

目标检测及抓取型机器人
软件测试说明书
SRS 202
版本号 3.0

分工说明

| 小组名称 | | |
|----------|-----|--------------------|
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 17373505 | 曹博文 | 2, 4 |
| 17231002 | 曹玥 | 1, 3, 5, 进行整体排版及组织 |
| 17373552 | 莫策 | 4 |
| 17373210 | 王博文 | 5, 6, 进行审核 |
| 17373128 | 全庆隆 | 4, 6 |

版本变更历史

| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
|------|-----------|-------|-----|------------------------|
| v0.7 | 2020.5.26 | 曹玥 | 王博文 | 主要针对文档用例以及之前部分 |
| v0.8 | 2020.5.28 | 曹玥 | 王博文 | 添加测试分析后半部分内容 |
| v1.0 | 2020.5.28 | 曹玥 | 王博文 | 添加测试结果, 测试分析部分, 完善测试用例 |
| v1.5 | 2020.6.3 | 曹玥 | 王博文 | 修改部分单元测试用例和测试结果 |
| v2.0 | 2020.6.4 | 曹玥 | 王博文 | 加入系统测试用例及结果 |
| v3.0 | 2020.6.7 | 曹玥 | 王博文 | 修改部分格式, 完善文档 |

1. 范围

1.1 项目概述

本项目意在开发一个目标检测及抓取型机器人。

本项目的开发基于启智 ROS 机器人。启智 ROS 机器人是一款为 ROS 机器人算法开发量身打造的机器人硬件平台，拥有硬件里程计、激光测距雷达、立体视觉相机和语音输入输出阵列等一系列硬件设备，完美适配 ROS 的 TF、Navigation、Actionlib 和 Pluginlib 子系统，是深入学习 ROS 和高级机器人算法验证开发的理想平台。

本项目目标实现一系列功能性需求，其中包括：实现机器人的主动控制；静态或动态障碍物避障；机器人利用传感器实时建立环境地图；机器人根据地图和自身的位置信息实现动态路径规划及导航控制；检测、识别并定位环境中的特定目标，动态接近目标物；抓取目标物；语音交互；多目标检测。

本项目目标完成的非功能性需求包括：在 5 秒内对于正确的用户需求做出响应；对于错误的用户需求进行提示并正确处理；若出现错误可在有限时间内恢复；拥有简单的指令格式，易于使用和学习。

此目标检测及抓取型机器人可作为服务机器人在现实生活中有所应用，例如在餐饮业中用于为顾客送餐、在酒店服务中帮助带领顾客到指定房间、在仓库管理中搬运货物等，应用前景广泛。

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（嵌入式方向）课程中护肝队的软件测试说明文档。本文档适用于基于 Ubuntu16.04 系统的 ROS 机器人操作系统开发，适用的硬件平台是启智 ROS 机器人，编写的软件的用途是在该硬件平台上实现一个具有目标检测和抓取功能的机器人，其功能主要包含机器人的主动控制，实时建立环境地图，静态/动态障碍物避障，路径规划和导航控制，单种/多种目标物的检测、识别和抓取，以及语音交互。软件测试

说明包括如下内容：任务概述，测试准备，测试用例，测试结果，测试结果分析。与本文档相关配套的，还有如下文档：SDP 软件开发计划文档、SRS 软件需求规格说明文档、SDD 软件设计说明文档。本文档初次撰写于 2020 年 5 月 22 日。

本项目的开发计划用于总体上指导 ROS 机器人软件项目顺利进行并得到通过最终评审的项目产品。本项目开发计划面向项目组的全体成员，项目周期为 3 个月。

1.3 术语和缩略词

| 缩略词 | 全称 |
|------|---|
| ROS | Robot Operating System/机器人操作系统 |
| URDF | Unified Robot Description Format/统一机器人描述格式 |
| IMU | Inertial Measurement Unit/惯性测量单元 |
| SLAM | Simultaneous Localization and Mapping/即时定位与地图构建 |
| TOF | 立体相机 Time of Flight/飞行时间技术 |

表 1 术语和缩略词表

1.4 引用文档

文档格式要求按照我国 GB8567-2006 计算机软件文档编制规范进行。引用文档包括以下文件：

软件设计说明书 GB8567-2006 (STD)

《启智 ROS 版_开发手册》

2. 任务概述

| 功能性需求 | | |
|-----------|-----|--|
| 名称 | 标号 | 简要描述 |
| 机器人建立环境地图 | 1.1 | 机器人应当能在人的控制下移动并完成对周围环境的建图操作。 |
| | 1.2 | 机器人完成建图后能正确地保存建好的地图。 |
| 机器人的动态避障 | 2 | 机器人在检测到路径中存在障碍物时尝试通过小幅度移动和偏转来绕过障碍物。 |
| 机器人的自主导航 | 3 | 在用户选定目的地后，机器人可基于环境地图进行路径规划，并沿最优运动至目的地。 |
| 机器人检测目标物体 | 4 | 机器人应能检测到并识别出视野中的物体，同时获取到目标物体的位置、形状信息。 |
| 机器人抓取目标物体 | 5 | 机器人应当能基于检测时获取到的目标物体的位置、形状信息，控制机械臂抓取目标物体。 |
| 机器人语音交互 | 6.1 | 机器人能够识别用户的语音指令并执行相应的任务。 |
| | 6.2 | 机器人能够通过语音播报任务的执行情况。 |
| 数据库需求 | 7.1 | 数据库的结构设计和存取过程应当保证数据的完整性和安全性。 |
| | 7.2 | 数据库的存取算法应当保证高效性。 |
| 用户界面需求 | 8.1 | 用户界面设计美观和谐，布局、色彩设计风格统一；操作简洁易懂。 |

| 非功能性需求 | | |
|--------|------|---|
| 名称 | 标号 | 简要描述 |
| 性能需求 | 9.1 | 机器人探测到路径上存在障碍物时，应能在 2s 内响应并尝试避障。 |
| | 9.2 | 语音交互时，应能够在 1s 内实时响应，语音答复或开始执行指令。 |
| | 9.3 | SLAM 建图应包括回环检测，回环检测在召回率为 50%时的准确率应不低于 80%。 |
| | 9.4 | 完成路径规划后，机器人运行的平均移动速度应不低于 0.1m/s。 |
| | 9.5 | 对于系统支持抓取的物品，目标识别准确率应不低于 95%，抓取成功率应不低于 95%。 |
| | 9.6 | 对于系统支持的语音指令，语音识别准确率应不低于 90%。 |
| | 9.7 | 系统功耗应低于 17W。 |
| 易用性需求 | 10 | 系统应在用户首次操作时给出操作指南，使用户经过简单的学习掌握机器人的操作方式。 |
| 可靠性需求 | 11 | 系统在正常使用情况下出现故障的频次应小于 2 次/月。 |
| | | 系统应具备一定的容错能力。在发生故障后，保存当前数据及运行状态并报告错误信息。严重故障时，系统应在重新启动后恢复正常状态。 |
| 可扩展性需求 | 12.1 | 系统应预留一定接口用于未来可能出现的新需求。 |
| | 12.2 | 新增、修改、删除系统功能时不需大幅修改原有代码，其成本应低于重新设计系统的成本。 |

| | | |
|--------|----|--|
| 可维护性需求 | 13 | 系统应支持维护检修，并且能够通过更换配件和修复系统bug的方式解决系统故障。 |
|--------|----|--|

表 2 任务概述表

3. 测试准备

3.1 测试流程

我们整体测试分为四部分：

1. 单元测试：对构成软件的每一个模块进行单元测试。
2. 系统测试：确认测试又称有效性测试。有效性测试是在模拟的环境下，运用黑盒测试的方法，验证被测软件是否满足需求规格说明书列出的需求。任务是验证软件的功能和性能及其他特性是否与用户的要求一致。对软件的功能和性能要求在软件需求规格说明书中已经明确规定，它包含的信息就是软件确认测试的基础。
其中，我们的系统测试包括对于功能性需求的测试和对于非功能性需求的测试。

3.2 测试安排

3.2.1 单元测试

在所有模块上运行黑盒和白盒测试，具体测试细节如下表所示。

| 测试流程 | 涉及功能 | 硬件/软件环境 | 测试人员 | 时间分配 |
|------|------|---------|------|------|
|------|------|---------|------|------|

| | | | | |
|------------|-----------------------|------------------------|-----|------|
| web 端 | 用户界面操作 | PyCharm，两 台电脑 | 曹玥 | 5.23 |
| 主控 | 实现与 web 端交 互以及整体控制 | PyCharm，机 器人模拟器 | 王博文 | 5.23 |
| 移动 | 前后左右四个方 向移动 | 机器人模拟器 及机器人运行 电脑 | 曹玥 | 5.24 |
| 建立环境地 图 | 实现环境地图的 建立 | 机器人模拟器 及机器人运行 电脑 | 王博文 | 5.24 |
| 导航 | 实现避障导航功 能 | 机器人模拟器 及机器人运行 电脑 | 莫策 | 5.24 |
| 目标检测 | 实现目标的正确 检测 | 机器人模拟器 及机器人运行 电脑 | 全庆隆 | 5.27 |
| 语音交互 | 实现通过语音控 制 | 机器人模拟器 及机器人运行 电脑 | 莫策 | 5.27 |

表 3 单元测试任务表

3.2.2 系统测试

| 涉及功能 | 硬件/软件环境 | 测试人员 | 时间分配 |
|------|--------------|-------------------|------|
| 系统测试 | 机器人模拟器以及两台电脑 | 曹博文，曹玥，王波文，全庆隆，莫策 | 6.4 |

表 4 系统测试任务表

4. 测试用例

4.1 单元测试

单元测试用例的选取采用分支覆盖标准，以功能模块为单元，对每个不可再分的函数进行白盒测试，对整个单元进行黑盒测试。

单元测试部分的用例标识的命名规则是：202UT(功能序号)-(用例序号)-(用例类型，黑 B 白 W)。

4.1.1 移动控制

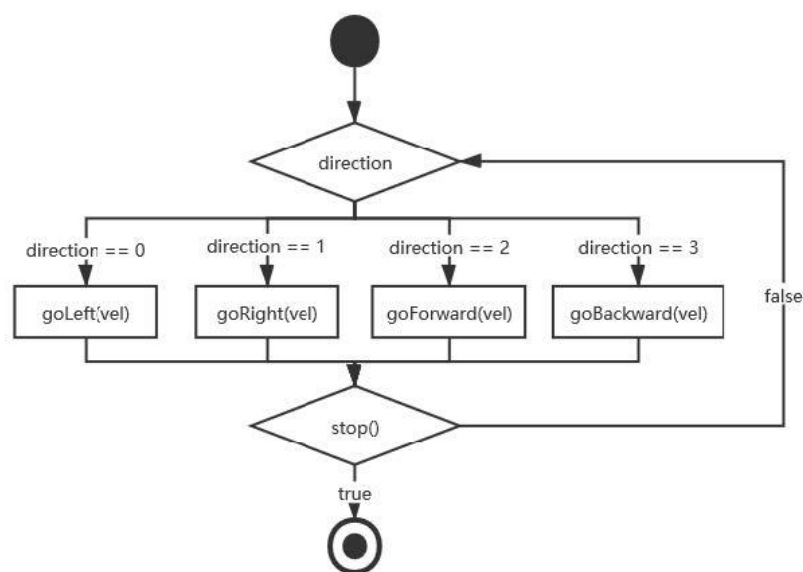


图 1 移动控制详细设计图

✧ 202UT1-1B

测试内容：测试手动键盘控制机器人移动

用例类型：白盒测试（测试各个运动方向和速度的覆盖情况）、黑盒测试（按照指令正确运动并输出速度信息）

初始状态与约束：机器人底盘工作正常

测试过程：键盘输入指令，其中 w/W 代表前进，s/S 代表后退，a/A 代表左平移，d/D 代表右平移，q/Q 代表左转，e/E 代表右转，Space 空格代表停止，Esc 代表退出手动控制状态。使用 python 创建该子程序，向管道内写入指令。

输入：写入指令 w, s, a, d, q, e 各 5 次，space, esc(0x1B)各一次；使用 python 脚本，模拟用户键盘输入，详见 **testmanualcontrol.py**

期望输出：对于 w, s, a, d, q, e 指令，机器人按照指令移动，速度增加到 0.3m/s 后不再增加；对于 space，机器人立即停止运动，各个方向速度降为 0；对于 esc 指令，机器人停止运动，退出手动控制。

评教准则：机器人按照指令运动；控制台输出速度信息，任意一个方向最大速度位 0.3m/s；且最后能正确停止运动并退出。

4.1.2 建立环境地图功能

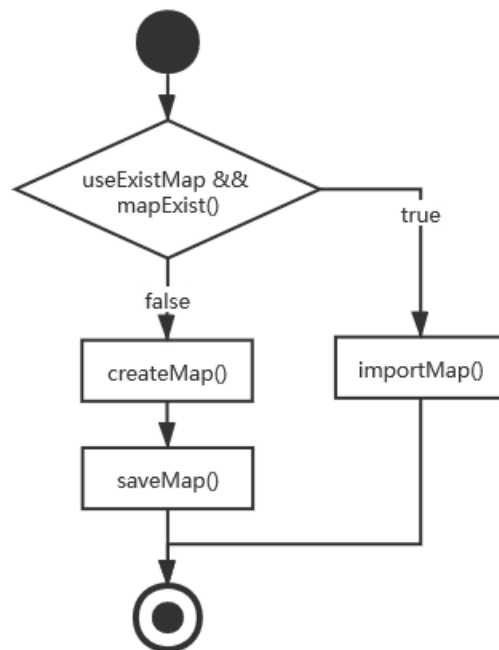


图 2 建立环境地图详细设计图

✧ 202UT2-1B

测试内容：机器人加载环境地图

用例类型：黑盒测试

初始状态与约束：地图已存在

测试过程：输入地图文件路径，机器人加载并显示地图

输入：地图文件路径

期望输出：地图信息

评价准则：地图被成功加载且未被修改

✧ 202UT2-2B

测试内容：机器人建立环境地图

用例类型：白盒测试（判断是否覆盖原地图文件）、黑盒测试（建图功能）

初始状态与约束：激光雷达工作正常

测试过程：机器人接受建图指令，检测目标路径下是否已存在地图文件，若是则询问是否覆盖原有地图文件，否则开始建图；询问是否覆盖后若用户选择是，则开始建图，否则结束进程

