

浙江大学

本科实验报告

课程名称：通信原理实验

姓 名：黄嘉欣

学 院：信息与工程学院

系：信息与工程学系

专 业：信息工程

学 号：3190102060

指导教师：龚淑君 金向东

2022 年 6 月 1 日

专业： 信息工程
姓名： 黄嘉欣
学号： 3190102060
日期： 2022 年 6 月 1 日
地点： 东四-319

浙江大学实验报告

课程名称： 通信原理实验 指导老师： 龚淑君 金向东 成绩： _____
实验名称： 帧同步和 DPSK 调制解调 实验类型： 设计性实验 同组学生： 张维豆

一、实验目的

- ① 了解帧同步的概念和实现方法；
- ② 了解差分二进制相移键控（DBPSK）的编码与解码实现方法；

二、实验设备

- ① USRP 设备 1 台；
- ② 安装 LabVIEW 环境的电脑 1 台。

三、实验概要

判断发送、接收系统是否工作正常，需要对发送和接收到的数据比特作比较，以确认它们的位模式是否一样。在上一个实验当中，接收机能正确接收到短的位序列（例如：1011100）。实际情况中对于长序列 接收机有必要识别出发送序列的起始位置。因此，发送机需要在每一帧数据的开头，插入一段训练序列（头或前导码），以此来标识有效数据的开始。接收机对接收到的每一帧数据搜索这个训练序列，当接收到的数据和这段固定模式序列的互相关性较高的时候，就表明实现了帧同步。

数字调制和解调类型分为两种：相干和不相干。相干调制/解调系统需要载波相位信息监测和确认发送的数据，而非相干系统不需要这些相位信息。在本实验中，将介绍一种非相干调制方式：差分二进制相移键控（DBPSK）。

四、实验内容与步骤

1. 帧同步

在这一部分实验内容中，需要完成以下子 VI：

- 1) AddTraining.gvi：在一帧数据中插入训练序列；

装
订
线

- 2) FrameSync.gvi: 检测训练序列并且实现帧同步;
- 3) Sim_framesync.gvi: 对以上子 VI 进行仿真;

1.1 添加训练序列

设计“AddTraining.gvi”子 VI，在发送电路中，将训练序列符号插入一帧中。表 4.1.1 是输入输出端口情况。

表 4.1.1 输入输出端口情况

	端口名称	类型	描述
输入	Original frame	双精度数组	发送符号中的原始数据帧
	Training sequence	双精度数组	
输出	Frame with training sequence	双精度数组	在起始位置有训练序列的数据帧

使用“数组插入模块”，在原始帧的起始位置插入训练序列（“数组插入”模块的“索引”端口值应该设置为 0）。

1.2 帧检测和同步

设计“FrameSync.gvi”子 VI，为了对齐和同步帧，将训练序列符号放置在发送机端的发送数据帧起始位置。表 4.1.2 是输入输出端口情况。

表 4.1.2 输入输出端口情况

	端口名称	类型	描述
输入	Unsynced frame	双精度数组	同步前接收到的数据帧
	Training sequence	双精度数组	
	Threshold	双精度	检测出的最小峰值
输出	Synced frame	双精度数组	同步数据帧
	Cross-correlation result	双精度数组	接收到的数据帧和训练序列之间的互相关结果
	Peak index	整数数组	互相关结果峰值索引

- 1) 使用“信号相关性”模块（分析/信号处理/运算），在函数设置中选择“互相关”选项，计算接收到的数据帧和训练序列的互相关；
- 2) 使用“阈值检测”模块检测互相关值的峰值和它们的索引位置（提示：“峰值检测模块”检测超出设定阈值的输入序列的峰值，它在“位置”端口输出峰值的索引值，在“计数”端口输出峰值的数量）；

- 3) 从接收到的数据帧中移除训练序列，得到同步数据帧（提示：使用“索引数组”模块获取第一个相关值峰值的索引，使用“数组子集”从“Unsynced frame”中得到“Synced frame”）。

