

# 浙江大学

## 本科设计报告

课程名称: 信息、控制与计算

设计名称: 远程声控系统

姓 名:

学 号:

学 院: 信电学院

专 业: 电子科学与技术

班 级: 电科 1803

指导老师: 叶志

2021 年 1 月 10 日

# 目录

1 问题背景 . . . . .	3
2 问题分析与解决思路 . . . . .	3
3 实现过程 . . . . .	3
3.1 语音信号采集 . . . . .	4
3.2 量化/编码 . . . . .	5
3.3 信源编码 . . . . .	6
3.4 信道编码 . . . . .	7
3.5 BPSK 调制和加性高斯白噪声 . . . . .	8
3.6 BPSK 解调 . . . . .	9
3.7 信道译码 . . . . .	10
3.8 信源译码 . . . . .	11
3.9 信号重建 . . . . .	11
3.10 语音识别 . . . . .	12
3.11 动作控制 . . . . .	14
参考文献 . . . . .	15

## 1 问题背景

实现一个远程声音控制系统。首先采集不同的语音指示信号，进行适当压缩；然后通过噪声信道实现远程传输，远端接收后再通过适当计算识别出是何指示，最后送入一个处于未知状态、但能控/能观的控制系统，完成不同的控制动作。

## 2 问题分析与解决思路

远程声控系统可以分解为三部分：语音指示信号采集、信号传输、指令识别。语音信号采集部分，我们可以用麦克风录制一个时长 1 秒的指令，该部分可以通过 MATLAB 自带的 *audiorecorder* 来实现。信号传输部分，我们采用传统的信号传输模型，将原始模拟语音信号量化后，经过信源编码、信道编码、调制解调、信道译码、信源译码得到传输后的信号。该模型如图 1.1 所示。其中信源编码我们采用了哈夫曼编码，可以达到较高的压缩率；信道编码我们采用了循环码，通过增加冗余以提高传输的可靠性；调制部分我们使用了 BPSK，来提高抗干扰能力；在传输过程中，我们人为引入了高斯白噪声。指令识别部分，我们通过使用 MATLAB 预训练的语音识别模型来识别我们解码后的信号。最后，我们根据指令来改变控制系统的状态。

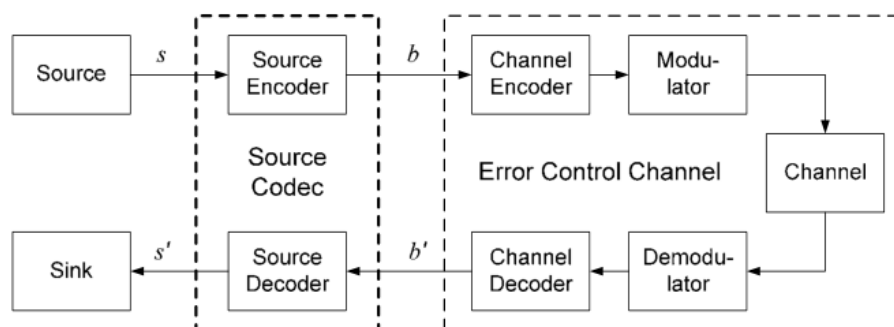


图 2.1: 传输模型

## 3 实现过程

图 3.1 为实现好的远程声控系统，我们在图 2.1 的基础上增加了语音信号采集和指令识别模块，下面我们将详细叙述整个系统的实现原理及结果。

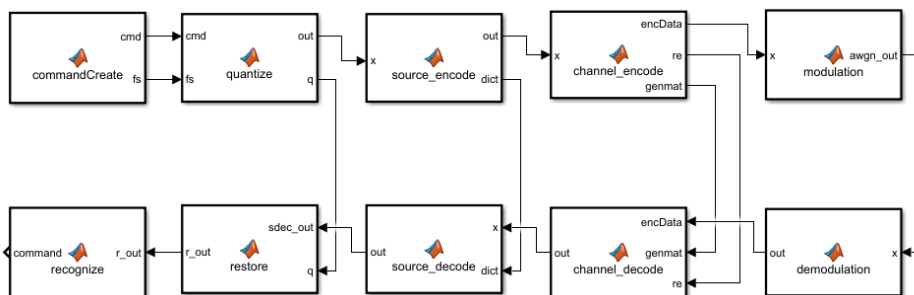


图 3.1: Simulink 仿真模型

### 3.1 语音信号采集

麦克风的输入幅值连续的模拟信号，而在数字通信系统中，信息由 1 和 0 组成，因此我们必须把模拟信号转换成数字格式才能在数字系统中使用。相比模拟信号，数字信号相对抗干扰性强、易于进行错误检测和纠正、易于复用、易于处理和存储。通过 A/D 转换，我们可以实现模拟信号到数字信号的转换，这涉及到采样、量化和编码。

