

大连理工大学

本科实验报告二

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

2.1 实验原理

2.1 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

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。

2.2 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。

2.3 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。

2.4 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 $A(n) = \sqrt{A_c^2(n) + A_s^2(n)}$ ， $\Phi(n) = \arctan \frac{A_s(n)}{A_c(n)}$ 。

2.2 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、主要仪器设备

Windows10(64 位)，MATLABR2016a

四、实验步骤与操作方法

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

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
```

```
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 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
```

3.2 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、Ac(t)、As(t)分量自相关函数

```
Rac=xcorr(Ac,'biased');           %低频过程 Ac(t)样本的自相关函数
Ras=xcorr(As,'biased');           %低频过程 As(t)样本的自相关函数
Racs=xcorr(As,Ac,'biased');       %低频过程 As(t)样本的自相关函数
Racw=abs(fft(Rac));               %低频过程 Ac(t)样本的功率谱密度
Rasw=abs(fft(Ras));               %低频过程 As(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 Ac(t)样本的自相关函数');
subplot(4,1,3);plot(Ras);
title('低频过程 As(t)样本的自相关函数');
subplot(4,1,4);plot(Racs);
title('低频过程 As(t)Ac(t)样本的互相关函数');
```

5. 展示窄带、Ac(t)、As(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 Ac(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 As(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 > chi2(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 symetrical, 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

fl=f0*2/fs;%输出 N 个再带随机信号样本

dfl=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
end
```

五、 实验数据记录和处理

实验一 希尔伯特变换的实现方法

D =

398

$T =$

1600

相位差为：0.4975pi

$T =$

1600

相位差为：0.49875pi

$D =$

399

$T =$

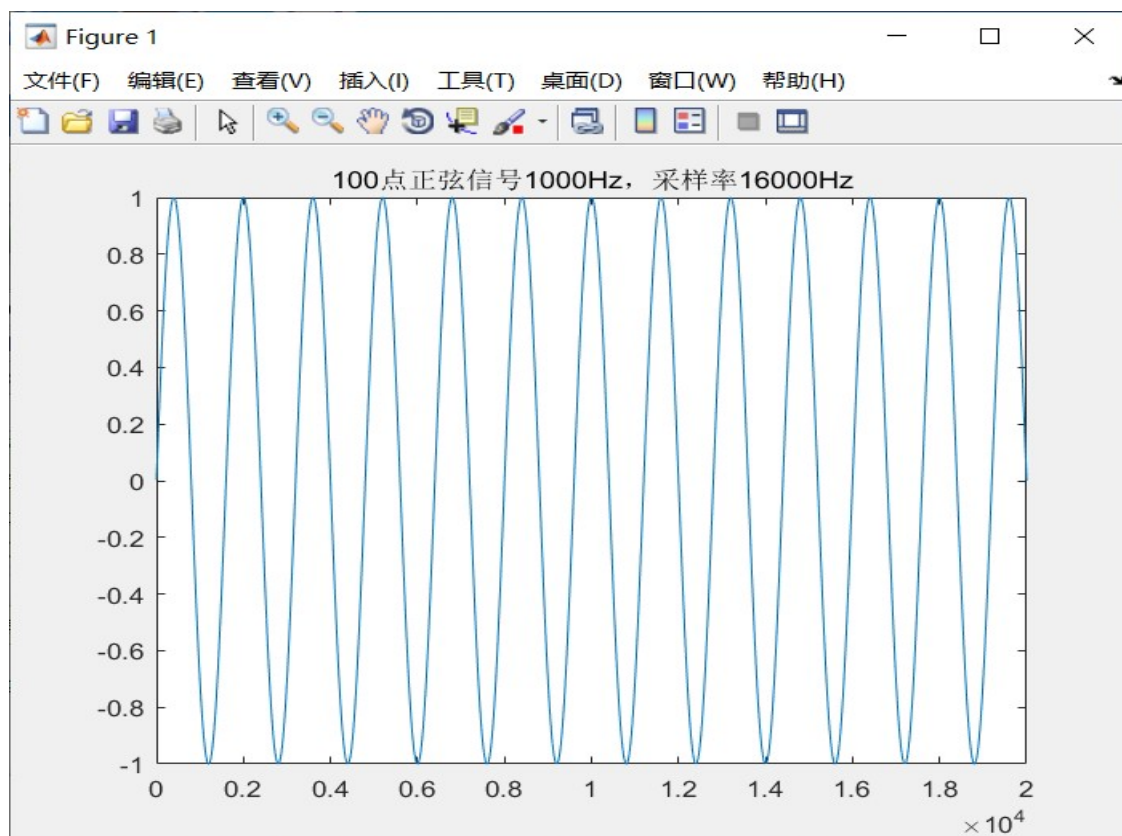
1600

相位差为：0.49875pi

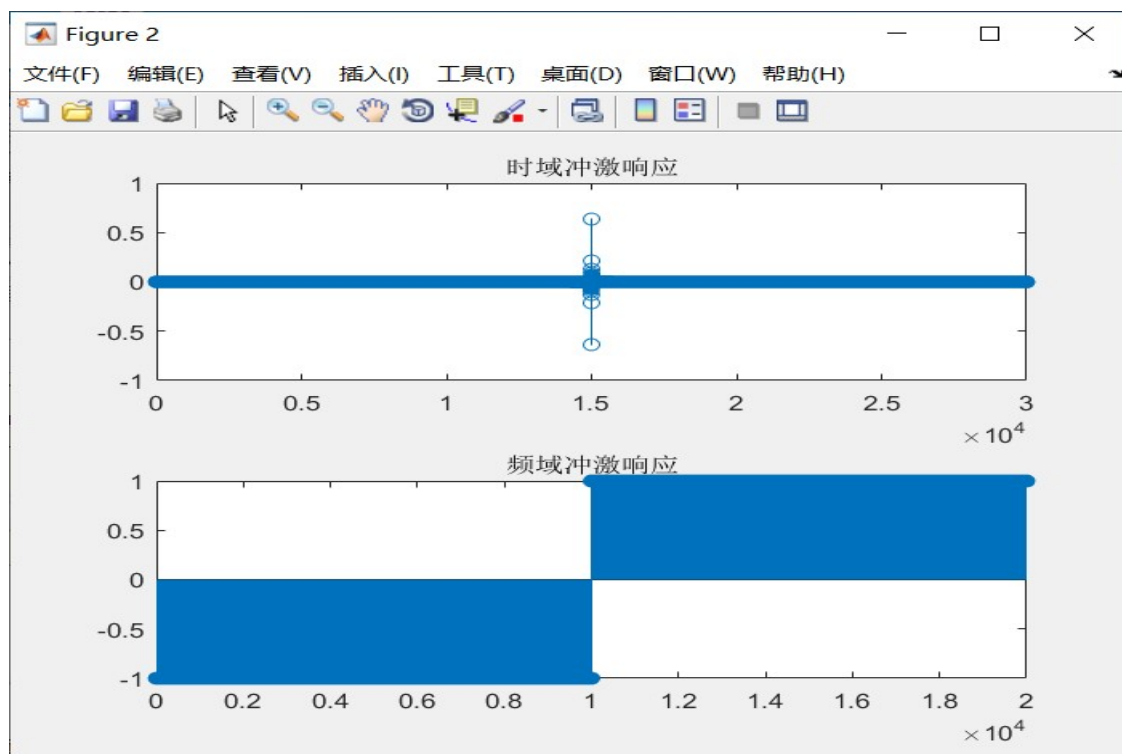
时域卷积变换	频域乘积变换	内置 Hilbert 函数变换
$\pi/2$	$\pi/2$	$\pi/2$

