浙江工业大学

MATLAB & Communication Simulations

Lab Report6

Name 凌智城

Teacher 张昱

Class 通信工程 1803 班

Student ID 201806061211

Department 信息工程学院

Date： May 29th,2021

I. Task_1

Consider 4ASK modulation. The signals are:

$$\mathrm{u_m}(t) = A_m g_T(t)\cos 2\pi f_c t\,, \qquad m = 1, 2, 3, ..., M$$

$$A_m = (2m-1-M)d\,, \qquad m = 1, 2, 3, ..., M$$

$$g_T(t)\begin{cases}\sqrt{\dfrac{2}{T}}, & 0 \le t \le T \\ 0, & otherwise\end{cases}$$

where d=1, T=1, fc=5Hz, the mapping rule from the bit pair to m is: 00->1, 01->2,11->3,10->4.

(1) If the message bits are 11 00 10 00 01, plot the corresponding 4ASK signal in both time and frequency domain, sampling frequency fs=1000Hz.

(2) Give the average signal energy per bit, i.e., Eb, for 4ASK.

(3) Denote SNR as Eb/N0, plot the theoretical SER (symbol error rate) and simulated SER, for SNR=0:2:8 dB.

Hint: refer to the slides and codes related to 4PAM baseband signal in Ch5 ppt.

A. Code

lab6_1.m

```
%% Consider the following 4ASK modulation, where T=1.

clear;
clc;

%% Task(1)
% Plot the corresponding 4ASK signal.
T = 1;                  % Symbol interval.
d = 1;
```

```
Tb = T/2;              % Bit interval.
m = [1,2,3,4];         % Equally spaced.
M = 4;                 % 4ASK:M=4
A_m = (2*m-1-M)*d;     % The amplitude of the m-th waveform.
fc = 5;                % Carrier frequency.
fs = 1000;             % Sampling frequency.
gT=sqrt(2/T);          % Transmission pulse.

% Message bits.
N = 5;
Sig = [1 1 0 0 1 0 0 0 0 1];
% Mapping.
for i=1:2:length(Sig)
    if(Sig(i)==0&&Sig(i+1)==0)
        m(fix(i/2)+1) = A_m(1);
    elseif(Sig(i)==0&&Sig(i+1)==1)
        m(fix(i/2)+1) = A_m(2);
    elseif(Sig(i)==1&&Sig(i+1)==1)
        m(fix(i/2)+1) = A_m(3);
    else
        m(fix(i/2)+1) = A_m(4);
    end
end

% Sampling.
[t,x] = Sampling(T,fs,m);
% Carrier wave.
carrier = cos(2*pi*fc*t);
% ASK modulation.
u_m=gT*x.*carrier;
% T2F.
[sf,U_m]=T2F(t,u_m);

% Plotting commands follow.
figure(1)
plot(t,u_m);
title('Time domain')
xlabel('t/s')
ylabel('u\_m(t)')
figure(2);
plot(sf,abs(U_m));
title('Frequency domain')
xlabel('f/Hz')
ylabel('U\_m')
```

```
%% Task(2)
% Give the average energy per symbol and average energy per bit.
Eav = sum(A_m.^2)/4    % Average energy per symbol.
Eb = Eav/2             % Average energy per bit.

%% Task(3)
% SER.

SNRindB=0:2:8;
for i=1:length(SNRindB)
  % Simulated error rate .
  smld_err_prb(i)=smldpe(SNRindB(i));
end
for i=1:length(SNRindB)
  % Signal-to-noise ratio.
  SNR_per_bit=exp(SNRindB(i)*log(10)/10);
  % Theoretical error rate.
  theo_err_prb(i)=(3/2)*Qfunct(sqrt((4/5)*SNR_per_bit));
end
% Plotting commands follow.
figure(3)
semilogy(SNRindB,smld_err_prb,'*');
hold
semilogy(SNRindB,theo_err_prb);
```

Sampling.m

```
function [T,Samp_Sig]=Sampling(t,Fs,sig)
% Fucntion Name:Sampling
%Input: Tb,Fs:sig OutPut:Samp_Sig
%When you call the Function ,u input the tiem for a bit,the
%Sampling rate and the source signal,then output the Samplint
Signal.
Ts=1/Fs;
Sig=sig;
len=length(Sig);
T=0:Ts:len*t-Ts;
Samp_Sig=T;
for i=0:1:len-1
    for j=1:1:t/Ts
        Samp_Sig(i*t/Ts+j)=Sig(i+1);
    end
end
```

