16QAM 调制解调的仿真分析

一、 QAM（Quadrature Amplitude Modulation）

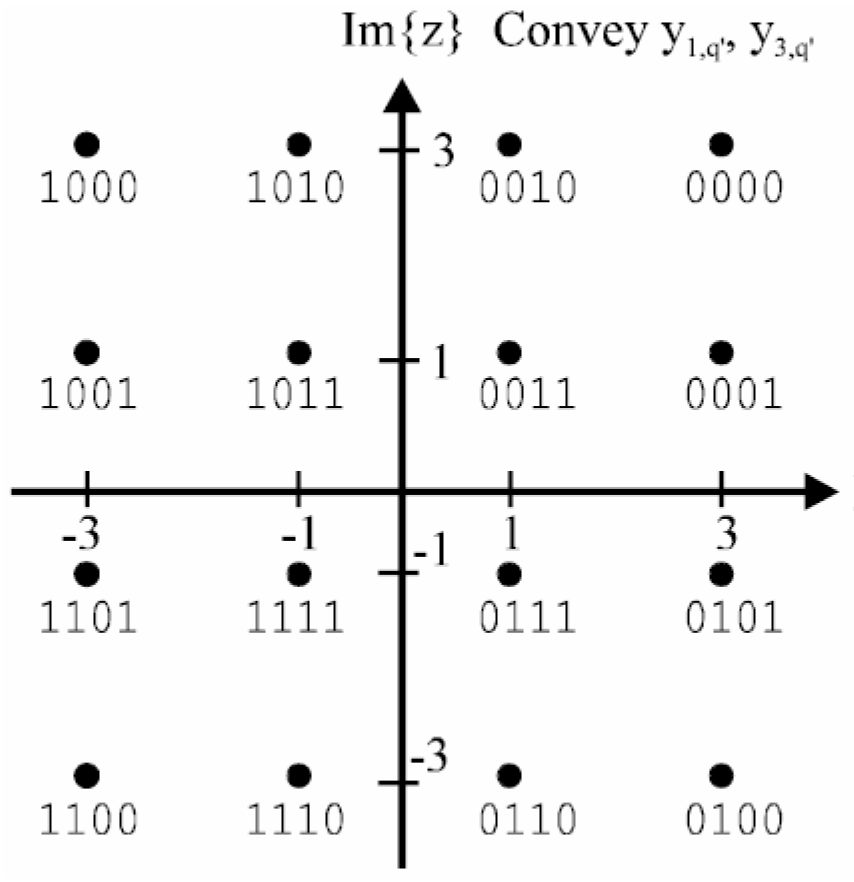
基本定义：

利用幅度和相位携带信息，完整正交幅度调制（QAM）表达式：

$$
\begin{aligned}
s_m(t) &= \mathrm{Re}\left[\left(A_{mc} + jA_{ms}\right)g(t)\exp j\left(2\pi f_c t\right)\right] \\
&= A_{mc}\,g(t)\cos\left(2\pi f_c t\right) - A_{ms}\,g(t)\sin\left(2\pi f_c t\right)
\end{aligned}
$$

星座图：

矩形 QAM 信号星座具有容易产生的独特优点，即通过在两个相位正交载波上施加两个 PAM 信号来产生。此外，他们也容易解调。虽然对于 M>=16 来说，该星座并不是最好的 M 元 QAM 信号星座，但是对于要达到给定的最小距离的要求来说，该星座所需要的平均发射功率仅稍大于最好的 M 元 QAM 信号星座所需要的平均功率。由于这些原因，矩形 M 元 QAM 信号在实际应用的最多。



显然 QAM 也是二维信号，可以用二维向量表示：

$$f_1(t)=\sqrt{\frac{2}{\varepsilon g}}g(t)\cos 2\pi f_c t \qquad f_2(t)=\sqrt{\frac{2}{\varepsilon g}}g(t)\sin 2\pi f_c t$$

QAM 信号表示为：

$$S_m=[\sqrt{\frac{\varepsilon_g}{2}}A_{mc},\sqrt{\frac{\varepsilon_g}{2}}A_{mS}]$$

任意一对信号向量间的欧氏距离为 ：

$$d_{mn}^{(e)}=\sqrt{(S_m-S_n)^2}=\sqrt{\frac{\varepsilon g}{2}[(A_{mc}-A_{nc})]^2+(A_{ms}-A_{ns})]^2]}$$

定义解析：

QAM 信号可以看成是幅度和相位的联合调制，决定了信号星座点在 I/Q 平面的位置。另外，可以考虑成 M1 个电平的 PAM 信号（I 路）与 M2 个电平的 PAM 信号（Q 路）联合构成的信号，也可以考虑成 M1 个电平的 PAM 信号与 M2 个相位 PSK 信号的组合。实际应用中，前者处理相对简单，若 M1 =2^n, M2 =2^m,则信号的可能传输值共有 M1*M2=2^{m+n},可能的传输比特数为 m+n，符号速率为 R，比特速率为(m+n)R

二、 调制产生 QAM 基带信号

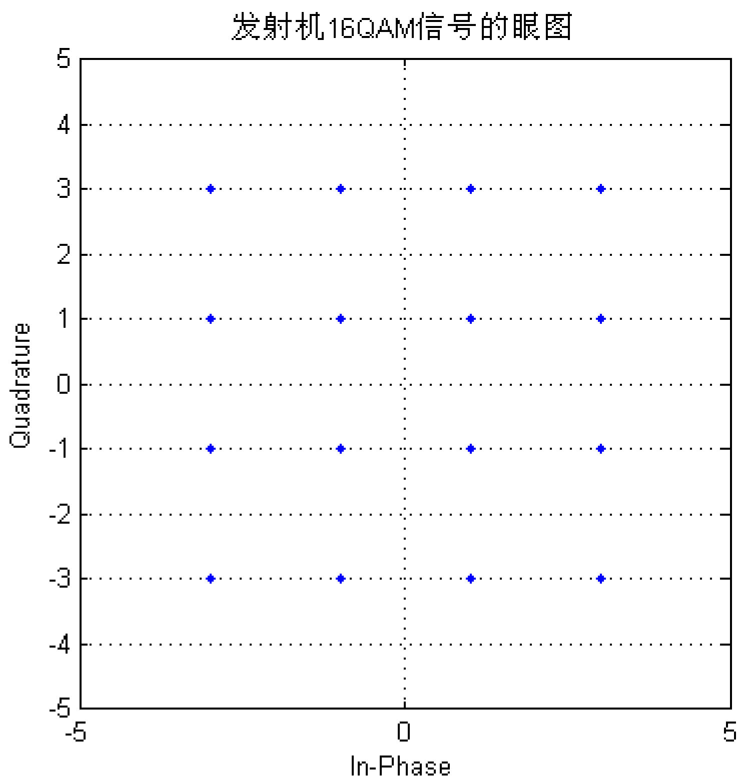
符号产生：

根据上面的星座图，单独观察 I 分量，”XXXX”中的第 1、3 位决定 PAM 信号的幅度值，同样的，单独观察 Q 分量，”XXXX”中的第 2、4 位决定 PAM 信号的幅度值。因此，由 bit 到符号的对应可以在 bit 流中四位一取，1、3 位进行 M=4 的 PAM 调制，2、4 位进行 M=4 的 PAM 调制，然后分别作为基带信号的 I、Q 路按照通用信号调制的流程生成基带发射信号。由于机理类似，且之前已经讨论过发射机的相关流程，本文不再赘述，仅给出仿真图。