对于延时点 $D=399$ 的原因是，输入信号时域波形的周期 $T=1600$ ，即对于信号延时 $\pi/2$ 的相位对应延时点 $D=399$ 。

1. 正弦信号样本时域波形

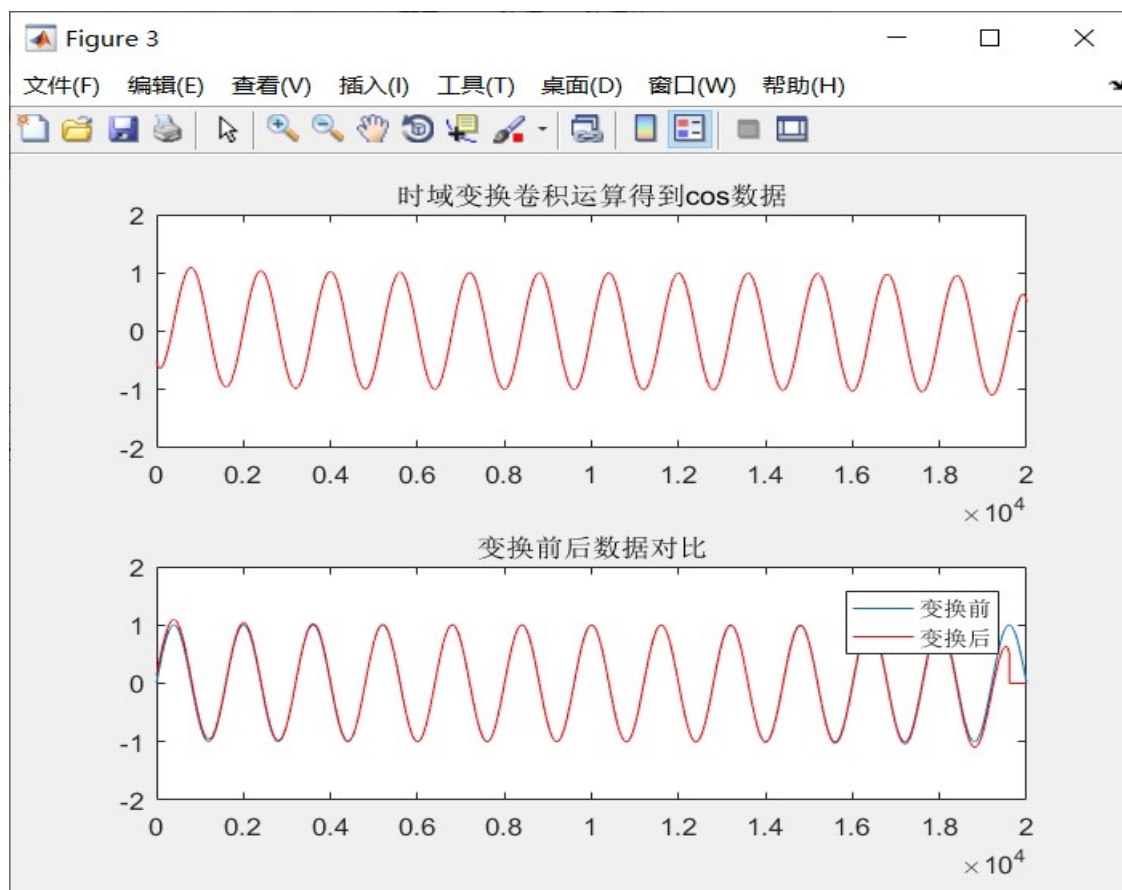


2. 设计的希尔伯特滤波器冲激响应波形和频域波形

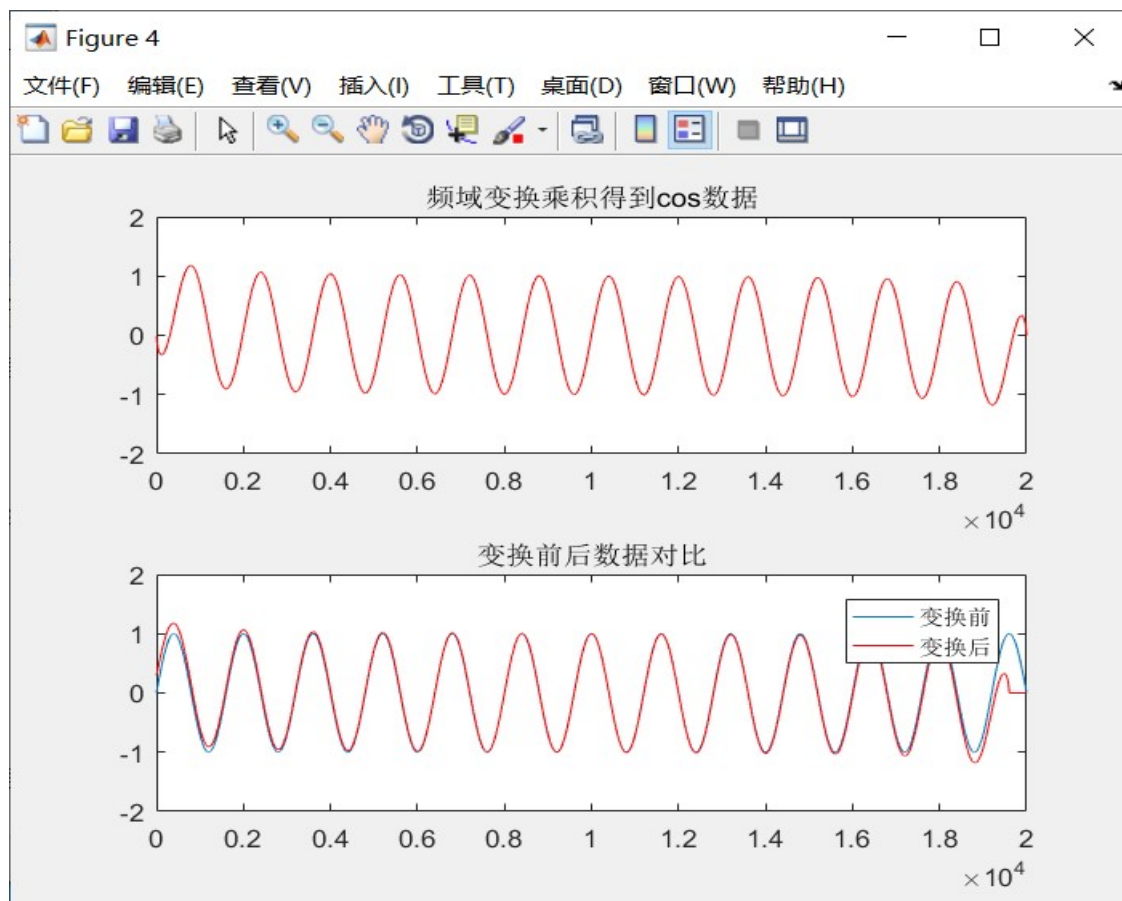


实验中要求产生 20000 个信号样本点，但此时图像变为一片蓝色，已无法清晰地呈现信号波形。

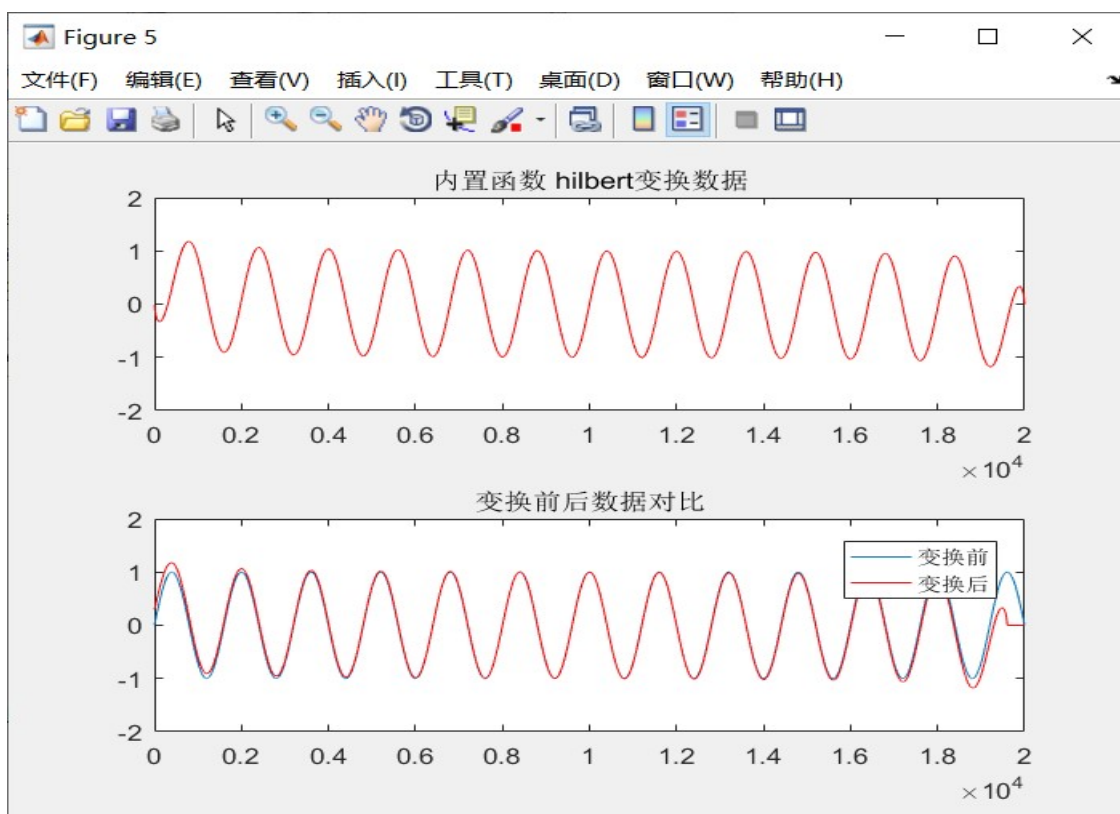
