

**简易机器人  
测试报告  
STR103  
V4.0**

## 分工说明

小组名称	undefined	
学号	姓名	本文档中主要负责的内容
16061160	陈麒先	编写测试用例，开展测试，记录分析测试结果
16061063	周尚纯	编写测试用例，记录分析测试结果
15071111	刘亮	编写测试用例，记录分析测试结果
16061005	秦枫	测试准备，参与测试工作
16061034	孙琦	测试准备，参与测试工作

## 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	2019.6.2	周尚纯等 5 人	陈麒先	测试报告第一版
V2.0	2019.6.2	孙琦等 5 人	陈麒先	测试报告第二版
V3.0	2019.6.2	陈麒先	秦枫	测试报告第三版
V4.0	2019.6.2	陈麒先	刘亮	测试报告第四版

# 1. 范围

## 1.1 项目概述

### 1.1.1 开发背景

机器人的研究和应用在很长一段时间内都是科学研究的热点。众所周知，每一次技术的变革都是对人类更进一步的解放，而机器人的诞生和应用也不例外。人们希望机器人能在农业、工业、服务业等行业中逐渐取代一些简单基本的人工，这要求机器人具有不亚于人的专业能力。

在机器人领域，日本、美国和欧洲已经有着牢固的技术基础和广泛的应用范围，而我国近年来工业机器人的生产和制造也取得成果并投入生产。世界各国对机器人的需求总体上呈增长趋势，人们对机器人的期望也向着智能化、技术化方向发展<sup>[1]</sup>。从技术发展的角度来看，就机器人移动和避障这一主题，除人工势场法、栅格法等传统算法外，更有基于神经网络、可视性二叉树、滚动时域控制等新颖算法的提出。

我们选择机器人嵌入式系统的开发，力图在前人开发经验的基础上自主实现具有避障、路径规划和目标抓取功能的机器人，并希望未来能够不断扩充和优化其性能，甚至实现技术突破。

### 1.1.2 功能和需求

我们项目的核心目标是实现一个可自主移动的简易机器人。其最主要也是最基本的功能是在行进中主动地避开障碍物，在此基础上，我们将追求更高的要求，即机器人能够依靠其视觉传感器分析地图信息，预先进行路径规划。我们的终极目标是实现机器人对目标物的检测和抓取功能。

除三个主要功能以外，我们对于成果机器人实现目标任务的程度也要进行评估和不断优化。具体包括越过障碍所需时间是否可以接受、在复杂情景中的避障行为是否正确、路径选择是否较优、识别检测到目标物的用时是否超长、抓取目标物的尝试次数是否有限、是否总能到达目标物所在的较小范围内、是否能稳定

实现有效抓取、能够抓取的目标种类和体量等，对于具体评测手段，我们将在实践和测试中给出量化指标。

1.1.3 用户价值与应用场景

我们认为具有自主移动和目标抓取功能机器人的实现是非常有意义的。这两项功能在现实中已经分别有广泛的应用，例如家用扫地机器人、工厂零件装配机器人等。而我们的项目集合这两项功能于一体，能够完成更为复杂的任务，适用于更多场景，比如火灾等危险场景中的救援和取物，物流业仓库物品的搬运、整理和摆放，家庭助理或老年人、残障人士助手，复杂环境和特殊对象的专业清洁，安全防爆检测，能源矿场采集，影视拍摄，甚至于送餐、送货行业等。

1.2 文档概述

本文档在设计说明书的基础上，对系统的实现情况进行测试。并对设计与实现阶段出现的问题进行复盘，并从不同的层次角度修复出现的系统缺陷，进而给出完整的项目测试流程。

1.3 术语和缩略词

表 1 术语和缩略词列表

测试用例编号	含义	问题编号	含义
KL-C-L-X	本地启动（Local Start） 第 X 个测试用例	KL-P-L-X	本地启动第 X 个问题
KL-C-W-X	网页启动（Web Start） 第 X 个测试用例	KL-P-W-X	网页启动第 X 个问题
KL-C-T-X	通信 （Tele-communication） 第 X 个测试用例	KL-P-T-X	通信第 X 个问题
KL-C-U-X	紧急制动（Urgent Stop） 第 X 个测试用例	KL-P-U-X	紧急制动第 X 个问题
KL-C-F-X	跟随（Follow）第 X 个	KL-P-F-X	跟随第 X 个问题

	测试用例		
KL-C-P-X	路径规划 (Path Plan) 第 X 个测试用例	KL-P-P-X	路径规划第 X 个问题
KL-C-B-X	建图 (Build Map) 第 X 个测试用例	KL-P-B-X	建图第 X 个问题
KL-C-M-X	标点 (Mark Point) 第 X 个测试用例	KL-P-M-X	标点第 X 个问题
KL-C-G-X	抓取 (Grab) 第 X 个测试用例	KL-P-G-X	抓取第 X 个问题
KL-C-V-X	语音 (Voice) 第 X 个测试用例	KL-P-V-X	语音第 X 个问题
KL-C-A-X	避障 (Avoid) 第 X 个测试用例	KL-P-A-X	避障第 X 个问题

## 1.4 引用文档

- 【1】 曹泓浩.工业机器人的应用现状及发展趋势[J].科技风,2019(05):145.
- 【2】 晋晓飞,王浩,宗卫佳,王鹏程,王策.自主移动机器人避障技术研究现状[J].传感器与微系统,2018,37(05):5-9.
- 【3】 SRS103 需求规格说明书
- 【4】 SDP103 项目开发计划
- 【5】 SDD103 设计说明书

## 2. 任务概述

表 2 功能需求

序号	优先级	名称	概述
----	-----	----	----

1	高	避障	实时监测障碍物并躲避
2	高	路径规划	进行全局路径规划
3	高	开关	通过机载电脑启动/关闭机器人
4	高	建图	完成初始化地图场景建模
5	低	查图	可以通过机载界面查看实时地图
6	中	标点	可通过语音指令记录关键物品地点
7	中	抓取	通过语音命令机器人抓取指定物品
8	高	异常	超市工作人员/顾客可关联错误条件
9	中	跟随	机器人跟随工作人员步行引导，能够语音控制结束
10	低	通信	网页控制机器人启动停止，网页显示机器人信息

表 3 非功能需求

序号	名称	概述
1	系统可靠性需求	系统应保证超市营业时间内不间断运行，具有冗余等安全措施软，以及软硬件故障在线恢复的能力。
2	系统可扩展性需求	机器人设备应能够适应系统容量的扩大和管理内容的增加，包括软硬件平台、系统结构、功能设计、管理对象。
3	系统易用性需求	使用者在使用该机器人的过程中不会产生压力或感到挫折，能用最少的努力发挥机器人最大的效能。
4	系统安全性需求	包括系统保密性需求、系统完整性需求、漏洞检测和安全风险评估、系统可用性和抗毁性以及系统防病毒。

### 3. 测试准备

#### 3.1 测试流程

测试流程分为以下几个部分。

1. 单元测试：单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等等。我们的单元测试分为以下几个部分：

（1）单元独立执行路径测试：对每一条独立执行路径进行测试；

（2）单元局部数据结构的测试：检查临时存储的数据在程序执行过程中是否正确、完整。检查是否有以下几类的错误：a.不合适或不相容的类型说明。b.变量初始化或缺省值有错。c.变量无初值。d.不正确的变量名。e.出现上溢、下溢或地址异常；

（3）单元接口测试：数据能正确输入输出；

（4）单元边界条件测试；

（5）单元容错性测试：针对出错进行正确的处理；

2. 集成测试：集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。我们的测试模式是非渐进式测试模式：先分别测试每个模块，再把所有模块按设计要求放在一起结合。我们采用自顶向下的集成方法。

3. 系统测试：系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求。我们依据需求规格说明书文档，对系统的功能和功能性需求进行测试。对界面、数据、操作、逻辑、接口、稳定性等方面进一步测试。

## 3.2 硬件、软件环境

### 3.2.1 硬件环境

嵌入式开发板+各类传感器+运动装置+机械臂装置

### 3.2.2 软件环境

Ubuntu16.04+ROS+CMAKE+RoboWare Studio

## 3.3 测试人员和时间分配

表 4 测试人员分工

测试阶段	测试人员	测试内容	截止时间
单元测试	陈麒先	void follow_test(); void start_test(); void FollowSwitch_test();	5/19
	刘亮	void Speak_test(); void AddNewWaypoint_test(); void InitKeyword_test();	5/19
	周尚纯	void GrabSwitch_test(); void PassSwitch_test(); void xxgrab_test();	5/19
	秦枫	void xxpass_test(); void GrabResultCallback_test(); void PassResultCallback_test();	5/19
	孙琦	void FindKeyword_test(); void Keyword_test_CB(); void xxnav_test();	5/19
集成测试	全员	软件单位之间的接口是否正确	5/23
系统测试	全员	对简易机器人系统的各项功能覆盖测试	6/2

### 3.4 测试数据准备

环境准备：选择场地为 7m×8m 的矩形封闭房间。

道具准备：准备规格为下表所示的塑料瓶各一个，作为待抓取物体：

表 5 测试道具准备

编号	颜色	形态	内容物
1	黄色	正常 500ml 饮料瓶大小	空
2	透明（带绿色标签）	正常 500ml 饮料瓶大小	空
3	透明（无标签）	正常 500ml 饮料瓶大小	空
4	透明（无标签）	正常 500ml 饮料瓶大小	有水
5	透明（带绿色标签）	大型 1.5L 饮料瓶大小	空