在这里，我们做的是 A/D 转换中的采样，对应模块为图 3.1 中的 *commandCreate*。我们调用了 MATLAB 自带函数 *audiorecorder* 来录制麦克风输入的音频数据，设置采样频率为 16000Hz，采样位数为 16bits，通道数为 1，录音时长为 1s，由此采集到语音信号指令。以语音输入“stop”为例，通过信号采集，我们得到了信号的时域图和频域图。由图 3.2 可见，语音采集后的幅值在 -1V 到 1V 之间，傅里叶变换后的频域图显示频率在 0-8000Hz，处于正常人说话的频率区间。

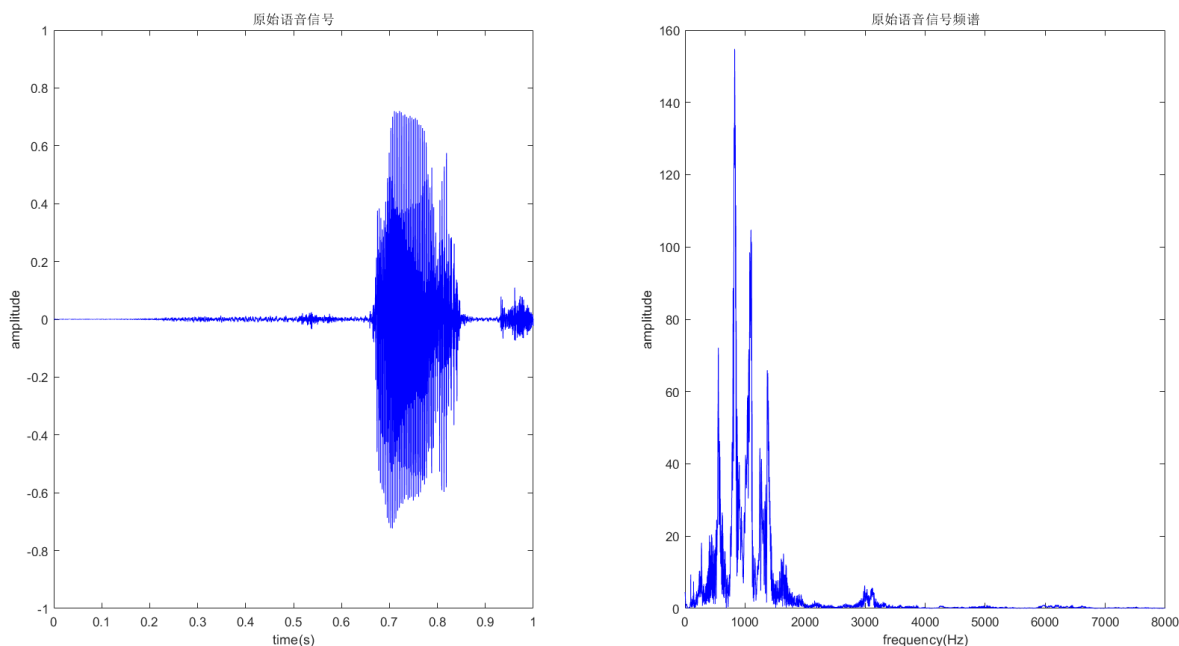


图 3.2: 原始语音信号时域和频域图

## 3.2 量化/编码

量化/编码是将采样的模拟电压值映射到离散电压电平，然后用二进制数字（位）表示。在任何数字系统中，只有有限数量的内存，因此只能使用有限数量的值代表模拟信号的样本。图 3.3 简单说明了 3 bit 量化/编码的原理。