3. 时域卷积变换前后波形图对比



4. 频域变换所得 cos 随机信号波形



5. 对比内置函数 hilbert 所得 cos 随机信号

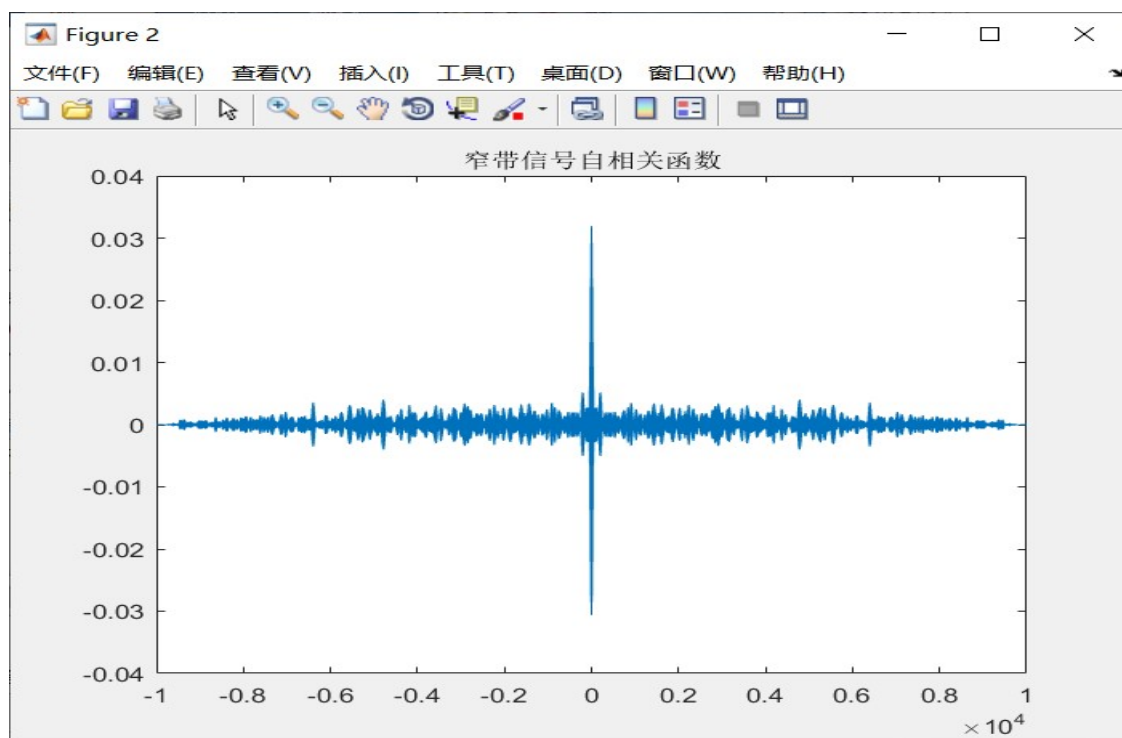
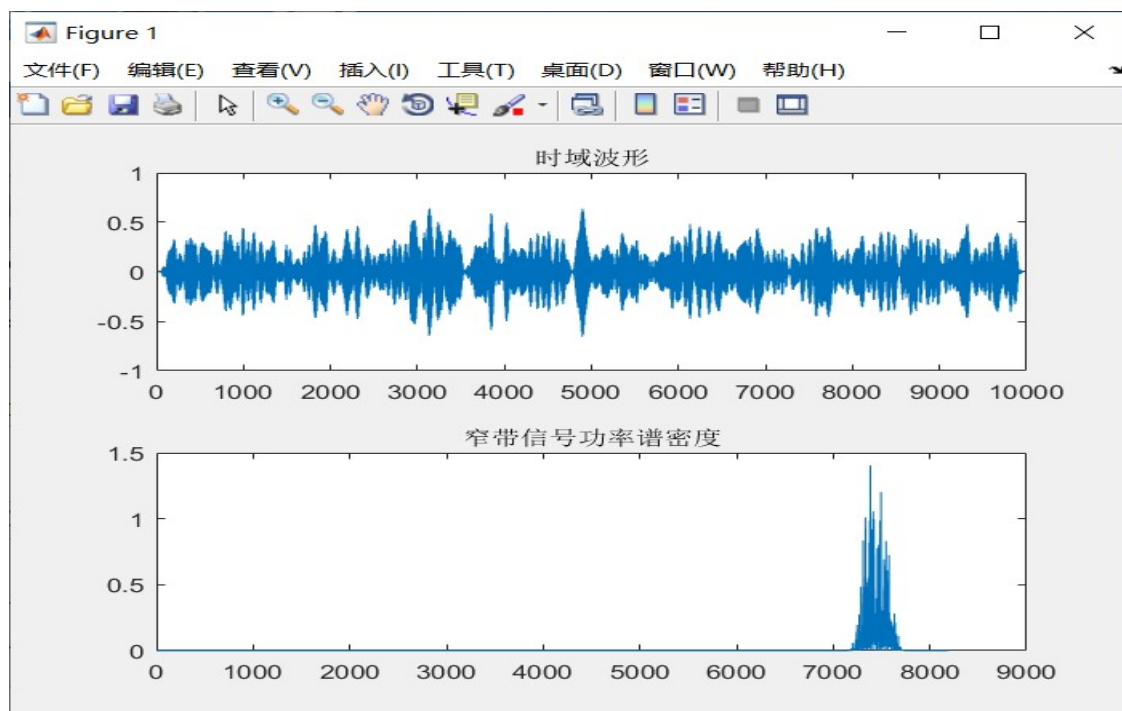


本实验通过时域和频域两种方法求解正弦信号的 Hilbert 变换，通过 MATLAB 计算结果可知，变换前后的信号相位差为 0.5π ，即变换后的信号为负的余弦信号，符合理论知识所学。

