

通信原理 实验报告

学号：2206113602
班级：信息 005
姓名：王靳朝

2 信道编解码

一 实验内容（10 分）

- 1.1 信道编解码的实现和验证
- 1.在已有的 demo 基础上，发送端合适位置加入信道编码（可自由选择编码类型及方式）
 - 2.在接收端合适位置加上对应的信道解码；
 - 3.系统测试，保证编解码的正确性，收端能得到发端信息；
 - 4.计算误码率，对比添加信道编解码和未添加信道编解码的误码率。

二 实验原理（50 分）

- 2.1 差错控制编码的分类（10 分）
- 最简单的差错控制编码是奇偶校验码，再信息位后加入一个校验位使得码流中 1 的个数位奇数或者偶数，从而在接收端发现错误。
- 此外还有根据交织编码+纠错编码=FEC 前向纠错编码，还有：
- 1. 线性分组码：由于信息位和校验位可以使用线性关系表示（矩阵），且信息位校验位分离，因此称线性分组码。
 - 2. 循环码：例如 BCH 码、CRC
 - 3. 卷积码：Turbo 码。Turbo 码主要使用于第四代移动通信技术中。第五代移动通信中的 polar 码由中国提出并写入国际规范。

- 2.2 线性分组码的编码、校验、纠错原理（20 分）
- 以 74 汉明码分析线性分组码的工作原理

Bit #	1	2	3	4	5	6	7
Transmitted bit	p_1	p_2	d_1	p_3	d_2	d_3	d_4
p_1	Yes	No	Yes	No	Yes	No	Yes
p_2	No	Yes	Yes	No	No	Yes	Yes
p_3	No	No	No	Yes	Yes	Yes	Yes

线性分组码可以用 (n,k) 表示，其中 n 是编码总长度， k 是信息位个数， $r=n-k$ 为监督位码长。线性分组码的生成和作用过程可以用两个矩阵表示，即生成矩阵 G 和校验矩阵 H ，设计一

个线性分组码的过程就是设计生成矩阵 G 的过程。

1. 编码原理 生成矩阵 G

设海明码为 $[d_1 d_2 d_3 d_4 p_1 p_2 p_3]$ 生成矩阵 G 为

$$[d_1 d_2 d_3 d_4] \cdot G_{4 \times 7} = [p_1 p_2 p_3 d_1 d_2 d_3 d_4]$$

根据 ppt 中的图得出生成矩阵 G 为

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

$\begin{matrix} & & d_1 & & d_2 & d_3 & d_4 \\ & & \uparrow & & \uparrow & \uparrow & \uparrow \\ & & & & & & \\ p_1 & p_2 & & & p_3 & & \end{matrix}$

2. 校验原理 校验矩阵 H

$$[p_1 p_2 p_3 d_1 d_2 d_3 d_4] \cdot H_{7 \times 3}^T = S_{1 \times 3} \quad S \text{ 为伴随阵}$$

S 向量解读即为发生错误的码字位序。例如当 $S=[100]$ 时代表 1 发生错误

$S=[101]$ 时 d_2 发生错误 $S=[000]$ 时无错误

由图 1 知 H 为

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

3. 纠错原理

在 H 中出现了代表错误的向量 S ，将对应的位取反即可

2.3 码间串扰产生的原因 (10 分)

码间串扰的定义是由于信道特性的不理想以及信道中加性噪声的存在，会存在抽样判决时刻的误码现象。

由于信道的带宽限制，当信号在有限带宽信道中传输时，会导致码元信号在时域上的拖尾，拖尾叠加到其它码元上影响其它码元的正确判决，称之为码间串扰 ISI。在二元幅度调制系统中，用收端信号的眼图来观察 ISI；在多元幅相调制系统中，用收端信号的星座云来观察 ISI。

信道的带限特性是客观实际存在的，实际中都是带限系统，所以不能完全消除信号的拖尾，但是可以想办法让拖尾按照我们想要的方式出现，即不影响其它码元的正确判决。

无码间串扰频域条件：

$$\sum_i H(\omega + \frac{2\pi i}{T_s}) = T_s (\text{或常数}) \quad \left| \omega \right| \leq \frac{\pi}{T_s}$$

无码间串扰时域条件：

$$h(kT) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$$

2.4 成型滤波、匹配滤波的原理和作用（10 分）

成型滤波与匹配滤波总是成对出现。常见的组合有：

- (1) 升余弦滤波（发射端）+ 单位冲击响应（接收端）
- (2) 平方根升余弦滤波（发射端）+ 平方根升余弦滤波（接收端）

匹配滤波器的传输函数：

$$H_M(\omega) = kX^*(\omega)e^{-j\omega t_0}$$

成型滤波设计：

- (1) 将信号的频谱进行限制，以匹配信道带宽的范围，能减小码间串扰。
- (2) 为了降低对滤波器的过渡带陡峭难度的要求，需先对信号进行过采样处理，如果每个符号 4 个采样点处理（符号速率不变），即 $F_s = 4 \cdot R_b$ ， $F_c = 1/2 \cdot R_b$

