2025 机器人软件工程学课程作业报告

项目名称：盲人引导移动机器人

英文名称：Blind Guide Mobile Robot

姓名：2213530 张禹豪

小组成员：

2213530 张禹豪

2211449 张涛

1911242 庞彦威

完成时间：2025 年 6 月

摘 要

本文设计并实现了一种基于多模态感知的智能机器人导航系统，通过融合语音识别、视觉感知与强化学习决策，实现了机器人在动态环境中的自主导航与任务执行。系统采用模块化分层架构：在感知层，集成火山引擎语音识别模块和OpenCV 视觉处理模块，分别实现语音指令的实时转换及视频流的目标检测；在决策层，基于智谱 AI 构建中心节点，通过多模态信息融合生成导航指令、机械臂控制信号及视觉反馈命令；在执行层，采用双向通信机制将指令分发至运动控制与机械臂模块。实验表明，该系统在室内环境下可实现较好的语音指令响应，视觉目标识别准确率达到 90%以上，且通过动态任务优先级调度算法，多任务并发执行成功率得到极大提升。本研究为服务机器人的多模态协同控制提供了可复用的技术框架。

关键词：多模态交互；机器人导航；语音识别；视觉伺服。

目 录

一、项目背景与相关研究

1.1 项目背景

根据中国盲人协会的相关数据，我国约有 1700 万盲人，这意味着每一百人中就有至少一位视障者。他们在日常生活和出行时往往面临着诸多不便，与庞大的盲人群体相对的是，我国现役导盲犬仅有四百余只，也就是平均每四万视障人士才配有一只导盲犬，且导盲犬的训练成本非常高昂，许多公共场所是否该允许导盲犬进入也仍在讨论中。因此，在这一时代背景下。关于高质量，高效率的盲人服务型引导机器人的研究非常有必要。

1.2 相关研究

（1）技术发展现状

多模态感知与定位技术当前导盲机器人普遍采用多传感器融合方案，包括激光雷达、超声波、视觉摄像头等，实现环境三维建模与精准定位。例如，上海交通大学高峰团队研发的六足导盲机器人[2]通过雷达-惯性里程计系统与滑动窗口算法，将定位误差控制在厘米级。此外，深度学习技术的应用（如端到端语音识别模型）使语音交互准确率达 90%以上，响应时间<1 秒。

路径规划与动态避障主流算法包括改进 A*算法、模型预测控制（MPC）和强化学习。六足导盲机器人通过实时滚动优化与多约束步态规划，可适应台阶、楼梯等复杂地形，动态避障成功率超 96%。轮足式设计则结合了轮式移动的效率和足式结构的适应性，显著提升地形通过性。

人机交互创新听觉、触觉与力觉交互正逐渐成为该领域的研究焦点。例如，力觉盲杖可传递牵引力和转向力矩，实现速度动态调整；语音交互系统支持指令下发与环境状态反馈，形成双向闭环。

