



東南大學
SOUTHEAST UNIVERSITY

数字通信的计算机仿真 (研讨)

10 载波调制技术(上)——等效基带仿真

讲解人: 王俊波

E-mail: jbwang@seu.edu.cn

Phone: 13770681926

QQ:308322767

主要内容

- 载波信号与等效低通信号
- 载波调制概述
- 脉冲幅度调制 (MPAM)
- 相移键控调制 (MPSK)
- 正交幅度调制 (MQAM)
- 频移键控调制 (MFSK)
- 差分编码相移键控调制(MDPSK)
- 平衰落信道下仿真性能评估

希尔伯特变换

- 希尔伯特 (Hilbert) 变换是一种构建某一已知函数的正交函数的变换，实函数 $f(t)$ 的希尔伯特变换

$$H(f(t)) \triangleq \hat{f}(t) = f(t) * \frac{1}{\pi t} = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{f(\tau)}{t - \tau} d\tau$$

- 希尔伯特变换的频率特性：等效于一个理想的相移器

$$\frac{1}{\pi t} \Leftrightarrow -j \operatorname{sgn}(\omega) = \begin{cases} -j & \omega > 0 \\ j & \omega \leq 0 \end{cases}$$

- ✓ 本质上，希尔伯特变换是一种实现 $\frac{\pi}{2}$ 移相的正交变换

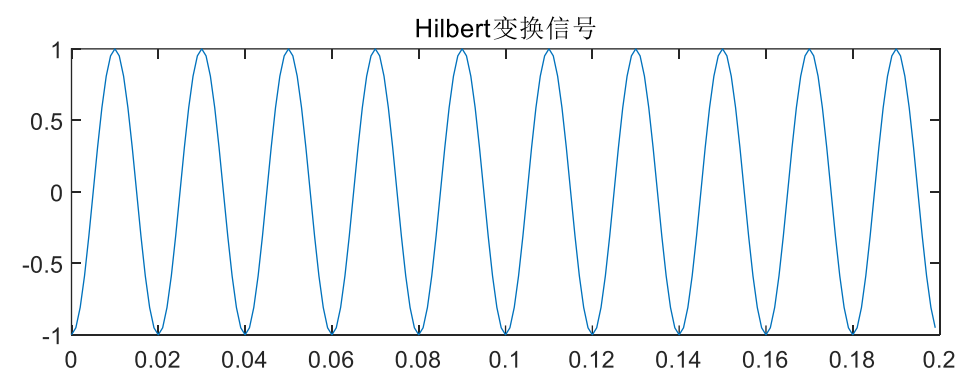
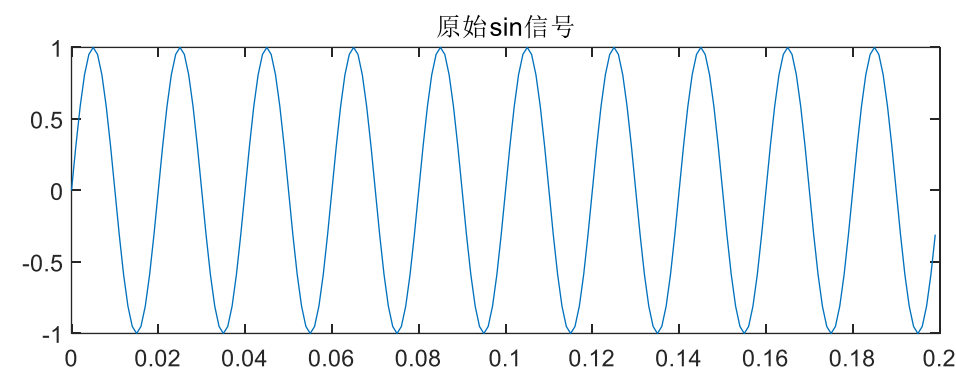
- 示例：正弦波的希尔伯特变换是余弦波

- ✓ 程序：simHilbertTransformSin.m

- ✓ 注意：matlab中的hilbert函数返回的是解析函数，不是hilbert变换，对变换取虚部得到hilbert变换结果

□ 程序: simHilbertTransformSin.m

```
02. ts = 0.001;
03. fs = 1/ts;
04. N = 200;
05. f = 50;
06. k = 0:N-1;
07. t = k*ts;
08. % 信号变换
09. % 结论: sin信号Hilbert变换后为cos信号
10. y = sin(2*pi*f*t);
11. yh = hilbert(y); % matlab函数得到信号是合成的复信号
12. yi = imag(yh); % 虚部为书上定义的Hilbert变换
13. figure
14. subplot(211)
15. plot(t, y)
16. title('原始sin信号')
17. subplot(212)
18. plot(t, yi)
19. title('Hilbert变换信号')
20. ylim([-1,1])
```



解析信号(分析信号)

□ 利用解析信号，可把带通信号转变为低通信号进行分析。

□ 定义实信号 $f(t)$ 的解析信号为

$$z(t) = f(t) + j\hat{f}(t)$$

✓ 其中 $\hat{f}(t)$ 为该实信号 $f(t)$ 的希尔伯特变换, $j = \sqrt{-1}$

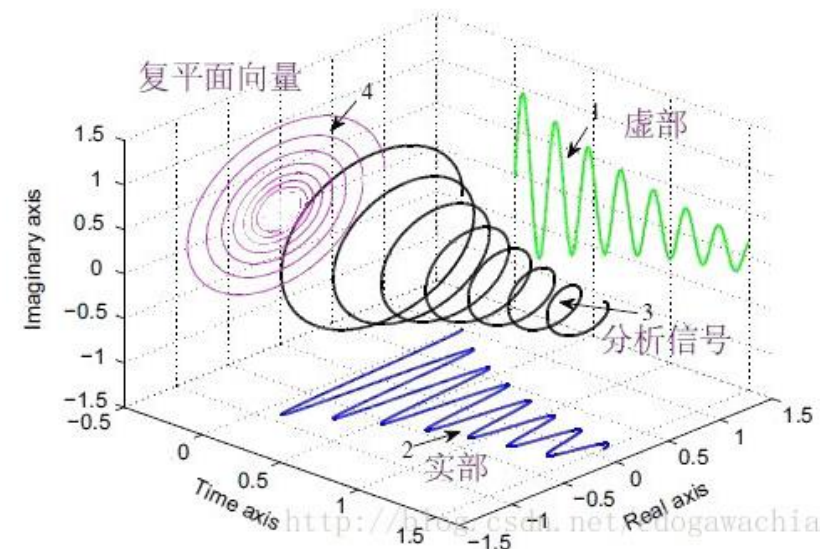
□ 若 $f(t) \Leftrightarrow F(\omega)$, 则 $z(t)$ 的傅里叶变换可以表示为

$$Z(\omega) = \begin{cases} 2F(\omega) & \omega > 0 \\ 0 & \omega \leq 0 \end{cases} = 2F(\omega)u(\omega)$$

✓ 证明:

$$\begin{aligned} Z(\omega) &= \mathcal{F}(z(t)) = \mathcal{F}(f(t) + j\hat{f}(t)) \\ &= \mathcal{F}(f(t)) + j\mathcal{F}(\hat{f}(t)) = F(\omega) + j[-j\text{sgn}(\omega)] \\ &= F(\omega) + \text{sgn}(\omega)F(\omega) = 2F(\omega)u(\omega) \end{aligned}$$

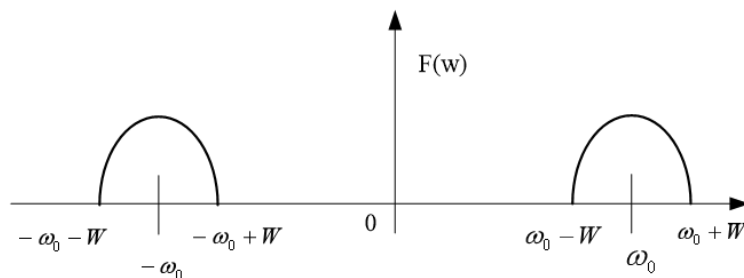
□ 信号包络: $\text{abs}(z(t))$



Matlab的函数hilbert(f), 返回的是实信号f的解析信号z, 若要取f的希尔伯特变换, 取z的虚部imag(z)

频带信号与窄带信号

- 频带信号(带通信号): 信号的频率分布集中在某一中心频率 ω_0 附近的信号
- 窄带信号: 假设频带信号带宽为 $2W$, 满足 $\omega_0 \gg 2W$ 的频带信号



频带信号的频谱分布

- 频带信号(带通信号) $f(t)$ 的解析信号

$$z(t) = f(t) + j\hat{f}(t) \Leftrightarrow 2F(\omega)u(\omega) = Z(\omega)$$

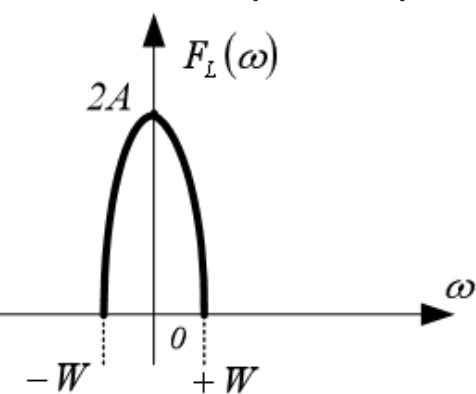
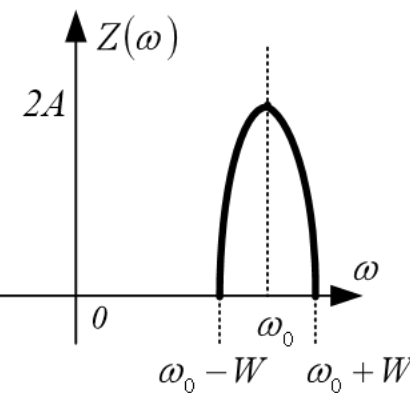
- 频带信号的复包络 (等效低通信号) 表示

$$f_L(t) = z(t)e^{-j\omega_0 t}$$

✓ 其满足

$$f_L(t) \Leftrightarrow F(\omega) = Z(\omega + \omega_0) = 2F(\omega + \omega_0)u(\omega + \omega_0)$$

注意: $z(t) = f_L(t)e^{j\omega_0 t}$



示例：希尔伯特变换、解析信号与等效低通信号

□ 已知信号 $x(t)=(1+0.5\cos(2\pi 5t))\cos(2\pi 50t+0.5\sin(2\pi 10t))$ ，绘制该信号的包络、信号的频谱、解析信号的频谱以及等效低通信号的频谱

