浙江大学

本科实验报告

课程名称：信息、控制与计算

姓　　名：袁东琦

学　　院：信息与电子工程学院

　　系：信电系

专　　业：信息工程

学　　号：3200103602

指导教师：张朝阳

2022 年 12 月 21 日

专业：___信息工程_____
姓名：___袁东琦_____
学号：___3200103602___
日期：___2022/12/21_____

浙江大学实验报告

课程名称：_信息、控制与计算_ 指导老师：__张朝阳_ 实验名称：__远程声控系统_

一、实验目的和要求

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。

二、实验内容和步骤

1、 使用麦克风录制一段音频，作为信源

2、 对信源进行量化和编码

3、 对信道进行编码

4、 对信源进行调制和加噪

5、 对信源进行解调

6、 对信道进行译码

7、 对信源进行译码并重建

8、 对重建的信号进行语音识别

9、 将识别得到的命令传输给控制系统，系统做出相应动作

三、主要仪器设备

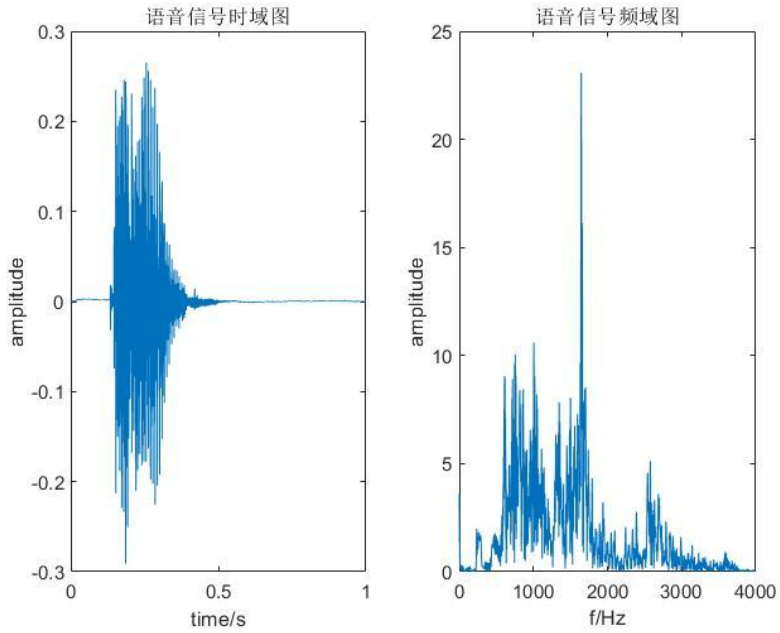
MATLAB。

四、实验过程

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。我们调用 MATLAB 自带的 audiorecorder 函数来录制音频。

设置采样频率为 8000Hz，量化位数为 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“down”为例，通过信号采集，我们得到了信号的时域图和频域图。如下图所示。



由上图可知，语音信号的幅度在-0.3~0.3V 之间，频率在 0~4000Hz，属于正常人说话的频率区间。

2、对信源进行量化和编码

由于对语音信号进行采集时已经设定量化位数为 16 位，因此量化器分辨率为：

$$q = \frac{V_{max} - V_{min}}{2^{nbits}} = \frac{0.3 - (-0.3)}{2^{16}} = 9.155 \times 10^{-6} V$$

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。在本实验中，我们采用霍夫曼编码对信源进行编码。霍夫曼编码是一种不等长无损编码，编码步骤为：

1) 将信息源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。

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，下图展示部分编码结果：

```
109206 : 1 0 0 0 0 1 1
109210 : 0 0 1 0 0 1 0
109213 : 0 0 0 1 1 0 1
109216 : 0 0 1 1 0 0 1
109220 : 0 0 0 0 1 0 0
109223 : 0 0 0 1 0 1 0
109226 : 0 0 1 1 1 0 1
109230 : 0 0 1 0 0 0 0
```

