**信息科学与工程学院**

**2021－2022学年第一学期**

实 验 报 告

课程名称： 通信原理实验

实验名称： PSK系统性能测试

专 业 班 级 通信工程 二班

学 生 学 号 201800121050

学 生 姓 名 孟麟芝

实 验 时 间 2021年4月

### 目 录

[第 1 章 原理分析 2](#_Toc70016438)

[1.1 PSK调制原理 2](#_Toc70016439)

[1.2最佳接收机与最佳检测器 3](#_Toc70016440)

[1.3 PSK系统性能分析 4](#_Toc70016441)

[1.4 k进制格雷码生成原理 6](#_Toc70016442)

[第 2 章 系统设计 7](#_Toc70016443)

[2.1概述 7](#_Toc70016444)

[2.2格雷码生成函数 8](#_Toc70016445)

[2.3格雷码编码函数 8](#_Toc70016446)

[2.4判决子函数 9](#_Toc70016447)

[2.5错误率统计函数 10](#_Toc70016448)

[2.6星座图绘制函数 11](#_Toc70016449)

[第 3 章 系统性能测试 12](#_Toc70016450)

[3.1从星座图看系统性能 12](#_Toc70016451)

[3.2比特数对性能测试准确度的影响 12](#_Toc70016452)

[3.3相关度量法与最小欧式距离法判决 13](#_Toc70016453)

[3.4 QPSK与8PSK的比较 13](#_Toc70016454)

[第 4 章 问题回顾与总结 14](#_Toc70016455)

[4.1问题回顾 14](#_Toc70016456)

[4.1.1 性能改进 14](#_Toc70016457)

[4.1.2 误比特率与误码率的关系 14](#_Toc70016458)

[4.1.3 对复数矩阵的操作 15](#_Toc70016459)

[4.2总结 15](#_Toc70016460)

### 第 1 章 原理分析

#### 1.1 PSK调制原理

我们知道，带通PAM信号是利用载波的不同幅度来传递不同的信息，相移键控则将信息包含在载波的相位中。

根据这一定义，我们给出个载波相位调制信号波形的一般表示形式如下：

其中，是基带脉冲波形，它决定了传输信号的频谱特性，方便起见，本实验中将其定义为一矩形脉冲，即：

相应的传输信号波形为：

将上式中余弦函数的角度看成两个角度之和，则可表示为：

其中：

所以，一个相位已调信号可以看成两个振幅分别为和的正交载波，振幅值取决于每个信号区间内的传输相位。

综上，我们可以得到数字相位已调信号的二维矢量表示形式：

正交基函数为：

将个比特映射到 个相位上的方法有很多，比较好的方法是使用格雷（Gray）编码，这种编码的相邻信号只有一个二进制数字不同，因为由噪声引起的发生概率最大的差错是错误地选择相邻相位作为发送信号相位，而对应于格雷编码中，比特序列中只会产生一个比特差错。

#### 1.2 最佳接收机与最佳检测器

AWGN信道下， 维信号经最佳接收机，得到的接收矢量为：

其中表示信号矢量，表示在信号空间的噪声矢量，噪声矢量的每一维服从分布 ，则。

可以证明，信号空间外的噪声矢量不含有与判决有关的任何信息。

显然，在未得到任何接收信息的情况下，最好的判决方法就是选择具有最高先验概率 的信号，得到接受矢量 后，如何作出最优的判决？其方法即用后验概率取代先验概率，选择使最大的，这个判决准则称为最大后验概率（MAP）准则。

根据贝叶斯公式，后验概率可表示为：

当个信号先验概率相等，则寻找 的最大值，就等价于寻找使最大的信号。条件概率密度或它的任何单调函数称为似然函数，在个信号上寻求最大的判决准则称为最大似然（ML）准则。各信号先验概率相等的情况下，MAP准则与ML准则是等价的。