（2）应用场景与市场进展

1. 实际应用案例

公共导航：导盲机器人已在机场、地铁站等场景测试，提供实时路径指引与红绿灯识别。

居家陪护：部分机型集成应急处理与互联网服务，实现远程监控和家庭环境

适应。

2. 产业化与专利布局

我国已取得《导盲机器人系统》等多项专利，上海交大、清华大学等机构推动技术落地。极擎科技等企业计划量产智能伴生机器人，成本有望低于传统导盲犬。

二、项目整体介绍

2.1 整体功能和框架（图）

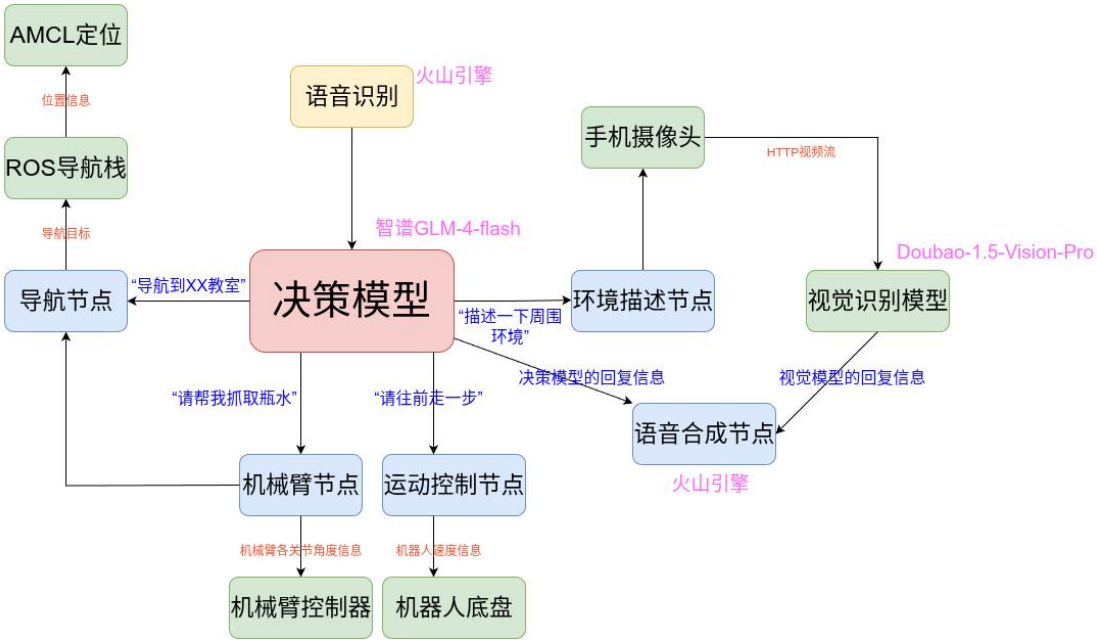

图 1.1 整体功能与框架示例

这张图展示了本盲人引导机器人系统的整体功能和架构，系统主要围绕“决策模型”为核心，整合了语音、视觉、导航与机械臂等多个模块，形成一个多模态交互、智能响应的引导系统。

◎ 语音交互模块

用户通过语音与机器人进行交互，语音识别由火山引擎完成，识别出的命令发送给“决策模型”，决策模型由 GLM-4-flash 模型承担。决策模型结合当前任务与环境状态，生成合适的响应或指令。回应信息通过“语音合成节点”生成语音反馈给用户，实现自然语言对话。

◎ 视觉与环境感知模块

我们使用手机摄像头采集视频流，使用 OpenCV 去接收手机发送的 HTTP 视频流，当接收到用户要求进行“场景识别”的指令时，该节点将在视频流中截取一张图片，通过视觉识别模型（Doubao-1.5-Vision-Pro）识别场景信息。视觉模型生成环境描述信息后，再由语音合成模块传达给用户。

◎ 导航与定位模块

系统利用 AMCL 定位和 ROS 导航栈提供室内导航功能。

用户可通过语音指令如“导航到 XX 教室”，由“导航节点”向 ROS 导航系统发送目标点信息，通过与 ROS 导航（包括 AMCL 定位和路径规划）的集成，实现房间级导航。

