



杭州电子科技大学  
HANGZHOU DIANZI UNIVERSITY

# 通信工程学院

## 信息与编码理论课程报告

题    目： 基于 3GPP 5G NR 移动通信系统中  
的极化码编码和译码算法项目

姓    名： 沈义富

学    号： 231080057

2023 年度

---

# 1 综述课题国内外研究动态,需求及现状分析,选题的依据和意义

通信系统在人们的日常生活中扮演着不可或缺的角色。随着社会的不断发展,信息的传播变得越来越便捷。智能化产品也日益普及,如智能手机、智能手表和智能机器人等已经成为我们生活中的常见存在。这些智能产品的涌现使人们对信息传输的要求不断提升<sup>[1]</sup>。

信息传输受到多个因素的影响,包括信噪比、传输时的带宽以及空间中的噪声等。为了提高传输的可靠性并减少外界因素的影响,引入纠错编码成为一种重要的手段<sup>[2]</sup>。纠错编码能够在信息接收端进行错误信息的纠正,不仅增强了传输的可靠性,还避免了因信息错误而需要重新传输的情况。

在 1993 年, Berrou 提出了 Turbo 码, 这是一种结合了并行级联卷积码和交织思想的编码方案。Turbo 码的设计使其性能接近了香农提出的信道容量  $C$ <sup>[3]</sup>。其编码方案相对易于实现, 只需将卷积码进行并行级联即可。在译码端, 可以采用 MAP 译码算法或 LOG-MAP 译码算法以实现最佳译码性能。由于 Turbo 码的计算复杂度较低, 可以使用复杂度上界形式进行约束, 并且在接近信道容量时能够实现可靠通信, 因此很快成为研究的热点<sup>[4]</sup>。此外, Turbo 码也是 3G 移动通信标准中 CDMA2000 和 WCDMA 的信道编码方案。

然而, Turbo 码的主要缺陷在于译码时的计算复杂度相对较高, 且存在较长的时间延迟, 主要是由于使用了交织器。与 Turbo 码类似, LDPC 码也是一种基于稀疏图的编码方式<sup>[5]</sup>。在码长较长时, LDPC 码能够接近香农极限的信道编码技术。在译码方面, 随着 Tanner 图的提出, 打开了新的对 LDPC 的研究领域。

在 2009 年, Arikan 提出了 Polar 码, 证明了 Polar 码是第一个具有无差错传输速率无限接近信道容量的编码方案<sup>[6]</sup>。极化码作为一种性能优越的信道编码引起了广泛关注, 并迅速成为信道编码领域的热点。极化码在信源编码、保密通信等领域都有着广泛的应用。其研究与改进主要集中在编译码算法和信道构造两个方面<sup>[7]</sup>。编译码算法, 如 SC 译码算法和 BP 译码算法以及各种改进算法, 可以显著提升信息传输的可靠性。而信道构造则通过准确选择大信道容量的信道来提升传输的正确率。目前存在多种信道构造方法, 包括蒙特卡洛构造方法、密度演化算法和极化权重算法等<sup>[8]</sup>。

为了更好地了解并学习极化码在信息传输、保密通信和数据存储等方面的知识, 加强对课程《信息论与编码》理解, 本此课程设计将侧重于对极化码的 SCL 译码算法进行仿真实现。

## 2 研究的基本内容，拟解决的主要问题

### 2.1 项目要求

极化码作为 5G 最为主要的两种信道编码之一，本项目基于 3GPP 5G NR 标准 3GPP TS 38.212，对 CRC 辅助极化（CA-Polar）编码方案进行建模。本项目基于 3GPP 5G 标准研究极化码的编解码方法，其中主要包括码构造、编码方法和解码算法以及速率匹配等。项目要求对 AWGN 上的 CA-Polar 信道编码 QPSK 调制链路进行建模，并给出编码方案中不同消息长度和码率的误块率性能结果。

