**《通信原理》实验报告**

|  |  |
| --- | --- |
| 实验名称： | FSK通信系统设计实验 |
| 院（系）： | 电子信息与通信学院 |
| 专业班级： |  |
| 姓名： |  |
| 学号： |  |
| 时间： | 2020.12.23 |
| 地点： | 南一楼中222 |
| 实验成绩： |  |
| 指导教师： |  |

2020 年 12 月 23 日

1. **实验任务**

#### 1.1 MATLAB设计性实验

利用卷积编码、FSK调制和前导码等技术构建通信系统，学习发射机结构，实现发射机代码，完成卷积编码、FSK调制；学习其接收机结构，实现接收机代码，完成接收信号的滤波、FSK解调、定时同步和卷积码译码。

通过该FSK系统实验，进一步认识通信系统的结构及其处理流程，同时掌握FSK调制解调方法。

#### USRP实验

在发送代码中加入SendBit、phase(SendSig)波形的绘图函数，用于观察发射波形；在接收代码中加入RecvSig、RecvFskDemod、RecvCorr、RecvSymbolSampled、RecvBit的绘图函数，用于观察接收波形。最后，观察误码率需要绘制RecvBit-SendBit的波形，当误码率为0时，该波形应该为一条恒定为0的直线。

完成代码嵌入后，运行Labview程序，观察实验现象。当接收结果与数据源载入的比特流完全一致时，表示误码率为0，实验成功

1. **实验基本原理**

#### 2.1 发射机结构

FSK通信系统发射机图1所示，具体步骤如下：



图 1 发射机结构

（1）随机信源比特从指定数据文件中读取。

（2）对二进制序列进行卷积编码，编码器参数是[171,133]，编码约束长度是7，编码前在信息比特的末尾添加6个0作为结尾比特。

（3）在编码比特之前插入前导码，前导码由16个固定比特组成，用于接收机的定时同步。

（4）进行FSK调制。

（5）最后将信号送往发射电路发射。

#### 2.2 接收机结构

FSK通信系统接收机如图2所示，具体步骤如下



图 2 接收机结构

（1）首先对来自接收电路的信号的载波1和载波2进行滤波。

（2）对两路滤波输出的幅度相减。

（3）通过搜索前导码，确定第一个数据码元的时间位置。

（4）对解调信号进行抽样，得到码元抽样序列。

（5）送入卷积码译码器译码，得到接收比特序列，译码采用matlab函数vitdec，译码结果要去掉6个尾比特。

#### 2.3 关键信号

SendBit：发送的信源比特序列

SendSig： FSK已调信号

RecvFskDemod：FSK解调信号

RecvCorr：前导码相关搜索结果

RecvSymbolSampled：码元抽样

RecvBit：恢复的数据比特

#### 2.4 关键参数

系统参数（不可更改）：

Fs = 200kHz，系统采样率

Rs = 10k码元/秒，码元速率

SigLen = 200k，发射信号SendSig的采样点数

信道参数：

Amax = 1，最大信号幅度

Pmax = pi，最大相位偏差

Fmax = 128，最大频率偏差，单位Hz

Tmax = 0.005，最大时间偏差，单位秒

SNR = -3，信噪比

1. **模块设计与实现**

#### 3.1 发送端：FskSysTx.m

1. **参数设置：**
   1. 代码：

UpSampleRate=Fs/Rs;

CoderConstraint = 7;%  约束长度

Preamble=[1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 0];

* 1. 说明：

设置升采样率为系统采样率/码元速率，约束长度为7（编码器中产生延迟的移位寄存器的个数+1），并约定了前导码。

1. **生成信源比特：**
   1. 代码：

% generate random message

load(SFile); %生成信源比特

MsgLen = length(SendBit);

1. **卷积编码：**
   1. 代码：

% convolutoinal coding

trel = poly2trellis(CoderConstraint, [171, 133]);

SendBitWithTail = [SendBit, zeros(size(1 : CoderConstraint - 1))];%  结尾处理, 在消息的结尾添加 coder\_constraint-1 个零

code = convenc(SendBitWithTail, trel);

%  调用库函数所生成卷积码

* 1. 说明：

poly2trellis ：将卷积码的码多项式转换为网格结构。常用函数格式为：

trellis = poly2trellis(ConstraintLength,CodeGenerator)

输入参数1——ConstraintLength，约束长度，=编码器中产生延迟的移位寄存器的个数+1；

输入参数2——CodeGenerator，用于表示编码器输入输出关系的码多项式；

输出参数——trellis，用于表示编码器输入输出关系的网格结构。

convenc：卷积码的编码（二进制数据）。常用函数格式为：

code = convenc(msg,trellis)

输入参数1——msg，未编码的信息符号序列，二进制矢量形式；

输入参数2——trellis，卷积码编码器的网格结构；

输出参数——code，编码后的卷积码符号序列，二进制矢量形式。

1. **加入前导码：**
   1. 代码：

% add preamble

data=[Preamble,code];

* 1. 说明：

添加前导码。

1. **FSK调制：**
   1. 代码：

% FSK modulation

%键控法实现

f\_1=Rs;%频率1

f\_2=3\*Rs;%频率2

t=(0:length(data)\*UpSampleRate-1)/Fs;%每个码元的采样点数

carrier\_1=exp(j\*(2\*pi\*f\_1\*t));%载波1

carrier\_2=exp(j\*(2\*pi\*f\_2\*t));%载波2

SendFskUp=zeros(1,length(data)\*UpSampleRate);%初始化

for iBits=0:(length(data)-1)

    for i=(iBits\*UpSampleRate+1):(iBits+1)\*UpSampleRate

        SendFskUp(i)=data(iBits+1);

    end

end

SendFskUp1=~SendFskUp; %反相器

SendSig = SendFskUp.\*carrier\_1+SendFskUp1.\*carrier\_2;%相加器

* 1. 说明：

先得到两个载波，载波1频率为Rs，载波2频率为3\*Rs。

采用键控法实现调制：先得到第一个单极性脉冲序列，再通过反相器得到第二个单极性脉冲序列。

再通过选通键控得到两个2ASK信号，最后再通过相加器将两个信号相加，得到2FSK信号。

1. **根升余弦滤波：**
   1. 代码：

% RRC filtering

filterDef=fdesign.pulseshaping(UpSampleRate,'Square Root Raised Cosine','Nsym,Beta',FilterSymbolLen,Rolloff);

myFilter = design(filterDef);

myFilter.Numerator=myFilter.Numerator\*UpSampleRate;

SendSig = conv(myFilter.Numerator,SendBpskUp);

* 1. 说明：

这里：SendSig 就是基带信号：



这种波形主瓣就跨了两个码元周期，故波形是重叠的。

另外滤波器的长度为120，滤波（conv）后引入了120/2的延时，加上SendBpskUp的数据是从20（可以从一开始）开始的，故第一个采样点在80。

这些引入的延时不用管它，在接收端通过搜索前导符可以去掉。

1. **确保信号长度是SigLen：**
   1. 代码：

% 确保信号长度是SigLen

if length(SendSig)>SigLen

    SendSig = 0.7\*SendSig(1:SigLen);

elseif length(SendSig)<SigLen

    SendSig = 0.7\*[SendSig zeros(1,SigLen-length(SendSig))];

end

* 1. 说明：

若长度不足SIgLen，则填充“0”。

**3.2 接收端：FskSysRx.m**

1. **参数设置：**
   1. 代码：

UpSampleRate=Fs/Rs;