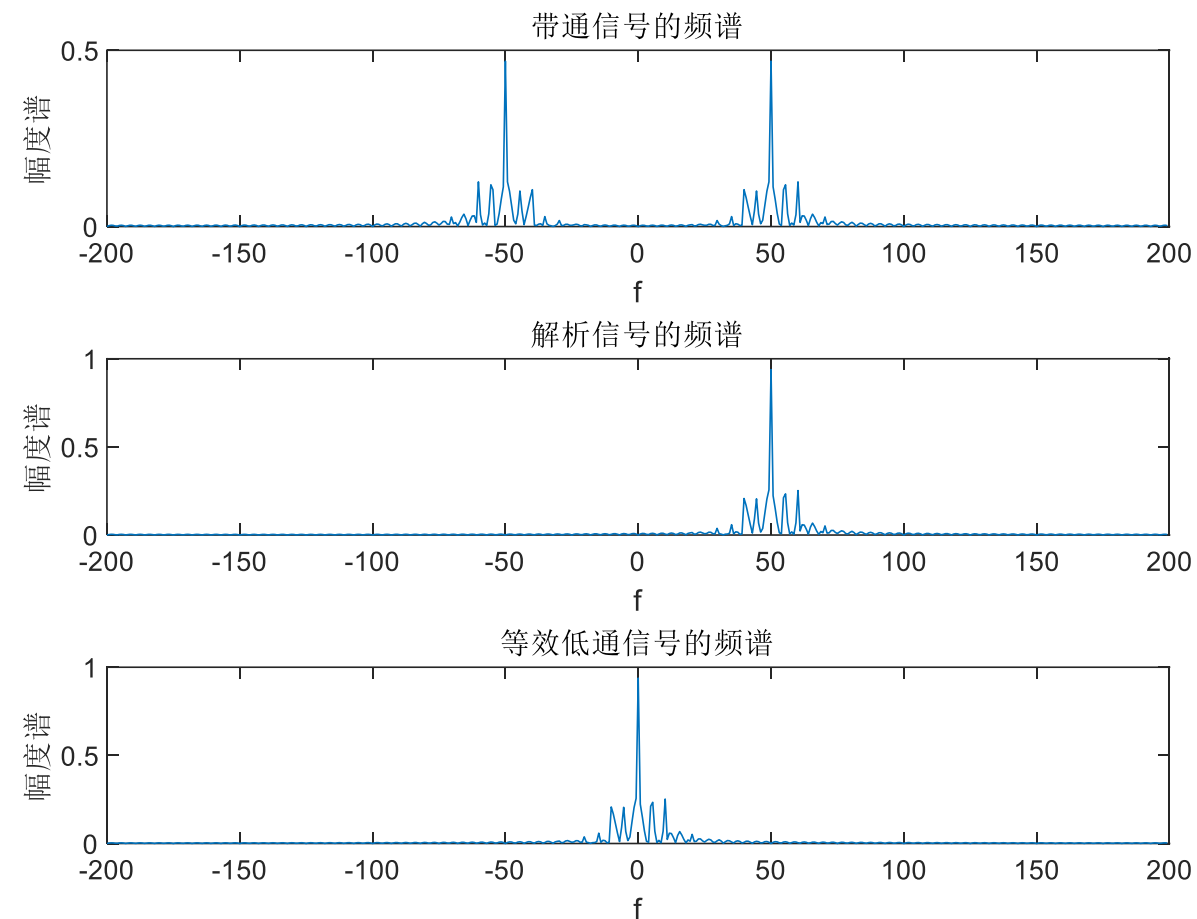
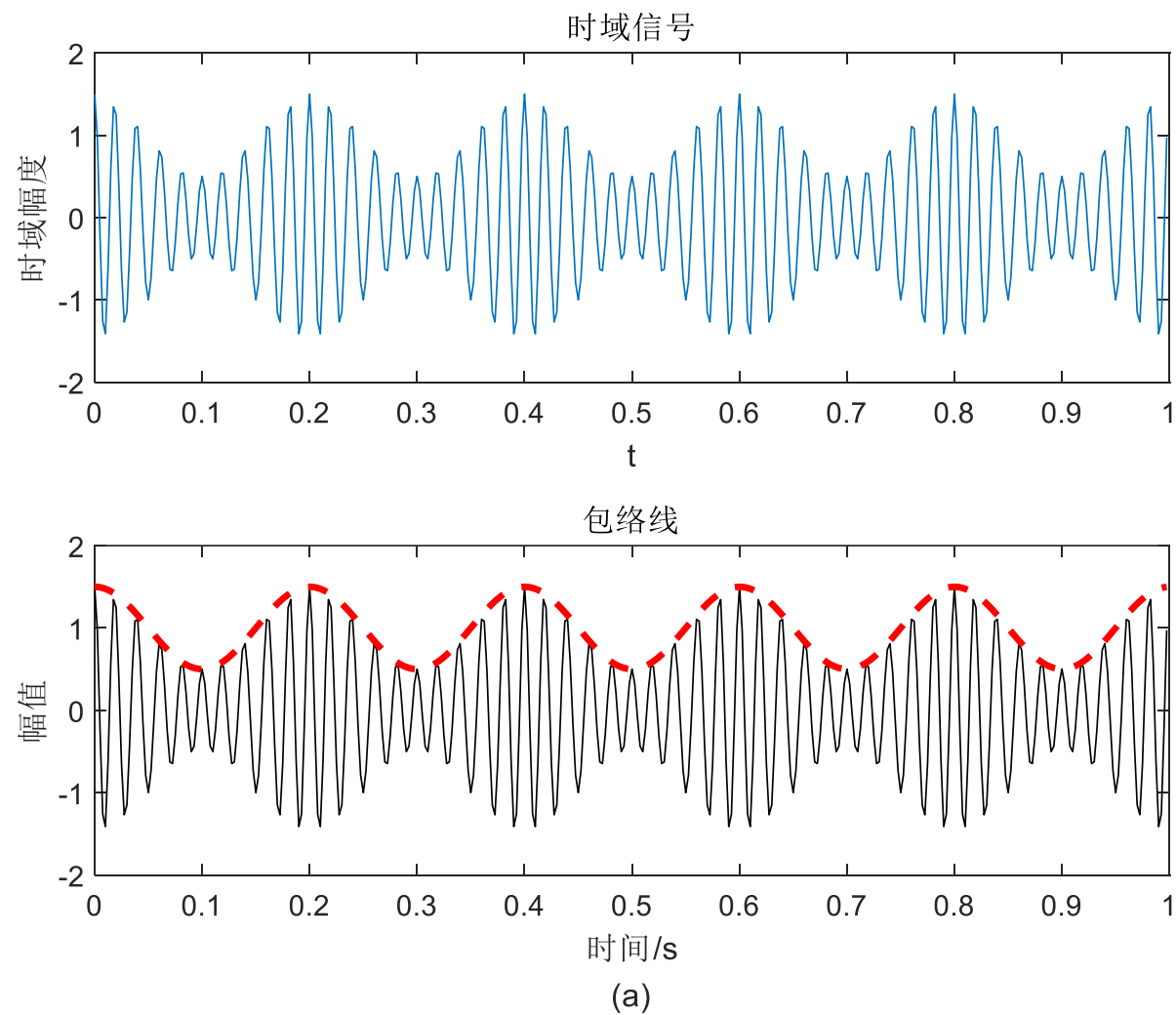
✓ 程序：simEquivalentLowPass.m

✓ 注意：matlab中的hilbert函数返回的是解析函数，不是hilbert变换

```
03. fs=400; % 采样频率
04. N=400; % 数据长度
05. n=0:1:N-1;
06. dt=1/fs;
07. t=n*dt; % 时间序列
08. A=0.5; % 相位调制幅值
09. x=(1+0.5*cos(2*pi*5*t)).*cos(2*pi*50*t+A*sin(2*pi*10*t)); % 信号序列
10. z=hilbert(x'); % 希尔伯特变换
11. a=abs(z); % 包络线
12. xL=z.*exp(-1i*2*pi*50*t); % 等效低通信号
13.
14. [Xf, f] = SpectrumViewer(x,fs, 'twosided'); %带通信号的频谱
15. [Zf, f] = SpectrumViewer(z,fs, 'twosided'); %解析信号的频谱
16. [XLf, f] = SpectrumViewer(xL,fs, 'twosided'); %等效低通信号的频谱
```

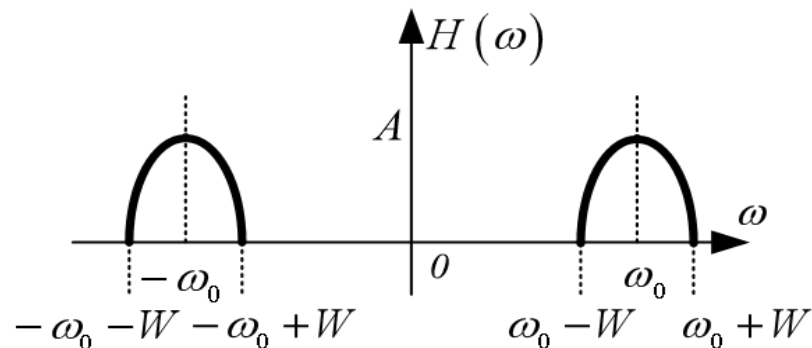
```
17. % 作图
18. figure
19. subplot(211)
20. plot(t, x);xlabel('t');ylabel('时域幅度');title('时域信号')
21. subplot(212)
22. pos = get(gcf, 'Position');
23. set(gcf, 'Position', [pos(1), pos(2)-100, pos(3), pos(4)]);
24. plot(t, x, 'k'); hold on;
25. plot(t, a, 'r--', 'linewidth', 2);
26. title('包络线'); ylabel('幅值'); xlabel(['时间/s' 10 '(a)']);
27. figure
28. subplot(311)
29. plot(f, abs(Xf));xlabel('f');ylabel('幅度谱');title('带通信号的频谱')
30. subplot(312)
31. plot(f, abs(Zf));xlabel('f');ylabel('幅度谱');title('解析信号的频谱')
32. subplot(313)
33. plot(f, abs(XLf));xlabel('f');ylabel('幅度谱');title('等效低通信号的频谱')
...
```

示例：绘制结果

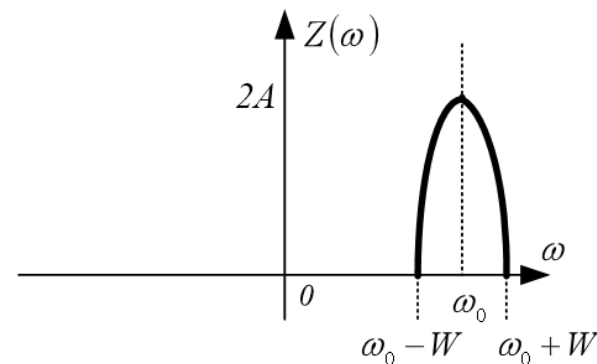


带通系统

□ 带通系统:通频带位于频谱的某一区域范围内的系统



带通信号的频谱



带通解析信号的频谱

✓ 系统的冲激响应和传递函数记为: $h(t) \leftrightarrow H(\omega)$

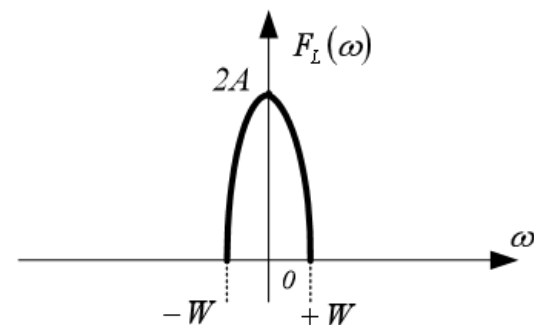
□ 带通系统的解析函数

$$z_h(t) = h(t) + j\hat{h}(t) \Leftrightarrow Z_h(\omega) = 2H(\omega)u(\omega)$$

✓ 注意: $z_h(t) = 2h_L(t)e^{j\omega_0 t}$

□ 带通系统的等效低通传递函数

$$h_L(t) = \frac{1}{2} z_h(t) e^{-j\omega_0 t} \Leftrightarrow H_L(\omega) = H(\omega + \omega_0) u(\omega + \omega_0)$$



带通等效低通信号的频谱

带通信号通过带通系统的低通分析方法

□ 带通信号 $f(t)$, 通过带通系统 $h(t)$, 输出为 $y(t)$

$$\begin{aligned}y(t) &= f(t) * h(t) = \frac{1}{2} [z_f(t) + z_f^*(t)] * \frac{1}{2} [z_h(t) + z_h^*(t)] \\&= \frac{1}{4} [z_f(t) * z_h(t) + z_f^*(t) * z_h(t) + z_f(t) * z_h^*(t) + z_f^*(t) * z_h^*(t)] \\&= \frac{1}{4} [z_f(t) * z_h(t) + z_f^*(t) * z_h^*(t)]\end{aligned}$$

□ 因为 $z_f(t) = f_L(t)e^{j\omega_0 t}$, $z_h(t) = 2h_L(t)e^{j\omega_0 t}$, 进一步推得

$$y(t) = \frac{1}{2} [(f_L(t) * h_L(t))e^{j\omega_0 t} + (f_L^*(t) * h_L^*(t))e^{-j\omega_0 t}]$$

□ 因为 $y(t) = \frac{1}{2} [z_y(t) + z_y^*(t)] = \frac{1}{2} [y_L(t)e^{j\omega_0 t} + y_L^*(t)e^{-j\omega_0 t}]$, 可以得到

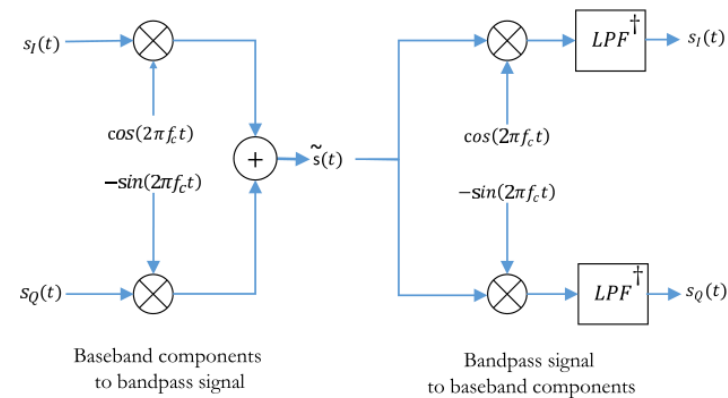
$$y_L(t) = f_L(t) * h_L(t)$$

□ 由 $y(t) = \text{Re}\{y_L(t)e^{j\omega_0 t}\}$ 可知, 带通信号通过带通系统, 等价于等效低通信号通过等效低通系统函数

