

# BSC信道编码实验报告

无04 2019012137 张鸿琳

## 实验目的

- 通过Simulink模块的搭建，理解信道编码在通信系统中的作用和定位。
- 通过实验，分析和对比重复码、Hamming码的实际误比特率性能和理论性能。

## 实验内容

- 编写实现重复码模块。
- 搭建使用重复码、Hamming码的BSC通信系统模型并进行仿真实验。
- 绘制不同码率的误码率曲线并和理论值进行对比分析。

## 实验原理

### 重复码

在信道编码理论中，重复码是最基本的纠错码之一。通过有噪声的信道传输消息，一些比特可能发生错误，重复码的基本思路是重复消息 $n$ 次，希望信道只破坏这些重复中的一小部分。例如(3,1)重复码，当编码消息为[1 0 1]时，编码结果为[1 1 1 0 0 0 1 1 1]。这样，接收方可以通过查看接收到的数据流中出现频率最高的消息来恢复原始消息。

由于过低的编码效率，在大多数情况下，其他纠错码是首选。重复码的优点是易于实现。

### Hamming码

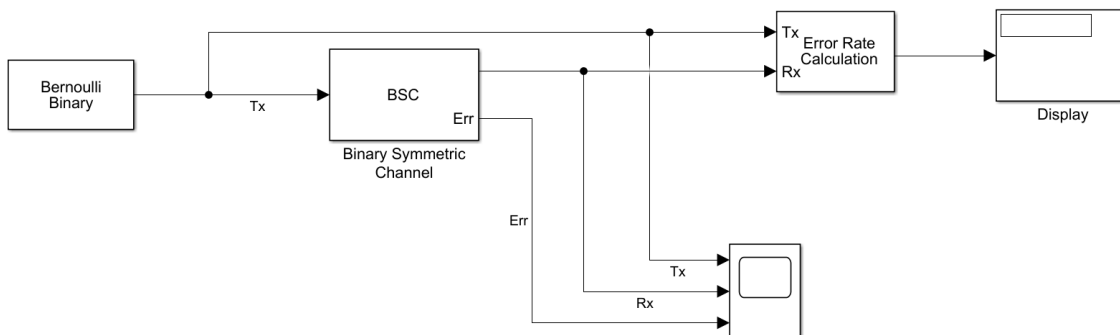
Hamming码是一种线性纠错码，其最小Hamming距离为3，可以纠正单个错误。给定正整数 $m(m \geq 2)$ ，其码长为 $n = 2^m - 1$ ，信息码位 $k = 2^m - 1 - m$ ，校验码位 $r = m$ 。校验矩阵 $H^T$ 的大小为 $n \times m$ ，它的行 $n$ 分别由除了全0之外的所有 $m$ 位码组成且不重复。其译码方法可以通过计算校正子，确定错误图样并纠正。