准备一张高 80cm 的桌子作为抓取平面，清除桌边无关杂物，保持沿桌边 40cm、距桌沿 30cm 范围内桌面空白；准备其余两张桌子作为障碍物。

编号准备：



表 6 测试用例的编号

测试用例编号	含义	问题编号	含义
KL-C-L-X	本地启动（Local Start） 第 X 个测试用例	KL-P-L-X	本地启动第 X 个问题
KL-C-W-X	网页启动（Web Start） 第 X 个测试用例	KL-P-W-X	网页启动第 X 个问题
KL-C-T-X	通信 （Tele-communication） 第 X 个测试用例	KL-P-T-X	通信第 X 个问题
KL-C-U-X	紧急制动（Urgent Stop） 第 X 个测试用例	KL-P-U-X	紧急制动第 X 个问题
KL-C-F-X	跟随（Follow）第 X 个 测试用例	KL-P-F-X	跟随第 X 个问题
KL-C-P-X	路径规划（Path Plan）第 X 个测试用例	KL-P-P-X	路径规划第 X 个问题
KL-C-B-X	建图（Build Map）第 X 个测试用例	KL-P-B-X	建图第 X 个问题
KL-C-M-X	标点（Mark Point）第 X 个测试用例	KL-P-M-X	标点第 X 个问题
KL-C-G-X	抓取（Grab）第 X 个测 试用例	KL-P-G-X	抓取第 X 个问题
KL-C-V-X	语音（Voice）第 X 个测 试用例	KL-P-V-X	语音第 X 个问题
KL-C-A-X	避障（Avoid）第 X 个测 试用例	KL-P-A-X	避障第 X 个问题

人员准备：小组成员 5 人，在测试中分工如下：

表 7 测试人员分工

陈麒先	主测试人员，在各项测试中代替用户角色。
-----	---------------------

刘亮	在语音测试中充当男声测试人员。在避障测试中充当临时障碍物。
周尚纯	在语音测试中充当女声测试人员。
孙琦	负责目标物的摆放与布置。
秦枫	负责测量各项距离指标是否符合测试用例标准。

## 4. 测试用例

### 4.1 本地启动

- 测试用例标识：KL-C-L-1。(case-localStart)
- 对应的需求：对应系统功能需求的【3】。能够正确地启动脚本，使得系统具备执行接下来的任务的能力。
- 条件或状态：各个硬件设施状况良好，正确连接至机载电脑。机载电脑连接到互联网。机载电脑与底盘均处于有电状态。ROS 的 main 节点没有被定位到其他机器。
- 输入：在机载电脑上打开新终端，输入命令“roslaunch Team103 team103.launch”
- 预期的输出：
  - 1) 会弹出 Rviz 界面，可以看到建图功能开启，同时在机器人的初始位置生成一个“start”航点。
  - 2) 会弹出 Rviz 界面，地图缺失，同时在机器人的初始位置生成一个“开始”航点，终端内出现红色的错误信息。
  - 3) 不弹出 Rviz 界面，终端内出现红色的错误信息。
- 评价准则：出现输出 1)时测试成功，出现 2)或 3)时测试失败。
- 测试流程：将机载电脑连接到互联网上，将机器人移动到建图的起点，打开终端程序，测试人员执行[输入]中的内容，其余人员观察并记录机载电脑控制台的显示情况，验证结果是否符合评价标准。

## 4.2 网页启动

- 测试用例标识：KL-C-W-1。(case-webStart)
- 对应的需求：对应系统功能需求的【3】【10】。能够通过网页正确地启动脚本，使得系统具备执行接下来的任务的能力，并实现网页与机载电脑的通信。
- 条件或状态：各个硬件设施状况良好，正确连接至机载电脑。机载电脑与底盘均处于有电状态。ROS 的 main 节点没有被定位到其他机器。机载电脑与个人电脑均通过 Wi-Fi 连接到同一局域网。
- 输入：
  - 1) 在机载电脑上打开新终端，输入命令“roscore”
  - 2) 在机载电脑上打开新终端，输入命令“roslaunch myserver shownstate.py”
  - 3) 在个人电脑上打开网页 114.115.205.63:8090 并点击机器人控制台
  - 4) 点击一键启动按钮
- 预期的输出：
  - a) 机载电脑在执行输入 1)后的输出：
    - 1) 终端无报错，节点启动。
    - 2) 终端出现红色报错，节点启动失败。
  - b) 机载电脑在执行输入 2)后的输出：
    - 1) 终端显示机载电脑无线网卡 IP 地址，显示“waiting for connecting...”。
    - 2) 终端显示错误信息。
  - c) 执行输入 3)后的输出：
    - 1) 个人电脑：主界面网页成功打开，机器人控制界面成功打开，机器人控制界面显示等待连接，变为连接成功。机载电脑：显示“connected from:”，显示个人电脑的 IP 及端口，显示"handshake success"，显示“new thread for client ...”
    - 2) 个人电脑：主界面网页打开失败。机载电脑：无特殊显示。
    - 3) 个人电脑：主界面网页打开成功，机器人控制界面打开失败。机载电脑：无特殊显示。

- 4) 主界面网页打开成功，机器人控制界面打开成功，机器人控制界面显示等待连接，变为连接失败。机载电脑：无特殊显示。
- 5) 主界面网页打开成功，机器人控制界面打开成功，机器人控制界面显示等待连接，变为连接失败。机载电脑：显示“handshake success”，显示“start new thread error: ”加上错误信息。
- d) 机载电脑在执行输入 4)后的输出:同 4.1 样例的输出情况。
- 评价准则：各部分输出全部符合 1)中输出，即为测试成功；其他情况均视为测试失败。
- 测试流程：将机载电脑和个人电脑连接到 BUAA-WiFi 上，将机器人移动到建图的起点，测试人员执行[输入]中的内容，其余人员观察并记录机载电脑控制台、个人电脑网页的显示情况，验证结果是否符合评价标准。

### 4.3 通信

- 测试用例标识：KL-C-T-1。(case-telecommunication)
- 对应的需求：能够对应系统功能需求的【10】。实现个人电脑上网页与机载电脑之间的通信：通过网页显示机器人状态。
- 条件或状态：在 4.2 样例正确执行之后进行测试。
- 输入：无。
- 预期的输出：
  - 1) 网页上实时显示机器人的状态信息。
  - 2) 网页上无变化。
- 评价准则：输出 1)代表测试成功，其他情况代表失败。
- 测试流程：执行 4.2 测试用例，在该用例成功的基础上，测试人员引导机器人进行标点、抓取任务，其余人员观察并记录机载电脑控制台、个人电脑网页的显示情况，验证结果是否符合评价标准。

### 4.4 紧急制动

- 测试用例标识：KL-C-U-1。(case-urgeStop)
- 对应的需求：能够对应系统功能需求的【10】。实现个人电脑上网页与机

载电脑之间的通信：通过网页按键点击，停止机器人运动。

- 条件或状态：在 4.2 样例正确执行之后进行测试。
- 输入：在网页上点击“紧急制动”按钮。
- 预期的输出：
  - 1) 机器人运动状态变为静止。
  - 2) 机器人保持运动状态。
- 评价准则：输出 1)代表测试成功，其他情况代表失败。
- 测试流程：执行 4.2 测试用例，在该用例成功的基础上，测试人员引导机器人进行运动，其余人员观察并记录机载电脑控制台及机器人行为，验证结果是否符合评价标准。

## 4.5 跟随测试

跟随测试总述：

- 对应的需求：能够对应系统功能需求的【9】。机器人跟随引导员运动。
- 测试流程：如图所示。按如图的方式固定好目标物体、机器人。测试人员站在机器人前 1m 位置。启动脚本，测试人员分别向前、后、左、右移动 0.4m。随后再分别向左、向右旋转错误!未找到引用源。rad。检查机器人能否跟随测试人员运动。

跟随测试分述：

- 测试用例标识：KL-C-F-1。(case-follow)
  - 输入：测试人员向前移动 0.4m。
  - 预期的输出：
    - 1) 机器人向前移动 0.4m。
    - 2) 机器人移动至错误的方向与距离。
  - 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
- 
- 测试用例标识：KL-C-F-2。
  - 输入：测试人员向后移动 0.4m。
  - 预期的输出：

- 1) 机器人向后移动 0.4m。
- 2) 机器人移动至错误的方向与距离。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
  
- 测试用例标识：KL-C-F-3。
- 输入：测试人员向左移动 0.4m。
- 预期的输出：
  - 1) 机器人向左移动 0.4m。
  - 2) 机器人移动至错误的方向与距离。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
  
- 测试用例标识：KL-C-F-4。
- 输入：测试人员向右移动 0.4m。
- 预期的输出：
  - 1) 机器人向右移动 0.4m。
  - 2) 机器人移动至错误的方向与距离。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
  
- 测试用例标识：KL-C-F-5。
- 输入：测试人员向左旋转**错误!未找到引用源。**rad。
- 预期的输出：
  - 1) 机器人向左旋转**错误!未找到引用源。**rad。
  - 2) 机器人移动至错误的方向与距离。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
  
- 测试用例标识：KL-C-F-6。
- 输入：测试人员向右旋转**错误!未找到引用源。**rad。
- 预期的输出：
  - 1) 机器人向右旋转**错误!未找到引用源。**rad。

