



LDPC编译码作业

信息论与编码理论 期中大作业

助教：朱洪飞

18813144329

zhuhongfei@pku.edu.cn

英杰交流中心330

2021年5月10日

编码部分书面作业



- ▶ 已交作业名单如下
- ▶ 尚未提交作业同学请抓紧交给我！

序号	姓名
1	吴海涛
2	何必博
3	刘容玮
4	陶子涵
5	汪奕汝
6	杨其鹏
7	郑家瑜
8	周欣怡
9	朱兰鑫
10	李文瑶
11	苏睿
12	毕涛
13	赵天一

14	陈伟
15	薛亮
16	池亚妮
17	方堃
18	焦响
19	李希仁
20	王泳诺
21	卫昱宏
22	魏子义
23	于慧波
24	周健
25	罗耿坤
26	余奕壕
27	刘致远

主要内容



- LDPC码简介（针对本次作业）
- LDPC编码
- 调制过信道
- LDPC译码
- 作业提交

主要内容



- **LDPC码简介（针对本次作业）**
- LDPC编码
- 调制过信道
- LDPC译码
- 作业提交

- ▶ 本次作业中采用系统码设计，信息序列长1008比特，码长2016比特，码率1/2，即 $N = 2016$, $K = 1008$, $R = 1/2$

- ▶ 校验矩阵H表示如下：

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,n_b} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,n_b} \\ \vdots & \vdots & \ddots & \vdots \\ H_{m_b,1} & H_{m_b,2} & \cdots & H_{m_b,n_b} \end{bmatrix}$$

- ▶ $H_{i,j}$ 是大小为 $z \times z$ 的循环移位矩阵，行重为1；它的值表示该矩阵的循环位移偏移量，也是第一行中元素1所处的列的位置；若为0，表示是全0矩阵

- ▶ 举例：若 $z=7$ ， $H_{ij}=4$ ，则其矩阵结构为

$$H_{ij} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- ▶ 使用 H_block 代表 H 矩阵的 H_{ij} 表示形式，则在本次作业中， $z=56$ ， $H_block(18*36)$ 由“Matrix(2016,1008)Block56.mat”给出


$$N = 2016, K = 1008, R = \frac{1}{2}$$

- ▶ 使用H_block生成的H能被分成两部分： $\left[H_p \mid H_s \right]$

$$H = \begin{bmatrix} H_{1,1} & \cdots & H_{1,m_b} & H_{1,m_b+1} & \cdots & H_{1,n_b} \\ H_{2,1} & \cdots & H_{2,m_b} & H_{2,m_b+1} & \cdots & H_{2,n_b} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_{m_b,1} & \cdots & H_{m_b,m_b} & H_{m_b,m_b+1} & \cdots & H_{m_b,n_b} \end{bmatrix}$$

- ▶ H_p 对应校验比特部分，大小为 $m_b z * m_b z$ ；
- ▶ H_s 对应信息比特部分，大小为 $m_b z * k_b z$ ，其中 $k_b = n_b - m_b$ 。

- ▶ H_p 具有如下结构 ($H_{i,j}$ 表示) :

$$H_p = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & \boxed{a} \\ 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & & & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots & \vdots \\ 0 & & & \ddots & 1 & 0 & 0 \\ 0 & 0 & & & 1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1 \end{bmatrix}$$


- ▶ 特别地，矩阵 a 具有如下结构 (一定注意!!!)

$$\begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

主要内容



- LDPC码简介（针对本次作业）
- **LDPC编码**
- 调制过信道
- LDPC译码
- 作业提交

- ▶ LDPC码通常由校验矩阵（也即前面的H）进行定义。尽管线性分组码可以使用传统的生成矩阵进行编码，但是要通过H求解G在实现上较为困难，因此根据校验矩阵直接进行编码。
- ▶ 由于本次作业采用系统码，所以编码方法：
 - ▶ 利用输入信息比特 $s = [s_1, s_2, \dots, s_{k_b z}]$
 - ▶ 得到校验比特为 $p = [p_1, p_2, \dots, p_{m_b z}]$
 - ▶ 按如下方法计算出p后，[p s]即可完成编码

