

# 2D 激光雷达移动机器人 SLAM 系统研究

何佳泽 张寿明

(昆明理工大学 信息工程与自动化学院 昆明 650500)

**摘要:** 针对传统 RBPF-SLAM 算法构建地图存在重影,粒子出现衰退等问题,设计了阈值重采样函数进行优化采用。并使用移动机器人实验平台,该移动机器人 SLAM 建图采用 Arduino 作为核心控制器,采用的是 2D 激光雷达传感器作为外部环境探测器。采集到的信号经过 Arduino 处理,通过 WiFi 传递给 ROS 操作系统,通过上位机处理,构建环境地图。在 ROS 机器人操作系统下通过移动机器人测试平台在两个不同环境下进行地图构建,比较算法优化前后的构图效果来比较两种算法的性能。对其中一种实验环境进行测量,通过比较两种算法构建地图的尺度精度,对优化前后的算法性能量化。通过直观建图效果和建图精度两种方法论证,结果表明优化算法建图效果更好,建图误差从 0.459% 降到 0.299%,该优化算法实验结果与理论推导一致。

**关键词:** SLAM 建图; Arduino; 激光雷达; ROS; RBPF-SLAM 算法

**中图分类号:** TP242.3 **文献标识码:** A **国家标准学科分类代码:** 510.8050

## Research on SLAM system of 2D lidar mobile robots

He Jiaze Zhang Shouming

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

**Abstract:** Arduino is used as the core controller and 2D lidar sensor is used as the external environment detector in the construction of mobile robot SLAM. The collected signal is processed by Arduino and transmitted to ROS operating system through WiFi. The environment map is constructed through processing by the upper computer. The traditional RBPF-SLAM algorithm was optimized, and the map was constructed by the mobile robot test platform under the ROS robot operating system in two different environments. The composition effects before and after the algorithm optimization were compared to compare the performance of the two algorithms. One of the experimental environments was measured. By comparing the scale accuracy of the map constructed by the two algorithms, the performance of the algorithm before and after optimization was quantified. It can be seen that the optimized algorithm has better effect of 2D plane map construction through the demonstration of two methods of visual map construction effect and map construction accuracy, the error of mapping is reduced from 0.459% to 0.299%, and the experimental results of the optimization algorithm are consistent with the theoretical derivation.

**Keywords:** SLAM mapping; Arduino; lidar; ROS; RBPF-SLAM algorithm

## 0 引言

随着中国制造 2025 的即将到来,工业水平的提高,必然依靠自动化水平的提高;而移动机器人在自动化领域起到连接各个生产环节的作用,如果一个环节生产效率比较高,而不能将这个生产环节的产品运送到下一个生产环节,所以必然造成整个生产系统生产效率低下<sup>[1]</sup>;而且移动机器人性能提高,能降低不必要的浪费,而且可以降低人力成本,并且高性能的移动机器人的搬运效率比人力更高<sup>[2]</sup>。

移动机器人通过激光和视觉探测器探测环境,通常有 2D 激光雷达探测传感器,单目视觉传感器和双目视觉感

器等<sup>[3-5]</sup>。同步定位与地图构建(simultaneous location and mapping, SLAM)在移动机器人研究方面占据着重要的位置;移动机器人 SLAM 常用的算法有基于 Rao-Blackwellisedz 粒子滤波器的 RBPF-SLAM 算法<sup>[6]</sup>, Gmapping 算法, Hector 算法和 Cartographer 算法<sup>[7-9]</sup>。传统 RBPF-SLAM 算法在实际环境中需要采集大量的粒子,这样会引起评估值与观测到的数值之间出现较大误差,这样就要通过大量更多的粒子来拟合误差。大量的粒子会消耗大量的内存,并且增加算法的计算量从而导致算法效率的降低。为了降低采集粒子的数量。本文采用自适应重采用方法去除无效和低效的粒子,将剩下的粒子构成新的粒

