



浙 江 科 技 学 院
本 科 毕 业 设 计
(2020 届)

| | |
|--------|-----------------------------|
| 题 目 | <u>基于多传感器融合的室内移动机器人定位系统</u> |
| 学 院 | <u>自动化与电气工程学院</u> |
| 专 业 | <u>自动化</u> |
| 班 级 | <u>自动化 163</u> |
| 学 号 | <u>1160399106</u> |
| 学生姓名 | <u>刘力铭</u> |
| 指导教师 | <u>孙勇智（讲师）</u> |
| 完成日期 | <u>2020 年 5 月 28 日</u> |

摘 要

随着科学技术的不断进步，移动机器人技术在越来越多的领域中得到应用。自主移动机器人对环境的识别以及精确定位是机器人智能化的关键，并且成为近年来的研究焦点。本文以无法安装车轮式里程计的低成本机器人为研究对象，提出了一种融合激光雷达和惯性测量单元的室内定位系统，并在四轮移动机器人上得到实现。

本设计使用 RF2O (Range Flow-based 2D Odometry) 算法从激光雷达的数据变化中计算得到机器人的运动数据，并使用扩展卡尔曼滤波(Extended Kalman Filter)融合惯性测量单元数据和 RF2O 算法计算结果得到初步定位数据，最后在此基础上使用自适应蒙特卡洛定位算法(Adaptive Monte Carlo Localization)对定位数据进行修正，以提高测量精度。

为了测试算法的性能，本设计首先在 Gazebo 仿真软件中进行仿真实验，定量计算了传感器数据通过不同算法处理后得到的误差。然后设计了室内四轮移动机器人实体，并在实际环境中进行了定位实验。仿真和现实实验结果表明：本设计提出的传感器数据处理方案具有比较高的精度和稳定性，并且避免了误差累计的问题。另外，该数据处理方案计算量较小，并且不依赖里程计传感器，可以满足低成本室内移动机器人的定位需求。

关键词：移动机器人；机器人操作系统；扩展卡尔曼滤波；室内定位

Abstract

With the development of science and technology, mobile robot technology has been applied in more and more fields. The recognition of the environment and localization play essential roles in the autonomous robot technology, and has become the research focus in recent years. In this paper, a new kind of indoor localization system, which combines lidar and inertial measurement unit, is proposed for the research of low-cost robot which cannot install wheel odometer, and this system is implemented in a four-wheel mobile robot.

In this design, rf2o (range flow based 2D odometry) algorithm is used to calculate the motion data of the machine from the data changes of lidar, and extended Kalman Filtering is used to fuse the inertial measurement unit data and rf2o algorithm to get the inaccurate pose of robot. Finally, Adaptive Monte Carlo localization algorithm is used to modify the data in order to improve the measurement accuracy.

In order to test the performance of the algorithm, the simulation experiment is carried out in the gazebo simulation software firstly, and the error of pose data processed by different algorithms is calculated quantitatively. Then the indoor four-wheel mobile robot is designed, and the localization experiment is carried out in the real environment. The simulated and real experiments show that the sensor data processing scheme proposed in this design has high precision and stability, and avoids the problem of error accumulation. In addition, the data processing scheme does not rely on odometer sensor, and the calculation is low computational cost, which makes it suitable for those robotic applications that require planar odometry.

Keywords: Mobile robots, Robot Operating System, Extended Kalman Filter, Localization

目录

| | |
|------------------------------|----|
| 摘要 | I |
| Abstract..... | II |
| 第 1 章 绪论 | 1 |
| 1.1 论文研究的背景及意义 | 1 |
| 1.2 国内外研究现状分析 | 2 |
| 1.3 论文内容与结构安排 | 4 |
| 第 2 章 移动机器人运动平台及其数学模型 | 5 |
| 2.1 运动平台简介 | 5 |
| 2.2 系统设计框架 | 6 |
| 2.3 坐标系统 | 7 |
| 2.4 机器人运动学模型 | 8 |
| 第 3 章 传感器数据处理研究 | 10 |
| 3.1 激光雷达数据处理 | 11 |
| 3.1.1 RF2O 算法基本原理..... | 11 |
| 3.1.2 RF2O 算法实现效果..... | 12 |
| 3.2 扩展卡尔曼滤波 | 15 |
| 3.2.1 扩展卡尔曼滤波基本原理 | 15 |
| 3.2.2 扩展卡尔曼滤波在定位系统上的实现 | 16 |
| 3.2.3 扩展卡尔曼滤波实现效果 | 17 |
| 3.3 自适应蒙特卡洛定位算法 | 21 |
| 3.3.1 自适应蒙特卡洛定位算法概述 | 21 |
| 3.3.2 自适应蒙特卡洛定位算法实现效果 | 22 |
| 第 4 章 基于 ROS 的机器人定位系统实现..... | 24 |
| 4.1 ROS 节点整体设计 | 24 |
| 4.2 坐标节点设计 | 25 |
| 4.3 运动控制节点设计 | 27 |
| 4.3.1 PID 控制基本原理..... | 27 |

| | |
|----------------------------------|-----------|
| 4.3.2 PID 控制实现..... | 27 |
| 4.4 定位实验与结果分析 | 28 |
| 第 5 章 非技术因素的考虑与方案评估 | 30 |
| 5.1 本课题对非技术因素的考虑 | 30 |
| 5.2 非技术层面的方案评估 | 30 |
| 5.3 成本估算与经济决策 | 30 |
| 5.4 本课题的创新性 | 31 |
| 结论 | 32 |
| 参考文献 | 33 |
| 致谢 | 35 |

第 1 章 绪论

1.1 论文研究的背景及意义

进入 21 世纪，机器人技术得到快速发展。机器人是有着高度智能的自动化机器，能够拥有感知、协同、动作等等和人类一样的智能特性，涉及到自动化、传感器技术、机械制造、计算机应用等多门学科的专业知识。近几年来，机器人在人类的生产生活中扮演着越来越重要的角色，比如物流装配、设施监测、流水线加工等等。与普通的工业机器人相比，自主移动机器人要求更高的自动规划能力、适应能力和运动能力，并且能够在更复杂的环境下完成工作。推动移动机器人技术不断完善可以使得移动机器人在各行各业中发挥巨大的作用，给人类的生产生活带来巨大改善。

自主移动机器人对环境的识别，以及在环境中的精确定位是实现其自主移动的重要条件，也是机器人智能化的关键。为了达成这一目标，国内外学者在过去的几十年时间中对机器人的定位问题进行广泛的研究。其中，在已知环境下的实现精准定位已经有了可靠的解决方案。因为移动机器人往往工作在不确定性较高的环境中，所以针对未知环境下的机器人定位方案的研究具有更高的价值。然而，机器人在未知环境下完成定位任务还有很多需要完善的理论和需要解决的技术问题。

目前，机器人室内定位最常用的解决方案是使用车轮式里程计和激光雷达传感器，通过 SLAM（Simultaneous Localization and Mapping, SLAM）技术完成室内定位和导航。但在实际生活中，有一部分机器人，由于轮子打滑或者机械结构限制等等原因无法使用车轮式里程计作为运动数据检测的传感器。而视觉传感器存在对 CPU 计算能力要求高、购买成本高、算法复杂等缺陷，因此并不适合在低成本的机器人系统上安装使用。为了解决这类低成本机器人系统的定位问题，本论文主要研究移动机器人在仅使用激光雷达和惯性测量单元作为传感器的情况下，通过传感器信息融合，精确地测量机器人的里程计数据，完成室内定位任务。

1.2 国内外研究现状分析

世界上最早的机器人可以追溯到上个世纪 60 年代，美国斯坦福研究所研制出了世界上第一台能够实现自主移动的机器人 Shakey。该机器人系统配备了多种传感器，如三角测距仪、电子摄像机、碰撞传感器等，能简单完成环境感知、运动规划和运动控制任务。由于受限于当时的计算机技术，Shakey 需要花费很长的时间来感知和分析周围的环境。

在过去的几十年时间中，移动机器人的定位技术受到国内外学者的高度关注，并进行广泛研究与应用。多年来，很多优秀的传感器和算法选择方案被提出来，使得不同机械结构的机器人能在不同的环境中，完成定位、环境感知和导航任务。

在室外可以利用已知的卫星地图数据和 GPS 全球定位系统实现准确定位。如在汽车上配备激光雷达和差分全球定位系统，采用激光雷达测距与建图算法进行三维同步定位与建图，从而保证了无人驾驶汽车环境感知系统的准确性和稳定性^[1]。然而在实际生活中，机器人往往工作在室内或者环境未知的环境中。在这种情况下，经常存在 GPS 信号差或者 GPS 误差过大而无法完成定位任务的情况。

此时，SLAM 技术是目前解决室内移动机器人的定位与环境感知问题的有效方法。SLAM 技术是指机器人系统在运动的过程中，通过自身搭载的传感器，完成对机器人运动数据和定位信息的计算以及对周围环境的感知。目前机器人室内定位最常用的解决方案是使用车轮式里程计和惯性测量单元测量机器人的运动数据，使用激光雷达传感器测量周围的障碍物信息，通过 SLAM 技术完成室内定位和导航^[2]。

在实际生活中，有一部分机器人因为一些原因无法安装车轮式里程计。为了解决这个问题，可以使用视觉里程计代替车轮式里程计，通过 VIO-B-SLAM、DPPTAM 等图像处理算法，从摄像头采集到的图像变化中得到机器人的线速度、角速度等数据，从而完成室内定位任务^[3,4]。

除了使用摄像头代替车轮式里程计以外，有研究员提出了激光雷达和惯性测量单元相融合的导航定位系统，用惯性测量单元采集的姿态信息通过卡尔曼滤波计算出较为精确的机器人运动数据^[5]。但是因为惯性测量单元无法抵消重力加速度分量，测量的位移数据会随时间的推移积累过多的横向误差，所以仅仅使用惯性测量单元完成机器人定位任务是不可靠的。

文献[6,7,8]分别提出了多种从激光雷达的数据变化中计算出机器人运动数据的算法，使得仅使用激光雷达和惯性测量单元完成精确定位成为可能。在该背景下，本论文打算结合这些激光雷达数据处理算法得到机器人运动的线速度和角速度数据，完成机器人定位任务。

在 SLAM 技术方面，SLAM 算法的研究方向最早是基于卡尔曼滤波（Kalman Filtering, KF）的 SLAM 算法^[9]，比如 EKF-SLAM。这些算法将非线性系统通过泰勒展开后保留一部分低次项来近似原系统，由于该近似方法比较简单，因此这些算法计算量通常较小，但由于忽略了泰勒展开式中的高阶项，算法误差较大，并且对于复杂度较高的非线性系统，滤波算法容易发散。所以，该算法适用于处理器性能较差并且对系统实时性要求较高的简单机器人系统中。

在上个世纪 90 年代，基于粒子滤波的 SLAM 算法被提出并引起重视^[10]，这些算法使用粒子来模拟非线性系统，通过将粒子输入到非线性系统中，根据粒子的输出来近似原系统，因此这些算法可以精确的还原非线性系统。通常这些算法计算量较大，但是在以激光雷达为传感器的机器人系统中表现出了更良好的稳定性和可靠性，如 Fast-SLAM^[11]、RBPF-SLAM^[12]算法。

综上，室内定位系统的发展大致可以分为两个方向。一种是追求定位精度的系统，这种系统大多使用摄像头、激光雷达和车轮式里程计等传感器，通过基于粒子滤波的 SLAM 算法融合传感器数据得到误差较小的定位数据。这种系统通常算法计算量大，对系统硬件条件要求高，因而系统制作成本高。而另一种系统要求低成本和高实时性，这种系统大多选择激光雷达、惯性测量单元和车轮式里程计作为传感器，通过基于卡尔曼滤波的 SLAM 算法融合传感器数据得到误差可以接受的定位数据。而本设计的初衷是为无法安装车轮式里程计的低成本机器人提供一种新的传感器和算法选择方案，因此选用后一种系统的设计理念来完成本设计。

