**大连理工大学实验报告**

学院（系）：信息与通信工程学院 专业： 班级：

姓 名： 学号： 组： \_\_\_

实验时间： 实验室： 实验台：

指导教师：

**实验I：随机信号的产生、相关分析及其应用实验**

1. 实验目的和要求

通过实验理解掌握随机信号样本生成的原理和方法、掌握随机过程相关函数的计算原理和方法。训练MATLAB程序代码编写能力，要求完成以下工作，并将实验结果与理论分析对照。

1.基于均匀分布伪随机数，掌握均匀分布白噪声典型生成方法。

2.基于均匀分布伪随机数，掌握高斯分布白噪声典型生成方法。

3.掌握随机信号相关函数计算、相关分析及实现方法。

1. 实验原理和内容

1.实验内容

(1)均匀分布随机数

较简单的伪随机序列产生方法是采用数论中基于数环理论的线性同余法（乘同余法、混 合同余法），其迭代公式的一般形式为 f(x) = (r·x + b) Mod M，其离散形式为 s(n + 1) = [r·s(n) + b] Mod M。其中，s(n)为 n 时刻的随机数种子，r 为扩展因子，b 为固定扰动项，M 为循环模，Mod M 表示对 M 取模。为保证 s(n)的周期为 M，r 的取值应满足 r = 4k + 1， M = 2p ，k 与p的选取应满足：r < M，r(M-1) + 1< 231 -1。通常公式中参数常用取值为s(0) =12357，r = 2045， b = 1，M =1048576。

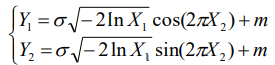
1. 混合同余法: 选定参数 a、c , M， 再选定一个初值 y(0), 按下式产生随机数:

y (n+1) = a\*y(n)+c (mod M ) (n=1,2…), x(n+1)=y(n+1)/M, 其中 M 为足够大的整数. 在 C 语言中, M = 216 , a、c和y(0)均为0至M中的常数。按上面算法公式，则y(1),…,y(N)为（0，M）之间的均匀随机数, x(1),…,x(N)为（0,1）之间的均匀随机数。

1. 乘同余法: 在混合同余法算法中， 令 c=0, 则为乘同余法。

(2)高斯随机数

a.变换抽样法，如果X1、X2是两个互相独立的均匀分布随机数，那么如下 Y1、Y2是期 望为 m，方差为 σ 2 的高斯分布函数，且互相独立：为简单起见，可以令期望 m=0, 方差为 σ2=1。



b.较简单的高斯白噪声产生方法是基于概率论中的中心极限定理。即无穷多个同分布随机变量之和构成随机变量服从高斯分布。方便起见，可用 N 个（通常 N=12）均匀分布随机变量之和Xi近似高斯分布随机变量。若Xi，i = 0, 1, …, 11 在[0, 1)上服从均匀分布，则 Y=近似服从均值为0, 方差为1的高斯分布。

(3)相关函数估计

离散随机序列自相关函数定义为 Rx(m)=E[x(n)x(n+m)]。对于各态历经随机过程，统计平均可用时间平均代替，即。工程实践中，无法获得无限长数据，只能用有限平均来近似，即，m=0, 1, …, K-1；K<N。为保证估计质量，通常要求 K<<N, 此时 Rx(m)也可以简化为。同理, 也类似地计算互相关函数。

2实验内容

(1) 编程实现产生10000个在(0, 1)区间均匀分布随机数。计算生成随机数的1~4阶矩，最大值，最小值，频度直方图。

(2) 编程实现产生10000个N(3, 4) 高斯随机数。计算生成随机数的1~4阶矩，最大值，最小值，频度直方图。

(3) 编程实现产生10000个N(1, 2) 高斯随机数和10000个N(3, 4) 高斯随机数。计算其自相关函数， 计算两个高斯随机信号的互相关函数。

(4)探究式实验内容：声音延迟及噪声分布律估计

研究随机信号相关分析方法在实际中的应用。针对指定的两路语音信号：已知其中一路是另外一路的延迟，延迟的时间未知，填充延迟时间的数据是标准的随机分布噪声数据，两路信号可能存在一定的干扰。要求找出两路信号之间的时间差，并且估计出延迟时间中随机数分布律及其参数。

