第3章 采样和量化

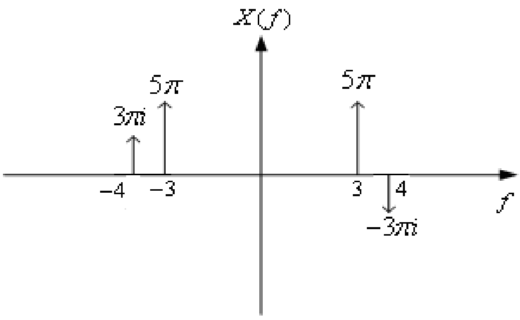
3-1

答：

输入：syms t w

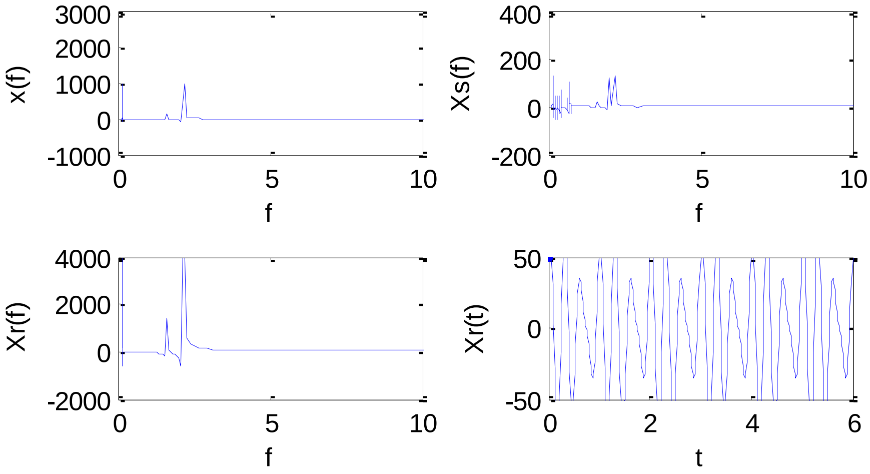
Xf=fourier(5*cos(6*pi*t)+3*sin(8*pi*t))

输出：Xf=pi*(5*dirac(w+6*pi)+3*i*dirac(w+8*pi)-3*i*dirac(w-8*pi)+5*dirac(w-6*pi))



matlab 程序：

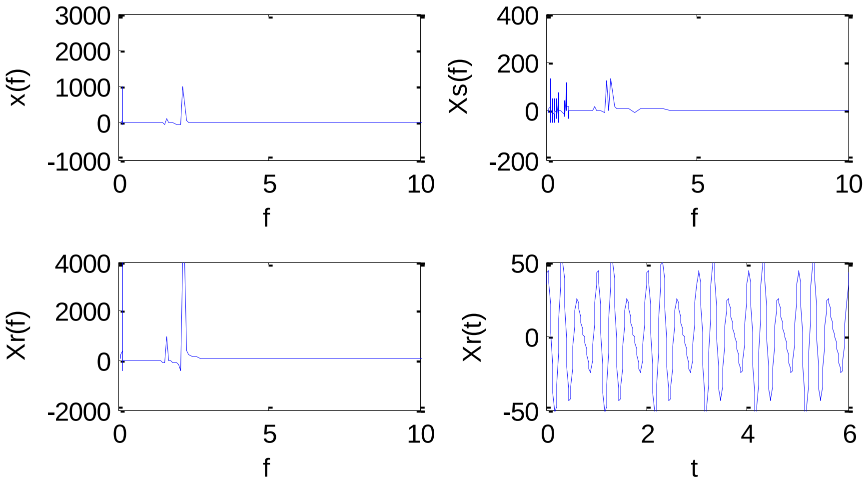
```
t=0:0.02:8;
L=length(t);
xt=5*cos(6*pi*t)+3*sin(8*pi*t);
f1=fft(xt);
fs=10;Ts=1/fs;
t1=8:-0.02:0;
f=1./t1;
Pt=zeros(1,L);
for i=1:8:L
        Pt(i)=1;
end
Xst=xt.*Pt;
f2=fft(Xst);
f3=fs*f1;
f4=ifft(f3);
subplot(2,2,1)
plot(f,f1)
axis([0 10 -1000 3000])
xlabel('f');ylabel('x(f)');
subplot(2,2,2)
plot(f,f2)
axis([0 10 -200 400])
xlabel('f');ylabel('Xs(f)');
subplot(2,2,3)
plot(f,f3)
axis([0 10 -2000 4000])
xlabel('f');ylabel('Xr(f)');
subplot(2,2,4)
plot(t,f4)
xlabel('t');ylabel('Xr(t)');
axis([0 6 -50 50])
```



3-2

答：matlab 程序：

```
t=0:0.02:8;
L=length(t);
xt=5*cos(6*pi*t)+3*sin(8*pi*t);
f1=fft(xt);
fs=7;Ts=1/fs;
t1=8:-0.02:0;
f=1./t1;
Pt=zeros(1,L);
for i=1:8:L
    Pt(i)=1;
end
Xst=xt.*Pt;
f2=fft(Xst);
f3=fs*f1;
f4=ifft(f3);
subplot(2,2,1)
plot(f,f1)
axis([0 10 -1000 3000])
xlabel('f');ylabel('x(f)');
subplot(2,2,2)
plot(f,f2)
axis([0 10 -200 400])
xlabel('f');ylabel('Xs(f)');
subplot(2,2,3)
plot(f,f3)
axis([0 10 -2000 4000])
xlabel('f');ylabel('Xr(f)');
subplot(2,2,4)
plot(t,f4)
xlabel('t');ylabel('Xr(t)');
axis([0 6 -50 50])
```



3.5

信号 $x(t)=5\sin(10\pi t)$，

（a）信号的动态范围为 $SNR=25.84,49.93,98.09,194.42$ dB。

（b）信号的波峰因素为 $F_c=\frac{2\sqrt{S}}{D}$。其中，S 为信号的功率 $S=\frac{A_m^2}{2}=\frac{5^2}{2}=12.5$，所以 $F_c=\frac{2*\sqrt{12.5}}{10}=0.7071$。

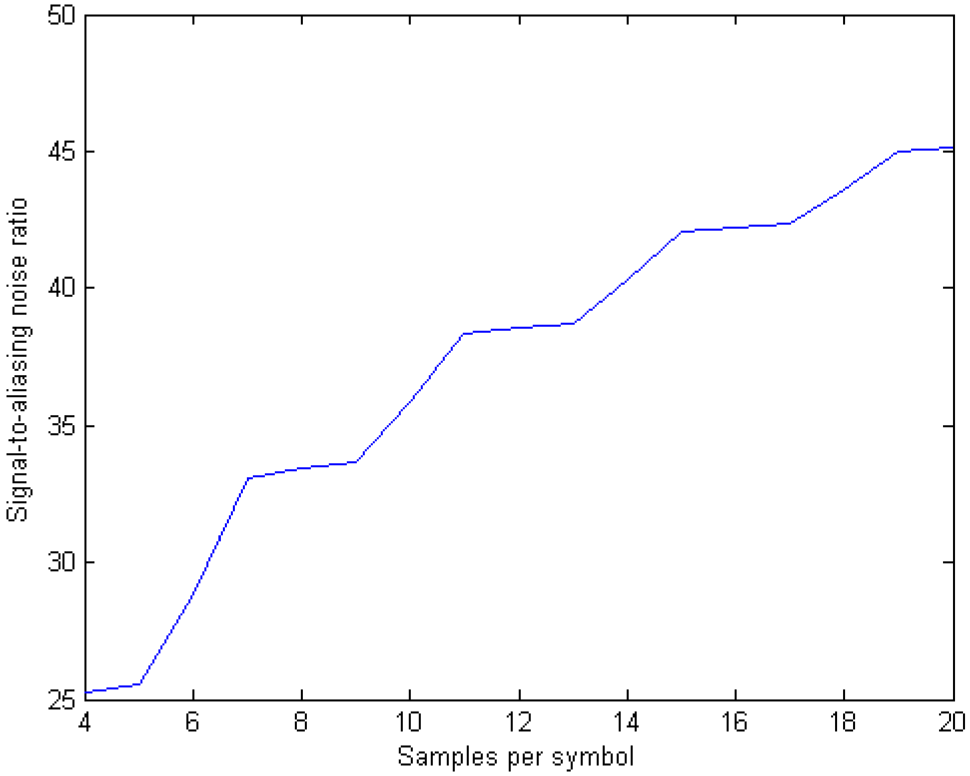
（c）信噪比 $(SNR)_q=3F_c^2 2^{2b}$，以 dB 为单位，$10\lg(SNR)_q=10\lg(3)+20\lg(F_c)+20b\lg 2$

当 $b=4,8,16,32$ 比特时，计算得 $SNR=25.84,49.93,98.09,194.42$ dB

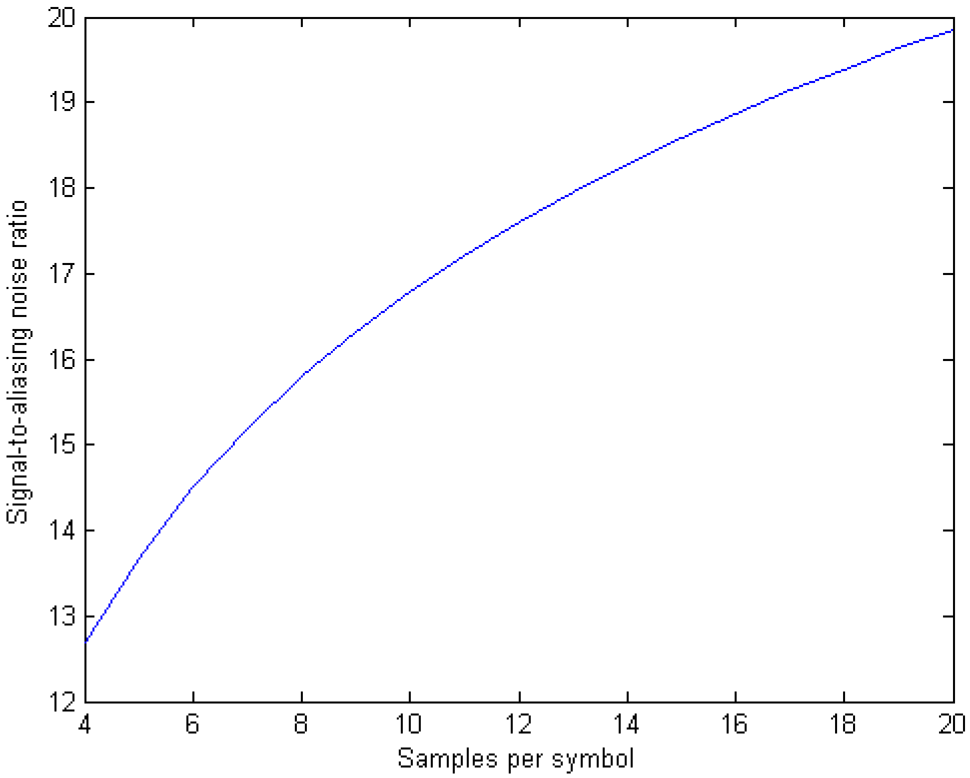
3.11

```
k = 50;                         % samples per lobe
nsamp = 50000;                  % total frequency samples
snrdb = zeros(1,17);            % initialize memory
x = 4:20;                       % vector for plotting
for m = 4:20
   signal = 0; noise = 0;       % initialize sum values
   f_fold = k*m/2;              % folding frequency
   for j = 1:f_fold
      term = (sin(pi*j/k/2)/(pi*j/k/2))^4;
      signal = signal+term;
   end
   for j = (f_fold+1):nsamp
```

```
        term = (sin(pi*j/k/2)/(pi*j/k/2))^4;
        noise = noise+term;
    end
    snrdb(m-3) = 10*log10(signal/noise);
end
plot(x,snrdb)                        % plot results
xlabel('Samples per symbol')
ylabel('Signal-to-aliasing noise ratio')
```



三角脉冲波形的信号混叠信噪比



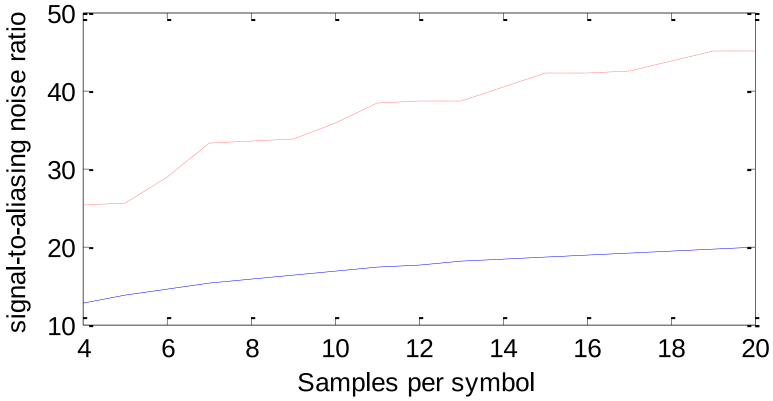
矩形脉冲波形的信号混叠信噪比

3-11

答：matlab程序：

```
k=50;
nsamp=50000;
snrdb=zeros(1,17);
snrdb_triangle=zeros(1,17);
x=4:20;
for m=4:20
    signal=0;noise=0;signal_triangle=0;noise_triangle=0;
    f_fold=k*m/2;
    for j=1:f_fold
        term=(sin(pi*j/k)/(pi*j/k))^2;
        term_triangle=(sin(pi*j/(2*k))/(pi*j/(2*k)))^4;
        signal=signal+term;
        signal_triangle=signal_triangle+term_triangle;
    end
    for j=(f_fold+1):nsamp
        term=(sin(pi*j/k)/(pi*j/k))^2;
        term_triangle=(sin(pi*j/(2*k))/(pi*j/(2*k)))^4;
        noise=noise+term;
        noise_triangle= noise_triangle+term_triangle;
    end
    snrdb(m-3)=10*log10(signal/noise);
    snrdb_triangle(m-3)=10*log10(signal_triangle/noise_triangle);
end
plot(x,snrdb,x,snrdb_triangle,'r:');
```

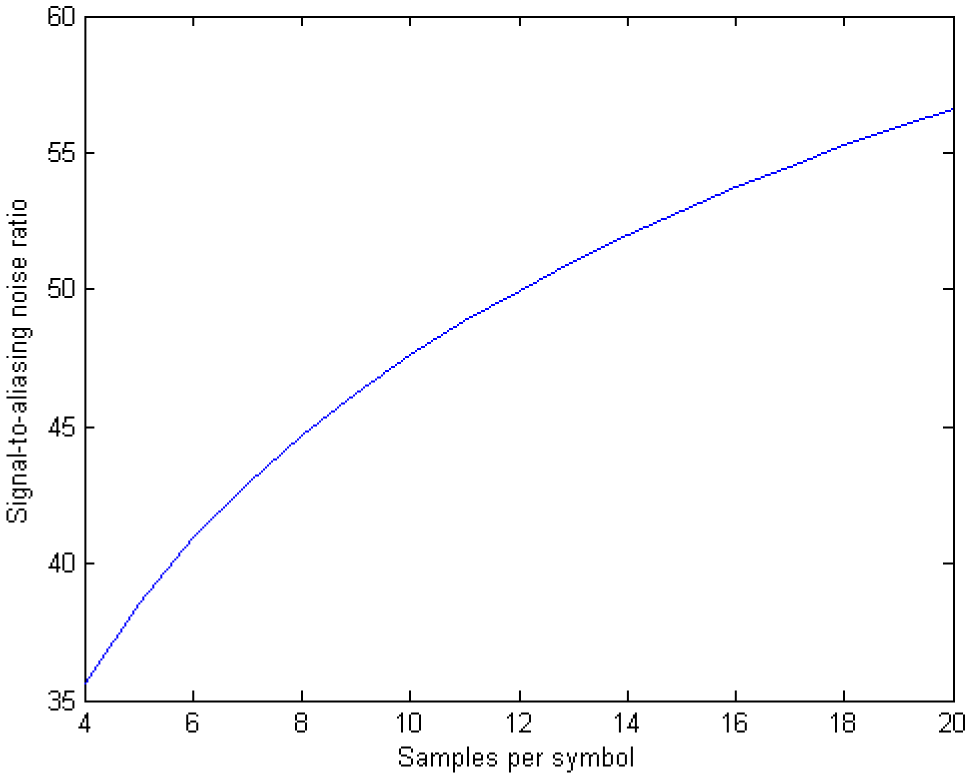
```
xlabel('Samples per symbol');
ylabel('signal-to-aliasing noise ratio');
```



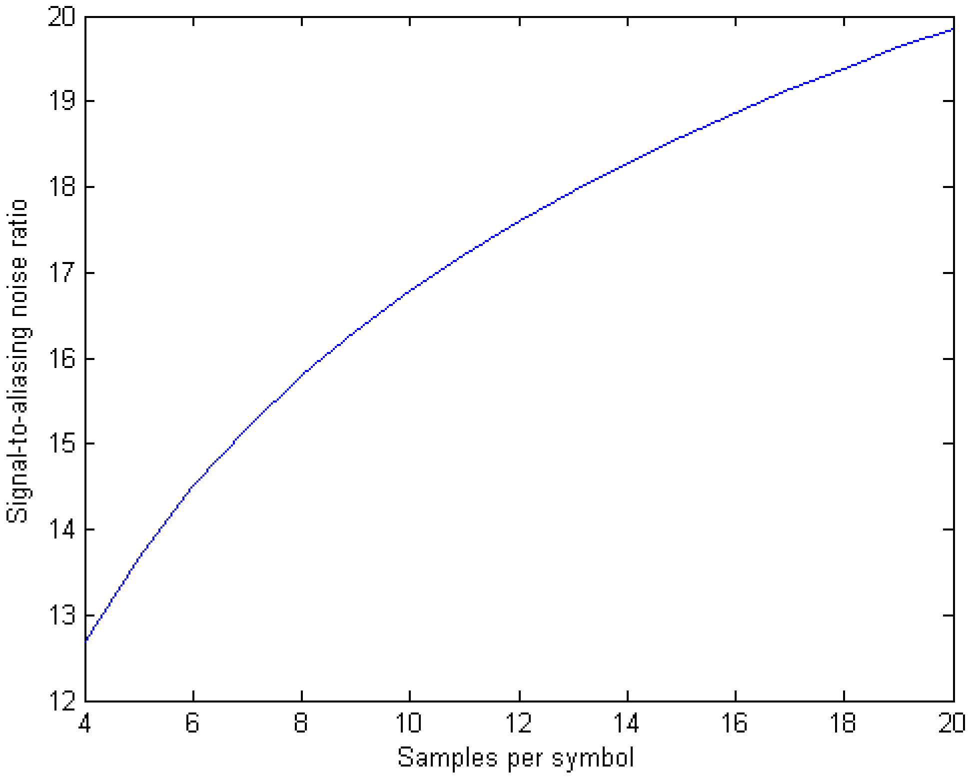
可见成形脉冲为三角脉冲时，效果更好一些。

3.12

```
k = 50;                        % samples per lobe
nsamp = 50000;                 % total frequency samples
snrdb = zeros(1,17);           % initialize memory
x = 4:20;                      % vector for plotting
for m = 4:20
   signal = 0; noise = 0;      % initialize sum values
   f_fold = k*m/2;             % folding frequency
   for j = 1:f_fold
      term = (cos(2*pi*j/k))^2/(pi^2)/((1-(4*j/k)^2))^2;
      signal = signal+term;
   end
   for j = (f_fold+1):nsamp
      term = (cos(2*pi*j/k))^2/(pi^2)/((1-(4*j/k)^2))^2;
      noise = noise+term;
   end
   snrdb(m-3) = 10*log10(signal/noise);
end
plot(x,snrdb)                  % plot results
xlabel('Samples per symbol')
ylabel('Signal-to-aliasing noise ratio')
% End of script file.
```



MSK 混叠信噪比

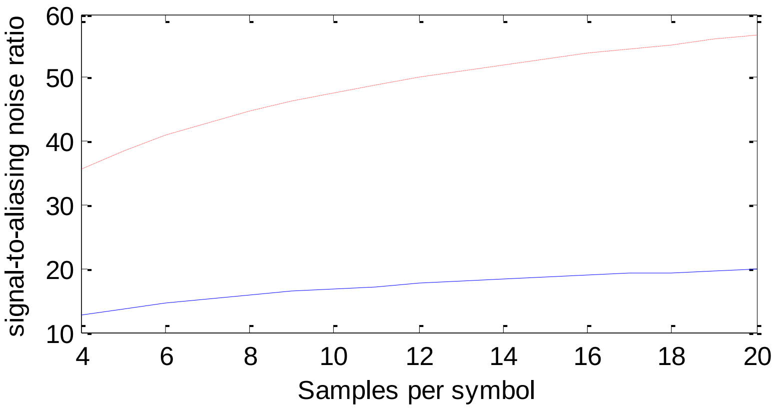


矩形脉冲波形的信号混叠信噪比

3-12

答：matlab程序： `k=50;`

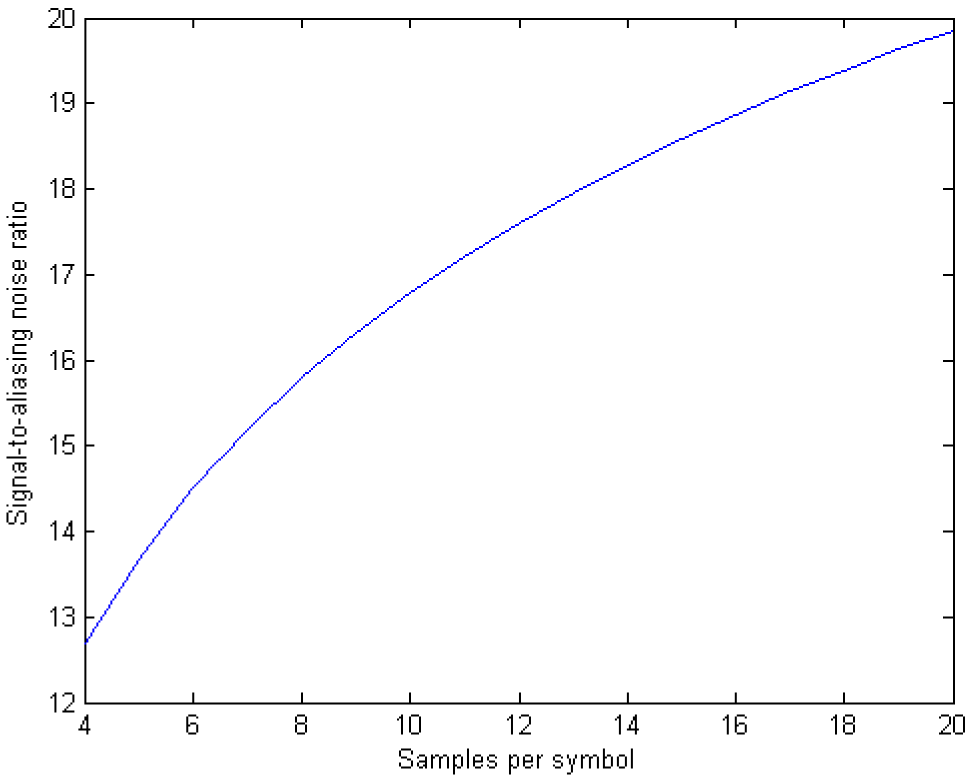
```
nsamp=50000;
snrdb=zeros(1,17);
snrdb_triangle=zeros(1,17);
x=4:20;
for m=4:20
    signal=0;noise=0;signal_triangle=0;noise_triangle=0;
  f_fold=k*m/2;
    for j=1:f_fold
        term=(sin(pi*j/k)/(pi*j/k))^2;
        term_triangle=(cos(2*pi*j/k))^2/(pi^2*(1-(4*j/k)^2)^2);;
        signal=signal+term;
        signal_triangle=signal_triangle+term_triangle;
    end
    for j=(f_fold+1):nsamp
        term=(sin(pi*j/k)/(pi*j/k))^2;
        term_triangle=(cos(2*pi*j/k))^2/(pi^2*(1-(4*j/k)^2)^2);;
        noise=noise+term;
        noise_triangle= noise_triangle+term_triangle;
    end
    snrdb(m-3)=10*log10(signal/noise);
    snrdb_triangle(m-3)=10*log10(signal_triangle/noise_triangle);
end
plot(x,snrdb,x,snrdb_triangle,'r:');
xlabel('Samples per symbol');
ylabel('signal-to-aliasing noise ratio');
```



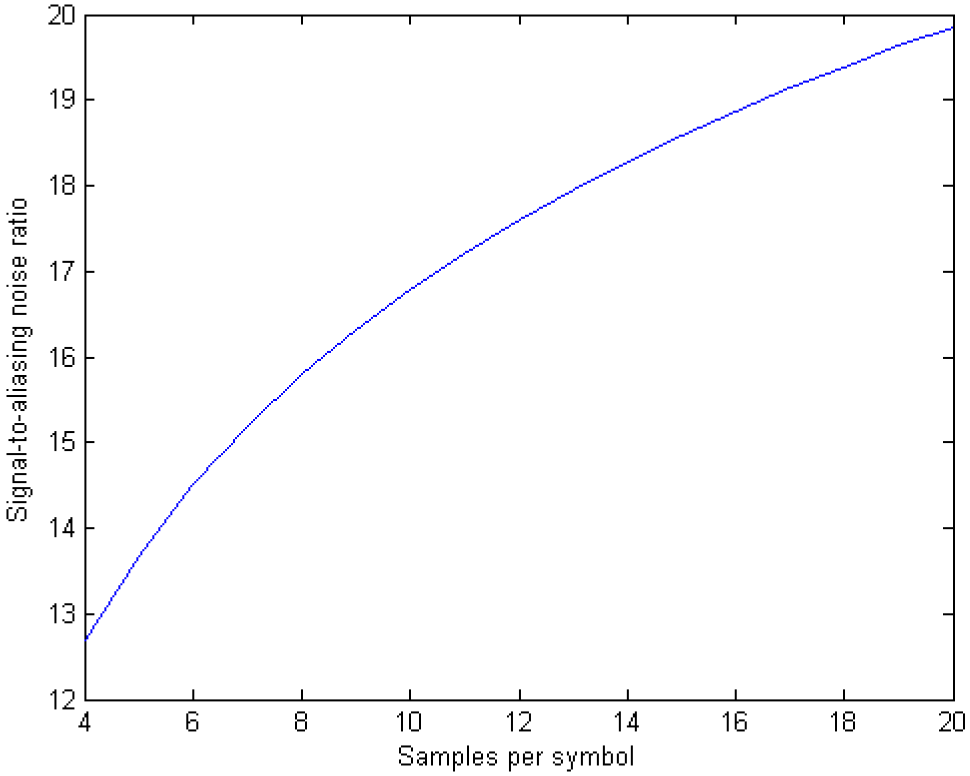
3.13

```
k = 50;                        % samples per lobe
nsamp = 50000;                 % total frequency samples
snrdb = zeros(1,17);           % initialize memory
x = 4:20;                      % vector for plotting
for m = 4:20
   signal = 0; noise = 0;      % initialize sum values
```

```
    f_fold = k*m/2;              % folding frequency
    for j = 1:f_fold
       term = (sin(pi*j/k)/(2*pi*j/k))^2;
       signal = signal+term;
    end
    for j = (f_fold+1):nsamp
       term = (sin(pi*j/k)/(2*pi*j/k))^2;
       noise = noise+term;
    end
    snrdb(m-3) = 10*log10(signal/noise);
end
plot(x,snrdb)                    % plot results
xlabel('Samples per symbol')
ylabel('Signal-to-aliasing noise ratio')
```



QPSK 混叠信噪比



矩形脉冲波形的信号混叠信噪比

第4章 带通信号与系统的低通仿真模型

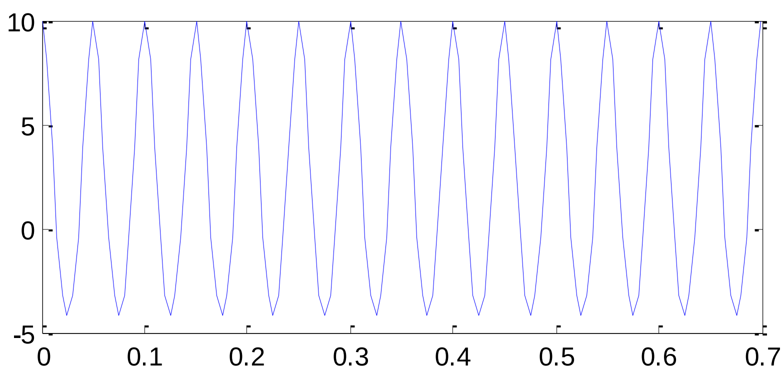
4-3

答：

（a）分析可知：$x_d = 10\cos(2\sin(20\pi t))$ $x_q = 1\,0\,\sin(2\,\sin\pi t($

画出同相分量 x_d： t= 0:0.005:0.7;

