

现代通信原理实验总结

题	目：	基于软件无线电平台的 5G 物理层通信系统设计
课	程	名称：现代通信原理
学	院：	信息科学技术学院
专	业：	电子信息工程
学	号：	PB22061345
学	生	姓名：浮焕然

目 录

一、综合设计目的	1
二、综合设计内容	1
三、基础算法部分	1
1、CRC 添加.....	1
2、速率匹配.....	3
3、调制映射	5
4、OFDM 调制	7
5、解调制映射.....	8
6、解扰.....	10
7、帧同步.....	10
8、信道估计与均衡.....	11
四、综合设计部分	12
1、系统总体组成与描述.....	12
2、系统各组成模块功能分析.....	13
3、系统运行结果分析.....	14
五、收获与体会	29
1.收获.....	29
2.体会.....	30
六、学生代码	30
1.CRC 添加.....	30
2.速率匹配.....	31
3.调制映射	32
4.OFDM 调制	33
5.解调制映射.....	33
6.解扰.....	36
7.帧同步.....	37
8.信道估计与均衡.....	38

一、综合设计目的

- (1) 了解通信领域前沿技术
- (2) 培养学生模块化+系统化思想搭建通信系统能力
- (3) 掌握 5G 协议算法实现原理及实现方法

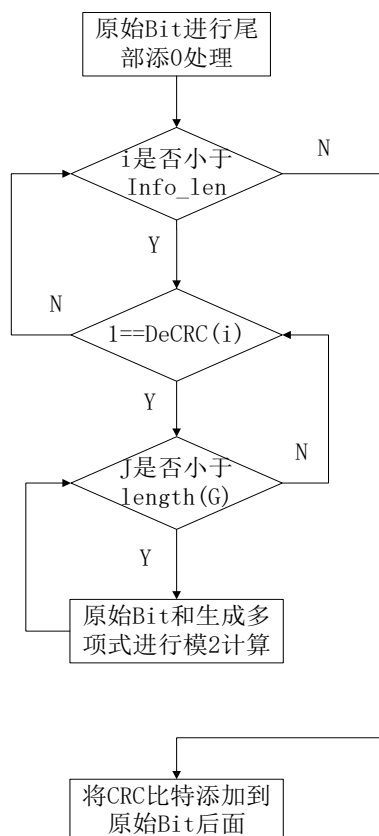
二、综合设计内容

1. 5G 物理层通信系统包括生成数据源、CRC 添加、LDPC 编码、速率匹配、交织、加扰、调制映射、预编码、产生导频数据、资源映射、OFDM 调制，最后将产生的数据通过以太网发送到 XSRP 软件无线电平台，在软件无线电平台中完成 IQ 数据 DA 转换、上变频载波调制、射频在指定频点将信号通过天线发射出去。无线比特经过空中无线信道，再通过射频的接收天线在对应的频点将数据接收、下变频、低通滤波、AD 转换得到 IQ 信号，通过以太网发送到电脑。在电脑上进行 OFDM 解调、解资源映射、信道估计与均衡、解预编码、解调制映射、解扰、解交织、解速率匹配、LDPC 译码、CRC 校验、信宿。
2. 和其他已经提供的功能模块合到一起，搭建完整通信系统。
3. 用 XSRP 软件无线电平台对系统进行软硬件联调，优化系统。

三、基础算法部分

1、CRC 添加

1.1、原理



CRC 校验码的作用是：发送方发送的数据在传输过程中受到了信号干扰，可能出现错误的码，造成的结果就是接收方不清楚收到的数据是否正确，所以就有了 CRC 校验码，CRC 是数据通信领域中最常用的一种差错校验码。

CRC 校验利用线性编码理论，在发送端根据要传送的 k 位二进制码序列，以一定的规则产生一个校验用的监督码（即 CRC 码） r 位，并附在信息后面，构成一个新的二进制码序列数共 $k+r$ 位，最后发送出去。在接收端，则根据信息码和 CRC 码之间所遵循的规则进行检验，以确定传送中是否出错。

设编码前的原始信息多项式为 $P(x)$ ，生成多项式为 $G(x)$ ，CRC 多项式为 $R(x)$ ；编码后带循环校验码 CRC 的信息多项式为 $T(x)$ 。其实现步骤如下：

a. 设待发送的数据块是 k 位的二进制多项式 $P(x)$ ，生成多项式为 r 阶的 $G(x)$ 。在数据块的末尾添加 r 个 0，数据块的长度增加到 $k+r$ 位，对应的二进制多项式为 。

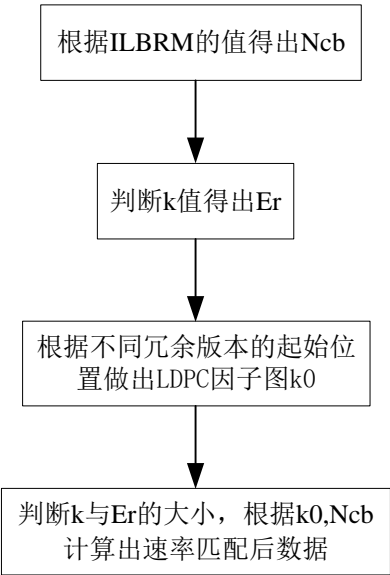
b. 用生成多项式 $G(x)$ 去模 2 除 。求得余数为 $r-1$ 阶的二进制多项式 $R(x)$ 。此二进制多项式 $R(x)$ 就是 $P(x)$ 经生成多项式 $G(x)$ 编码的 CRC 校验码。

将校验码 $R(x)$ 添至 $P(x)$ 的末尾，即可得到包含 CRC 校验码的待发送字符串。

CRC 校验的算法依据是 3GPP 36.212 协议中的 5.1.1 节中的 CRC 计算部分，根据不同的信道使用的 CRC 计算方式是不一样的，例如 LTE 的 PUSCH 信道是采用 CRC24A

的方式进行加 CRC 的，本实验也采用 CRC24A 的方式添加 CRC。

2、速率匹配



2.1、原理



速率匹配是根据信道编码后的不同码流长度做不同的处理，从而使得码流长度与实际传输能力相匹配，速率匹配的方案是与编码的方式强相关的。

根据 3GPP TS 38.212 协议 5.4.2

5.4.2.1 Bit selection

The bit sequence after encoding $d_0, d_1, d_2, \dots, d_{N-1}$ from Section 5.3.2 is written into a circular buffer of length N for the r -th coded block, where N is defined in Section 5.3.2.

For the r -th code block, let $N_{cb} = N$ if $I_{LBRM} = 0$ and $N_{cb} = \min(N, N_{ref})$ otherwise, where $N_{ref} = \left\lfloor \frac{TBS_{LBRM}}{C \cdot R_{LBRM}} \right\rfloor$,

$R_{LBRM} = 2/3$, TBS_{LBRM} is determined according to section X.X in [6, TS38.214] assuming the following:

- maximum number of layers supported by the UE for the serving cell;
- maximum modulation order configured for the serving cell;
- maximum coding rate of 948/1024;
- $\bar{N}_{RE} = 156$;
- $n_{PRB} = n_{PRB,LBRM}$ is given by Table 5.4.2.1-1;

3GPP

Release 15

27

3GPP TS 38.212 V1.2.1 (2017)

- C is the number of code blocks of the transport block determined according to Section 5.2.2.

rv_{id}	k_0	
	Base graph 1	Base graph 2
0	0	0
1	$\left\lfloor \frac{17N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{13N_{cb}}{50Z_c} \right\rfloor Z_c$
2	$\left\lfloor \frac{33N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{25N_{cb}}{50Z_c} \right\rfloor Z_c$
3	$\left\lfloor \frac{56N_{cb}}{66Z_c} \right\rfloor Z_c$	$\left\lfloor \frac{43N_{cb}}{50Z_c} \right\rfloor Z_c$

确定 Ncb 和 k0 后的算法伪代码如下


```

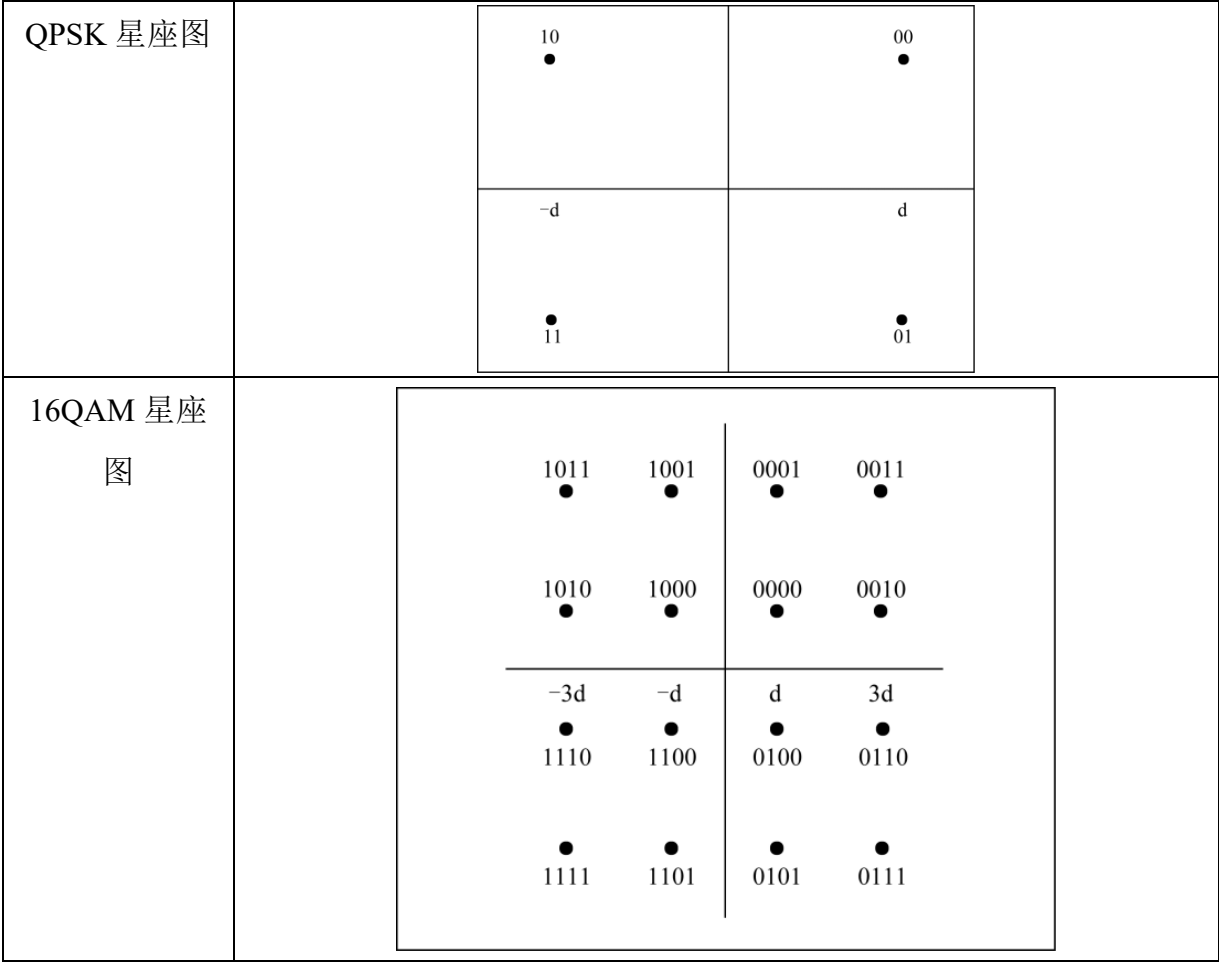
k = 0;
j = 0;
while k < E
    if d(k0+j)modNrs ≠ < NULL >
        ek = d(k0+j)modNrs;
        k = k + 1;
    end if
    j = j + 1;
end while

```

3、调制映射

3.1 原理：

对加扰后的比特进行调制，产生复值调制符号。采用的调制方式有 QPSK、16QAM、64QAM 和 256QAM。这几种调制方式的星座图下图所示：



64QAM 星座图	
256QAM 星座图	

为了保证统计上平均每个符号的能量为 1，需要进行符号的能量归一化。
 四种调制方式的归一化因子如下：

QPSK 星座图	$1 / \sqrt{2}$
16QAM 星座图	$1 / \sqrt{10}$
64QAM 星座图	$1 / \sqrt{42}$
256QAM 星座图	$1 / \sqrt{170}$