1.3 论文内容与结构安排

本课题针对室内移动机器人定位算法进行了研究，提出了一种新的传感器数据处理方案，并将该方案应用在一个能够在室内自由移动的机器人系统上，使得机器人系统在仅安装惯性测量单元和激光雷达的情况下，通过扩展卡尔曼滤波以及自适应蒙特卡洛定位算法进行多传感器数据融合，从而完成定位任务。全文分为五个章节，论文的内容安排如下：

第一章主要描述了室内移动机器人定位算法的研究背景及意义，并对国内外研究现状做了综述。

第二章主要介绍了机器人运动平台，进而描述了室内移动机器人的总体设计框架，最后分析并建立机器人系统的坐标模型以及运动模型。

第三章是传感器数据处理研究。本章主要介绍了 RF2O 算法、扩展卡尔曼滤波算法以及自适应蒙特卡洛定位算法的基本原理，并通过仿真软件和现实环境中的实验分析各个算法的性能。

第四章主要阐述了 ROS 操作系统的基本概念，进而描述了在 ROS 操作系统下的机器人定位系统设计，然后介绍了机器人的坐标节点设计和运动控制节点设计，最后通过现实环境中的定位实现证明了定位系统的可行性。

第五章主要介绍了非技术因素的考虑与方案评估。其中包括本课题对非技术因素的考虑、非技术层面的方案评估、成本估算与经济决策以及本课题的创新性分析。

第 2 章 移动机器人运动平台及其数学模型

2.1 运动平台简介

本设计使用的机器人平台如图 2-1 所示。其中，机器人主要硬件电路为车载电脑和 KEA128 单片机主板。车载电脑上安装 Ubuntu16.04 以及 Kinect 版本的 ROS 操作系统。电脑通过 USB 接口连接 LS01G 激光雷达、ART-IMU-02A 姿态传感器，通过相应的驱动程序采集传感器数据。同时，车载电脑连接路由器可以实时发送数据给上位机电脑，方便人机交互。单片机主板包含提供整个系统正常工作的电源模块，同时通过串口与车载电脑通讯，使用 KEA128 单片机驱动电机与舵机，完成机器人的运动控制。



图 2-1 移动机器人实验平台

2.2 系统设计框架

在 ROS 的软件架构下，系统设计结构的模块化和层次化有助于增加机器人系统的稳定性，并方便调试和查找问题。基于该思想，设计室内移动机器人定位系统总设计架构如图 2-2 所示，包括硬件层、驱动层、操作系统层以及人机交互层共四个层次。

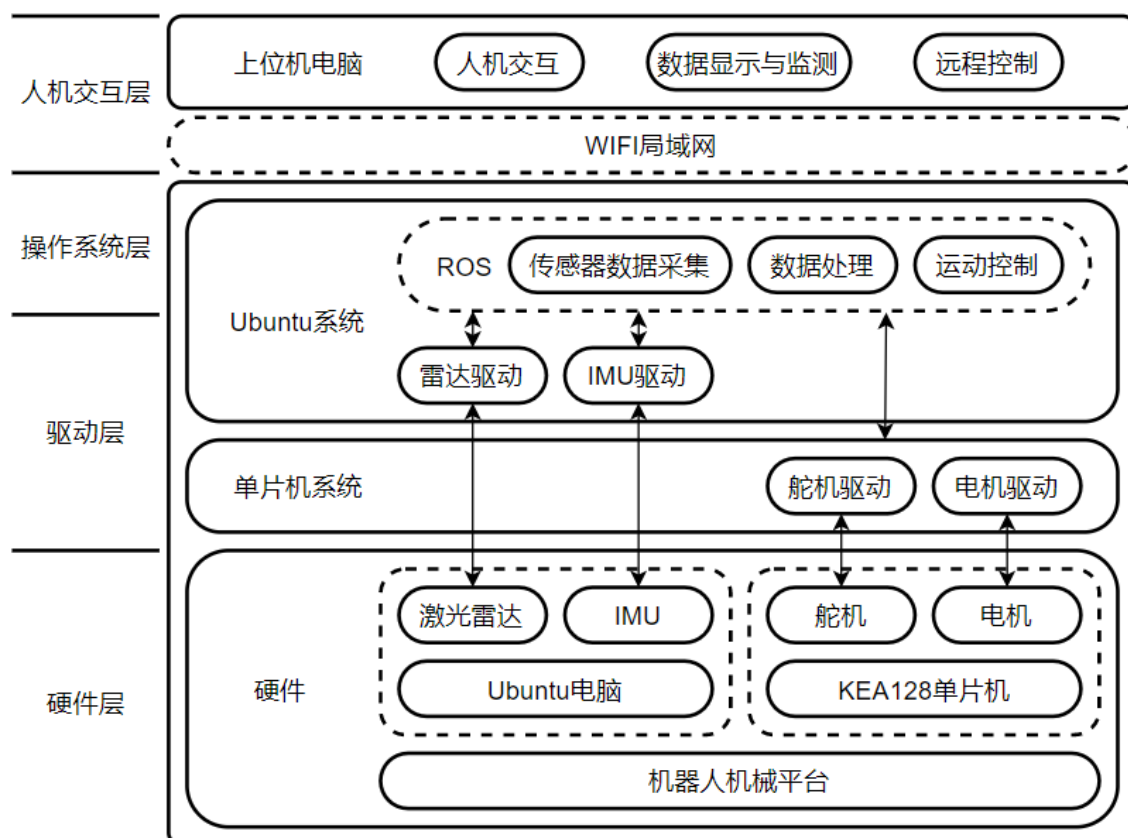


图 2-2 室内移动机器人定位系统总设计架构图

硬件层包括控制单元、外围电路以及机械平台。控制单元为一台车载电脑和 KEA128 单片机。两个控制单元根据自身的性能和外部接口驱动不同硬件模块。其中 KEA128 单片机是 NXP 公司旗下一款基于 ARM-Cortex M0+ 系列内核的单片机，它具有低功耗、宽电压工作、稳定性强等特点。芯片内自带 PWM 信号输出模块，通过驱动电路，控制舵机打角和电机输出扭矩；车载电脑是以 I3-7100U 为处理器的 ART-I3U7 电脑，电脑上运行 Ubuntu 操作系统以及操作系统中的 ROS 软件框架。电脑通过 USB 连接激光雷达模块、惯性测量单元模块和 KEA128 单片机。机器人机械平台即两驱车运动平台，前轴使用舵机控制转弯，后轴使用单电机控制速度，能够在室内自由移动。

在驱动层中，KEA128 单片机负责电机和舵机的驱动，它通过串口得到车载电脑输出的决策

信号，并输出电机和舵机的驱动信号，以此来驱动机器人达到预期的运动状态。车载电脑负责对传感器和单片机进行通讯，它通过 ROS 框架中对应的驱动程序节点，完成对激光雷达和惯性测量单元的初始化、设置和数据采集，以及对单片机输出决策信号。

在操作系统层中，各个模块分别是 ROS 框架中的各个节点，它们主要负责对传感器采集到的数据进行处理、融合以及根据计算的数据控制机器人的运动。

最后，人机交互层主要由个人笔记本电脑或者台式电脑组成。机器人系统通过 WIFI 局域网发送数据与远端电脑通讯。远端电脑通过 Ubuntu 中的 SSH 指令，远程登录到车载电脑中，可以完成实时数据监测、数据可视化以及远程指令发送等功能。极大程度上方便了人机交互，简化了调试流程，提高开发效率。

2.3 坐标系

在移动机器人研究中，使用笛卡尔坐标系可以大大方便描述机器人的姿态数据。本文用笛卡尔坐标系 Σ_w 描述世界坐标系，用 Σ_R 描述机器人坐标系。

由于本论文仅研究机器人在平面上的运动并且假设移动过程中机器人可视为刚体，那么机器人的各坐标系如图 2-3 所示。图中，为了简化机器人模型，将机器人模型简化为车的一轴；机器人坐标系为 $x_{local} o_{local} y_{local}$ ，该坐标系的坐标原点是机器人的重心；世界坐标系为 $x_{world} o_{world} y_{world}$ ，该坐标系原点是机器人运动的出发点；假设在 t 时刻机器人所在的位置相对运动起点的位移为 (x_t, y_t) ，偏航角为 θ_t ，那么，在 t 时刻机器人在世界坐标系下的姿态可以表示为 (x_t, y_t, θ_t) 。

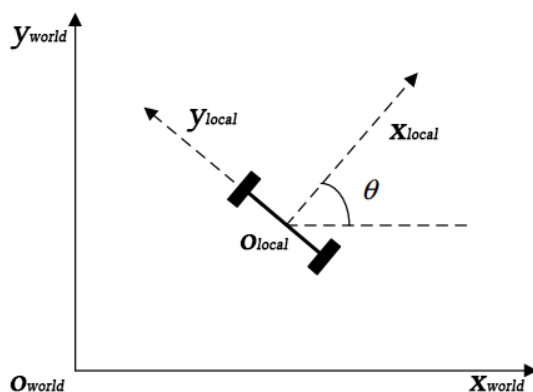


图 2-3 移动机器人坐标系

世界坐标系和机器人坐标系可以经过矩阵变换相互转换，其中，机器人坐标系转换为世界坐

标系的公式为：

$$\begin{pmatrix} x_{world} \\ y_{world} \\ \theta_{world} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{local} \\ y_{local} \\ \theta_{local} \end{pmatrix} \quad (2-1)$$

世界坐标系转换为机器人坐标系的公式为：

$$\begin{pmatrix} x_{local} \\ y_{local} \\ \theta_{local} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{world} \\ y_{world} \\ \theta_{world} \end{pmatrix} \quad (2-2)$$

2.4 机器人运动学模型

机器人的运动模型如图 2-4 所示，机器人的前轴使用舵机控制转弯，后轴使用单电机控制速度。根据几何关系，机器人的移动速度 v 是后轴左轮速度 v_l 与后轴右轮速度 v_r 的平均值，即：

$$v = \frac{v_r + v_l}{2} \quad (2-3)$$

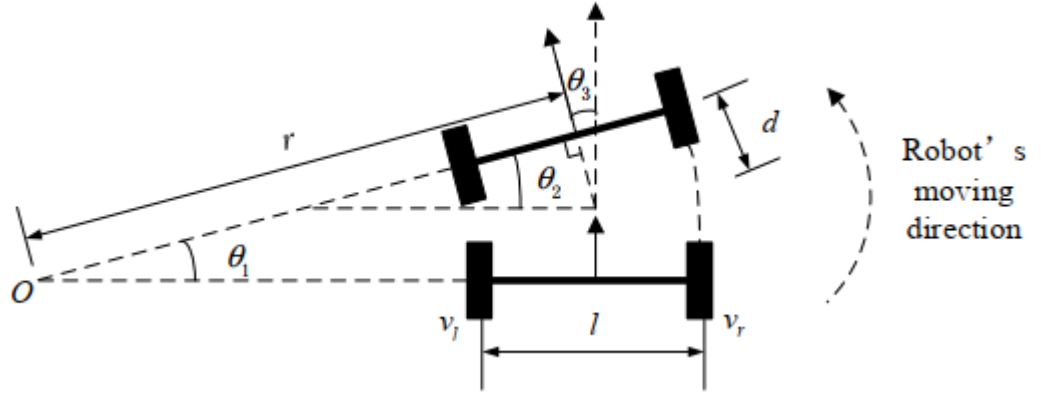


图 2-4 机器人的运动学模型

为了简化公式的难度，近似地认为 $\theta_1 = \theta_2 = \theta_3$ ，其中 θ_3 为机器人的转向角度。那么根据几何关系，就可以推导出右轮与左轮的距离 d 与转弯半径 r 之间的关系。同时，由于每次位移所经历的时间 Δt 和角度 θ_2 很小，可以近似地推出如下关系：

$$\theta_2 \approx \sin(\theta_2) = \frac{d}{l} = \frac{(v_r - v_l)\Delta t}{l} \quad (2-4)$$