汉明码是完备码，它在校验位长度相同、最小距离为3的码中能达到最高的码率。

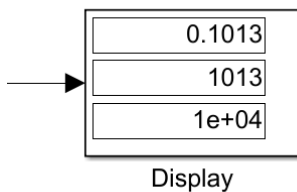
## 实验过程与记录

### 5.1 BSC 差错信道

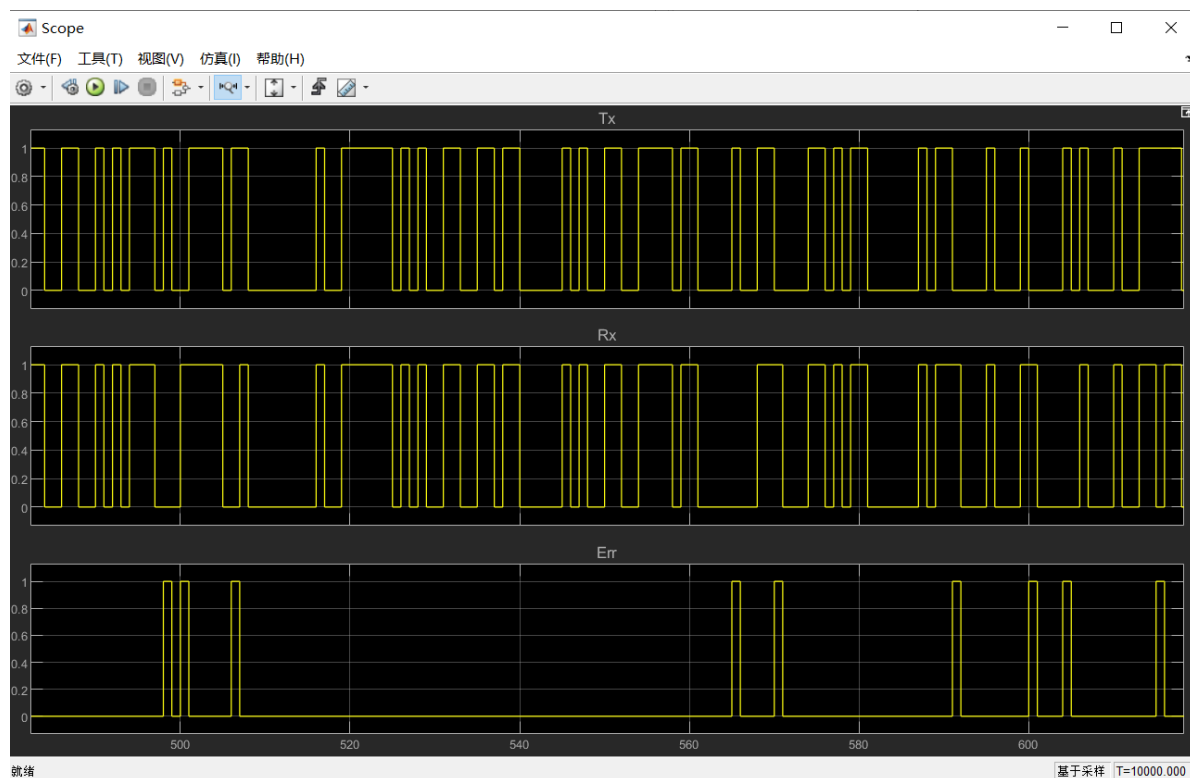
搭建如下模型：



设置 BSC 信道模块的错误率为0.1，随机种子为学号后三位，仿真时间设置为 10000，得到误比特率如下：



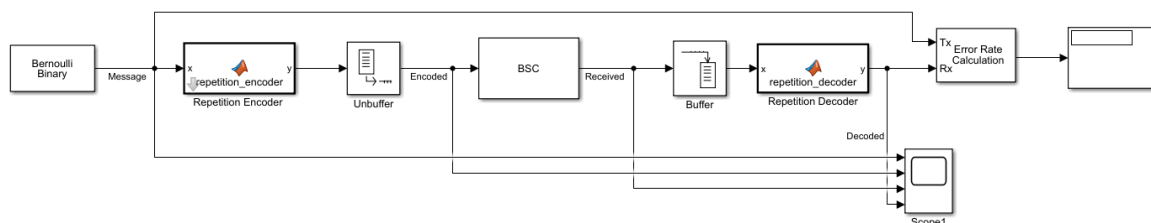
误比特率为0.1013，和设置的错误率0.1相吻合。示波器图像如下：



可以看到在500-600这段时间内，共发生了大约7、8次比特错误，和设置的错误率是基本吻合的。

## 5.2 重复码

搭建如下模型：



其中 repetition\_encoder 模块中自定义函数如下：

```
1 function y = repetition_encoder(x, n)
2 y = repelem([x],n)';
```

repetition\_decoder 模块中自定义函数如下：

```

1 function y = repetition_decoder(x)
2 n0 = length(find(x == 0));
3 n1 = length(find(x == 1));
4 if(n0 >= n1)
5     y = 0;
6 else
7     y = 1;
8 end

```

改变信道错误率，记录两种重复码的误比特率，如下：

- (3,1) 重复编码：

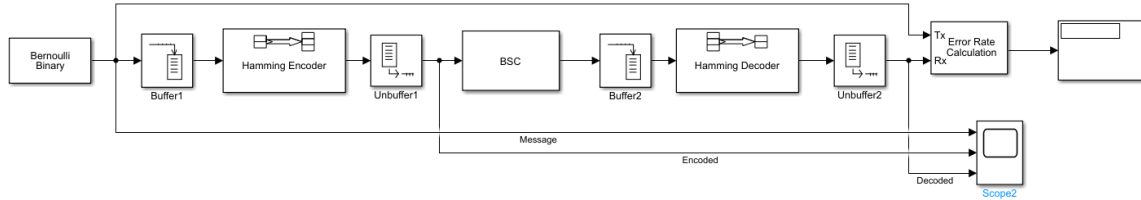
信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.216	0.02739	0.00264	0.0003	0.00005