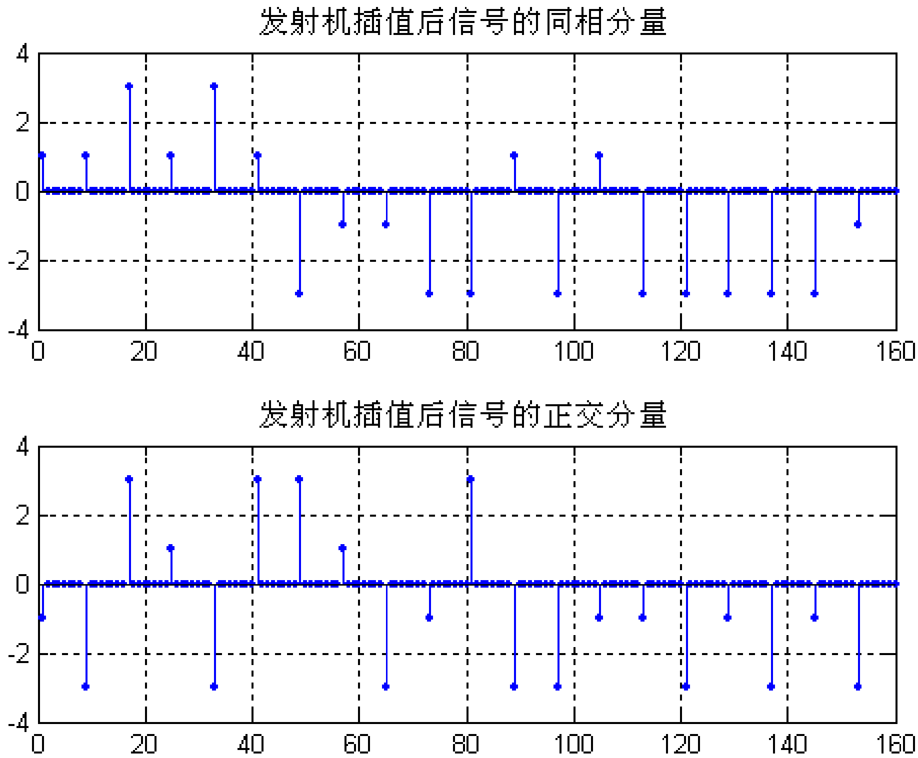
```
%调制
signal_base_band1I=message(1:4:end-3);
signal_base_band3I=message(3:4:end-1);
signal_base_band2Q=message(2:4:end-2);
signal_base_band4Q=message(4:4:end);
signal_base_bandI=zeros(1,SYM_LEN);
signal_base_bandQ=zeros(1,SYM_LEN);
for i=1:1:SYM_LEN
    tempI=[signal_base_band1I(i),signal_base_band3I(i)];
    if      tempI==[1,0]
             signal_base_bandI(i)=-3;
    elseif tempI==[1,1]
             signal_base_bandI(i)=-1;
    elseif tempI==[0,1]
```

```
            signal_base_bandI(i)=1;
    else
            signal_base_bandI(i)=3;
    end;
    tempQ=[signal_base_band2Q(i),signal_base_band4Q(i)];
    if      tempQ==[1,0]
            signal_base_bandQ(i)=-3;
    elseif tempQ==[1,1]
            signal_base_bandQ(i)=-1;
    elseif tempQ==[0,1]
            signal_base_bandQ(i)=1;
    else
            signal_base_bandQ(i)=3;
    end;
end;
```

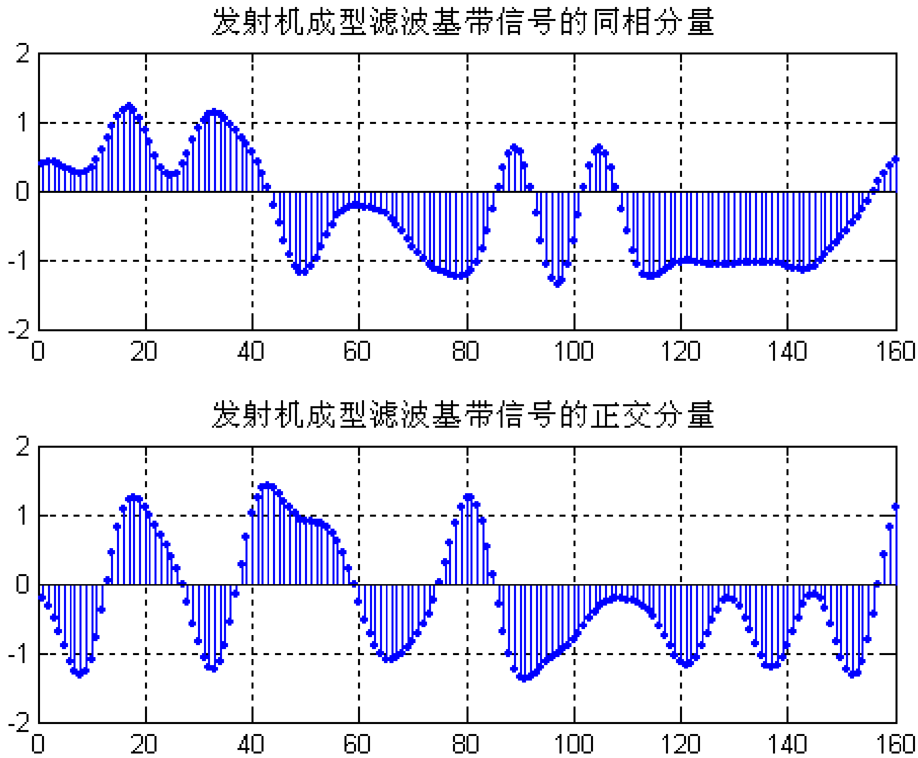
星座图：



插值：


发射机插值后信号的同相分量
4
2
0
-2
-4
0
20
40
60
80
100
120
140
160
发射机插值后信号的正交分量
4
2
0
-2
-4
0
20
40
60
80
100
120
140
160

