浙江大學

本科生设计报告

学年、学期：2021 — 2022 学年 秋冬 学期

课程名称：信息、控制与计算

任课教师：张朝阳

题　　目：远程声控系统

学生姓名：黄嘉欣

学　　号：3190102060

目录

信息、控制与计算：远程声控系统

3190102060, 黄嘉欣, 信工 1903 班

***摘要*—本远程声控系统可以利用指令语音对远端的系统进行控制，其主要由预处理系统、通信系统、控制系统三个部分组成。其中，预处理系统采集模拟语音信号并将其量化为数字信号，以供计算机处理；通信系统实现信号的远程传输，包括信源编/解码、信道编/解码、调制解调以及信道传输，主要涉及到 Huffman 编码、BCH 码、二进制相移键控（BPSK）、加性高斯噪声（AWGN）信道；控制系统利用卷积神经网络对输入的语音信号进行识别，得出指示并控制系统完成相应动作。**

***关键词*—Huffman 编码，BCH 码，BPSK，AWGN，PID**

I. 系统概述

本项目实现的是一个远程声控系统，其能够对采集到的语音指示信号进行适当压缩，通过噪声信道实现远程传输。远端接收到信号后，可经过适当计算识别出其是何指示，最后控制系统完成相应动作。因此，可以将此声控系统的功能分解成三个部分：信号采集、信号传输、信号识别，分别对应于预处理系统、通信系统、控制系统等三个子系统，其功能具体阐述如下。

A. 预处理系统

远程声控系统的入口，主要完成语音信号的采集和量化操作。使用者输入的语音是模拟信号，必须经数字化处理后才能在通信系统中传输。

B. 通信系统

远程声控系统的主体部分，主要完成信号的传输，其结构如图 Fig. 1 所示。根据传统的信号传输模型，在发送端，为了使信源减少冗余、有效传输，需要对数据进行压缩，即信源编码；与此同时，为了提高系统的抗干扰能力和纠错能力，可以增加校验码等冗余，即信道编码。通过调制，我们可以提高信号的抗噪能力，使其传输更远，更符合实际应用需求。通过噪声信道后，相应地，在接收端收到的信号，需要进行解调、信道解码、信源解码、信号重建等操作，方才能被还原为原始的语音信号。

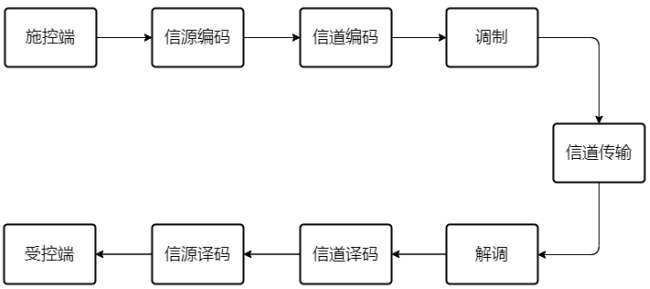


Fig. 1. 传统通信系统的结构图。

C. 控制系统

远程声控系统的执行部分，主要完成远端的控制，可被细分为语音识别和系统控制两个模块。语音识别模块采用简单的卷积神经网络，可对语音信号的频谱进行分类处理，进而识别出对应指令；系统控制模块将指令作为远端系统输入，控制系统对不同指令执行相应动作，实现完整的系统功能。

整个系统的伪代码如下：

算法 1 RemoteVoiceControlSys

输入： 模拟语音指令$voice$

输出： 相应控制操作$signal$

$sampledSignal \leftarrow$ signalsampling($voice$) // 语音信号的采样
$quantizedSignal \leftarrow$ signalquantization($sampledSignal$) // 语音信号的量化
$sourceCodes \leftarrow$ sourceencode($quantizedSignal$) // 信源编码
$channelCodes \leftarrow$ channelencode($sourceCodes$) // 信道编码
$modulatedCodes \leftarrow$ modulation($channelCodes$) // 调制
$transmittedCodes \leftarrow$ channeltransmission($modulatedCodes$) // 信道传输
$demodulatedCodes \leftarrow$ demodulation($transmittedCodes$) // 解调
$channelDecodes \leftarrow$ channeldecode($demodulatedCodes$) // 信道解码
$sourceDecodes \leftarrow$ sourcedecode($channelDecodes$) // 信源解码
$signal \leftarrow$ speechrecognition($sourceDecodes$) // 语音识别

系统控制模块将根据语音识别结果对具体的远端系统进行控制，故不用代码描述。

II. MATLAB 环境配置

1) Deep learning Toolbox
2) Statistics and Machine Learning Toolbox
3) System Identification Toolbox
4) Audio Toolbox

III. 预处理系统

当使用者发出指令控制远端系统时，系统得到的输入为模拟语音信号，其无法在数字系统中处理和传输。考虑到稳定性和实用性需求，在系统最前端，我们必须将输入系统的模拟信号数字化，其主要包括语音信号的采样和量化两个过程。

A. 语音信号的采样

1) 系统原理

由 A/D 转换原理，在将模拟信号转换为数字信号时，需要先对其进行采样，即按照一定时间间隔采样获得时间上离散的信号。因为人耳可听的频率范围最高为20kHz，为了验证后续信号的重建效果，根据奈奎斯特采样定理，我们可以使用44.1kHz 的采样频率进行采样，从而将输入的语音信号在时间上离散化。