子集合,从而降低使用粒子的数量,从而提高算法的运算效率<sup>[10]</sup>。

通过提高算法效率从而提高建图效果,可以降低硬件成本。为了提高移动机器人的性能,本文按照低成本,易实现,易操作,实用性的设计理念。采用激光雷达摄像头作为外部环境探测器,使用 Arduino 控制主板,控制两个伺服电机,驱动移动机器人移动。通过机器人操作系统(ROS)进行实验,检验算法的性能。

### 1 总体设计方案

Arduino 控制板在电机控制方面因为具有价格低,性能好等特点被广泛采用,结合 RPLIDAR A1 激光雷达传感器进行环境探测与地图构建。并通过 ROS 机器人操作系统,将构建的地图返回到上位机上面。从而实现了整套 SLAM 建图。

系统主要分为硬件传感器部分,ROS 机器人操作系统,SLAM 地图构建算法部分。SLAM 系统的流程是 Arduino 主板控制伺服电机驱动整套系统在环境中移动,使 2D 激光雷达通过两点定位方法将采集到的粒子经过无线网络传递给上位机(即 ROS 机器人操作系统),上位机通过算法处理采集到的传感器信号,从而在上位机上形成即时的环境地图。

### 2 系统硬件设计

移动机器人结构如图 1 所示,系统包括激光雷达传感器部分和 Arduino 主控板部分。

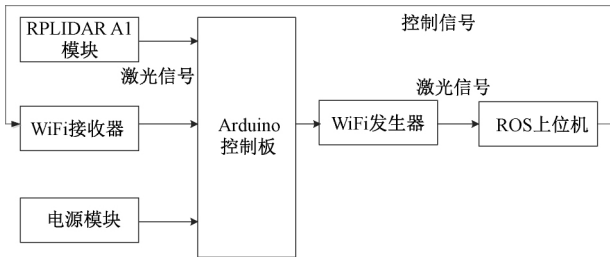


图 1 系统结构

#### 2.1 2D 激光雷达传感器模块

本文采用的是 RPLIDAR A1 激光雷达传感器。此传感器是基于三角测距原理的。它的价格便宜,并且在近距离测量精度比较高,适合在室内环境下使用。它的基本原理是激光雷达传感器首先通过发射器发出激光雷达信号,发出去的激光雷达遇到障碍物时就会被接收器检测到信号,从而根据三角测距原理测出距离<sup>[11]</sup>。其原理如图 2 所示。

激光雷达安装在移动机器人平台上的,所以激光雷达与障碍物之间的距离,可以大致认为就是移动机器人与障碍物之间的距离,此距离就是图 2 中的  $d$ 。 $L$  表示接收器与激光发生器之间的距离, $L_1$  表示接收器与障碍物垂直线之间的距离, $L_2$  表示障碍物与激光发生器之间的距离,

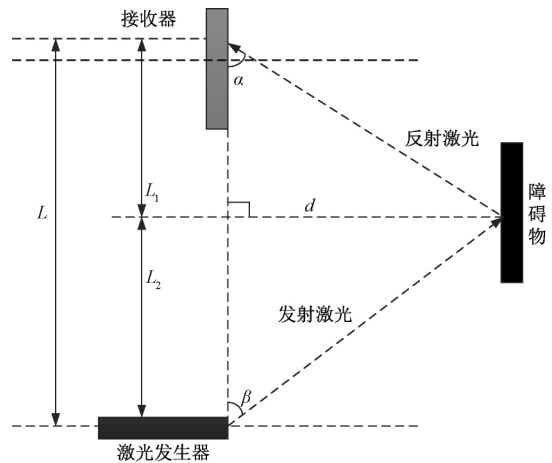


图 2 激光雷达传感器测距原理

$\alpha$  表示接收器与障碍物之间夹角, $\beta$  表示激光发生器与障碍物之间夹角,根据三角原理如式(1)所示。

$$\begin{cases} \tan\alpha = \frac{d}{L_1} \\ \tan\beta = \frac{d}{L_2} \end{cases} \quad (1)$$

