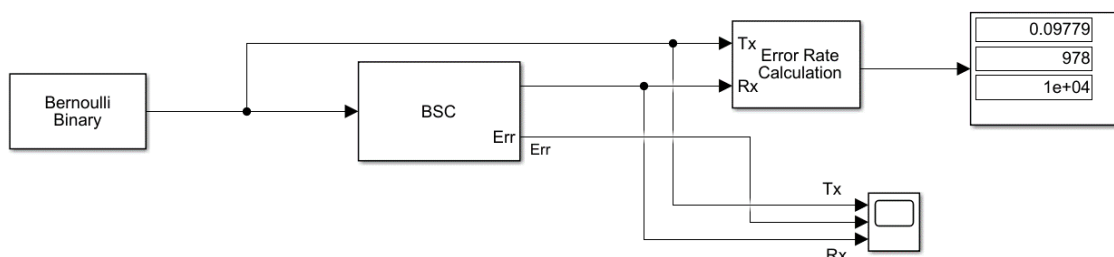


## 实验 5 BSC信道编码实验

### 一、BSC 差错信道

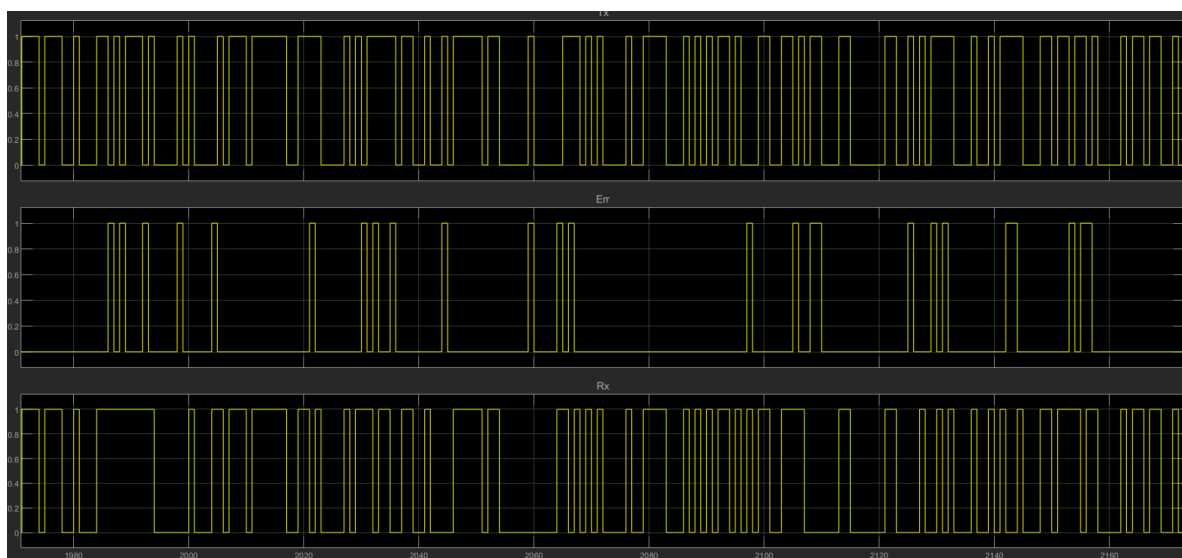
#### a) 搭建模型



参考模块设计图搭建的模型

#### b) 观察BSC 信道对二元数字信号的影响

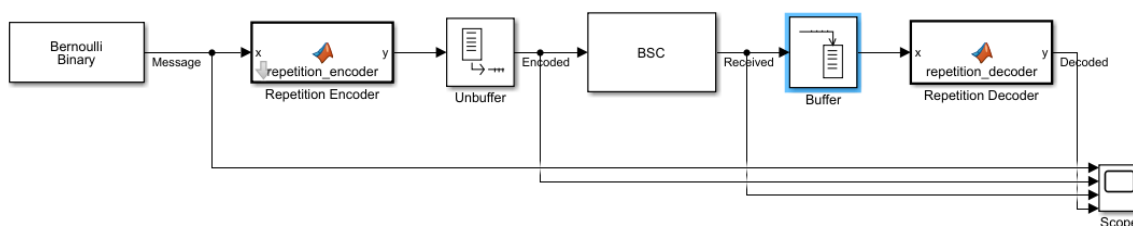
仿真的误比特率为0.09779，和设置的信道错误概率基本相等



较短时间内BSC 信道对二元信号的影响

### 二、重复码

#### a) 搭建模型并编写函数实现重复码



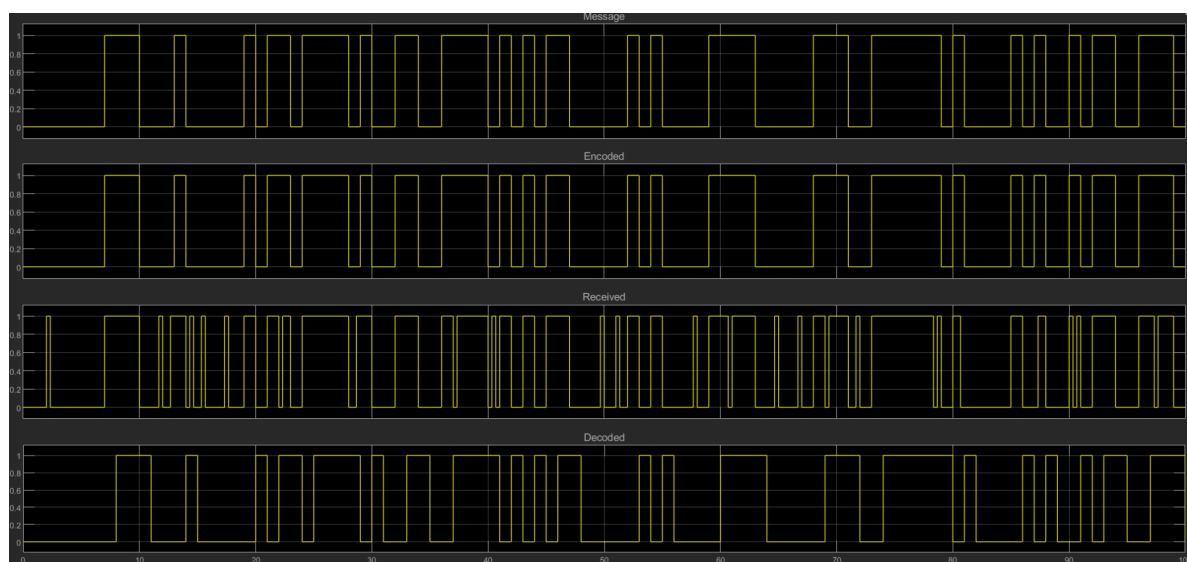
搭建的模型

```
hw5_2 ▶ Repetition Encoder
1  function y = repetition_encoder(x, n)
2      % TODO: y = ?
3      if x>0
4          y=ones(n,1);
5      else
6          y=zeros(n,1);
7      end
```

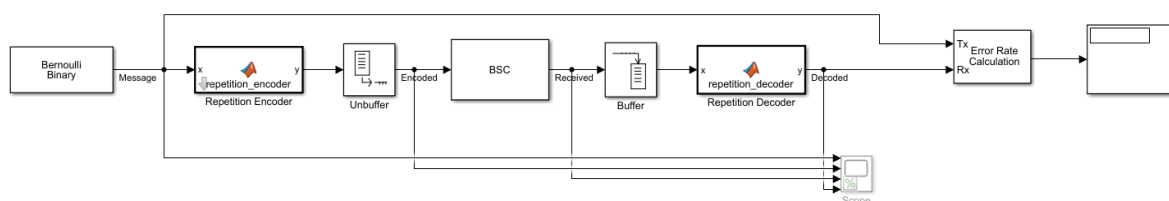
编码器

```
hw5_2 ▶ Repetition Decoder
1  function y = repetition_decoder(x)
2      % TODO: y = ?
3      if 2*sum(x)>length(x)
4          y=1;
5      else
6          y=0;
7      end
```