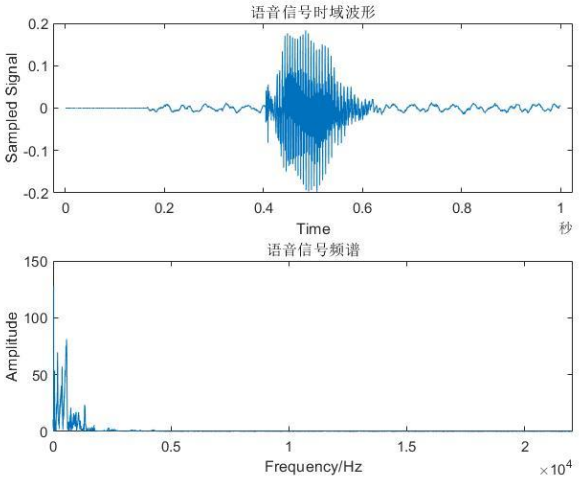


Fig. 2. 采样后语音信号的时域和频域波形。

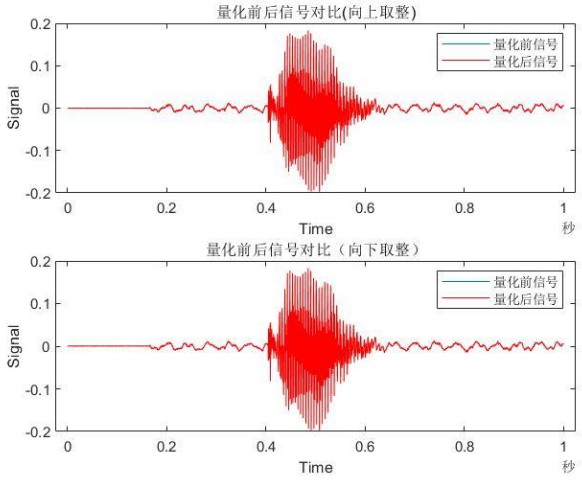


Fig. 3. 采用向上取整和向下取整量化后的信号与量化前信号的对比图。

2) 系统的 MATLAB 实现

在 MATLAB 中， audiorecorder 函数可用来录制来自输入设备的音频数据，其会根据我们设置的参数对语音信号进行采样，结合 getaudiodata 即可将录制的音频数据转换为数字信号。为了使精度更高，我们设置采样位数为 16 位，声道为单声道，录制时长为 1 秒。核心代码如下：

```
myRecording = audiorecorder(Fs, nBits, nChannels);
disp('Start speaking.');
recordblocking(myRecording, 1); % 录制时长为1s
disp('End of Recording');
play(myRecording); % 播放录制的音频
sampledSignal = getaudiodata(myRecording); % 获取录音数据
plot(sampledSignal); % 绘制语音信号
audiowrite('../command.flac', sampledSignal, Fs); % 存储语音信号
```

3) 结果分析

利用 MATLAB 录制输入指令“go”，得到的信号时域波形和频谱如图 Fig. 2 所示。可以看到，采样后的语音信号的时域幅度约在−0.2~0.2之间，频谱主要集中在0~2kHz以内。

B. 信号的量化

1) 系统原理

当模拟语音信号经采样成为离散时间信号后，需要进一步将其幅值离散化，使其连续取值近似为有限多个离散值。不妨设信号的最大幅度值为 Max，最小幅度值为 Min，当采用均匀量化，且量化位数为 n 时，则可得量化的最小分辨率为：

$$\text{delta} = \frac{\text{Max} - \text{Min}}{2^n}$$

即每两个量化取值之间的间隔为 delta。当然，在对连续值进行判决时，也有多种处理方法。若采用向上取整策略，则需要判断采样信号每个点的幅值位于哪个量化区间，并取区间的中点进行量化；若采用向下取整策略，则判断该点幅值是否位于某个量化取值±0.5delta范围内，并取该量化值进行量化。当采用非均匀量化时，则将需要根据信号的取值区间变化量化间隔。在本项目中，量化将主要采用均匀量化。

2) 系统的 MATLAB 实现

根据 *1)*中所述的量化方法和判决手段，可令量化位数 nQuantizationBits 为 8 位，实现 MATLAB 核心代码如下：

```
Min = min(sampledSignal); % 找出信号的最小值
Max = max(sampledSignal); % 找出信号的最大值
delta = (Max-Min)/2^nQuantizationBits; % 最小分辨率
for i = 1:2^nQuantizationBits+1
    q(i) = Min + delta*(i-1); % 量化取值
end

% 向上取整
N = length(sampledSignal); % 采样语音信号的每个点索引
quantizedSignal = sampledSignal;
for i = 1:N
    for j = 1:2^nQuantizationBits
        % 若信号某点取值在两个量化值之间，则取其量化值的中间值
        if sampledSignal(i)>=q(j) && sampledSignal(i)<=q(j+1)
            quantizedSignal(i) = (q(j)+q(j+1))/2;
            break
        end
    end
end

% 向下取整
N = length(sampledSignal); % 采样语音信号的每个点索引
quantizedSignalRoundDown = sampledSignal;
for i = 1:N
    for j = 1:2^nQuantizationBits+1
        % 若信号某点取值在量化值正负半个量化区间内，则取该量化值
        if sampledSignal(i)>=q(j)-0.5*delta && sampledSignal(i)<=q(j)+0.5*delta
            quantizedSignalRoundDown(i) = q(j);
            break
        end
    end
end
```

3) 结果分析

如图 Fig. 3，采用向上取整和向下取整方法量化后的信号与量化前的语音信号基本一致。在后续系统运行时，我们将使用向上取整获得的数字信号。当然，需要注意的是，当使用 MATLAB 录制指令时，其已经对模拟语音信号进行过采样和量化（即程序中 sampledSignal 实际已量化过）。为了能够体现出区别，我将 *A* 中采样时的采样位数设为 16 位（幅值量化间隔基本很小），而 *B* 中量化时的量化位数