- (7,1) 重复编码：

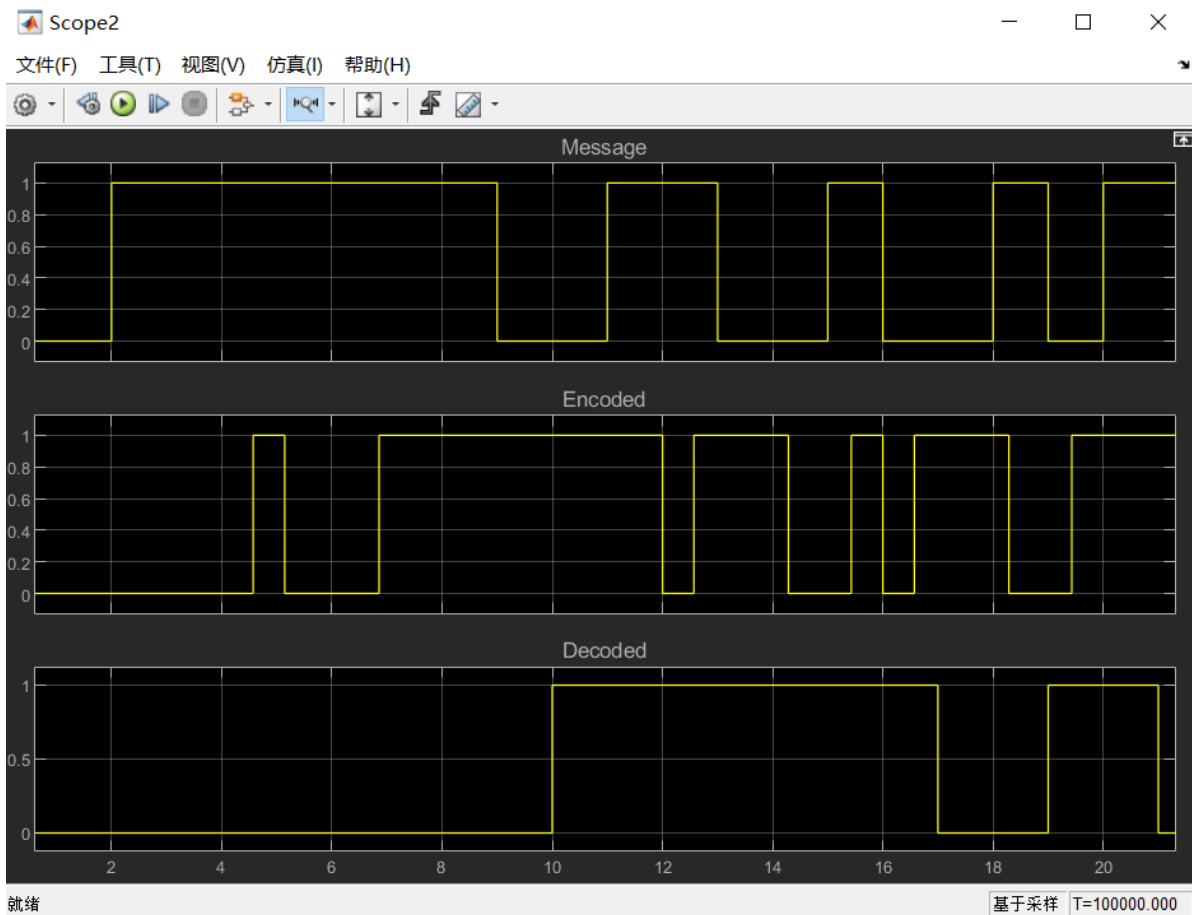
信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.1279	0.00261	0.00004	0	0