◎ 运动与机械臂控制模块

当用户提出如“请帮我抓取瓶水”的请求，决策模型会激活“机械臂节点”，通过对机械臂控制器发出特定关节角度信息指令，控制机械臂动作。

若用户请求“请往前走一步”，则会通过“运动控制节点”更改底盘运动速度，从而操控机器人运动。

2.2 各模块之间关系与接口方法

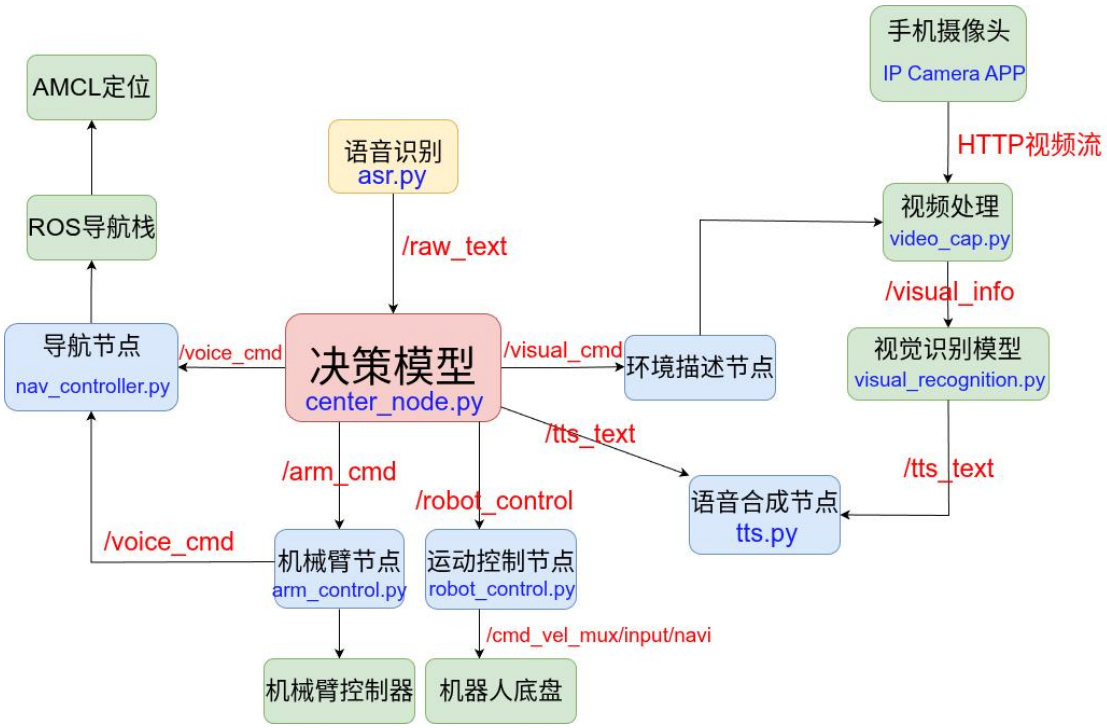

图 2.2 各模块之间关系与话题接口

该项目各模块之间连接关系如上图所示，其中“语音识别”与“决策模型”组成的语音交互系统是整个系统的核心，决策模型将对识别到的用户要求进行分析，将具体要求信息发送给指定模块进行执行。

整个系统之间的跨模块通信主要使用 ROS 话题进行异步通信（如上图红色字体所示），同时部分操作（如启动/停止语音识别）使用 ROS 服务进行同步请求/相应操作。

三、具体功能原理和实现

3.1 决策模型的原理和实现方法

这个决策模型的核心在于利用大型语言模型 (LLM)（这里具体使用的是智谱 AI 的 glm-4-flash）来理解自然语言的语音命令，并将其转化为机器人可执行的结构化命令。以下是其工作原理的详细分解：

接收语音识别文本：该决策模型节点持续接收/raw_text 中的语音识别文本信息。

LLM 解释：转录后的文本会被发送给 LLM。LLM 被提供了一个非常严格的 system_prompt（系统提示词），该提示词定义了它作为盲人引导机器人的角色，并规定了针对不同机器人功能的精确 JSON 输出格式。具体的 prompt 如下所示：

```
self.system_prompt = """你现在是一个盲人引导机器人，具有运动控制/导航/机械臂夹取/场景识别的功能，请分析用户的要求，如果你认为用户的要求属于运动控制/导航/机械臂夹取/场景识别中的一类，请按照下属 json 格式响应，请严格按以下 JSON 格式响应：
{
"content": "回复内容", # 当用户的要求为“场景识别”时，"content"保持空字符串即可，不要回复任何内容
"request": "运动控制/导航/机械臂夹取/场景识别", # 仅用户的要求为"运动控制"或者"导航"或者"机械臂夹取"或者"场景识别"时需要，选择"运动控制"或者"导航"或者"机械臂夹取"或者“场景识别”四者之一输出，其他情况设为"NONE"
"text": "前进/后退/左转/右转/停止", # 仅"request"为运动控制时需要，选择"前进"或者"后退"或者"左转"或者"右转"或者"停止"五者之一输出，其他情况设为"NONE"
"place": "116" # 仅导航时需要 # 仅"request"为导航时需要，且只有两个选项：327，302，303,且只响应数字即可，其他情况设为"NONE"
```

```
    "arm": "true"    # 仅机械臂夹取时需要，且只要"request"为机械臂夹取，"arm"即为 true，其他情况设为"NONE"
    "visual": "true"    # 仅场景识别时需要，且只要"request"为场景识别，"visual"即为 true，其他情况设为"NONE"
```

结构化命令生成：根据用户的输入和 system_prompt，LLM 会生成一个 JSON 响应。这个 JSON 包含以下信息：

content：给用户的口头回复（例如，“好的，我将前进。”）。

request：机器人动作的类型（例如，“运动控制”、“导航”、“机械臂夹取”、“场景识别”或“NONE”）。

text：如果 request 是“运动控制”，则为具体的运动命令（例如，“前进”、“后退”）。

place：如果 request 是“导航”，则为导航目的地（例如，“327”）。

arm：如果 request 是“机械臂夹取”，则为“true”。

visual：如果 request 是“场景识别”，则为“true”。

命令执行：决策模型节点会解析这个 JSON 输出

它首先将 content 字段发布在话题/tts_text，用于语音合成节点向用户生成语音反馈。

然后，它会检查 request 字段及其相关参数（text、place、arm、visual），并将特定命令发布到相应的 ROS 话题，从而触发实际的机器人动作。