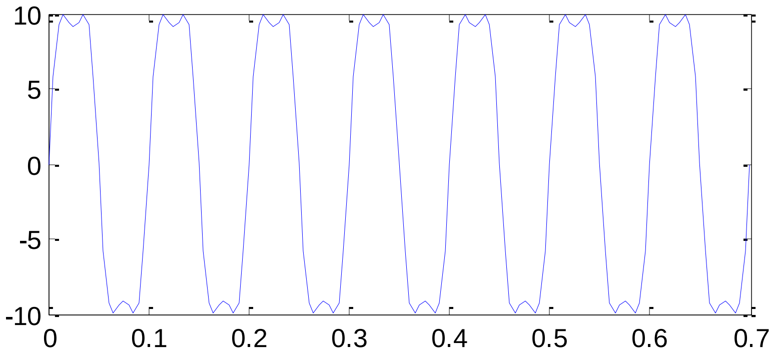
```
xd=10*cos(2*sin(20*pi*t));
plot(t,xd)
```



画出正交分量 x_q： t= 0:0.005:0.7;

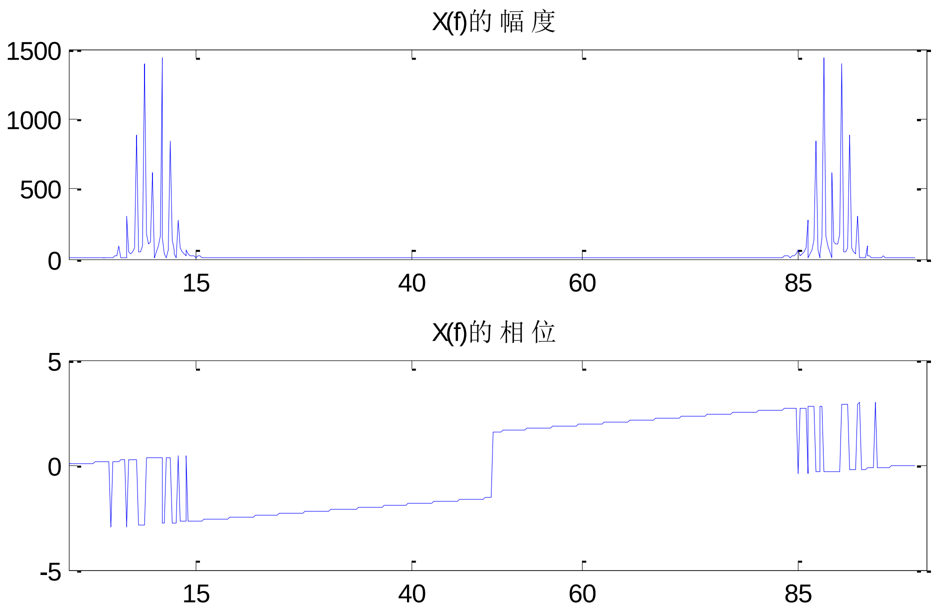
```
xq=10*sin(2*sin(20*pi*t));
```

plot(t,xq)



（b） matlab程序：

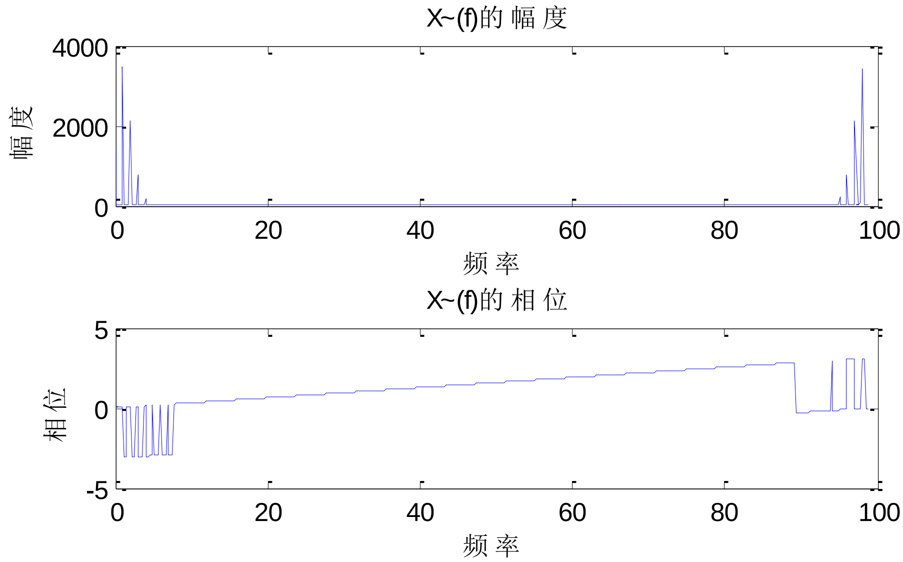
```
t = 0:0.001:0.5;
x = 10*cos(200*pi*t+2*sin(20*pi*t));
y = fft(x);
m = abs(y); p = angle(y);
f = (0:length(y)-1)*99/length(y);
subplot(2,1,1);
plot(f,m); title('X(f)的幅度');
set(gca,'XTick',[15 40 60 85]);
subplot(2,1,2);
plot(f,p); title('X(f)的相位');
set(gca,'XTick',[15 40 60 85]);
```



（c）matlab程序：

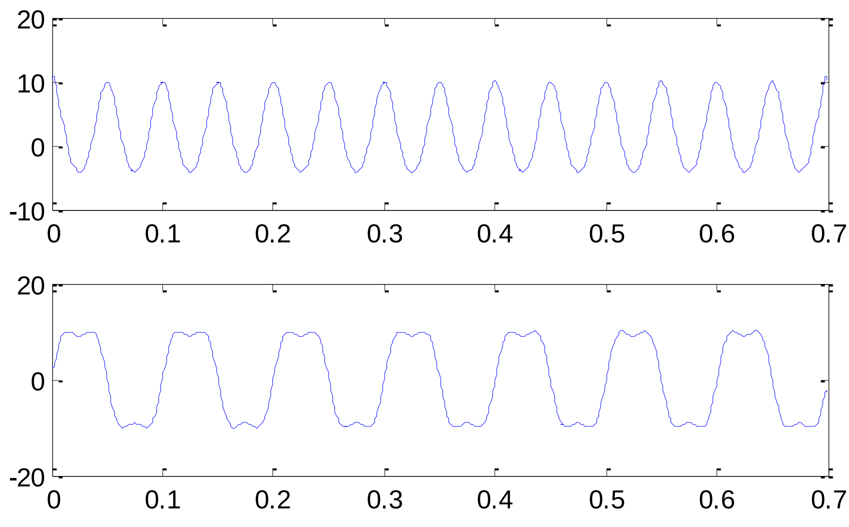
```
t = 0:0.001:0.6;
x = 10*cos(200*pi*t+2*sin(20*pi*t));
z=hilbert(x);
x1=fft(z.*exp(-j*2*pi*100*t));
m = abs(x1);
f = (0:length(x1)-1)*99/length(x1);
subplot(2,1,1);
plot(f,m);
title(' X~(f)的幅度');
xlabel('频率');
```

```
ylabel('幅度');
p =angle(x1);
subplot(2,1,2);
plot(f,p);
title(' X~(f)的相位');
xlabel('频率');
ylabel('相位');
```



(d) matlab程序：

```
t = 0:0.001:0.7;
x = 10*cos(200*pi*t+2*sin(20*pi*t));
z=hilbert(x);
xl=z.*exp(-j*2*pi*100*t);
xd=real(xl);
xq=-j*(xl-xd);
subplot(2,1,1)
plot(t,xd);
subplot(2,1,2);
plot(t,xq);
```

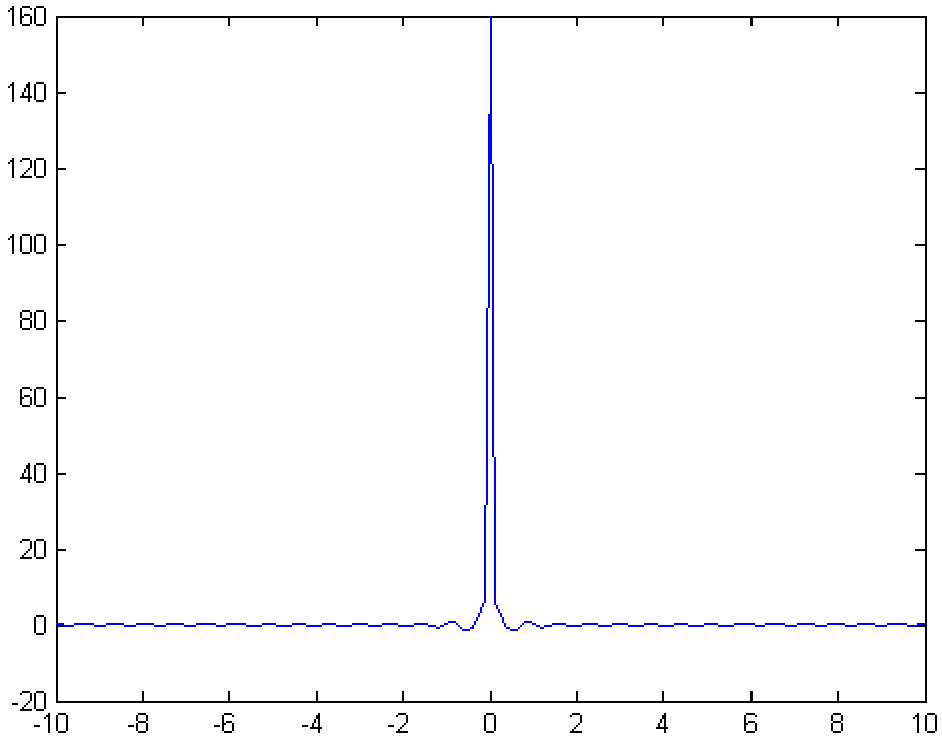


（e）由图可知，该结果与解析结果是一致的。

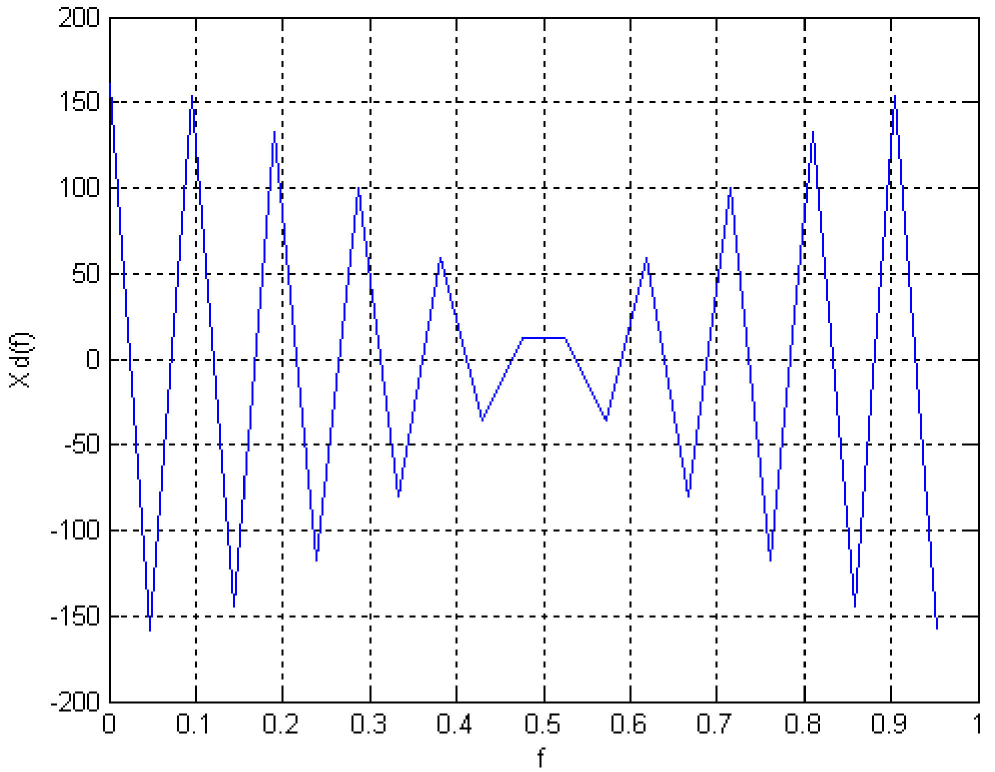
4.8

由傅立叶反变换：$x(t)=\int_{-\infty}^{\infty}X(f)\exp(j2\pi t)df$ 可以得：$X_d(f)=(8/\pi t)\sin(20\pi t)$

（a） 当 $f_0=100$， $x(t)=(8/\pi t)\sin(20\pi t)$，所以 $x_d(t)=(8/\pi t)\sin(20\pi t)$，

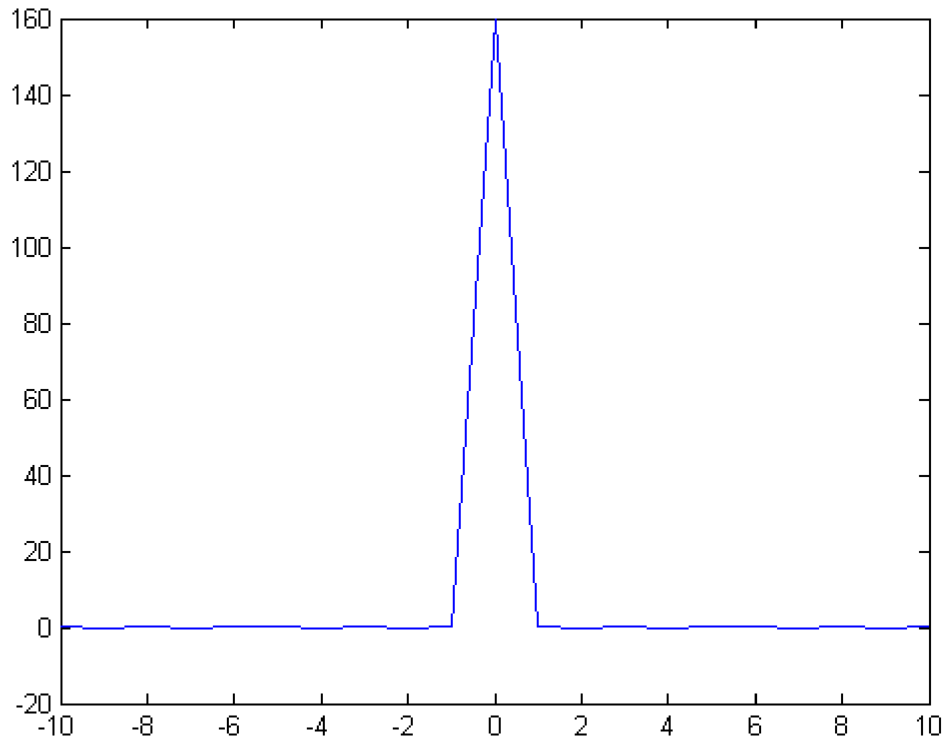


$x_d(t)$ 的时域图形

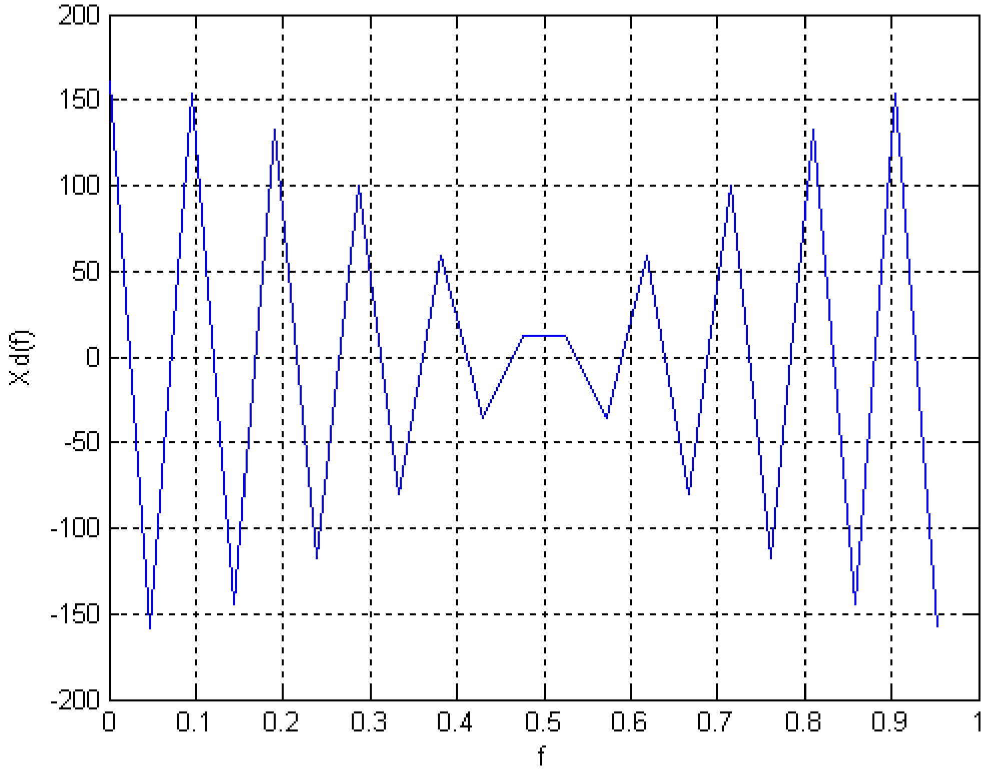


$X_d(f)$ 的频域图形

（b） 当 $f_0 = 90$， $x_d(t) = (8/\pi t)\sin(20\pi t)\cos(10\pi t)$，

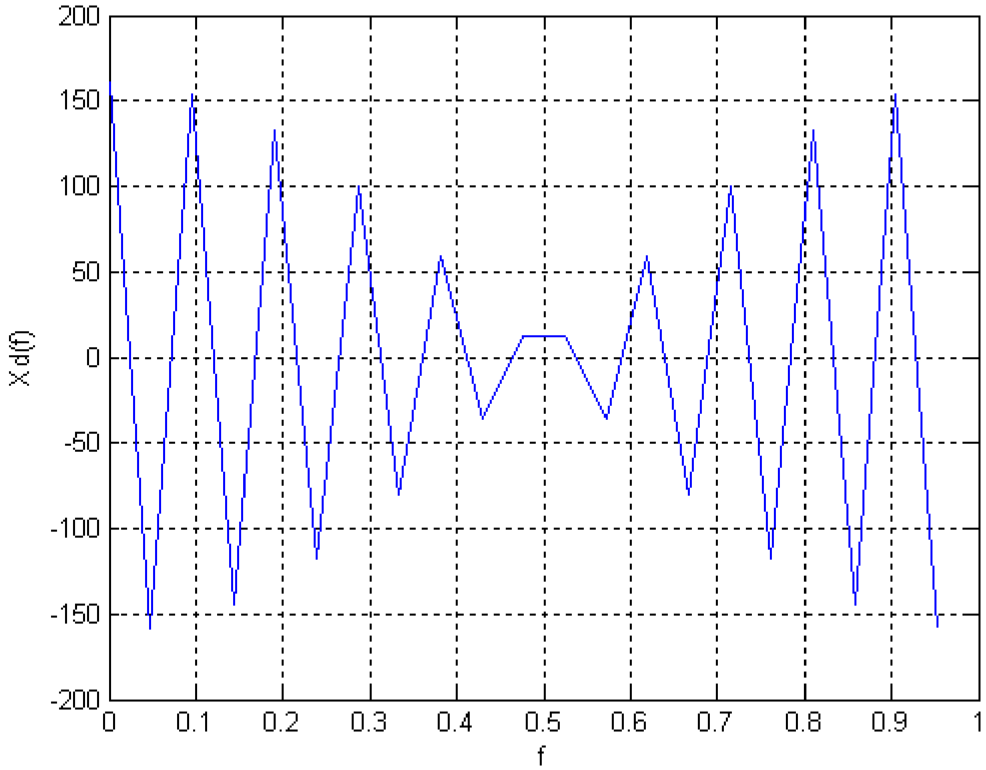


$x_d(t)$ 的时域图形

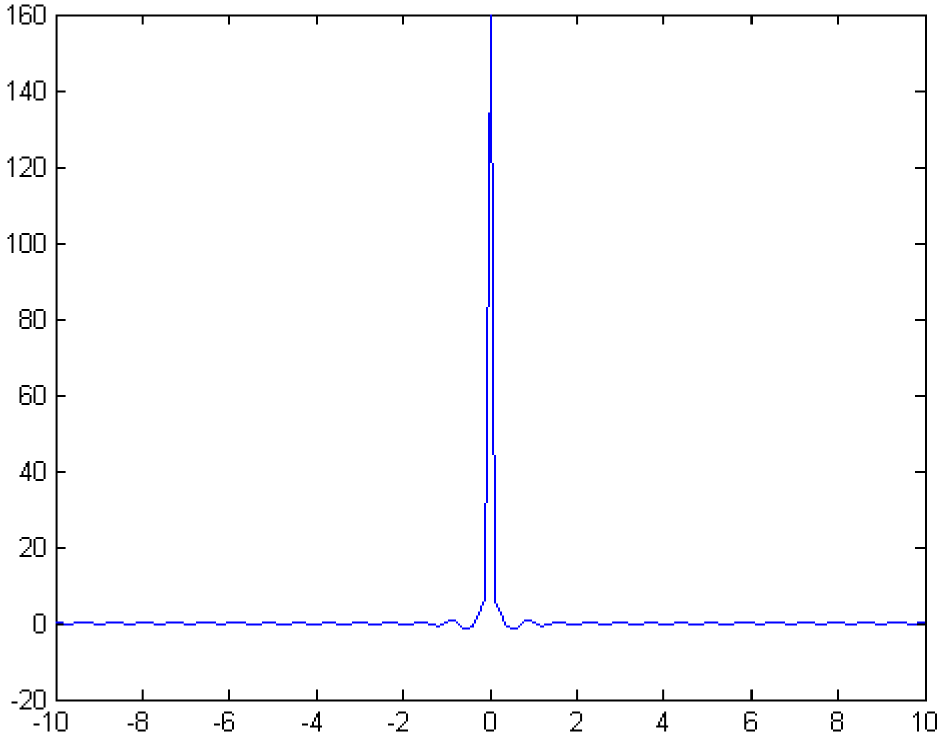


$X_d(f)$ 的频域图形

（c） 当 $f_0 = 90$， $x_d(t) = (8/\pi t)\sin(20\pi t)\cos(20\pi t)$



$X_d(f)$ 的频域图形



$x_d(t)$ 的时域图形

4. 11

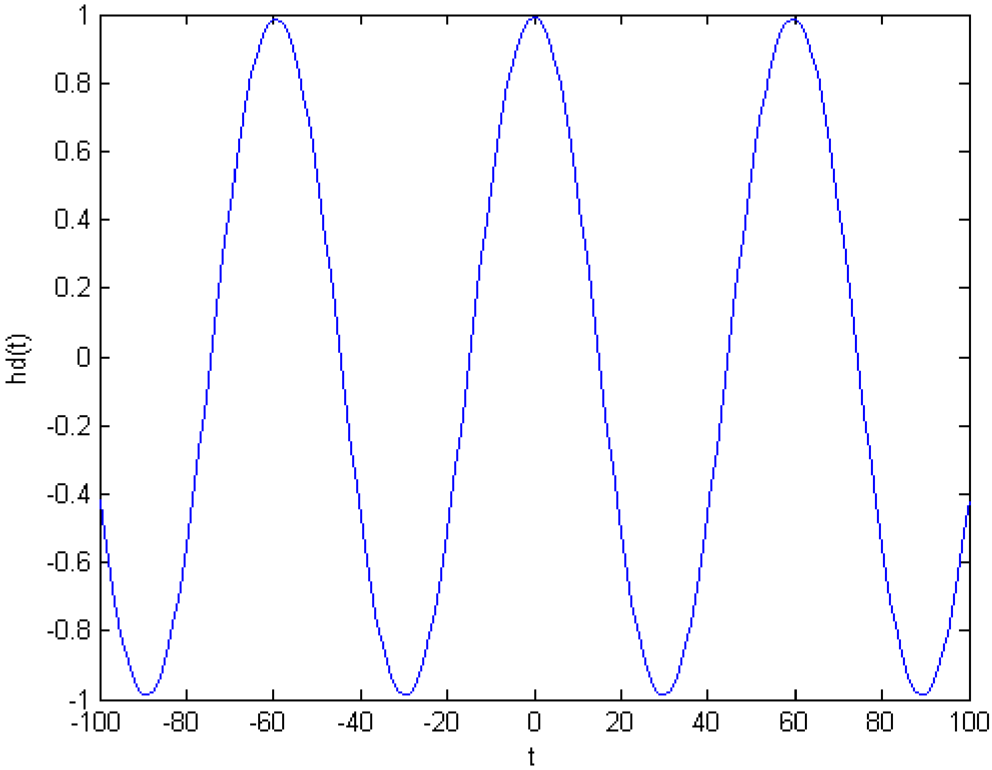
（a） $H(s) \leftrightarrow h(t) = 0.8944\exp(\omega_b/2)\cos(\omega_0 t - 0.1056\omega_0 t)$

$$h(t) = 0.8944\exp(\omega_b/2)\exp(-0.1056\omega_0 t)$$

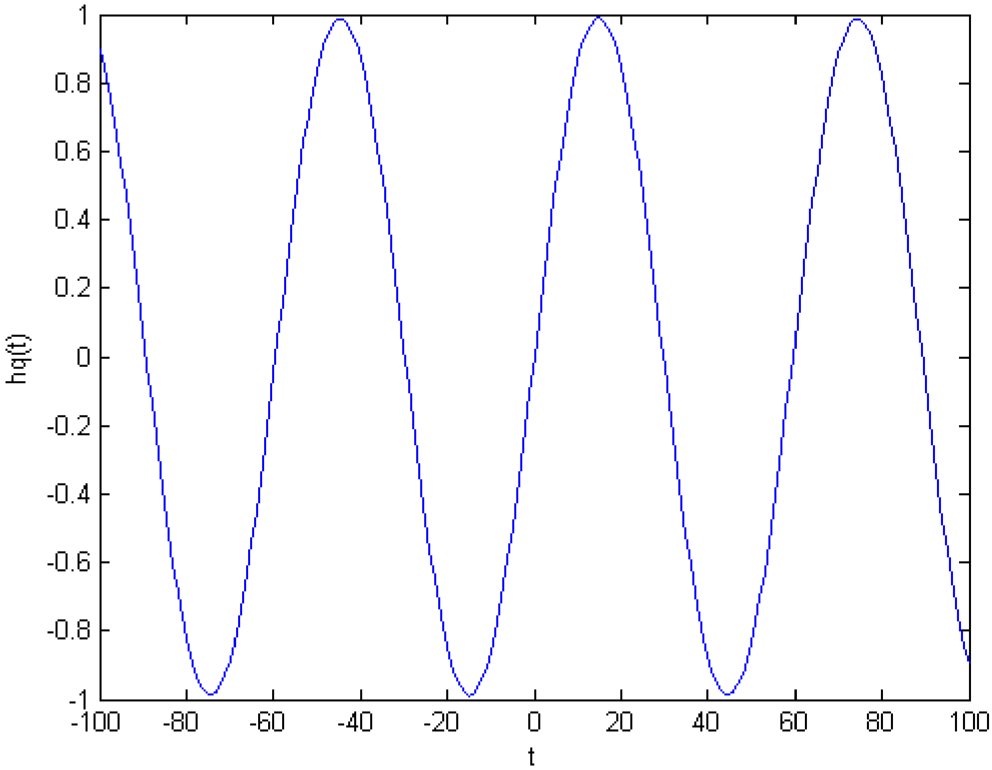
$$h_d(t) = 0.8944\exp(\omega_b/2)\cos(0.1056\omega_0 t)$$