### 2.2 项目内容

#### 2.2.1 3GPP 5G NR 极化码编码方式

下图 1 为 3GPP 5G NR 标准下行信道中的发送端信息处理框图。对于下行链路，在极化编码之前对输入比特进行交织。在信息比特末尾附加的 CRC 比特被分配用于 CA-Polar 方案。极性编码使用一种与信噪比无关的方法，其中每个子信道的可靠性都是离线计算的，并且有序序列被存储用于最大码长。极性码的嵌套特性允许该序列用于任何码率和所有小于最大码长的码长。

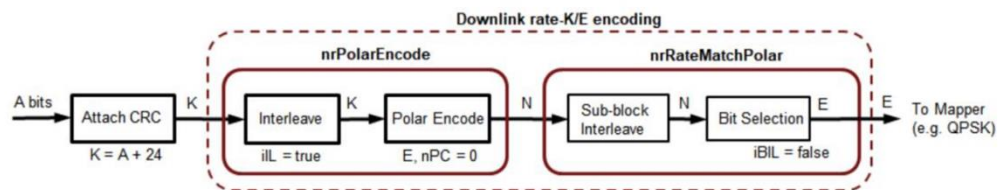


图 1 标准下行信道中的发送端信息处理框图

下图 2 为 3GPP 5G NR 标准上行信道中的发送端信息处理框图

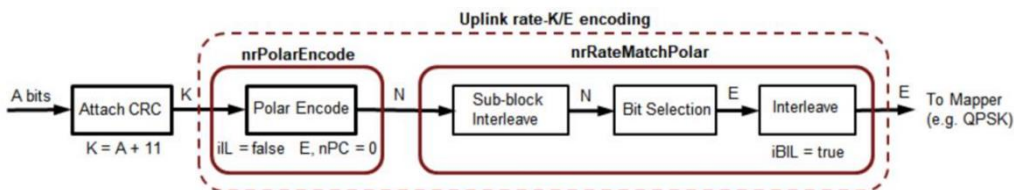


图 2 标准上行信道中的发送端信息处理框图

---

### 2.2.1 3GPP 5G NR 极化码译码算法

下行链路（DCI 或 BCH）或上行链路（UCI）消息比特的隐式 CRC 编码指示使用 CRC 辅助连续取消列表解码（CA-SCL）作为信道解码器算法。众所周知，CA-SCL 解码可以优于 turbo 或 LDPC 码，这是 3GPP 采用极化码的主要因素之一。Tal 和 Vardy 根据似然性（概率）描述了 SCL 解码算法。然而，由于定点计算的溢出效应，该译码算法计算在数值上是不稳定的。为了克服这个问题，Stimming 等人仅在对数似然比（LLR）域中提供 SCL 解码。列表解码的特征在于 L 参数，它表示保留的最可能解码路径的数量。在解码结束时，L 路径中最可能的代码路径是解码器输出。随着 L 的增加，解码器性能也会提高，但存在收益递减效应。对于加入 CRC 的输入消息，CA-SCL 解码会删除 CRC 无效的任何路径，前提是至少有一条路径具有正确的 CRC。与 SCL 解码相比，这种对最终路径加以选择的方法进一步提高了性能。对于下行链路，使用 24 位的 CRC，而对于上行链路，指定了 6 位和 11 位的 CRC，这取决于 K 值的选取。

### 3 研究步骤、方法及措施

#### 3.1 流程设计

下图 3 为基于 3GPP 5G NR 极化码编码和译码算法的仿真流程图。系统仿真主要步骤包括：生成随机信息比特、计算 CRC 并将其附加到信息快上、CRC 附加位对母码块长度进行极性编码、速率匹配、QPSK 调制、过高斯白噪声信道、对噪声信号进行软 QPSK 解调、使用 CA-SCL 算法进行极性译码、统计并输出误码率。