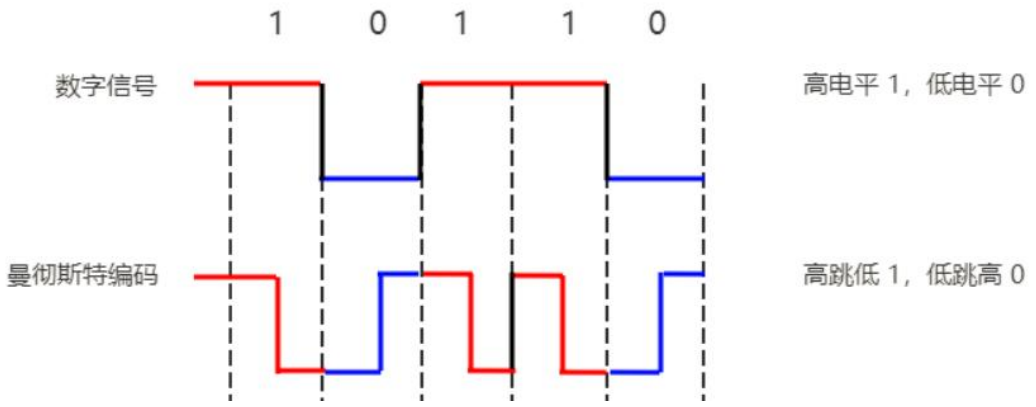
```
112440 : 0 1 1 1 0 0 1 0 0 0 0 1 1
112470 : 0 1 1 1 0 0 1 0 0 0 0 1 0
112480 : 0 1 1 1 0 0 1 0 0 1 1 0 1
112483 : 0 1 1 1 0 0 1 0 0 1 1 0 0
112530 : 0 1 1 1 0 0 1 0 0 1 1 1 1
112560 : 0 1 1 1 0 0 1 0 0 1 1 1 0
112566 : 0 1 1 1 0 0 1 0 0 1 0 0 1
112573 : 0 1 1 1 0 0 1 0 0 1 0 0 0
```

由上图可知，对于出现次数较多即出现概率较大的信号编码获得的码字长度较短，而出现次数较少即出现概率较小的信号编码获得的码字长度较长，平均码长为 8.74450，信源熵为 8，716394，编码效率为 0.996757，压缩率为 1.829669，较好的实现了压缩的功能。

平均码长 ： 8.744750
信源熵 ： 8.716394
编码效率 ： 0.996757
编码前字符串总长度 ： 128000
编码后字符串二进制总长度 ： 69958
压缩率 ： 1.829669

3、对信道进行编码

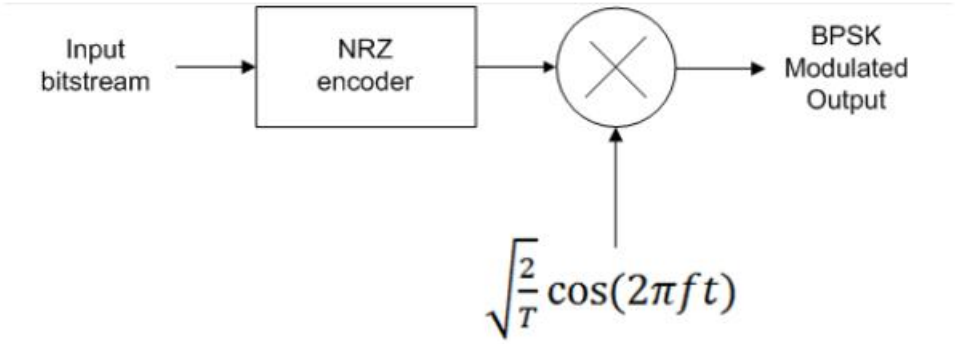
由于实际信道存在噪声和干扰，使得经过信道传输后收到的码字和发送码字之间存在差错。一般情况下，信道噪声和干扰越大，码字产生差错的可能性也越大。信道编码的目的在于改善通信系统的传输质量，发现或者纠正差错，以提高通信系统的可靠性。在本实验中，我们采取曼彻斯特编码对信道进行编码。曼彻斯特编码，常用于局域网传输。在曼彻斯特编码中，每一位的中间有一跳变，位中间的跳变既作时钟信号，又作数据信号；从高到低跳变表示“0”，从低到高跳变表示“1”，即“10”表示“1”，“01”表示“0”。