式中: $L_1, L_2$  的数值是固定的,并且可以测量,并且由公式  $L = L_1 + L_2$  可以算出  $L$  的数值;式(1)表示的是角度关系,通过反三角函数如式(2)所示,可以推导出  $d$ :

$$d = L \frac{\sin\alpha \sin\beta}{\sin(\alpha + \beta)} \quad (2)$$

然后通过 TF 坐标变换,即可以得到移动机器人坐标位置,激光信号通过 WiFi 发送给上位机,通过上位机处理。从而在环境中定位机器人位置,感受障碍物,构建地图。

#### 2.2 Arduinio 模块

主控模块是采用的 Arduino 控制器,使用 Arduino IDE 集成开发环境,使用 C、C++ 编程语言。主要是用来将激光雷达采集到的信号通过 WiFi 传递给上位机机器人操作系统,将上位机传来的控制信号转换为移动机器人的运动信号。Arduino 主要是通过控制小伺服电机来控制整套系统运动。控制方式为差速控制,通过两个伺服电机的差速来实现移动机器人平台的转向<sup>[12]</sup>。要通过差速控制移动机器人运动,首先建立差速模型如图 3 所示。

$V_L$  为左轮运动速度, $V_R$  为右轮运动速度,两轮的中间距为  $D$ ,所以机器人的运行线速度如式(3)所示。

$$V = \frac{V_L + V_R}{2} \quad (3)$$

$\omega$  表示移动机器人角速度,可以求出它的角速度如式(4)所示。

$$\omega = \frac{V_R}{\frac{D(V_L + V_R)}{2(V_L - V_R)} + D/2} \quad (4)$$

化简得到式(5):

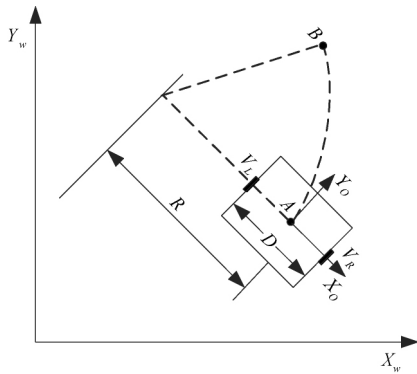


图 3 Arduino 控制差速运动模型

$$\omega = \frac{V_L - V_R}{D} \quad (5)$$

所以 Arduino 就是通过差速模型来控制移动机器人运动的。

### 3 SLAM 建图算法

移动机器人通过 WiFi 将采集到的激光粒子信号传递给上位机(ROS 机器人操作系统)。通过 ROS 操作系统进行滤波处理,从而构建环境地图。针对传统 RBPF-SLAM 滤波算法不足的地方加以改进,以提高建图效果。

#### 3.1 RBPF-SLAM 算法

RBPF-SLAM 算法是基于卡尔曼滤波的滤波 SLAM 框架。RBPF-SLAM 算法是通过粒子滤波来估计机器人的位置姿态。此算法通过不断的迭代更新来预测各个时刻的机器人的位姿姿态<sup>[13]</sup>。流程如图 4 所示。



图 4 RBPF-SLAM 算法流程

RBPF-SLAM 算法,每一个粒子都包含着一幅地图,那么将使整个系统计算量变得非常的大,并且由于采集粒子数较多,所以会造成整个系统构建的二维平面地图出现重影。

对 RBPF-SLAM 算法的改进主要是从改进提议分布和减少重采样次数两个方面考虑。本文系统主要是从流程中的第一步,采样方面考虑,采用选择性重采样提高整体效果。

#### 3.2 改进 RBPF-SLAM 算法

因为激光雷达时时刻刻都在传递激光粒子信号给 ROS 系统处理,系统将选择其中有效的粒子进行构建地图,这个步骤就叫采样。但是其中一些粒子会衰减,一些粒子会一直采样。所以本文系统通过设置 3 段阈值,只有在中间阈值区间内才执行重采样,这样减少了采样次数,也就可以减缓粒子衰退<sup>[14-16]</sup>。

本文系统通过设置两个阈值,将采样值区间划分成为 3 个子区间;对采集到的粒子按照权重进行高低排序,中间部分阈区间的粒子集  $S_B$  效果最好,此粒子集可以直接使

