

大连理工大学

本科实验报告三

课程名称： 随机信号分析实验

学院（系）： 电子信息与电气工程学部

专 业： 电子信息工程

班 级： 电信 1806 班

学 号： 201871080

学生姓名： 刘祎铭

2020 年 12 月 3 日

大连理工大学实验预习报告

学院（系）： 电子信息与电气工程学部 专业： 电子信息工程 班级： 电信 1806

姓 名： 刘祎铭 学号： 201871080 组：

实验时间： 2020.11.25 实验室： 创新园 C220 实验台：

指导教师： 李小兵

实验 III：通信信号统计特性分析、仿真实现及应用

一、实验目的和要求

掌握 Hilbert 变换的时域/频域实现方法；掌握窄带随机过程样本生成方法；掌握窄带随机过程及其低频分量的功率估计方法；掌握高斯窄带随机过程包络和相位的分布估计方法。

二、实验原理和内容

（一）实验原理

1. Hilbert 变换的实现方法

Hilbert 变换系统传递函数为
$$H(e^{j\omega}) = \begin{cases} -j & 0 < \omega < \pi \\ j & -\pi < \omega < 0 \end{cases}$$
 其时域等效系统单位脉冲响应为

应为
$$h(n) = \begin{cases} 0 & n = 2k \\ \frac{2}{n\pi} & n = 2k+1, n = -\infty, \dots, \infty. \end{cases}$$
 $x(n)$ 的 Hilbert 变换相当于为 $x(n)$ 与 $h(n)$ 的卷积，即 Hilbert 变换可以看作是一个冲激响应为 $h(n)$ 的线性移不变网络。工程上实现时，需要将 $h(n)$ 截短，右移(因果化)后才行。

2. 窄带随机过程样本生成方法

窄带随机过程可以表示为下面的准正弦振荡的形式 $X(t) = A(t)\cos[\omega_0 t + \Phi(t)]$ ，或者表示为同相分量与正交分量的合成： $X(t) = A_c(t)\cos\omega_0 t - A_s(t)\sin\omega_0 t$ ，其中 $A_c(t)$ 与 $A_s(t)$ 均为低频变化的随机过程，可以通过模拟其分布及功率谱特性来实现窄带随机过程的产生。

3. 随机过程及其低频分量的功率估计方法

先产生随机信号样本 $X(t) = A_c(t) \cos \omega_0 t - A_s(t) \sin \omega_0 t$ ，再利用变换列方程相干解调求得 $A_c(t)$ 与 $A_s(t)$ ，再估计此窄带随机过程及其低频过程 $A_c(t)$ 、 $A_s(t)$ 的功率谱密度。

4. 高斯窄带随机过程包络和相位的分布估计方法

先产生随机信号样本 $X(t) = A_c(t) \cos \omega_0 t - A_s(t) \sin \omega_0 t$ ，再利用变换列方程相干解调求得 $A_c(t)$ 与 $A_s(t)$ ，进一步求得包络和相位 $A(n) = \sqrt{A_c^2(n) + A_s^2(n)}$ ， $\Phi(n) = \arctan \frac{A_s(n)}{A_c(n)}$ 。

(二) 实验内容

1. Hilbert 变换的实现方法

生成 20000 点正弦波信号数据，与给定的冲激响应函数卷积或通过频域方法，得到 Hilbert 变换信号，并验证产生数据的正确性，利用 MATLAB 函数 hilbert 重新计算变换值。

Hilbert 变换等效系统单位响应为 $h(n) = \begin{cases} 0 & n = 2k \\ \frac{2}{n\pi} & n = 2k+1 \end{cases}$ ，其等效系统函数为

$$H(e^{j\omega}) = \begin{cases} -j & 0 < \omega < \pi \\ j & -\pi < \omega < 0 \end{cases}$$

2. 随机过程及其低频分量的功率估计方法

生成 10000 点中心频率 10KHz、带宽 500Hz 的窄带随机信号 $X(t)$ ，估计此窄带随机过程及其低频过程 $A_c(t)$ 、 $A_s(t)$ 的功率谱密度。

