

# 浙江大学

## 本科设计报告



题目 远程声控系统

姓名 \_\_\_\_\_

学号 \_\_\_\_\_

# 目 录

1 问题说明.....	3
2 问题分解与解决思路.....	3
3 实现 .....	3
3.1 语音信号采集与编码.....	3
3.1.1 采集.....	3
3.1.2 量化.....	4
3.1.3 信源编码.....	5
3.2 信号传输.....	6
3.2.1 信道编码.....	6
3.2.2 BPSK 调制和加性高斯白噪声.....	6
3.2.3 BPSK 解调.....	7
3.2.4 信道译码.....	7
3.3 语音指令信号译码识别.....	8
3.3.1 信源译码.....	8
3.3.2 信号重建.....	9
3.3.3 语音识别.....	9

# 1 问题说明

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 问题分解与解决思路

该远程声控系统可以分解为四个部分：语音指令信号采集与编码、信号传输、语音指令信号译码识别、输出控制状态转移。

- (1) 语音指令信号采集与信源编码：设置指令为“向左”、“向右”、“向上”、“向下”、“左下”、“右下”、“左上”、“右上”。每条指令时长 1s，采用 MATLAB 的 `audiorecorder()` 实现。将语音信号进行归一化、量化处理后，进行 Huffman 编码。
- (2) 信号传输：采用传统的信号传输模型，将信源编码进行信道编码和调制。信道编码采用卷积码的形式，调制部分使用 BPSK 提高抗干扰能力；为了仿真真实的信道传输，人为增加高斯白噪声，并对信噪比的大小对卷积码性能的影响做了一定的讨论。
- (3) 语音指令信号译码识别：对从信道接收的信号进行信源译码，采用 MATLAB 的 `huffmandeco()` 函数进行。再对信号进行恢复，并于标准指令信号的 MFCC 特征对比，得到最可能的指令。
- (4) 输出控制状态转移：根据得到的最大概率指令控制小车的坐标移动，并呈现出来。

## 3 实现

### 3.1 语音信号采集与编码

#### 3.1.1 采集

采用 MATLAB 自带函数 `audiorecorder()` 函数，采样率为 44800Hz，采样位数 16bits，单通道，时长为 2s，分别对“向左”、“向右”、“向上”、“向下”、“左下”、“右下”、“左上”、“右上”作采集。

