularUpdate</code>	float	0.5	旋转angularUpdate角度，进行scanmatch
<code>resampleThreshold</code>	float	0.5	重采样的判断阈值
<code>particles</code>	int	30	粒子数量
<code>xmin、ymin</code>	float	-100	初始化地图大小
<code>ymax、ymax</code>	float	100	初始化地图大小
<code>delta</code>	float	0.05	地图分辨力，即每个栅格的长度 (m)
<code>occ_thresh</code>	float	0.25	判断栅格是否被占用的阈值
<code>inverted_laser</code>	string	false	判断激光旋转方向是否倒转
<code>throttle_scans</code>	int	1	一次处理激光数据个数

步骤 5: 远程运动控制

在地图创建过程中，为了防止人为近距离控制机器人运动会影响激光扫描的数据，从而影响建图效率和准确度，所以我们需要远程控制机器人在需建图环境中的运动。另外，实现远程通信还可以方便我们实时观察 `rviz` 上地图创建的情况。

在本文实验环境下需要实现一台台式主机和移动机器人上的 MiniPC 通信。台式机的主要作用是通过键盘控制机器人 FinaRobot 运动和用 `rviz` 可视化界面监控建图情况，MiniPC 通过 ROS 实现移动机器人的定位、建图和路径规划等程序。两台 PC 都装有 `ubuntu 14.04` 系统，且都在一个局域网下。

远程运动控制的结构框图如下图 6-1 所示：

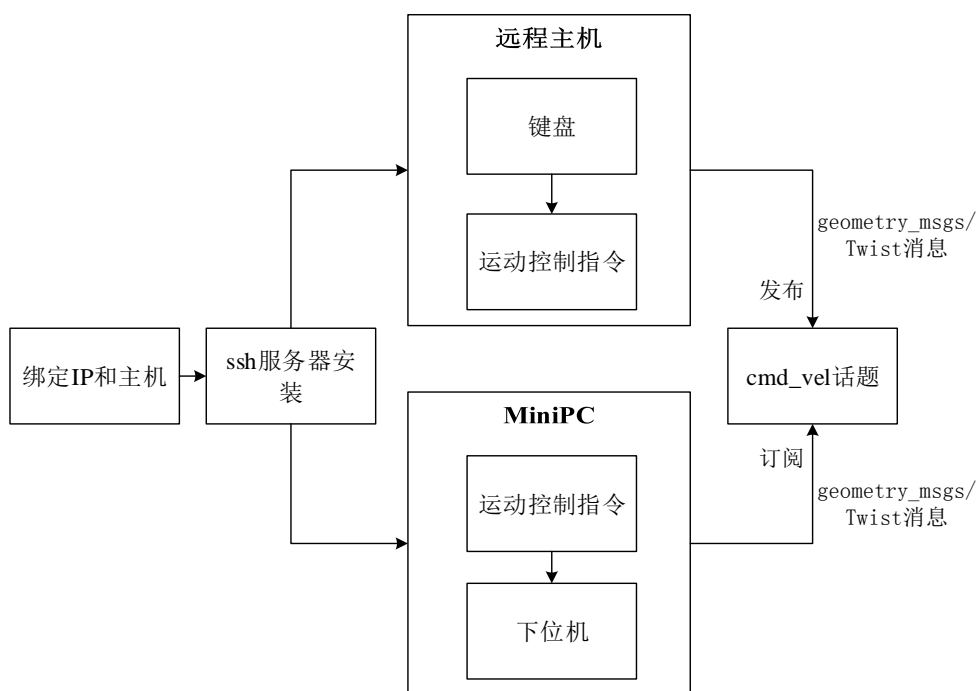


图 6-1 远程运动控制结构框图

（1）绑定 IP 与主机

分别得到两台主机的 IP 和 `hostname`，修改两台主机的 `hosts` 文件，均添加如下内容：

```
IP1      hostname1
IP2      hostname2
```

其中 `hostname1`、`hostname2` 分别代表远程主机和 MiniPC 的主机名。

（2）安装 ssh 服务器

SSH（Secure Shell）是一种用于计算机通信或者加密登录的网络安全协议，两

台计算机使用 SSH 协议进行通信，那么通信内容是经过加密处理的，需要在接收方进行解密才能获取内容。因此，这种通信方式是相对安全的。

在两台主机上装上 `chrony` 包，用于实现同步，同时都安装 `ssh` 服务器，并使用 `ssh hostnameX` 开启服务器。此时可以基于 `ping` 命令检查两台主机是否可以正常通信。

