《随机过程与统计信号处理》

第一次上机作业

# 实验问题

# 

# 问题分析

## 实验一：

# 现已知目标随机序列的概率密度：

# （1）

# 由此可求出该随机数列的理想均值与理想均差：

# （2）

# 同时可求出相应的概率分布函数及其反函数：

# 图形用户界面, 文本 描述已自动生成 （3）

# 生成随机变量 R 使其满足均匀分布：

# 图片包含 图示 描述已自动生成 （4）

# 于是根据反函数法可以生成随机变量 X：

# （5）

# （具体实现见下方代码）由此产生 1000 个样本，计算出该序列均值为 1.3353，方差为 0.2262 。而理想序列均值为 4/3，方差为 2/9，算得均值与方差的误差分别为：1.3353 – 1.3333 = 0.0020，0.2262 – 0.2222 = 0.0040 。增加样本数量后，其均值与方差更接近理想值。

## 实验二：

# 绘制序列的直方图与概率密度函数如下图所示

# 实验代码

|  |
| --- |
| clear; clc; close all;  % Inverse function method generates random sequence and probability density function  n = 1:1000;  R = rand(1,1000);  inverse\_function = @(x) sqrt(4\*x);  X = inverse\_function(R);  % mean and variance of the random sequence  m\_X = mean(X);  v\_X = var(X);  figure();  % plot the curve of the sequence  subplot(3, 1, 1);  plot(n,X);  title('Random sequence');  xlabel('n');  ylabel('X(n)');  grid on;  % plot the histogram of the sequence  subplot(3, 1, 2);  histogram(X, 100);  title('Histogram of the random sequence');  xlabel('X(n)');  ylabel('Frequency');  grid on;  % plot the probability density function  subplot(3, 1, 3);  [elements, centers] = hist(X, 100);  plot(centers, elements/sum(elements));  title('Probability density function');  xlabel('X(n)');  ylabel('f\_X(x)');  grid on; |

# 结果与分析

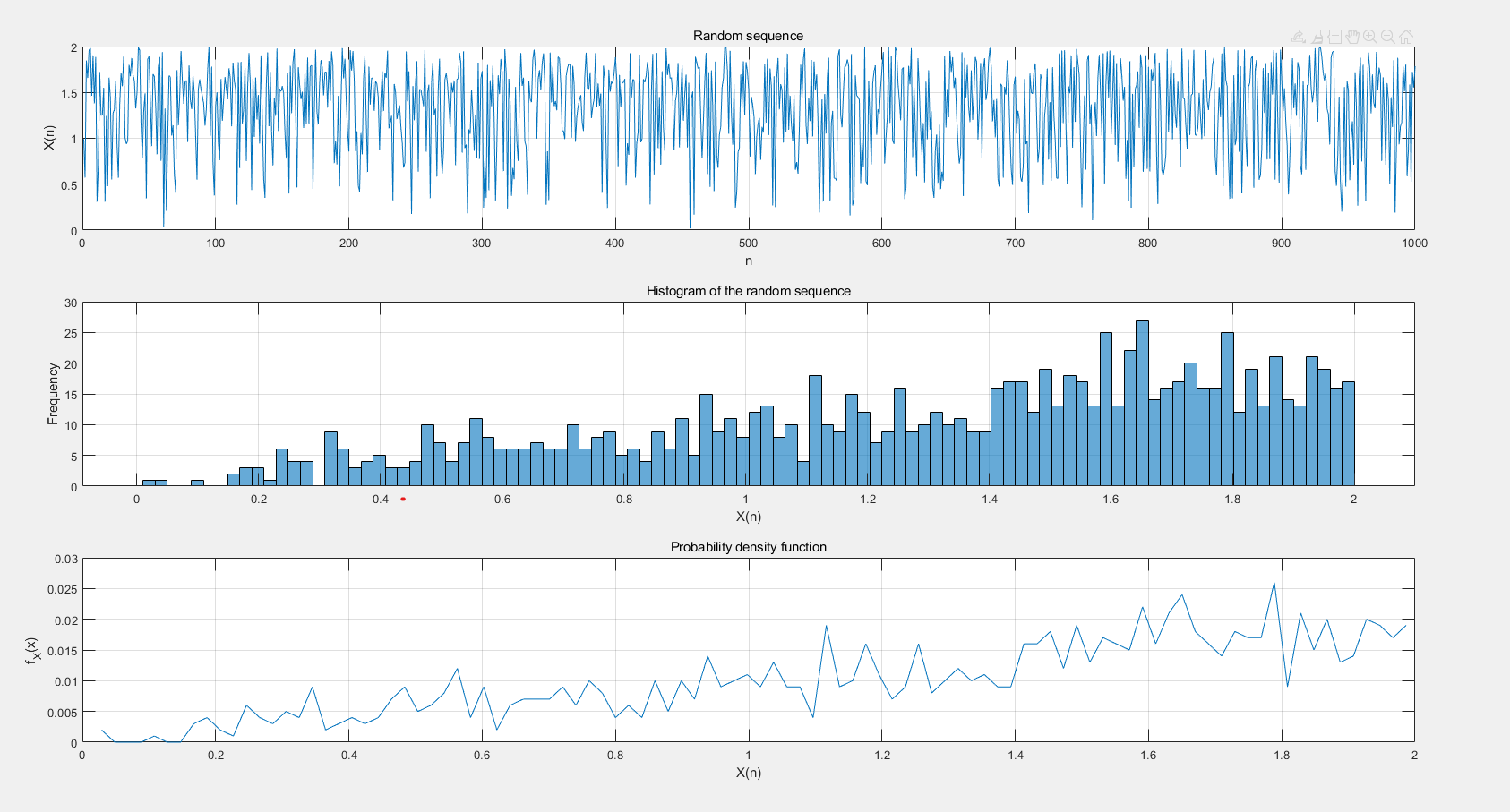
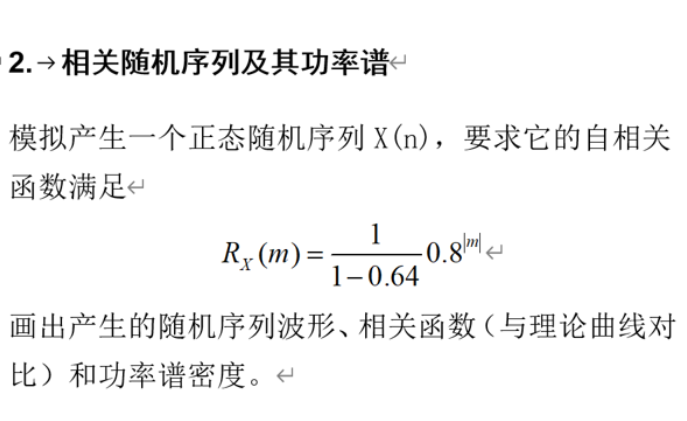


