

BPSK 系统

本例中仿真如图 1 所示的 BPSK 系统。此系统采用了升余弦滚降，滚降因子是 $\alpha=0.5$ 。

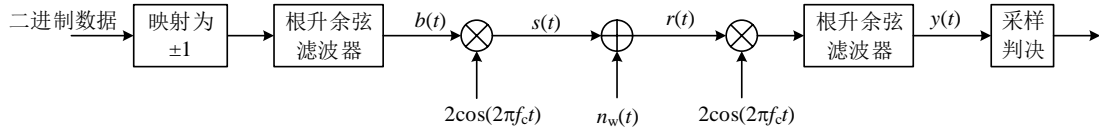


图 1 BPSK 系统

发送的 BPSK 信号 $s(t)$ 是 PAM 信号 $b(t)$ 对载波 $c(t) = 2\sin(2\pi f_c t)$ 的 DSB 调制：

$$s(t) = b(t)c(t) \quad (1)$$

其中

$$b(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT_b - t_0) = \sum_{n=-\infty}^{\infty} (-1)^{d_n} g_T(t - nT_b - t_0) \quad (2)$$

其中 $d_n \in \{0,1\}$ 是独立等概的二进制数据， $a_n \in \{\pm 1\}$ 是基带双极性 PAM 信号的幅度序列¹。发送端的基带成形脉冲 $g_T(t)$ 的傅氏变换是升余弦滚降频谱的平方根（RRC，root raised cosine）， t_0 代表第 0 个码元对应的的时间位置。接收滤波器的冲激响应 $g_R(t) = g_T(-t) = g_T(t)^2$ 。

以下图 2 是 BPSK 信号波形。图 3 是功率谱密度。发送信号的功率谱形状就是升余弦滚降特性的形状，本例中带宽是 750Hz。经过 DSB 调制后，OOK 信号的主瓣带宽是 kHz。

图 4 是 OOK 的误比特率曲线。OOK 的理论误比特率公式为

$$P_e = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad (3)$$

¹ 式(2)中是将二进制数据 0 映射为电平+1，也可以反过来把二进制 1 映射为电平+1

² 注意本例中的仿真是非实时仿真，不需要将滤波器因果化。也可以认为，本例仿真的滤波器延迟是仿真中信号 $s(t)$ 的总持续时间 T ：需要拿到全部 T 时间内的信号数据后才能进行频域的滤波。