$$h_q(t) = 0.8944\exp(\omega_b/2)\sin(0.1056\omega_0 t)$$

（b）当 $\omega_b = 0.2, \omega_0 = 1$时，

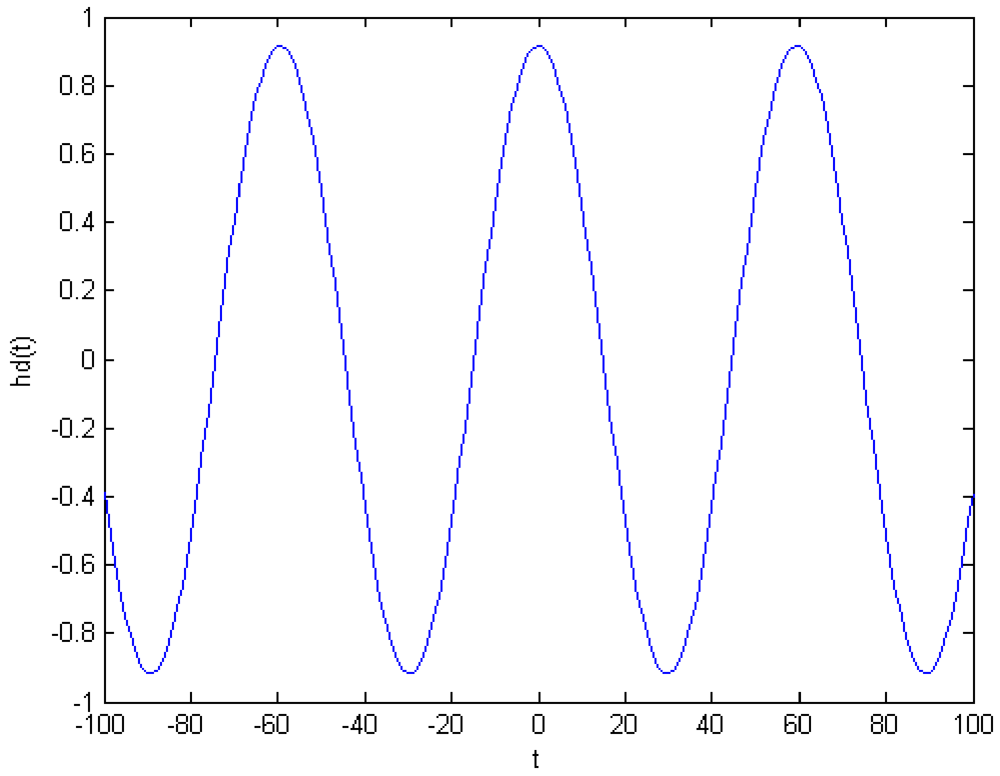


$h_d(t)$的图形

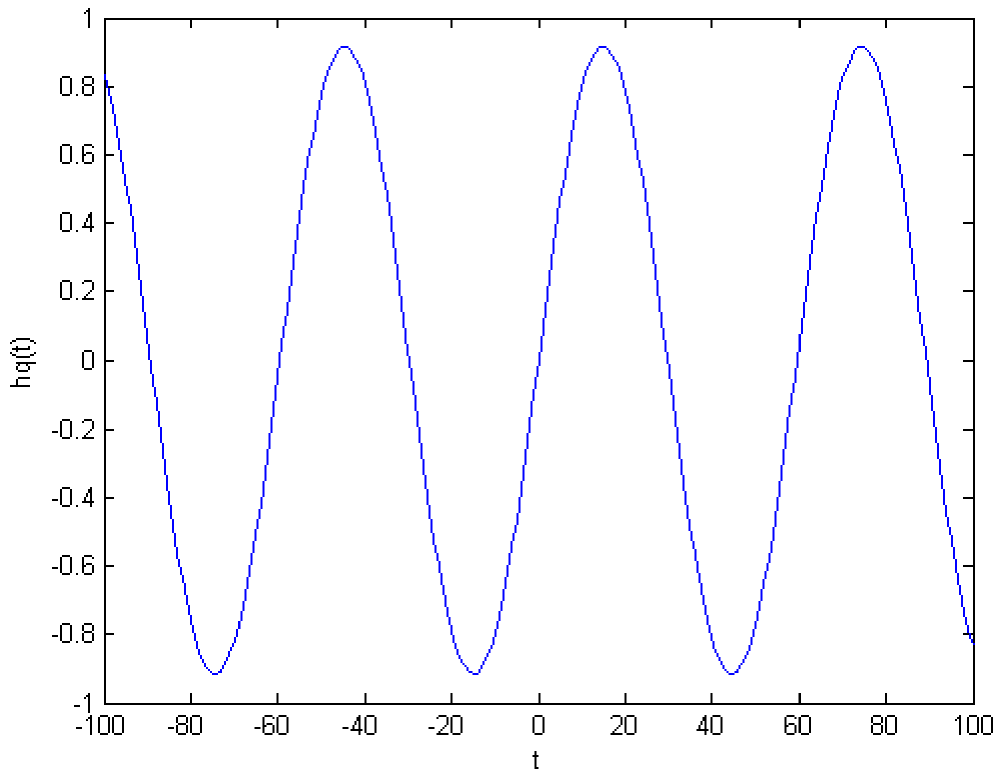


$h_q(t)$的图形

（d） 当$\omega_b=0.05,\omega_0=1$时，



$h_d(t)$ 的图形



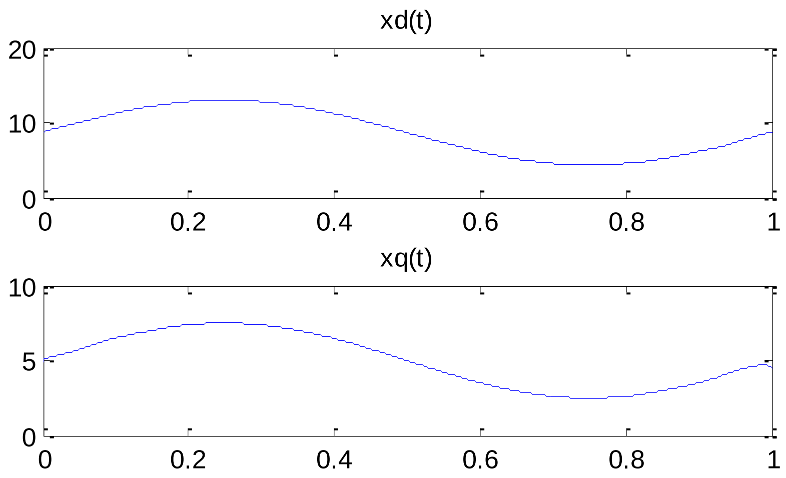
$h_q(t)$ 的图形

4-12

答：

（a）matlab 程序：

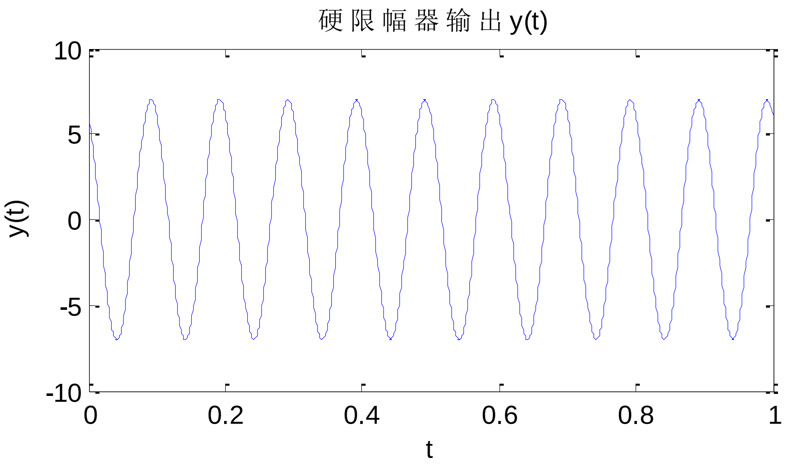
```
t=0:0.001:1;
x=5*(2+sin(2*pi*t)).*cos(20*pi*t+pi/6);
z=hilbert(x);
f0=10;
xl=z.*exp(-j*2*pi*f0*t);
xd=real(xl);
xq=-j*(xl-xd);
subplot(2,1,1)
plot(t,xd); title('xd (t)');
subplot(2,1,2);
plot(t,xq); title('xq (t)');
```



其中 x1 即为 x(t)的复包络 $\tilde{x}(t)$。

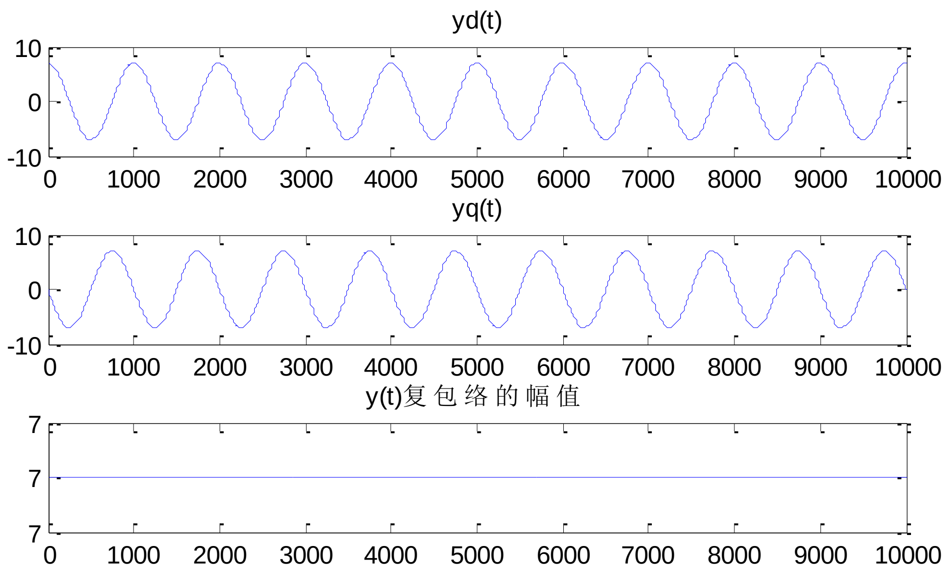
(b) matlab 程序：

```
t=0:0.001:1;
x=5*(2+sin(2*pi*t)).*cos(20*pi*t+pi/6);
z=hilbert(x);
f0=10;
xl=z.*exp(-j*2*pi*f0*t);
xd=real(xl);
xq=-j*(xl-xd);
yd=(7*xd)/sqrt(xd.*xd+xq.*xq);
yq=(7*xq)/(sqrt(xd.*xd+xq.*xq));
yl=yd+j*yq;
z1=exp(j*2*pi*f0*t)*yl;
y=real(z1);
plot(t,y);
title(' 硬限幅器输出y(t)');
xlabel('t');
ylabel(' y(t)');
```



（c）matlab程序：

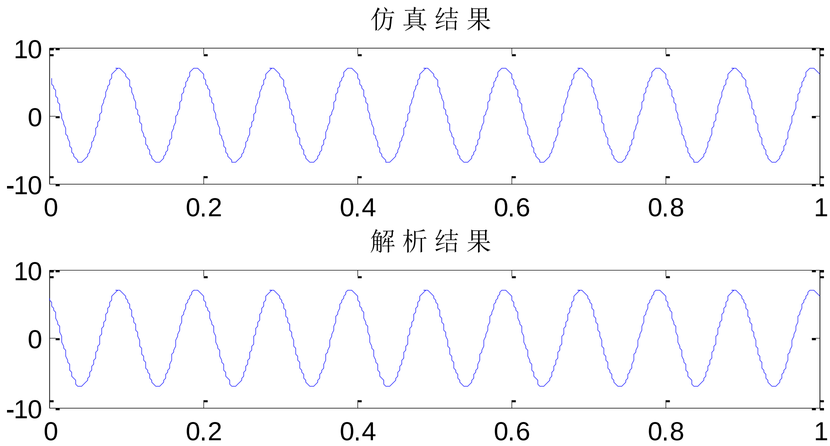
```
t=0.0001:0.0001:1;
At=5*(2+sin(2*pi*t));f0=10;seta=pi/6;
xt=At.*cos(2*pi*f0*t+seta);
xlt=At.*exp(j*seta);
B=7;
Y2=hilbert(real(xlt));
y=y2.*exp(-j*2*pi*f0*t);
xdty=real(y);xqty=imag(y);
ydt=(B*xdty)./sqrt((xdty.^2)+(xqty.^2));
yqt=(B*xqty)./sqrt((xdty).^2+(xqty).^2);
ylt=ydt+j*yqt;
subplot(3,1,1);plot(ydt); title('yd(t)');
subplot(3,1,2);plot(yqt); title('yq(t)');
subplot(3,1,3);plot(abs(ylt)); title('y(t)复包络的幅值');
```



（d）matlab 程序：

```
t=0.001:0.001:1;
x=5*(2+sin(2*pi*t)).*cos(20*pi*t+pi/6);
z=hilbert(x);
f0=10;
xl=z.*exp(-j*2*pi*f0*t);
xd=real(xl);
```

```
xq=-j*(xl-xd);
yd=(7*xd)/sqrt(xd.*xd+xq.*xq);
yq=(7*xq)/(sqrt(xd.*xd+xq.*xq));
yl=yd+j*yq;
z1=exp(j*2*pi*f0*t)*yl;
y=real(z1);
yt=7*cos(20*pi*t+pi/6);
subplot(2,1,1);plot(t,y);title('仿真结果')；
subplot(2,1,2);plot(t,yt);title('解析结果')；
```



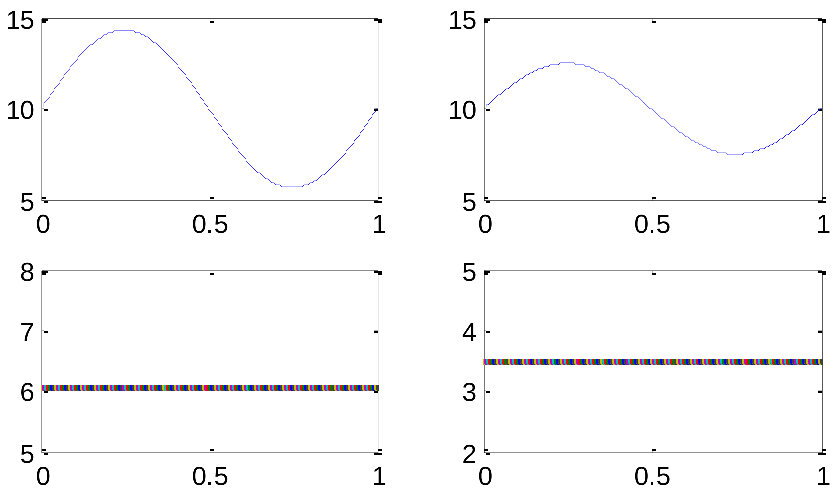
故，由上图分析可知，仿真产生的结果和解析结果是基本一致的。

（e）解析表达式：$\mathrm{x_d}(t)=5(2+\sin 2\pi t)\cos(\dfrac{\pi}{6})$，$\mathrm{x_q}(t)=5(2+\sin 2\pi t)\sin(\dfrac{\pi}{6})$，

$$y_{\mathrm{d}}(t)=7\cos(\frac{\pi}{6}),\quad y_{\mathrm{q}}(t)=7\sin(\frac{\pi}{6})$$

matlab 程序：

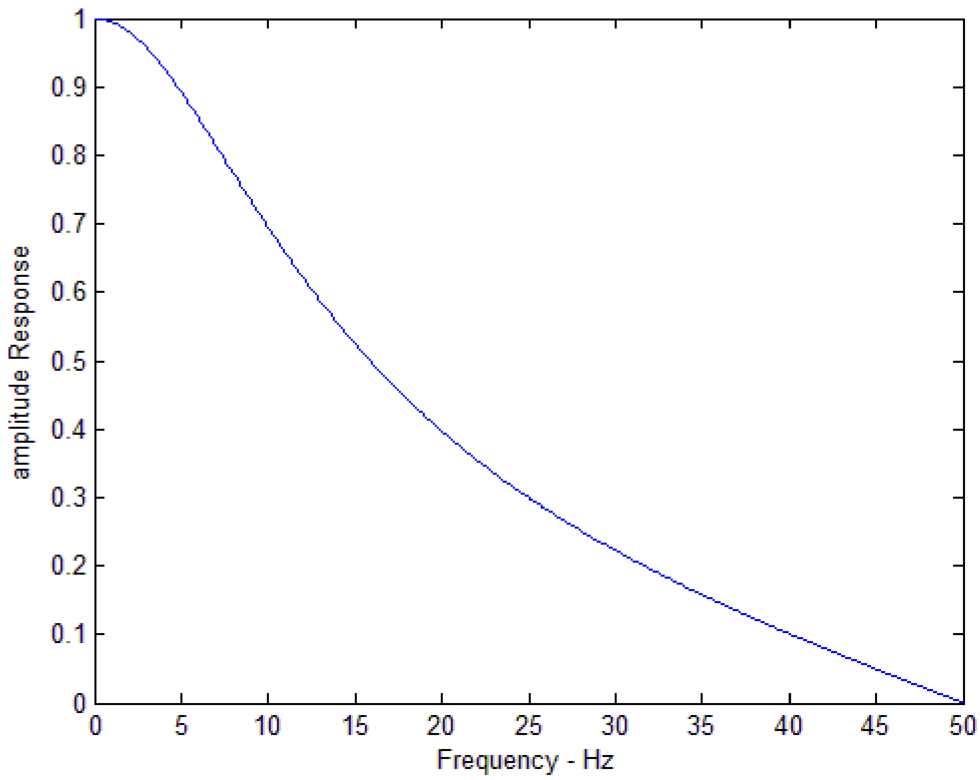
```
t=0:0.001:1;
xd=5*(2+sin(2*pi*t)*cos(pi/6));
xq=5*(2+sin(2*pi*t)*sin(pi/6));
yd=7*cos(pi/6);
yq=7*sin(pi/6);subplot(2,2,1);plot(t,xd);
subplot(2,2,2);plot(t,xq);
subplot(2,2,3);plot(t,yd);
subplot(2,2,4);plot(t,yq);
```



第 5 章 滤波器模型与仿真方法

5.10

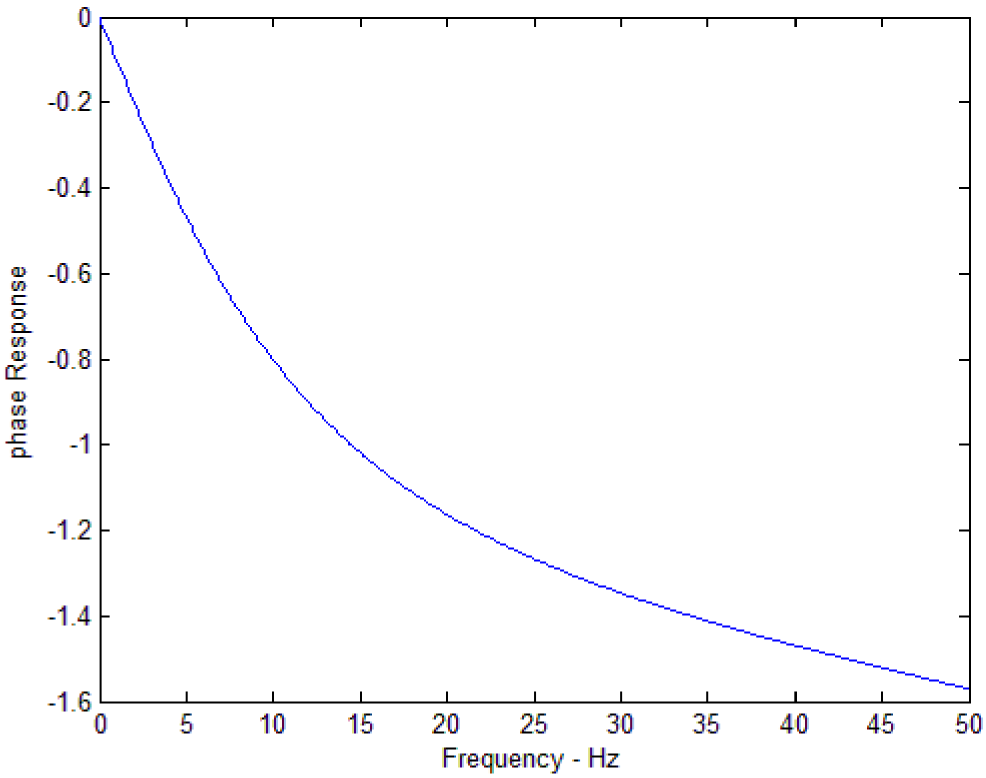
```
clear all;
T = 0.01;
f = 0:0.1:50;
z = exp(-i*2*pi*f*T);                            % see (5.4)
a0 = 0.239057; a1 = 0.239057; b1 = 0.521886;     % bilinear invariant
num = a0+a1*z;
den = 1-b1*z;
ampx=abs(num./den);
plot(f,ampx);
xlabel('Frequency - Hz')
ylabel('amplitude Response')
% End of script file.
```



幅度响应

```
clear all;
T = 0.01;
f = 0:0.1:50;
z = exp(-i*2*pi*f*T);                            % see (5.4)
a0 = 0.239057; a1 = 0.239057; b1 = 0.521886;     % bilinear invariant
num = a0+a1*z;
den = 1-b1*z;
anglex=angle(num./den);
```

```
plot(f,anglex);
xlabel('Frequency - Hz')
ylabel('phase Response')
% End of script file.
```



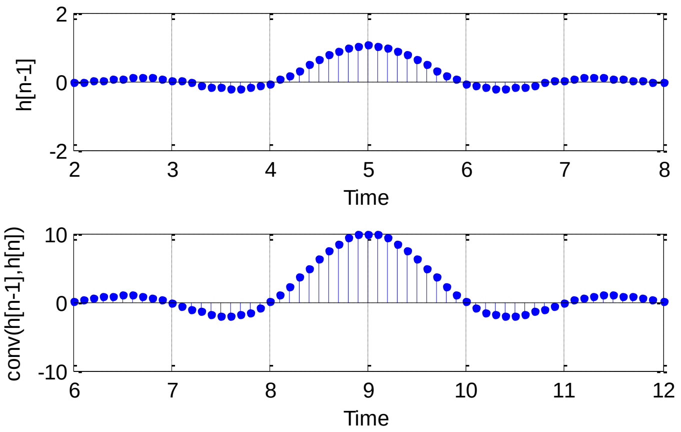
相位响应

5-12

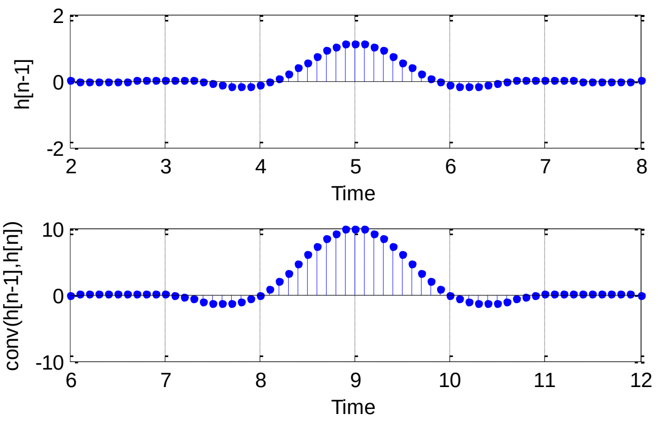
答：matlab 程序：

```
T = 1;
k = 10;
m = 4;
beta = 0.2;
n = 0:2*m*k;
z = (n/k)-m+eps;
t1 = cos((1+beta)*pi*z);
t2 = sin((1-beta)*pi*z);
t3 = 1./(4*beta*z);
den = 1-16*beta*beta*z.*z;
num = t1+t2.*t3;
c = 4*beta/(pi*sqrt(T));
h = c*num./den;
in = zeros(1,101); in(11) = 1;
out = conv(in,h);
out1 = conv(out,h);
t = 2:0.1:8;
subplot(2,1,1);
stem(t,out(21:81),'.')
```

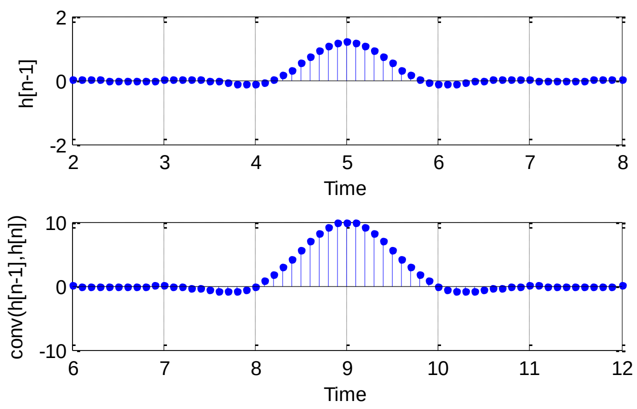
```
grid;
xlabel('Time');
ylabel('h[n-1]');
subplot(2,1,2);
t = 6:0.1:12;
stem(t,out1(61:121),'.');
grid;
xlabel('Time');
ylabel('conv(h[n-1],h[n])');
```



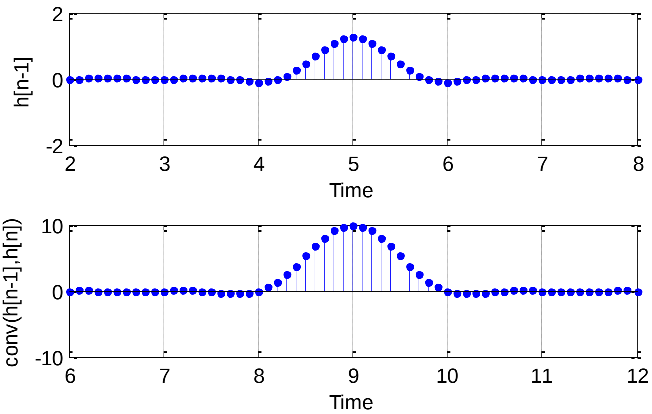
beta = 0.5 时：



beta = 0.7 时：



beta = 0.9 时：

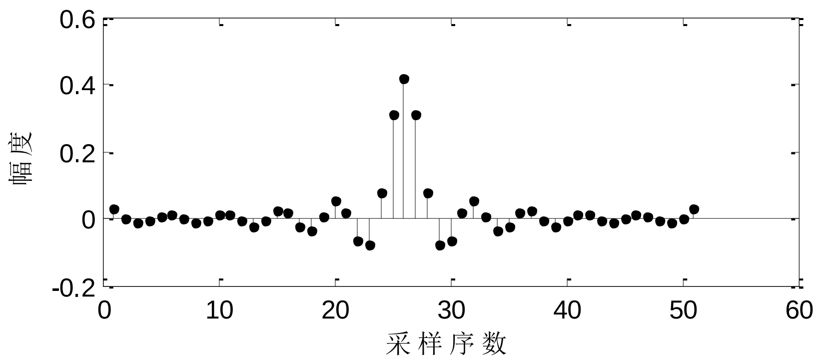


5-13

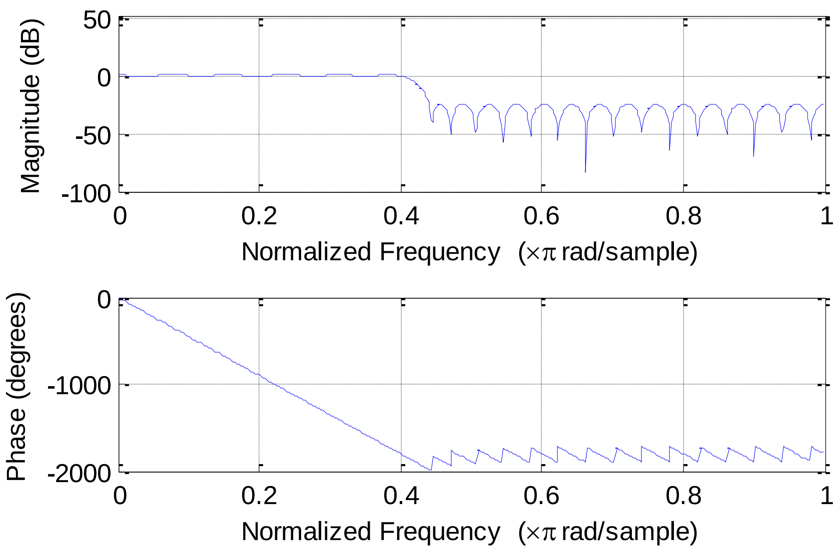
答：

（a）matlab 程序：

```
order=50;
f=[0,0.4,0.44,1];
amp=[1 1 0 0 ];
b=remez(order,f,amp);
stem(b,'.k')
xlabel('采样序数')
ylabel('幅度')
pause
freqz(b,1)
```



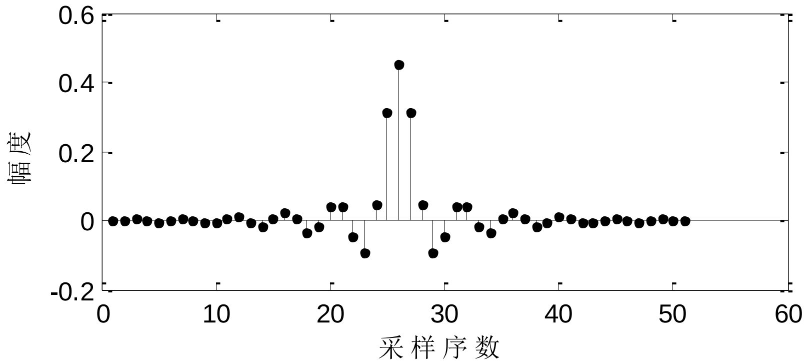
（b）由（a）中程序可知幅度响应和相位响应如下：



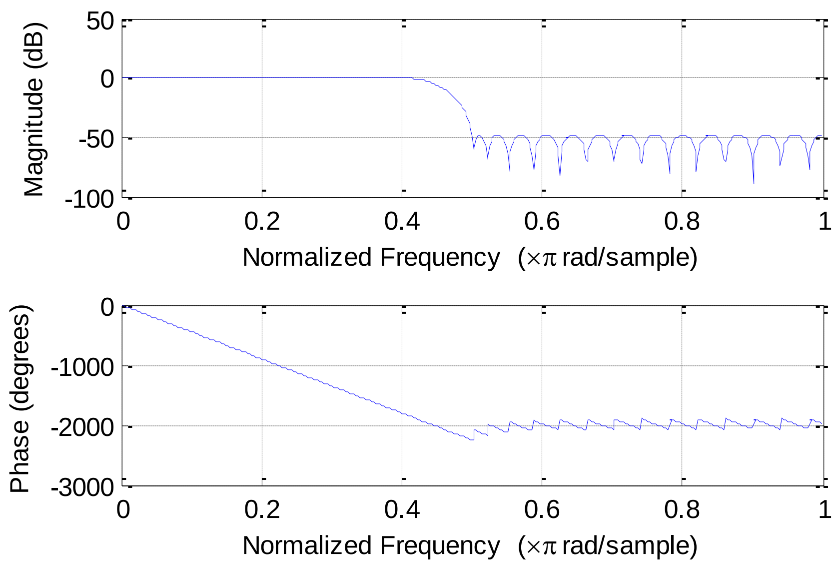
阻带衰减约为 25dB.