Ts=UpSampleRate; % symbol period

SymbolLen = (MsgLen+6)\*2;

CoderConstraint = 7;%  约束长度

Preamble=[1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 0];

PreambleSig=[-1 -1 -1 -1 +1 -1 +1 -1 -1 +1 +1 -1 +1 +1 +1 +1];

PreambleLen = 16;

RecvFskDemod = 0;

RecvCorr = 0;

RecvSymbolSampled = 0;

RecvBit = 0;

* 1. 说明：

设置升采样率为系统采样率/码元速率，SymbolLen为所有比特数据个数的两倍，约束长度为7（编码器中产生延迟的移位寄存器的个数+1），并定义了和发送端一致的前导码。

1. **滤波器设计：**
   1. 代码：

% filter design

fd1 = fdesign.bandpass('N,F3dB1,F3dB2',10,1,2\*Rs,Fs);

filter1 = design(fd1,'butter');

fd2 = fdesign.bandpass('N,F3dB1,F3dB2',10,2\*Rs,4\*Rs,Fs);

filter2 = design(fd2,'butter');

fd3 = fdesign.lowpass('N,F3dB',10,Rs,Fs);

filter3 = design(fd3,'butter');

t = (0:length(RecvSig)-1)/Fs;

* 1. 说明：

为下面的FSK解调设计滤波器。使用fdsign快速设计，该函数返回一个滤波器设计规范对象，该对象包含滤波器的规范，如通带频率、停止带频率、通带纹波和滤波器顺序。使用设计功能从滤波器设计规范对象设计过滤器。

下面是各个滤波器的设计：

滤波器1：中心频率为载波1频率的带通滤波器：

* 参数填充方式设为：'N,F3dB1,F3dB2'
* 滤波器阶数设置为10。也可以设置的更高一些。
* 第一个通带频率（First 3dB Frequency）设置为 1。
* 第二个通带频率（First 3dB Frequency）设置为 2\*Rs。

Fs must be specified as a scalar trailing the other numerical values provided.

滤波器2：中心频率为载波2频率的带通滤波器：

* 参数填充方式设为：'N,F3dB1,F3dB2'
* 滤波器阶数设置为10。也可以设置的更高一些。
* 第一个通带频率（First 3dB Frequency）设置为 1。
* 第二个通带频率（First 3dB Frequency）设置为 2\*Rs。

Fs must be specified as a scalar trailing the other numerical values provided.

滤波器3：为了模拟包络检波效果的低通滤波器。

* 参数填充方式设为：'N,F3dB'
* 滤波器阶数设置为10。也可以设置的更高一些。
* 通带频率（3dB Frequency）设置为 Rs。

Fs must be specified as a scalar trailing the other numerical values provided.

1. **FSK解调：**
   1. 代码：

% filtering of carrier1

coss1 = exp(1j\*2\*pi\*t\*Rs);

signal1 = filter(filter1, RecvSig);

signal1 = signal1 .\* conj(coss1);

signal1 = filter(filter3, signal1);

% filtering of carrier2

coss2 = exp(1j\*2\*pi\*t\*3\*Rs);

signal2 = filter(filter2, RecvSig);

signal2 = signal2 .\* conj(coss2);

signal2 = filter(filter3, signal2);

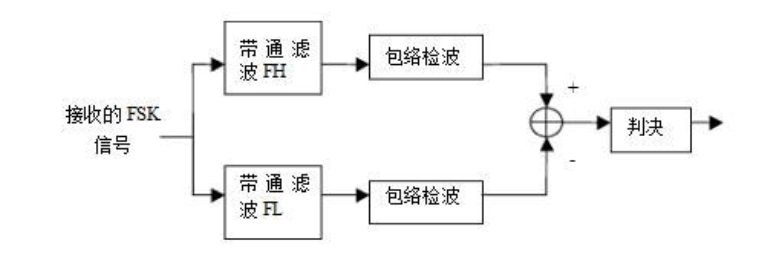
% FSK demodulation

RecvFskDemod = abs(signal1) - abs(signal2);

* 1. 说明：

对于FSK信号的解调方式很多:相干解调、滤波非相干解调、正交相乘非相干解调。

而FSK的非相干解调一般采用滤波非相干解调，解调原理是将2FSK信号分为上下两路2ASK信号分别进行解调，然后判决，这里的抽样判决是直接比较两路信号抽样值的大小，可以不专门设置门限。判决则应与调制规则相呼应，调制时若规定“1”符号为对应载波频率f1,则接受时上支路的样值大小，应判为“1”反之则判为“0”。程序框图如下所示：



1. **前导码搜索：**
   1. 代码：

% preamble search

RecvCorr = zeros(1,floor(length(RecvFskDemod)/3));

for k=1:length(RecvFskDemod)/3

    RecvCorr(k) = sum(RecvFskDemod(k:Ts:k+PreambleLen\*Ts-1).\*PreambleSig);

end

RecvCorr = abs(RecvCorr);

* 1. 说明

确定第一个数据码元的时间位置，是输入输出码元同步的关键，我们通过计算RecvFskDemod 与前导码的相关函数，并找到相关函数最大的位置，即为我们的前导码第一个码元位置。

RecvFskDemod 是经过了上采样的信号，在与前导码进行相乘的时候，需要使用Ts步长来计算相关函数，最佳的抽样时刻便是相关性最大的时刻，采用max函数找到最佳抽样时刻的第一个码元的位置。

由上述分析可得相关函数计算式：

由于RecvFskDemod是复数，在计算相关性时使用real只考虑实部。

1. **码元采样：**
   1. 代码：

% sampled symbol

[peak, pos] = max(RecvCorr);

RecvSymbolSampled = RecvFskDemod(pos:Ts:pos+(PreambleLen+SymbolLen)\*Ts-1);

RecvSymbolSampled = real(sign(RecvSymbolSampled(PreambleLen+1:end)));

* 1. 说明：

码元采样模块比较简单，只需要从确定的第一个码元位置后，按照上采样率为步长，依次采样需要的码元长度即可。

1. **卷积码译码：**
   1. 代码：

% convolutoinal decoding

sampled\_code = round(max(RecvSymbolSampled,0));%round((1-RecvSymbolSampled)/2);

t = poly2trellis(CoderConstraint, [171, 133]);

tb = 2;

decoded = vitdec(sampled\_code,t,tb,'trunc','hard');

RecvBit = decoded(1:end-6);

* 1. 说明：

vitdec ：卷积码的维特比译码（二进制数据）

decoded = vitdec(code,trellis,tblen,opmode,dectype)；

decoded = vitdec(code,trellis,tblen,opmode,'soft',nsdec)