输入：建图指令，存储地图文件的路径

期望输出：指定文件夹下的地图文件 map.yaml 与 map.pgm

评价准则：机器人生成正确的地图

4.1.3 导航功能

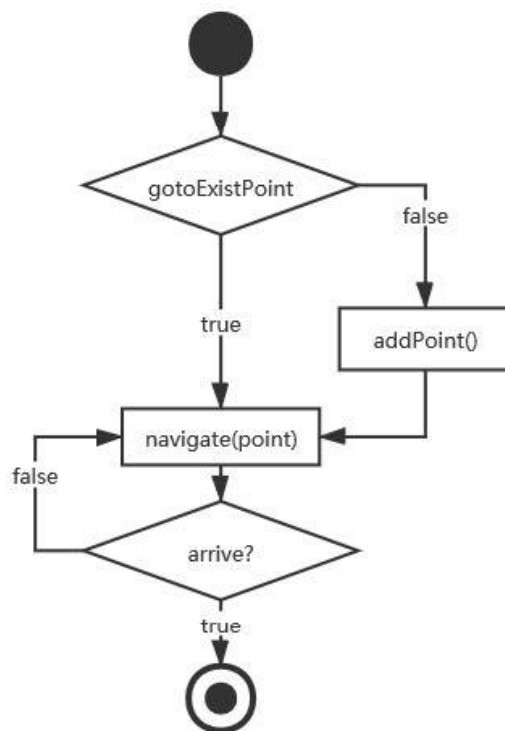


图 3 导航功能详细设计图

✧ 202UT3-1B

测试内容: 添加与保存导航点

用例类型: 黑盒测试（测试添加导航点并保存）

初始状态与约束: 机器人连接正常，roscore 正确启动；

测试过程: 测试人员启动相应程序，随机选择 10 个不同的导航点并保存；查看程序是否正常退出，以及指定路径是否有保存的导航点文件。

输入: 用户标注多个导航点；

期望输出: 导航点文件 waypoints.xml；

评教准则: 用户标注的导航点与文件中保存的一致。

✧ 202UT3-2B

测试内容: 导航节点算法

用例类型: 黑盒测试（机器人正确规划路径并达到设定位置）

初始状态与约束: 机器人连接正常，roscore 正确启动，建立完全的地图，导航

点文件；

测试过程：用户在仿真环境中，输入导航点文件中的不同导航点，开始导航，观察机器人状态。

输入：用户选择的导航点

期望输出：开始导航后，图中出现规划好的紫色路线。机器人沿紫色路径行进，到达终点时，机器人输出到达目标点语句。

评教准则：机器人输出到达目标点语句。

◇ 202UT3-3B

测试内容：测试导航过程中的避障功能；

用例类型：黑盒测试（机器人导航的避障功能）；

初始状态与约束：机器人连接正常，导航模块已启动，避障传感器工作正常。

测试过程：用户在仿真环境中，为机器人设置不同类型的目标地点，开始导航，在途中添加不同类型的障碍物。

输入：用户选择的导航点，障碍物

期望输出：开始导航后，图中出现规划好的紫色路线。机器人沿紫色路径行进，遇到障碍物时机器人停下，等待障碍物移开，继续运动。

评教准则：

对于在用户选择的点，机器人都能正确规划路径途径各个设定的航点并到达，遇到障碍物时停下，等待障碍物移开，继续运动。

4.1.4 目标检测

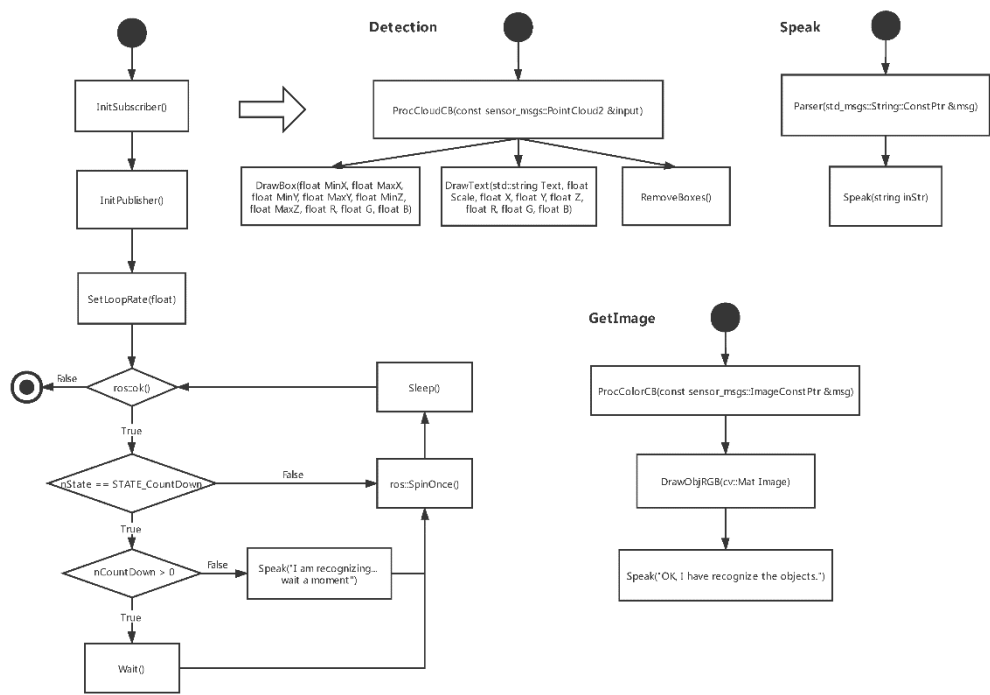


图 4 目标检测功能详细设计图

✧ 202UT4-1W

测试内容：机器人目标识别

用例类型：白盒测试、黑盒测试

初始状态与约束：机器人正在执行指令中

测试过程：在机器人执行建图、导航目标识别等指令时，向机器人发出目标识别指令

输入：目标识别指令

期望输出：机器人正忙

评价准则：机器人执行流程及执行结果正确

✧ 202UT4-2W

测试内容：机器人目标识别

用例类型：白盒测试、黑盒测试

初始状态与约束：机器人处于静止待命状态，机器人视野内无物体

测试过程：在 gazebo 仿真环境中将机器人停在空旷处，向机器人发出目标识别指令

输入：目标识别指令

期望输出：未检测到物体

评价准则：机器人执行流程及执行结果正确

✧ 202UT4-3W

测试内容：机器人目标识别

用例类型：白盒测试、黑盒测试

初始状态与约束：机器人处于静止待命状态，机器人视野内有物体

测试过程：在 gazebo 仿真环境中放置物品，物品距离机器人不远且与 kinect 相机高度相近，向机器人发出目标识别指令

输入：目标识别指令

期望输出：机器人检测到物品，返回物体位置

评价准则：机器人执行流程及执行结果正确

4.1.5 语音交互功能

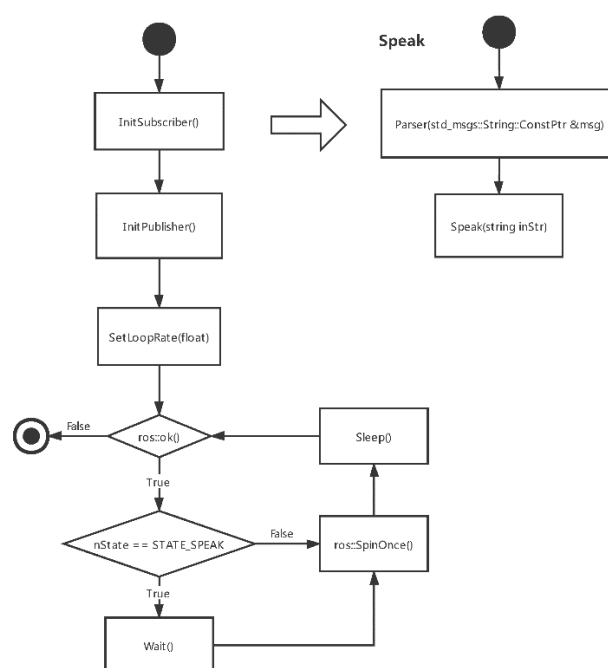


图5 语音交互详细设计图

✧ 202UT5-1B

测试内容：识别语音指令

用例类型：白盒测试（测试识别流程）、黑盒测试（正别识别语音指令）

初始状态与约束：网络连接畅通，麦克风工作正常

测试过程：测试人员对麦克风发出语音指令，而后观察是否识别过程满足流程——①音频被存至本地②调用科大讯飞 API 检测③返回结果并删除该音频文件

输入：语音指令（包括“前进”、“后退”、“左转”、“右转”、“建立地图”、“导航”、“抓取”、“检测”）

期望输出：语音指令的识别结果

评价准则：识别结果与测试人员说出的内容一致

✧ 202UT5-2B

测试内容：语音播报功能

用例类型：黑盒测试

初始状态与约束：扬声器工作正常

测试过程：测试人员分别输入开始移动、停止移动、开始建图、建图完成、开始导航、到达目的地、发现障碍物、开始检测目标、检测目标完成、开始抓取、抓取成功、发生故障的状态更新指令，机器人通过扬声器播报状态

输入：状态更新指令

期望输出：机器人进行语音播报

评价准则：语音播报的内容与测试人员输入的状态一致

4.1.6 主控模块

✧ 202UT6-1W

测试内容：control_shell()函数，主控程序

用例类型：白盒测试，黑盒测试

初始状态与约束：ros 环境已打开

测试过程：测试人员分别输入代表不同命令的字符串，观察是否运行相应命令行。

输入：代表动作的字符串。

期望输出：根据字符串的不同，命令行上显示不同程序的运行信息。

评价准则：可以运行相应命令行，无报错信息。

4.1.7 web 端

4.1.7.1 用户数据库测试(dbmessage.py; login.html; register.html)

