

实验 5 BSC 信道编码实验

1. 实验目的

1. 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。
2. 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。

2. 实验内容

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

3. 实验原理

3.1 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。

3.2 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 n 行分别由除了全 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。

4. 实验环境

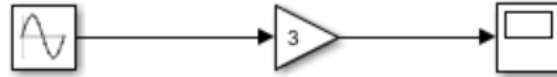
Simulink 是 The MathWorks 公司开发的用于动态系统和嵌入式系统的多领域模拟和基于模型的设计工具，常集成于 MathWorks 公司的另一产品 MATLAB 中与之配合使用。

Simulink 提供一个交互式的图形化环境及可定制模块库 (Library)，可对各种时变系统，例如通讯、控制、信号处理、影像处理和图像处理系统等进行设计、模拟、执行和测试，也可以进行基于模型的设计。

模块是 Simulink 编辑器的基本建模结构。我们可以从内置的 Simulink 库中添加模块以执行特定操作，也可以创建自定义模块。模块从输入端口接受信号，执行运算，并在输出端口输出信号。模块之间的连接接口称为**端口**。



信号线在仿真中将数据从一个模块传输到另一个模块。**信号**是随时间变化的量，在所有时间点（连续）或指定的时间点（离散）都有对应的值。信号线连接模块端口，信号从模块的输出端口流向另一个模块的输入端口。常见的信号是数值或矢量。



操作方法请参阅 Simulink 预习实验指导书。

5. 实验流程

5.1 BSC 差错信道

a) 搭建模型

需要使用的模块：

Communications Toolbox-Comm Sources-Random Data Sources-Bernoulli Binary Generator

Communications Toolbox-Channels-Binary Symmetric Channel

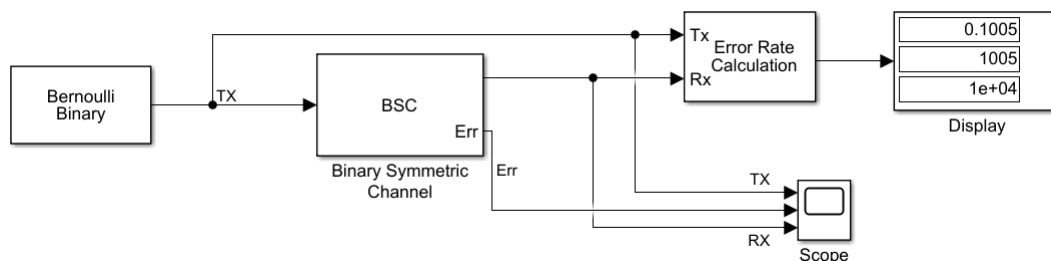
Communications Toolbox-Comm Sinks-Error Rate Calculation