那么，机器人的转弯角速度 ω 和转弯半径 r 与速度的关系式可以表达为：

$$\omega = \frac{\theta_1}{\Delta t} = \frac{v_r - v_l}{l} \quad (2-5)$$

$$r = \frac{v}{\omega} = \frac{(v_r + v_l) \cdot l}{2(v_r - v_l)} \quad (2-6)$$

根据前文坐标系统中推导的机器人坐标系转换到世界坐标系公式(2-2)，由于机器人的机械特性，机器人在坐标轴 y_{local} 上的速度始终为 0，则式(2-2)可以化简为：

$$\begin{pmatrix} x_{world} \\ y_{world} \\ \theta_{world} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{local} \\ \theta_{local} \end{pmatrix} \quad (2-7)$$

假设经过一段时间间隔 Δt 后，后轴左右轮分别经过的位移为 Δs_l 和 Δs_r ，机器人模型平移的距离为 $\Delta s = v\Delta t$ ，那么在时间 Δt 中，机器人的姿态变化关系式为：

$$\Delta \theta = \frac{\Delta s_r - \Delta s_l}{l} \quad (2-8)$$

$$\Delta s = \frac{\Delta s_l + \Delta s_r}{2} \quad (2-9)$$

$$Pose = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \Delta s \cos\left(\theta + \frac{\Delta \theta}{2}\right) \\ \Delta s \sin\left(\theta + \frac{\Delta \theta}{2}\right) \\ \Delta s \cos\left(\theta + \frac{\Delta \theta}{2}\right) \end{pmatrix} \quad (2-10)$$

第 3 章 传感器数据处理研究

机器人的定位数据可以分为相对定位和绝对定位。相对定位是指基于惯性传感器、里程计等可以获取机器人运动数据的传感器，通过积分公式获得的机器人相对初始状态的姿态数据。由于传感器测量总是存在误差，在积分的过程中，定位数据的误差会随时间逐渐累积增大，因此相对定位数据在长时间运行的机器人系统中并不可靠；绝对定位则是指根据一些已知的参照信息，如地图数据，通过激光雷达等能够获取周围环境信息的传感器，从而计算出机器人相对参照目标的姿态数据。绝对定位数据不会有误差的累计问题，故适合作为长期参考，但是定位数据不连续，存在跳变。

根据相对定位法以及绝对定位法的优劣势，设计传感器数据处理方案框图如图 3-1 所示。

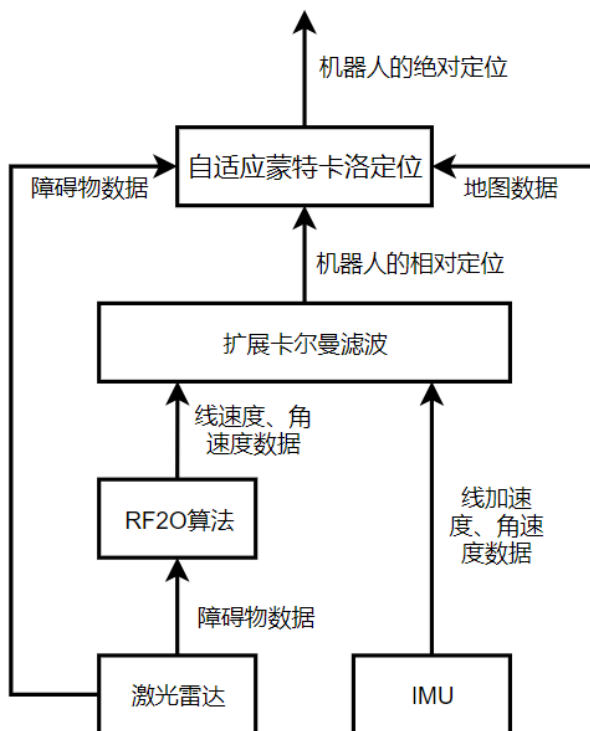


图 3-1 传感器数据处理方案框图

该方案需要使用惯性测量单元和激光雷达传感器。惯性测量单元能够测量机器人的线加速度和角速度。激光雷达能够测量机器人周围的障碍物数据，该数据通过 RF20 算法计算得到机器人的线速度和角速度。使用扩展卡尔曼滤波融合惯性测量单元数据和 RF20 算法计算结果得到机器人的相对定位数据。最后，为了克服相对定位数据的误差累计问题以及在该数据的基础上进一步减少测量误差，使用自适应蒙特卡洛定位算法来修正机器人的相对定位，得到系统的绝对定位数据。通过绝对定位数据可以得到比较准确并且没有误差累计的机器人位移、偏航角数据，通过相

对定位可以得到准确并且连续的机器人线速度、角速度、加速度数据。

3.1 激光雷达数据处理

3.1.1 RF2O 算法基本原理

低成本惯性测量单元中的加速度传感器测量噪声较大，并且在机器人运行的过程中，机械结构的震动会对加速度传感器的测量造成较大影响。因此，通过加速度积分得到位移数据是不可靠的，需要引入另外的传感器来完成定位任务。激光雷达传感器能够测量周围环境的障碍物数据，但是无法直接测量位移、速度等数据。RF2O (Range Flow-based 2D Odometry)算法基于激光雷达测量的环境数据，根据数据的变化近似地计算出激光雷达的速度、角速度数据^[6]。

激光雷达的扫描方式如图 3-2 所示，以激光雷达的扫描中心为原点，建立极坐标系 $\Sigma_L(r, \theta)$ 。

当激光雷达在 t 时刻对周围障碍物进行扫描时，得到一组雷达扫描数据 $U_t = \{d_\alpha, \alpha = 1, 2, \dots, N\}$ ，其中 N 代表单次扫描的扫描样本总数，本设计使用的激光雷达 $N=360$ ， d_α 代表该次扫描中第 α 个角度 θ 测得的障碍物距离。为方便计算，使用 $R(t, \alpha)$ 来表示在 t 时刻，第 α 个角度测得某点 P 的障碍物距离。由于建立的极坐标系 Σ_L 方向与本设计使用的激光雷达数据方向存在 180 度旋转，则 α 与 θ 的关系式为：

$$\alpha = \frac{N-1}{FOV} \theta + \frac{N-1}{2} = k_\alpha \theta + \frac{N-1}{2} \quad (3-1)$$

式中，FOV 是激光雷达的扫描角度，本文使用的激光雷达为 360 度。

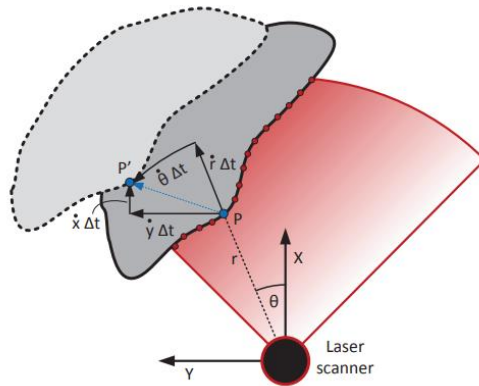


图 3-2 激光雷达扫描示意图

假设 R 是可微的，那么下一时刻该点 P 扫描的距离可以用泰勒展开表示：

$$R(t + \Delta t, \alpha + \Delta \alpha) = R(t, \alpha) + \frac{\partial R}{\partial t}(t, \alpha) \Delta t + \frac{\partial R}{\partial \alpha}(t, \alpha) \Delta \alpha + O(\Delta t^2, \Delta \alpha^2) \quad (3-2)$$

为了简化表达式，舍去泰勒展开后的高次项，那么 P 点距离的变化率可以表示为：

$$\frac{\Delta R}{\Delta t} = \frac{R(t + \Delta t, \alpha + \Delta \alpha) - R(t, \alpha)}{\Delta t} \simeq \frac{\partial R}{\partial t}(t, \alpha) + \frac{\partial R}{\partial \alpha}(t, \alpha) \frac{\Delta \alpha}{\Delta t} = R_t + R_\alpha \frac{\Delta \alpha}{\Delta t} \quad (3-3)$$

式中， $\dot{r} = \Delta R / \Delta t$ 即该点 P 的平均移动速度； $\dot{\alpha} = \Delta \alpha / \Delta t$ 即该点 P 的平均角速度。用式(3-1)代入式(3-3)可得：

$$\dot{r} \simeq R_t + R_\alpha \dot{\alpha} = R_t + R_\alpha k_\alpha \dot{\theta} \quad (3-4)$$

由于 $(\dot{r}, \dot{\theta})$ 都是在极坐标系之下，需要转换到笛卡尔坐标系之下，转换公式如下：

$$\begin{cases} \dot{r} = \dot{x} \cos \theta + \dot{y} \sin \theta \\ r \dot{\theta} = \dot{y} \cos \theta - \dot{x} \sin \theta \\ \dot{x} = -v_{x,s} + y \omega_s \\ \dot{y} = -v_{y,s} - x \omega_s \end{cases} \quad (3-5)$$

将式(3-5)代入式(3-4)可得激光雷达速度的约束方程。

$$\begin{aligned} & \left(\cos \theta + \frac{R_\alpha k_\alpha \sin \theta}{r} \right) v_{x,s} + \left(\sin \theta - \frac{R_\alpha k_\alpha \cos \theta}{r} \right) v_{y,s} \\ & + (x \sin \theta - y \cos \theta - R_\alpha k_\alpha) \omega_s + R_t = 0 \end{aligned} \quad (3-6)$$

根据该公式，可以直接从激光雷达的数据变化中计算出激光雷达的速度、角速度数据。但是该公式的误差产生于式(3-3)舍去了泰勒展开后的高次项，以及假设了周围环境是静止的。因此需要将该公式计算出来的数据融合惯性测量单元的测量结果以得到更精准的结果。

3.1.2 RF2O 算法实现

在现实生活中，由于传感器的误差总是存在的，测量出机器人运动数据的真实值是不可能实现的，因此，也就无法在现实的环境下定量地测量算法的性能。本文使用机器人仿真平台 Gazebo 搭建仿真环境，并在该仿真软件中测试 RF2O 的性能。Gazebo 是基于物理引擎的仿真软件，可以精确地模拟出复杂的机器人工作环境并测试机器人系统算法的可靠性。

在 RF2O 算法仿真中，给仿真机器人的激光雷达传感器配置和 LS01G 激光雷达传感器一样的参数，并在激光雷达的每一个样本的测量上增加一个服从 $N(0,0.01)$ 的高斯噪声。分别在机器人静止、直线和旋转三种情况下，计算测量的均方根误差（Root Mean Squared Error, RMSE）来体现测量的精度。其中，RMSE 的计算公式为：

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x_i) - y_i)^2} \quad (3-7)$$

在静止实验中，控制机器人的线速度与角速度为零，保持 30 秒，记录机器人的运动数据。由于机械结构上的限制，机器人只能产生 X 轴的线速度以及 Z 轴的角速度，因此仅记录这两组数据。测量部分结果如图 3-3 和图 3-4 所示，算法在 X 轴线速度的误差平均值为 -0.000134，均方根误差为 0.01384；在 Z 轴角速度的静态误差平均值为 0.000157，均方根误差为 0.0165。总体来说，Z 轴角速度的静态测量噪声略大于 X 轴线速度的静态噪声，算法的静态误差比较小，可以接受。