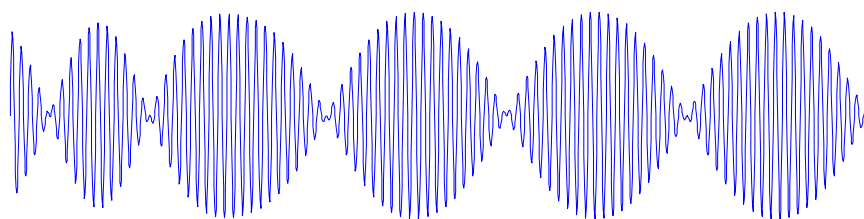


图 2 基带采用 RRC 成形时的 BPSK 信号波形

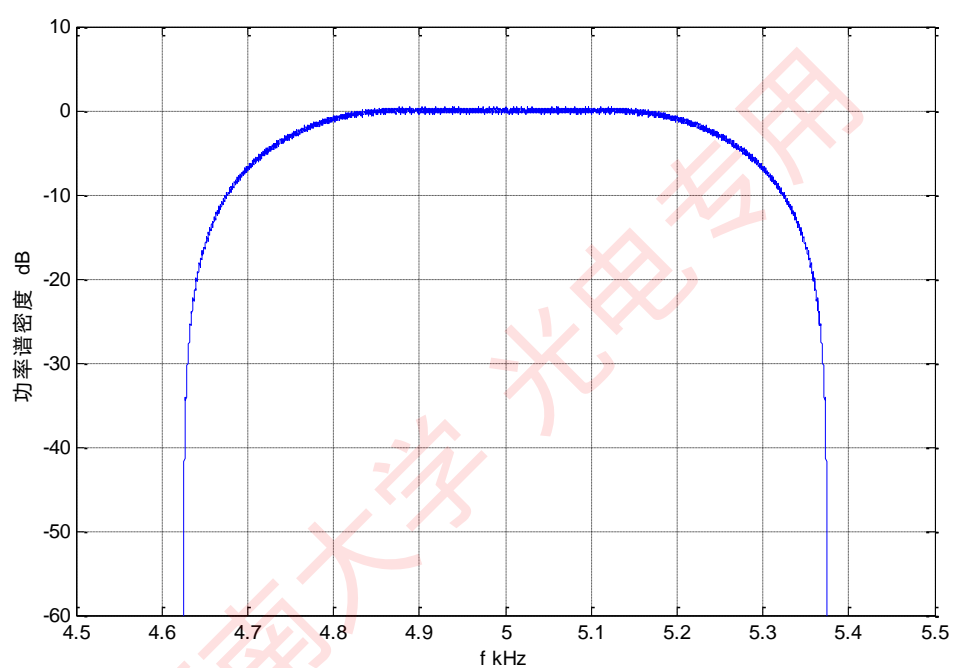


图 3 基带采用 RRC 成形时 BPSK 信号的功率谱密度

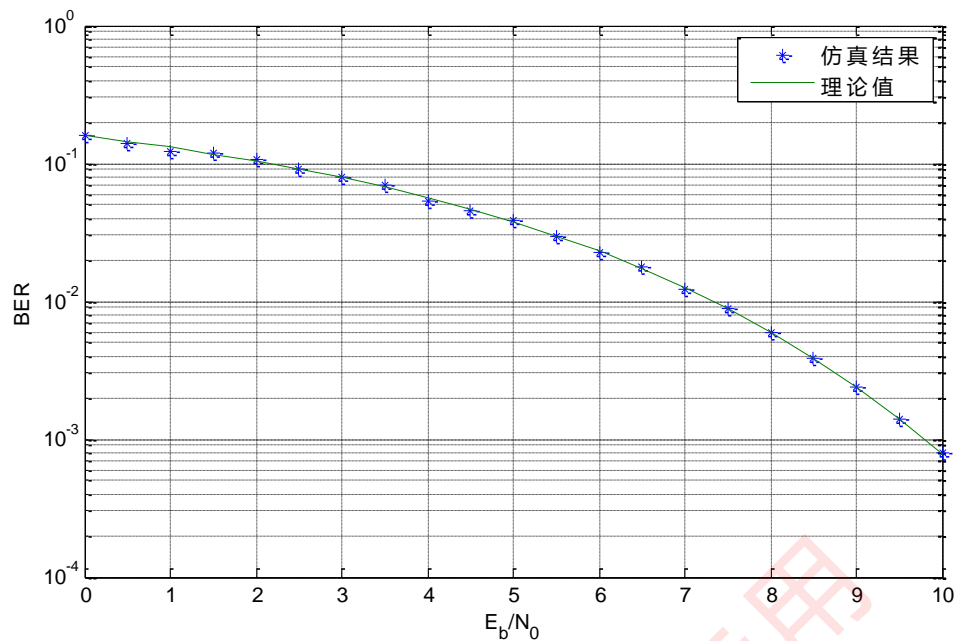


图 4 误比特率曲线

本例中的仿真代码如下：

```
clear all
close all

M=10000; %码元个数
Tb=2; %ms
Rb=1/Tb;
fc=10/Tb;
Nfc=16; %每载波周期内的采样点数
Ns=Tb*fc*Nfc; % 每码元的采样点数

T=M*Tb; %持续时间
dt=Tb/Ns; %采样间隔
fs=1/dt; %采样频率
df=1/T; %频域的采样间隔
t=[-T/2:dt:T/2-dt]; %ms
f=[-fs/2:df:fs/2-df]; %kHz

c=2*sin(2*pi*fc*t); % carrier

EbN0dB=[0:.5:10];
Pet=.5*erfc(10.^(EbN0dB/20));

Pe=zeros(1,length(EbN0dB));
nb_err=zeros(1,length(EbN0dB));
nb_bit=zeros(1,length(EbN0dB));
Psf=zeros(1,length(f));

ii=[Ns/2:Ns:length(t)]; %采样时刻

max_nb_err=1000;
max_nb=1e6;
```

```

%根升余弦滤波器
alpha=0.5;
W=Rb/2*(1+alpha);
W1=Rb/2*(1-alpha);
idx=find(abs(f)>W1 & abs(f)<=W);
X=Tb*ones(size(f));
X(abs(f)>W)=0;
X(idx)=Tb/2*(1+cos(pi*Tb/alpha*(abs(f(idx))-W1)));
RRC=sqrt(X);

s=zeros(1,length(t));

ie0=1;
while ie0<=length(EbN0dB)
    while min(nb_err(ie0))<max_nb_err && nb_bit(ie0)<max_nb
        for ie=ie0:length(EbN0dB)

            d=rand(1,M)<0.5;
            a=(-1).^d;

            tmp=zeros(size(t));
            tmp(ii)=a/dt;
            Tmp=t2f(tmp,fs).*RRC;
            b=real(f2t(Tmp,fs));

            s=b.*c;

            Tmp=t2f(s,fs);
            Psf=Psf+abs(Tmp).^2/T;

            Ps=mean(s.^2);
            Eb=Ps*Tb;