解码器



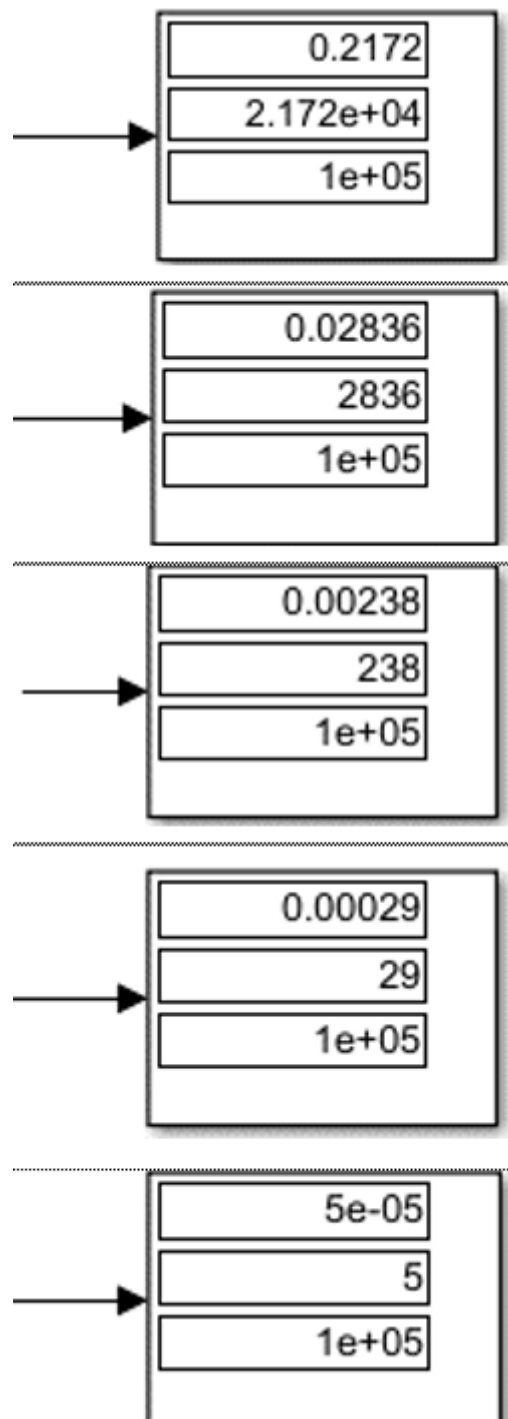
观察Message、Encoded、Received和Decoded四处的信号



重复码误比特率的模型搭建

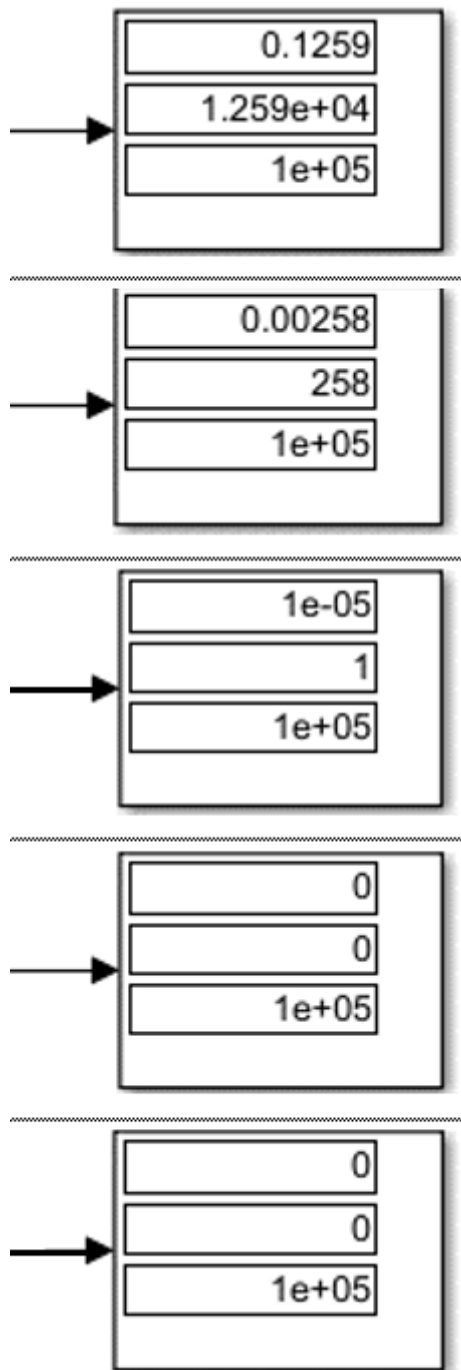
b) 改变信道错误率，记录重复码误比特率

(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	0.2172	0.02836	0.00238	0.00029	0.00005