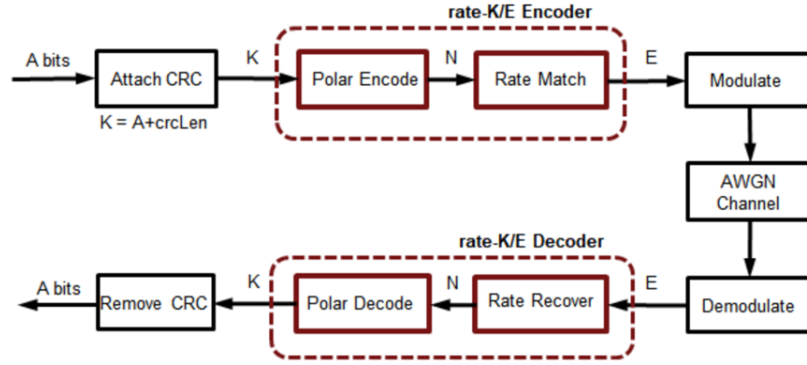


图 3 极化码编码与译码仿真流程图

#### 3.2 主要模块介绍

##### 3.2.1 极化码编码

极化变换主要包括两个步骤，分别是信道极化的分裂与联合结构。编码结构公司可表示为：

$$\mathbf{x}_1^N = \mathbf{u}_1^N \mathbf{G}_N \quad (1)$$

其中  $\mathbf{u}_1^N$  为发送序列矩阵， $\mathbf{x}_1^N$  为输出序列矩阵， $\mathbf{G}_N$  的公式为：

$$\mathbf{G}_N = \mathbf{B}_N \cdot \mathbf{F}^{\otimes n} = \mathbf{F}^{\otimes n} \cdot \mathbf{B}_N \quad (2)$$

其中， $n = \log_2^N$ ， $\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ ， $\mathbf{F}^{\otimes n}$  的公式为  $\mathbf{F}^{\otimes n} = \mathbf{F} \times \mathbf{F}^{\otimes n-1}$ ， $\mathbf{B}_N$  的含义是翻

转排序，假设原始的比特索引号变为二进制按从高到低排列为  $i-1 \rightarrow (b_1, b_2, \dots, b_n)$ ，

接着把新的索引反序排列，最后再将其变为十进制，即  $(b_1, b_2, \dots, b_n) \rightarrow j-1$ ，输出

比特  $j$  位的序号与  $i$  的序号相对应。

### 3.2.2 极化码译码

在确定编码方式后，极化码的性能直接由译码方法决定。Arikan 最初提出了极化码的基本译码算法，随后出现了更多的译码算法。目前，主流的极化码译码算法包括串行抵消译码算法（SC）和带列表的串行抵消（SCL）译码算法。

#### 3.2.2.1 SC 译码

SC 译码的本质是逐步对每个比特进行译码，利用已知信息和决策树来逼近未知比特的值。下面分别是极化码译码算法的译码示意图，其中图 4 使用传统的 SC 译码算法，图 5 使用快速 SC 译码算法。

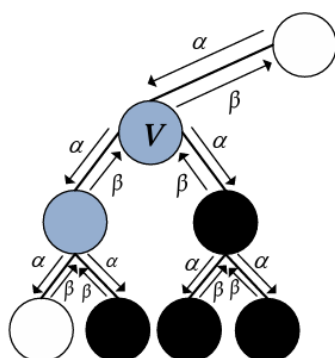


图 4 传统 SC 译码示意图

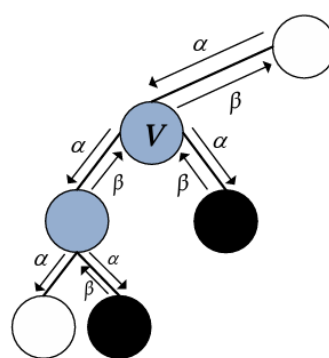


图 5 快速 SC 译码示意图

图中的  $\alpha$  和  $\beta$  分别表示下层极化节点的接收信息和发送信息，代表 SC 译码时的操作步骤。

在 SC 译码中，可以使用二叉码树表示。这棵树的每个节点都对应于对应码字的一个比特。通过对这棵树的递归遍历，可以逐步推断出每个比特的可能取值，从而实现译码。如下图 6 所示。图中各个节点的值代表该节点为 0 或者 1 的转移概率，也被称为路径度量值。第 0 层不对应具体的节点，仅用于表示整个码字的概念；第 1 层对应于极化码的第一个比特，也就是树的根节点；第 2 层对应于极化码的第二个比特，根节点的两个子节点分别代表第二个比特的两个可能取值；依次类推，直到第 N 层：每一层都对应着极化码的一个比特，形成了一棵深度为 N 的二叉树。在这个描述中，每个节点都代表了对应比特的一个可能取值。通过对树的递归遍历，可以逐步还原整个极化码的信息，推断出每个比特的值。

