

智能抓取式机器人
需求规格说明书
SRS209
【版本号 1.0.4】

分工说明

小组名称	嵌不入队	
学号	姓名	本文档中主要承担的工作内容
17373548	吴天昊	对文档第一部分工作内容进行撰写，设计用户界面
17373545	聂志捷	对文档第三、四部分工作内容进行撰写
17373513	吴仪周	对文档第二部分内容进行撰写，并对文档进行整理
17373541	薛晨祺	对文档第五部分工作内容进行撰写，并完成审稿
16061178	刘晓洁	对文档第六、七部分工作内容进行撰写

版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
1.0.0	3.28	吴仪周	薛晨祺	软件需求规格说明书第一版
1.0.1	4.2	吴仪周	薛晨祺	对于原文档进行细化和加工
1.0.2	4.2	吴仪周	薛晨祺	对于界面展示进行修改
1.0.3	4.24	聂志捷	薛晨祺	对于用户需求细节进行修改
1.0.4	6.4	聂志捷	薛晨祺	添加登陆部分的需求细节

目录

1. 范围.....	1
1.1 项目概述.....	1
1.1.1 项目背景.....	1
1.1.2 主要功能.....	1
1.1.3 非功能性需求.....	2
1.1.4 应用场景.....	2
1.2 文档概述.....	2
1.3 术语和缩略词.....	3
1.4 引用文档.....	3
2. 业务需求.....	5
2.1 业务环境.....	5
2.2 用例描述及活动图.....	5
2.2.1 用户手动控制机器人.....	5
2.2.2 机器人自主控制.....	7
2.3 需求用例图.....	9
2.3.1 用户用例图.....	9
2.3.2 机器人用例图.....	10
3. 数据需求.....	11
3.1 类的抽象.....	11
3.1.1 从潜在类中确定实体类.....	11
3.1.2 从潜在类中确定分析类.....	11
3.2 类的类型与关系.....	12
3.3 UML 类图.....	15
4. 功能需求.....	16
4.1 系统功能说明.....	16
4.1.1 机器人的主动控制.....	16
4.1.2 静态或动态障碍物避障.....	16

4.1.3	利用传感器实时建立环境地图.....	16
4.1.4	根据地图和自身位置实现动态路径规划和导航控制.....	17
4.1.5	检测、识别并定位环境中的特定目标，动态接近目标物.....	17
4.1.6	抓取目标物.....	17
4.1.7	语音交互.....	18
4.1.8	异常处理.....	错误！未定义书签。
4.2	Control 类数据流图.....	19
4.3	状态图.....	21
5.	非功能需求.....	22
5.1	性能指标.....	22
5.1.1	响应时间和处理能力.....	22
5.1.2	功耗.....	22
5.2	质量属性.....	23
5.2.1	可用性.....	23
5.2.2	安全性.....	23
5.2.3	可移植性.....	24
5.2.4	可维护性与可扩展性.....	24
6.	用户界面需求.....	24
6.1	用户界面的基本要求.....	24
6.2	界面布局草案.....	25
7.	运行与开发环境.....	28
7.1	运行环境.....	28
7.1.1	硬件环境.....	28
7.1.2	硬件工作环境.....	28
7.1.3	软件环境.....	29
7.2	开发环境.....	29
7.2.1	硬件环境.....	29
7.2.2	软件环境.....	30

1. 范围

1.1 项目概述

1.1.1 项目背景

现如今，智能服务已经占据了相当部分的市场，为此，各商家研发设计了各种专用型机器人来为我们的生活提供便利，例如扫地机器人、点餐机器人，这些机器人能够非常完美的处理重复、单一的任务，并且大大降低了出错的概率，这是人工所无法达到的。

而本项目的开发目的就是要设计制作一款具有包括抓取在内的多重功能的嵌入式机器人，以迎合各类市场的需求。嵌入式系统具有专用性强、实时性好、可靠性高的特点，并且还有较低的功耗，成本也非常低，因此对于我们开发机器人是一个非常好的选择。我们的开发平台基于启智 ROS 机器人，是一款为 ROS 机器人算法开发打造的硬件平台。我们的机器人将能够在一定范围内按照要求识别并抓取我们选定的目标，并且自动探测周围环境、实时建立地图并规划路径，同时还要具有躲避障碍物的功能，当然，为了进一步迎合用户需求，我们还会让机器人具备手动操纵以及语音控制的功能。

1.1.2 主要功能

本项目机器人将具备以下功能：

- 单一类型目标的识别；
- 多类型目标的识别；
- 机器人的主动控制；
- 利用传感器实时建立周边环境的地图；
- 机器人导航过程中动静态避障；
- 根据地图及目标点位置动态规划路径；
- 对目标物按照姿态进行抓取；
- 对异常情况进行识别与处理。

1.1.3 非功能性需求

我们对机器人的非功能性需求如下:

- 低响应时间: 我们希望系统对于用户的每个指令都能够快速反应, 并且有较高的吞吐量和资源利用率;
- 低功耗和较长的续航时间;
- 可靠性: 系统有较强的鲁棒性, 出错率较低, 有一定的容错能力, 并且机器人的整体结构也要足够牢固;
- 易用性: 用户界面美观, 对用户使用较友好, 易于操作;
- 可扩展性: 代码的编写按照模块化的进行, 具有较高的复用性和可移植性;
- 安全性: 能够用户的权限进行区分, 从而执行不同的功能, 以免使用者出现误操作造成安全事故。

1.1.4 应用场景

家庭服务机器人: 我们设计的机器人可以作为未来家庭服务机器人的一个重要功能模块, 能够在用户家中实现精确导航避障, 并且准确地对用户需要的目标进行抓取和传送。

分拣机器人: 当前的快递点快递分拣的任务都是人工完成的, 内容具有很大的重复性, 而且人工还存在一定的犯错概率, 我们设计的机器人可以用于对快递进行高效的分拣归类, 并且出错率也可以保持在非常低的水平。

货运机器人: 在商场中, 当前有很多上货的任务都是人工进行的, 很多时候货物重, 难以分拣, 我们就可以利用具有高强度抓取能力的机器人来进行搬运、上货的任务, 既高效又节省成本。

1.2 文档概述

本文档是北京航空航天大学计算机学院 2020 年春季学期软件工程-嵌入式课程中的软件开发计划文档。用于描述小组项目的需求规格, 理清各个部分需要达到的标准等。小组成员包括吴天昊、聂志捷、吴仪周、薛晨祺和刘晓洁, 其中吴天昊负责担任组长职务。在新冠肺炎疫情尚未结束之前, 整个开发将保持以线上

沟通的方式来进行，具体的硬件实施将在后续依相关情况进行跟进。

计划书的内容按照项目概述、业务需求、数据需求、功能需求、非功能需求、用户界面需求和运行与开发环境来进行组织和编写，具体每个部分还进行了更细致的划分和补充，最终目的就是用于描述小组整个设计和开发 ROS 机器人软件工程项目各个部分的计划和要求，每个部分都由负责该部分的成员进行了认真的分析和调查，从而设计出更符合用户需要、更人性化的产品。

1.3 术语和缩略词

表 1-1 术语和缩略词

术语/缩略词	解释/全称
ROS	Robot Operating System/机器人操作系统
Gmapping	Gmapping 是基于滤波 SLAM 框架的常用开源 SLAM 算法，可以实时构建室内地图，在构建小场景地图所需的计算量较小且精度较高
Rviz	rviz 是 ROS 官方提供的一款 3D 可视化工具，我们需要用到的所有机器人相关数据都可以在 rviz 中展现
DFD	数据流图 (Data Flow Diagram)
航点	含有明确坐标和编号的导航路径节点
地面点云	实时 SLAM 所探测到的周围障碍方位信息
基准格栅	地图文件中存储的障碍方位信息
行为脚本	针对某一关键词命令所设定的一串行动序列

1.4 引用文档

T. Wiedemeyer, “IAI Kinect2,” https://github.com/code-iai/iai_kinect2, Institute for Artificial Intelligence, University Bremen, 2014 – 2015, accessed June 12, 2015.