调制信号的复基带表示

□ 通带信号是指其单边能量谱以非零载频 f_c 为中心而不扩展到DC的信号

□ 通带信号或任何数字调制的射频波形表示为



$$\tilde{s}(t) = a(t) \cos [2\pi f_c t + \phi(t)] = s_I(t) \cos(2\pi f_c t) - s_Q(t) \sin(2\pi f_c t)$$

✓ 其中

$$a(t) = \sqrt{s_I(t)^2 + s_Q(t)^2} \quad \phi(t) = \tan^{-1} \left(\frac{s_Q(t)}{s_I(t)} \right)$$

✓ 由于正弦和余弦项是相互正交的，信号可以用复数形式表示为

$$s(t) = s_I(t) + js_Q(t)$$

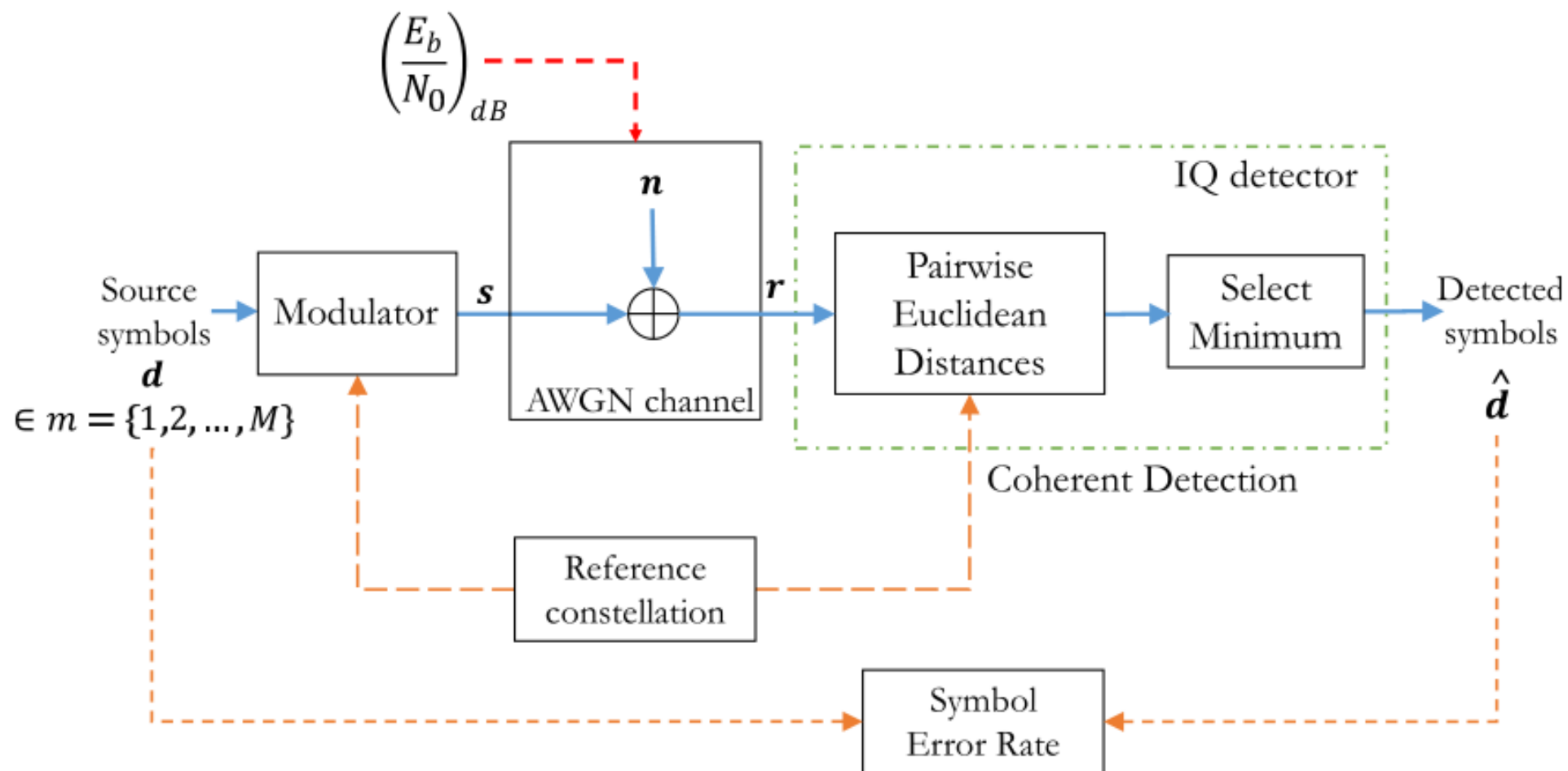
✓ 信号 $s(t)$ 称为复包络或实信号的复基带等效表示， $s_I(t)$ 和 $s_Q(t)$ 分别称为同相分量和正交分量

载波调制技术概述

- 在任何通信系统中，发射机通过将带有信息的基带波形调制到正弦射频载波上而工作，从而产生通带信号
 - ✓ 选择用于传输的载波频率因应用不同而不同
 - 例如：调频收音机使用88-108兆赫载波频率
 - ✓ 载波频率不是包含信息的分量，而是包含正在传送的信息的基带信号
- 实际的射频传输是通过上变频将基带信号转换成通带信号开始的。类似地，在实际解调开始之前，通带信号在接收器处被下变频到基带。因此，通信系统仿真中存在两种基本模型：通带模型及其等效基带模型
 - ✓ 在通带模型（波形仿真模型），详细模拟了射频载波的每个周期，功率谱集中在载波频率 f_c 附近：通带模型需要消耗更多的内存，因为射频载波中的每个点都需要存储在计算机内存中进行模拟
 - ✓ 基带模型（等效基带仿真模型）：信号集中在零频率附近，射频载波被消除，模拟所需的样本数量大大减少，基带模型可以进一步简化系统仿真，通过基于符号采样时刻的采样来实现

相干检测性能仿真的通用仿真框架

□ 适用于MPAM、MPSK、MQAM



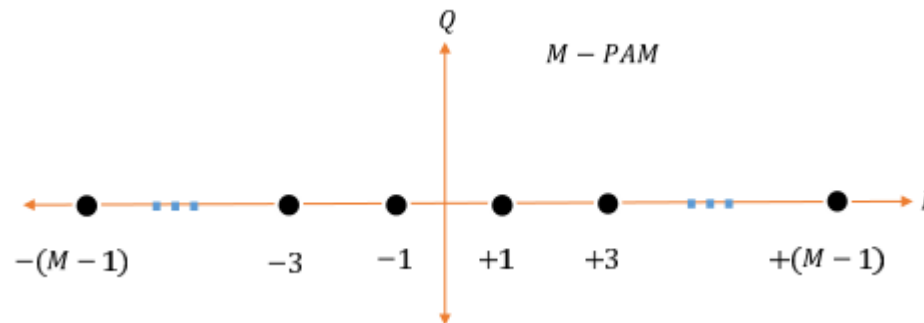
脉冲幅度调制 (MPAM)

□ MPAM调制是一种无正交分量的一维调制技术，所有的信息都被编码在信号幅度中

- ✓ MPAM调制器的离散基带模型是从集合 $m \in \{1, 2, \dots, M\}$ 中提取的一串信息符号进行传输，每个传输符号包含 k 位信息 ($k = \log_2(M)$)
- ✓ 生成M-PAM信号星座集的一般表达式如下

$$A_m = 2m - 1 - M, \quad m = 1, 2, \dots, M$$

- ✓ M是调制阶数，由每符号表示的比特数 k 决定
- ✓ 星座点位于 $\pm 1, \pm 3, \dots, \pm (M-1)$



```
01. function [s,ref]=mpam_modulator(M,d)
02. % 输入参数
03. % M 调制阶数
04. % d 输入符号序列，为行矢量
05. % 输出参数
06. % s 调制输出
07. % ref 参考星座点
08.
09. m=1:1:M;
10. Am=complex(2*m-1-M); %生成星座图
11. s = complex(Am(d)); %符号映射到对应的星座上
12. ref = Am;
13. end
```

接收检测技术

- 常用于检测数字调制数据的两大类检测技术是相干检测和非相干检测
 - ✓ 相干检测：发射机和接收机使用相同参考星座
 - ✓ 非相干检测：接收机不知道发射机使用的参考星座、初始相位信息，采用包络检测等方法对数据进行解调
- 相干检测：IQ检测技术
 - ✓ 第1步：计算参考星座点和被噪声破坏的接收符号之间的成对欧氏距离
 - ✓ 第2步：返回提供最小欧几里德距离的符号
 - ✓ 成对欧氏距离定义为

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_p - y_p)^2}$$

□ MPAM的SER

$$P_M = \frac{2(M-1)}{M} Q \left(\sqrt{\frac{6\mathcal{E}_{av}}{(M^2-1)N_0}} \right).$$

MPAM相干解调

□ 程序: mpam_detector.m, OptCoherentDetector.m

```
01. function Out= mpam_detector(M,r)
02. % 输入参数
03. % M 调制阶数
04. % r 接受的信号点, 为行向量
05. % 输出参数
06. % Out 解调输出
07. m=1:1:M;
08. Am=complex(2*m-1-M); %所有星座点
09. ref = Am;           %解调用的参考星座点
10. Out= OptCoherentDetector(r,ref); %最优相干检测
11. end
```

```
01. function idealPoints= OptCoherentDetector(received,ref)
02. %最优相干检测, 适用于MPAM,MPSK,MQAM和MFSK
03. %received - 接收到的复信号
04. %ref - 参考星座点
05. %idealPoints 解调输出符号
06.
07. [m,~] = size(received);
08. [n,~] = size(ref);
09.
10. if m==1 %输入序列为一位向量, 如MPAM,MPSK,MAQM,把信号构造成二维上的星座坐标
11.     x=[real(received); imag(received)]'; %接收信号的实部与虚部
12.     y=[real(ref); imag(ref)]';           %参考星座点
13. else %对于正交信号, 如MFSK, 本身已是一个多维信号
14.     x=received;
15.     y=ref;
16. end
17.
18.
19. X = sum(x.*x,2);
20. Y = sum(y.*y,2)';
21. d = X(:,ones(1,n)) + Y(ones(1,m),:) - 2*x*y'; %欧氏距的平方.
22. [~,idealPoints]=min(d,[],2); %寻找最小欧式距的符号
23. idealPoints=idealPoints.';
24. end
```


AWGN信道模型

□ 若发射信号为 x ，则接收信号表示为 $y = x + n$

✓ 其中 n 是均值为0，方差为 σ^2 的高斯随机变量

□ 给定信噪比 γ_{req} ，由输入信号能量，生成所需的噪声信号。若信号的过采样率为 L ，则发射信号功率可估算为

$$P_{\text{signal}} = L \times \frac{1}{N} \sum_{i=0}^{N-1} |x_i|^2$$

✓ 其中 N 为信号序列的长度

✓ 由 γ_{req} 可推得噪声功率为

$$P_{\text{noise}} = \frac{P_{\text{signal}}}{\gamma_{\text{req}}}$$

□ 若输入信号为复信号，则噪声为复高斯信号

$$n = n_c + jn_i \quad n_c \sim \mathcal{N}(0, P_{\text{noise}} / 2) \quad n_i \sim \mathcal{N}(0, P_{\text{noise}} / 2)$$

□ 若输入信号为实信号，则噪声为实高斯信号

$$n \sim \mathcal{N}(0, P_{\text{noise}})$$

AWGN信道生成

□ 程序: add_awgn_noise.m

```
01. function r = add_awgn_noise(s,SNRdB)
02. % 输入参数
03. %   s 输入符号序列
04. %   SNRdB 符号信噪比
05. % 输出参数
06. %   r 叠加噪声后的输出符号序列
07. s_temp=s;
08. if iscolumn(s), s=s.'; end
09. gamma = 10^(SNRdB/10); %SNR dB值转成线性值
10.
11. % 计算信号功率
12. if isvector(s) %对矢量输入信号, 如MPAM,MPSK,MQAM
13.     P=sum(abs(s).^2)/length(s);
14. else % 对多维输入信号, 如MFSK
15.     P=sum(sum(abs(s).^2))/length(s); %if s is a matrix [MxN]
16. end
17.
18. N0=P/gamma; % 计算噪声功率谱
19. %生成噪声
20. if(isreal(s))
21.     n = sqrt(N0)*randn(size(s));
22. else
23.     n = sqrt(N0/2)*(randn(size(s))+1i*randn(size(s)));
24. end
25. r = s + n; %接收信号叠加噪声
26. if iscolumn(s_temp), r=r.'; end
27. end
```