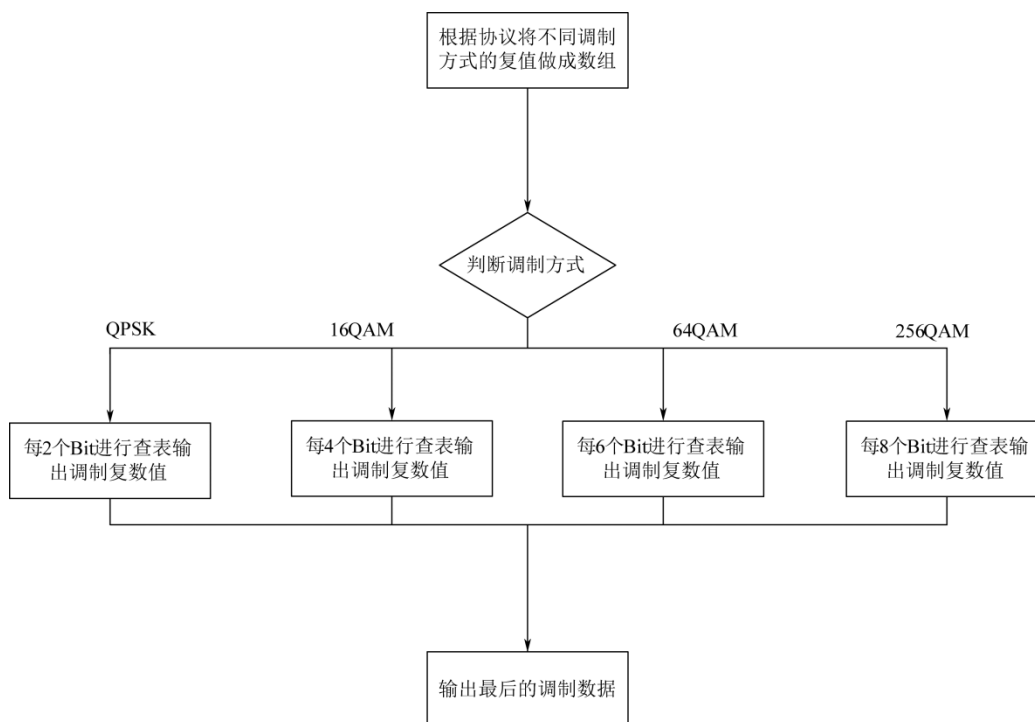
QPSK 的调制如下

$b(i), b(i+1)$	I	Q
00	$1/\sqrt{2}$	$1/\sqrt{2}$
01	$1/\sqrt{2}$	$-1/\sqrt{2}$
10	$-1/\sqrt{2}$	$1/\sqrt{2}$

11	$-1/\sqrt{2}$	$-1/\sqrt{2}$
----	---------------	---------------

其他的调制方法省略

3.2 代码：



以 256QAM 的调制算法为例：

$$x = \frac{1}{\sqrt{170}} \{ (1-2b(i)) [8 - (1-2b(i+2)) [4 - (1-2b(i+4)) [2 - (1-2b(i+6))]]] \\ + j(1-2b(i+1)) [8 - (1-2b(i+3)) [4 - (1-2b(i+5)) [2 - (1-2b(i+7))]]] \}$$

4、OFDM 调制

原理：

将频域数据根据符号个数分别进行 IFFT 变换，组成 OFDM 符号，并加入循环前缀。

正交频分复用(OFDM: Orthogonal Frequency Division Multiplexing)是多载波调制方式(MCM:Multiple Carriers Modulation)的一种。在 OFDM 系统中，各个子载波在时域相互正交，它们的频谱相互重叠，因而具有较高的频谱利用率。OFDM 变换(时域到频域的转换)：这是各个独立的频域子载波信号到时域信号的转换。OFDM 技术一般应用在无线系统的数据传输中，由于无线信道的多径效应，从而使符号间产生干扰。为了消除符号间干扰(ISI)，应该在符号间插入保护间隔。插入保护间隔的一般方法是符号间置零，即发送第一个符号后停留一段时间(不发送任何信息)，接下来再发送第二个符号。在 OFDM 系统中，这样虽然减弱或消除了符号间干扰，由于破坏了子载波间的正交性，从而导致

了子载波之间的干扰(ICI)。因此，这种方法在 OFDM 系统中不能采用。在 OFDM 系统中，为了既可以消除 ISI，又可以消除 ICI，通常保护间隔是由循环前缀来充当。当循环前缀的长度大于或等于信道冲击响应长度时，可以有效地消除 ISI 和 ICI。

OFDM 符号:一个时隙内的 OFDM 符号的数量是固定的，有 14 个 OFDM 符号（常规 CP)具体每个符号 CP 的点数如下所示：

配置	循环前缀长度
常规循环前缀	160 for l=0 144 for l=1, 2, ..., 6

代码：将频域数据根据符号个数分别进行 IFFT 变换，组成 OFDM 符号，并加入循环前缀。

5、解调制映射

原理：

解调是调制的逆变化，是指从已调信号中恢复消息的过程。5G 的采用的调制方式有 QPSK、16QAM、64QAM 和 256QAM。解调分为硬解调和软解调。硬解调是通过设置阈值来判断输出，以二进制来说，一般大于 0 判为 0，小于 0 判决为 1，硬解调输出的是 0 或者 1。软解调的输出不是一个具体的比特，而是一个数值，QAM 解调通常使用简化的双最小距离量度方法和简单度量算法。

以 256QAM 为例说明解调制映射的原理乳如下：

设第 k 个解调输入信号 I 路数据为 X_k ，Q 路数据为 Y_k ，设解调输出为： $\Lambda(S_{k,0})$ ， $\Lambda(S_{k,1})$ ， $\Lambda(S_{k,2})$ ， $\Lambda(S_{k,3})$ ， $\Lambda(S_{k,4})$ ， $\Lambda(S_{k,5})$ ， $\Lambda(S_{k,6})$ ， $\Lambda(S_{k,7})$

$$\Lambda(S_{k,0}) = X_k$$

$$\Lambda(S_{k,1}) = Y_k$$

$$\Lambda(S_{k,2}) = 8A - |X_k|$$

$$\Lambda(S_{k,3}) = 8A - |Y_k|$$

$$\Lambda(S_{k,4}) = 4A - |\Lambda(S_{k,2})|$$

$$\Lambda(S_{k,5}) = 4A - |\Lambda(S_{k,2})|$$

$$\Lambda(S_{k,6}) = 2A - |\Lambda(S_{k,4})|$$

$$\Lambda(S_{k,7}) = 2A - |\Lambda(S_{k,5})|$$

接收到信号之后，要根据发送端的调制方式讲信号解调出来，具体的对应方式如下表所示

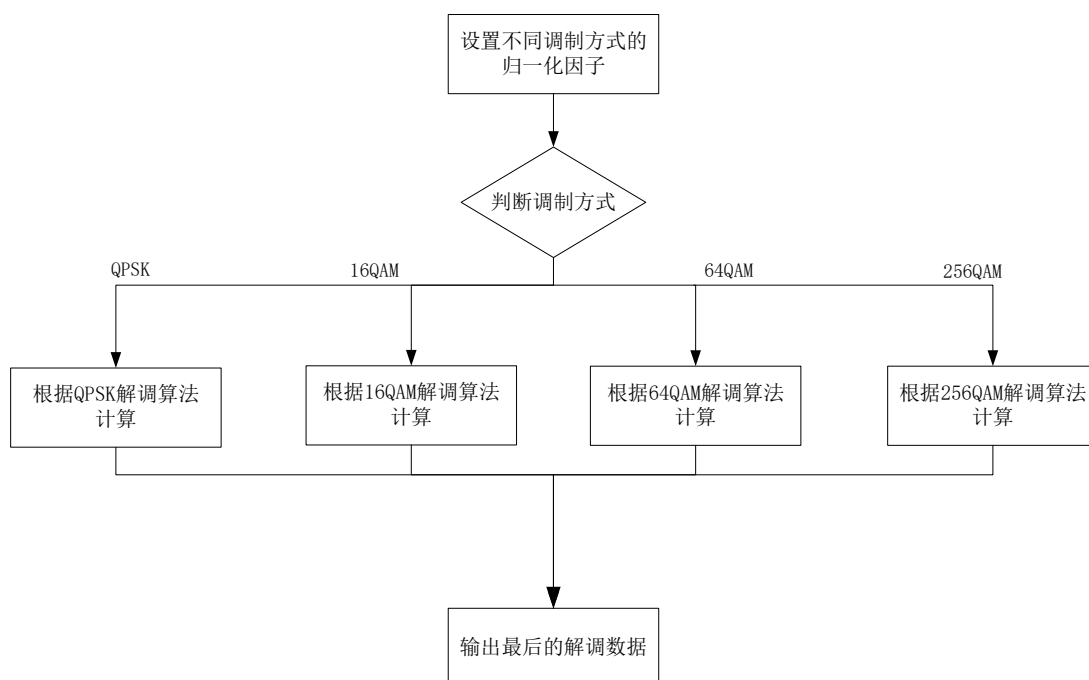
Qm	调制方式
2	QPSK
4	16QAM
6	64QAM
8	256QAM

过程：

解调是将复值信号恢复比特流。本项目中使用硬解调方式，通过设置阈值来判断输出，即大于 0 判为 0，小于 0 判决为 1。

解调流程：

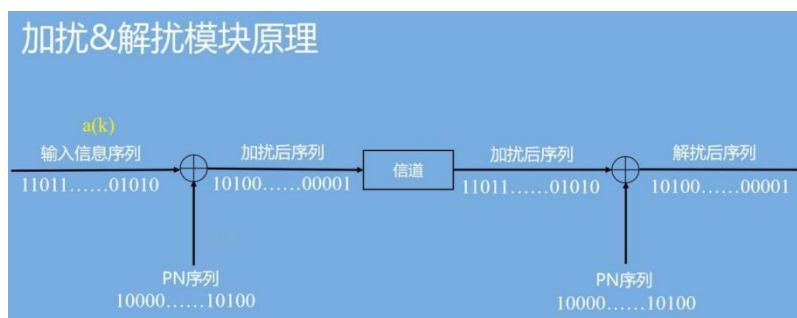
- 1)根据 qm 值大小判断调制方式。
- 2)根据不同的调制方式选择相应的解调算法，将复值信号恢复成 Bit 流。
- 3)对每个 bit 位映射进行判断：大于 0 判 0，小于 0 判 1。



6、解扰

原理：

加扰/解扰的步骤按位异或：加扰过程涉及输入数据流（由码块级联步骤产生）和加扰序列之间的按位异或 (XOR) 运算。这个操作的输出是一个加扰的数据流。由于这个过程是可逆的，所以根据规定生成 PN 序列之后，对接收到的数据进行异或 xor 运算，即可得到解扰后的数据。

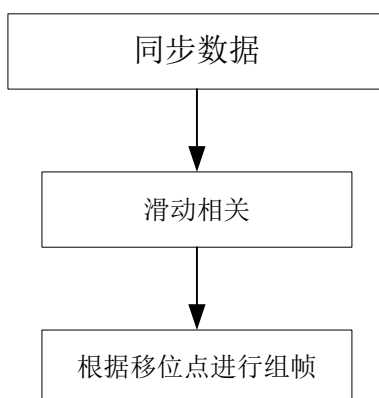


加扰的步骤主要是按位异或：加扰过程涉及输入数据流（由码块级联步骤产生）和加扰序列之间的按位异或 (XOR) 运算。这个操作的输出是一个加扰的数据流。

$$\tilde{b}(i) = b(i) \oplus c(i) = (b(i) + c(i)) \bmod 2, i = 0, 1, \dots, M_{bit} - 1$$

其中 $b(i)$ 表示码块级联后数据， $\tilde{b}(i)$ 表示加扰后数据。

7、帧同步



原理：

PSS 使用长度为 127 的 BPSK 调制的 m 序列，PSS 仅与小区组内 ID 有关，有较好的

自相关特性和互相关特性。

得到 PSS 信号的方法与得到导频数据的方法一致，所以经过 IFFT 变换之后的同步数据 PSS 应该与接收信号中的第一个导频数据部分相关性最好。由于导频符号的位置 $\text{Frist_DMRS_L0}+0+1=4$ ，它应该在 $160+2048+144+2048+144+2048+144+2048+1=6737$ 的位置上。

如果接收信号发生偏移，我们也可以根据滑动相关模块计算的位置进行左移或右移来调整。

8、信道估计与均衡

导频数据的产生：

根据所给的参数按照协议生成导频数据

Table 6.4.1.1.3-4: PUSCH DM-RS positions \bar{l} for double-symbol DM-RS.