(3) 通过 `cmd_vel` 话题发送 `geometry_msgs/Twist` 消息进行机器人运动控制，`geometry_msgs/Twist` 消息包含两个向量：`linear` 和 `angular`，每个向量中又保存了三个 `float64` 类型的数据（分别为 `x`、`y` 和 `z`）。

在本文中，对于键盘操作所对应的 `geometry_msgs/Twist` 消息进行的规定如下表 6-5 所示：

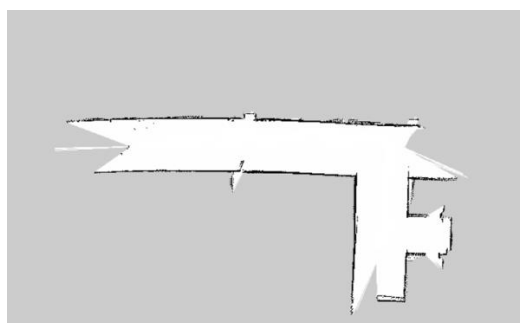
表 6-5 按键和机器人运动方向的对应关系

数字按键 1	forward	<code>linear.x=0.3, angular.z=0</code>
数字按键 2	back	<code>linear.x=-0.3, angular.z=0</code>
数字按键 3	left	<code>linear.x=0, angular.z=0.5</code>
数字按键 4	right	<code>linear.x=0, angular.z=-0.5</code>
数字按键 5	stop	<code>linear.x=0, angular.z=0</code>
default	forward	<code>linear.x=0.1, angular.z=0</code>

步骤 6：在完成以上所有步骤后，通过机器人控制程序控制机器人在实验环境中行走，打开 `rviz` 可视化界面即可看到不断膨胀的栅格地图。本次实验分别在龙湖 1501 和主楼走廊两个实验环境中进行测试，最终地图如下图 6-2 (a) 和 (b) 所示。



(a)



(b)

图 6-2 环境建图实验结果 (a) 龙湖 1501 室地图；(b) 走廊地图

经过反复的实验，采用改进的 RBPF-SLAM 算法，在降低粒子数量的情况下还能够精确地建立环境地图，计算速度也得到了一定的提升。

6.2 机器人语音交互导航实验与测试

步骤 1：语音识别功能。

下图 6-3 (a)、(b) 分别是正在语音识别和说了“请问 A 区怎么去”后识别结果。



图 6-3 语音识别实验结果 (a) 识别中；(b) 识别结果

步骤 2：问答系统--财务地点查询功能

基于步骤 1 转化后的问题文本，经过分词算法最后得到的关键词组为{请问，a，区，怎么，去}，删除停用词后的关键词组为{a，区}，客户端将该问题关键词组通过 http 协议发送到云服务器，在服务器端的 php 程序中计算该问题词组和数据库答案库中所有答案词组进行余弦相似度匹配，最终匹配到的答案为标明 a 区的地图图片，具体如下图 6-4 所示：



图 6-4 地点查询答案反馈

点击【带我去】按钮，开启智能导航模块。

步骤 3：基于 ROS 的导航系统测试

启动 `007_nav_configuration.launch`、`move_base.launch`、`send_pose` 节点和 `send_goal` 节点。`007_nav_configuration.launch` 中包含 `rplidar`、`odometry` 和 `tf` 节点，`move_base.launch` 中包含 `map_server` 节点、`move_base` 节点和 `amcl_omni.launch`，该测试使用的地图是 `1501.yaml`。另外，启动 `move_base` 节点需要为其设置运行成本、机器人半径、到达目标位置的距离、机器人移动速度等参数。

客户端将 1501 室 A 区的答案通过通信系统发往 MiniPC，MiniPC 在地址库内匹配到该地点的坐标后，通过 `send_pose` 节点将坐标封装成 `Turtlesim::Pose` 消息类型发布到 `pose` 话题，`send_goal` 节点订阅该话题，一旦有接收到位姿消息便会调用 `sendToMoveBase()` 回调函数，在该函数中通过 `MoveBaseClient` 将目标位姿发布到 `move_base` 节点，并开启导航，当前程序被阻塞。全局路径规划路线图如下图 6-5 所示：