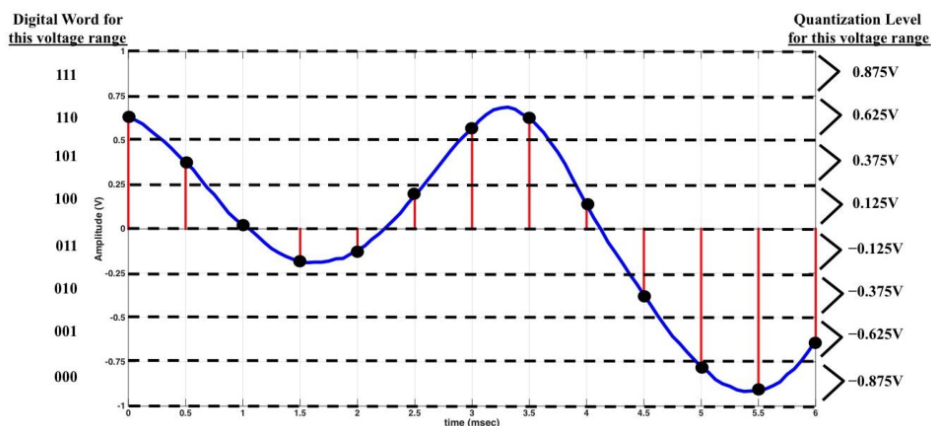


图 3.3: 3 bit 量化/编码示意图

在本系统中，我们使用的量化级为 8，量化器的分辨率为：

$$q = \frac{V_{max} - V_{min}}{2^N} = \frac{1 - (-1)}{2^8} = 3.90625 \times 10^{-3}V$$

图 3.4 展示了原始语音信号量化前后的效果对比。

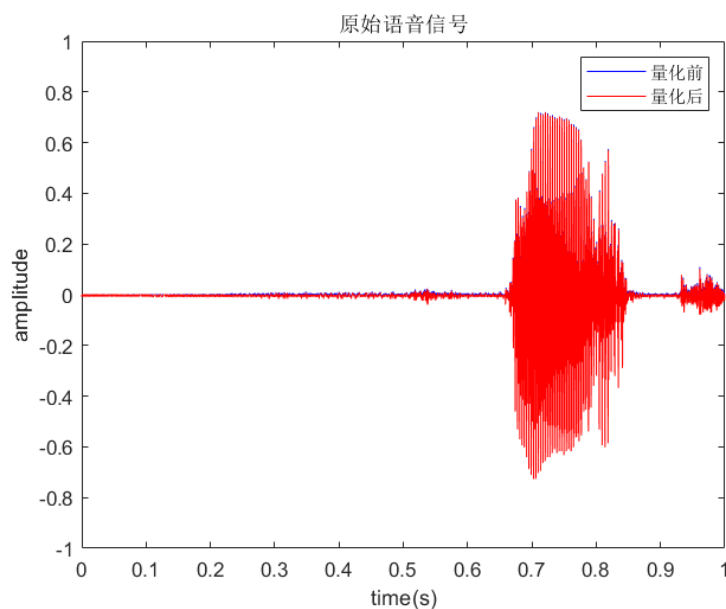


图 3.4: 原始语音信号量化前后对比

### 3.3 信源编码

信源编码的目的是提高编码的有效性，使信源减少冗余，更加有效、经济地传输。在量化和 8bit 编码后的结果中，我们可以很明显地看到部分信号出现频率较高，部分信号几乎没有出现，若分配同样的码长，必然会浪费信道的带宽，因此我们采用了可以根据信号出现的频率分配码长的哈夫曼编码，以实现压缩和节约带宽的目的。哈夫曼编码的一个优点是它是一种无损编码，下面介绍哈夫曼编码的基本步骤：

- 1) 将信息员符号出现的概率从大到小顺序排雷。
- 2) 将两处最小的概率相加，形成一个新概率。
- 3) 再将概率进行重新排列，如此重复直到只有两个概率为止。
- 4) 分配码字。码字分配从最后一步开始反方向进行，对最后两个概率一个赋予“0”码字，一个赋予“1”码字，如此反方向进行到开始的概率排列。

在本系统中，我们先计算各符号出现的概率，然后通过调用 MATLAB 内置函数 *huffmandict* 生成字典，通过调用 *huffmanenco* 生成哈夫曼编码，相应的解码可以通过调用 *huffmandeco* 实现。图 3.5 是信源编码的统计结果，展示了部分符号的哈夫曼编码码字和最终输出（部分）。我们可以看到哈夫曼编码的平均码长为 3.263625，而比起原先的 8bit 量化，平均码长减少了一半多，压缩率高达 2.451262。