## 5.3 Hamming 码

搭建如下模型：

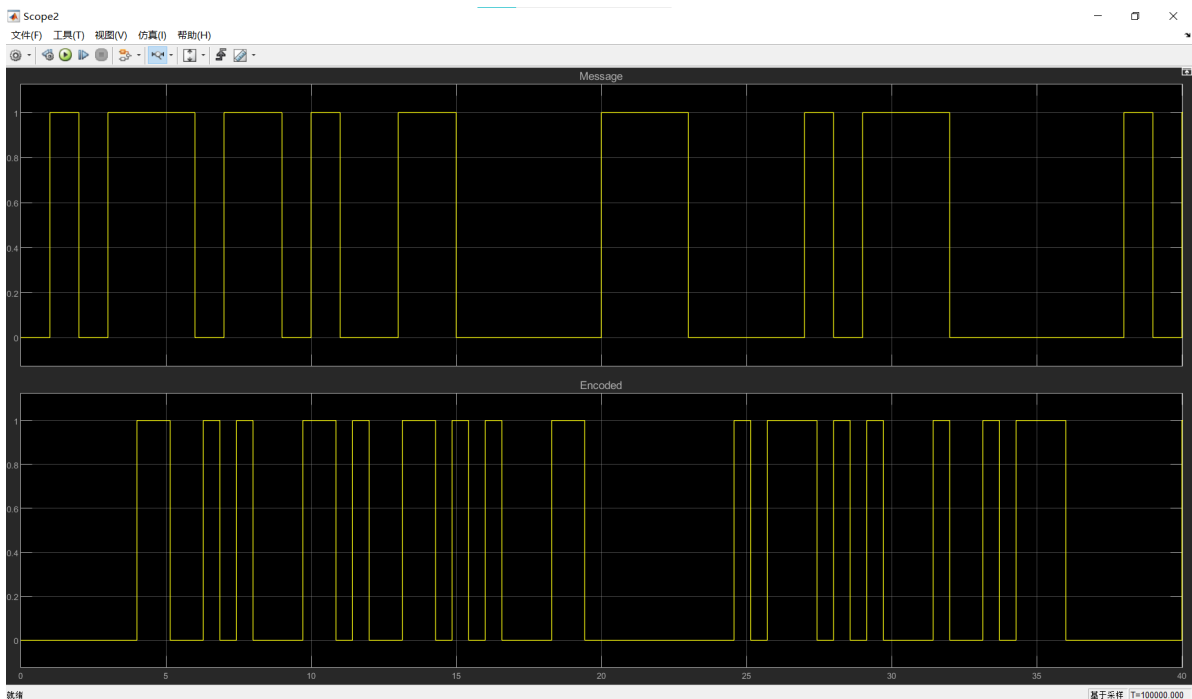


通过示波器测得信宿处信号的延时为8，如下图：



据此设置Error Rate Calculation模块的Receive Delay进行补偿后便可以测量误比特率。

利用示波器观察 (7,4) Hamming码编码前后的信号，如下图：



由于Buffer引入了4个时间单位的延时，去除延时后，可以看到输入序列为  
 $[0, 1, 0, 1; 1, 1, 0, 1; 1, 0, 1, 0; 0, 1, 1, 0]$ ，其对应的编码输出为  
 $[1, 1, 0, 0, 1, 0, 1; 0, 0, 0, 1, 1, 0, 1; 0, 0, 1, 1, 0, 1, 0; 1, 0, 0, 0, 1, 1, 0]$ ，比较容易发现，每次编码出的7位比特码的后四位都和原码相同，所以比较容易推测生成矩阵形式为  $G = [Q, I]$ ，通过上面的数

据，求解多元线性方程，比较容易得出，生成矩阵为  $G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$ ，进而可得校

验矩阵  $H^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ ，由于生成码字包含消息（原码）本身，所以实验 (7,4) Hamming码是系

统码。

改变信道错误率，记录两种Hamming码的误比特率如下：

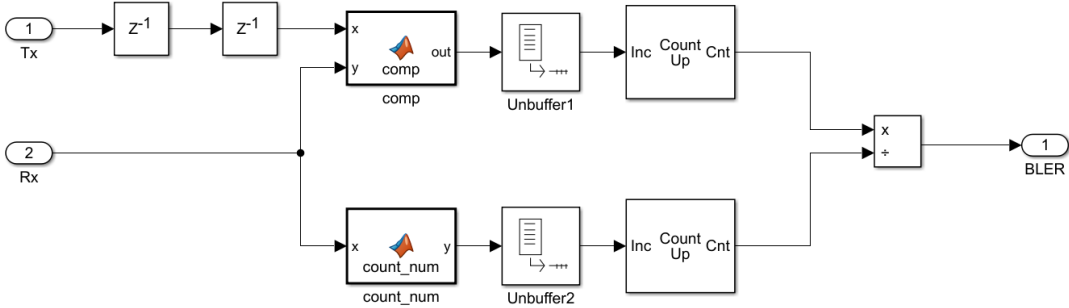
- (7,4) Hamming码：

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.3222	0.06804	0.007521	0.0009601	0.00006