图 6-5 机器人引导路线

当到达目的地后，`sendToMoveBase()` 函数中阻塞部分被唤醒，将到达信息发送给通信系统，又通信系统将该信息返回给客户端，客户端解析得到指令信息，利用语音合成技术转化为自然语音反馈给用户。

6.3 机器人自动充电技术实验与测试

在本文第五章提出了一种基于突变点检测的充电桩检测技术，下文将针对该技术和基于激光雷达的远程导航技术给出详细的实验步骤和实验结果。

步骤 1：初始位置标定

运行 `amcl` 节点后，首先加载环境地图，接着对充电桩的位置做了初始标定，

记录充电桩的具体位姿和角度。该位姿就是步骤 2 路径规划的终点。打开 rviz 即可显示地图、机器人位置和方向。

步骤 2：低电量状态开启远程路径规划

启动 007_nav_configuration.launch、move_base.launch、send_pose 节点、send_goal 节点和 judge_charge 节点。judge_charge 节点每 10 分钟通过串口获取一次电量信息，一旦进入低电量状态，通过 send_pose 节点发送充电桩位姿。

完成以上步骤机器人就会按照规划路径导航前进。最后我们运行 rviz 可以很方便地观察全局路径规划路线以及机器人在导航过程中的实时位姿，导航过程如图 6-6 所示。

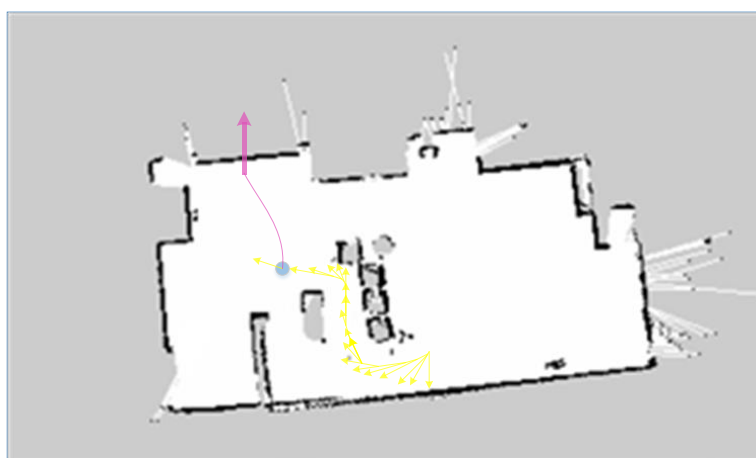


图 6-6 机器人全局路径规划路线

步骤 3：充电桩对接

完成远距离导航后，send_goal 节点中阻塞位置被唤醒，接着调用充电桩识别和对接程序，具体实现过程如图 6-7 所示。

- (1) 到达充电桩附近后（2.1m 范围内），开启近程对接程序。
- (2) 判断当前机器人方向与初始标定的方向是否一致，如不一致则进行调整。
- (3) 订阅激光雷达数据，做无效数据过滤，跨越法补全激光数据，对突变点进行标定，识别出连续 3-5 个突变点为充电桩位置，最终采用角度折半划分法来大致确定偏向角度。当在 2.1m 范围内没有搜索到充电桩，则调用 send_pose 重新进行远程路径规划。
- (4) 调用机器人运动控制程序，沿着步骤（3）中确定位置的角度前进。
- (5) 重复步骤 2-步骤 4 直至最终偏向角度到达误差容忍范围内，前进对接，完成自动充电。