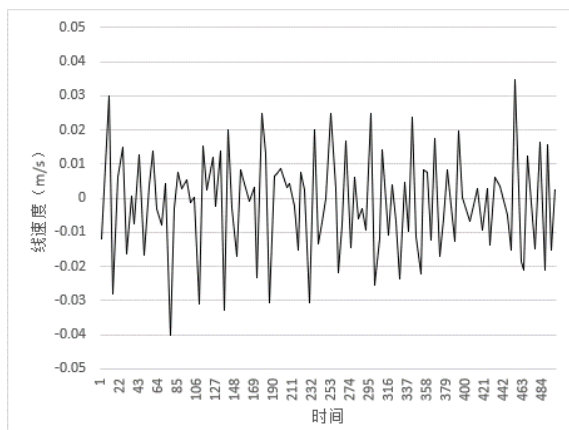


图 3-3 X 轴线速度静态噪声测量图

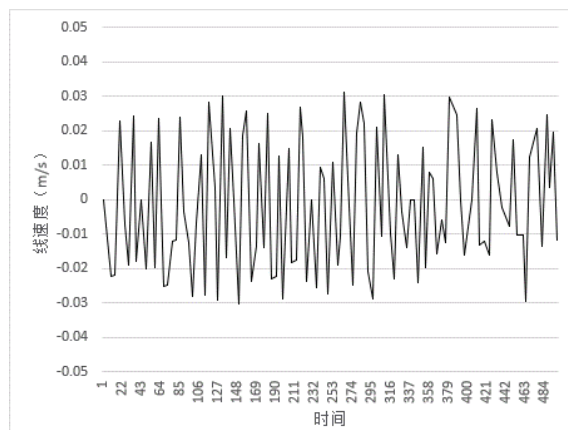


图 3-4 Z 轴角速度静态噪声测量图

在直行实验中，仅仅控制机器人缓慢地加速直行，保持 30 秒，记录机器人在 X 轴上的线速度并通过积分得到位移数据。测量部分结果如图 3-5 所示，积分后得到的位移数据如图 3-6 所示。算法在 X 轴线速度的动态误差平均值为 -0.00092，均方根误差为 0.04。从测量结果可以看出，RF2O 算法在机器人运动速度较小时，测量误差比较小，当机器人运动线速度大于 0.25m/s 时，测量的误差变大。对于位移的计算来说，线速度计算位移是积分的过程，虽然大致上测量值与真实值保持一致，但误差随时间逐渐增大。

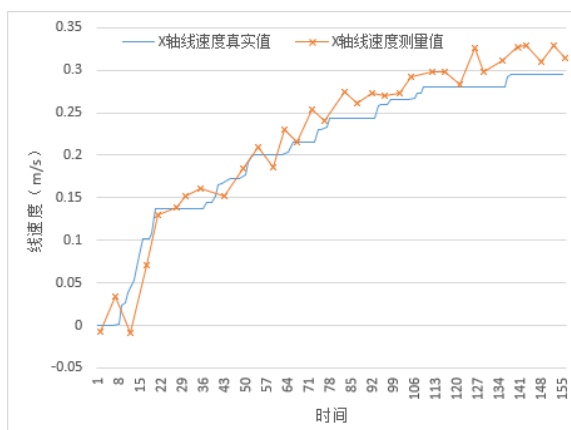


图 3-5 X 轴线速度动态测量图

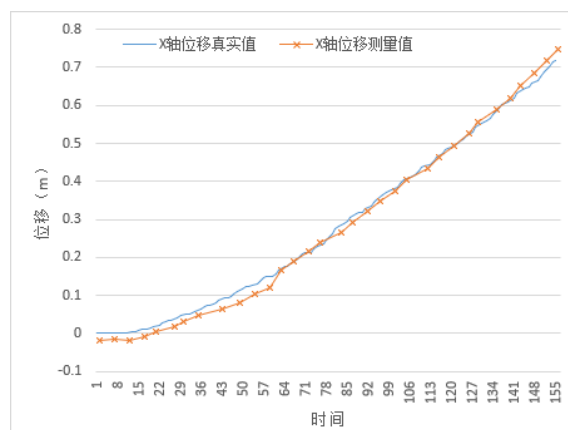


图 3-6 积分得到 X 轴位移动态测量图

在旋转实验中，仅仅控制机器人定轴加速旋转，保持 30 秒，记录机器人在 Z 轴上的角速度并通过积分得到偏航角数据。测量部分结果如图 3-7 所示，积分后得到的偏航角数据如图 3-8 所示。算法在 Z 轴角速度的动态误差平均值为-0.0028，均方根误差为 0.039。从测量结果可以看出，RF2O 算法在机器人角速度较小时，测量误差比较小，当机器人运动线速度大于 0.25r/s 时，测量的误差变大。对于偏航角的计算来说，同样误差随时间逐渐增大。

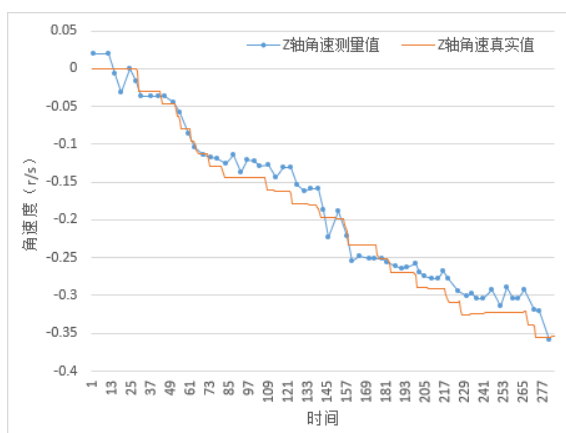


图 3-7 Z 轴角速度动态测量图

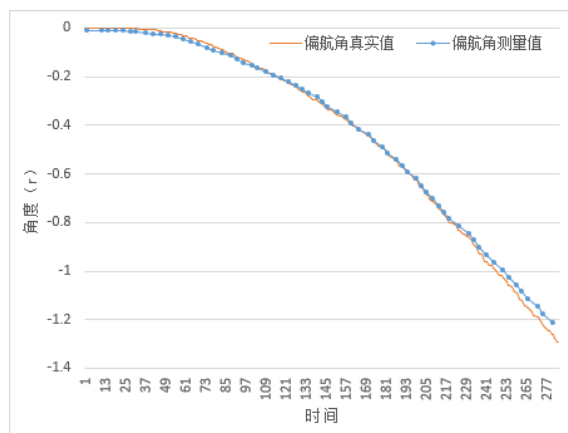


图 3-8 积分得到 Z 轴偏航角动态测量图

综上，得出 RF2O 算法在不同实验中的测量误差如表 3-1 所示。RF2O 算法在三个场景下的误差平均值都无限接近于零，说明测量漂移较小。RF2O 算法在直行和旋转的情况下测量噪声相差不大，但都远远大于静态时的测量噪声。并且，随着运动速度的增大，测量误差呈现增加的趋势。尽管如此，RF2O 的测量噪声依然在可以接受的范围之内。由于偏航角和位移的计算都是积分的过程，误差随时间逐渐增大，故仅使用该算法无法完成机器人准确定位任务。

表 3-1 RF2O 算法在不同场景下的测量误差表

| | 静态 | 直行 | 旋转 |
|-------|-----------|---------|---------|
| 误差平均值 | -0.000134 | 0.00092 | -0.0028 |
| 均方根误差 | 0.01384 | 0.04 | 0.039 |

3.2 扩展卡尔曼滤波

3.2.1 扩展卡尔曼滤波基本原理

卡尔曼滤波在 19 世纪 60 年代由 Kalman 提出，它建立在线性代数和隐马尔可夫模型上，是一种递归的估计^[9]。但是该滤波器假设系统的状态转换模型和传感器观测模型都是线性模型，实际上，大多数工程的动态系统都并不确切的符合这个模型。为了克服系统的非线性，扩展卡尔曼滤波对非线性函数采用泰勒级数展开，并忽略高阶项，将非线性的模型近似地线性化，从而能更好地解决包含噪声的非线性估计问题。

在机器人模型中，假设系统的真实状态与上一时刻的状态和当前系统输入有关，系统的测量状态与当前系统的真实状态有关，则建立系统的状态转换模型和传感器观测模型如下：

$$\begin{cases} x_k = f(x_{k-1}, u_k) + w_k \\ z_k = h(x_k) + v_k \end{cases} \quad (3-8)$$

式中， k 表示离散时间； x_k 表示在 k 时刻下系统的真实状态； u_k 表示输入矩阵； w_k 表示过程噪声，假定符合 $w_k \sim N(0, Q_k)$ 的多元高斯噪声； z_k 表示在 k 时刻下的测量； v_k 表示测量噪声，假定符合 $v_k \sim N(0, R_k)$ 的多元高斯噪声； $f(x, u)$ 、 $h(x)$ 分别为非线性可微的状态转换函数和测量函数。

基于以上模型，扩展卡尔曼滤波求解分为两个阶段，分别为预测阶段和更新阶段。

(1) 预测阶段：上一时刻的最佳估计 $\hat{x}_{k-1|k-1}$ 和协方差矩阵 $P_{k-1|k-1}$ 被用来计算当前时间的预测 $\hat{x}_{k|k-1}$ 和预测协方差矩阵 $P_{k|k-1}$ 。由于状态换函数 $f(x, u)$ 为非线性函数，扩展卡尔曼滤波使用泰勒级数展开后保留低次项，将函数近似地线性化后得到状态转换矩阵 $F_k = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k-1|k-1}, u_k}$ ，预测阶段

求解公式如下：

$$\begin{cases} \hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \\ P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \end{cases} \quad (3-9)$$

(2)更新阶段：同样测量函数 $h(x)$ 被用泰勒级数展开后保留低次项，得到观察矩阵 $H_k = \frac{\partial h}{\partial x} \Big|_{\hat{x}_{k|k-1}}$ 。随后，在式(3-9)计算出来的预测协方差矩阵 $P_{k|k-1}$ 进一步计算得到卡尔曼增益 K_k 。最后，根据卡尔曼增益来补偿预测阶段得到的预测 $\hat{x}_{k|k-1}$ 和和预测协方差矩阵 $P_{k|k-1}$ 得到当前最佳估计 $\hat{x}_{k|k}$ 和协方差矩阵 $P_{k|k}$ ，更新阶段公式如下：

$$\begin{cases} K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \\ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1})) \\ P_{k|k} = (I - K_k H_k) P_{k|k-1} \end{cases} \quad (3-10)$$

更新阶段计算出的 $\hat{x}_{k|k}$ 就是在 k 时刻扩展卡尔曼滤波的估计结果。在 $k+1$ 时刻，该结果又是扩展卡尔曼滤波的输入，通过迭代，扩展卡尔曼滤波逐渐收敛，逐渐准确地估计动态系统的状态。但是扩展卡尔曼滤波也有缺陷：在复杂的系统中，如果扩展卡尔曼滤波近似的线性系统与原非线性系统误差过大，滤波器可能会迅速发散，从而无法得到准确的估计结果。尽管如此，扩展卡尔曼滤波还是因为它优秀的性能和计算量较小的特点在导航和 GPS 中被广泛使用。

3.2.2 扩展卡尔曼滤波在定位系统上的实现

基于扩展卡尔曼滤波的设计原理，需要建立室内移动机器人系统的状态转换模型和传感器观测模型。在状态转换模型方面，由于本设计的机器人系统仅考虑 2D 平面上的运动，故系统的状态矩阵 x 和输入矩阵 u 可表示为：

$$\begin{cases} u = [V_\theta \quad A_x \quad A_y]^T \\ x = [X \quad Y \quad \theta \quad V_x \quad V_y \quad V_\theta \quad A_x \quad A_y]^T \end{cases} \quad (3-11)$$

式中， X 、 Y 分别为机器人在 X、Y 轴上的位移； θ 为机器人的偏航角； V_x 、 V_y 分别为机器人在 X、Y 轴上的速度分量； V_θ 为机器人的旋转角速度； A_x 、 A_y 分别为机器人在 X、Y 轴上的加速度分量。

为了简化模型，假设机器人系统在运动的过程中，状态转换都是理想的，并且状态转化误差服从正态分布。根据状态矩阵中各个状态的物理意义，可以推导出各个状态的转换关系，即状态转换函数 $f(x, u)$ 可以用以下矩阵来表示：

$$\begin{bmatrix} 0 & 0 & 0 & \cos(\theta) \cdot \Delta t & -\sin(\theta) \cdot \Delta t & 0 & 0.5 \cdot \cos(\theta) \cdot \Delta t^2 & -0.5 \cdot \sin(\theta) \cdot \Delta t^2 \\ 0 & 0 & 0 & \sin(\theta) \cdot \Delta t & \cos(\theta) \cdot \Delta t & 0 & 0.5 \cdot \sin(\theta) \cdot \Delta t^2 & 0.5 \cdot \cos(\theta) \cdot \Delta t^2 \\ 0 & 0 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \Delta t & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & \Delta t \\ \vdots & \vdots & \vdots & \vdots & \vdots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \theta \\ V_x \\ V_y \\ V_\theta \\ A_x \\ A_y \end{bmatrix} \quad (3-12)$$