- (31, 26) Hamming码

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.3131	0.1187	0.02676	0.004432	0.0003902

【选做】设计实现一个误块率计算模块，如下图：



其中 comp 模块内置函数如下：

```

1 function out = comp(x,y)
2 len = length(x);
3 out = zeros(len,1);
4 for p = 1:len
5     if (x(p) ~= y(p))
6         out(1,1) = 1;
7     return
8 end
9 end
10 return

```

comp 模块的作用就是比较两个输入数组（也就是两个分组）是否一致，也就是接收方解码出的块是否错误，若出现错误，则（和后续 Unbuffer 模块）生成一个脉冲，使后续 count 模块计数加一。

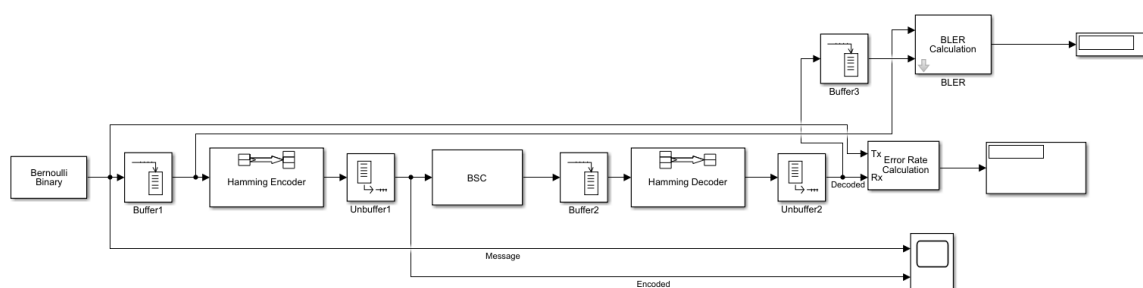
count\_num 模块内置函数如下：

```
1 function y = count_num(x)
2 len = length(x);
3 y = zeros(len,1);
4 y(1,1) = 1;
5 return
```

count\_num 模块用于计数共有几个分组输入，每收到一个分组，则（和后续 Unbuffer 模块）生成一个脉冲，使后续 count 模块计数加一。

利用上面两个模块，对输入 Tx 延时两个采样点（因为外部 Hamming 编码模型中存在两个 Buffer 模块），将输入的 Tx 数组和 Rx 数组输入到设计好的 comp 和 count\_num 模块中，最后将二者输出作比值，得到的就是误块率。最后将该模块进行封装即可。

对此前搭建的 Hamming 码模型稍作修改（加入测量误块率的模块），如下：



改变信道错误率，记录两种Hamming码的误块率如下：

- (7,4) Hamming码：

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.3222	0.06804	0.007521	0.0009601	0.00006
误块率	0.6717	0.1516	0.01756	0.00232	0.00012

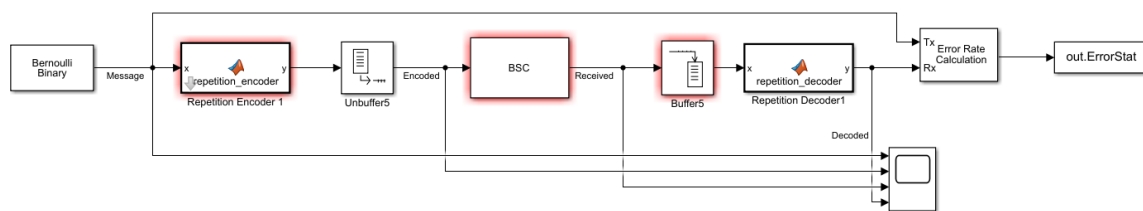
- (31, 26) Hamming码

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.3131	0.1187	0.02676	0.004432	0.0003902
误块率	0.9997	0.8346	0.2467	0.04342	0.00416