            ebn0=10^(EbN0dB(ie)/10);
            N0=Eb/ebn0;
            sigma=sqrt(N0*fs/2);

            nw=sigma*randn(1,length(t));

            r=s+nw;

            y=r.*c;
            Tmp=t2f(y,fs);
            y1=real(f2t(Tmp.*RRC,fs));

            yk=y1(ii);

            d_dec=(yk<0);
            nb_err(ie)=nb_err(ie)+nnz(d~=d_dec);
            nb_bit(ie)=nb_bit(ie)+M;
        end
        fprintf('\n
EbN0=%.2fdB   %d %d %d',EbN0dB(ie0),nb_bit(ie0),nb_err(ie0),ceil(nb_bit(ie0)*Pet(ie0))
)
    end
    ie0=ie0+1;

```

```
end

Psf=Psf/sum(Psf*df); %归一化的功率谱密度

Pe=nb_err./nb_bit;

figure(1)
semilogy(EbN0dB,Pe,'*',EbN0dB,Pet)

figure(2)
plot(f,10*log10(Psf))
axis([3,7,-60,30])
```

江南大学 光电专用

MASK

本例仿真 MASK 的 BER（bit error rate）及 SER（symbol error rate）。信道模型如下

y = x + z (1)

其中 $x \in \Omega$ 是星座点， Ω 是星座图， $z \sim \mathcal{N}\left(0, \frac{N_0}{2}\right)$ 是高斯噪声。各星座点等概出现，符号能量是 $E_s = \mathbb{E}[x^2]$ 。考虑格雷映射的星座图，如图 1 所示，比特与星座点的映射见表 1。

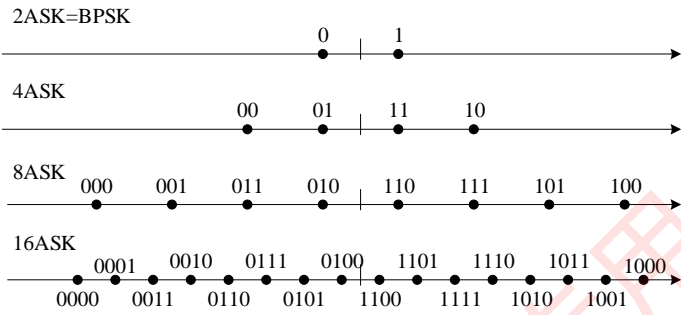


图 1 MASK 星座图

表 1 星座点映射

十进制序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二进制	0	1														
制比特	00	01	10	11												
	000	001	010	011	100	101	110	111								
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
电平映射	-1	+1														
	-3	-1	+3	+1												
	-7	-5	-1	-3	+7	+5	+1	+3								
	-15	-13	-9	-11	-1	-3	-7	-5	+15	+13	+9	+11	+1	+3	+7	+5