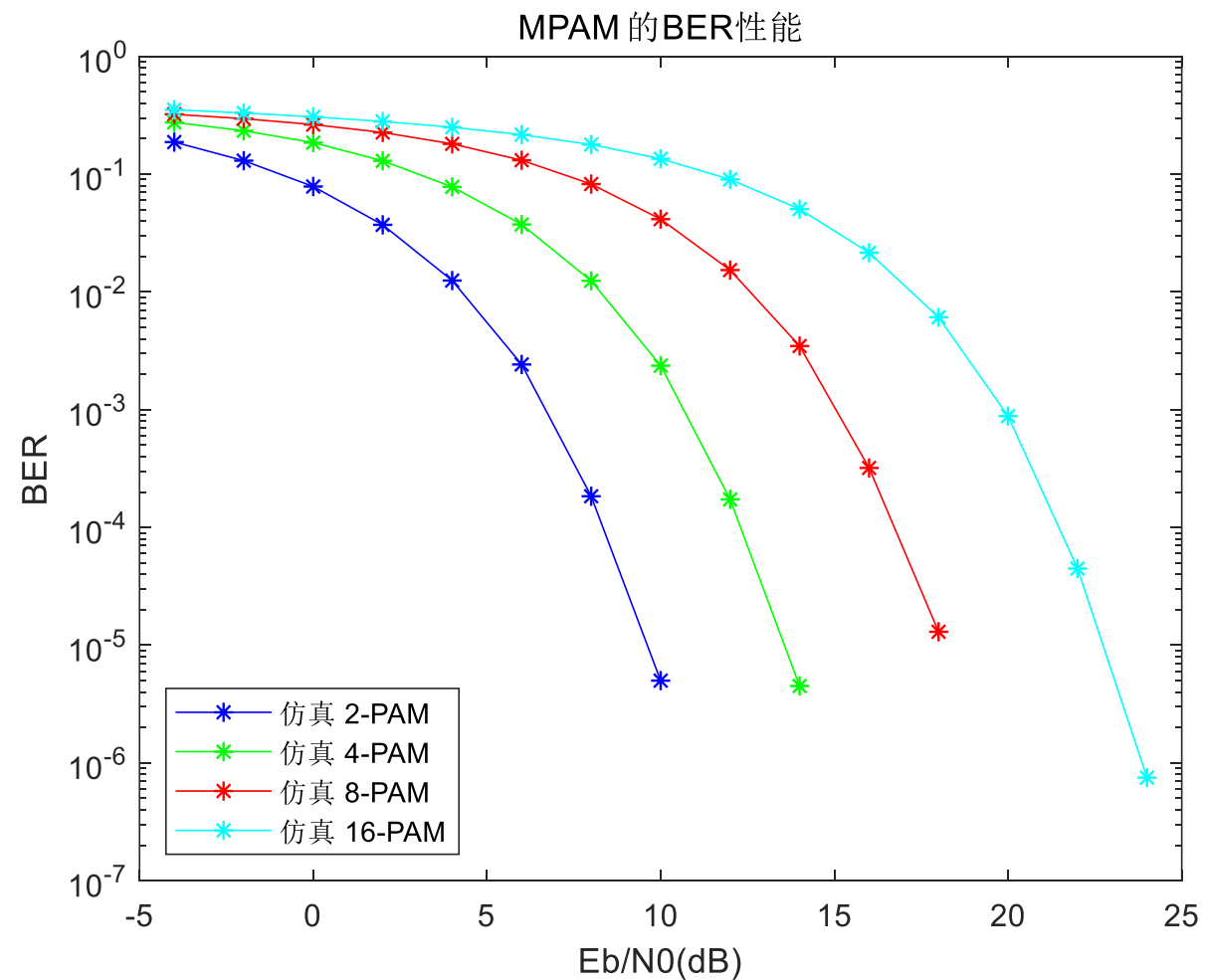
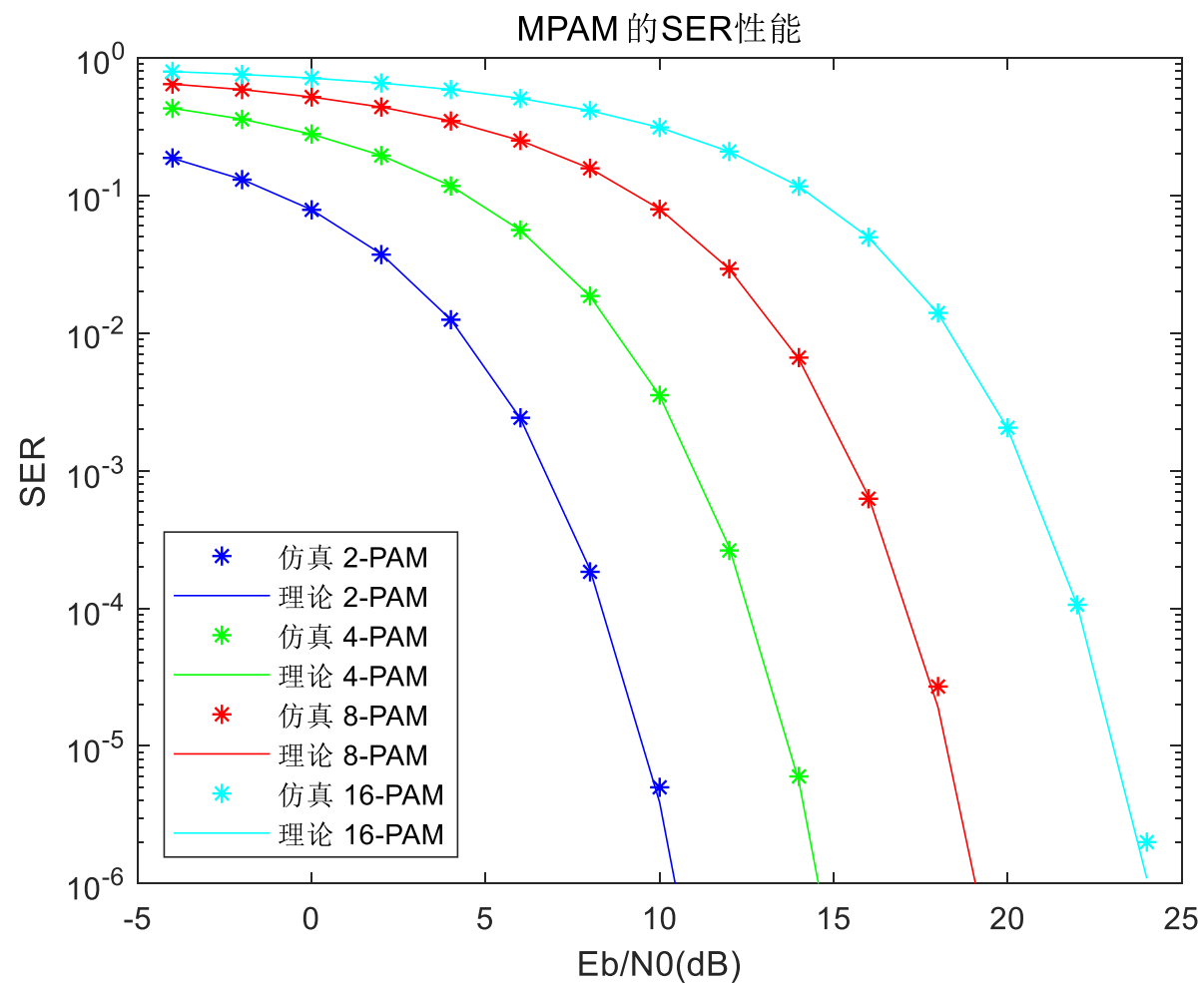
MPAM在AWGN信道下的性能

□ simMPAMoverAWGN.m

```
%% 参数设置
05. N=1e6; %符号数目
06. EbN0dB = -4:2:24; % Eb/N0[dB] 的仿真范围
07. arrayofM=[2 4 8 16]; %调制阶数
08.
09.
10. SER=zeros(length(arrayofM),length(EbN0dB));
11. BER=zeros(length(arrayofM),length(EbN0dB));
12. theoreticalSER=zeros(length(arrayofM),length(EbN0dB));
```

```
%% 性能仿真
14. for i = 1:length(arrayofM)
15.     M=arrayofM(i);
16.     k=log2(M); %每符号的比特数目
17.     m=1:1:M;
18.     Am=2*m-1-M; % MPAM的星座点
19.     dataIn=randi([0,1],1,k*N); %发送的信息比特
20.     dataInMatrix = reshape(dataIn,length(dataIn)/k,k); %转成每行k个元素的矩阵
21.     SymbolIn= bi2de(dataInMatrix)+1; %转成整数, 即符号
22.     SymbolIn = reshape(SymbolIn,1,[]);
23.
24.     [s,ref]=mpam_modulator(M,SymbolIn);
25.
26.     EbN0lin = 10.^(EbN0dB/10);
27.     EsN0dB = 10*log10(k)+EbN0dB; %EsN0dB calculation
28.
29.     for j=1:length(EsN0dB)
30.         y = add_awgn_noise(s,EsN0dB(j));%通过AWGN信道
31.
32.         [SymbolOut]= mpam_detector(M,y);
33.         SER(i,j)=sum((SymbolIn~=SymbolOut))/N;%计算SER
34.         %理论SER: 性能公式为《Digital Communications》第五版 (4-3-5)
35.         theoreticalSER(i,j)=2*((M-1)/M)*qfn(sqrt(6*log2(M)/(M^2-1)*10.^(EbN0dB(j)/10)));
36.
37.         SymbolOut = reshape(SymbolOut,[],1)-1;
38.         dataOutMatrix = de2bi(SymbolOut,k);
39.         dataOut = reshape(dataOutMatrix,1,[]);
40.         BER(i,j)=sum((dataIn~=dataOut))/(N*k);%计算BER
41.     end
42. end
43. end
```

MPAM在AWGN信道下的性能

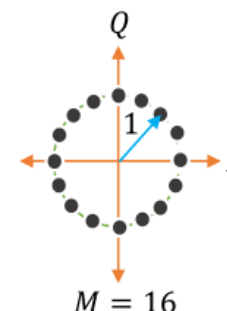
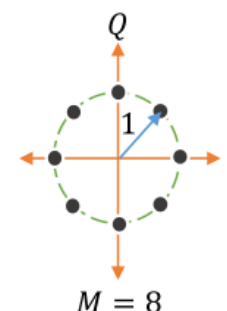
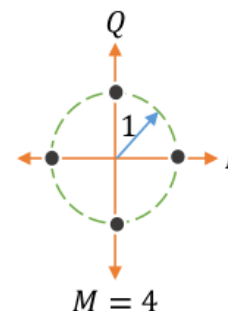


相移键控调制(MPSK)

□ 在相移键控中，所有的信息都被编码在载波信号的相位中

- ✓ MPAM调制器的离散基带模型是从集合 $m \in \{1, 2, \dots, M\}$ 中提取的一串信息符号进行传输，每个传输符号包含 k 位信息 ($k = \log_2(M)$)
- ✓ M-PSK信号集的一般表达式如下

$$s_m(t) = A \cos \left[2\pi f_c t - \frac{(m-1)2\pi}{M} \right], \quad m = 1, 2, \dots, M$$



- ✓ M是调制阶数，由每符号表示的比特数 k 决定

□ 利用三角恒等式，MPSK信号可分为余弦和正弦基函数

$$s_m(t) = A \cos \left[\frac{(m-1)2\pi}{M} \right] \cos(2\pi f_c t) + A \sin \left[\frac{(m-1)2\pi}{M} \right] \sin(2\pi f_c t), \quad m = 1, 2, \dots, M$$

- ✓ 也可表示为I-Q平面上的同相和正交相位分量的组合

$$s_m(t) = A \cos \left[\frac{(m-1)2\pi}{M} \right] + j A \sin \left[\frac{(m-1)2\pi}{M} \right], \quad m = 1, 2, \dots, M$$