在传感器观测模型方面，机器人系统安装了一个惯性测量单元和激光雷达传感器。惯性测量单元能够测量机器人的加速度和角速度，而激光雷达通过 RF2O 算法可以计算出车模的 X、Y 轴线速度和 Z 轴角速度。为了简化模型，假设传感器的每次测量和数据处理都是线性的，传感器测量的误差服从正态分布。那么，式(3-8)中的传感器观测模型可以表达为：

$$z_k = \begin{bmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & 0 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \theta \\ V_x \\ V_y \\ V_\theta \\ A_x \\ A_y \end{bmatrix} + v_k \quad (3-13)$$

本系统的状态转化模型和传感器测量模型都已经被推导出来。在机器人系统运行的过程中，传感器每次测量得到数据后，通过扩展卡尔曼滤波的预测公式(3-9)和更新公式(3-10)的不断迭代，逐渐准确地估计动态系统的状态。

3.2.3 扩展卡尔曼滤波实现效果

本设计使用扩展卡尔曼滤波融合传感器数据，算法数据输入为惯性测量单元的加速度、角速度测量值以及 RF2O 算法对机器人系统的线速度以及角速度的测量值，输出为以上各数据的滤波值。

(1) 扩展卡尔曼滤波对惯性测量单元的滤波效果：

在机器人运行的过程中，惯性测量单元受到多种不同的噪声影响，包括高斯噪声、粉红噪声、机械噪声等等，而这些噪声很难用仿真软件模拟，所以本测试在现实环境中进行。但因为无法测量传感器的理想读数，所以无法进行定量比较。

在机器人系统运动的过程中，分别记录惯性测量单元的读数与扩展卡尔曼滤波的滤波结果，并进行比对。惯性测量单元测量到的 X 轴加速度测量值与扩展卡尔曼滤波的滤波结果如图 3-9 所示。可以看出，惯性测量单元的测量噪声较大，并且当加速度达到正向峰值时，由于力的方向突然改变，机械结构的振动引起加速度传感器读数的巨幅震荡，显然，该数据无法直接使用。而扩展卡尔曼滤波处理后的数据比较平缓，原数据峰值的震荡得到了较好的处理，数据滞后程度较小，处理效果比较好。

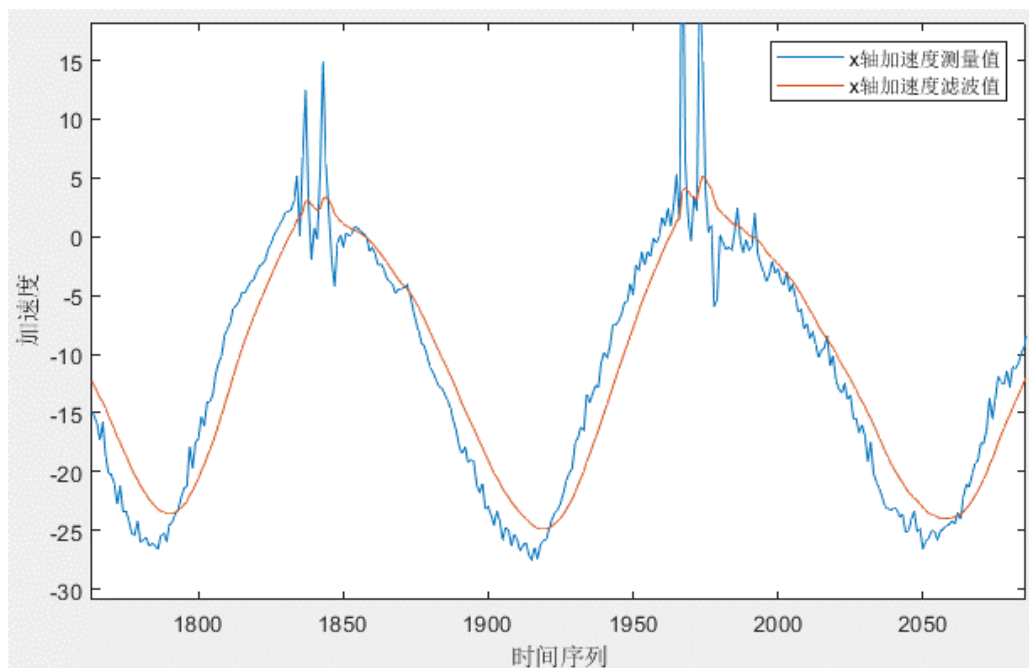


图 3-9 X 轴加速度测量值与扩展卡尔曼处理滤波结果比对图

为了更好地体现滤波算法的作用，分别对着两个数据进行快速傅里叶变换，得到数据的频域数据如图 3-10 所示。可以看出，原数据的有效数据主要集中在 5Hz 以内的频域内，而高频段充满了高频噪声。扩展卡尔曼滤波在基本保留低频段信号的情况下，对高频的噪声进行了有效处理。因此，扩展卡尔曼滤波对惯性测量单元的数据处理较好。

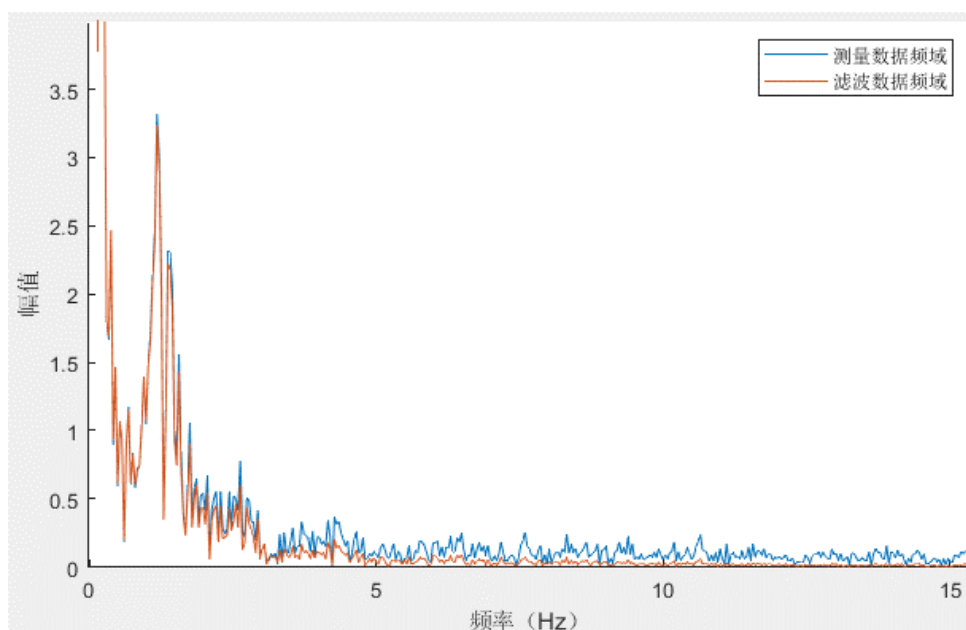


图 3-10 X 轴加速度测量值与扩展卡尔曼滤波处理结果的频域对比图

(2) 扩展卡尔曼滤波对 RF2O 算法计算结果的滤波效果:

为了定量比较滤波结果, 本实验在仿真环境中进行。在 Gazebo 仿真软件中, 建立室内仿真场景并在室内安放多种不同形状的障碍物。仿真场景图和该仿真场景的栅格地图如图 3-11 和图 3-12 所示。为了保证仿真的结果更加符合现实, RF2O 算法和扩展卡尔曼滤波算法的工作频率都设置为 10Hz, 同时, 在惯性测量单元和激光雷达的每一次样本的测量上都增加一个服从正态分布的噪声。

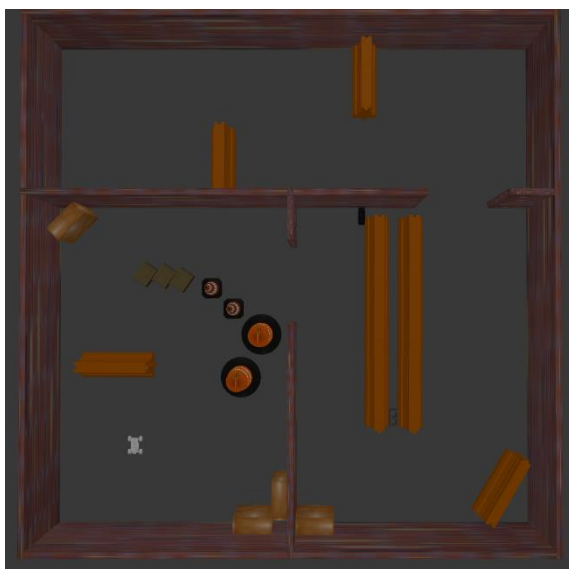


图 3-11 室内仿真场景图

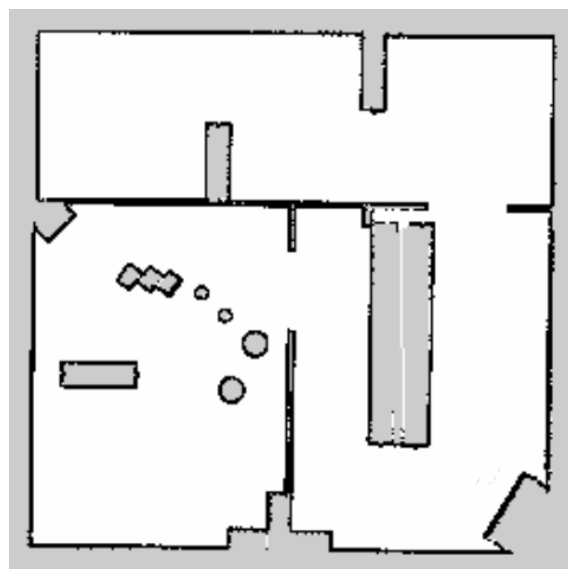


图 3-12 仿真场景的栅格地图

为测试扩展卡尔曼滤波的效果，机器人以场景的左下角作为出发点和世界坐标系的原点，并运动到场景的左上角作为终点。在机器人系统运动的过程中，分别记录机器人姿态的真实值（红色曲线）、RF2O 算法的计算值（蓝色曲线）以及扩展卡尔曼滤波对多传感器数据融合后的姿态数据（绿色曲线）后描绘轨迹如图 3-13 所示，并进行比对。为方便描述，将场景房间分为一号，二号，三号房间。