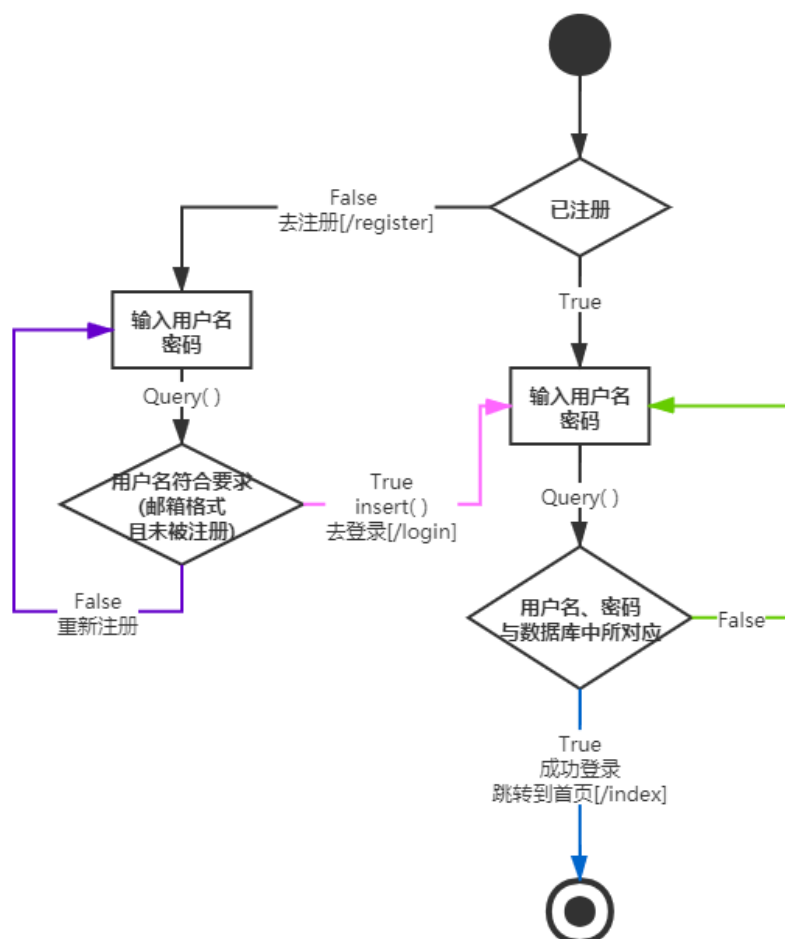


图 6 用户数据库详细设计图

✧ 202UT71-1

测试内容: 注册新用户（注册项不符合格式要求）

用例类型: 黑盒测试；白盒测试

初始状态与约束: 已进入注册页面

测试过程: 注册时输入的用户名不符合邮箱格式，或用户名及密码字段有空缺，或用户名已被注册；点击注册按钮，观察界面状态

输入: 不符合格式的注册项

期望输出: 提醒用户重新输入符合格式的注册项

评价准则: 注册失败且提醒用户不符合格式的原因

✧ 202UT71-2

测试内容： 注册新用户（注册项符合格式要求）

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入注册页面

测试过程： 注册时输入的用户名符合邮箱格式，且用户名及密码字段无空缺；
点击注册按钮，观察界面状态

输入： 符合格式的注册项

期望输出： 注册成功

评价准则： 注册成功且跳转到登录页面

✧ 202UT71-3

测试内容： 已注册用户登录（个人信息与数据库中数据不相符）

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入登录页面

测试过程： 登录时输入的用户名及密码与数据库中数据不相符；
点击登录按钮，观察界面状态

输入： 与数据库中数据不相符的个人信息（用户名及密码）

期望输出： 提醒用户用户名或密码不正确

评价准则： 登陆失败并提醒用户用户名或密码不正确

✧ 202UT71-4

测试内容： 已注册用户登录（个人信息与数据库中数据相符）

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入注册页面

测试过程： 登录时输入的用户名及密码与数据库中数据相符；
点击登录按钮，观察界面状态

输入： 与数据库中数据相符的个人信息（用户名及密码）

期望输出： 登录成功

评价准则： 登录成功且跳转到首页

4.1.7.2 基本移动模式测试（index.html）

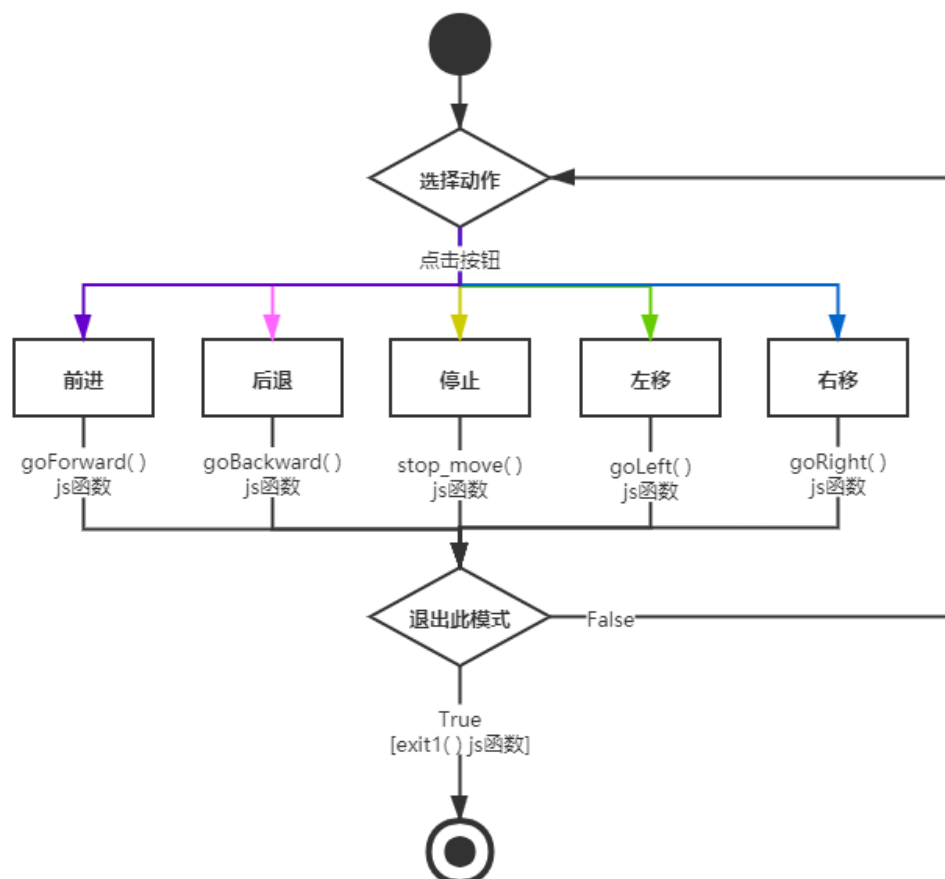


图 7 基本移动页面详细设计图

✧ 202UT72-1

测试内容： 点击“前进”按钮，触发 goForward() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“基本移动模式”弹窗

测试过程： 点击“前进”按钮，观察 Terminal 中的输出

输入： 点击“前进”按钮

期望输出： GET/index?pattern=1&action=forward HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

✧ 202UT72-2

测试内容： 点击“后退”按钮，触发 goBackward() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“基本移动模式”弹窗

测试过程： 点击“后退”按钮，观察 Terminal 中的输出

输入： 点击“后退”按钮

期望输出： GET/index?pattern=1&action=backward HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

✧ 202UT72-3

测试内容： 点击“停止”按钮，触发 stop_move() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“基本移动模式”弹窗

测试过程： 点击“停止”按钮，观察 Terminal 中的输出

输入： 点击“停止”按钮

期望输出： GET/index?pattern=1&action=stop HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

✧ 202UT72-4

测试内容： 点击“左移”按钮，触发 goLeft() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“基本移动模式”弹窗

测试过程： 点击“左移”按钮，观察 Terminal 中的输出

输入： 点击“左移”按钮

期望输出： GET/index?pattern=1&action=left HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

✧ 202UT72-5

测试内容： 点击“右移”按钮，触发 goRight() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“基本移动模式”弹窗

测试过程： 点击“右移”按钮，观察 Terminal 中的输出

输入： 点击“右移”按钮

期望输出： GET/index?pattern=1&action=right HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

✧ 202UT72-6

测试内容： 点击“Exit”按钮，触发 exit1() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“基本移动模式”弹窗

测试过程： 点击“Exit”按钮，观察 Terminal 中的输出

输入： 点击“Exit”按钮

期望输出： GET/index?pattern=1&action=exit HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

4.1.7.3 建立环境地图测试（index.html）

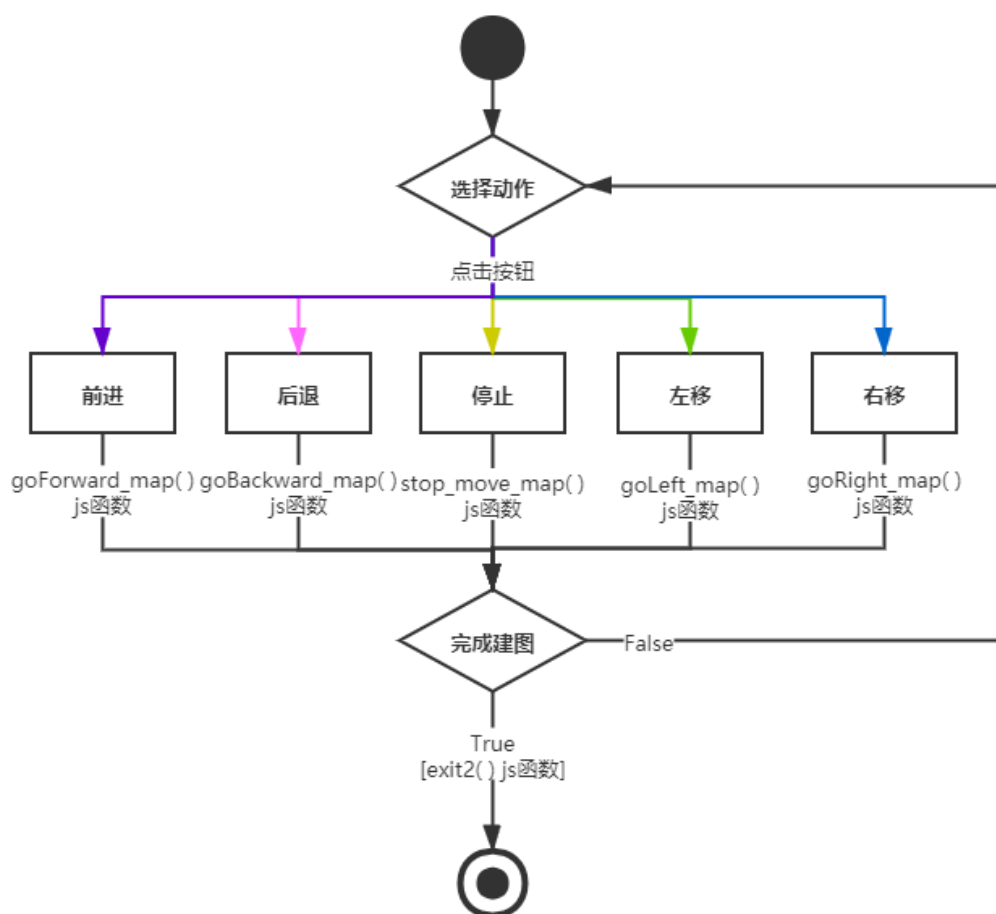


图 8 建立环境地图页面详细设计图

✧ 202UT73-1

测试内容： 点击“前进”按钮，触发 goForward_map() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“建立环境地图”弹窗

测试过程： 点击“前进”按钮，观察 Terminal 中的输出

输入： 点击“前进”按钮

期望输出： GET/index?pattern=2&action=forward HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

✧ 202UT73-2

测试内容： 点击“后退”按钮，触发 goBackward_map() js 函数

用例类型：黑盒测试；白盒测试

初始状态与约束：已进入首页中“建立环境地图”弹窗

测试过程：点击“后退”按钮，观察 Terminal 中的输出

输入：点击“后退”按钮

期望输出：GET/index?pattern=2&action=backward HTTP/1.1?

评价准则：无报错信息且 Terminal 中有正确输出

✧ 202UT73-3

测试内容：点击“停止”按钮，触发 stop_move_map() js 函数

用例类型：黑盒测试；白盒测试

初始状态与约束：已进入首页中“建立环境地图”弹窗

测试过程：点击“停止”按钮，观察 Terminal 中的输出

输入：点击“停止”按钮

期望输出：GET/index?pattern=2&action=stop HTTP/1.1?

评价准则：无报错信息且 Terminal 中有正确输出

✧ 202UT73-4

测试内容：点击“左移”按钮，触发 goLeft_map() js 函数

用例类型：黑盒测试；白盒测试

初始状态与约束：已进入首页中“建立环境地图”弹窗

测试过程：点击“左移”按钮，观察 Terminal 中的输出

输入：点击“左移”按钮

期望输出：GET/index?pattern=2&action=left HTTP/1.1?

评价准则：无报错信息且 Terminal 中有正确输出

✧ 202UT73-5

测试内容：点击“右移”按钮，触发 goRight_map() js 函数

用例类型：黑盒测试；白盒测试

初始状态与约束： 已进入首页中“建立环境地图”弹窗

测试过程： 点击“右移”按钮，观察 Terminal 中的输出

输入： 点击“右移”按钮

期望输出： GET/index?pattern=2&action=right HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

✧ 202UT73-6

测试内容： 点击“保存地图”按钮，触发 exit2()js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“建立环境地图”弹窗

测试过程： 点击“保存地图”按钮，观察 Terminal 中的输出

输入： 点击“保存地图”按钮

期望输出： GET/index?pattern=2&action=exit HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

4.1.7.4 导航模式测试 (index.html)

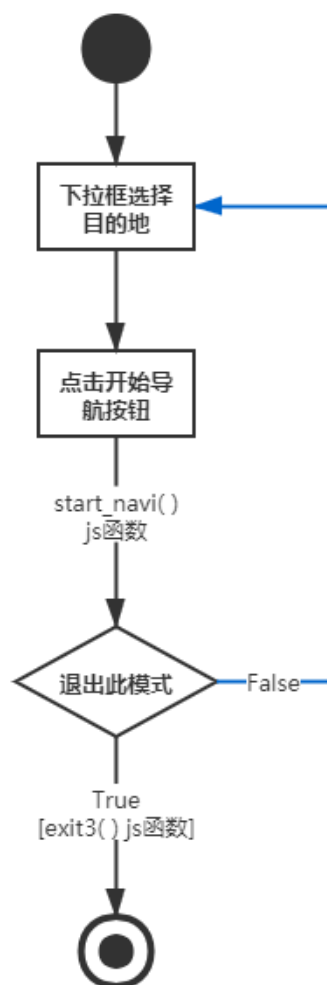


图 9 导航页面详细设计图

✧ 202UT74-1

测试内容: 在下拉框中选择目的地, 点击“开始导航”按钮, 触发 `start_navi()` js 函数

用例类型: 黑盒测试; 白盒测试

初始状态与约束: 已进入首页中“导航模式”弹窗

测试过程: 在下拉框中选择目的地, 点击“开始导航”按钮, 观察 Terminal 中的输出

输入: 在下拉框中选择目的地, 点击“开始导航”按钮

期望输出: `GET/index?pattern=3&action=start&place=xxx HTTP/1.1?`

评价准则： 无报错信息且 Terminal 中有正确输出，place 为所选择的目的地

✧ 202UT74-2

测试内容： 点击“Exit”按钮，触发 exit3() js 函数

用例类型： 黑盒测试；白盒测试

初始状态与约束： 已进入首页中“导航模式”弹窗

测试过程： 点击“Exit”按钮，观察 Terminal 中的输出

输入： 点击“Exit”按钮

期望输出： GET/index?pattern=3&action=exit HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出

4.1.7.5 抓取模式测试（index.html）

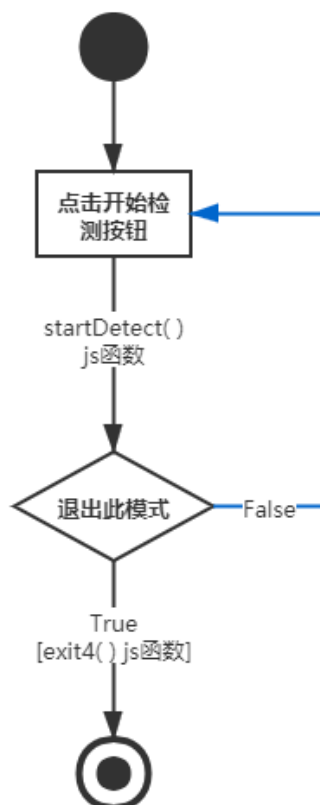


图 10 抓取页面详细设计图

✧ 202UT75-1

测试内容： 点击“开始检测”按钮，触发 startDetect() js 函数

用例类型：黑盒测试；白盒测试

初始状态与约束：已进入首页中“抓取模式”弹窗

测试过程：点击“开始检测”按钮，观察 Terminal 中的输出

输入：点击“开始检测”按钮

期望输出：GET/index?pattern=4&action=start HTTP/1.1?