- ✓ 振幅归一化为 $A = \sqrt{0.5}$

MPSK调制与解调仿真实现

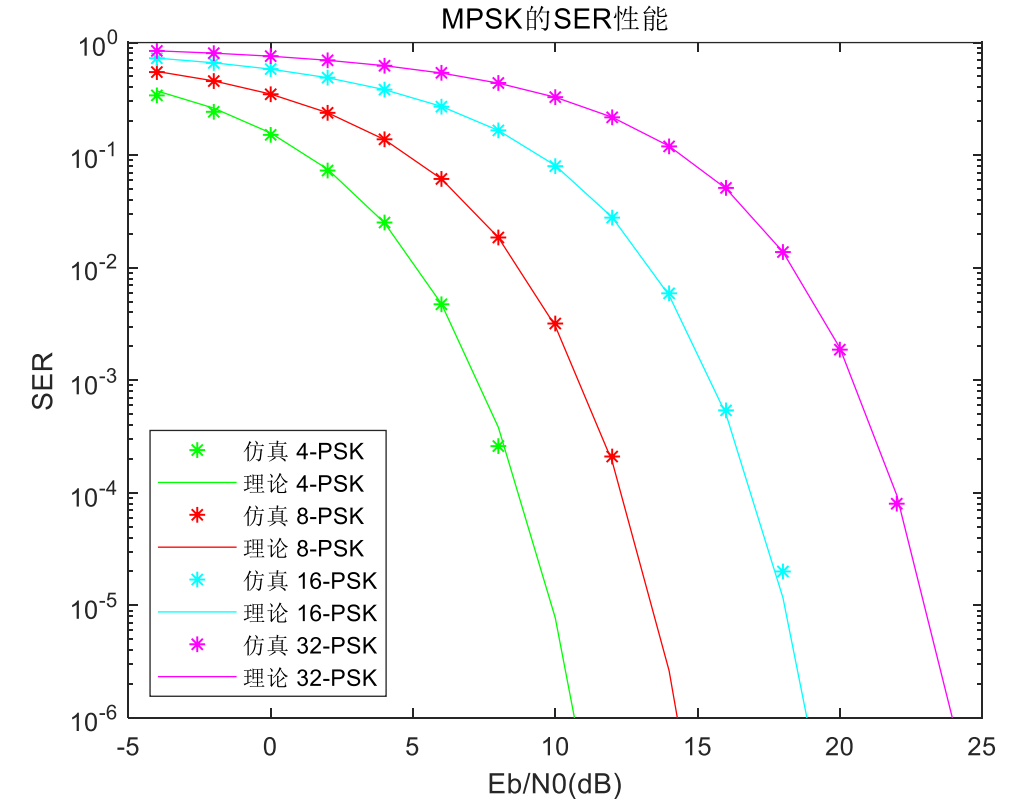
```
01. function [s,ref]=mpsk_modulator(M,d)
02. % 输入参数
03. % M 调制阶数
04. % d 输入符号序列，为行矢量
05. % 输出参数
06. % s 调制输出
07. % ref 参考星座点
08.
09. ref_i= 1/sqrt(2)*cos(((1:1:M)-1)/M*2*pi);
10. ref_q= 1/sqrt(2)*sin(((1:1:M)-1)/M*2*pi);
11. ref = ref_i+1i*ref_q; %生成星座图
12. s = ref(d); %符号影射到对应的星座上
13. end
```

```
01. function [Out]= mpsk_detector(M,r)
02. % 输入参数
03. % M 调制阶数
04. % r 接受的信号点，为行矢量
05. % 输出参数
06. % Out 解调输出
07. ref_i= 1/sqrt(2)*cos(((1:1:M)-1)/M*2*pi);
08. ref_q= 1/sqrt(2)*sin(((1:1:M)-1)/M*2*pi);
09. ref = ref_i+1i*ref_q; %解调用的参考星座点
10. Out= OptCoherentDetector(r,ref); %最优相干检测
11. end
```

MPSK在AWGN信道下的性能

□ simMPSKoverAWGN.m

```
05. %% 参数设置
06. nSym=1e6;           %符号数目
07. EbN0dB = 0:2:24;    % Eb/N0[dB] 的仿真范围
08. arrayOfM=[4,8,16,32]; %MPSK的阶数
09.
10. %% 仿真
11. SER_theory=zeros(length(arrayofM),length(EbN0dB));
12. SER_sim=zeros(length(arrayofM),length(EbN0dB));
13.
14. for i = 1:length(arrayofM)
15.     M=arrayofM(i);
16.     k=log2(M);
17.     EsN0dB = 10*log10(k)+EbN0dB; %计算Es/N0的dB值
18.     a=randi([1,M],1,nSym);%生成1~M的符号值
19.     [s,ref]=mpsk_modulator(M,a);
20.
21.     for j=1:length(EbN0dB)
22.         r = add_awgn_noise(s,EsN0dB(j));%通过AWGN信道
23.         acap= mpsk_detector(M,r);
24.         SER_sim(i,j)=sum((a~=acap))/nSym; %计算仿真SER
25.         %计算理论SER: 性能公式为《Digital Communications》第五版 (4-3-17)
26.         SER_theory(i,j) =2*qfn(sqrt(2*k*10.^(EbN0dB(j)/10))*sin(pi/M));
27.     end
28. end
```



正交幅度调制 (MQAM)

□ 在MQAM调制中，信息比特被编码为信号的幅度和相位的变化

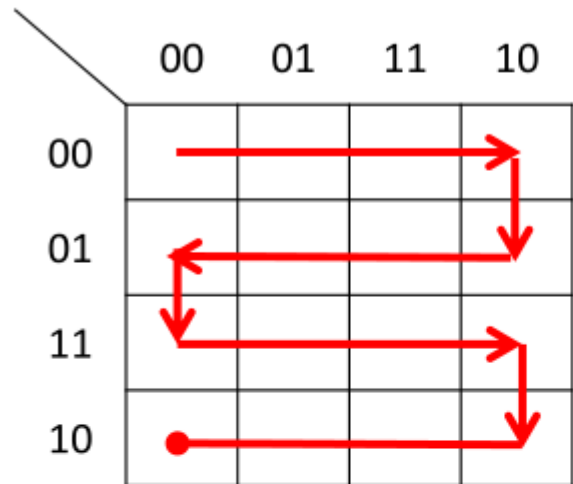
- ✓ MQAM调制器的离散基带模型是从集合 $m \in \{1, 2, \dots, M\}$ 中提取的一串信息符号进行传输，每个传输符号包含 k 位信息 ($k = \log_2(M)$)
- ✓ 为了将错误的接收器决策限制为单比特错误，信息符号被格雷 (Gray) 编码，然后使用矩形M-QAM技术对信息符号进行数字调制

$$s = a + jb \quad \text{where } a, b \in \{\pm 1, \pm 3, \dots, \pm(\lceil \sqrt{M} \rceil - 1)\}$$

□ 利用卡诺图生成Gray编码

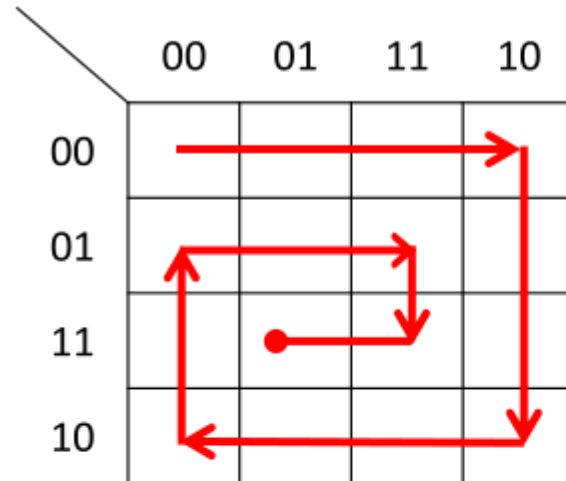
- ✓ 在卡诺图上走一圈，将产生一套Gray编码
- ✓ 循环Gray编码：最后一个码字与第一个码字只1位差异
- ✓ 非循环Gray编码：最后一个码字与第一个码字不只1位差异

循环和非循环Gray编码



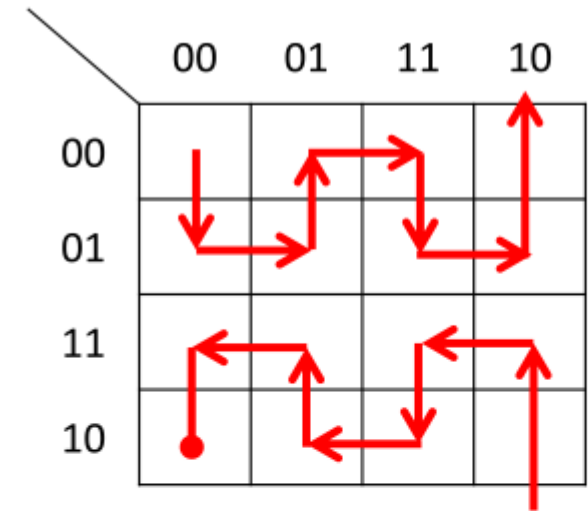
Walk type 1:
Cyclic Gray code possible

Binary	Dec
0000	0
0001	1
0011	3
0010	2
0110	6
0111	7
0101	5
0100	4
1100	12
1101	13
1111	15
1110	14
1010	10
1011	11
1001	9
1000	8



Walk type 2:
Cyclic Gray code **not** possible

Binary	Dec
0000	0
0001	1
0011	3
0010	2
0110	6
1110	14
1010	10
1011	11
1001	9
1000	8
1100	12
0100	4
0101	5
0111	7
1111	15
1101	13



Walk type 3:
Cyclic Gray code possible

Binary	Dec
0000	0
0100	4
0101	5
0001	1
0011	3
0111	7
0110	6
0010	2
1010	10
1110	14
1111	15
1011	11
1001	9
1101	13
1100	12
1000	8

格雷编码的仿真生成

- **n位的二进制格雷编码表可以通过翻转n - 1位编码表递归地生成**
 - ✓ 将原编码表中的码字加上二进制0前缀，将翻转编码表加上二进制1前缀
 - ✓ 然后将原编码表与翻转原编码表连接起来
 - ✓ **GrayCodeGen.m**

2-bit list:	00, 01, 11, 10
Reflected:	10, 11, 01, 00
Prefix old entries with 0:	000, 001, 011, 010,
Prefix new entries with 1:	110, 111, 101, 100
Concatenated:	000, 001, 011, 010, 110, 111, 101, 100

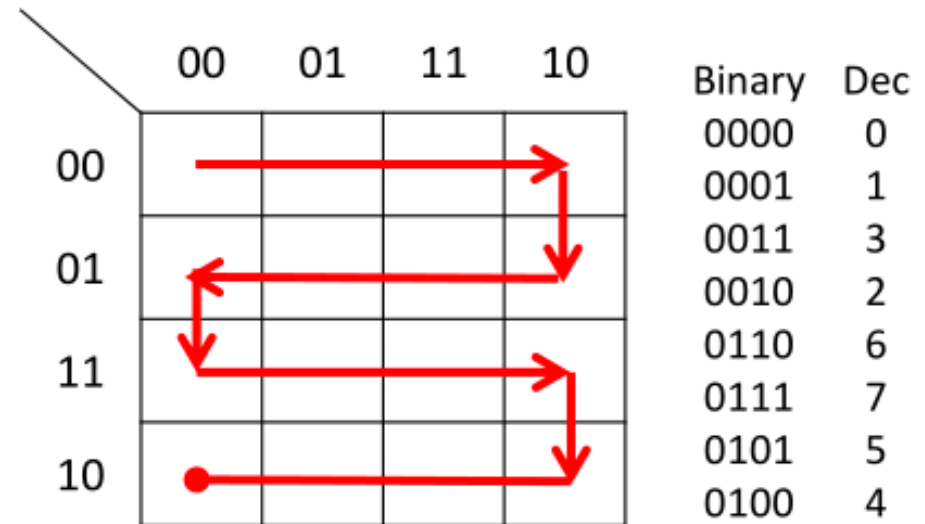
```
01. function gray_code=GrayCodeGen(M)
02. % 生成格雷编码
03. % 输入M: 阶数, 也即生成0: M-1的格雷码
04.
05. N=log2(M);
06. SubGray=[0;1];
07. for n=2:N
08.     top_gray=[zeros(1,2^(n-1))' SubGray];
09.     ReflectedSubGray=flipud(SubGray);
10.     bottom_gray=[ones(1,2^(n-1))' ReflectedSubGray];
11.     SubGray=[top_gray;bottom_gray];
12. end
13. gray_code=bi2de(flip1r(SubGray))';
14. end
```