例如当“request”字段为“运动控制”时，它会将“text”字段信息发布在话题/robot_command 中用于运动控制节点接收；当“request”字段为“导航”时，它会将“place”字段信息发布在话题/voice_cmd 中用于导航节点接收。

具体效果如下图所示：

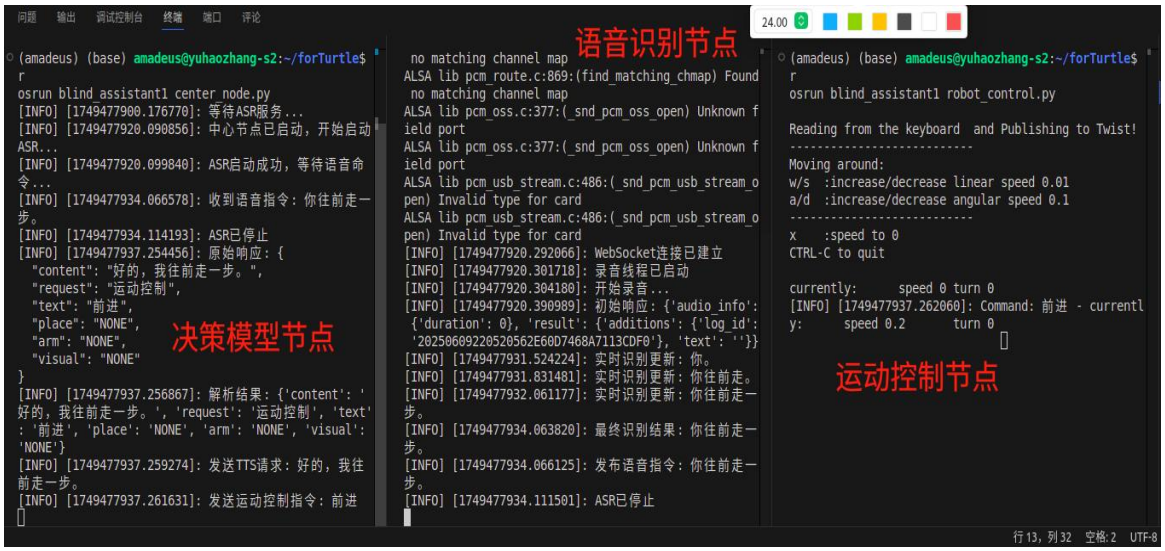

图 3.1 通过语音进行运动控制示例

语音识别开关管理：

当 LLM 处理命令和机器人进行语音回复时，ASR 服务会被暂停，以防止误识别。一旦机器人说完话（通过 tts_status 话题指示），ASR 会重新启动，允许机器人监听下一个命令。

3.2 视觉与环境感知模块的原理和实现

机器人视觉和环境感知模块其核心原理是通过 ROS 将 IP 摄像头的视频流传输到图像处理节点，并利用大模型（Doubao-1.5-Vision-Pro 多模态模型）对图像进行场景识别，然后将识别结果以文本形式发布出去。

该模块的原理可以分为以下几个步骤：

◎ 图像捕获与发布（video_cap.py）：

原理：使用 OpenCV 从 IP 摄像头捕获视频流，并将每一帧图像转换为 ROS Image 消息。

实现：图像捕获与发布节点负责从指定的 IP 摄像头 URL 读取视频帧。它将原始视频帧包装成 sensor_msgs/Image 格式。为了实现“场景识别”的按需触发，该节点还订阅了 visual_cmd 话题。当 visual_cmd 接收到“true”命令时，它会发布当前最新的一帧图像到 visual_info 话题。这意味着只有当需要进行场景识别时，图像才会被发送给识别模块，减少不必要的处理。

◎ 视觉识别处理 (visual_recognition.py)：

原理：订阅 /visual_info 话题上的图像数据。当接收到图像时，将其转换为 Base64 编码的 JPEG 格式，然后作为输入发送给多模态大模型进行分析。大模型返回一个关于图像内容的文本描述。

