通信原理 实验报告

**2 信道编解码**

## 一 实验内容（10分）

* 1. 信道编解码的实现和验证

信道编解码的实现和验证是通信系统中确保数据传输可靠性的关键步骤。在发送端，信道编码通过增加冗余信息来提高数据的抗干扰能力。常见的编码方式包括线性分组码（如汉明码）、卷积码、Turbo码等。发送端首先对原始二进制码流进行编码，生成带有冗余的码字，然后进行星座映射、组帧和成型滤波等处理，最终通过DAC转换为模拟信号发送。在接收端，信号经过ADC采样后，进行匹配滤波、信道均衡等处理，然后进行信道解码，恢复原始信息。验证过程包括系统测试和误码率计算。通过对比添加信道编解码前后的误码率，可以评估编码对系统性能的提升效果。实验结果表明，信道编解码能显著降低误码率，提高通信系统的可靠性。

**1. 编码实现：**

编码器根据特定算法将原始数据转换为冗余码字，以增强抗干扰能力。常见的编码方式包括：

* **卷积编码：** 通过移位寄存器和模2加法器生成冗余位，适用于连续数据流。
* **Turbo码：** 结合两个卷积编码器和交织器，接近香农极限，适合高误码率场景。
* **LDPC码：** 基于稀疏校验矩阵，具有接近Turbo码的性能，且译码复杂度较低。
* **交织码、Polar码、BCH**

**2. 解码实现：**

解码器通过接收到的码字恢复原始数据，常见的解码算法有：

* **Viterbi算法：** 用于卷积码，通过最大似然估计找到最可能的发送序列。
* **BCJR算法：** 用于Turbo码，基于最大后验概率估计，性能优于Viterbi算法。
* **置信传播算法：** 用于LDPC码，通过迭代更新变量节点和校验节点的置信度进行解码。

**3. 验证：**

验证信道编解码性能通常通过仿真和实际测试完成：

* **仿真验证：** 使用MATLAB等工具模拟不同信道条件下的误码率（BER）和误帧率（FER），评估编码增益。
* **实际测试：** 在真实通信环境中测试，结合硬件平台验证编解码器的实时性和可靠性。

通过实现和验证，信道编解码技术能够有效提升通信系统的抗干扰能力，确保数据传输的准确性。

## 二 实验原理（50分）

2.1 差错控制编码的分类（10分）

信道编解码的实现和验证是通信系统中确保数据传输可靠性的关键步骤。在发送端，信道编码通过增加冗余信息来提高数据的抗干扰能力。常见的编码方式包括线性分组码（如汉明码）、卷积码、Turbo码等。发送端首先对原始二进制码流进行编码，生成带有冗余的码字，然后进行星座映射、组帧和成型滤波等处理，最终通过DAC转换为模拟信号发送。在接收端，信号经过ADC采样后，进行匹配滤波、信道均衡等处理，然后进行信道解码，恢复原始信息。验证过程包括系统测试和误码率计算。通过对比添加信道编解码前后的误码率，可以评估编码对系统性能的提升效果。实验结果表明，信道编解码能显著降低误码率，提高通信系统的可靠性。

2.2 线性分组码的编码、校验、纠错原理（20分）

线性分组码的编码过程是将k位信息位通过生成矩阵G扩展为n位码字，其中n-k位为监督位。生成矩阵G的每一行对应一个基向量，码字是信息位与基向量的线性组合。校验矩阵H用于检测和纠正错误，满足G\*H^T=0。接收端通过计算伴随式S=v'\*H^T来判断是否存在错误，S只与错误位置向量e有关。如果S不为零，则说明存在错误，通过查找错误位置向量e可以纠正错误。线性分组码的纠错能力取决于码距，码距越大，纠错能力越强。例如，汉明码(7,4)可以纠正单个错误，检测两个错误。线性分组码通过增加冗余信息，提高了数据传输的可靠性。