由公式（8）可得到条件概率密度函数的表达式：

取自然对数，得：

寻找使 最大的信号，等价于寻找取得最小欧氏距离的信号，则有如下公式：

称为距离度量，因此对于AWGN信道，基于ML准则的判决简化为了寻求在距离上最接近接受矢量 的信号矢量。

将距离度量展开，可得：

其中， 项对所有的判决度量是相同的，可忽略，则得到相关度量：

那么寻求距离度量最小的信号矢量，等价于寻求相关度量最大的信号。即相关度量与距离度量是等价的。

#### 1.3 PSK系统性能分析

下面，我们讨论采用最佳接收机和最佳检测器时，在AWGN信道上传输的进制相位调制的错误概率。运算过程中我们采用相位度量为基础的最佳检测。

假设发送信号的相位 ，发送信号矢量为：

接受信号矢量为：

其中 ， 均服从分布 ，则有：

相干相位检测的度量是相位 ，进行变量代换：

由此可得：

将上式对积分，并记，可得到，即：

当发送信号时，如果由于噪声干扰而使相位超出的范围，就会导致错判，由此可得码元错误概率为：

一般情况下，该积分都不能化为一个简单函数，但在时，该式有较为简单的表达形式。记 ，则有：

值得注意的是，四相调制的误比特率与二相调制相同，均可由公式（22）计算得出，这是因为四相调制实际上可以看成两对正交相位的二相调制。对每一路二相调制，其导致的误比特率均为，则总体的误比特率为。

当时，该积分只能由数值积分法得出，下面直接给出大信噪比情况下的近似表达式：

#### 1.4 k进制格雷码生成原理

这一节中的讨论里使用了矩阵，仅是为了方便表示，并不具有实际的数学意义。

首先，我们给出一对单比特格雷码作为“基”，用于后面生成进制格雷码（每个格雷码含有比特）：

将两个矩阵横向拼接：

向第一列补’0’，第二列补’1’：

将第二列上下翻转后纵向拼接到第一列下方，即可得到二进制格雷码：

再将作为基，重复上面的操作，即可生成三进制格雷码，不断循环，可生成任意进制的格雷码。

### 第 2 章 系统设计

#### 2.1 概述

前面已经对PSK调制与解调原理及性能进行了详尽的介绍，下面我们要利用Matlab对该系统进行仿真，系统的原理框图如下：



图2. 1 系统原理框图

首先，可以利用randi函数生成长为的随机数序列作为信源，将信源每比特为一码元进行PSK调制（包含格雷码映射过程）。由于PSK调制后为二维信号，编程时方便起见，我们可以利用复数代表每个信号点。再利用awgn函数这向这个信号点的实部虚部同时加以高斯白噪声，便得到了接收信号，再对接收信号进行判决即可。

这一过程的代码实现如下，其思路已在注释中详尽描述，其中randi、pskmod、awgn函数是Matlab中内置的函数，它们能够满足本实验的需求，其他各个子函数的功能及实现方式将在后面给出。

|  |  |
| --- | --- |
|  | M=8;                            %MPSK进制数选择 |
|  | flag=1;                         %判决方式选择 |
|  | k=log2(M); |
|  | sigma=sqrt(20);                 %噪声方差 |
|  | SNR=10\*log10((1/k)/(2\*sigma^2)); |
|  | L=1000; |
|  | table=generateGrayCode(k);      %码表生成 |
|  | an=randi([0,1],1,L\*k);          %信源序列 |
|  | source=grayEncoder(an,k,table); %格雷码映射 |
|  | MPSK=pskmod(source,M);          %PSK调制 |
|  | MPSK=awgn(MPSK,SNR);            %通过AWGN信道 |
|  | result=decision(MPSK,M,flag);   %判决 |
|  | [bitError,symbolError]=getErrorRate(result,an,source,table); |