1. 主要仪器设备

Maltab

1. 实验步骤与操作方法

1.实验步骤

(1) 实现产生 10000 个在(0, 1)区间均匀分布随机数。

(2) 计算生成随机数的1~4阶矩，最大值，最小值，频度直方图。

(3) 编程实现产生10000个N(3, 4)高斯随机数。

(4) 计算生成随机数的1~4阶矩，最大值，最小值，频度直方图。

(5) 编程实现产生10000个N(1, 2)高斯随机数和10000个N(3, 4)高斯随机数。

(6) 计算其自相关函数，计算两个高斯随机信号的互相关函数。

(7) 读取对应的两个声音文件。

(8) 计算他们的互相关函数，搜索峰值估算延迟。

(9) 计算延迟点数之前信号的数字特征，估计分布类型。

(10)用内置函数假设检验

2.操作方法

内容1：

(1)自编产生随机数函数：

function s=randuniform(M,r,b,first,num)

s=zeros(1,num);

s(1)=first;

for i=2:num

s(i)=mod(s(i-1)\*r+b,M);

end

s=s/M;

(2)使用自编函数并估计数字特征：

%% 准备环境

clear all

close all

clc

%% 生成随机数

M=1048576;

r=2045;

b=1;

first=12357;

num=10000;

s=randuniform(M,r,b,first,num);

%% 作频度直方图

figure,subplot(2,1,1),hist(s)

title('分10个区间的直方图')

subplot(2,1,2),hist(s,100)

title('分100个区间的直方图')

suptitle('自编代码生成数据直方图展示')

figure,

subplot(2,2,1),plot(s) %全部

title('全部数据连线')

subplot(2,2,2),plot(s(1:100)) %前100个数据

title('前100个数据连线')

subplot(2,2,3),plot(s,'.') %全部

title('全部数据画点')

subplot(2,2,4),plot(s(1:100),'\*') %前100个数据

title('前100个数据画点')

suptitle('自编代码生成数据展示')

%% 相关数字特征计算

maxs=max(s);

mins=min(s);

m = zeros(1,4); %生成[0 0 0 0]

for i = 1 : num

m(1) = m(1) + s(i); % 均值

m(2) = m(2) + s(i)^2; % 二阶矩

m(3) = m(3) + s(i)^3; %三阶

m(4) = m(4) + s(i)^4; %四阶

end

m=m/num;

disp(['生成数据的数字特征'])

disp(['最大值 = ',num2str(maxs)] );

disp(['最小值 = ',num2str(mins)] );

disp(['均值 = ',num2str(m(1))] );

disp(['均方值 = ',num2str(m(2))] );

disp(['三阶原点矩 = ',num2str(m(3))] );

disp(['四阶原点矩 = ',num2str(m(4))] );

disp(['生成数据的数字特征'])

[f,xi]= ksdensity(s,'support',[0,1]);

figure,subplot(2,1,1),plot(xi,f);

title('利用MATLAB函数ksdensity估计的概率密度')

t=0:1/100:1;

subplot(2,1,2),plot(t,ones(size(t)));

axis([0 1 0 1]);

title('预设的[0 ,1]均匀分布理论概率图')

suptitle('自编代码生成数据概率图与理论概率比较')

sm = rand(1,num);

[f,xi]=ksdensity(sm,'Kernel','epanechnikov','support',[0,1]);

figure,plot(xi,f);

title('利用MATLAB函数ksdensity估计的概率密度')

suptitle('MATLAB内置函数生成数据展示')

内容2：

（1）自编变换法生成高斯分布：

function z=rnd1(mn,a,num)

z=zeros(1,num);

x1=rand(1,num);

x2=rand(1,num);

for i=1:num

y1=sqrt((-2)\*log(x1(i)))\*cos(2\*pi\*x2(i));

z1=a\*y1+mn;

z(i)=z1;

end

（2）自编中心极限法生成高斯分布：

function z=rnd2(mn,a,num)

z=zeros(1,num);

x=randuniform(1048576,2045,1,12357,num\*12);

for i=1:num

y=sum(x(1+12\*(i-1):1+12\*(i-1)+11)-1/2)/sqrt(12/12);

z(i)=a\*y+mn;

end

（3）使用中心极限法产生高斯分布，并估计数字特征：

%% 准备环境

clear all

close all

clc

%% 生成高斯随机数

num=100000;

mn=3;

a=2;

s1=rnd2(mn,a,num);

