### DQPSK调制解调实验

**一、实验目的**

1. DQPSK差分码编码及译码的原理；
2. 掌握DQPSK调制和解调的原理及实现方法。

**二、实验仪器**

1. RZ9681实验平台
2. 实验模块：

* 主控模块
* 基带信号产生与码型变换模块-A2
* 信道编码与频带调制模块-A4
* 纠错译码与频带解调模块-A5

1. 信号连接线
2. 100M双通道示波器
3. PC机（二次开发）

**三、实验原理**

**3.1 DQPSK调制**

在前面实验中进行QPSK和OQPSK实验时，我们都观察到相位模糊的情况，在相位模糊情况下，无法解调出正确的数据，这在实际通信系统中是不符合要求的，但是要如何解决这个问题呢？在本节实验内容中，将结合差分编码的理论，研究如果解决QPSK调制解调中的相位模糊情况。

在实际通信系统中，会采用DQPSK的编码方法解决相位模糊的现象。DQPSK又叫四相相对相移键控。QPSK调制具有固定的参考相位，它是以四进制码元本身的相位值来表示信息的，而DQPSK调制没有固定的参考相位，后一个四进制码元总是以它相邻的前一个四进制码元的终止相位为参考相位（或称为基准相位），因此，它是以前后两个码元的相位差值来表示信息的。由于DQPSK传输信息的特有方式，使得解调时不存在相位模糊问题，这是因为不论提取的载波取什么起始相位，对相邻两个四进制码元来说都是等价的，相邻两个四进制码元的相位差与起始相位无关，也就不存在由于相干载波起始相位不同而引起的相位模糊问题，所以，在实际使用中一般都采用相对四相调制。

DQPSK调制中，将输入的二进制序列先经串/并转换分为两路并行数据DI和DQ输出，再对双比特码进行码变换，其后的调制过程与QPSK调制原理框图类似，如下图所示。



图3.4.4.1 DQPSK调制器框图



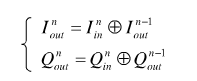
图3.4.4.2 DQPSK相干解调器框图

**3.2 差分码编码原理**

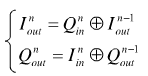
数字相位调制中，若信号以绝对相位来表示携带信息的0和1，则接收机中就需要恢复出一个与发送端相位和频率完全一样的参考载波以便实现正确的解调。但这个参考载波很难做到同频同相，因此接收机容易出现信息的0,1倒置的情况。这个问题的解决可以通过差分编码来实现，差分编码的结果是把绝对相位调制变成相对相位调制，利用载波相位的相对跳变来传递信息，这样一来即使载波恢复时出现相位模糊的情况也不会影响正确解调。在前面的DPSK调制中我们已经学习了2进制的差分编码原理，可以解决相位模糊的情况。

在QPSK调制也有载波恢复的难题，并且情况更复杂一些，但同样可以采用类似的办法解决。在QPSK调制中，差分编码有多种，常用的有自然差分编码和格雷差分（Gary）编码。由于QPSK调制每次传递2个符号，因此有4种相位状态，而差分编码是相对上一次状态的编码，本次输出是4中状态，因此差分编码可能的状态共有16种。对采用正交方式实现的QPSK调制，采用格雷差分编码可以归纳为如下两种情况：

1. 若上次输出满足则此次输出为：



1. 若上次输出满足，则此次输出为：



在上述两式，表示异或运算，表示I路此次输出，表示I路上一次输出，其余类似。经过以上规则的对应，QPSK调制就转变成了DQPSK调制。由上述公式给出的编码关系可以发现格雷码差分编码很容易实现。

在实现时，我们用另外一个角度来看，若把格雷差分编码中的输入做符号对应：

**00->0,01->1,10->3,11->2**

这样一来，格雷差分编码过程就可以看成是一个做模4加法运算。即把上一次的结果与本次的输入先做符号对应，然后再做模4加法运算，得到的结果就是本次的输出。这一概念其实是把二进制差分编码的概念推广到多进制，在DPSK中异或运算其实也是模2加运算，对应的DQPSK就是模4加运算。差分编码结构图如下所示：

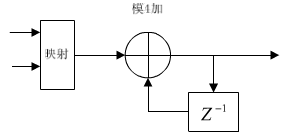


图3.4.4.3 DQPSK差分编码结构图

DQPSK差分编码的译码可按减法原理来获得，其实现框图如下所示：

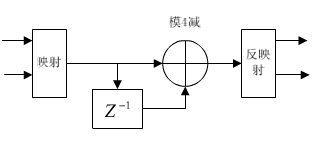


图3.4.4.4 DQPSK差分译码结构图

对于接收端输入为发送端输出，因此有



由译码原理可以得到：







这就是实验中实现DQPSK的差分译码的原理。

下面针对差分编译码举例说明编译码流程，分析其是否解决了相位模糊的问题，其他几种反向情况尝试通过该方法验证：

* 解调1流程对应的正确解调译码过程；
* 解调2流程对应的有相位模糊；

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **原始码** |  | **1 0** | **1 1** | **0 0** | **0 1** | **0 1** | **1 0** | **1 1** | **0 0** |
| **对应符号** | 0 | 3 | 2 | 0 | 1 | 1 | 3 | 2 | 0 |
| **差分编码** | 0 | **3** | 1 | 1 | 2 | 3 | 2 | 0 | 0 |
| **调制输出** | **00** | **10** | 01 | 01 | 11 | 10 | 11 | 00 | 00 |
| ***解调1*** | ***00*** | ***10*** | *01* | *01* | *11* | *10* | *11* | *00* | *00* |
| **对应符号** | **0** | **3** | 1 | 1 | 2 | 3 | 2 | 0 | 0 |
| **对应译码** |  | **3** | 2 | 0 | 1 | 1 | 3 | 2 | 0 |
| **对应解调** |  | **10** | 11 | 00 | 10 | 11 | 01 | 01 | 00 |
| ***解调2*** | ***11*** | ***01*** | ***10*** | ***10*** | ***00*** | ***01*** | ***00*** | ***11*** | ***11*** |
| **对应符号** | **2** | **1** | 3 | 3 | 0 | 1 | 0 | 2 | 2 |
| **对应译码** |  | **3** | 2 | 0 | 1 | 0 | 3 | 2 | 0 |
| **解调输出** |  | **10** | 11 | 00 | 01 | 01 | 10 | 11 | 00 |

采用差分编码后，相位的相对跳变表示传送码的信息，根据相位的相对跳变，解调器可正确解调出对应信息。使用差分编码时需要注意，差分编码是相对编码，因此第一个数据只起参考作用，对编码而言，第一个数据可以随意设定；对信号解调来说第一个数据是无法确定传送的具体信息。下表给出差分编码与相对相位跳变对应关系。