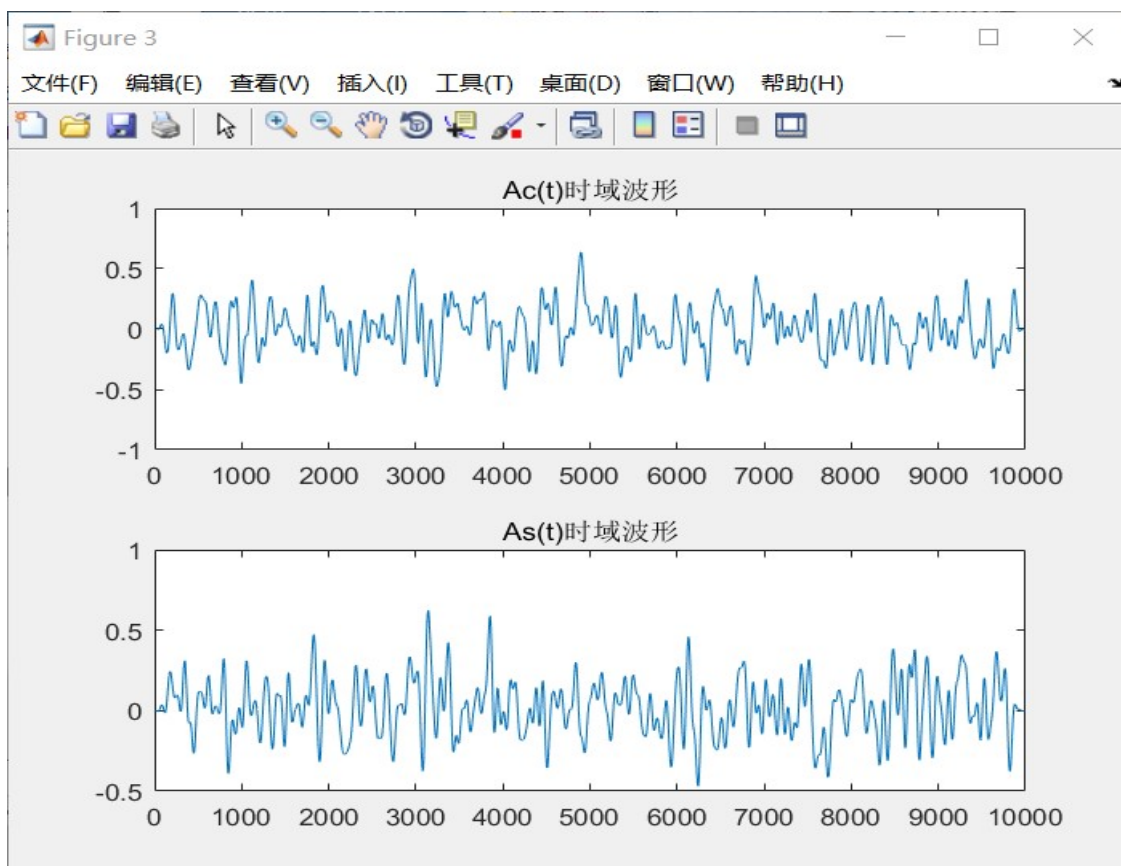
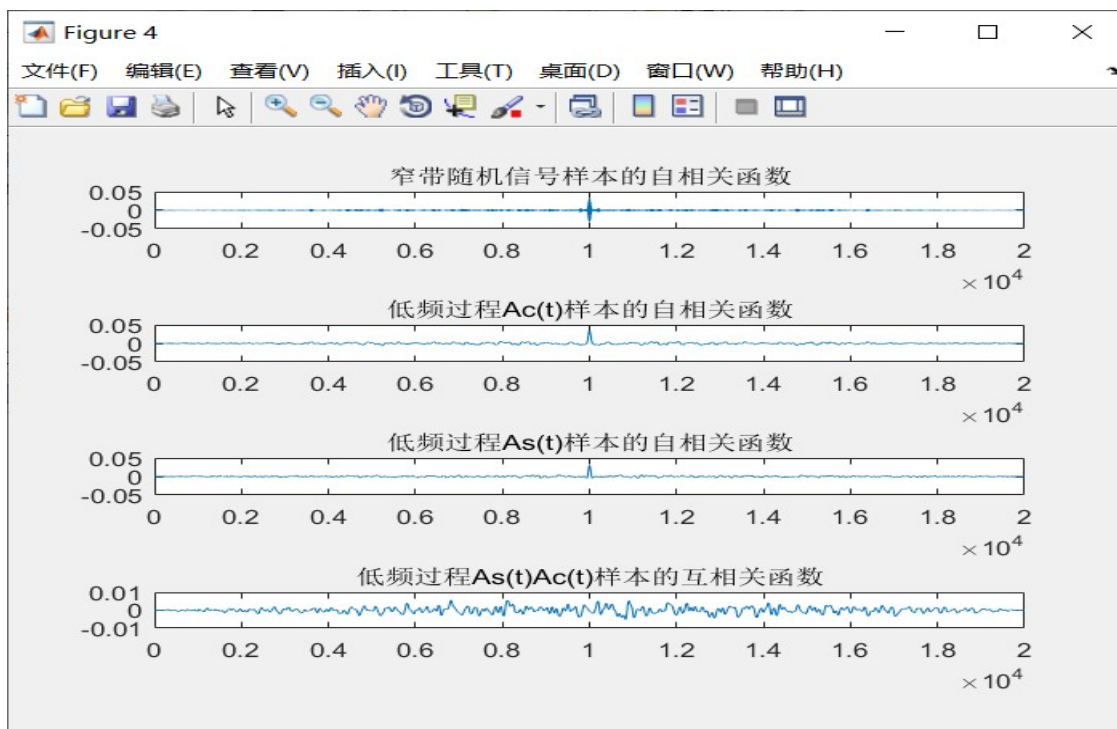
可以得出通过时域卷积和频域变换得到的希尔伯特输出同内置希尔伯特函数类似。