## 5.4 绘制误比特率曲线

## a) 重复码

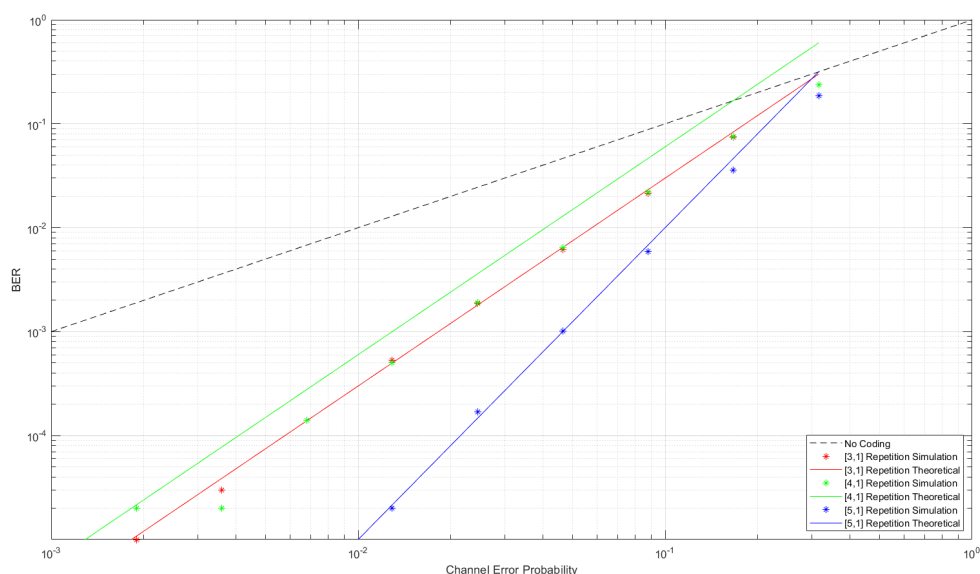
对此前的重复码实验模型稍作修改，如下图：



将所给 exp5.m 代码“TODO”部分补全为：

```
1 ber_rep_th(i,j) = nchoosek(n,round(n/2))*eprob^ (round(n/2));
```