启智 ROS 版_开发手册——20181109

启智 ROS 机器人

2. 业务需求

2.1 业务环境

用户下载相应平台通过语音操作或手动操控，控制机器人对物品的搬运工作。机器人可以实时与用户进行交流，明确自己的任务目标。同时，机器人能在复杂场景中进行环境建模，并进行路线规划，在行进过程中实现避障，最终查找并定位捕获所需物品。

2.2 用例描述及活动图

2.2.1 用户手动控制机器人

主要参与者：用户。

目标：机器人根据用户的指示做出相应的行进、旋转和抓取动作。

前置条件：完整配置系统，用户具有合法的账号密码，能成功登陆。

启动：用户选择手动控制。

场景：

1. 用户点击登陆按钮。
2. 用户输入账号密码，登陆机器人控制软件。
3. 系统显示主要功能按钮。
4. 用户选择“手动控制”。
5. 用户点击选择前后左右、旋转和机器臂运动控制。
6. 机器人根据相应指示做出行动。
7. 机器人做完动作等待下一题指令（指令捕捉速度快，一直点击行进不会停顿等待）。
8. 退出功能按钮。

优先级：中。

使用频次：中。

次要参与者: 机器人。

异常情况:

1. 行进途中出现障碍物, 出现场景 6。
2. 行进途中遭受碰撞等意外情况, 出现场景 6。

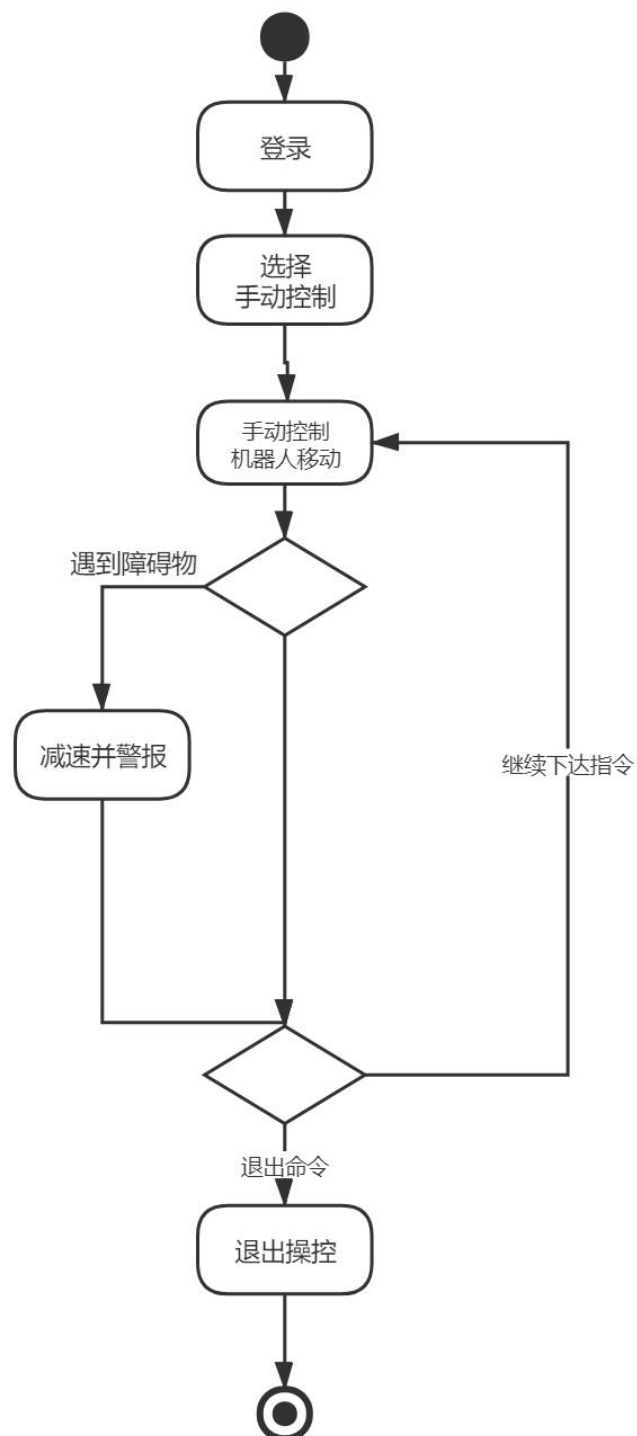


图 2-1 手动控制活动图

2.2.2 机器人自主控制

主要参与者: 机器人。

目标: 机器人根据用户语音命令, 自主做出路径选择和抓取操作, 最终抓获目标物, 并返回用户所在地址。

前置条件: 完整配置系统, 用户具有合法的账号密码, 能成功登陆。

启动: 用户选择自动控制, 进行语音操控。

场景:

1. 用户点击登陆按钮。
2. 用户输入账号密码, 登陆机器人控制软件。
3. 系统显示主要功能按钮。
4. 用户选择“自动控制”。
5. 机器人接受用户语音指令, 确定所需抓取的目标物。
6. 机器人根据已有地图进行目标搜索, 若不存在, 则实时旋转 360 度对场景进行建模并识别目标物。
7. 进行路径规划并自驱动行进寻找目标物, 在适合地方进行抓取。
8. 退出功能按钮(可强制终止程序)。

优先级: 中。

使用频次: 高。

次要参与者: 用户。

异常情况:

1. 语音无法识别, 出现场景 5。
2. 行进途中出现障碍物, 出现场景 7。
3. 行进途中遭受碰撞等意外情况, 出现场景 6, 7。

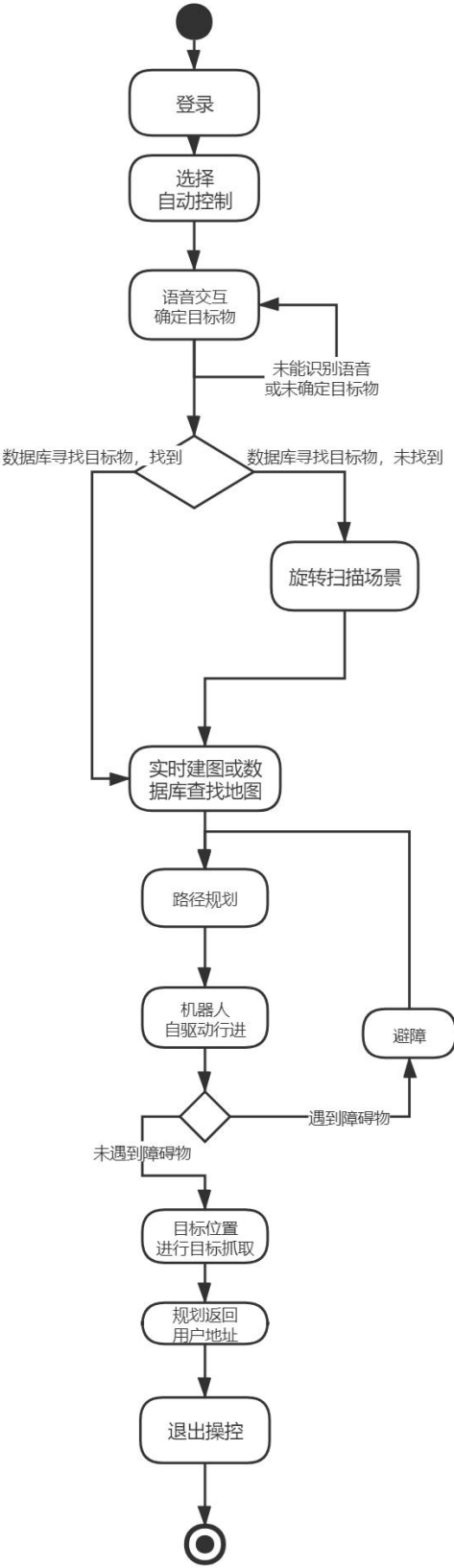


图 2-2 自动控制活动图

2.3 需求用例图

2.3.1 用户用例图

用户分为普通用户和管理者。

普通用户进行用户注册和用户登录后可以对于机器人进行手动操作或者要求机器人自动操控，完成其抓捕任务。在手动控制模式下，用户可以使用操作界面操控机器人进行前后左右的位置移动、左旋右旋的旋转移动和三维机械臂抓取操控。在自动控制模式下，用户与机器人进行语音交互，并传达需要抓取的目标物。

