

# 第二次实验报告

赵博瑄 2015201907 樊树霖 2015201919 牟善磊 2015201909 王宇睿 2015201906

## 项目进度

我们已经实现的功能有：

- 1、小车利用超声波自动避开障碍物
- 2、利用蓝牙端口给小车发送消息，用手机遥控小车
- 3、语音控制小车
- 4、小车自动追踪红色电烙铁

其中前两项功能已于上次实验实现，本次实验中，我们主要实现了后两项功能。

## 语音控制

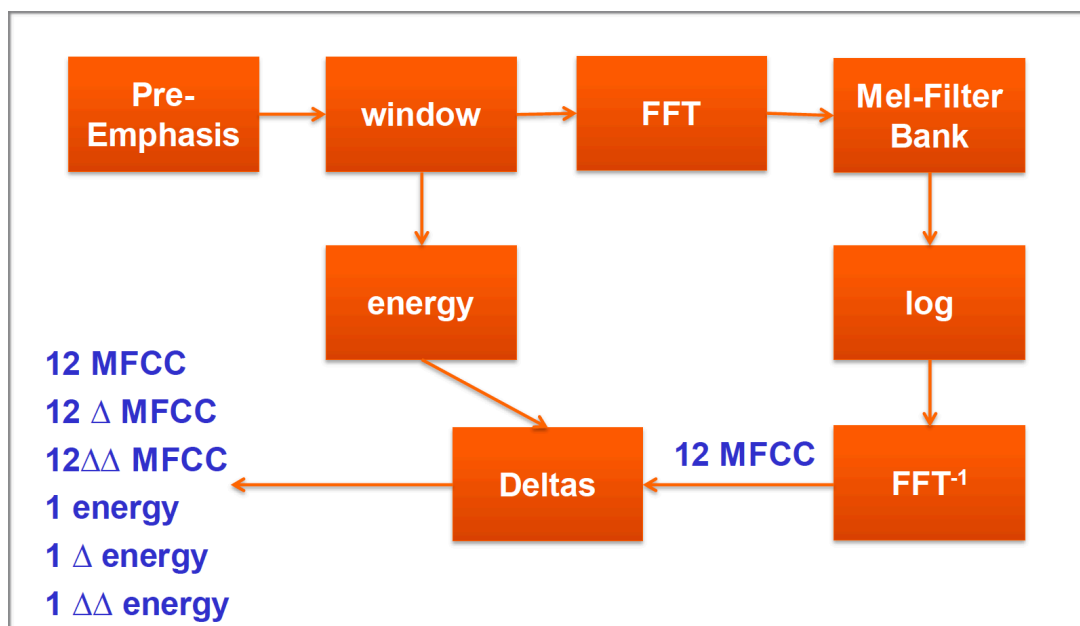
---

我们尝试了自己构建简单的语音识别系统,识别包括“左”，“右”，“前”，“后”，“停”等控制小车前进方向的关键字，构建和识别的主要过程为：

- 数据采集及特征提取

我们通过录制音频和开源数据中采集两种方式获得了用于输入大量的语音数据及其对应的文字标签。对获得的原始语音信号进行处理（由于录制时在较为安静的环境中，因此不需进行降噪处理），对语音信号进行分帧（近似认为在 10-30ms 内是语音信号是短时平稳的，将语音信号分割为一段一段进行分析）以及预加重（提升高频部分）等处理。

用 kaldi 提取出能够反映语音信号特征的关键参数 MFCC 特征。提取 MFCC 的计算过程如下：



- 模型训练

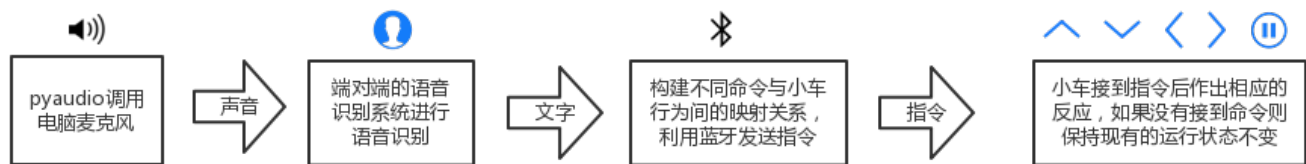
将整理好的 MFCC 特征和对应的类标签输入到构建的分类器中进行训练，利用交叉验证和网格搜索调整分类器的超参数，获得构建好的语音识别模型。