设为 8 位，从而模拟出信号从连续幅值到离散幅值的过程。

IV. 通信系统

当语音指令数字化后，便可以通过数字系统将其传输到远端。由传统的传输模型，从施控端到受控端，信号需要经过信源编码、信道编码、调制/解调、信道解码、信源解码等过程，各部分具体设计如下。

A. *信源编码*

1) *系统原理*

由 II 中处理可知，对单次的语音输入而言，经过 A/D 转换后得到的数字信号在时间上和取值上都已经离散；且其前后输出的幅值都独立、不相关，输出序列的概率分布与起点无关。因此，对通信系统来说，当施控端连续发出指令时，每条语音对应的数字信号都可看作一个平稳离散无记忆信源，其量化后的不同幅值构成一符号表，使得我们可以用在有限字符集上取值的独立随机变量序列作为模型，利用相关知识对其进行理论分析。除此之外，考虑到系统的准确性和有效性需求，我们将对信源消息进行无损不等长编码，这里将采用二元 Huffman 编码，以在准确传输信号的同时获得最短的平均码长，更好地减少冗余。

根据 II 中的量化结果，此离散无记忆信源U产生的输出序列长度为$L = 44100$，信源字符表由$K = 255$个符号组成。利用 MATLAB 可以算出信源每个符号出现的概率$p_i(i = 1, 2, \dots\dots, K)$，并得到信源的熵（熵速率）：

$$H(U) = -\sum_{i=1}^{K} p_i \log(p_i) = 4.872872\ bit/\text{符号}$$

即平均每个信源符号所携带的信息量为$4.872872\ bit$。

熵的相对率为：

$$\eta = \frac{H(U)}{\log(K)} = 0.609539$$

冗余度为：$R = 1 - \eta = 0.390461$

在对信源符号进行二元 Huffman 编码时，需要根据每个符号的概率大小进行排序，将概率最小的两个符号排在最后两位，标以 0 和 1。如此重复，最后得到仅有两个符号的辅助源，其中一个标以 0，另一个标以 1。最后从反方向开始进行码字分配，得到各个符号对应的码字。之后可将信源输出序列中各符号用分配的相应码字替换，完成编码。

2) *系统的 MATLAB 实现*

MATLAB 内置的 huffmandict 函数可根据已知符号及其概率分布生成对应的 Huffman 码字，因此，在使用该函数前，需要先统计出信源符号的出现概率。生成码书后，再利用 huffmanenco 函数完成编码工作。相关的 MATLAB 代码如下：

```
% 计算信源概率分布及信源熵
L = length(quantizedSignal); % 信源输出序列长度
symbols = unique(quantizedSignal); % 统计信源字符表
K = length(symbols); % 字符数
U = zeros(3,K); % 信源字符概率表
U(1,:) = symbols; % 第一行为字符
for i = 1:K
    U(2,i) = length(find(quantizedSignal==U(1,i))); % 第二行为出现次数
    U(3,i) = U(2,i)/L; % 第三行为出现概率
end

% Huffman编码
[dict, avglen] = huffmandict(U(1,:),U(3,:)); % 得到每个符号对应的码字及平均码长
sourceCodes = huffmanenco(quantizedSignal, dict); % 将信号中各符号替换为码字
```

3) *结果分析*

根据 huffmandict 的输出结果，可得 Huffman 编码的平均码长为$\bar{n} = 4.908163$，基于此，可以算出编码速率为：

$$R = \bar{n} log D = \bar{n} = 4.908163\ bit/\text{符号} > H(U)$$

即每个信源符号需要的编码符号比特数为4.908163，可达；相对应的编码效率为：

$$\eta = \frac{H(U)}{R} = 0.992810$$

可见采用 Huffman 编码的效率是比较理想的，可以用于实际使用。经过编码后得到的输出序列码长为 216450，而若采用量化后的 8 位编码，得到的序列码长为 352800，压缩率为1.629938（码书和信源编码结果可见 data 文件夹中的 SourceEncode.txt 文件）。

综上，考虑到系统的处理时间和运行效率，在对输入信号进行量化后，我们并没有立即对其进行 8 位量化编码，而是对离散的幅值进行 Huffman 编码。相比于量化后的 8 位编码，通过 Huffman 信源编码后的序列码长明显缩短，编码速率与信源的熵速率几乎相同，信号冗余被有效剔除，压缩效果符合预期。

B. *信道编码*

1) *系统原理*

由于信道中噪声的存在，信号从发送端传输到接收端往往会出现偏差。因此，我们需要在信源编码的基础上向码字序列中添加一定的冗余，以增加系统的抗噪声能力。与此同时，为了模拟真实的传输环境，在信道部分，系统将采用平稳无记忆高斯信道，其离散时间输出Y_i是输入X_i和高斯随机变量Z_i之和。若信道输入信号的功率限制为P，噪声方差为N，则高斯信道的容量：

$$C = \frac{1}{2}\log\left(1 + \frac{P}{N}\right)\ bit/\text{次}$$

当且仅当输入X为正态分布，$X \sim N(0, P)$时，达到信道容量。同时，信道编码的速率R必须小于C，才能确保信号在信道中能够无差错地传输。

在系统中，为了获得更强的抗干扰能力，信道编码部分将使用$(15, 7)$BCH 码，其编码长度$n = 15$，信息位$k = 7$，校验位为 8 位，能纠正$t = 2$位错误。该 BCH 码由八进制表示的多项式“721”生成，对应的生成多项式为：$g(x) = x^8 + x^7 + x^6 + x^4 + 1$。当进行编码时，需要先将输入序列