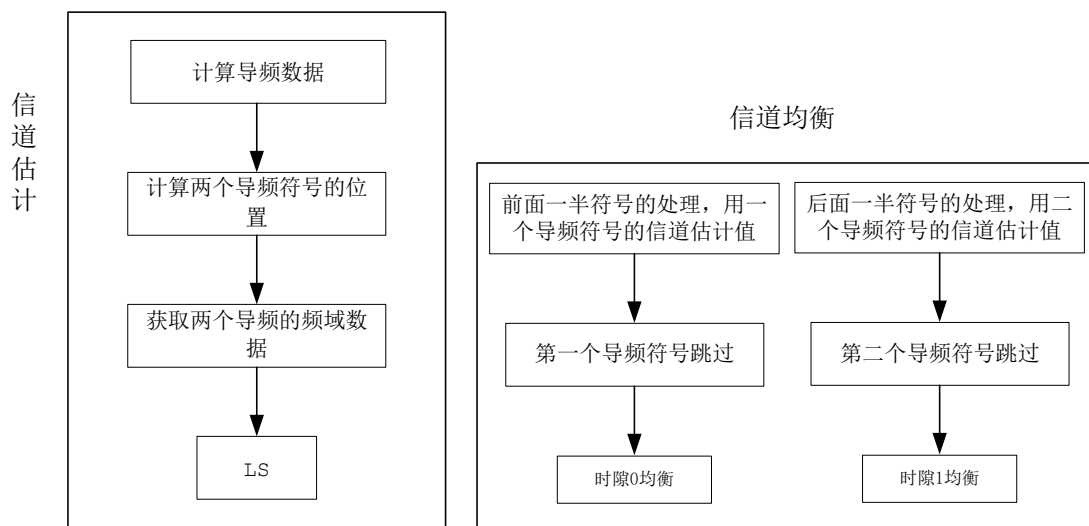
PUSCH duration in symbols	DM-RS positions \bar{l}							
	PUSCH mapping type A				PUSCH mapping type B			
	DL-DMRS-add-pos				DL-DMRS-add-pos			
	0	1	2	3	0	1	2	3
▪ ≤ 7	l_0	-			0	-		
▪ 8	l_0	-			0	0, 5		
▪ 9	l_0	-			0	0, 5		
▪ 10	l_0	$l_0, 8$			0	0, 7		
▪ 11	l_0	$l_0, 8$			0	0, 7		
▪ 12	l_0	$l_0, 8$			0	0, 9		
▪ 13	l_0	$l_0, 10$			0			
▪ 14	l_0	$l_0, 10$			-	-		

DM-RS duration	l'	Supported antenna ports p	
		Configuration type 1	Configuration type 2
▪ single-symbol DM-RS	0	1000 – 1003	1000 – 1005
▪ double-symbol DM-RS	0, 1	1000 – 1007	1000 – 1011

信道估计与均衡：

信道估计与均衡，就是利用导频做信道估计和均衡，接收的导频和发送的导频相除得到估计值，再利用估计值进行均衡，接收的数据除以估计值得到均衡值。

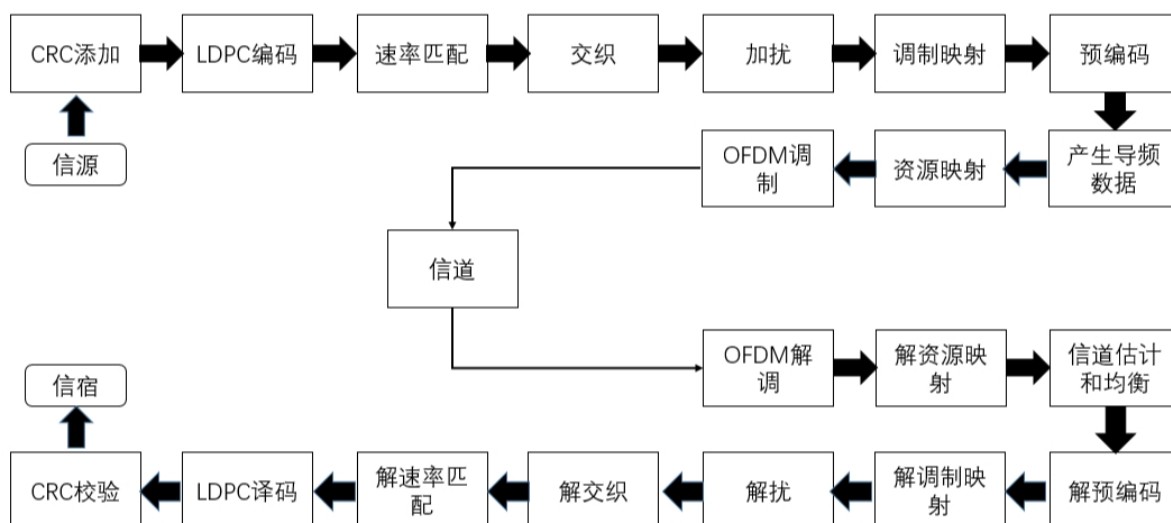
算法流程：



四、综合设计部分

1、系统总体组成与描述

总体流程图如下：



描述：5G 物理层协议算法是按照 3GPP 38.211、38.212 、38.213 协议的定义，完成了 PUSCH 信道的实现过程。实现了 UE 上行发数过程和基站的上行处理过程，实现了一个完整的 UE 发数，基站侧解数的整体链路。

2、系统各组成模块功能分析

主要流程	子模块	模块功能
UE 上行发数	信源	根据配置参数产生随机信源数据，也可以导入数据
	CRC 添加	包含 CRC 添加、码块分割 CRC 添加：对传输块进行 CRC24A 计算，并将计算得到的 24 比特校验码添加在传输块后面 码块分割：根据 LDPC_base_graph 来对传输块进行码块分割，并在每个码块做 CRC24B 校验码添加
	LDPC 编码	对每个码块进行 LDPC 编码
	速率匹配	对每个 LDPC 编码后的码块数据进行速率匹配
	交织	将速率匹配后的数据按照调制符号一组进行交织
	加扰	包含码块级联、加扰 码块级联：将速率匹配后的所有码块拼接起来 加扰：计算伪随机加扰序列，对输入比特数据进行按位加计算
	调制映射	根据配置的调制方案对加扰后数据进行调制
	预编码	包含 层映射、传输预编码、预编码 层映射：根据映射表将调制后的复值映射到各个层 传输预编码：根据传输预编码指示将每个符号进行传输预编码处理 预编码：将传输预编码的数据根据 PMI 索引进行预编码
	产生导频数据	根据所给的参数按照协议生成导频数据
	资源映射	将预编码数据和导频数据映射到每个子载波的资源位置
	OFDM 调制	将频域数据根据符号个数分别进行 IFFT 变换，并添加

		循环前缀组成 OFDM 符号
基站侧 解数	OFDM 解调	以符号为单位，先除掉 CP，然后进行 FFT 变换，将时域数据变换成频域数据
	解资源映射	从频域数据中提取有用子载波数据和导频数据
	信道估计与均衡	根据导频数据估计信道的矩阵，用信道估计的值来补偿收到的有用子载波数据
	解预编码	包含 解预编码、解传输预编码、解层映射 解预编码：根据 PMI 索引对均衡后的数据进行解预编码 解传输预编码：根据传输预编码指示将每个符号进行解传输预编码处理 解层映射：根据映射表将多层的数据进行解层映射
	解调制映射	根据配置的调制方案将复值信号恢复成 Bit 流
	解扰	包含 解扰、解码块级联 解扰：加扰的逆运算，将加扰后数据还原为加扰前的码字数据 解码块级联：根据协议计算的码块长度将解扰后的数据分成多个码块
	解交织	将解码块级联的数据按照调制符号为一组进行解交织
	解速率匹配	速率匹配的逆运算，还原出解 LDPC 编码的数据
	LDPC 译码	LDPC 编码的逆运算，得到编码前的 Bit 数
	CRC 校验	对传输块进行 CRC 校验
	信宿	将还原信宿比特，写入图片

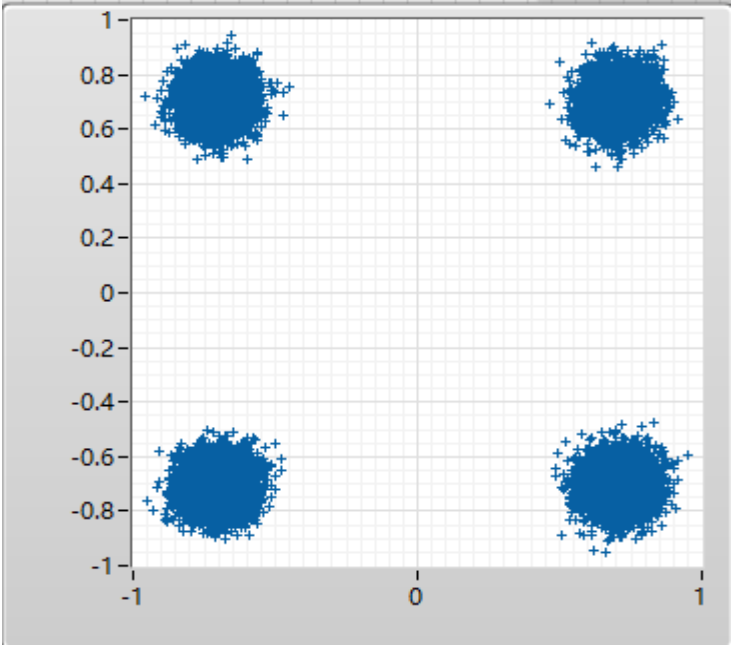
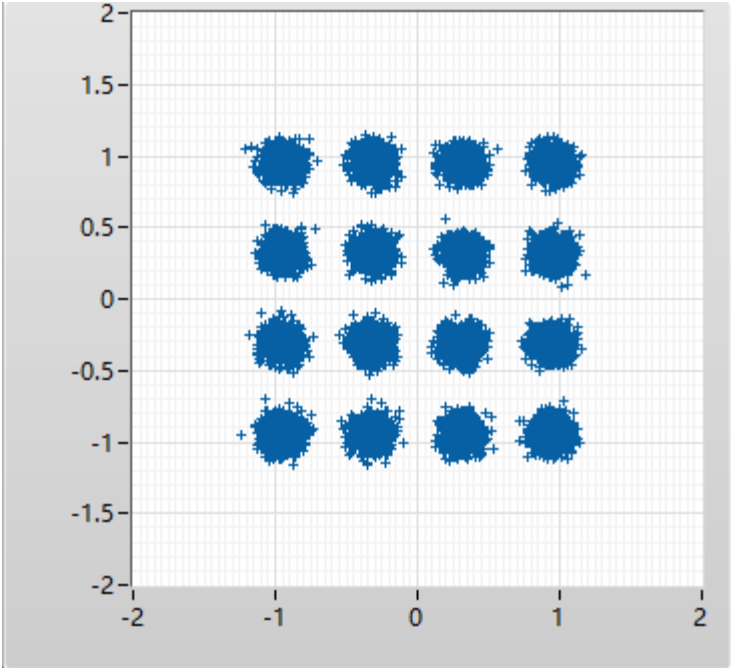
3、系统运行结果分析

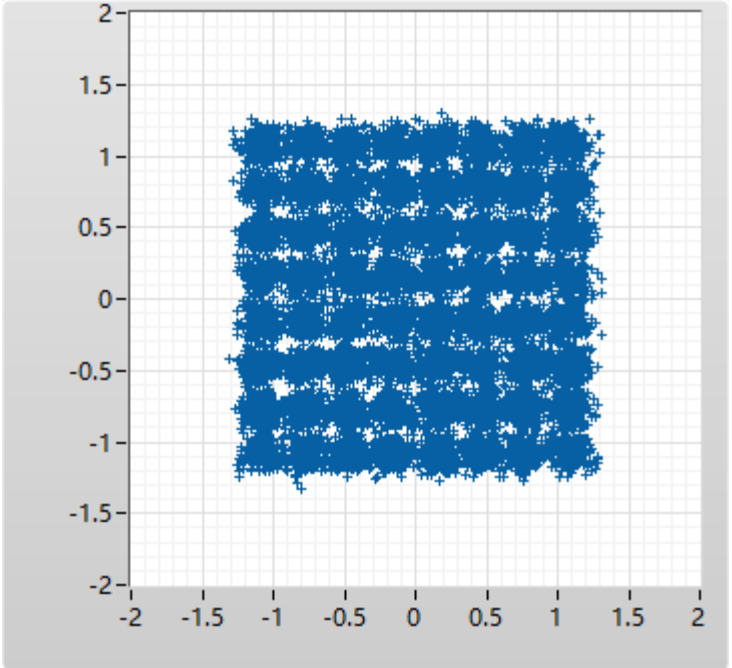
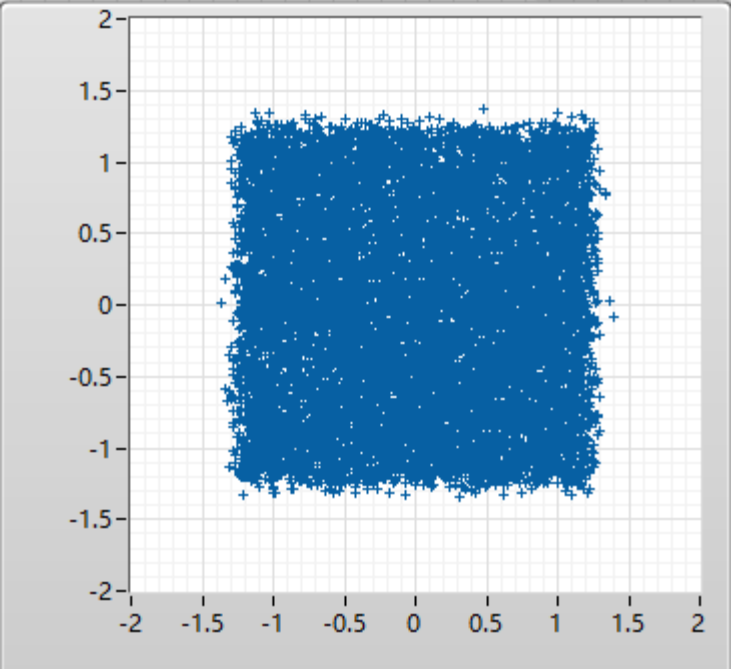
3.1 仿真结果分析