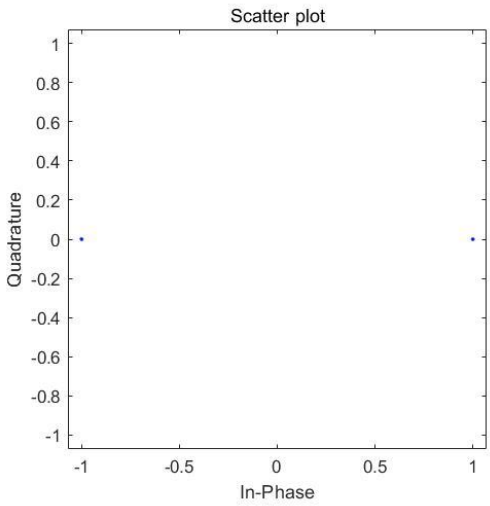
下图为部分编码结果。

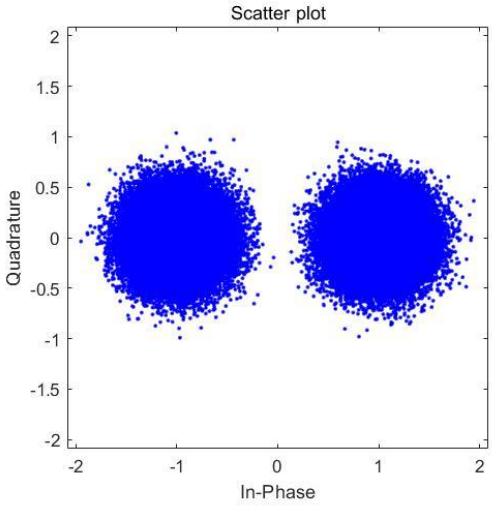
10101010010110101010010101010101101001101010010110100101100110010101101010010110011001011010100110101010101

4、调制和加噪



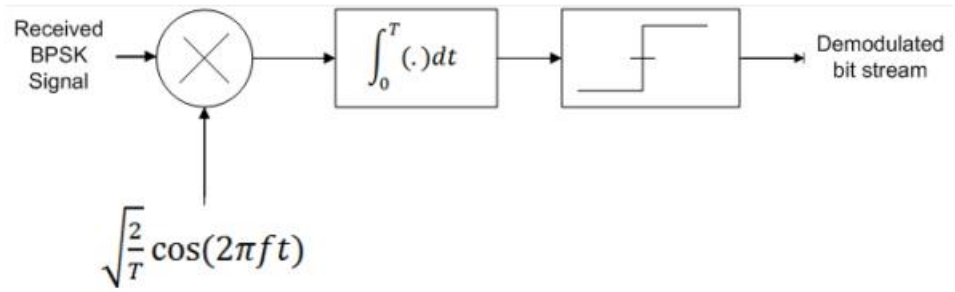
上图为 BPSK 调制的基本原理图，在 BPSK 中，通常用初始相位 0 或 π 来表示二进制的“1”和“0”。在 MATLAB 中，我们可以调用 pskmod 实现 BPSK 调制，调用 awgn 在信号中加入高斯白噪声，信噪比为 10dB。





上述左图为未添加高斯白噪声所得到的 BPSK 调制结果，右图为添加高斯白噪声后所得到的 BPSK 调制结果，由图中的结果可以看出添加的噪声均值为 0，方差为 1，与高斯白噪声的定义一致。

5、解调



上图为 BPSK 解调原理图，在 MATLAB 中我们调用 pskdemod 函数进行解调，解调得到的码字序列误码率为 0。

出错数 ： 0

误码率 ： 0.000000

6、信道译码

由于我们信道编码使用的是曼彻斯特编码，因此我们只需依次判断两位的码字为“10”或“01”，即可将序列进行译码。具体实现如下图所示。

```
for i = 2:2:length(y_channel_MC)
    if(y_channel_MC(i-1)==1&&y_channel_MC(i)==0)
        y_dechannel(i/2)=1;
    else if(y_channel_MC(i-1)==0&&y_channel_MC(i)==1)
            y_dechannel(i/2)=0;
        end
    end
end
```