%% 作频度直方图

figure,subplot(2,1,1),hist(s1)

title('分10个区间的直方图')

subplot(2,1,2),hist(s1,100)

title('分100个区间的直方图')

suptitle('自编代码生成数据直方图展示')

%% 计算数字特征

m = zeros(1,4); %生成[0 0 0 0]

for i = 1 : num

m(1) = m(1) + s1(i); % 均值

m(2) = m(2) + s1(i)^2; % 二阶矩

m(3) = m(3) + s1(i)^3; %三阶

m(4) = m(4) + s1(i)^4; %四阶

end

m=m/num;

maxs=max(s1);

mins=min(s1);

disp(['生成数据的数字特征'])

disp(['最大值 = ',num2str(maxs)] );

disp(['最小值 = ',num2str(mins)] );

disp(['均值 = ',num2str(m(1))] );

disp(['均方值 = ',num2str(m(2))] );

disp(['三阶原点矩 = ',num2str(m(3))] );

disp(['四阶原点矩 = ',num2str(m(4))] );

figure,

subplot(2,2,1),plot(s1) %全部

title('全部数据连线')

subplot(2,2,2),plot(s1(1:100)) %前100个数据

title('前100个数据连线')

subplot(2,2,3),plot(s1,'.') %全部

title('全部数据画点')

subplot(2,2,4),plot(s1(1:100),'\*') %前100个数据

title('前100个数据画点')

suptitle('自编代码生成数据展示')

%% 与标准高斯分布进行对比

meanValue = mean(s1);

stdValue = std(s1);

disp('----------')

disp(['预设参数，均值为：',num2str(mn),',标准差为：',num2str(a)]);

meanErr = (meanValue - mn)/(mn)\*100;

stdErr = (stdValue - a)/(a)\*100;

disp(['计算参数，均值为：',num2str(meanValue),',标准差为：',num2str(stdValue)]);

disp(['相对误差分别为：',num2str(meanErr),' %, 和：',num2str(stdErr),' %'])

内容3：

用中心极限法产生高斯分布并估计自，互相关函数：

%% 设定环境

clear all

close all

clc

%% 设定均值和标准差，以及产生随机高斯数

mn1=1;

mn2=3;

a1=sqrt(2);

a2=2;

num=100;

s1=rnd2(mn1,a1,num);

s2=rnd2(mn2,a2,num);

%% 估计自相关和互相关

[x1,lags]=xcorr(s1);

[x2,lags]=xcorr(s2);

[x12,lags]=xcorr(s1,s2);

subplot(2,1,1);plot(lags,x1);

[~,indx]=max(x1);

title(['最大值点位于',num2str(abs(indx-num))]);

hold on

subplot(2,1,2);plot(lags,x2,'r');

[mx,indx]=max(x2);

title(['最大值点位于',num2str(abs(indx-num))]);

suptitle('两路信号的自相关展示')

figure;plot(lags,x12)

title(['两路信号的互相关展示'])

内容4：

读取语音并估计分布：

%% 准备环境

clear all

close all

clc

%% 读取数据

[filename1,path1]=uigetfile;

audeofile1= strcat(path1,filename1);

[y1,Fs1] = audioread(audeofile1);

[filename2,path2]=uigetfile;

audeofile2= strcat(path2,filename2);

[y2,Fs2] = audioread(audeofile2);

num=length(y1);

%% 判断延迟

[x,lags]=xcorr(y1,y2,'coeff');

figure;plot(lags,x)

title('两路同源信号的互相关，峰值处最相关');

[mx,indx] = max (x);

TLag = abs(num - indx);

disp(['经计算相关函数，估计延迟点数为：',num2str(TLag),'。'])

TsLag=TLag/Fs1;

disp(['估计延迟时间为：',num2str(TsLag),'。'])