用,直接保留。将高阈值区间和低阈值区间的粒子集合进行整合形成新的粒子集  $S_A$ ,这样就可以对新的粒子集合优化重采样。

设置一个粒子数阈值  $\lambda$ ,它的值会随着环境的变化而变化,此系统假定将阈值为 100,按照如式(6)所示划分标准。

$$\begin{cases} \omega_L = \frac{2}{3} \cdot \frac{1}{N}, \omega_H = \frac{3}{2} \cdot \frac{1}{N}, & \lambda \leq 100 \\ \omega_L = \frac{4}{5} \cdot \frac{1}{N}, \omega_H = \frac{5}{4} \cdot \frac{1}{N}, & \lambda > 100 \end{cases} \quad (6)$$

式中:  $\omega_L$  为低权重阈值;  $\omega_H$  为高权重阈值。将新合并的粒子集  $S_A$  进行如式(7)所示的重采样:

$$\chi_n = \chi_s + KL(\chi_a - \chi_s) \quad (7)$$

式中:  $\chi_n$  为新产生的粒子;  $\chi_s$  为粒子集  $S_A$  中被重复选择权重较大的粒子;  $\chi_a$  为粒子集  $S_A$  中被重复选择权重较小的粒子;  $K$  为步长系数;  $L$  为  $(\chi_a - \chi_s)$  方向上的合适步长。

### 4 实验检验

本文系统通过 ROS 机器人上位机系统进行实验比较算法改进前后的效果,体现改进算法的优点。

#### 4.1 实验平台

上位机硬件:CPU i7-8750H;内存 16G。

移动机器人平台:天子博特 mini ROS 机器人底座;RPLIDAR A1 激光雷达传感器。

上位机操作系统:Ubuntu 18.04 机器人操作系统 Ros Melodic Morenia。算法层用 python 算法,驱动层 C++ 编程语言。

构建小型实验地图,设置方形和原型实验物体,如图 5 所示。



图 5 实验平台

#### 4.2 实验检验

如图 6 所示,将上位机和移动机器人放在一个网络的同一网段内,就可以开始检验。为了降低其他因素对实验结果的影响,所有实验都是用的 0.1 rad/s 的角速度和 0.3 m/s 的速度,并且同一地图中移动机器人按照同一路线绕地图运行一周。通过移动机器人构建地图得到如图 7 和 8 所示结果。

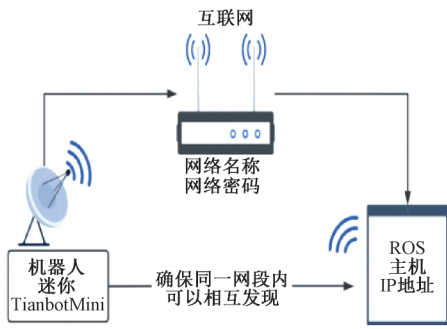


图 6 网络配置

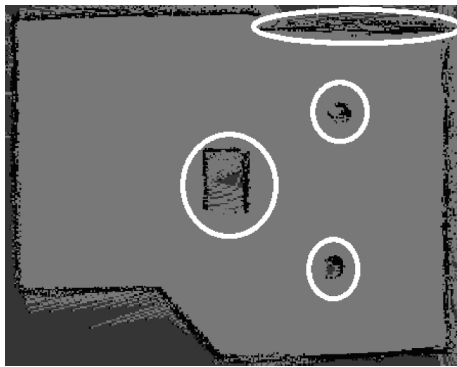


图 7 传统 RBPF-SLAM 算法构建实验地图

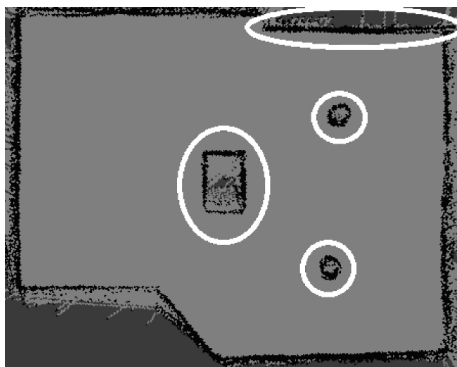


图 8 优化 RBPF-SLAM 构建实验地图

在实验室环境下,分别使用传统 RBPF 算法和优化算法构建两次环境地图,通过比较构建的地图,可以看出优化算法在实验环境中构图效果更佳,并将构图明显地方通过圆圈标记。