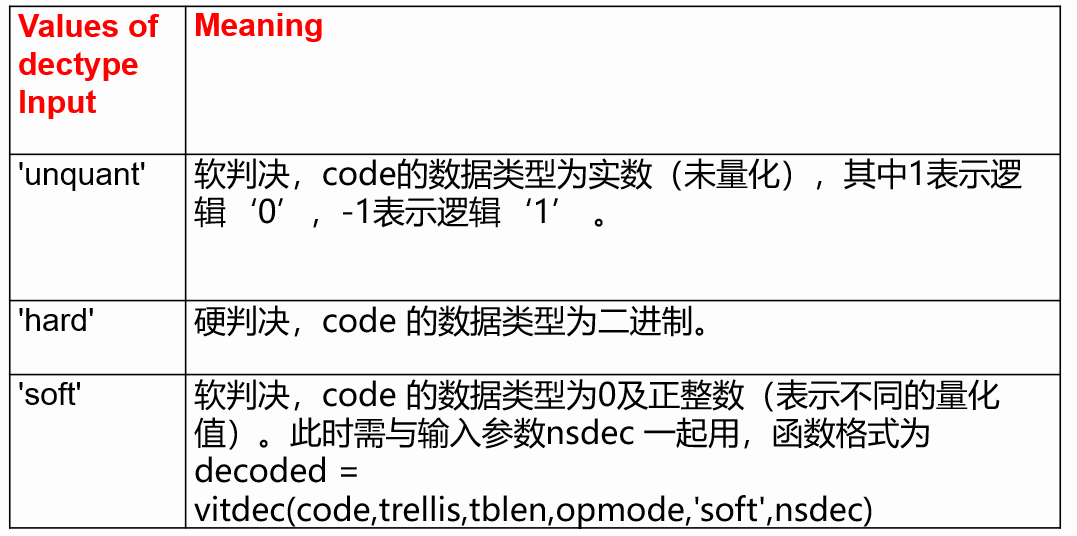
输入参数1——code，维特比译码器的输入符号序列，矢量形式。以前述2/3码率的编码器结构为例，每个符号代表编码器输出的3个bit；

输入参数2——trellis，卷积码编码器的网格结构；

输入参数3——tblen，a positive integer scalar，用于规定回溯深度。If the code rate is 1/2, a typical value for tblen is about five times the constraint length of the code；

输入参数4—— opmode，指示译码器的操作模式，与编码器的结尾处理方式有关。通常取值=‘term’，此时假设编码器是从移位寄存器的全零状态开始，也结束于全零状态。为此，编码时，信息符号输入完毕后，需要额外输入若干0比特，以使移位寄存器恢复为全零状态；