T2T.m

```
function [f,sf]=T2F(t,st)
%input is time and the signal vectors
%output is freguency and signal spectrum

dt=t(2)-t(1);
T=t(end)-t(1)+dt;
df=1/T;                   %smapling rate
N=length(st);

f=-N/2*df:df:(N/2-1)*df;  % Frequency sampling point.
sf=fft(st);
% Compensation for time shift
sf=T/N*fftshift(sf).*exp(-j*2*pi*f*t(1));
```

smldpe.m

```
function [p]=smldpe(snr_in_dB)
% [p]=smldPe(snr_in_dB)
%      SMLDPE  simulates the probability of error for the given
%         snr_in_dB, signal to noise ratio in dB.
d=1;
SNR=exp(snr_in_dB*log(10)/10);    % signal to noise ratio per
bit
sgma=sqrt((5*d^2)/(4*SNR));  % sigma, standard deviation of noise
N=10000;                  % number of symbols being simulated
% generation of the quarternary data source follows
for i=1:N
  temp=rand;              % a uniform random variable over (0,1)
  if (temp<0.25)
    dsource(i)=1;         % with probability 1/4, source output
is "00"
  elseif (temp<0.5)
    dsource(i)=2;         % with probability 1/4, source output
is "01"
  elseif (temp<0.75)
    dsource(i)=3;         % with probability 1/4, source output
is "11"
  else
    dsource(i)=4;         % with probability 1/4, source output
is "10"
  end
end
```

```
% detection, and probability of error calculation
numoferr=0;
for i=1:N
  % The matched filter outputs
  if (dsource(i)==1)
    r=-3*d+gngauss(sgma);         % if the source output is "00"
  elseif (dsource(i)==2)
    r=-d+gngauss(sgma);           % if the source output is "01"
  elseif (dsource(i)==3)
    r=d+gngauss(sgma);            % if the source output is "11"
  else
    r=3*d+gngauss(sgma);          % if the source output is "10"
  end
  % detector follows
  if (r<-2*d)
    decis=1;                  % decision is "00"
  elseif (r<0)
    decis=2;                  % decision is "01"
  elseif (r<2*d)
    decis=3;                  % decision is "11"
  else
    decis=4;                  % decision is "10"
  end
  if (decis~=dsource(i))      % if it is an error, increase the
error counter
    numoferr=numoferr+1;
  end
end
p=numoferr/N;                 % probability of error
```

gngauss.m

```
function [gsrv1,gsrv2]=gngauss(m,sgma)
% [gsrv1,gsrv2]=gngauss(m,sgma)
% [gsrv1,gsrv2]=gngauss(sgma)
% [gsrv1,gsrv2]=gngauss
%      GNGAUSS  generates two independent Gaussian random
variables with mean
%         m and standard deviation sgma. If one of the input
arguments is missing
%         it takes the mean as 0, and the standard deviation as
the given parameter.
%         If neither mean nor the variance is given, it generates
two standard
```

```
%          Gaussian random variables.
if nargin == 0,
  m=0; sgma=1;
elseif nargin == 1,
  sgma=m; m=0;
end;
u=rand;                               % a uniform random variable in
(0,1)
z=sgma*(sqrt(2*log(1/(1-u))));        % a Rayleigh distributed
random variable
u=rand;                               % another uniform random variable
in (0,1)
gsrv1=m+z*cos(2*pi*u);
gsrv2=m+z*sin(2*pi*u);
```

Qfunct.m

```
function [y]=Qfunct(x)
% [y]=Qfunct(x)
%      QFUNCT  evaluates the Q-function.
%          y = 1/sqrt(2*pi) * integral from x to inf of exp(-
t^2/2) dt.
%          y = (1/2) * erfc(x/sqrt(2)).
y=(1/2)*erfc(x/sqrt(2));
```