▶ 基于H矩阵的通用LDPC编码：

$$Hx^T = 0 \Rightarrow \begin{bmatrix} H_p & H_s \end{bmatrix} \begin{bmatrix} p^T \\ s^T \end{bmatrix} = 0 \Rightarrow p = s H_s^T (H_p^T)^{-1}$$

Diagram illustrating the dimensions of the matrices in the equation:

- p is a $1 \times K$ vector (indicated by an upward arrow from $1 \times K$ to p).
- s is a $1 \times (N-K)$ vector (indicated by a downward arrow from s to $1 \times (N-K)$).
- H_s^T is a $(N-K) \times (N-K)$ matrix (indicated by an upward arrow from $(N-K) \times (N-K)$ to H_s^T).
- $(H_p^T)^{-1}$ is a $K \times (N-K)$ matrix (indicated by a downward arrow from $(H_p^T)^{-1}$ to $K \times (N-K)$).

▶ 要求： H_p 可逆



LDPC 编码算法 2

Step1 : 利用输入矢量s计算中间结果

$$w = sH_s^T \quad w = [w_1, w_2, \dots, w_{m_b z}]$$

Step2 : 利用x计算校验比特

$$p_i = \begin{cases} w_i & , i = 1 \\ w_i \oplus p_{(m_b-1)z+i-1} & , 1 < i \leq z \\ w_i \oplus p_{i-z} & , i > z \end{cases} \quad p = [p_1, p_2, \dots, p_{m_b z}]$$

Step3 : 组合p与s

$$x = [p, s]$$

检验 : (一定得多试几组数据, 保证编码正确)

$$Hx^T = 0$$



两种算法的比较



▶ 时间复杂度比较

	比特乘法次数	比特加法次数
算法1	$K(N - K) + (N - K)(N - K)$	$(K - 1)(N - K) + (N - K - 1)(N - K)$
算法2	$K(N - K)$	$(K - 1)(N - K) + N - K - 1$

注： $(H_p^T)^{-1}$ 可以离线求解，所以不占用复杂度

▶ 算法1适用于通用的H矩阵（只要 H_p 可逆）

▶ 算法2适用于本次作业特定的H矩阵

▶ 本次作业要求采用时间复杂度较低的**算法2**进行编码

主要内容



- LDPC码简介（针对本次作业）
- LDPC编码
- **调制过信道**
- LDPC译码
- 作业提交



调制过信道

▶ 本次作业采用**实数**BPSK调制

- ▶ 比特0映射成符号1，比特1映射成符号-1

$$d = 1 - 2x$$

- ▶ 实数BPSK，SNR与 E_b/N_0 的转换关系：

$$(SNR)_{dB} = \left(\frac{E_b}{N_0}\right)_{dB} + 10\log_{10}(R) - 10\log_{10}\left(\frac{1}{2}\right), \quad R = \frac{K}{N}$$

- ▶ 信道加噪用SNR，性能画图用 E_b/N_0 ！

E_b/N_0 与SNR转换详细解释见压缩包内word文档！

▶ 本次作业在AWGN信道下进行误码率性能仿真

▶ MATLAB给信号加AWGN噪声的两种方式（建议用法一）：

- ▶ 法一：根据信号功率及SNR计算噪声功率 σ^2 ，用randn函数产生噪声 n 再添加

$$y = d + n$$

- ▶ 法二：直接用awgn函数产生加噪之后的信号

$$y = \text{awgn}(d, \text{SNR_dB}, 'measured')$$

▶ 实数BPSK调制，AWGN信道下接收信号 y 的对数似然比（LLR）

$$LLR(y) = \ln \frac{p(y | x = 0)}{p(y | x = 1)} = \ln \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+1)^2}{2\sigma^2}}} = \frac{2}{\sigma^2} y, \quad \text{其中 } \sigma^2 = \frac{1}{(\text{SNR})_{\text{linear}}}$$