3. 高斯窄带随机过程包络和相位的分布估计方法

生成 10000 点中心频率 10KHz、带宽 500Hz、方差为 1 的窄带随机信号 $X(t)$ ，计算此窄带随机过程包络 $A(t)$ 和相位 $\Phi(t)$ 以及包络平方 $A^2(t)$ ，画出它们的分布直方图，并与理论结论做比较。

三、实验步骤

实验一

1. 生成正弦信号样本

```
% N=100 点正弦信号 1000Hz，采样率 16000Hz  
  
N = 20000;  
  
f = 1000;  
  
fs = 1600000;  
  
sinData = sin(2*pi*f/fs*(1:N));  
  
figure, plot(sinData)  
  
title('100 点正弦信号 1000Hz，采样率 16000Hz')  
  
%axis([0 1000 -1 1]);
```

2. 设计 hilbert 滤波器

```
%时域冲激响应  
  
step = 30000;  
  
h = zeros(1,step);  
  
for i = 1 : step  
  
    ind = bitand(abs(i-step/2),1);  
  
    if(ind == 1)  
  
         $h(i) = 2/\pi/(i-step/2);$   
  
    end  
  
end
```

```
% show h(n)

figure,subplot(2,1,1),stem(h)

title('时域冲激响应')

%频域传递函数

H = zeros(1,N);

for p = 1 : N

    if(p<= N/2)

        H(p) = -j;

    else

        H(p) = j;

    end

end

% show H(p)

subplot(2,1,2),stem(imag(H))

%axis([9975 10025 -1 1]);

title('频域冲激响应')
```

3.1 时域变换卷积运算得到 cos 数据

```
%时域卷积

cosData = conv(sinData,h,'same');

figure,subplot(2,1,1),plot(cosData,'r')

%axis([0 1000 -1 1]);
```

```
title('时域变换卷积运算得到 cos 数据')
```

3.2 确认相位差 $\pi/2$ 自相关确定延迟点数 D

```
[c,l]=xcorr(sinData,cosData,'coeff');
```

```
% figure,stem(l,c)
```

```
[m,indx] = max(c);
```

```
D = abs(indx - N) + 1
```

3.3 变换前后数据对比

```
subplot(2,1,2),plot(sinData);hold on
```

```
plot([cosData(D:end) zeros(1,D-1)],'r')
```

```
title('变换前后数据对比')
```

```
%axis([0 1000 -1 1]);
```

```
legend('变换前','变换后')
```

3.4D 转换成弧度 找到周期

```
[C,I]=find(sinData==1);
```

```
T = I(2)-I(1)
```

```
%%相位差
```

```
delta = D/T;
```

```
disp(['相位差为: ',num2str(delta*2),'pi'])
```

4.1 频域变换乘积得到 cos 数据

```
%域乘积后反变换
```

```
cosData = real(ifft(fft(sinData).*H));
```

```
% figure,plot(sinData);hold on
```

```
figure,subplot(2,1,1),plot(cosData,'r')
```

```
%axis([0 1000 -1 1]);
```

```
title('频域变换乘积得到 cos 数据')
```

4.2 确认相位差 $\pi/2$ 自相关确定延迟点数 D

```
[c,l]=xcorr(sinData,cosData,'coeff');
```

```
% figure,stem(l,c);
```

```
[m,indx] = max(c);
```

```
D = abs(indx - N) ;
```

4.3 变换前后数据对比

```
subplot(2,1,2),plot(sinData);hold on
```

```
plot([cosData(D:end) zeros(1,D-1)],'r')
```

```
title('变换前后数据对比')
```

```
%axis([0 1000 -1 1]);
```

```
legend('变换前','变换后')
```

4.4 D 转换成弧度 找到周期

```
[C,I]=find(sinData==1);
```

```
T = I(4)-I(3)
```

4.5 相位差

```
delta = D/T;
```

```
disp(['相位差为: ',num2str(delta*2),'pi'])
```

5.1 对比内置函数 hilbert

```
cosData = imag(hilbert(sinData));  
  
figure;subplot(2,1,1),plot(cosData,'r')  
  
%axis([0 1000 -1 1]);  
  
title('内置函数 hilbert 变换数据')
```