[illegible]

图 3.5: 哈夫曼编码效果

### 3.4 信道编码

由于实际信道存在噪声和干扰，使得经过信道传输后收到的码字和发送码字之间存在差错。一般情况下，信道噪声和干扰越大，码字产生差错的可能性也越大。信道编码的目的在于改善通信系统的传输质量，发现或者纠正差错，以提高通信系统的可靠性。其基本思路根据一定的规律在待发送的信息码元中加入一些冗余的码元，即监督码元。在本系统中，我们采用循环码的方法。

先介绍线性分组码，它包括两个基本步骤：首先将信源输出的信息序列以  $k$  个划分为一组；然后根据一定的编码规则由这  $k$  个信息码元产生  $r$  个监督码元，构成  $n(k+r)$  个码元组成的码字，一般用  $(n, k)$  表示。线性码的监督码元和信息码元可以用一组线性代数方程联系起来。对一个线性分组码  $C$ ，若它的任一码字的每一循环移位寄存器都是  $C$  的一个码字，则称  $C$  是一个循环码。循环码编码流程如下图所示。

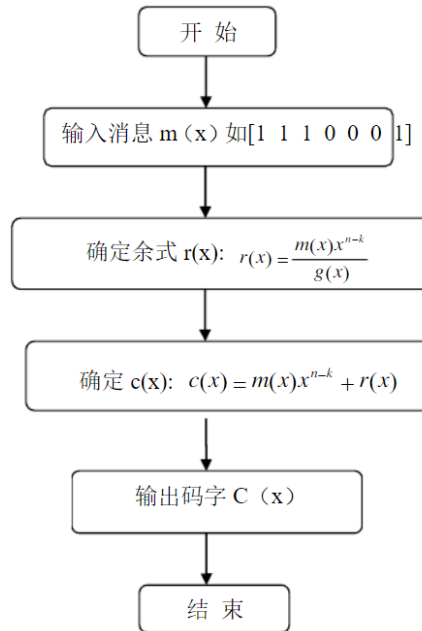


图 3.6: 循环码编码流程图

在本系统中，我们使用  $(7, 4)$  循环码，通过调用 MATLAB 函数 *cyclpoly*、*cyclgen*、*gen2par*，创建一个循环码的生成多项式，然后，创建奇偶校验矩阵并将其转换为生成矩阵，最后调用 *encode* 进行编码。我们使用的生成多项式  $G(x) = x^3 + x^1 + 1$ ，生成矩阵为：

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

图 3.7: 信道编码效果 (部分)

相移键控是利用载波的相位变化来传递数字信息，而振幅和频率保持不变。在 BPSK 中，通常用初始相位  $0$  或  $\pi$  来表示二进制的“1”和“0”。



在 MATLAB 中，我们可以调用 *pskmod* 实现 BPSK 调制，调用 *awgn* 在信号中加入高斯白噪声，信噪比为 10dB。图 3.9 为该模块输出信号。