以下是 BER 曲线，是 SER 曲线。

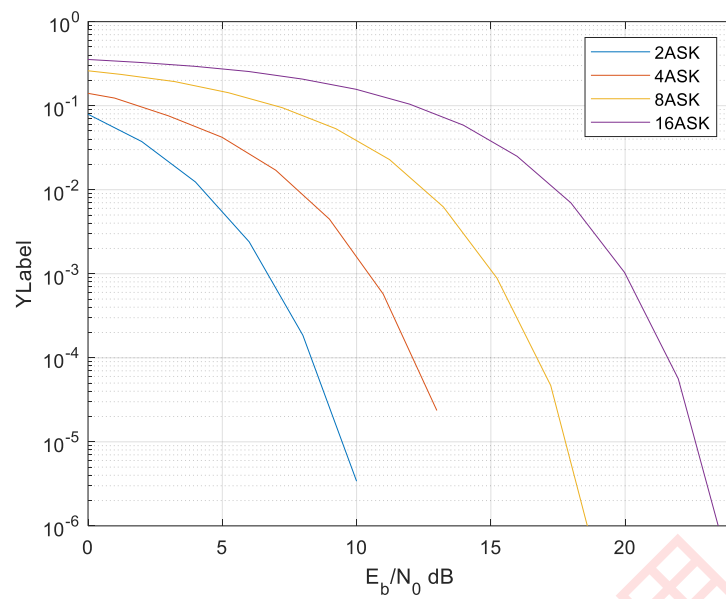


图 2 BER 曲线

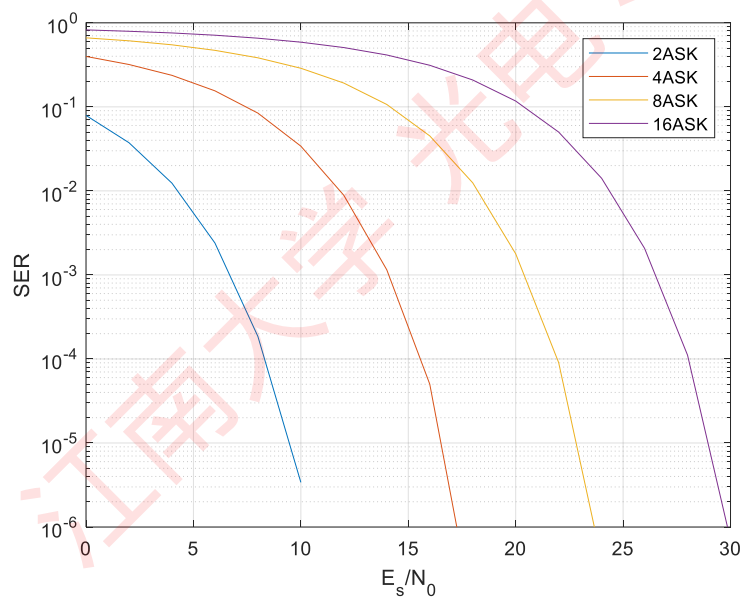


图 3 SER 曲线

本例中的仿真代码如下：

```
clear all
close all

K=10000; %比特数

EsN0dB=[0:2:30];

Peb=zeros(4,length(EsN0dB));
Pes=zeros(4,length(EsN0dB));
```

```

nb_errbit=zeros(4,length(EsN0dB));
nb_errsymb=zeros(4,length(EsN0dB));
nb_symb=zeros(1,length(EsN0dB));
max_nb_err=10000;
max_nb=1e7;

%归一化的星座图
cons2=[-1,+1];
cons4=[-3,-1,+3,+1]/sqrt(5);
cons8=[-7,-5,-1,-3,7,5,1,3]/sqrt(21);
cons16=[-15 -13 -9 -11 -1 -3 -7 -5 15 13 9 11 1 3 7 5]/sqrt(85);

%十进制转二进制的表
Tbl2=[0 0 ; 0 1; 1 0; 1 1];
Tbl3=[0 0 0; 0 0 1; 0 1 0; 0 1 1; 1 0 0; 1 0 1; 1 1 0; 1 1 1];
Tbl4=[0 0 0 0; 0 0 0 1; 0 0 1 0; 0 0 1 1; 0 1 0 0; 0 1 0 1; 0 1 1 0; 0 1 1 1; 1 0 0 0; 1 0 0 1; 1 0 1 0; 1 0 1 1; 1 1 0 0; 1 1 0 1; 1 1 1 0; 1 1 1 1];

ie0=1;
while ie0<=length(EsN0dB)
    while min(nb_errbit(:,ie0))<max_nb_err && nb_symb(ie0)<max_nb
        for ie=ie0:length(EsN0dB)

            b=(rand(4,K)>0.5);

            %2ASK
            d=b(1,:)+1;
            x(1,:)=cons2(d);

            %4ASK
            d=b(1,:)*2+b(2,:)+1;
            x(2,:)=cons4(d);

            %8ASK
            d=b(1,:)*4+b(2,:)*2+b(3,:)+1;
            x(3,:)=cons8(d);

            %16ASK
            d=b(1,:)*8+b(2,:)*4+b(3,:)*2+b(4,:)+1;
            x(4,:)=cons16(d);

            sigma=10^(-EsN0dB(ie)/20)/sqrt(2);

            z=sigma*randn(4,K);