主要内容



- LDPC码简介（针对本次作业）
- LDPC编码
- 调制过信道
- **LDPC译码**
- 作业提交



- ▶ **本次作业要求实现四种译码算法，分别是：**
 - ▶ 和积算法 (Sum-Product , SP , 系统仿真中常用)
 - ▶ 最小和算法 (Min-Sum , MS , 硬件实现常用)
 - ▶ 归一化最小和算法(Normalized Min-Sum , NMS)
 - ▶ 偏置最小和算法(Offset Min-Sum , OMS)

- ▶ **后两种算法是在最小和算法的基础上进行一点修正以提高其性能**



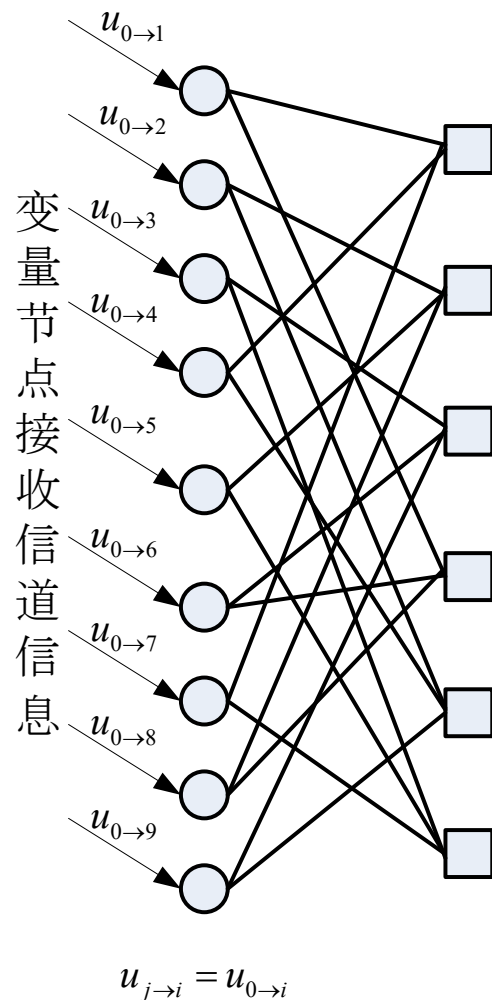
- ▶ **(N,K)LDPC码的二分图包括**
 - ▶ N个变量节点
 - ▶ N-K个校验节点
 - ▶ 若干条边

- ▶ **度 (Degree)**
 - ▶ 变量节点的度
 - ▶ 校验节点的度

- ▶ **和积算法译码过程可以分为三步**
 - ▶ 初始化
 - ▶ 置信度传递
 - ▶ 判决与迭代

▶ 第一步：初始化

- ▶ 每个比特从信道中接收到的信道信息作为初始置信度，也即变量节点的置信度 $u_{j \rightarrow i} = u_{0 \rightarrow i}$ ，其中 $u_{0 \rightarrow i}$ 表示第 i 个比特从信道接收到的LLR $2y_i / \sigma^2$ 。