MQAM星座点生成

□ 生成算法

- ✓ 生成一维Gray编码顺序
- ✓ 将一维Gray编码顺序转成二维矩阵
- ✓ 将偶数行的元素左右翻转
- ✓ 将二维矩阵转回一维矢量
- ✓ 根据一维矢量结果，生成星座点

```
01. function [ref]=constructQAM(M)
02. % 输入
03. % M MQAM的阶数
04. % ref Gray编码的MQAM星座点
05.
06. a=GrayCodeGen(M); %生成一维Gray编码的顺序
07. N=sqrt(M); % K-Map的维度 - N x N matrix
08. a=reshape(a,N,N).'; % 将一位Gray编码改成N x N 矩阵
09. evenRows=2:2:size(a,1); % 找出偶数行
10. a(evenRows,:)=fliplr(a(evenRows,:)); % 偶数行元素左右翻转
11. nGray=reshape(a.',1,M); % 再将N x N 矩阵转成一维矢量
12.
13. % 构建MQAM的星座点
14. x=floor(nGray/N);
15. y=mod(nGray,N);
16. Ax=2*(x+1)-1-N; %PAM 幅度 2m-1-D - 实轴
17. Ay=2*(y+1)-1-N; %PAM 幅度 2m-1-D - 虚轴
18. ref=Ax+1i*Ay; %MQAM 星座点 (I+jQ)
19. end
```



Walk type 1:
Cyclic Gray code possible

MQAM调制与解调的仿真实现

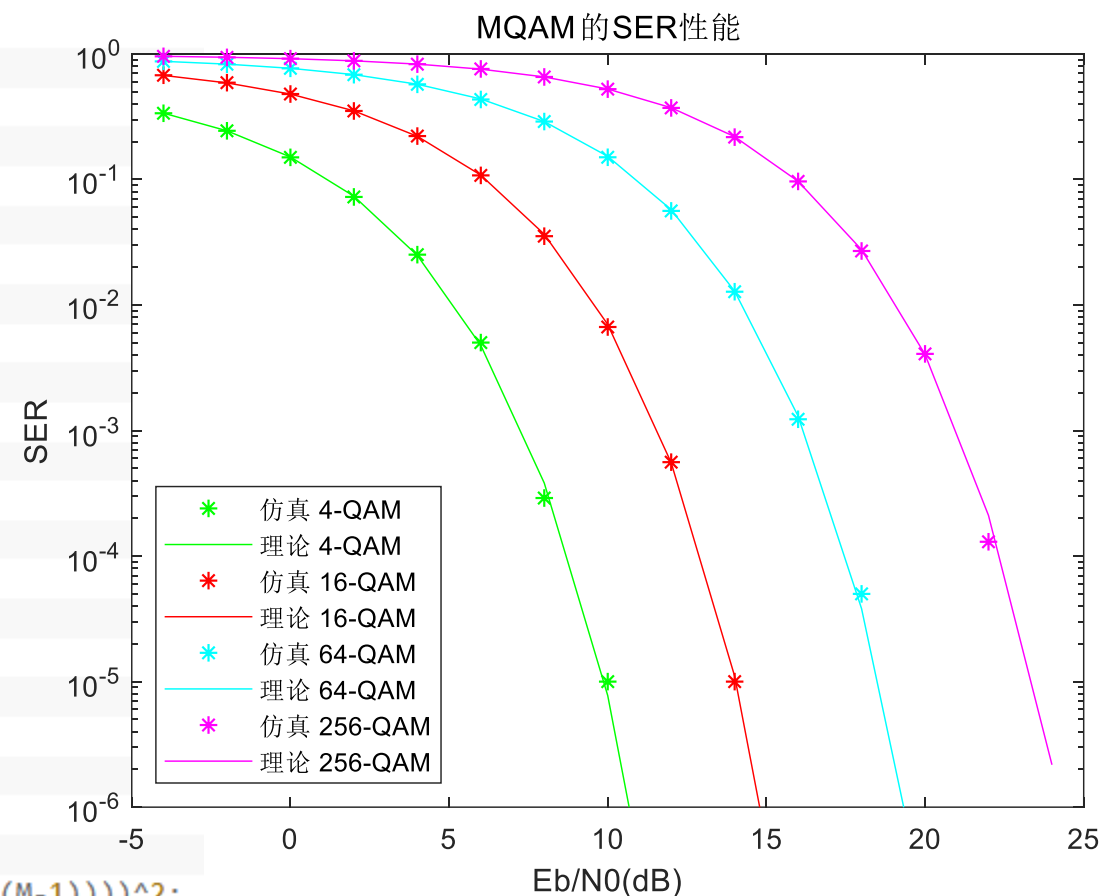
```
01. function [s,ref]=mqam_modulator(M,d)
02. % 输入参数
03. % M 调制阶数
04. % d 输入符号序列, 为行矢量
05. % 输出参数
06. % s 调制输出
07. % ref 参考星座点
08.
09. if(((M~=1) && ~mod(floor(log2(M)),2))==0), %M 必须是2的偶数幂
10.     error('只支持方形MQAM. M必须是2的偶数幂');
11. end
12.     ref=constructQAM(M); %生成星座图
13.     s=ref(d); %符号影射到对应的星座上
14. end
```

```
01. function [Out]= mqam_detector(M,r)
02. % 输入参数
03. % M 调制阶数
04. % r 接受的信号点, 为行矢量
05. % 输出参数
06. % Out 解调输出
07.
08. if(((M~=1) && ~mod(floor(log2(M)),2))==0) %M 必须是2的偶数幂
09.     error('只支持方形MQAM. M必须是2的偶数幂');
10. end
11. ref=constructQAM(M); %生成星座图
12. Out= OptCoherentDetector(r,ref); %IQ detection
13. end
```

MQAM在AWGN信道下的性能

□ simMQAMoverAWGN.m

```
05. %% 参数设置
06. nSym=1e6;           %符号数目
07. EbN0dB = 0:2:24;    % Eb/N0[dB] 的仿真范围
08. arrayOfM=[4,16,64,256]; %MQAM的阶数
09.
10. %% 仿真
11. SER_theory=zeros(length(arrayofM),length(EbN0dB));
12. SER_sim=zeros(length(arrayofM),length(EbN0dB));
13. for i = 1:length(arrayofM)
14.     M=arrayofM(i);
15.     k=log2(M);
16.     EsN0dB = 10*log10(k)+EbN0dB; %计算Es/N0的dB值
17.     a=randi([1,M],1,nSym);%生成1~M的符号值
18.     [s,ref]=mqam_modulator(M,a);
19.
20.     for j=1:length(EbN0dB)
21.         r = add_awgn_noise(s,EsN0dB(j));%通过AWGN信道
22.         acap= mqam_detector(M,r);
23.         SER_sim(i,j)=sum((a~=acap))/nSym;%计算仿真SER
24.         %计算理论SER: 性能公式为《Digital Communications》第五版 (4-3-27)
25.         %《Fundamentals of Communication Systems》第2版 (8.7.16)
26.         SER_theory(i,j) =1-(1-2*(1-1/sqrt(M))*qfn(sqrt(3*log2(M)*10.^(EbN0dB(j)/10)/(M-1))))^2;
27.     end
28. end
```



频移键控调制 (MFSK)

- 频率调制最简单的形式是二进制频率调制2FSK，使用两个不同的频率，例如 f_0 和 $f_1 = f_0 + \Delta f$ ，两个信号的波形为

$$u_0(t) = \sqrt{\frac{2\varepsilon_b}{T_b}} \cos 2\pi f_0 t, \quad 0 \leq t \leq T_b,$$

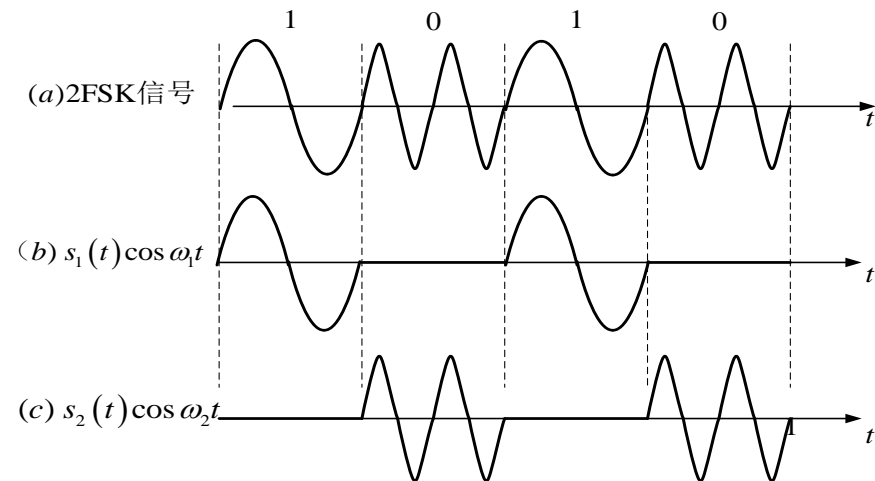
$$u_1(t) = \sqrt{\frac{2\varepsilon_b}{T_b}} \cos 2\pi f_1 t, \quad 0 \leq t \leq T_b,$$

✓ ε_b 是每比特的信号能量， T_b 为比特持续时间

- 将2FSK扩展到MFSK，也就是每个波形传递 $k = \log_2(M)$ 个比特信息，这 M 个波形可以表示为

$$u_m(t) = \sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + 2\pi m \Delta f t), \quad m = 0, 1, \dots, M-1, \quad 0 \leq t \leq T$$

- ✓ 其中 $E_s = kE_b$ 是每个符号的能量， $T = kT_b$ 是符号区间， Δf 是相邻两个频率之间的间隔，即 $\Delta f = f_m - f_{m-1}$ ， $m = 1, 2, \dots, M-1$ ， $f_m = f_c + m\Delta f$
- ✓ 每个信号具有相等的能量，频率间隔 Δf 决定了从 M 个可能发送信号的识别出发送信号的程度



相关系数与正交

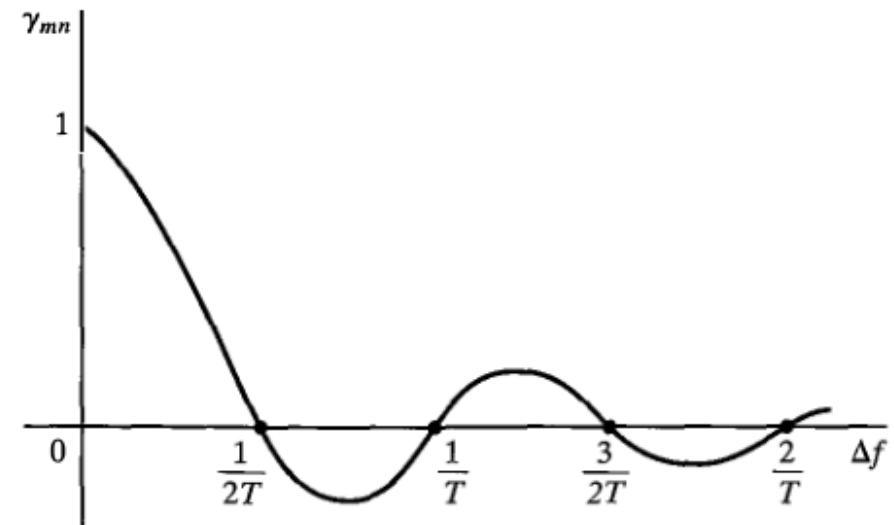
□ 相关系数 γ_{mn} 定义为

$$\gamma_{mn} = \frac{1}{E_s} \int_0^T u_m(t) u_n(t) dt$$

$$\begin{aligned} \gamma_{mn} &= \frac{1}{E_s} \int_0^T \frac{2E_s}{T} \cos(2\pi f_c t + 2\pi m \Delta f t) \cos(2\pi f_c t + 2\pi n \Delta f t) dt \\ &= \frac{1}{T} \int_0^T \cos(2\pi(m-n)\Delta f t) dt + \frac{1}{T} \int_0^T \cos[4\pi f_c t + 2\pi(m+n)\Delta f t] dt \\ &= \frac{\sin[2\pi(m-n)\Delta f T]}{2\pi(m-n)\Delta f T} \quad f_c \gg 1/T \end{aligned}$$

□ γ_{mn} 为频率间隔 Δf 的函数

- ✓ Δf 是 $1/(2T)$ 的倍数时，这些信号波形正交
- ✓ 从正交性来说，最小频率间隔为 $1/(2T)$



正交信号的等效基带模型

□ 对类似于MFSK这样的正交信号，其发射信号等效基带模型可以表示为

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_M \end{bmatrix} = \begin{bmatrix} \sqrt{E_s} e^{j\phi_1} & 0 & \cdots & 0 \\ 0 & \sqrt{E_s} e^{j\phi_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{E_s} e^{j\phi_M} \end{bmatrix}$$