1.3 仿真

创建一个子 VI 并将它命名为“sim_framesync.gvi”，用来确认“AddTraining”和“FrameSync”是否正确。

- 1) 使用“MT Generate Bits”模块生成数据比例序列。如：配置模块为“Fibonacci”PN 序列，total bits 为 50，PN sequence order 为 5；
- 2) 将数据比特映射成符号：1 映射成符号“1”，0 映射成符号“-1”。创建一个显示控件以监测产生的符号序列，它将作为不包含训练序列的原始数据帧；
- 3) 用 13 位 Barker 码创建一个常量数组，作为训练符号序列。13 位 Barker 码为：
+1+1+1+1+1-1-1+1+1-1+1-1+1；
- 4) 使用“AddTraining”模块将训练序列插入到原始数据帧的起始位置，并产生加入了训练序列的数据帧；
- 5) 使用“FrameSync”模块确定上一步骤中产生的数据帧中训练序列的位置。需要观察相关值，并基于互相关值选择一个合适的阈值以找出需要的峰值。

1.4 问题 1

- 1) 帧同步需要的互相关峰值在哪里？峰值与原始数据帧的起始位置有什么关系？你的“FrameSync.gvi”子 VI 是否产生了同步数据帧？
- 2) 一个好的训练序列应该要能给出一个明显的互相关峰值，以便于检测。因此可能需要增加训练序列的长度。Barker 码的最大长度是 13，可以将两个或更多的 Barker 码连接起来。使用 13 位序列创建一个 26 位训练序列（提示：使用“创建数组”模块，并且勾选其行为特性“连接输入”来连接两个 Barker 码）。阈值将作何改变？此时，互相关的峰值在哪里？原始数据帧的起始位置在哪里？
- 3) 长训练序列的优点和缺点是什么？

2. 差分二进制相移键控（DBPSK）

2.1 发送电路

在上一个实验“TxBPSK.gvi”的基础上完成此部分实验。新建一个项目，将此文件

拷贝到新项目中，重命名为“TxDBPDK.gvi”。在这部分实验中，将要构建一个传输电路，完成差分二进制相移键控调制（DBPSK）。同 BPSK 一样，数据信息由载波信号的相位携带。与 BPSK 不同之处在于：DBPSK 传输的符号不仅取决于现在的位信息，也取决于前一个位信息。

在前面板中设置以下参数：

- ① IQ rate = 400k;
- ② carrier frequency = 1G;
- ③ gain = 0;
- ④ active antenna = TX1;
- ⑤ Message length = 1000;
- ⑥ Symbol rate = 10k;
- ⑦ Enabled Channels = 0。

1) 参考表 4.2.1，对添加了训练序列的数据帧进行差分编码：

表 4.2.1 差分编码参考

输入符号	-1	-1	1	1
前一符号	-1	1	-1	1
当前符号	1	-1	-1	1

- 2) 重复利用发送电路其余部分电路：脉冲成形滤波器、卷积等；
- 3) 同时在发送和接收电路中对 NI-USRP 属性模块添加“LO Frequency”参数的配置，如图 4.2.1 所示：

