【注意】本文档中所有蓝色文字，是教师对内容的要求或补充说明。学生在提交报告前，应将它们全部删除。

**目录**

[1. 系统设计 2](#_Toc9346)

[1.1. 设计目标 2](#_Toc4243)

[1.2. 设计方案 2](#_Toc6454)

[2. 系统实现 2](#_Toc10810)

[2.1. 信源 2](#_Toc529)

[2.2. 信道 3](#_Toc30422)

[2.3. 信源编解码 3](#_Toc3293)

[2.4. 信道编解码 4](#_Toc29236)

[3. 系统仿真与分析 5](#_Toc32408)

[3.1. 理想情况下 5](#_Toc19489)

[3.2. 信源非理想情况下 6](#_Toc21400)

[3.3. 信道非理想情况下 6](#_Toc21610)

[3.4. 一般非理想情况下 7](#_Toc5634)

[4. 系统改进【可选】 7](#_Toc10245)

[5. 总结 7](#_Toc17607)

[5.1. 设计完成情况 7](#_Toc12429)

[5.2. 经验与收获 8](#_Toc14280)

[6. 附录——相关技术简介 8](#_Toc20583)

**要求**：定稿后更新目录，以保证目录的正确

# 系统设计

## 设计目标

使用计算机编程语言，完整实现一个包含信源、信道、信源编解码、信道编解码等模块的信息传输系统模型，并通过与理论推导的结果进行比对，验证系统实现的正确性。再进一步通过多种参数组合下的仿真，模拟不同通信条件下的信息传递，掌握信息论与编码理论的理论和实践基础。

以上“设计目标”段落文字可以直接复制使用。

## 设计方案

**要求**

* 整体系统的原理方框图、文字说明
* 每个模块的文字说明

# 系统实现

**要求**

* 每个模块是一个独立可执行的程序，模块间通过文件进行数据交互
* 数据文件的有效载荷(payload)，必须经过比特打包(bit-packing)
* 报告正文中，每个模块给出关键的代码“片段”，并提供简要的文字解说

## 信源

### 二元离散无记忆信源（DMS）模块

**要求**

* 输入参数：信源消息概率分布*P*(0)
* 输入参数：消息序列的长度（字节为单位，一般至少10K字节长）
* 模块输出：信源输出消息序列文件

程序API如下：

byteSource.exe INPUT LEN OUTPUT

– INPUT：信源消息概率分布P(0)

– LEN：消息序列的长度（字节为单位，一般至少10K字节长）

– OUTPUT：信源输出消息序列文件的路径

例如：调用格式byteSource.exe 0.5 64 t:\data\byteSource.output.txt

### 信源指标计算模块

**要求**

* 模块输入：信源输出消息序列文件
* 模块输出：（输出形式自定）
  + 数据比特概率分布（即二元DMS的概率分布统计）：*P*(0)
  + 字节概率分布（即二元DMS的8次扩展信源的概率分布统计）：*P*(0), ..., *P*(255)
  + 二元DMS的信息熵（信息比特/二元消息）
  + 二元DSM的信源冗余度

程序API如下：

calcByteSource.exe INPUT OUTPUT

– INPUT：信源输出消息序列文件的路径

– OUTPUT：存放计算结果的路径

其中，OUTPUT文件最好是CSV文件。每次运行calcByteSource.exe之后，将会在OUTPUT文件后添加计算结果（注意：不是覆盖），格式为

"二元DMS的概率分布统计","二元DMS的信息熵","二元DSM的信源冗余度"

“二元DMS的8次扩展信源的概率分布统计”

例如：当调用命令行calcByteSource.exe t:\data\byteSource.output.txt t:\data\calcByteSource.output.csv时，t:\data\calcByteSource.output.csv文件应会增加一行类似以下的结果：

0.5,7.9999,0.00001

0,0.1

1,0.3

2,0.0

3,0.2

4,0.4

5,0

...

255,0

### 单元测试

**要求（同时适用于下文相同部分，有特殊要求的再另行补充）**

* 理论推导
  + 为各输入参数假设至少1个数值，推导出相关指标的结果
  + 详细描述假设条件、推导过程、推导结果
* 测试结果
  + 按上述理论推导中的假设条件设定该模块的输入参数
  + 收集相关指标的实测结果
* 对比理论推导的结果和测试结果，分析异同及原因

byteSource部分：

P(0)=0.2

长度均为64KB

例如：输出文件命名格式：byteSource.p0=0.2.len=64KB.bin

calcByteSource部分：

例如：输出文件命名格式：calc.byteSource. p0=0.2.len=64KB.csv

## 信道

### 二元对称信道（BSC）模块

**要求**

* 模块输入：信道输入消息序列文件
* 输入参数：错误传递概率*p*
* 模块输出：信道输出消息序列文件

程序API如下：

byteChannel.exe INPUT p OUTPUT

例如：调用格式byteChannel.exe 0.5 t:\data\byteChannel.output.txt

### 信道指标计算模块

**要求**

* 模块输入：信道输入消息序列文件、信道输出消息序列文件
* 模块输出：（输出形式自定）
  + 输入消息序列的信息熵（信息比特/二元消息）
  + 输出消息序列的信息熵（信息比特/二元消息）
  + 平均互信息量（信息比特/二元消息）