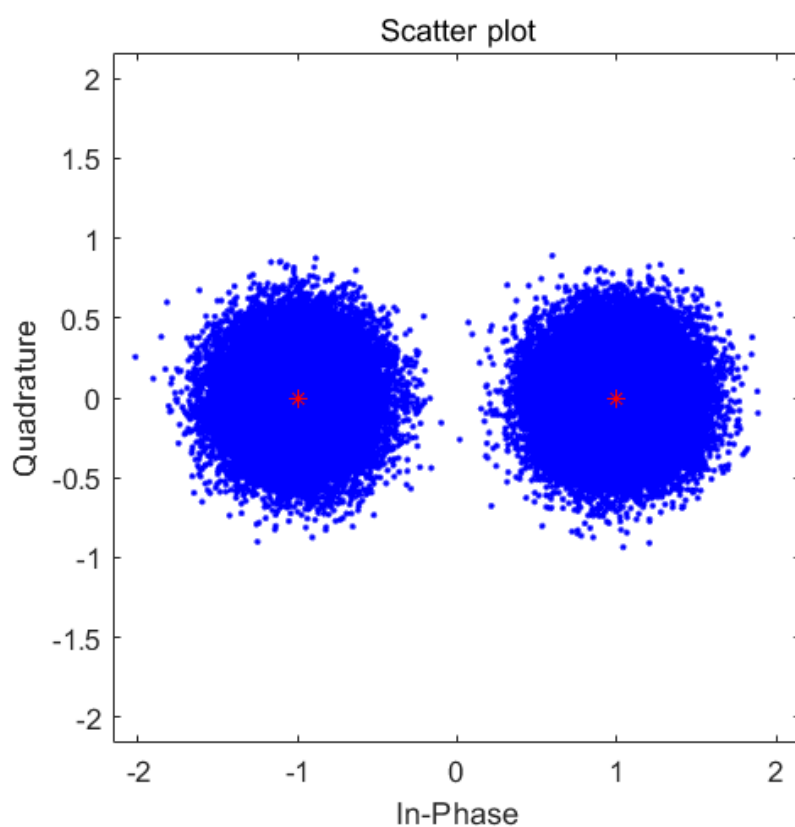


图 3.9: BPSK 调制和引入高斯白噪声后的信号

### 3.6 BPSK 解调

图 3.10 为 BPSK 解调原理框图。

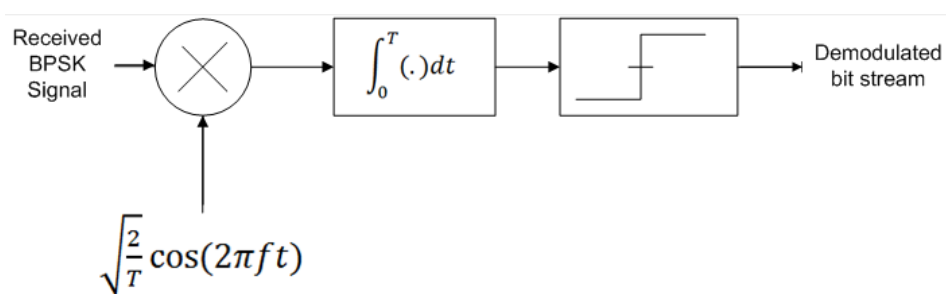


图 3.10: BPSK 解调

在该过程中，我们计算了解调后序列相比调制前序列的比特出错数和误码率，得到的结果如下：

```

出错数 : 0
误码率 : 0.000000
fx >>

```

图 3.11: 错误统计

### 3.7 信道译码

由于我们信道编码采用的是循环码，其译码过程如下图所示。

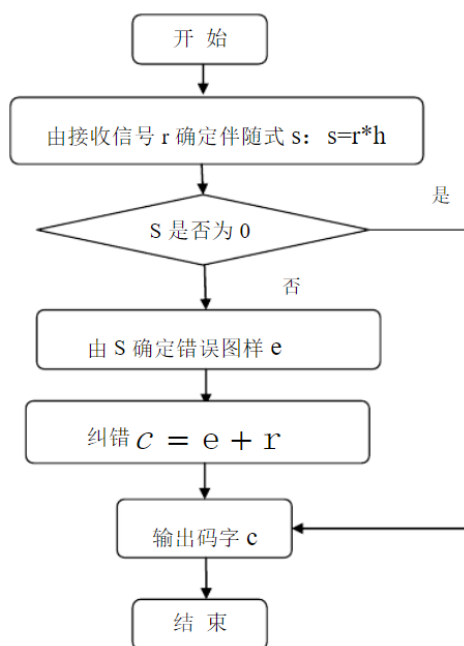


图 3.12: 循环码译码流程图

我们使用 MATLAB 内置函数 *decode* 进行循环码的译码。输出结果如下：

```

出错数 : 0
误码率 : 0.000000
fx >>

```

图 3.13: 错误统计

另外，我们测试了不同信噪比（-20dB-20dB）下的信道译码的比特出错数和误码率，得到如下结果，由图可见，高斯白噪声信道信噪比越高，信道译码出错率越低。到后面，序列通过信道传输基本没有发生错误。