每$k=7$位分为一组，记为多项式$m(x)=m_0+m_1x+m_2x^2+\cdots+m_6x^6$，其中$m_i\in\{0,1\}$；记校验位为$r(x)$，且$r(x)=x^8m(x)\ mod\ g(x)$。最终编码得到的 BCH 码字多项式为：

$$C(x)=x^8\ m(x)+r(x)\ mod\ g(x)$$

2) 系统的 MATLAB 实现

在 MATLAB 中，BCH 码的编码过程可由函数 bchenc 完成，其会根据我们设置的参数n、k对输入列数为k的矩阵进行(n,k)BCH 编码，相对应的核心代码为：

```
% 信道编码
N = length(sourceCodes); % 信源编码序列长度
blocks = ceil(N/k); % 需要分成的组数
if (blocks>N/k)
    sourceCodes(N:blocks*k) = 0; % 若不够，则补0
end
sourceCodes = gf(reshape(sourceCodes, k, blocks)'); % 转成blocks*k的矩阵
gfCodes = bchenc(sourceCodes, n, k); % BCH编码(GF域)
channelCodes = double(gfCodes.x); % BCH编码(实数域)
channelCodes = reshape(channelCodes', [], 1); % 转成列向量
```

3) 结果分析

根据编码结果，经过$(15,7)$BCH 编码后得到的输出序列码长为 463830，而输入的信源编码序列码长为 216450，其编码速率为：

$$R=\frac{k}{n}=0.46667\ bit/次$$

可见，为了提高系统的抗噪声能力和纠错能力，信道编码的冗余程度较高，每次传输能够传递的信息量很少，编码效率较低。因此，当信道信噪比较低时，其仍有很大的概率速率可达，经过信道以后，理论上误码率将会很低，信息的准确性会得到比较好的保障（信道编码结果可见 data 文件夹中的 ChannelEncode.txt 文件）。

C. 调制

1) 系统原理

在信号进入信道以前，往往需要对其进行调制，将其变换成适合信道传输的信号后再进行传输，以提高系统的抗噪声能力。在本系统中，我们将采用二进制相移键控，利用二进制数字信号控制正弦载波的相位，而载波的幅度和频率保持不变，其发送的波形表达式为：

$$e(t)=Acos(2\pi ft+\varphi_n)=\begin{cases}Acos(2\pi ft), \text{输入“0”时}\\ -Acos(2\pi ft), \text{输入“1”时}\end{cases}$$

即当输入的二进制信号为“0”时，载波相位$\varphi_n=0$；当二进制信号为“1”时，载波相位$\varphi_n=\pi$。对于调制输出，若其为离散序列，不妨记某一位的值为s，则同理可得其与对应的输入值b之间存在的关系：$s=1-2b$。对输入序列的每一位进行上述变换，即可完成 BPSK 调制，最终的调制输出结果应当转化为“1”与“-1”的序列，其中“1”对应原始信号中的“0”，“-1”对应原始信号中的“1”。

2) 系统的 MATLAB 实现

根据 *1)*中所述原理，可以对信道编码后的码字序列进行逐位替换，相关核心代码如下：

```
modulatedCodes = 1-2*channelCodes; % BPSK调制
```

需要注意的是，根据实际实验结果，BPSK 调制不能使用 MATLAB 中自带的 pskmod 和 comm.BPSKModulator 函数，其调制出的结果时复数形式，虽然虚部为 0，但当后续使用 awgn 函数时，序列的实部和虚部都会被加上高斯白噪声，这必然会导致实际的噪声功率为设想的一半，即仿真结果与理论相比由 3dB 的增益，应当纠正。

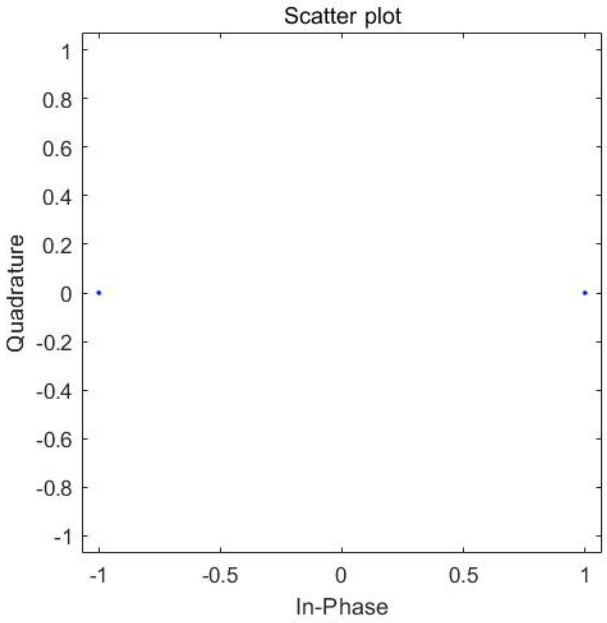


Fig. 4. BPSK 调制后的信号星座图。

3) 结果分析

如图 Fig. 4 为利用 MATLAB 绘制的信号星座图。可以看到，经过 BPSK 调制后，序列中所有值都位于 1 和-1 两个点正中央。显然，这是由于尚未通过噪声信道，得到的结果十分理想。