（c）过渡带为 20HZ<|f|<25 HZ 时，

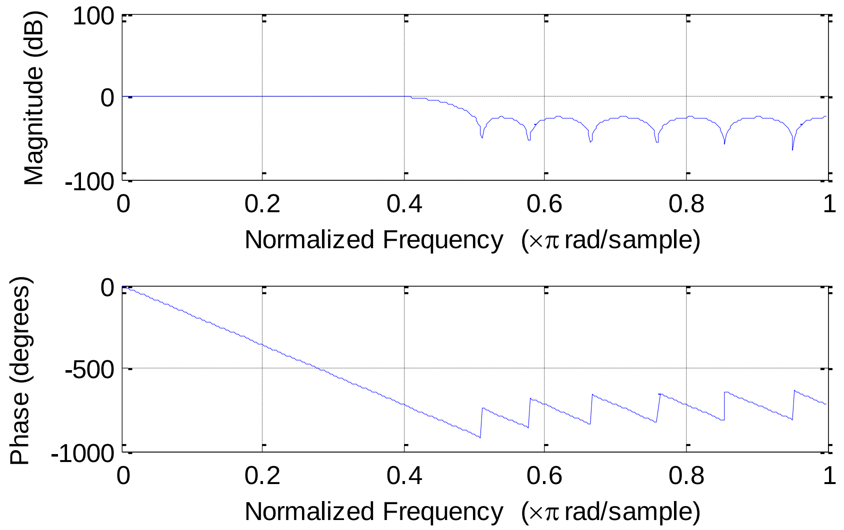
```
order = 50;
f = [0 0.4 0.5 1];
amp = [1 1 0 0];
b = remez(order,f,amp);
stem(b,'.k')
xlabel('采样序数')
ylabel('幅度')
pause
freqz(b,1)
```



幅度和相位响应如下：



当阶数 order 为 20 时，幅度响应与相位响应如下图所示：



所以，可以通过降低滤波器阶数来降低阻带衰减。

第 6 章 案例研究：锁相环与微分方程方法

6.3

第一步：

```
% File: pllpre.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
%
clear all                          % be safe
disp(' ')                          % insert blank line
fdel = input('Enter the size of the frequency step in Hertz > ');
fn = input('Enter the loop natural frequency in Hertz > ');
lambda = input('Enter lambda, the relative pole offset > ');
disp(' ')
disp('Accept default values:')
disp('  zeta = 1/sqrt(2) = 0.707,')
disp('  fs = 200*fn, and')
disp('  tstop = 1')
dtype = input('Enter y for yes or n for no  > ','s');
if dtype == 'y'
   zeta = 1/sqrt(2);
   fs = 200*fn;
   tstop = 1;
else
zeta = input('Enter zeta, the loop damping factor > ');
fs = input('Enter the sampling frequency in Hertz > ');
tstop = input('Enter tstop, the simulation runtime > ');
end
%
npts = fs*tstop+1;                 % number of simulation points
t = (0:(npts-1))/fs;               % default time vector
```

```
nsettle = fix(npts/10);          % set nsettle time as 0.1*npts
tsettle = nsettle/fs;            % set tsettle
%
% The next two lines establish the loop input frequency and phase
% deviations.
%
fin = [zeros(1,nsettle),fdel*ones(1,npts-nsettle)];
phin = [zeros(1,nsettle),2*pi*fdel*t(1:(npts-nsettle))];
%
disp(' ')                        % insert blank line
%       end of script file pllpre.m
```

在 command window 中进行一下对话:

```
Enter the size of the frequency step in Hertz > 40

Enter the loop natural frequency in Hertz > 10

Enter lambda, the relative pole offset > 0


Accept default values:

 zeta = 1/sqrt(2) = 0.707,

 fs = 200*fn, and

 tstop = 1

Enter y for yes or n for no  > n

Enter zeta, the loop damping factor > 1/sqrt(2)

Enter the sampling frequency in Hertz > 5000

Enter tstop, the simulation runtime > 0.8
```

第二步:

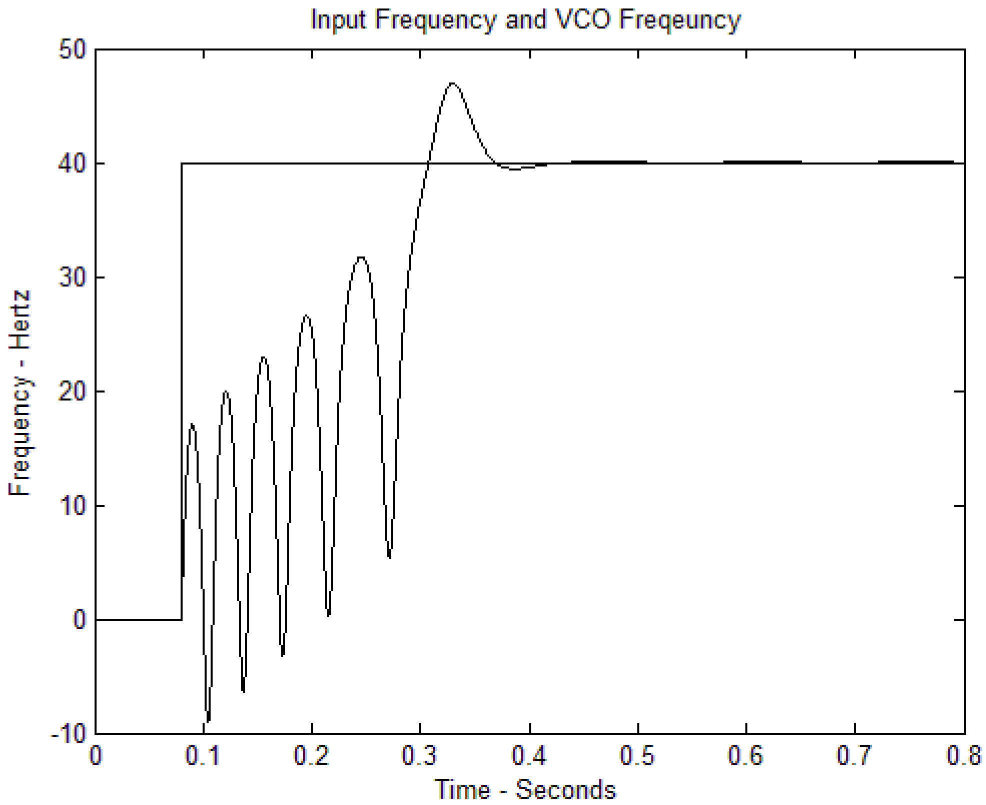
```
%File: c6_PLLsim.m
w2b=0;w2c=0;s5=0;phivco=0;      %initialize
twopi=2*pi;           %define 2*pi
twofs=2*fs;            %define 2*fs
G=2*pi*fn*(zeta+sqrt(zeta*zeta-lambda));    %set loop gain
a=2*pi*fn/(zeta+sqrt(zeta*zeta-lambda));     %set filter parameter
a1=a*(1-lambda);a2=a*lambda;     %define  constants
phierror=zeros(1,npts);    %initialize  vector
fvco=zeros(1,npts);        %initialize vector

% beginning of simulation loop
for i=1:npts
    s1=phin(i)-phivco;  %phase error
    s2=sin(s1);   %sinusoidal phase detector
    s3=G*s2;
```

```
    s4=a1*s3;
    s4a=s4-a2*s5;  %loop filter integrator input
    w1b=s4a+w2b;   %filter integrator (step 1)
    w2b=s4a+w1b;  %filter integrator (step 2)
    s5=w1b/twofs;  %generate filter output
    s6=s3+s5;  %VCO integrator input
    w1c=s6+w2c;   %VCO integrator (step 1)
    w2c=s6+w1c;   %VCO integrator (step 2)
    phivco=w1c/twofs;   %generate VCO output
    phierror(i)=s1;   %build phase error vector
    fvco(i)=s6/twopi;   %build VCO input vector
end
% end of simulation loop
freqerror=fin-fvco;   %build frequency error vector
%End of script file
```

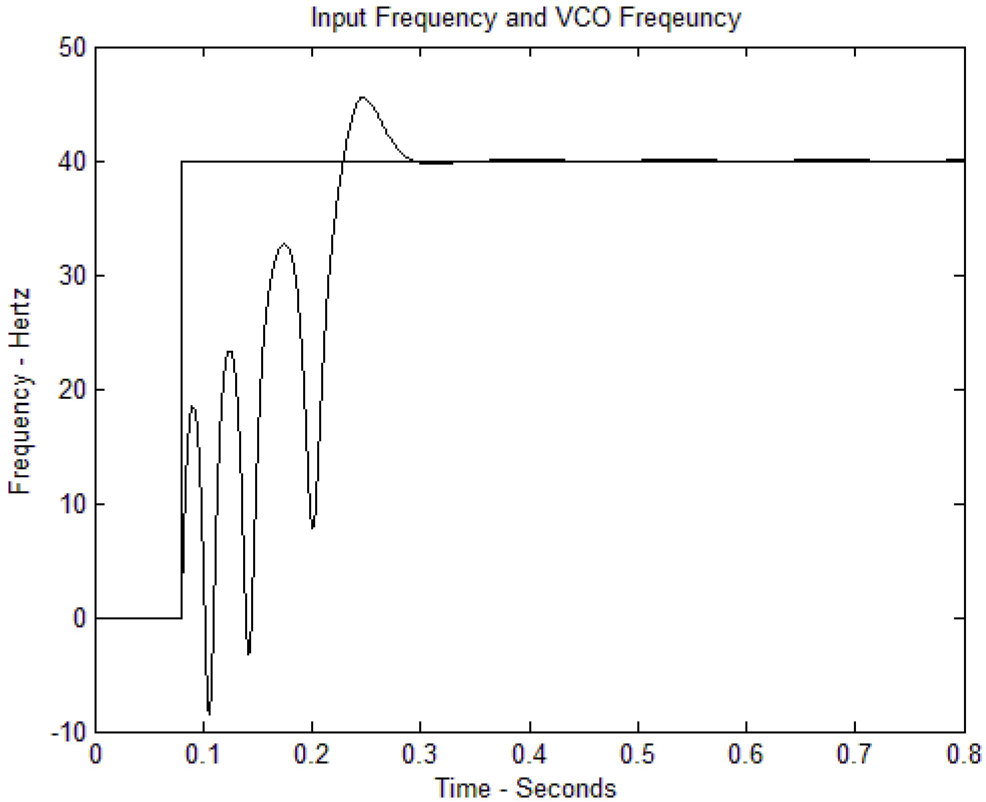
第三步：

```
% File: pllpost.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
%
kk = 0;
while kk == 0
k = menu('Phase Lock Loop Postprocessor',...
        'Input Frequency and VCO Frequency',...
        'Input Phase and VCO Phase',...
        'Frequency Error','Phase Error','Phase Plane Plot',...
        'Phase Plane and Time Domain Plots','Exit Program');
        if k == 1
                plot(t,fin,'k',t,fvco,'k')
                title('Input Frequency and VCO Freqeuncy')
                xlabel('Time - Seconds');ylabel('Frequency - Hertz');pause
        elseif k ==2
                pvco=phin-phierror;plot(t,phin,t,pvco)
                title('Input Phase and VCO Phase')
                xlabel('Time - Seconds');ylabel('Phase - Radians');pause
        elseif k == 3
                plot(t,freqerror);title('Frequency Error')
                xlabel('Time - Seconds');ylabel('Frequency Error - Hertz');pause
        elseif k == 4
                plot(t,phierror);title('Phase Error')
                xlabel('Time - Seconds');ylabel('Phase Error - Radians');pause
        elseif k == 5
                 ppplot
        elseif k == 6
```

```
        subplot(211);phierrn = phierror/pi;
        plot(phierrn,freqerror,'k');grid;
        title('Phase Plane Plot');xlabel('Phase Error /Pi');
        ylabel('Frequency Error - Hertz');subplot(212)
        plot(t,fin,'k',t,fvco,'k');grid
        title('Input Frequency and VCO Freqeuncy')
        xlabel('Time - Seconds');ylabel('Frequency - Hertz');subplot(111)
    elseif k == 7
        kk = 1;
    end
end
% End of script file.
```



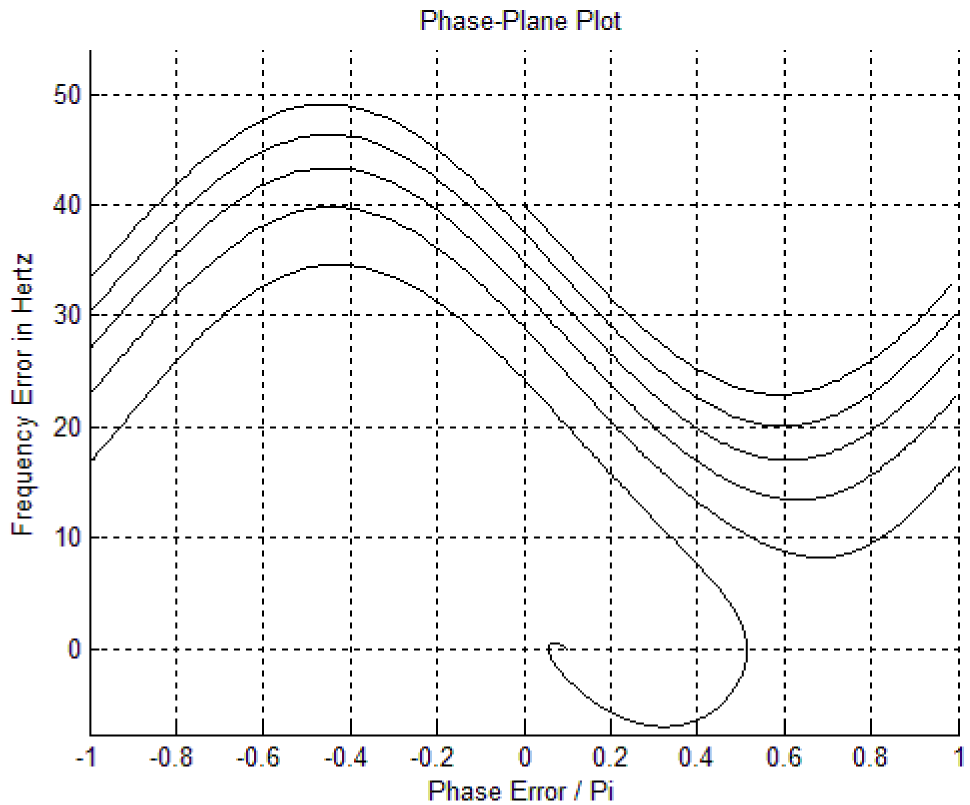
当相对极点偏移$\lambda = 0.10$时，有6个滑动周期



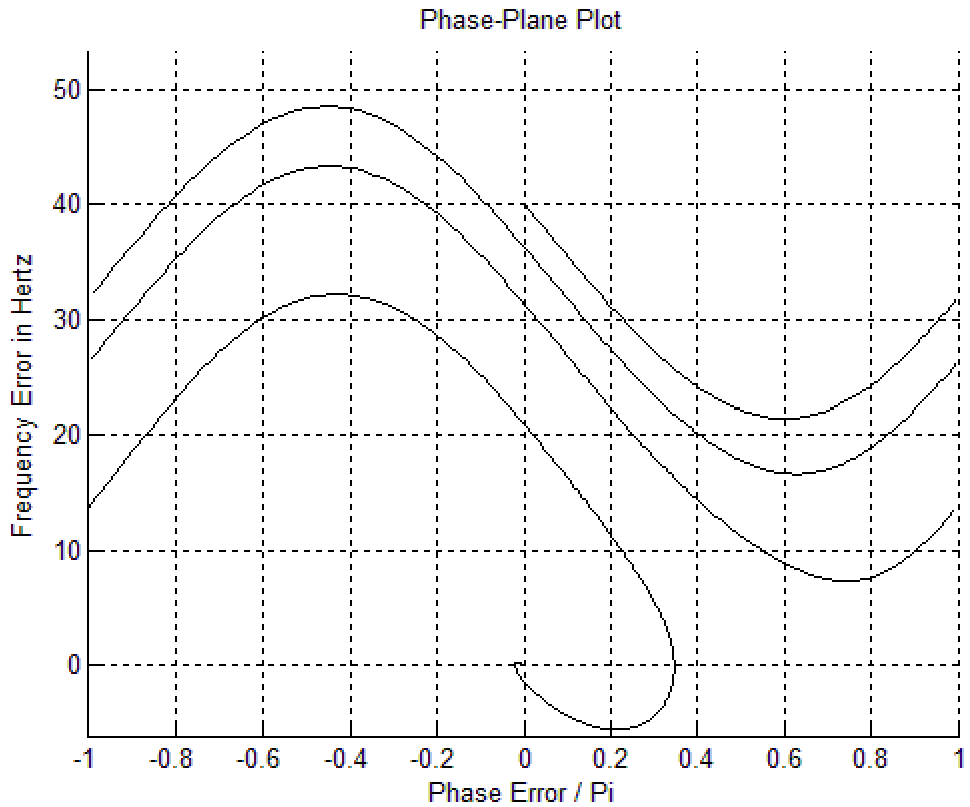
当相对极点偏移$\lambda=0$时，有4个滑动周期，缩短了2个周期，时间降低了0.1s。

6.4

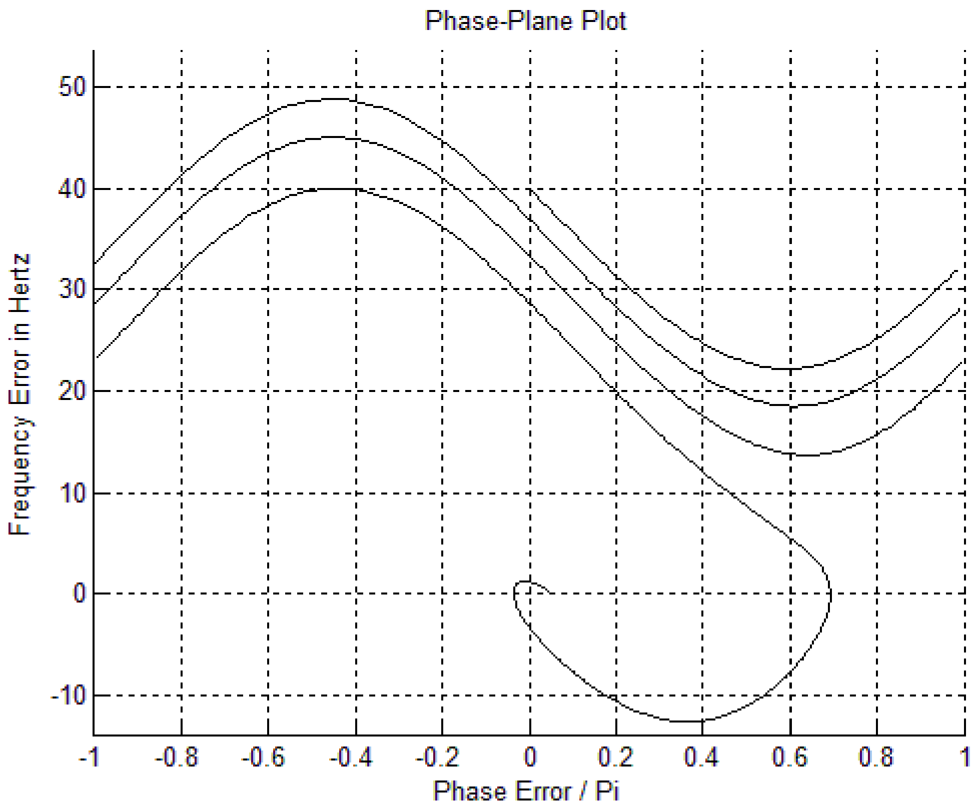
当相对极点偏移$\lambda=0.10$时，相位差随频率差如下图：



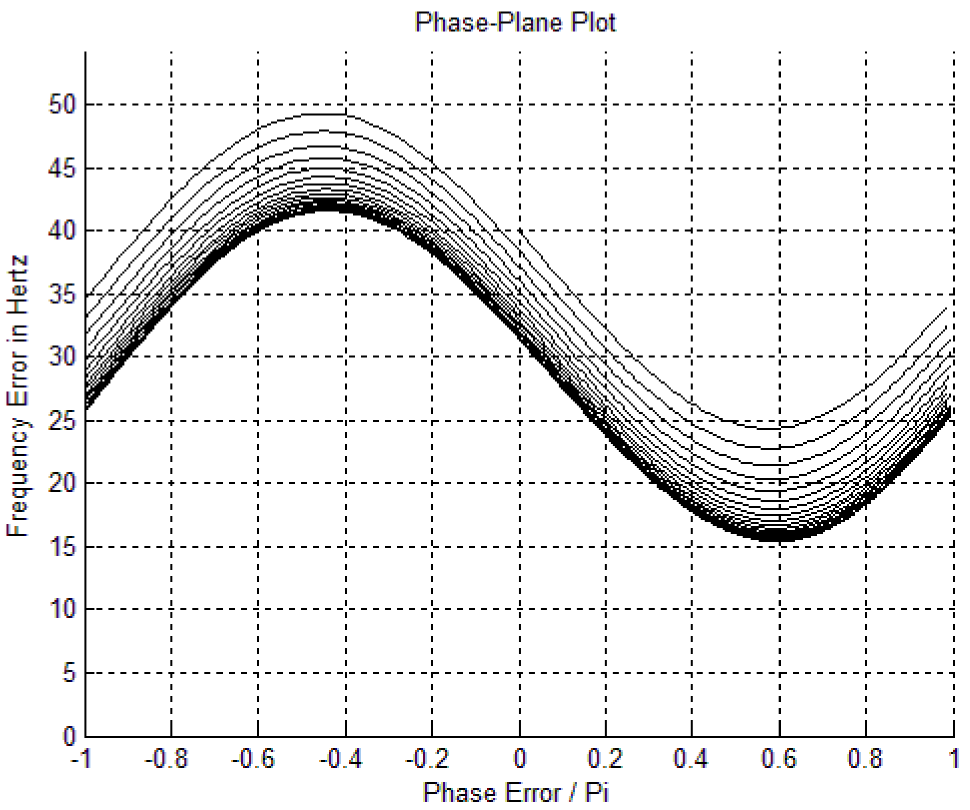
当相对极点偏移$\lambda=0$时，相位差随频率差如下图：



当相对极点偏移$\lambda = 0.05$时，相位差随频率差如下图：



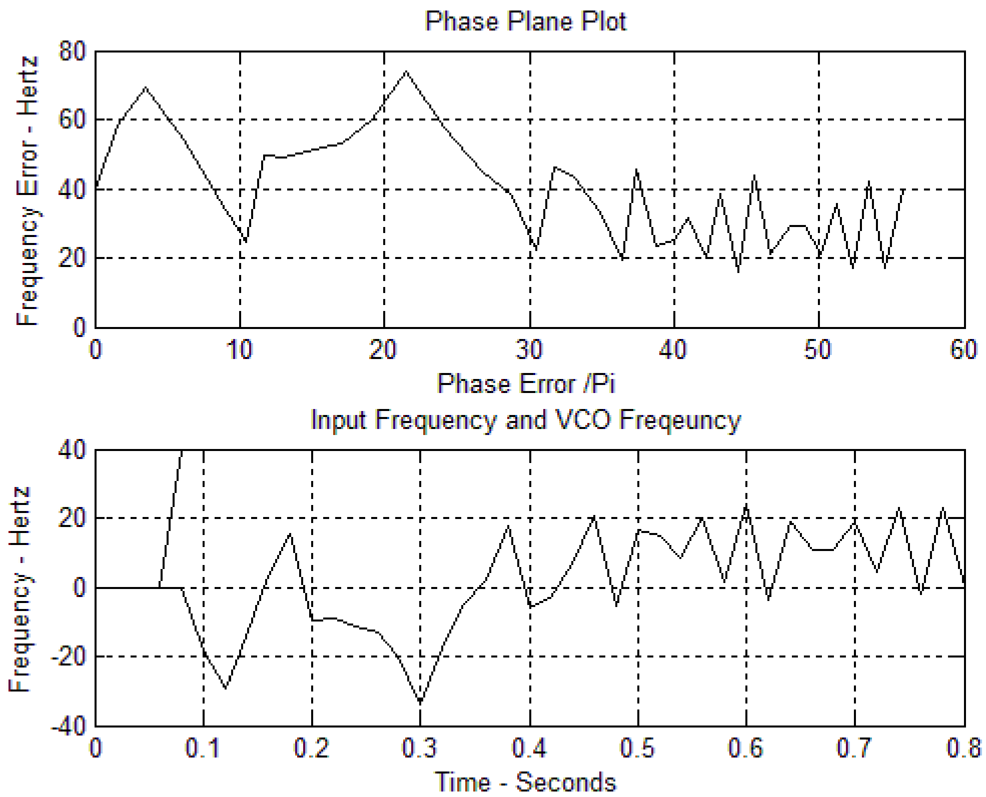
当相对极点偏移$\lambda = 0.2$时，相位差随频率差如下图：



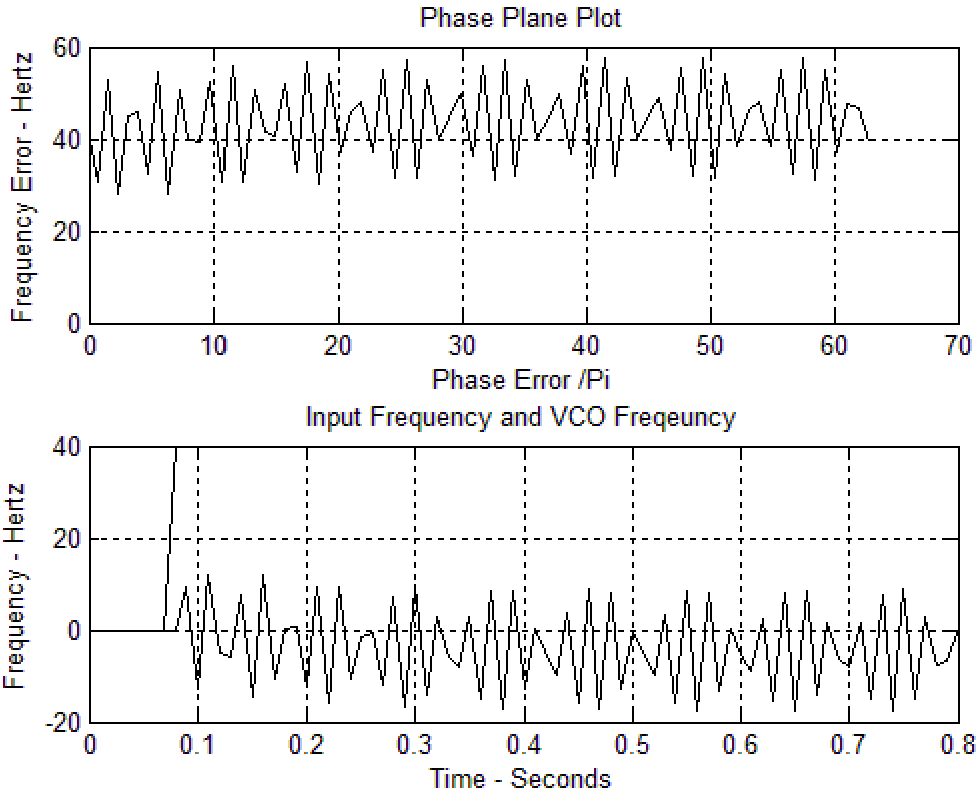
由上面几副图可知：当$\lambda = 0.2$时，不能捕获，即无法达到稳态误差。

