

实验一 信道编码

董皓戡 无 37 2023010659

2025 年 11 月 3 日

1 联合调试

1.1 模块划分

整体工作流中按顺序包含以下模块：

1. 原始图像：lena_128_bw.bmp；
2. 量化：采用经单独调试后效果最好的 JPEG 量化；
3. 熵编码：生成单、双符号的 Huffman 码本并进行基于条带的编码；
4. 比特-电平映射：采用 ROTATED_BPSK, QPSK, 8PSK 分别进行 1, 2, 3 比特/符号的映射；
5. 符号处理：采用在数据符号中插入导频的方式进行信道估计；
6. 信道：经过复电平序列信道传输；
7. 符号处理：通过导频进行信道估计，并进行 ML 判决；
8. 比特-电平反映射：进行 ROTATED_BPSK, QPSK, 8PSK 的反映射；
9. 熵解码：根据对应码本进行基于条带的解码；
10. 反量化：JPEG 反量化；
11. 重构图像。

1.2 调试过程

实验中将信源编解码部分和信道部分分别进行调试，主要针对场景三和条件三，噪声方差 $\sigma_n^2 = 0.01$ ，信道调试结果为 $N = 2, x_p = 2 + 0j$ 。具体含义和原因见第 2 节。

在联合调试过程中，扫描参数按如下设置：

- 单、双符号 Huffman 编码；
- JPEG 量化因子：[5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]；
- VLC 条带选项：[0, 1, 2, 3, 4]；
- 比特/符号：[1, 2, 3]；

- Huffman 编码逃逸码阈值: [1, 2, 3, 4, 5].

扫描上述参数发现信道中真实的信噪比几乎不变, 于是选择重建图像 PSNR 最高的参数组合作为最终联合调试结果: 双符号 Huffman 编码、量化因子: 70、VLC 条带选项: 1、Huffman 编码逃逸码阈值: 4、比特/符号: 1. 最终重构图像如图 1 所示, 重构图像 PSNR 为 18.246784 dB, 信道传输的信噪比为 24.770162 dB. 人眼观察, 图像虽有部分失真, 但都局限在小区域内. 在较远处观察与原图差别不大. 具体的参数扫描结果见压缩包中文件 (代码结构见附录 ??).



图 1: 重构图像

2 信道处理

本人主要负责信道处理模块的设计与实现. 主要对场景三: $b = 0.7, \rho = 0.996$ 进行仿真. 其他场景可看作场景三的简化:

1. 场景一: $b = 0$, 信道变为 $y_i = x_i + n_i$, 即 AWGN 信道, 取消信道估计即可;
2. 场景二: $b = \rho = 1$, 信道变为 $y_i = a_i x_i + n_i$, 但信道增益 a_i 恒定不变, 仅需在一次通信开始时进行信道估计即可.

主要针对条件三: 收发双方均不知道 β_i 进行仿真. 其他条件:

1. 条件一: 收发双方在完整一次通信过程之前, 已提前知道所有的 β_i : 发送方根据信道增益调整发送符号或接收方根据信道增益调整判决电平均可;
2. 条件二: 收方提前知道所有的 β_i , 而发方完全不知道: 接收方根据信道增益调整判决电平即可, 发送方无需调整发送符号.

2.1 比特-电平映射

本部分由林天皓同学设计，对 1, 2, 3 比特/符号分别采用 ROTATED_BPSK, QPSK, 8PSK 进行映射，相应的星座图如图 2 所示。

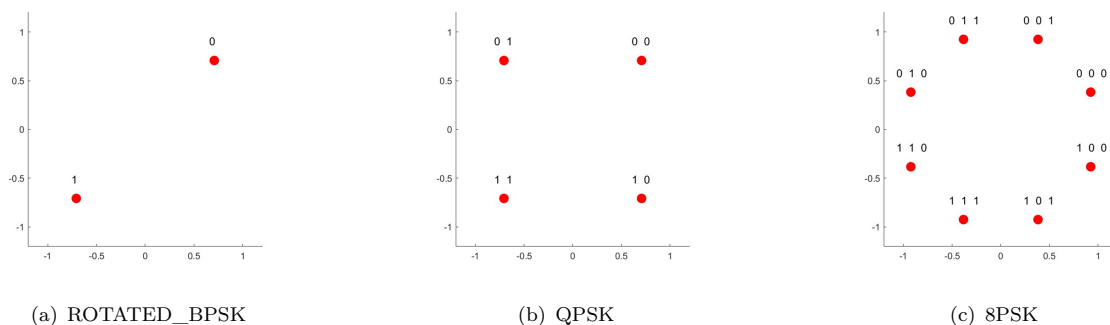


图 2: 比特-电平映射星座图

2.2 复采样信道

复采样信道的输入输出关系为:

$$y_i = a_i x_i + n_i \quad (1)$$

其中 $a_i = \sqrt{1 - b^2} + b\beta_i$, $b \in [0, 1]$, $\beta_i = \rho\beta_{i-1} + \sqrt{1 - \rho^2}z_i$, z_i 和 β_1 的实部和虚部独立，且服从零均值、方差为 0.5 的高斯分布。所有 n_i 的实部和虚部独立，且服从零均值、方差为 $\sigma_n^2/2$ 的高斯分布。