D. 信道传输

1) 系统原理

正如信道编码部分所述，为了与实际情况接近，我们将使信号在加性高斯噪声信道中传输，其信道容量为：

$$C=\frac{1}{2}\log\left(1+\frac{P}{N}\right)\ bit/次$$

其中P为信道输入信号的功率限制，N为噪声方差。在高斯信道，输入X_i会受到高斯白噪声的影响，使得离散时间输出Y_i是输入X_i和高斯随机变量Z_i的和，即：$Y_i=X_i+Z_i$，$Z_i\sim N(0,N)$。

除此之外，对信噪比SNR，一般定义为信号功率与噪声功率的比值，以dB作为单位。若输入信号的功率为P，噪声的功率为N，则其表达式为：

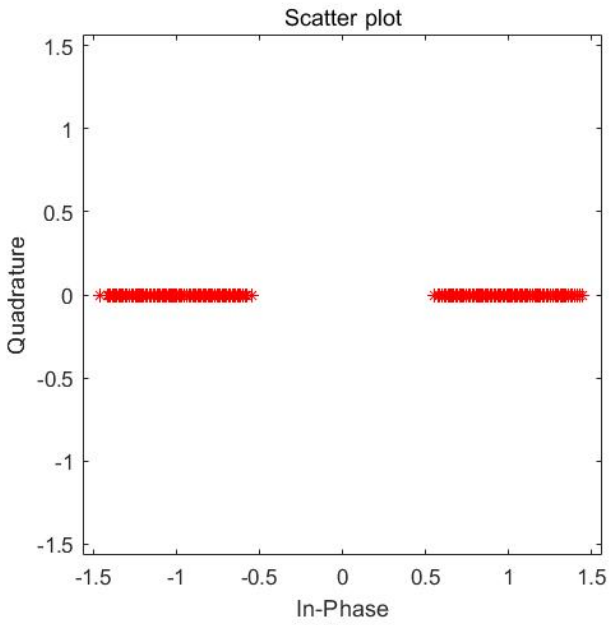


Fig. 5. 加高斯白噪声后的信号星座图，信噪比$SNR = 20\text{dB}$。

$$SNR = 10\lg\left(\frac{P}{N}\right)$$

因此，在具体使用时，需要注意SNR与信号、噪声功率直接比值之间的区别。

2) 系统的MATLAB实现

MATLAB 内置的 awgn 函数可以在信号中加入高斯白噪声，模拟其在信道中传输的过程。同时，正如 *1)*中原理所述，awgn 函数的参数 snr 可以dB作为单位，也可为线性比值，需要加以区分。在系统的实现过程中，我们将采用$SNR = 20\text{dB}$。当然，为了比较在不同信噪比条件下$(15, 7)$BCH 信道编码速率R的可达性及其误码率，我们会在后续完成信道解码后设置不同的SNR值并对解码结果进行分析。模拟信号在高斯信道中传输的相关 MATLAB 核心代码如下：

```
transmittedCodes = awgn(modulatedCodes, SNR); % 添加高斯白噪声
```

3) 结果分析

将加入高斯白噪声的输出信号与调制后的信号进行对比，如图 Fig. 5 所示，可以发现经过高斯信道后，得到的数据并未正好位于星座图中 1 和-1 位置，而是分布在两点周边约-0.5~0.5 的范围内，与预期的输出结果是相吻合的。

E. 解调

1) 系统原理

从 D 中传输结果可知，在将经噪声干扰后的调制信号解调回原始信号时，需要对其进行判决：对大于 0 的值，可以认为其是由 1 加上高斯噪声得到；对小于 0 的值，可以认为其是从-1 受噪声影响而来，进而将信道传输得到的信号还原为未受到干扰的编码。完成判决后，由 BPSK 调制的原理，调制后信号中的“1”对应原始信号中的“0”，“-1”对应原始信号中的“1”，据此即可实现解调，得到信道编码序列的观测值。

2) 系统的MATLAB实现

如 *1)*中原理所述，可以编写相关的 MATLAB 核心代码如下：

```
% 判决
tempCodes(transmittedCodes>-0) = 1;
tempCodes(transmittedCodes<0) = -1;
% 解调
demodulatedCodes(tempCodes>-0) = 0;
demodulatedCodes(tempCodes<0) = 1;
% 转成列向量
demodulatedCodes = reshape(demodulatedCodes,[],1);
```

3) 结果分析

利用 MATLAB 提供的 biterr 函数，我们将解调后的序列与信源编码后得到的序列进行了比较，发现误码数为 0，误码率为 0，表明信号经过调制、信道传输、解调等过程后可以被很好地还原，系统设计符合使用需求（解调后的结果可见 data 文件夹中的 demodulation.txt 文件）。

F. 信道解码

1) 系统原理

众所周知，信道解码是信道编码的逆过程。在对$(15, 7)$BCH 码进行解码时，需要计算接收多项式$R(x)$对应的伴随式$s(x)$，根据伴随式$s(x)$，查表寻找对应的错误多项式即陪集首项。完成后，把接收多项式和错误多项式相加就纠正了相应的错误。一般而言，若$(15, 7)$ BCH 码的错误位置在第i, j位，则错误位置多项式为：

$$s_1 x^2 + s_1^2 x + s_1^3 + s_3 = 0$$

其中$s_1 = \alpha^i + \alpha^j$，$s_3 = \alpha^{3i} + \alpha^{3j}$，$\alpha$为$\text{GF}(2^4)$的本原元。通过试探方法解出多项式的根$\alpha^i$，$\alpha^j$，即可找出错误并纠正。