B. Figure

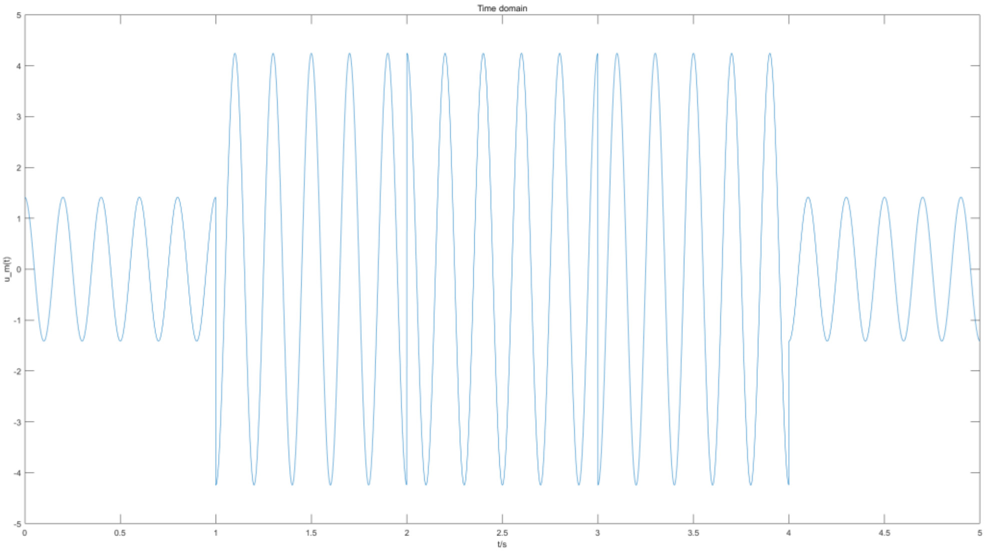


Fig.1 Message bits in Time domain(4ASK).

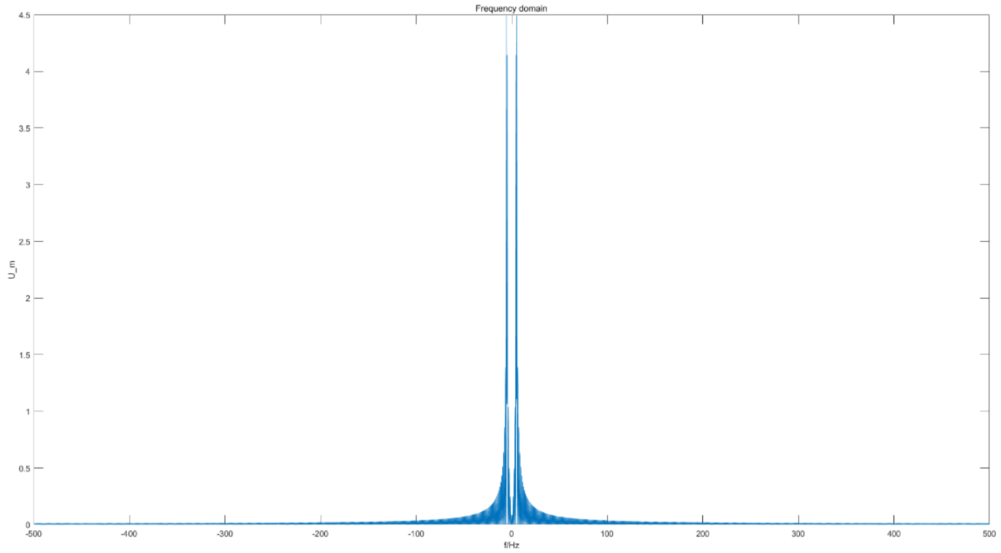
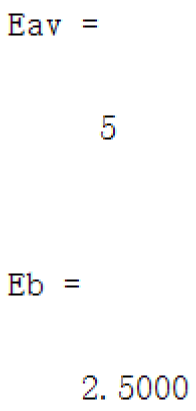


Fig.2 Message bits in Frequency domain.



```
Eav =

     5


Eb =

    2.5000
```

Fig.3 The average signal energy per bit.

