

随机信号分析

大作业实验报告



院 系： 信息与通信工程学院

班 级： 2019211112

学 号： 2021523016

类 别： 交流生

姓 名： 徐川峰

指导老师： 司中威

2021 年 11 月 30 日

实验 2.1 随机过程的模拟与特征估计

随机过程的特征估计是信号处理最基本的内容，通过本实验熟悉和掌握特征估计的基本方法及其 **MATLAB** 实现。实验原理参与 2.7 节，实验内容如下：

(1) 按如下模型产生一组随机序列 $x(n) = 0.8x(n-1) + w(n)$ ，其中 $w(n)$ 为均值为 1，方差为 4 的正态分布白噪声序列。估计过程的自相关函数与功率谱。

MATLAB 代码：

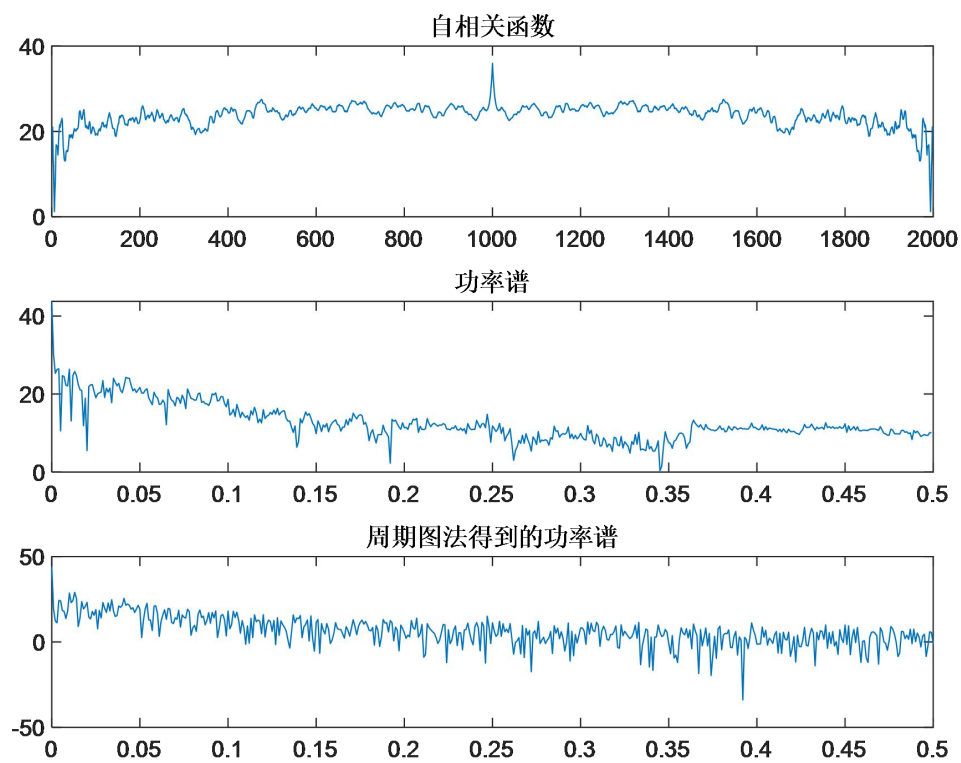
```
Fs=1; %采样频率
n=0:1/Fs:1000;%生成均值为 1 方差为 4 的正态分布白噪声序列
w=randn(1,1000);
w=w/std(w);
w=w-mean(w);
a=1; %均值为 1
b=4; %方差为 4
w=a+sqrt(b)*w;
x=zeros(1,1000);
x(1)=w(1);
for n=2:1000
    x(n)=0.8*x(n-1)+w(n);
end
nfft=1000;
cxn=xcorr(x,'unbiased'); %计算 x(n) 的自相关函数 figure(1);
subplot(3,1,1);
plot(cxn); %绘制自相关函数图
title('自相关函数')
%自相关法功率谱估计
CXk=fft(cxn,1000);
Pxx=abs(CXk);
index=0:round(nfft/2-1);
k=index*Fs/nfft;
```

```

plot_Pxx=10*log10(Pxx(index+1));
subplot(3,1,2)
plot(k,plot_Pxx);
title('功率谱');
%周期图法功率谱估计
window=boxcar(length(x));%矩形窗
[Pxx,f]=periodogram(x,window,nfft,Fs);%直接法
subplot(3,1,3)
plot(f,10*log10(Pxx))
title('周期图法得到的功率谱')