评价准则：无报错信息且 Terminal 中有正确输出

✧ 202UT75-2

测试内容：点击“Exit”按钮，触发 exit4() js 函数

用例类型：黑盒测试；白盒测试

初始状态与约束：已进入首页中“抓取模式”弹窗

测试过程：点击“Exit”按钮，观察 Terminal 中的输出

输入：点击“Exit”按钮

期望输出：GET/index?pattern=4&action=exit HTTP/1.1?

评价准则：无报错信息且 Terminal 中有正确输出

4.1.7.6 页面跳转测试（app.py; help.html; about_us.html; index.html）

✧ 202UT76-1

测试内容：首页、“帮助文档”页面、“关于我们”页面间的跳转

用例类型：黑盒测试

初始状态与约束：已进入首页

测试过程：点击首页左侧工具栏中相应界面的按钮，观察 Terminal 中的输出和页面的跳转情况

输入：点击首页左侧工具栏中相应界面的按钮

期望输出：GET /index?name=xxx HTTP/1.1?

GET /help HTTP/1.1?

GET /about_us HTTP/1.1?

评价准则： 无报错信息且 Terminal 中有正确输出，页面跳转状态正常

4.1.8 目标抓取

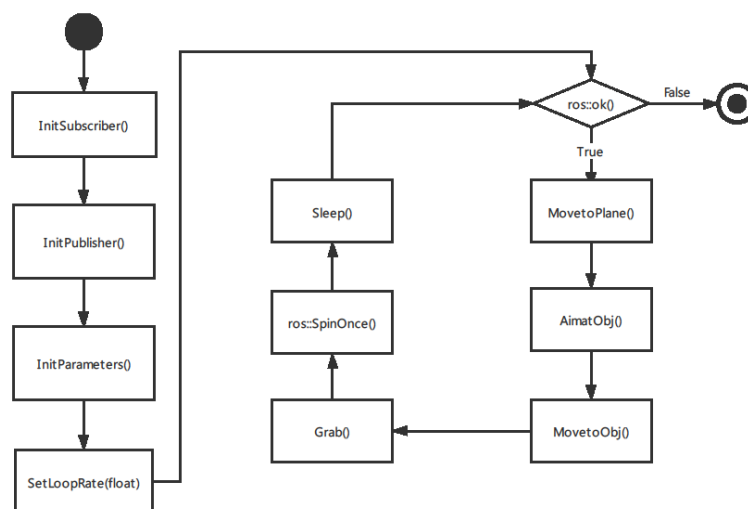


图 11 目标抓取详细设计图

✧ 202UT8-1

测试内容： 订阅 ProcCloudCB(点云分割)节点和 PoseDiff（测距）节点

用例类型： 白盒测试

初始状态与约束： 软件运行正常

测试过程： 在 InitSubscriber（新建订阅节点）函数中加入输出的调试信息。进入抓取功能，观察控制台输出的信息

输入： 在机器人面前设置一个桌子，并在桌子上放置多个啤酒瓶；运行 grab_action.launch 进入抓取功能

期望输出： 在控制台有“subscribe ProcCloudCB”输出；在控制台输出机器人和桌子的距离

评价准则： 机器人成功订阅点云分割节点，并正确地判断自己和桌子的位置（使用 Rviz 界面中的 measure 功能进行判断）

✧ 202UT8-2

测试内容： 发布点云分割结果和移动控制信息

用例类型： 白盒测试

初始状态与约束： 软件运行正常

测试过程：在 `InitPublisher`（新建发布节点）函数中加入输出的调试信息。进入抓取功能，观察控制台输出的信息

输入：在机器人面前设置一个桌子，并在桌子上放置多个啤酒瓶；运行 `grab_action.launch` 进入抓取功能

期望输出：在 `Rviz` 界面中看到点云分割结果；在控制台输出移动控制信息

评价准则：机器人正确地对点云数据进行分割，并基于对自己和桌子间距离的判断发布移动控制信息，控制自己接近桌子

✧ 202UT8-3

测试内容：机器人对准目标物体

用例类型：白盒测试（`AimatObj` 函数和 `MovetoObj` 函数）、黑盒测试（机器人有对准物体、接近物体的行为）

初始状态与约束：软件运行正常，且 `InitSubscriber` 函数和 `InitPublisher` 函数正确

测试过程：在 `AimatObj` 函数和 `MovetoObj` 函数中加入输出的调试信息。进入抓取功能，观察控制台输出的信息

输入：在机器人面前设置一个桌子，并在桌子上放置多个啤酒瓶；运行 `grab_action.launch` 进入抓取功能

期望输出：在控制台输出移动控制信息，控制机器人首先左右移动瞄准物体、而后控制机器人前后移动靠近物体；在 `Rviz` 界面中看到机器人表现出对准物体、接近物体的行为

评价准则：机器人基于对物体的点云分割结果，控制自己瞄准并接近目标物体

✧ 202UT8-4

测试内容：机器人的抓取功能

用例类型：白盒测试（机械臂、机械手控制信息）、黑盒测试（机器人抓取行为）

初始状态与约束：软件运行正常，`InitSubscriber` 函数和 `InitPublisher` 函数正确，且机器人已经接近目标物体

测试过程：在 `Grab` 函数中加入输出的调试信息。进入抓取功能，在机器人接近目标物体之后观察控制台输出的信息

输入：在机器人面前设置一个桌子，并在桌子上放置多个啤酒瓶；运行 `grab_action.launch` 进入抓取功能

期望输出：在控制台输出控制机械臂升降、机械臂开合的信息，控制机器人抓取物体；在 Rviz 界面中看到机器人有抓取物体的行为

评价准则：机器人基于对物体的点云分割结果，控制机械臂抓取物体

4.2 系统测试

由于系统测试的主要目的是确认整个机器人系统满足设计需求，我们根据需求文档中的需求说明设计了基于黑盒测试的测试用例。

总体用例包括测试需求说明文档中的功能性需求和非功能需求。其中功能性需求包括：

- 移动控制：用户可以控制机器人的前、后、左转、右转移动；
- 建立环境地图：用户建立根据环境的改变环境地图；
- 导航：用户可以让机器人去往指定的预先标定好的导航点，机器人导航过程中实现避障；
- 目标检测：机器人可以对于物体进行检测；

参与测试的非功能性需求包括：

- 性能需求
- 易用性需求
- 可靠性需求

我们的用例设计整体上依照 SRS 文档中的需求图，其中，系统流程图为：

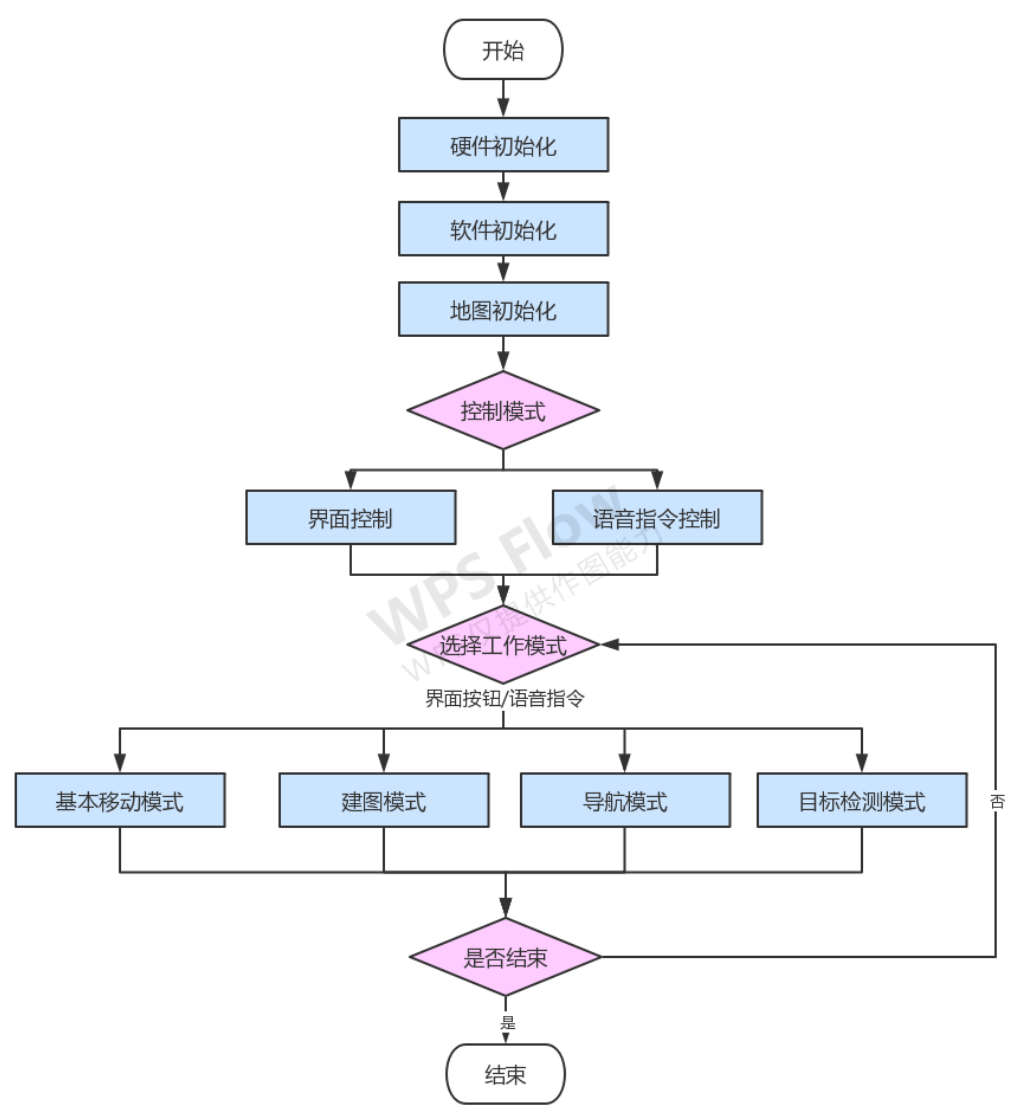


图 12 总体业务需求流程图

系统测试部分的用例标识的命名规则是: 202ST(功能序号)-(用例序号)。

4.2.1 移动控制

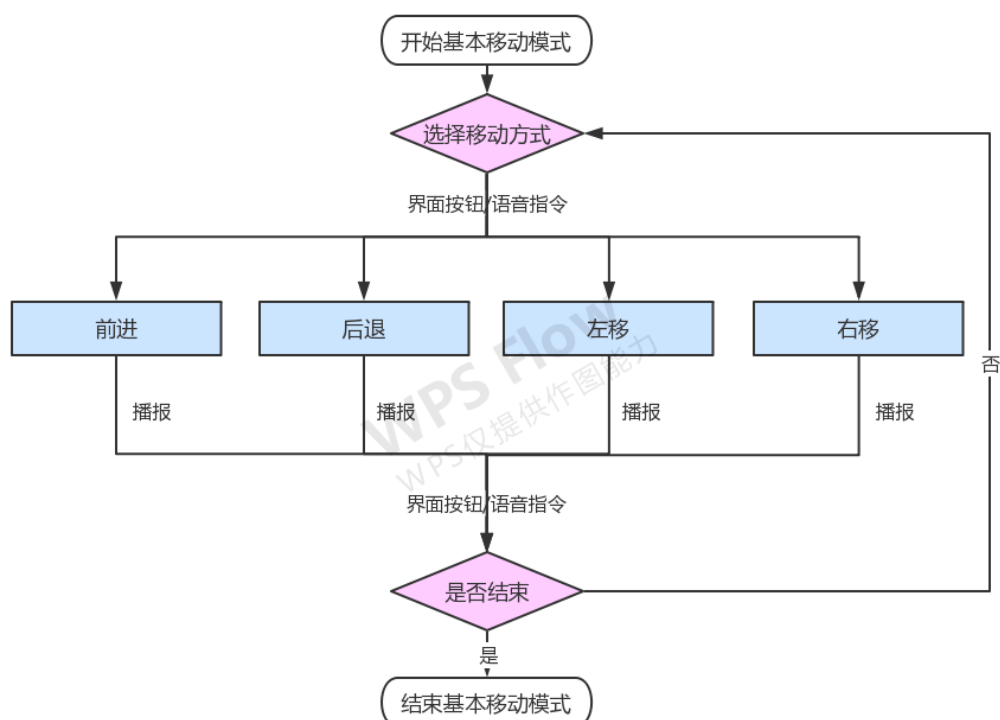


图 13 移动控制需求流程图

✧ 202ST1-1

测试内容: web 前端与后端移动控制整体的连接与控制的正确性

用例类型: 黑盒测试（系统测试）

初始状态和约束: web 端已与后端主控正常连接通信，软件和机器人硬件工作正常

测试过程: 用户登录 web 端，点击移动模式，进入机器人控制界面，点击按钮，观察机器人运动状态，和控制电脑的终端输出信息

输入: 用户在 web 点击控制方向的 7 个按钮，分为前进、后退、左平移、右平移、左转、右转和停止各 5 次

期望输出: 机器人按照方向和指定速度移动，终端输出速度信息；

评价准则: 机器人各个方向运动速度不超过 0.3m/s；用户退出移动界面，机器人停止运动，终端输出 EXIT 信息；**非功能性需求:** ①性能需求——用户发出指令后，机器人在 1s 内响应并开始移动；②易用性需求——web 端给出操作指南，