（Error Rate Calculation 模块设置 Output Data 为 Port 后连接 Display）

Simulink-Sinks-Scope（示波器）

Simulink-Sinks-Display（数值显示器）

请参考模块设计图搭建模型：



这里信源是 Bernoulli 二元随机信号，信道是 BSC 信道，在信宿处计算误比特率。

b) 观察 BSC 信道对二元数字信号的影响

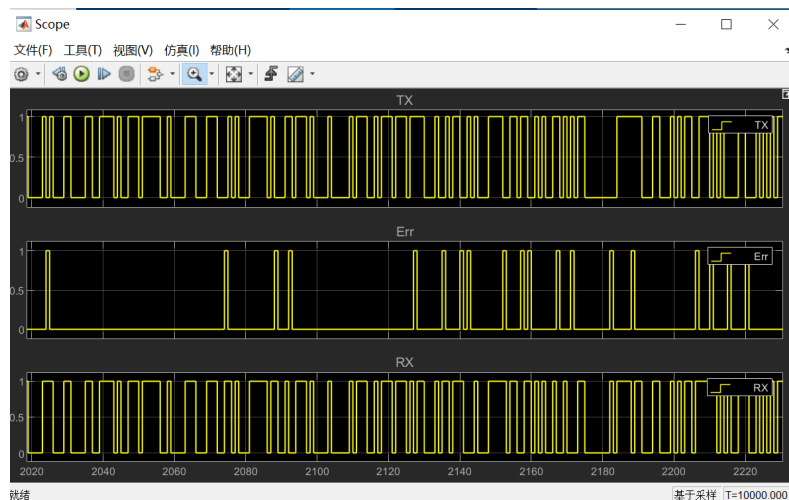
● 参数设置：

Binary Symmetric Channel 模块设置 Error Probability 为 0.1，initial seed(随机种子)设置你的学号后三位。

在上方仿真栏设置停止时间为 10000。

- 观察记录仿真的误比特率（BER）；
- 将发送信号（TX）、错误（Err）和接收信号（RX）一同送入 Scope（示波器），观察截图一段较短时间（100 左右）内 BSC 信道对二元数字信号的影响。

结果样例：（Scope 显示）



5.2 重复码

需要额外使用的模块：

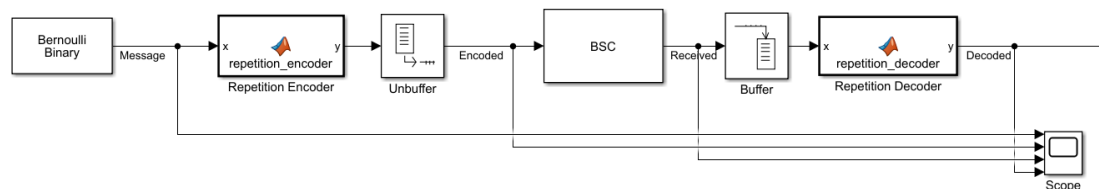
Simulink-User Defined Functions-MATLAB Function

DSP System Toolbox-Signal Management-Buffers-Buffer

DSP System Toolbox-Signal Management-Buffers-Unbuffer

a) 搭建模型并编写自定义 Matlab 函数实现(n,1)重复码

基于 5.1 模型，在信源和信道之间、信道和信宿之间分别插入两个自定义 Matlab 函数模块（编码、解码模块）以及 Buffer、Unbuffer，**如图所示搭建模型**。对模块标签进行适当标注。在 BSC 模块的属性中反选“Output Error Vector”使得模块只有一个输出端口。



Unbuffer 的作用是将并行的数据（向量）转换为高速串行数据以在 BSC 信道中发送;相反，Buffer 的作用是将高速串行的数据转换为低速并行数据（向量），然后交给解码器。两个自定义函数分别定义如下：（可直接复制）

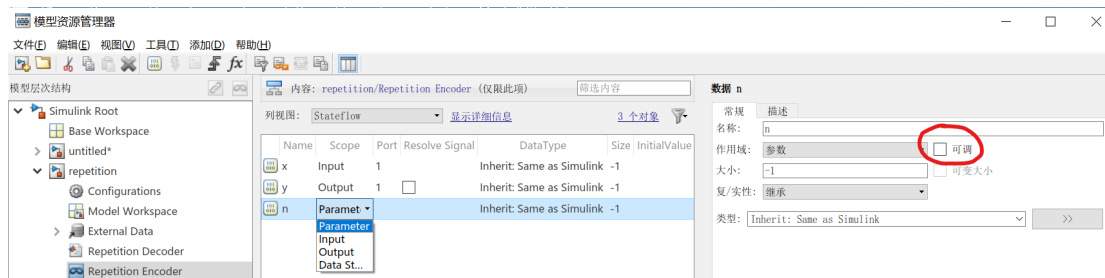
```
function y = repetition_encoder(x, n)
    % TODO: y = ?
```

```
function y = repetition_decoder(x)
    % TODO: y = ?
```