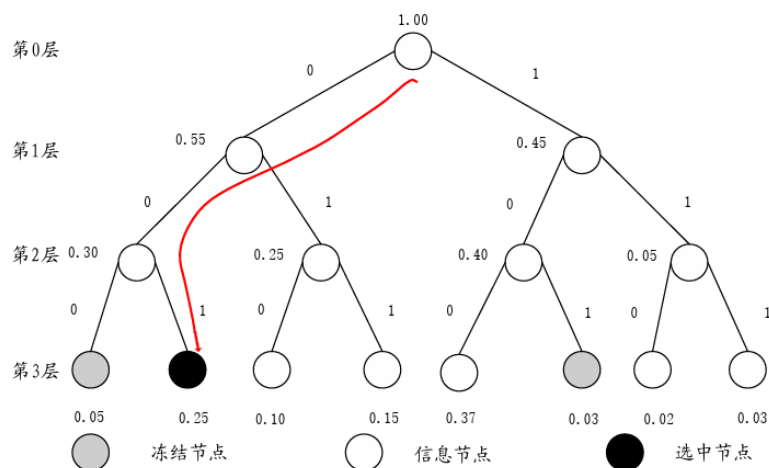


图 6 SC 译码树图

### 3.2.2.2 SCL 译码

SCL (Successive Cancellation List) 译码是一种用于极化码的高性能译码算法，它是在 SC (Successive Cancellation) 译码的基础上进行改进的。SCL 译码引入了一个列表结构，以更好地处理多个假设。SCL 算法采用广度优先的遍历原则，在每一层保留一个列表，其中包含多条路径，而不是像 SC 算法那样仅保留两个路径。这些路径具有不同的度量值，表示它们的可能性。在译码过程中，SCL 会维护这个列表，并在每一层根据路径的度量值进行排序，保留度量值最大的路径。

一旦译码到达树的底部，SCL 会对保留的多条路径进行 CRC 校验。CRC 校验是一种错误检测方法，通过对路径的比特序列进行校验，判断是否存在错误。在通过 CRC 校验的路径中，SCL 会选择度量值最大的一条路径作为最终的译码结果。这种方法允许 SCL 在译码过程中保留多个假设路径，并在最后通过 CRC 校验和度量值的比较来确定最可靠的解。这提高了极化码的译码性能，特别是在有噪声的通信信道中。SCL 译码数图如下图 7 所示。

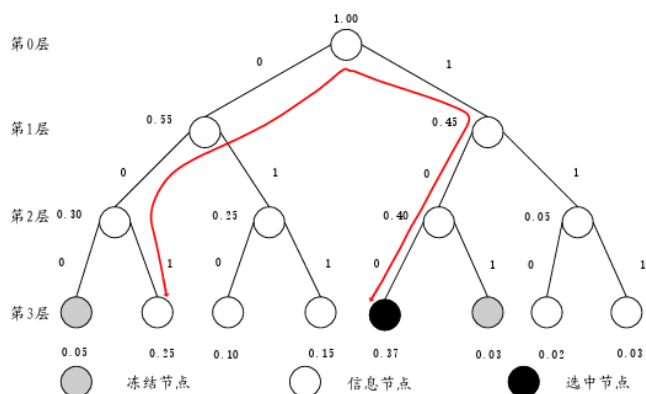


图 7 SCL 译码数图

---

### 3.2.3 速率匹配

极化码的速率匹配过程包括子块交织和比特选择两个关键步骤。在子块交织过程中，编码后的比特被均分为 32 个子块，然后按照预定的图样进行交织，最后将交织后的比特写入循环缓存中。在比特选择过程中，采用了重复、打孔和缩短三种方式。

当传输比特长度  $E$  不小于编码后比特长度  $N$  时，采用重复方式。即从循环缓存的第 1 个比特开始，依次选取  $E$  个比特，以满足传输长度的需求。