输入参数5—— dectype，指示译码器的判决类型。其取值不同，对应的输入参数1——code的数据类型也不同。其取值如下表：

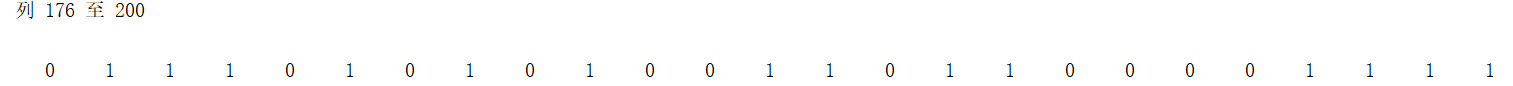


输出参数——decoded，维特比译码器的输出符号序列，矢量形式。以前述2/3码率的编码器结构为例，每个符号代表编码器输入的2个bit。

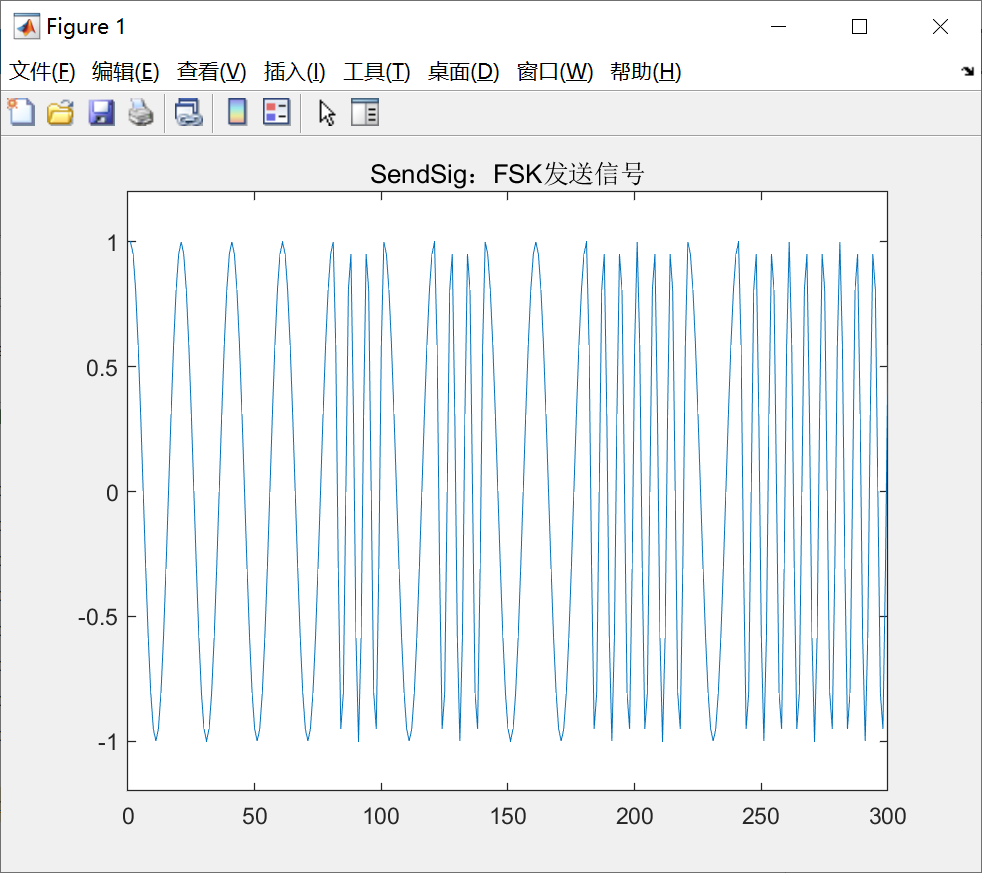
1. **实验系统搭建与调试**

#### 4.1 发送端：

首先观察数据源比特部分序列如图所示：

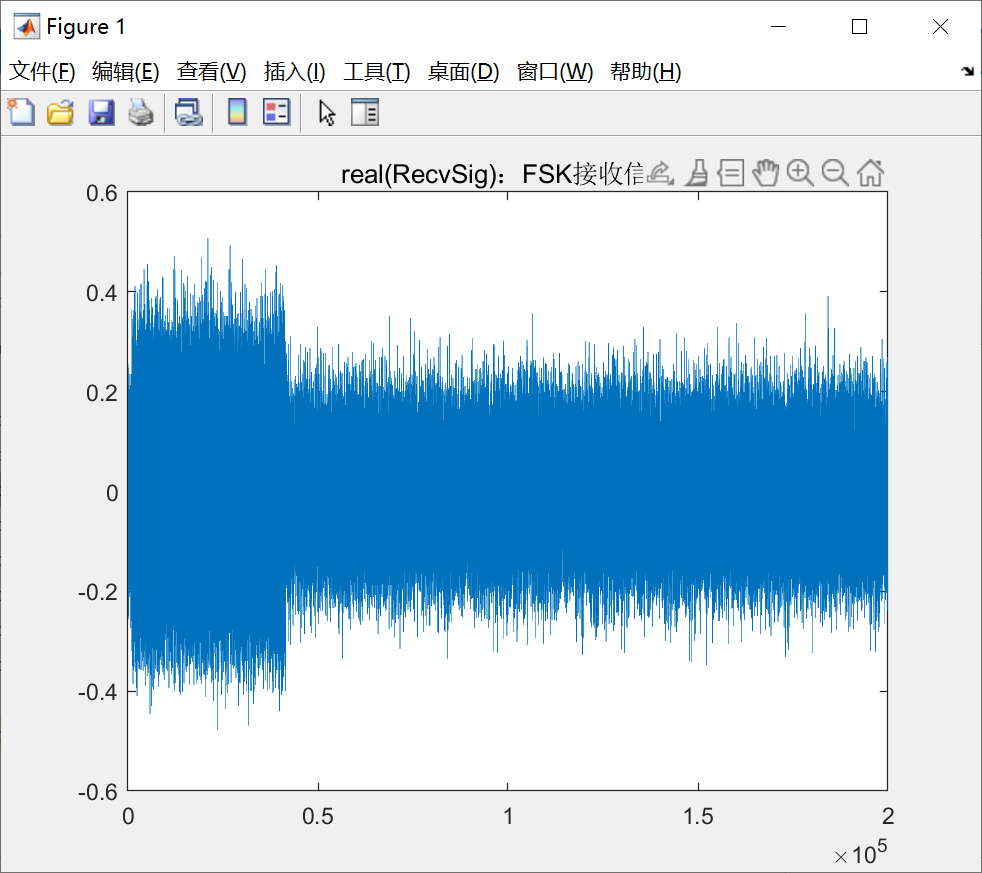


观察发送信号波形，可见实现了FSK调制。

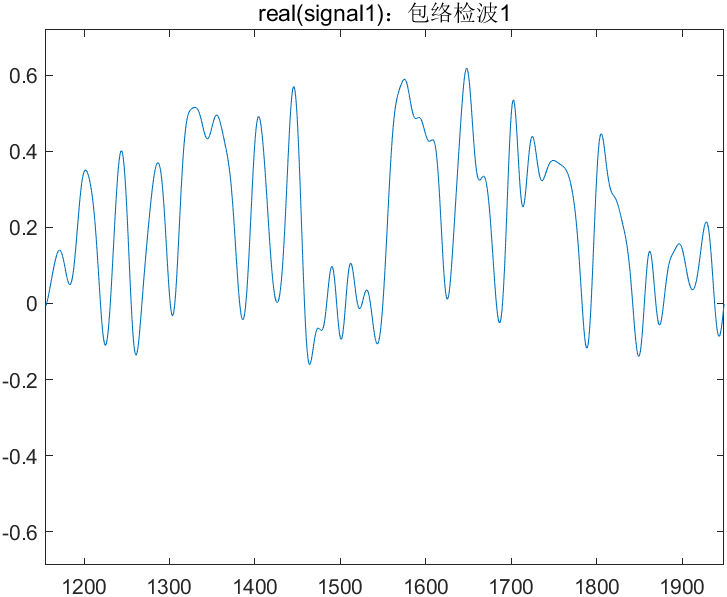


#### 4.2 接收端：

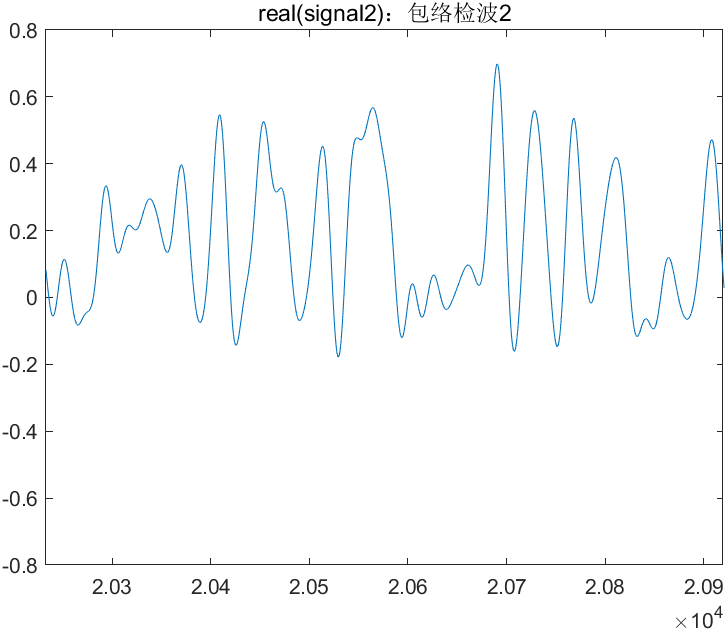
FSK接收端信号如图所示，可见信道正常，可以正常接收。