#### 2.2 格雷码生成函数

依照前面给出的原理，我们可编写如下代码生成格雷码，该函数将返回一个码表table便于使用。

最初设计该函数时考虑到需要查表的问题，使用了cell类型，将格雷码使用字符串存放，利用strcmp函数配合find函数可很方便完成查找。但由于对Matlab字符串操作较为缓慢，故最终改为了矩阵形式，这样一来，尽管查表时使用了较“傻瓜”的for循环，但仍使得程序速度有了质的飞跃。将100000个比特通过系统，原先需要5分钟左右，现只需15秒。

|  |  |
| --- | --- |
|  | function table=generateGrayCode(k) |
|  | % ------------------------------------ |
|  | % 格雷码生成器 |
|  | % k为每码元比特位数 |
|  | % ------------------------------------ |
|  | base=[0;1]; |
|  | joint=base; |
|  | for m=1:k-1 |
|  | joint=[zeros(size(joint,1),1),joint,ones(size(joint,1),1),joint]; |
|  | joint=[joint(1:size(joint,1),1:size(joint,2)/2);flipud(joint(1:size(joint,1),size(joint,2)/2+1:size(joint,2)))]; |
|  | end |
|  | table=joint; |
|  | end |

#### 2.3 格雷码编码函数

编码过程原理较为简单，即将信源序列每比特划分为一码元，将该码元与码表中的码字一一比较，从而将这一比特序列映射为格雷码码元。

其中查表比较有两种方法，一种是利用for循环逐个比较，查到后即利用break退出循环，再对下一个码元进行映射；第二种是将这一比特序列与码表矩阵相减，利用find函数寻找全零行。两种方法在时间复杂度上基本一致，下面采用的是for循环方法。

|  |  |
| --- | --- |
|  | function symbol=grayEncoder(an,k,table) |
|  | % ------------------------------------ |
|  | % 将比特序列使用格雷码编码 |
|  | % an为待编码的比特序列 |
|  | % k为每码元比特位数 |
|  | % table为格雷码编码表 |
|  | % ------------------------------------ |
|  | m=1; |
|  | symbol=zeros(1,length(an)/k); |
|  | for i=1:k:length(an)-k+1 |
|  | judge=an(i:i+k-1)-table; |
|  | [line,row]=size(judge); |
|  | for j=1:line |
|  | if judge(j,:)==zeros(1,row) |
|  | symbol(m)=j-1; |
|  | break |
|  | end |
|  | end |
|  | m=m+1; |
|  | end |
|  | end |

#### 2.4 判决子函数

相关度量与距离度量的原理及等价性在原理分析中已经叙述，在此不再赘述。而在程序实现方面，有一些需要注意的技巧。

首先，两种方法都需要先利用公式（6）生成一组发送信号

对于最小欧氏距离法，我们只需将输入序列进行转置（对复数矩阵转置时应使用“.’”，若使用“’”将进行共轭转置），将转置后的序列减去 矩阵，再对差值取模，即得到了每个码元对各个发送信号的距离度量。利用min函数寻找每个码元的距离度量的最小值，可完成判决。