2) 机器人移动至错误的方向与距离。

➤ 评价准则：得到预期输出 1) 即为测试通过。反之不通过。

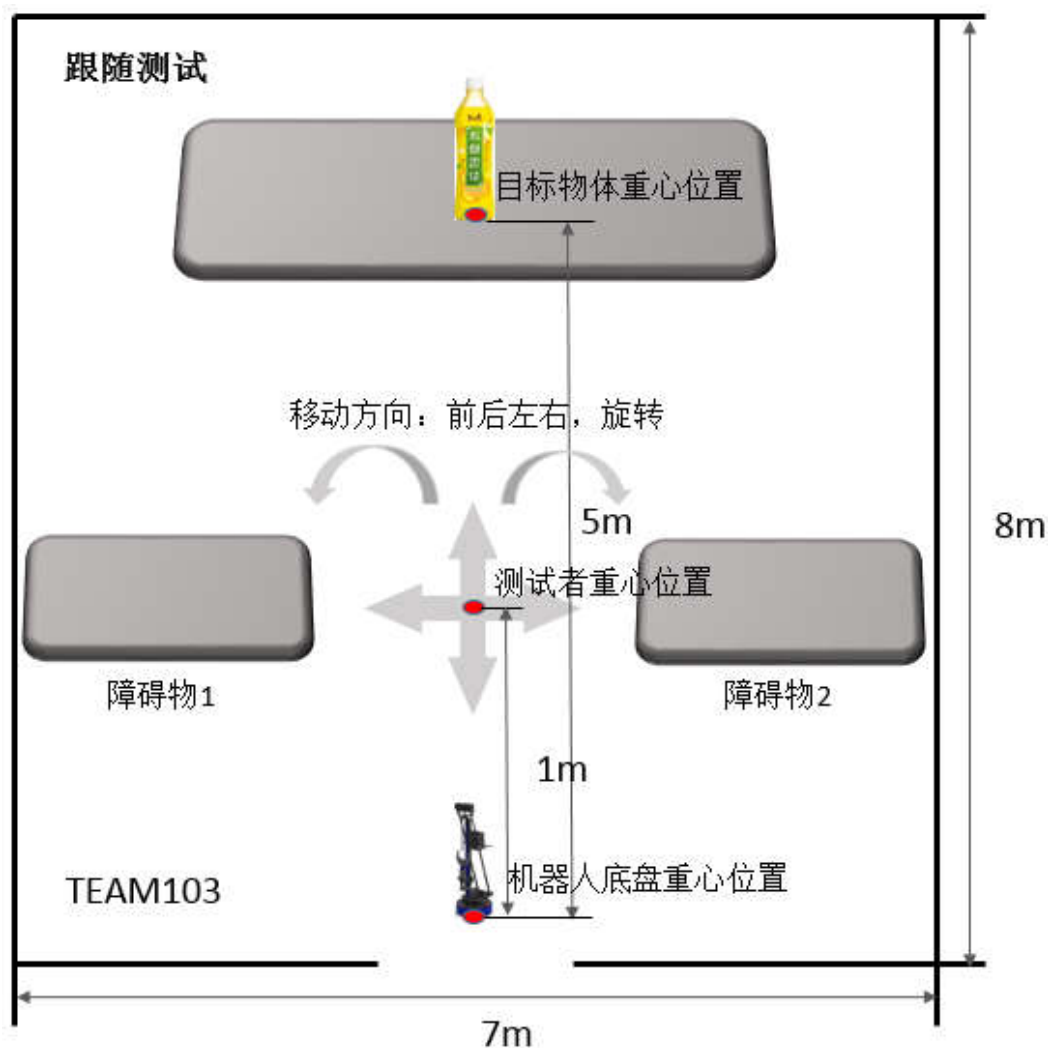


图 1 跟随测试用例

## 4.6 路径规划

- 测试用例标识：KL-C-P-1。(case-pathPlan)
- 对应的需求：能够对应系统功能需求的【2】。机器人能够规划出一条从起始点到达目的地之间的最短路径。
- 输入：测试人员向左前方沿一个半径为 0.5m 的弧前进 错误!未找到引用源。 rad，再向右前方沿一个半径为 0.5m 的弧前进 错误!未找到引用源。 rad，最后再向左前方沿一个半径为 0.5m 的弧前进错误!未找到引用源。 rad 到达目的地。

- 预期的输出：
  - 1) 机器人沿一条从起始点指向目的地的直线前进 5m 到达目的地。
  - 2) 机器人沿其他路线走向目的地。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
- 测试流程：如图所示。按如图的方式固定好目标物体、机器人。测试人员站在机器人前 1m 位置。启动脚本，测试人员向左前方沿一个半径为 0.5m 的弧前进 **错误!未找到引用源。** rad, 再向右前方沿一个半径为 0.5m 的弧前进 **错误!未找到引用源。** rad, 最后再向左前方沿一个半径为 0.5m 的弧前进 **错误!未找到引用源。** rad 到达目的地。回到出发点后，向机器人下达取物指令。验证结果是否符合评价标准。

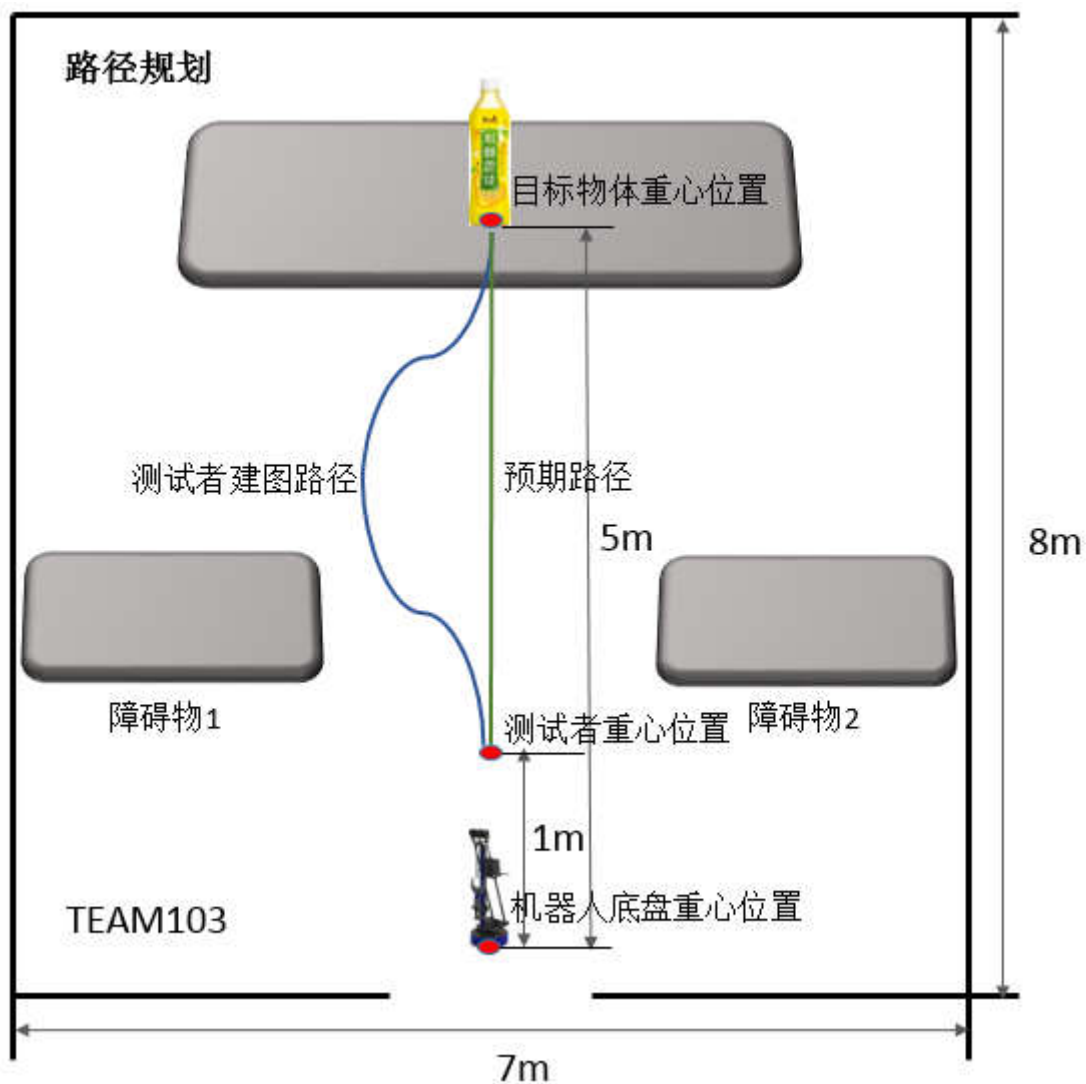


图 2 路径规划测试用例



## 4.7 建图测试

- 测试用例标识：KL-C-B-1。(caseBuildMap)
- 对应的需求：能够对应系统功能需求的【4】。要求在工作人员指引下，机器人能够在指定路径下生成正确的地图文件。
- 输入：测试人员按照如图所示的路线下达指令引导机器人探索整个环境空间。
- 预期的输出：
  - 1) 建图完成后，在~/catkin\_ws/src/team103/maps/目录下应生成了两个文件，分别为 map.pgm 和 map.yaml。
  - 2) 在其他路径下生成地图文件。
  - 3) 生成的文件不完整或无法正常读取其中的数据。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
- 测试流程：如图所示。按如图的方式固定好目标物体、机器人测试人员按照如图所示的路线下达指令引导机器人探索整个环境空间。启动脚本。建图动作完成后，检查机器人机载电脑相关目录，查看是否成功获取到相应的地图文件。