5.2 确认相位差 $\pi/2$ 自相关确定延迟点数 D

```
[c,l]=xcorr(sinData,cosData,'coeff');  
  
% figure,stem(l,c)  
  
[m,indx] = max(c);  
  
D = abs(indx - N)
```

5.3 变换前后数据对比

```
subplot(2,1,2),plot(sinData);hold on  
  
plot([cosData(D:end) zeros(1,D-1)],'r')  
  
title('变换前后数据对比')  
  
%axis([0 1000 -1 1]);  
  
legend('变换前','变换后')
```

5.4 D 转换成弧度 找到周期

```
[C,I]=find(sinData==1);  
  
T = I(2)-I(1)
```

5.5 相位差

```
delta = D/T;
```



```
disp(['相位差为: ',num2str(delta*2),'pi'])
```

实验二

1.1 生成窄带随机信号

```
N=10000;f0=10000;deltf=500;fs=22000;M=200; %调用参数设置
X=NarrowBand(N,f0,deltf,fs,M); %调用产生窄带随机信号
的函数
```

```
function X=Narrowbandsignal(N,f0,deltf,fs,M)%产生窄带随机过程
N1=N-M;%N 为要产生样本个数，deltf 表示信号的带宽，f0 随机过程单边
功率谱的中心频率为 f0
xt=random('norm',0,1,[1,N1]);%fs 表示信号采样频率，M 为产生宽带信号滤
波器阶数
f1=f0*2/fs;%输出 N 个再带随机信号样本
df1=deltf/fs;%N-M 个高斯随机数 %归一化中心频率 %归一化带宽
ht=fir1(M,[f1-df1 f1+df1]);%ht 为带通滤波器的冲击响应，M 为阶数
X=conv(xt,ht);%输出 N 个窄带随机信号样本采样
return
```

1.2 展示窄带数据

```
%时域
```

```
figure,
subplot(2,1,1);plot(X);
title('时域波形')
```

% 功率谱

```
subplot(2,1,2);plot( periodogram( X));
```

```
title('窄带信号功率谱密度')
```

2. 自相关函数

```
[Rx,lag]=xcorr(X,'biased');
```

%窄带随机信

号样本的自相关函数

```
figure,plot(lag,Rx);
```

```
title('窄带信号自相关函数')
```

3.1 取得 $A_c(t)$ 、 $A_s(t)$ 分量

```
[Ac, As]=Lowfsignal(X,f0,fs);
```

%调用产生 $A_c(t)$ 、

$A_s(t)$ 的函数

```
function [Ac As]=Lowfsignal(X,f0,fs)
```

%产生低频过程 A_c 和 A_s 的样本

```
HX=imag(hilbert(X));
```

```
[M N]=size(X);
```

```
t=0:1/fs:((N-1)/fs);
```

```
Ac=X.*cos(2*pi*f0*t)+HX.*sin(2*pi*f0*t);
```

```
As=HX.*cos(2*pi*f0*t)-X.*sin(2*pi*f0*t);
```

```
return
```

3.2 展示 $A_c(t)$ 、 $A_s(t)$ 分量时域波形

```
%Ac(t)
figure,
subplot(2,1,1);plot(Ac);
title('Ac(t)时域波形')
% As(t)
subplot(2,1,2);plot(As);
title('As(t)时域波形')
```

3.3 展示窄带、 $A_c(t)$ 、 $A_s(t)$ 分量自相关函数

```
Rac=xcorr(Ac,'biased');           %低频过程  $A_c(t)$ 样
本的自相关函数
Ras=xcorr(As,'biased');           %低频过程  $A_s(t)$ 样
本的自相关函数
Racs=xcorr(As,Ac,'biased');       %低频过程
 $A_s(t)$ 样本的自相关函数
Racw=abs(fft(Rac));               %低频过程  $A_c(t)$ 
样本的功率谱密度
Rasw=abs(fft(Ras));               %低频过程  $A_s(t)$ 
样本的功率谱密度
Rxw=abs(fft(Rx));                 %窄带随机信号
样本的功率谱密度
```

```
N1=2*N-1;
```

```
f=fs/N1:fs/N1:fs/2;
```