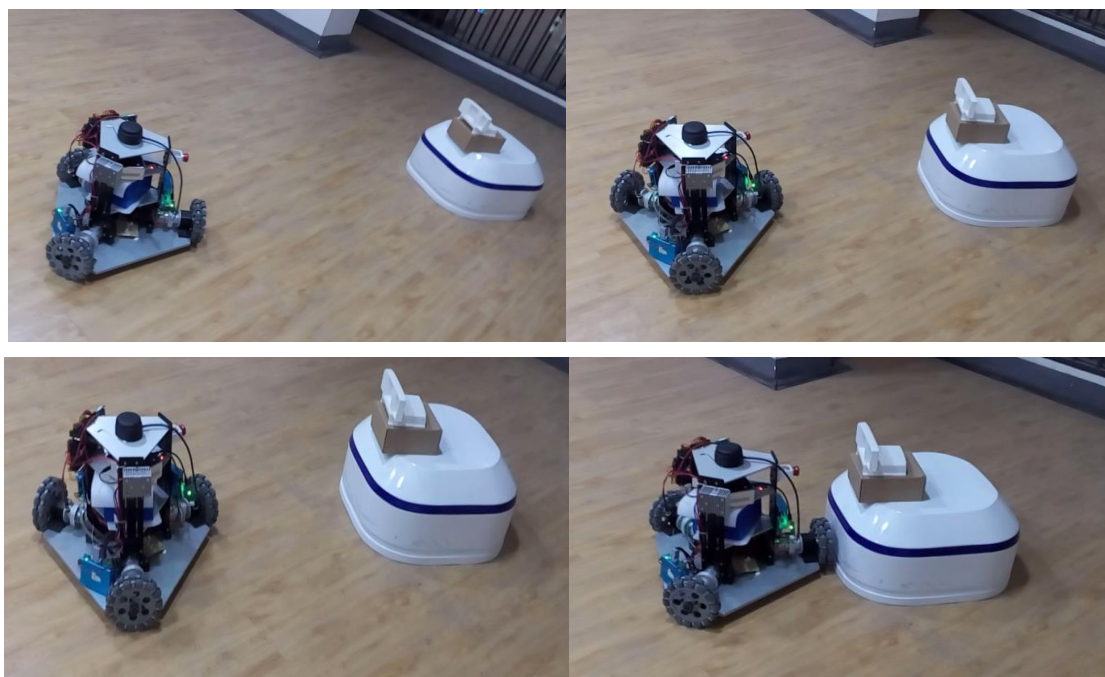


图 6-7 自动充电近程对接实验过程

6.4 本章小结

本章在机器人 FinaRobot 平台基础上搭建了基于语音交互的移动机器人智能导航系统和基于突变点检测的自动充电系统，并且针对上述两个系统的各个功能都进行了详细的测试和分析对比。通过一次语音的财务地点咨询和带领验证语音导航系统的可行性，并且通过低电量状态下的机器人动作以及客户端状态验证了 FinaRobot 导航系统中各个算法的可靠性和有效性。另外，客户端在提供全面的财务咨询功能的同时还具有拟人化的人机交互界面，可以给用户带来更好的使用感和趣味性。总的来说机器人 FinaRobot 具有很强的实用价值。

第七章 全文总结与展望

7.1 全文工作总结

本文在电子科技大学财务机器人 FinaRobot 项目的研发基础上,重点对室内移动机器人的建图以及导航技术进行研究和实验。在理论研究部分,针对地图创建提出了一种可减少计算时间、降低粒子退化速度的改进算法。工程实验阶段设计了一套机器人自主导航软件系统,包括环境地图创建、语音导航系统和自动充电系统。在论文后期对上述功能进行一一测试,解决当前存在的可解决的问题,对待解决的问题形成文档。实验结果验证了电子科技大学财务机器人 FinaRobot 可以在财务咨询方面给予用户很大的帮助,作为客户端人机交互运行的坚实基础,下层的导航软件系统在算法和设计上表现出有效性和正确性。本文的主要研究工作如下:

(1) 针对本文对导航系统的功能需求,对财务机器人 FinaRobot 的总体架构(包含软件系统和硬件系统)进行构思和设计,其中最主要的软件系统主要包括三大模块:环境地图创建、语音导航和自动充电,同时选用 Socket 和 HTTP 通信作为两大导航系统中的网络通信方式。最后统一整个软硬件开发环境。

(2) 了解 SLAM 和激光雷达的原理,引入了一种非参数化滤波器,即粒子滤波器。接着分析了传统 RBPF-SLAM 存在的问题,包括计算复杂度高以及粒子退化问题。基于上述两个问题,本文引入了一种改进的 RBPF-SLAM,通过一种改进的粒子重采样策略,降低粒子退化的速度,保持粒子多样性,减少粒子个数,进一步提交建图效率。基于上述改进算法,完成代码编写,最终选取两个不同的环境验证该算法的优化效果。最后对如何在 ROS 下面创建环境地图的执行步骤进行了详细介绍。