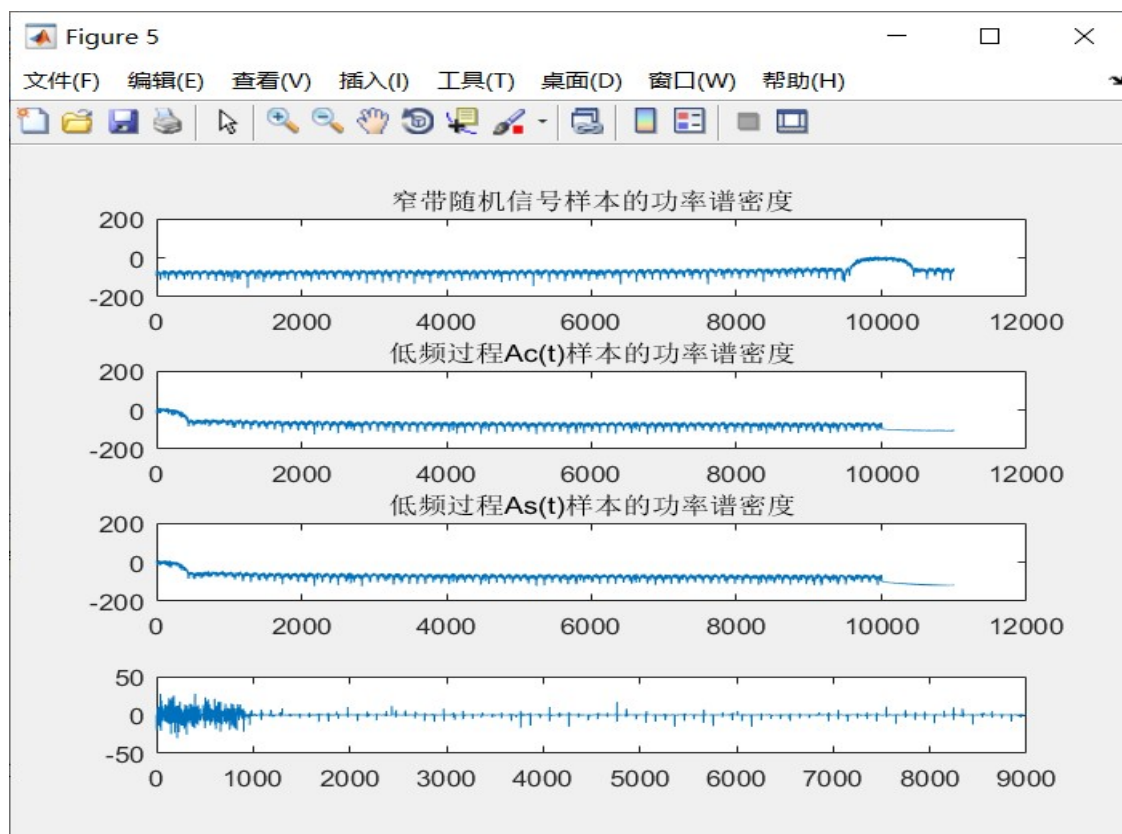
实验二 随机过程及其低频分量的功率估计方法

1. 生成窄带随机信号时域波形，功率谱波形，自相关函数波形



2. 取得的 $A_c(t)$ 、 $A_s(t)$ 分量时域波形

3. 取得的 $Ac(t)$ 、 $As(t)$ 自相关、互相关函数波形4. 取得的 $Ac(t)$ 、 $As(t)$ 分量功率谱波形



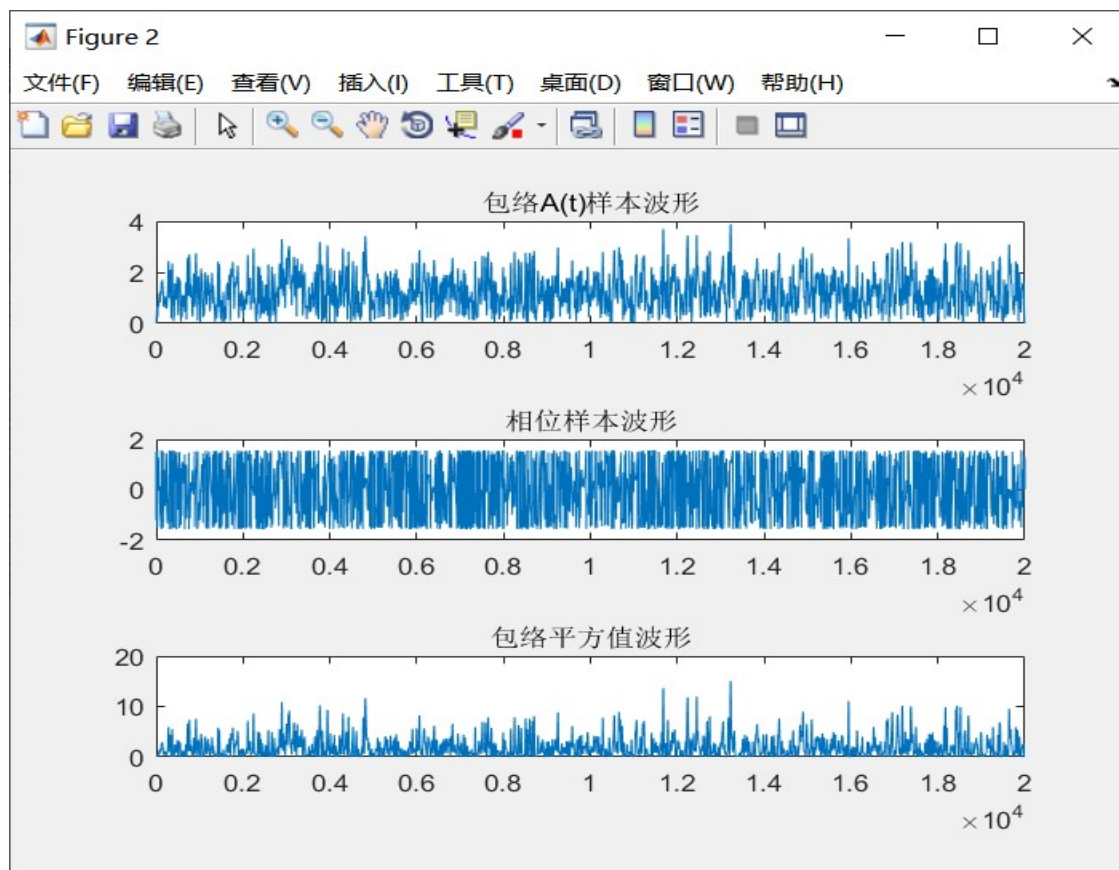
本实验通过 MATLAB 仿真探究了窄带随机信号及表达式中正余弦分量的特性，在图中，第一张图是中心频率为 10000Hz，带宽为 500Hz 的窄带过程的功率谱密度，第二张和第三张图分别是 $A_c(t)$ 和 $A_s(t)$ 的功率谱密度，通过三者功率谱密度的对比，验证了式 3.5。

$$G_{A_c}(\omega) = G_{A_s}(\omega) = \begin{cases} G_x(\omega - \omega_0) + G_x(\omega + \omega_0), & |\omega| < \Delta\omega/2 \\ 0, & \text{其他} \end{cases} \quad (3.5)$$

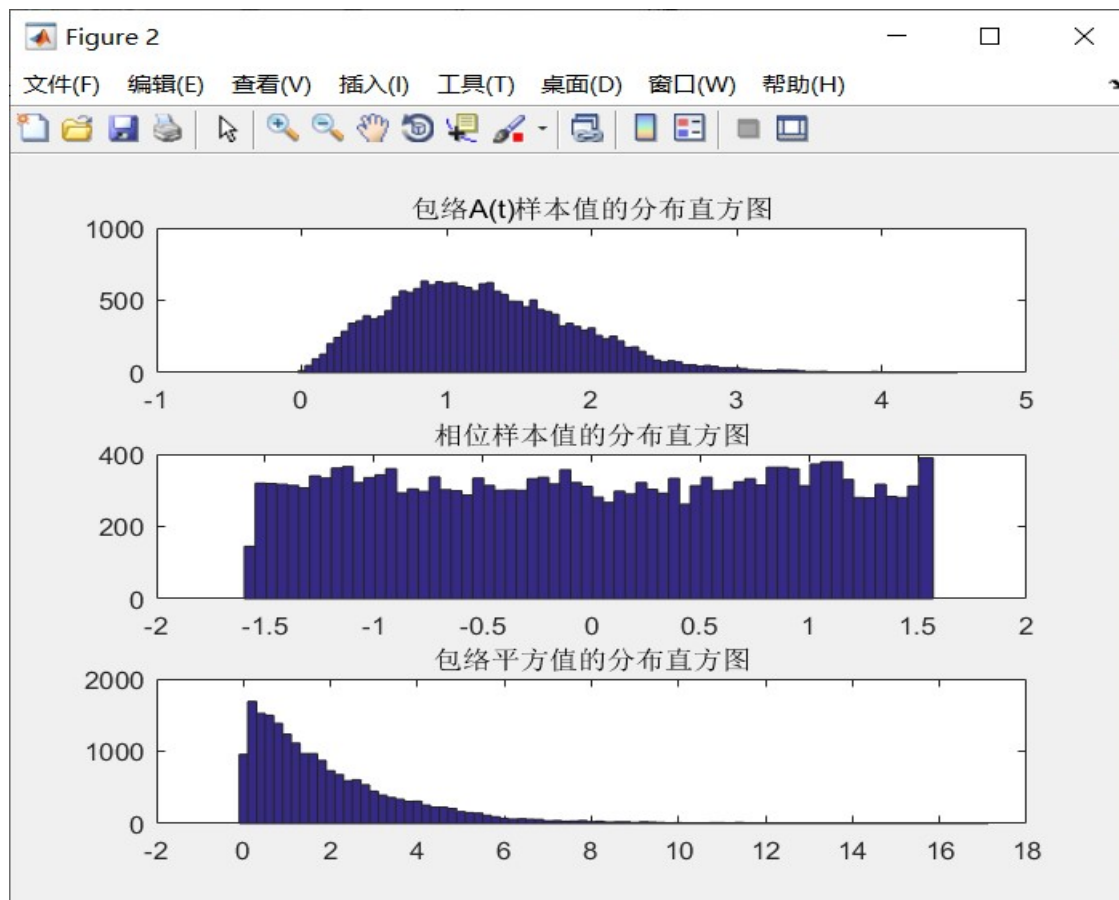
使用 ks 检验 A_s 和 A_c 的相似程度，概率为 1，可以认为二者相等。