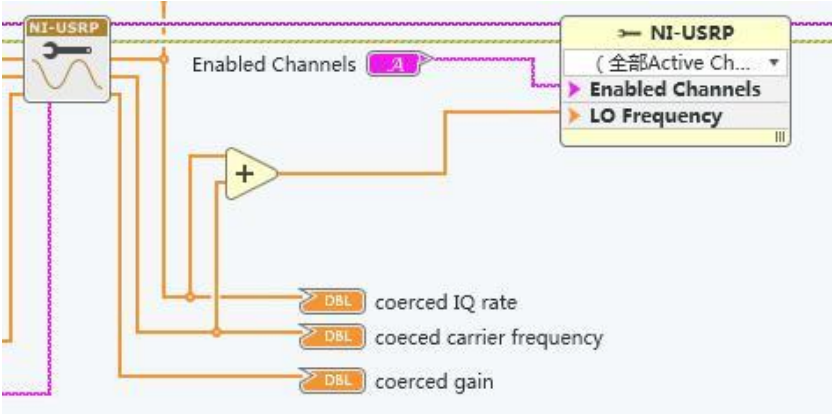


图 4.2.1 参数配置

2.2 接收电路

在上一个实验“RxBPSK.gvi”的基础上完成这部分实验，将该文件重命名为“RxDBPSK.gvi”。

在前面板中设置以下参数：

- ① IQ rate = 400k;
 - ② carrier frequency = 1G;
 - ③ gain = 0;
 - ④ active antenna = RX2;
 - ⑤ Message length = 1000;
 - ⑥ Symbol rate = 10k;
- 1) 计算捕捉两帧数据所需要的采样数，并将它作为“niUSRP Fetch Rx Data”模块 number of samples 参数端的输入;
 - 2) 使用原有电路中的匹配滤波器、卷积、脉冲成形和降采样模块;
 - 3) 需要重新构建一个归一化模块，命名为“norm1D_median.gvi”。与上一个实验中使用最大值进行归一化不同，这里用中值对向量进行归一化。创建一个子 VI，如图 4.2.2 所示，使用它对接收到的符号进行归一化:

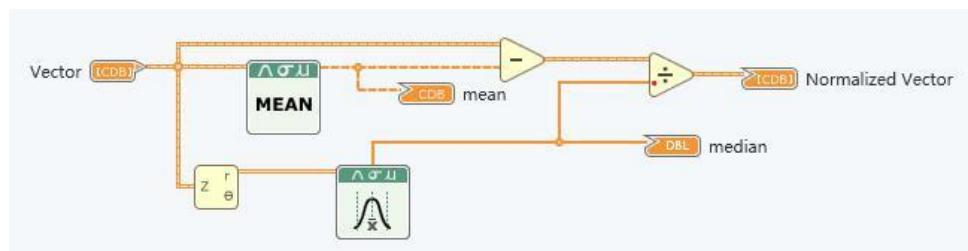


图 4.2.2 归一化电路

- 4) (差分译码) 完成图 4.2.3 所示电路，对归一化符号进行译码:

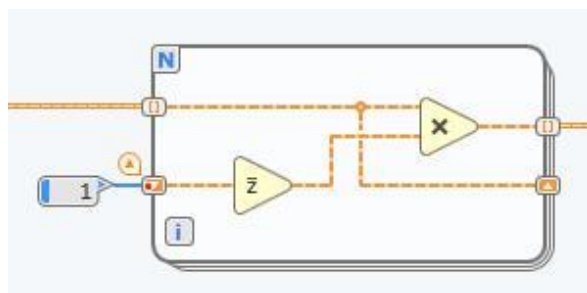


图 4.2.3 差分译码电路

- 5) 获取译码符号的实部数值，使用“FrameSync”模块以及合适的训练序列对数据帧进行同步。阈值建议设置为 18;
- 6) (符号到位比特) 采用与 BPSK 同样的方法，将符号数据映射为比特数据;

- 7) （误比特率）通过与发送的比特数据序列进行比较，来计算误比特率确认译码比特数据的正确性。创建一个和发送电路有相同配置和参数的 PN 序列数据（即：Fibonacci, PN sequence order, total bits, seed in 等）。使用一个 for 循环将译码得到的比特数据与参考数据作比较，计算错误的个数（提示：使用“不等于？”、“布尔值至整数转换”、“加”模块和移位寄存器）。将错误数除以信息长度（Message length）得到误比特率；
- 8) （星座图）在前面板放置一个图形显示控件，将 X 轴名称重命名为 I，Y 轴重命名为 Q。曲线类型选择为“点”，设置自动调整标尺为“无”、标尺最大值设置为 1.5、最小值设置为-1.5。在图形显示控件中显示译码符号数据。