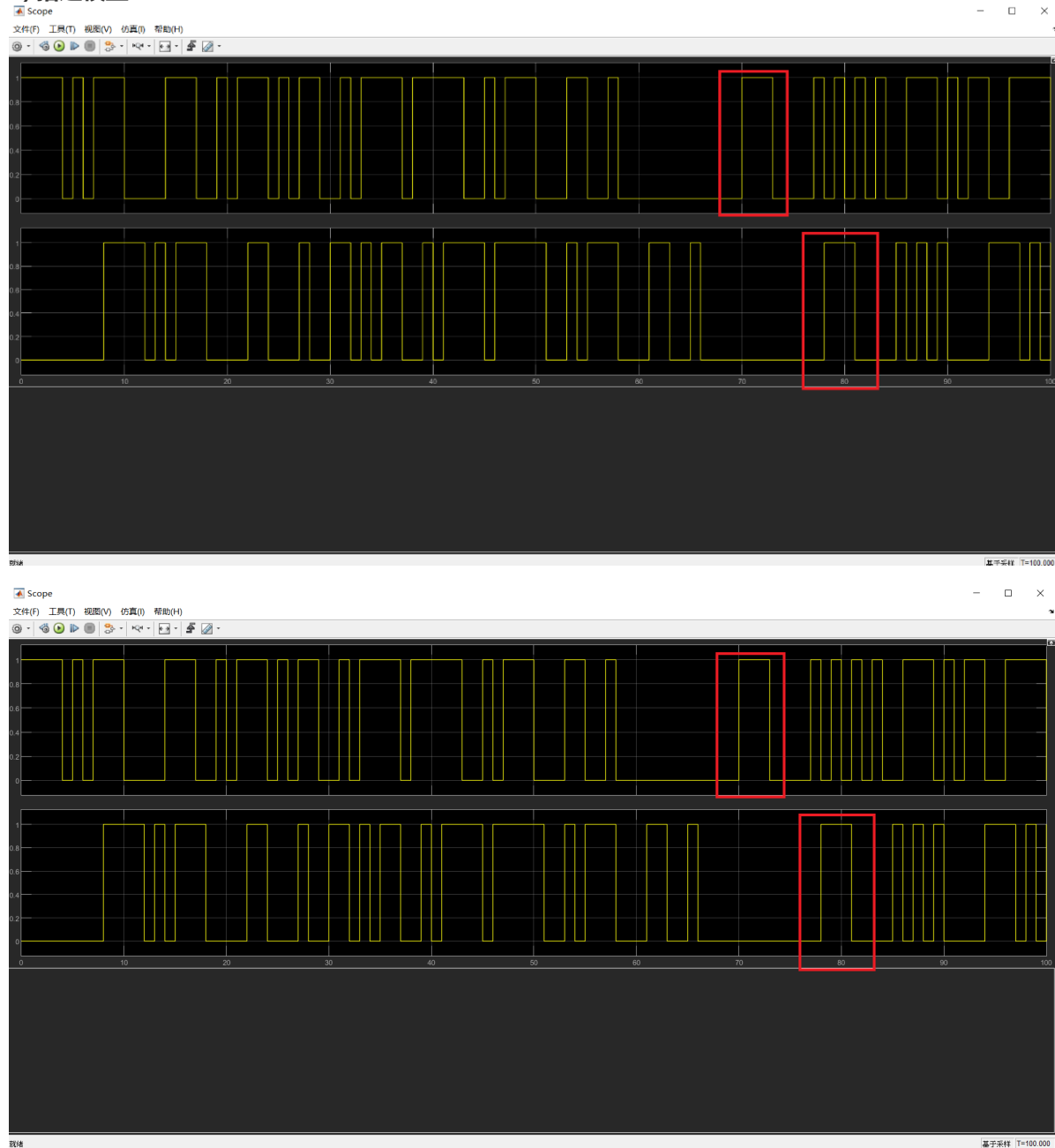
(7,1)重复编码



信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.1259	0.00258	0.00001	0	0

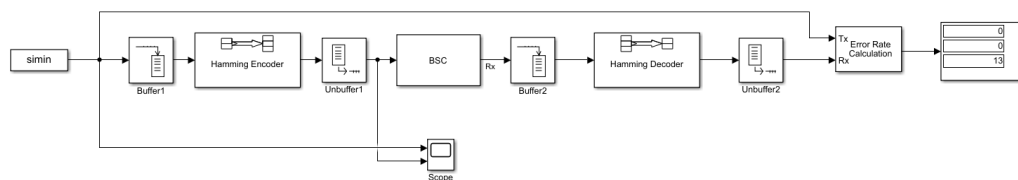
### 三、Hamming 码

## a) 搭建模型

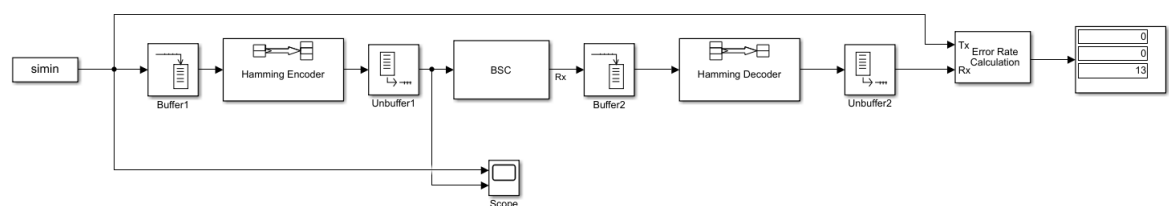


由图可得延时为8，设置正确的Receive Delay后得到误比特率0.03226

## b) 推断生成矩阵 $G$ 和校验矩阵 $H^T$



设置 $simin=[0:15;1,0,0,0,1,0,0,1,1,0,1,1,1,1]'$ ;