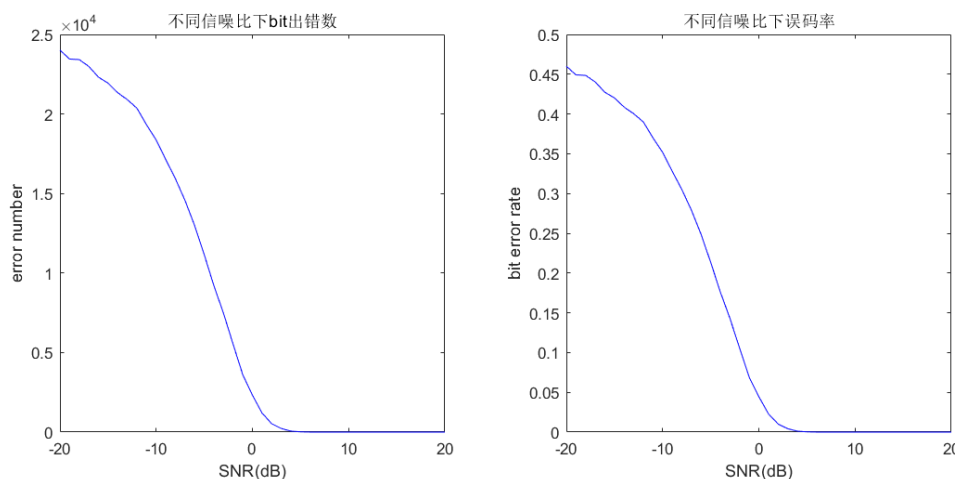


图 3.14: 不同信噪比下的译码效果

### 3.8 信源译码

我们的信源编码采用的是哈夫曼编码，在信源译码部分，我们采用 MATLAB 内置函数 `huffmandeco`。与最初量化后的信号比较，得到的反馈如下：

```
出错数 : 0
误码率 : 0.000000
fx >>
```

图 3.15: 错误统计

### 3.9 信号重建

由于在量化阶段我们把模拟电压变成了在 00000000-11111111 之间的离散信号，因此我们需要把译码后得到的离散信号还原为模拟电压信号。具体操作为用先前量化的分辨率  $q$  与离散信号相乘，然后减 1，以使电平信号在 -1V 到 +1V 之间。下图展示了原始语音信号与重建后的语音信号的均方误差，为 0.000019，可见信号传输基本没有差错。

```
均方误差为 : 0.000019
fx >>
```