2.3 问题 2

- 1) 给出误比特率和 DBPSK 星座图。运行接收电路一段时间，观察实验结果是否稳定。
- 2) 给出 BPSK 的误比特率和星座图，与 DBPSK 的结果作比较，并解释原因；
- 3) 理解并解释实验步骤 4) 中差分译码的工作原理；

五、实验数据分析与问题回答

① 问题 1：

- 1) 帧同步需要的互相关峰值在哪里？峰值与原始数据帧的起始位置有什么关系？你的“FrameSync.gvi”子 VI 是否产生了同步数据帧？

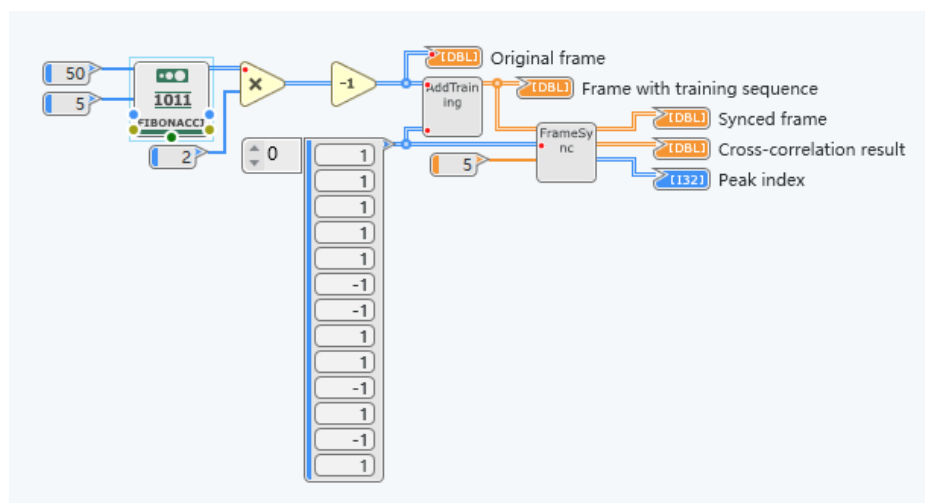


图 5.1.1 帧同步仿真电路

如图 5.1.1，为帧同步电路的仿真电路图，其中 FrameSync 模块的阈值输入设置为 5。运行电路，得到前面板图显示如下：