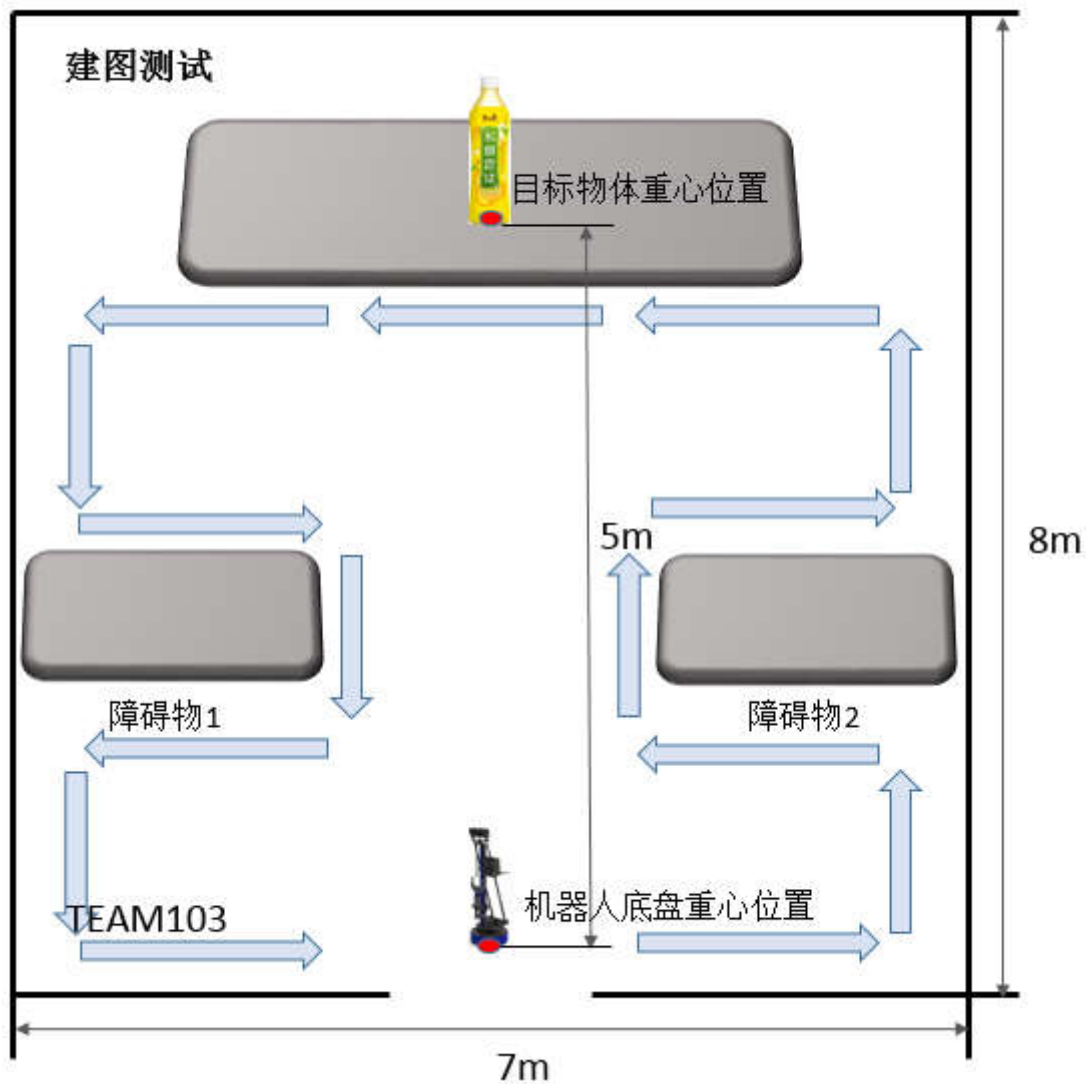


图 3 建图测试用例

## 4.8 标点测试

标点测试总述:

- 对应的需求：能够对应系统功能需求的【10】。机器人需要标定不同的航点。
- 测试流程：如图所示。按如图的方式固定好目标物体、机器人。测试人员站在机器人前 1m 位置。启动脚本，测试人员向左前方沿一个半径为 0.5m 的弧前进 **错误!未找到引用源。** rad, 再向右前方沿一个半径为 0.5m 的弧前进 **错误!未找到引用源。** rad, 最后再向左前方沿一个半径为 0.5m 的弧前进 **错误!未找到引用源。** rad 到达目的地。到达目的地后，测试

者位于机器人前方 0.7m，距离目标物体 0.5m 关键词和地点。测试人员面朝机器人麦克风，距离麦克风 0.3m，向机器人下达取物指令。验证机器人任务执行结果是否符合评价标准。

标点测试分述：

- 测试用例标识：KL-C-M-1。(case-markPoint)
- 输入：

测试者位于机器人前方 0.7m，距离目标物体 0.5m 关键词和地点。测试人员面朝机器人麦克风，距离麦克风 0.3m，向机器人下达取物指令。

测试者高声宣读关键词库中的内容：“这是一瓶水”。
- 预期的输出：
  - 1) 机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。
  - 2) 其他输出或不响应标点指令。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
  
- 测试用例标识：KL-C-M-2。
- 输入：

测试者位于机器人前方 0.7m，距离目标物体 0.5m 关键词和地点。测试人员面朝机器人麦克风，距离麦克风 0.3m，向机器人下达取物指令。

测试者高声宣读关键词库中的内容：“这是可乐”。
- 预期的输出：
  - 1) 机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。
  - 2) 其他输出或不响应标点指令。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。
  
- 测试用例标识：KL-C-M-3。
- 输入：

测试者位于机器人前方 0.7m，距离目标物体 0.5m 关键词和地点。测试人员面朝机器人麦克风，距离麦克风 0.3m，向机器人下达取物指令。

- 测试者高声宣读关键词库中的内容：“这是一杯奶茶”。
- 预期的输出：
  - 1) 机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。
  - 2) 其他输出或不响应标点指令。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。

- 测试用例标识：KL-C-M-4。

- 输入：

测试者位于机器人前方 0.7m，距离目标物体 0.5m 关键词和地点。测试人员面朝机器人麦克风，距离麦克风 0.3m，向机器人下达取物指令。

测试者高声宣读不包含任何关键词的内容：“我们爱先先”。

- 预期的输出：
  - 1) 机器人不应产生任何响应。
  - 2) 其他输出。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。

- 测试用例标识：KL-C-M-5。

- 输入：

测试者位于机器人前方 0.7m，距离目标物体 0.5m 关键词和地点。测试人员面朝机器人麦克风，距离麦克风 0.3m，向机器人下达取物指令。

测试者高声宣读一个包含已标记过的关键词：“这是一瓶水”。

- 预期的输出：
  - 1) 机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。不会覆盖掉先前已标记的航点位置。
  - 2) 其他输出或不响应标点指令。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。

- 测试用例标识：KL-C-M-6。
- 输入：

测试者位于机器人前方 0.7m，距离目标物体 0.5m 关键词和地点。测试人员面朝机器人麦克风，距离麦克风 0.3m，向机器人下达取物指令。

测试者在听到提示音后，不下达任何指令。
- 预期的输出：
  - 1) 机器人不产生任何反馈。
  - 2) 其他输出结果。
- 评价准则：得到预期输出 1) 即为测试通过。反之不通过。