2) 系统的MATLAB实现

在 MATLAB 中，BCH 码的解码过程可由函数 bchdec 完成，其会根据我们设置的参数n、k对输入列数为n的矩阵进行(n, k)BCH 解码，相对应的核心代码为：

```
% 信道解码
blocks = length(demodulatedCodes)/n; % 需要分成的组数
demodulatedCodes = gf(reshape(demodulatedCodes, n, blocks)'); % 转成blocks*n的矩阵
gfCodes = bchdec(demodulatedCodes, n, k); % BCH解码(GF域)
channelDecodes = double(gfCodes.x); % BCH编码(实数域)
channelDecodes = reshape(channelDecodes', [], 1); % 转成列向量
channelDecodes = channelDecodes(1:length(sourceCodes)); % 去除补0的值
```

3) 结果分析

与 E 中类似，将信道解码的结果与信源编码所得序列进行比较，发现误码数为 1，误码率为 0.000005，系统的抗干扰能力较好。当改变高斯信道的信噪比后，可以得到信道编码误码率与信噪比之间的关系如图 Fig. 6。当$SNR = 6\text{dB}$左右时，误码率几乎为 0，此时对应的信号功率P与噪声功率N比值为：

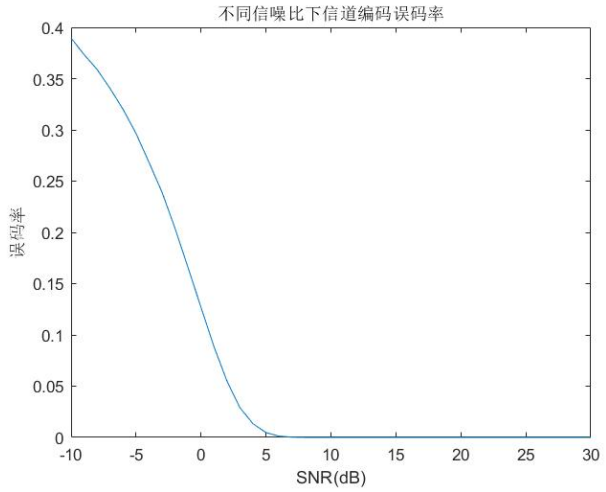


Fig. 6. 当信噪比$SNR \in [-10, 30]$dB时，信道编码的误码率。

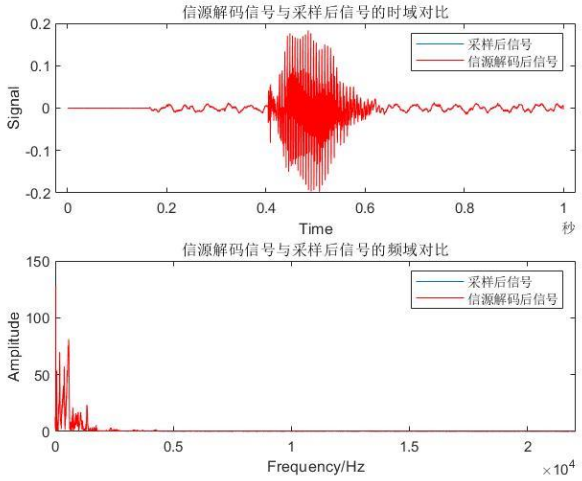


Fig. 7. 信源解码后所得信号与采样后信号在时域和频域的对比图。

$$\frac{P}{N} = 10^{0.1SNR} = 3.98$$

则高斯信道容量为：

$$C = \frac{1}{2}\log\left(1 + \frac{P}{N}\right) = 1.1582\ bit/次$$

显然，由于输入的信号非正态分布，我们在系统中无法达到信道容量值。同理，当SNR约为 2dB时，误码率在 0.05 左右，此时高斯信道的信道容量为$C = 0.5\log(1 + 10^{0.2}) = 0.685\ bit/次$，略大于 BCH 码的编码速率$R = 0.46667\ bit/次$。可见，当$R < C$时，信道编码的误码率$\lambda^{(n)} \to 0$，符合高斯信道编码定理。$SNR = 20$dB时的信道解码结果可见 data 文件夹中的 ChannelDecode.txt 文件。

G. *信源解码*

1) *系统原理*

与信源编码类似，信源解码的过程实际上是根据码书将编码序列中的码字还原为原始的序列。由于 Huffman 编码是异字头码，其唯一可译和无延时的性质使得我们可以正确识别每个长度码字的起点，而不必担心接收到序列后不能正确译出和译码延时的问题。

2) *系统的 MATLAB 实现*

MATLAB 内置的 huffmandeco 函数可以根据码书对输入序列进行 Huffman 解码，相关的 MATLAB 核心代码如下：

```
% 信源解码
sourceDecodes = huffmandeco(channelDecodes, dict);
```

3) *结果分析*

利用 find 函数，我们将信源解码后的信号与量化后的信号值进行了对比，发现两者差异很小，错误数为 1，错误率为 0.000023。如图 Fig.7，为信源解码所得信号与采样后信号在时域和频域上的比较图，可以发现两者的时域波形和频谱基本相同，信号能够通过通信系统进行准确地传输。当输入其他语音指令时，得到结果的准确度差异不大。信源解码的输出可查阅 data 文件夹下的 SourceDecode.txt 文件。

V. 控制系统

当语音指令通过通信系统传输到受控端后，并不能直接对远端系统进行控制，而是需要将其转换成电信号。对不同的指令，电信号应当存在差异，以“告诉”系统应该完成何种响应。因此，对接收到的语音信号，我们要先判断出其为何种指令，此步骤可由神经网络进行语音识别；根据不同的指令产生不同的电信号，使用一定的控制算法完成响应，将在系统控制中设计。具体如下：