成型滤波：



三、 AWGN 信道

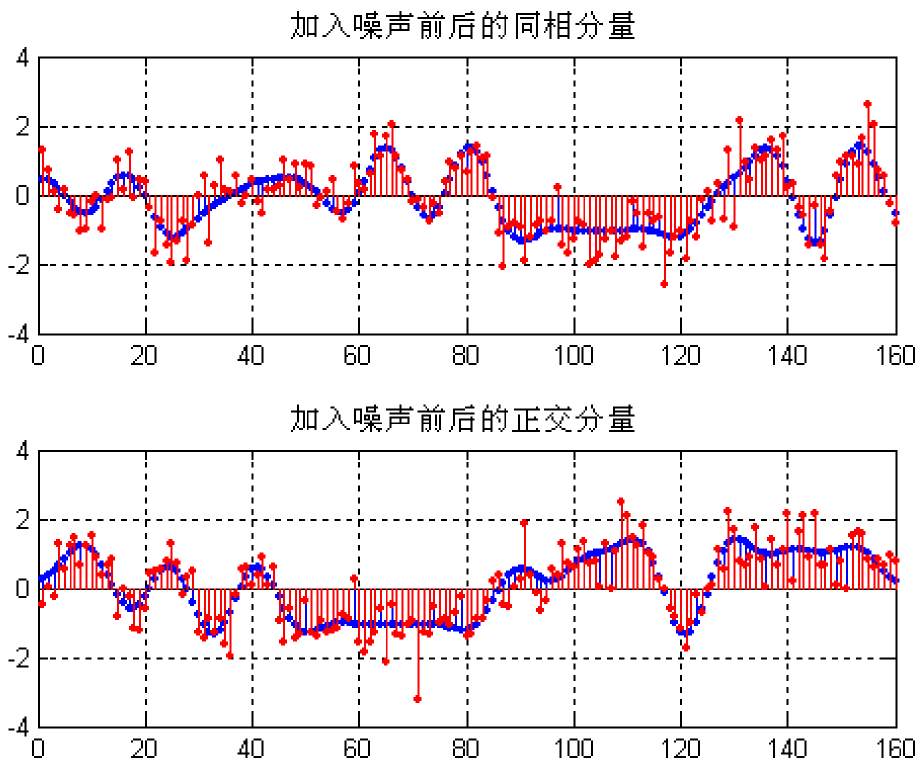
利用 Matlab 自带函数 awgn()在信道中加入高斯高噪声，函数 y=awgn(x,snr,'measured')表示：x 为输入序列，snr 为加入的信噪比，'measured'表示在加入噪声之前先对输入序列的功率进行测量。在利用 awgn()函数对发射信号加入噪声过程中需要考虑过采样的影响。注意到，awgn()在信道中加噪声是在每一个采样点在加入噪声，故采样前后的信噪比有如下的关系：

$$SNR_{采样后} = SNR_{采样前} - 10\lg(SampRate)$$

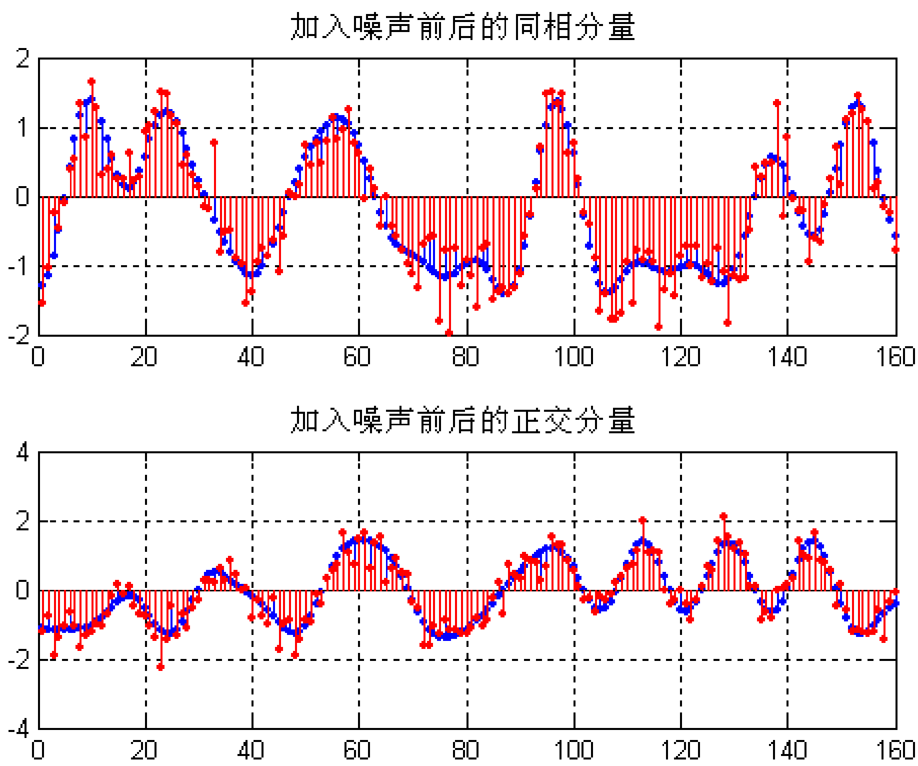
对于一般的 I、Q 两路调制信号，信号功率平均分配在 I 路和 Q 路，噪声功率同样平均分配在 I 路和 Q 路上，此时使用 awgn 对 I、Q 路分别操作时，每一路单独操作即可。仿真中可以设置的为符号信噪比。

```
signal_receiveI = awgn(signal_sendI,SNR-10*log10(INSERT_TIMES),'measured');
signal_receiveQ = awgn(signal_sendQ,SNR-10*log10(INSERT_TIMES),'measured');
```

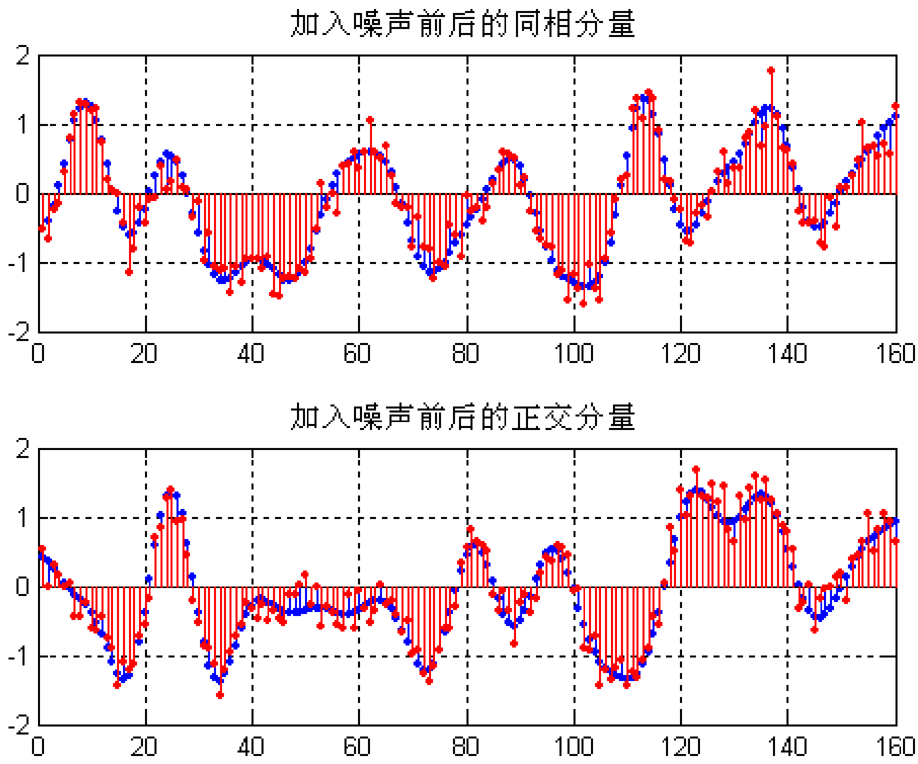
在符号 SNR 分别为 12dB、16dB、20dB 时，添加噪声前后波形对照如下，显然信噪比越小，信号的不确定性越大。