(3) 设计了一套基于语音交互的智能导航系统,系统采用语音识别技术将输入问题转化为文本,通过自然语言处理和余弦相似度匹配最终在答案库中匹配到该地点的平面地图,接着便开启导航系统,将用户引导到目的地点,同时客户端通过语音合成技术向用户进行反馈。

(4) 提出了一种基于突变点检测的充电桩对准算法并完成了自主充电系统的设计与实现,同时引入了一套较高误差容忍度的充电头和充电座,充分提高机器人自动充电的精准对接的成功率。

(5) 最后,在已有的硬件平台上搭建和整合导航系统软件模块,针对完整的机器人导航系统做了若干测试实验用于验证各个软件模块的合理性和有效性。

7.2 后期工作展望

在本论文工程试验阶段，在对系统架构设计和代码编写的过程中遇到的一些需要进一步改进的问题，具体如下：

（1）第三章引入的改进的 RBPF-SLAM 算法，虽然在小范围环境下验证了其优化效果，但由于环境条件限制没有在更大、更复杂的环境下进行实验，后期将该改进算法在复杂环境下的建图优化能力做进一步的验证。

（2）在自动充电模块中，未对最后对接部分做详细设计，包括对接前的速度控制，对接后的连接判断。

（3）以上实验所用的底盘和安卓平台还未组装成完整机器人，后期会进行加工组装。

致 谢

首先感谢电子科技大学，给我们提供非常好的科研和学习氛围以及良好的生活环境。感谢我的导师—葛树志教授在学业上对我的帮助，葛教授经常在精神上给予我们很多鼓励，帮助我们建立逻辑思维能力，他在机器人模式识别领域前沿的观点也深深影响了我，我对此受益匪浅。感谢叶茂教授，在科研项目和论文方面给予我很多的指导和帮助，让我少走了很多弯路。感谢机器人中心的各位老师、各位同学，感谢大家对我的论文工作做出的贡献。

最后感谢所有关心帮助过我的人，祝大家万事胜意！

参考文献

- [1] 孙志杰, 王善军, 张学鑫. 工业机器人发展现状与趋势[J]. 吉林工程技术师范学院学报, 2011(7): 61-62.
- [2] 嵇鹏程, 沈惠平. 服务机器人的现状及其发展趋势[J]. 常州大学学报(自然科学版), 2010(2): 3-78.
- [3] 沈绪明, 董鹏. 我国机器人发展现状, 需求及产业化探讨[J]. 物流技术(装备版), 2012(22): 68-71.
- [4] 王田苗, 陶永, 陈阳. 服务机器人技术研究现状与发展趋势[J]. 中国科学: 信息科学, 2012(9): 1049-1066.
- [5] S. S. SRINIVASA, D. FERGUSON, C. J. HELFRICH, et al. a home exploring robotic butler[J]. Autonomous Robots, 2010, 28(1): 5-20.
- [6] T. Breuer, G. R. G. Macedo, R. Hartanto, et al. an autonomous service robot for domestic environments[J]. Journal of Intelligent & Robotic Systems, 2012, 66: 245-272.
- [7] S. Schiffer, A. Ferrein, G. Lakemeyer. an intelligent domestic service robot[J]. Intelligent Service Robotics, 2012.
- [8] L. Zhao, R. Li, T. Zang, et al. A Method of Landmark Visual Tracking for Mobile Robot[M]. Intelligent Robotics and Applications. Springer Berlin Heidelberg, 2008:901-910.
- [9] 于清晓. 轮式餐厅服务机器人移动定位技术研究[D]. 上海: 上海交通大学机械与动力学院, 2013.
- [10] 沈奎林, 邵波, 陈力军, 单国锋. 基于超高频 RFID 的图书盘点机器人的设计和实现[J]. 图书馆学研究, 2016, (07): 24-28.
- [11] K. Park, D. Chung, H. Chung, J. G. Lee. Dead Reckoning Navigation of a Mobile Robot Using an Indirect Kalman Filter[A]. ICMFIIS1996[C]. 1996: 132-138.
- [12] K. Park, H. Chung, G. Choi, J. G. Lee. Dead Reckoning Navigation for an Mobile Robot Using a Differential Encoder and a Gyroscope[A]. Proceedings of the ICRA 1997[C]. 1997: 441-446.
- [13] 席志红, 原新, 许辉. 基于视觉的移动机器人实时避障和导航[J]. 哈尔滨工程大学学报, 2002, 23(5): 107-109.
- [14] 包桂秋, 周兆英, 熊沈蜀, 叶维英. 图像导航技术的发展和应[J]. 航空计测技术, 2003, 23(6): 1-4.
- [15] G. N. DeSouza, A. C. Kak. Vision for Mobile Robot Navigation: A Survey[J]. IEEE Trans. on PAMI, 2002, 24(2): 237-266.