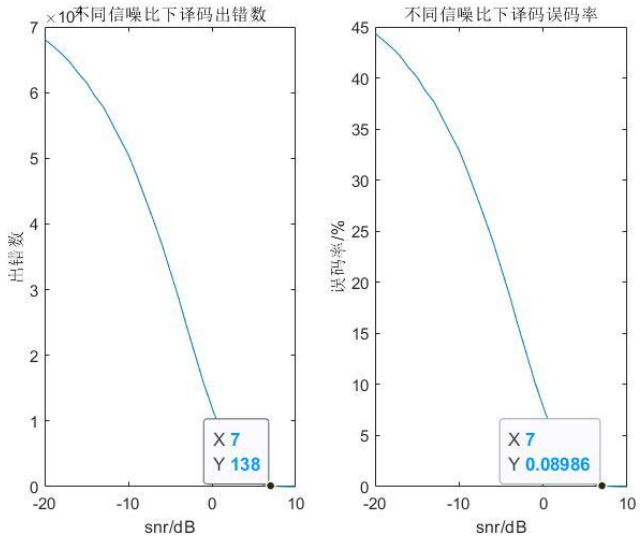
部分译码结果如下所示，误码率为 0。

```
1111001111000000110111001100101000111100101001110111111110011011101100100000110000101111000000001011110111001
```

出错数 ： 0

误码率 ： 0.000000

我们还探究了在不同信噪比下译码的准确程度，结果如下图所示：



由图中可以看出，当信噪比较小时译码的误码率较高，在信噪比为-20 时误码率接近 50%；当信噪比大于等于 7 时，误码率接近于 0，译码结果较为准确。

7、信源进行译码并重建

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，误码率为 0。

重建的目的是把译码得到的离散数字信号恢复成模拟电压信号。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 0.3，以使电平信号在-0.3V 到 0.3V 之间。