图1：从上到下依次为随机序列样本分布函数图、序列直方图藉概率密度函数

# 实验问题



# 问题分析

## 实验一：

# 根据自相关函数，可以得出该序列的均值 mX：

## （1）

# 由此可得：

## 手机屏幕截图 中度可信度描述已自动生成（2）

# 于是只需求出 相应的协方差矩阵，并对生成的正态分布白噪声进行变化即可（代码实现见下方）

# 实验代码

|  |
| --- |
| clc; clear; close all;  Sigma = 1; a = 0.8; M = 1000; m = 0:1:M;  Rx = Sigma^2 / (1 - a^2) \* a.^abs(m);  mule = 0;  % calculate covariance matrix  N = 1000;  K = zeros(N, N);  for i = 1:N  K(i, :) = cat(2, Rx(i:-1:1), Rx(2:N-i+1));  end  figure;  % generate normal distribution random sequence  subplot(3, 1, 1);  A = chol(K).';  u = randn(N, 1);  x = A \* u + mule;  plot(x);  xlabel('n'); ylabel('x(n)'); title('x(n)');  % draw autocorrelation function of x  subplot(3, 1, 2);  Rx = cat(2, Rx(N:-1:2), Rx(1:N));  plot(-N+1:N-1, Rx, 'b');  hold on;  Rx = xcorr(x, 'biased');  plot(-N+1:N-1, Rx, 'r');  hold off;  xlabel('m'); ylabel('Rx(m)'); title('Rx(m)');  legend('theoretical', 'estimated');  % draw power spectral density of x (double-sided)  subplot(3, 1, 3);  Gx = fftshift(fft(Rx));  w = linspace(-pi, pi, 2\*N-1);  plot(w, Gx);  xlabel('w'); ylabel('Gx(w)'); title('Gx(w)'); |