图像接收与编码：它使用 cv_bridge 将 ROS Image 消息转换回 OpenCV 图像格式，然后使用 cv2.imencode 将其编码为 JPEG，并进一步使用 base64 编码为 Data URI 格式的字符串。这种格式是多模态大模型通常接受的图像输入方式。

大模型调用：将编码后的图像数据和提示词（“请描述上述图片内容。要求尽量简短，缩减到在 20 字以内。”）构建成大模型 API 请求的 content。

文本描述发布：从大模型的响应中提取文本回复，然后将这个文本描述发布到 /tts_text 话题。这个 /tts_text 话题会由中心节点订阅，然后传递给 TTS 系统，实现语音播报识别结果。

实现效果如下图所示：

图 3.2 图像识别示例

3.3 定点导航的实现原理和方法

◎ 系统架构与核心思想

模块化设计代码采用分层架构，将导航任务分解为环境感知（语音/视觉输入）、决策规划（目标设定）和执行控制（move_base 交互）三个层次，符合 ROS 导航栈的模块化理念。在本次实验中，由于项目应用场景较为简单，且是室内环境，环境的稳定性相对较高，因此中心思想是通过预设地图坐标与多源输入触发实现自主导航，兼顾效率与可扩展性。

事件驱动机制通过订阅"/voice_cmd"和"/current_room"话题实现异步触发，当语音指令或视觉识别到目标房间时，自动调用路径规划，体现响应式编程思想。

◎ 导航实现原理

1. move_base 集成

动作客户端通过"SimpleActionClient"连接"move_base"动作服务器，实现目标提交与状态监控。"MoveBaseGoal"消息包含目标位姿（x, y, w），其中"w"为四元数的实部，简化了二维导航中的朝向控制。

2. 静态地图与坐标预设

本次实验中采取 slam 建图后，将得到的地图文件存放在对应位置后，采用房间位置字典的思想进行位置的预存。"room_locations"预存各房间在"map"坐标系下的位姿，避免实时计算，适合结构化环境（如固定房间布局）。"frame_id="map""确保目标位姿基于全局地图坐标系，与 SLAM 或静态地图对齐。

3. 多模态输入处理

(1) 语音控制："voice_callback"解析语音指令，直接触发导航，适用于用户交互场景。

(2) 视觉反馈：预留"room_callback"接口，可扩展为视觉定位辅助（如二维码识别房间号）。

◎ 与 ROS 导航栈的协同

1.全局与局部规划

代码仅提交目标，实际路径规划由"move_base"的全局规划器（如 A*/Dijkstra）和局部规划器（如 DWA/TEB）完成。

2.代价地图依赖

"move_base"会根据全局/局部代价地图动态避障，无需代码显式处理障碍物。

3. 状态机逻辑

代码未处理导航结果反馈，但"move_base"内部通过状态机管理规划、执行、恢复等流程，例如超时重规划或避障失败触发恢复行为。

◎ 优化与扩展方向

1. 动态适应性

(1) 实时坐标更新：可结合"amcl"定位节点，动态更新"room_locations"以适应地图偏移。

(2) 障碍物感知：通过订阅"/move_base/feedback"获取实时路径状态，增强异常处理。

2. 功能扩展

（1）多目标序列：扩展"navigate_to_room"支持路径点序列，实现巡检任务。

（2）语义导航：引入自然语言处理（如"带我去最近的卫生间"），动态解析目标位置。

总之，该代码的核心是基于事件触发的轻量级导航控制器，通过 ROS 动作库与"move_base"深度集成，将复杂的路径规划、避障等任务委托给导航栈，自身专注于高层指令解析与目标管理。其设计平衡了简洁性与功能性，适合室内服务机器人等场景。

3.4 核心语音交互与调度

在本系统中混合使用话题(Topics)与服务(Services)，话题用于传递异步数据流，如语音识别文本、TTS 播报内容等；服务用于执行同步控制命令，如启动与停止 ASR，确保指令被精确执行并获得反馈。