输入输出信号如图

输入	输出
1000	1101000
1001	0111001
1010	0011010
1111	1111111

由于输入四个信息线性无关，直接对输出矩阵左乘输入矩阵的逆即可得到生成矩阵 $G$

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

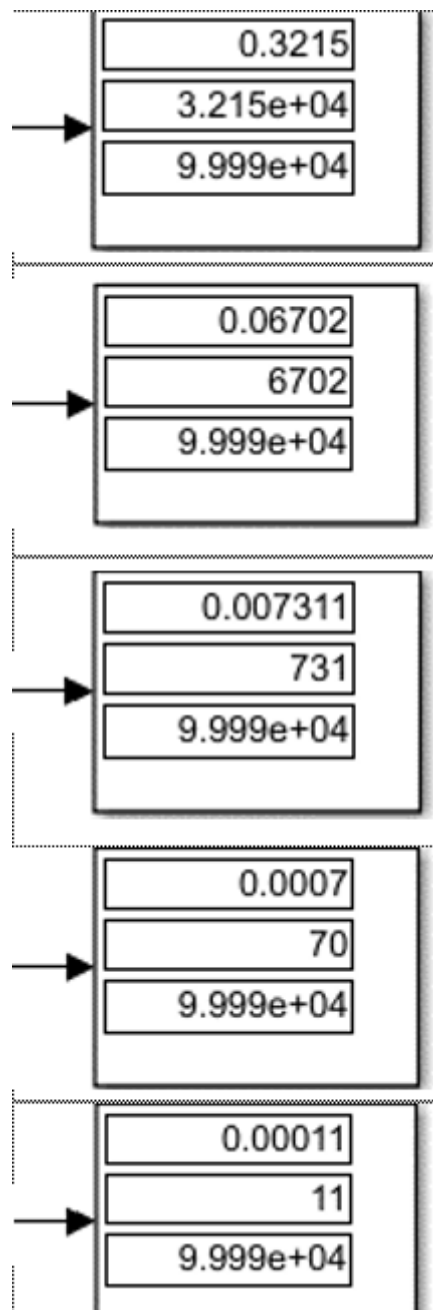
对 $G$ 中的 $Q$ 矩阵做转置拼上 $I_4$ 得到 $H^T$

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

生成矩阵 $G$ 中包含 $I_4$ ，是系统码，信息位是7位中的后4位

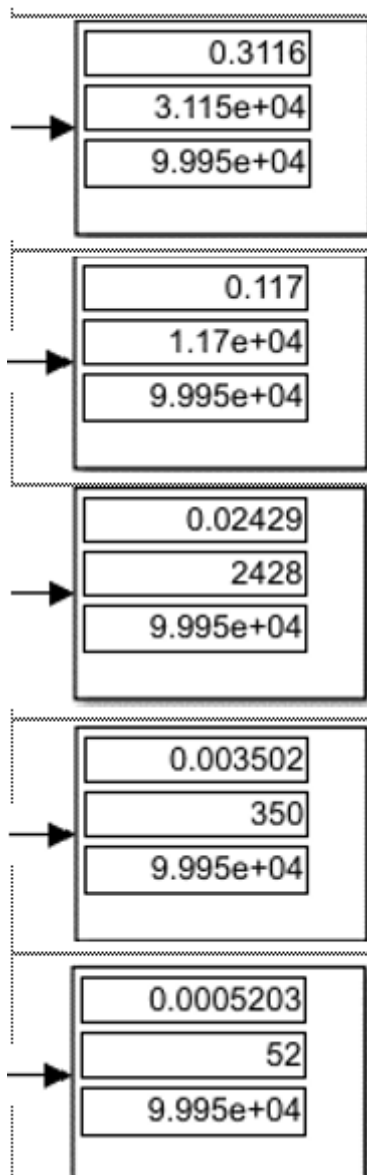
### c) 改变信道错误率，记录重复码误比特率

(7,4) Hamming 码



信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.3215	0.06702	0.007311	0.0007	0.00011