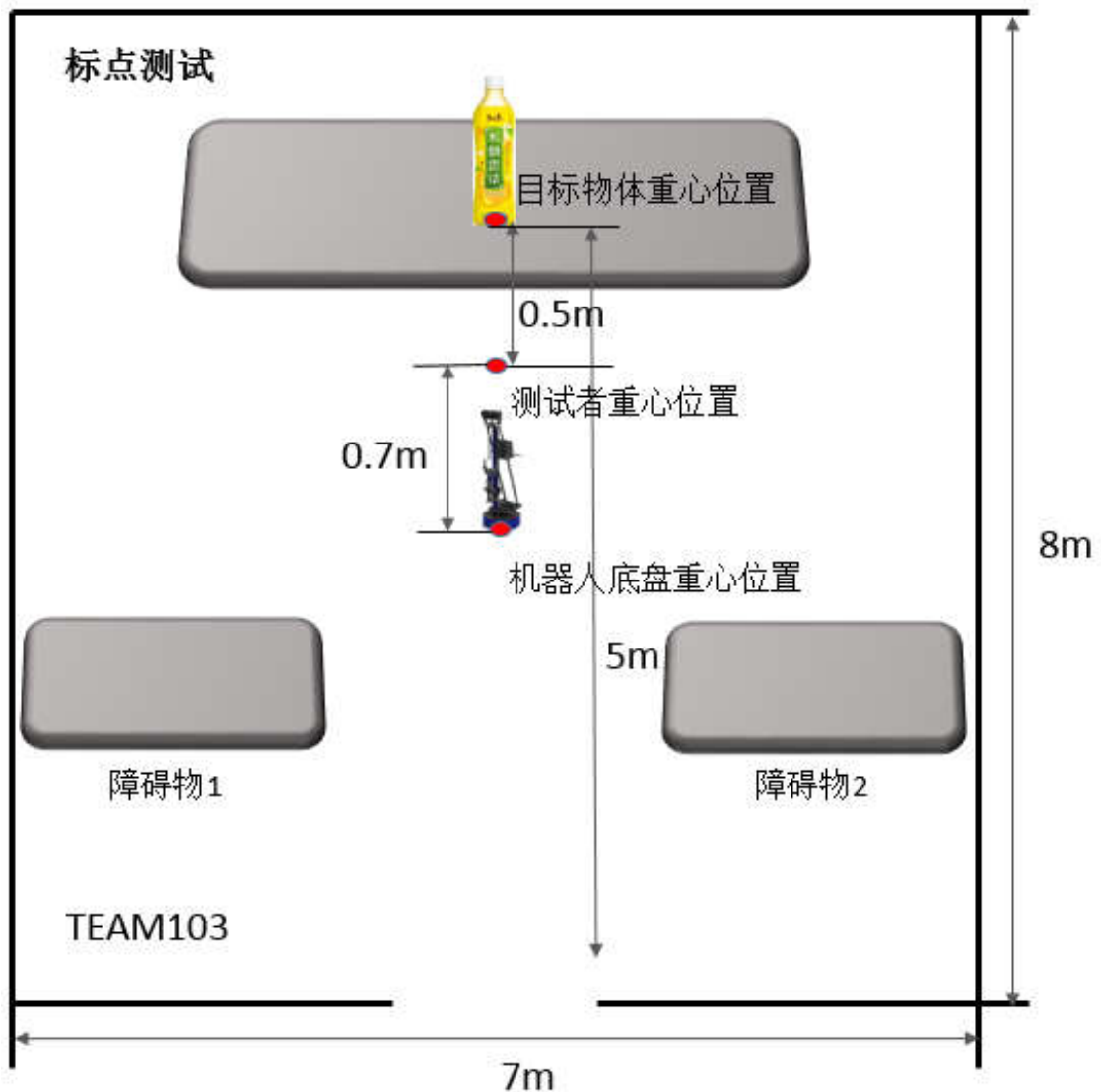


图 4 标点测试用例

## 4.9 抓取

抓取模块测试总述：

- 对应的需求：对应系统功能需求【7】。可以准确抓到指定物体，并能返回并将物品成功递给用户。
- 条件或状态：在测试样例 4.2 正确执行后，机器人正常启动并完成跟随和标点，已录入的关键词包括：“开始”、“水”、“奶茶”、“咖啡”，其中，位置已标定的有：“水”、“奶茶”。测试桌面上除测试

测试用例分述：

- 测试用例标识：KL-C-G-1。(case-grab)
- 测试针对性：测试机器人能否到达指定航点抓取对应物品并返回交给测试人员，多次改变物体距桌边距离。
- 输入： 测试人员标准普通话语音：“给我拿一瓶水。”

代表水的黄色塑料瓶距桌边距离：0cm, 5cm, 10cm, 15cm, 20cm, 25cm, 30cm。

- 预期的输出：对每一次距离调整后可能的输出情况有：

1) 机器人语音播报：“I got it. I am coming back.”，机器人成功抓取物体并返回递给测试人员；

2) 机器人语音播报：“I got it. I am coming back.”，机器人成功抓取物体并返回，递给测试人员时发生错误；

3) 机器人语音播报：“I got it. I am coming back.”，机器人成功抓取物体，但未返回“开始”航点；

4) 机器人语音播报：“I got it. I am coming back.”，机器人抓取物体失败，包括碰倒物体或姿态调整有误；

5) 机载电脑控制台输出：“[ WARN] [时间]: Could not estimate a planar model for the given dataset.” 机器人到达航点后直接返回。

- 评价准则：输出 1)代表测试成功，其他情况代表失败。
- 测试流程：将黄色塑料瓶放置在 80cm 高的桌面上，正常启动机器人，由一名测试人员引导机器人标点“水”后，等待机器人停止跟随并发出“噔噔噔”的准备录音提示后，每次调整塑料瓶距离桌边距离如[输入]，

并用标准普通话对机器人说出[输入]中的语句，其余人员观察并记录机载电脑控制台，机器人语音播报的输出信息及机器人行为。

- 测试用例标识：KL-C-G-2。
- 测试针对性：测试机器人能否到达放有多个物品的指定航点，抓取对应物品并返回交给测试人员，多次改变待抓取物体与其他物体之间的距离。
- 输入：测试人员标准普通话语音：“给我拿一瓶水。”

代表水的黄色塑料瓶距桌边距离：10cm，其与其他物体之间的距离：5cm，10cm，15cm，20cm，25cm，30cm。

- 预期的输出：对每一次距离调整后可能的输出情况有：
  - 1) 机器人语音播报：“I got it. I am coming back.”，机器人成功抓取目标物体并返回递给测试人员；
  - 2) 机器人语音播报：“I got it. I am coming back.”，机器人抓取非目标物体并返回递给测试人员；
  - 3) 同上一测试用例的输出 2)~5)。
- 评价准则：输出 1)代表测试成功，其他情况代表失败。
- 测试流程：将黄色塑料瓶放置在 80cm 高的桌面上，距离桌边 10cm，其他物体放置在与桌边平行的 10cm 水平线上。正常启动机器人，由一名测试人员引导机器人标点“水”后，等待机器人停止跟随并发出“噔噔噔”的准备录音提示后，每次调整其他物品距塑料瓶距离如[输入]，并用标准普通话对机器人说出[输入]中的语句，其余人员观察并记录机载电脑控制台，机器人语音播报的输出信息及机器人行为。

- 测试用例标识：KL-C-G-3。
- 测试针对性：测试机器人能否到达指定航点，抓取对应物品并返回交给测试人员，多次改变代表关键词“水”的物品的形态、颜色以及内容物。
- 输入：测试人员标准普通话语音：“给我拿一瓶水。”

代表水的物体距桌边距离：10cm，其形态、颜色和内容物如下表所示：

表 8 测试道具准备

编号	颜色	形态	内容物
1	黄色	正常 500ml 饮料瓶大小	空
2	透明（带绿色标签）	正常 500ml 饮料瓶大小	空
3	透明（无标签）	正常 500ml 饮料瓶大小	空
4	透明（无标签）	正常 500ml 饮料瓶大小	有水
5	透明（带绿色标签）	大型 1.5L 饮料瓶大小	空

➤ 预期的输出：对每一次距离调整后可能的输出情况有：

- 1) 机器人语音播报：“I got it. I am coming back.”，机器人成功抓取目标物体并返回递给测试人员；
- 2) 同上一测试用例的输出 2)~5)。

➤ 评价准则：输出 1)代表测试成功，其他情况代表失败。

测试流程：将待抓取物放置在 80cm 高的桌面上，距离桌边 10cm。正常启动机器人，由一名测试人员引导机器人标点“水”后，等待机器人停止跟随并发出“噔噔噔”的准备录音提示后，每次更换待抓取物体如[输入]，并用标准普通话对机器人说出[输入]中的语句，其余人员观察并记录机载电脑控制台，机器人语音播报的输出信息及机器人行为。

## 4.10 语音

语音模块测试总述：

- 对应的需求：对应系统功能需求中【4】和【9】的语音控制。机器人能够听取并正确识别测试人员发出的中文语音，从中提取出预先设定的关键词句，并进一步响应用户请求。
- 条件或状态：机器人正常启动后，周围环境安静，可清晰听到测试人员说话。预先已在代码中录入关键词：“开始”、“水”、“奶茶”、“咖啡”。

测试用例分述：

- 测试用例标识：KL-C-V-1。(case-voice)



- 测试针对性：测试机器人能否将男性语音转换为对应字符串，并从中捕获关键词（如果存在）。
- 输入：男性测试人员标准普通话语音：
  - 1) “请给我拿一杯奶茶。”
  - 2) “水”
  - 3) “我喜欢喝冰糖雪梨。”
  - 4) “美酒加咖啡。”
- 预期的输出：
  - 1) 机载电脑控制台依次出现如下信息：
    - 1) [识别结果]：请给我拿一杯奶茶  
[WARN][时间]: ----- Keyword = 奶茶 -----
    - 2) [识别结果]：水  
[WARN][时间]: ----- Keyword = 水 -----
    - 3) [识别结果]：我喜欢喝冰糖雪梨  
[WARN][时间]: ----- Keyword = -----
    - 4) [识别结果]：美酒加咖啡  
[WARN][时间]: ----- Keyword = 咖啡 -----
  - 2) 机载电脑控制台只能显示部分以上信息，无法完整获取语音内容；
  - 3) 机载电脑控制台可以显示以上[识别结果]，但输出关键词为空；
  - 4) 机载电脑控制台无对应输出信息，即[识别结果]为空。
- 评价准则：如出现情况一中情形，则测试正确通过；出现情况二中情形，则测试部分通过，可能原因有环境干扰或语音发出时间不当；出现情况三中或测试四情形，则测试不通过。
- 测试流程：正常启动机器人，按下运动锁，由两名男性测试人员站在机器人前，等待机器人发出“噔噔噔”的准备录音提示后，用标准普通话对机器人交替说出[输入]中的语句，其余人员观察并记录机载电脑控制台输出信息。
- 测试用例标识：KL-C-V-2。