6.5

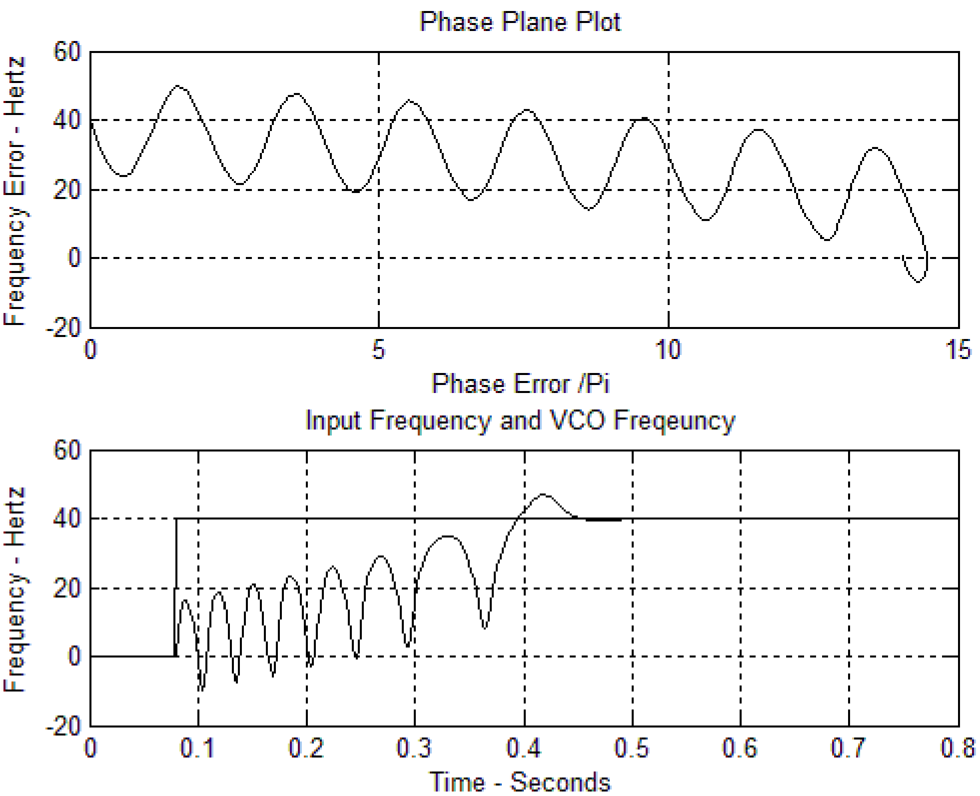
当采样频率为50Hz时：



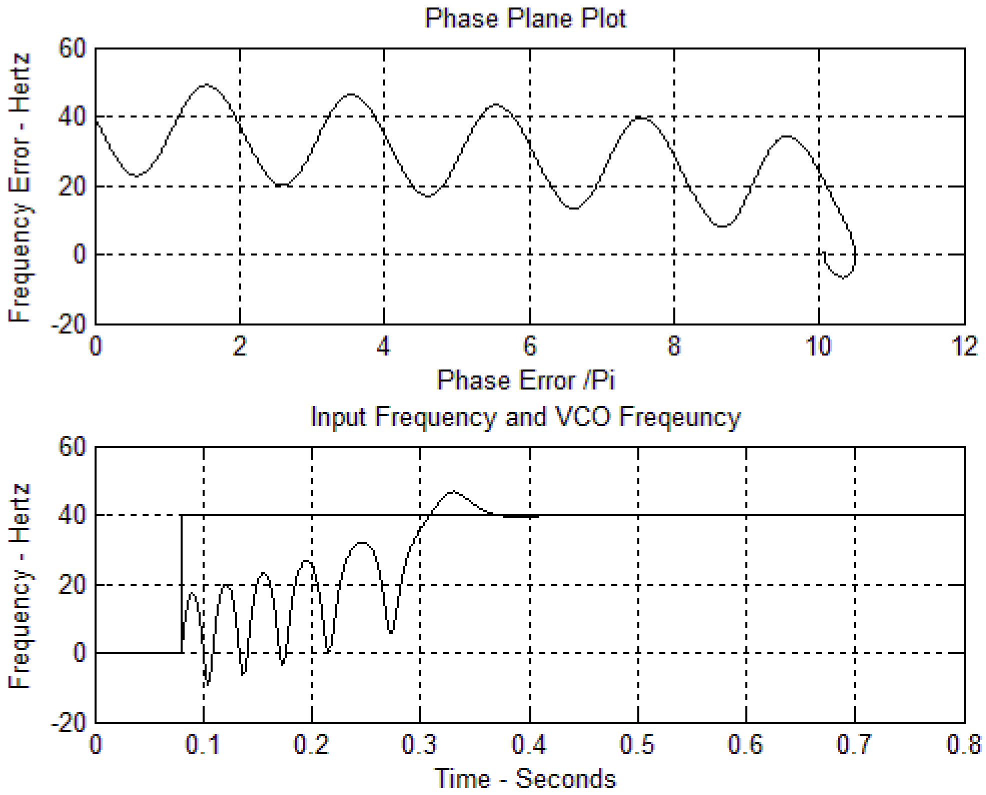
当采样频率为100Hz时：



当采样频率为500Hz时：



当采样频率为10000Hz时：



由上面四幅图可知：采样频率越大，结果越精确，收敛时间越快，但是当相位误差能够趋于0时，采样频率即为合适的。

6-11

答：matlab 程序：

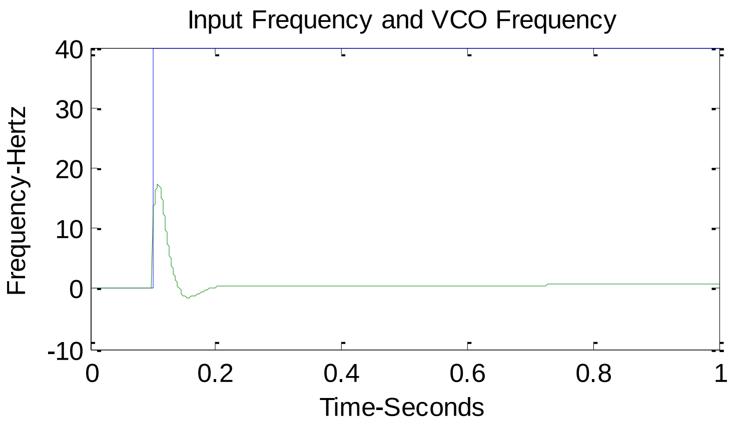
```
clear all
disp(' ')
fdel=input('Enter the size of the frequency step in Hertz>');
fn=input('Enter the loop natural frequency in Hertz>');
lambda=input('Enter lambda,the relative pole offset>');
disp(' ')
disp('Accept default values:')
disp('zeta=1/sqrt(2)')
disp('tstop=0.8')
dtype=input('Enter y for yes or n for no>','s');
if dtype=='y'
    zeta=1/sqrt(2);
    fs=200*fn;
    tstop=1;
else
    zeta=input('Enter zeta,the lop dmping factor>');
    fs=input('Enter the sampling frequency in Hertz>');
    tstop=input('Enter tstop,the simulation runtime>');
end
npts=fs*tstop+1;
t=(0:(npts-1))/fs;
nsettle=fix(npts/10);
tsettle=nsettle/fs;
fin=[zeros(1,nsettle),fdel*ones(1,npts-nsettle)];
phin=[zeros(1,nsettle),2*exp(t(1:(npts-nsettle)))];
```

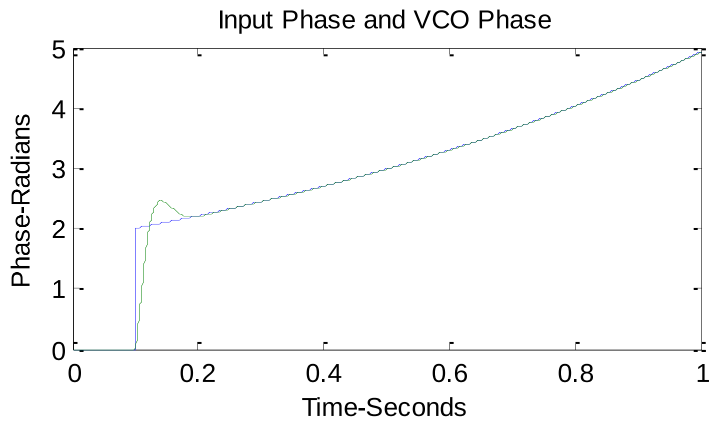
```
disp(' ')

w2b=0;w2c=0;s5=0;phivco=0;    %initialize
twopi=2*pi;
twofs=2*fs;
G=2*pi*fn*(zeta+sqrt(zeta*zeta-lambda));
a=2*pi*fn/(zeta+sqrt(zeta*zeta-lambda));
a1=a*(1-lambda);a2=a*lambda;
%npts=1+fs*tfinal;
phierror=zeros(1,npts);
fvco=zeros(1,npts);
for i=1:npts
        s1=phin(i)-phivco;
        s2=sin(s1);
        s3=G*s2;
        s4=a1*s3;
        s4a=s4-a2*s5;
        w1b=s4a+w2b;
        w2b=s4a+w1b;
        s5=w1b/twofs;
        s6=s3+s5;
        w1c=s6+w2c;
        w2c=s6+w1c;
        phivco=w1c/twofs;
        phierror(i)=s1;
        fvco(i)=s6/twopi;
end
freqerror=fin-fvco;
  kk=0;
while kk==0
        k=menu('Phase Lock Loop Postprocessor',...
                'Input Frequency and VCO Frequency','Input Phase and VCO Phase',...
                'Phase Plane and Time Domain Plots','Exit Program');
        if k==1
                plot(t,fin,t,fvco)
title('Input Frequency and VCO Frequency')
                xlabel('Time-Seconds');ylabel('Frequency-Hertz');
                pause
        elseif k==2
                        pvco=phin-phierror;
                        plot(t,phin,t,pvco)
                        title('Input Phase and VCO Phase')
                        xlabel('Time-Seconds');ylabel('Phase-Radians');
                        pause
        elseif k==3
                plot(t,freqerror);
                title('Frequency Error')
                xlabel('Time-Seconds');ylabel('Frequency Error-Hertz');
```

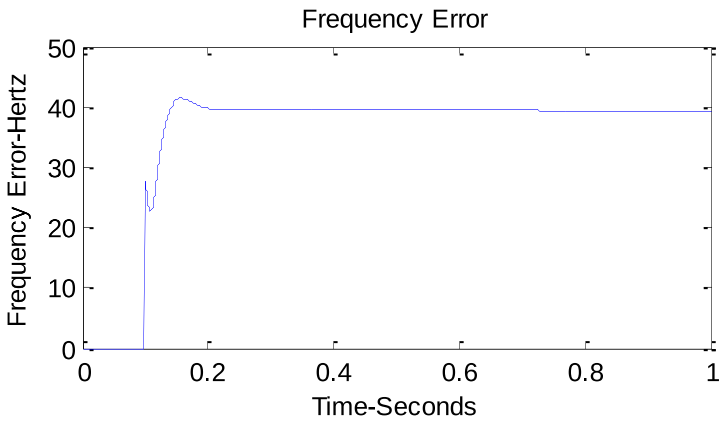
```
            pause
        elseif k==4
            plot(t,phierror);
            title('Phase Error')
            xlabel('Time-Seconds');ylabel('Phase Error-Radians');
            pause
        elseif k==5
            ppplot
        elseif k==6
            subplot(211);
            phierrn=phierr/pi;
            plot(phierrn,freqerror);
            grid;
            title('Phase Plane Plot');
            xlabel('Phase Error/pi');ylabel('Frequency Error-Hertz');
            subplot(212)
            plot(t,fin,t,fvco);grid
            title('Input Frequency and VCO Frequency')
            xlabel('Time-Seconds');ylabel('Frequency-Hertz');
        subplot(111)
        elseif k==7
            kk=1;
        end
end

kz=0;
while kz==0
        k=menu('Phase Plane Options',...
            'Extend Phase Plane',...
            'Phase Plane mod(2pi)',...
            'Exit Phase Plane Menu');
            if k==1
            phierrn=phierror/pi;
            plot(phierrn,freqerror,'k')
            title('Phase Plane Plot')
            xlabel('Phase Error/Pi');ylabel('Frequency Error-Hertz')
            grid
            pause
            elseif k==2
                pplane(phierror,freqerror,nsettle+1);
                pause
            elseif k==3
                kz=1;
            end
end

function pplane=pplane(x,y,nsettle)
ln=length(x);
```

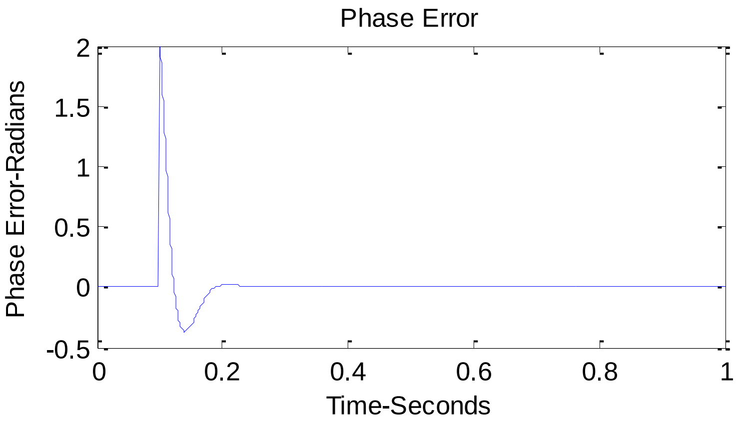
```
maxfreq=max(y);
minfreq=min(y);
close
axis([-1,1,1.1*minfreq 1.1*maxfreq]);
hold on
j=nsettle;
while j<ln
    i=1;
    while x(j)<pi&j<ln
        a(i)=x(j)/pi;
        b(i)=y(j);
        j=j+1;
        i=i+1;
    end
    plot(a,b,'k')
    a=[];
    b=[];
    x=x-2*pi;
end
hold off
title('Phase-Plane Plot')
xlabel('Phase Error/Pi');ylabel('Frequency Error in Hertz')
grid
```

```
Enter the size of the frequency step in Hertz>40
Enter the loop natural frequency in Hertz>10
Enter lambda,the relative pole offset>0.1

Accept default values:
zeta=1/sqrt(2)
tstop=0.8
Enter y for yes or n for no>y
```







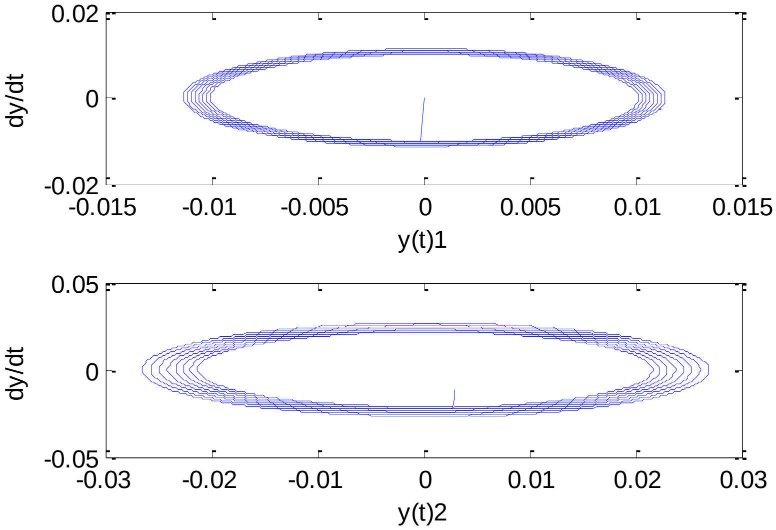


6-14

答：matlab 程序：

```
w2b=0;w2c=0;
yd=0;y=0;
tfinal=50;
fs=100;
delt=1/fs;
npts=1+fs*tfinal;
ydv=zeros(1,npts);
yv=zeros(1,npts);
for i=1:npts
    t=(i-1)*delt;
    ydd1=-y-abs(y)*yd;
w1b=ydd1+w2b;
w2b=ydd1+w1b;
yd=w1b/(2*fs);
```

```
w1c=yd+w2c;
w2c=yd+w1c;
y=w1c/(2*fs);
ydv(1,i)=yd;
yv(1,i)=y;
        if t==0
                y=1;yd=0;
                ydd1=0;ydd2=0;
        end
end
subplot(2,1,1);
plot(yv,ydv)
xlabel('y(t)1')
ylabel('dy/dt')
for i=1:npts
        t=(i-1)*delt;
        ydd2=-y-y*abs(yd);
w1b=ydd2+w2b;
w2b=ydd2+w1b;
yd=w1b/(2*fs);
w1c=yd+w2c;
w2c=yd+w1c;
y=w1c/(2*fs);
ydv(1,i)=yd;
yv(1,i)=y;
        if t==0
                y=1;yd=0;
                ydd1=0;ydd2=0;
        end
end
subplot(2,1,2);
plot(yv,ydv)
axis([-0.03 0.03 -0.05 0.05])
xlabel('y(t)2')
ylabel('dy/dt')
```



第 7 章 随机信号的产生与处理

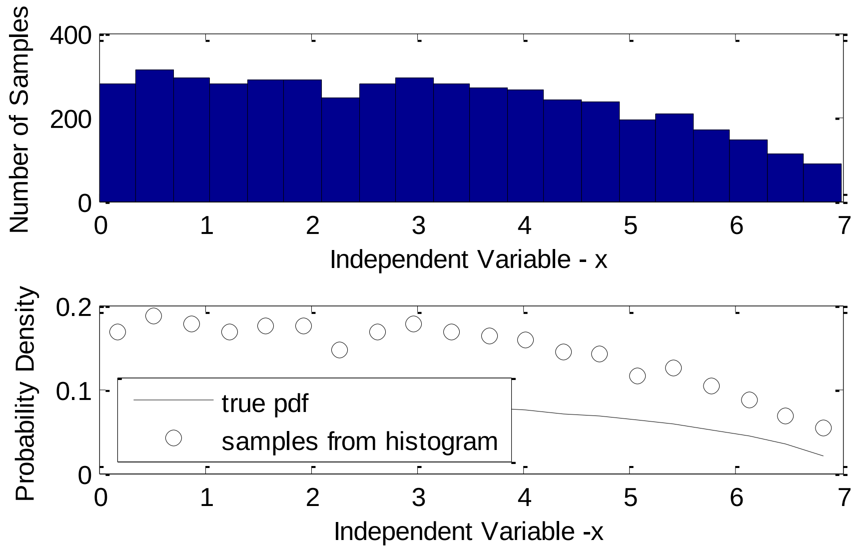
7-19

答：matlab 程序：

```
R=7;M=2/pi;
N=input('Input number of points N > ');
fx=zeros(1,N);
u1=rand(1,N);u2=rand(1,N);
v1=R*u1;
v2=(M/R)*rand(1,N);
kpts=0;
for k=1:N
if v2(k)<(M/(R*R))*sqrt(R*R-v1(k)*v1(k));
       kpts=kpts+1;
       fx(kpts)=v1(k);
       end
end
fx=fx(1:kpts);
[N_samp,x]=hist(fx,20);
subplot(2,1,1)
bar(x,N_samp,1)
ylabel('Number of Samples')
xlabel('Independent Variable - x')
yt=(M/R/R)*sqrt(R*R-x.*x);
del_x=x(3)-x(2);
p_hist=N_samp/(kpts*del_x);
subplot(2,1,2)
plot(x,yt,'k',x,p_hist,'ok')
ylabel('Probability Density');
xlabel('Independent Variable -x');
legend('true pdf','samples from histogram',3);
text=['The number of points accepted is',...
       num2str(kpts,15),'and N is',num2str(N,15),'.'];
disp(text);
```

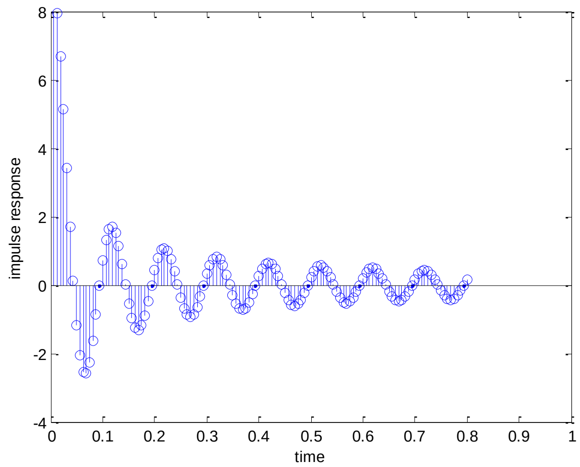
输入数据：　Input number of points N > 6000

The number of points accepted is4753and N is6000.



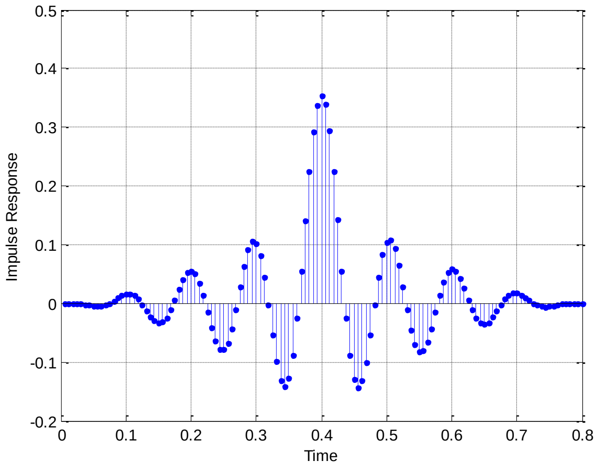
7.20

```
fd=10;
fs=16*fd;ts=1/fs;
time=1*ts:ts:128*ts;
htt=(pi*fd*time).^(-1/4)/gamma(3/4)*fd.*Bessel(1/4,2*pi*fd*time);
stem(time,htt)
xlabel('time');
ylabel('impulse response');
axis([0 1 -4 8])
```




```
fd = 10;
impw = jakes_filter(fd);
fs = 16*fd; ts = 1/fs;
time = [1*ts:ts:128*ts];
 subplot(1,1,1)
stem(time,impw,'.'); grid
xlabel('Time'); ylabel('Impulse Response')

 [h f] = linear_fft(impw,128,ts);

x = randn(1,1024);
y = filter(impw,1,x);
```

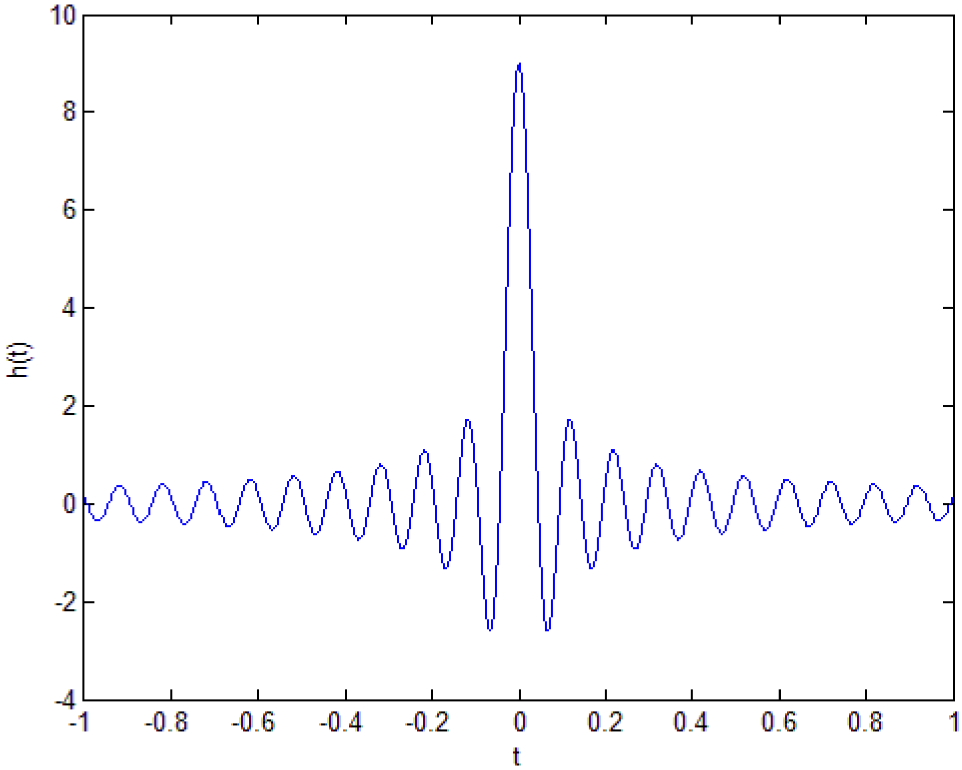
```
[output_psd ff] = log_psd(y,1024,ts);
figure;
subplot(2,1,1)
plot(ff,output_psd); grid;
axis([-500 500 -50 0])
xlabel('Frequency'); ylabel('PSD')
z = randn(1,1024)+i*randn(1,1024);
zz = filter(impw,1,z);
time = (0.0:ts:1024*ts);
zz = zz/max(max(abs(zz)));
subplot(2,1,2)
plot(time(161:480),10*log10(abs(zz(161:480)))); grid
axis([0.1 0.3 -20 0])
xlabel('Time'); ylabel('Log Amplitude')
```