%频率轴的变换

4. 自相关函数

```
figure,subplot(4,1,1);plot(Rx);
```

```
title('窄带随机信号样本的自相关函数');
```

```
subplot(4,1,2);plot(Rac);
```

```
title('低频过程  $A_c(t)$  样本的自相关函数');
```

```
subplot(4,1,3);plot(Ras);
```

```
title('低频过程  $A_s(t)$  样本的自相关函数');
```

```
subplot(4,1,4);plot(Racs);
```

```
title('低频过程  $A_s(t)A_c(t)$  样本的互相关函数');
```

5. 展示窄带、 $A_c(t)$ 、 $A_s(t)$ 分量功率谱密度

%功率谱密度

```
figure,subplot(4,1,1);plot(f,10*log10(Rxw(1:(N1-1)/2)+eps));
```

```
title('窄带随机信号样本的功率谱密度');
```

```
subplot(4,1,2);plot(f,10*log10(Racw(1:(N1-1)/2)+eps));
```

```
title('低频过程  $A_c(t)$  样本的功率谱密度');
```

```
subplot(4,1,3);plot(f,10*log10(Rasw(1:(N1-1)/2)+eps));
```

```
title('低频过程  $A_s(t)$  样本的功率谱密度');
```

```
xxx=[10*log10(Racw(1:(N1-1)/2)+eps)',10*log10(Rasw(1:(N1-1)/2)+eps)'];
```

```
[p,q]=chi2test(xxx);
```

```
AL=10*log10(Racw(1:(N1-1)/2)+eps);
```

```
ALL=10*log10(Rasw(1:(N1-1)/2)+eps);
```

```
Av=AL(1:9000)-ALL(1:9000);
```

```
subplot(4,1,4);plot(Av);
```

p

6.

```
function [p, Q]= chi2test(x)
```

```
% Usage: [p, Q]= chi2test(x)
```

```
%
```

```
% The chi-squared test.
```

```
%
```

```
% Given a number of samples this function tests the hypothesis that the samples  
are
```

```
% independent. If  $Q > \chi^2(p, nu)$ , the hypothesis is rejected.
```

```
%
```

```
% Each column represents a variables, each row a sample.
```

```
%
```

% If you find any errors, please let me know: .

%

% ARGUMENTS:

% x Absolut numbers.

% p The prob ability value, calculated from Q.

% Q The resulting Q-value.

%

% EXAMPLE 1

% In region A, 324 of 556 cows were red, whereas in region B 98 of 260 were red.

% [p, Q]= chi2test([324, 556-324; 98, 260-98])

% p=

% 4.2073e-08

% Q=

% 30.0515

% With an error risk of about 4e-08, we can claim that the samples are independent.

%

% EXAMPLE 2

% Throw two different dices to see if they have the same probability of 1 (and 2, 3, 4, 5, 6).

% We don't check if they are symmetrical, only if the both behave in the same

way.

```
% [p,Q] = chi2test([15,10; 7,11; 9,7; 20,15; 26,21; 19,16])
```

```
% p=
```

```
%      0.8200
```

```
% Q =
```

```
%      2.2059
```

```
% The dices don't significantly behave differently. That is, they seem to behave  
in the same way.
```

```
%
```

```
% HISTORY:      v.1.0, first working version, 2007-08-30.
```

```
%
```

```
% COPYRIGHT:   (c) 2007 Peder Axensten. Use at own risk.
```

```
% KEYWORDS:    chi-squared test, chi-squared, chi2, test
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%
```

```
% Check the arguments.
```

```
if(nargin ~= 1),          error('One and only one argument required!');
```

```
end
```

```

if(ndims(x) ~= 2),          error('The argument (x) must be a 2d matrix!');
    end

if(any(size(x) == 1)),      error('The argument (x) must be a 2d matrix!');
    end

if(any(~isreal(x))),        error('All values of the argument (x) must be real
values!'); end

% Calculate Q = sum( (a-np*)^2/(np*(1-p*)) )

s=    size(x, 1);

r=    size(x, 2);

np=    sum(x, 2)/sum(sum(x)) * sum(x);    % p=sum(x, 2)/sum(sum(x))
and n=sum(x)

Q=    sum(sum((x-np).^2./(np)));

% Calculate cdf of chi-squared to Q. Degrees of freedom, v, is (r-1)*(s-1).

p=    1 - gammainc(Q/2, (r-1)*(s-1)/2);

end