构图效果可以通过直观观测,因为不同算法构图会导致实际算法与构图算法尺寸精度有所不同;RVIZ 的 measure 工具可以测得建图尺寸,为了使算法优化后的效果可以直观量化用数字体现,将建图尺寸和图测尺寸进行比较,求出误差,来证明算法的优劣。分别取传统 RBPF-SLAM 算法和优化后的 RBPF-SLAM 算法中的 7 处同一关键位置,进行比较。如图 9 所示为选取的 7 个关键距离。

选取 7 个关键位置点之后,通过实际高精度测量尺得到如表 1、2 所示结果:表中实际编号与图中的编号位置一

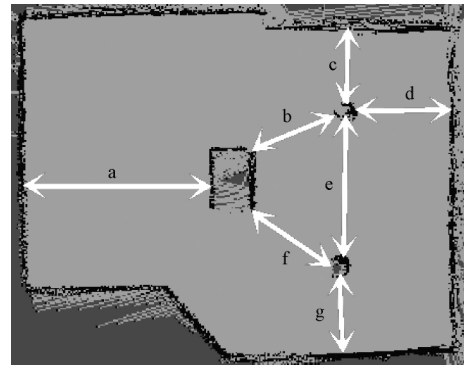


图 9 比较尺寸精度

一对应,绝对误差值=实际尺寸值-图测尺寸值;相对误差值=绝对误差值/实际误差值,平均误差值为 7 个位置误差的算术平均值;为了提高精度,表中所有值均保留 3 位小数。

表 1 传统算法建图尺寸

| 位置<br>编号 | 实际尺寸/<br>m | 图测尺寸/<br>m | 绝对误差/<br>m | 相对误差/<br>% |
|----------|------------|------------|------------|------------|
| a        | 80.135     | 79.907     | 0.228      | 0.28       |
| b        | 54.126     | 54.305     | -0.179     | 0.33       |
| c        | 34.625     | 34.423     | 0.202      | 0.58       |
| d        | 38.596     | 38.325     | 0.271      | 0.70       |
| e        | 81.056     | 81.336     | -0.280     | 0.35       |
| f        | 52.786     | 52.578     | 0.208      | 0.40       |
| g        | 35.698     | 35.903     | -0.205     | 0.57       |

注:计算求得传统算法的平均误差为 0.459%。

表 2 优化算法建图尺寸

| 位置<br>编号 | 实际尺寸/<br>m | 图测尺寸/<br>m | 绝对误差/<br>m | 相对误差/<br>% |
|----------|------------|------------|------------|------------|
| a        | 80.135     | 80.015     | 0.120      | 0.14       |
| b        | 54.126     | 54.234     | -0.108     | 0.20       |
| c        | 34.625     | 34.510     | 0.115      | 0.33       |
| d        | 38.596     | 38.403     | 0.193      | 0.50       |
| e        | 81.056     | 81.225     | -0.169     | 0.21       |
| f        | 52.786     | 52.659     | 0.127      | 0.24       |
| g        | 35.698     | 35.867     | -0.169     | 0.47       |