取 $b = 0.7, \rho = 0.996, \sigma_n^2 = 0.1$ ，并取输入为 $\{1 + 0j, -1 + 0j, 0 + 1j, 0 - 1j\}$ ，得到信道输出星座图如图 3 所示。

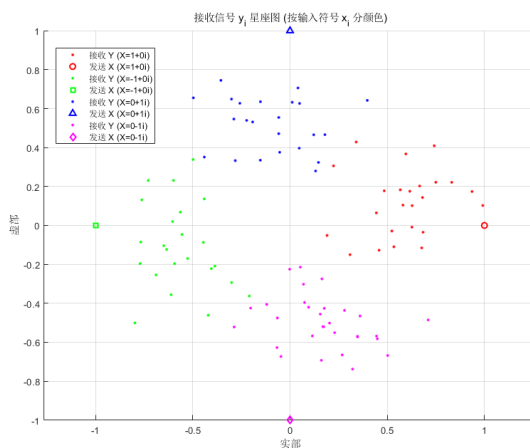


图 3: 复采样信道输出星座图

2.3 复电平序列信道

连续使用复采样信道，抽象成复电平序列信道。具体关系如图 4 所示。

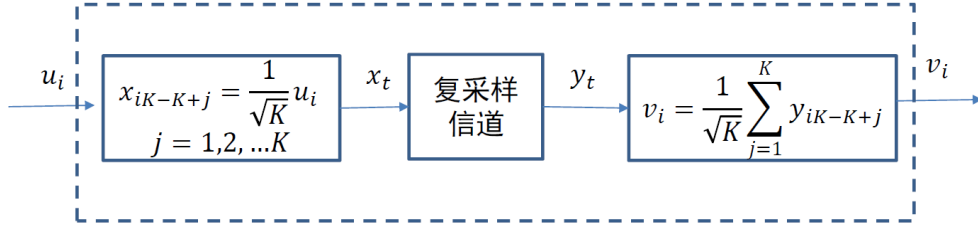


图 4: 复电平序列信道示意图

将复电平序列信道抽象为:

$$y_i = h_i x_i + n_{eff,i} \quad (2)$$

其中, $h_i = \frac{1}{K} \sum_{j=1}^K a_{ij}$, $n_{eff,i} = \frac{1}{\sqrt{K}} \sum_{j=1}^K n_{ij}$. 分析复电平序列信道与内核的复采样信道的信噪比得:

$$SNR_{core} = \frac{(1 - 0.5b^2)P_x}{\sigma_n^2/2}, \quad SNR_{seq} = \frac{\mathbb{E}[|h_i|^2]}{\sigma_n^2/2} P_x \quad (3)$$

$$G \triangleq \frac{SNR_{seq}}{SNR_{core}} = K \frac{\mathbb{E}[|h_i|^2]}{\mathbb{E}[|a_i|^2]} = \begin{cases} K, & \rho \rightarrow 1 \\ \frac{K(1-b^2)+0.5b^2}{1-0.5b^2}, & \rho \rightarrow 0 \end{cases}$$

在 $K = 20, b = 0.7$ 下, 分别取 $\rho = 0.999$ 和 $\rho = 0.001$, 仿真增益与理论增益如表 1 所示.

ρ	理论增益 G	仿真增益 \hat{G}	误差
0.999	19.9564	20.0387	0.0823
0.001	13.8369	13.8940	0.0571

表 1: 复电平序列信道信噪比增益

仿真值与理论值高度吻合, 验证了复电平序列信道的信噪比增益分析.

2.4 信道估计与 ML 判决

在复电平序列信道上, 在数据符号中插入导频符号进行信道估计. 设每隔 N 个数据符号插入一个导频符号, 导频符号已知且设为 x_p . 则第 i 个导频对应的信道增益估计为:

$$h_{est,i} = \frac{y_{p,i}}{x_p} \quad (4)$$

其中 $y_{p,i}$ 为第 i 个导频符号对应的接收符号. 对于两个导频符号之间的 $N - 1$ 个数据符号, 采用线性插值进行信道增益估计. 接受方对原始接收符号乘以对应的信道增益估计值, 然后进行 ML 判决. 显然, $|x_p|$ 越大, 信道估计越准确, 但会占用更多的功率资源. 本次仿真中采用 $x_p = 2 + 0j$. 在 $b = 0.7, \rho = 0.996, \sigma_n^2 = 0.01, x_p = 2 + 0j$ 情况下对 N 取不同值进行仿真, 得到误符号率与 N 的关系如表 2 所示 (仿真 5000 个符号).