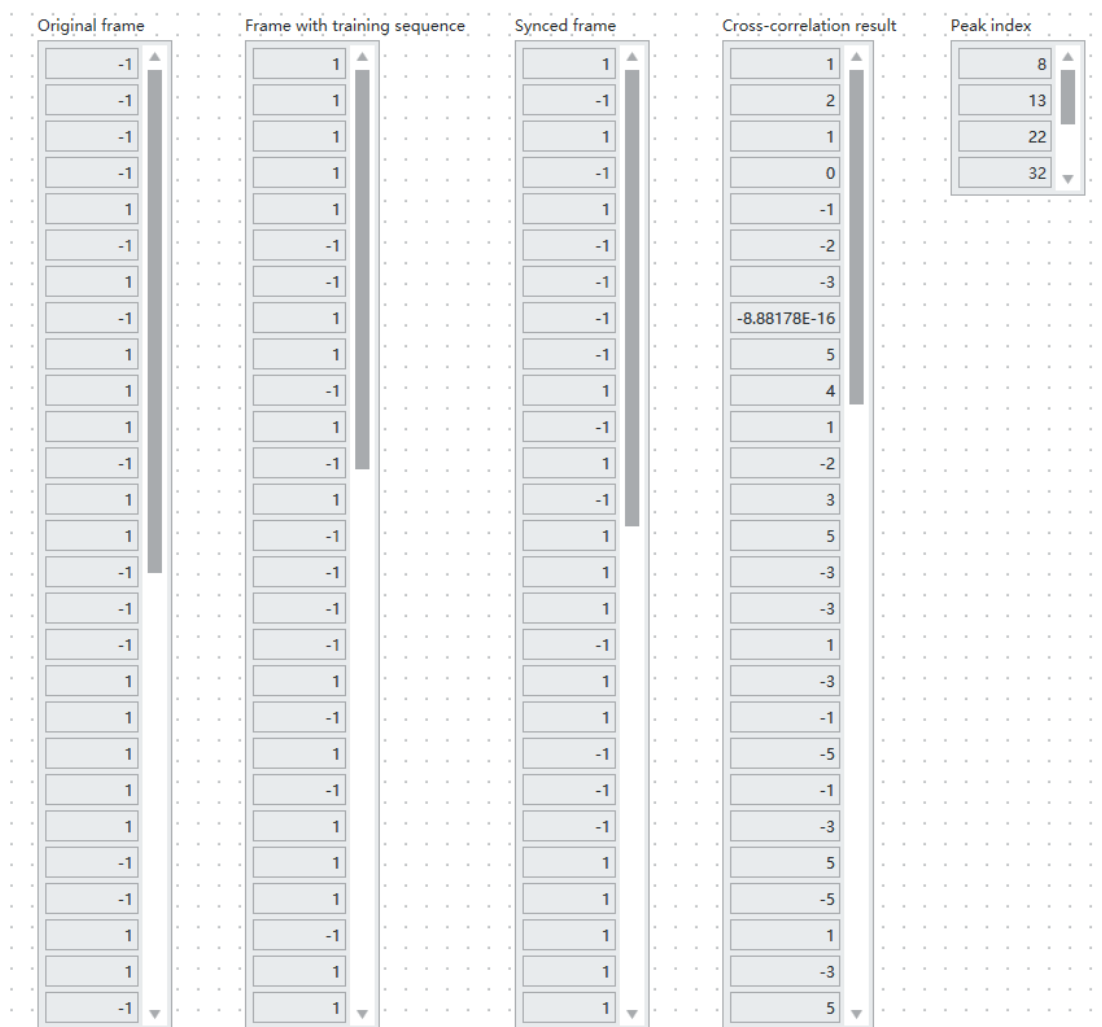


图 5.1.2 前面板显示

比较原始数据帧和加入训练序列后的数据帧，显然，帧同步需要的互相关峰值对应的索引为 13（起始索引为 0），对应的峰值为 5。互相关峰值与原始数据帧的起始位置相对应，当互相关值达到最大时，该位置极有可能就是原始数据帧的起始点。然而，在实验当中，由于第 8 位互相关值也为 5，使得输出的同步数据帧与原始数据帧并不相同，即数据帧同步失败，未产生同步数据帧。

- 2) 一个好的训练序列应该要能给出一个明显的互相关峰值，以便于检测。因此可能需要增加训练序列的长度。Baker 码的最大长度是 13，可以将两个或更多的 Baker 码连接起来。使用 13 位序列创建一个 26 位训练序列，阈值将作何改变？此时，互相关的峰值在哪里？原始数据帧的起始位置在哪里？