```

实验三

1.1 生成窄带随机信号

```

N=20000;f0=10000;deltf=500;fs=22000;M=50;    %调用参数设置

x= NarrowBand(N,f0,deltf,fs,M);                %调用产生窄带随机信号的函数

```



```

function X=NarrowbandSignal(N,f0,deltf,fs,M)%产生窄带随机过程
N1=N-M;%N 为要产生样本个数，deltf 表示信号的带宽，f0 随机过程单边
功率谱的中心频率为 f0
xt=random('norm',0,1,[1,N1]);%fs 表示信号采样频率，M 为产生宽带信号滤
波器阶数
f1=f0*2/fs;%输出 N 个窄带随机信号样本
df1=deltf/fs;%N-M 个高斯随机数    %归一化中心频率    %归一化带宽
ht=fir1(M,[f1-df1 f1+df1]);%ht 为带通滤波器的冲击响应，M 为阶数
X=conv(xt,ht);%输出 N 个窄带随机信号样本采样
return

```

1.2 展示窄带数据

%时域

figure,

subplot(2,1,1);plot(x);

title('时域波形')

% 功率谱

subplot(2,1,2);plot(periodogram(x));

title('窄带信号功率谱密度')

x=x/sqrt(var(x));

%归一化方差处理窄带高斯

过程 $X(t)$

2.1 取得窄带随机信号的包络、相位、包络平方

[At, Ph, A2]=EnvelopPhase(x,f0,fs); %调用 A(t)、包络平方产生函数

```
function [At Ph A2]=EnvelopPhase(X,f0,fs)
```

```
HX=imag(hilbert(X));
```

```
[M N]=size(X);
```

```
t=0:1/fs:((N-1)/fs);
```

```
Ac=X.*cos(2*pi*f0*t)+HX.*sin(2*pi*f0*t);
```

```
As=HX.*cos(2*pi*f0*t)-X.*sin(2*pi*f0*t);
```

```
Ph=atan(As./Ac);
```

```
A2=Ac.^2+As.^2;
```

```
At=sqrt(A2);
```

2.2 展示窄带随机信号的包络、相位、包络平方的分布

%包络 A(t)样本值的分布直方图

```
LA =0:0.05:4.5;
```

%包络样本值的分布区间

```
figure;subplot(3,1,1);hist(At,LA);
```

%包络 A(t)样本值的分布直方图

```
disp('')
```

```
p_judge(At',0.001);
```

```
title('包络 A(t)样本值的分布直方图');
```

```
LP=-pi/2:0.05:pi/2;
```

%相位样本值的分布区间

```
subplot(3,1,2);hist(Ph,LP);
```

%相位样本值的分布直方图

```
disp('')
```

```
p_judge(Ph',0.0001);
```

```
title('相位样本值的分布直方图');
```

```
LA2=0:0.2:16; %包络平方值的分布区间
subplot(3,1,3);hist(A2,LA2); %包络平方值的分布直方图
disp(' ')
p_judge(A2',0.001);
title('包络平方值的分布直方图');
```