7.20

由matlab程序可以得到h(t)的图：

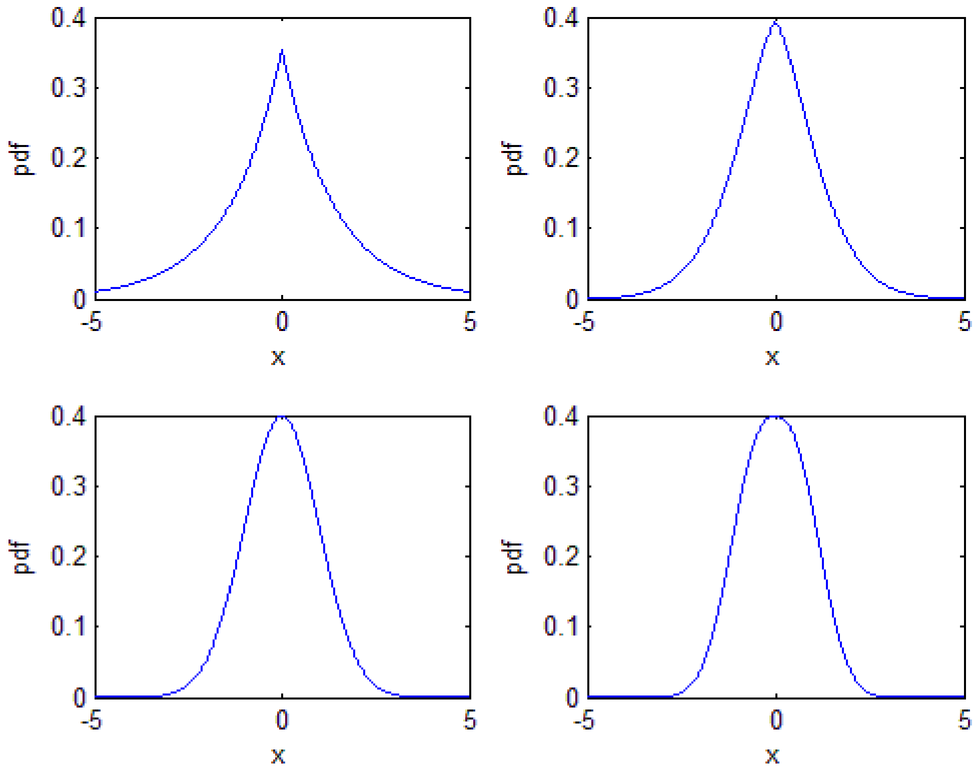
```
%c7_20 impulse
t=-1:0.0001:1;
h=(pi*10*t).^(-1/4)./gamma(3/4)*10.*besselj(1/4,20*pi*t);
plot(t,h);
xlabel('t');
ylabel('h(t)');
```



7.21

(a)当 $f_X(x)$，$v=1,1,5,2,25$时，$f_X(x)$的图形如下：

```
clear all
v=[1 1.5 2 2.5];
for i=1:4
    subplot(2,2,i)
    x=-5:0.0001:5;
    f=(v(i)/(sqrt(8)*gamma(1/v(i))))*exp(-(abs(x/sqrt(2))).^v(i));
    plot(x,f);
end
```



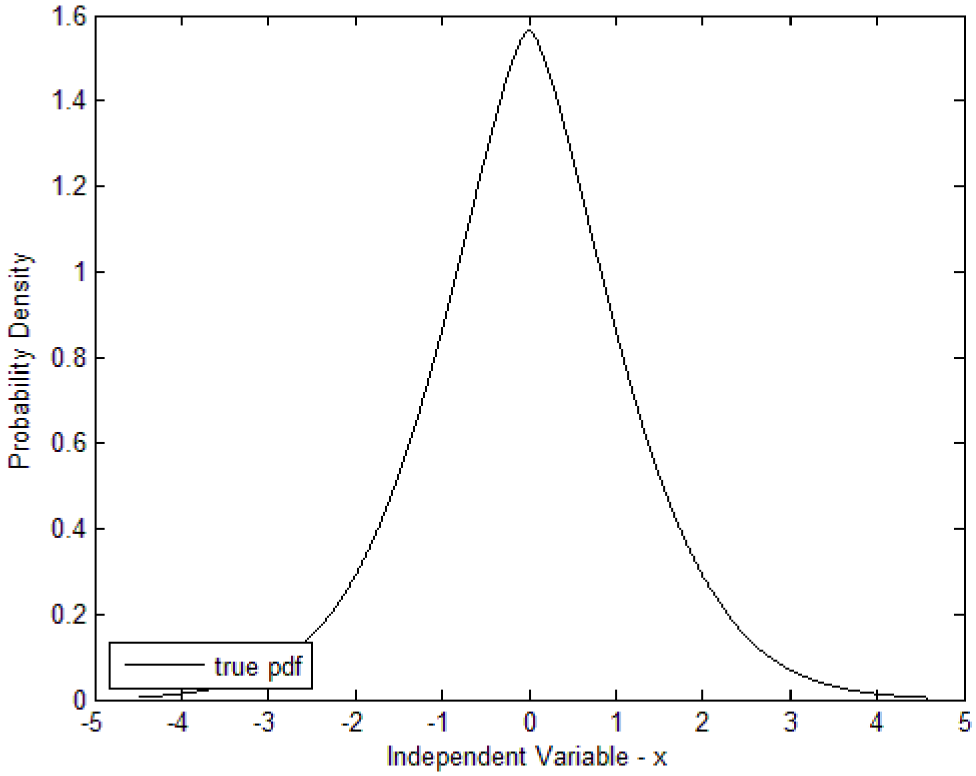
(b)

由上面的几副图可得： $M = 0.4*(5-(-5)) = 4$

```
clear all
a = 5;                                          % default value of R
M = 4;                                      % value of M
N = input('Input number of points N > ');    % set N
fx = zeros(1,N);                                % array of output samples
u1 = rand(1,N); u2 = rand(1,N);                 % generate u1 and u2
v1 = 2*a*u1-5;                                      % generate v1
v2 = M/10*rand(1,N);
kpts = 0;                                       % initialize counter
for k=1:N
   if v2(k)<M*(1.5/(sqrt(8)*gamma(1/1.5)))*exp(-(abs(v1(k)/sqrt(2))).^1.5);
      kpts=kpts+1;                              % increment counter
      fx(kpts)=v1(k);                           % save output sample
   end
end
fx = fx(1:kpts);
[N_samp,x] = hist(fx,2000);                     % get histogram parameters
yt =M*(1.5/(sqrt(8)*gamma(1/1.5)))*exp(-(abs(x/sqrt(2))).^1.5);          % calculate pdf
plot(x,yt,'k')%,x,p_hist,'ok')                  % compare
ylabel('Probability Density')
xlabel('Independent Variable - x')
legend('true pdf','samples from histogram',3)
text = ['The number of points accepted is ',...
      num2str(kpts,15),' and N is ',num2str(N,15),'.'];
```

```
disp(text)
% End of script file.
```
运行该程序：

```
Input number of points N > 2000
The number of points accepted is 993 and N is 2000.
```



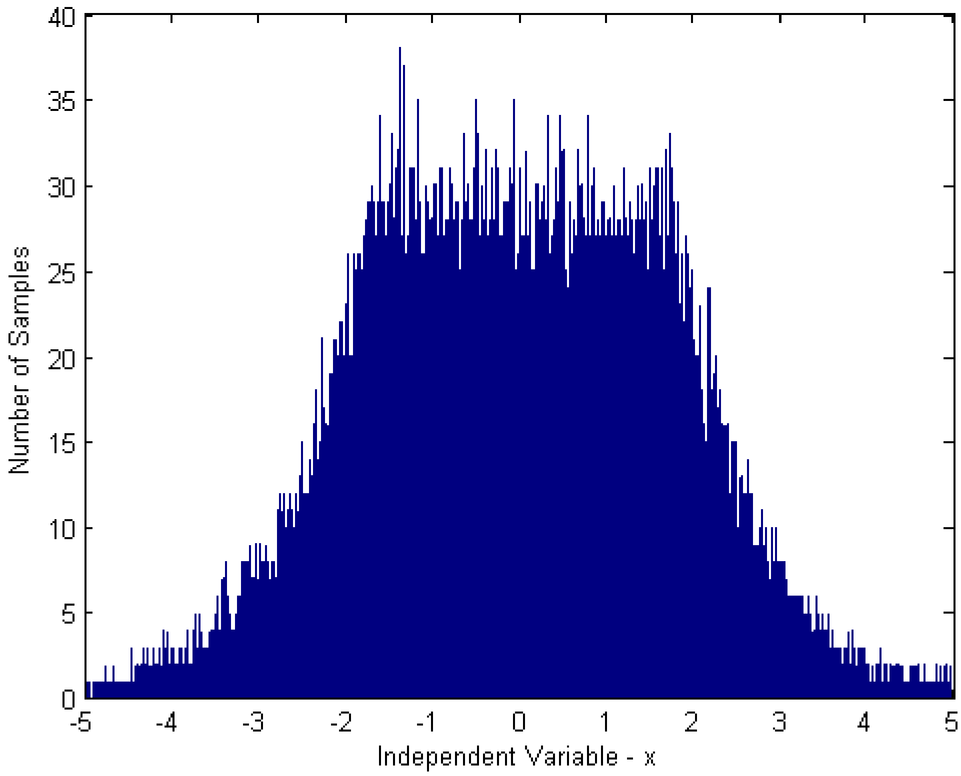
(c)

```
clear all
a = 5;                                        % default value of R
M = 4;                                    % value of M
N = input('Input number of points N > ');    % set N
fx = zeros(1,N);                              % array of output samples
u1 = rand(1,N); u2 = rand(1,N);               % generate u1 and u2
v1 = 2*a*u1-5;                                    % generate v1
v2 = M/10*rand(1,N);
kpts = 0;                                     % initialize counter
for k=1:N
   if v2(k)<M*(1.5/(sqrt(8)*gamma(1/1.5)))*exp(-(abs(v1(k)/sqrt(2))).^1.5);
      kpts=kpts+1;                            % increment counter
      fx(kpts)=v1(k);                         % save output sample
   end
end
fx = fx(1:kpts);
[N_samp,x] = hist(fx,10000);                      % get histogram parameters
bar(x,N_samp,1)                               % plot histogram
ylabel('Number of Samples')
xlabel('Independent Variable - x')
text = ['The number of points accepted is ',...
```

```
    num2str(kpts,15),' and N is ',num2str(N,15),'.'];
disp(text)
% End of script file.
```

运行程序：

```
Input number of points N > 200000
The number of points accepted is 98788 and N is 200000.
```



7-23

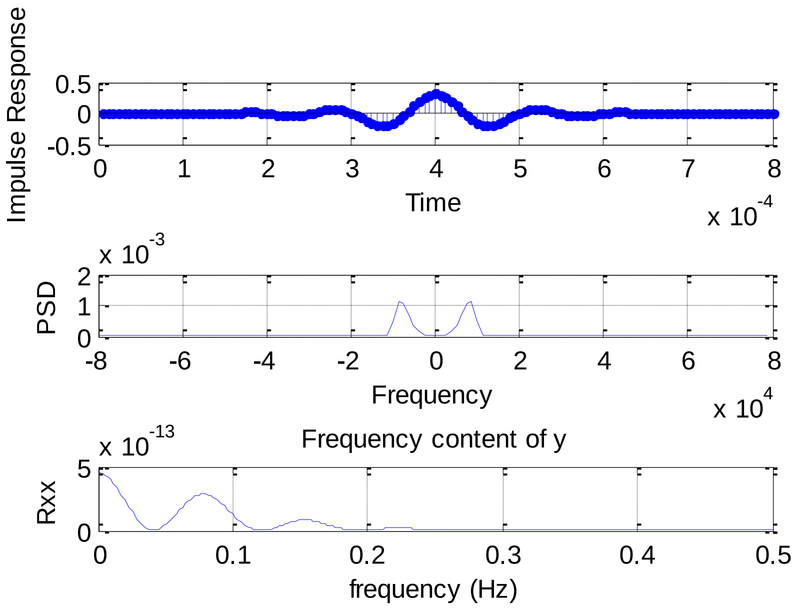
答：matlab 程序：

```
% FIR implementation of the Jakes filter (128 points)
n = 512; nn = 2*n;
fd=10000;
fs = 0:fd/64:fd;
H = zeros(1,n);
for k=1:(n/8+1)
    jpsd(k)=(1-cos(pi*fs(k)/fd))^2;
  H(k)=jpsd(k)^0.5;
end;
for k=1:n
    H(n+k) = H(n+1-k);
end
[inv,time] = linear_fft(H,nn,fd/64);
imp = real(inv(450:577));
impw = imp.*hanning(128)';
energy = sum(impw.^2)
impw = impw/(energy^0.5);
% End of function file.
ts = 1/(16*fd);
time = [1*ts:ts:128*ts];
```

```
subplot(3,1,1)
stem(time,impw,'.'); grid
xlabel('Time'); ylabel('Impulse Response')
% Square the fft and check the power transfer function.
[h f] = linear_fft(impw,128,ts);
subplot(3,1,2)
plot(f,abs(h.*h)); grid;
xlabel('Frequency'); ylabel('PSD')
%get the Rxx fuction
Y=ifft(abs(h.*h),512);
Pyy=Y.*conj(Y)/512;
f =(0:256)/512;
subplot(3,1,3)
plot(f,Pyy(1:257));grid;
title('Frequency content of y')
xlabel('frequency (Hz)'); ylabel('Rxx');
% End of file.
```

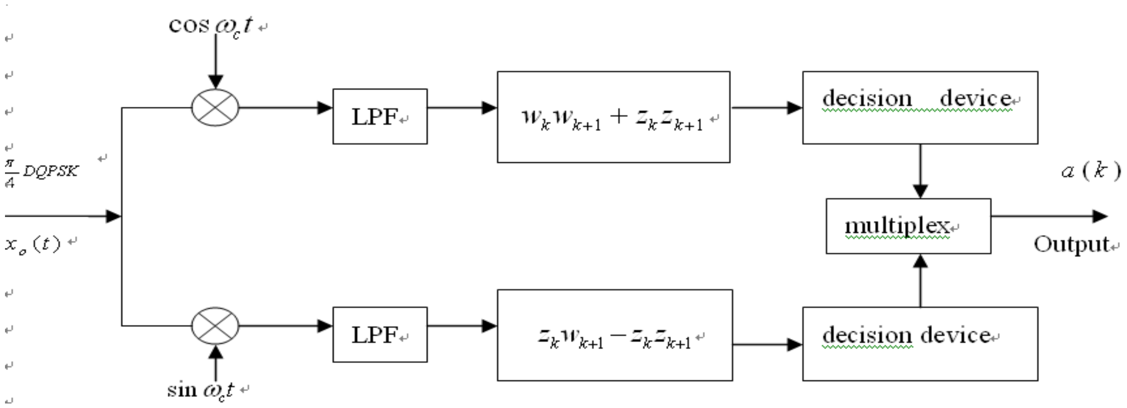
其中， File: linear_fft.m 如下：

```
function [fftx,freq] = linear_fft(x,n,ts)
y = zeros(1,n);
for k=1:n
  freq(k) =(k-1-(n/2))/(n*ts);
     y(k) = x(k)*((-1.0)^(k+1));
end;
fftx = fft(y)/n;
% End of function file.
```



第 8 章 后处理

8-1



答：

matlab 程序：

```
m = 200;      bits = 2*m;
sps = 10;
iphase = 0;
order = 5;
bw = 0.2;
data = zeros(1,bits); d = zeros(1,m); q = zeros(1,m);
dd = zeros(1,m); qq = zeros(1,m); theta = zeros(1,m);
thetaout = zeros(1,sps*m);
data = round(rand(1,bits));
dd = data(1:2:bits-1);
qq = data(2:2:bits);
theta(1) = iphase;
thetaout(1:sps) = theta(1)*ones(1,sps);
for k=2:m
    if dd(k) == 1
        phi_k = (2*qq(k)-1)*pi/4;
    else
        phi_k = (2*qq(k)-1)*3*pi/4;
    end
    theta(k) = phi_k + theta(k-1);
    for i=1:sps
        j = (k-1)*sps+i;
        thetaout(j) = theta(k);
    end
end
d = cos(thetaout);
q = sin(thetaout);
[b,a] = butter(order,bw);
df = filter(b,a,d);
qf = filter(b,a,q);
kk = 0;
while kk == 0
k = menu('pi/4 QPSK Plot Options',...
          'Unfiltered pi/4 QPSK Signal Constellation',...
          'Unfiltered pi/4 QPSK Eye Diagram',...
          'Filtered pi/4 QPSK Signal Constellation',...
          'Filtered pi/4 OQPSK Eye Diagram',...
          'Unfiltered Direct and Quadrature Signals',...
```

```
        'Filtered Direct and Quadrature Signals',...
        'Exit Program');
        if k == 1
            sigcon(d,q)
            pause
        elseif k ==2
            dqeye(d,q,4*sps)
            pause
        elseif k == 3
            sigcon(df,qf)
            pause
        elseif k == 4
            dqeye(df,qf,4*sps)
            pause
        elseif k == 5
            numbsym = 10;
            dt = d(1:numbsym*sps);
            qt = q(1:numbsym*sps);                    dqplot(dt,qt)
            pause
        elseif k == 6
            numbsym = 10;
            dft=df(1:numbsym*sps);
            qft=qf(1:numbsym*sps);
            dqplot(dft,qft)
            pause
        elseif k == 7
            kk = 1;
        end
    end
```

其中，子程序1：

```
function []=sigcon(x,y)
plot(x,y)
axis('square')
axis('equal')
xlabel('Direct Channel')
ylabel('Quadrature Channel')
% End of function file.
```

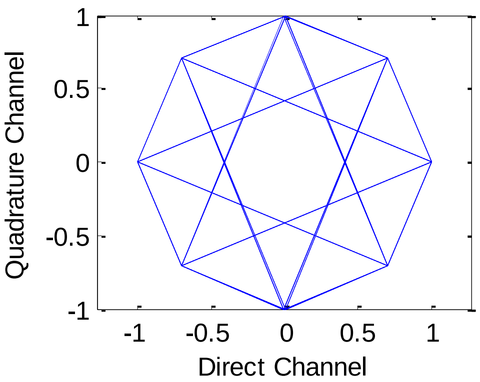
子程序2：

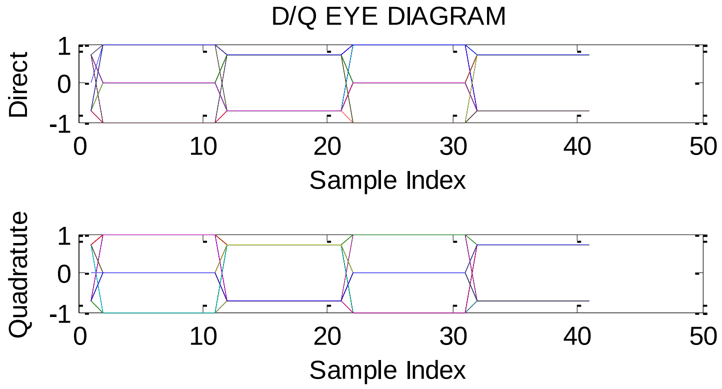
```
function [] = dqeye(xd,xq,m)
lx = length(xd);
kcol = floor(lx/m);
xda = [0,xd]; xqa = [0,xq];
for j = 1:kcol
for i = 1:(m+1)
        kk = (j-1)*m+i;
        y1(i,j) = xda(kk);
        y2(i,j) = xqa(kk);
    end
end
subplot(211)
```

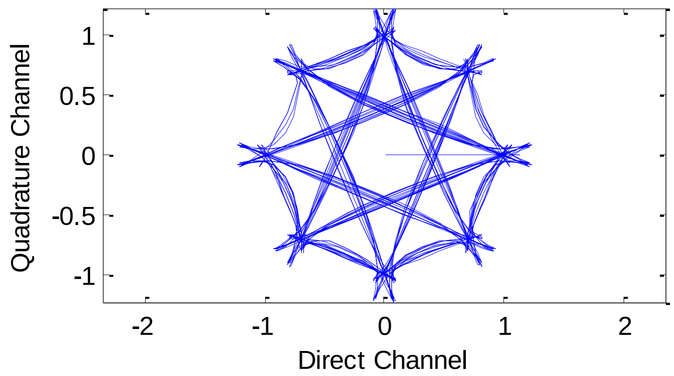
```
plot(y1);
title('D/Q EYE DIAGRAM');
xlabel('Sample Index');
ylabel('Direct');
subplot(212)
plot(y2);
xlabel('Sample Index');
ylabel('Quadratute');
subplot(111)
% End of function file.
```

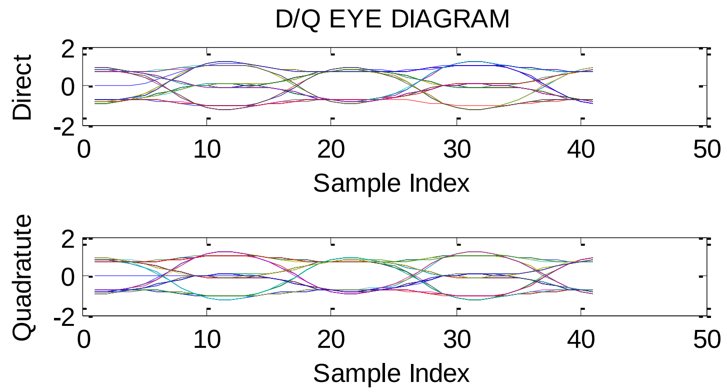
子程序3：

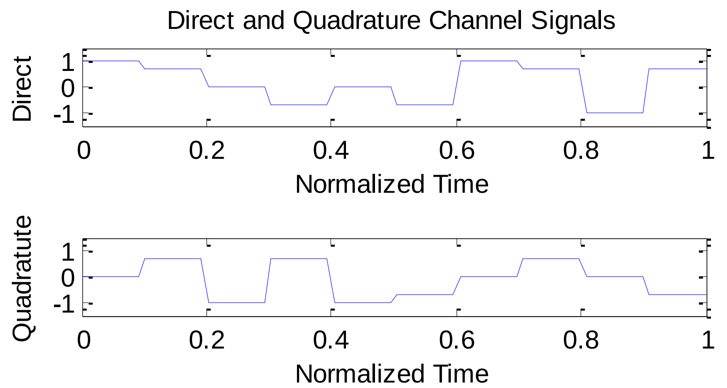
```
function [] = dqplot(xd,xq)
lx = length(xd);
t = 0:lx-1;
nt = t/(lx-1);
nxd = xd(1,1:lx);
nxq = xq(1,1:lx);
subplot(211)
plot(nt,nxd);
a = axis;
axis([a(1) a(2) 1.5*a(3) 1.5*a(4)]);
title('Direct and Quadrature Channel Signals');
xlabel('Normalized Time');
ylabel('Direct');
subplot(212)
plot(nt,nxq);
a = axis;
axis([a(1) a(2) 1.5*a(3) 1.5*a(4)]);
xlabel('Normalized Time');
ylabel('Quadratute');
subplot(111)
% End of function file.
```

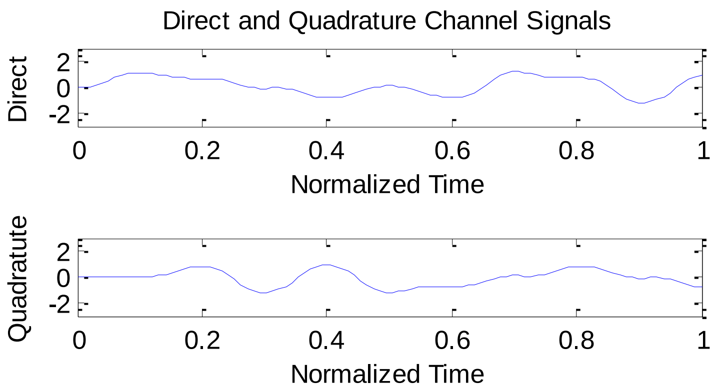



D/Q EYE DIAGRAM
Direct
Sample Index
Quadrature
Sample Index


Quadrature Channel
Direct Channel


D/Q EYE DIAGRAM
Direct
Sample Index
Quadrature
Sample Index

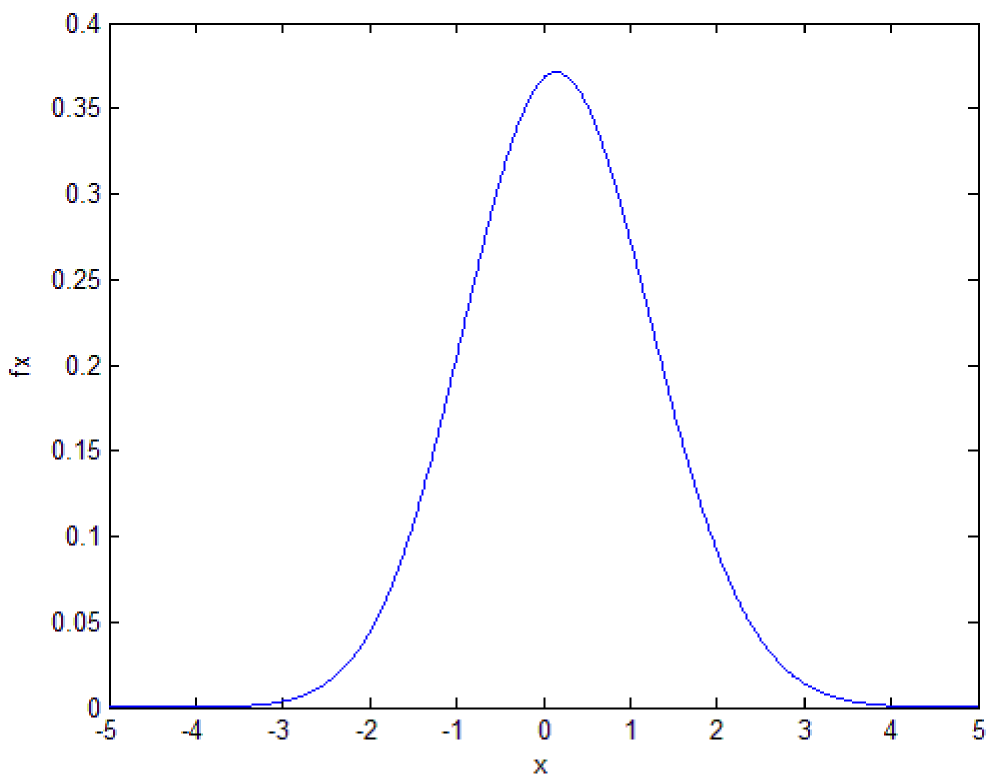

Direct and Quadrature Channel Signals
Direct
Normalized Time
Quadrature
Normalized Time



8.5

当 $a=0.8, m_1=0, m_2=1, \sigma_1=\sigma_2=1$ 时，程序如下：

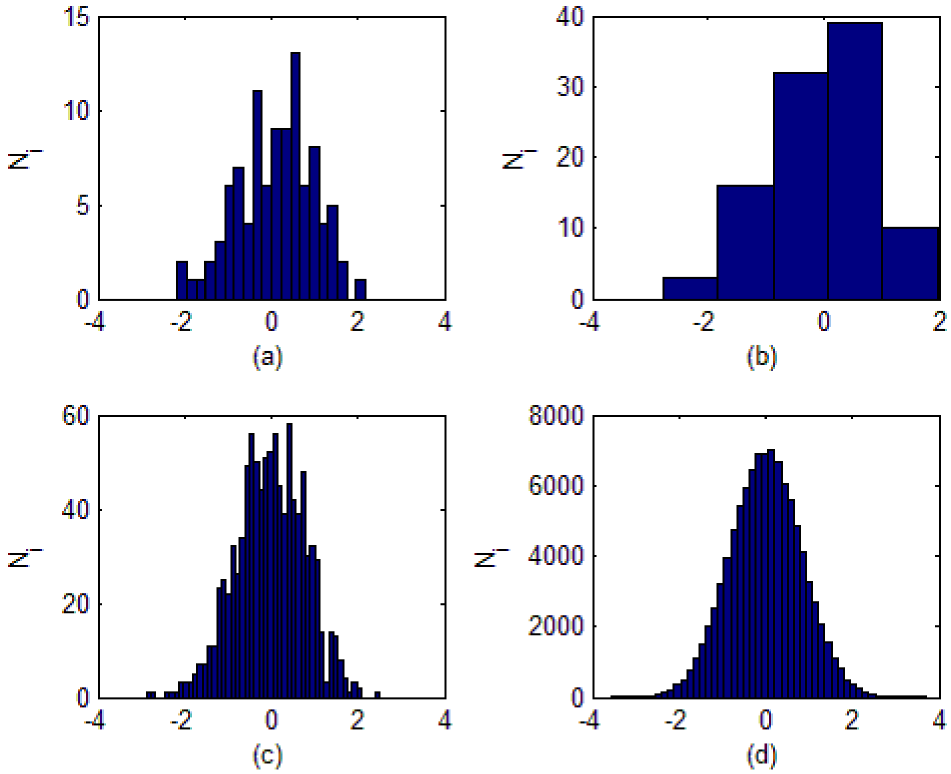
```
clear all
x=-5:0.01:5;
a=0.8;m1=0;m2=1;segma1=1;segma2=1;
f=a/sqrt(2*pi)/segma1*exp(-((x-m1).^2/(2*segma1.^2)))+(1-a)/sqrt(2*pi)/segma2*exp(-((x-m2).^2/(2*segma
2.^2)));
plot(x,f);
```