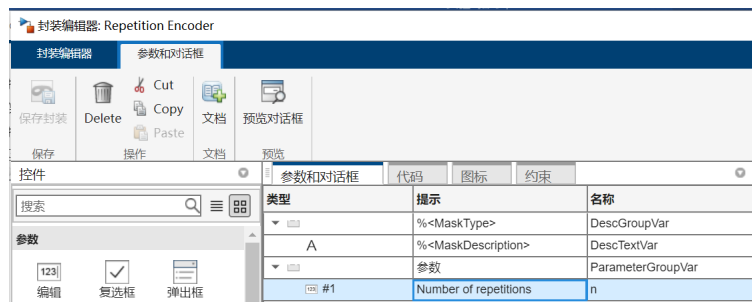
（注：n=2k 时 decoder 在 0,1 数量相等时可以判决为任意值）

请根据重复码的编解码规则进行实现编码器、解码器，编码结果应输出为列向量(n*1)。

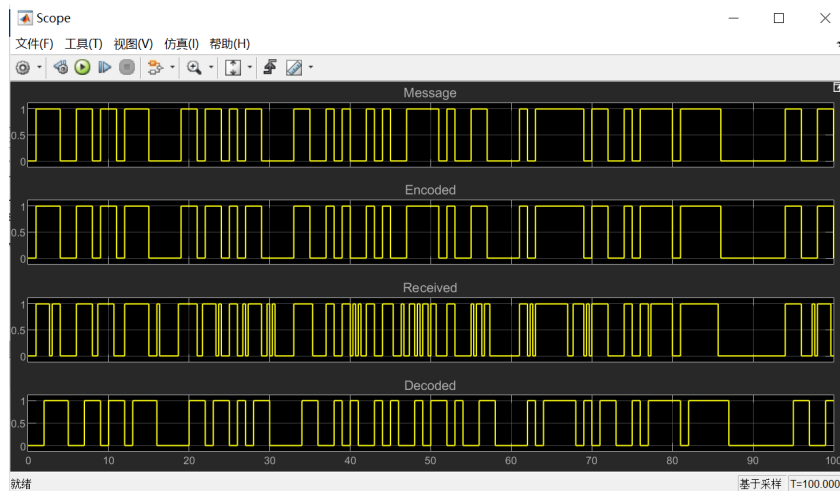
编写函数后，在模型资源管理器中设置函数 **repetition_encoder** 的输入 n 为参数，然后反选“可调”，如下图。这样函数只有一个输入端口 x，且确保编码器输出的信号维度在每次仿真中固定。



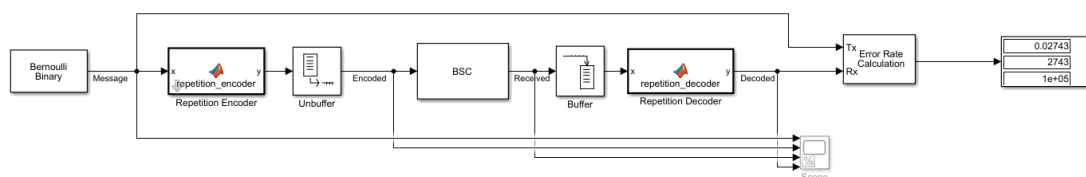
进一步，我们将编码函数封装为模块。右键点击该函数，创建封装，添加一个“编辑”控件，修改其（外部）“提示”和（内部）“名称”，为封装内部的编码函数的参数 n 赋值，如下图：



封装后，设置编码模块的参数为变量 n ，设置 Buffer 模块的 size 为变量 n 。
注意，（与刚刚设为参数并封装到内部的 n 不同）此处的变量 n 对应 Matlab 工作区。
 在 Matlab 命令行中给 n 赋值为 3，设置停止时间为 100，设置 BSC 信道的错误概率为 0.1，运行仿真后用示波器观察图中四个位置的信号：