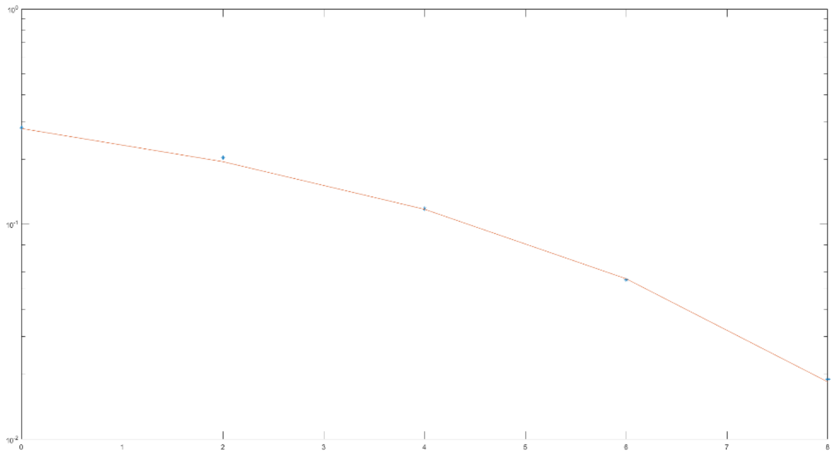


Fig.4 The theoretical SER and simulated SER

C. Discussion

Using carrier wave to transmit information is different from baseband transmission. Carrier-modulated digital transmission moves the frequency of the signal carrying the information to the frequency band of the signal.

The baseband signal waveform is multiplied by the sine carrier, and the frequency spectrum of the baseband signal is shifted by an amount of fc, and thus the signal is placed in the bandwidth of the channel.

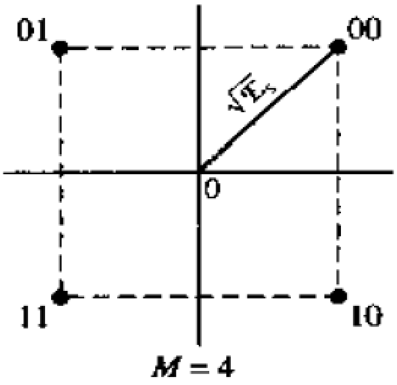
First map the 10-bit learning sequence to a binary number according to the law, and then change the corresponding amplitude according to the 4ASK mapping relationship. After sampling the baseband signal with fs=1000Hz, it is multiplied by the 5Hz sine carrier, and it is found that only the amplitude has occurred in the time domain. Change, the frequency has shifted in the frequency spectrum.

II. Task_2

Consider the 4PSK modulation. The signals are:

$$u_m(t)=\sqrt{\frac{2E_S}{T}}\cos\left(2\pi f_c t+\frac{2\pi m}{4}+\frac{\pi}{4}\right), m=0,1,2,3$$

where the average signal energy Es=1, T=1, fc=5Hz, and the constellation diagram is:



$$s_m=\left(\sqrt{E_s}\cos\left(\frac{2\pi m}{4}+\frac{\pi}{4}\right),\sqrt{E_s}\sin\left(\frac{2\pi m}{4}+\frac{\pi}{4}\right)\right)$$

(1) If the message bits are 11 00 10 00 01, plot the corresponding 4PSK signal in both time and frequency domain, sampling frequency fs=1000Hz.

(2) Plot the constellation diagram of the 4PSK experiencing the AWGN channel with N_0=0.5. (You can generate a large number of 4PSK signals and noise samples.)

(3) Give the BER and SER for N_0=0.5, from theoretical calculation and numerical simulation, respectively.