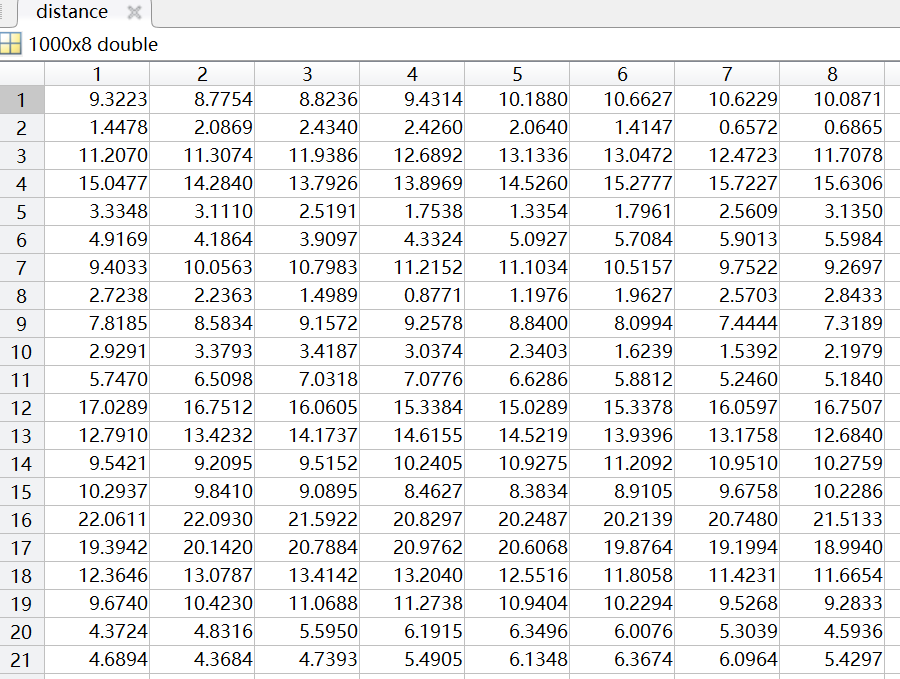


图2. 2 每个码元对各个发送信号的距离度量

对于相关度量法，采用了复数代表向量， 与的数量积运算该如何用复数运算代替呢？下面我们将进行推导。

设：

则有：

即要求数量积，只需将与共轭复数的乘积取实部即可。

判决子函数的程序实现如下：

|  |  |
| --- | --- |
|  | function result=decision(input,M,flag) |
|  | % ------------------------------------ |
|  | % 对码元序列进行判决 |
|  | % input为待判决的序列 |
|  | % M为所需生成发送信号的个数 |
|  | % flag为判决方法选择标志，0为最小欧式距离法，1为相关度量法 |
|  | % ------------------------------------ |
|  | m=0:1:M-1; |
|  | sm=cos(2\*pi\*m/M)+1j\*sin(2\*pi\*m/M); |
|  | if flag==0 |
|  | %最小欧氏距离检测法 |
|  | distance=abs(input.'-sm);   %获取距离度量 |
|  | [~,position]=min(distance');%寻求距离度量最小值 |
|  | end |
|  | if flag==1 |
|  | %最大相关度量检测法 |
|  | C=real((input.').\*conj(sm));%获取相关度量 |
|  | [~,position]=max(C');       %寻求相关度量最大值 |
|  | end |
|  | result=position-1; |
|  | end |

#### 2.5 错误率统计函数

这一函数的实现较简单，在此只给出代码，不予分析。

|  |  |
| --- | --- |
|  | function [bitError,symbolError]=getErrorRate(result,an,source,table) |
|  | % ------------------------------------ |
|  | % 对码元序列进行判决 |
|  | % result为抽样判决结果 |
|  | % an为发送比特序列 |
|  | % source为所发送码元序列 |
|  | % table为格雷码编码表 |
|  | % ------------------------------------ |
|  | L=length(source); |
|  | k=log2(length(table)); |
|  | symbolError=length(find(result~=source))/L; |
|  | for i=1:1:L |
|  | bit((i-1)\*k+1:i\*k)=table(result(i)+1,:); |
|  | end |
|  | bitError=length(find(bit~=an))/(L\*k); |
|  | end |

#### 2.6 星座图绘制函数

Matlab内部提供的scatterplot函数运行较快，但无颜色区分，下面这一函数将不同信号点用不同颜色进行了区分，但运行速度稍慢，且较难进行优化，实现代码如下：

|  |  |
| --- | --- |
|  | function constellation(source,MPSK) |
|  | % ------------------------------------ |
|  | % QPSK星座图绘制 |
|  | % source为发送码元序列 |
|  | % MPSK为通过AWGN信道后的抽样输出 |
|  | % ------------------------------------ |
|  | figure |
|  | for i=1:1:length(source) |
|  | switch(source(i)) |
|  | case 0 |
|  | scatter(real(MPSK(i)),imag(MPSK(i)),'r','.') |
|  | hold on; |
|  | case 1 |
|  | scatter(real(MPSK(i)),imag(MPSK(i)),'g','.') |
|  | hold on; |
|  | case 2 |
|  | scatter(real(MPSK(i)),imag(MPSK(i)),'b','.') |
|  | hold on; |
|  | case 3 |
|  | scatter(real(MPSK(i)),imag(MPSK(i)),'m','.') |
|  | hold on; |
|  | end |
|  | end |
|  | end |