符号 SNR=12dB

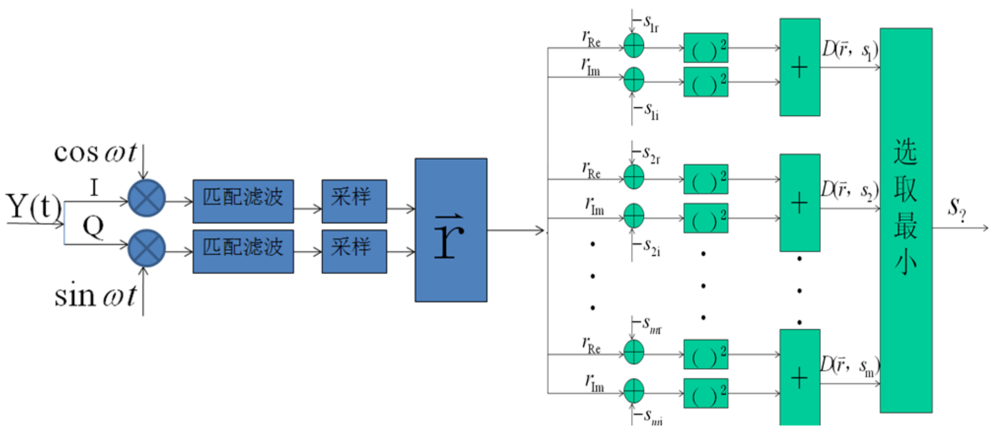


符号 SNR=16dB



符号 SNR=20dB

四、 接收机解调



匹配滤波：

发射机前端的低通滤波器实现了成型和频带限制的作用，一般用升余弦滤波器实现。为了在接收端实现最大比合并，得到最大的信噪比，在接收端使用匹配滤波器。故将一个升余弦滤波器分拆为两个平方根升余弦滤波器，两个平方根升余弦滤波器的卷积可以实现无码间

干扰。利用 Matlab 自带的 rcosflt()函数可实现滤波：

y = rcosflt(x,Fd,Fs,'filter_type/Fs',r,delay)

其中 y 为输出序列，x 为输入序列，Fd 为符号速率，Fs 为采样速率，filter_type 设置为 sqrt 表示平方根升余弦，'sqrt/Fs' 使得系统认为输入序列 x 已经具有 Fs 的采样速率，不需进行进一步的上采样，r 为滚降系数，delay 为滤波器一半的边瓣和主瓣数目（即若边瓣数目为 PETAL，则 delay 设置为 PETAL+1。

