
物品检测与抓取机器人

开发计划

SDP102

V1.0.0

分工说明

小组名称	rushbot	
学号	姓名	本文档中主要承担的工作内容
17373060	杨开元	负责项目任务范围的编写以及本文档的编写与定稿
17373517	王正达	负责项目任务概要的编写以及本文档的审核与校对
15241035	彭文鼎	负责风险管理、过程模型和资源计划模块的编写
17373288	尹俊成	负责项目时间节点确定以及里程碑计划的编写

版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
V1.0.0	2020.3.9	杨开元	王正达	本小组软件开发计划初稿

1. 范围

1.1 项目概述

1.1.1 项目背景

随着计算机技术的发展,网络的普及,硬件和制造业的进步,我们在越来越多的领域都能看到机器人的身影。人类希望将那些枯燥重复、有流程约定的任务交到价格低廉、可以全天候工作的机器手中去,进而提高效率,减少成本。

机器人最早应用于工业领域,二战后不久,工业机器人出现,用来替代人类完成一系列高危动作,如搬运危险材料、开采矿物等。[1]之后大学实验室和社会中的公司也投身于这个领域,开发了各种各样的机器人,应用到航空航天、交通、服务等行业中去。

而机器人技术的进步,离不开以嵌入式计算机为核心的嵌入式系统。人类对机器人的要求越来越高,想要让其能够实现越来越多的功能,开发了种种特定的机器人,但是对于这些机器人来说,底层的模块,如雷达模块、感知模块、导航模块又非常相似,于是 ROS 系统应运而生[2]。

ROS 系统是一个适用于机器人开发的开源操作系统,提供了操作系统应有的服务,如抽象、消息传递、包管理等等。ROS 还是一个分布式的节点框架,进程被封装在易于分享和发布的程序包和功能包中,工程也可以被 ROS 的基础工具方便地整合在一起。

与此同时,越来越多的新技术也正在 ROS 系统中发挥作用,如开源的 OpenCV、PCL 图像库、语音识别,结合这些技术机器人也有了更广泛的作用领域,更好的人机交互体验等等。

1.1.2 主要功能

首先,我们要保证机器人的移动功能,在机器人内置的三轴姿态测量系统的辅助下,通过三个全向轮和底层硬件驱动,实现可控的移动、转向、起停。

其次,我们会使用机器人底盘上安装的 360°激光雷达,结合 HectorSLAM 和 GMapping 算法实现 SLAM 建图[3],同时实现静态和动态的避障功能,使得机器人能够智能地停止或后退。

之后,我们会使用类似 A* 的路径规划算法实现机器人的路径规划,实现机器人到目标点的较优路径规划。

我们的最终目标,是在上述三个目标的基础上,实现目标物品的检测和抓取,我们会使用机器人上的 ToF 模组、摄像头采集二维和三维数据,结合相关算法和对机械臂的操控,实现该功能。

为改善人机交互体验,我们也会尝试着加入语音识别与控制系统。为验证机器人的有效性,我们会设置实验场景对机器人目标检测结果、接近目标和抓取进行可靠性和效率性测试,这些是项目中的非功能需求。

1.1.3 应用场景

该机器人有很多值得展望的应用场景。

该机器人可以在无人职守的商店里 24 小时值班并销售物品,面对提前规格化的货架和机器人自己建立的地图,只需要让其学习每种商品的特征,加上语音识别模块,就能很好地和人类交互,让这种机器人在夜间或全天服务再好不过。

通过优化分类算法和大量数据的学习,改进机械臂的构造,该机器人可以实现物品的分捡,可以被安排在物流公司这种急需劳动力、垃圾分类点这种环境较差的场景中。

该机器人可以在家中充当保姆的角色,将其安置在行动不便老人的家中,通过老人和机器人的互动,机器人可以学习用户身边的物品,并在用户发出指令之后将物品取来或归位。

1.2 文档概述

本文档用于在开发前明确项目背景、功能，项目使用的相关开发模型，明确时间节点和分工，列出项目开始之前我们想到的技术难点，对整个项目起指导作用。

本文档首先描述 ROS 系统和机器人的大体发展情况，描述机器人当前服务的领域，进一步引导出本小组项目的大体构思，我们在项目描述一节简要写明了项目要实现的功能和非功能性需求，点明我们的机器人项目的目标及潜在应用领域，论述了我们机器人项目的先进性和必要性。

文档接着在项目任务概要一节详细展开我们计划的功能，描述每个功能依赖的硬件软件，需要的开发环境，技术选型及其实现的大体思路。在这一节中，我们同样写明了队伍成员的资质，项目结构框架等等。展开描述了项目中每个文档的大致内容和意义，明确了初步的项目期限。

在风险管理一节中，我们描述了项目开发、测试和部署过程中可能遇到的风险，项目时间节点估算风险、技术困难、人员分配不合理、需求需要调整等等多种可能性，并给出了一些应对风险的策略，使得我们对风险有一定抵抗能力。