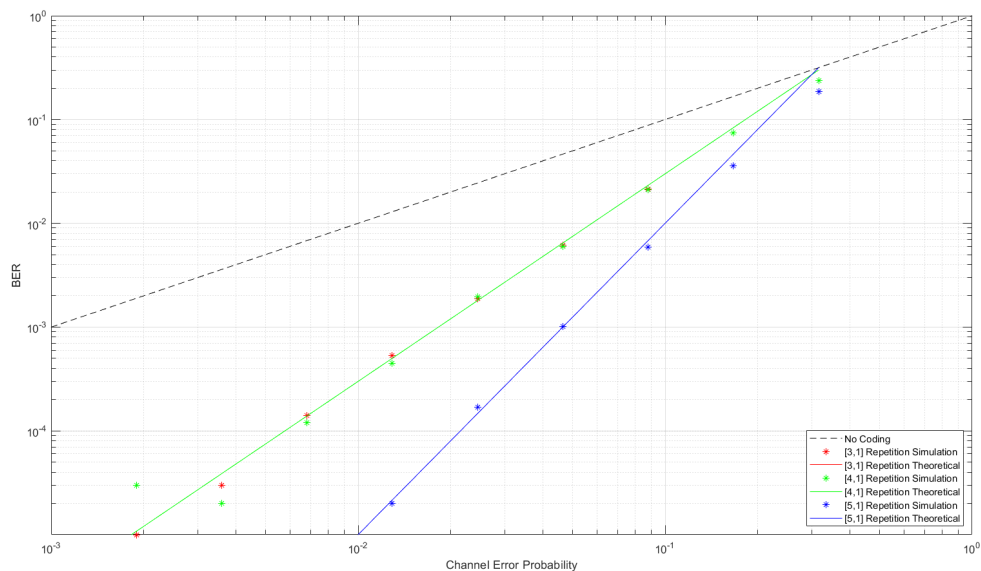
运行代码，得到图像如下：



对于  $(n,1)$  重复编码，其理论误比特率为  $P_b \approx \binom{n}{\lfloor \frac{n}{2} \rfloor} \epsilon^{\lfloor \frac{n}{2} \rfloor}$ （ $(3,1)$  重复码理论误比特率  $P_b \approx \binom{3}{2} \epsilon^2 = 3\epsilon^2$ ， $(4,1)$  重复码理论误比特率  $P_b \approx \binom{4}{2} \epsilon^2 = 6\epsilon^2$ ， $(5,1)$  重复码理论误比特率  $P_b \approx \binom{5}{3} \epsilon^3 = 10\epsilon^3$ ）。从上图可以看到实验所得误比特率和理论误比特率基本吻合，只是对于  $(4,1)$  重复编码，直接套用上述公式得到的理论误比特率和实验所得值有较为明显的偏差，因为我所设计的解码模块中，若收到分组中存在 2 个 0 时，将该组判定为 0，若原本发送的是 1，则该情况下一定判错，若原本发送的是 0，则该情况下一定判对，（发送 0 的概率为 0.5）此时实际上的理论误比特率约为  $P_b \approx \frac{1}{2} \binom{4}{2} \epsilon^2$ ，若把“TODO”部分更改为如下代码：

```
1 if (n == 4)
2     ber_rep_th(i,j) = nchoosek(n,round(n/2))*eprob^(round(n/2))/2;
3 else
4     ber_rep_th(i,j) = nchoosek(n,round(n/2))*eprob^(round(n/2));
5 end
```

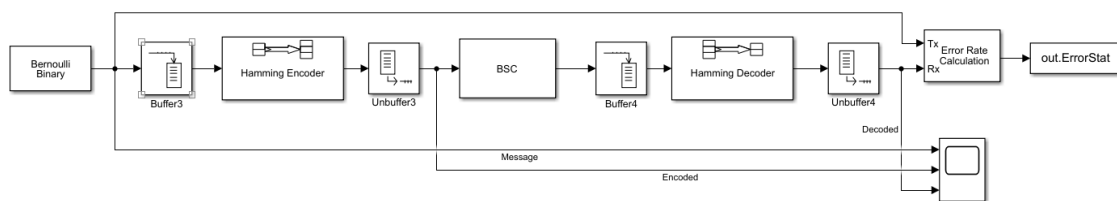
再运行代码，得到下图：



可见更改后，理论值基本和实验所得值完全吻合了（此时 (3,1) 重复编码和 (4,1) 重复编码的理论误比特率都约为  $P_b \approx 3\varepsilon^2$ ，故而二者理论值曲线重合）。

## b) Hamming 码

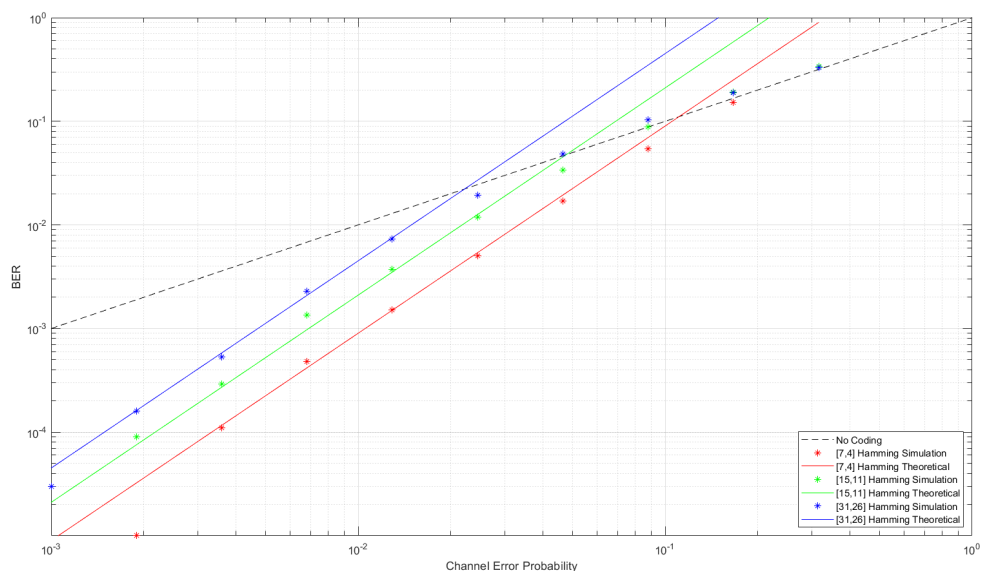
对此前的 Hamming 码实验模型稍作修改，如下图：



将所给 exp5.m 代码“TODO”部分补全为：

```
1 | ber_rep_th(i,j) = 3/n*nchoosek(n,2)*eprob^(2);
```