包络检波1的输出信号为：

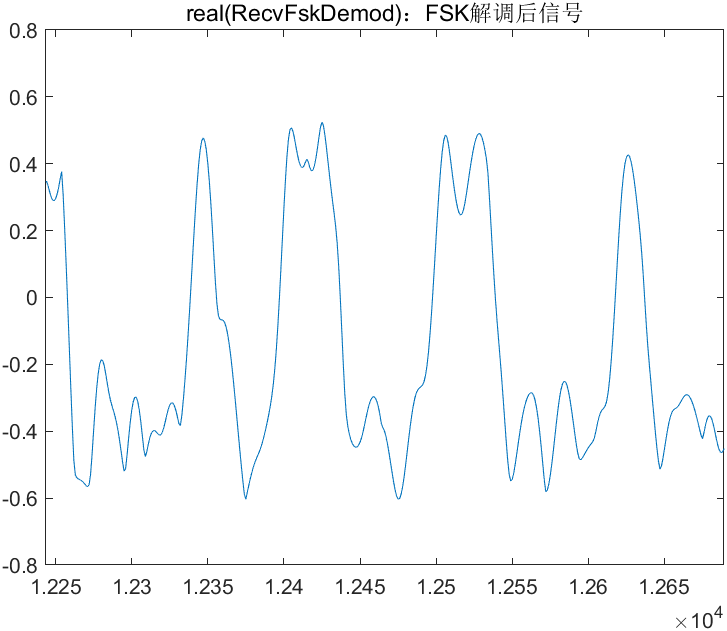


包络检波2的输出波形为：



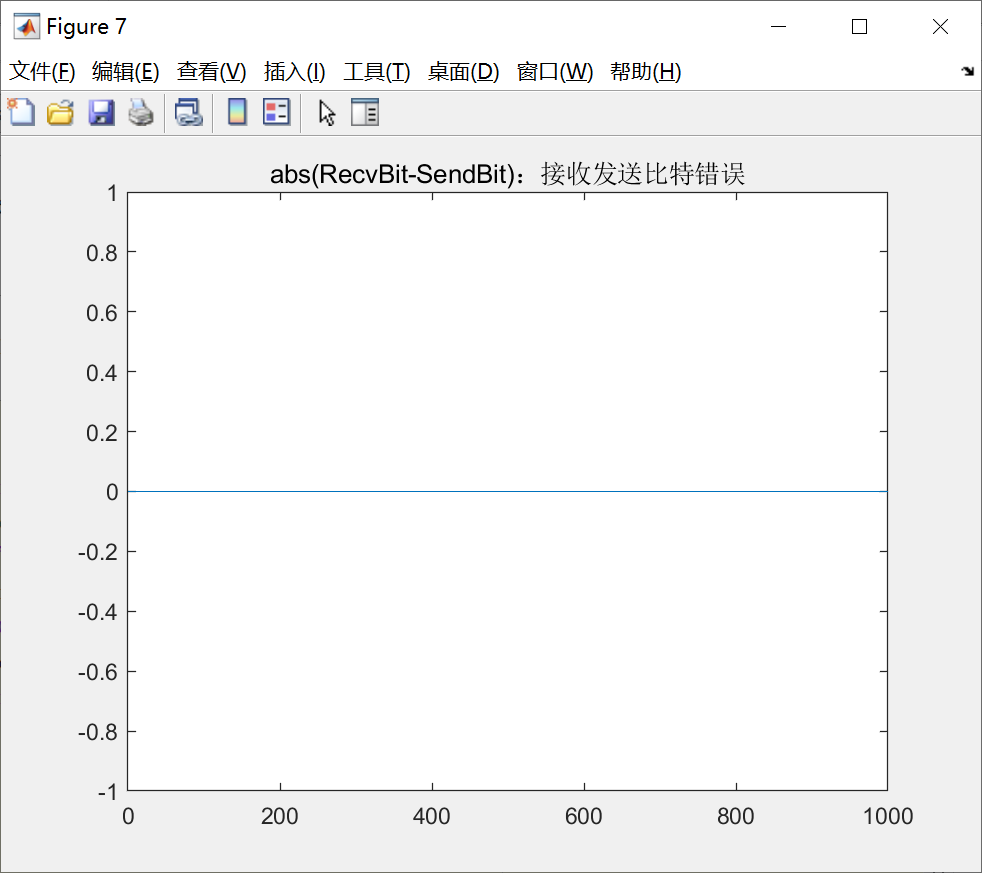
可见，滤波与包络检波部分功能已实现。

在观察解调输出波形：



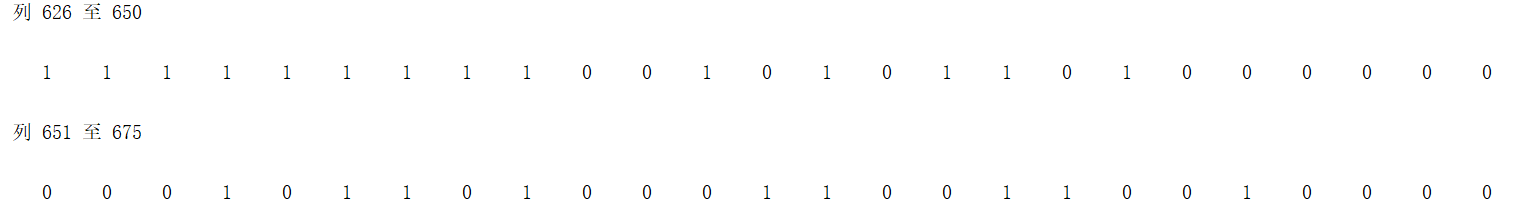
可见，基本实现FSK解调，但还存在一定的噪声。

最后通过观察接收发送比特错误，判断整个系统工作正常。



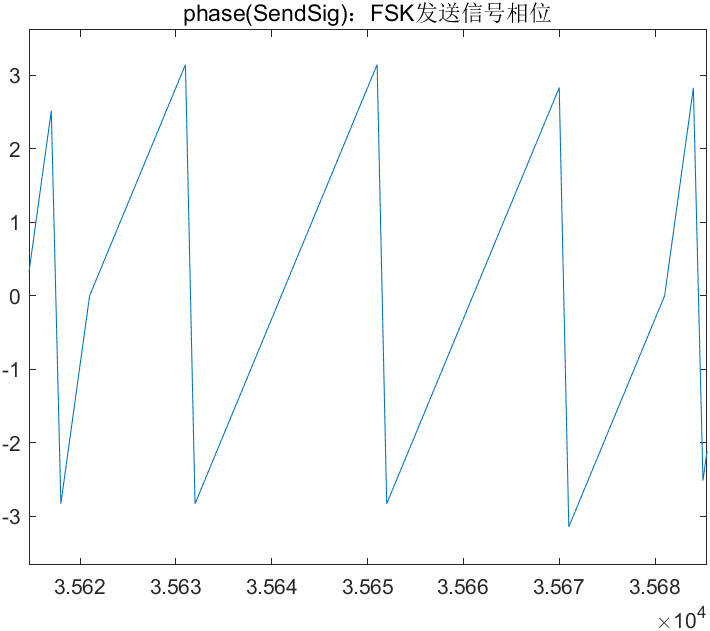
1. **实验数据**
2. 数据源比特：

观察截取626-675部分：



1. FSK发送信号相位：

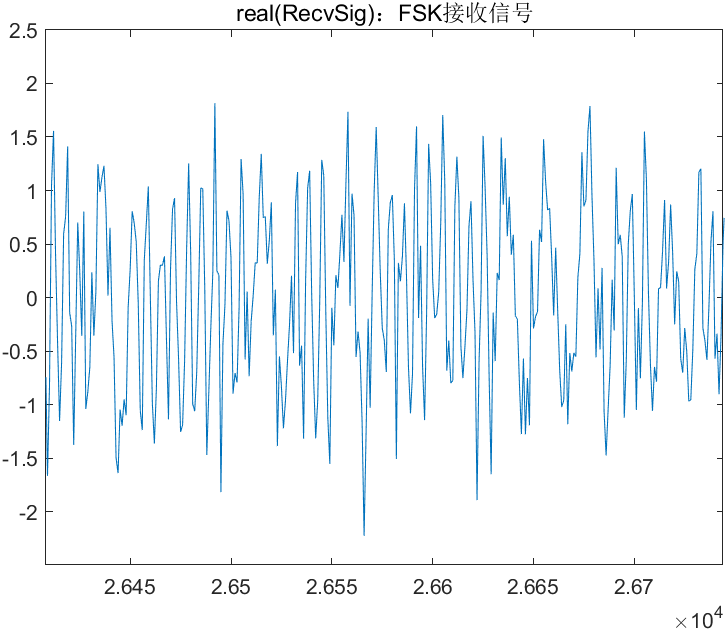
观察部分：



1. 信道参数：

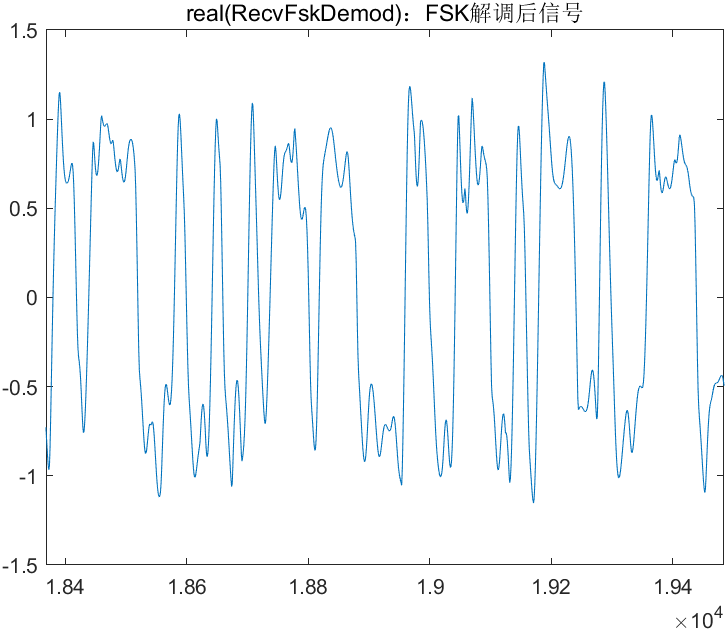


1. FSK接收信号：