均方误差为 ： 0.000000

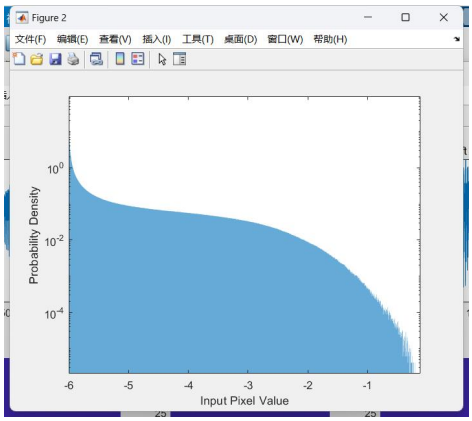
重建信号与原始信号之间的均方误差为 0，说明信号完全恢复。

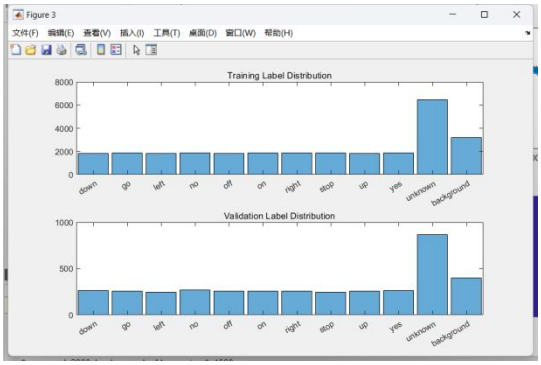
8、语音识别

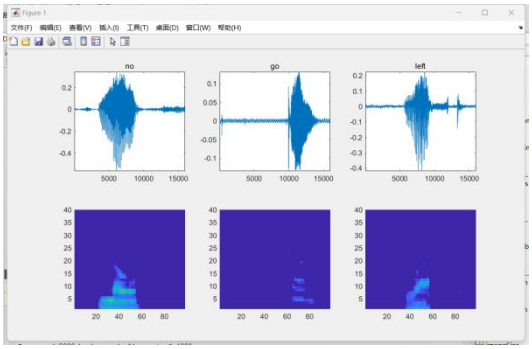
语音识别的其目标是用计算机自动将人类的语音内容转换为相应的文字。在 MATLAB 中，我们通过 Deep Learning Toolbox 来实现语音识别的功能。在该工具箱中已经有预训练好的语音识别模型 commandNet.mat，但为了更好的了解语音识别的过程，我们通过下述命令打开了语音识别实例，下载并解压 speech_commands_v0.01.tar.gz 文件后，运行程序，得到如下图像。

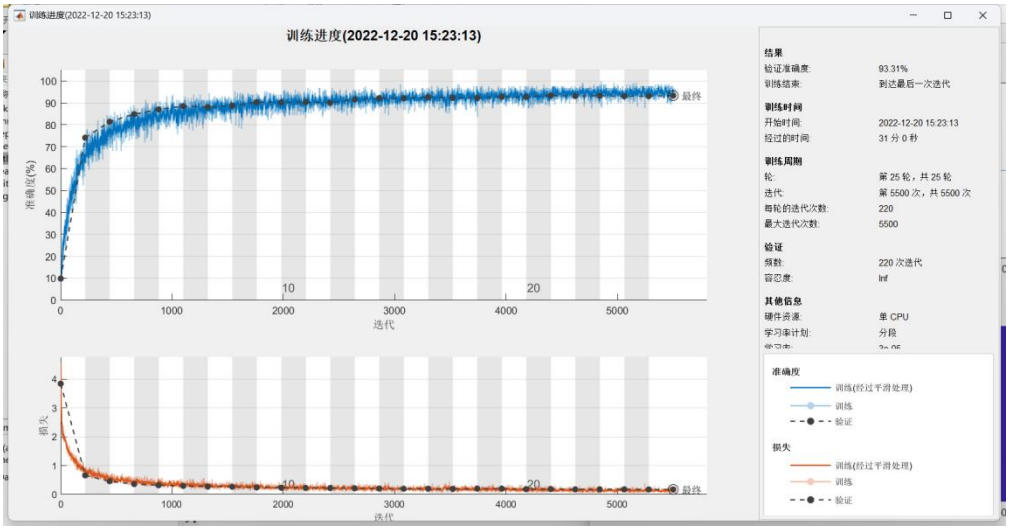
```
openExample('deeplearning_shared/DeepLearningSpeechRecognitionExample')
```

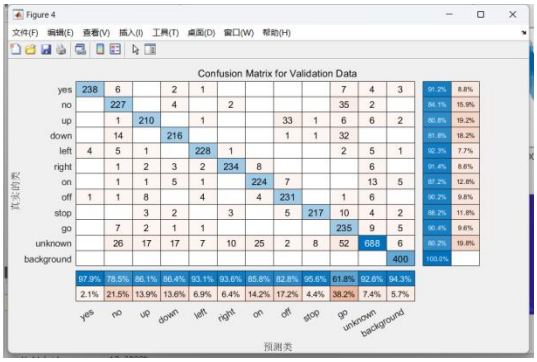
down	2359
go	2372
left	2353
no	2375
off	2357
on	2367
right	2367
stop	2380
unknown	8242
up	2375
yes	2377