- 测试针对性：测试机器人能否将女性语音转换为对应字符串，并从中捕获关键词（如果存在）。
- 输入：女性测试人员标准普通话语音：
  - 1) “开始。”
  - 2) “我在测试机器人的语音功能。”
  - 3) “我喜欢喝冰糖雪梨。”
  - 4) “美酒加咖啡。”
- 预期的输出：
  - 1) 机载电脑控制台依次出现如下信息：
    - [识别结果]：开始  
[ WARN] [时间]: ----- Keyword = 开始 -----
    - [识别结果]：我在测试机器人的语音功能  
[ WARN] [时间]: ----- Keyword = -----
    - [识别结果]：我喜欢喝冰糖雪梨  
[ WARN] [时间]: ----- Keyword = -----
    - [识别结果]：美酒加咖啡  
[ WARN] [时间]: ----- Keyword = 咖啡 -----
  - 2) 机载电脑控制台只能显示部分以上信息，无法完整获取语音内容；
  - 3) 机载电脑控制台可以显示以上[识别结果]，但输出关键词为空；
  - 4) 机载电脑控制台无对应输出信息，即[识别结果]为空。
- 评价准则：如出现情况一中情形，则测试正确通过；出现情况二中情形，则测试部分通过，可能原因有环境干扰或语音发出时间不当；出现情况三中或测试四情形，则测试不通过。
- 测试流程：正常启动机器人，按下运动锁，由两名女性测试人员站在机器人前，等待机器人发出“噎噎噎”的准备录音提示后，用标准普通话对机器人交替说出[输入]中的语句，其余人员观察并记录机载电脑控制台输出信息。
- 测试用例标识：KL-C-V-3。

- 测试针对性: 测试机器人能否从识别到的语音中捕获关键词(如果存在)。
- 输入: 测试人员标准普通话语音:
  - 1) “开始”
  - 2) “水”
  - 3) “奶茶”
  - 4) “咖啡” (以上四句为单独一个关键词)
  - 5) “给我拿一杯水。” (关键词在句末)
  - 6) “奶茶真好喝。” (关键词在句首)
  - 7) “有一杯咖啡放在桌子上。” (关键词在句中)
  - 8) “给我拿杯雪碧。” (不包含关键词)
  - 9) “我想喝奶茶和咖啡。”
  - 10) “我想喝咖啡和奶茶。” (以上两句为同时包含两个关键词)
- 预期的输出:
  - 1) 机载电脑控制台依次出现如下信息:
    - [识别结果]: 开始  
[ WARN] [时间]: ----- Keyword = 开始 -----
    - [识别结果]: 水  
[ WARN] [时间]: ----- Keyword = 水 -----
    - [识别结果]: 奶茶  
[ WARN] [时间]: ----- Keyword = 奶茶 -----
    - [识别结果]: 咖啡  
[ WARN] [时间]: ----- Keyword = 咖啡 -----
    - [识别结果]: 给我一杯水  
[ WARN] [时间]: ----- Keyword = 水 -----
    - [识别结果]: 奶茶真好喝  
[ WARN] [时间]: ----- Keyword = 奶茶 -----
    - [识别结果]: 有一杯咖啡放在桌子上  
[ WARN] [时间]: ----- Keyword = 咖啡-----
    - [识别结果]: 给我拿杯雪碧

[ WARN] [时间]: ----- Keyword = -----

■ [识别结果]: 我想喝奶茶和咖啡

[ WARN] [时间]: ----- Keyword = 奶茶 -----

■ [识别结果]: 我想喝咖啡和奶茶

[ WARN] [时间]: ----- Keyword = 奶茶 -----

2) 单独一个关键词以及关键词位于句首和句末时[识别结果]不完整, 输出关键词为空;

3) 最后两条语句[识别结果]正确, 但关键词为“咖啡”;

4) 最后两条语句[识别结果]正确, 但第一句关键词为“奶茶”, 第二句为“咖啡”;

5) 机载电脑控制台可以显示以上[识别结果], 但输出关键词为空。

➤ 评价准则: 如出现 1)中情形, 则测试正确通过; 出现 2)中情形, 则测试部分通过, 可能原因有环境干扰或语音发出时间不当; 出现 3)或 4)中情形, 则关键词查找逻辑与预期不符; 出现 5)中情形, 则测试不通过。

➤ 测试流程: 正常启动机器人, 按下运动锁, 由一名测试人员站在机器人前, 等待机器人发出“噚噚噚”的准备录音提示后, 用标准普通话对机器人说出[输入]中的语句, 其余人员观察并记录机载电脑控制台输出信息。

➤ 测试用例标识: KL-C-V-4。

➤ 测试针对性: 测试机器人在标点阶段能否对测试人员的语音指令做出正确的语音应答。

➤ 输入: 测试人员标准普通话语音:

1) “这里有水。” (包含关键词)

2) “这是一瓶雪碧。” (不包含关键词)

3) “这是水, 别跟了。”

4) “别跟了, 这是水。” (以上两句为关键词与停止跟随指令同时出现, 其间需重新设置机器人回到标点状态)

➤ 预期的输出:

1) 机载电脑控制台依次出现如下信息：

■ [识别结果]：这里有水

[ WARN] [时间]: ----- Keyword = 水 -----

■ [识别结果]：这是一瓶雪碧

[ WARN] [时间]: ----- Keyword = -----

■ [识别结果]：这是水，别跟了

[ WARN] [时间]: ----- Keyword = 水 -----

■ [识别结果]：别跟了，这是水

[ WARN] [时间]: ----- Keyword = 水 -----

机器人发出语音播报：“OK. I have memoried. Next one , please.”

机器人发出语音播报：“OK. I have memoried. Next one , please. OK. What do you want me to fetch?”

机器人发出语音播报：“OK. I have memoried. Next one , please. OK. What do you want me to fetch?”

2) 机载电脑控制台能显示以上信息，但机器人无语音播报；

3) 机载电脑控制台能显示以上信息，前两条语音执行结果同 1)，后机器人只发出语音播报：“OK. I have memoried. Next one , please.”

4) 机载电脑控制台能显示以上信息，前两条语音执行结果同 1)，后机器人只发出语音播报：“What do you want me to fetch?”

5) 机载电脑控制台无显示信息，机器人无语音播报。

➤ 评价准则：如出现 1)中情形，则测试正确通过；出现 2)中情形，则测试不通过；出现 3)或 4)中情形，则机器人指令处理逻辑与预期不符，测试部分通过；出现 5)中情形，可能原因有环境干扰或语音发出时间不当，应回溯到用例 XX，测试不通过。

➤ 测试流程：正常启动机器人，由一名测试人员站在机器人前，等待机器人开始跟随测试人员并发出“噎噎噎”的准备录音提示后，用标准普通话对机器人说出[输入]中的语句，其余人员观察并记录机载电脑控制台及机器人语音播报的输出信息。

- 测试用例标识：KL-C-V-5。
- 测试针对性：测试机器人在抓取阶段能否对测试人员的语音指令做出正确的语音应答。
- 输入：测试人员标准普通话语音：
  - 1) “给我拿瓶水。”（包含关键词）
  - 2) “给我拿一瓶雪碧。”（不包含关键词）
- 预期的输出：
  - 1) 机载电脑控制台依次出现如下信息：
    - [识别结果]：给我拿瓶水  
[ WARN] [时间]: ----- Keyword = 水 -----
    - [识别结果]：给我拿一瓶雪碧  
[ WARN] [时间]: ----- Keyword = -----机器人发出语音播报：“OK. I will go to get it for you.”
  - 2) 机载电脑控制台能显示以上信息，但机器人无语音播报；
  - 3) 机载电脑控制台无显示信息，机器人无语音播报。
- 评价准则：如出现 1)中情形，则测试正确通过；出现 2)中情形，则测试不通过；出现 3)中情形，可能原因有环境干扰或语音发出时间不当，应回溯到用例 XX，测试不通过。
- 测试流程：正常启动机器人，由一名测试人员引导机器人标点“水”后，等待机器人停止跟随并发出“噚噚噚”的准备录音提示后，用标准普通话对机器人说出[输入]中的语句，其余人员观察并记录机载电脑控制台及机器人语音播报的输出信息。

## 5. 测试结果

### 5.1 本地启动

- 测试用例标识：KL-C-L-1
- 实际输出：
  - 1) 第一次测试：会弹出 Rviz 界面，但界面显示没有地图，机器人的

初始位置生成一个“start”航点。控制台报错雷达和 Kinect 无法正常连接。

2) 第二次测试：会弹出 Rviz 界面，但界面显示没有地图，机器人的初始位置生成一个“start”航点。控制台不报错。

3) 第三次测试：会弹出 Rviz 界面，可以看到建图功能开启，同时在机器人的初始位置生成一个“start”航点。

➤ 完成状态：遇到问题，与要求有偏差

➤ 问题详述：

1) 机器人节点正常启动，但无法连接到硬件，疑似硬件未启动或是连接异常。【问题标识：KL-P-L-1】；

2) 机器人节点正常启动，硬件连接正常，但建图功能失败，疑似雷达出现问题。【问题标识：KL-P-L-2】；

## 5.2 网页启动

➤ 测试用例标识：KL-C-W-1

➤ 实际输出：

1) 第一次测试：机载电脑上节点成功启动，终端显示机载电脑无线网卡 IP 地址，显示“waiting for connecting...”主界面网页打开成功，机器人控制界面打开成功，机器人控制界面显示等待连接，变为连接失败。