按键简单易用③可靠性需求——移动过程中出现故障的频次小于 2 次/月；若非严重故障，系统在重启后应恢复正常状态

✧ 202ST1-2

测试内容：控制器处理网络连接中断；

用例类型：黑盒测试（系统测试）

初始状态和约束：web 端已与后端主控正常连接通信，软件和机器人硬件工作正常；

测试过程：机器人开始运动，然后断开网络连接；

输入：手动断开 web 与机器人的网络连接；

期望输出：机器人停止运动；

评价准则：网络连接中断后，主控向机器人发送停止运动的指令；机器人各个方向速度变为 0，并退出运动状态，等待重新连接；

4.2.2 建立环境地图

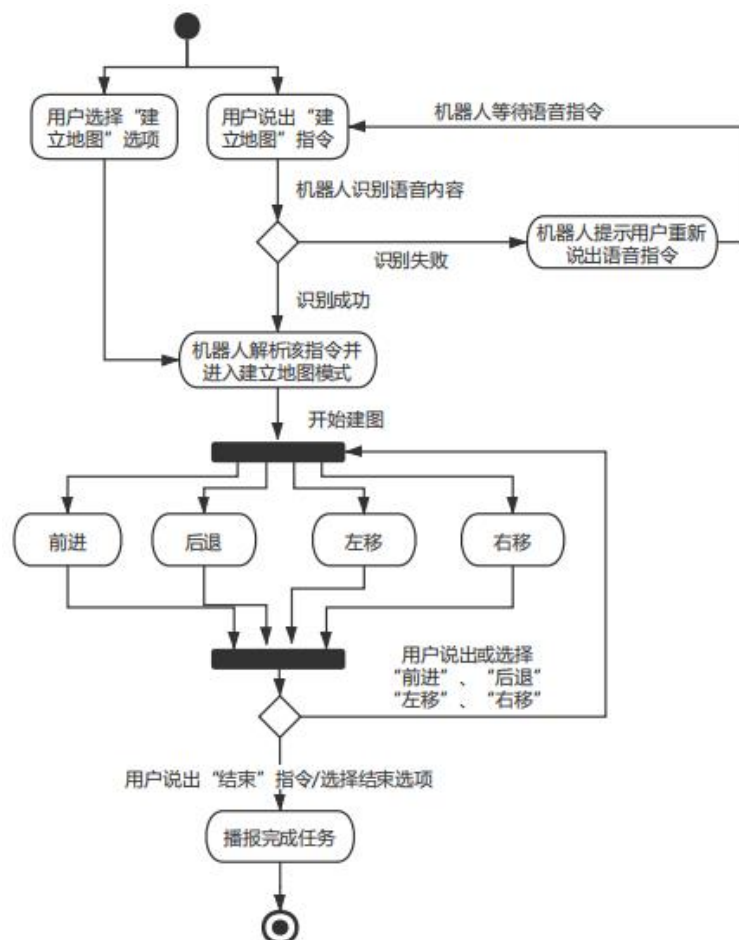


图 14 建立环境地图需求流程图

◇ 202ST2-1

测试内容：web 前端与用户建立（更新）地图功能的连接与控制的正确性

用例类型：黑盒测试

初始状态与约束：软件运行正常

测试过程：测试人员在 web 端选择“建立地图”选项，web 端向机器人发送建图指令；而后测试人员在 web 端通过“前进”、“后退”、“左移”、“右移”键控制机器人绕场移动、建立地图；

输入：在 web 端点击“建立地图”选项

期望输出：机器人完成建图并在 web 端显示反馈信息

评价准则：①性能需求——用户发出指令后，机器人在 1s 内响应并开始建图；地图准确率不低于 80%；识别“建立地图”指令的准确率不低于 90%②易用性

需求——web 端给出操作指南③可靠性需求——建图过程中出现故障的频次小于 2 次/月；若非严重故障，系统在重启后应恢复正常状态

4.2.3 导航

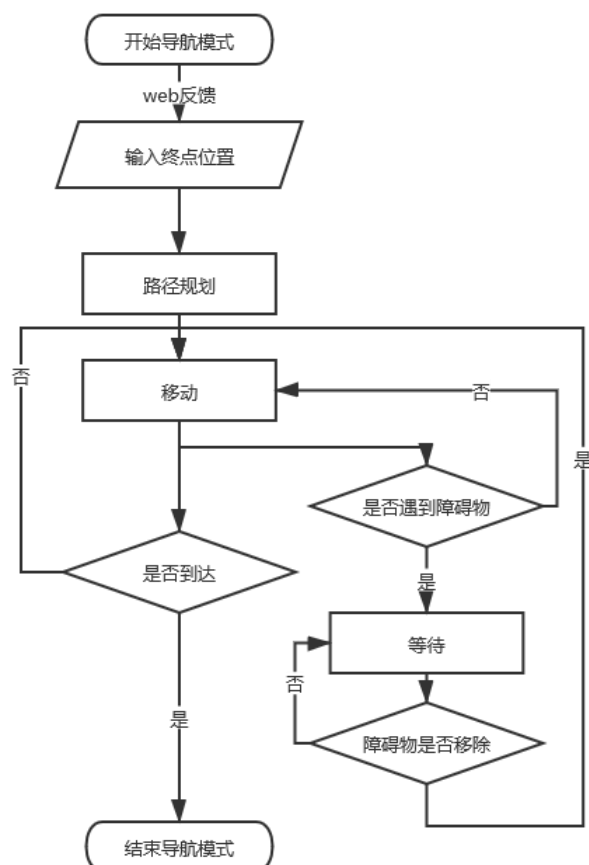


图 15 导航功能需求流程图

◇ 202ST3-1

测试内容：web 前端与导航功能连接与控制的正确性

用例类型：黑盒测试（系统测试）

初始状态和约束：web 端已与后端主控正常连接通信，软件和机器人硬件工作正常，拥有建立完全的地图，导航点文件；

测试过程：用户登录 web 端，点击导航模式，选择前往的点，开始导航，web 端反馈状态，观察机器人的运动状态。

输入：用户在 web 端选择前往的导航点名称

期望输出：开始导航后，web 端反馈状态，图中出现规划好的紫色路线。机器人

沿紫色路径行进，到达终点时，终端输出到达目标点语句

评价准则：机器人显示的路径是正确的路径，机器人可以沿紫色路径行进，到达终点时，终端输出到达目标点语句。非功能性需求：①性能需求——用户发出指令后，机器人在 1s 内响应并在 3 秒内开始导航；导航地点准确；②易用性需求——web 端给出操作指南③可靠性需求——导航过程中出现故障的频次小于 2 次/月；若非严重故障，系统在重启后应恢复正常状态

4.2.4 目标检测

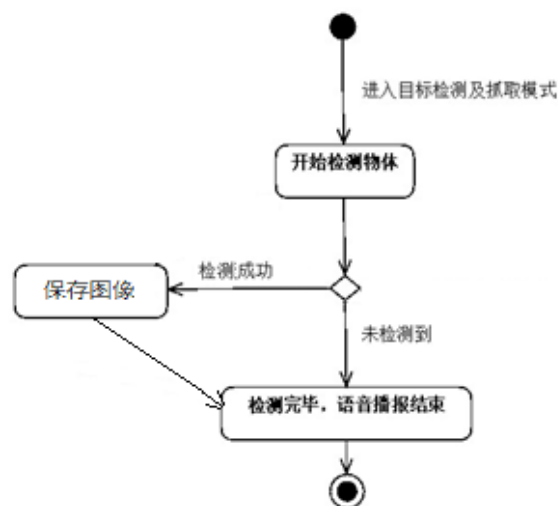


图 16 目标检测需求流程图

✧ 202ST4-1

测试内容：web 前端与目标检测功能连接与控制的正确性

用例类型：黑盒测试（系统测试）

初始状态和约束：web 端已与后端主控正常连接通信，软件和机器人硬件工作正常

测试过程：用户登录 web 端，进行目标检测，观察机器人的反馈结果

输入：gazebo 仿真世界中放置能被机器人检测到的物品

期望输出：机器人反馈检测到物品，并返回物体位置

评价准则：机器人正确检测出有物品存在，并返回物体位置

4.2.5 语音交互

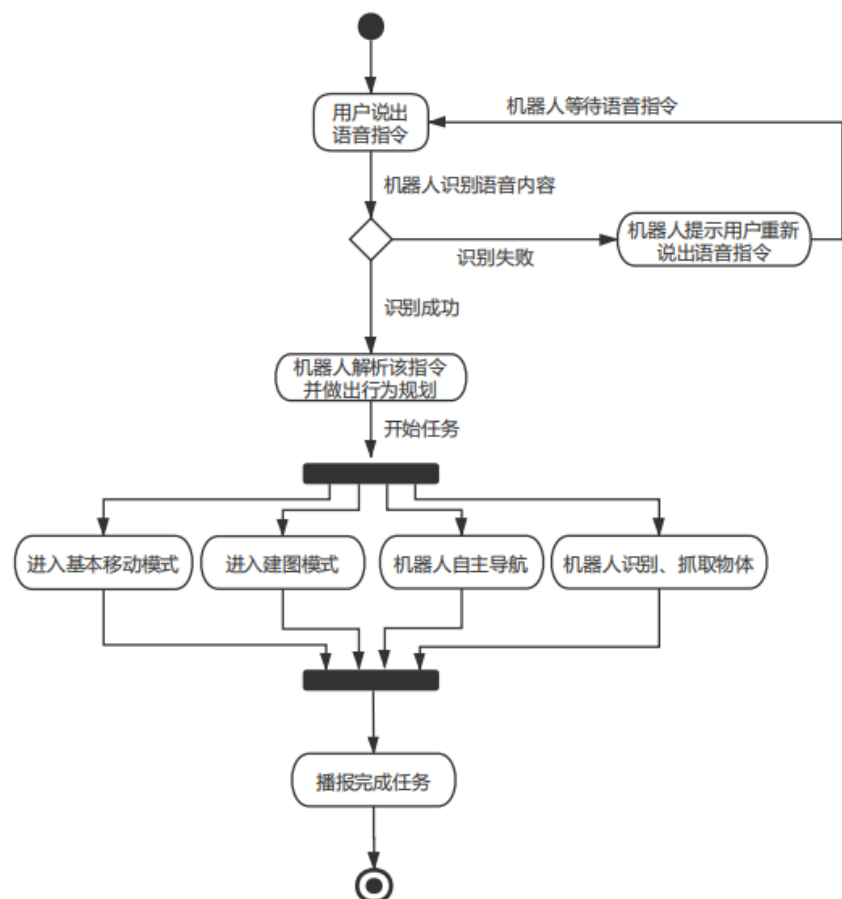


图 17 语音交互需求流程图

✧ 202ST5-1

测试内容：测试用户与机器人的语音交互功能

用例类型：黑盒测试

初始状态与约束：软件运行正常

测试过程：本功能将在测试其他用例过程中同步进行测试

评价准则：对于系统支持的语音指令，语音识别准确率应不低于 90%；语音播报声音洪亮、吐字清晰；语音播报内容与机器人所处状态一致

4.2.6 目标抓取

✧ 202ST6-1

测试内容：web 前端与目标抓取功能连接与控制的正确性

用例类型：黑盒测试

初始状态和约束：web 端已与后端主控正常连接通信，软件和机器人硬件工作正常

测试过程：用户登录 web 端，进行目标抓取，观察机器人的反馈结果

输入：在 gazebo 仿真世界中放置一张桌子，并在桌子上放置多个啤酒

期望输出：机器人反馈物品的点云数据，动态地接近物品并表现出抓取行为

评价准则：机器人正确检测到目标物体，并表现出抓取行为

5. 测试结果

5.1 单元测试

5.1.1 移动模式

✧ 202UT1-1B

测试内容：测试手动键盘控制机器人移动

实际输出：机器人按照指令正确移动，终端输出相应的速度和状态信息；

完成状态：优

发现问题：开始时未设定运行的最大速度，后来补充上，保证机器人运行的安全性；机器人停止没有逐渐减速的过程，实际运行中可能由于惯性造成隐患。控制台对输入字符有回显，影响输出效果，后来关闭回显，输出更加简洁明了。