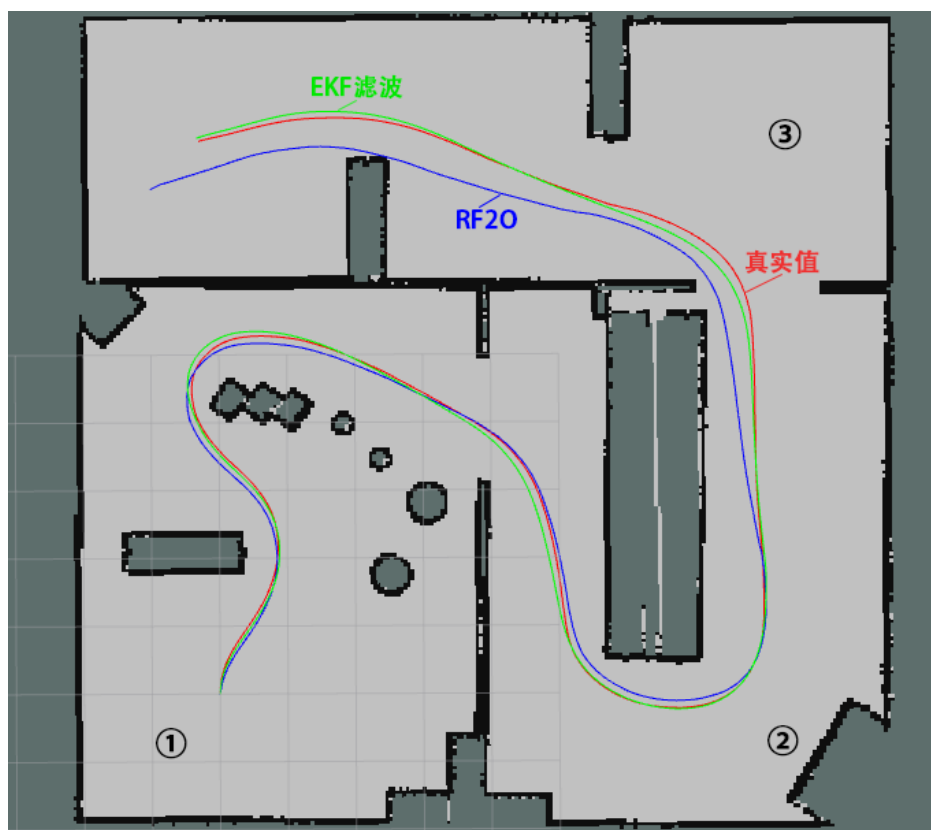


图 3-13 单次仿真中，不同数据处理方法得到的机器人运动轨迹图

从轨迹图中，可以看出：通过 RF2O 算法计算的轨迹曲线（蓝色曲线）在二号房间的后半段以及三号房间由于误差的累计逐渐偏离实际曲线（红色曲线）。在机器人运动到终点后，RF2O 计算得到的 X 轴位移测量误差为 0.84 米，Y 轴位移测量误差为 -0.74 米。因此，单独使用 RF2O 算法无法完成定位任务。而扩展卡尔曼滤波算法通过融合 RF2O 算法的计算结果以及惯性测量单元的测量数据，轨迹曲线（绿色曲线）能够较好地贴合实际曲线。尽管如此，扩展卡尔曼滤波数据在二号房间和三号房间入口的转弯处都与真实数据存在明显偏差。并且，扩展卡尔曼滤波能够减少误差累计对系统的影响，在机器人运动的终点，扩展卡尔曼滤波计算得到的 X 轴位移测量误差仅为 -0.07 米，Y 轴位移测量误差为 0.1 米。

为了定量比较扩展卡尔曼滤波的滤波效果，本文采用均方根误差（RMSE）来描述测量的误

差^[15]。假设 P_1, \dots, P_n 为预测的姿态数据序列； Q_1, \dots, Q_n 为真实的姿态数据序列； Δ 为每秒的数据间隔，即每秒钟预测的姿态数据个数，本次实验 $\Delta=10$ 。则均方根误差的计算公式如下所示：

$$RMSE(E_{1:n}, \Delta) := \left(\frac{1}{n-\Delta} \sum_{i=1}^{n-\Delta} \left\| (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta}) \right\| \right)^{1/2} \quad (3-14)$$

将记录的位移数据通过式(3-14)计算，得到 RF2O 算法以及扩展卡尔曼滤波对多传感器数据融合后的误差数据如表 3-2 所示。在机器人系统运行的过程中，RF2O 算法测量得到的 X 轴位移误差小于扩展卡尔曼滤波算法，但是 RF2O 算法测量的 Y 轴位移以及偏航角的误差都远远大于扩展卡尔曼滤波算法。综上，扩展卡尔曼滤波算法处理后的定位数据误差减小，并且该滤波算法减少了误差累计对系统测量的影响。使用扩展卡尔曼滤波基本能完成室内移动机器人定位任务。

表 3-2 RF2O 算法和扩展卡尔曼滤波算法的测量误差表

| | RF2O 算法 | 扩展卡尔曼滤波 |
|--------------|-------------|-------------|
| 均方根误差（X 轴位移） | 0.315175295 | 0.372654996 |
| 均方根误差（Y 轴位移） | 4.695963809 | 0.490947403 |
| 均方根误差（偏航角） | 20.3348107 | 0.117389029 |

3.3 自适应蒙特卡洛定位算法

3.3.1 自适应蒙特卡洛定位算法概述

自适应蒙特卡洛定位算法（Adaptive Monte Carlo Localization, AMCL）是一种估计机器人在地图中的姿态和位移数据的定位方法。AMCL 算法使用粒子来表示机器人可能的假设姿态，通过粒子集采样、权值运算以及重采样步骤逐渐丢弃与观测不符的粒子。最终，粒子会不断收敛，从而得到机器人在地图中实际的位置数据。在图 3-11 所示的仿真场景中，AMCL 算法粒子群逐渐收敛的过程如图 3-14 所示。



图 3-14 自适应蒙特卡洛定位算法粒子群逐渐收敛的过程图

AMCL 算法会遇到绑架问题，即当机器人丢失了先前的位置信息或者得到了一个错误的位姿信息，机器人就无法依靠 AMCL 算法继续计算当前自己在地图上的位置信息。同时，AMCL 算法是基于粒子滤波思想的算法，计算量较大，对 CPU 处理能力要求较高。为了解决这两个问题，本设计提出扩展卡尔曼滤波和 AMCL 算法相结合的思想，即使用扩展卡尔曼滤波计算出机器人的相对定位后，再使用 AMCL 算法修正该相对定位得到精度更高的绝对定位。扩展卡尔曼滤波可以减少误差累计的影响，从而解决 AMCL 算法的绑架问题。此外，由于 AMCL 算法仅起到辅助修正作用，该算法的工作频率可以远远低于扩展卡尔曼滤波算法，以此来减少定位系统的总体计算量。

3.3.2 自适应蒙特卡洛定位算法实现效果

为了定量比较 AMCL 算法的效果，测试实验与扩展卡尔曼滤波的测试实验步骤相同。在程序上，增加 AMCL 算法，该算法通过修正扩展卡尔曼滤波算法计算得到的相对定位数据计算出车模的绝对定位。扩展卡尔曼滤波算法的工作频率设置为 10Hz，AMCL 算法的工作频率设置为 0.2Hz。在运行的过程中分别记录扩展卡尔曼滤波与 AMCL 算法得到的定位误差。其中，这两个算法在位移上的误差对比如图 3-15 和图 3-16 所示。在绝大部分情况下，AMCL 算法的测量误差都要小于扩展卡尔曼滤波的测量误差。但是由于 AMCL 算法仅仅是修正扩展卡尔曼滤波算法的

测量值，所以 AMCL 算法测量误差的变化趋势直接受到扩展卡尔曼滤波算法测量误差的影响。

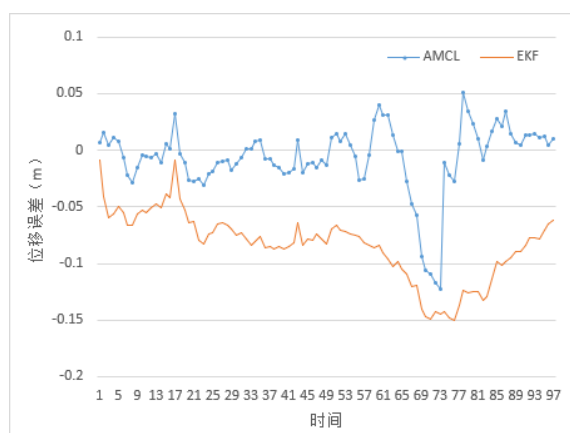


图 3-15 X 轴位移测量误差对比图

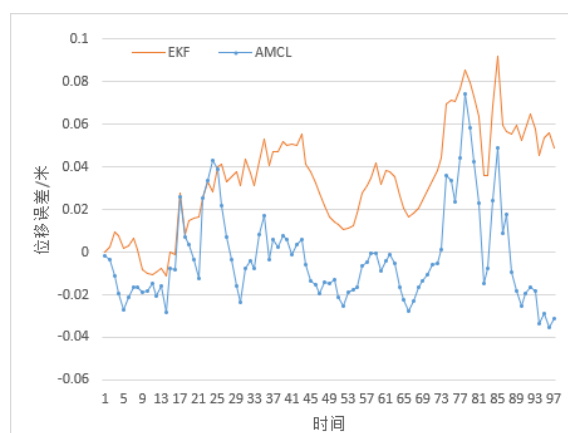


图 3-16 Y 轴位移测量误差对比图

将采集到的数据使用公式(3-14)计算 AMCL 算法测量的均方根误差，得到误差数据如表 3-3 所示。将该表格数据与表 3-2 数据进行对比可以得出：使用 AMCL 算法对扩展卡尔曼滤波算法的测量结果进行修正，可以有效降低 X 轴和 Y 轴的位移测量误差，最大测量偏差不超过 7.4 厘米。但是由于机器人系统使用的角速度传感器测量噪声较小，使得扩展卡尔曼滤波算法融合角速度传感器数据计算的偏航角误差较小，引入 AMCL 算法反而增大了偏航角的测量误差，但是该误差仍然较小，最大测量偏差不超过 1.6° ，在可以接受的范围之内。最重要的是，使用 AMCL 算法对扩展卡尔曼滤波数据进行修正解决了积分引起的误差累计问题和 AMCL 算法的机器人绑架问题。同时，因为 AMCL 算法的工作频率可以远远低于扩展卡尔曼滤波算法，该传感器数据处理方案的计算量并不大。综上，室内移动机器人系统的定位问题得到较好的解决。

表 3-3 自适应蒙特卡洛定位算法的测量误差表

| | 自适应蒙特卡洛定位算法 |
|----------------|---------------|
| 均方根误差 (X 轴位移) | 0.140620729 |
| 均方根误差 (Y 轴位移) | 0.29658343 |
| 均方根误差 (偏航角) | 0.307936455 |
| 最大测量偏差 (X 轴位移) | 0.057 米 |
| 最大测量偏差 (Y 轴位移) | 0.074 米 |
| 最大测量偏差 (偏航角) | -1.59° |

第 4 章 基于 ROS 的机器人定位系统实现

机器人操作系统（Robot Operating System, ROS）是机器人项目开发的开源软件平台。ROS 本身并不是实时操作系统，但它可以提供许多类似操作系统的服务，例如进程之间的通讯，常用传感器的硬件驱动以及程序包管理等功能。

ROS 的设计理念是减少机器人系统开发过程中的代码重复使用、重复开发的现象。在此理念之上，ROS 的功能模块都是以功能包（Packages）的形式存在，开发者通过下载需要的开源功能包可以有效减少代码复用率，降低开发成本。

ROS 同时还集成了很多工具，这些工具可以使得开发工作变得更加简单。其中，Gazebo 是基于强大物理引擎的仿真软件，可以精确的模拟出复杂的机器人工作环境并测试机器人系统算法的可靠性；Rviz 是数据可视化工具，它通过订阅机器人系统的实时数据，使得开发者能够及时观察机器人系统的运行状态；rqt 上集成了多种开发工具，包括数据示波器 rqt_plot，负责消息记录的 console 等等；

在机器人的定位设计上，ROS 提供了很多已经完成的软件包，如 tf 软件包提供了机器人系统的坐标变换、slam_gmapping 软件包提供了基于激光雷达的开源 SLAM 算法实现、navigation 软件包提供了定位、导航等功能。合理使用这些软件包可以大大减少开发难度。

4.1 ROS 节点整体设计

分布式计算是 ROS 操作系统的一个最显著的特征。在 ROS 中，节点是表示完成一定功能的进程。在此基础上，ROS 提供了话题（Topics）和服务（Services）使得节点与节点之间能够互相通讯。本设计的 ROS 节点通讯简化图如图 4-1 所示，其中，使用椭圆形代表运行的节点，使用矩形代表话题。

机器人系统的运动控制由 /telep_control 和 /art_driver_node 两个节点控制完成。其中 /telep_control 为远程控制节点，上位机通过该节点发布机器人的线速度与角速度控制话题 /cmd_vel。/art_driver_node 接收该数据后通过自动控制算法计算电机、舵机的控制信号，并把该信号发送给单片机完成运动控制。

/art_ls01g 和 /art_imu_02a 分别为激光雷达和惯性测量单元的驱动接口。它们负责初始化、设置传感器以及读取传感器测量数据并发布相应的传感器数据话题。/rf2o_node、/ekf_node 和