2) 第二次测试：机载电脑上节点成功启动，终端显示机载电脑无线网卡 IP 地址，显示“waiting for connecting...”个人电脑主界面网页打开成功，机器人控制界面打开成功，机器人控制界面显示等待连接，变为连接成功。机载电脑上显示“connected from:”，显示个人电脑的 IP 及端口，显示"handshake success"，显示“new thread for client ...”。

➤ 完成状态：遇到问题，与要求有偏差

➤ 问题详述：

1) 机载电脑正常启动服务器端程序，个人电脑网页成功打开，但未连接成功，疑似网络问题。【问题标识：KL-P-W-1】；

### 5.3 通信

- 测试用例标识: KL-G-T-1
- 实际输出:
  - 1) 第一次测试: 网页上实时显示机器人的状态信息;
- 完成状态: 所预期的

### 5.4 跟随测试结果

- 测试用例标识: KL-C-F-1
  - 实际输出:
    - 1) 第一次测试: 机器人跟随测试人员过远。
    - 2) 第二次测试: 机器人向前移动 0.4m, 执行成功。
  - 完成状态: 执行成功。
  - 问题详述: 机器人跟随测试人员过远, 导致机器人失去跟随焦点目标, 跟随测试失败。【问题标识: KL-P-F-1】
- 
- 测试用例标识: KL-C-F-2
  - 实际输出:
    - 1) 第一次测试: 机器人跟随测试人员过近。
    - 2) 第二次测试: 机器人向后移动 0.4m, 执行成功。
  - 完成状态: 执行成功。
  - 问题详述: 机器人跟随测试人员过近, 导致容易发生碰撞危险【问题标识: KL-P-F-2】
- 
- 测试用例标识: KL-C-F-3
  - 实际输出: 机器人向左移动 0.4m
  - 完成状态: 执行成功。
- 
- 测试用例标识: KL-C-F-4



- 实际输出：机器人向右移动 0.4m
- 完成状态：执行成功。
  
- 测试用例标识：KL-C-F-5
- 实际输出：机器人向左旋转错误!未找到引用源。rad
- 完成状态：执行成功。
  
- 测试用例标识：KL-C-F-6
- 实际输出：机器人向右旋转错误!未找到引用源。rad 。
- 完成状态：执行成功。

## 5.5 路径规划测试结果

- 测试用例标识：KL-C-P-1
- 实际输出：机器人沿一条从起始点指向目的地的直线前进 5m 到达目的地。
- 完成状态：执行成功。

## 5.6 建图测试结果

- 测试用例标识：KL-C-B-1
- 实际输出：建图完成后，在~/catkin\_ws/src/team103/maps/目录下应生成两个文件，分别为 map.pgm 和 map.yaml。
- 完成状态：执行成功。

## 5.7 标点测试结果

- 测试用例标识：KL-C-M-1

- 实际输出：机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。
  - 完成状态：执行成功。
- 
- 测试用例标识：KL-C-M-2
  - 实际输出：机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。
  - 完成状态：执行成功。
- 
- 测试用例标识：KL-C-M-3
  - 实际输出：机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。
  - 完成状态：执行成功。
- 
- 测试用例标识：KL-C-M-4
  - 实际输出：机器人无任何响应。
  - 完成状态：执行成功。
- 
- 测试用例标识：KL-C-M-5
  - 实际输出：机器人语音输出” OK. I have memoried. Next one , please”，控制台输出识别到的关键词。并覆盖掉先前已标记的航点位置。
  - 完成状态：执行成功。
- 
- 测试用例标识：KL-C-M-6
  - 实际输出：机器人回到初始点。
  - 完成状态：执行成功。
- 
- 测试用例标识：KL-C-M-7
  - 实际输出：机器人回到初始点。

- 完成状态：执行成功。
- 测试用例标识：KL-C-M-8
- 实际输出：机器人不产生任何反馈。
- 完成状态：执行成功。

## 5.8 抓取测试结果

- 测试用例标识：KL-C-G-1
- 实际输出：
  - 4) 第一次测试：机器人成功前往待抓取物体所在航点。到达航点约 3s 后机器人语音播报：“I failed to grab the item. I am coming back.”并返回出发地，抓取失败；
  - 5) 第二次测试：机器人成功前往待抓取物体所在航点，机械臂向前向下旋转约  $\pi/5$  后向上升起一段距离，机械爪合并，而后语音播报：“I got it. I am coming back.”，返回出发地，抓取失败；
  - 6) 第三次测试：机器人成功前往待抓取物体所在航点，机械臂向前向下旋转约  $\pi/2$  后向上升起一段距离，但机械臂仍低于桌面时机器人已开始前进，机械臂几乎碰上桌沿，小组人员及时按下急停锁，抓取失败；
  - 7) 第四次测试：当待抓取物体距桌边距离为 0cm、5cm、10cm、15cm、20cm、25cm 时，输出同预期输出 1)，抓取成功。
- 完成状态：遇到问题，与要求有偏差
- 问题详述：
  - 3) 机器人到达航点后直接返回，抓取进程未被启动，疑似代码逻辑出现相关问题【问题标识：KL-P-G-1】；
  - 4) 机器人到达航点后虽发起抓取尝试，但机械臂运转异常，而机器人的语音播报说明系统自身并未发现该异常情况【问题标识：KL-P-G-2】；
  - 5) 机器人到达航点后虽发起抓取尝试，且机械臂运转流程正常，但抓取高度判断有误，疑似抓取参数设置不正确【问题标识：KL-P-G-3】；
  - 6) 机器人在物体距桌边距离  $\leq 25\text{cm}$  时可以正确完成抓取，但超过该距

离将导致抓取失败【问题标识：KL-P-G-4】。

➤ 测试用例标识：KL-C-G-2

➤ 实际输出：

1) 第一次测试：当待抓取物体距其他物体 5cm 时，机器人将旁边水瓶错误识别为指定物体并取回，抓取失败，其余情况输出同预期输出 1)，抓取成功；

2) 第二次测试：当待抓取物体距其他物体 5cm 时，机器人将旁边水瓶碰倒，抓取成功，当待抓取物体距其他物体 10cm 时，机器人将旁边水瓶错误识别为指定物体并取回，抓取失败，其余情况输出同预期输出 1)，抓取成功；

➤ 完成状态：与要求有偏差

➤ 问题详述：

机器人在待抓取物体距其他物体 $\leq 10\text{cm}$ 时可能将非指定物体抓回，或者将其他物体碰倒，总体上说性能不稳定，超过该距离抓取基本成功【问题标识：KL-P-G-5】。

➤ 测试用例标识：KL-C-G-3

➤ 实际输出：

3) 第一次测试：当待抓取物体为道具【3】【4】时，控制台显示信息：“[WARN] [时间]: [FIND\_OBJ] nObjCnt  $\leq 0$  No object detected ...”后返回，抓取失败，其余情况输出同预期输出 1)，抓取成功；

4) 第二次测试：当待抓取物体为【3】和【4】时，控制台显示信息：“[WARN] [时间]: [FIND\_OBJ] nObjCnt  $\leq 0$  No object detected ...”后返回，抓取失败，其余情况输出同预期输出 1)，抓取成功；

➤ 完成状态：与要求有偏差

➤ 问题详述：

机器人在待抓取物体为通体透明时可能抓取失败，总体上说性能不稳定，的物体抓取基本成功【问题标识：KL-P-G-6】。

## 5.9 语音测试结果

➤ 测试用例标识: KL-C-V-1

➤ 实际输出:

1) 第一次测试: 机器人没有发出“噎噎噎”的准备录音提示, 但控制台有准备录音的文字信息。测试人员直接开始语音, 控制台获得如下输出:

■ [识别结果]: 我拿一杯奶茶

[ WARN] [时间]: ----- Keyword = 奶茶 -----

■ [识别结果]: 水

[ WARN] [时间]: ----- Keyword = 水 -----

■ [识别结果]: 雪梨

[ WARN] [时间]: ----- Keyword = -----

■ [识别结果]: 美酒加咖啡

[ WARN] [时间]: ----- Keyword = 咖啡 -----

2) 第二次测试: 同预期输出 1)

➤ 完成状态: 与要求有偏差

➤ 问题详述: 由于系统没有“噎噎噎”提示音, 导致测试人员说话时机与系统录音时机不吻合, 所获结果不完全。【问题标识: KL-P-V-1】

➤ 测试用例标识: KL-C-V-2

➤ 实际输出: 同预期输出 1)

➤ 完成状态: 所预期的

➤ 测试用例标识: KL-C-V-3

➤ 实际输出: 同预期输出 1)

➤ 完成状态: 所预期的

➤ 测试用例标识: KL-C-V-4

➤ 实际输出: 同预期输出 1)