A. Code

lab6_2.m

```
%% Consider the following 4PSK modulation.

clear;
clc;

%% Task(1)
% Plot the corresponding 4PSK signal.
T = 1;                  % Symbol interval.
Es = 1;                 % Energy of every symbol.
Tb = T/2;               % Bit interval.
m = [0,1,2,3];          % Equally spaced.
M = 4;                  % 4PSK:M=4
fc = 5;                 % Carrier frequency.
fs = 1000;              % Sampling frequency.

% Message bits.
N = 5;
Sig = [1 1 0 0 1 0 0 0 0 1];
% Mapping.
for i=1:2:length(Sig)
    if(Sig(i)==0&&Sig(i+1)==0)
        phase_m(fix(i/2)+1) = m(1);
    elseif(Sig(i)==0&&Sig(i+1)==1)
        phase_m(fix(i/2)+1) = m(2);
    elseif(Sig(i)==1&&Sig(i+1)==1)
        phase_m(fix(i/2)+1) = m(3);
    else
        phase_m(fix(i/2)+1) = m(4);
    end
end

% Sampling.
[t,x] = Sampling(T,fs,phase_m);
for i=1:1:N*T*fs
    u_m(i) =sqrt(2*Es/T)*cos(2*pi*fc*i/fs+2*pi*x(i)/4+pi/4);
end
% T2F.
[sf,U_m]=T2F(t,u_m);

% Plotting commands follow.
```

```
figure(1)
plot(t,u_m);
title('Time domain')
xlabel('t/s')
ylabel('u\_m(t)')
figure(2);
plot(sf,abs(U_m));
title('Frequency domain')
xlabel('f/Hz')
ylabel('U\_m')

%% Task(2)
% Plot the constellation diagram.
N0 = 0.5;
nc=sqrt(N0/2)*randn(5000,1);
ns=sqrt(N0/2)*randn(5000,1);
for i=1:1:N*T*fs
    Sm_x(i) = sqrt(Es)*cos(2*pi*x(i)/4+pi/4)+nc(i);
    Sm_y(i) = sqrt(Es)*sin(2*pi*x(i)/4+pi/4)+ns(i);
end

% Plotting commands follow.
figure(3)
for i=1:1:5000
    if(x(i)==0)
        plot(Sm_x(i),Sm_y(i),'r*')
        hold on;
    elseif(x(i)==1)
        plot(Sm_x(i),Sm_y(i),'g+')
        hold on;
    elseif(x(i)==2)
        plot(Sm_x(i),Sm_y(i),'b.')
        hold on;
    else
        plot(Sm_x(i),Sm_y(i),'c.')
        hold on;
    end
    axis square
end
title('The constellation diagram of the 4PSK')

%% Task(3)
% BER and SER.
```

```
N=100000;
Eb=Es/2;
SNR = Eb/N0;
sgma=sqrt(Es/SNR/2);          % noise variance
% The signal mapping.
s00=[1 1];
s01=[-1 1];
s11=[-1 -1];
s10=[1 -1];
% Generation of the data source.
for i=1:N
  temp=rand;               % a uniform random variable between 0
and 1
  if (temp<0.25)           % With probability 1/4, source output
is "00."
    dsource1(i)=0;
    dsource2(i)=0;
  elseif (temp<0.5)        % With probability 1/4, source output
is "01."
    dsource1(i)=0;
    dsource2(i)=1;
  elseif (temp<0.75)       % With probability 1/4, source output
is "11."
    dsource1(i)=1;
    dsource2(i)=1;
  else                     % With probability 1/4, source output
is "10."
    dsource1(i)=1;
    dsource2(i)=0;
  end
end
% Detection and the probability of error calculation.
numofsymbolerror=0;
numofbiterror=0;
for i=1:N
  % The received signal at the detector, for the ith symbol, is:
  n(1)=gngauss(sgma);
  n(2)=gngauss(sgma);
  if ((dsource1(i)==0) && (dsource2(i)==0))
    r=s00+n;
  elseif ((dsource1(i)==0) && (dsource2(i)==1))
    r=s01+n;
  elseif ((dsource1(i)==1) && (dsource2(i)==1))
    r=s11+n;
  else
    r=s10+n;
  end
  % The correlation metrics are computed below.
  c00=dot(r,s00);
  c01=dot(r,s01);
  c11=dot(r,s11);
  c10=dot(r,s10);
  % The decision on the ith symbol is made next.
  c_max=max([c00 c01 c11 c10]);
  if (c00==c_max)
    decis1=0; decis2=0;
  elseif (c01==c_max)
    decis1=0; decis2=1;
  elseif (c11==c_max)
    decis1=1; decis2=1;
  else
    decis1=1; decis2=0;
  end
  % Increment the error counter, if the decision is not correct.
  symbolerror=0;
  if (decis1~=dsource1(i))
    numofbiterror=numofbiterror+1;
    symbolerror=1;
  end
  if (decis2~=dsource2(i))
    numofbiterror=numofbiterror+1;
    symbolerror=1;
  end
  if (symbolerror==1)
    numofsymbolerror = numofsymbolerror+1;
  end
end
% Since there are totally N symbols.
ps=numofsymbolerror/N
% Since 2N bits are transmitted.
pb=numofbiterror/(2*N)


theo_err_prb=Qfunct(sqrt(2*SNR))     % Theoretical bit-error rate.
theo_err_sym=theo_err_prb*2          % Theoretical symbol-error
rate.
```