# 结果与分析

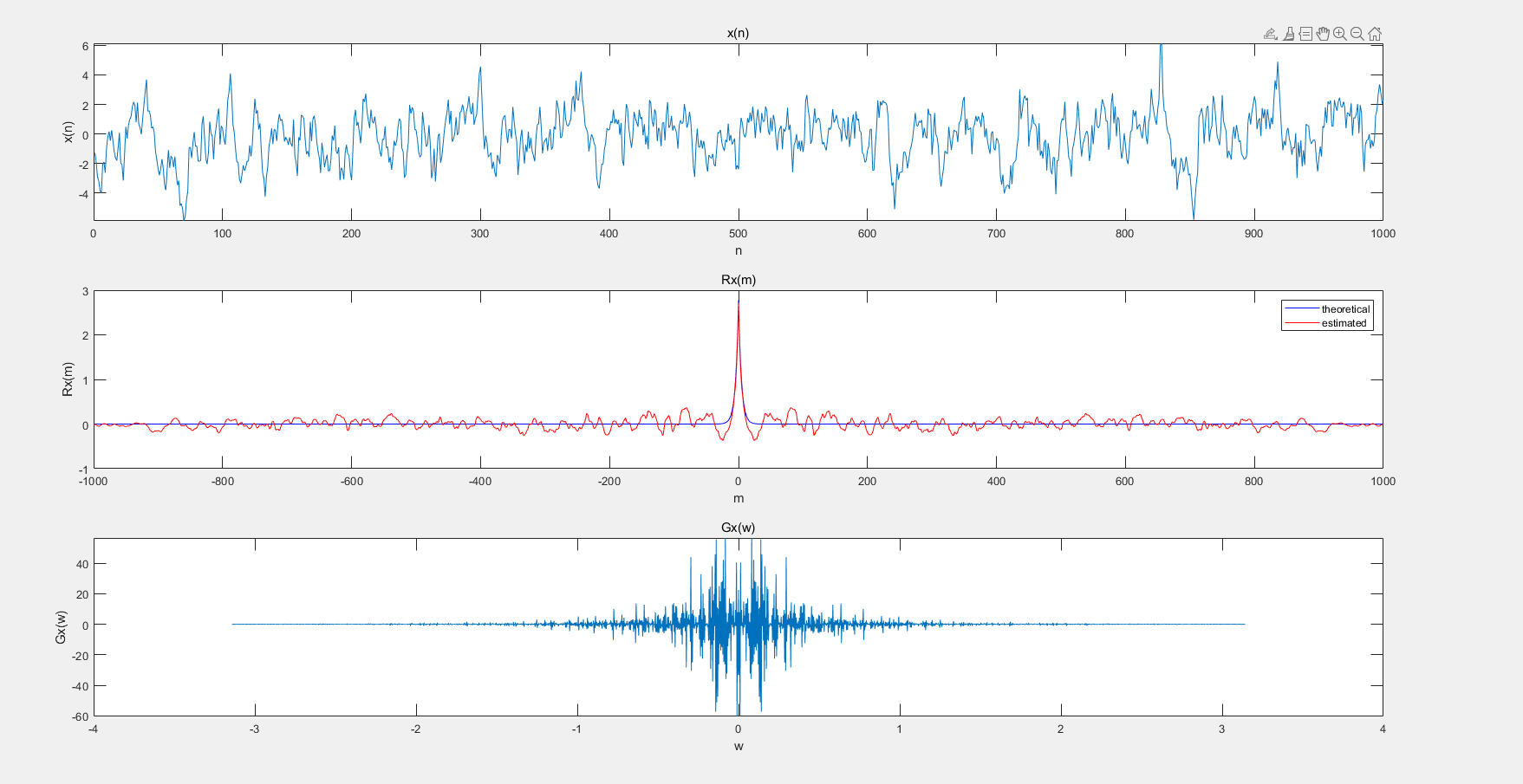


图2：从上到下分别为随机序列的波形、相关函数（及其理论曲线）和功率谱

# 实验问题

手机屏幕截图

描述已自动生成

# 问题分析

## 实验一：

# 已知有：

## （1）

# 根据上式：

## 手机屏幕的截图 中度可信度描述已自动生成（2）

## 图示 中度可信度描述已自动生成（3）

# 于是有：

## 文本 描述已自动生成（4）

# 根据上述步骤即可模拟产生 X(n)（代码实现以及样本函数波形见下方）

## 实验二：

## 样本函数均值为 0.0691，方差为 9.2570 。

## 实验三、四

# 具体实现及图像见下方

# 实验代码

|  |
| --- |
| clc; clear; close all;  %% generate X(n) by impulse response method  % generate h(n) by freqz and ifft  % - B: numerator  % - A: denominator  B = 1;  A = [1 0.8];  [H, w] = freqz(B, A, (0:2\*pi/1000:2\*pi).');  h = ifft(H);  % generate W(n) by randn  Sigma = 2; Ts = 1; Fs = 1/Ts;  W = Sigma .\* randn(500, 1) + 0;  % X(n) = W(n) \* h(n)  X = conv(W,h);  X = X(1:500);  figure, plot(abs(X)); xlabel('order'); ylabel('amplitude');  % calculate mean and variance  mX = mean(X); vX = var(X);  % theorectical Rx and Gx  Gx = abs(H) .^ 2 \* 4;  fset = (0:length(Gx)-1) \* Fs/length(Gx);  figure, plot(fset, Gx); xlabel('frequency'); ylabel('amplitude(dB)');  Rx = fftshift(ifft(Gx));  figure, plot(abs(Rx)); xlabel('order'); ylabel('amplitude'); axis([0 1000 0 12]);  % estimated Rx and Gx  Rx1 = xcorr(X);  figure, plot(abs(Rx1)); xlabel('order'); ylabel('amplitude');  nfft = 1024;  window = hann(length(X));  [Pxx, fset2] = periodogram(X, window, nfft, Fs);  figure, plot(fset2, Pxx); xlabel('frequency'); ylabel('amplitude(dB)'); |

# 结果与分析

## 实验一

## 图形用户界面, 图表 描述已自动生成

图3：X(n)的样本函数

## 图表 描述已自动生成

图4：X(n)的理论自相关函数

图形用户界面

中度可信度描述已自动生成

图5：X(n)的理论功率谱密度

图形用户界面, 图表

描述已自动生成

图6：X(n)的理论功率谱密度

直方图

描述已自动生成

图7：X(n)的功率谱密度