2.3 码间串扰产生的原因（10分）

码间串扰（ISI）是由于信道带宽限制导致的信号拖尾现象。当信号在有限带宽信道中传输时，时域上的拖尾会叠加到其他码元上，影响其他码元的正确判决。ISI的产生主要是因为信道的带限特性，导致信号在时域上无限延伸，拖尾部分会干扰相邻码元的判决。此外，信号的调制方式、信道特性以及接收端的采样时刻也会影响ISI的严重程度。为了减少ISI，通常采用成型滤波器和匹配滤波器来优化信号的时域和频域特性，使得信号的拖尾在判决时刻为零，从而消除码间串扰的影响。

2.4 成型滤波、匹配滤波的原理和作用（10分）

成型滤波器和匹配滤波器是通信系统中用于减少码间串扰（ISI）和优化信号传输的重要工具。成型滤波器在发送端使用，通常采用根升余弦滤波器，其作用是对调制后的信号进行带限处理，使得信号在时域上的拖尾在判决时刻为零，从而减少ISI。匹配滤波器在接收端使用，与发送端的成型滤波器相匹配，其作用是最大化信噪比（SNR），并进一步减少ISI。匹配滤波器的频率响应与发送信号的频谱共轭匹配，使得在最佳采样时刻信号的功率最大，噪声的功率最小。通过成型滤波和匹配滤波的配合，可以有效消除码间串扰，提高信号的传输质量和系统的抗干扰能力。

## 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）

1、编码

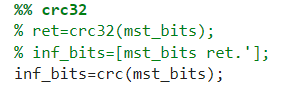
试验文件crc.mlx

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | g = [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];  bits = [1 0 1 0];  n = numel(bits);  bitts = reshape(bits, 4, 1).';  bit\_h = zeros(n/4, 7);    for i = 1:n/4      bit\_h(i, :) = mod(bitts(i, :) \* g, 2);  end    bit\_s = reshape(bit\_h.', 7 \* n/4, 1).' |

验证可靠后将其修改为能在demo文件中使用的函数crc74.m

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14 | function inf\_bits = crc74(bits)  g=[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];%生成矩阵  n=length(bits);  bitts=reshape(bits,4,n/4);  bitts=bitts';  for i=1:n/4      bit\_h(i,:)=bitts(i,:)\*g;      for j=1:7          bit\_h(i,j)=mod(bit\_h(i,j),2);      end  end  bit\_s=reshape(bit\_h',7\*n/4,1);  bit\_s=bit\_s';  inf\_bits=bit\_s(1:2048); %取前2048位 |

再将其插入qam16\_tx\_func.m即可使用



2、解码

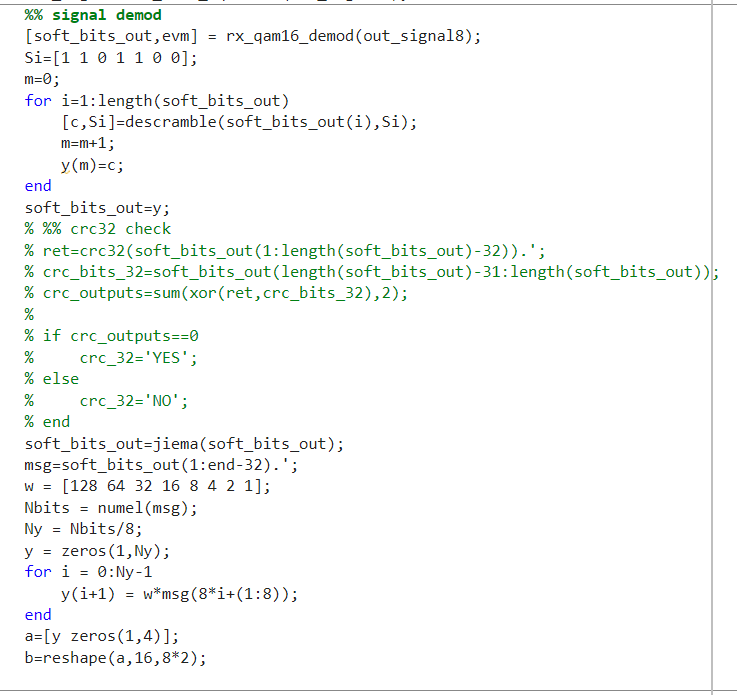
试验文件jiema.mlx（由于试验文件在删掉分号后出现难以解决的报错，反复修改无法解决，故使用了交大deepseek重构了代码来解决报错问题，因此代码结构和函数文件不同）