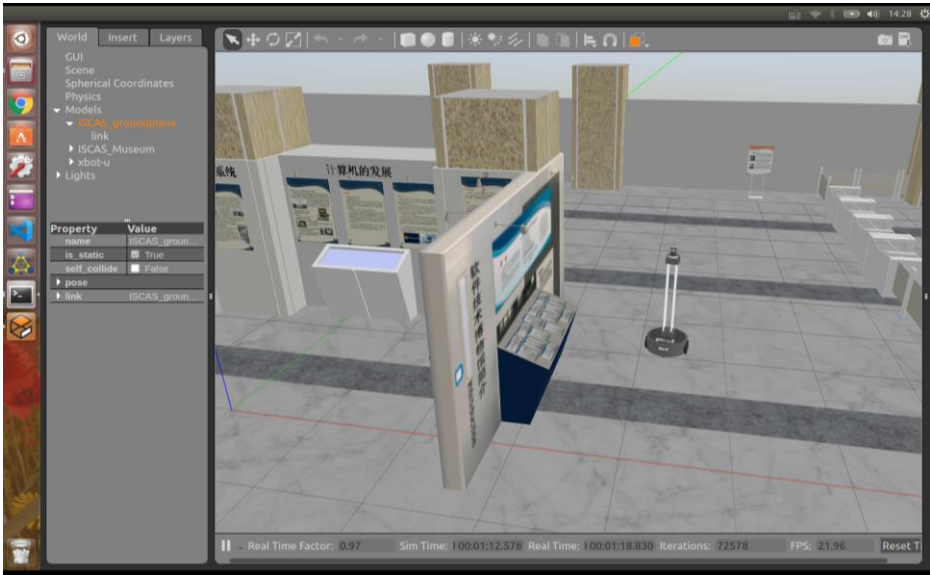


图 18 移动控制单元测试截图 1

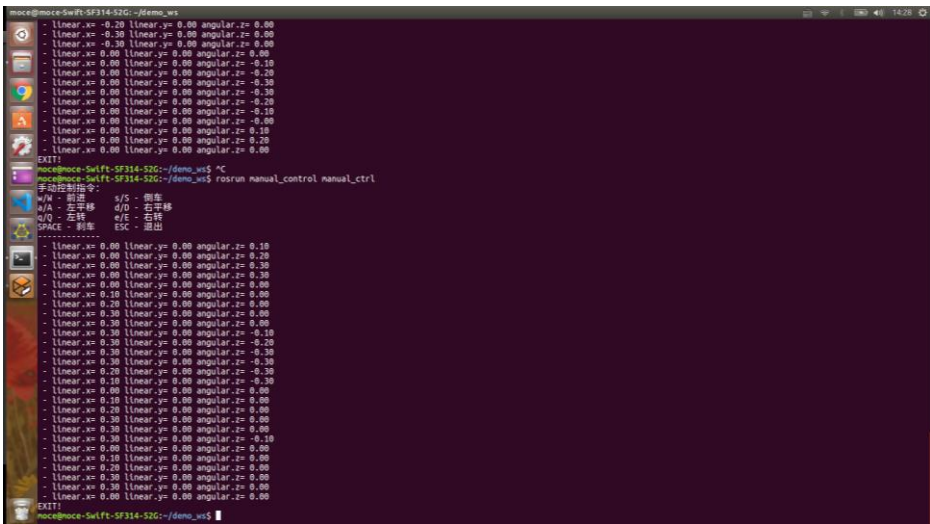


图 19 移动控制单元测试截图 2

5.1.2 建立环境地图功能

◇ 202UT2-1B

测试内容：机器人加载环境地图

实际输出：成功加载地图

完成状态：优

发现问题：无

◇ 202UT2-2B

测试内容：机器人建立环境地图

实际输出：指定路径下生成地图文件 `map.yaml` 与 `map.pgm`

完成状态：优

发现问题：虽然地图完整度较高，但测试发现激光雷达只能扫描到特定高度的物体——①面前出现救护车时，激光雷达只扫描到轮胎，而未发现车辆底盘以上的部分②面前出现可乐瓶时，激光雷达并未发现

解决方案：下移激光雷达所在位置，使其可以检测到小物体，并在机器人中段加设一个激光雷达，使其可以完整地检测到车辆等存在部分结构悬空的物体。

测试过程截图：



图 20 建立环境地图单元测试截图 1

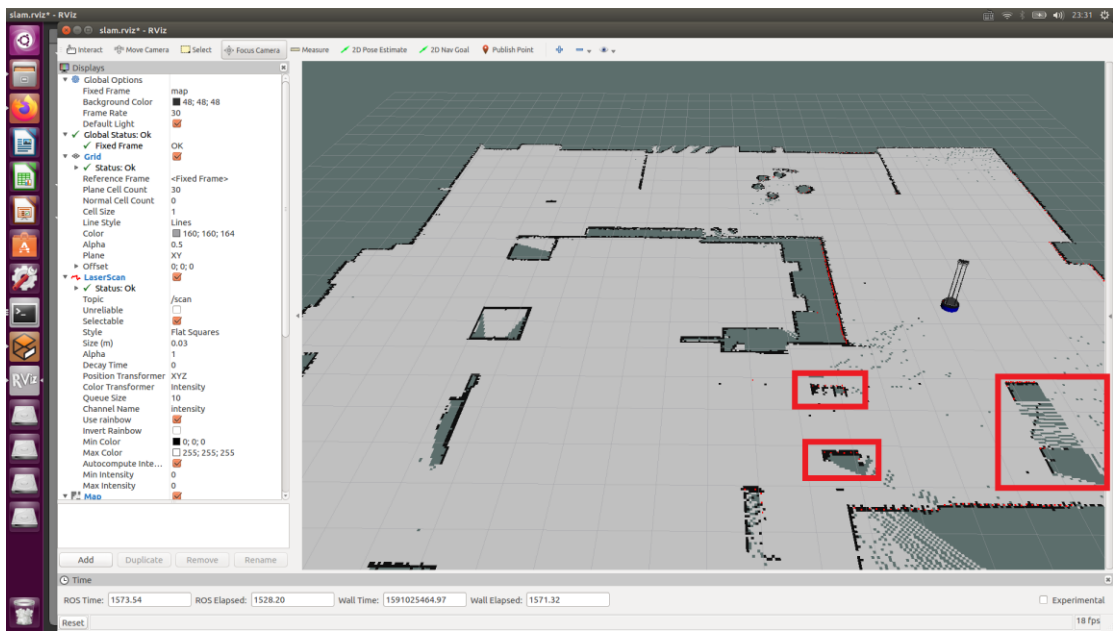


图 21 建立环境地图单元测试截图 2

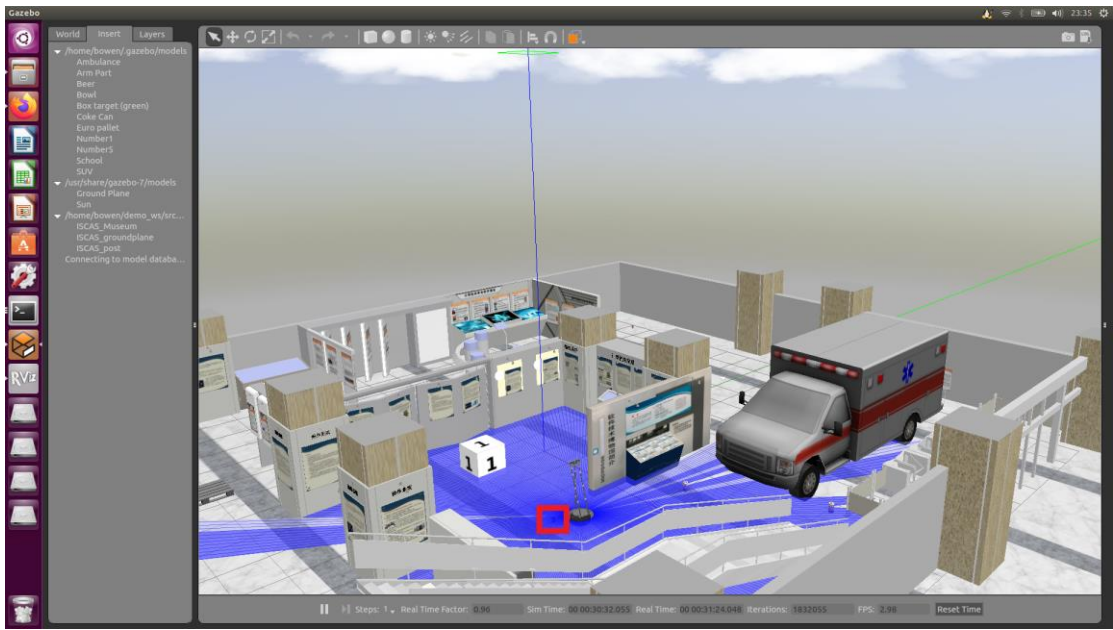


图 22 建立环境地图单元测试截图 3

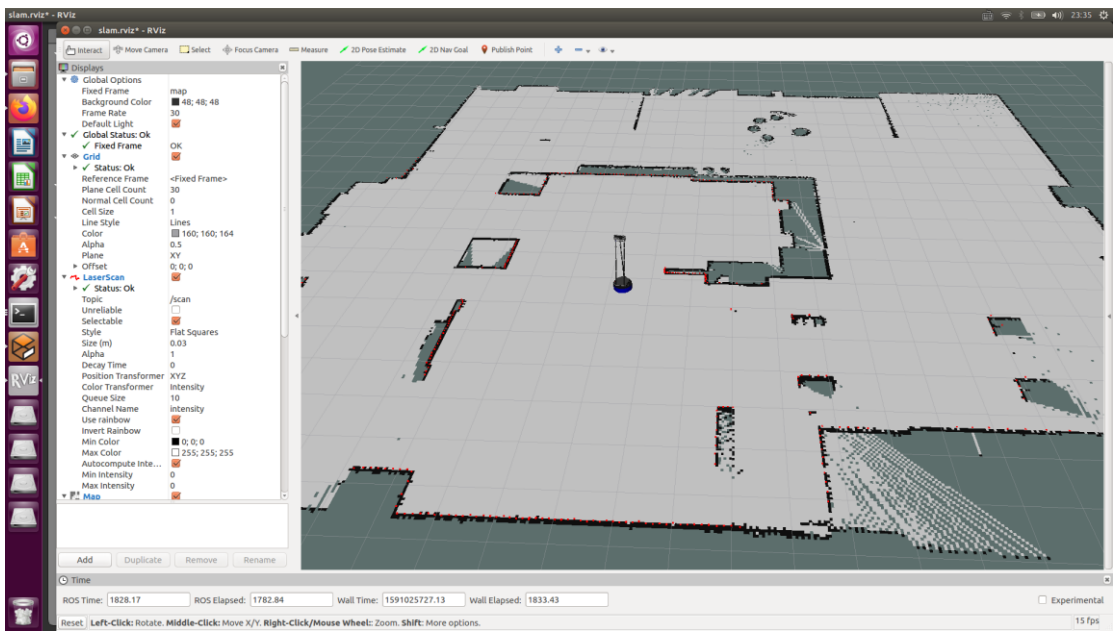


图 23 建立环境地图单元测试截图 4

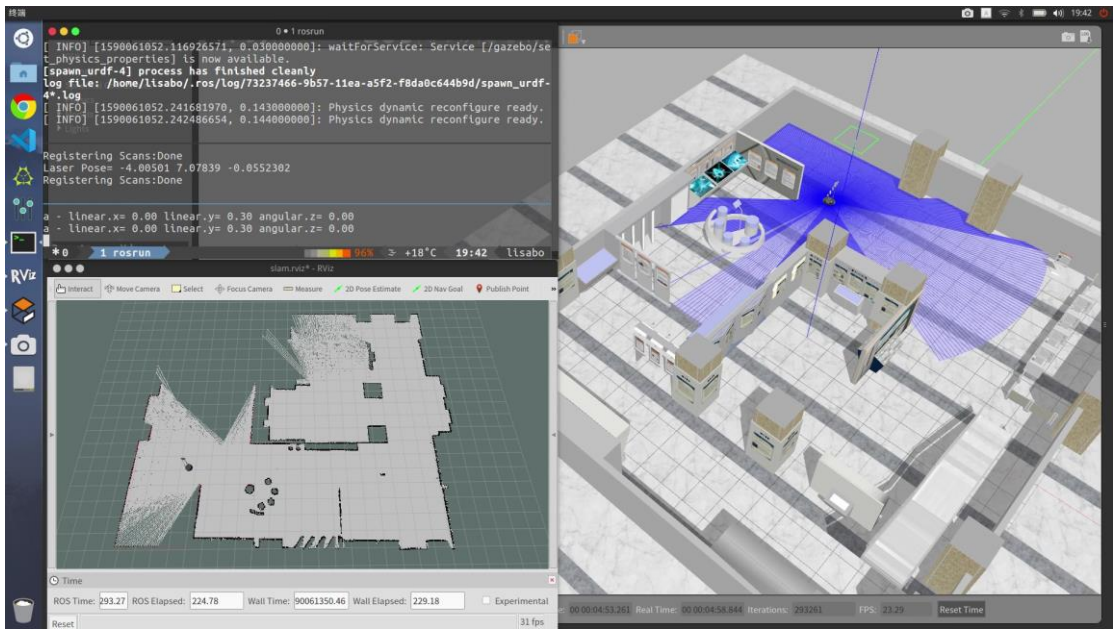


图 24 建立环境地图单元测试截图 5

5.1.3 导航功能

✧ 202UT3-1B

测试内容：添加与保存导航点

实际输出：指定路径下生成导航点文件 waypoints.xml

完成状态：优

发现问题: 无

✧ 202UT3-2B

测试内容: 导航节点算法

实际输出: 指定路径下生成导航点文件 waypoints.xml

完成状态: 开始导航后, 图中出现规划好的紫色路线。机器人沿紫色路径行进, 到达终点时, 机器人输出到达目标点语句。

发现问题: 有时机器人沿规定路径行进的时候会出现转圈的行为。

解决问题: 尝试改变机器人参数, 目前已解决相关问题。

✧ 202UT3-3B

测试内容: 测试导航过程中的避障功能

实际输出: 开始导航后, 图中出现规划好的紫色路线。机器人沿紫色路径行进, 遇到障碍物时机器人停下, 等待障碍物移开, 继续运动, 到达终点时, 机器人输出到达目标点语句。