将两个 13 位 Baker 码连接后插入到原始数据帧的起始位置，得到电路图如图 5.1.3 所示：

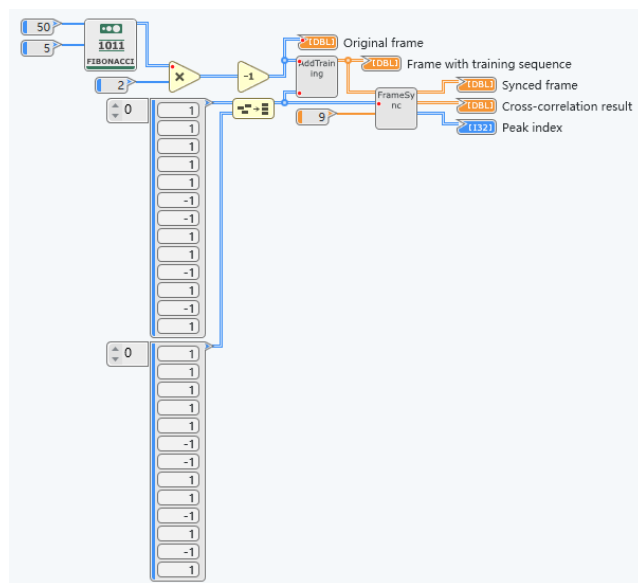


图 5.1.3 26 位训练序列

运行电路，前面板输出如图 5.1.4 所示：

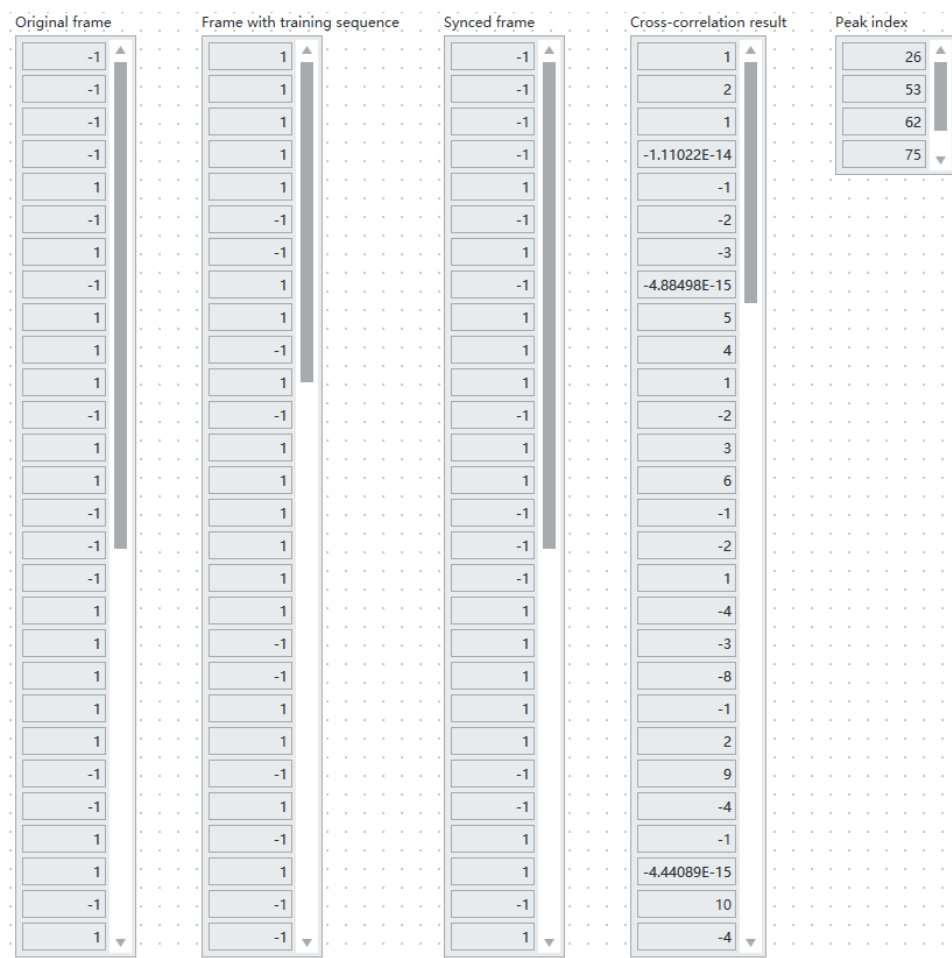


图 5.1.4 前面板显示

此时的互相关峰值为 10，对应的索引位置为 26，原始数据帧的起始索引也为 26。

因此，需要将 FrameSync 模块的阈值输入增大，此处设置为 9，得到的同步数据帧与

原始数据帧完全一致，帧同步成功。

3) 长训练序列的优点和缺点是什么？

根据上述实验，当使用长训练序列后，我们可以得到一个更加明显的互相关峰值，更利于准确检测原始数据帧的起始位置；然而，增加训练数据长度会导致发送更多数据，使得冗余信号增多，信号功率效率下降。

② 问题 2:

1) 给出误比特率和 DBPSK 星座图。运行接收电路一段时间，观察实验结果是否稳定：

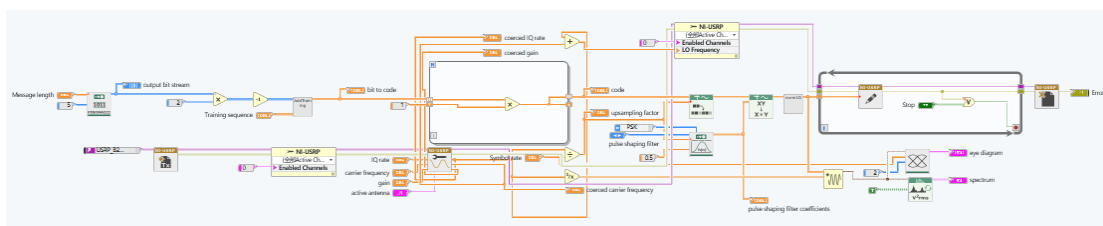


图 5.2.1 发送电路程序框图

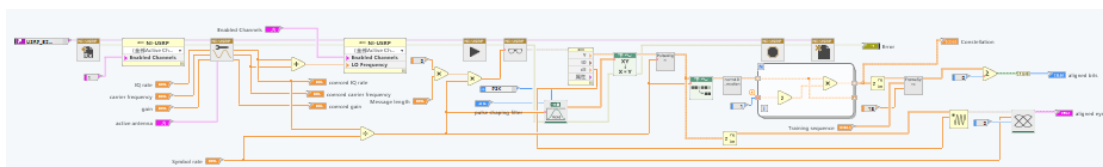


图 5.2.2 接收电路程序框图

如图 5.2.1 和 5.2.2，分别为 DBPSK 的发送、接收电路程序框图。运行电路，得到接收信号星座图和判决后的序列如下：

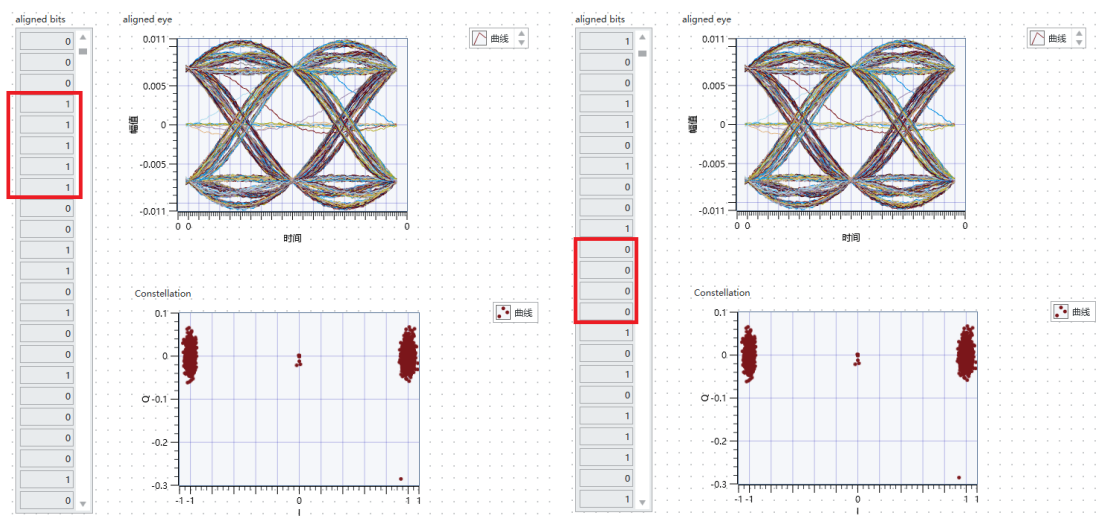


图 5.2.3 DBPSK 接收信号

可以看到，接收信号的星座图基本分布在 $(-1,0)$ 和 $(1,0)$ 两点附近，比较集中，性

能较为稳定；眼图的线迹也比较清晰，张开度较好。观察接收到的序列，可以从中找到五个“1”连续出现，如图 5.2.3 左图所示；以及四个“0”连续出现，如图 5.2.3 右图所示。多次运行接收电路，所得结果都比较稳定。因此，使用 DBPSK 能够很好的实现数据的传输，系统可靠性较高。