图 3.16: 重建后语音信号与原始信号均方误差

我们可以通过原始语音信号和重建语音信号的时域图、频域图对两者进行更深入的分析。由频域图和时域图可见，两者几乎没有差别。

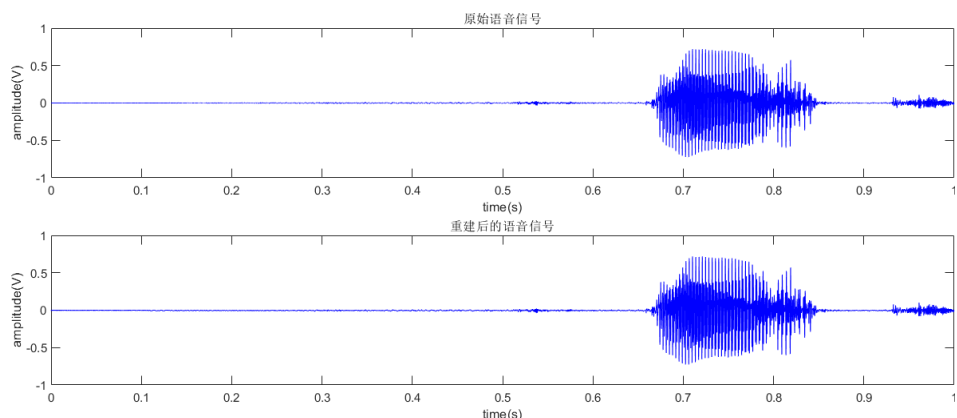


图 3.17: 时域图对比

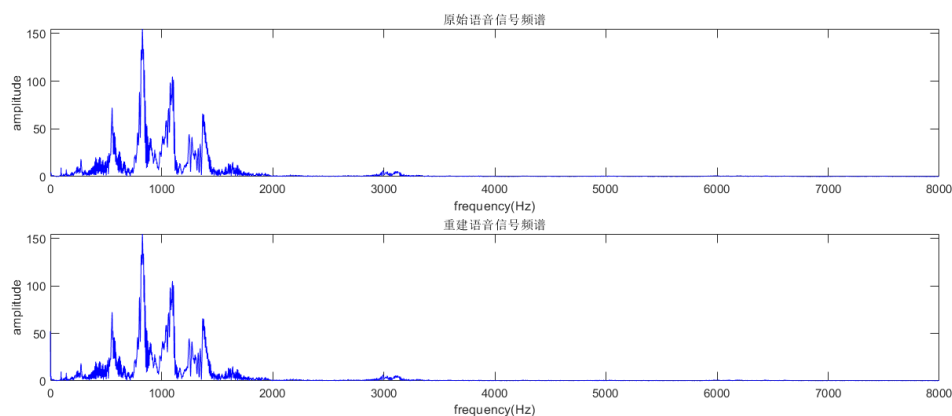


图 3.18: 频域图对比

### 3.10 语音识别

语音识别 (Speech Recognition) 技术, 也被称为自动语音识别 (Automatic Speech Recognition, ASR), 其目标是用计算机自动将人类的语音内容转换为相应的文字。一个语音识别系统主要包括信号处理和特征提取、声学模型训练、语言模型训练以及识别引擎等几个核心部分:

**特征提取:** 语音识别的第一步就是特征提取, 去掉语音信号中对于语音识别无用的冗余信息 (如背景噪音), 保留能够反映语音本质特征的信息 (为后面的声学模型提取合适的特征向量), 并用一定的形式表示出来; 较常用的特征提取算法又 MFCC。

**声学模型训练:** 根据语音库的特征参数训练出声学模型参数, 在识别的时候可以将待识别的语音特征参数同声学模型进行匹配, 从而得到识别结果。目前主流的语音识别系统多采用 HMM 进行声学模型建模。

**语言模型训练:** 就是用来计算一个句子出现的概率模型, 主要用于决定哪个词序列的可能性更大。语言模型分为三个层次: 字典知识、语法知识、句法知识。对训练

文本库进行语法、语义分析，经过基于统计模型训练得到语言模型。

**语音解码与搜索算法：**其中解码器就是针对输入的语音信号，根据已经训练好的声学模型、语言模型以及字典建立一个识别网络，再根据搜索算法在该网络中寻找一条最佳路径，使得能够以最大概率输出该语音信号的词串，这样就确定这个语音样本的文字。

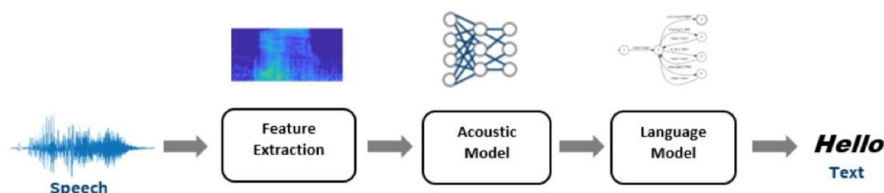


图 3.19: 语音识别系统

在本设计中，我们没有采用传统的 HMM 声学模型，而是使用了现下较为流行的 CNN，利用卷积的不变性来克服语音信号本身的多样性。CNN 模型将整个语音信号分析得到的时频谱当作一张图像一样来处理，采用图像中广泛应用的深层卷积网络对其进行识别。下图为典型的 CNN 模型。

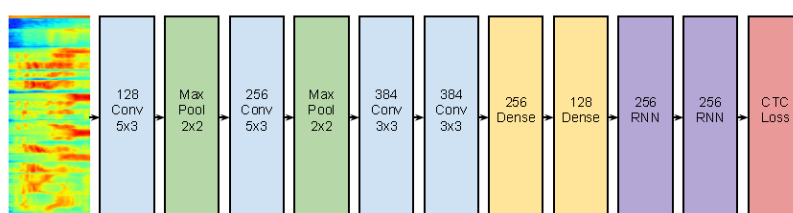


图 3.20: CNN 模型

在语音识别模块，为了节省训练开销，我们直接使用了 MATLAB toolbox 中的 Deep Learning Toolbox，该工具箱中已经有了预训练的语音识别 CNN 模型，可以识别“yes”，“no”，“up” “down”，“left”，“right”，“on”，“off”，“stop”，“go” 共 10 条指令，其余指令为 unknown。值得注意的是，我们在采样阶段的录音时长为 1 秒，也就是一个指令的时长被限制在 1 秒内。