A. *语音识别*

1) *系统原理*

语音识别系统本质上是一种模式识别系统，其包括特征提取、模式匹配、参考模式库等三个基本单元。通常情况下，语音识别都是基于时频分析后的语音频谱完成的，其谱图具有比较明显的结构特点。为了克服语音信号中包含的各种多样性，可以将卷积的不变性思想应用到语音识别的声学建模中，将整个语音信号分析得到的时频谱当作一张图像来处理，即利用卷积神经网络（CNN）完成语音的识别，其主要由输入层、卷积神经层、下采样层、全连接层及输出层构成。

2) *系统的 MATLAB 实现*

根据官方文档的描述，MATLAB 中内置有一个预训练语音识别网络，可用来识别{“yes”、“no”、“up”、“down”}等 10 条语音命令及背景噪声，相关核心代码如下：

```
load('commandNet.mat'); % 加载该预训练网络
% 计算听觉频谱图
auditorySpect = helperExtractAuditoryFeatures(sourceDecodes, Fs);
% 根据听觉频谱图对命令进行分类
command = classify(trainedNet,auditorySpect);
```

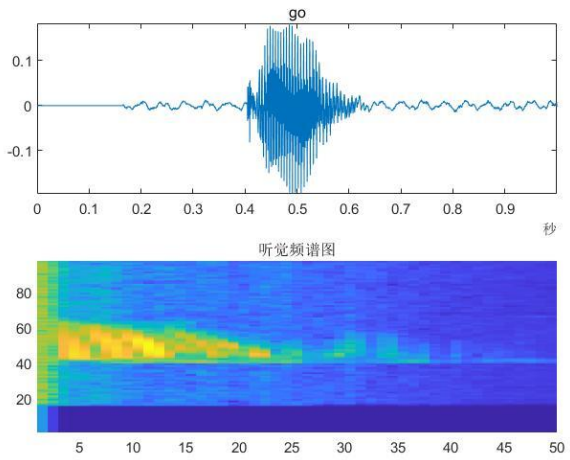


Fig. 8. 语音识别结果。

3) 结果分析

如图 Fig. 8 所示，通过 CNN 网络后，系统能够准确识别出输入的语音指令为“go”，图中的下半部分即为其听觉频谱图。综上，语音识别部分的功能符合预期。

B. 系统控制

1) 系统原理

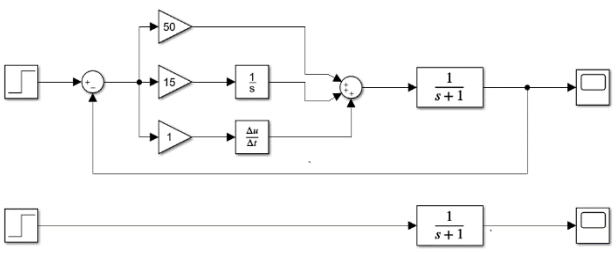
当受控端将指令正确识别后，需要将其转换为电信号以控制系统。因此，在系统控制部分，我们将预设多个指令，并为每个指令分配相应的电平响应方式。当接收到不同指令后，各种电平响应将使系统进入不同的状态，完成多种操作。除此之外，考虑一般的电机控制，为了使系统更加准确而稳定，往往需要加入反馈回路，使电平等保持在参考值。因此，在本系统中，我们将引入 PID 控制器，即比例、积分、微分控制器，其中比例用于减少偏差，积分用于消除自控系统的稳态误差，微分用于克服被控对象的滞后。根据工程经验，可以对 P、I、D 三个参数进行调整，进而调节系统的输出，使其更符合我们的需求。