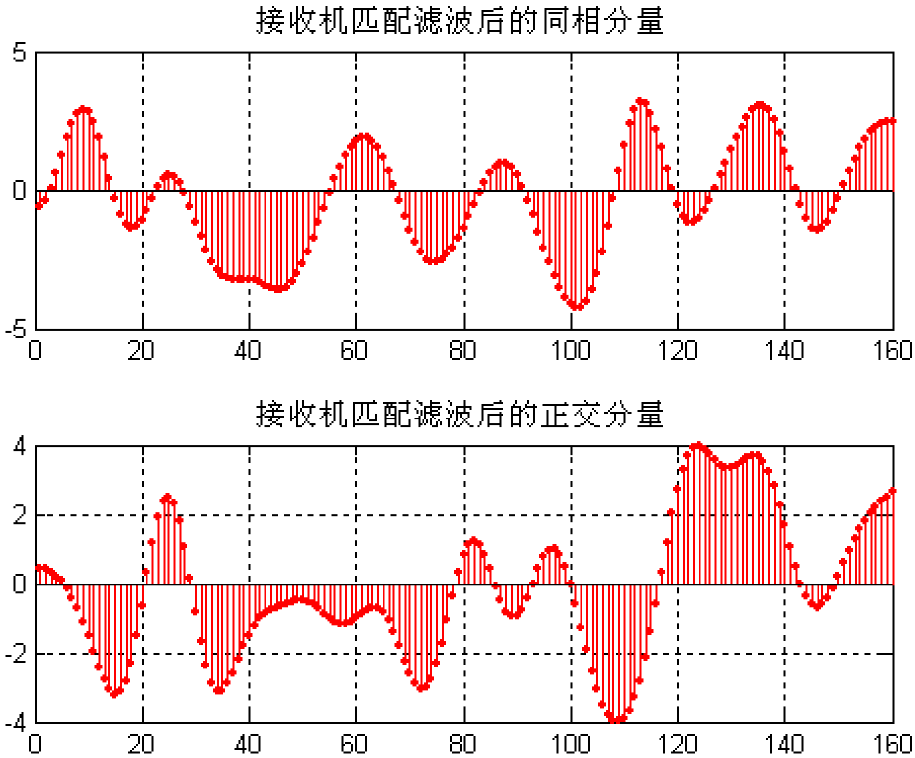
在滤波器类型的参数设置中，如果加入/Fs，那么：

length(y) = length(x)+2*delay*Fs/Fd

滤波器的长度都为 N=2*delay*Fs/Fd+1。这样，滤波导致的信号群延时为：

(N-1)/2=delay*Fs/Fd

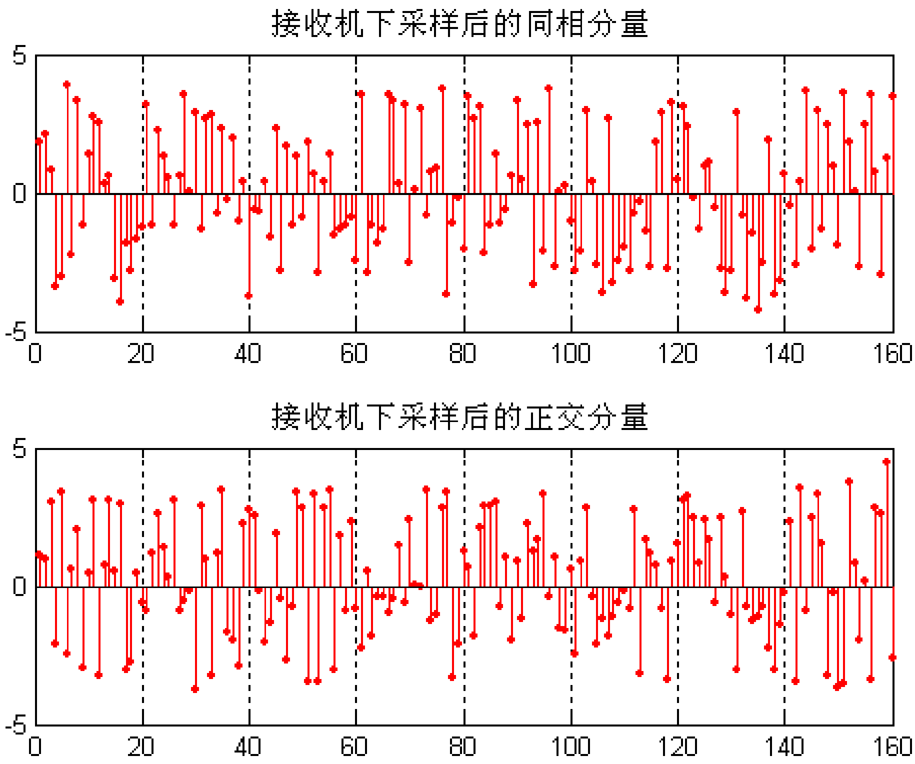
本次仿真中，取边瓣个数 PETAL=5，INSERT_TIMES=8 时，理论上信号群延时为(5+1)*2*8=96，实际仿真中观察也确实如此。



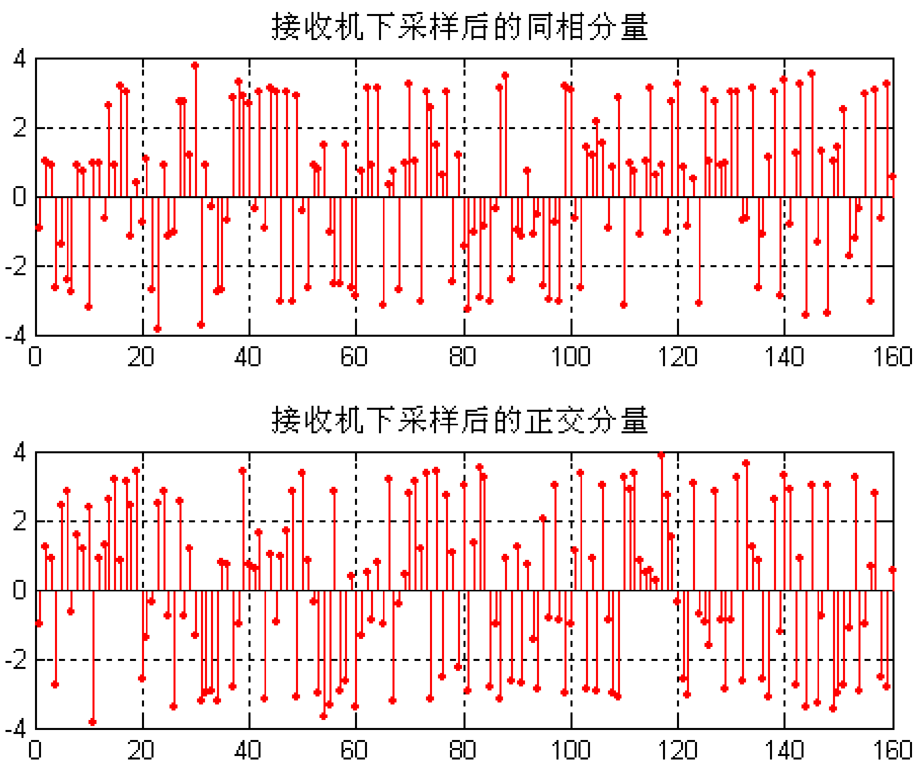
下采样：

滤波之后的信号下采样是为了信号的判决，根据匹配滤波理论，下采样时刻需要在信号功率最大处进行，仿真中由于确知接收信号的相位，因此保证这点不难，只需要根据上面滤波器模块中推导的信号延时来计算采样时刻即可。

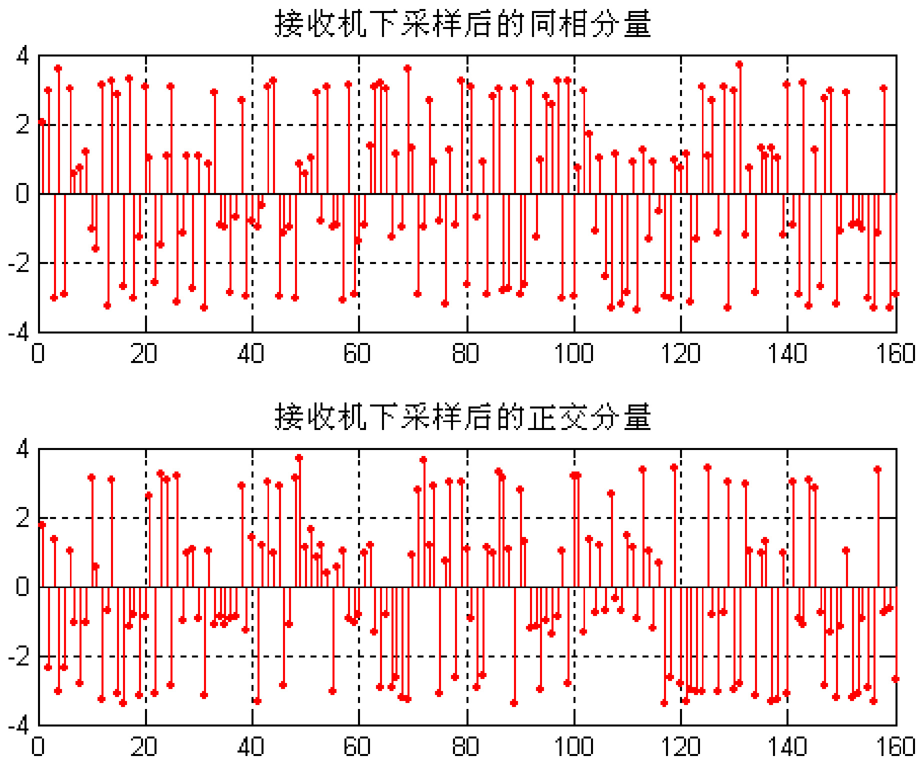
下面列出在符号信噪比分别为 12dB、16dB、20dB 时的下采样结果，由仿真图可以看出信噪比越小，下采样后的信号幅度波动越大，对后面的判决也有越不利。