原理：该功能的核心是基于服务调用的精确状态机，以 center_node 为主导，解决了传统语音交互中的"声音回环"和"API 资源浪费"两大痛点。

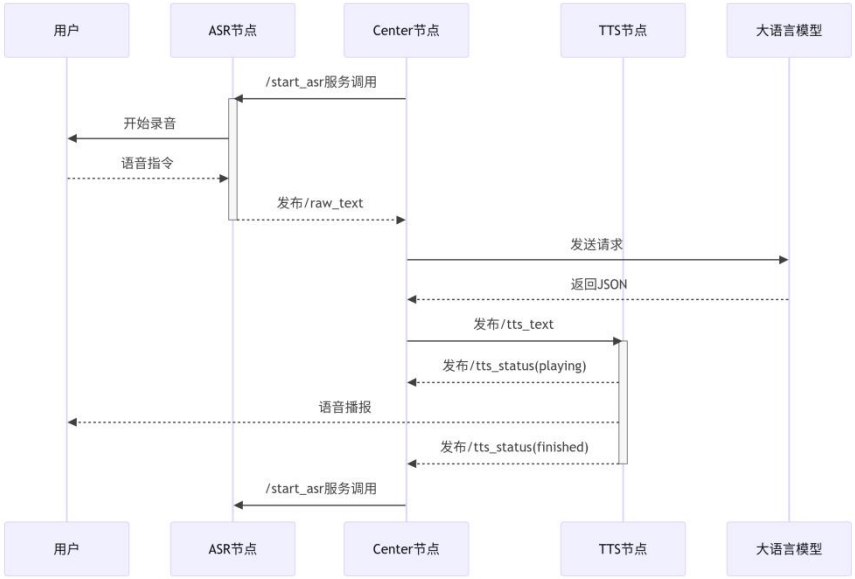

图 3.3 核心交互示意图

◎ ASR 节点 (asr.py)：

受控启动：节点默认不工作。它提供/start_asr 和/stop_asr 两个服务接口。

自动停止：在用户一句话说完（检测到静音）并成功发出识别文本后，它会自动停止录音和 API 请求，进入待命状态，等待 center_node 的下一次唤醒。

◎ 调度中心 (center_node.py)：

主导 ASR：在一次交互结束后（如 TTS 播报完毕），center_node 会调用/start_asr 服务来"唤醒"ASR。当收到 ASR 的识别结果后，它立即调用/stop_asr 服务，确保 ASR 在后续的 LLM 处理和 TTS 播报期间处于关闭状态。

意图理解与任务分发功能：即决策模型。

◎ TTS 节点 (center_node.py)：

状态反馈：这是形成闭环的关键。TTS 节点在开始播放音频时，会向/tts_status 话题发布"playing"；在播放完全结束后，会发布"finished"。

精确重启：center_node 通过监听/tts_status 话题。一旦收到"finished"消息，

就认为本次交互的"说"环节已结束，可以安全地启动 ASR，准备下一次"听"，从而形成一个完美的交互循环。

3.5 机械臂操作

该功能实现了通过语音指令控制 my_dynamixel 机械臂执行预设的抓取序列，并在动作完成后自动导航回指定位置。

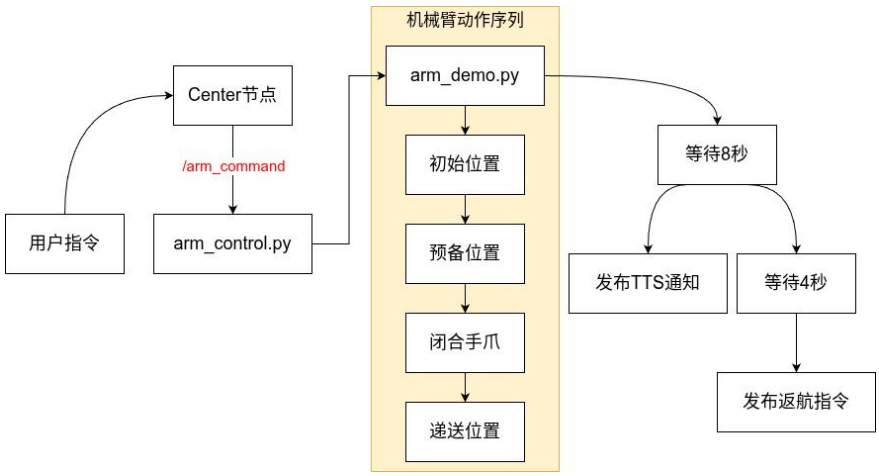

图 3.4 机械臂抓取流程图

原理：这是一个分层控制的实现。blind_assistant1 包中的 arm_control.py 作为高层控制器，负责接收 center_node 的简单指令。my_dynamixel 包中的 arm_demo.py 作为底层控制器，负责驱动机械臂硬件执行具体的、预设的动作序列。