- [16] E. Menegatti, T. Maeda, H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images[J]. *Robotics and Autonomous Systems*, 2004, 47(4): 251-267.
- [17] 童峰, 许天增. 一种移动机器人超声波导航系统[J]. *机器人*, 2002, 24(1): 55-57.
- [18] K. T. Song, C. C. Chang. Reactive navigation in dynamic environment using a multisensor predictor[J]. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society*, 1999, 29(6): 870.
- [19] H. Maaref, C. Barret. Sensor-based navigation of a mobile robot in an indoor environment[J]. *Robotics & Autonomous Systems*, 2002, 38(1): 1-18.
- [20] D. Ratner, P. Mckerrow. Navigating an outdoor robot along continuous landmarks with ultrasonic sensing[J]. *Robotics & Autonomous Systems*, 2003, 45(2): 73-82.
- [21] 宋爱国, 王庆, 黄惟. 基于数字地形图移动机器人的 GPS 自主导航[J]. 1995(4): 63-67.
- [22] S. Li, A. Hayashi. Robot navigation in outdoor environments by using GPS information and panoramic views[C]. *IEEE International Conference on Intelligent Robots and Systems*, 1998. *Proceedings. IEEE*, 1998: 570-575.
- [23] S. Panzneri, F. Pascucci, G. Ulivi. An outdoor navigation system using GPS and inertial platform [J]. *IEEE Transactions on Mechatronics*, 2002, 7(2): 134-142.
- [24] M. W. M. G. Dissanayake, P. Newman, S. Clark, et al. A solution to the simultaneous localization and map building (SLAM) problem[J]. *Robotics & Automation IEEE Transactions on*, 2001, 17(3): 229-241.
- [25] H. Durrant-Ahyte, T. Bailey. Simultaneous Localization and Mapping(SLAM)[J]. *The Essential Algorithms. IEEE Journal of Robotics and Automation*, 2006, 13(2): 99-110.
- [26] H. Durrant-Whyte. Where am I? A tutorial on mobile vehicle localization[J]. *Industrial Robot*, 1994, 21(2): 11-16.
- [27] 李群明, 熊蓉, 褚健. 室内自主移动机器人定位方法研究综述[J]. *机器人*, 2003, 25(6): 560-567.
- [28] 基于激光测距仪的室内机器人 SLAM 研究[D]. 彭晟远. 武汉大学 2012.
- [29] H. P. Moravec, A. Elfes. High resolution maps from angle sonar[C]. *ICRA*. 1985: 116-121.
- [30] A. Elfes. *Sonar-Based Real-World Mapping and Navigation*[C]. Springer-Verlag New York, Inc. 1990: 249-265.
- [31] T. Lemaire, C. Berger, I. K. Jung, et al. Vision-Based SLAM: Stereo and Monocular Approaches[J]. *International Journal of Computer Vision*, 2007, 74(3): 343-364.
- [32] 秦晖. 轮式家庭服务机器人的移动控制设计与研究[D]. 电子科技大学, 2016.