B. Figure

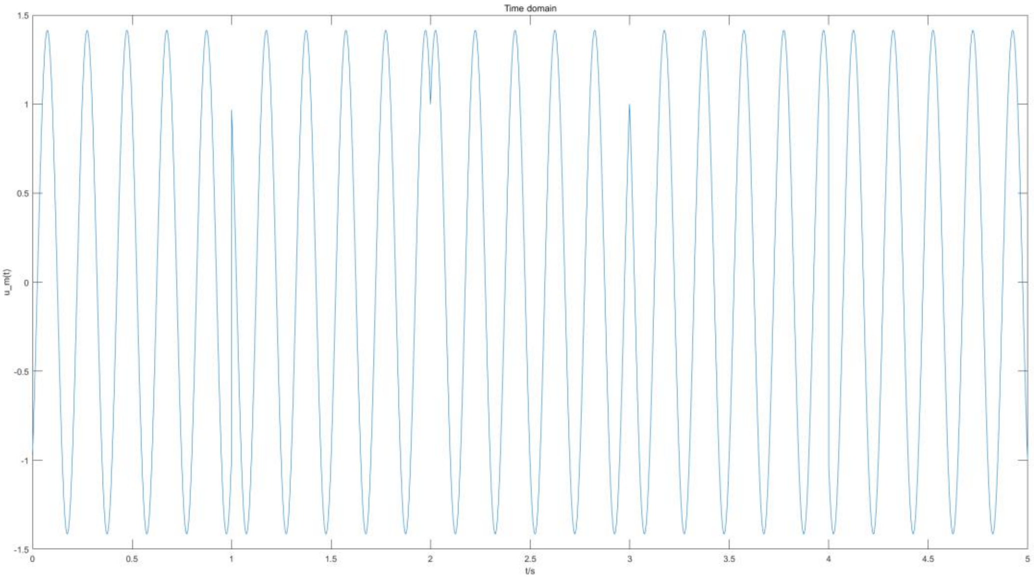


Fig.5 Message bits in Time domain(4PSK).

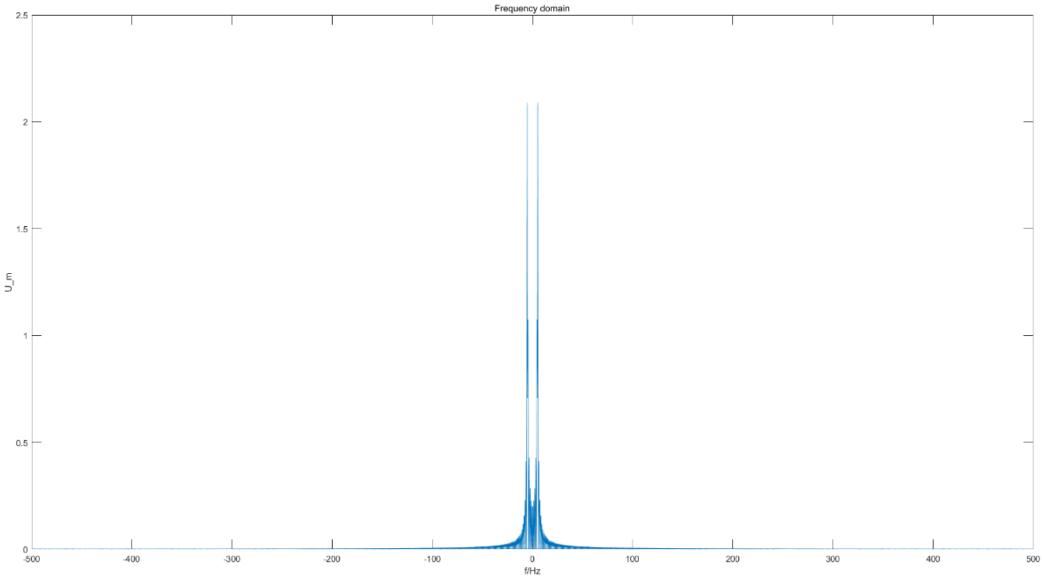


Fig.6 Message bits in Frequency domain(4PSK).