### 第 3 章 系统性能测试

#### 3.1 从星座图看系统性能

设噪声方差 ，使系统通过1000个比特，观察信道输出形成的四张星座图的区别。

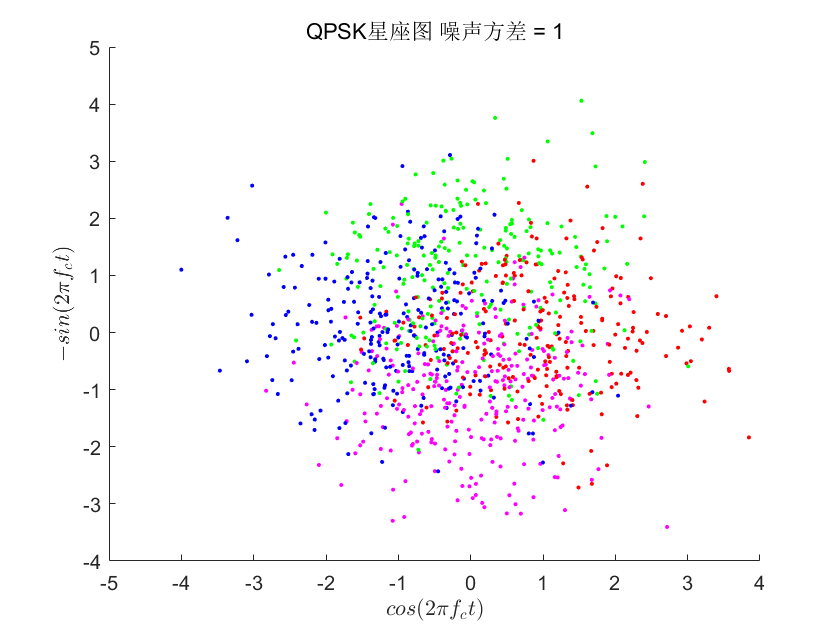