管理员具有普通用户的一切权限，且可以对于普通用户进行账户删除等信息表管理工作，同时管理员可以对于后台地图建模数据库进行特定的修改和删除操作。

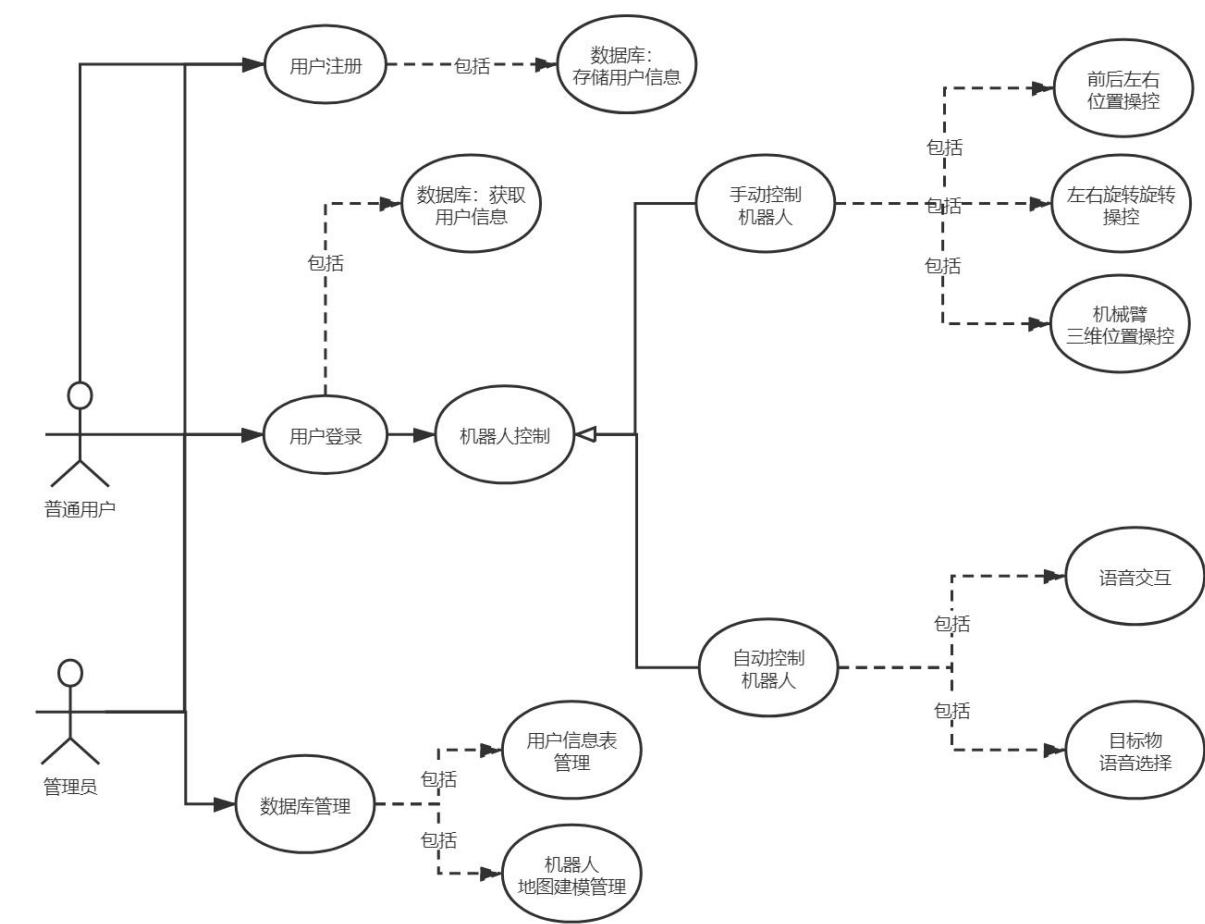


图 2-3 用户用例图

9

2.3.2 机器人用例图

在人工抓取状态下，机器人识别用户发射来的红外信号进行移动，此时机器人只进行路障检测，当发现路障时进行警报并减速提醒用户。在遇到突发情况和异常时，将紧急制动。

在自动模式下，机器人识别用户的语音请求，同时具有实时地图建模、障碍物检测并规避、路径规划并自驱动行进、紧急制动和异常处理的功能。

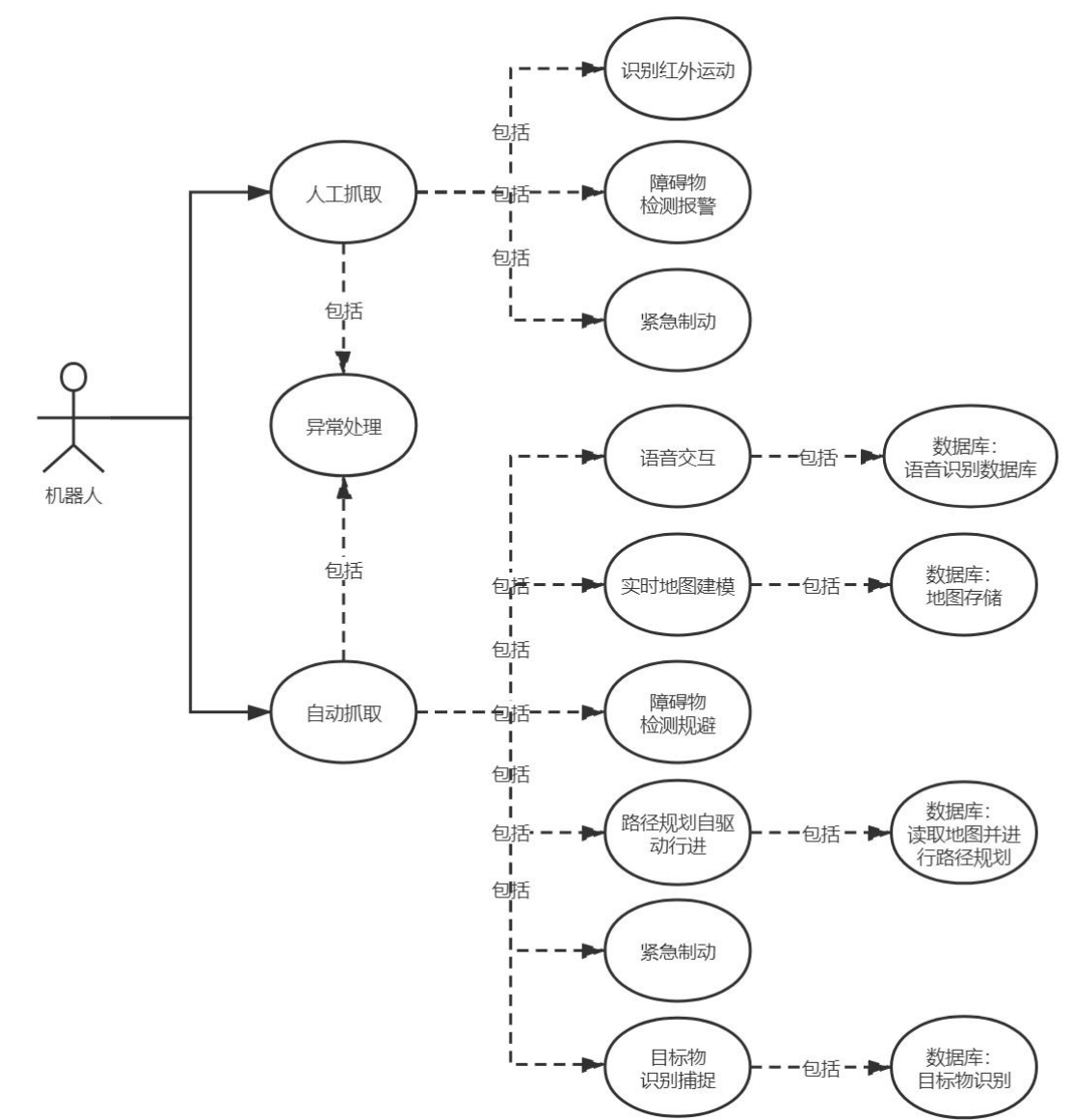


图 2-4 机器人用例图

3. 数据需求

3.1 类的抽象

3.1.1 从潜在类中确定实体类

本产品的用户分为普通家庭用户和管理者。

普通家庭用户可以对于机器人进行声控或手控要求，完成其抓捕任务。其具体表现为，登录进入界面之后，在初始界面选择不同的模式。

在手动操控状态下，机器人的运动轨迹由用户进行操作，操作界面将提供给用户方向键进行操作，此时机器人只进行路障检测，关闭地图建模和路径规划功能，当发现路障时进行警报并减速提醒用户，用户手动操作避过路障。当到达航点准备进行捕捉时，用户可从方向控制界面切换成目标捕捉界面，对机器人抓取臂进行操作，抓取到所需的目標物。

在自动模式下，用户使用语音或输入特定的物品与机器人进行交流，机器人在得到准确的物品指令后，会首先在自身存储的地图中寻找目标物，如果有将直接进行路径规划去寻找；如果没有，将从当前位置进行 360 度扫描进行建模和目标物的确定，再实行查找捕捉。在捕捉完成之后，搜索用户的位置，选择路径把物品交给用户。在此过程中，机器人将有避障、建模、路径选择的能力，过程中，用户可以随时语音或者手动选择停止机器人的抓捕任务。

管理者可以对于机器人后台的路径进行进一步规划和调整，对于普通家庭用户的权限进行控制。在登录管理者界面后，可对机器人的基本情况进行操作，如清空后台数据，管理增加和删除普通用户信息等，同时管理者账号也可以对机器人进行如普通用户一样的操作。