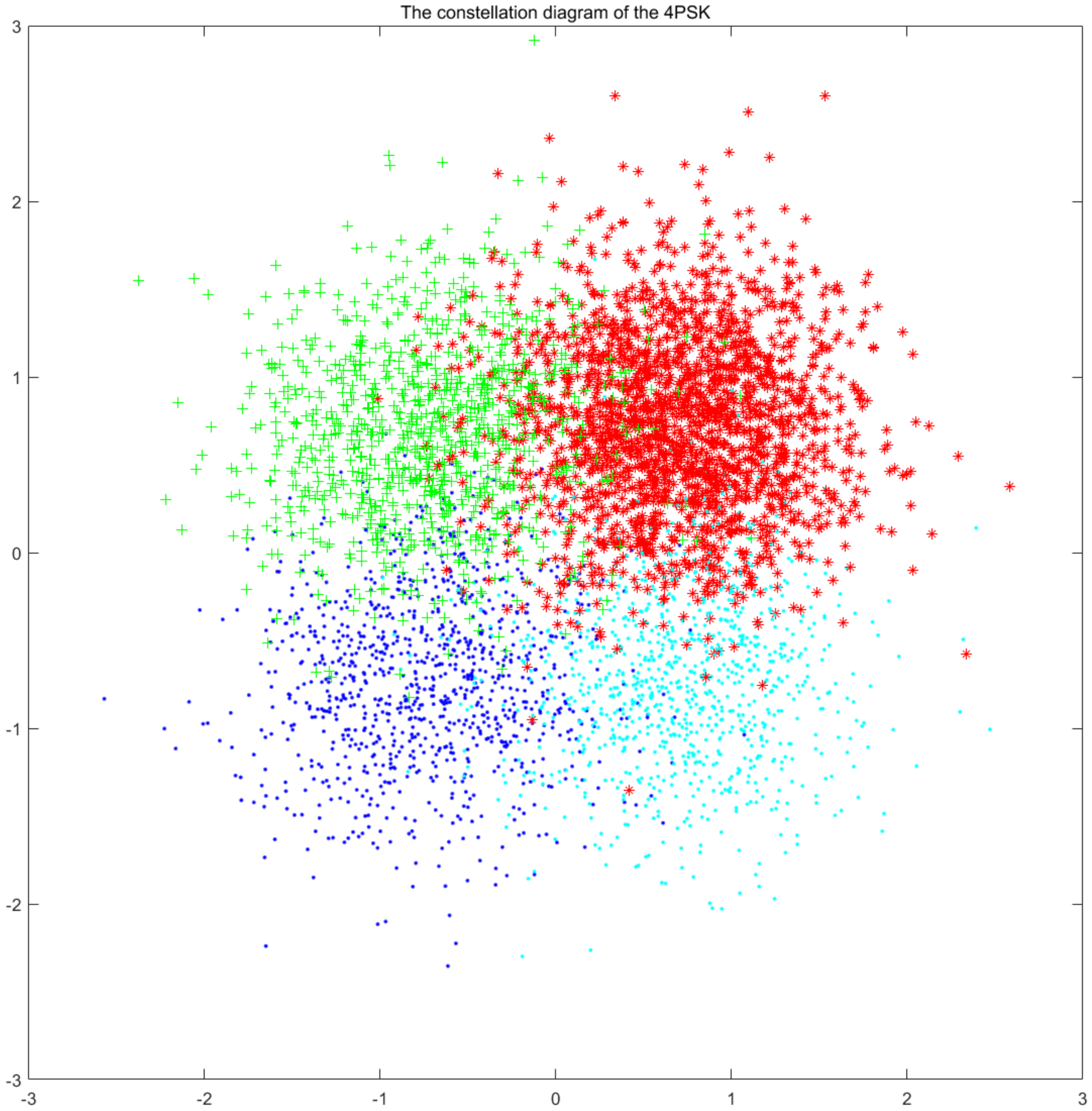
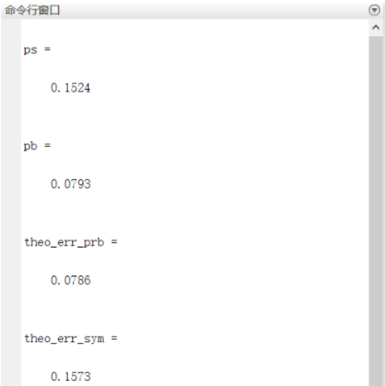


Fig.7 The constellation diagram of the 4PSK.



```
ps =

    0.1524


pb =

    0.0793


theo_err_prb =

    0.0786


theo_err_sym =

    0.1573
```

Fig.8 BER and SER(theoretical calculation and numerical simulation).

C. Discussion

In carrier phase modulation, the signal to be transmitted through a communication channel is embedded in the phase of the carrier, and the information sequence is binary-coded and mapped to the corresponding phase. Each piece of data corresponds to a phase. The amplitude does not change, but it changes. Initial phase.

When the phase of the two bits of information undergoes a sudden change, the time-domain waveform will also undergo a sudden change. The reason is the discontinuity of the phase, which may cause bit errors during demodulation.

In the constellation diagram, we can see that the PSK signal will drift due to noise. If the noise is too large, a large number of signals will drift to other phase points, causing an increase in the bit error rate.