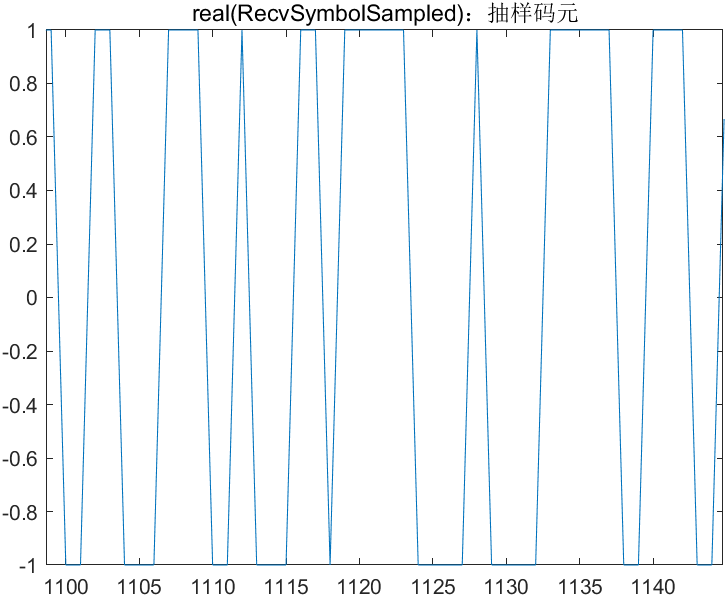


可见，存在较强的噪声干扰。

1. FSK解调后信号：

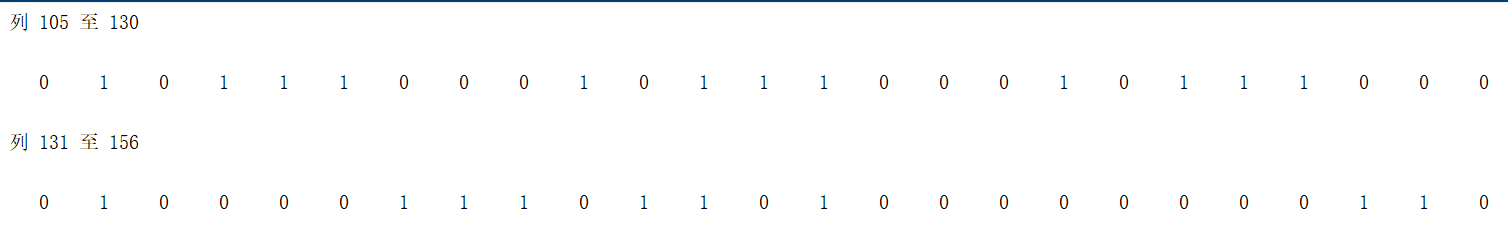


1. 抽样码元：

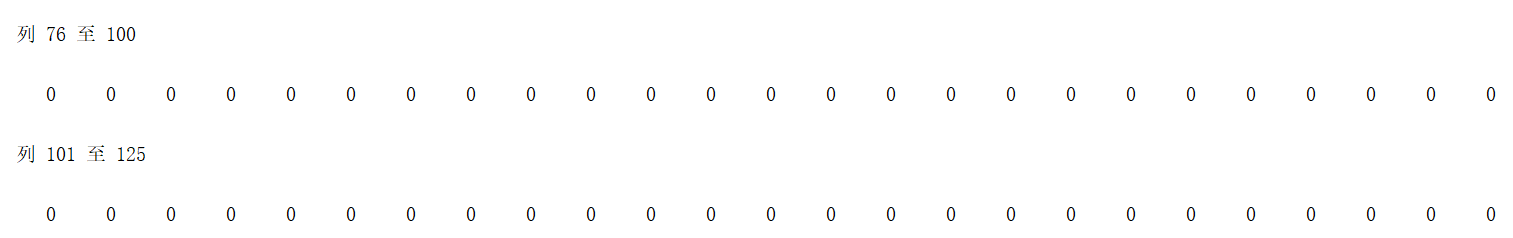


1. 译码比特：

观察105-156

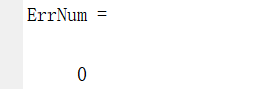


1. 接收发送比特错误：

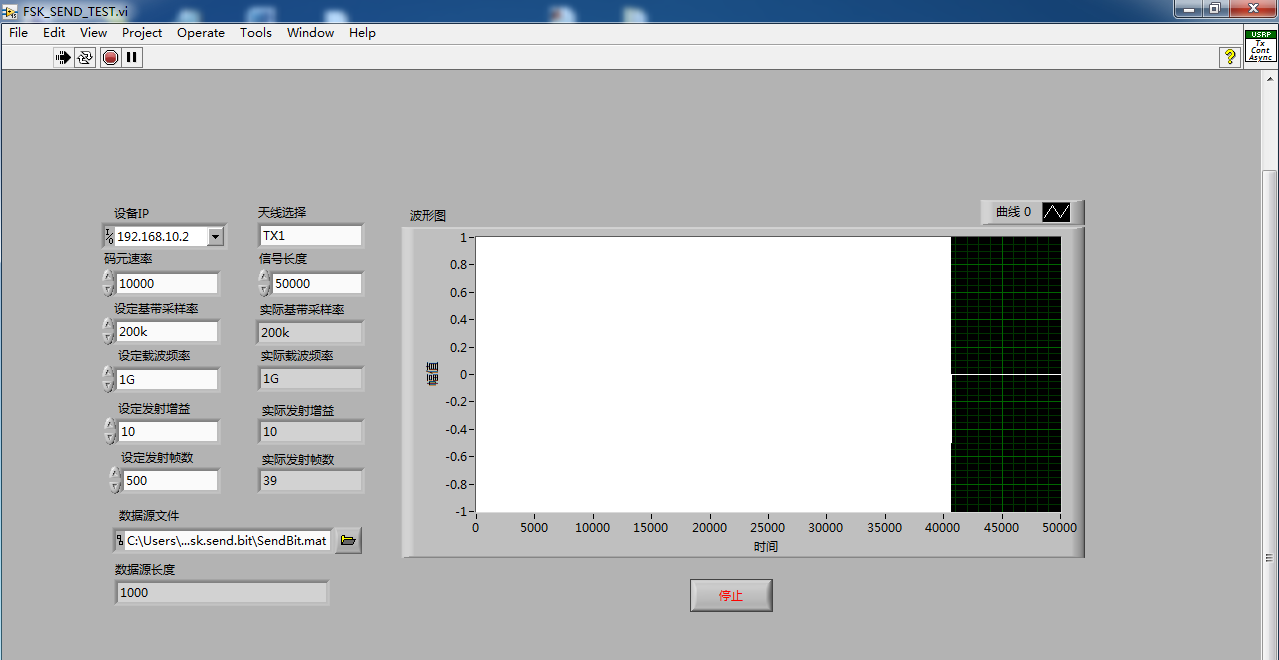


可见，全为0，实验系统成功。

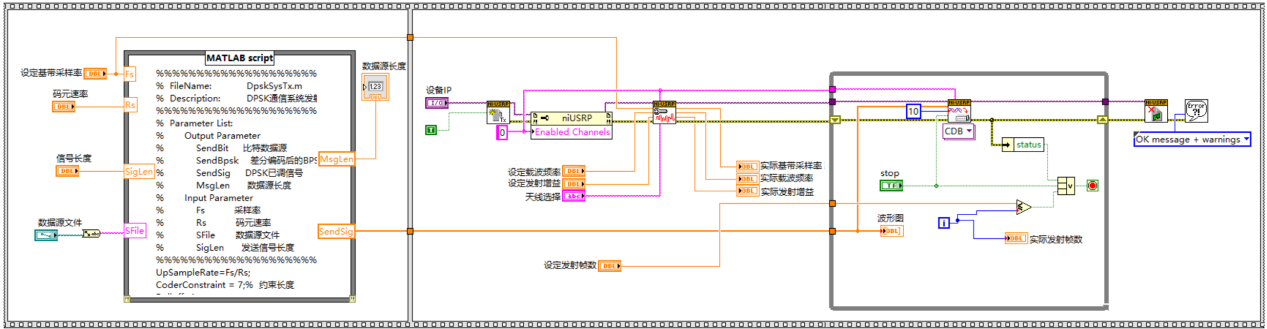
1. 总误码数：



1. **USRP实验结果记录**
2. 发射机面板与框图：

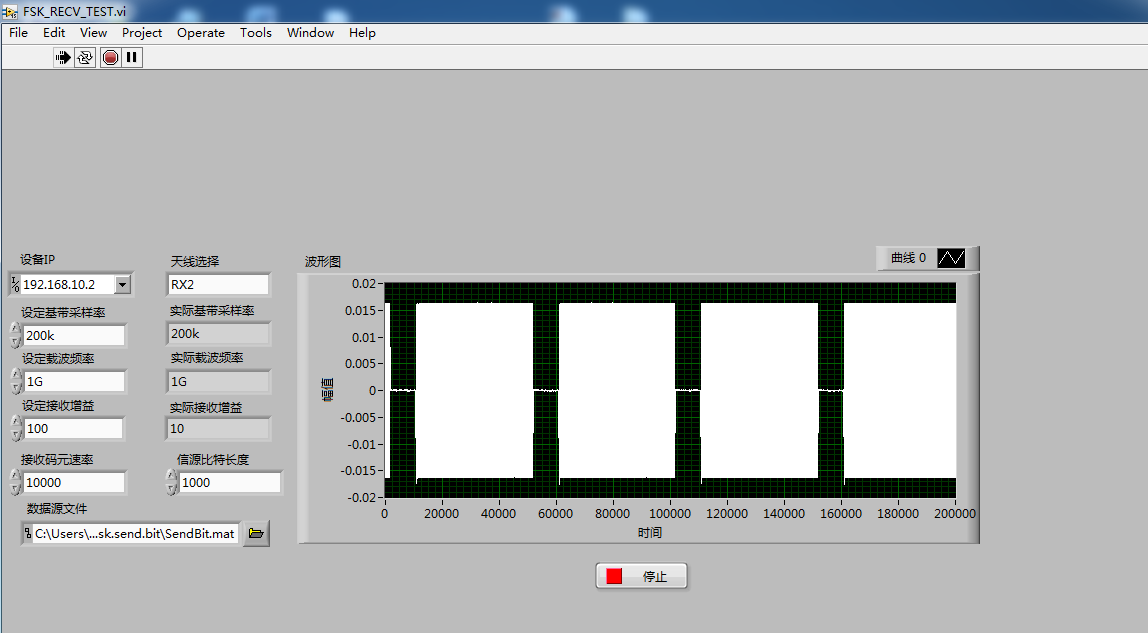


(a) 发射机前面板

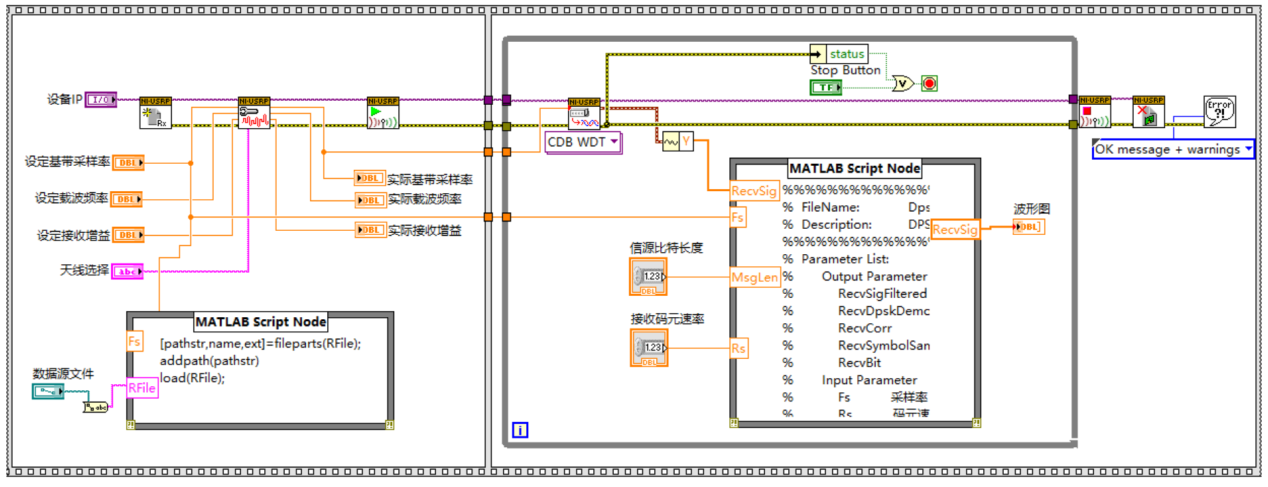