注意虽然编码后的信号和编码前的信号完全相同，编码后的信号频率是编码前的 3 倍。可以观察到，Buffer 处引入了 1 个时间单位的延时，而编码/解码过程未引入任何延时。因此，在计算误比特率时，需要将原始信息做 1 个时间单位延时后再与解码信息做对比，因此设置 Error Rate Calculation 模块的 Receive Delay 为 1。



请按上图完成计算重复码误比特率的模型搭建。

b) 改变信道错误率，记录两种重复码的误比特率

设置仿真停止时间为 100000。（可注释/删除示波器以加快仿真速度）。

(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					

(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					

5.3 Hamming 码

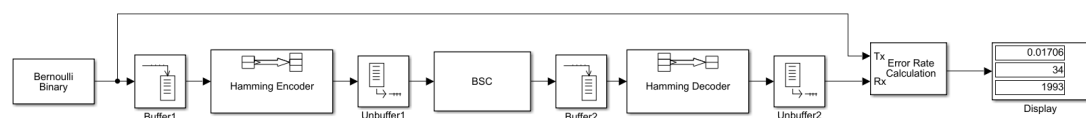
额外需要使用的模块：

Communications Toolbox-Error Detection & Correction/Hamming Encoder

Communications Toolbox-Error Detection & Correction/Hamming Decoder

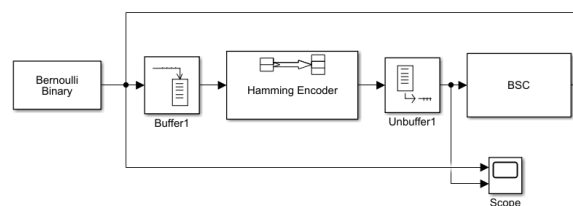
a) 搭建模型

基于 5.1 的模型，在信源和信道之间依次插入 Buffer, Hamming Encoder, Unbuffer 模块；类似地，在信道和信宿之间依次插入 Buffer, Hamming Decoder, Unbuffer，如图搭建模型。采用默认的(7,4)码长设置，注意设置 Buffer 的 size 应与下一个模块的输入尺寸一致。

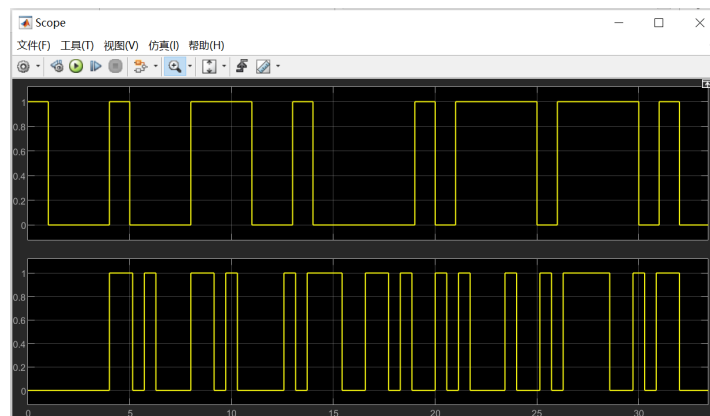


通过示波器测量得到信宿处信号的延时，并设置 Error Rate Calculation 模块的 Receive Delay 进行补偿后测量误比特率。

b) 通过示波器观察(7,4) Hamming 码编码前后的信号，推断生成矩阵 G 和校验矩阵 H^T



在上图的两个位置用示波器观察编码前后的信号（修改示波器的布局为两路）。



两个信号的时钟不同，编码前信号的周期为 1，编码后为4/7。可以观察到，编码前后的信号延时为 4 个单位，这是由于 Buffer 处引入了 $k = 4$ 个时间单位的延时。在上图例中，编码结果为 11010001…。注意，如果编码结果以'0'开头，则编码后的首个上升沿会推迟。

请根据你的仿真记录若干组编码前 4 位消息与编码后 7 位码字的对应，推断生成矩阵 G ，进一步得到校验矩阵 H^T ，并判断是否为系统码。

c) 改变信道错误率，记录两种 Hamming 码的误比特率

设置仿真停止时间为 100000。(可注释/删除示波器以加快仿真速度)。

(7,4) Hamming 码:

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率					

(31,26) Hamming 码:

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率					

d) (选做) 设计实现一个模块计算误块率 (Block Error Rate, BLER)

对于分组码，误块率定义为解码错误的块（每个块即一个分组，有 k 个比特）占传输的总块数的比例。对于 Hamming 码，每个块错误有可能导致多个错误比特。

请首先设计计算模块率的模块，组成一个子系统，然后封装为模块（见预习指导书），并命名为"BLER Calculation"。

提示：该模块无参数。两个输入信号分别为 $k \times 1$ 的数组，输出信号为一个数值。由输入信号之间有延时，需要通过延时模块补偿。这里补偿的延时设为 1，因为这里延时模块基于采样，输入信号的周期为 k ，所以延时设为 1 实际对应的仿真时间是 k 。

可能需要使用的模块（一种实现方法）:

Simulink-Commonly Used Blocks-Delay（注意不是 DSP System Toolbox 中的 Delay）

Simulink-Logic and Bit Operations-Relational Operator

Simulink-Logic and Bit Operations-Logical Operator

DSP System Toolbox-Signal Management-Switches and Counters-Counter

Simulink-Commonly Used Blocks-Constant

Simulink-Math Operations-Divide

设置仿真停止时间为 100000。(可注释/删除示波器以加快仿真速度)，扩展 b)的表。

(7,4) Hamming 码:

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率					
误块率					

(31,26) Hamming 码:

信道错误率	0.3	0.1	0.03	0.01	0.003
误比特率					
误块率					

5.4 绘制误比特率曲线

在这一小节我们编写 Matlab 代码调用 Simulink 进行批量仿真。

额外需要使用的模块：

Simulink-Sinks-To Workspace

a) 重复码：

将 BSC 模块的差错概率设置为变量 eprob, 此后可通过 Matlab 脚本修改 Base Workspace 中为 eprob 改变取值。添加 To Workspace 模块, 连接到 Error Rate Calculation 的输出端口, 并设置“将数据点限制为最后”为 1, “保存格式”为数组, 设置变量名称为 ErrorStat。

设置仿真停止时间为 100000。

将样例代码 exp5.m 复制到模型所在文件夹, 并修改以下两行对应你的模型名称：

```
open_system('repetition.slx');
```

```
stat = sim('repetition');
```

样例代码遍历仿真了不同重复位数 $n=\{3,4,5\}$ 和不同的信道错误概率下的误比特率。

请根据理论课知识填写(n,1) 重复码的错误概率估计：

```
ber_rep_th(i,j) = % TODO: Theoretical BER
```

然后运行代码, Matlab 将反复调用 Simulink 模型进行仿真, 并最终分别绘制实验数据和理论误比特率的曲线。请根据结果分析仿真结果与理论值的对应关系。

b) Hamming 码：

令 $n = 2^M - 1$, $k = 2^M - 1 - M$, 用变量 M 表示各参数, 修改 Hamming Encoder 和 Hamming Decoder 模块的参数 N 、 K 分别为 $2^M - 1$ 、 $2^M - 1 - M$, 并同时修改延时模块、Buffer 模块的参数 (请推理得到)。修改 BSC 模块差错概率为 eprob。

同样, 添加 To Workspace 模块, 同样连接到 Error Rate Calculation 的输出端口, 并设置“将数据点限制为最后”为 1, “保存格式”为数组, 设置变量名称为 ErrorStat。

设置仿真停止时间为 100000。

另存修改 exp5.m 以绘制(7,4), (15,11), (31,26) Hamming 码的仿真/理论误比特率曲线, 根据结果分析仿真结果与理论值的对应关系。

(提示: (n,k) Hamming 码的错误概率 $P_{BER} \approx \frac{3}{n} P_{BLER} \approx \frac{3}{n} \binom{n}{2} e^2$)

6. 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?
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?
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)