            r=x+z;

            for ib=1:K
                %2ASK
                nb_errsymb(1,ie)=nb_errsymb(1,ie)+(r(1,ib)*x(1,ib)<0);

                %4ASK
                [tmp,idx]=min(abs(r(2,ib)-cons4));
                x_dec=cons4(idx);
                if x_dec~=x(2,ib)
                    b_dec=Tbl2(idx,:);
                    nb_errsymb(2,ie)=nb_errsymb(2,ie)+1;
                end
            end
        end
        ie0=ie0+1;
    end
end

```



```

        nb_errbit(2,ie)=nb_errbit(2,ie)+nnz(b_dec~=b(1:2,idx));
    end

    %8ASK
    [tmp,idx]=min(abs(r(3,ib)-cons8));
    x_dec=cons8(idx);
    if x_dec~=x(3,ib)
        b_dec=Tbl3(idx,:);
        nb_errsymb(3,ie)=nb_errsymb(3,ie)+1;
        nb_errbit(3,ie)=nb_errbit(3,ie)+nnz(b_dec~=b(1:3,idx));
    end

    %16ASK
    [tmp,idx]=min(abs(r(4,ib)-cons16));
    x_dec=cons16(idx);
    if x_dec~=x(4,ib)
        b_dec=Tbl4(idx,:);
        nb_errsymb(4,ie)=nb_errsymb(4,ie)+1;
        nb_errbit(4,ie)=nb_errbit(4,ie)+nnz(b_dec~=b(:,idx));
    end

    end
    nb_errbit(1,ie)= nb_errsymb(1,ie);
    nb_symb(ie)=nb_symb(ie)+K;
end
fprintf('\n EsN0=%.2fdB : %d : %d %d %d %d : %d %d %d %d\n',EsN0dB(ie0),nb_symb(:,ie0),nb_errbit(:,ie0),nb_errsymb(:,ie0))
end
ie0=ie0+1;
end

for ie=1:length(EsN0dB)
    Peb(:,ie)=nb_errbit(:,ie)/nb_symb(ie).*[1 1/2 1/3 1/4]';
    Pes(:,ie)=nb_errsymb(:,ie)/nb_symb(ie);
    EbN0dB(:,ie)=EsN0dB(ie)-10*log10([1,2 3,4]);
end

```