2) 给出 BPSK 的误比特率和星座图，与 DBPSK 的结果作比较，并解释原因：

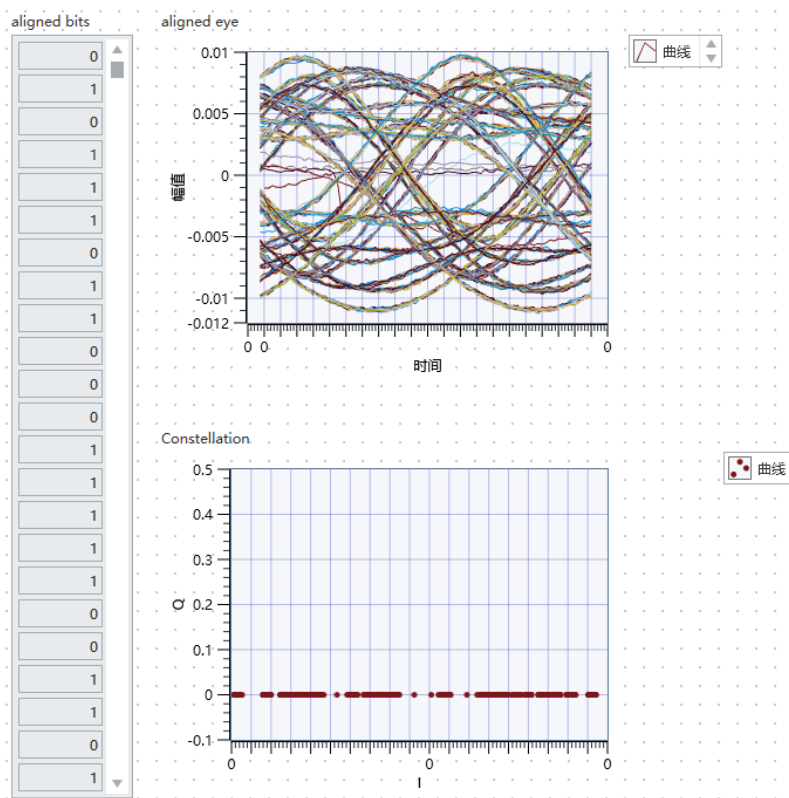


图 5.2.4 DBPSK 接收信号

如图 5.2.4，为利用上一个实验的 BPSK 电路接收到的信号星座图。可以看到，与 DBPSK 相比，BPSK 的星座图更为分散，眼图的线迹也比较模糊，张开度明显缩小，表明 BPSK 的准确性和可靠性较 DBPSK 更低。这是因为在 BPSK 的实现当中，我们并没有加入训练序列和帧同步，当数据序列较长时容易出现错误；另一个方面，BPSK 前后两位符号之间基本可以认为相互独立，易受到噪声干扰，而 DBPSK 传输的符号不仅取决于现在的位信息，还会与前一个位信息有关，其并不依赖于某一固定的载波相位参考值，只要前后位的相对相位关系不被破坏，即可根据此关系正确恢复信息，从而更加准确、稳定。

3) 理解并解释实验步骤 4) 中差分译码的工作原理：

根据表 5.1 所示的编码方式以及差分译码电路，在进行差分译码时，只需要将接收到的符号与前一位符号取共轭后相乘，即可得到输入符号。在进行差分编码时，也是将输入符号与前一符号相乘，得到编码后的符号。显然，两者存在异曲同工之妙，可以比较快捷、方便地实现编/译码工作。

表 5.1 差分编码参考

输入符号	-1	-1	1	1
前一符号	-1	1	-1	1
当前符号	1	-1	-1	1

装
订
线