程序API如下：

calcByteChannel.exe INPUT1 INPUT2 OUTPUT

– INPUT1：信道输入消息序列文件的路径

– INPUT2：信道输出消息序列文件的路径

– OUTPUT：信道计算结果文件的路径

格式为

"输入消息序列的信息熵","输出消息序列的信息熵","平均互信息量"

例如：当调用这个API时，输出结果的文件中会追加以下内容：

"7.98888","7.898998","7.09099090"

### 单元测试

byteChannel部分：

例如：

错误传递概率定为0.4

信道输入消息序列文件命名格式：byteSource.p0=0.2.len=64KB.bin

信道输出消息序列文件命名格式：byteChannel.p=0.4.p0=0.2.len=64KB.bin(这个p是错误传递概率)

calcByteChannel部分：

例如：输出文件命名格式：calc.byteChannel. p=0.4.p0=0.2.len=64KB.txt

## 信源编解码

### 霍夫曼编码模块

**要求**

* 模块输入：编码前的文件
* 输入参数：编码前的文件的字节概率分布
* 模块输出：编码后的文件，包含码书、码字序列

程序API如下：

byteSourceCoder.exe PMF INPUT OUTPUT

PMF path to probability mass function CSV file

INPUT path to the encoder input file

OUTPUT path to the encoder output file

### 霍夫曼解码模块

**要求**

* 模块输入：解码前的文件（格式与霍夫曼编码模块的输出相同）
* 模块输出：解码后的文件

程序API如下：

byteSourceDecoder.exe INPUT OUTPUT

INPUT path to the decoder input file

OUTPUT path to the decoder output file

### 信源编码指标计算模块

**要求**

* 模块输入：编码前的文件、编码后的文件
* 模块输出：（输出形式自定）
  + 压缩比（编码前文件长度:编码后文件长度）
  + 平均码长（码字数据比特/信源字节）
  + 编码效率
  + 编码前的文件的信息熵（信息比特/字节）
  + 编码后的文件的信息熵（信息比特/字节）

程序API如下：

calcByteSourceCoder.exe INPUT1 INPUT2 OUTPUT

– INPUT1：编码前的文件的路径

– INPUT2：编码后的文件的路径

– OUTPUT：信源编码计算结果文件的路径

格式为

"压缩比","平均码长","编码效率"," 编码前的文件的信息熵","编码后的文件的信息熵"

例如：当调用这个API时，输出结果的文件中会追加以下内容：

"7.98888","7.898998","7.09099090","7.898998","7.09099090"

### 单元测试

信源编解码部分：

例如：

编码部分：

信源编码编码前的格式：byteSource.p0=0.2.len=64KB.bin

信源编码编码后的格式：byteSourceCoder.p0=0.2.len=64KB.bin

输入的PMF的格式：PMF.p0=0.2.csv

解码部分：

输入格式：byteSourceCoder.p0=0.2.len=64KB.bin

输出格式：byteSourceDecoder.p0=0.2.len=64KB.bin

信源编码计算部分：

输出格式：calcByteSourceCoder. p0=0.2.len=64KB.txt

## 信道编解码

### 信道编码模块

**要求**

* 模块输入：编码前的文件
* 输入参数：编码方案选择，允许用户自由选择以下6种方案中的一种
  + N长的重复码：N是5、7、9
  + (*n*, *k*)线性分组码：(7,4)、(15,11)、(31,26)
* 模块输出：编码后的文件，包含编码方案相关参数、码字序列

程序API如下：

byteChannelCoder.exe INPUT NUMBER OUTPUT

INPUT path to the encoder input file

NUMBER ： 这六种方案的代号，比如输入参数为1表示第一种方案

OUTPUT path to the encoder output file

### 信道解码模块

**要求**

* 模块输入：解码前的文件（格式与信道编码模块的输出相同）
* 模块输出：解码后的文件

程序API如下：

byteChannelDecoder.exe INPUT OUTPUT

INPUT path to the decoder input file

OUTPUT path to the decoder output file

### 信道编解码指标计算模块

**要求**

* 模块输入：编码前的文件、编码后的文件、解码后的文件
* 模块输出：（输出形式自定）
  + 误码率（汉明失真，错误二元消息数/总二元消息数）
  + 编码前的信源信息传输率（信息比特/二元消息）
  + 编码后的信源信息传输率（信息比特/二元消息）
  + 压缩比（编码前文件长度:编码后文件长度）