- a. 同一信噪比下不同调制方式下基于软件无线电平台的 5G 物理层通信系统设计实验结果

配置：更改调制类型分别为 256QAM 、64QAM、16QAM、QPSK，打开添加噪声开关，SNR 设置为 14，其余参数为默认参数

测试结果如下：

调制方式	总比特数	接收端星座图
QPSK	16896	
16QAM	21000	

64QAM	38936	
256QAM	75792	

结论：随着调制方式越来越复杂，接收端口接受的比特数越来越多，说明传输时一帧能携带的信息越来越多。

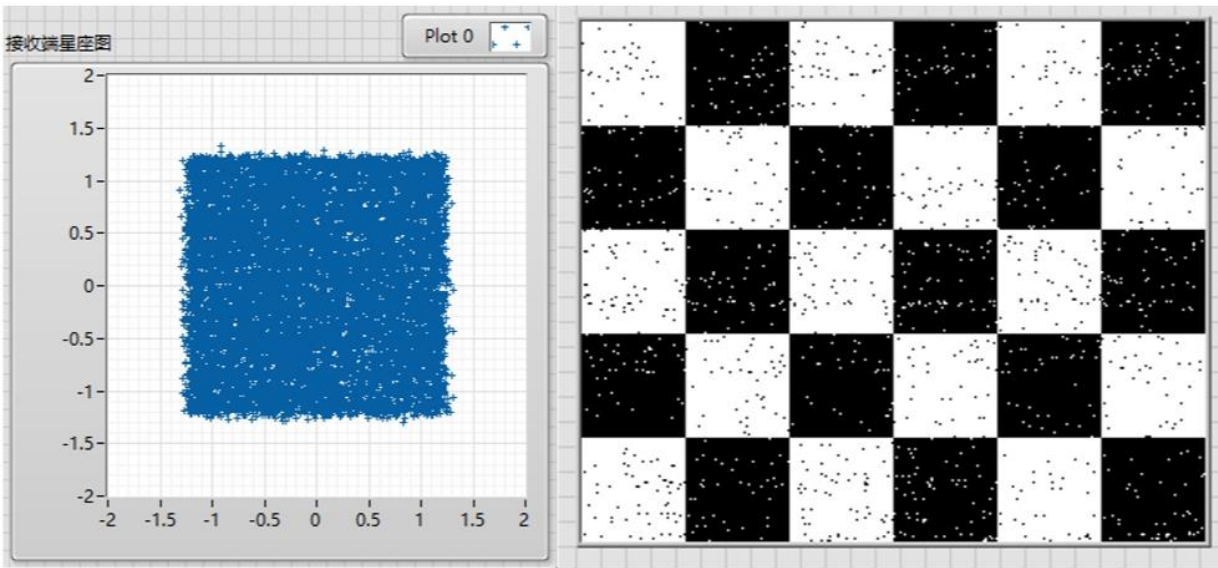
b.LDPC 模块作用验证

配置：添加噪声信噪比 17，配置 LDPC 开关开启及关闭，调制类型为 256QAM，其余参数为默认参数。

当 LDPC 打开时，接收端星座图正常，接收信号误码率为 0，CRC 检验通过。

当 LDPC 关闭时，接收端星座图有干扰，误码率 0.0176008，CRC 检验不通过，接收

图片有噪声点，如下图所示

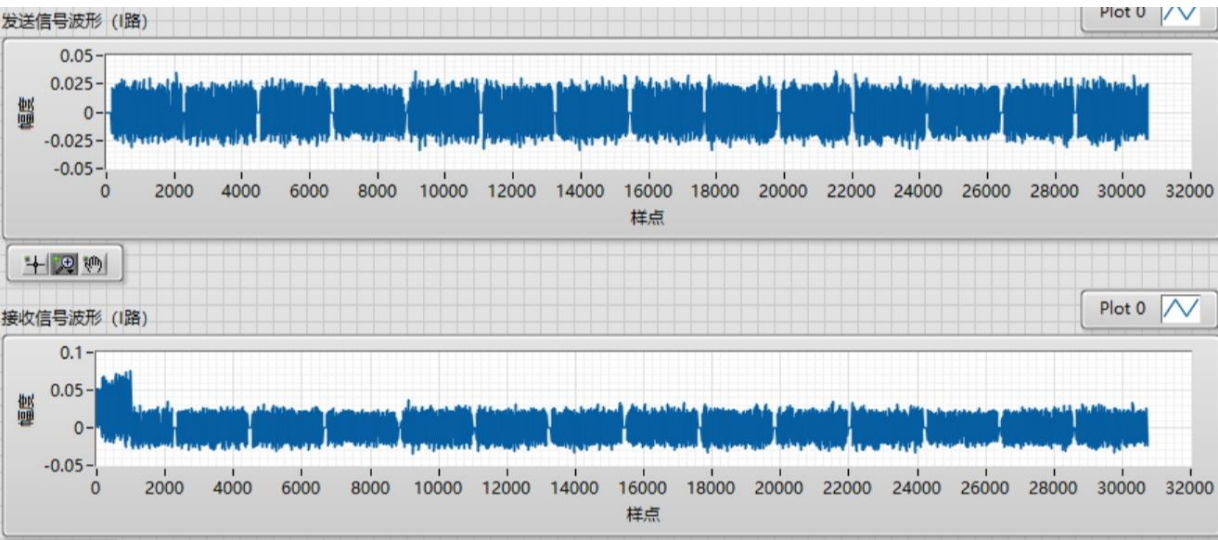


LDPC 模块的作用：信道编码通过在信源编码的输出上增加监督位/纠错位，使数据具有自检错和自纠正的能力，从而增加通信的可靠性。

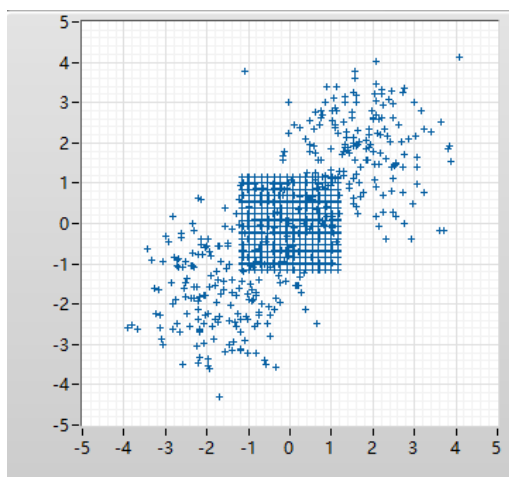
c.交织模块作用验证

配置：突发干扰开关打开，配置交织开关开启及关闭，不添加噪声，调制类型为256QAM，其余参数为默认参数

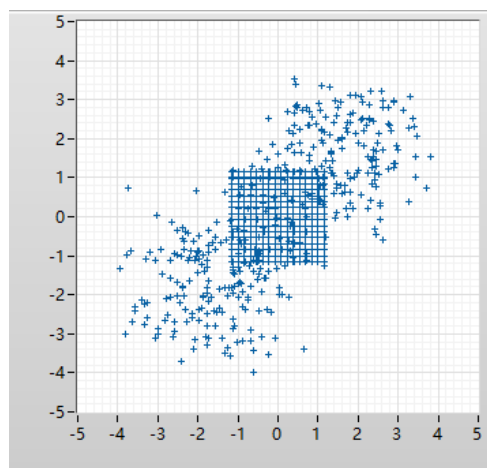
打开突发干扰后，发送信号与接收信号幅度出现不匹配，说明发生了突发干扰



当交织模块打开时，接收星座图出现分布错误的点，证明发生了突发干扰，解调制映射的误码率达到 0.0127691，最后信宿的误码率为 0，能正确接收图片



当交织模块关闭时，接收星座图出现分布错误的点，证明发生了突发干扰，解调制映射的误码率达到 0.013099

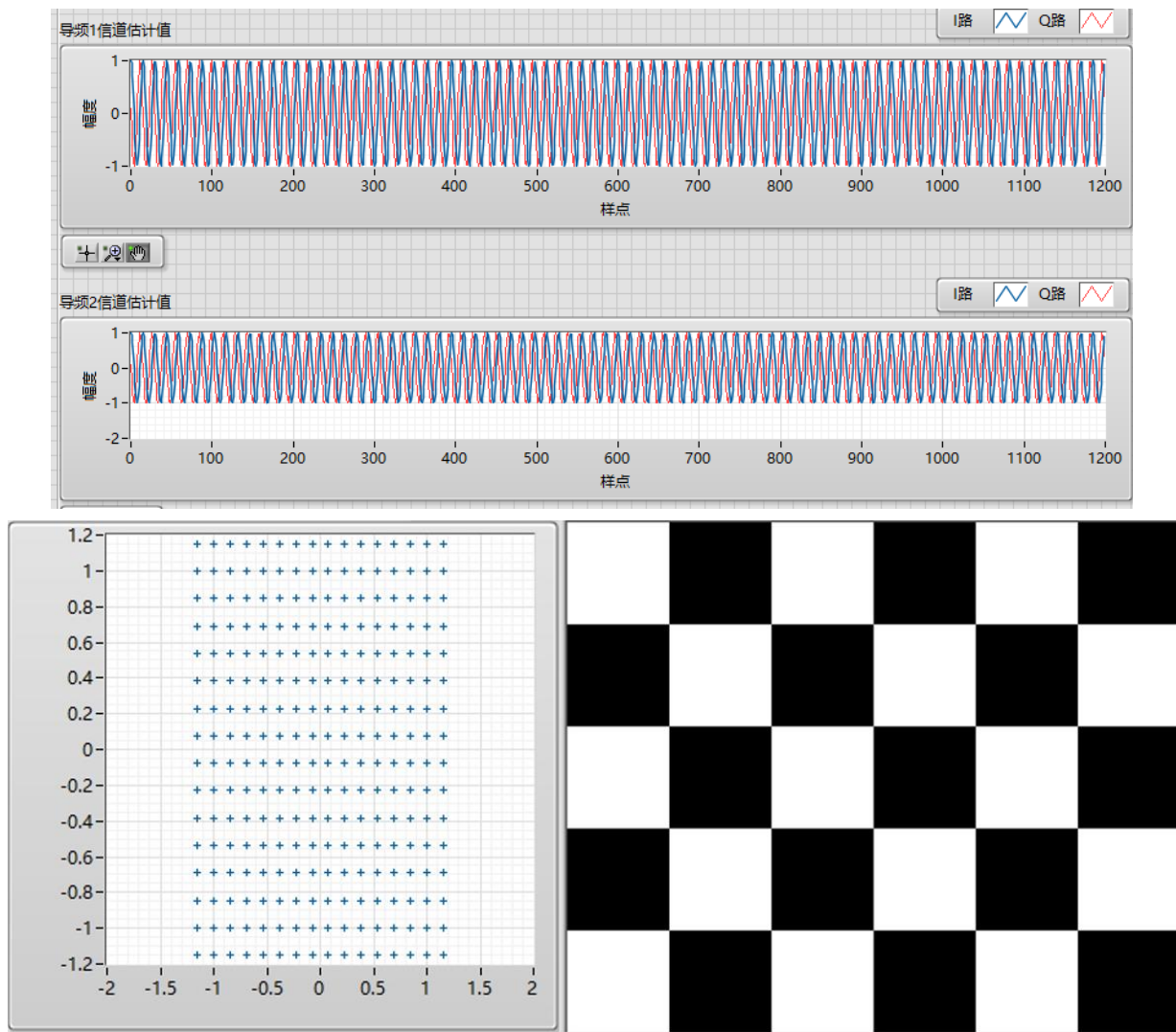