完成状态: 优

发现问题: 无

测试截图:

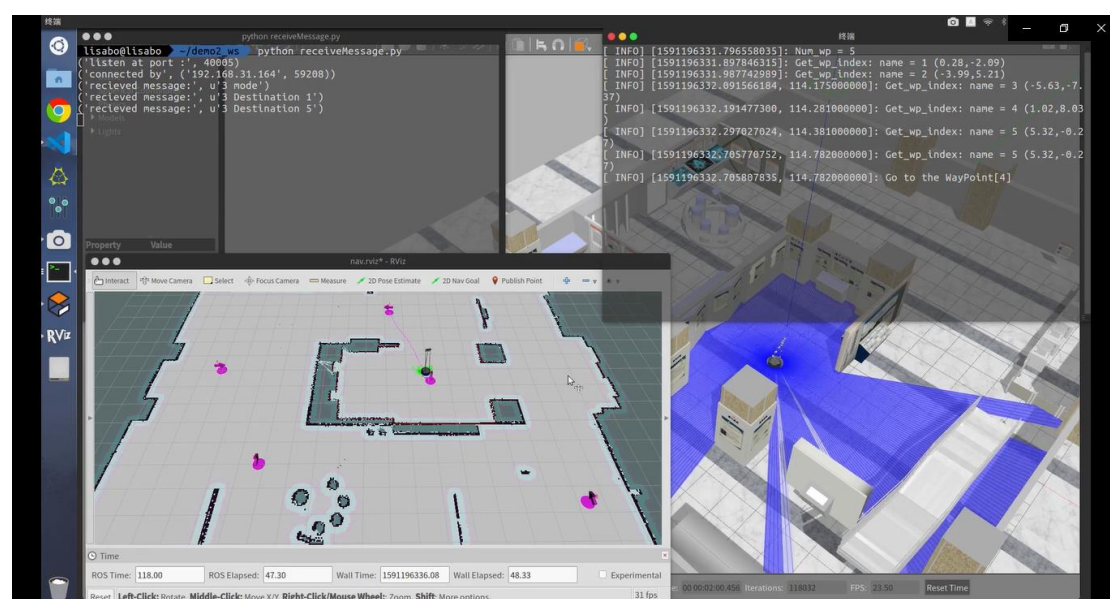


图 25 导航单元测试截图 1

5.1.4 目标检测

✧ 202UT4-1W

测试内容：机器人在非闲置时的目标识别

实际输出：机器人正忙

完成状态：优

发现问题：无

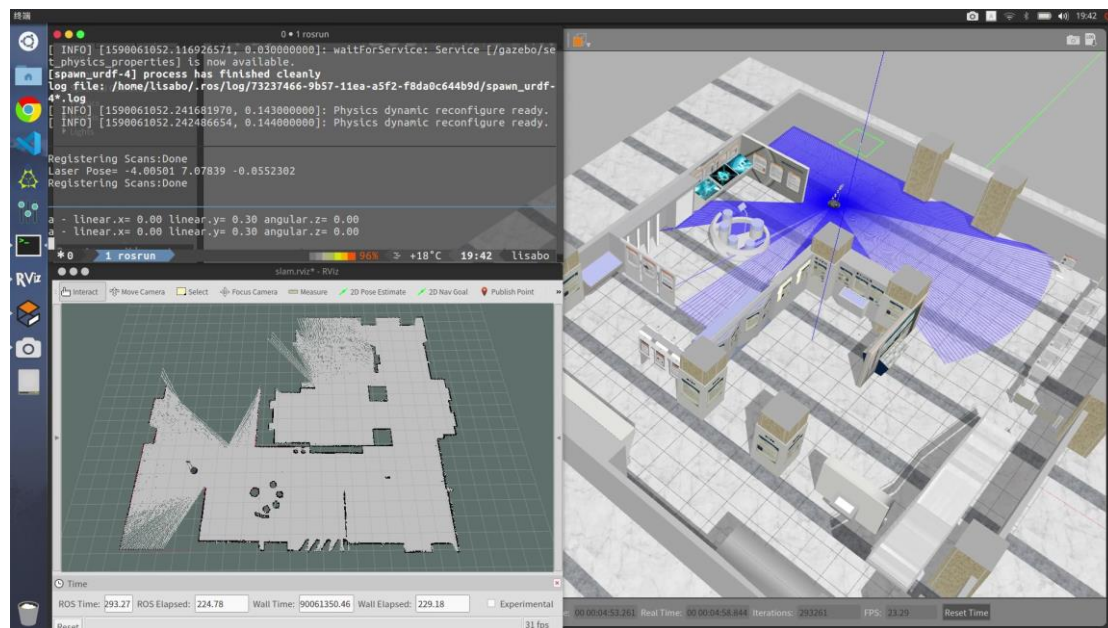


图 26 目标检测单元测试截图 1

✧ 202UT4-2W

测试内容：机器人闲置、无物品时的目标识别

实际输出：机器人未检测到物品

完成状态：优

发现问题：无



图 27 目标检测单元测试截图 2

◇ 202UT4-3W

测试内容：机器人闲置、有物品时的目标识别

实际输出：机器人检测到物品，返回物体位置

完成状态：优

发现问题：PCL 点云分割使用的一致性分割算法只能提取面，球体，圆柱等物品，复杂物品检测效果较差

解决方案：同时保存下分割前和分割后的图像，识别结果进行交叉验证，提升准确率

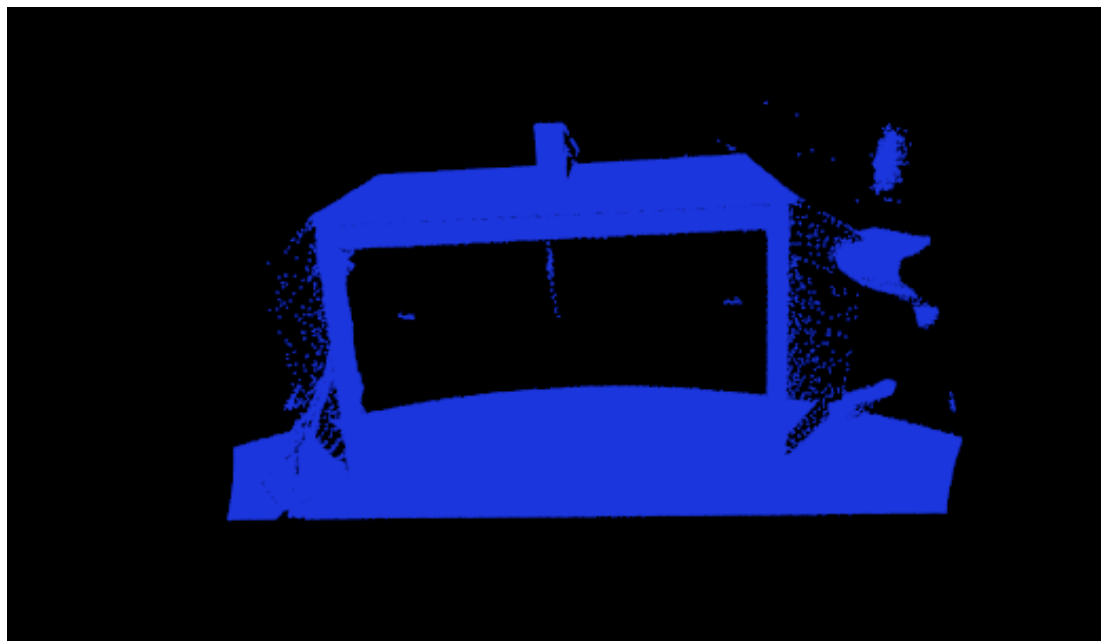


图 28 目标检测单元测试截图 3

5.1.5 语音交互功能

✧ 202UT5-1B

测试内容：识别语音指令

实际输出：与多名测试人员所说内容一致的字符串（包括“前进”、“后退”、“左转”、“右转”、“建立地图”、“导航”、“抓取”、“检测”）

完成状态：优

发现问题：虽然语音识别准确率较高，但测试发现调用科大讯飞语音识别 API 有时做不到实时检测（受网速及科大讯飞实时业务量影响）

解决方案：关闭其他联网进程以提高网速（暂无法提高科大讯飞 API 受理速度）

测试结果截图：

```
Start Recording...
Recording Done...
前进
后退,
左转
右转,
建立地图,
导航
抓取
检测。
```

图 29 语音交互单元测试截图 1**✧ 202UT5-2B**

测试内容：语音播报功能

实际输出：测试人员分别输入开始移动、停止移动、开始建图、建图完成、开始导航、到达目的地、发现障碍物、开始检测目标、检测目标完成、开始抓取、抓取成功、发生故障的状态更新指令，机器人通过扬声器播报状态

完成状态：优

发现问题：无

5.1.6 主控端**✧ 202UT6-2B**

测试内容：control_shell()函数，主控程序

实际输出：测试人员分别输入代表不同命令的字符串，观察是否运行相应命令行。

完成状态：优

发现问题：刚开始无法在运行一个命令行时无法打开下一个终端，之后改动了程序指令，目前已经解决了问题。

5.1.7 web 端**5.1.7.1 用户数据库测试****✧ 202UT71-1**

测试内容：注册新用户（注册项不符合格式要求）

实际输出：用户名不符合邮箱格式及用户名已被注册会收到提醒，但信息空缺不会进行报错

完成状态：良好

发现问题：信息空缺未报错

解决问题：在 input 框中添加 required data-msg 属性，解决了信息空缺不会对进行提醒的问题



图 30 用户数据库单元测试截图 1



图 31 用户数据库单元测试截图 2



图 32 用户数据库单元测试截图 3

✧ 202UT71-2

测试内容：注册新用户（注册项符合格式要求）

实际输出：注册成功并跳转到登录页面

完成状态：优

发现问题：无



我是一个还没设计完的logo

欢迎加入我们!

12@qq.com

..

注册

已有账户? [登录](#)

This is a registration form. It features a light blue header with a placeholder logo and a welcome message. Below this, there are two input fields: the first contains the email '12@qq.com' and the second contains two dots '..'. A prominent blue button labeled '注册' (Register) is positioned below the inputs. At the bottom, there is a link '已有账户? 登录' (Already have an account? Login).

图 33 用户数据库单元测试截图 4



我是一个还没设计完的logo

你好! 欢迎使用

注册成功! 请登录 ×

用户名/邮箱

密码

登录

[忘记密码?](#)

没有账户? [注册一个](#)

This is a login form. It features a light blue header with a placeholder logo and a greeting. Below this, there is a purple notification bar stating '注册成功! 请登录' (Registration successful! Please login) with a close button. The form includes two input fields: '用户名/邮箱' (Username/Email) and '密码' (Password). A prominent blue button labeled '登录' (Login) is positioned below the inputs. At the bottom, there are two links: '忘记密码?' (Forgot password?) and '没有账户? 注册一个' (No account? Register one).

图 34 用户数据库单元测试截图 5

✧ 202UT71-3

测试内容：已注册用户登录（个人信息与数据库中数据不相符）

实际输出：登陆失败并提醒用户用户名或密码不正确

完成状态：优

发现问题：无



图 35 用户数据库单元测试截图 6

✧ 202UT71-4

测试内容：已注册用户登录（个人信息与数据库中数据相符）

实际输出：登陆成功并跳转到首页

完成状态：优

发现问题：无

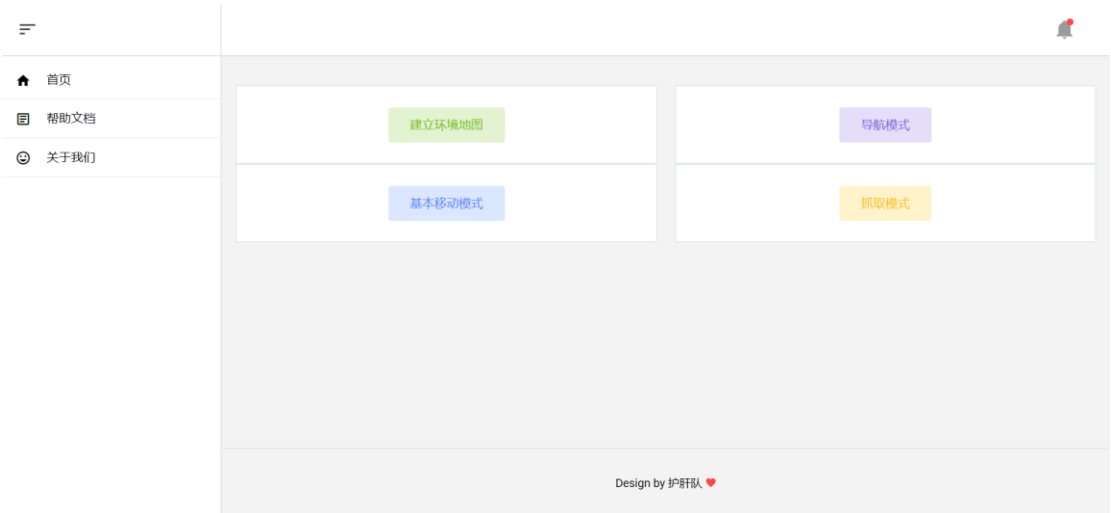


图 36 用户数据库单元测试截图 7

5.1.7.2 基本移动模式测试

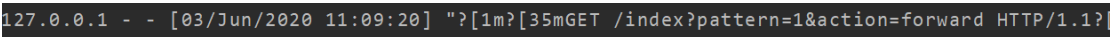
✧ 202UT72-1

测试内容：点击“前进”按钮，触发 goForward() js 函数

实际输出：GET/index?pattern=1&action=forward HTTP/1.1?