注:计算求得传统算法的平均误差为 0.299%。

通过对传统算法和优化算法建图尺寸精度比较,可以看出优化后的算法建图误差平均值更小,所以优化后的算法构建地图更接近真实尺寸,建图效果更好。

实验环境比对后,将移动机器人放在真实环境中分别使用传统算法和优化算法进行环境建图,将构建的环境地图保存,通过比较建图效果来比较算法的效果。

图 10 所示为真实环境,针对环境首先采用传统 RBPF-SLAM 算法构建地图,其构图结果如图 11 所示。



图10 真实环境



图11 传统算法构建地图

可以看出,因为没有过滤衰退粒子和重复粒子,在地图的边缘,会形成重影。

然后使用同样的实验平台,在同样的外部环境下,使用优化后的建图算法,让移动机器人以同样的速度、方向和路线在运行一次,得到如图12所示的环境地图。



图12 使用优化算法构建地图

可以看出,使用优化算法后,同样条件下构建的环境地图,边缘位置变得非常清晰,并且地图重影明显减少。

## 5 结 论

本文系统通过2D激光雷达传感器与Arduino控制器构成移动机器人。通过对移动机器人采集的激光雷达传感器的信号进行优化;主要是减少一些衰减的粒子和一些重复的粒子,这样不仅仅使构建的地图效果更加接近实际的环境,而且还大大降低了系统的运行条件。因为采集到大

量的无用粒子会使得上位机系统处理量变大,占用内存和CPU空间,通过对重采样优化,还能降低上位机硬件条件。本文通过真实实验,将算法在真实移动机器人上实现,通过实际建图效果直观比较直接看出优化算法建图效果明显提高;通过建图环境尺寸精度量化比较,以构建的真实实验环境尺寸做指标可以得出,传统算法构图误差为0.459%,优化后的RBPF-SLAM算法误差为0.299%,可以看出误差变小,精度提高,建图效果得到明显提搞。由于2D激光雷达相对于其他建图传感器来讲成本更低,提高2D激光建图效果,这样就能降低整套系统的成本。

## 参考文献

- [1] 李安旭. 移动机器人即使定位与地图构建技术研究[D]. 成都:电子科技大学,2019.
- [2] 孙小婷. 基于群智能优化粒子滤波的机器人SLAM算法研究[D]. 西安:西安科技大学,2020.
- [3] 王彦. 基于激光雷达的移动机器人SLAM研究[D]. 汉中:陕西理工大学,2020.
- [4] 李辉. 基于激光雷达的2D-SLAM的研究[D]. 杭州:浙江工业大学,2017.
- [5] 雷皓玮,林晓焕. 一种混合单目视觉SLAM系统研究[J]. 国外电子测量技术,2019,38(9):90-93.
- [6] 陈建军,廖小飞,吴赞,等. 基于小生境遗传优化的 Rao-Blackwellised SLAM 算法[J]. 计算机应用研究,2017,34(8):2368-2371.
- [7] 任纪颖. 基于图优化的差速移动机器人激光SLAM研究[D]. 济南:山东大学,2020.
- [8] 杨佳. 基于栅格地图的改进SLAM算法研究与实现[D]. 武汉:华中科技大学,2015.
- [9] 宗文鹏,李广云,李明磊,等. 激光扫描匹配方法研究综述[J]. 中国光学,2018,11(6):916-918.
- [10] 罗元,苏琴,张毅,等. 基于优化RBPF的同时定位与地图构建[J]. 华中科技大学学报(自然科学版),2016,44(5):30-34.
- [11] 朱朔凌,毛建旭,王耀南,等. 基于惯性导航角度补偿的室内激光SLAM方法[J]. 电子测量与仪器学报,2020,33(3):2-4.
- [12] 于镭,张国强,王泽龙. 基于单舵轮搬运机器人的导航系统设计[J]. 电子测量技术,2020,43(18):11-12.
- [13] 李晓婷,史健芳. 基于重采样技术改进的粒子滤波算法[J]. 微电子学与计算机,2016,33(9):164-168.
- [14] 王晓初,李宾,刘玉县,等. 一种基于改进卡尔曼滤波的姿态解算算法[J]. 科学技术与工程,2019(24):71.
- [15] 周旭. 基于改进粒子滤波的SLAM算法研究[D]. 南京:南京理工大学,2014.
- [16] 伍永健,陈跃东,陈孟元. 量子粒子群优化下的RBPF-SLAM算法研究[J]. 智能系统学报,2018,13(5):829-835.

## 作者简介

何佳泽,硕士研究生,主要研究方向为移动机器人SLAM建图和路径规划。

张寿明(通信作者),博士,教授,主要研究方向为复杂工业工程控制。

E-mail:1411834974@qq.com