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | bit\_s = [1 1 1 0 1 0 0];  n = length(bit\_s);  h = [1 0 1 0 1 0 1; 0 1 1 0 0 1 1; 0 0 0 1 1 1 1];  a = 0;  bits = zeros(floor(n / 7), 4);  s = zeros(floor(n / 7), 3);    for i = 1:floor(n / 7)      start\_idx = (i - 1) \* 7 + 1;      end\_idx = i \* 7;      current\_bits = bit\_s(start\_idx:end\_idx);        s(i, :) = mod(current\_bits \* h', 2);      t = s(i, :) \* [1; 2; 4]        if t == 0          bits(i, 1:4) = current\_bits([3, 5, 6, 7]);      else          a = a + 1;          current\_bits(t) = 1 - current\_bits(t);          bit\_s(start\_idx:end\_idx) = current\_bits;          bits(i, 1:4) = current\_bits([3, 5, 6, 7]);      end  end    bits = reshape(bits', 1, 4) |

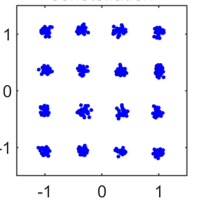
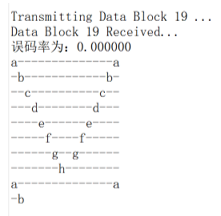
验证可靠后将其修改为能在demo文件中使用的函数jiema74.m

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | function out\_bits=jiema74(bit\_s)  n=length(bit\_s);  h=[1 0 1 0 1 0 1;0 1 1 0 0 1 1;0 0 0 1 1 1 1];  bit\_s=bit\_s(1:2044);  a=0;  bit\_h=reshape(bit\_s,7,2044/7);  bit\_h=bit\_h';  for i=1:floor(n/7)      s(i,:)=bit\_h(i,:)\*h';      for j=1:3          s(i,j)=mod(s(i,j),2);      end      t=s(i,:)\*[1;2;4];      if t==0           bits(i,1)=bit\_h(i,3);         bits(i,2)=bit\_h(i,5);         bits(i,3)=bit\_h(i,6);         bits(i,4)=bit\_h(i,7);      else         a=a+1;         bit\_h(i,t)=1-bit\_h(i,t);         bits(i,1)=bit\_h(i,3);         bits(i,2)=bit\_h(i,5);         bits(i,3)=bit\_h(i,6);         bits(i,4)=bit\_h(i,7);      end  end  fprintf('误码率为：%f\n',a/floor(n/7))    bits=reshape(bits',4\*floor(n/7),1);  out\_bits=bits';  out\_bits=[out\_bits zeros(1,880)]; |

再将其插入qam16\_rx\_func.m即可使用

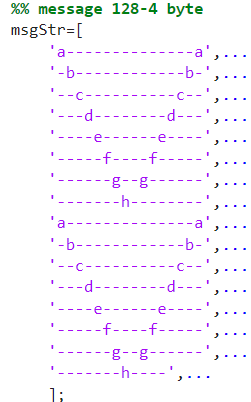


3.2 从误码率分析系统性能（10分）

在刚开始输出时存在乱码，是因为手法不同步导致的，到后面几次收发就稳定下来，误码率变为0。同时输出其实是不完全的，因为发送的数据为原始数据截取2048得到的，省略了后面的数据，所以输出的数据只有一半多一点。