```

MATLAB 截图：



实验原理：本题要求按它所给的公式产生一个伪随机序列，在这里我利用了 `randm` 函数，用它来产生高斯白噪声，方程式为 $\omega(n) = 1 + 2 \times \text{randm}(1,1001)$ 。这样，就可以得到题目要求均值和方差的高斯白噪声。然后通过一个循环就产生了题目要求的随机序列。接着求出这个随机序列的功率谱密度，方法是使用了 `periodogram` 函数，这个函数的作用就是产生随机序列的功率谱，利用系统的 `xcorr` 函数画出其自相关函数，从而输出结果。

(2) 设信号为 $x(n) = \sin(2\pi f_1 n) + 2\cos(2\pi f_2 n) + w(n)$, $n = 1, 2, \dots, N$, 其中 $f_1 = 0.05$, $f_2 = 0.12$, $w(n)$ 为正态白噪声。试在 $N = 256$ 和 1024 点时, 分别产生随机序列 $x(n)$, 画出 $x(n)$ 的波形并估计 $x(n)$ 的相关函数和功率谱。

(N=256) MATLAB 代码:

```
Fs=1000; %采样频率
n=0:1/Fs:1;

xn=sin(2*pi*0.05*n)+2*cos(2*pi*0.12*n)+randn(size(n)); %产生 xn
的随机序列

nfft=256;

subplot(3,1,1)

plot(n,abs(xn)) %绘制 xn 的波形

title('x(n)=sin(2*pi*f1n)+2*cos(2*pi*f2n)+w(n) 的信号')

%自相关法功率谱估计

cxn=xcorr(xn,'unbiased'); %计算自相关函数

CXk=fft(cxn,nfft); %傅立叶变换

Pxx=abs(CXk);

index=0:round(nfft/2-1);

k=index*Fs/nfft;

plot_Pxx=10*log10(Pxx(index+1));

subplot(3,1,2)

plot(k,plot_Pxx) %绘制功率谱图

title('自相关法得到的功率谱')

%周期图法功率谱估计

window=boxcar(length(xn)); %矩形窗

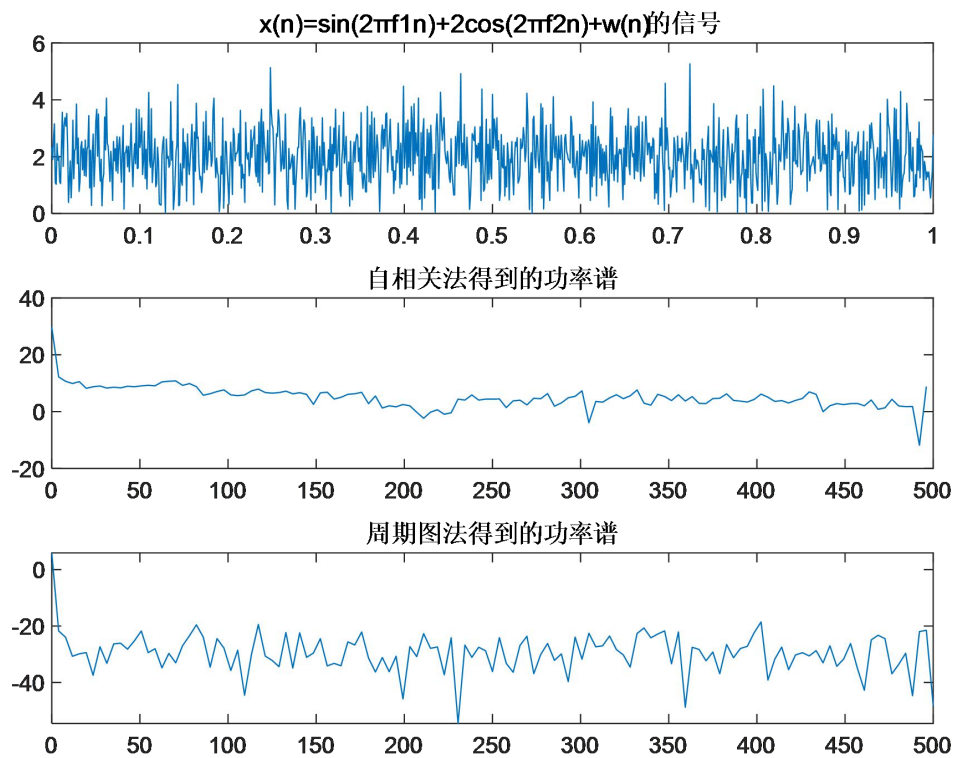
[Pxx,f]=periodogram(xn,window,nfft,Fs);

subplot(3,1,3)

plot(f,10*log10(Pxx))

title('周期图法得到的功率谱')
```