3.1.2 从潜在类中确定分析类

确定分析类的依靠原则为：

1. 保留信息 2. 所需服务 3. 多个属性
4. 公共属性 5. 公共操作 6. 必要要求

表 3-1 分析类筛选表

潜在类	表现形式	不符合条件
普通家庭用户	角色	
管理者	角色	
机器人	事物	
控制界面	事物	
模式设置	事件	3
手动控制状态	事物	3
轨迹	事物	1、2
用户	角色	
路障	事物	
航点	事物	
抓取臂	事物	
目标物	事物	
自动模式	事物	1、2
地图	事物	3
物品	事物	5
目前位置	场所	1、2
语音	事物	3
后台数据	事物	1、2
账号	事物	1、2

3.2 类的类型与关系

实体类: 普通用户、管理员、用户、机器人、机械臂、物体、航点、障碍物

- **普通用户:** 使用者是一类**权限较低**的使用者, 一般情况下使用者仅负责给出该项任务或实验所需的一些必要的输入, 部分属性继承于用户基类。

- **管理员:** 机器人管理人员负责**机器人的使用和维护**, 负责机器人内部相关程序的编写。是机器人的实际操作者。可对机器人的基本情况进行操作, 如使用 SLAM 算法构建地图清空后台数据, 管理增加和删除普通用户信息等, 部分属性继承于用户基类。
- **用户:** 用户基类, 包含**普通用户和管理员共同的属性**, 也可作为对象属性, 包含在控制类中。
- **机器人:** **具体任务的执行者**, 在机器人管理人员的指令下, 结合用户的输入, 完成相应的任务。机器人最主要的任务是负责**运动控制**。在此项功能之外, 机器人还负责各项传感器**读入数据**的处理、**寻路算法**的实现、**追踪设置的路径**等功能。
- **机械臂:** 机器人的**可选安装部分**, 安装后可进行物品的抓取可以使用控制模块对机械臂的**状态, 间距和高度**进行定量控制, 配合机器人的其他模块, 可实现特定目标的抓取与运送。
- **物体:** 机器人 Kinect2 扫描器所识别出的物品, 具有**三维坐标, 特征与宽度**等属性, 是机器人所要**识别和抓取**的目标。
- **航点:** 机器人在导航过程中所要到达的**目标点位**, 具有**二维坐标和特定编号**, 可一次设定**若干个**, 机器人需要按照编号**依次**导航通过航点。
- **障碍物:** 机器人在移动过程中所碰到的障碍物, 具有**二维坐标**, 分为动态和静态两种, 其中静态障碍物可以根据**路径规划算法**躲避, 动态障碍物需要用户**手动操纵**或者等待远离才可继续行动。

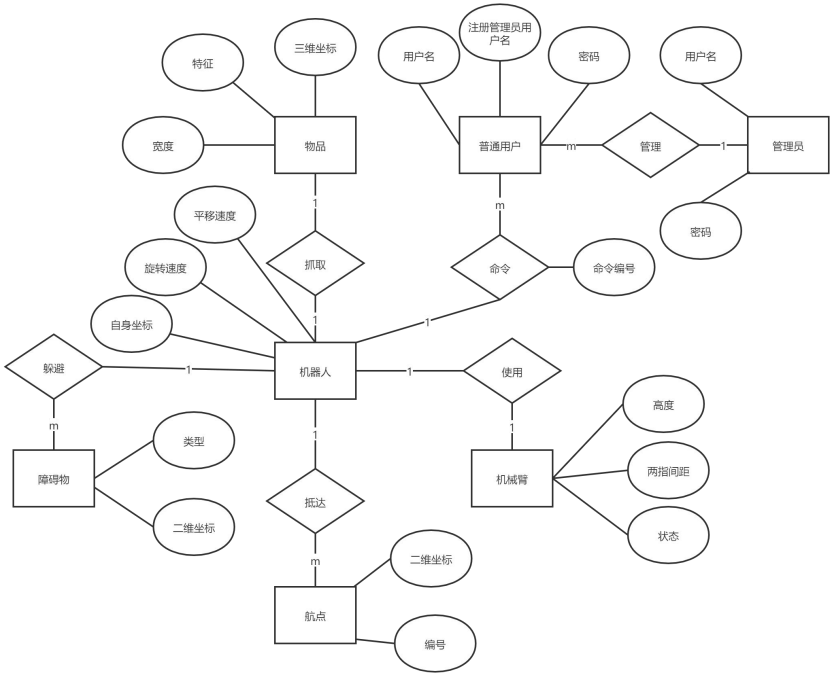


图 3-1 E-R 图

边界类：移动控制、语音交互

- **移动控制**：使用物理摇杆对机器人的**平移旋转与混合运动**进行操纵，在遇到障碍时可**紧急避险**并发出警告，也可进行地图的构建任务。
- **语音交互**：用户针对特定需求说出**预先约定好的关键词语**，语音识别模块将根据科大讯飞语音识别功能进行关键字的**文字转换**，并命令机器人执行对应的**行动序列**。

控制类：用户控制、管理员控制、异常处理

- **用户控制**：用户控制类管理用户的**登陆、注销与验证**。在通过验证后，普通用户通过用户控制类调用相应的**移动、抓取**等接口使机器人做出相应动作。
- **管理员控制**：管理员控制类继承于用户控制类，包括用户控制类的所有功能，并添加了**构建 Gmapping 地图，删除普通用户账号**的功能。
- **异常处理**：对机器人发生的异常进行**播报和处理**的类，当机器人在发生**翻倒，关键组件受损**等严重事故时发出警报，请求维护人员迅速解决故障问题；当机器人发生**障碍物过多、导航失效、抓取物体掉落**等轻微事故时，将发生故障的部分参数进行保存，并发送到**指定接口**方便代码编写人员排查故障。

3.3 UML 类图

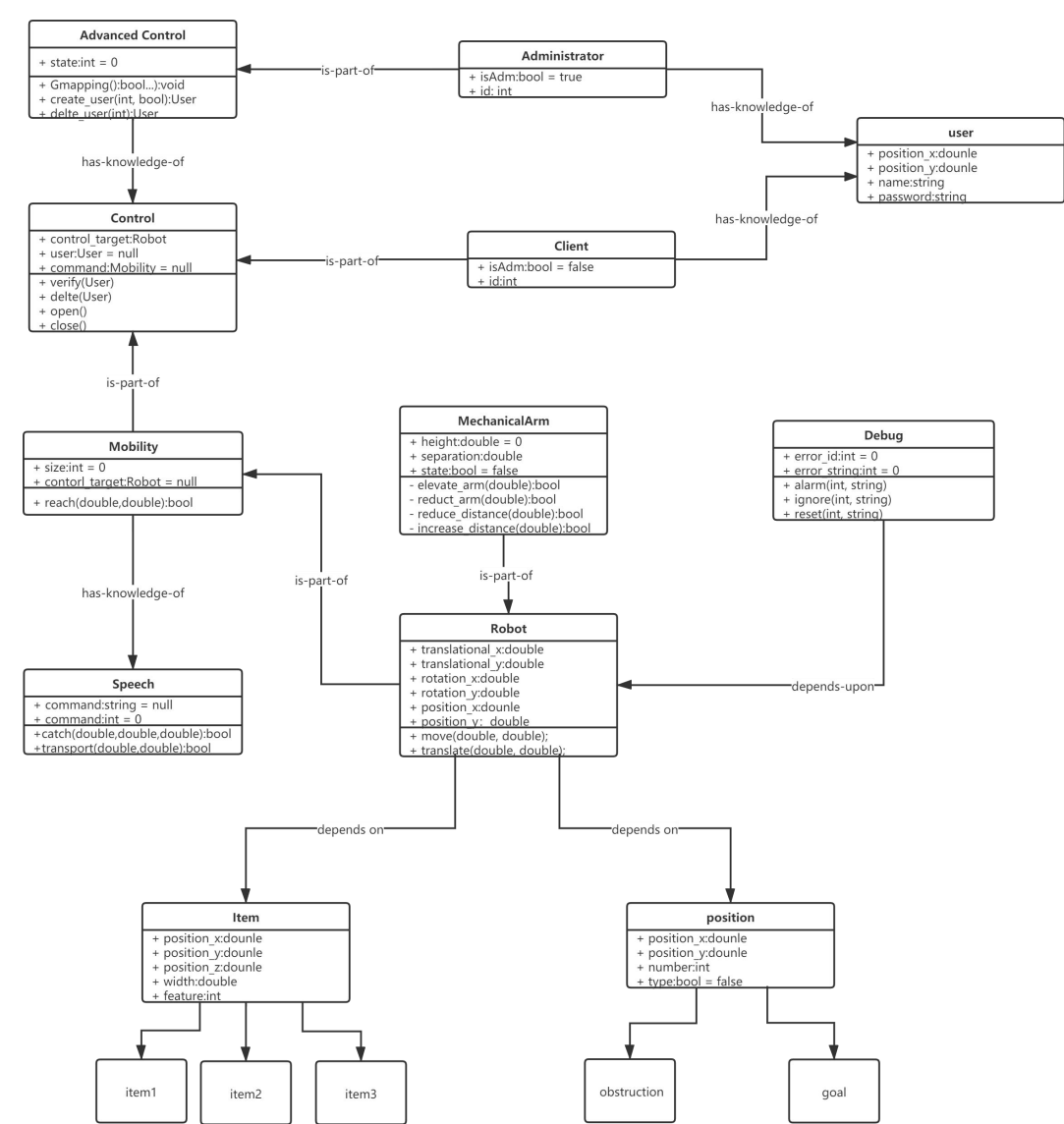


图 3-2 UML 图

4. 功能需求

4.1 系统功能说明

4.1.1 机器人的主动控制

本功能的实现依靠 `ros.h` 与速度结构体类型 `geometry msg::Twist` 的定义文件实现。在打开机器人底盘面板的红色急停开关后,可以采用以下几种方式控制机器人的移动:

- (1) 前后左右移动: 手柄的左侧摇杆, 控制机器人运动的平移向量, 上下分别控制前后移动, 左右分别控制侧向平移;
- (2) 旋转: 手柄的右侧摇杆, 控制机器人运动的转动向量, 拨向左是左转, 拨向右是右转;
- (3) 旋转移动: 机器人的运动平移向量和转动向量可以叠加的, 即可以一边平移一边旋转。

4.1.2 静态或动态障碍物避障

本功能的实现依靠机器人所带的激光雷达实现。在开启激光雷达后, 可在 `Rviz` 界面看到周围 360° 的障碍物分布情况, 并形成障碍物轮廓的俯视二维点阵输入到 ROS 系统中。静态或动态障碍物避障功能可实现:

- (1) 监测: 在移动过程中实时监测运动路径上的障碍物和距离;
- (2) 停止: 在距离障碍物一定距离时, 停止运动
- (3) 判断: 选择左侧或右侧中更空旷的一侧, 并将机身进行旋转;
- (4) 躲避: 向当前方向前进一定距离, 直至原本行驶方向路径上无障碍物;
- (5) 恢复: 停止平移并旋转回原本行驶方向, 继续移动。

4.1.3 利用传感器实时建立环境地图

本功能的实现依靠 `Gmapping` 算法技术实现。运行 `Gmapping` 算法期间可以观察到以下情况:

- (1) 扫描: 机器人周围的地面基准变成了深灰色, 出现白色扫描区域;
- (2) 识别: 探测到的障碍点在 Rviz 上呈红色; 静态障碍物轮廓呈黑色;
- (3) 更新: 移动机器人时, 白色区域增加, 静态障碍物的轮廓越来越接近;
- (4) 存储: 绕场地移动一圈后, 可以得到保存的后缀为.pgm 的地图文件。

4.1.4 根据地图和自身位置实现动态路径规划和导航控制

本功能的实现主要依靠蒙特卡洛自适应定位算法、move_base 软件包和 MapTools 软件。在调节优化至红色激光雷达数据点和静态障碍物轮廓大致贴合后, 可以通过以下步骤进行导航控制:

- (1) 在 Rviz 视图加入若干个目标位置的航点以及机器人的朝向;
- (2) 机器人自动规划路径并按照顺序依次移动至航点;
- (3) 到达一个航点后, 机器人不停止, 立即规划至下一个航点;
- (4) 在抵达最后一个航点后, 机器人停止移动, 并旋转至设定的朝向;

在导航期间可以实现的其他功能包括:

- (5) 可在代码中修改航点坐标或增减航点数目, 机器人将重新规划路径;
- (6) 对路径上的障碍物可以有效躲避, 不发生碰撞;

4.1.5 检测、识别并定位环境中的特定目标, 动态接近目标物

本功能的实现主要依靠 Kinect2 视觉传感器、PCL 平面检测算法。在对 Kinect2 的角度倾角参数进行标定、调节地面点云和基准栅格完全重合后, 可以实现以下功能:

- (1) 接近: 根据导航指令接近桌子或货架、机器人正面面向桌子或货架;
- (2) 检测: 将桌面上的物品标注出来, 并计算其三维坐标;
- (3) 识别: 机器人根据指令选定目标物品;
- (4) 动态调整: 机器人调整自己与桌面的距离并左右平移对准物品。

4.1.6 抓取目标物

本功能的实现依靠 ros.h 与机器臂结构体类型 sensor_msgs::JointState 的定义文

件实现。在正确安装机器臂的前提下，在目标物正前方，能按照如下顺序使机械臂状态切换：

- (1) 初始：手臂处于最下端折叠收起状态；
- (2) 展开：手臂从零位上升到展开状态的最低位；
- (3) 上升：机器臂上升指定高度，直至物体中心位置高度；
- (4) 抓取：机械臂调节手抓的两指间距，直至夹住物体
- (5) 提起：机器臂继续上升制定高度，直至物体脱离桌面，悬停空中

4.1.7 语音交互

本功能的实现依靠科大讯飞的语音识别引擎和阵列麦克风实现。主要能够进行以下涉及语音交互的行为：

- (1) 设置：自定义目标关键词，并可以配置关键词所对应的行为脚本；
- (2) 识别：用户对着麦克风说出目标关键词，可准确识别语音命令结果；
- (3) 执行：机器人根据行为脚本无差错执行命令；
- (4) 待命：在完成命令后进入监听状态，等待下一条命令输入。

最终能够实现“特定地点物品抓取”和“递送饮料”等功能。

4.1.8 登陆/注册与用户管理

本功能实现依靠数据库的会员信息与管理员高级控制实现。任意使用者均需要注册和登陆后才能使用机器人的功能，注册时根据权限不同，可分为管理员和普通用户。在登陆/注册过程中，需要确保以下功能：

- (1) 用户名重复检测：当新注册用户名与已注册用户名重复时，发出警告
- (2) 密码错误检测：当登陆用户输入错误密码时，发出警告
- (3) 用户不存在检测：当登陆用户输入不存在用户名时，发出警告

4.1.9 异常中断与恢复

本功能实验依靠 C++ 内自带的异常类与编写的异常处理程序实现。当机器人由于外部因素（机器人侧翻，底盘抱死等）或内部因素（代码死循环，指令无法

识别等），将进行以下步骤：

- （1） 保存日志信息：将产生异常时的数据存储在数据库中
- （2） 等待维护人员解决：维护人员通过修理或代码调试使系统正常运行
- （3） 中断恢复：系统从日志中读取异常时正在进行的任务，并继续执行

4.2 Control 类数据流图

DFD Level0:

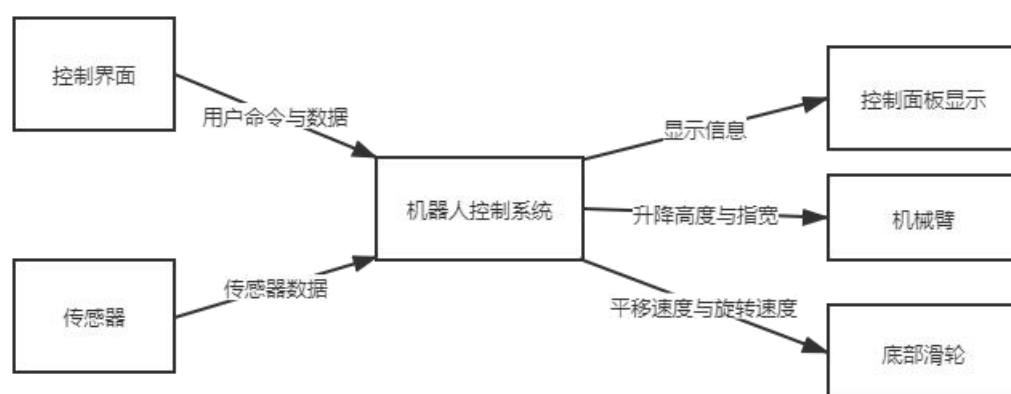


图 4-1 DFD Level0 图

DFD Level1:

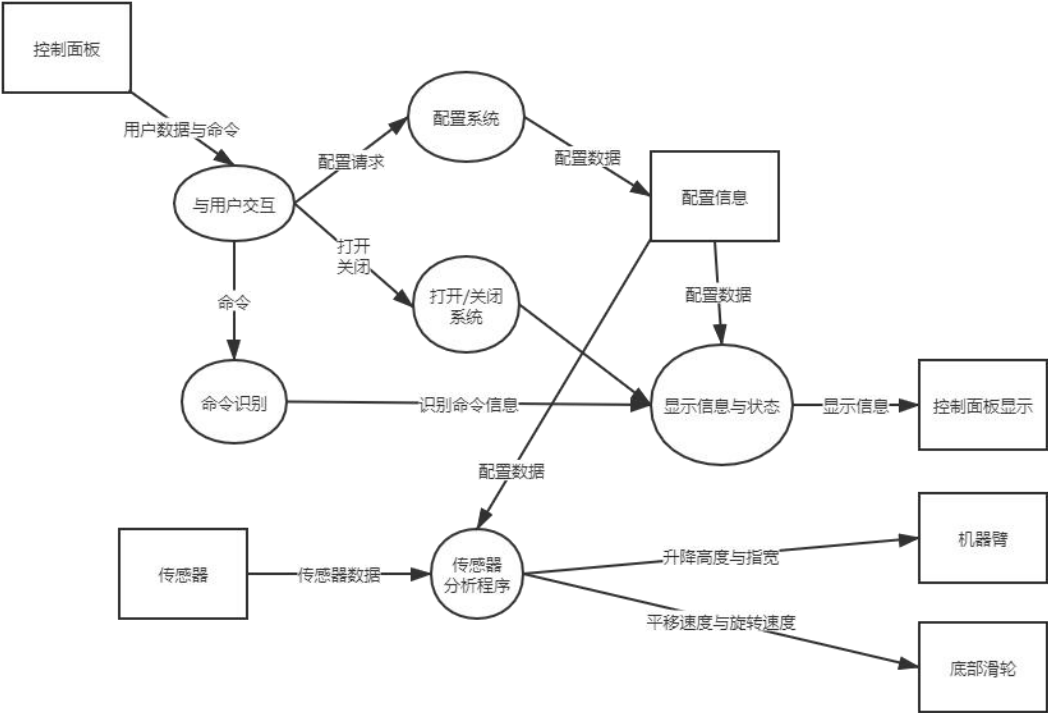


图 4-2 DFD Level1 图

DFD Level2:

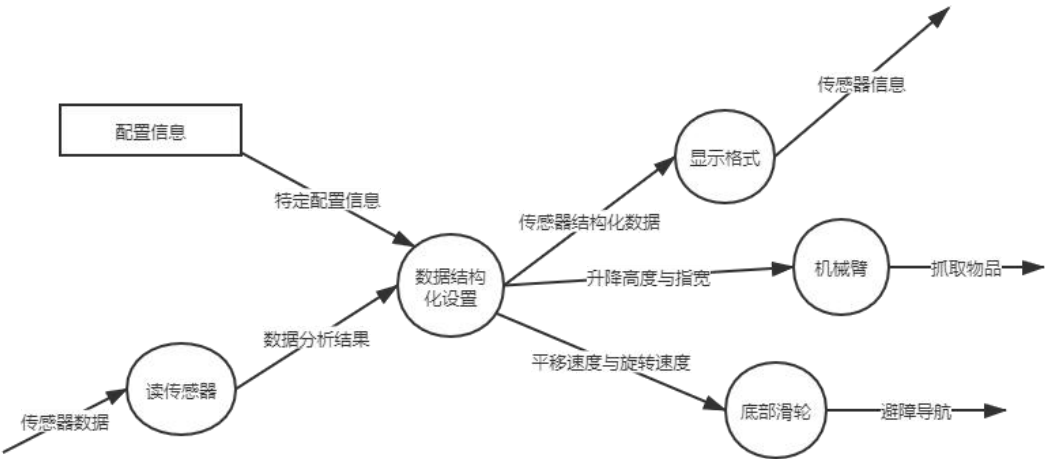


图 4-3 DFD Level2 图

DFD Level3:

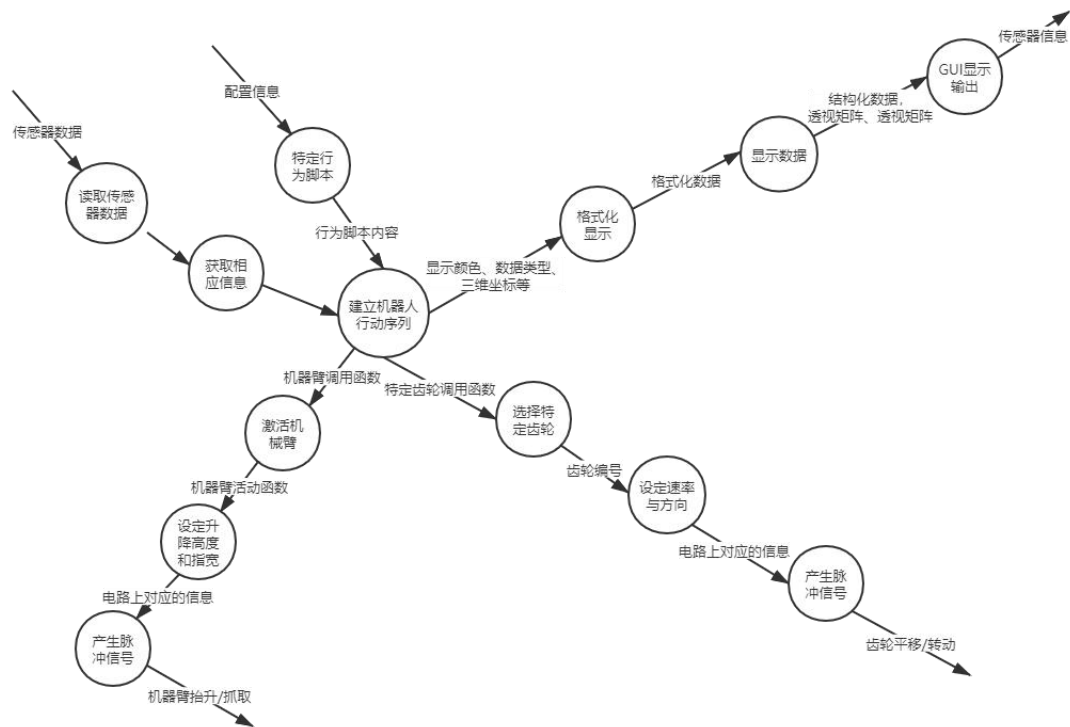


图 4-4 DFD Level3 图

4.3 状态图

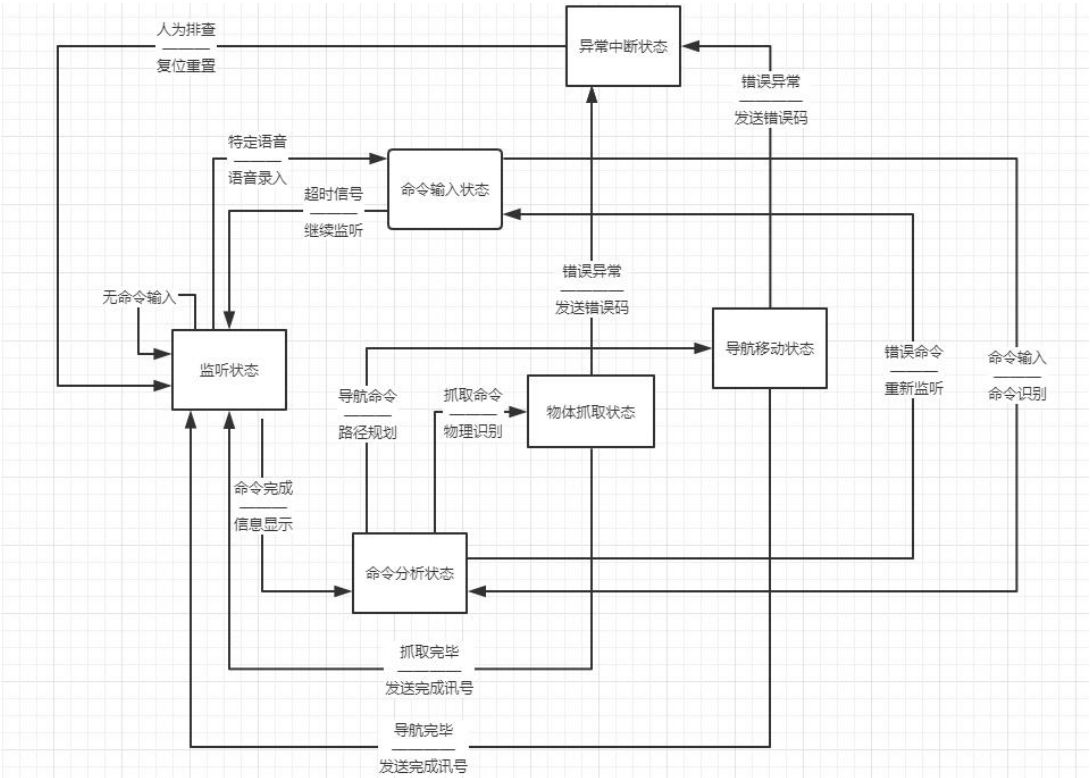


图 4-5 状态图

5. 非功能需求

5.1 性能指标

5.1.1 响应时间和处理能力

1. 机器人内部的信息具有较高的更新率和反馈率。
2. 给定起点和终点,机器人可以在 80s 内完成路径规划。对于复杂的环境情况,应在 300s 内完成路径规划。
3. 机器人能够在 1s 内从静止加速到 0.1m/s 进行前后移动,从起始位置到终点平均移动速度在 0.08-0.15m/s 之间。并且可以以 0.1 弧度/s 的速度向左或向右旋转。
4. 机器人移动过程中,激光雷达检测到障碍物时,应在 1s 内减速至停止移动。
5. 具有较高的精准度,能够准确移动到目标位置,准确识别,抓取物体,误差 $\leq 0.1\text{m}$ 。
6. 在 95%的情况下,能在 1s 内对一般指令做出反应。
7. 在 95%的情况下,能在 5s 内对语音指令做出反应。