实现：系统分为高层指令控制（arm_control.py）和底层执行（arm_demo.py）。当中心节点检测到抓取意图时，会通过发布"true"消息到/arm_command 话题触发高层控制模块，随后该模块立即向/arm_action 话题发送"start"以激活底层控制器。为确保动作完成，系统采用 8 秒固定延时等待底层执行，之后通过/tts_text 发布语音通知，并在 4 秒后向/navigation_command 发布"302"指令，启动自动返航。底层控制器 arm_demo.py 预设了多个机械臂状态（如初始、预备、夹取、递送），每个状态对应一组五个关节的目标角度。它在收到"start"信号后按序执行动作，每步动作通过向各关节控制器的话题发送目标角度进行控制，并使用 2 秒的固定延时等待每个动作完成，整个过程无反馈，属于开环控制。

四、个人完成的主要工作

4.1 负责决策模型与项目整体结构构建

作为项目核心部分的负责人，我设计并实现了整个盲人引导机器人系统的决策模型和系统架构。决策模型是本项目的"大脑"，负责将用户的语音指令转化为机器人可执行的操作。这项工作主要包括以下几个方面：

系统架构设计：

我设计了基于 ROS 的模块化系统架构，将复杂的盲人引导机器人系统拆分为语音交互、视觉感知、导航定位、运动控制和机械臂控制五个主要功能模块。这种模块化设计使各模块之间通过统一的 ROS 话题接口进行通信，实现了松耦合的系统结构，便于各模块的独立开发和测试。

具体而言，我定义了以下关键的 ROS 话题接口：

- /raw_text：语音识别结果传输
- /tts_text：需要语音合成的文本
- /robot_command：机器人运动控制指令
- /voice_cmd：导航目标点信息
- /arm_cmd：机械臂控制指令
- /visual_cmd：视觉识别触发信号
- /visual_info：视觉信息传输通道

这种话题设计确保了各功能模块能够通过明确的接口进行数据交换，同时保持足够的独立性。具体的各模块之间关系图请查阅图 2.2。

决策模型的实现：

在决策模型实现方面，我基于大型语言模型（GLM-4-flash）设计了一套专门的提示词工程解决方案，使系统能够智能理解用户意图并转化为机器人可执行的指令：

◎ 智能意图理解：我编写的提示词模板能够引导大语言模型准确识别用户意图，将其归类为"运动控制"、"导航"、"机械臂夹取"或"场景识别"四类基本功能。

◎ 结构化输出设计：我设计了严格的 JSON 输出格式，包括 content（语音回复内容）、request（请求类型）、text（运动命令）、place（导航目标点）、

arm（机械臂控制标志）和 visual（视觉识别标志）等字段，确保系统能够准确解析语言模型的输出。

◎ 多模式指令处理：针对不同类型的用户指令，我实现了差异化的处理逻辑：

- 运动控制指令（如"请往前走一步"）：提取方向信息并发送至运动控制模块
- 导航指令（如"带我去 327 教室"）：提取目的地并触发导航功能
- 机械臂操作（如"帮我抓一瓶水"）：激活机械臂控制流程
- 场景识别（如"描述我眼前的景象"）：触发视觉感知模块

◎ 状态管理与错误处理：我实现了完整的系统状态管理机制，包括指令处理中的状态标记、超时处理以及错误恢复机制，确保系统在各种异常情况下能够正常运行。

系统整合与调试：

作为架构设计者，我还负责了整个系统的整合与调试工作：

◎ 节点启动配置：编写了统一的 ROS 启动文件（start_all.launch），实现了一键启动所有功能模块的功能，简化了系统操作流程。

◎ 系统行为调优：通过实际测试，不断调整决策模型的提示词和参数设置，优化系统对自然语言指令的理解能力和响应效果。

4.2 负责视频发布、接收与处理节点构建

作为项目中视觉系统的负责人，我设计并实现了视频捕获、传输和处理的完整流程，使盲人引导机器人能够"看见"周围环境并为用户提供场景描述。

IP 摄像头接入方案设计：

考虑到机器人自带摄像头在移动机器人上的角度调整限制，我创新性地提出并实现了基于智能手机 IP 摄像头的解决方案：

◎ 手机摄像头视频流发布：我使用手机 APP IP Camera 通过网络将实时采集的手机摄像头视频数据以视频流的方式传输出去，并被连接在同一网络 IP 下的其他设备接收。