交织模块的作用：对比可知，交织模块通过将信号重新排序的方式，可以有效应对突发干扰，减轻突发干扰的影响，从而恢复出正确的数据，增加信道的可靠性。

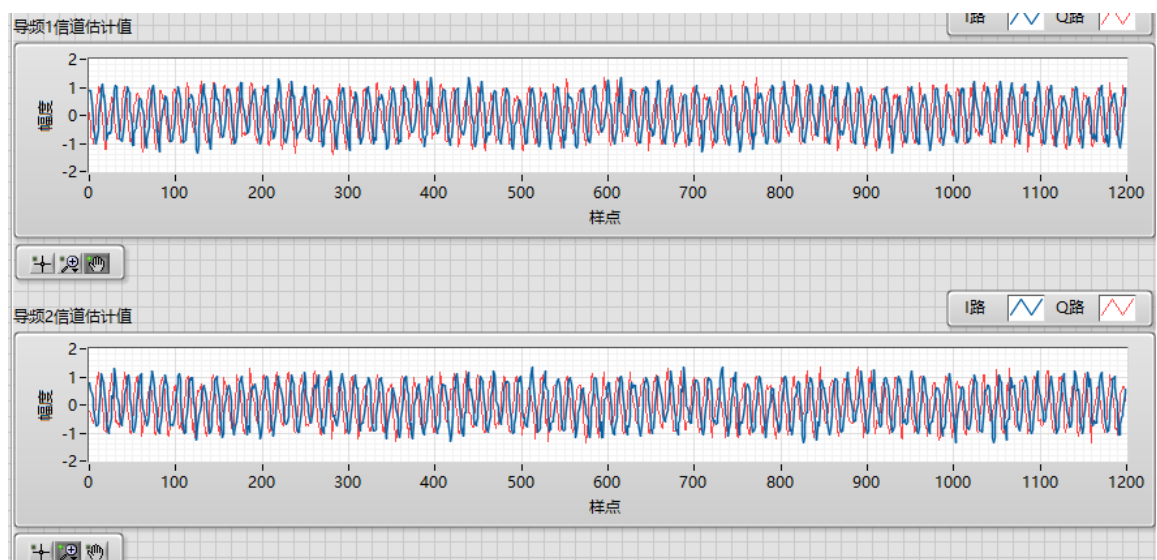
d.CP 模块作用验证

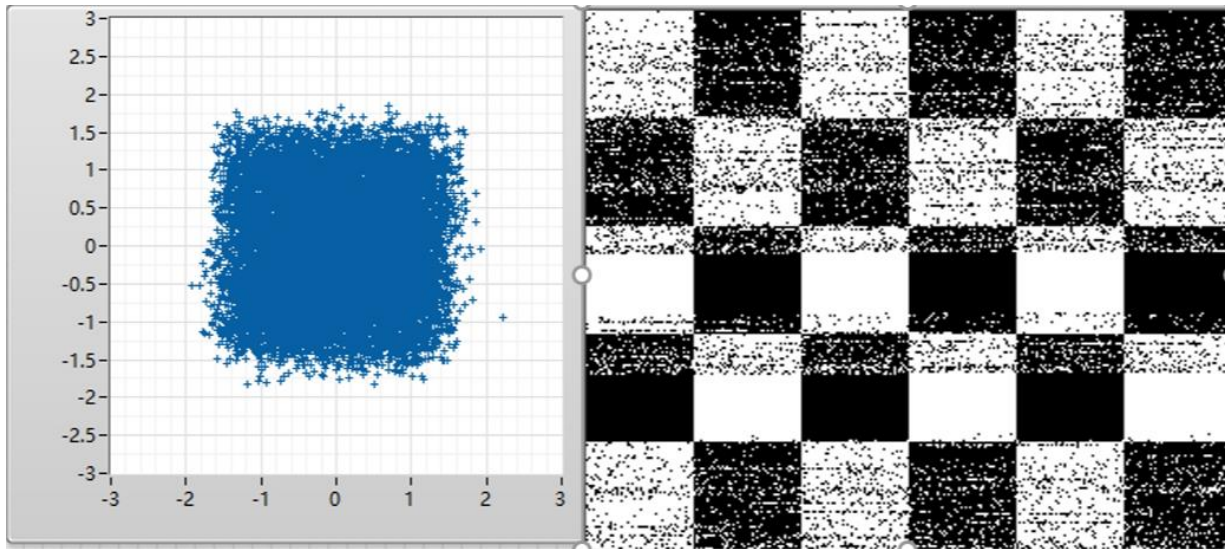
配置：CP 多径时延开关打开，配置 CP 开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

当 CP 模块打开时，接收信号正常，导频估计表现良好



当 CP 模块关闭时，星座图混乱，误码率达到 0.106792，CRC 校验不通过，导频估计出现不规则



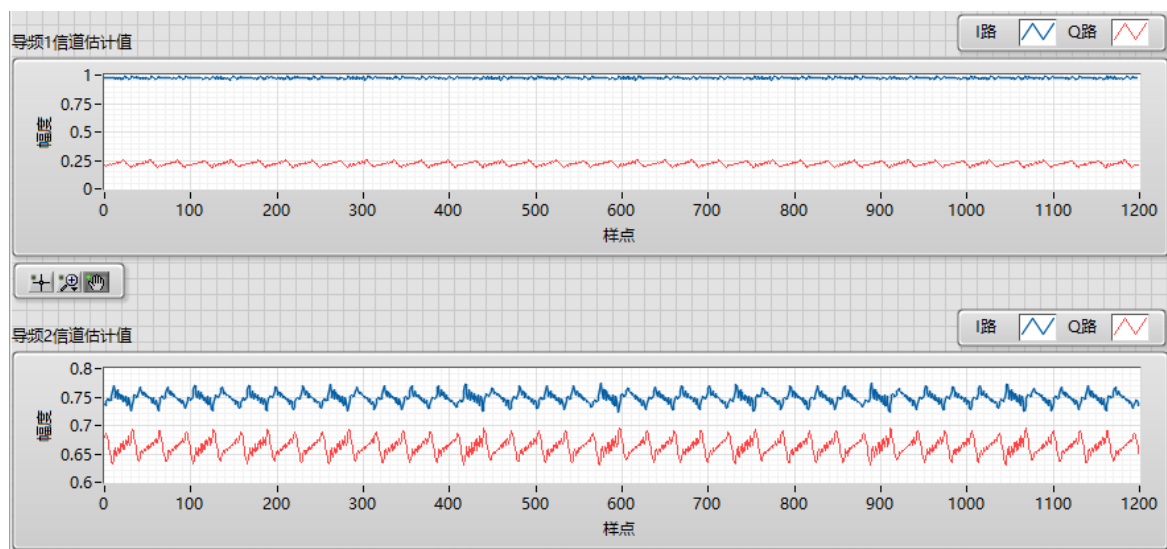


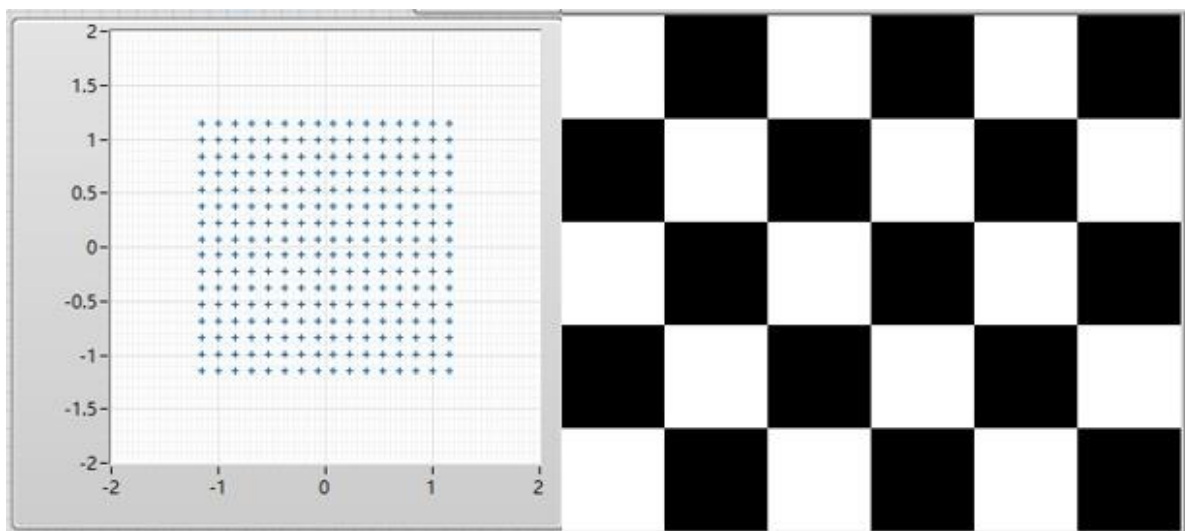
结论：OFDM 调制添加的 CP 模块能消除符号间干扰(ISI)的同时不破坏子载波之间的正交性，可以抵抗多径效应的影响。

e.信道估计模块作用验证

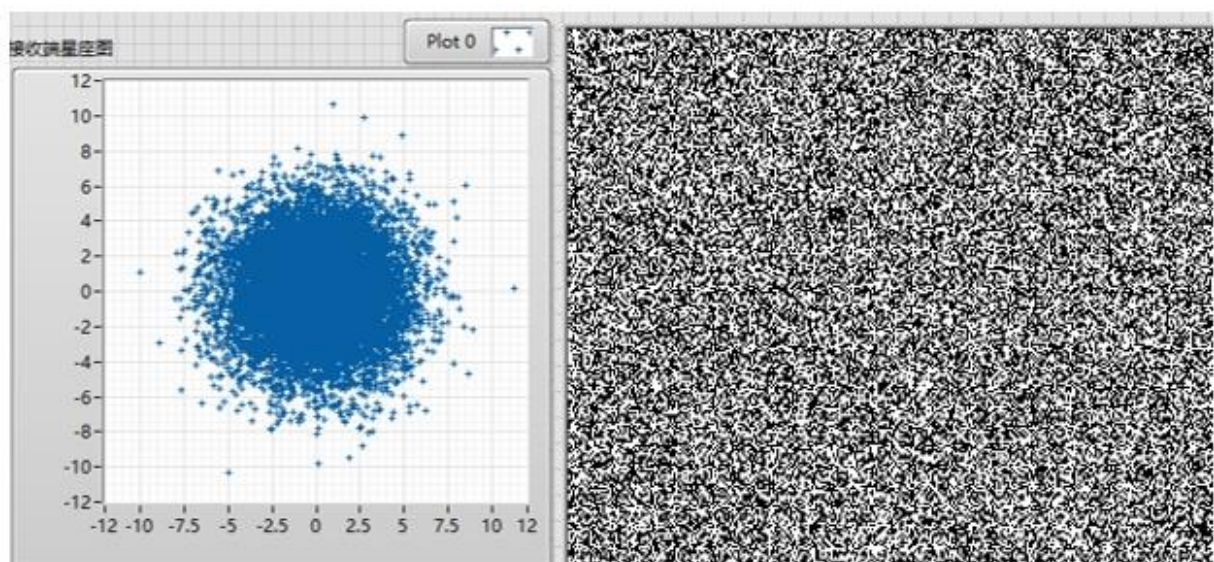
配置：频偏开关打开，频偏 140Hz，配置信道估计开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

当信道估计模块打开时，如下图可以根据对导频信号进行信道估计，根据信道估计的值，可以将信息还原正常，从而获得正确的信息





当信道估计模块关闭时，不会进行信道估计从而还原接收信息，这样接受的星座图便会有幅度上的误差和角度的偏移，如下图所示，此时误码率为 0.49332，CRC 检验不通过。

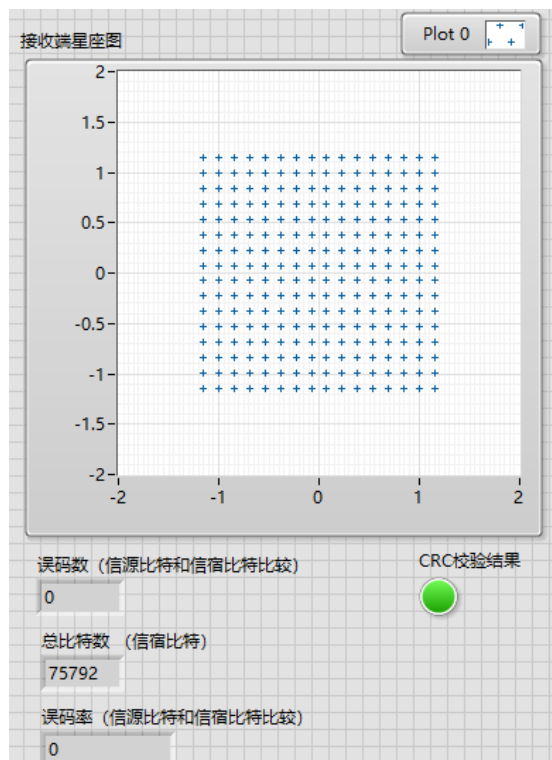
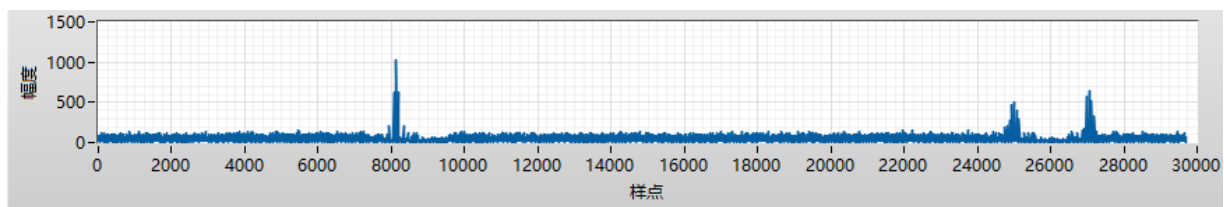


信道估计的作用：它涉及使用接收到的信号来估计信道的特性。信道估计的准确性直接影响到接收信号的解调质量和整个通信系统的性能。在 OFDM 系统中，信道估计尤为重要，因为它能够帮助接收机准确恢复发送端的信号。