5.1.2 功耗

1. 机器人的电源由电池模块供给,该电源模块内部由 7 枚 3500mA/h 容量的锂离子电池串联组成,正常工作输出,输出电压范围 23.1V 至 29.4V。
2. 电机工作电压 24V,额定功率 14W,持续工作电流 1.4A。
3. 机器人可以连续工作 4h 以上。

5.2 质量属性

5.2.1 可用性

1. 易操作:控制指令的设计简单,用户可以在较短时间内学习使用方法,易于用户操作。
2. 可靠性: 我们预计整个机器人设备在正常使用下可以稳定地运行**两年以上**, 并且对于通常物品的识别和**抓取正确率达到 98%及以上**。
3. 容错能力:系统具有一定的容错能力和抗干扰能力。在非硬件故障或非通讯故障时,系统能保证正常运行。**在 95% 的故障中,系统最多可以在 30s 内重启或解决故障**。
4. 提供数据备份和恢复功能,若由于系统错误或其他原因导致系统数据丢失或被破坏时,可以及时的恢复和还原数据。

5.2.2 安全性

1. 启智 ROS 机器人是室内机器人, 在此环境之外运行可能会损坏机器人。工作平面需要能够承载**不小于 40kg 的重量**。启智 ROS 机器人原则上在水平平面上工作,**坡道坡度不大于 15 度**,坡道倾斜度过大可能导致倾覆。
2. 启智 ROS 机器人**不具备防水功能**, 在任何情况下, 启智 ROS 机器人都不应该与雨水, 雾, 地面积水以及任何其他液体接触, 否则可能导致电路和机构损坏。
3. 启智 ROS 机器人设计工作温度为 **15° C 到 35° C 之间**,使用中务必远离明火和其他热源。
4. 机器人移动过程中应避免与室内人员和物体相撞, 以免发生危险。
5. 对用户的权限进行区分, 从而执行不同的功能, 以免使用者出现误操作造成安全事故。
6. 记录运行日志,对运行中的系统错误和用户关键操作信息进行记录,便于追踪。

5.2.3 可移植性

1. 本机器人只能运行在 Ubuntu16.04+ROS Kinetic 环境下,暂不支持移植。

5.2.4 可维护性与可扩展性

1. 保留历史各版本的源代码。
2. 将系统功能模块化,支持灵活开发,减少重复开发量。
3. 普通的修改请求或 bug 修复应在 1~2 天内完成,对于重大的需求或设计修改应在一周内完成。
4. 应有清晰的系统结构,命名规范,界面规范,提示和帮助信息规范。

6. 用户界面需求

6.1 用户界面的基本要求

1. 界面元素: 用户界面中的所有元素不能出现错误或存在二义性的内容,保证界面文字、颜色、图案的风格一致。界面整洁、美观,字体大小适当,易于使用。
2. 用户角色: 界面面向有一定计算机使用基础的用户。
3. 操作方式: 界面中设置帮助键,用户能够查阅操作指南;可通过鼠标,键盘与界面各个元素进行简单便捷的互动。颜色使用适当,颜色统一,色彩搭配简洁清楚,选择长期使用不易疲劳的颜色。遵循对比原则,深色背景使用浅色文字;浅色背景使用深色文字。
4. 功能完整: 软件的全部功能都能够在用户界面和菜单中找到。界面中设置一些常用的快捷按钮,便于用户快速进行操作。例如启动键,停止键,登录键,语音控制相关按键,和抓取选择的按钮等。
5. 区域排列: 整齐合理;一般的标签右对齐,编辑框左对齐。字符左对齐,数字右对齐。