当传输比特长度  $E$  小于编码后比特长度  $N$ ，并且码率  $R$  不超过  $7/16$  时，采用打孔方式。即从循环缓存的第  $N-E+1$  个比特开始，依次选取  $E$  个比特，以满足传输长度的需求。

当传输比特长度  $E$  小于编码后比特长度  $N$ ，并且码率  $R$  超过  $7/16$  时，采用缩短方式。即从循环缓存的第 1 个比特开始，依次选取  $E$  个比特，以满足传输长度的需求。

这种设计能够根据传输比特长度和码率的不同，灵活选择合适的比特选择方式，从而提高系统的性能。

### 3.2.4 比特交织

比特交织用于上行信道速率匹配之后，该交织器其实为一个等腰直角三角形交织。按行写入等腰直角三角形交织器，再按列依次读出。



## 4 设计实施方案及结果展示

对仿真流程图中的各个模块进行编程仿真实现，仿真统计了码率相同的情况下消息码长与信噪比之间的关系示意图以及编码长度  $N$  相同的情况下码率不同与信噪比之间的关系示意图。在仿真中，分别统计了误码率和误块率，其中总块数设置为 50，编码长度  $N$ ，消息长度  $A$  以及  $K$  具体数值对应图例中的数值，CRC 长度设置为 11。