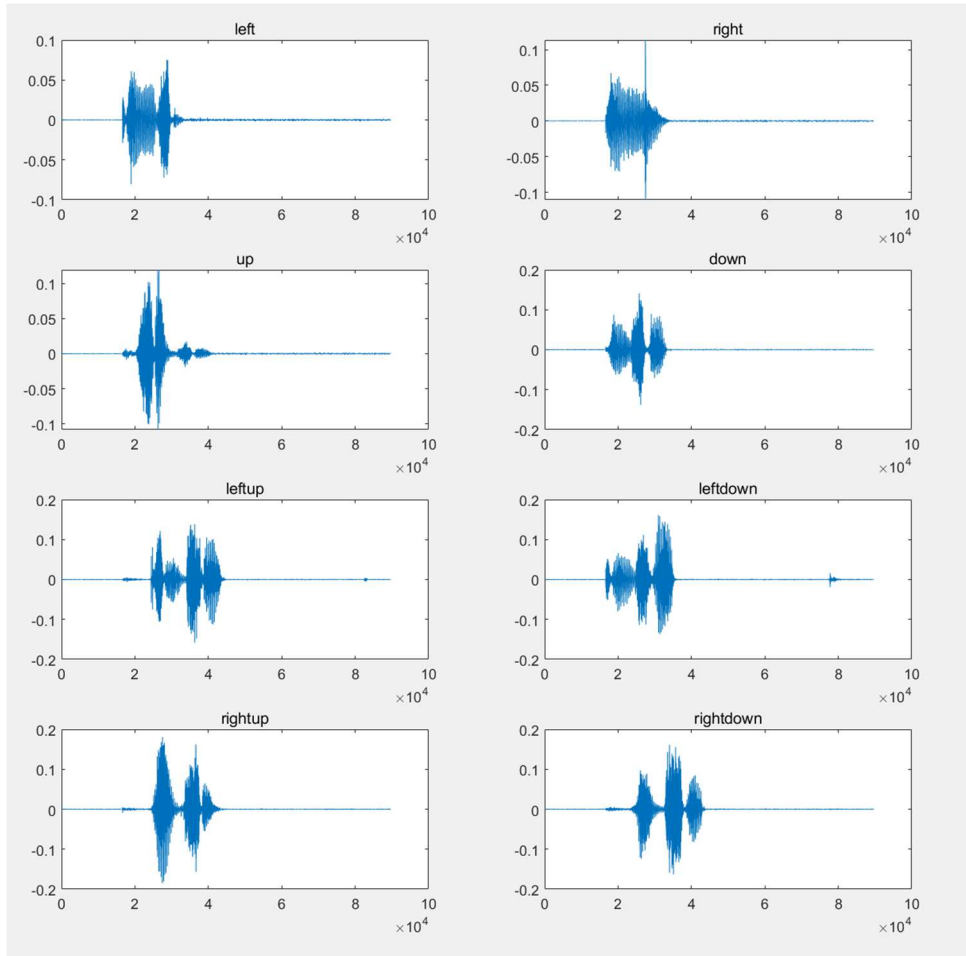


图 3.1.1: 原始语音信号时域图

### 3.1.2 量化

对信号采用  $N$  bit 量化方法。先对信号进行归一化，幅值在 $[-1,1]$ 之间，并作为指令的标准信号。本次设计取量化位数为 $N = 5$ ，分辨率为 $res = \frac{V_{max}-V_{min}}{2^N} = 0.0625$ 。如图 3.1.2，以 leftup 信号的量化为例，展示了 leftup 信号量化前后的对比，虽然损失了一些精度，但一定程度上也过滤了一些噪声。下面将对量化后的信号进行信源编码。

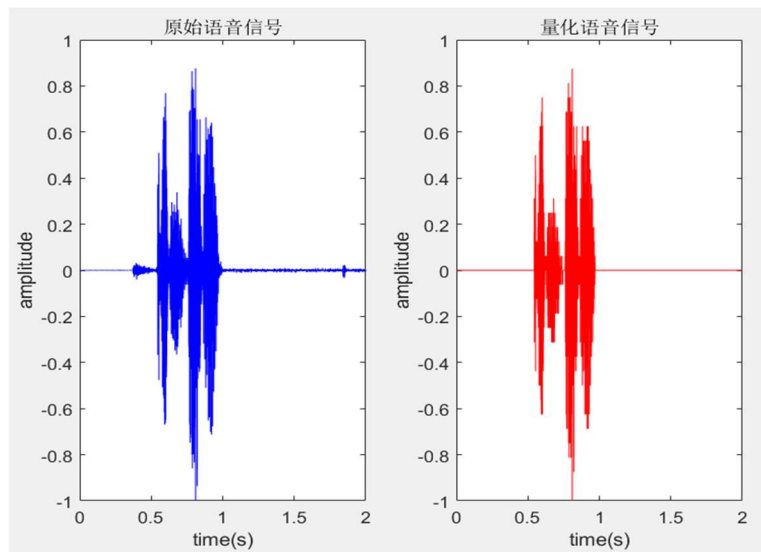


图 3.1.2: 信号量化前后对比图

### 3.1.3 信源编码

信源编码采用 Huffman 编码。假定八条指令的出现概率相同，只需要对总体信号中各种量化值进行统计，得出每个值的出现概率，再按照 Huffman 编码准则进行编码即可。

概率的统计由循环遍历完成，Huffman 编码由 MATLAB 的 `huffmandict()` 函数完成。得到的结果如图 3.1.3(a)。由图 3.1.3(b)可见，信源熵为 1.326129，该 Huffman 编码的平均码长为 1.326129，相比原 5bit 编码，压缩率达 2.77。