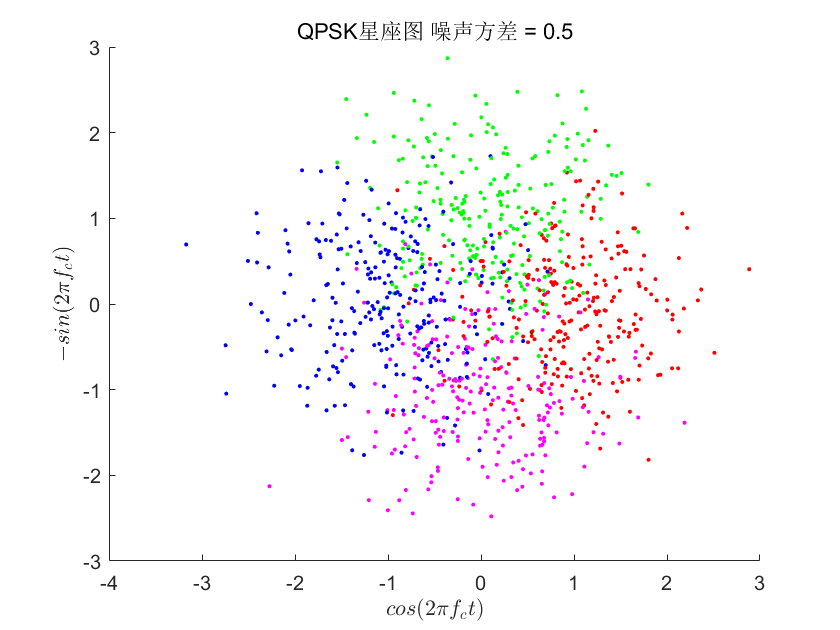
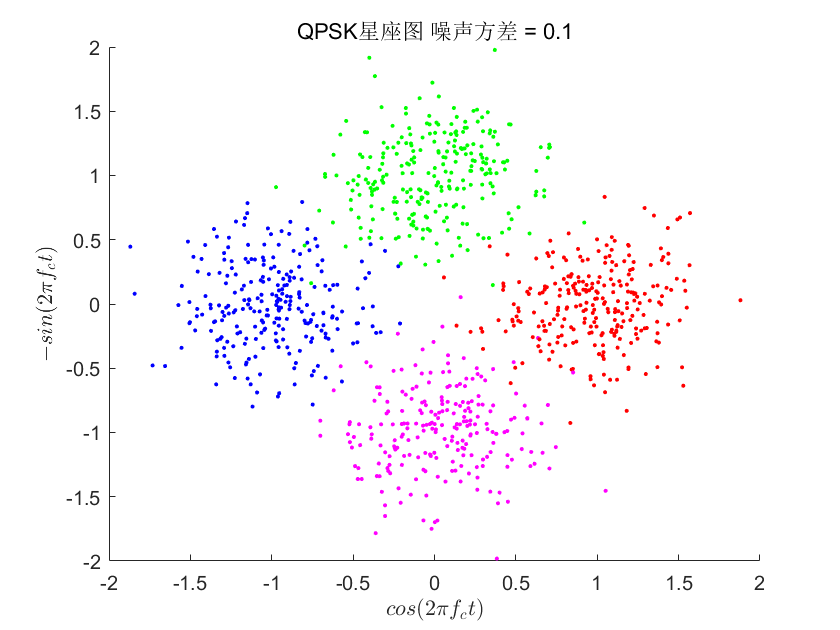
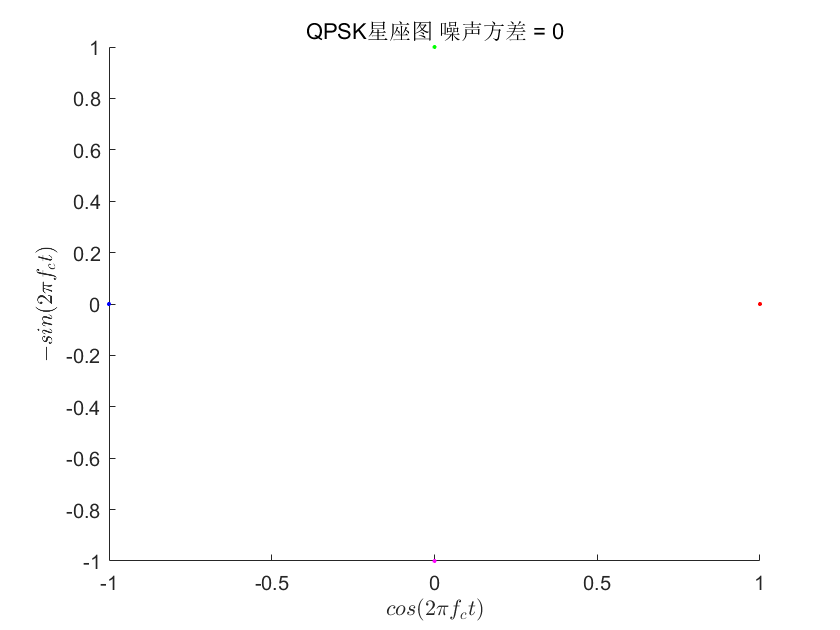