N	1	2	3	4	5	6
误符号率	0.0054	0.005	0.007	0.0094	0.0082	0.0112

表 2: 不同导频间隔下的误符号率

取最优的导频间隔 $N = 2$ 进行联合调试. 在 $b = 0.7, \rho = 0.996, \sigma_n^2 = 0.01, N = 2, x_p = 2 + 0j$ 情况下, 仿真 h_{est} 与真实 h 的实部和虚部如图 5 所示.

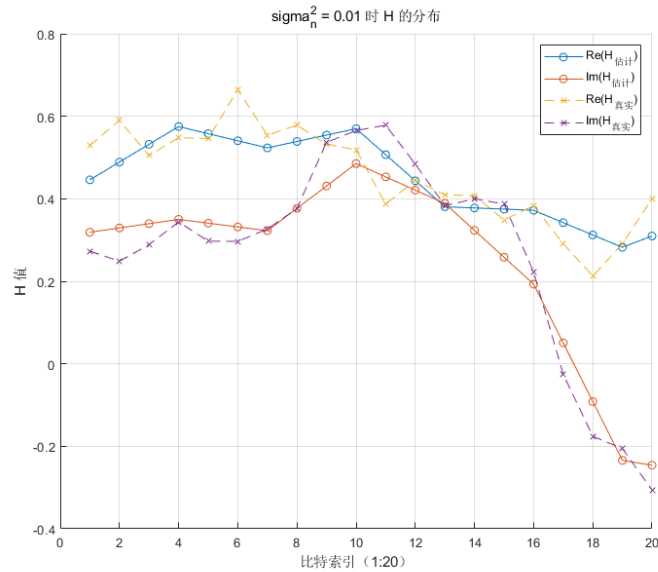


图 5: 信道估计图

从图中可以看出, 信道估计值与真实信道增益符合较好, 验证了信道估计方法的有效性.

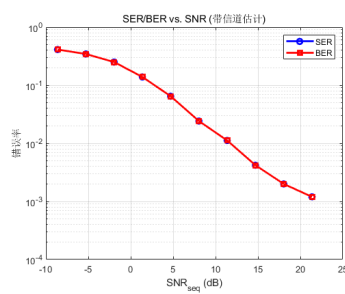
2.5 比特电平映射与信道估计联合仿真

在 $b = 0.7, \rho = 0.996, \sigma_n^2 = 0.01, N = 2, x_p = 2 + 0j$ 情况下, 对 1, 2, 3 比特/符号分别进行联合仿真, 得到误比特率如表 3 所示 (仿真 5000 个符号).

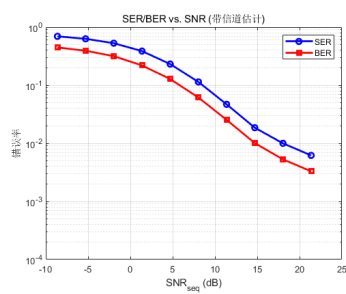
比特/符号	1	2	3
误比特率	0.0012	0.0033	0.0096

表 3: 不同比特/符号下的误比特率

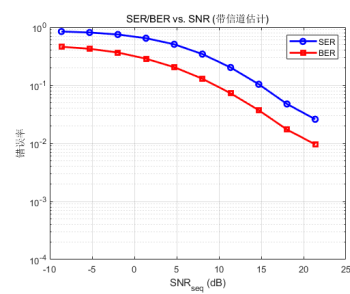
在 $b = 0.7, \rho = 0.996, N = 2, x_p = 2 + 0j$ 情况下, 对 1, 2, 3 比特/符号分别进行联合仿真, 得到误符号率、误比特率与信噪比关系如图 6 所示, 比特错误图案如图 7 所示.