符号 SNR=12dB



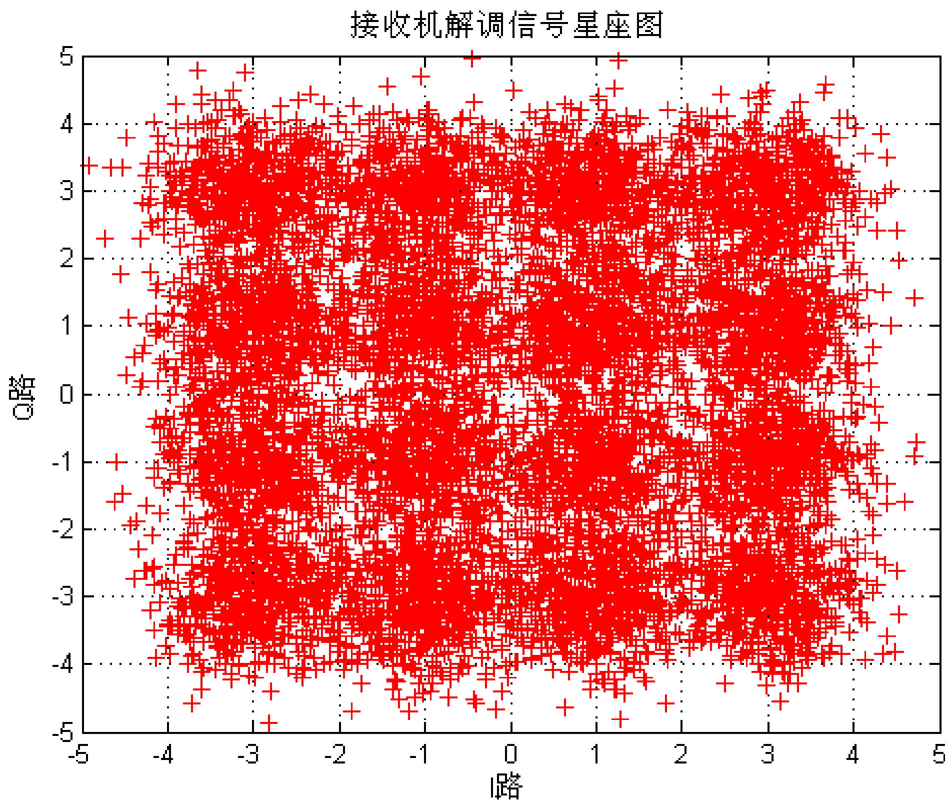
符号 SNR=16dB



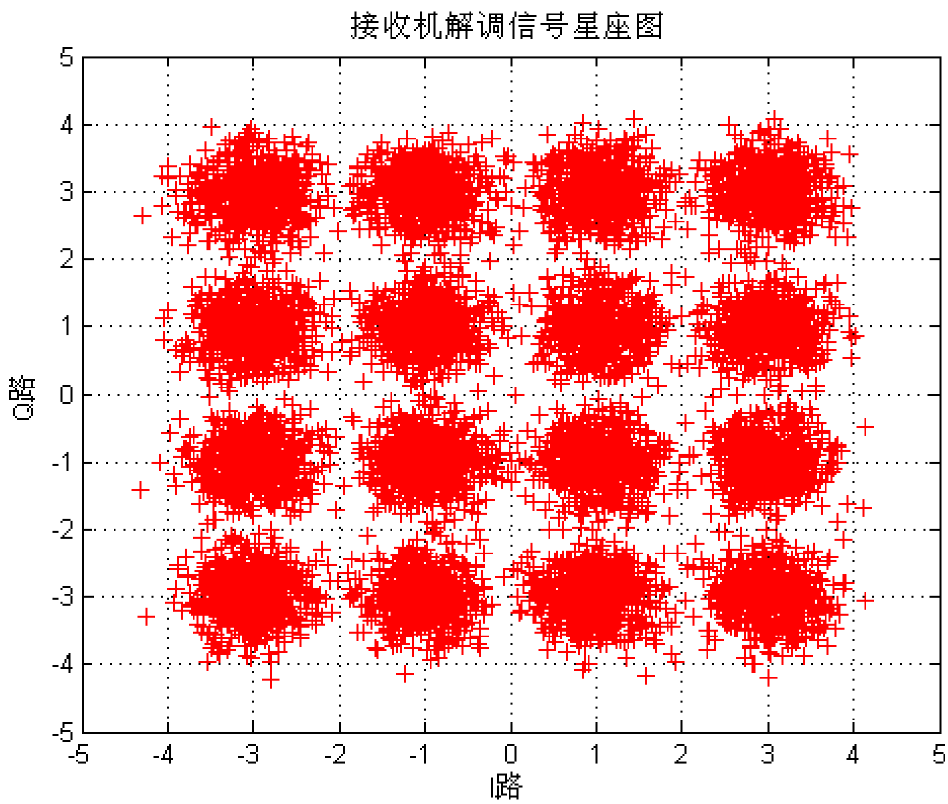
符号 SNR=20dB

星座图：

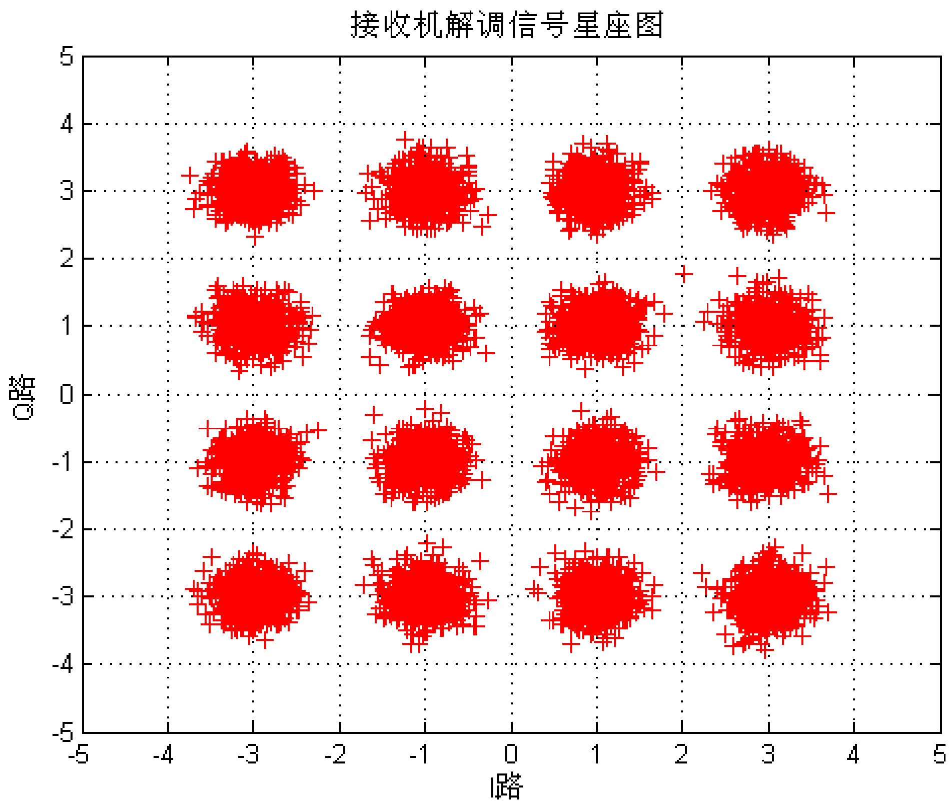
接收信号收到噪声的影响，星座图不再高度集中于 16 个点，而是在各自中心点周围符合高斯分布，符号信噪比越小，星座图越分散。