实验三 高斯窄带随机过程包络和相位的分布估计方法

1. 高斯窄带随机信号包络，相位，包络平方波形图



2. 窄带随机信号包络，相位，包络平方的分布直方图



仿真结果可以看出，高斯窄带随机信号包络的分布为瑞利分布，相位的分布为均匀分布，包络平方的分布为指数分布，从而验证了理论知识。

用 KS 检验：

包络检验：

该数据源不服从正态分布。

该数据源不服从均匀分布。

该数据源不服从指数分布。

该数据源不服从泊松分布。

该数据源不服从 γ 分布。

*该数据源服从 rayleigh 分布。

相位检验：

该数据源不服从正态分布。

*该数据源服从均匀分布。

包络平方 KS 检验：

该数据源不服从正态分布。

该数据源不服从均匀分布。

*该数据源服从指数分布。

仿真结果可以看出，高斯窄带随机信号包络的分布为瑞利分布，相位的分布为均匀分布，包络平方的分布为指数分布，从而验证了理论知识。

六、实验结果与分析

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，f0 表示窄带随机过程 X(t)单边功率谱的中心频率 f_0 ；fs 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 At, Ph 和 A2 样本 EnvelopPhase

函数 EnvelopPhase(X,f0,fs)，用来产生 At, Ph 和 A2。形参 X 为要提取的随机过程 X(t)的样本，f0 表示窄带随机过程 X(t)单边功率谱的中心频率 f_0 ，fs 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);

```

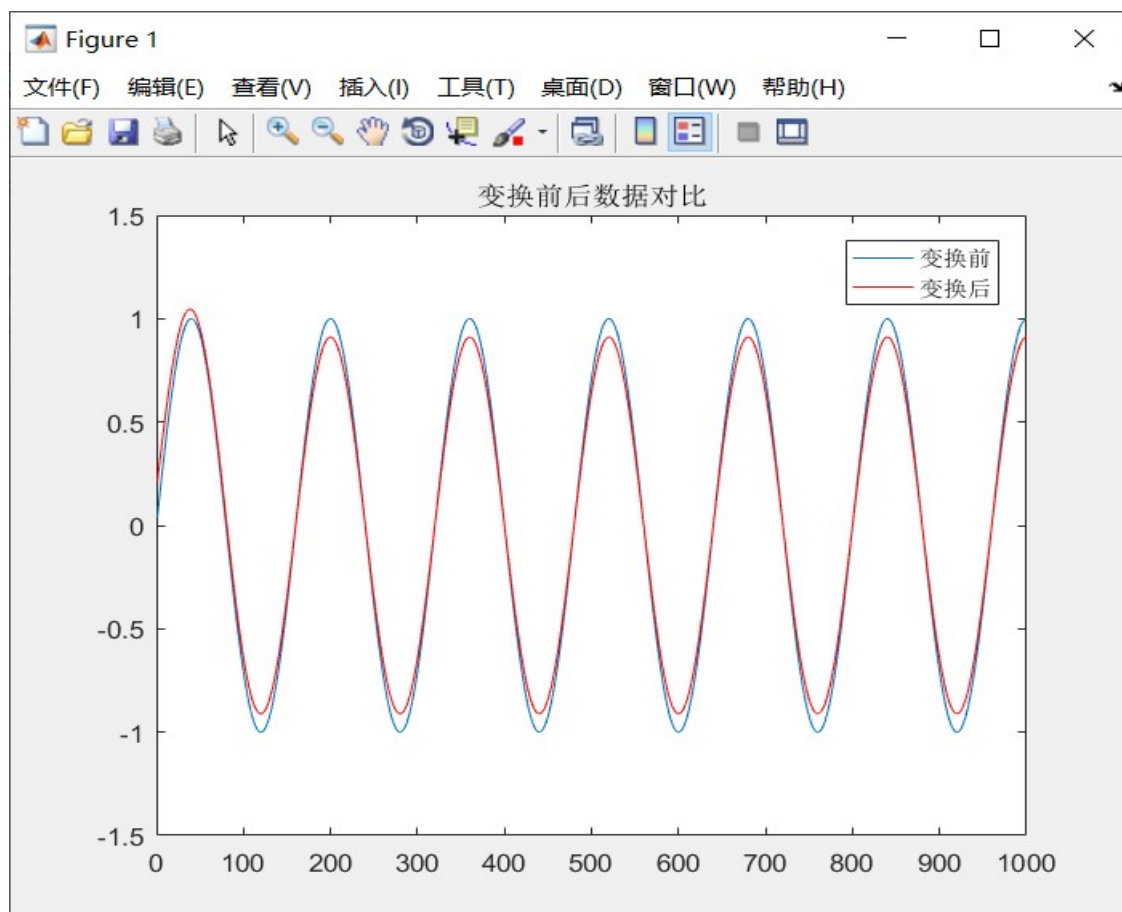
分析：

1. 对于实验一为何时域的希尔伯特变换器会出现只有奇数点有高度并且在奇数时是 2 倍的 $1/\pi t$ ，与连续的 $1/\pi t$ 并不一致。这个区别可以采用数字信号处理的 IDTFT 进行解释，对时域进行离散化，即对时域进行隔点采样，从频域上看，会造成频域周期化，并且周期为 2π 。在 $-\pi \sim 0$ 这个频域上，频域响应是 j ，在 $0 \sim \pi$ 上是 $-j$ ，对它进行离散时域傅里叶反变换 IDTFT，我们可以非常容易得推出我们

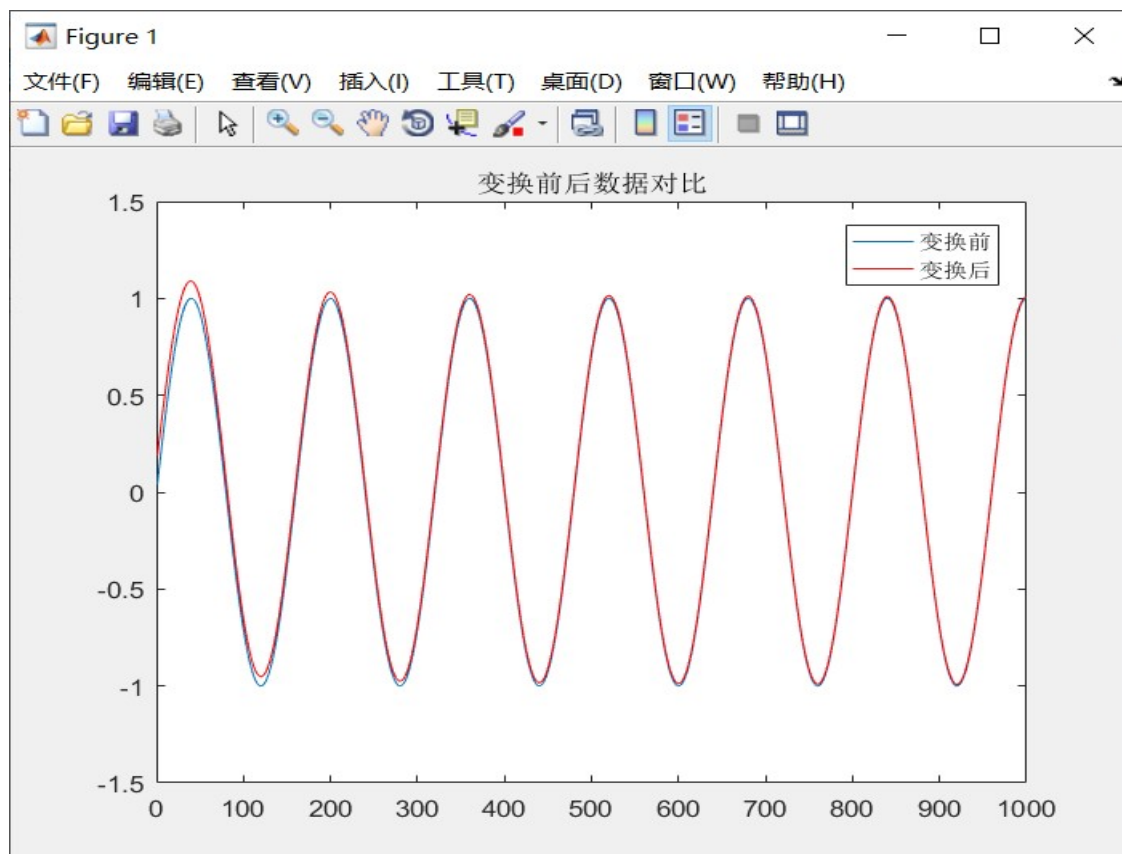
的时域响应，

$$h(n) = \begin{cases} 0 & n = 2k \\ \frac{2}{n\pi} & n = 2k+1, n = -\infty, \dots, \infty \end{cases}。$$

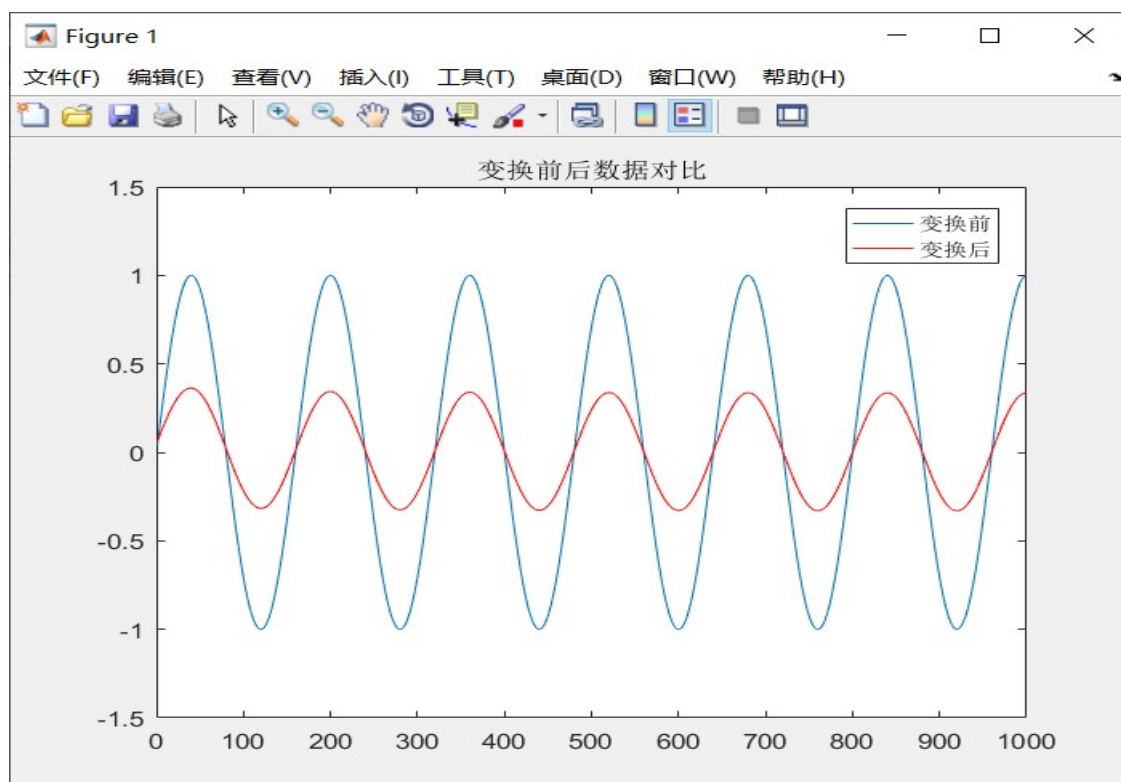
2. 对于希尔伯特的时域变换的步长 step ，它反应的是希尔伯特滤波器的采样点数，在相同的采样间隔之下，希尔伯特滤波器采样点数越多，精度越高。如果我们只增加 step 但是并不增加对信号时域波形的采样频率，我们实际上在信号时域上对应的时域长度会变短，即对应点数会减少，这回导致我们变换后的能量变小，我们可以非常清晰得看到单纯增加 step 采样点数后输出的波形它的幅值是要比原有波形要低很多的。增加 10 倍的采样频率，确实在幅值上变为了 $\sqrt{3}$ ，这是符合预期的。而我们如果想通过增加希尔伯特滤波器的采样点数增加对信号变换的精度，而其它不想被改变，我们只需要等比例对采样频率 f_s 和 step 进行等比例变化就好了。