运行代码，得到图像如下：





对于  $(n,k)$  Hamming 编码, 理论误比特率为  $P_b \approx \frac{3}{n} \binom{n}{2} \epsilon^2$ 。从上图可以看到实验所得误比特率和理论误比特率基本吻合, 只是当 BSC 信道错误率逐渐增大, Hamming 编码的误比特率走势逐渐和图中虚线相吻合, 图中虚线就是  $y = \epsilon$ , 也就是说当 BSC 信道错误率大到一定程度时, Hamming 编码基本已经不具有纠错能力, 编码经过信道解码并纠错后的误比特率基本等于信道错误率了。

## 实验思考题

**1. 在Simulink中, 信道编码为什么会引起延时? 假设每次实际使用信道传输一个bit的时间为1个单位, 请计算 $(n,1)$ 重复码和 $(n,k)$  Hamming码的实际延时。延时对于设计信道编码有什么启示?**

在 Simulink 中信道编码引起延时的原因是存在 Buffer 模块, 假设 Buffer 模块的参数为  $n$ , 则 Buffer 模块需要等待  $n$  个 bit 的时间才会给出一次输出, 模拟实验中的单位时间是由 Bernoulli Binary 生成模块确定的, 故而实验测得在上面搭建的模型中  $(n,1)$  重复码的延时为 1 个单位时间,  $(n,k)$  Hamming 码的延时为  $2k$  个单位时间。

在实际使用信道时, 单位时间是由信道本身确定的,  $(n,1)$  重复码在解码端需要等待  $n$  个 bit 全部收到后才能判决出一个 bit 结果, 故而  $(n,1)$  重复码的实际延时为  $n$  个单位;  $(n,k)$  Hamming 码在编码端需要等待收到  $k$  个 bit 后才能进行一次编码, 为了保障信道正常工作, 编码端接收  $k$  个 bit 的时间应该等于其发出  $n$  个 bit 的时间 (也就是信道中  $n$  个 bit 的传输时间), 而后在解码端则需要等待收到  $n$  个 bit 后才能进行一次解码, 故而  $(n,k)$  Hamming 码的实际延时为  $2n$  个单位。

延时对设计信道编码的启示: 由上面分析可以看出延时基本和编码后的 bit 数成正相关, 所以不能无限制增长单个分组编码后的 bit 数规模, 否则编解码本身会引起较为明显的延时。

**2. 在高信道错误概率时, 为什么Hamming码的误比特率超过了信道错误概率?**

因为 Hamming 码只能用于纠正 1 bit 错误, 当信道的错误概率很高时, 就会导致一个分组 (块) 内出现多个 bit 错误的概率明显增大, 此时 Hamming 码会越纠越错, 即纠错后误比特数更多了, 因此就出现了 Hamming 码的误比特率超过了信道错误概率的现象。

**3. 若给定信道与可靠性 (误比特率) 要求下, 怎样确定最优编码方案? 例如已知信道错误率  $\epsilon = 0.01$ , 要求编码后误比特率  $P_e \leq 0.003$ , 请选择一套编码方案和参数。 (提示: 根据误比特率曲线)**

可以将多种不同方案的编解码后理论误比特率曲线放置于同一图像中, 找到给定信道错误率的条件下, 编解码后误比特率满足要求的方案, 再从这些方案中选出延时和编解码复杂度综合考虑最优的一个方案就是最优编码方案。

若已知信道错误率  $\epsilon = 0.01$ , 要求编码后误比特率  $P_e \leq 0.003$ , 则根据前面实验中绘制出的误比特率图像, 可知  $(15,11)$  Hamming 码、 $(7,4)$  Hamming 码、 $(3,1)$  重复码、 $(4,1)$  重复码、 $(5,1)$  重复码都是可行的选择, 如果考虑到延时和编解码复杂度, 则  $(3,1)$  重复码是最优的方案。