符号 SNR=12dB



符号 SNR=16dB



符号 SNR=20dB

符号判决：

通信系统中，一般采用最大似然准则（maximum likelihood, ML），根据 $p(r|s_m)$ 进行判决。假设噪声为零均值高斯分布的，则对于每个正交分量上，有：

$$p(r_k|s_{mk})=\frac{1}{\sqrt{\pi N_0}}\exp\left(-\frac{(r_k-s_{mk})^2}{N_0}\right),\quad k=1,2,\ldots N$$

上式是一个一维高斯分布。对于多维信号，共有 N 个基函数，似然函数为 N 维的高斯分布：

$$p(\mathbf{r}|s_m)=\frac{1}{(\pi N_0)^{N/2}}\exp\left(-\sum_{k=1}^{N}\frac{(r_k-s_{mk})^2}{N_0}\right),\quad m=1,2,..M$$

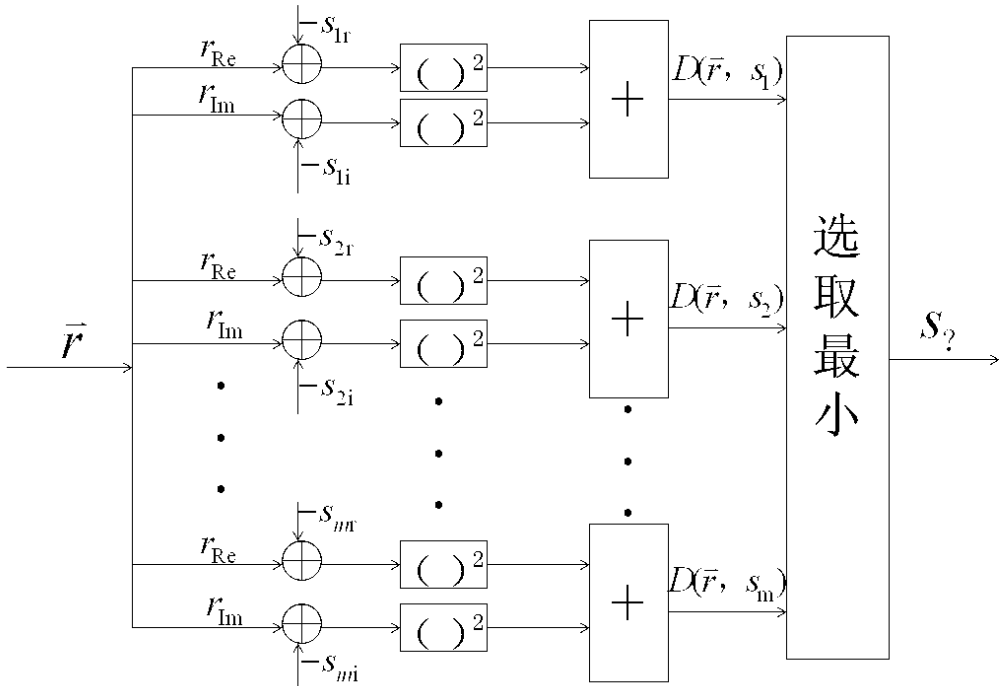
ML 准则就是要使此似然函数最大化。
对似然函数取对数，得到对数似然函数：

$$\ln(p(\mathbf{r}|\mathbf{s}_m))=-\frac{1}{2}N\ln(\pi N_0)-\frac{1}{N_0}\sum_{k=1}^{N}(r_k-s_{mk})^2$$

由于第一项为常数，使上式最大化等效于使上式中的第二项最小化，也就是：

$$D(\mathbf{r},\mathbf{s_m})=\sum_{k=1}^{N}(r_k-s_{mk})^2$$

D(r,s_m)：是一种距离度量。基于 ML 的判决准则也叫做最小距离准则（最小欧几里得距离）。



本文中，代码实现中选取最小值的方法较简单，只需要考察信号所在的区间，然后判定星座图中与接收信号星座点 I 分量和 Q 分量距离均为最小的那点即可。

```
%符号判决
mysignal_base_bandI=zeros(1,SYM_LEN);
mysignal_base_bandQ=zeros(1,SYM_LEN);
for i=1:1:SYM_LEN
    if signal_after_downsampleI(i)>=2
        mysignal_base_bandI(i)=3;
    elseif signal_after_downsampleI(i)>=0
        mysignal_base_bandI(i)=1;
    elseif signal_after_downsampleI(i)>=-2
        mysignal_base_bandI(i)=-1;
    else
        mysignal_base_bandI(i)=-3;
    end;
    if signal_after_downsampleQ(i)>=2
        mysignal_base_bandQ(i)=3;
    elseif signal_after_downsampleQ(i)>=0
        mysignal_base_bandQ(i)=1;
    elseif signal_after_downsampleQ(i)>=-2
        mysignal_base_bandQ(i)=-1;
    else
        mysignal_base_bandQ(i)=-3;
    end;
end;
```

符号到比特的变换：

只是比特到符号变换的逆过程，按照星座图设计一一对应取值。

```
%符号到比特变换
mymessage=zeros(1,MES_LEN);
for i=1:1:SYM_LEN
    tempS=[mysignal_base_bandI(i),mysignal_base_bandQ(i)];
    tempB=zeros(1,4);
    if       tempS==          [-3,-3]
            tempB=[1,1,0,0];
    elseif tempS==          [-3,-1]
            tempB=[1,1,0,1];
    elseif tempS==          [-3,  1]
            tempB=[1,0,0,1];
    elseif tempS==          [-3,  3]
            tempB=[1,0,0,0];
    elseif tempS==          [-1,-3]
            tempB=[1,1,1,0];
    elseif tempS==          [-1,-1]
            tempB=[1,1,1,1];
    elseif tempS==          [-1,  1]
            tempB=[1,0,1,1];
    elseif tempS==          [-1,  3]
            tempB=[1,0,1,0];
    elseif tempS==          [ 1,-3]
            tempB=[0,1,1,0];
    elseif tempS==          [ 1,-1]
            tempB=[0,1,1,1];
    elseif tempS==          [ 1,  1]
            tempB=[0,0,1,1];
    elseif tempS==          [ 1,  3]
            tempB=[0,0,1,0];
    elseif tempS==          [ 3,-3]
            tempB=[0,1,0,0];
    elseif tempS==          [ 3,-1]
            tempB=[0,1,0,1];
    elseif tempS==          [ 3,  1]
            tempB=[0,0,0,1];
    elseif tempS==          [ 3,  3]
            tempB=[0,0,0,0];
    else
            null;
    end;
```

```
    mymessage((i-1)*4+1:i*4)=tempB;
end;
```