◎ HTTP 视频流接收：我实现了 video_cap.py 节点，该节点能够使用 OpenCV

从手机 IP 摄像头应用程序实时接收 HTTP 协议的 MJPEG 视频流，减少了对专业摄像头硬件的依赖。

视觉识别触发机制实现：

为了避免持续不必要的图像处理和 API 调用，我设计了一套高效的按需触发机制，该节点订阅了 /visual_cmd 话题。当 /visual_cmd 接收到“true”命令时，它会发布当前最新的一帧图像到 /visual_info 话题。这一设计确保系统只在用户明确请求场景描述时才会触发视觉识别流程，大大节约了计算资源和 API 调用次数。

4.3 负责运动控制节点构建

在本项目中，我负责实现机器人运动控制模块，该模块基于 ROS 平台，支持通过键盘控制和语音指令（文本命令）控制机器人进行移动操作。该模块主要功能如下：

控制接口设计：

我编写了一个 Python 脚本 teleop_twist_keyboard 的增强版本，实现了双输入控制方式：

◎ 键盘控制：支持通过键盘按键 w/s/a/d/x 控制机器人线速度和角速度的增减；

◎ 语音命令控制（通过订阅话题 robot_command）：支持解析包括“前进”、“后退”、“左转”、“右转”、“停止”等命令，实现对机器人的远程自然语言控制。

功能实现细节：

◎ 利用 geometry_msgs/Twist 消息类型，通过发布 /cmd_vel_mux/input/navi 话题，控制机器人底盘运动；

◎ 采用 std_msgs/String 类型接收决策模型发布的语音指令，处理逻辑清晰易扩展；

◎ 运动控制采用增量方式，即多次按键/指令会持续加速，提高控制灵敏度；

◎ 特别加入 "x" 和 "停止" 指令以实现一键紧急停止，增强安全性。

通过以上工作，我成功实现了一套可靠、安全且用户友好的运动控制系统，

使盲人用户能够通过自然语言指令轻松控制机器人的移动，大大提高了盲人引导机器人的实用性和可用性。

五、项目总结与展望

5.1 项目总结

本次实验得到的结果较好，设计的机器人能够较好的完成导航、递物、语音识别、环境描述、语音交互等工作内容，多模态之间的协作也能很好的运行，为相关领域的研究提供了一种可复用的技术框架

5.2 项目展望

由于本次实验的应用环境较为简单，因此“房间位置词典”的设计思路能够较好的完成任务，但是在后续应用过程中，势必会面临应用场景复杂化，用户需求多样化的趋势，在这一趋势下，如果继续采用提前标定目标位置的方法，会造成代码的冗余，且导航效果也难以保证，后续可以采用其他方法来替换提前标定房间位置的思路来提高导航的性能。

此外，机器人对于用户的任务需求表述能力要求较高，对于一些模糊化的导航需求存在响应不及时甚至是响应错误的情况，后续可以通过扩张“房间位置词典”或对房间进行更概括的“标签化”标定的方法改善此类情况，提升用户使用该机器人的便利程度。

参考文献

[1] Kuriakose, B., Shrestha, R. & Sandnes, F. E. (2020). Multimodal Navigation Systems for Users with Visual Impairments — A Review and Analysis. Multimodal Technol. Interact., 4(4), 73.

[2] Kang D O, Kim S H, Lee H, et al. Multiobjective navigation of a guide mobile robot for the visually impaired based on intention inference of obstacles[J].

Autonomous Robots, 2001, 10: 213-230.

[3] Lu C L, Liu Z Y, Huang J T, et al. Assistive navigation using deep reinforcement learning guiding robot with UWB/voice beacons and semantic feedbacks for blind and visually impaired people[J]. Frontiers in Robotics and AI, 2021, 8: 654132.

[4] Kim J T, Yu W, Kothari Y, et al. Transforming a quadruped into a guide robot for the visually impaired: Formalizing wayfinding, interaction modeling, and safety mechanism[J]. arXiv preprint arXiv:2306.14055, 2023.

[5] Chen Y, Xu Z, Jian Z, et al. Quadruped guidance robot for the visually impaired: A comfort-based approach[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023: 12078-12084.

[6] Liu Z, Zhu T, Xiang J, et al. Controllable and Diverse Data Augmentation with Large Language Model for Low-Resource Open-Domain Dialogue Generation[J]. arXiv preprint arXiv:2404.00361, 2024.

[7] Buckley B, O'Hagan A, Galligan M. Variational Bayes latent class approach for EHR-based phenotyping with large real-world data[J]. arXiv preprint arXiv:2304.03733, 2023.