✓ $\phi_1, \phi_2, \dots, \phi_M$ 为信号的初始相位

□ 若发端发的是 s_i ，接收信号为

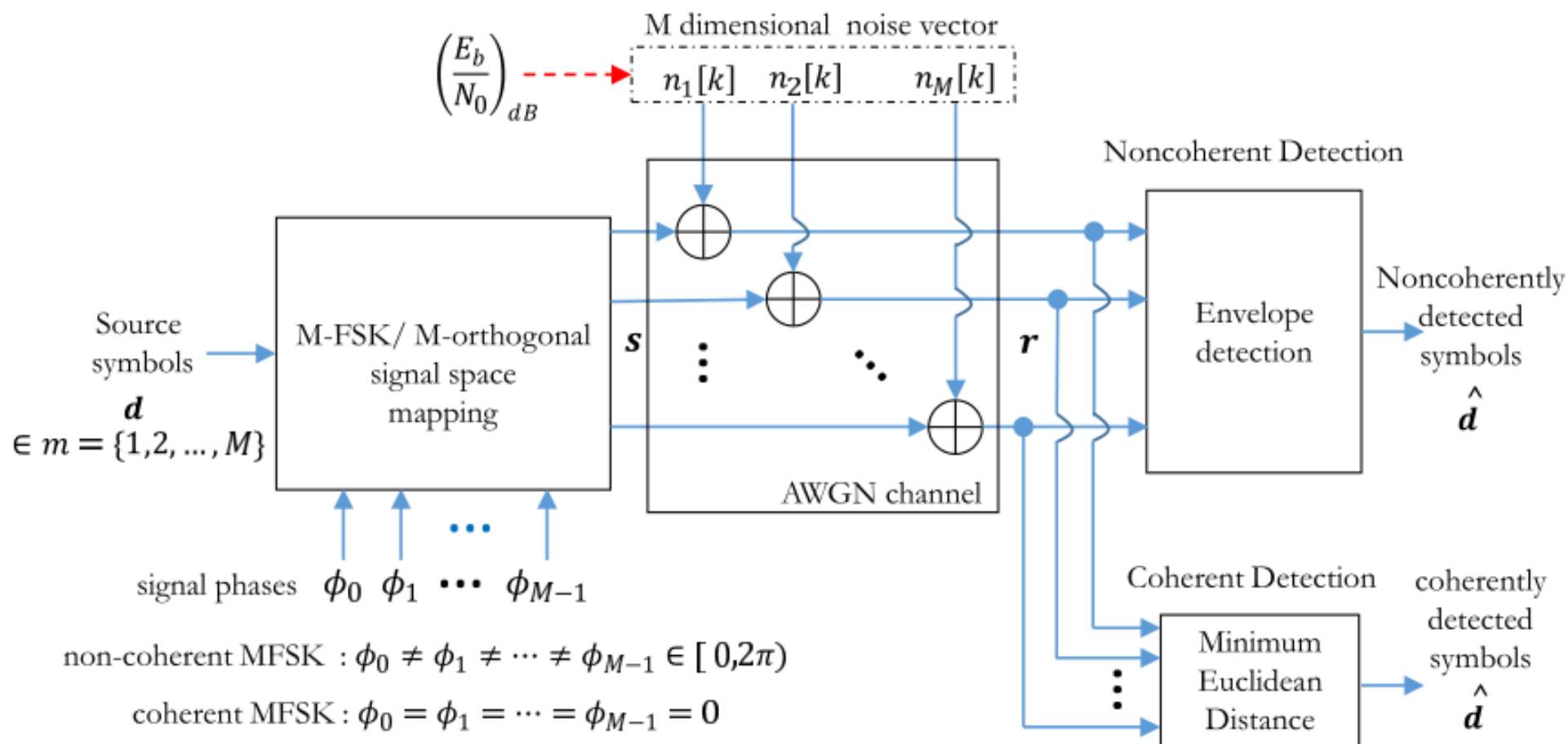
$$\mathbf{r} = [r_1, \dots, r_i, \dots, r_M] = [n_1, \dots, \sqrt{E_s} e^{j\phi_i} + n_i, \dots, n_M]$$

✓ n_1, n_2, \dots, n_M 为零均值，方差 $\sigma^2 = N_0/2$ 的高斯随机变量

✓ 对相干检测， $\phi_i = 0$

✓ 对非相干检测， ϕ_i 为 $(0, 2\pi)$ 上均匀分布的随机数

MFSK相干和非相干检测性能仿真的框架



相干解调与非相干解调

□ 相干检测

- ✓ 正交信号为多维信号，直接计算信号与参考星座点之间的欧式距离
- ✓ 返回提供最小欧几里德距离的符号
- ✓ 误符号率

$$P_M = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \{1 - [1 - Q(x)]^{M-1}\} e^{-(x - \sqrt{2E_s/N_0})^2/2} dx.$$

□ 非相干检测

- ✓ 没有初始相位信息
- ✓ 对接收信号的计算幅度
- ✓ 返回幅度最大符号 $i = \arg \max_{i=1, \dots, M} |r_i|$
- ✓ 误符号率

$$P_c = \sum_{n=0}^{M-1} (-1)^n \binom{M-1}{n} \frac{1}{n+1} e^{-n\rho_s/(n+1)},$$

```
01. function idealPoints= OptCoherentDetector(received,ref)
02. %最优相干检测，适用于MPAM,MPSK,MQAM和MFSK
03. %received - 接收到的复信号
04. %ref - 参考星座点
05. %idealPoints 解调输出符号
06.
07. [m,~] = size(received);
08. [n,~] = size(ref);
09.
10. if m==1 %输入序列为一位矢量，如MPAM,MPSK,MAQM,把信号构造成二维上的星座坐标
11.     x=[real(received); imag(received)]'; %接收信号的实部与虚部
12.     y=[real(ref); imag(ref)]'; %参考星座点
13. else %对于正交信号，如MFSK，本身已是一个多维信号
14.     x=received;
15.     y=ref;
16. end
17.
18.
19. X = sum(x.*x,2);
20. Y = sum(y.*y,2)';
21. d = X(:,ones(1,n)) + Y(ones(1,m),:) - 2*x*y'; %欧氏距的平方.
22. [~,idealPoints]=min(d,[],2); %寻找最小欧式距的符号
23. idealPoints=idealPoints.';
24. end
```

MFSK调制与解调的仿真实现

```
01. function [s,ref]= mfsk_modulator(M,d,COHERENCE)
02. % 输入参数
03. % M 调制阶数
04. % d 输入符号序列, 为行矢量
05. % COHERENCE MFSK的解调方案'coherent'/'noncoherent'
06. % 输出参数
07. % s 调制输出
08. % ref 参考星座点
09.
10.     if strcmpi(COHERENCE,'coherent')
11.         phi= zeros(1,M); %对相干检测, phase为0
12.         ref = complex(diag(exp(1i*phi)));
13.         s = complex(ref(d,:)); %符号影射到对应的星座上
14.     else
15.         phi = 2*pi*rand(1,M);%M个在(0,2pi)上的随机相位
16.         ref = diag(exp(1i*phi));
17.         s = ref(d,:);
18.     end
19. end
```

```
01. function [Out]= mfsk_detector(M,r,COHERENCE)
02. % 输入参数
03. % M 调制阶数
04. % r 接受的信号点, 为行矢量
05. % COHERENCE MFSK的解调方案'coherent'/'noncoherent'
06. % 输出参数
07. % Out 解调输出
08.     if strcmpi(COHERENCE,'coherent')
09.         phi= zeros(1,M); %对相干检测, phase为0
10.         ref = complex(diag(exp(1i*phi)));%生成参考星座
11.         Out= OptCoherentDetector(real(r),ref);%相干检测
12.     else %非相干MFSK
13.         [~,Out]= max(abs(r),[],2);%相干幅度检测
14.         Out = Out.';
15.     end
16. end
```

MFSK在AWGN信道下的性能仿真

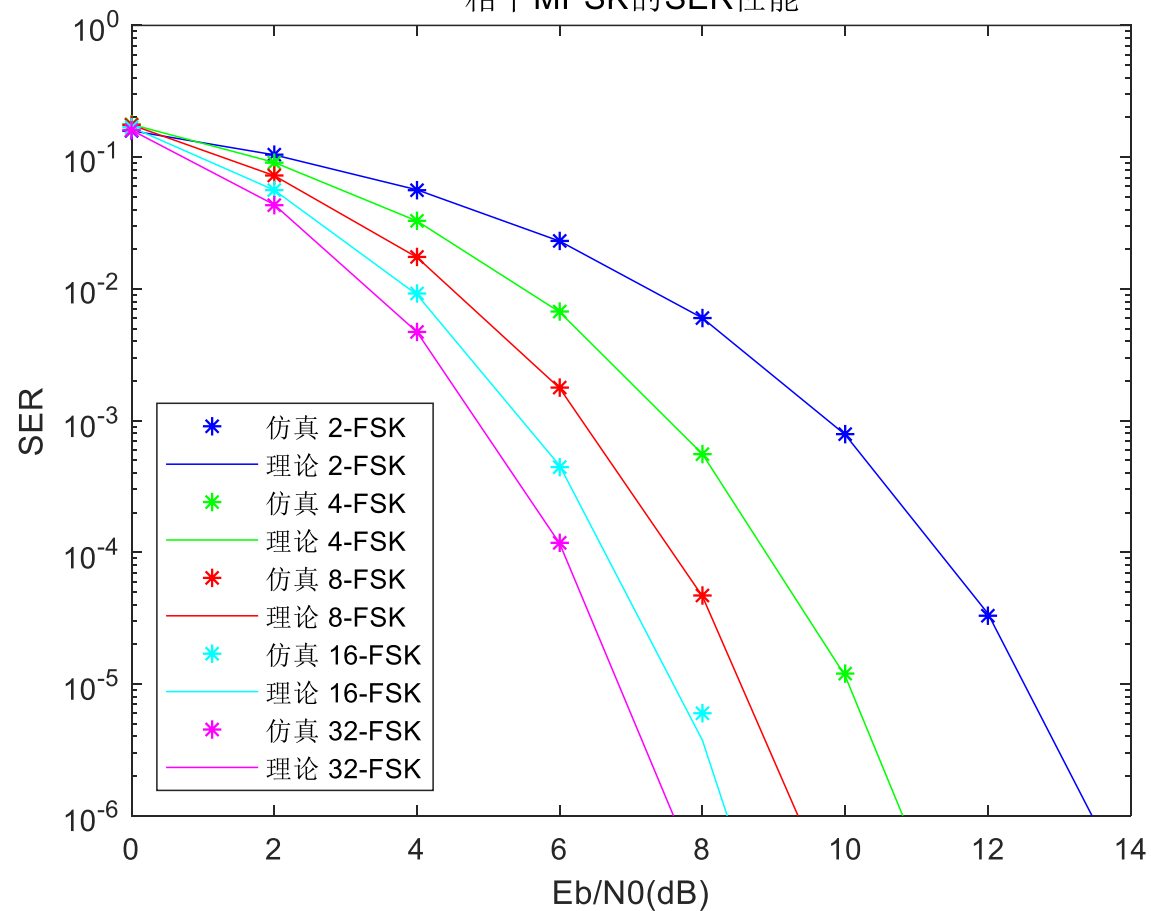
□ simMFSKoverAWGN.m

```
05. %% 参数设置
06. nSym=1e6;           %符号数目
07. EbN0dB = 0:2:24;    % Eb/N0[dB] 的仿真范围
08. arrayOfM=[2,4,8,16,32]; %MFSK的阶数
09. % arrayOfM=[2,4]; %MFSK的阶数
10. COHERENCE = 'coherent'; % MFSK的解调方案: 'coherent'/'noncoherent'
```