➤ 完成状态: 所预期的

- 测试用例标识: KL-C-V-5
- 实际输出: 同预期输出 1)
- 完成状态: 所预期的

## 6. 测试结果分析

### 6.1 对被测试软件的总体评估

- 问题标识: KL-P-G-1
  - 原因分析: 机器人抓取异常处理部分代码逻辑错误, 将连续的抓取成功判断循环提前切断, 导致抓取进程刚刚开启, 还未检测到平面和物体就被迫中止。
  - 解决方案: 对“\team103\src\exception\_handler.cpp”中相关逻辑进行修改, 不再提前终止判断循环。
  - 影响分析: 本错误为系统逻辑错误, 直接导致抓取功能不能使用, 必须完全修复。该错误的修复涉及到对系统异常处理部分设计的重新构思, 但不影响其余正常功能的实现。
- 
- 问题标识: KL-P-G-2
  - 原因分析: 机器人机械臂导线方向插反, 导致机械臂运动异常。
  - 解决方案: 重新反向连接机械臂导线。
  - 影响分析: 本错误为系统硬件组装错误, 导致抓取功能异常。该错误的修复不涉及对系统软件设计和逻辑的修改, 不影响其余正常功能的实现。
- 
- 问题标识: KL-P-G-3
  - 原因分析: ROS 环境中抓取相关服务包“wpb\_home\wpb\_home\_bringup”中的抓取参数“.\config\wpb\_home.yaml”文件中原本参数设置为“grab\_y\_offset: 0.0; grab\_lift\_offset: 0.0; grab\_forward\_offset: 0.0”不符合抓取的实际情况。

- 解决方案：对“wpb\_home\wpb\_home\_bringup\config\wpb\_home.yaml”中相关参数进行修改，修改后的参数设置为“grab\_y\_offset: 0.0; grab\_lift\_offset: 0.3; grab\_forward\_offset: 0.15”。
- 影响分析：本错误为参数调整不当，导致抓取功能出现严重误差，必须完全修复。该错误的修复不涉及对系统软件设计和逻辑的修改，不影响其余正常功能的实现。
  
- 问题标识：KL-P-G-4
- 原因分析：物体距桌边距离在一定程度上影响抓取效果，可能与点云相关计算有关。
- 影响分析：本误差为系统非功能性需求极限，由于在距桌面 cm 范围内抓取功能正常，因此建议使用时待抓取物体应保持在这一范围内为宜。
  
- 问题标识：KL-P-G-5
- 原因分析：物体距其他物体距离在一定程度上影响抓取效果，可能与点云相关计算有关。
- 影响分析：本误差为系统非功能性需求极限，由于在距其他物体 15cm 范围内抓取功能正常，因此建议使用时待抓取物体之间应保持在这一范围内为宜。同时，本系统设计初衷为抓取某一航点上的物体，原则上所有放置在该范围内的物体都可以算作该航点上的物体。
  
- 问题标识：KL-P-G-6
- 原因分析：物体形态颜色在一定程度上影响抓取效果，可能与点云相关计算有关。
- 影响分析：本误差为系统非功能性需求极限，由于在物体为 时抓取功能正常，因此建议使用时待抓取物体应尽量选择此类物体。
  
- 问题标识：KL-P-V-1
- 原因分析：机器人语音识别相关开发包“xfyun\_waterplus”可能发生一

定程度的损坏，导致机器人无法发出“噚噚噚”的录音准备提示。没有提示音的情况下，测试人员（及用户）无法判断下达指令的准确时机，于是导致录音前半部分缺失。

- 解决方案：重新下载安装 ROS 环境中的“xfyun\_waterplus”。
- 影响分析：“噚噚噚”录音准备提示并非录音必需条件，但是对于用户使用机器人的便捷性有很大程度影响，且贯穿其余测试流程。该问题的修复办法并未涉及改变系统程序逻辑和设计。

- 问题标识：KL-P-F-1

- 原因分析：设置的跟随距离参数过远，导致红外雷达无法检测到测试人员的位置。
- 解决方案：减小跟随距离参数范围至 0.5m 以下。
- 影响分析：跟随过远导致失去跟随目标可能导致不能正常完成标点等后续功能，因此有必要修复该缺陷。

- 问题标识：KL-P-F-2

- 原因分析：设置的跟随距离参数过小，导致机器人紧贴用户。
- 解决方案：增大跟随距离参数范围至 0.3m 以上。
- 影响分析：跟随过近将造成机器人与用户相撞，导致危险。因此有必要修复该缺陷。

- 问题标识：KL-P-L-1

- 原因分析：机器人开关未打开，使得 Kinect、雷达处于关闭状态。
- 解决方案：按下底座开关。
- 影响分析：未打开底座开关使得所有的硬件设备均无法连接到机载电脑，无法进行测试。修正此问题是所有测试的基石。

- 问题标识：KL-P-L-2



- 原因分析：控制台未报错说明各个硬件设备均无损害，经过观察 Rviz 界面左侧雷达部分显示状态异常，猜测是因为 QSize 太小使得信息无法完全显示。
- 解决方案：将左侧 QSize 改为 200。
- 影响分析：QSize 太小使得 Rviz 中无法显示地图，因此产生阻塞，机器人将停滞在原地，无法按需执行正常功能。修正后并未涉及改变系统程序逻辑和设计。
  
- 问题标识：KL-P-W-1
- 原因分析：进一步测试发现机载电脑和个人电脑无法 ping 通，此时两者都连接到了 BUAA-WiFi 下。可能是连接到的主机不同，或是校园网对主机连接进行了一些限制。
- 解决方案：将两台电脑连接到热点上。
- 影响分析：无法连接成功将会影响通信、紧急制动等通过网页展示的功能，影响用户使用机器人的体验，带来许多不方便。修正后并未涉及改变系统程序逻辑和设计，只是需要规定系统使用的网络环境。

## 6.2 测试环境的影响

本产品实际应用场景为超市，主要功能为辅助顾客获取指定商品，以上测试并未在超市开展。本节将对超市环境与测试环境的差异进行一系列评估和对差异带来的影响进行分析。

### 1) 差异一：更广大的运行空间

- 差异评估：测试环境为 7m×8m 的矩形封闭房间，而超市环境将远远大于这一范围，约为测试空间的 5~10 倍大小。
- 影响分析：机器人可在小场景中行进，证明其运动方式、速度指标不存在问题，因此在更广大的场景中也不致出现基本运动问题。

### 2) 差异二：更复杂的避障场景

- 差异评估：测试环境中设置了两张桌子作为额外障碍物，而超市场景中可能有多组货架，并且有来回穿梭的顾客和员工。

- 影响分析: 测试用例 KL-C-P 系列和 KL-C-A 系列证明了机器人在小范围内路径规划与紧急避障不存在问题, 将此功能迁移到复杂场景中可能有一定风险。但从将大场景划分为多局部区域的角度来看, 某一时刻机器人其实是运行在一个特定的局部场景中, 只要该局部与我们测试场景物体放置密度在同一水平, 运行就不会出现障碍。

#### 3) 差异三: 更嘈杂的背景噪声

- 差异评估: 考虑到超市环境属于人员众多的公共场合, 其场景中的噪声分贝要高于测试环境。
- 影响分析: 由于机器人主要靠语音获得用户指令, 嘈杂的背景音效很可能干扰语音的识别。但在 KL-C-V 系列测试的实际执行过程中, 也曾出现背景嘈杂的情形, 此类情况下机器人基本能够捕获关键词, 由此推测如顾客靠近机器人发出指令, 则影响不致很大。

#### 4) 差异四: 更丰富的货物摆放

- 差异评估: 超市货架中货物摆放密集, 品种繁多, 且不同种类商品之间相隔很近, 与测试环境中按照一定距离隔开摆放的优先种类待抓取物体有很大差异。
- 影响分析: 目前机器人对不同物体的识别只依靠位置标定, 所以精度不高, 且由 KL-C-G-2 的测试结果可知, 机器人只有在物体之间有一定间隔时才能稳定抓取, 这给实际应用带来较大困难。如未来加入更精准地对物品分类的识别方式, 或可改进此类抓取缺陷。

## 6.3 改进建议

- 1) 需求上: 实际超市应用场景中, 背景音音量较大, 对于语音标点, 语音下达抓取命令两方面, 有一定可能出现机器人无法捕获到正确信息的情况, 因此可以采取多种方式下达指令。例如: 通过网页输入文本来标定地点; 通过网页点击对应物体的按钮来实现抓取。若要实现该功能, 需要在网页上加入表单, 按钮等控件, 需要在服务器端加入处理相应消息

的控制代码，需要在脚本 `main` 函数中更改程序流程，使得标点和抓取信息的获取不仅仅依赖于语音回调函数。

- 2) 设计上：实际超市应用场景中，物体摆放较为密集，一个区域摆放多个物体。而当前的软件系统没有融入不同物体间识别，容易出现错抓、漏抓的情况。可以在当前的基础上加入目标检测框架，使得能够识别指定货物，精准抓取。若要实现该功能，需要采集超市货物的图像，训练出一个能够对其进行检测和分类的模型。在脚本中加入调用该模型的接口，改变抓取模块的代码，加入识别判断的内容。为了实现此内容可能需要将待抓取的货物信息作为全局变量传递。
- 3) 实现上：可以选择两个更有应用价值的方向：具有更高性价比或具有更好的性能。前者需要采用更为廉价的硬件设备，若要采用此建议需要抓取、语音等模块进行部分重写，工程量大。后者选择更好的硬件设施，如更好的摄像头，更好的激光雷达，若要实现此也需要改变代码中相关数据的处理算法。
- 4) 测试上：在本次测试过程中，出现不符合预期的错误情况时测试人员应该及时记录。每次修复问题时都应该记录 `commit` 的编号，以追溯被改变的内容。这样才能对系统的整体变革有一个清晰的认识，便于团队在某位成员不在时，也能找到他改变的代码，方便版本管理。