%% 进行参数计算

noise=y2(1:TLag);

m = zeros(1,4); %生成[0 0 0 0]

for i = 1 : TLag

m(1) = m(1) + noise(i); % 均值

m(2) = m(2) + noise(i)^2; % 二阶矩

m(3) = m(3) + noise(i)^3; %三阶

m(4) = m(4) + noise(i)^4; %四阶

end

m=m/TLag;

disp(['生成数据的数字特征'])

disp(['均值 = ',num2str(m(1))] );

disp(['均方值 = ',num2str(m(2))] );

disp(['三阶原点矩 = ',num2str(m(3))] );

disp(['四阶原点矩 = ',num2str(m(4))] );

meanValue=mean(noise);

stdValue =std(noise);

disp(['方差 = ',num2str(stdValue^2)] );

figure,

subplot(2,2,1),plot(noise) %全部

title('全部数据连线')

subplot(2,2,2),plot(noise(1:100)) %前100个数据

title('前100个数据连线')

subplot(2,2,3),plot(noise,'.') %全部

title('全部数据画点')

subplot(2,2,4),plot(noise(1:100),'\*') %前100个数据

title('前100个数据画点')

suptitle('添加噪声数据展示')

%% 和预设值对比

disp(['预设参数，均值为：',num2str(0.1),',方差为：',num2str(0.01)]);

meanErr = (meanValue - 0.1)/(0.1)\*100;

stdErr = (stdValue^2 - 0.01)/(0.01)\*100;

disp(['相对误差分别为：',num2str(meanErr),' %, 和：',num2str(stdErr),' %'])

[f,xi]=ksdensity(noise);

figure,subplot(2,1,1),plot(xi,f);

title('利用MATLAB函数ksdensity估计的概率密度')

t=0:1/1000:1;

y=10\*exp((-10)\*t);

subplot(2,1,2),plot(t,y);

title('预设参数为10的指数分布理论概率图')

suptitle('估计数据概率图与理论概率比较')

H=kstest(noise,[noise,expcdf(noise,0.1)],0.05);

if H==0

disp('该数据服从参数为10的指数分布。')

else

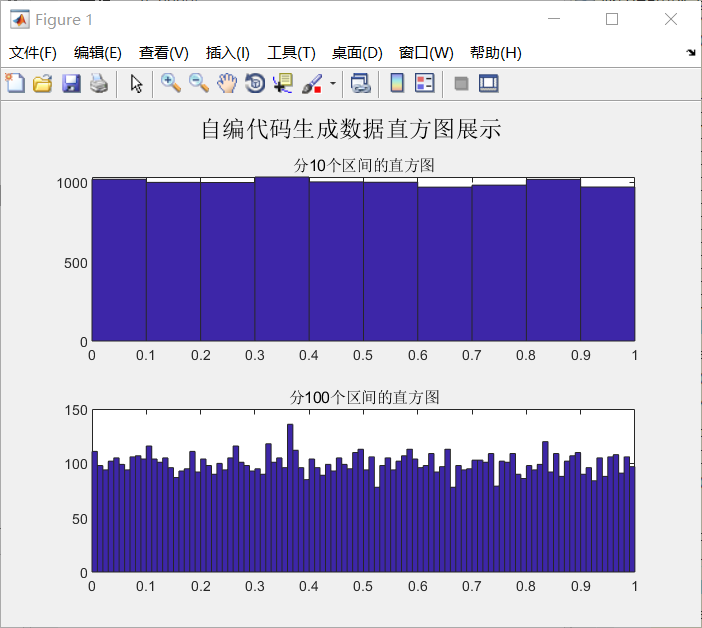
disp('该数据不服从参数为10的指数分布。')