1		2
1 -15		<i>1x15 double</i>
2 -14		<i>1x14 double</i>
3 -13		<i>1x12 double</i>
4 -12		<i>1x12 double</i>
5 -11		[1,0,0,1,0,1,1,1,1]
6 -10		[1,0,0,1,0,1,1,0,1,0]
7 -9		[1,0,0,1,0,1,1,0,0,0]
8 -8		[1,0,1,1,0,0,0,1]
9 -7		[1,0,1,1,0,0,1]
10 -6		[1,0,1,0,1,0,0]
11 -5		[1,0,1,0,1,1]
12 -4		[1,0,1,1,1]
13 -3		[1,0,1,0,0]
14 -2		[1,1,0,0]
15 -1		[1,1,1]
16 0		0
17 1		[1,0,0,0]
18 2		[1,1,0,1]
19 3		[1,0,0,1,1]
20 4		[1,0,0,1,0,0]
21 5		[1,0,1,1,0,1]
22 6		[1,0,0,1,0,1,0]
23 7		[1,0,1,0,1,0,1]
24 8		[1,0,1,1,0,0,0,0]
25 9		[1,0,0,1,0,1,1,0]
26 10		[1,0,0,1,0,1,1,0,1,1]
27 11		<i>1x11 double</i>
28 12		<i>1x11 double</i>
29 13		<i>1x12 double</i>
30 14		<i>1x14 double</i>
31 15		<i>1x14 double</i>
32 16		<i>1x15 double</i>

```
>> main
avglen:1.326129
Entropy:1.724903
Press Rate:2.770373
fx >>
```

图 3.1.3: (a)Huffman 编码结果图 (b)Huffman 编码的平均码长、信源熵和压缩率

## 3.2 信号传输

### 3.2.1 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,我们采用卷积码方法。

卷积码( $n, k, N$ )是一种非分组码,  $n$  和  $k$  均指比特(分组码可以看作帧/矢量编码,卷积码可以看作数据流编码)。因此卷积码是将  $k$  比特的信息段编成  $n$  比特的码组,其编码生成的监督码元不止和  $k$  比特的信息段有关,还和前面的  $m=(N-1)$  个信息段有关。可知,一个监督码元同时和前  $m$  比特以及当前输入的一个比特信息有关,即监督着  $N=m+1$  比特,我们将  $N$  成为约束度,所有监督码元的长度  $nN$  称为约束长度。

MATLAB 中实现卷积码需要用到函数 `poly2trellis()` 和 `convenc()`,前者用于生成相应卷积码的网格表,后者用于根据网格表生成卷积码。此处选择(177,133,6)。卷积码的译码则采用 `vitdec()` 函数,选择译码器回溯深度为 6 倍约束长度。

### 3.2.2 BPSK 调制和加性高斯白噪声

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”。

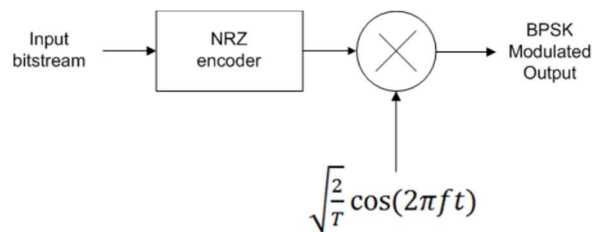


图 3.2.2: BPSK 调制原理图

加性高斯白噪声 (AWGN), 加性指的是叠加在某信号上; 高斯指的是概率分布里的正态分布函数, 白噪声指的是它的二阶矩不相关、一阶矩为常数, 即为先后信号在时间上的相关性。如果这个噪声的幅度服从高斯分布, 功率谱密度又是均匀分布的, 则称它为高斯白噪声。在 MATLAB 中, 我们可以利用 `pskmod()` 实现 BPSK 调制, 再调用 `awgn()` 在信号中加入高斯白噪声。

### 3.2.3 BPSK 解调

图 3.2.3 为 BPSK 解调原理框图。令码率  $M = 2$ 。

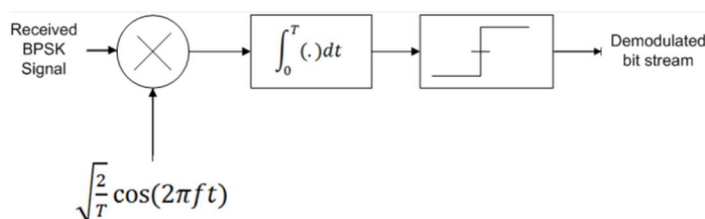


图 3.2.3: BPSK 解调原理图

在 MATLAB 实现中, 采用 `pskdemod()` 函数即可。

### 3.2.4 信道译码

信道采用 `vitdec()` 函数对卷积码进行译码, 图 3.2.4 显示了信噪比与卷积码性能曲线, 可见高斯白噪声信道信噪比越高, 信道译码出错率越低。当信噪比大于 6 时, 序列通过信道传输基本没有发生错误。