- 识别命令

利用 python 中的 pyaudio 包调用电脑的麦克风，从麦克风获取声音。每隔一定的秒数将获取到的声音文件实时保存到本地。在 linux 环境中调用 shell 脚本利用现有的工具对音频文件进行 MFCC 特征提取，将得到的特征结果以文件形式保存，然后程序从文件中读取数据，输入到语音识别模型中进行分类，得到对应的标签（左、右、前、后、停）。

然而，由于从麦克风中获得声音后进行操作较多，速度慢，同时，自己构建的语音识别系统正确率欠佳，无法满足控制小车的实时性要求，因此我们决定采用现有的模型对小车进行语音控制。

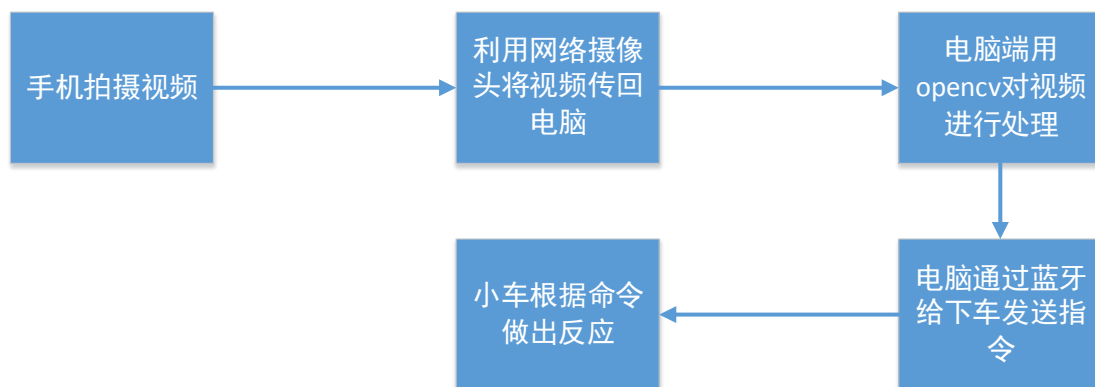
我们首先在电脑端利用端对端的语音识别系统（IBM Speech to Text）进行语音识别，识别出所说的内容，然后程序读取语音的内容，并根据语言的内容给小车发送相应的指令，小车接到指令后作出相应的反应，如果没有接到命令则保持现有的运行状态不变。整个语音识别过程见下图：



## 物体追踪

我们将手机固定在小车上，利用特殊的软件将手机作为网络摄像头，使得电脑可以通过相应的 ip 访问手机摄像头，通过这种方式，将手机拍摄到的画面实时传送到电脑。在电脑端，我们使用 opencv 对传回的视屏进行处理，本次实验中，我们对识别的规则进行了设定，利用红色电烙铁上的红色方形区域，使得程序能够根据颜色和形状的信息识别视频中的电烙铁，并给出电烙铁上的红色方形区域的质心在小车的视野中的坐标，根据坐标信息判断电烙铁在小车的那个方向，然后通过蓝牙串口给小车发送改变运动方向的指令，小车收到指令后会做出相应的反应，从而使小车跟踪视屏中的电烙铁。

大致流程如下：



## 问题及解决方案

- 1、在测试的过程中小车的电机突然出现故障，我们不得不将底板重新进行组装，跟换新的电机后问题得到了解决。
- 2、由于在安装和调试的过程中插拔的次数过多，小车 arduino 板的多个接口出现问题，插入电线后线路无法接通，最后我们重新更换了 arduino 板，将小车的主要部分重新进行了组装，最后成功解决了问题。
- 3、在测试过程中，由于电池的使用时间较长，电池的电压不足，两个车轮的速度出现了差异，我们在代码中进行了相应的调整，确保小车的两轮有相近的速度。
- 4、在语音识别的过程中，我们尝试了自己搭建语音识别系统，但是由于过程步骤较多，电脑的计算资源不足，同时麦克风有延迟，特征提取的工具工作耗时长，训练数据不足，因此尝试搭建的语音识别系统的识别准确率不高并且识别系统的速度很慢，无法实现实

时控制小车的要求，因此最后我们采用现有的模型进行了语音识别，提升了整个系统的运行效率。