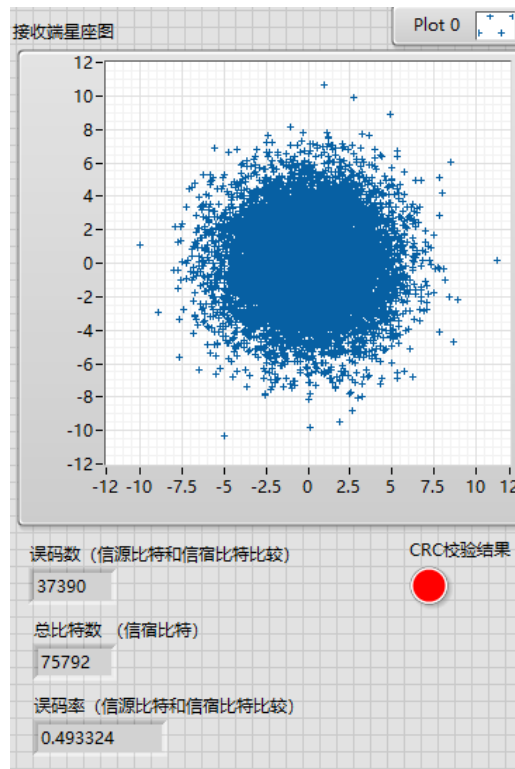
f.同步模块作用验证

配置：收发时延开关打开，配置同步开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

当帧同步模块打开时，接收信道能够检测出相关峰的位置并对其调整，接受的信息正常，如下图误码率为 0，CRC 检验通过



当帧同步关闭时，接受端口不能根据相关峰去调整接受的信息，此时就会受到延迟的影响导致接受的信息混乱，如下图误码率达到 0.493324，CRC 检验不通过

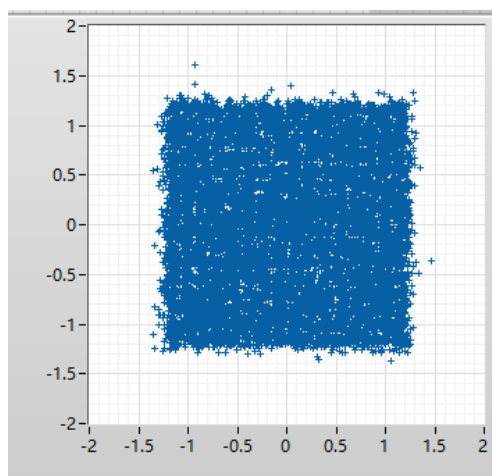
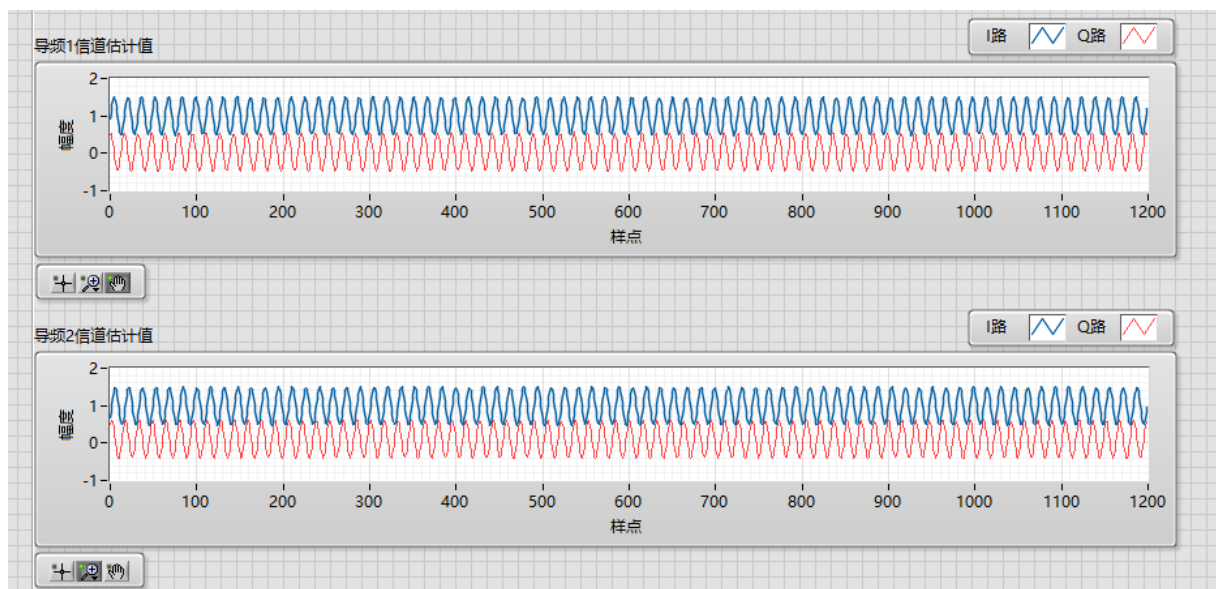


同步模块的作用：当存在收发延时，通过 PSS 信号的相关峰运算找到正确的顺序并将接收信息进行调整，从而恢复正常的信息。

g.多径使能验证

配置：配置多径使能打开，配置第 1 径和第 2 径时间延迟采样点，第 1 径和第 2 径幅度，第 1 径和第 2 径频偏，第 1 径和第 2 径相偏。调制类型为 256QAM，其余参数为默认参数

打开多径使能后，接收的星座图出现误差，信道估计起作用，调制映射前后有误码数，相关峰发生偏移



多径使能的影响：在实际的无线电波传播信道中(包括所有波段)，常有许多时延不同的传输路径，称为多径现象。由于传播路径的不同，通过各个路径接受的信息会有不同的时延、幅度变化、频偏、相偏等

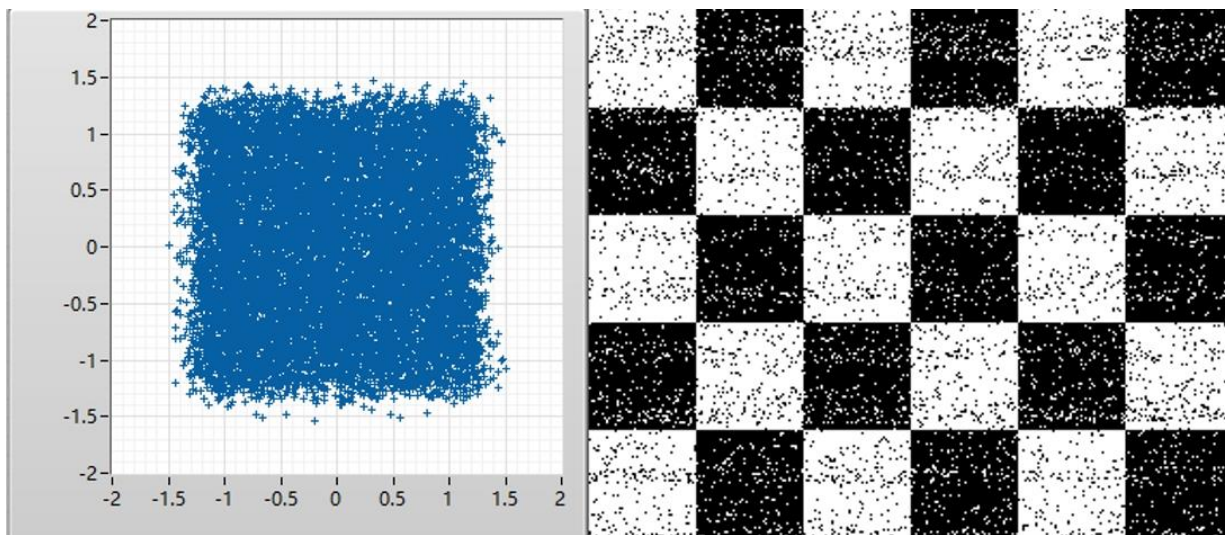
3.2 真实系统时基于软件无线电平台的 5G 物理层通信系统设计实验结果

在真实系统中各模块都打开发挥作用时，接收星座图表现良好，误码率为 0，CRC 校验通过

a. 真实系统下 LDPC 模块作用验证

配置：运行方式改为真实系统，配置 LDPC 开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

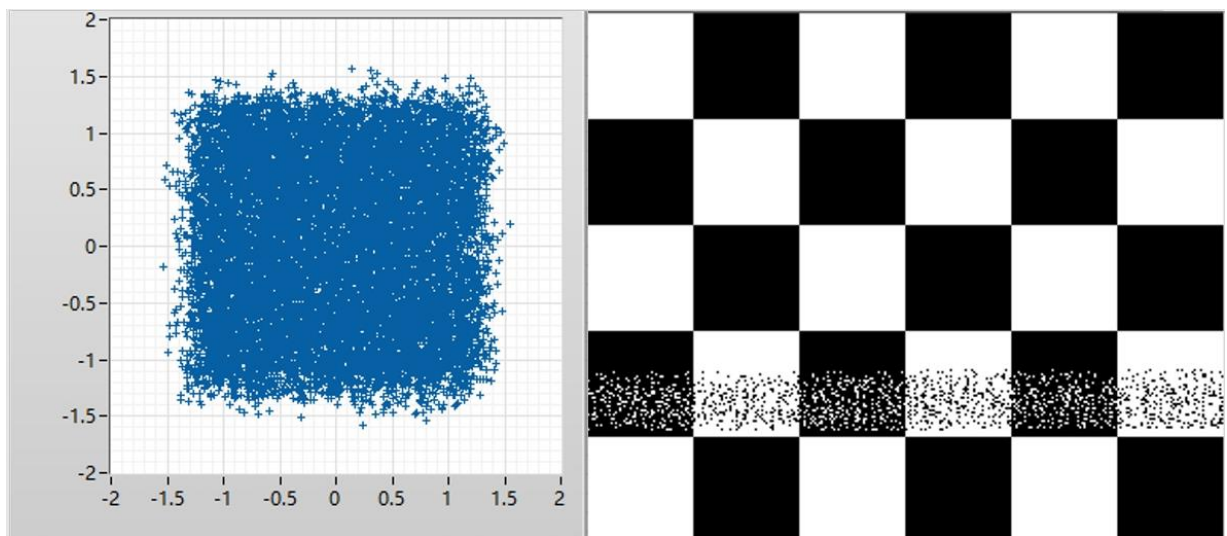
现象：关闭 LDPC 模块后，星座图出现误差，信宿误码率为 0.0694269，CRC 校验不通过



b. 真实系统下交织模块作用验证

配置：运行方式改为真实系统，配置交织开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

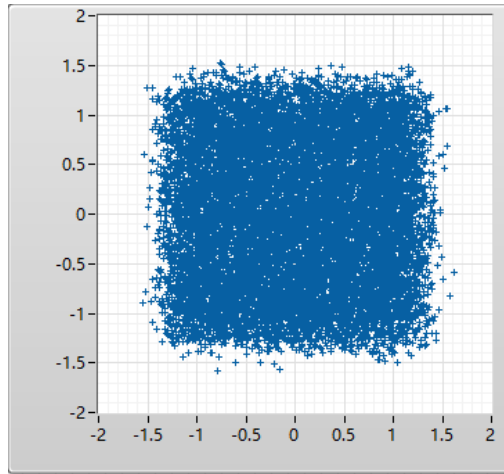
现象：关闭交织模块后，接收星座图出现误差，信宿误码率为 0.0204111，接收图片出现较为集中的干扰噪声。



c. 真实系统下 CP 模块作用验证

配置：运行方式改为真实系统，配置 CP 开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

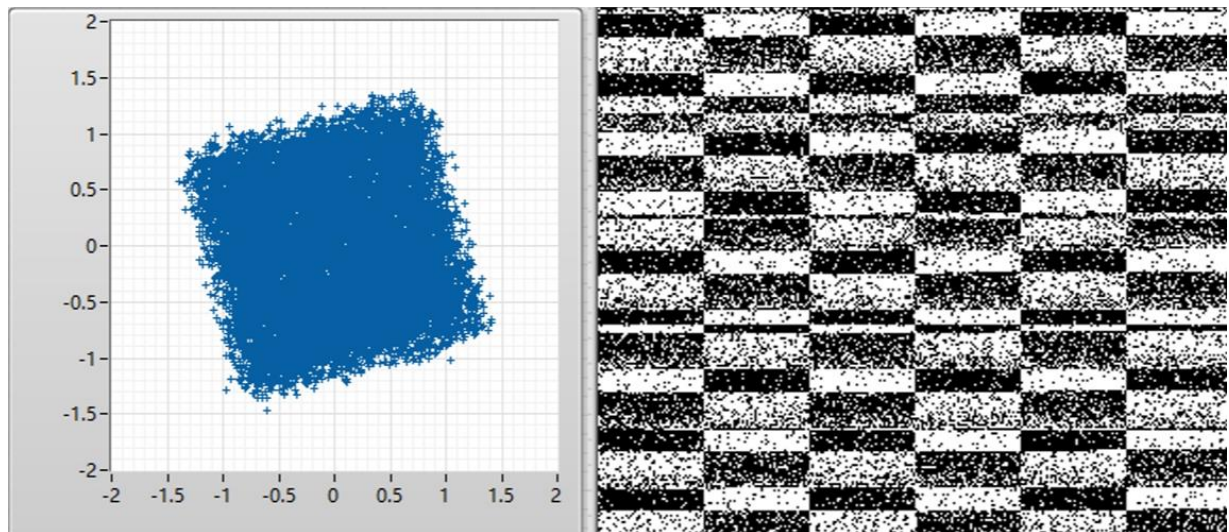
现象：关闭 CP 模块后接收星座图出现误差，但可能在真实信号中的 ISI 影响较小，因此最后信宿误差率为 0（即靠其他模块可以还原出正确的信号）