图3. 1 噪声方差对星座图的影响

从上图中可以看出，噪声方差越大，信号的信噪比越低，信号点越分散。反之，噪声方差越小，信号点越集中。

#### 3.2 比特数对性能测试准确度的影响

由于使用的是Monte Carlo仿真方法，随机测试数据点越多，越能显现出系统性能的原貌。

在AWGN信道下，利用最大投影点准则（相关度量）进行判决，分别画出比特数为1000、10000、100000时的仿真误比特率曲线和理论误比特率曲线，比较其区别：



图3. 2 比特数对性能测试准确度影响

从上图可见，随着数据点变多，曲线更加平滑，更加贴合理论值。另外在数据点较少而较大时，可能会出现误比特率为0的情况，这些点无法在图上正确显示出来，随着数据点增多，在较大下更准确的误比特率也得以呈现。

#### 3.3 相关度量法与最小欧式距离法判决

在1.2节中我们已经证明，相关度量法及最小欧氏距离法这两种判决都是基于ML准则的判决方法，本质上是等价的，下面传输100000个比特，比较使用两种判决方法得到的误比特率曲线，来直观的体会二者的等价性：



图3. 3 相关度量法与最小欧氏距离法比较

从图中我们可以看出，二者的性能是没有区别的，也可间接证明两种方法是等价的。

#### 3.4 QPSK与8PSK的比较

对QPSK与8PSK的误码率、误比特率计算可见公式（22）~（24）及其相关推导，在此不再赘述。接下来，我们采用最小欧氏距离法进行判决，传输100000个比特，比较二者的性能区别如下：



图3. 4 QPSK与8PSK性能比较

从图中我们可以看出，QPSK性能要优于8PSK，随着进一步增大，MPSK系统性能将逐渐降低。但更高阶的PSK系统能带来更高的带宽利用率。

### 第 4 章 问题回顾与总结

#### 4.1 问题回顾

4.1.1 性能改进

最初设计格雷码编译码时，考虑到需要查表的问题，使用了cell类型，将格雷码使用字符串存放，利用strcmp函数配合find函数可很方便完成查找。但由于对Matlab字符串操作较为缓慢，故最终改为了矩阵形式，这样一来，尽管查表时使用了较“傻瓜”的for循环，但仍使得程序速度有了质的飞跃。将100000个比特通过系统，原先需要5分钟左右，现只需15秒。

另外，在Matlab中进行欧氏距离计算、相关度量计算、错误概率统计等，都不需要使用for循环，使用矩阵运算将有效提高运行速度，也使代码更加简洁，具体方法已在2.4与2.5节中给出。

4.1.2 误比特率与误码率的关系

对于高阶的PSK系统，在较小 下，格雷码映射的误比特率不能再使用 来计算，不然会导致很大偏差。因为这时某个码元被误判时，大概率被误判为邻居码元的假设不再成立，需要更精确的计算方法。



图4. 1 32PSK误码率与误比特率测试

W. Jeong等人在文章《New BER Expression of MPSK》中给出了一种新的MPSK系统BER计算方法，此文章研究了AWGN信道及衰落信道下MPSK系统的性能，作者提出的方法还可扩展到非格雷码映射的MPSK系统BER计算。此文应当可以解决这里提到的问题，但由于该方法较为复杂，本实验中未对其进行仿真实现。

4.1.3 对复数矩阵的操作

该实验中为方便起见，将二维向量用复数进行表示，大大简化了编程工作，但要注意的是复数矩阵的一些操作与实数矩阵不同。如转置运算符为“.’”，而“’”为共轭转置运算符。可见随计算数值类型不同，运算符意义也可能不同，使用时要格外留意。

#### 4.2 总结

通过本实验，在原理理解方面，我们对MPSK系统有了更直观、深入的认识，加深了对一些公式来龙去脉的印象。在对高阶PSK系统误比特率的分析时，通过查阅文献，了解到了更加准确的BER计算方法。课本中的公式、假设是否准确？近似公式的误差究竟有多少？在日常学习中并不容易得到重视，该文章的工作则令我感悟颇深，意识到了自己所学的肤浅，更应继续钻研。

编程方面，本实验由于需要进行蒙特卡洛仿真，其准确度依赖于测试点数，代码运行效率便显得尤为重要。结合MathWorks提供的详尽的官方文档，做出了一些优化代码的尝试。对复数矩阵的应用、如何减少for循环的思考、对cell元胞数组的尝试，都令我大开眼界，相信这些看似微不足道的工作，都将为今后的学习工作提供帮助。