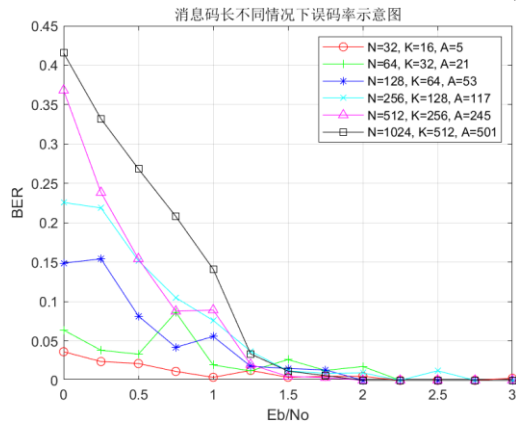


图 8 消息码长不同情况下误码率示意图

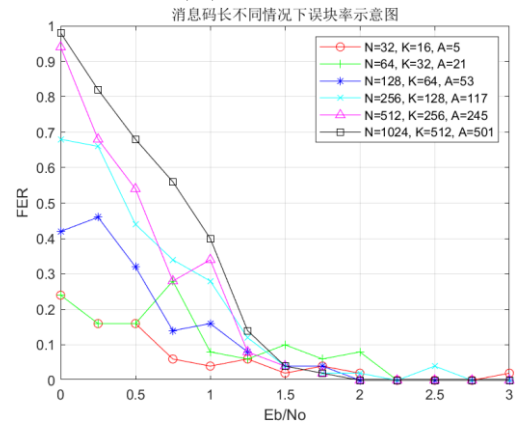


图 9 消息码长不同情况下误块率示意图

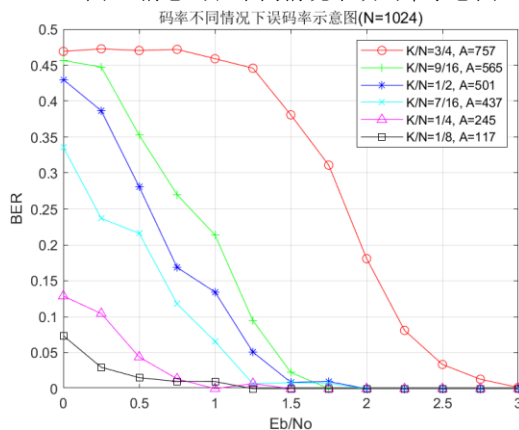


图 10 码率不同情况下误码率示意图

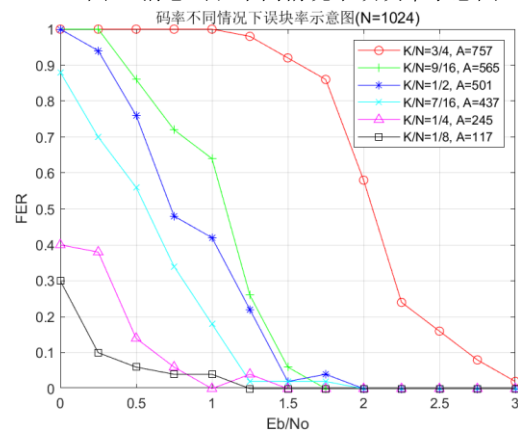


图 11 码率不同情况下误块率示意图

由图 8 和图 9 可知，当码率为  $1/2$  时，消息码长在低信噪比情况下消息码长越长，误码率和误块率越高，在信噪比大于  $1.5\text{dB}$  时，误码率和误块率普遍很低，同时消息码长越长，误码率和误块率越低。由图 10 和图 11 可知，当编码长度  $N$  固定时，码率越高，误码率和误块率越高，码率越低，误码率和误块率越低，但从译码结果上看译码效果相对较好。

---

## 5 团队任务分配

- 1. 每位任务的详细描述
- 2. 团队任务分配表

序号	姓名	学号	贡献度	签名
1	沈义富	231080057	100%	

注：贡献度填写每个成员完成的任務及百分比。

---

## 6 课程感想及团队合作情况

写到这，不经意向左下角低头瞟了一眼电脑时间，已经是一月二十日晚上九点四十分左右了，距离提交截止时间不到三个小时。好像每一次提交作业都是匆匆忙忙，但又觉得每天也没有刻意荒废时间，总是有做不完的杂事。

之前在群里问了一下，没有人应答，于是我边一人一组去弄这个课程设计。至于为什么选这道题呢，因为这道题涉及到编码，由于之前看过关于 WCDMA 的 3GPP 的协议，当时看着真是一头雾水，里面有好多份文档，涉及信道复用与编码、各种信道和指令等等，后来又被安排了其他杂事，导致对于编码这块知识就给耽搁了。自此，我总以为这块内容是一座难以攀登的高山，但我还是下决心想着将来一定找个时间好好研究以下。碰巧遇上了这个课程设计，因此果断抓住这个机会，想以作业的方式逼迫自己抽出时间再多了解了解这块知识。

课程设计真正开始时间在上周五，也就是一月十二号，也都是背着导师在偷偷做这个课设。对于我导师，他呢，总希望我们糊弄一下平时的课程，一心一意当作无脑机器给他做事，这样才是对的！有些教导还是不能全听，毕竟已经是成年人了，也该有一些主见了。

至于这个课设的进展，说来也觉得有些幸运，在开始的第二天，不经意在网上找到了关于 polar 码的教学视频，印度的一个网上公开课网站 NPTEL，同时也获取了教学视频对应的课件板书以及相关代码，于是在上周六周日两天就抓紧把教学课件和视频看了看，对 polar 码有了大致的了解。因此，在了解的基础上赶快实操，不然看多了容易懵，于是利用课程中提供的基于 SC 的编码和译码相关代码，整理出了我的第一版代码，对应压缩文件中的 polar\_self\_v1.m 文件，这版代码中，在调制模块部分我添加了成型、采样率变换、上变频、以及载波同步和码元同步的代码，通过最终运行对比前后译码结果，发现误码率为 0，暂时可行。

于是开始在万方、知网上下载关于 polar 码的相关论文，大致看了几篇之后对这个编码有了进一步了解，尤其是看论文的引言部分，使我脑中对这块知识有了大致的轮廓。

于是才意识到 polar\_self\_v1.m 文件使用的 SC 译码算法，于是接着搜索资料，查找 SCL 的译码算法及实现。就这样，这周三，便有了我的第二份代码 polar\_self\_v2.m 文件，这版代码中使用了 SCL 译码算法。老实说，其中 SCL 译码算法代码也是从网上找到的，暂时还没完全理解，就拿来使用了。其中路径条数 L 设置为 4，N=1024，A=500，CRC 长度取 11，通过最终运行对比前后译码结果，发现误码率为 0，暂时可行。