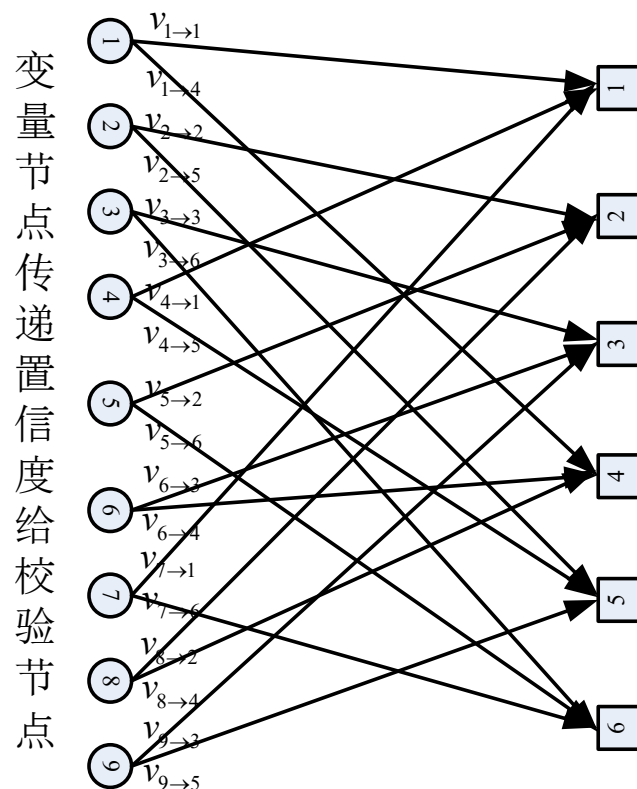


▶ 第二步：置信度传递

- ▶ 1) 每个变量节点要将自己的置信度传递给具有校验关系的校验节点。第*i*个变量节点传递给第*j*个校验节点的置信度计算方法

$$v_{i \rightarrow j} = u_{0 \rightarrow i} + \sum_{k=1, k \neq j}^{d_i^v} u_{k \rightarrow i}$$

其中 d_i^v 表示第*i*个变量节点的度

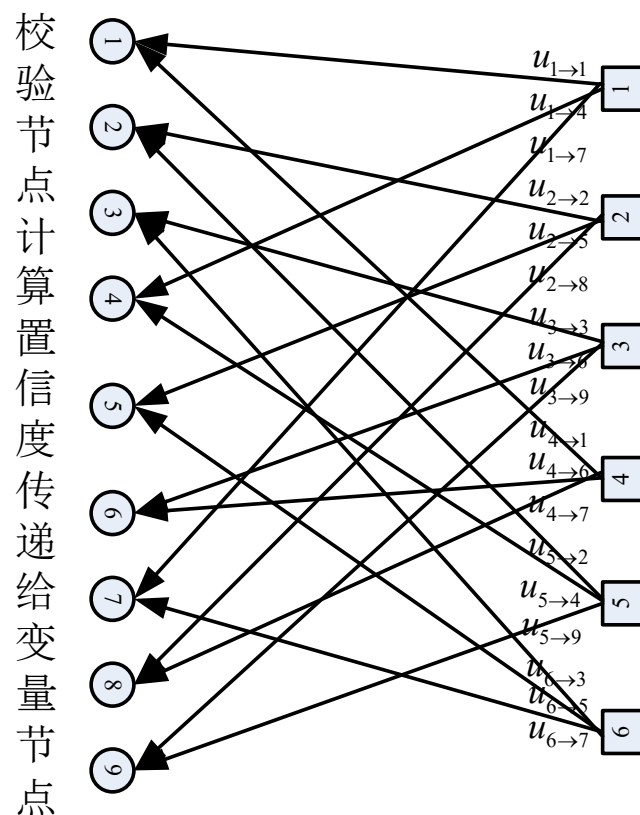


► 第二步：置信度传递

- 2) 校验节点根据变量节点传递来的置信度，更新每一个变量节点的置信度。第j个校验节点为第i个变量节点更新的置信度为

$$u_{j \rightarrow i} = 2 \tanh^{-1} \left(\prod_{k=1, k \neq i}^{d_j^c} \tanh \left(\frac{v_{k \rightarrow j}}{2} \right) \right)$$