(b) 发射机程序框图

1. 接收机面板与框图：

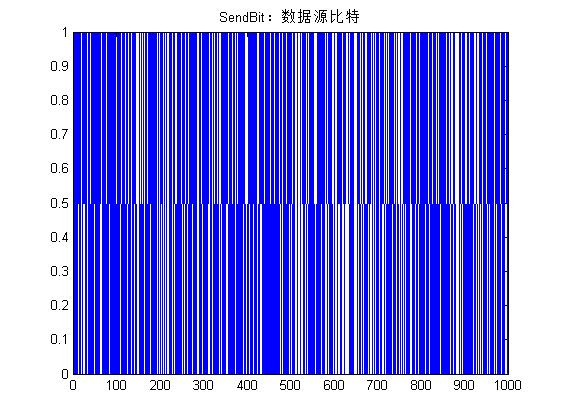


(a) 接收机前面板

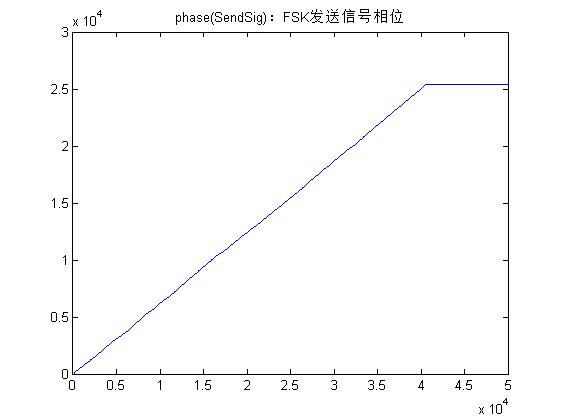


(b) 接收机程序框图

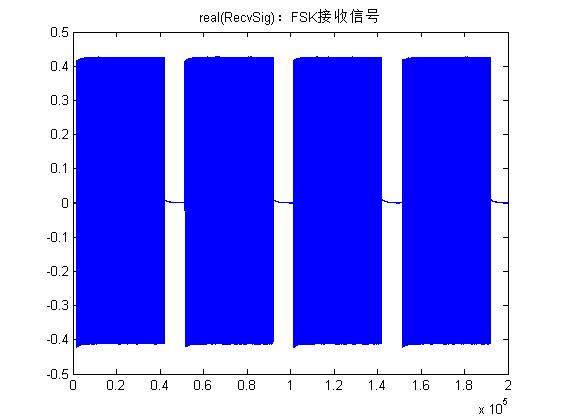
1. SendBit波形：



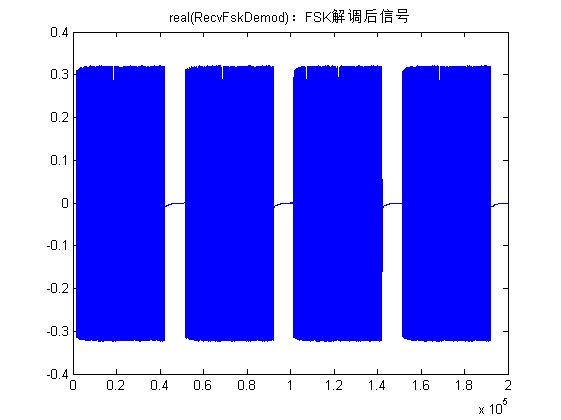
1. phase(SendSig)波形：



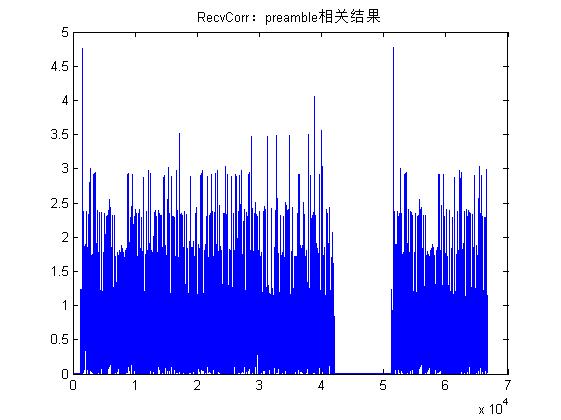
1. RecvSig波形：



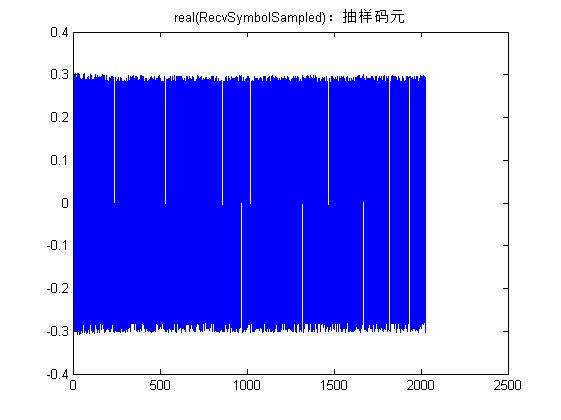
1. RecvFskDemod波形：



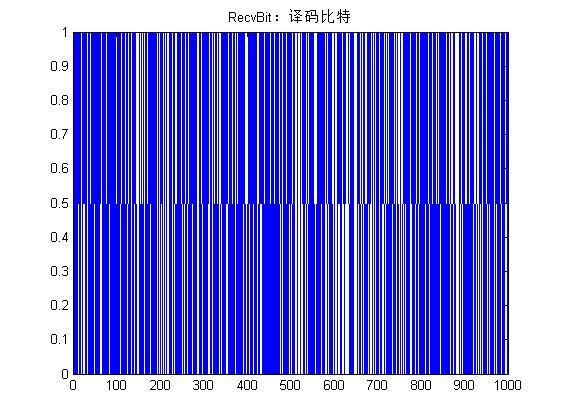
1. RecvCorr波形：



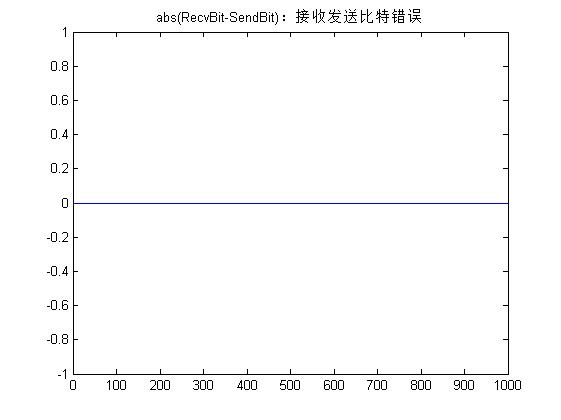
1. RecvSymbolSampled波形：



1. RecvBit波形：



1. RecvBit-SendBit波形：



1. **实验结论**

本次课程实验主要涉及到Matlab和Labview在通信系统方面的应用，加深了我们对数字基带传输系统的理解和掌握。

在Matlab编程方面，知道了如何用函数的方法，实现2FSK的调制与解调，并且生成重要信号的波形，以此来判断高斯噪声对信号的影响。

另外，我们也初步学会了使用专用在通信方面的硬件USRP完成实验，在这次USRP实验中，我尝试采用了相干解调和非相干解调两种解调方法。不过，有两处需要注意，一个是滤波器的设置，这个我花了比较长的时间，通过上网搜索和自己一次一次的设置，终于选取了较好的基带频率和载波频率，解调出了原始波形；另一个需要注意的是抽样判决的实现，最初以为需要引入定时脉冲，后来才知道只需要加一个比较器便可实现。在对比基带信号和解调信号后发现解调信号相对于基带信号有比较大的延迟，后来改变了一下滤波器参数，减小了延迟时间。