小 step, 二者相近, 但拟合的不是很好。



等比例增加 step 和 f_s ，采样精度有了提高。二者的拟合度有了极大增加



单增加 step ，变换后的幅度会减小

3. 我们可以看到在变换信号的两端并不理想，左侧是吉伯斯现象，是无法彻底消除的，也是时频域变换特有的现象，这本质是由于加了一个带权值的特殊矩形窗，由于其切断效应，必然会导致这种现象，由理论和实际测量可知，增加采样频率和滤波器采样长度和精度可以改善这种现象，但是不能彻底消除。而增加采样频率会增加对算力的要求，也会对延迟有比较大的影响，这是我们必须考虑的。

4. 从相位角度说，滤波器和信号都是从 0 开始的，无论是时域还是频域滤波器都做了 $-j$ 的延迟，所以卷积出来的是希尔伯特逆变换，只有总延迟的必须是等于 $k \cdot \text{step}$ ($k=0, +1, +2, \dots$) 的，造成的相位差才能够保证输出波形不会出现有延迟的问题。而如果是 $K \cdot \text{step}/2$ 的话，则必然导致题中的反相的情况的发生。同时，之所以比较时没有显现出来滤波器的延迟，是对信号进行延迟后采用了舍弃的补偿手段认为把二者相位调节至一致。滤波器和信号的相对位置的改变是会对输出造成延迟的，本次就造成了有 π 的延迟，把正变换延迟成了反变换，这是我们必须注意的。

5. 生成窄带信号的方法：

- (1) .白噪声通过窄带系统
- (2) .低频信号与载波信号混频后滤波

(3) 相干信号的调制与解调, $A(n) = \sqrt{A_c^2(n) + A_s^2(n)}$, $\Phi(n) = \arctan \frac{A_s(n)}{A_c(n)}$, 这里要求 A_s

和 A_c 必须是相干的

(4) 使用多级混频和滤波方式, 级联把信号频率提高

(5) 使用 DDS 方式生成

(6) 使用多级插值的方法

6. 通过求自相关函数求对应的功率谱密度, 可看出 $A(t)$ 样本值频率远远高于其包络频率。并且验证了低频分量 $A_c(t)$ 和 $A_s(t)$ 具有相同的自相关函数、相同的功率谱密度, 则有相同的方差与平均功率。

同时也符合数学上的公式: $R_{A_c}(\tau) = R_{A_s}(\tau) = R_X(\tau) \cos \omega_0 \tau + \hat{R}_X(\tau) \sin \omega_0 \tau$ 其中 ω_0 为高频频率。

$A_c(t)$ 和 $A_s(t)$ 具有相同的自相关函数、相同的功率谱密度, 这个可以使用 ks 检验的方式进行验证, 概率为 1, 实验结果有力支撑了我们的理论。这个检验不可以采用相减的方式, 这是因为窄带信号不够理想, 在变换时也会有失真, 试图相减, 但是出现了零散的冲击, 但这些冲击的能量趋近于 0, 而且十分零散, 完全可以忽略掉, 所以不可以否认二者的相等, 尤其是考虑到变换的误差, 是可以接受的。

检测两者相等的假设检验函数:

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