其中 d_j^c 表示第j个校验节点的度



第三步：判决

1) 置信度更新完毕后做硬判决

$$\hat{x}_i = \begin{cases} 1 & \text{if } \sum_{k=0}^{d_i^v} u_{k \rightarrow i} < 0 \\ 0 & \text{if } \sum_{k=0}^{d_i^v} u_{k \rightarrow i} \geq 0 \end{cases}$$

2) 如果所有的校验关系得到满足，即 $H\hat{x}^T = 0$ ，其中 $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N)$ ，则译码结束，取出

$$\hat{s} = (\hat{x}_{N-K+1}, \hat{x}_{N-K+2}, \dots, \hat{x}_N)$$

即为信息位 s 的估计；如果不成立，则返回迭代的第二步继续迭代，直至达到迭代次数的上限。

- ▶ 因为和积运算的积运算部分需要进行的反双曲正切连乘等非常复杂的运算，在硬件实现中通常只能利用查表等相对繁琐的操作来完成，因此需要进行改进。
- ▶ 最小和算法与和积算法大同小异，区别在于积算法不再用

$$u_{j \rightarrow i} = 2 \tanh^{-1} \left(\prod_{k=1, k \neq i}^{d_j^c} \tanh \left(\frac{v_{k \rightarrow j}}{2} \right) \right)$$

而用

$$u_{j \rightarrow i} = \left(\prod_{k=1, k \neq i}^{d_j^c} \text{sign}(v_{k \rightarrow j}) \right) \left(\min_{k \in \{1, 2, \dots, d_j^c\} \setminus i} (|v_{k \rightarrow j}|) \right)$$

- ▶ 归一化最小和算法建立在最小和算法之上，在计算校验节点为变量节点更新的置信度时引入归一化系数 α ，以减小Min-Sum译码消息的幅度

$$u_{j \rightarrow i} = \left(\prod_{k=1, k \neq i}^{d_j^c} \text{sign}(v_{k \rightarrow j}) \right) \left(\min_{k \in \{1, 2, \dots, d_j^c\} \setminus i} (|v_{k \rightarrow j}|) \right) * \alpha$$

- ▶ α 的取值在0到1之间，这种算法中通过引入 α 减小了译码消息的幅度，取值合适时性能逼近和积算法
(为了简化要求， α 取值精确到小数点后一位即可)

- ▶ 偏置最小和算法也是建立在最小和算法之上，在计算校验节点为变量节点更新的置信度时引入偏置量 β

$$u_{j \rightarrow i} = \left(\prod_{k=1, k \neq i}^{d_j^c} \text{sign}(v_{k \rightarrow j}) \right) \left(\max \left[\min_{k \in \{1, 2, \dots, d_j^c\} \setminus i} (|v_{k \rightarrow j}|) - \beta, 0 \right] \right)$$