MATLAB 截图：



(N=1024) MATLAB 代码：

```
Fs=1000; %采样频率
n=0:1/Fs:1;
xn=sin(2*pi*0.05*n)+2*cos(2*pi*0.12*n)+randn(size(n)); %产生 xn
的随机序列
nfft=1024;
subplot(3,1,1)
plot(n,abs(xn)) %绘制 xn 的波形
title('x(n)=sin(2 π f1n)+2cos(2 π f2n)+w(n) 的信号')
%自相关法功率谱估计
cxn=xcorr(xn,'unbiased'); %计算自相关函数
CXk=fft(cxn,nfft); %傅立叶变换
Pxx=abs(CXk);
index=0:round(nfft/2-1);
```

```

k=index*Fs/nfft;

plot_Pxx=10*log10(Pxx(index+1));

subplot(3,1,2)

plot(k,plot_Pxx) %绘制功率谱图

title('自相关法得到的功率谱')

%周期图法功率谱估计

window=boxcar(length(xn)); %矩形窗

[Pxx,f]=periodogram(xn,window,nfft,Fs);

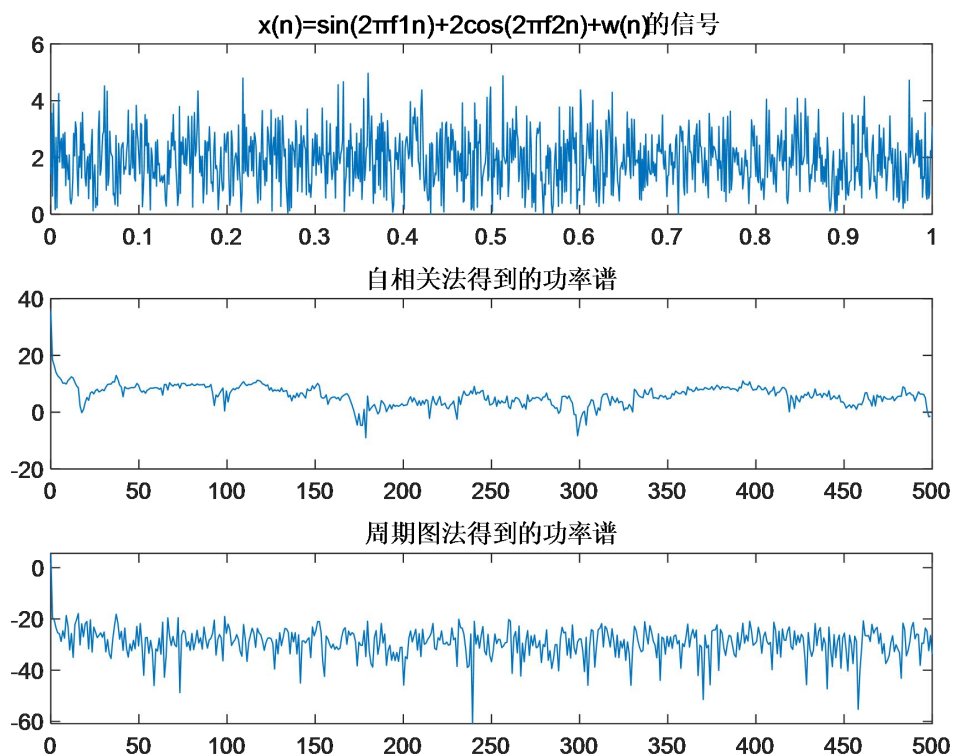
subplot(3,1,3)

plot(f,10*log10(Pxx))

title('周期图法得到的功率谱')

```

MATLAB 截图：



实验原理：本题的要求是在 $N = 256$ 和 $N = 1024$ 处分别产生随机序列，并分别画出它们的功率谱密度和自相关函数，这里已知 $w(n)$ 是均值为 0，方差为 1 的高斯白噪声，所以我用生成高斯分布的函数 `randn` 直接产生高斯白噪声，然后产生所要求的随机序列，并将其画出。接下来利用 `periodogram` 函数求出了功率谱密度，它的作用是实现周期图