三 实验结果图示及分析（30 分）

3.1 信道编解码的具体实现（20 分）

信道编码：

%%信道编码-海明码（系统码）

```
l=length(mst_bits);
switch mod(l,4) %化成 4 的整数倍
    case 0
        temp=mst_bits;
    case 1
        temp=[mst_bits,0,0,0];
    case 2
        temp=[mst_bits,0,0];
    case 3
        temp=[mst_bits,0];
end

G=[1 0 0 0 1 0 1;
   0 1 0 0 1 1 1;
```

```

0 0 1 0 1 1 0;

0 0 0 1 0 1 1]; %%生成矩阵

temp1=[0,0,0,0];
temp2=[0,0,0,0,0,0,0,0];

for i=1:4:(length(temp)-3)           %4 位一组
    temp1(mod(i,4))=temp(i);
    temp1(mod(i,4)+1)=temp(i+1);
    temp1(mod(i,4)+2)=temp(i+2);
    temp1(mod(i,4)+3)=temp(i+3);

    temp2=mod(temp1*G,2);
    if(i<2)
        hamming_code=temp2;
    else
        hamming_code=[hamming_code,temp2];
    end
end

信道解码:

%%信道解码

%%检验

temp3=[0,0,0,0,0,0,0,0];

switch mod(length(soft_bits_out),7)    %%使输入数组为 7 的倍数
    case 0
        soft_bits_out=soft_bits_out
    case 1

soft_bits_out=soft_bits_out(:,1:length(soft_bits_out)-1);
    case 2

```

```

soft_bits_out=soft_bits_out(:,1:length(soft_bits_out)-2);
    case 3

soft_bits_out=soft_bits_out(:,1:length(soft_bits_out)-3);
    case 4

soft_bits_out=soft_bits_out(:,1:length(soft_bits_out)-4);
    case 5

soft_bits_out=soft_bits_out(:,1:length(soft_bits_out)-5);
    case 6

soft_bits_out=soft_bits_out(:,1:length(soft_bits_out)-6);
end

temp4=soft_bits_out;
S=[0,0,0];           %%伴随式

H=[1,1,1,0,1,0,0;
    0,1,1,1,0,1,0;
    1,1,0,1,0,0,1];
Ht=H.';

for i=1:7:(length(temp4)-6)           %%每7个一组进行校验解码
    temp3(mod(i,7))=temp4(i);
    temp3(mod(i,7)+1)=temp4(i+1);
    temp3(mod(i,7)+2)=temp4(i+2);
    temp3(mod(i,7)+3)=temp4(i+3);
    temp3(mod(i,7)+4)=temp4(i+4);
    temp3(mod(i,7)+5)=temp4(i+5);
    temp3(mod(i,7)+6)=temp4(i+6);

    S=mod(temp3*Ht,2);

    s=char(S+'0');           %%判断错误图样
    e=zeros(1,7);
    switch s
        case '000'

```

```

        e=[0 0 0 0 0 0 0];
    case '001'
        e=[0 0 0 0 0 0 1];
    case '010'
        e=[0 0 0 0 0 1 0];
    case '011'
        e=[0 0 0 1 0 0 0];
    case '100'
        e=[0 0 0 0 1 0 0];
    case '101'
        e=[1 0 0 0 0 0 0];
    case '110'
        e=[0 0 1 0 0 0 0];
    case '111'
        e=[0 1 0 0 0 0 0];
    end

    if(i<2)

        signal=mod(temp3+e,2); %%纠错

    else
        signal=[signal,mod(temp3+e,2)];
    end

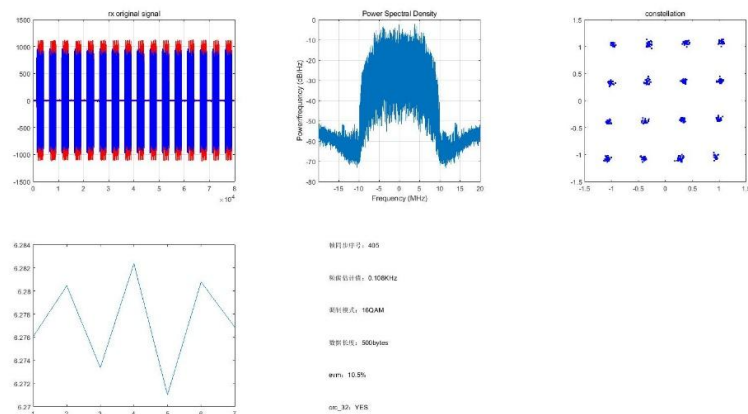
end        %%for 循环结束

a=length(signal)/7;        %%将监督位去除，恢复原数组

signal=reshape(signal,7,a);
final_signal=signal(1:4,:);
final_signal=reshape(final_signal,1,4*a);

```

经过以上两步编码解码，得到如下结果：



由图可见，在自发自收时，可以完成信道编解码任务，同时再收端能够得到正确的信源。但由于自发自收的误码率本身较小，所以没办法观察到加入海明码纠错后对误码率的提升。

3.2 从误码率分析系统性能（10 分）

理论上来说，74 海明码的加入可以降低误码率，提升系统性能。

由于收发两端距离较短，信道条件相对较好，因此加入海明码前后误码率相差不大。

目前想到的可能的验证方法是在实验室中一人发，另一人收，从而观察加入海明码前后的误码率变化。

四 思考题（10 分）

4.1 线性分组码中码距、纠错位数、检错位数的关系

码距，又称汉明距离，是码组中任意两个码字之间对应位上码元取值不同的个数。它等于两个码字对应位模 2 相加后“1”的个数。

最小码距指的是码组中各码字之间最小的码距，用 d_0 表示， d_0 与纠错位数 e 和查错位数 t 的不等关系如下：

$$d_0 \geq e + 1$$

$$d_0 \geq 2t + 1$$

$$d_0 \geq e + t + 1, (e > t)$$