原始数据如下：



在实验条件下（用线连接），干扰较小，使用74海明码很容易将误码率降为0，是因为其纠错单个错误的能力。可以推测出，当干扰较大的条件下，只能纠正部分错误，因此只能降低部分误码率。

## 四 思考题（10分）

4.1 符号率和DAC采样率的关系

4.1 调研4G和5G中的信道编码的名称、分类、码长、编码效率。

在4G和5G通信系统中，信道编码是确保数据传输可靠性的关键技术。4G LTE系统主要采用Turbo码和咬尾卷积码，而5G NR系统则引入了LDPC码和Polar码。

1. 4G LTE中的信道编码：
   * Turbo码：Turbo码是一种并行级联卷积码，具有接近香农极限的性能。它通过两个卷积编码器和一个交织器实现，编码效率较高，通常用于数据信道的编码。Turbo码的码长可以根据需要调整，常见的码长为几十到几千比特，编码效率通常在1/3到1/2之间。
   * 咬尾卷积码：咬尾卷积码是一种特殊的卷积码，其编码器的初始状态和结束状态相同，避免了传统卷积码的尾比特开销。咬尾卷积码主要用于控制信道的编码，码长较短，通常为几十比特，编码效率为1/3。
2. 5G NR中的信道编码：
   * LDPC码：LDPC（低密度奇偶校验码）是一种线性分组码，具有稀疏的校验矩阵，译码复杂度低且性能优异。LDPC码在5G中用于数据信道的编码，码长可以从几十到几千比特不等，编码效率通常在1/3到9/10之间。
   * Polar码：Polar码是一种基于信道极化理论的编码方式，具有理论上达到香农极限的性能。Polar码在5G中主要用于控制信道的编码，码长较短，通常为几十到几百比特，编码效率为1/2。

4G和5G中的信道编码在码长和编码效率上有所不同，5G通过引入LDPC码和Polar码，进一步提升了编码效率和译码性能，适应了更高数据速率和更低延迟的需求。

4.2 选择4.1中的1种，概述编码过程和当前主流的译码方法。

LDPC码（低密度奇偶校验码）是5G NR系统中用于数据信道的主要编码方式。其编码过程基于稀疏的校验矩阵，具有较高的编码效率和较低的译码复杂度。

1. LDPC码的编码过程：

* 生成校验矩阵：LDPC码的核心是一个稀疏的校验矩阵H*H*，矩阵中的大部分元素为0，少数元素为1。校验矩阵的稀疏性使得译码复杂度较低。
* 信息位与校验位的计算：给定信息位向量u*u*，编码器通过校验矩阵H*H*生成校验位向量p*p*，使得H⋅[u,p]T=0*H*⋅[*u*,*p*]*T*=0。编码后的码字v*v*由信息位和校验位组成，即v=[u,p]*v*=[*u*,*p*]。
* 系统化编码：LDPC码通常采用系统化编码方式，即编码后的码字中直接包含原始信息位，便于译码时直接提取信息。

2. LDPC码的主流译码方法：

* 置信传播算法（Belief Propagation, BP）：BP算法是LDPC码最常用的译码方法。它通过在因子图上进行消息传递，迭代更新每个比特的置信度，最终根据置信度判决每个比特的值。BP算法的性能接近最大后验概率（MAP）译码，但复杂度较低。
* 最小和算法（Min-Sum Algorithm）：为了进一步降低复杂度，最小和算法对BP算法进行了简化，用最小值运算代替了复杂的对数似然比计算。虽然性能略有损失，但复杂度显著降低，适合硬件实现。
* 分层译码（Layered Decoding）：分层译码是一种改进的BP算法，将校验矩阵分为多个层，逐层进行消息传递。这种方法加快了收敛速度，适合高吞吐量的应用场景。

LDPC码通过稀疏的校验矩阵和高效的译码算法，在5G系统中实现了高编码效率和低译码复杂度，成为5G数据信道编码的主流选择。