(a) 1 比特/符号

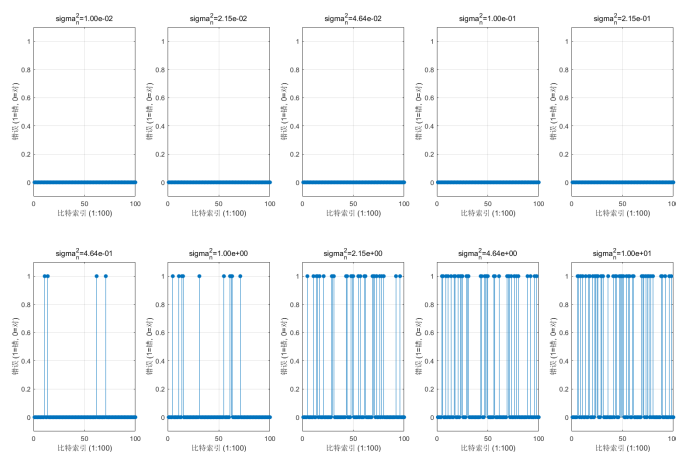


(b) 2 比特/符号

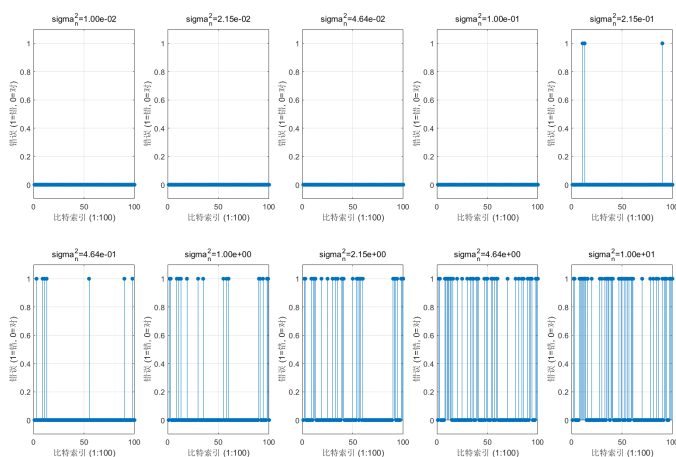


(c) 3 比特/符号

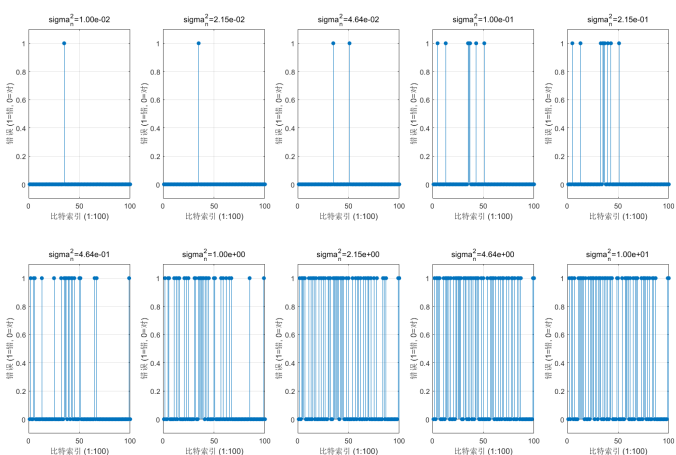
图 6: 不同映射方式下的误符号率与误比特率随信噪比变化图



(a) 1 比特/符号



(b) 2 比特/符号



(c) 3 比特/符号

图 7: 不同映射方式下的错误图案

A 代码结构

以下为压缩包中代码文件的结构说明（平台提供的文件的说明略去）：

```
.
|-- bit2sym.m                # 比特-符号映射
|-- calculate_coding_stats.m # 计算编码统计信息（用于Huffman编解码）
|-- constellation_map.m      # 构建比特-符号星座图
|-- cplxchan.m              # 复采样信道实现
|-- dct2D.m
|-- decode_huffman.m        # Huffman解码（未使用）
|-- decode_vlc1.m           # 单符号VLC解码
|-- decode_vlc2.m           # 双符号VLC解码
|-- encode_huffman.m        # Huffman编码（用于生成码供VLC编解码）
|-- encode_vlc1.m           # 单符号VLC编码
|-- encode_vlc2.m           # 双符号VLC编码
|-- idct2D.m
|-- jpeg_dequantization.m    # JPEG反量化
|-- jpeg_quantization.m     # JPEG量化
|-- lena_128_bw.bmp
|-- PSNR.m
|-- qf_test.m               # 联合调试主程序
|-- qf_test_log_double.txt   # 联合调试双符号Huffman编码结果日志
|-- qf_test_log_single.txt   # 联合调试单符号Huffman编码结果日志
|-- seqcplxchan.m           # 复电平序列信道实现
|-- sim_cplxchan.m           # 复电平信道仿真
|-- sim_err_ser_vs_snr.m     # 采用信道估计后复电平序列信道误码率仿真
|-- sim_seqcplxchan.m        # 复电平序列信道仿真
|-- sim_snr_compare.m        # 复电平序列信道和内核的复采样信道的信噪比比较仿真
|-- sym2bit.m               # 符号-比特反映射
|-- utils                   # 辅助函数目录
    |-- getmax.py            # 从联合调试日志文件中获取最大PSNR
```