/amcl_node 则分别是第 3 章中提到的 RF2O 算法、扩展卡尔曼滤波算法以及自适应蒙特卡洛定位算法的软件实现。它们通过订阅传感器数据的话题，通过程序计算，得到机器人系统的定位数据。

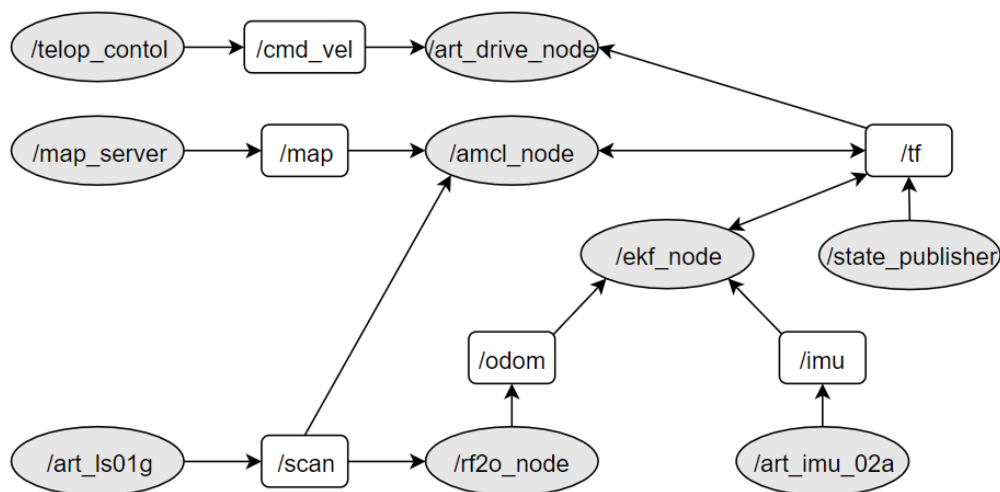


图 4-1 ROS 节点与话题通讯关系简化图

/tf 话题则是用来保存机器人系统中的各个坐标结构。/ekf_node 和/amcl_node 节点通过该话题更新机器人系统的相对定位以及绝对定位数据。下面具体介绍本设计的坐标系统设置。

4.2 坐标节点设计

在复杂的机器人系统中，会涉及到的复杂的坐标系转化关系。ROS 提供了 TF 功能来高效管理各个坐标系。在本设计中，室内移动机器人简化模型以及各个坐标系定义如图 4-2 所示。

base_link 为机器人坐标系，即 2.3 节提到的坐标系 Σ_R ，该坐标系一般以机器人系统的重心为原点。

laser_link 和 imu_link 分别为激光雷达坐标系和惯性测量单元坐标系，它们也都是以传感器的重心为原点，并且由于机器人的机械结构固定，它们与 base_link 坐标系之间的转换关系是固定不变的。激光雷达和惯性测量单元测量到的传感器数据需要通过坐标轴变换转换到 base_link 上才可以计算使用。

odom 坐标系为里程计坐标系，base_link 坐标系相对于 odom 坐标系的姿态即为机器人的相对定位。该坐标通常由线速度以及角速度传感器测量，通过积分计算得到，会存在误差累计的问题，故不能成为长时间的参考。但是 odom 坐标系变化连续，在短时间内准度可以保证，故该坐标系下的速度、角速度数据可以作为运动控制的反馈数据。

map 坐标系为世界坐标系，即 2.3 节提到的坐标系 Σ_w ，base_link 坐标系相对于 odom 坐标系的姿态即为机器人的绝对定位。该坐标通常是 odom 坐标系通过定位算法计算得到。map 坐标系会随时间变化发生跳变，但是不存在误差累计问题，故适合作为长期参考。解决室内移动机器人定位问题本质上就是得到 map 坐标系与 base_link 坐标系之间的关系。

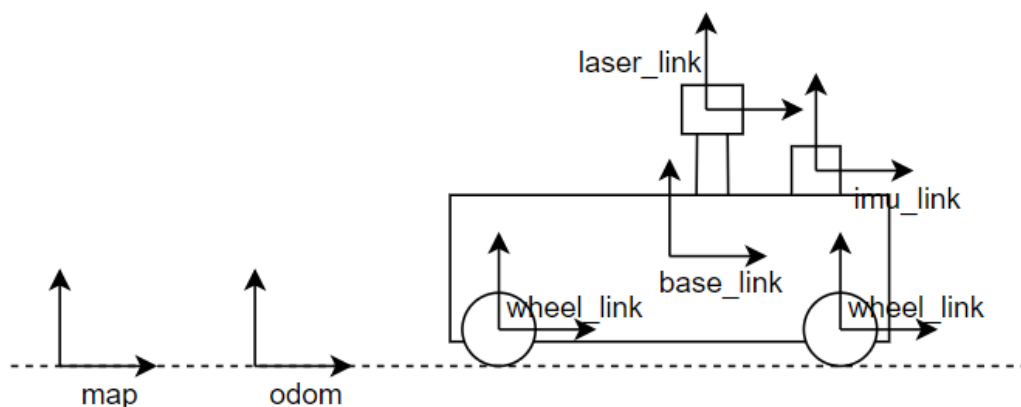


图 4-2 机器人坐标系框架图

在 TF 功能包中，各个坐标系都按照树形结构连接。本设计的各个坐标系之间的关系如图 4-3 所示。其中，各个坐标系的转换关系如下：由于机械结构的关系，base_link、imu_link 以及 laser_link 坐标系之间的转换关系固定不变，传感器测量的数据都会转换到 base_link 坐标系下，以方便统一计算；odom 坐标系由惯性测量单元和激光雷达数据通过卡尔曼滤波算法计算得到；map 坐标系在 odom 坐标系的基础上通过自适应蒙特卡洛定位算法计算得到。得益于该树形连接关系，任意两个坐标系的转换关系可以根据其在树形图上的连接关系迭代计算得出。

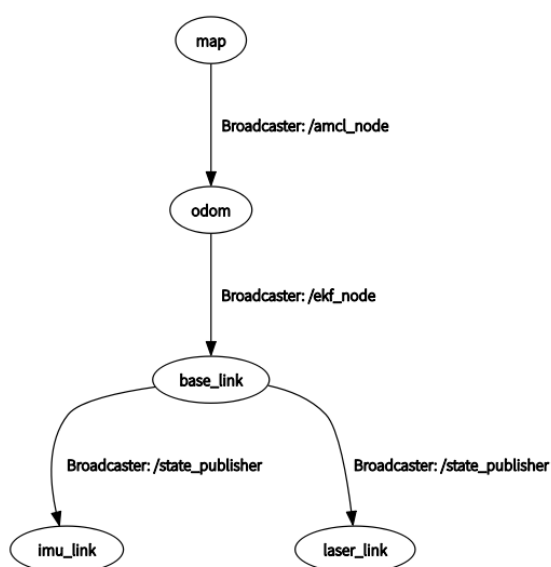


图 4-3 各个坐标系的树形关系图

4.3 运动控制节点设计

4.3.1 PID 控制基本原理

在控制系统中,控制器最常用的是比例积分微分控制(proportional-integral-derivative control),简称 PID 控制。PID 控制算法简单,稳定性强,被广泛使用,表达式如下所示:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (4-1)$$

式中, K_p 为控制器的比例系数; T_i 为控制器的积分时间常数; T_d 为控制器的微分时间常数;

$u(t)$ 为 t 时刻控制器的输出; $e(t)$ 为 t 时刻控制系统的期望输入与测量的实际输入的差值。从表达式可以看出, PID 控制分为比例、积分和微分部分。其中比例部分主要对系统的瞬时误差做出响应,比例系数越大,控制作用越强;积分部分主要对系统的累计误差做出响应,恰当的积分常数可以有效消除系统的稳态误差;微分部分主要对系统误差的变化率做出响应,该部分使得控制器能够对误差变化的趋势做出预测,提高控制系统的响应速度。

由于计算机控制是一种采样控制,计算机控制在时间上是离散的。因此,式(4-1)需要离散化处理后才能在计算机系统中使用。以 T 为采样周期, k 为采样序号的离散 PID 控制的表达式如下所示:

$$u_k = K_p \left[e_k + \frac{T}{T_i} \sum_{j=0}^k e_j + T_d \frac{e_k - e_{k-1}}{T} \right] \quad (4-2)$$

4.3.2 PID 控制实现

设计机器人系统的速度闭环控制框图如图 4-4 所示,该闭环控制的输入为期望线速度,输出为机器人实际线速度,控制器采用 PI 控制,执行器为电机驱动。为得到更准确的线速度反馈,线速度测量值由 RF2O 算法测量的线速度和惯性测量单元测量的加速度通过扩展卡尔曼滤波融合计算得到。采用 PI 控制是因为 P 系数增加控制的反应速度, I 系数保证实际机器人线速度能始终跟随设置的期望线速度。

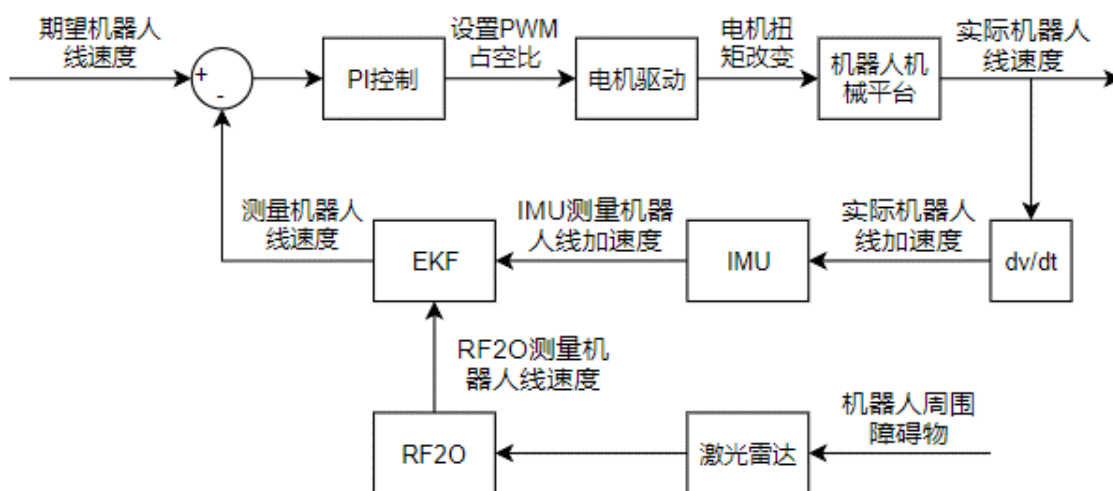


图 4-4 机器人系统的线速度闭环控制框图

机器人系统的角速度闭环控制框图如图 4-5 所示，该闭环控制的输入为期望角速度，控制器采用 PD 控制。通过控制器输出 PWM 控制舵机打角来控制车模达到期望角速度。采用 PI 控制是因为 P 系数增加控制的反应速度，D 系数保证该系统的稳定。

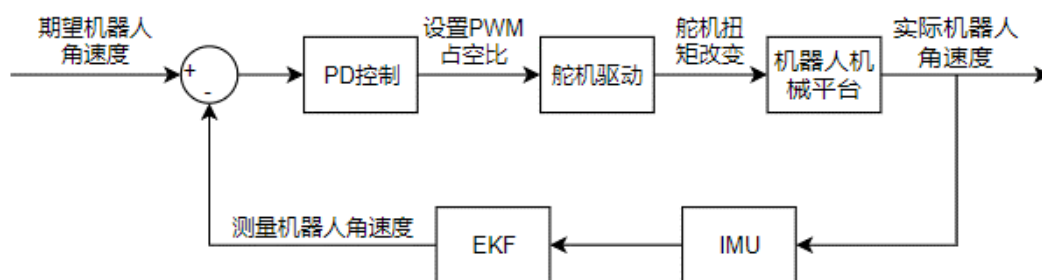


图 4-5 机器人系统的角速度闭环控制框图

图 4-1 中的/art_driver_node 节点即为运动控制节点，该节点通过/tf 节点得到/ekf_node 节点发布的滤波后机器人线速度和角速度，随后通过图 4-4 和图 4-5 所示的控制框图计算控制量。由于舵机驱动和电机驱动是由 KEA 单片机通过 PWM 信号驱动，车载电脑需要将计算得到的控制量通过 USB 转串口接口发送给单片机，从而完成控制。