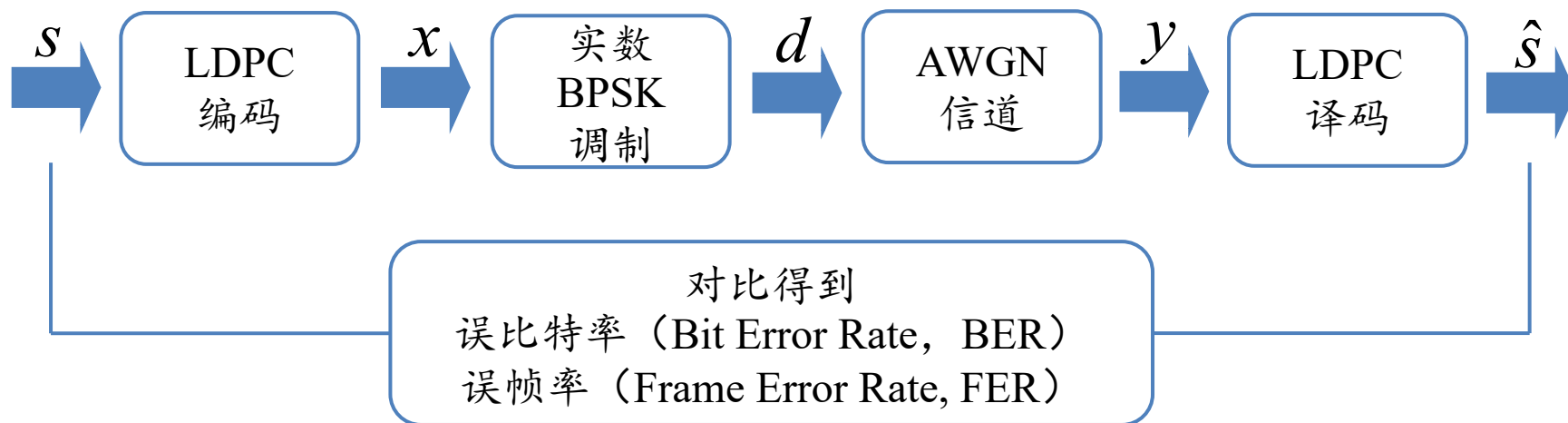
- ▶ $\min_{k \in \{1, 2, \dots, d_j^c\} \setminus i} (|v_{k \rightarrow j}|)$ 小于 β 时，我们将其置为零，也就是消除了其对于下一步信息节点更新中的贡献
- ▶ β 的取值在0到1之间，这种算法使得过小的校验信息不再传递到变量节点中，取值合适时性能逼近和积算法

主要内容



- LDPC码简介（针对本次作业）
- LDPC编码
- 调制过信道
- LDPC译码
- **作业提交**

- 作业要求：用MATLAB平台搭建一个简单的仿真系统，验证LDPC编解码的性能





▶ 仿真要求

- ▶ 对于两个修正最小和算法，应选定一个 E_b/N_0 （1dB附近），仿真BER得出 α 和 β 最佳值。 α 和 β 取值范围均为0到1，步进为0.1， α 和 β 的BER曲线各画一张图。
- ▶ 仿真得出 E_b/N_0 在-1dB到2dB（步进0.5dB）时四种译码算法的BER和FER，注意两个修正最小和算法的 α 和 β 都取最佳值。应画BER一张图，FER一张图，每一张图都包含四种算法。
- ▶ 作图应使用MATLAB的semilogy函数，横坐标为 $(E_b/N_0)_{dB}$ 。

▶ 仿真建议

- ▶ 每一帧的仿真，建议设置最大迭代次数30。
- ▶ 每一个信噪比的仿真，建议停止条件： $E_b/N_0 \leq 1dB$ ，错误的帧数超过50； $E_b/N_0 > 1dB$ ，错误的帧数超过3。



► Tips :

- 善用矩阵运算、向量运算，结合LDPC编解码的方法可以有效提升程序运行速度。
- 若发现性能总有一个门限，或高信噪比时依然经常迭代满次数上限，考虑检查H矩阵是否严格和PPT的一样。
- 双曲正切、反正切运算中由于Matlab精度问题可能出现值为“无穷”的情况，此时应特殊处理一下。
- 可以尝试使用C/C++编写解码模块，用mex命令在matlab中调用C/C++程序，可以有效提升程序运行速度（mex混合编程相关说明见word文档）。此部分为附加内容，有能力的同学建议尝试。

作业提交



◆提交途径：

- ◆教学网——教学内容——2021LDPC编译码大作业

◆提交内容：

- ◆全部源码文件及PDF说明文档
- ◆说明文档请包含性能曲线以及相关分析说明

◆提交形式

- ◆将全部文件打包为zip文件上传
- ◆zip文件命名为“学号_姓名_LDPC编译码大作业”，
如“1701111292_朱洪飞_LDPC编译码大作业”

◆截止日期：2021年5月23日（周日）24点

温馨提示：

代码量少，但是仿真时间较长，建议尽早做



**有任何问题随时问我！
祝大家编程愉快！
谢谢大家！**