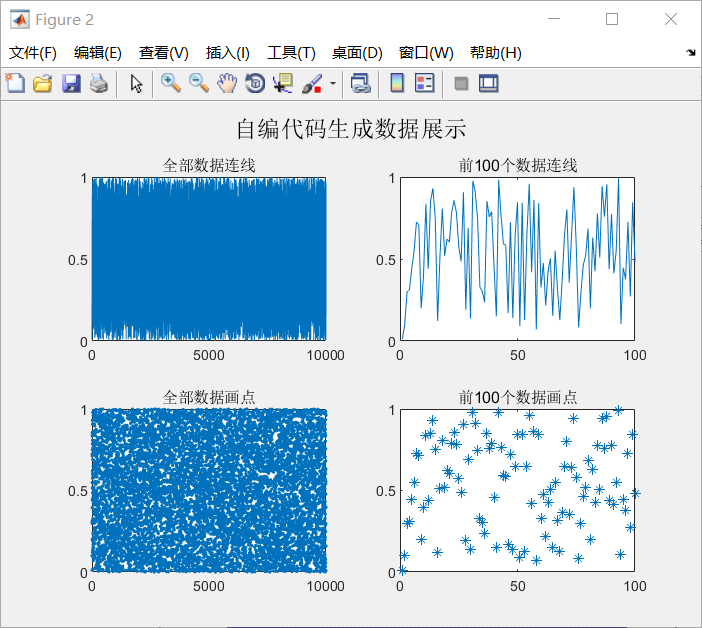
end

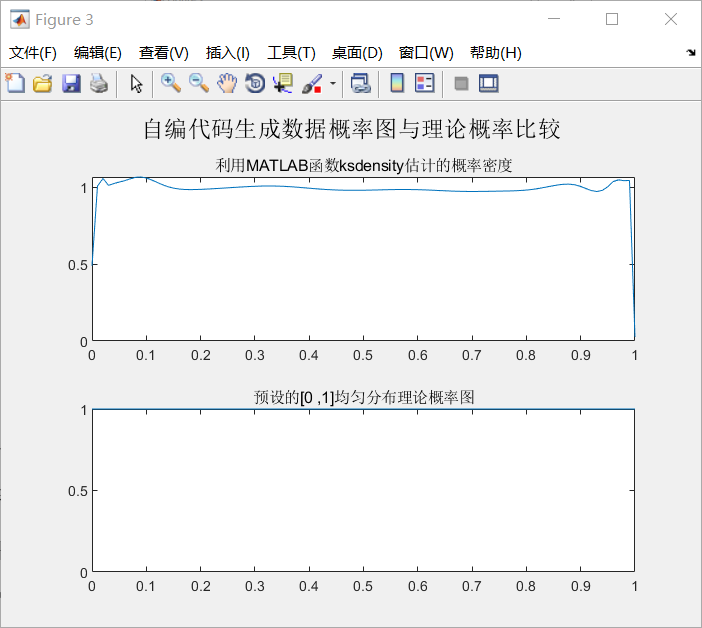
1. 实验数据记录和处理

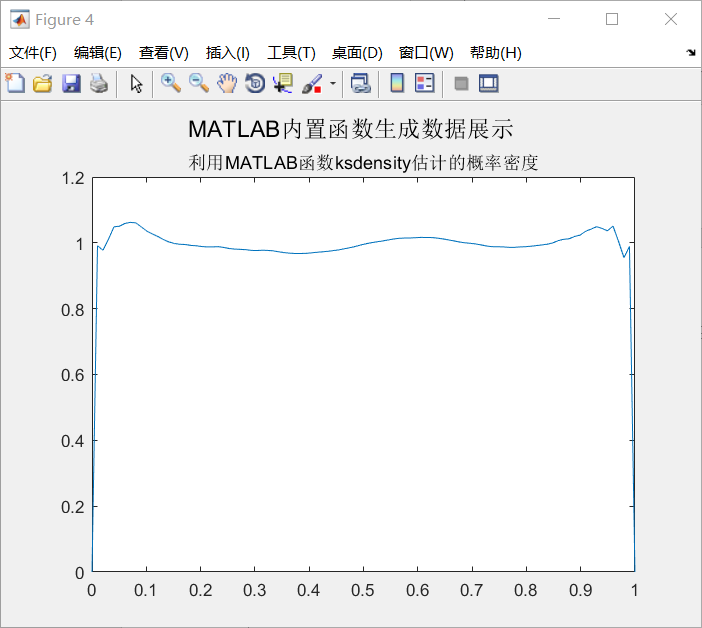
1.均匀随机数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 一阶矩 | 二阶矩 | 三阶矩 | 四阶矩 | 最大值 | 最小值 |
| 0.49724 | 0.33062 | 0.24762 | 0.19795 | 0.99986 | 7.6294e-06 |