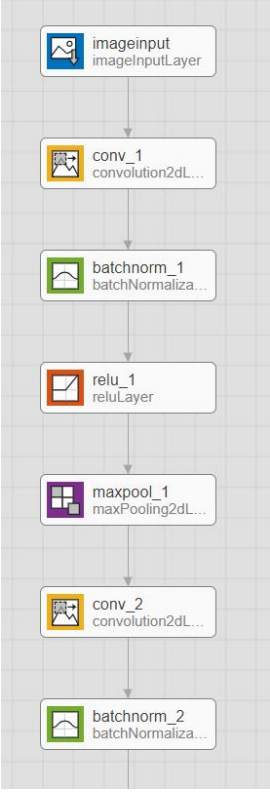




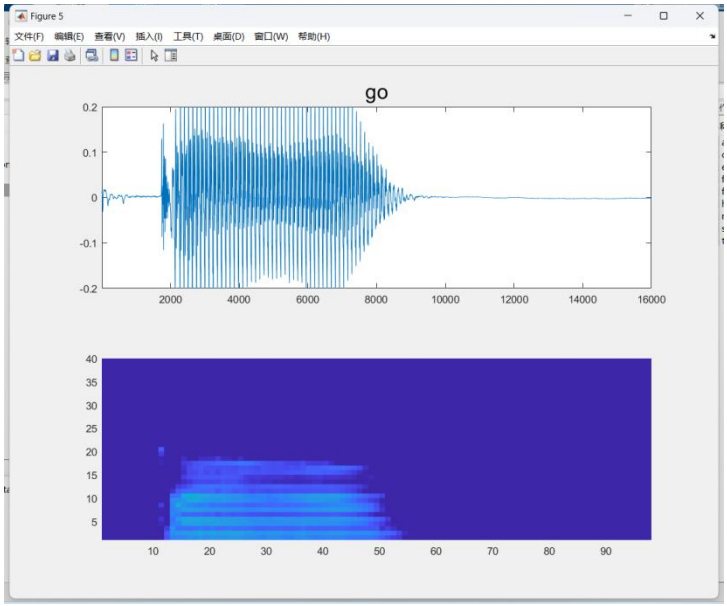




模型网络部分结构如下图所示：



上述图像显示了模型的训练进度以及预测结果，下述图像为使用训练完成后的模型进行实时语音识别得到的结果。



由于上述训练最终得到的是实时语音识别，与我们的实验所需有一定的差别，因此我们通过其他函数来实现本实验中的语音识别。下面介绍语音识别的基本流程：

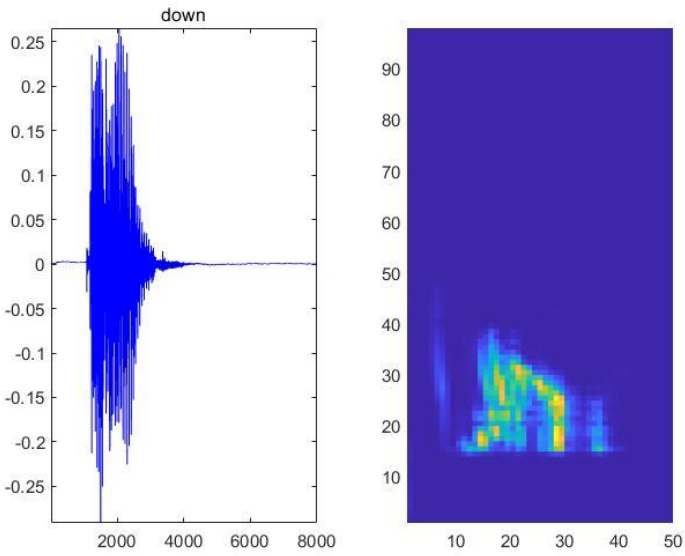
1）通过 load 函数导入预训练模型的参数。

2）通过 extractAuditorySpectrogram 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。

由于实验时使用的 MATLAB 的版本为 2019b，因此在使用 extractAuditorySpectrogram 函数时缺少了 trimOrPad 函数，需要自行添加。另外，通过 DeepLearningSpeechRecognitionExample 得到的预训练模型对语音识别特征矩阵的要求是

40*98，因此需要将 extractAuditorySpectrogram 函数中的参数 numBands 的值改为 40。识别结果如下图所示。



9、动作控制

由于没有硬件设备可以仿真指令的实现，我们编写了一个简单的程序来模拟控制系统对指令的执行具体指令，对于正确的指令，小车进行语音播报并执行相应动作，对于错误指令或无法识别的指令，小车进行语音播报但不会执行动作。如下所示：