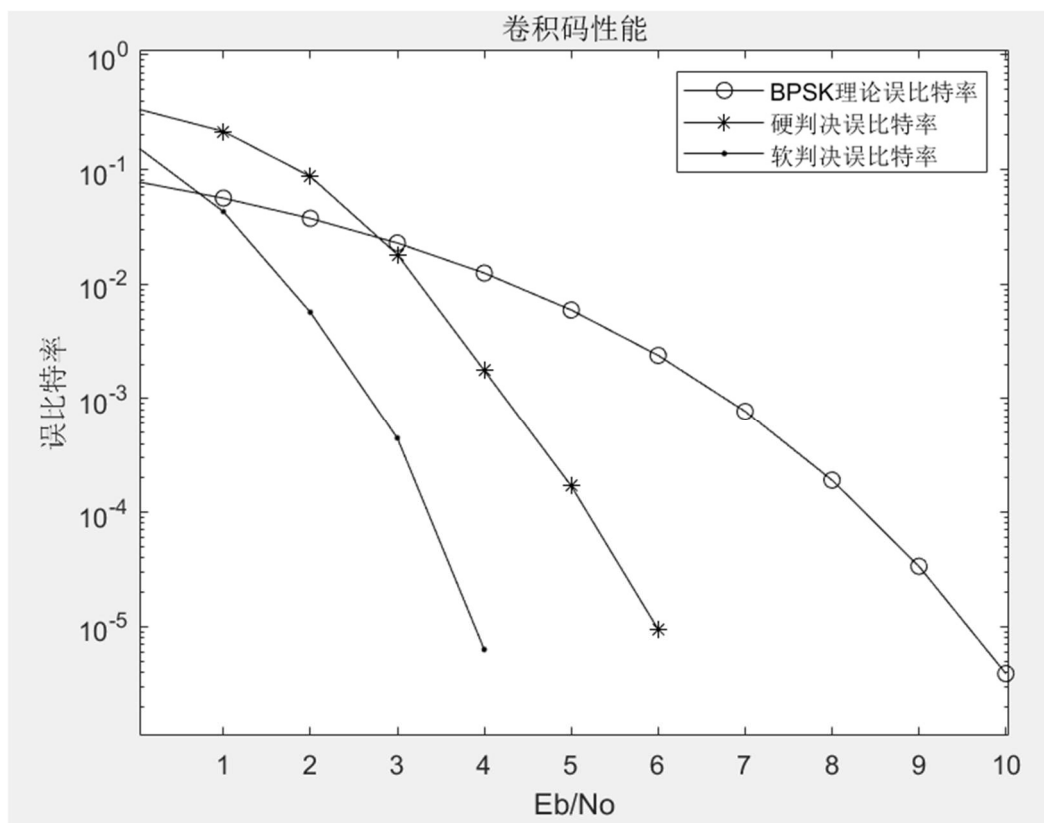


图 3.2.4: 卷积码性能随 SNR 变化曲线图

## 3.3 语音指令信号译码识别

### 3.3.1 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，得到的反馈如图 3.3.1:

```

命令行窗口
>> Error = length(OutAudio_q)-length(find(OutAudio_q-InputAudio_q==0));
>> Error

Error =

    0

fx>> |

```

图 3.3.1:信源译码的错误统计图



### 3.3.2 信号重建

由于在量化阶段我们把模拟电压转换成了 00000-111111 之间的离散信号，因此需要把译码后得到的离散信号还原为电压信号。可以看出信号的精度略有损失，但总体恢复较好。