语音识别模块的操作流程为：

- 1) 通过 *load* 函数导入预训练模型的参数。
- 2) 通过 *helperExtractAuditoryFeatures* 提取重新后语音信号的特征信息。
- 3) 通过 *classify* 函数对声音的特征信息进行分类，输出分类后的指令信息。

图 3.19 展示了语音信号及其频谱图，我们的语音识别模型正确地输出了指令。更多指令的识别见附录 1。

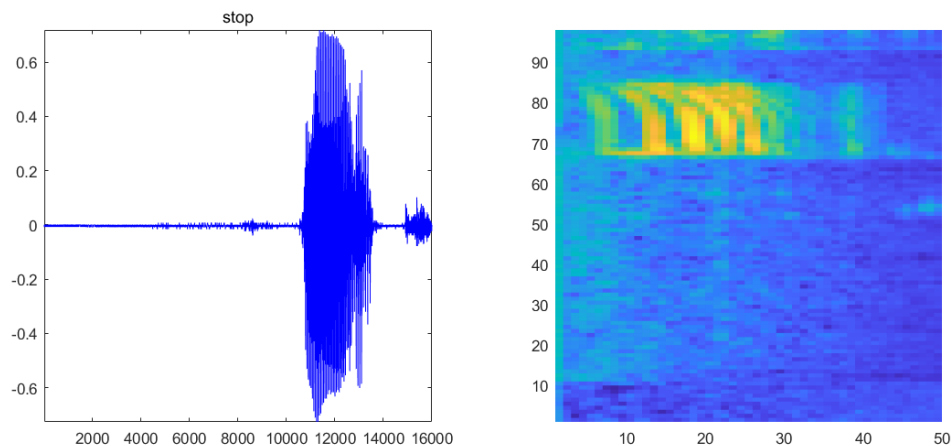


图 3.21: 语音信号及其频谱图

### 3.11 动作控制

由于缺乏原材料，该模块我们仅以语言形式阐述实现过程。我们的控制系统是一个有限状态机，所有指令都是预定义的，即上文提到的 10 种指令，这些指令可以存放在一个字典中，字典的键值为递增的数字。在动作控制部分，我们可以在字典中查询语音识别后的指令，得到相应的键值，然后将通过该键值来改变系统的状态，从而完成不同的控制动作。

## 参考文献

1. 张雪英. 数字语音处理 [M]. 北京: 电子工业出版社, 2010.
2. 郑君里, 应启珩, 杨为理. 信号与系统 [M]. 北京: 高等教育出版社, 2000.
3. 薛年喜. MATLAB 在数字信号处理中的应用 [M]. 北京: 清华大学出版社, 2003.
4. 胡航. 语音信号处理 [M]. 哈尔滨: 哈尔滨工业大学出版社, 2000.
5. 易克初, 田斌, 付强. 语音信号处理 [M]. 北京: 国防工业出版社, 2000.
6. 万永革. 数字信号处理的 MATLAB 实现 [M]. 北京: 科学出版社, 2007.
7. 刘卫国. MATLAB 程序设计与应用 [M]. 北京: 高等教育出版社, 2006.
8. 王嘉梅. 基于 MATLAB 的数字信号处理与时间开发 [J]. 西安: 西安电子科技大学出版社, 2007: 10-14.
9. 程佩青. 数字信号处理教程 (第二版) [M]. 北京: 清华大学出版社, 2010.
10. 韩纪庆, 张磊, 郑铁然. 语音信号处理 [M]. 北京: 清华大学出版社, 2004.
11. 徐明远, 邵玉斌. Matlab 仿真在通信与电子工程中的应用 [M]. 西安: 西安电子科技大学出版社, 2005.
12. 邓华. Matlab 通信仿真及应用实例详解 [M]. 北京: 人民邮电出版社, 2005.
13. 张照明, 刘政波, 刘斌等. 应用 Matlab 实现信号分析处理 [C]. 北京: 科学出版社, 2006.
14. 徐守时. 信号与系统理论方法和应用 [M]. 合肥: 中国科学技术大学出版, 1999.
15. 高俊斌. Matlab 语言与程序设计 [M]. 武汉: 华中理工大学出版社, 1998.