当 $a=0.8, m_1=0, m_2=0, \sigma_1=\sigma_2=1$ 时，用高斯混合的pdf，重做例8-2：

```
clear all
a=0.8;
subplot(2,2,1)
x = a*randn(1,100)+(1-a)*randn(1,100); hist(x,20)
```

```
ylabel('N_i'); xlabel('(a)')
subplot(2,2,2)
x = a*randn(1,100)+(1-a)*randn(1,100); hist(x,5)
ylabel('N_i'); xlabel('(b)')
subplot(2,2,3)
x = a*randn(1,1000)+(1-a)*randn(1,1000); hist(x,50)
ylabel('N_i'); xlabel('(c)')
subplot(2,2,4)
x = a*randn(1,100000)+(1-a)*randn(1,100000); hist(x,50)
ylabel('N_i'); xlabel('(d)')
```



8.11

求 $y_1(n)$ 利用函数：

```
% File: snrmse.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
%
function [gain,delay,px,py,rxy,rho,snrdb] = snrmse(x,y)
ln = length(x);                 % Set length of the reference (x) vector
fx = fft(x,ln);                 % FFT the reference (x) vector
fy = fft(y,ln);                 % FFT the measurement (y) vector
fxconj = conj(fx);              % Conjugate the FFT of the reference vector
sxy = fy .* fxconj;             % Determine the cross PSD
rxy = ifft(sxy,ln);             % Determine the cross correlation function
```

```
rxy = real(rxy)/ln;              % Take the real part and scale
px = x*x'/ln;                    % Determine power in reference vector
py = y*y'/ln;                    % Determine power in measurement vector
[rxymax,j] = max(rxy);           % Find the max of the crosscorrelation
gain = rxymax/px;                % Here's the gain
delay = j-1;                     % Here's the delay
rxy2 = rxymax*rxymax;            % Square rxymax for later use
rho = rxymax/sqrt(px*py);        % Here's the correlation coefficient
snr = rxy2/(px*py-rxy2);         % Here's the snr
snrdb = 10*log10(snr);           % Here's the snr in db
% End of script file.
```

再自己编写程序：

```
kpts = 1024;                         % FFT Block size
k = 1:kpts;                          % sample index vector
fd = 2;                              % desired signal frequency
Ax = 1; Ayd = 5;         % amplitudes
theta = 2*pi*k/kpts;                 % phase vector
x = Ax*sin(fd*theta);                % desired signal
yd = Ayd*sin(fd*theta);     % desired signal at receiver input
yy = yd;            % receiver input
[gain,delay,px,py,rxy,rho,snrdb] = snrmse(x,yy);
cpx = ['The value of Px is ',num2str(px),'.'];
cpy = ['The value of Py is ',num2str(py),'.'];
cgain = ['The value  gain is ',num2str(gain),'.'];
cdel = ['The value of delay is ',num2str(delay),'.'];
csnrdb = ['The value of SNR is ',num2str(snrdb),' dB.'];
disp(' ')                            % insert blank line
disp(cpx)
disp(cpy)
disp(cgain)
disp(cdel)
disp(csnrdb)
% End of script file.
```

运行结果：

```
The value of Px is 0.5.
The value of Py is 12.5.
The value  gain is 5.
The value of delay is 0.
The value of SNR is 155.4635+13.64376i dB.
```

同理，求 $y_2(n)$ 自己再编写程序：

```
kpts = 1024;                         % FFT Block size
k = 1:kpts;                          % sample index vector
fd = 2;                              % desired signal frequency
Ax = 1; Ayd = 5;% Ayi =4;            % amplitudes
phase = pi/2-pi/512;                         % phase shift
theta = 2*pi*k/kpts;                 % phase vector
x = Ax*sin(fd*theta);                % desired signal
```

```
yd = Ayd*sin(fd*theta-phase);        % desired signal at receiver input
yy = yd;                 % receiver input
[gain,delay,px,py,rxy,rho,snrdb] = snrmse(x,yy);
cpx = ['The value of Px is ',num2str(px),'.'];
cpy = ['The value of Py is ',num2str(py),'.'];
cgain = ['The value  gain is ',num2str(gain),'.'];
cdel = ['The value of delay is ',num2str(delay),'.'];
csnrdb = ['The value of SNR is ',num2str(snrdb),' dB.'];
disp(' ')                            % insert blank line
disp(cpx)
disp(cpy)
disp(cgain)
disp(cdel)
disp(csnrdb)
% End of script file.
```

运行结果：

```
The value of Px is 0.5.
The value of Py is 12.5.
The value  gain is 4.9999.
The value of delay is 127.
The value of SNR is 44.2423 dB.
```

8-12

答：

（a） %M 文件：snrmse(x,y)

```
function [gain,delay,px,py,rxy,rho,snrdb] = snrmse(x,y)
ln = length(x);
fx = fft(x,ln);
fy = fft(y,ln);
fxconj = conj(fx);
sxy = fy .* fxconj;
rxy = ifft(sxy,ln);
rxy = real(rxy)/ln;
px = x*x'/ln;
py = y*y'/ln;
[rxymax,j] = max(rxy);
gain = rxymax/px;
delay = j-1;
rxy2 = rxymax*rxymax;
rho = rxymax/sqrt(px*py);
snr = rxy2/(px*py-rxy2);
snrdb = 10*log10(snr);
% End of script file.
```

%设置频率和样本数

```
kpts = 1024;
k = 1:kpts;
fd = 5;
% desired signal frequency
fi =10;
```

```
Ax = 2; Ayd = 5; Ayi =5;
phase = pi/2;
% phase shift
nstd = 5;
theta = 2*pi*k/kpts;
% phase vector
x = Ax*sin(fd*theta);
yd = Ayd*sin(fd*theta-pi/2);
yi = Ayi*sin(fi*theta-3*pi/4);
noise = nstd*sin(15*theta-pi);
yy = yd+yi+noise;
[gain,delay,px,py,rxy,rho,snrdb] = snrmse(x,yy);
cpx = ['The value of Px is ',num2str(px),'.'];
cpy = ['The value of Py is ',num2str(py),'.'];
cgain = ['The value    gain is ',num2str(gain),'.'];
cdel = ['The value of delay is ',num2str(delay),'.'];
csnrdb = ['The value of SNR is ',num2str(snrdb),' dB.'];
disp(' ')
disp(cpx)
disp(cpy)
disp(cgain)
disp(cdel)
disp(csnrdb)
% End of script file.
```

结果： The value of Px is 2.

The value of Py is 37.5.

The value   gain is 2.5.

The value of delay is 256.

The value of SNR is -3.0103 dB.

（b） 从（a）中的显示结果可以看到，增益，延迟，信噪比等数据，得到这些差错源并不是很严重的。

（c） 系统幅度失真，相位失真。

（d） 参数 A=B=1，a=b=0 时，系统无失真，下面对这些参数进行实验,并通过和真实值比较证明结论正确。

```
kpts = 1024;
k = 1:kpts;
fd = 5;
fi =10;
Ax = 2; Ayd = 5; Ayi =1;
phase = pi/2;
nstd = 1;
theta = 2*pi*k/kpts;
x = Ax*sin(fd*theta);
  yd = Ayd*sin(fd*theta-pi/2);
yi = Ayi*sin(fi*theta);
noise = nstd*sin(15*theta);
yy = yd+yi+noise;
  [gain,delay,px,py,rxy,rho,snrdb] = snrmse(x,yy);
cpx = ['The value of Px is ',num2str(px),'.'];
cpy = ['The value of Py is ',num2str(py),'.'];
```

```
cgain = ['The value    gain is',num2str(gain),'.'];
    cdel = ['The value of delay is ',num2str(delay),'.'];
    csnrdb = ['The value of SNR is ',num2str(snrdb),' dB.'];
disp(' ')
disp(cpx)
disp(cpy)
disp(cgain)
disp(cdel)
disp(csnrdb)
% End of script file.
```

结果： The value of Px is 2.

The value of Py is 13.5.

The value   gain is 2.5.

The value of delay is 256.

The value of SNR is 10.9691 dB

第 9 章 蒙特卡罗方法导论

9.10

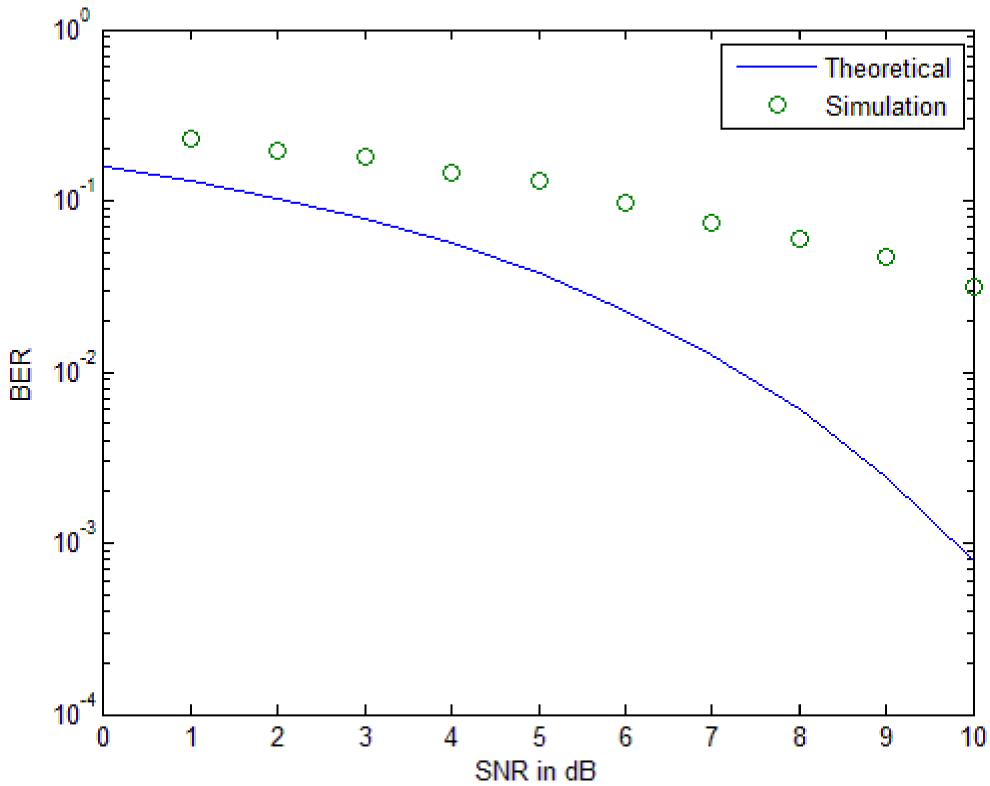
先利用q函数：q.m

```
function y=q(x)
y = 0.5*erfc(x/sqrt(2));
```

再编写程序：

```
clear all
snrdB_min = 0; snrdB_max = 10;          % SNR (in dB)limits
snrdB = snrdB_min:1:snrdB_max;
Nsymbols = input('Enter number of symbols > ');
snr = 10.^(snrdB/10);                    % convert from dB
h = waitbar(0,'SNR Iteration');
len_snr = length(snrdB);
for j=1:len_snr                          % increment SNR
   waitbar(j/len_snr)
   sigma = sqrt(1/(2*snr(j)));           % noise standard deviation
   error_count = 0;
   for k=1:Nsymbols                      % simulation loop begins
      d = round(rand(1));                % data
      if d ==0
         x_d = sqrt(3)/2;                        % direct transmitter output
        x_q = 0;                         % quadrature transmitter output
      else
       x_d = 0;                      % direct transmitter output
         x_q = 1/2;                      % quadrature transmitter output
      end
      n_d = sigma*randn(1);              % direct noise component
      n_q = sigma*randn(1);              % quadrature noise component
      y_d = x_d + n_d;                   % direct receiver input
      y_q = x_q + n_q;                   % quadrature receiver input
      if y_d > y_q                       % test condition
         d_est = 0;                      % conditional data estimate
```

```
        else
            d_est = 1;                      % conditional data estimate
        end
        if (d_est ~= d)
            error_count = error_count + 1; % error counter
        end
    end                                     % simulation loop ends
    errors(j) = error_count;                % store error count for plot
end
close(h)
ber_sim = errors/Nsymbols;                  % BER estimate
ber_theor = q(sqrt(snr));                   % theoretical BER
semilogy(snrdB,ber_theor,snrdB,ber_sim,'o')
axis([snrdB_min snrdB_max 0.0001 1])
xlabel('SNR in dB')
ylabel('BER')
legend('Theoretical','Simulation')
```



9-13

答：matlab 程序：

```
iter=input('the parameter N iter:=');
high=4;low=1.5;
x=1.5+2.5*rand(iter,1);
y=2*rand(iter,1);
s=2.5*2*sum(4*exp(-x(:,1)/2)-y(:,1)>=0)/iter;
s1=quad('4*exp(-x/2)',1.5,4);
disp(s);
disp(s1);
```

```
the parameter N iter:=100
     2.7500

     2.6963
the parameter N iter:=500
     2.6900

     2.6963
the parameter N iter:=1000
     2.6450

     2.6963
```

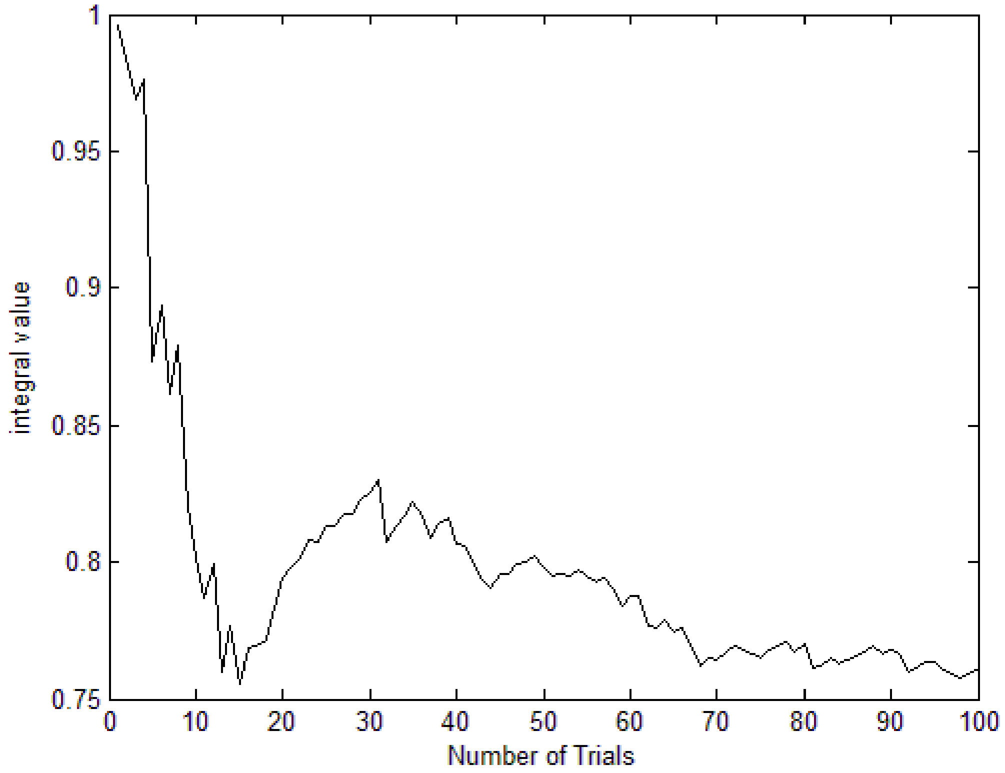
其中，s: 蒙特卡罗积分值

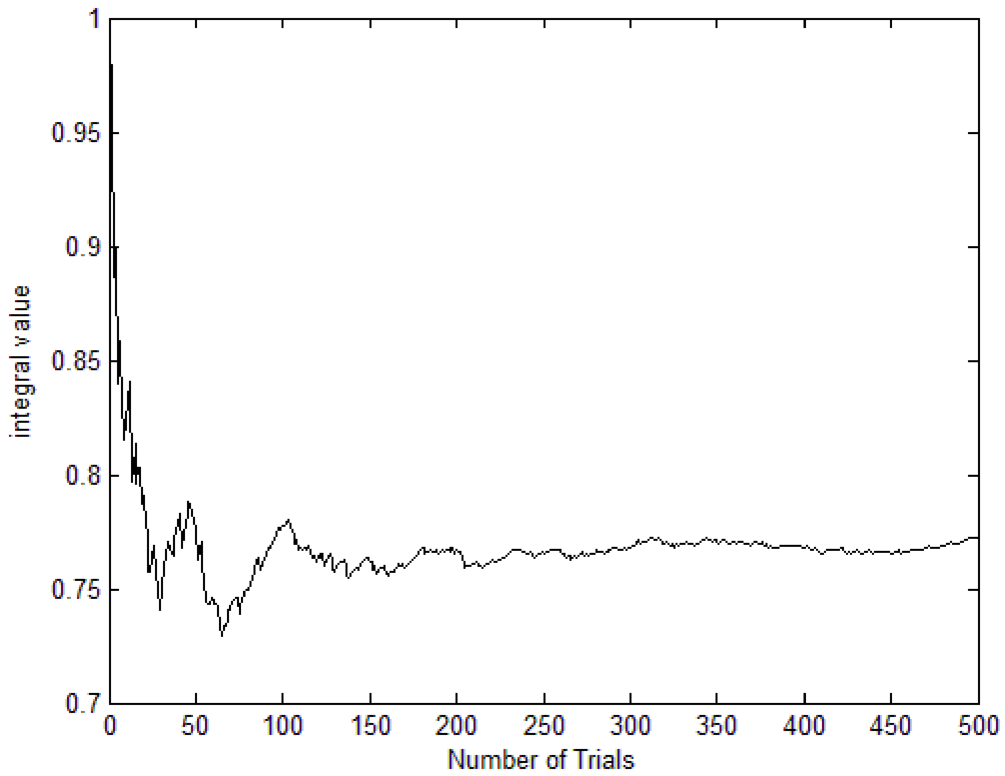
s1:真实积分值

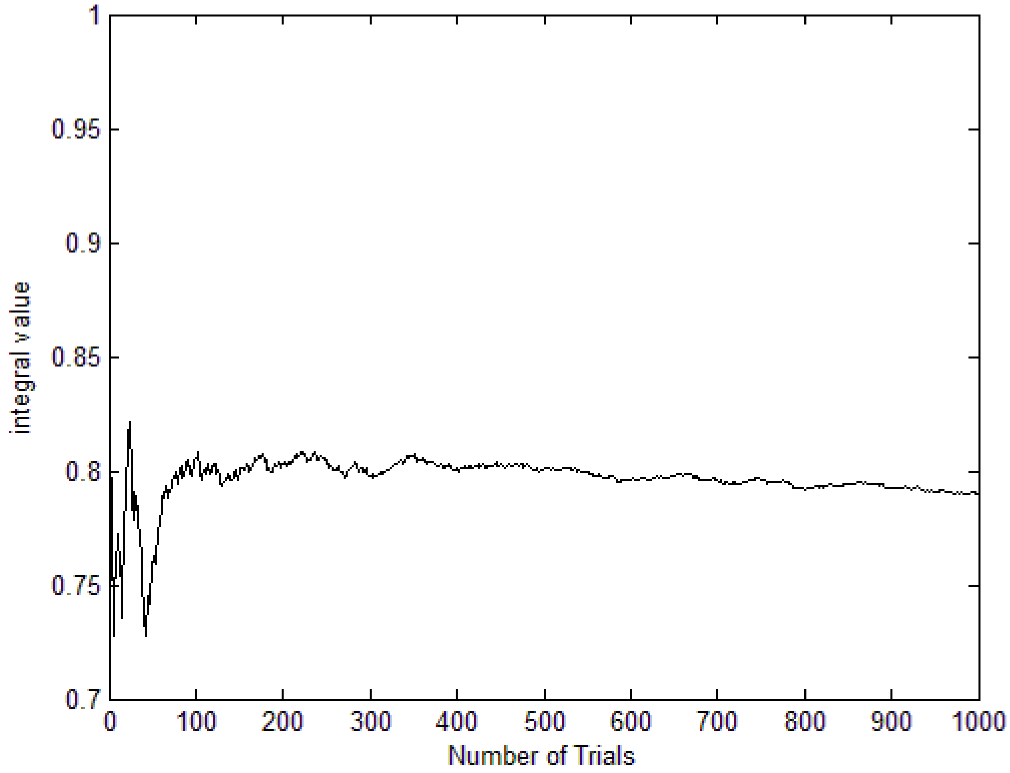
由此可知，N 越大，蒙特卡罗积分结果就越接近真实的积分值。

9.14

```
clear all
N=input('the parameter N iter:=')                    % Number of experiments                    % Trials per experiment
u = rand(N,1);               % Generate random numbers
uu =sqrt(1-u.*u);% Define function
data = zeros(N,1);           % Initialize array
% The following four lines of code determine
% M estimates as a function of j, 0<j<=N.
data(1,:)=uu(1,:);
for j=2:N
   data(j,:)=sum(uu(1:j,:))/j;
end
est = data(N,:)              % M estimates of pi
s1=quad('sqrt(1-u.*u)',0,1)
plot(data,'k')               % Plot results
xlabel('Number of Trials')
ylabel('integral value')
```


1
0.95
0.9
0.85
0.8
0.75
integral value
0
10
20
30
40
50
60
70
80
90
100
Number of Trials


1
0.95
0.9
0.85
0.8
0.75
0.7
integral value
0
50
100
150
200
250
300
350
400
450
500
Number of Trials

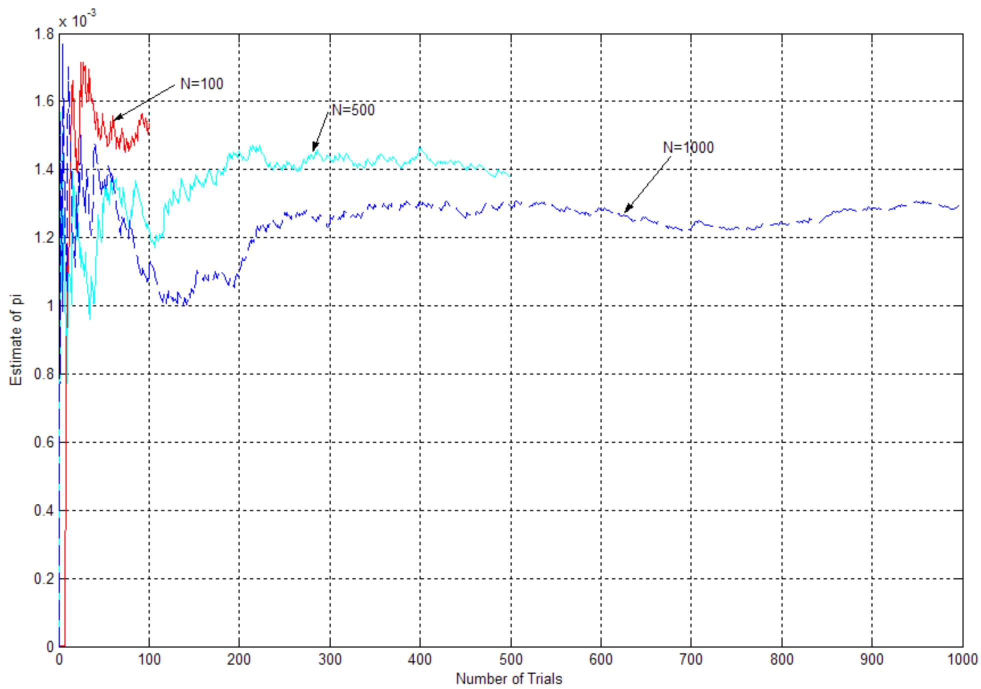


9-16

答：$Q(3)=\dfrac{1}{\sqrt{2\pi}}\int_3^{\infty}\exp(-y^2/2)dy=\dfrac{1}{2}erfc(3/\sqrt{2})=0.0013$

$Q_1(3)=\dfrac{1}{\pi}\int_0^{\pi/2}\exp(-\dfrac{3^2}{2\sin^2\theta})d\theta=\dfrac{1}{\pi}\cdot\dfrac{\pi}{2}\int_0^1\exp(-\dfrac{3^2}{2\sin^2(\pi\theta/2)})d\theta=\dfrac{1}{2}\int_0^1\exp(-\dfrac{3^2}{2\sin^2(\pi\theta/2)})d\theta$ Matlab 程序基本同前：

gx=exp(-9/(2*sin(pi*x*0.5).^2));



第 10 章 通信系统的蒙特卡罗仿真

10.3

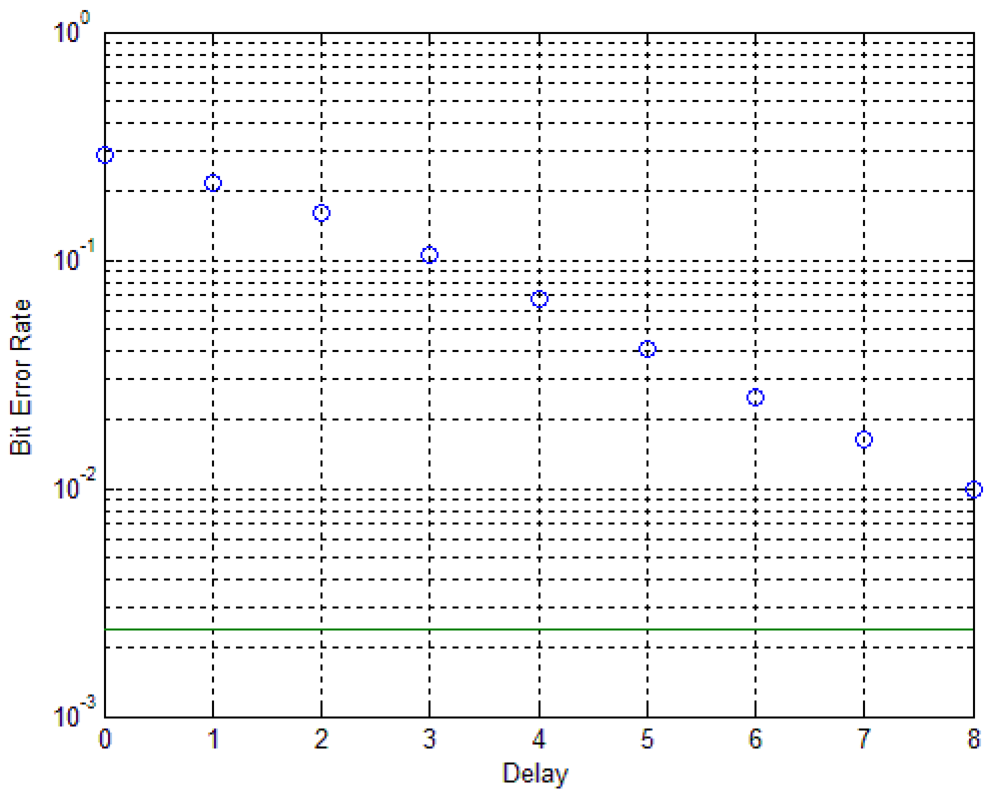
首先要修改采样次数：

```
% File: c10_MCBPSKrun.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
%
function [BER,Errors]=MCBPSKrun(N,EbNo,delay,FilterSwitch)
SamplesPerSymbol = 20;                        % samples per symbol
BlockSize = 1000;                             % block size
NoiseSigma = sqrt(SamplesPerSymbol/(2*EbNo)); % scale noise level
DetectedSymbols = zeros(1,BlockSize);         % initialize vector
NumberOfBlocks = floor(N/BlockSize);          % number of blocks processed
[BTx,ATx] = butter(5,2/SamplesPerSymbol);     % compute filter parameters
[TxOutput,TxFilterState] = filter(BTx,ATx,0); % initialize state vector
BRx = ones(1,SamplesPerSymbol); ARx=1;        % matched filter parameters
Errors = 0;                                   % initialize error counter
%
% Simulation loop begine here.
%
for Block=1:NumberOfBlocks
     %
     % Generate transmitted symbols.
     %
     [SymbolSamples,TxSymbols] = random_binary(BlockSize,SamplesPerSymbol);
     %
     % Transmitter filter if desired.
     %
     if FilterSwitch==0
        TxOutput = SymbolSamples;
     else
        [TxOutput,TxFilterState] = filter(BTx,ATx,SymbolSamples,TxFilterState);
     end
     %
     % Generate channel noise.
     %
     NoiseSamples = NoiseSigma*randn(size(TxOutput));
     %
     % Add signal and noise.
     %
     RxInput = TxOutput + NoiseSamples;
     %
     % Pass Received signal through matched filter.
     %
     IntegratorOutput = filter(BRx,ARx,RxInput);
     %
```

```
     % Sample matched filter output every SamplesPerSymbol samples,
     % compare to transmitted bit, and count errors.
     %
     for k=1:BlockSize,
        m = k*SamplesPerSymbol+delay;
        if (m < length(IntegratorOutput))
           DetectedSymbols(k) = (1-sign(IntegratorOutput(m)))/2;
           if (DetectedSymbols(k) ~= TxSymbols(k))
              Errors = Errors + 1;
           end
        end
     end
end
BER = Errors/(BlockSize*NumberOfBlocks);    % calculate BER
% End of function file.
```