(31,26) Hamming 码

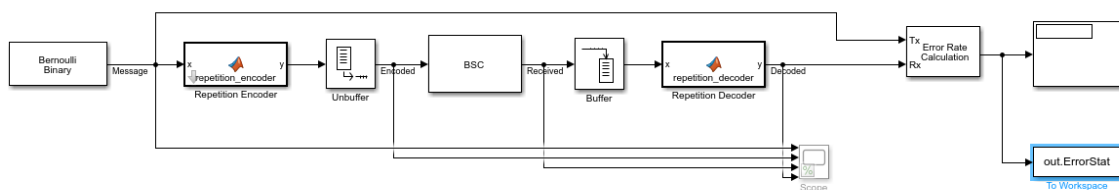


其中两个buffer引入延时 $2 * k = 52$

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率	0.3116	0.117	0.02429	0.003502	0.0005203

#### 四、绘制误比特率曲线

##### a) 重复码



模型构建

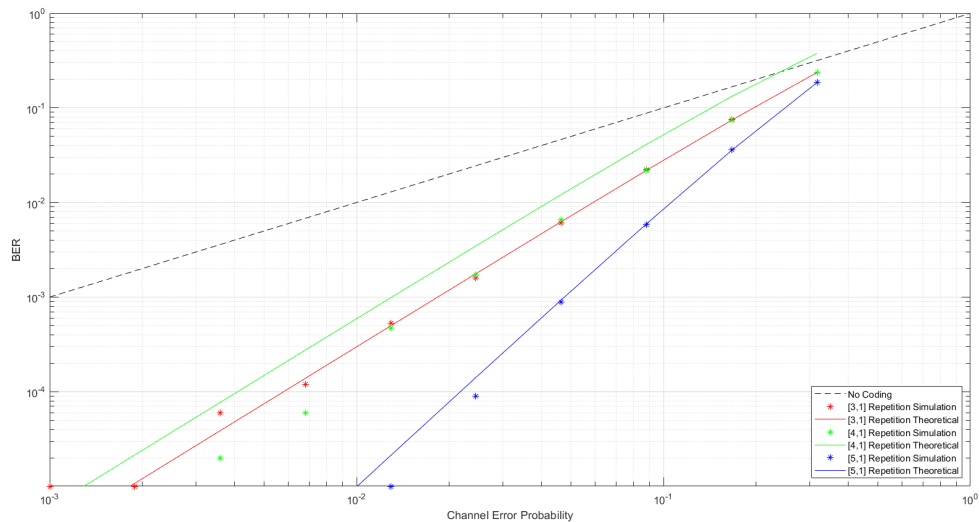


```

ber_rep_th(i,j) = 0;
    for k=floor((n+1)/2):n
        ber_rep_th(i,j) =ber_rep_th(i,j)+nchoosek(n,k)*(eprob^k)*(1-
eprob)^(n-k);% TODO: Theoretical BER
    end

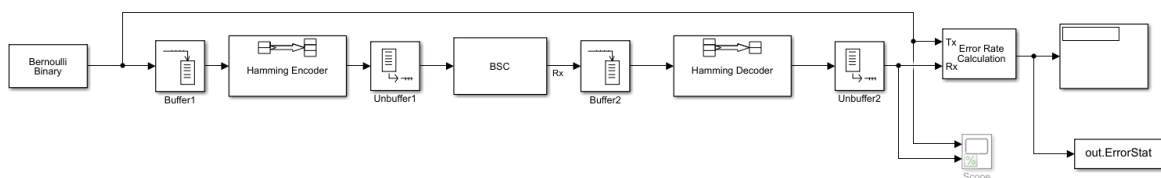
```