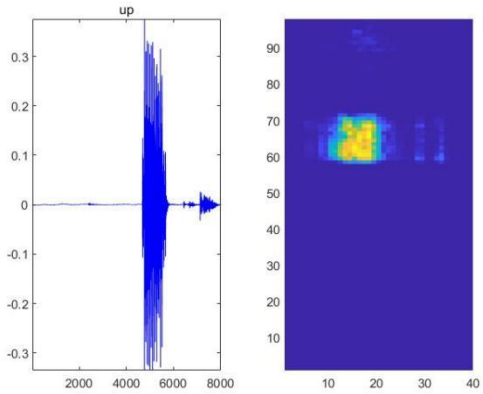
```
case 'go'
    control='小车前进';
case 'right'
    control='小车右转';
case 'up'
    control='小车抬起前轮';
case 'left'
    control='小车左转';
case 'down'
    control='小车放下前轮';
case 'stop'
    control='小车停止';
```

```
case 'unknown'
    control='The command is unknown.';
case 'background'
    control='The command is background noise.';
case 'on'
    control='The command is not right';
case 'off'
    control='The command is not right';
case 'yes'
    control='The command is not right';
case 'no'
    control='The command is not right';
```

运行结果如下所示：

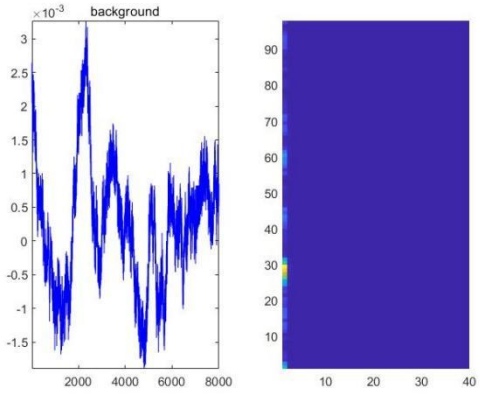
```
go

小车前进
```

对于其他的指令，该系统也可正确识别，下图展示几种识别结果：



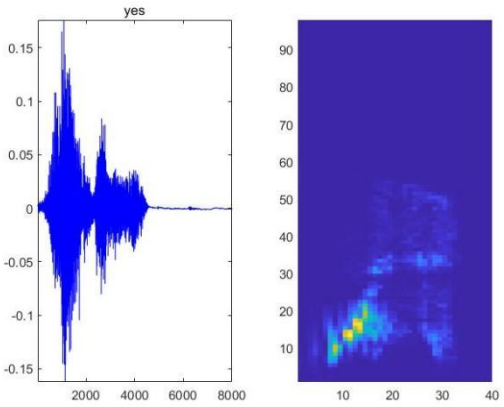
up

小车抬起前轮



background

The command is background noise.



yes

The command is not right

说明该控制系统能够正确识别并执行对应的指令。

五、总结与体会

通过本次实验，我了解了信息编码、传输、计算、控制的基本过程，对课本上的知识也有了更加全面系统的理解。本次实验的主要目标是搭建一个信息的传输与识别系统，通过合适的编码方式对信源和信道进行编码，可以有效的压缩传输的比特数，提高传输信息的准确度，通过导入 MATLAB 预训练的语音识别模型，可以较准确的识别出输入的语音指令。在做语音识别模块时也遇到一些困难，比如在运行 DeepLearningSpeechRecognitionExample 后发现只能实现实时的语音识别，对里面的代码进行改写后虽然可以对录制好的音频进行识别，但识别效果不尽人意，因此又通过在 mathworks 上搜索其他可用于语音识别的函数，最终找到现在使用的 extractAuditorySpectrogram 函数，完成语音识别的任务。总的来说，通过此次实验，我对这门课的内容了解更深，对 MATLAB 以及相关工具箱的使用也有了更多的了解，

但由于深度学习相关的内容并不是很了解，因此对语音识别模块依旧只是一知半解，之后也会学习更多这个方面的知识，进一步拓宽自己的知识面。