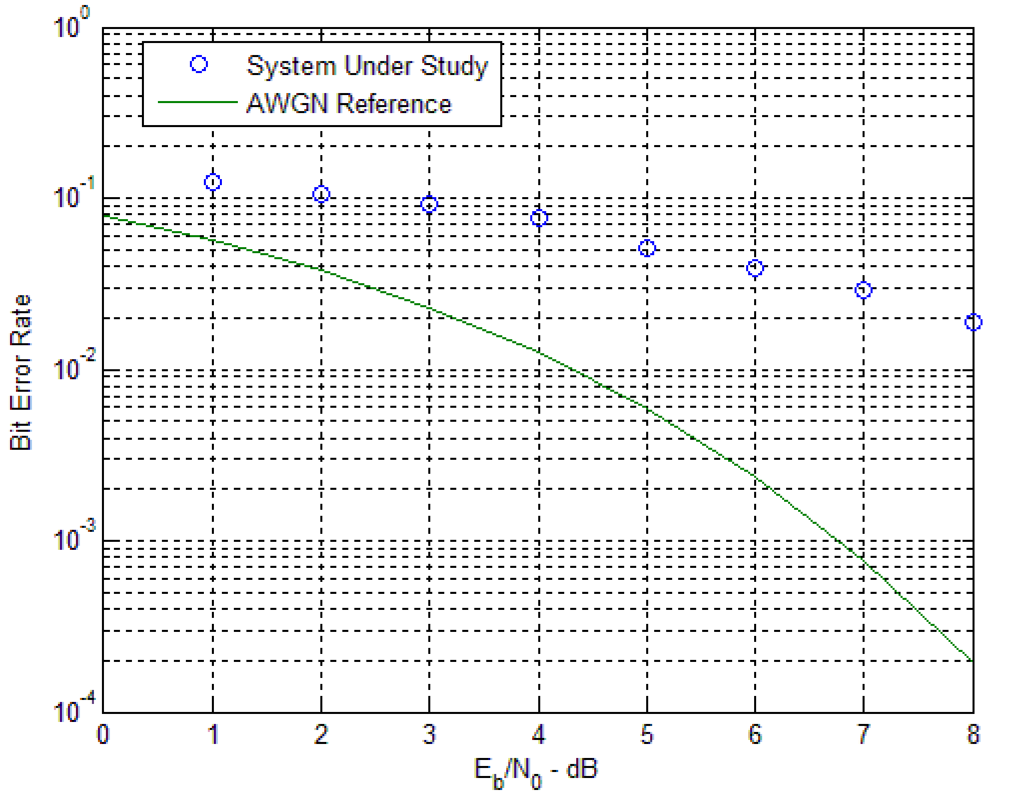
Delay仿真估计：

```
% File: c10_MCBPSKdelay.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
%
EbNodB = 6;                                    % Eb/No (dB) value
z = 10.^(EbNodB/10);                           % convert to linear scale
delay = 0:8;                                   % delay vector
BER = zeros(1,length(delay));                  % initialize BER vector
Errors = zeros(1,length(delay));               % initialize Errors vector
BER_T = q(sqrt(2*z))*ones(1,length(delay));    % theoretical BER vector
N = round(100./BER_T);                         % 100 errors for ideal (zero ISI) system
FilterSwitch = 1;                              % set filter switch (in=1 or out=0)
for k=1:length(delay)
   [BER(k),Errors(k)] = c10_MCBPSKrun(N(k),z,delay(k),FilterSwitch)
end
semilogy(delay,BER,'o',delay,BER_T,'-'); grid;
xlabel('Delay'); ylabel('Bit Error Rate');
% End of script file.
```

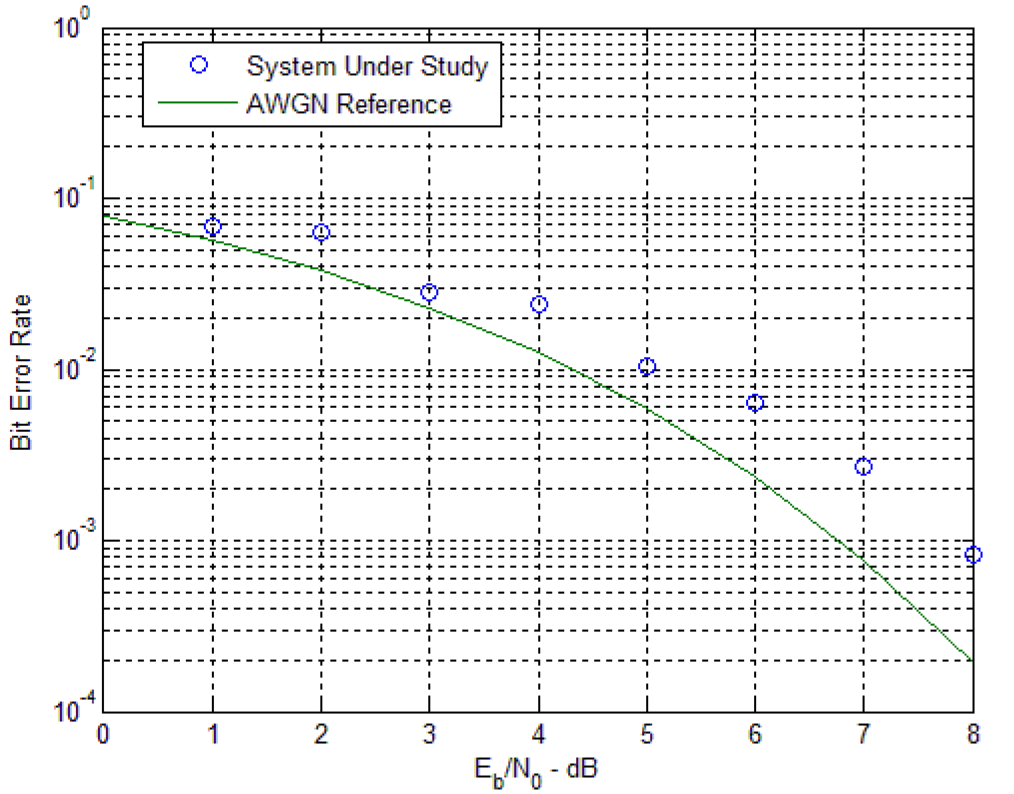


BER仿真估计：

```
% File: c10_MCBPSKber.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
%
EbNodB = 0:8;                    % vector of Eb/No (dB) values
z = 10.^(EbNodB/10);             % convert to linear scale
delay = 5;                       % enter delay value (samples)
BER = zeros(1,length(z));        % initialize BER vector
Errors = zeros(1,length(z));     % initialize Errors vector
BER_T = q(sqrt(2*z));            % theoretical (AWGN) BER vector
N = round(20./BER_T);            % 20 errors for ideal (zero ISI) system
FilterSwitch = 1;                % Tx filter out (0) or in (1)
for k=1:length(z)
   N(k) = max(1000,N(k));        % ensure at least one block processed
   [BER(k),Errors(k)] = c10_MCBPSKrun(N(k),z(k),delay,FilterSwitch)
end
semilogy(EbNodB,BER,'o',EbNodB,BER_T)
xlabel('E_b/N_0 - dB'); ylabel('Bit Error Rate'); grid
legend('System Under Study','AWGN Reference',0)
% End of script file.
```



10-1中的图形为：



10.4

蒙特卡罗方法：

```
tic
EbNodB = 0:8;                          % vector of Eb/No (dB) values
```

```
z = 10.^(EbNodB/10);            % convert to linear scale
delay = 5;                      % enter delay value (samples)
BER = zeros(1,length(z));       % initialize BER vector
Errors = zeros(1,length(z));    % initialize Errors vector
BER_T = q(sqrt(2*z));           % theoretical (AWGN) BER vector
N = round(20./BER_T);           % 20 errors for ideal (zero ISI) system
FilterSwitch = 1;               % Tx filter out (0) or in (1)
for k=1:length(z)
   N(k) = max(1000,N(k));       % ensure at least one block processed
   [BER(k),Errors(k)] = c10_MCBPSKrun(N(k),z(k),delay,FilterSwitch)
end
semilogy(EbNodB,BER,'o',EbNodB,BER_T)
xlabel('E_b/N_0 - dB'); ylabel('Bit Error Rate'); grid
legend('System Under Study','AWGN Reference',0)
toc
% End of script file.
Elapsed time is 0.484517 seconds.
```

半解析估计法：

```
tic
NN = 256;                                  % number of symbols
tb = 1;                                    % bit file
p0 = 1;                                    % power
fs = 16;                                   % samples/symbol
ebn0db = [0:1:8];                          % Eb/No vector in dB
[bt,at] = butter(5,2/fs);                  % transmitter filter parameters
x = random_binary(NN,fs);                  % establish PSK signal
y1 = x;                                    % save signal
y2a = y1*sqrt(p0);                         % scale amplitude
y2 = filter(bt,at,y2a);                    % transmitter output
br = ones(1,fs); br = br/fs; ar = 1;       % matched filter parameters
y = filter(br,ar,y2);                      % matched filter output
%
% End of simulation.
%
% The following code sets up the semianalytic estimator. Find the
% maximum magnitude of the cross correlation and the corresponding lag.
%
[cor lags] = vxcorr(x,y);                  % compute crosscorrelation
[cmax nmax] = max(abs(cor));               % maximum of crosscorrelation
timelag = lags(nmax);                      % lag at max crosscorrelation
theta = angle(cor(nmax));                  % determine angle
y = y*exp(-i*theta);                       % derotate
%
% Noise BW calibration.
%
hh = impz(br,ar);                          % receiver impulse response
nbw = (fs/2)*sum(hh.^2);                   % noise bandwidth
```

```
%
% Delay the input and do BER estimation on the NN-20+1 128 bits.
% Use middle sample. Make sure the index does not exceed number
% of input points. Eb should be computed at the receiver input.
%
index = (10*fs+8:fs:(NN-10)*fs+8);
xx = x(index);
yy = y(index-timelag+1);
eb = tb*sum(abs(y2).^2)/length(y2);
eb = eb/2;
[peideal,pesystem] = psk_berest(xx,yy,ebn0db,eb,tb,nbw);
semilogy(ebn0db,pesystem,'ro-',ebn0db,peideal); grid;
xlabel('E_b/N_0 (dB)'); ylabel('Bit Error Rate')
legend('System Under Study','AWGN Reference',0)
toc
% End of script file.
Elapsed time is 0.171073 seconds.
```

10-6

答：matlab 程序：

```
EbNodB = 0:10;
z = 10.^(EbNodB/10);
delay = 5;
BER = zeros(1,length(z));
Errors = zeros(1,length(z));
BER_T = q(sqrt(2*z));
N = round(20./BER_T);
FilterSwitch = 1;
for k=1:length(z)
    N(k) = max(1000,N(k));
    [BER(k),Errors(k)] = c10_MCBPSKrun(N(k),z(k),delay,FilterSwitch) ;
end
semilogy(EbNodB,BER,'o',EbNodB,BER_T) ;
xlabel('E_b/N_0 - dB'); ylabel('Bit Error Rate');
grid
legend('System Under Study','AWGN Reference',0) ;
% End of script file.
```

其中，子程序 1：

```
function [BER,Errors]=MCBPSKrun(N,EbNo,delay,FilterSwitch)
SamplesPerSymbol = 10;
BlockSize = 1000;
NoiseSigma = sqrt(SamplesPerSymbol/(2*EbNo));
DetectedSymbols = zeros(1,BlockSize);
NumberOfBlocks = floor(N/BlockSize);
[BTx,ATx] = butter(5,2/SamplesPerSymbol);
[TxOutput,TxFilterState] = filter(BTx,ATx,0);
BRx = ones(1,SamplesPerSymbol); ARx=1;
Errors = 0;
for Block=1:NumberOfBlocks
    [SymbolSamples,TxSymbols] =random_binary(BlockSize,SamplesPerSymbol);
    if FilterSwitch==0
```

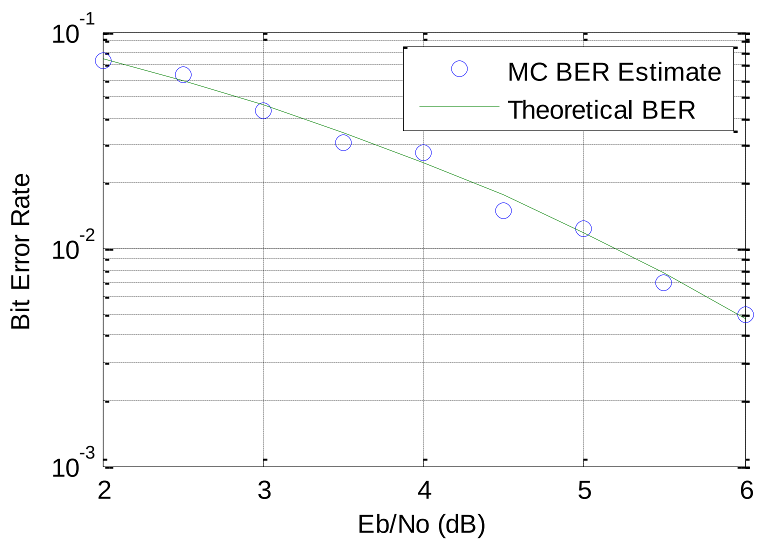
```
            TxOutput = SymbolSamples;
        else
            [TxOutput,TxFilterState] =filter(BTx,ATx,SymbolSamples,TxFilterState);
        end
      NoiseSamples = NoiseSigma*randn(size(TxOutput));
      RxInput = TxOutput + NoiseSamples;
      IntegratorOutput = filter(BRx,ARx,RxInput);
      for k=1:BlockSize,
            m = k*SamplesPerSymbol+delay;
            if (m < length(IntegratorOutput))
                DetectedSymbols(k) = (1-sign(IntegratorOutput(m)))/2;
                if (DetectedSymbols(k) ~= TxSymbols(k))
                    Errors = Errors + 1;
                end
            end
        end
  end
  BER = Errors/(BlockSize*NumberOfBlocks);
  % End of function file.
```

子程序 2：function y=q(x)

```
  y = 0.5*erfc(x/sqrt(2));
  % End function file.
```

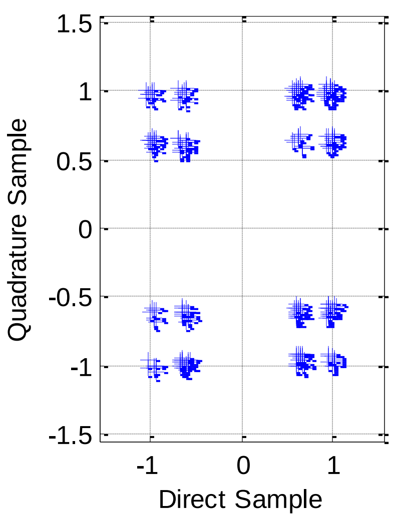
子程序 3：function [x, bits] = random_binary(nbits,nsamples)

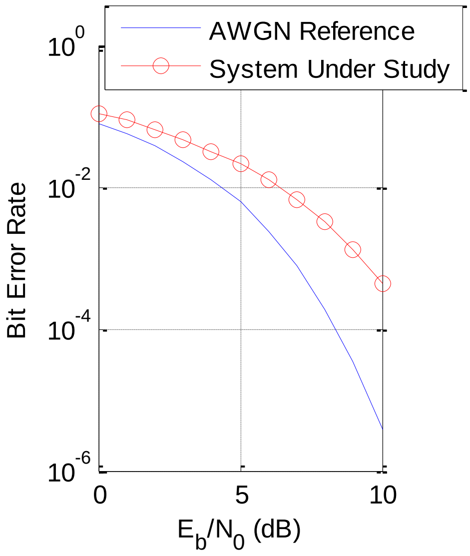
```
  x = zeros(1,nbits*nsamples);
  bits = round(rand(1,nbits));
  for m=1:nbits
      for n=1:nsamples
          index = (m-1)*nsamples + n;
          x(1,index) = (-1)^bits(m);
      end
  end
  % End of function file.
```



答：matlab 程序：

```
NN = 256;
tb = 0.5;
p0 = 1;
fs = 16;
ebn0db = [0:1:10];
[b,a] = butter(5,1/16);
x = random_binary(NN,fs)+i*random_binary(NN,fs);
y1 = x;
y2a = y1*sqrt(p0);
y2 = filter(b,a,y2a);
b = ones(1,fs); b = b/fs; a = 1;
y = filter(b,a,y2);
 [cor lags] = vxcorr(x,y);
cmax = max(abs(cor));
nmax = find(abs(cor)==cmax);
timelag = lags(nmax);
theta = angle(cor(nmax));
y = y*exp(-i*theta);
hh = impz(b,a);
nbw = (fs/2)*sum(hh.^2);
index = (10*fs+8:fs:(NN-10)*fs+8);
xx = x(index);
yy = y(index-timelag+1);
[n1 n2] = size(y2); ny2=n1*n2;
eb = tb*sum(sum(abs(y2).^2))/ny2;
eb = eb/2;
[peideal,pesystem]= qpsk_berest(xx,yy,ebn0db,eb,tb,nbw);
subplot(1,2,1)
yscale = 1.5*max(real(yy));
plot(yy,'+')
xlabel('Direct Sample'); ylabel('Quadrature Sample'); grid;
axis([-yscale yscale -yscale yscale])
subplot(1,2,2)
semilogy(ebn0db,peideal,ebn0db,pesystem,'ro-'); grid;
xlabel('E_b/N_0 (dB)'); ylabel('Bit Error Rate')
legend('AWGN Reference','System Under Study')
% End of script file.
```





第 11 章 无线系统仿真的方法论

11-1

答：

（a） 因为 Ak=+1 或-1，所以从 j=1 到 j=5 相对应的 Ak 值共有 32 种组合。当全 1 时，Dkmax=0.5；当全-1 时，Dkmin=-0.5. 得出 ISI 的分布范围是（-0.5，0.5）。

（b） Aj 的奇数阶距是零，偶数阶距是 1。由 $D_k = \sum_{j=1}^{5} h_j A_{j-k}$ ，可得出奇数阶距 D=0（k=1，3，5，7，9）；偶数阶距 D=0.5（k=2，4，6，8，10）

（c） 由 Dk 的各阶矩，可以获得 Dk 分布的一个离散近似。在该近似中，Dk 被当作一个可取 J 值的离散随机变量，这个 J 值为 d1,……dj，对应的概率为 p1,……，pj。

选择横坐标 x1,……,xj 和纵坐标 p1,……,pj，使得连续分布和离散分布具有相同的矩

$$E(X^n) = \mu_n = \sum_{k=1}^{j} x_k^n p_k$$

已知 X 的前 3 阶距，就可以求出 xk 和 pk，k=1，2，3.

（d） 使用 ISI 分布的离散近似，按下式就可以计算两种情况的差错概率：

$$E\{Q(A + D_k / \sigma_k)\} = \sum_{j=1}^{j} Q(A + d_j / \sigma_x) p_j$$

11-2

答：

（a） 因为 Ak，Bk=+1 或-1，所以 5 相对应的 Sk 值为 2，-2，0 三种。用下面表达式来表示确切分布：

$$D_x = \sum_{j=1}^{5} \alpha_j A_j - \sum_{j=0}^{5} \beta_j B_j$$

其中，$\overrightarrow{h_k} = \alpha_k + j\beta_k$

（b）Dx 的矩可通过下式进行计算：

$$E\{D_x^k\} = E\{[\sum_{j=1}^{5} \alpha_j A_j - \sum_{j=0}^{5} \beta_j B_j]^k\}, k = 1, \cdots\cdots, 10$$

这样就可以得到其前 10 阶距。

（c）同样道理，已知 X 的前 3 阶距，就可以求出 xk 和 pk，k=1，2，3.计算出 ISI 分布的一阶距匹配近似。
（d）使用 ISI 分布的离散近似，按下式就可以计算两种情况的差错概率：

$$E\{Q(A+D_k/\sigma_k)\}=\sum_{j=1}^{j}Q(A+d_j/\sigma_x)p_j$$

15.6

```
N = 200;
pie = zeros(N,3);
A = [0.8 0.15 0.05;0.2 0.7 0.1;0 0.1 0.9];
pie(1,:) = [1 0 0];
S=zeros(1,N);
S(1)=1;
for k=2:N
   pie(k,:) = pie(k-1,:)*A;
   R=rand(1);
   if R>0&R<=pie(k,1)
       S(k)=1;
   elseif R>pie(k,1)&R<=(pie(k,2)+pie(k,1))
        S(k)=2;
   else R>(pie(k,2)+pie(k,1))&R<=1
        S(k)=3;
   end
end
disp('The value of A^N is');A^N
disp('状态S各个时刻为：');S
t=1:200;
plot(t,S,'*')
% End of script file.
```

运行结果：

```
The value of A^N is

ans =

    0.2857    0.2857    0.4286
    0.2857    0.2857    0.4286
    0.2857    0.2857    0.4286

状态S各个时刻为：

S =

  Columns 1 through 13

     1     3     1     2     1     3     3     2     1     3     2     2     3

  Columns 14 through 26
```

3 3 1 2 3 3 2 3 1 2 3 1 1

Columns 27 through 39

1 1 3 1 1 1 3 2 3 2 2 3 2

Columns 40 through 52

1 3 3 1 3 2 3 2 3 2 2 1 1

Columns 53 through 65

3 2 2 1 3 2 3 3 3 2 3 3 3

Columns 66 through 78

3 3 2 2 2 2 3 2 3 2 2 3 2

Columns 79 through 91

2 3 3 3 3 2 3 1 3 1 1 3 3

Columns 92 through 104

1 1 3 1 2 3 1 2 1 3 3 2 2

Columns 105 through 117

2 2 1 3 3 2 3 1 2 3 3 2 2

Columns 118 through 130

3 1 3 1 2 2 3 1 3 3 3 3 2

Columns 131 through 143

2 1 3 2 3 3 2 3 1 2 3 3 1

Columns 144 through 156

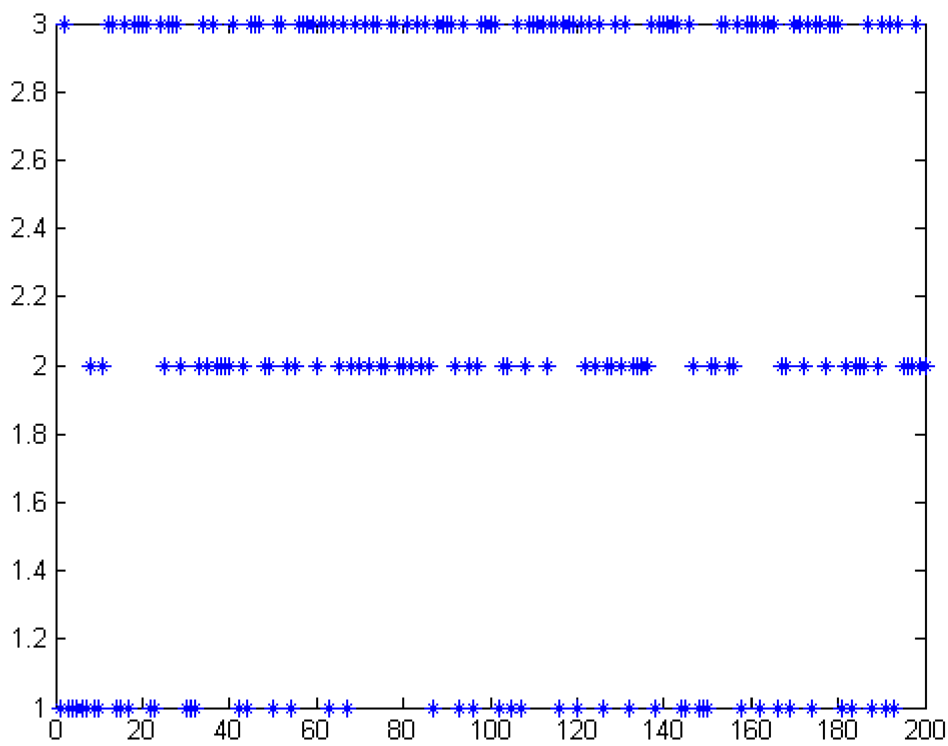
3 3 1 1 3 3 2 3 2 1 1 2 1

Columns 157 through 169

2 3 1 1 3 3 1 1 1 3 1 2 3

Columns 170 through 182

```
    3     3     1     2     2     2     1     3     3     3     2     2     1

  Columns 183 through 195

    2     3     3     1     1     3     1     3     3     1     1     1     1

  Columns 196 through 200

    3     1     3     1     1
```



1— 表示状态1
2— 表示状态2
3— 表示状态3

当N=40000；
先运行：

```
% File: c15_errvector.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
%
disp(' ')
disp('Default values are:')
N = 20000                                        % default N
A = [0.8 0.1 0.1; 0.2 0.6 0.2; 0.02 0.08 0.90]   % default A
B = [0.999 0.95 0.99; 0.001 0.05 0.01]           % default B
```

```
disp(' ')
disp('Accept default values?')
dtype = input('Enter y for yes or n for no  > ','s');
if dtype == 'n'
   N = input('   Enter N, the number of points to be generated > ');
   A = input('   Enter A, the state transition matrix > ');
   B = input('   Enter B, the error distribution matrx > ');
end
state = 1;                              % initial state
total_states = size(A,1);
out = zeros(1,N);                       % initialize error vector
state_seq = zeros(1,N);                 % initialize state sequence
h = waitbar(0,'Calculating Error Vector');
%
u2 = rand(1);                           % get random number
if u2>B(1,state)                        % test for error
   out(1) = 1;                          % record error
end
state_seq(1) = state;                   % record state
for t=2:N
   u1 = rand(1);                        % get random number
   cum_sum = [0 cumsum(A(state,:))];
   for i=1:total_states                 % loop to determine new state
      if u1>=cum_sum(i) & u1<cum_sum(i+1);
         state = i;                     % assign new state
      end
   end
   state_seq(t) = state;                % new record state
   u2 = rand(1);                        % get random number
   if u2>B(1,state)
      out(t) = 1;                       % record error
   end
   waitbar(t/N)
end
close(h)
% End of script file.
```

在对话框输入：

```
Accept default values?
Enter y for yes or n for no  > n
   Enter N, the number of points to be generated > 40000
   Enter A, the state transition matrix > [0.8 0.15 0.05;0.2 0.7 0.1;0 0.1 0.9]
   Enter B, the error distribution matrx > [0.9990 0.9500 0.9900;0.0010 0.0500 0.0100]
```

再运行以下程序：

```
% File: c15_hmmtest.m
% Software given here is to accompany the textbook: W.H. Tranter,
% K.S. Shanmugan, T.S. Rappaport, and K.S. Kosbar, Principles of
% Communication Systems Simulation with Wireless Applications,
% Prentice Hall PTR, 2004.
```

```
%
pe = sum(out)/N;
state_sum = zeros(1,total_states);
for k=1:N
   if state_seq(k)==1
      state_sum(1)=state_sum(1)+1;
   end
   if state_seq(k)==2
      state_sum(2)=state_sum(2)+1;
   end
   if state_seq(k)==3
      state_sum(3)=state_sum(3)+1;
   end
end
a = ['The probability of State 1 is ',num2str(state_sum(1)/N),'.'];
b = ['The probability of State 2 is ',num2str(state_sum(2)/N),'.'];
c = ['The probability of State 3 is ',num2str(state_sum(3)/N),'.'];
d = ['The error probability is ',num2str(pe),'.'];
disp('Simulation results:')
disp(a)         % display probability of state 1
disp(b)         % display probability of state 2
disp(c)         % display probability of state 3
disp(d)         % display error probability
% End script file.
Simulation results:
The probability of State 1 is 0.28092.
The probability of State 2 is 0.28405.
The probability of State 3 is 0.43503.
The error probability is 0.018.
>> A^N

ans =

    0.2857    0.2857    0.4286
    0.2857    0.2857    0.4286
    0.2857    0.2857    0.4286
```