的功率谱估计，这样就估计出了它的功率谱密度。然后再次使用了 `xcorr` 函数，实现了对自相关函数的求解。这两次实验我使用了类似于 C 语言中的全局变量性质的变量 `K`，这样可以在不改变源程序的情况下，只需要改变 `K`，就可以求出任意 `N` 值处的随机序列，并对他们分析，这给我们带来了巨大的方便。

比较两次实验结果，在不同的 `N` 处，得到了两组不同的图，可以看出，在 `N` 较大时，图比较密集，将它放大后可以看出，当 `N` 较大时得到的结果精度较高，但是图的整体形状基本不变，只有一些细小的差别。由于 `N` 小的时候精度不够而没有体现出来，但是从总体上看，两图还是基本上是一致的，都表示出了这个随机序列的特性。

（PS）实验 2.2 数字图像直方图均衡

2.8.2 节介绍了一个信号处理应用实例——图像直方图均衡，本实验用 **MATLAB** 实现均衡算法，请自己找一幅图像（可以从 **MATLAB** 图像工具箱找一幅例子图像），画出原始图像的直方图和经过均衡处理后的直方图。

MATLAB 代码：

```
H= imread('lzy.jpg');

%判断是否为三通道彩色图片 若是 则将其灰度化

if length(size(H))>2

    H=rgb2gray(H);

end

%获取图片的尺寸 便于计算总像素数 即 m*n

[m,n]=size(H);

%生成一个一行 256 列的矩阵

p=zeros(1,256);

% 统计各灰度的像素个数

%find(H==i) 是在图像矩阵里面寻找灰度为 i 的点坐标

% 因为矩阵是从 1 开始的 所以为 p(i+1)

for i=0:255

    p(i+1)=length(find(H==i))/(m*n);

end

subplot(2,2,1);
```

```

imshow(H);

title('原图');

subplot(2,2,2);

% 显示原图的直方图
bar(0:255,p,'b');

title('原图直方图');

% 利用循环 累加概率值
s=zeros(1,256);

for i=1:256
    for j=1:i
        s(i)=p(j)+s(i);
    end
end

%对 s 中的数先乘以 255，再取整
a=round(s*255);

b=H;

%更新原图像的灰度
for i=0:255
    b(find(H==i))=a(i+1);
end

subplot(2,2,3);

imshow(b)

title('均衡化后图像');

%统计更新后的概率
for i=0:255
    GPeq(i+1)=sum(p(find(a==i)));
end

subplot(2,2,4);

bar(0:255,GPeq,'b'); title('均衡化后的直方图');

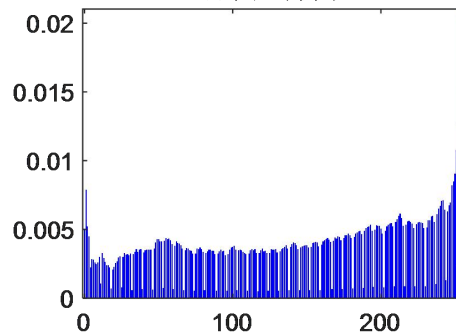
```


MATLAB 截图：

原图



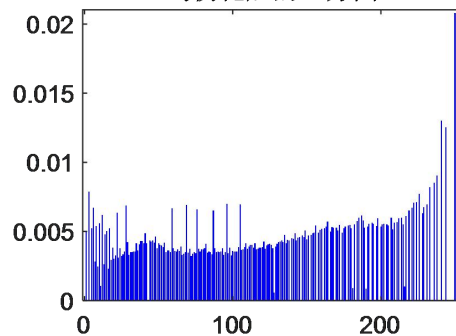
原图直方图



均衡化后图像



均衡化后的直方图



结果分析:把原始图像的灰度直方图从比较集中的某个灰度区间变成在全部灰度范围内的均匀分布。直方图均衡化就是对图像进行非线性拉伸,重新分配图像像素值,使一定灰度范围内的像素数量大致相同。直方图均衡化就是把给定图像的直方图分布改变成“均匀”分布直方图分布(把原来的图像的灰度分配均匀,使得 0-255 都有一定的取值,这样对比度相对大一些,视觉上更好看一点)