d.真实系统下信道估计模块作用验证

配置：运行方式改为真实系统，配置信道估计开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

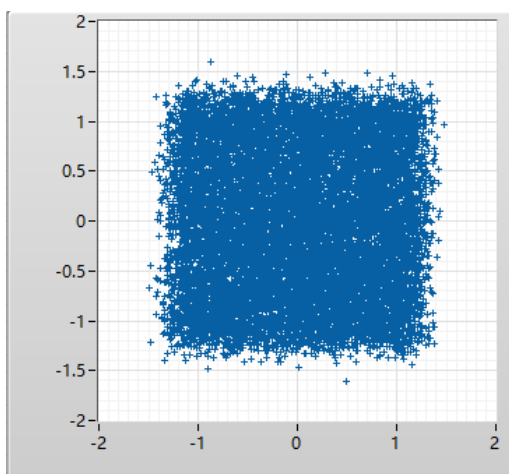
现象：关闭信道估计模块后，从接收星座图可以看出接受的信号出现明显的相位和幅度误差，误码率达到 0.508893，CRC 校验不通过



e.真实系统下同步模块的验证

配置：运行方式改为真实系统，配置同步开关开启及关闭，调制类型为 256QAM，其余参数为默认参数

现象：关闭同步模块后接收星座图出现误差，但可能在真实信号中时延为 0，因此最后信宿误差率为 0（即靠其他模块可以还原出正确的信号）

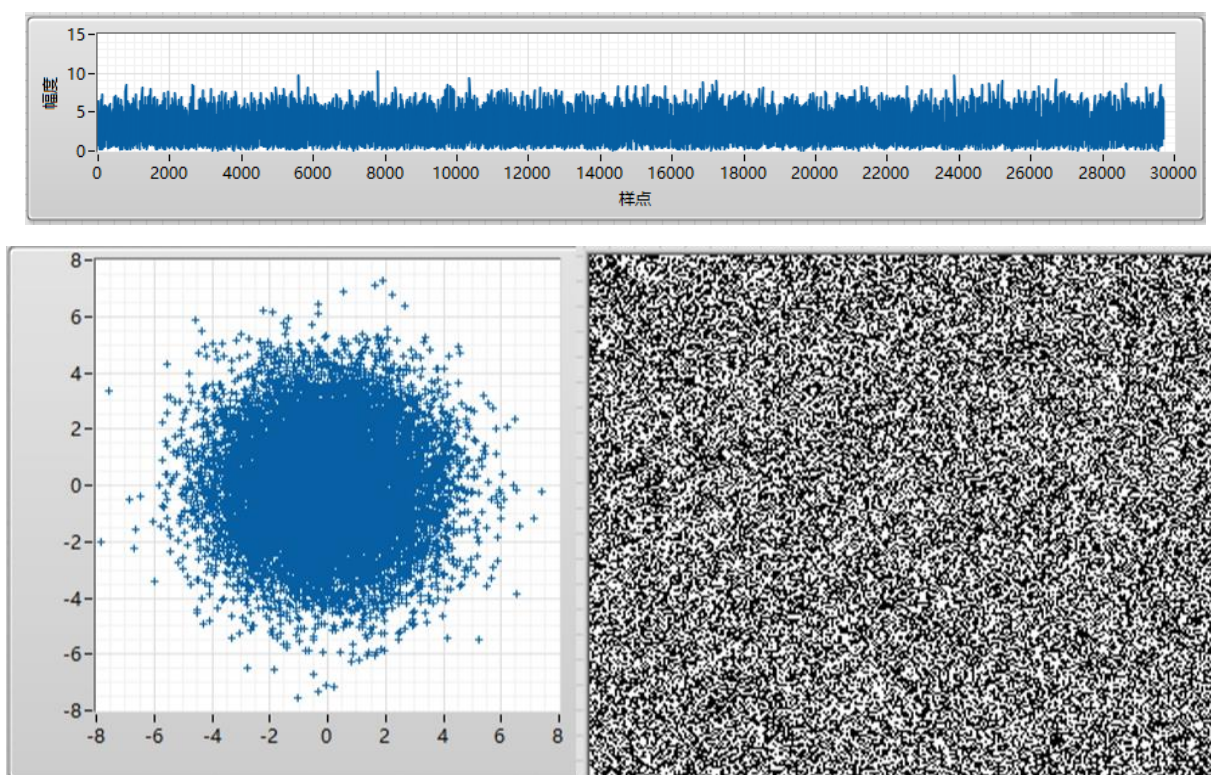


3.3 更改射频后基于软件无线电平台的 5G 物理层通信系统设计实验结果

a.更改发射频率和接收频率

配置：接收频率改为 2G，发射频率改为 2.38G，调制类型为 256QAM，其余参数配置为默认参数

接受的星座图混乱，误码率达到 0.500，CRC 检验不通过，相关峰不能有效检测出
timestart



结论：更改发射和接受的频率不一致，会导致接收到的信息解算完全错误，造成严重的信号失真，所以通信需要保持频率一致才能做到正确接收。

b.更改发射衰减和接收增益

配置：更改发射衰减、接收增益，运行方式配置为真实系统，调制类型为 256QAM，其余参数配置为默认参数

在试验中更改增益得到结果如下表所示

衰减增益（dB）	接收增益（dB）	调制映射误码率	信宿误码率
40	20	0	0
50	20	0.308845	0.210787
50	30	0.320764	0.226858
50	40	0.315972	0.224113

衰减增益的作用：衰减增益主要指的是信号在信道中传输过程中由于各种因素（如自由空间衰减、传输介质损耗、障碍物阻挡等）导致的信号强度减弱的现象。信号衰减会影响通信质量，因为它会导致信号的功率降低，从而可能使得信号在到达接收端时变得难以被正确识别和处理。在无线通信中，信号衰减还可能受到大气吸收、多径效应等因素的影响。为了对抗衰减，可以采用多种策略，如增加发射功率、使用中继和增强器、选择适当的频率等。

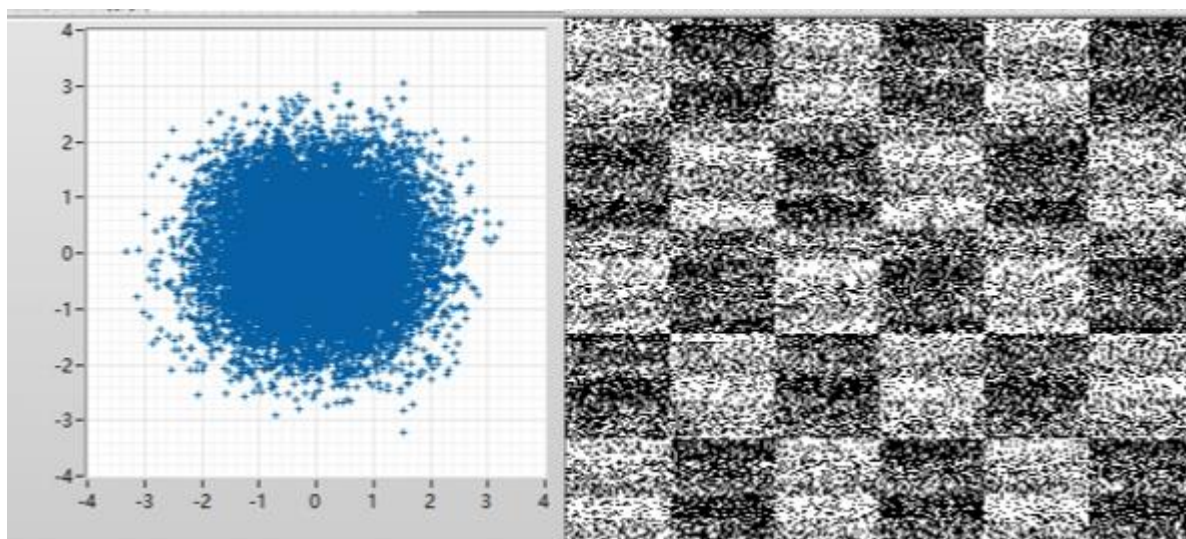
接收增益的作用：接收增益则是指在通信系统中，为了提高信号的接收质量，通过技术手段（如天线设计、信号处理技术等）增强接收到的信号强度的过程。

结论：在衰减增益一定范围内，增加接收增益可以抵消衰减增益的影响，但衰减增益超过一定范围以后，继续增加接收增益的作用不大。

c.更改天线

配置：拔掉天线，运行方式配置为真实系统，调制类型为 256QAM，其余参数配置为默认参数

现象：在拔掉天线之后，误码率达到 0.328913，接收星座图混乱



结论：天线的存在可以提高信号质量，减少干扰，并提升频谱利用率；减少信号的衰减和干扰，提高信号的接收质量，尤其是在多径传播环境中；通过精确的波束成形，减少能量的浪费，因为信号能量更集中地发送给目标用户，从而降低整体能耗

五、收获与体会

1. 收获

1. 学到了 matlab 的基础编写规则与使用。
2. 解 5G 物理层通信系统设计的整体流程，5G 物理层通信系统包括生成数据源、CRC 添加、LDPC 编码、速率匹配、交织、加扰、调制映射、预编码、产生导频数据、资源映射、OFDM 调制，最后将产生的数据通过以太网发送到 XSRP 软件无线电平台，在软件无线电平台中完成 IQ 数据 DA 转换、上变频载波调制、射频在指定频点将信号通过天线发射出去。无线比特经过空中无线信道，再通过射频的接收天线在对应的频点将数据接收、下变频、低通滤波、AD 转换得到 IQ 信号，通过以太网发送到电脑。在电脑上进行 OFDM 解调、解资源映射、信道估计与均衡、解预编码、解调制映射、解扰、解交织、解速率匹配、LDPC 译码、CRC 校验、信宿。
3. 我了解到 5G 是一种基于新的无线技术标准的通信技术，它具有更高的传输速率和更低的延迟。这使得 5G 能够支持更广泛的应用，包括物联网、虚拟现实和增强现实等新兴技术。
4. 我学习了 5G 网络的架构。与传统的 4G 网络相比，5G 网络采用了更为灵活的虚拟化和切片技术，可以根据不同的应用场景和需求进行灵活配置和部署。这使得 5G 网络

可以更好地适应不同的数据传输和处理需求。尤其是以 256QAM 未调制映射方法使得 5G 通信在相同的信道长度下能够携带更多的信息。帧同步、交织、加/解扰等模块增强了 5G 通信的安全性，使得通信更加可靠

2.体会

1. 本次 5G 实验难度较大，尤其是手边现成资料少，网上资料又难懂的情况下，更需要我们加强与指导老师和同学的交流，从而能够顺利完成实验，闭门造车只会拖延实验的进度。
2. 5G 通信实验作为一个完整的大实验，在完成每一个分模块时需要联系它在整体中起的作用，才能在完成分模块时准确理解它的思路，否则跟着教程即使做出来了一个模块，也不能准确掌握它的意义。
3. 在做实验的过程中要准确理解指导资料、寻找相关信息、阅读数据手册，才能少走弯路。

六、学生代码

1. CRC 添加

```
function p = crc24a(c)
% 功能：以校验多项式为除数的多项式除。计算 24 位 crc, 得到校验二进制校验数组 p
% 输入：
%      c: 需要进行 CRC 校验的信息
% 输出：
%      p: 进行 CRC 校验的校验位，由低位到高位排列
%
L = 24;
p = zeros(1,L); % 24 位校验位
lenD = length(c);
G=[1 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 1 1 1 0 1 1]; % 生成多项式，由低位到高位
DataCRC1=[c,zeros(1,L)];
for i= 1:lenD
    if DataCRC1(i)==1
        for j=(1:length(G))
            DataCRC1(i+j-1)=xor(DataCRC1(i+j-1),G(j));
        end
    end
end
```

```

        end
    end
    for k= 1:L
        p(k)=DataCRC1(lenD+k);
    end

```

2. 速率匹配

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% student code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%根据 I_LBRM 确定 Ncb
TBS_LBRM = 100;%有待确定
[a1,N]=size(Info_data);%是输入序列的一维长度
R_LBRM = 2/3;
Nref = floor(TBS_LBRM/(C*R_LBRM));
if (I_LBRM==0)
    Ncb = N;
    disp('根据 I_LBRM==0, Ncb 取 N');
else
    Ncb = min(N,Nref);
    disp('根据 I_LBRM==1, Ncb 取 min(N,Nref)');
end
%根据 rvid 确定
switch rvid
    case 0
        k0 = 0;
    case 1
        if(LDPC_base_graph==1)
            k0 = Zc*floor((17*Ncb)/(66*Zc));
        elseif(LDPC_base_graph==2)
            k0 = Zc*floor((13*Ncb)/(50*Zc));
        else
            disp('LDPC_base_graph 条件出错')
        end
    case 2
        if(LDPC_base_graph==1)
            k0 = Zc*floor((33*Ncb)/(66*Zc));
        elseif(LDPC_base_graph==2)
            k0 = Zc*floor((25*Ncb)/(50*Zc));
        else
            disp('LDPC_base_graph 条件出错')
        end
    case 3
        if(LDPC_base_graph==1)
            k0 = Zc*floor((56*Ncb)/(66*Zc));

```