6. 异常处理：当用户进行危险操作时，会弹出警告窗口并且提醒用户做出对应的应急操作。

6.2 界面布局草案

1. 初始界面



图 6-1 初始界面

主要包括登录和注册界面，用户可以登录和注册，系统将自动识别是普通用户还是管理员。

2. 选择操作页面

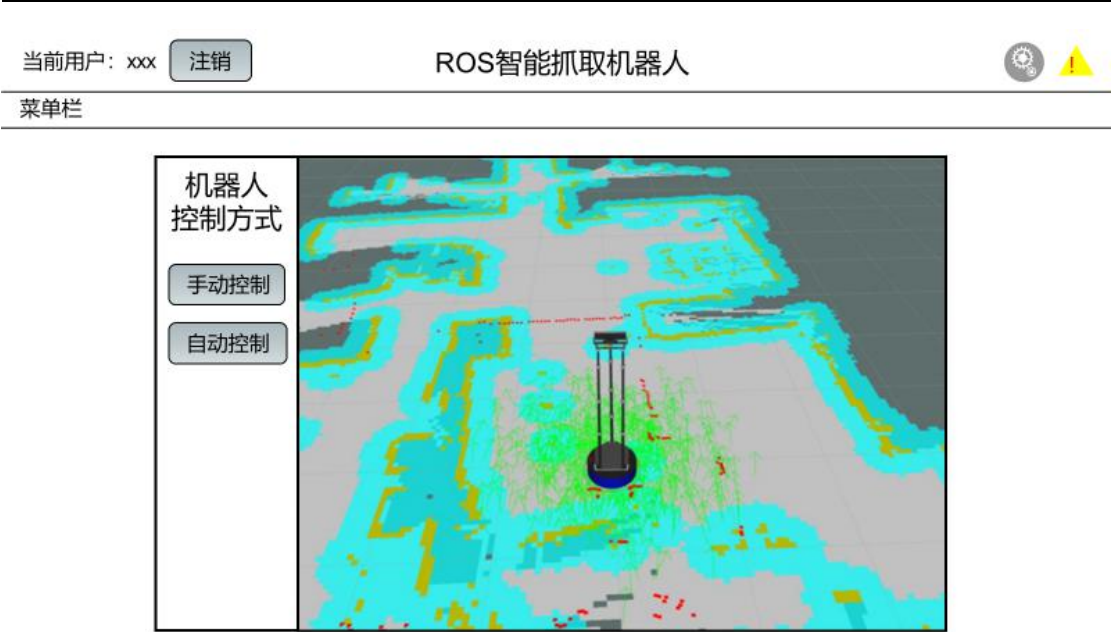


图 6-2 选择操作页面

用户登录后会在左上角显示当前用户名和注销键，同时将显示机器人目前所在位置。用户可根据需要选择手动控制或者自动控制。

3. 手动控制：

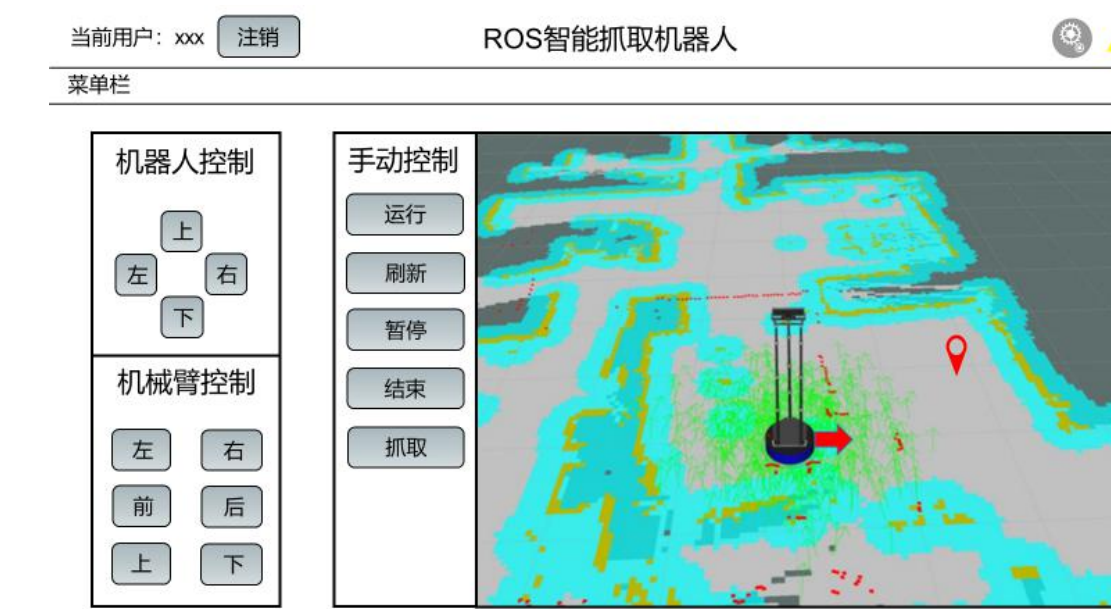


图 6-3 手动控制页面

用户可依照左边的按钮对于机器人进行手动控制，当调整到合适位置按抓取按钮进行抓取。同时，还有暂停，结束等功能。

4. 自动控制界面

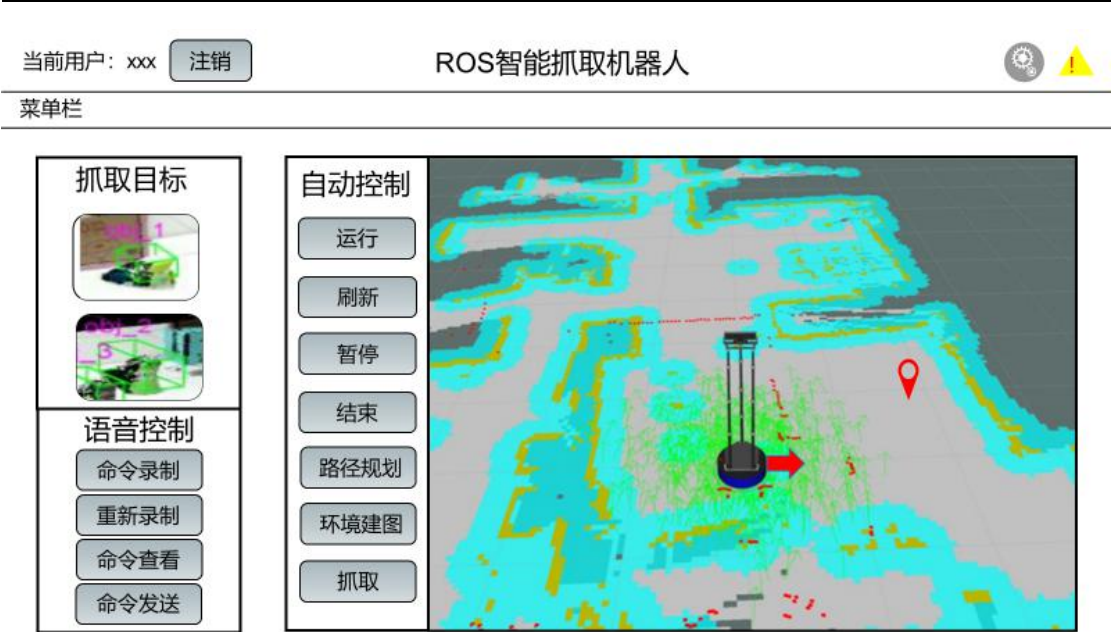


图 6-4 自动控制界面

用户选择自动控制时，与机器人进行语音交互，机器人识别语音后会在左上方显示目标抓取物的示例，用户可依此进行判断是否为自己所需抓取的物品，右侧包括暂停、结束等功能按键。

5. 弹出警告

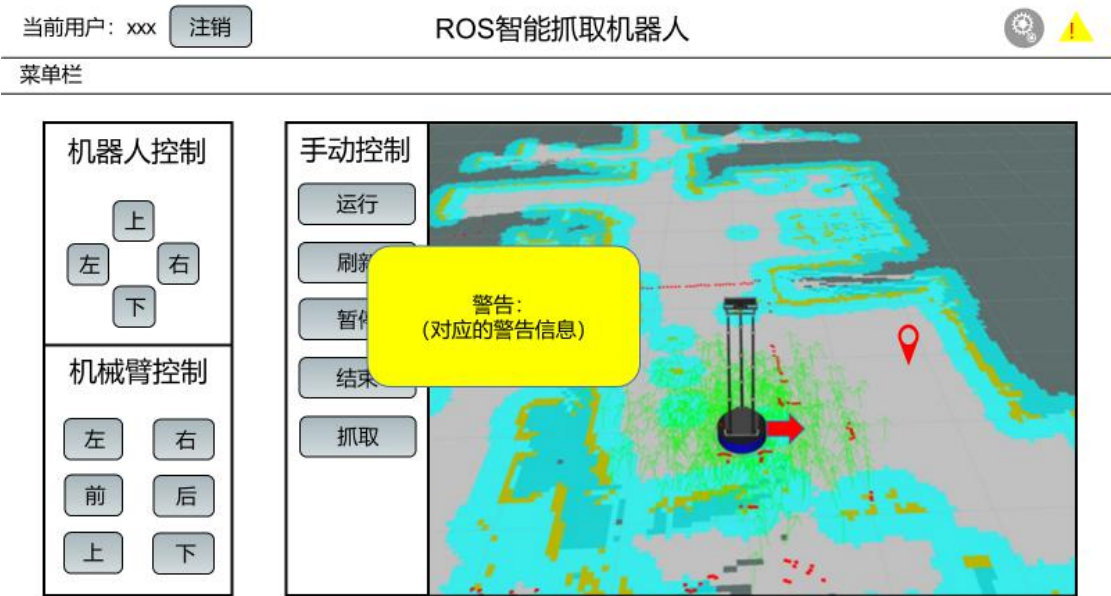


图 6-5 警告样例

在手动控制过程中，当机器人遇到障碍物时会在页面提示警告，同时报警减速。

7. 运行与开发环境

7.1 运行环境

7.1.1 硬件环境

表 7-1 运行硬件环境

名称	数量	参数/功能
启智 ROS 机器人	1	重量约为 30kg（包含抓取组件），承载能力 10kg
机载平板电脑	1	运行 ROS 操作系统，通过 USB 接口与机器人底盘内的 USB_HUB 连接。
视觉传感器	1	kinect2 视觉传感器
激光雷达	1	思岚（SLAMTEC）RPLIDAR A2。测距范围：0.15 米-12 米；扫描角度：360°；测距分辨率：<实际距离的 1%；角度分辨率：0.9°；扫描频率：10Hz。
IMU	1	启智控制器内置 MPU6050 惯性测量传感器
启智伺服电机模块	1	内置驱动控制板,电流环伺服周期为 50 μ s,速度环、位置环伺服周期为 1ms。
启智控制器	1	通过 RS485 总线与启智伺服电机模块通讯
启智电池模块	1	正常工作输出电压范围为 23.1V 至 29.4V
轮子	3	全向轮
机械手臂	1	用于抓取桌面上物品的机械臂，该机械臂提供两个控制量：上升高度和手爪的闭合宽度。

7.1.2 硬件工作环境

1. 硬质工作平面，可承载不少于 40kg 的重量，坡度不大于 15 度。
2. 温度 15°C~35°C。
3. 远离雾，地面积水，雨水及任何其他液体。

7.1.3 软件环境

系统环境: kinetic 版本 ROS 系统, 基于 Ubuntu16.04

软件包:

表 7-2 运行软件环境

软件包	说明	版本限制
启智 ROS 源代码		kinetic 版本
OpenCV	计算机视觉库	2.4.X
cmake	安装(编译)工具	无
PCL	点云相关开源 C++编程库	1.7.X
libusb	USB 驱动	>=1.0.20
TurboJPEG	图像编辑解码器	无
OpenGL	开放式图形库, 用于渲染 2D、3D 矢量图形	无
OpenCL/CUDA/VAPPI	GPU 加速(不同 GPU 芯片可选不同软件包)	无
OpenNI2	开放自然交互的框架	无
科大讯飞在线语音识别包		kinetic 版本
导航地图工具包		无
RPLIDAR ROS	激光雷达驱动	无

7.2 开发环境

7.2.1 硬件环境

表 7-3 开发硬件环境

处理器	i7-4720HQ/i5-7400
内存	8GB
系统类型	64 位操作系统, 基于 x64 的处理器
GPU	NVIDIA GTX960M

7.2.2 软件环境

表 7-4 开发软件环境

操作系统	Ubuntu 16.04
	ROS Kinetic
IDE	Robo Ware Studio
python 功能支持（调试）	pylint
clang-format 功能支持（自动化整理代码）	clang-format3.8 及以上版本