2) 系统的 MATLAB 实现

不失一般性地，可以假设电机的传递函数为：

$$H(s) = \frac{1}{s+1}$$

当接收到“go”指令时，系统向电机输入阶跃信号，则利用 Simulink 仿真搭建其结构图如下：



其中上半部分为引入 PID 后的控制系统，其 P、I、D 三个参数分别为 50、15、1，下半部分为开环系统。

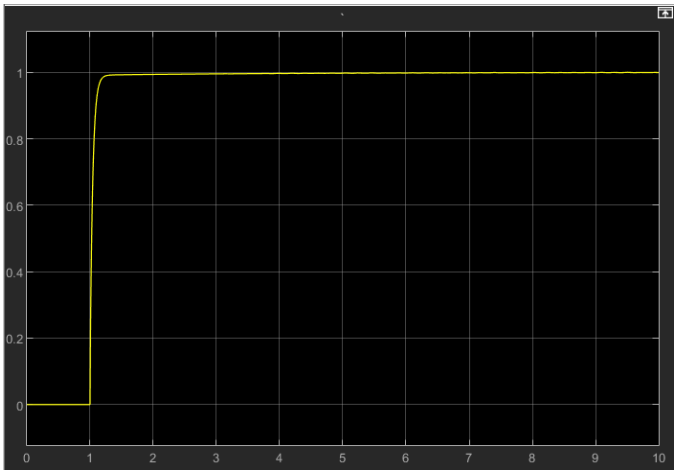


Fig. 9. 引入 PID 后系统对阶跃信号的输出。

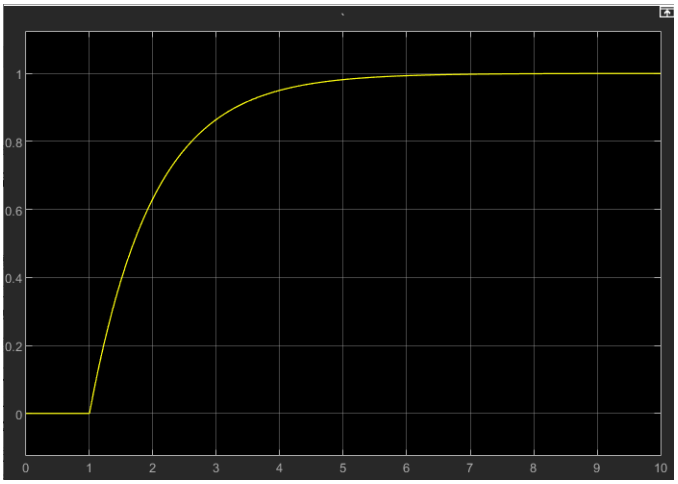


Fig. 10. 开环系统对阶跃信号的输出。

3) 结果分析

经过 Simulink 仿真，可以得到 PID 系统的输出和开环系统的输出如 Fig. 9 和 Fig. 10 所示。显然，引入 PID 反馈后，系统的响应速度更快，且不存在超调、震荡等情况，表明电机接收到“go”指令后能够更快、更准地做出响应，系统性能更佳，更适合在实际中使用。

VI. 总结

此次实验，我们在详细理解远程声控系统结构的基础上，利用 MATLAB 对其进行了仿真，不仅熟悉、掌握了信号的采样、传输、计算等过程的设计方法及其步骤，也对信道和各种编码的性质有了更好的认识，不失为一次良好的学习体验。

从实验得到的结果来看，整个系统的功能得到了完整的实现，信号传输和识别的准确度符合预期，在一定程度上能够应对使用的需要。一方面，经过通信系统的传输，信号到达受控端时的误码率几乎为 0，系统的准确性、抗干扰能力具有很好的保障。另一方面，我们也计算了不同

信噪比条件下信道编码的误码率，得出了随着信噪比上升，误码率将逐步降低的结论，在一定程度上验证了高斯信道编码定理。最后，在受控端，通过 PID 反馈控制，受控系统的响应速度、稳定性和准确性比开环系统有了很大的提升，符合预期的设计初衷。

当然，在实验的过程当中，我也遇到过一些困难，其中最主要的还是对知识的理解掌握和 MATLAB 的仿真。虽然整个过程较为繁琐，但在不断回顾所学、查阅文档和资料的过程中，我加深了对知识的掌握，理解了信息系统设计的一般准则，也对编码速率、信道容量等重要概念的性质有了更加深刻的认识，不禁由衷佩服于香农的开创性思维，对我们以后的专业生活产生了很大的启发意义。

除此之外，考虑到声控系统的简单性和即时性，可以在语音信号的预处理部分使用神经网络识别出用户输入的指令。根据预设的指令类型，我们只需要有限个字符即可表示所有指令。在信源编码时，可以根据每种字符出现的概率对其进行 Huffman 编码，得到的输出序列码长将会远小于现有的系统。同理，在通信系统的其他部分，我们所需要处理的数据量也将会大幅降低。当受控端接收到初始的字符时，可以通过查表得到用户发出的指令，再执行对应操作。可见，通过这样的系统架构，我们不仅能够缩短系统需要传输的序列长度，还可以有效减少处理时间，避免资源的浪费，但也同时面对着相对错误概率更高的问题（传输过程出错使得信源解码后得到的字符与编码前的字符不同，最终导致查表得到的指令错误），需要进行利弊的权衡。

总的来说，经过一学期的学习，我们了解了许多信息理论中的重要概念和知识，认识了计算理论和控制理论，也从这个过程当中领悟到了前人大牛在领域开创过程中的创造性思维，干货满满。在以后的学习生活中，我会继续努力用实践去验证理论所学、用工具辅助自身学习，通过仿真、设计理解知识架构，在交叉融会的过程中掌握必备的技能、培养必须的思维方式、掌握更多的专业要领，树立起大信息系统观，不断将理论与实践结合，为成为新型的高素质人才而前进。

VII. References

[1] 门爱东, 苏菲, 王雷,等. 数字信号处理[M]. 科学出版社, 2005.

[2] Oppenheim A V , Willsky A S . Signals and Systems[M]. Prentice-Hall, 2006.

[3] 文超. 信道编码中循环码的译码纠正[J]. 信息通信, 2012(1):3.

[4] Thomas M. Cover, Joy A. Thomas. Elements of Information Theory. John Wiley & Sons, New York. 2006.

[5] 仇佩亮, 张朝阳, 谢磊, 余官定,等. 信息论与编码.第 2 版[M]. 高等教育出版社, 2011.

[6] 孟利民, 朱健军, 赵新建,等. 全数字 BPSK 调制解调系统的仿真[J]. 浙江工业大学学报, 2003, 31(1):6.

[7] 张博, 张玉静. 加性高斯白噪声信道中典型信号的信噪比估计[J]. 滨州学院学报, 2015, 31(2):4.

[8] 王艳华, 袁秀湘. 加性高斯白噪声信道的最佳接收机性能研究[J]. 交通科学与工程, 2000, 16(004):19-22.

[9] 张晴晴, 刘勇, 王智超,等. 卷积神经网络在语音识别中的应用[J]. 网络新媒体技术, 2014(6):4.

[10] 王彪. 基于Matlab的语音识别系统研究[J]. 计算机与数字工程, 2011, 39(12):4.

[11] 刘金琨. 先进 PID 控制 MATLAB 仿真[M]. 电子工业出版社, 2004.

[12] 王蕾, 宋文忠. PID 控制[J]. 自动化仪表, 2004, 25(4):6.