程序API如下：

calcByteChannelCoder.exe INPUT1 INPUT2 OUTPUT

– INPUT1：编码前的文件的路径

– INPUT2：编码后的文件的路径

– OUTPUT：信源编码计算结果文件的路径

格式为

"误码率","编码前的信源信息传输率","编码后的信源信息传输率"," 压缩比"

例如：当调用这个API时，输出结果的文件中会追加以下内容：

"0.98743","0.86754","0.9999","1.5345"

### 单元测试

信道编解码部分：

例如：

编码部分：

信道编码编码前的格式：byteSourceCoder.p0=0.2.len=64KB.bin

信道编码编码后的格式：byteChannelCoder.p0=0.2.len=64KB.bin

解码部分：

输入格式：byteChannelCoder .p0=0.2.len=64KB.bin

输出格式：byteChannelDecoder.p0=0.2.len=64KB.bin

信源编码计算部分：

输出格式：calcByteChannelCoder. p0=0.2.len=64KB.txt

# 系统仿真与分析

要求

* 在本节推导以下待测指标的计算公式
  + 信源的信息率*RS*（信息比特/秒）
  + 信道的数据率*rc*（数据比特/秒）
  + 信道的输入信息率*Rci*（信息比特/秒）
  + 信道的输出信息率*Rco*（信息比特/秒）
  + 信宿关于信源的信息率*RI*（信息比特/秒）
  + 信宿的误码率*er*
* 在以下每种情况下
  + 在“条件设定”小节中，详细描述固定条件、可变条件的具体数值
    - 所有情况下的“信源的数据率”固定为*rs*=1（数据比特/秒）
  + 在“理论推导”小节中，计算所有待测指标的理论数值
  + 在“仿真”小节中，描述仿真的操作过程和待测指标的仿真结果
  + 在“分析”小节中，对比待测指标的理论数值和仿真结果，分析异同及原因

## 理想情况下

### 条件设定

**要求**

* 固定条件
  + 信源的数据率*rs*（数据比特/秒）
  + 信源为等概率二元DMS
  + 信道为无噪无损的离散无记忆信道
  + 不使用信源编码
  + 不使用信道编码
* 可变条件
  + 无

### 理论推导

### 仿真

### 分析

## 信源非理想情况下

### 条件设定

**要求**

* 固定条件
  + 信源的数据率*rs*（数据比特/秒）
  + 信道为无噪无损的离散无记忆信道
  + 不使用信道编码
* 可变条件
  + 2种信源概率分布：0.1, 0.3
  + 无信源编码、有信源编码

### 理论推导

### 仿真

### 分析

## 信道非理想情况下

### 条件设定

**要求**

* 固定条件
  + 信源的数据率*rs*（数据比特/秒）
  + 信源为等概率二元DMS
  + 不使用信源编码
* 可变条件
  + 3种信道错误传递概率：0.01, 0.2, 0.5
  + 无信道编码、重复码、(*n*,*k*)线性分组码

### 理论推导

### 仿真

### 分析

## 一般非理想情况下

### 条件设定

**要求**

* 固定条件
  + 信源的数据率*rs*（数据比特/秒）
* 可变条件
  + 2种信源概率分布：0.1, 0.3
  + 3种信道错误传递概率：0.01, 0.2, 0.5
  + 无信源编码、有信源编码
  + 无信道编码、重复码、(*n*,*k*)线性分组码

### 理论推导

### 仿真

### 分析

# 系统改进【可选】

**要求**

* 针对仿真所出现的现象，提出对系统设计、实现、仿真等方面的改进设想。
* 分析说明改进的目的、原理和可行性依据。
* 实施对系统的改进，并通过仿真验证改进的效果。

# 总结

## 设计完成情况

**要求**

* 列出已完成的内容
* 列出未完成的内容，及导致未能完成的原因

## 经验与收获

**要求**：在本次设计过程中，所得到的经验与收获

# 附录——相关技术简介

**要求：**简要介绍本设计中所使用的

* 计算机语言
  + 官方名称、版本号、官网URL
* 主要函数包/模组
  + 只需包含非标准函数包/模组等（不是计算机语言自带的）
  + 官方名称、版本号、官网URL
  + 在本设计中的用途
* 工具软件
  + 编程开发、报告撰写、作图、其他辅助工具等
  + 官方名称、版本号、官网URL
  + 在本设计中的用途