对n比特中错误大于等于一半比特的情况概率求和，即重复码的理论BER估计



实验数据和理论误比特率的曲线。除了n=4理论和仿真偏差较大，n=3和n=5两种情况理论和仿真都比较接近。在解码器代码编写时设置只有大于一半的比特是1时才判为1，而在n=4的理论计算时把码重大于等于2的都统一算作解码错误，这实际上多算了一部分发端发0，收端收到码重为2的概率。所以n=4的理论曲线更高一些。

## b) Hamming 码



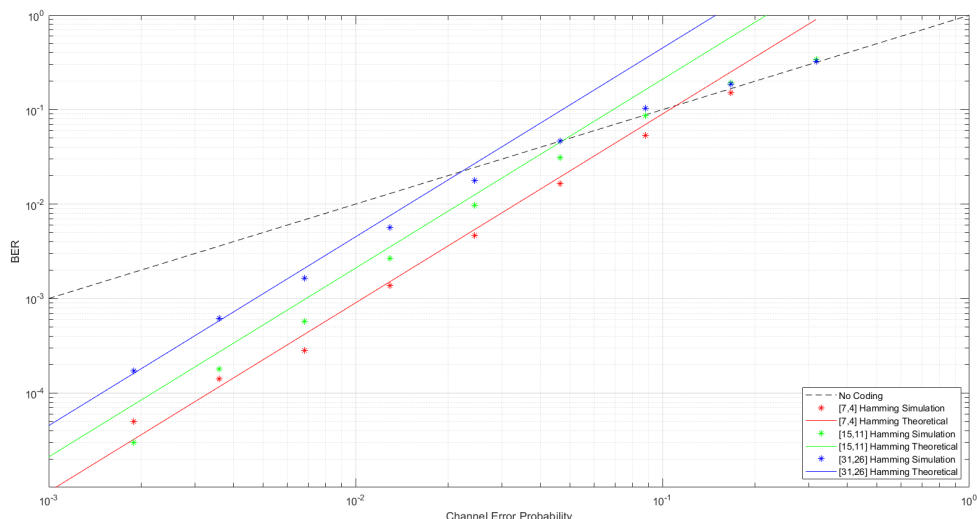
模型构建

利用提示的错误概率编写理论误比特率的代码

```

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

```



在低信道错误概率的情况下，理论和仿真的误比特率较为接近，随着信道错误概率的升高仿真得到的误比特率增长渐渐慢于理论的估计误比特率。因为估计公式中只取了 $\epsilon$ 的最低次项作为估计，在 $\epsilon$ 较小时舍弃的项非常小，随着 $\epsilon$ 增大舍弃的项逐渐增大，理论估计和仿真的误差也随着增大。

## 五、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？

答：因为信道编码需要用到buffer模块，当收到 $k$ 个串行比特时再并行传输给编码器，引起 $k$ 个单位的延时。在解码器前也要用到buffer，但因为编码后的信号频率变为原信号的 $\frac{n}{k}$ 倍，buffer收到 $n$ 个比特只需要 $k$ 个单位的延时。重复码较为特殊，编码只需要一个信息位，编码器前不需要接buffer，因此延时是 $k$ ；Hamming码延时是 $2k$ 。

$(n,1)$ 重复码实际延时：buffer需要累计收到 $n$ 个比特才能进行解码，这 $n$ 个比特经过BSC信道延时是 $n$ 个单位时间，因此实际延时是 $n$ 。

$(n,k)$ Hamming 码实际延时：由于有两个buffer，因此实际延时是重复码实际延时的二倍，即 $2n$ 个单位时间。

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？

答：在高信道错误概率时，传输一个码字的错误比特数近似为 $n\epsilon > 1$ ，由于Hamming 码的纠错能力只有1位，在错误比特位大于等于2时可能会把收到的码字纠正成其他码字，造成“越纠越错”错误比特数增加的情况，误比特率随之也会超过信道错误概率。

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）

答：由于重复码信道利用率较低，先考虑Hamming 码能否满足要求。观察图中 $\epsilon = 0.01$ 对应的误比特率， $(31,26)$ Hamming 码对应的误比特率略高于0.003，不符合要求。 $(15,11)$ Hamming 码的误比特率约为0.002，满足要求，并且相比同样满足误比特率要求的 $(7,4)$ Hamming 码效率更高，因此选择 $(15,11)$ Hamming 码。