符码率分析：

- **理论计算：**

QAM 信号的误符号率可由 PAM 信号得到，一个 M 进制的 QAM 信号可以表示成两个进制的 PAM 信号。正确接收的条件是要求两个 PAM 信号都正确接收：

$$P_c = \left(1 - P_{\sqrt{M}}\right)^2$$

其中：

$$P_{\sqrt{M}} = 2\left(1 - \frac{1}{\sqrt{M}}\right)Q\left(\sqrt{\frac{3}{M-1}\frac{\varepsilon_{av}}{N_0}}\right)$$

最终：

$$\mathrm{P_M} = 1 - P_c$$

考虑到 Q 函数与 erfc 函数的对应关系：

$$Q(x) = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right)$$

Matlab 代码实现为：

```
M=16;
X=sqrt(3/(M-1)*10.^((SNR)/10));
P_sqrM=2*(1-1/sqrt(M))*1/2*erfc(X/sqrt(2));
PM=1-(1-P_sqrM).^2;
```

- **误码分析：**

首先清楚，上面推导的都是在符号信噪比已知的情况下得到的误符号率的关系，而对于 $M=2^k$ 元的 QAM 调制来说，符号信噪比与比特信噪比的关系等效的定义为：

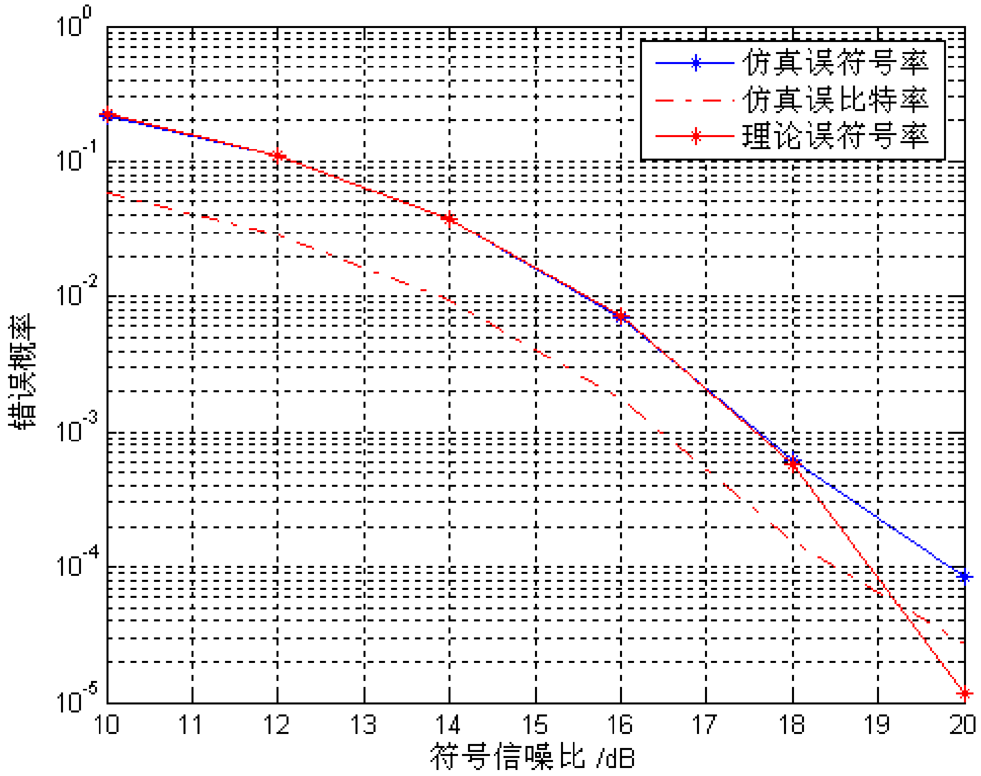
$$SNR_{sys} = k \times SNR_{bit}$$

由于符号信噪比和比特信噪比的确定对应关系，本次仿真中提到的信噪比如无特殊说明均为符号信噪比。误码率绘制代码如下：

```
%模式2：误码曲线的绘制
SNR = [10 12 14 16 18 20]; %符号SNR值
SumBit = zeros(1,length(SNR));
SumErrBit = zeros(1,length(SNR));
SumSym = zeros(1,length(SNR));
SumErrSym = zeros(1,length(SNR));
h=waitbar(0,'正在绘制，请稍候 …… ');
for k = 1:length(SNR)
    SumSym(1,k) = 0;
    SumErrSym(1,k) = 0;
```

```
    SumBit(1,k) =0;
    SumErrBit(1,k) =0;
    while(SumErrSym(1,k)<100 && SumSym(1,k)<SYM_LEN*50) %出现100个误符号 或者 仿真
        点数太多时停止
        function产生调制信号
         function加高斯白噪声
         function接收机解调
         %误码率统计
             ErrBit = MES_LEN-sum((mymessage == message));
             ErrSym = SYM_LEN-sum((mysignal_base_band == signal_base_band));
              SumErrBit(1,k) = SumErrBit(1,k) + ErrBit;
              SumErrSym(1,k) = SumErrSym(1,k) + ErrSym;
              SumBit(1,k) = SumBit(1,k) +MES_LEN;
              SumSym(1,k) = SumSym(1,k) +SYM_LEN;
      End
      waitbar(k/length(SNR),h);
end
close(h);
MyErrBitRate = SumErrBit./SumBit      %仿真得到的误比特率
MyErrSymRate = SumErrSym./SumSym      %仿真得到的误符号率
TherSer=SER_16QAM(SNR);    %理论误符号率
%绘制误符号率和误比特率曲线
figure;
semilogy(SNR, MyErrSymRate, 'b-*');
hold on;
semilogy(SNR, MyErrBitRate, 'r-.');
hold on;
semilogy(SNR, TherSer, 'r-*');
legend('仿真误符号率','仿真误比特率','理论误符号率');
xlabel('符号信噪比 /dB');
ylabel('错误概率');
grid on;
```

其误码率曲线为：



由图可以看出，仿真误符号率与理论误符号率在低信噪比的情况下吻合较好，在高信噪比情况下出现偏差，这是由仿真点数有限所影响的，仿真中发先点数越多，在高信噪比情况下越吻合。

另外，仿真中发现误比特率大约是误比特率的 1/4，这是因为加的是高斯白噪声，因此符号错误几乎全部出现在相邻符号之间，而星座图采用格雷编码，相邻符号之间只有 1bit 不同，因此误比特率理论上为误符号率的大约 1/4 大小。