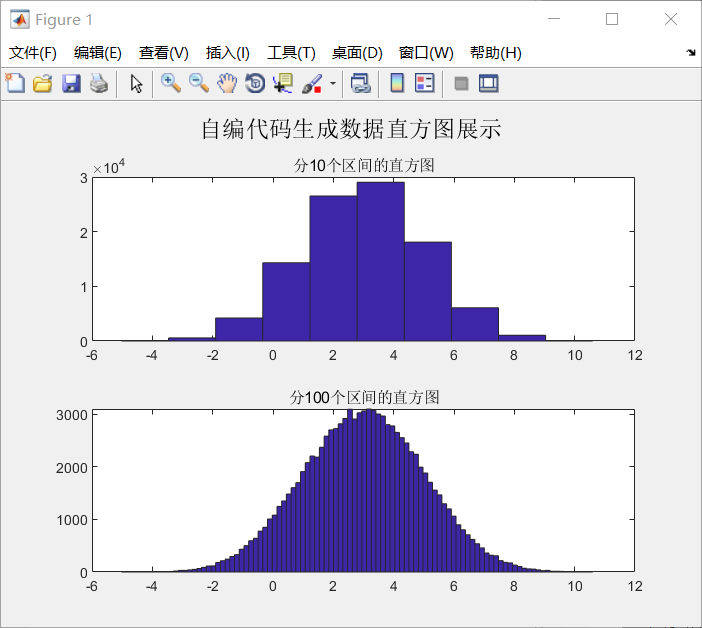


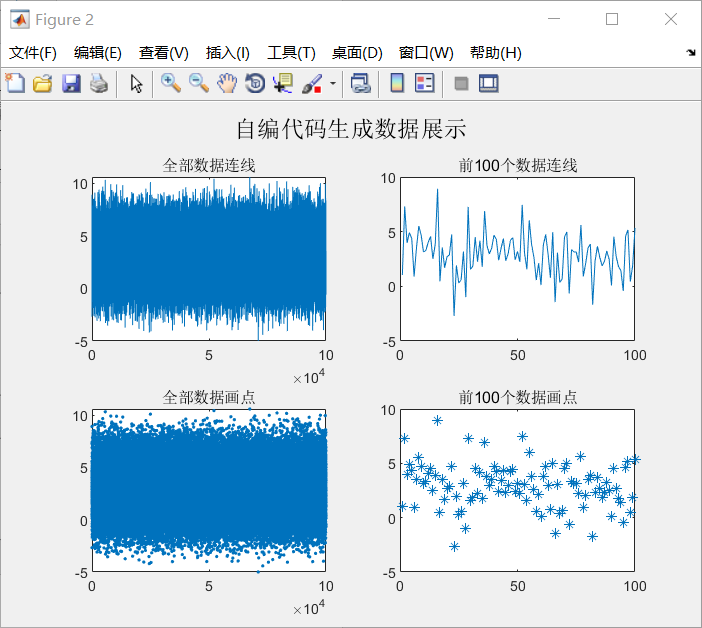




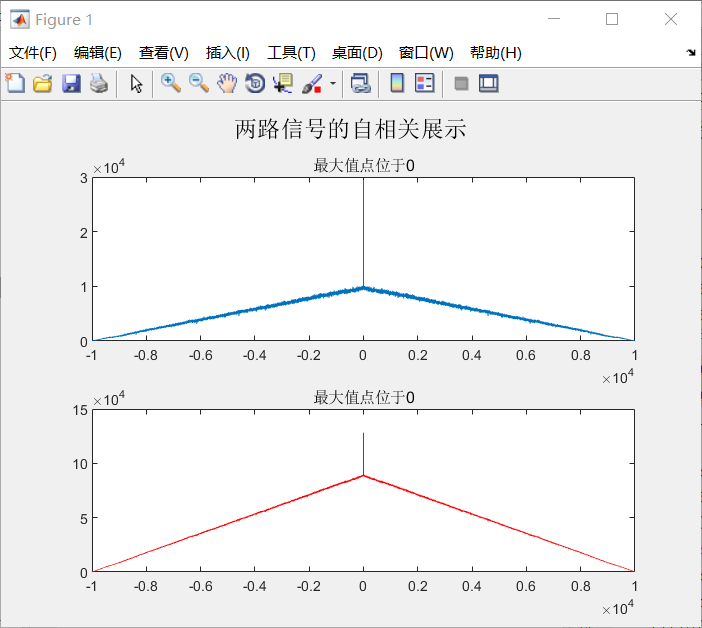
2.高斯随机数

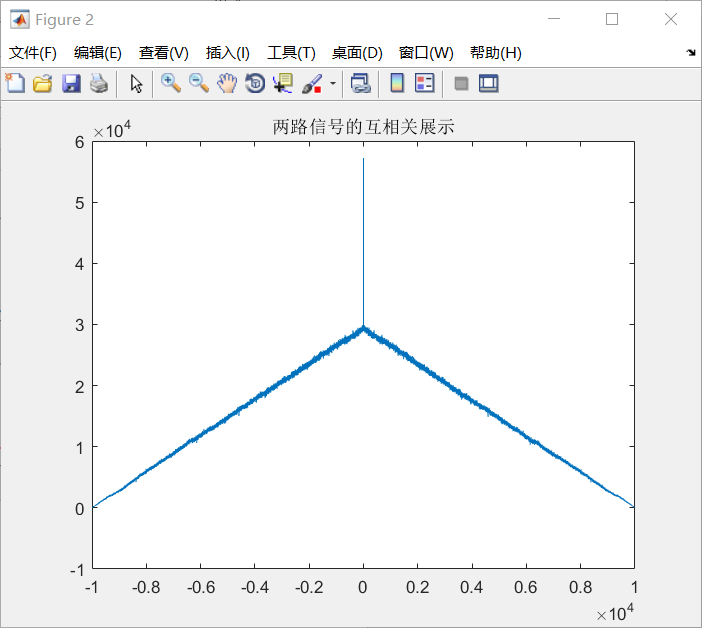
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 一阶矩 | 二阶矩 | 三阶矩 | 四阶矩 | 最大值 | 最小值 |
| 2.9987 | 12.9793 | 62.7803 | 340.8509 | 10.5986 | -5.025 |