```

elseif(LDPC_base_graph==2)
    k0 = Zc*floor((43*Ncb)/(50*Zc));
else
    disp('LDPC_base_graph 条件出错')
end

otherwise
    disp('rvid 出错, 应为 0、1、2、3')
end
%判断 k, Er 大小, 由 k0, Ncb 计算匹配后的数据
j = 0;
Er
length= size(Info_data)
Output = zeros(1,Er);
for k=1:Er
    Output(1,k) = Info_data(mod(k0+j,Ncb)+1);
    j=j+1;
end
out_len = Er;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

3. 调制映射

```

if(modulertype == 1) %QPSK
    for(kkk=1:symbol_len)
        temp = inputdata(2*kkk-1)*2+inputdata(2*kkk)+1;
        out(kkk) = QPSK_table(temp);
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% student code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif(modulertype == 2) %16QAM
    for(kkk=1:symbol_len)
        temp = inputdata(4*kkk-3)*8+inputdata(4*kkk-2)*4+inputdata(4*kkk-1)*2+inputdata(4*kkk)+1;
        out(kkk) = QAM16_table(temp);
    end

elseif(modulertype == 3) %64QAM
    for(kkk=1:symbol_len)
        temp = inputdata(6*kkk-5)*32+inputdata(6*kkk-4)*16+inputdata(6*kkk-3)*8 ...
            + inputdata(6*kkk-2)*4+inputdata(6*kkk-1)*2 +inputdata(6*kkk)+1;
        out(kkk) = QAM64_table(temp);
    end
end

```

```

else %256QAM
    for(kkk=1:symbol_len)
        temp = inputdata(8*kkk-7)*128+inputdata(8*kkk-6)*64 ...
            + inputdata(8*kkk-5)*32+inputdata(8*kkk-4)*16+inputdata(8*kkk-3)*8 ...
            + inputdata(8*kkk-2)*4+inputdata(8*kkk-1)*2 +inputdata(8*kkk)+1;
        out(kkk) = qam256(temp);
    end
end

```

4. OFDM 调制

```

function [out] = GenTimeData_5G(fredata,Nsym_slot,Nu)
%GENTIMEDATA_5G 此处显示有关此函数的摘要
% 此处显示详细说明
nfft = Nu;
start =1;
for iii=1:Nsym_slot
    %端口 0, 符号 iii
    % tmpdata=[inputdata(iii,(half+1):total),zeros(1,(nfft-
total)),inputdata(iii,1:half)];
    tmpdata=fredata(iii,:);
    ifftdata = ifft(tmpdata,Nu);
    tail = start+nfft-1;
    out([start:tail]) =ifftdata([1:nfft]);
    start = tail+1;
end
end

```

5. 解调制映射

```

switch qm
case 2 %QPSK 硬解调
    for(iii=1:len)
        if(realdata(iii)>0)
            outdemod(2*(iii-1)+1) = 0;
        else
            outdemod(2*(iii-1)+1) = 1;
        end
        if(imagedata(iii)>0)

```

```

        outdemod(2*(iii-1)+2) = 0;
    else
        outdemod(2*(iii-1)+2) = 1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% student code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 4 %16QAM 硬解调
    for(iii=1:len)
        tmpdata1 = 2*d1 - abs(realdata(iii));
        tmpdata2 = 2*d1 - abs(imagedata(iii));
        if(realdata(iii)>0)
            outdemod(4*(iii-1)+1) = 0;
        else
            outdemod(4*(iii-1)+1) = 1;
        end
        if(imagedata(iii)>0)
            outdemod(4*(iii-1)+2) = 0;
        else
            outdemod(4*(iii-1)+2) = 1;
        end
        if(tmpdata1>0)
            outdemod(4*(iii-1)+3) = 0;
        else
            outdemod(4*(iii-1)+3) = 1;
        end
        if(tmpdata2>0)
            outdemod(4*(iii-1)+4) = 0;
        else
            outdemod(4*(iii-1)+4) = 1;
        end
    end
end

case 6 %64QAM 硬解调
    for(iii=1:len)
        tmpdata1 = 4*d2 - abs(realdata(iii));
        tmpdata2 = 4*d2 - abs(imagedata(iii));
        tmpdata3 = 2*d2 -abs(tmpdata1);
        tmpdata4 = 2*d2 -abs(tmpdata2);
        if(realdata(iii)>0)
            outdemod(6*(iii-1)+1) = 0;
        else
            outdemod(6*(iii-1)+1) = 1;
        end
        if(imagedata(iii)>0)
            outdemod(6*(iii-1)+2) = 0;

```

```

else
    outdemod(6*(iii-1)+2) = 1;
end
if(tmpdata1>0)
    outdemod(6*(iii-1)+3) = 0;
else
    outdemod(6*(iii-1)+3) = 1;
end
if(tmpdata2>0)
    outdemod(6*(iii-1)+4) = 0;
else
    outdemod(6*(iii-1)+4) = 1;
end
if(tmpdata3>0)
    outdemod(6*(iii-1)+5) = 0;
else
    outdemod(6*(iii-1)+5) = 1;
end
if(tmpdata4>0)
    outdemod(6*(iii-1)+6) = 0;
else
    outdemod(6*(iii-1)+6) = 1;
end
end

case 8 %256QAM 硬解调
for(iii=1:len)
    tmpdata1 = 8*d3 - abs(realdata(iii));
    tmpdata2 = 8*d3 - abs(imagedata(iii));
    tmpdata3 = 4*d3 -abs(tmpdata1);
    tmpdata4 = 4*d3 -abs(tmpdata2);
    tmpdata5 = 2*d3 -abs(tmpdata3);
    tmpdata6 = 2*d3 -abs(tmpdata4);
    if(realdata(iii)>0)
        outdemod(8*(iii-1)+1) = 0;
    else
        outdemod(8*(iii-1)+1) = 1;
    end
    if(imagedata(iii)>0)
        outdemod(8*(iii-1)+2) = 0;
    else
        outdemod(8*(iii-1)+2) = 1;
    end
    if(tmpdata1>0)
        outdemod(8*(iii-1)+3) = 0;

```

```

else
    outdemod(8*(iii-1)+3) = 1;
end
if(tmpdata2>0)
    outdemod(8*(iii-1)+4) = 0;
else
    outdemod(8*(iii-1)+4) = 1;
end
if(tmpdata3>0)
    outdemod(8*(iii-1)+5) = 0;
else
    outdemod(8*(iii-1)+5) = 1;
end
if(tmpdata4>0)
    outdemod(8*(iii-1)+6) = 0;
else
    outdemod(8*(iii-1)+6) = 1;
end
if(tmpdata5>0)
    outdemod(8*(iii-1)+7) = 0;
else
    outdemod(8*(iii-1)+7) = 1;
end
if(tmpdata6>0)
    outdemod(8*(iii-1)+8) = 0;
else
    outdemod(8*(iii-1)+8) = 1;
end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

6. 解扰

```

function [descrambledata] = DeScramble_5G(demoddata,vrb_num,qm,ue_index,cellid)
totallen = vrb_num * 12 * 12 * qm;%我需求输出的长度
cinit = ue_index * ( 2 ^ 15 ) + cellid;
NC = 1600;
setlen= NC +totallen - 31;%我需要生成的 x1x2 长度（需求长度+增强随机性变量-已有长度）

x1 = zeros( 1, 31 );
x1( 1 ) = 1;

```



```

for ( i = 1:31 )
    x2( i ) = mod( cinit, 2 );
    cinit = floor( cinit / 2 );
end

for ( i = 1:setlen )
    x1( i + 31 ) = xor( x1( i + 3 ), x1( i ) );
    x2( i + 31 ) =xor(xor(x2(i+3),x2(i+2)),xor(x2(i+1),x2(i)));
end
for ( i = 1:totallen )

    scrambit( i ) = xor(x1( i + NC ) , x2( i + NC ));%后半部分

end

for ( i = 1:totallen )
    descrambledata( i ) = xor( scrambit(i) , demodata( i ));%b+c) mod2
end
end

```

7. 帧同步

```

function [ searchflag, timestart, corrrdata ] = LTECorrFun( rxdata, corrlen,
rsdata )
pssdata1 = ifft( rsdata, 2048 );
pssdata = pssdata1( 1, 1:2048 );
c=sum(abs(pssdata))
corr = zeros( 1, corrlen );
for data_num = 1:corrlen-1024
    for n = 1:length( pssdata )
        corr( data_num ) = corr( data_num ) + rxdata( data_num + n -
1 )*conj(pssdata( n ));
    end
end
corrrdata = abs( corr );

timestart = find( max( corrrdata ) == corrrdata );
if max( corrrdata ) > 2 * ( sum( corrrdata ) - max( corrrdata ) ) / ( corrlen - 1 )
    searchflag = 1;
else
    searchflag = 0;
end

```

```
end
```

8. 信道估计与均衡

```
%% 包含信道估计、信道均衡
```

```
function [outdata,slotlschanneldata] =
```

```
NR_LSChannel( deremapdata,TransformPrcode_en,cellid,nslotnum,beltaDMRS,UL_DMRS_Conf  
ig_type,Frist_DMRS_L0,Second_DMRS_L,prb_num,EN_CHES,DMRS_symbol,Nsym_slot)
```

```
%% 信道估计
```

```
if(EN_CHES)
```

```
    %计算导频数据
```

```
    rsdata =
```

```
NR_RS_Gen(TransformPrcode_en,cellid,nslotnum,beltaDMRS,UL_DMRS_Config_type,Frist_DM  
RS_L0,Second_DMRS_L,prb_num);
```

```
    scnum = 12*prb_num;
```

```
    %计算 2 个导频符号的位置，由 L0+L' 计算获得，
```

```
    l1 = Frist_DMRS_L0 +0;
```

```
    l2 = Second_DMRS_L+1;
```

```
    %matlab 没有符号 0
```

```
    rs_station0 = l1+1;
```

```
    rs_station1 = l2+1;
```

```
    %获取 2 个导频的频域数据
```

```
    rs0fredata = deremapdata(rs_station0,1:scnum);
```

```
    rs1fredata = deremapdata(rs_station1,1:scnum);
```

```
    for(iii = 1:scnum)
```

```
        if(0 == rsdata(1,iii))    %检测除数为 0 的情况
```

```
            out(1,iii) = 1;
```

```
            out(2,iii) = 1;
```

```
        else
```

```
            out(1,iii) = rs0fredata(iii)/rsdata(1,iii);
```

```
            out(2,iii) = rs1fredata(iii)/rsdata(2,iii);
```

```
        end
```

```
    end
```

```
else
```

```
    scnum = 12*prb_num;
```

```
    out = ones(2,scnum);
```

```
end
```

```
slotlschanneldata = out;
```

```
%% 信道均衡
```

```
start=1;
```

```
start1 =1;
```

```

datasymbol = 0;
carry_len = prb_num*12;
halfsymbol = Nsym_slot/2;
slotssymbol = Nsym_slot-DMRS_symbol;
tail1 = start1+carry_len-1;
if(2 ==DMRS_symbol)
    %计算 2 个导频符号的位置，由 L0+L' 计算获得，
    rsl1 = Frist_DMRS_L0 +0;
    rsl2 = Second_DMRS_L+1;
    %matlab 没有符号 0
    rs_station0 = rsl1+1;
    rs_station1 = rsl2+1;
else
    rsl1 = Frist_DMRS_L0 +0;
    %matlab 没有符号 0
    rs_station0 = rsl1+1;
    rs_station1 = 20; %一个 slot 的符号不会超过 20，不进行资源分配
end
for(iii=1:halfsymbol) %前面一半符号的处理，用一个导频符号的信道估计值
    if(iii == rs_station0) %第一个导频符号跳过
        continue;
    end
    datasymbol = datasymbol+1;
    tail = start+carry_len-1;
    outdata(start:tail) =
deremapdata(iii,start1:tail1)./slotlschanneldata(1,start1:tail1);
    start = tail+1;
end

for(iii=(halfsymbol+1):Nsym_slot) %后面一半符号的处理，用二个导频符号的信道估计值
    if(iii == rs_station1) %第二个导频符号跳过
        continue;
    end
    datasymbol = datasymbol+1;
    tail = start+carry_len-1;
    % outdata(datasymbol,start:tail) =
deremapdata(iii,start1:tail1)./slotlschanneldata(2,start1:tail1);
    outdata(start:tail) =
deremapdata(iii,start1:tail1)./slotlschanneldata(2,start1:tail1);
    start = tail+1;
end
end

```