接下来便是速率匹配这块代码，这块部分简直就是我的噩梦，因为之前看关于 WCDMA 的 3GPP 协议，里面的信道复用与编码老复杂了，根本了看不懂。于是后来

---

继续看网上的论文，期间又被安排了各种杂事，磕磕绊绊，最后在这两天才出了我的第三份代码 `polar_self_v3.m`。里面添加了速率匹配和交织模块代码。能写出这两块代码主要靠看了论文徐俊前辈的论文，再去和协议 3GPP TS 38.212 中 5.4.1 节对照理解。不过，考虑时间问题以及复杂程度，这份代码中速率匹配我都简化了步骤，比如固定速率匹配后长度  $E$  大于等于编码长度  $N$ ，然后使用重复操作去进行速率匹配。通过最终运行对比前后译码结果，发现误码率为 0，于是改变码长，发现误码率还是 0。这就不太对了，理论误码率也不可能一直为 0 啊，肯定有问题。

于是，便有了我的第四份代码 `polar_self_v4.m`。尽管第三份代码我还没找到原因，但为了课设进行，我再第四份代码中删除了采样率变换以及相关解调模块代码。最后运行，得出的误码率和误块率示意图不至于太离谱。示意图中的数据我已经保存成 `.mat` 文件，`BER_sim_1024_Rarechange.mat`、`FER_sim_1024_Rarechange.mat` 和 `BER_sim_Nchange_half.mat`、`FER_sim_Nchange_half.mat`。

由于时间准备不充分，这次课设还存在很多问题。比如 QPSK 软解调，这块我还不是很明白，还有给 QPSK 信号加高斯白噪声，利用  $E_b/N_0$ ，我就不太明白了，从我绘制的误码率示意图和论文中比较，感觉我预设的 1.5dB，但实际添加好像不是 1.5dB，具体代码我编写得感觉有些问题，我不知道会不会存在这个问题；还有就是速率匹配没有写完全，各种情况都应该考虑一下，同时再复杂一点，各种信道或许也可以考虑去仿真一下，不过这确实很难，非常难，毕竟协议那么多那么复杂，以我的水平要想实现肯定得付出巨大的时间和精力。有机会去问问相关老师，多了解一点还是好的。

不过今天就先到这，之后也可以再改进。对于这次课程设计，我还是可以对自己说：“我确实有所得、有所获，值了”。

---

## 7. 主要参考文献（所列出的参考文献不得少于 8 篇，发表在期刊上的学术论文不得少于 4 篇。）

- [1] 秦康剑. 极化码在 5G 应用场景下的编译码算法研究 [D]. 浙江大学, 2020. DOI:10.27461/d.cnki.gzjdx.2020.001185
- [2] Arikan, E., "Channel Polarization: A Method for constructing Capacity Achieving Codes for Symmetric Binary-Input Memoryless Channels," IEEE Transactions on Information Theory, vol. 55, No. 7, pp. 3051-3073, July 2009.
- [3] Tal, I, and Vardy, A., "List decoding of Polar Codes", IEEE Transactions on Information Theory, vol. 61, No. 5, pp. 2213-2226, May 2015.
- [4] Niu, K., and Chen, K., "CRC-Aided Decoding of Polar Codes," IEEE Communications Letters, vol. 16, No. 10, pp. 1668-1671, Oct. 2012.
- [5] Niu, K., Chen, K., and Lin, J.R., "Beyond turbo codes: rate compatible punctured polar codes", IEEE International Conference on Communications, pp. 3423-3427, 2013.
- [6] Stimming, A. B., Parizi, M. B., and Burg, A., "LLR-Based Successive Cancellation List Decoding of Polar Codes", IEEE Transaction on Signal Processing, vol. 63, No. 19, pp. 5165-5179, 2015.
- [7] 杜铮. 5G NR 物理层上行控制信道的研究及实现 [D]. 重庆邮电大学, 2021. DOI:10.27675/d.cnki.gcydx.2021.000795
- [8] 徐俊, 彭佛才, 许进. 5G NR 信道编码研究 [J]. 邮电设计技术, 2019, (03): 16-21.

---

## 八、附件

- 1.程序使用说明;
- 2.程序文档（有条理，注释清晰）;
- 3.其他需要说明的问题。

**说明：另要求递交程序文件夹**