3.高斯函数的自相关和互相关



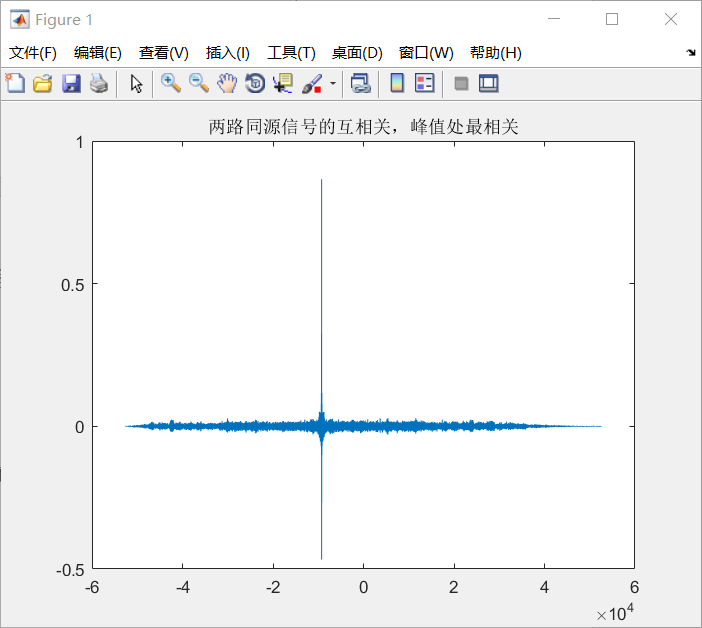


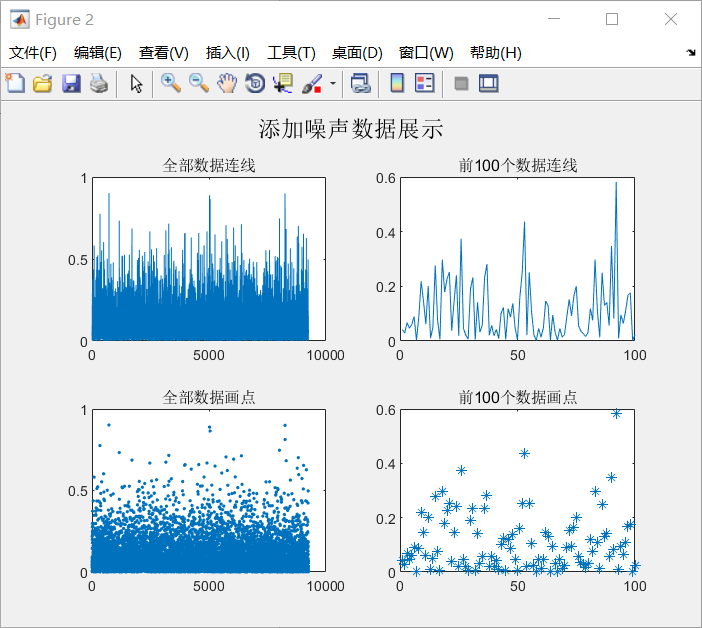
4.探究性实验

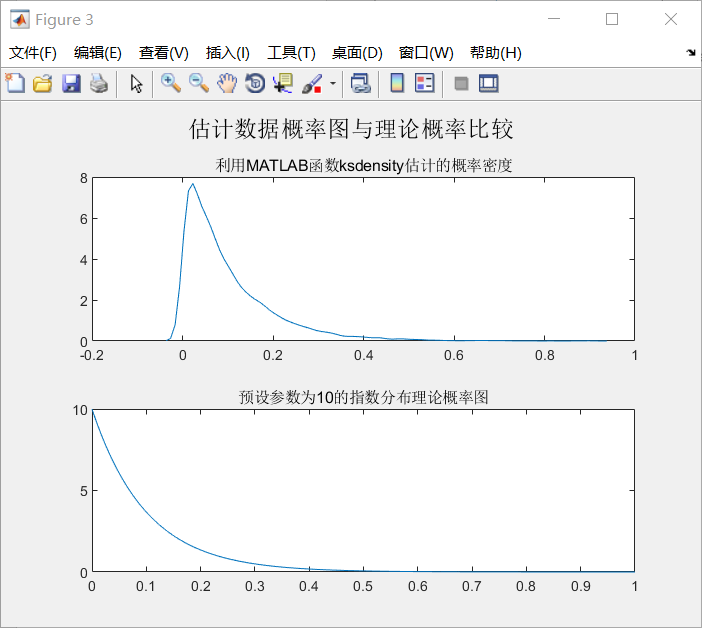
延迟时间

|  |  |
| --- | --- |
| 延迟点数 | 延迟时间/s |
| 9250 | 1.1292 |

分布类型：参数为10的指数分布







1. 实验结果与分析

（1）均匀随机数

自编函数生成的均匀随机数的均值为0.49724，均方值为 0.33062，可以算出方差为0.08340，与理论值0.08333十分接近，满足数字特征。画出的频度直方图以及数据描点进一步说明了数据的均匀性，最后最能说明的是概率密度。由于matlab内置的画概率密度函数会用一些数据平滑的方式（矩形拟合，三角拟合等）对样本点进行平滑，所以会出现虽然样本都是正值，但画出的概率密度曲线却有负值的情况。可以通过设置ksdensity的参数来优化概率密度曲线，比如将取值区间限制在[0,1]，以及改变样本的平滑方式，但只能做到优化，而不能完全解决。

（2）高斯随机数

使用自编的函数（中心极限法）产生的高斯随机数。预设参数，均值为：3,标准差为：2