```
% 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
```

7. 包络的一维分布符合瑞利分布 (或称广义莱斯分布), 相位的一维分布符合均匀分布, 包络的平方的分布符合指数分布这个规律。

用 ks 进行检验可以看出, 结果有力支撑了我们的理论。具体 ks 原理再第一次实验已经完成, 不再赘述。

Ks 检验函数：

```

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
end

```

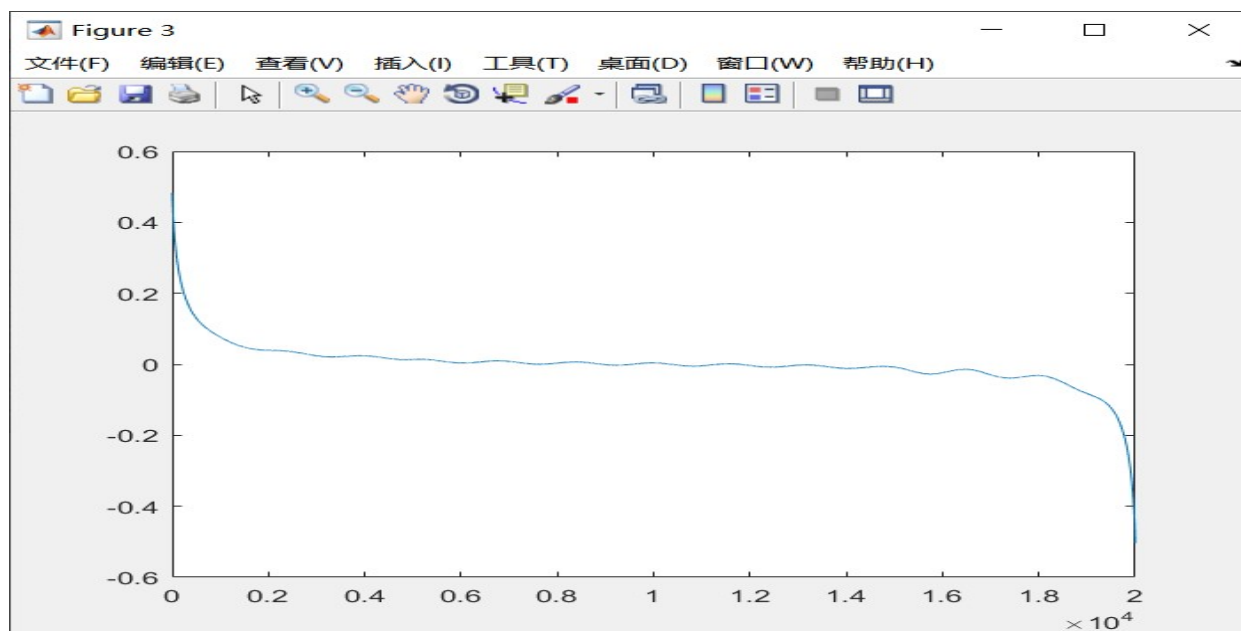
8. 实验二和实验三的不同之处在于实验二是窄带信号就可以，而实验三必须是窄带高斯信号，只有信噪比不是特别大的时候，我们才能认为包络服从瑞利分布，相位的一维分布符合均匀分布，包络的平方的分布符合指数分布这个规律，只有在高斯时，我们才能把不相关转换为独立，就能从 A_s 和 A_c 的概率密度推出 A 和相位的概率密度。而实验二中的性质是任何的窄带信号都具有的。

七、讨论、建议、质疑

本次实验既进行了理论课上着重强调的内容——希尔伯特变换，又进行了窄带高斯过程包络和相位的分析，可以看出实验课可以和理论课互补，更可以互相加强，使我对理论和实验的认识都更加充分。另外最后老师的提问也让我明白要摆脱依赖调用函数去做检验的固定思维，要看到最本质的东西，一个简单的转换就可以验证自编函数的结果而不是必须调用所谓的系统函数（如检验 \cos

希尔伯特变换正确性的方法不必用一个更复杂的函数去做希尔伯特变换，直接编程 $-\cos$ 即可)。

理想 \cos 和生成的 \cos 函数相减的出的重要波形，可以看出除了两端的由切断和泄漏效应引发的失真，拟合非常好。



另外，我也使用了 ks 检验的方法对 \cos 函数进行了检验，结果符合预期。

之后，我还使用 ks 的方法对 A_s 和 A_c 的功率谱进行了检测，并且用 ks 方法对任务 3 的分布进行了检测，这次实验，使我对 ks 检验的运用和理解有了更加深入的认识。

同时本实验再次体现了 MATLAB 编程的强大之处，在课堂上数学理论推导十分复杂的结论却能在 MATLAB 中十分形象的显现出来。在日后的学习中，可以首先通过 MATLAB 观察现象，再有目的的进行理论证明，从而提高效率。

最后，随机信号分析实验就告一段落了，通过本课程，我又一次提高了 MATLAB 编程的能力，相信这一强大的工具会伴随我继续走过学习和工作的道路。非常感谢李老师和陈老师在实验中对我的帮助和指导！

八、matlab 代码

任务 1

```
%% 准备环境
clear all
close all
clc
%% 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));
```

```
cosdata=cos(2*pi*f/fs*(1:N)-pi);
```

```
figure,plot(sinData)
```

```
title('100 点正弦信号 1000Hz，采样率 16000Hz')
```

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

```
%% 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')
```

```
%% 1: 时域变换卷积运算得到 cos 数据
```

```
%时域卷积
```

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

```
k=cosData-cosdata;
```

```
figure,plot(k)
```

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

```

plot(cosdata,'b');
xxx=[cosData',cosdata'];
[p,q]=chi2test(xxx);
%axis([0 1000 -1 1]);
title('时域变换卷积运算得到 cos 数据')

%% 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

%% 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')

%% D 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'])

%% 2: 频域变换乘积得到 cos 数据
%域乘积后反变换
cosData = real(iff(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')

%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) ;

%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')

%D 转换成弧度 找到周期
[C,I]=find(sinData==1);
T = I(4)-I(3)

%相位差
delta = D/T;
disp(['相位差为: ',num2str(delta*2),'pi'])

%% 3.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')

%%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)

%%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')

%%D 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'])
```

任务 2

```
%% 准备环境
clear all
close all
clc

%% 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
%% 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')

%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')

%% 取得 Ac(t)、As(t)分量
[Ac, As]=Lowfsignal(X,f0,fs); %调用产生 Ac(t)、As(t)的函数

%% 展示 Ac(t)、As(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')

%% 展示窄带、Ac(t)、As(t)分量自相关函数
Rac=xcorr(Ac,'biased'); %低频过程 Ac(t)样本的自相关函数
Ras=xcorr(As,'biased'); %低频过程 As(t)样本的自相关函数
Racs=xcorr(As,Ac,'biased'); %低频过程 As(t)样本的自相关函数
Racw=abs(fft(Rac)); %低频过程 Ac(t)样本的功率谱密度
Rasw=abs(fft(Ras)); %低频过程 As(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
```

```

% 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 Ac(t)样本的自相关函数');
subplot(4,1,3);plot(Ras);
title('低频过程 As(t)样本的自相关函数');
subplot(4,1,4);plot(Racs);
title('低频过程 As(t)Ac(t)样本的互相关函数');

%% 展示窄带、Ac(t)、As(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 Ac(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 As(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

```

任务 3

```

%% 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
%% 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)
```

```
%% 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
```

```
%% 展示窄带随机信号的包络、相位、包络平方的时域波形
```

```
%包络 A(t)样本值的分布直方图
```

```
figure;subplot(3,1,1);plot(At); %包络 A(t)样本值的分布直方图
```

```
title('包络 A(t)样本波形');
```

```
subplot(3,1,2);plot(Ph); %相位样本值的分布直方图
```

```
title('相位样本波形');
```

```
subplot(3,1,3);plot(A2); %包络平方值的分布直方图
```

```
title('包络平方值波形');
```

```
%% 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');
```

```
ks 分布检验
```

```
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
end

```

ks 相似性检验：

```

function [p, Q]= chi2test(x)
    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

```

附加函数：

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, f_0 随机过程单边功率谱的中心频率为 f_0

$\text{xt}=\text{random}('norm',0,1,[1,N1]);$ % f_s 表示信号采样频率, M 为产生宽带信号滤波器阶数
 $\text{f1}=f_0*2/f_s;$ %输出 N 个再带随机信号样本

$\text{df1}=\text{deltf}/f_s;$ % $N-M$ 个高斯随机数 %归一化中心频率 %归一化带宽

$\text{ht}=\text{fir1}(M,[f_1-\text{df1} \ f_1+\text{df1}]);$ % ht 为带通滤波器的冲击响应, M 为阶数

$\text{X}=\text{conv}(\text{xt},\text{ht});$ %输出 N 个窄带随机信号样本采样

return

2. 产生 A_c, A_s 样本 Lowfsignal

函数 Lowfsignal(X, f_0, f_s) 用于产生 A_c 和 A_s 。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);
```