在过程模型一节中，我们考虑到模块之间的依赖关系，提出使用迭代式模型来进行机器人的开发，以求稳定和效率。

在资源计划中，我们列出了我们需要的硬软件资源。在进度计划中，我们给出了初步计划约定的时间节点。

1.3 术语和缩略词

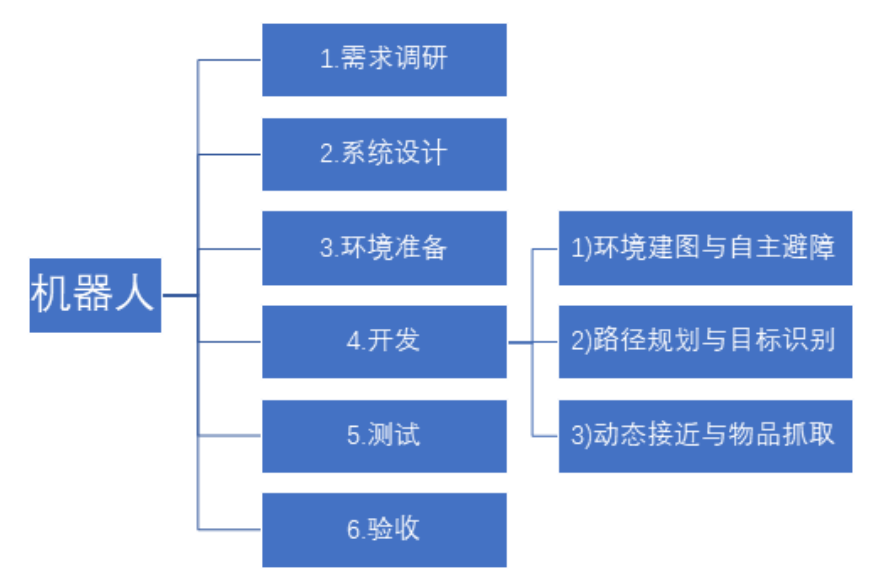
术语和缩略词	简要解释
ROS	Robot Operating System 编写机器人软件的高度灵活性框架
OpenCV	Open Source Computer Vision Library 轻量高效的开源计算机视觉库
PCL	Point Cloud Library 实现 3D 视图下处理计算的点云库
SLAM	Simultaneous Localization and Mapping 机器人的自主定位和导航
HectorSLAM	一种上述SLAM算法
GMapping	结合激光技术的一种实现 SLAM 的算法
A*算法	A-star 求解最优路径时的启发式算法
ToF	Time of Flight 通过给目标连续发送光脉冲的 3D 成像方法

1.4 引用文档

- [1] 李肇惠,郝昭.工业机器人的技术发展及其应用分析[J].内燃机与配件,2020(01):249-250.
- [2] 王鹏飞. 基于 ROS 的多传感器信息融合自主导航控制系统设计与实现[D].南京邮电大学,2019.
- [3] 北京六部工坊科技有限公司的. 启智 ROS 机器人开发手册 V1.1.0

2. 项目任务概要

2.1 工作内容



2.2 主要人员

开发人员	承担工作(侧重)	工作时间	工作经验	技术水平
王正达	项目经理 文档评审	85 小时左右	一定规模系统 开发经验	熟练
杨开元	程序开发与架构	85 小时左右	较强算法能力	熟练
彭文鼎	程序开发与架构	85 小时左右	较强底层实现 能力	熟练
尹俊成	系统测试 文档整理	85 小时左右	较丰富开发测 试经验	熟练

2.3 产品

2.3.1 程序、数据或设备

源代码:

- 机器人可控移动源代码
- 机器人实时建图源代码
- 机器人路径规划源代码
- 机器人目标检测与抓取源代码

程序:

- 机器人启动程序
- 机器人自主寻路程序
- 机器人目标检测与抓取程序

数据:

- 机器人工作环境参数
- 机器人工作性能参数

2.3.2 文档

1. SDP-软件开发计划

明确项目背景与开发目标, 设定项目时间节点与人员分工, 安排项目开发进度, 完成项目风险预测。

2. SRD-软件需求规格说明书

进一步确定所要开发机器人的用户需求与目标功能, 并对应地设计所要开发软件之界面、功能, 对将来实际开发起到参考与标尺作用。

3. SDD-软件设计说明

从高到低描述软件各个层次的设计工作, 包括但不限于数据库设计、体系结构设计、接口设计、详细设计。

4. STD-软件测试说明

按逻辑顺序精简描述软件测试过程, 依次包括测试准备、测试用例、测试结果、测试结果分析。

2.4 运行与开发环境

2.4.1 运行环境

硬件环境:

- 实验室提供嵌入式开发板
- 激光雷达
- RGB-D 摄像头

软件环境:

- Ubuntu16.04
- RoboWare Studio

2.4.2 开发环境

硬件环境:

- 个人笔记本

软件环境:

- Ubuntu16.04
- ROS Kinetic

2.5 项目期限

起始时间: 2020.3.4

结束时间: 2020.6.10

3. 风险管理

本项目可能有但不限于以下风险:

3.1 估算 (Estimation) 方面

总体开发时间较为充足, 延期风险低, 但各任务持续时间还需明确, 人员分配工作量较为平均合理。风险发生可能性低。应对策略: 后期需要根据具体任务调整策略, 详细记录文档, 调整系统结构。

3.2 技术 (Technology) 方面

开发平台、可选构件集相关。这里由于前期是远程无法接触到实际机器人硬件, 可能存在一些风险。风险发生可能性高。应对策略: 熟悉 ROS 平台, 做好充足的前期准备, 设计多种预期方案, 给后期实际环境测试预留足够的缓冲。

3.3 人员 (People) 方面

具体分工可能还会受到项目开展实际情况的影响, 团队人员技术水平预期可以完成项目。前期是远程沟通且返校时间依然未知。风险发生可能性中等。应对策略: 远程定期沟通, 细化分解任务, 完善文档, 即时交付。

3.4 需求 (Requirements) 方面

目前的需求还是不够明确和具体, 后续会继续迭代调整。风险发生可能性中等。应对策略: 采用敏捷开发模型, 保证细粒度日志记录, 设计时保证模块之间低耦合, 以应对后续需求调整。根据实际完成进度对照里程碑迭代增量式完善。

4. 过程模型

综合考虑本项目的具体任务,由于部分任务之间存在依赖性,选择迭代-增量式过程模型,可以提高稳定性和效率。

主要阶段为:

1. 机器人的基本移动
2. SLAM 实时建图+简单避障
3. 路径规划
4. 物品识别与检测
5. 物品抓取
6. 语音识别

其中阶段 2, 阶段 3 依赖于阶段 1。阶段 5 依赖于阶段 4。阶段 1, 阶段 5, 阶段 6 则互不依赖,可以同时并行,分工完成。通过迭代式模型,每阶段测试驱动,以高效率逐步完善系统的开发。最终完成一个完整的物品检测与抓取机器人。

5. 资源计划

硬件资源需求:

配有 Kinect2 视觉传感器、激光雷达、语音输入输出阵列、机械臂等硬件的启智 ROS 机器人

软件资源需求:

ROS 版本(ROS Kinetic)和 Ubuntu 版本(16.04), ROS 系统还需要相关相关硬件驱动如激光雷达、语音、导航、IAI-Kinect2 等

6. 进度计划

6.1 里程碑计划

序号	应完成的工作内容	时间节点	产品状态条件
1	完成并提交项目开发计划	2020.03.09	初步明确开发目标
2	配置环境	2020.03.15	开发环境配置完毕
3	调研并学习 ROS 有关操作	2020.03.30	个人对分工部分技能掌握熟练
4	完成需求分析文档	2020.03.30	充分了解产品需求
5	需求分析评审	2020.04.01	完成评审
6	进一步写成软件需求分析文档	2020.04.20	产品需求与设计进一步细化
7	软件需求分析/设计文档评审	2020.04.22	完成评审
8	各部分开发完成	2020.05.01	各部分开发完成
9	代码对接	2020.05.05	构建系统
10	系统调试	2020.05.12	系统稳定性与正确性均合格
11	完成主体代码并准备评审内容	2020.05.13	代码评审
12	单元测试, 编写测试文档	2020.06.02	系统正确性进一步保证
13	完成测试文档编写并评审	2020.06.03	测试文档一次评审
14	继续进行测试, 改进文档不足	2020.06.09	系统进一步完善, 文档更加完备
15	测试文档评审内容准备完毕	2020.06.10	测试文档二次评审

6.2 里程碑任务映射

序号	需完成任务	人员	起始时间	截止时间
1	完成开发计划	全体	2020.03.03	2020.03.09
2	配置开发环境	全体	2020.03.09	2020.03.15
3	调研机器人使用需求(7 天) 调研机器人运动与导航(7 天) 调研抓取目标的相关参数(7 天)	全体	2020.03.09	2020.03.30
4	阅读机器人开发手册相关章节	全体	2020.03.09	2020.03.31
5	准备评审资料	王正达 尹俊成	2020.03.09	2020.04.01
6	进一步完善软件需求分析	王正达 尹俊成	2020.04.02	2020.04.21
7	完成概要设计 确定软件系统的总体布局 各个子模块的功能 模块间、与外部系统的关系 技术选型	全体	2020.04.04	2020.04.10
8	完成详细设计 概要设计进一步细化 落实每个模块的具体技术细节	全体	2020.04.10	2020.04.21
9	准备评审资料	王正达 尹俊成	2020.04.02	2020.04.22
10	分工后开发个人部分代码	杨开元 彭文鼎	2020.04.02	2020.05.01
11	集成代码	杨开元 彭文鼎 尹俊成	2020.05.02	2020.05.05
12	系统调试	杨开元 彭文鼎 尹俊成	2020.05.06	2020.05.12
13	准备代码评审	王正达 尹俊成	2020.04.22	2020.05.13
14	进行系统测试	尹俊成	2020.05.14	2020.06.02
15	准备测试文档评审	王正达 尹俊成	2020.05.14	2020.06.03
16	更完备的系统测试并完善文档	全体	2020.06.04	2020.06.10