- [33] A. Doucet, N. D. Freitas, K. Murphy, et al. Rao-blackwellised particle filtering for dynamic Bayesian networks[C]. Sixteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc. 2000: 176-183.
- [34] G. Grisetti, C. Stachniss, W. Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters[J]. IEEE Transactions on Robotics, 2007, 23(1): 34-46.
- [35] N. D. Freitas. Rao-Blackwellised particle filtering for fault diagnosis[C]. Aerospace Conference Proceedings. IEEE, 2002: 4-1767-4-1772 vol.4.
- [36] M. Montemerlo, S. Thun, D. Koller, et al. FastSLAM: A factored solution to simultaneous mapping and localization[C]. National Conference on Artificial Intelligence. 2002.
- [37] M. Montemerlo, S. Thrun, D. Roller, et al. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges[C]. International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 2003: 1151-1156.
- [38] G. Grisetti, C. Stachniss, W. Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters[J]. IEEE Transactions on Robotics, 2007, 23(1): 34-46.
- [39] 张建伟, 张立伟, 胡颖,等. 开源机器人操作系统—ROS[M]. 科学出版社, 2012.
- [40] B. Gerkey. Slam_gmapping [OL]. [2010-08-05]. http://wiki.ros.org/slam_gmapping.
- [41] H. C. Lee, S. K. Park, J. S. Choi, et al. PSO-FastSLAM: An improved FastSLAM framework using particle swarm optimization[C]. IEEE International Conference on Systems, Man and Cybernetics. IEEE, 2009: 2763-2768.
- [42] D. H. Won, S. Chun, S. Sung, et al. SLAM system using distributed particle filter[J]. International Journal of Control Automation & Systems, 2010, 8(6): 1232-1240.
- [43] 朱磊, 樊继壮, 赵杰,等. 改进粒子滤波器的移动机器人同步定位与地图构建方法[J]. 重庆大学学报:自然科学版, 2014, 37(4): 39-45.
- [44] 李娟, 刘晓龙, 卢长刚,等. 改进的粒子滤波重采样算法[J]. 吉林大学学报(工), 2015, 45(6): 2069-2074.
- [45] J. S. Liu, R. Chen. Sequential Monte Carlo Methods for Dynamic Systems[J]. Publications of the American Statistical Association, 1998, 93(443): 1032-1044.
- [46] R. Havangi, H. D. Taghirad, M. A. Nekoui, et al. A Square Root Unscented FastSLAM With Improved Proposal Distribution and Resampling[J]. IEEE Transactions on Industrial Electronics, 2013, 61(5): 2334-2345.
- [47] C. Feng , M. Wang, Q. B. Ji. Analysis and comparison of resampling algorithms in particle filter[J]. Journal of System Simulation, 2009, 14(2): 323-331.

- [48] J. S. Liu, R. Chen. Sequential Monte Carlo Methods for Dynamic Systems[J]. Publications of the American Statistical Association, 1998, 93(443): 1032-1044.
- [49] Y. U. Jin-Xia, Z. X. Cai, Z. H. Duan. Survey on some key technologies of mobile robot localization based on particle filter[J]. Application Research of Computers, 2007, 24(11): 9-14.
- [50] R. V. D Merwe, A. Doucet, N. D. Freitas, et al. The unscented particle filter[C]. International Conference on Neural Information Processing Systems. MIT Press, 2000: 563-569.
- [51] M. Bolic, P. M. Djuric, S. Hong. New resampling algorithms for particle filters[C]. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. IEEE, 2003: II-589-92 vol.2.
- [52] B. GERKEY. Gmapping [EB/OL]. [2010-08-05]. http://wiki.ros.org/slam_gmapping.
- [53] Y. Dong, F. Seide, L. Gang. Conversational speech transcription using context-dependent deep neural networks[C]. International Conference on Machine Learning. Omnipress, 2012: 1-2.
- [54] 柳春. 语音识别技术研究进展[J]. 甘肃科技, 2008, 24(9): 41-43.
- [55] Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Daniel Jurafsky, James H Martin. 2000.
- [56] 张歆奕, 吴今培, 张其善. 车载导航仪中路径规划算法及其实现[J]. 计算机测量与控制, 2001, 9(4): 15-17.
- [57] D. Fox, W. Burgard, S. Thrun. The dynamic window approach to collision avoidance[J]. IEEE Robotics & Automation Magazine, 2002, 4(1): 23-33.
- [58] 徐保来, 管贻生, 苏泽荣,等. 改进动态窗口法的阿克曼移动机器人局部路径规划器[J]. 机电工程技术, 2016, 45(9): 21-26.
- [59] D. G. Aid. Automatic charging system: US, US2905926[P]. 1959.