完成状态：优

发现问题：无



发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:09:47] "[1m?[35mGET /index?pattern=1&action=stop HTTP/1.1?"
```

图 39 基本移动页面单元测试截图 3

✧ 202UT72-4

测试内容：点击“左移”按钮，触发 goLeft() js 函数

实际输出：GET/index?pattern=1&action=left HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:09:44] "[1m?[35mGET /index?pattern=1&action=left HTTP/1.1?"
```

图 40 基本移动页面单元测试截图 4

✧ 202UT72-5

测试内容：点击“右移”按钮，触发 goRight() js 函数

实际输出：GET/index?pattern=1&action=right HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:09:48] "[1m?[35mGET /index?pattern=1&action=right HTTP/1.1?"
```

图 41 基本移动页面单元测试截图 5

✧ 202UT72-6

测试内容：点击“Exit”按钮，触发 exit1() js 函数

实际输出：GET/index?pattern=1&action=exit HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:09:51] "[1m?[35mGET /index?pattern=1&action=exit HTTP/1.1?"
```

图 42 基本移动页面单元测试截图 6

5.1.7.3 建立环境地图测试

✧ 202UT73-1

测试内容：点击“前进”按钮，触发 goForward_map() js 函数

实际输出：GET/index?pattern=2&action=forward HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:14:07] "[1m?[35mGET /index?pattern=2&action=forward HTTP/1.1?"
```

图 43 建立环境地图页面单元测试截图 1

✧ 202UT73-2

测试内容：点击“后退”按钮，触发 goBackward_map() js 函数

实际输出：GET/index?pattern=2&action=backward HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:14:09] "[1m?[35mGET /index?pattern=1&action=backward HTTP/1.1?"
```

图 44 建立环境地图页面单元测试截图 2

✧ 202UT73-3

测试内容：点击“停止”按钮，触发 stop_move_map() js 函数

实际输出：GET/index?pattern=2&action=stop HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:14:12] "[1m?[35mGET /index?pattern=2&action=stop HTTP/1.1?"
```

图 45 建立环境地图页面单元测试截图 3

✧ 202UT73-4

测试内容：点击“左移”按钮，触发 goLeft_map() js 函数

实际输出：GET/index?pattern=2&action=left HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:14:14] "[1m?[35mGET /index?pattern=2&action=left HTTP/1.1?"
```

图 46 建立环境地图页面单元测试截图 4

✧ 202UT73-5

测试内容：点击“右移”按钮，触发 goRight_map() js 函数

实际输出：GET/index?pattern=2&action=right HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:14:15] "?"[1m?[35mGET /index?pattern=2&action=right HTTP/1.1?
```

图 47 建立环境地图页面单元测试截图 5

✧ 202UT73-6

测试内容：点击“保存地图”按钮，触发 exit2() js 函数

实际输出：GET/index?pattern=2&action=exit HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:14:15] "?"[1m?[35mGET /index?pattern=2&action=right HTTP/1.1?
```

图 48 建立环境地图页面单元测试截图 6

5.1.7.4 导航模式测试

✧ 202UT74-1

测试内容：在下拉框中选择目的地，点击“开始导航”按钮，触发 start_navi() js 函数

实际输出：GET/index?pattern=3&action=start&place=目的地 2 HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:16:44] "?"[1m?[35mGET /index?pattern=3&action=start&place=目的地 2 HTTP/1.1?
```

图 49 导航页面单元测试截图 1

✧ 202UT74-2

测试内容：点击“Exit”按钮，触发 exit3() js 函数

实际输出：GET/index?pattern=3&action=exit HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:16:46] "?"[1m?[35mGET /index?pattern=3&action=exit HTTP/1.1?
```

图 50 导航页面单元测试截图 2

5.1.7.5 抓取模式测试

✧ 202UT75-1

测试内容：点击“开始检测”按钮，触发 startDetect() js 函数

实际输出：GET/index?pattern=4&action=start HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:17:38] "?"[1m?[35mGET /index?pattern=4&action=start HTTP/1.1?
```

图 51 导航页面单元测试截图 3

✧ 202UT75-2

测试内容：点击“Exit”按钮，触发 exit4() js 函数

实际输出：GET/index?pattern=4&action=exit HTTP/1.1?

完成状态：优

发现问题：无

```
127.0.0.1 - - [03/Jun/2020 11:17:39] "?"[1m?[35mGET /index?pattern=4&action=exit HTTP/1.1?
```

图 52 导航页面单元测试截图 4

5.1.7.6 页面跳转测试

✧ 202UT76-1

测试内容：首页、“帮助文档”页面、“关于我们”页面间的跳转

实际输出：GET /index?name=xxx HTTP/1.1?

GET /help HTTP/1.1?

GET /about_us HTTP/1.1?

完成状态：良好

发现问题：在“帮助文档”页面及“关于我们”页面无法回到首页

解决问题：help.html 及 about_us.html 页面中左侧工具栏中首页按钮 href 属

性需调用 app.py 中的路由跳转，即解决跳转问题

```
127.0.0.1 - - [03/Jun/2020 11:18:26] "?"[37mGET /help HTTP/1.1?[0m" 200 -  
127.0.0.1 - - [03/Jun/2020 11:18:27] "?"[37mGET /about_us HTTP/1.1?[0m" 200 -  
127.0.0.1 - - [03/Jun/2020 11:18:28] "?"[37mGET /index HTTP/1.1?[0m" 200 -
```

图 53 页面跳转单元测试截图

5.1.8 目标抓取

◇ 202UT8-1

测试内容：订阅 ProcCloudCB 节点和 PoseDiff 节点

实际输出：subscribe ProcCloudCB

dis = 0.9

完成状态：良好

◇ 202UT8-2

测试内容：发布点云分割结果和移动控制信息

实际输出：在 Rviz 显示出点云分割结果；

移动控制信息：w – linear.x = 0.10 linear.y = 0.00 angular.z = 0.00 等信息

完成状态：良好

测试过程截图：

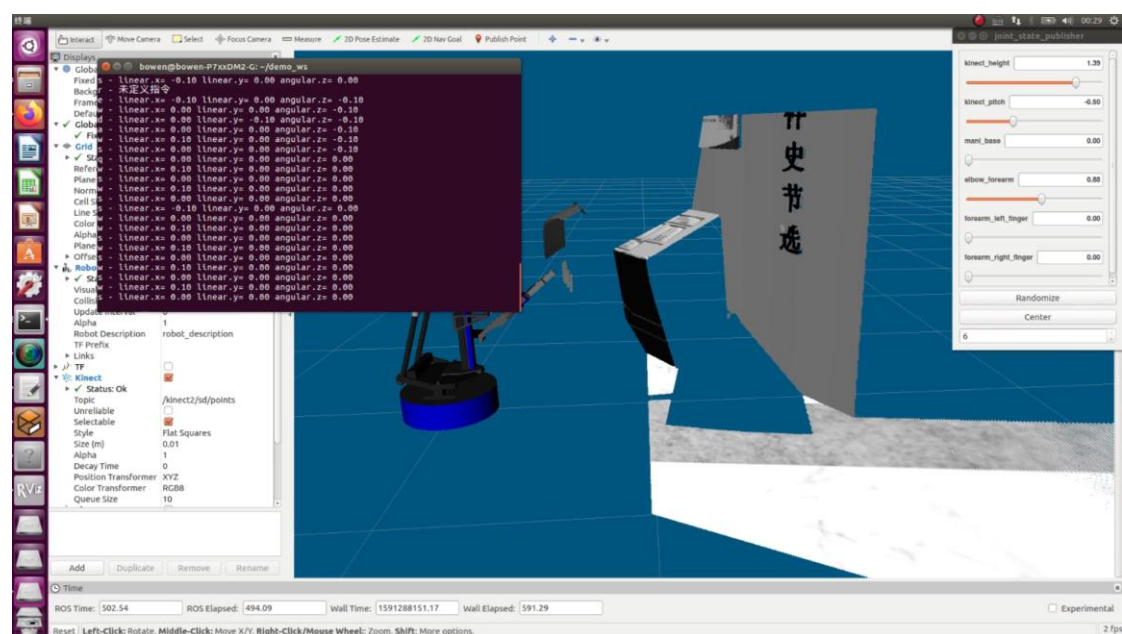


图 54 目标抓取单元测试截图 1

◇ 202UT8-3

测试内容：机器人对准目标物体

实际输出：机器人表现出接近物体行为。

完成状态：良好

◇ 202UT8-4

测试内容：机器人抓取功能

完成状态：较差

发现问题：只能在 Rviz 界面中通过手动调整机械臂高度、开合程度等参数控制机械臂抓取目标物体，无法实现自动抓取

5.2 系统测试

5.2.1 移动控制

✧ 202ST1-1

测试内容：web 前端与后端移动控制整体的连接与控制的正确性

实际输出：机器人按照用户点按的按键方向移动，控制台输出相关信息，用户退出手动控制界面后，机器人停止运动。主控电脑上，有速度信息的输出；用户退出后能，机器人返回退出状态。

完成状态：所预期的

发现问题：roslaunch 结束后，机器人还会继续移动，目前问题已经解决，在退出前主控端用 virtkey 模拟按空格键停止运动，然后关闭程序。由于网络连接的问题，可能出现传输不稳定或者延迟较大。

✧ 202ST1-2

测试内容：控制器处理网络连接中断；

实际输出：机器人能在断开连接后停止运动，注意终端显示退出移动控制状态；

完成状态：遇到问题；

发现问题：TCP 网络连接断开后，此时机器人有可能已经接近障碍物，导致系统刹车不及时而撞上去。后期考虑在手动的移动控制中，加入通过传感器主动减速刹车的功能。

[视频展示](#)

5.2.2 建立环境地图

✧ 202ST2-1

测试内容：用户建立（更新）地图

实际输出：机器人完成建图并语音播报“完成建图”。

完成状态：优

发现问题：无

[视频展示](#)

5.2.3 导航

✧ 202ST3-1

测试内容：web 前端，语音交互与导航功能连接与控制的正确性

实际输出：开始导航后，web 端反馈状态，图中出现规划好的紫色路线。机器人沿紫色路径行进，到达终点时，终端输出到达目标点语句

完成状态：所预期的

发现问题：无。

[视频展示](#)

5.2.4 目标检测

✧ 202ST4-1

测试内容：web 前端与目标检测功能连接与控制的正确性

实际输出：机器人反馈检测到物品，并返回物体位置

完成状态：所预期的

发现问题：无。

[视频展示](#)

5.2.5 语音交互

✧ 202ST5-1

测试内容：机器人的语音交互功能

实际输出：机器人正确地识别了包括“前进”、“后退”、“左转”、“右转”、“建立地图”、“导航”、“抓取”、“检测”在内的所有语音指令，正确地执行了对应的任务，并在完成任务后播报了完成任务。

完成状态：优

发现问题：由于调用百度 api，语音交互功能有一定的延迟。

[视频展示](#)

5.2.6 目标抓取

✧ 202ST6-1

测试内容：web 前端与目标抓取功能连接与控制的正确性

实际输出：Rviz 界面中有物体的点云分割结果，机器人接近目标物体后，只有机械臂控制信息，无抓取行为

完成状态：较差

发现问题：只能在 Rviz 界面中通过手动调整机械臂高度、开合程度等参数控制机械臂抓取目标物体，无法实现自动抓取

[视频展示](#)

6. 测试结果分析

6.1 对被测试软件的总体评估

单元测试中基本移动模块、建图模块、导航模块、语音交互模块、web 端及主控模块通过单元测试（包括白盒测试或黑盒测试），实现了模块内部功能。抓取模块未能参与测试，主要原因是机械臂与模拟器结合时出现问题，后续完成结合后，需要进行补充测试。

系统测试中对于对于移动控制功能，建立环境地图功能，导航功能，目标检

测功能，语音交互功能进行了测试，测试了 web 前端与目标检测功能连接与控制的正确性，并且对于非功能性需求的满足情况进行了测试。

6.2 测试环境的影响

测试环境为 gazebo 模拟器，该仿真环境在运行时未考虑实际物理接口连接等，与真实世界有一定区别，测试用例在真实物理环境中的运行结果可能与测试结果有细微不同。另外 gazebo 开启时的不稳定性可能造成建图失败，需要重启软件才可以建图，现实中应该不会出现该问题。

6.3 改进建议

强化对本地操作的反馈，尤其是失败情况下的反馈。丰富的反馈可以改善使用者的体验，提高系统易用性。

加快用户界面与机器人控制系统结合部分的开发，以及对语音模块和避障模块出现的问题尽快解决，并及时补充相应测试。

添加由于 gazebo 环境不稳定所造成的无法建图的异常处理：发出建图指令后，每隔 200 秒检测是否生成地图。若 600 秒后仍没有生成地图，则重启 GUI。目前已完成该部分改进。

进行总体测试，完善系统功能，丰富测试样例，根据代码和功能设计样例，覆盖性测试整个项目，进行集成的功能测试与非功能测试，提升系统可靠性。