实际均值为：2.9987,标准差为：1.9968。相对误差分别为：-0.043954 %, 和：-0.15955 %。误差很小，可以认为服从高斯分布。并且由画出的频度分布直方图可以进一步证明正确性。

（3）高斯分布的自相关以及互相关函数

其中自相关函数的图像能够比较好的用理论知识进行解释，为偶函数并且在0处取得最大值。比较难解释的是互相关函数为什么呈偶对称，从理论上看，互相关函数并不一定是偶对称的，但是我测了多组高斯分布的互相关函数，无一例外的都是偶对称。按我的理解，两个独立的高斯分布，其中一个高斯函数可以通过线性变换得到另一个高斯函数，所以他们俩的互相关函数实质上是自相关函数，所以有着自相关函数的性质。但也只是猜想，并没有验证。

（4）探究性实验

计算两个函数互相关函数，得到延迟点数为9250点，根据采样率算出延迟时间为1.1292s。并且由均值为0.10108，方差为0.010402猜想为参数10的指数分布，接着画出概率密度图像，发现与指数分布的图像相似。最后使用matlab自带的kstest函数进行置信度为95%的假设检验，得出结果为符合指数为10的指数分布。猜想正确。

七、讨论、建议、质疑

之前使用matlab时基本都是无脑地直接应用现成的函数，但是通过这个实验，发现追本溯源也是非常有意义的，能够发现一些理论上正确但用计算机模拟却出现偏差的原因。做完本实验，还有一些疑惑，高斯分布互相关函数为什么对称? 以及kstest函数假设检验的机理是什么? 还有待我继续进行探究。