4.KS 检验分布函数

```
function p_judge(A,alpha)
[mu,sigma]=normfit(A);
p1=normcdf(A,mu,sigma);
[H1,s1]=kstest(A,[A,p1],alpha);
n=length(A);
if H1==0
disp('该数据源服从正态分布。')
else
disp('该数据源不服从正态分布。')
end
if H1==1
[mu,sigma]=unifit(A);
p1=unifcdf(A,mu,sigma);
[H6,s6]=kstest(A,[A,p1],alpha);
n=length(A);
if H6==0
```

```
disp('该数据源服从均匀分布。')

else

disp('该数据源不服从均匀分布。')

end

if H6==1

H2=1;

try

mu=expfit(A,alpha);

p2=expcdf(A,mu);

[H2,s2]=kstest(A,[A,p2],alpha);

end

if H2==0

disp('该数据源服从指数分布。')

else

disp('该数据源不服从指数分布。')

end

if H2==1

H3=1;

try

lamda=poissfit(A,alpha);

p3=poisscdf(A,lamda);

[H3,s3]=kstest(A,[A,p3],alpha);
```

```
end

if H3==0

disp('该数据源服从泊松分布。')

else

disp('该数据源不服从泊松分布。')

end

if H3==1


H4=1;

try

phat=gamfit(A,alpha);

p4=gamcdf(A,phat(1),phat(2));

[H4,s4]=kstest(A,[A,p4],alpha);

end

if H4==0

disp('该数据源服从  $\gamma$  分布。')

else

disp('该数据源不服从  $\gamma$  分布。')

end

if H4==1

[phat, pci] = raylfit(A, alpha);

p5=raylcdf(A,phat);
```

```

[H5,s5]=kstest(A,[A,p5],alpha);

if H5==0

disp('该数据源服从 rayleigh 分布。')

else

disp('该数据源不服从 rayleigh 分布。')

end

end

end

end

end

end

```

实验中用到的几个自定义函数

1. 产生窄带随机过程样本 Narrowbandsignal

函数 Narrowbandsignal(N,f0,deltf,fs,M)的功能是产生窄带随机过程样本。形参 N 为要产生样本的个数，f0 表示窄带随机过程单边功率谱的中心频率为 f_0 ，deltf 表示信号的带宽，fs 表示信号采样频率，M 为产生窄带信号的滤波器阶数，必须满足 $M \ll N$ 。

```

function X=Narrowbandsignal(N,f0,deltf,fs,M)%产生窄带随机过程
N1=N-M;%N 为要产生样本个数，deltf 表示信号的带宽，f0 随机过程单边功率谱的
中心频率为 f0
xt=random('norm',0,1,[1,N1]);%fs 表示信号采样频率，M 为产生宽带信号滤波器阶数
f1=f0*2/fs;%输出 N 个再带随机信号样本
df1=deltf/fs;%N-M 个高斯随机数 %归一化中心频率 %归一化带宽
ht=fir1(M,[f1-df1 f1+df1]);%ht 为带通滤波器的冲击响应，M 为阶数
X=conv(xt,ht);%输出 N 个窄带随机信号样本采样
return

```

2. 产生 Ac, As 样本 Lowfsignal

函数 Lowfsignal(X,f0,fs)用于产生 Ac 和 As。形参 X 为要提取随机过程 X(t)的样本，

f_0 表示窄带随机过程 $X(t)$ 单边功率谱的中心频率 f_0 ； f_s 表示要信号采样频率 f_s 。

```
function [Ac As]=Lowfsignal(X,f0,fs)
%产生低频过程 Ac 和 As 的样本
HX=imag(hilbert(X));
[M N]=size(X);
t=0:1/fs:(N-1)/fs;
Ac=X.*cos(2*pi*f0*t)+HX.*sin(2*pi*f0*t);
As=HX.*cos(2*pi*f0*t)-X.*sin(2*pi*f0*t);
return
```

3. 产生 A_t , Ph 和 A_2 样本 EnvelopPhase

函数 EnvelopPhase(X, f_0, f_s)，用来产生 A_t , Ph 和 A_2 。形参 X 为要提取的随机过程 $X(t)$ 的样本， f_0 表示窄带随机过程 $X(t)$ 单边功率谱的中心频率 f_0 ， f_s 表示要信号采样频率 f_s 。

```
function [At Ph A2]=EnvelopPhase(X,f0,fs)
HX=imag(hilbert(X));
[M N]=size(X);
t=0:1/fs:(N-1)/fs;
Ac=X.*cos(2*pi*f0*t)+HX.*sin(2*pi*f0*t);
As=HX.*cos(2*pi*f0*t)-X.*sin(2*pi*f0*t);
Ph=atan(As./Ac);
A2=Ac.^2+As.^2;
At=sqrt(A2);
```

四、实验数据记录表格