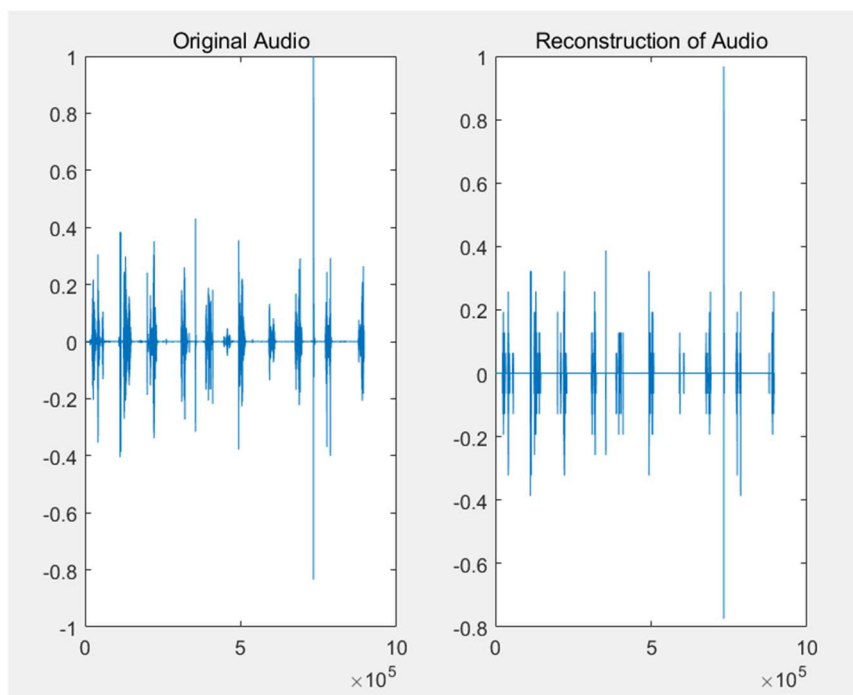


图 3.3.2:重建后语音与原信号时域对比图

### 3.3.3 语音识别

语音识别核心步骤为：

- (1) 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。本文采取 MFCC 算法。
- (2) 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。本文采用 CNN 实现模型。
- (3) 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。
- (4) 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。

最后小车的移动轨迹无误：

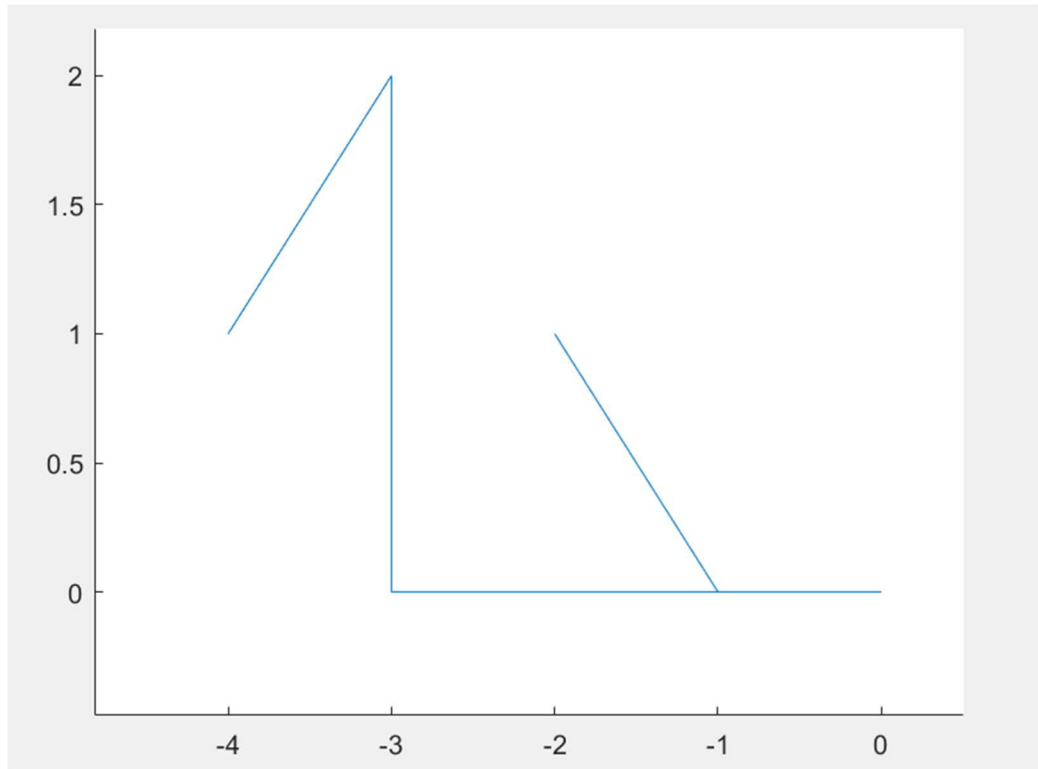


图 3.3.3:小车轨迹图