4.4 定位实验与结果分析

为了验证在仿真实验中得到的结果，本节使用真正的机器人在现实环境中完成定位任务。但由于无法测量出定位数据的真实值，也就无法对测量结果进行定量比较。因此，在本节通过定位数据以及激光雷达测量的障碍物数据绘制地图来定性比较不同定位方案的性能。同时，由于机器人每次运行的运动轨迹、以及传感器数据都不相同，为避免这些差异对实验结果的影响，本节

使用 `roslaunch` 工具记录机器人某次运动时的传感器数据以及数据发布的时间。在运动完成后，依据时间顺序对数据进行回放，再根据各个定位方案，计算出定位数据并完成建图。

在本次实验中，机器人途经的路径大约有 60 米长，平均速度为 0.5 米/秒，并且除自适应蒙特卡洛定位算法的工作频率为 0.2Hz 以外，其他各算法的工作频率都设置为 10Hz。不同定位方案得到的机器人运动轨迹以及建立的地图如图 4-6 所示。在图中，左侧地图为参考建图。图中的定位数据是由目前普遍使用的基于粒子滤波的 SLAM 算法生产，最符合真实情况；右侧地图的定位数据源自于 RF2O 算法，可见 RF2O 算法计算得到的角速度误差较大（见 3.2.3 条），导致转弯处的地图出现弯曲、形变。中间地图中的定位数据则是来自于本论文提出的传感器数据处理方案（见第 3 章），传感器数据通过数据融合并修正后得到的定位数据更加准确，得到的定位数据所建立的地图非常接近参考地图，证明数据误差较小。

在计算量方面，虽然参考 SLAM 算法和本设计所使用的自适应蒙特卡洛定位算法同样都是基于粒子滤波思想的算法，但是自适应蒙特卡洛定位算法仅起到辅助修正作用，其工作频率远远低于参考 SLAM 算法。所以，本论文提出的定位系统计算量要小于参考 SLAM 算法。

综上，仿真实验和现实实验证明：本论文提出的传感器数据处理方案具有测量误差较小、计算量较小的优点，较好地解决了低成本机器人的室内定位问题。



图 4-6 不同方案估计轨迹以及地图对比图

第 5 章 非技术因素的考虑与方案评估

5.1 本课题对非技术因素的考虑

由于机器人系统的重量较大，为了给与机器人机械平台足够的牵引力，机器人装配有大功率的电机，该电机能够输出较大扭矩并且使机器人达到较高的运行速度。因此，机器人在环境中如果无法及时避障可能会损伤周围人员、环境以及机器人自身，造成安全问题。

其次，由于机器人系统装配有电机、舵机等大功率电子器件。在系统运行时，电路板上的电流远远大于人体安全电流 10mA。因此，当电路发生故障、短路、漏电等情况时，过大的电流不但会直接损坏电路元件，造成经济损失，而且会对调试人员的身体健康造成影响。

5.2 非技术层面的方案评估

为了解决机器人系统动能过大而可能造成的安全问题，本设计在机器的前端安装弹簧防撞结构。当机器人失控并碰冲撞到其他物体时，利用弹簧的弹性势能缓冲掉大多数动能，以此来保护机器人系统。同时，在机器人的速度闭环控制算法中，将系统的预期速度设定在比较低的数据，并且限制控制器的最大输出来减少机器人系统的速度和加速度，使得系统更加安全。

同时，为了解决电路电流的不安全问题，本设计在电池电路中串联一根保险丝，当电流过大时，保险丝自动熔断来保护电流和调试人员。此外，在电路中增加电池检测电路，在单片机中编程实现实时检测电路电流。当电流过大时，及时报警提醒调试人员处理。

5.3 成本估算与经济决策

本设计所牵扯到的成本预算主要包括材料成本以及设计人工成本。

材料成本包含机器人机械结构、电子元件、电机、舵机、车载电脑、传感器模块等器件的选择和购买。其中，为了减少制作成本，不同功能模块在成本考虑上的优先级各不相同。其中，为了保证定位系统的精度以及工作频率等性能，传感器模块为最高优先级，即购买传感器不考虑成本，尽可能购买性能好的传感器来减少数据处理的麻烦。车载电脑的性能会影响到传感器处理算法的运行速度，因此，车载电脑选择性能刚好达到要求即可。至于机器人的其他功能模块则是尽量考虑节省成本，比如单片机选用低成本、低功耗的 KEA128 单片机。

设计人工成本主要为时间成本，其中包含机器人机械结构的设计预搭建、电路图的设计与焊接、算法在仿真软件上的验证和软件设计。系统的机械和硬件设计应该投入足够时间以确保系统的稳定性。在仿真软件上测试算法同样也是必要的，该步骤可以在模拟的环境中验证算法的可行性，节省后续软件开发的时间。

5.4 本课题的创新性

相比较于一般的室内移动机器人，本系统与其他系统在方案上最大的区别在于传感器和算法的选择。

在传感器选择方面：目前，在室内移动机器人定位方案中，主要使用激光雷达传感器以及里程计传感器作为主要传感器。其中，激光雷达传感器在平面移动机器人系统中不可或缺，在机器人的环境感知、避障等算法中起到关键作用；里程计传感器以摄像头传感器以及车轮编码为主。而本设计使用激光雷达传感器和惯性测量单元。首先，在成本上，惯性测量单元相较于一般的摄像头传感器以及精密的车轮式里程计成本更低。其次，本设计为那些无法安装车轮式里程计的机器人在定位方案上提供了一种新的技术路线。

在算法选择方面：目前，大部分学者主要使用基于粒子滤波的数据处理方法来融合传感器数据得到机器人系统的定位数据。而粒子滤波算法计算量大，对 CPU 性能要求高。本设计使用扩展卡尔曼滤波算法融合传感器数据得到机器人的定位数据，再使用自适应蒙特卡洛定位算法修正结果以提高测量的精度。由于基于粒子滤波思想的自适应蒙特卡洛定位算法工作频率低，定位系统的计算量较少，并且计算得到的定位数据误差可以接受，因此，本论文提出的传感器数据处理方案较好地完成了定位任务。

结 论

本论文针对只使用激光雷达和惯性测量单元作为传感器的低成本机器人系统进行研究设计，并定制传感器数据处理方案，完成室内定位任务。

本论文首先从机器人系统建模出发，对室内移动机器人的系统设计框架、坐标系统以及运动模型做出深入研究，为定位算法的推导奠定了基础。针对传感器的数据处理问题，本文阐述了所使用的 RF2O 算法、扩展卡尔曼滤波算法以及自适应蒙特卡洛定位算法的基本原理。根据机器人模型和传感器数据特点决定采用以下方案进行传感器数据处理：使用 RF2O 算法从激光雷达的数据变化中计算得到机器的运动数据，并使用扩展卡尔曼滤波融合惯性测量单元数据和 RF2O 算法计算结果得到机器人系统的相对定位数据，最后在此基础上使用自适应蒙特卡洛定位算法对定位数据进行修正，以得到精度更高的绝对定位数据。

为验证算法的性能，本设计首先在 Gazebo 仿真软件中进行仿真实验，定量计算了传感器数据通过不同算法处理后得到的误差。然后设计了室内四轮移动机器人实体，并在实际环境中进行了定位实验。实验数据表明：1. 本论文提出的数据处理方案解决了自适应蒙特卡洛定位算法的绑架问题和积分导致的误差累计问题；2. 算法得到的定位数据精度较高，在仿真实验中位移以及偏航角误差控制在 7.4cm 以及 1.6° 之间；3. 由于自适应蒙特卡洛定位算法仅起到辅助修正作用，因此该算法的工作频率可以远远低于扩展卡尔曼滤波算法的工作频率，因此，算法总体计算量较小。并且不依赖于里程计传感器，可以满足低成本室内移动机器人的定位需求。

参考文献

- [1] 吴晶. 无人驾驶汽车三维同步定位与建图精度评估[J]. 装备机械,2019(04):17-20+43.
- [2] Zhi L, Xuesong M. Navigation and control system of mobile robot based on ROS[C]//2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2018: 368-372.
- [3] Ibragimov I Z, Afanasyev I M. Comparison of ros-based visual slam methods in homogeneous indoor environment[C]//2017 14th Workshop on Positioning, Navigation and Communications (WPNC). IEEE, 2017: 1-6.
- [4] 龚学锐,闵华松. 基于 ROS 的惯性和视觉里程计的机器人室内定位[J]. 组合机床与自动化加工技术,2019(05):106-110.
- [5] 严小意,郭杭. 激光 SLAM 移动机器人室内定位研究[J]. 测绘通报,2019(12):8-11.
- [6] Jaimez M, Monroy J G, Gonzalez-Jimenez J. Planar odometry from a radial laser scanner. A range flow-based approach[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 4479-4485.
- [7] Censi A. An ICP variant using a point-to-line metric[C]//2008 IEEE International Conference on Robotics and Automation. Ieee, 2008: 19-25.
- [8] Diosi A, Kleeman L. Fast laser scan matching using polar coordinates[J]. The International Journal of Robotics Research, 2007, 26(10): 1125-1153.
- [9] Kalman R E. A new approach to linear filtering and prediction problems[J]. 1960.
- [10] 李辉. 基于激光雷达的 2D-SLAM 的研究[D].浙江工业大学,2017.
- [11] 王古超. 基于 ROS 的全向移动机器人系统设计与研究[D].安徽理工大学,2019.
- [12] He H, Jia Y H, Sun L. Simultaneous Location and Map Construction Based on RBPF-SLAM Algorithm[C]//2018 Chinese Control And Decision Conference (CCDC). IEEE, 2018: 4907-4910.
- [13] 柴瑞强,王瑞. 基于 MK60FN1MOVLQ15 单片机的智能车设计[J]. 农业装备与车辆工程,2019,57(11):68-71.
- [14] 陶文鹏. 基于 PID 算法对智能车自动控制的研究与实现[J]. 河南科技,2019(19):30-32.
- [15] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE,

2012: 573-580.

[16] 雷碧波. 基于 ROS 平台的室内定位算法设计与实现[D].浙江理工大学,2017.

[17] 熊安. 基于 SLAM 的轮式机器人定位与导航技术研究[D].中国科学院大学(中国科学院国家空间科学中心),2018.

[18] Thrun S, Fox D, Burgard W, et al. Robust Monte Carlo localization for mobile robots[J]. Artificial intelligence, 2001, 128(1-2): 99-141.

[19] 张敏, 李凯, 韩焱, et al. 基于卡尔曼滤波的陀螺仪降噪处理[J]. 传感技术学报, 2018.

[20] 吴敬阳. 基于激光雷达定位导航的多自主移动机器人系统研制[D].哈尔滨工业大学,2017.

[21] 贾云辉.基于 ROS 系统的移动机器人室内定位方法研究[D].天津:天津理工大学,2019.

致 谢

时光荏苒，大学四年转瞬即逝，感谢浙江科技学院赠与我这四年岁月光阴。在此论文完成之际，我希望向那些帮助过我的老师、同学表达我最真诚的谢意。

首先感谢我的指导老师孙勇智老师。四年来，孙老师在学业方面给予我许多帮助，为我指明未来努力的方向，在整个毕业设计期间在本论文的选题、课题研究、论文的修改等各个方面给予我悉心指导和帮助。孙老师认真严谨的科研作风与丰富的知识一直影响着我，使我受益匪浅。

同时，我也要感谢孙老师领导下的浙江科技学院智能车队对本论文的支持，移动机器人的搭建得益于车队优良的开发条件。最后，我也要感谢我的同学，四年以来我们的互相学习、互相进步、互相帮助，度过了一段难忘的大学时光。