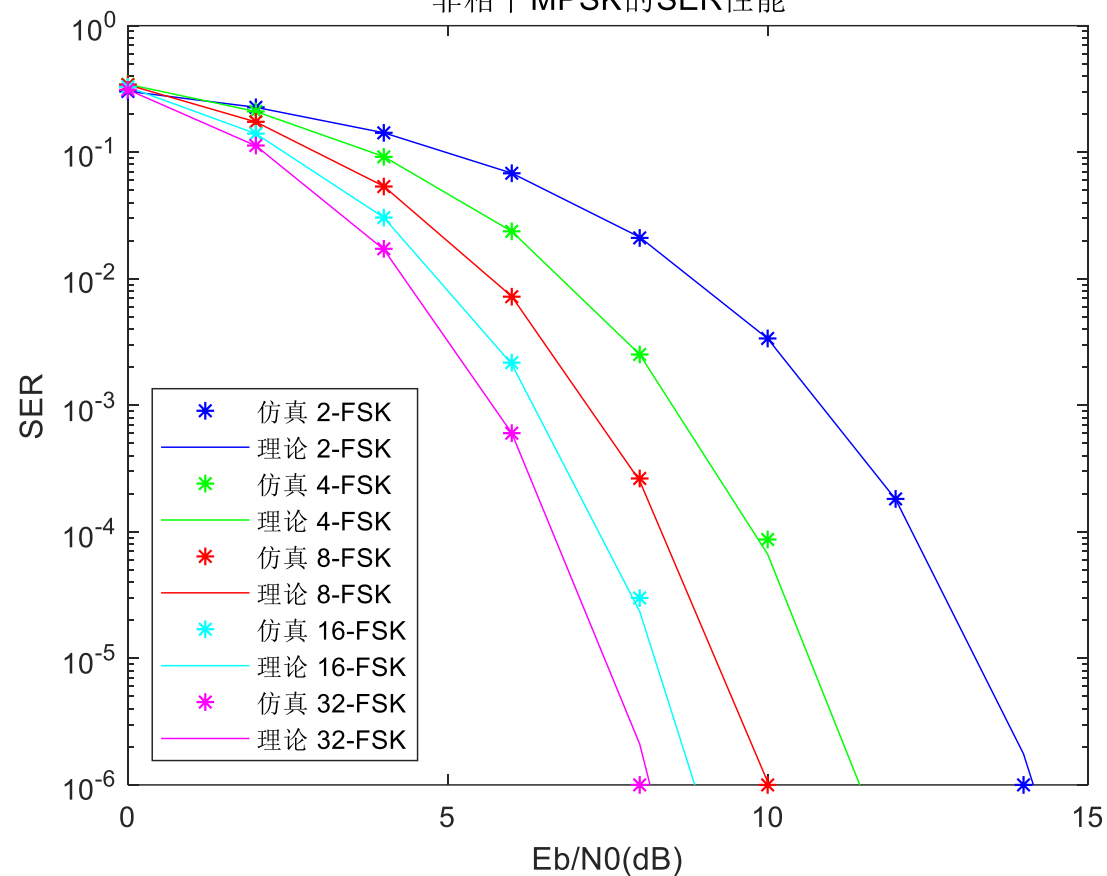
```
12. %% 仿真
13. SER_theory=zeros(length(arrayofM),length(EbN0dB));
14. SER_sim=zeros(length(arrayofM),length(EbN0dB));
15. for i = 1:length(arrayofM)
16.     M=arrayofM(i);
17.     k=log2(M);
18.     EsN0dB = 10*log10(k)+EbN0dB; %计算Es/N0的dB值
19.
20.     d=randi([1,M],1,nSym);%生成1~M的符号值
21.     s=mfsk_modulator(M,d,COHERENCE);
22.     for j=1:length(EsN0dB)
23.         r = add_awn_noise(s,EsN0dB(j)); %通过AWGN信道
24.         dCap = mfsk_detector(M,r,COHERENCE);
25.         SER_sim(i,j) = sum((d~=dCap))/nSym; %计算SER
26.     end
27.     %计算SER
28.     %相干检测性能公式为《Digital Communications》第五版 (4-4-10)
29.     % 或《Fundamentals of Communication Systems》第2版 (9.1.15)
30.     %非相干检测性能公式为《Fundamentals of Communication Systems》第2版 (9.5.40)
31.     gamma_s=10.^(EsN0dB/10);
32.     if strcmpi(COHERENCE,'coherent')
33.         for ii=1:length(gamma_s)
34.             fun=@(x)(1-(1-0.5*(erfc(x/sqrt(2))))).^(M-1)).*exp(-(x-sqrt(2*gamma_s(ii))).^2/2);
35.             SER_theory(i,ii) = 1/sqrt(2*pi)*integral(fun,-inf,inf);
36.         end
37.     else
38.         for ii=1:length(gamma_s)
39.             summ=0;
40.             for jj=1:M-1
41.                 n=M-1;
42.                 r=jj;
43.                 summ=summ+(-1).^(jj+1)./(jj+1).*prod((n-r+1:n)./(1:r)).*exp(-jj./(jj+1).*gamma_s(ii));
44.             end
45.             SER_theory(i,ii)=summ;
46.         end
47.     end
48. end
```

MFSK在AWGN信道下的性能结果

相干MPSK的SER性能



非相干MPSK的SER性能



差分编码相移键控调制(MDPSK)

- 差分编码相移调制不需要载波的相位，因此MDPSK被归为非相干检测
- 信息承载在相位转移中，必须在两个符号间隔时间上进行检测
- 两个符号间隔时间上的第m个等效低通信号矢量表示为

$$s_{ml} = (\sqrt{2\mathcal{E}_s} \quad \sqrt{2\mathcal{E}_s} e^{j\theta_m}), \quad 1 \leq m \leq M$$

$$\theta_m = \frac{2\pi(m-1)}{M}$$

- 接收信号为

$$r_l = (r_1 \quad r_2) = (\sqrt{2\mathcal{E}_s} \quad \sqrt{2\mathcal{E}_s} e^{j\theta_m}) e^{j\phi} + (n_{1l} \quad n_{2l}), \quad 1 \leq m \leq M$$

- 检测性能

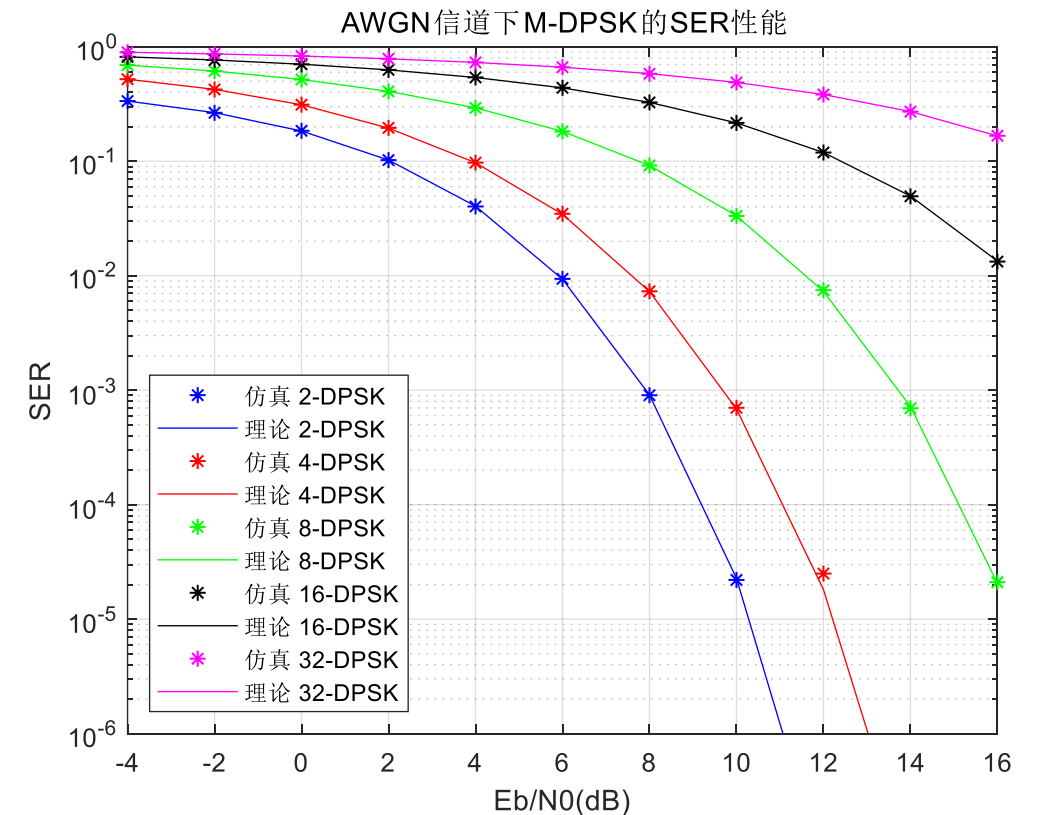
$$P_s(E) = \frac{\sin \frac{\pi}{M}}{2\pi} \int_{-\pi/2}^{\pi/2} \frac{\exp\left(-\frac{E_s}{N_0} \left[1 - \cos \frac{\pi}{M} \cos \theta\right]\right)}{1 - \cos \frac{\pi}{M} \cos \theta} d\theta$$

MDPSK在AWGN信道下的性能仿真

□ simMDPSKoverAWGN.m

```
08. %% 仿真
09. SER_sim = zeros(length(arrayofM),length(EbN0dB));
10. SER_theory = zeros(length(arrayofM),length(EbN0dB));
11. for i=1:length(arrayofM)
12.     M=arrayofM(i);
13.     k=log2(M);
14.     EsN0dB = EbN0dB + 10*log10(k);
15.
16.     data = randi([0,M-1],1,N-1);
17.
18.     % 生成差分调制信号
19.     % phi[k] = phi[k-1] + Dphi[k]
20.     data_diff = filter(1,[1 -1],data);
21.     x=exp(1i*2*pi*data_diff/M);
22.
23.     for j=1:length(EsN0dB)
24.         y=add_awgn_noise(x,EsN0dB(j));
25.         estPhase = angle(y);
26.         % Dphi[k] = phi[k] - phi[k-1]
27.         est_diffPhase = filter([1 -1],1,estPhase)*M/(2*pi);
28.         dec = mod(round(est_diffPhase),M); % 量化解调
29.
30.         SER_sim(i,j) = sum(dec~=data)/(N); %仿真SER
31.         %理论SER
32.         gamma_s=10^(EsN0dB(j)/10);
33.         gamma_b=10^(EbN0dB(j)/10);
34.
35.         %MDPSK检测性能公式为《Digital Communication over Fading Channels 2nd》第2版 (8.84)
36.         g = sin(pi/M).^2;
37.         fun=@(x)exp(-gamma_s*(1-cos(pi/M)*cos(x)))/(1-cos(pi/M)*cos(x));
38.         SER_theory(i,j)=sin(pi/M)/(2*pi)*integral(fun,-pi/2,pi/2,'ArrayValued',true);
39.     end
40. end
```

```
03. %% 参数设置
04. N = 1e6; %符号数目
05. EbN0dB = -4:2:16; % Eb/N0[dB] 的仿真范围
06. arrayofM = [2,4,8,16,32]; %MDPSK的阶数
```



平衰落信道

□ 平衰落也称为块衰落，意味着衰落过程在给定的传输时间间隔内近似不变，但块与块之间随机变化

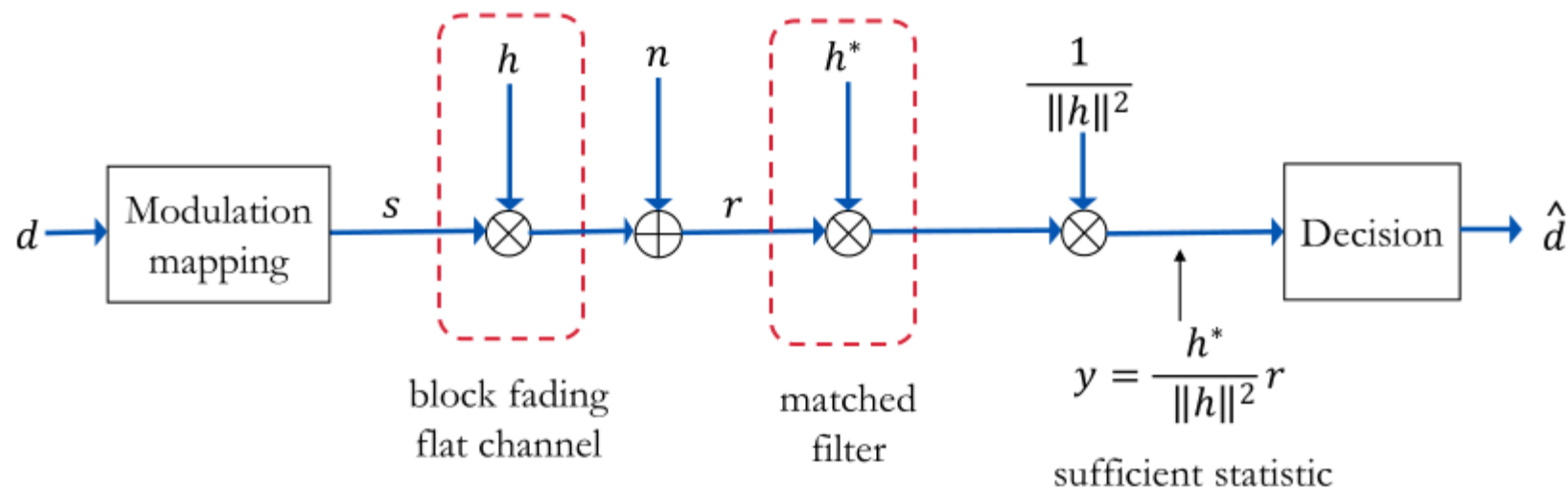
□ 平衰落模型

$$y = hx + n$$

□ 接收端检测

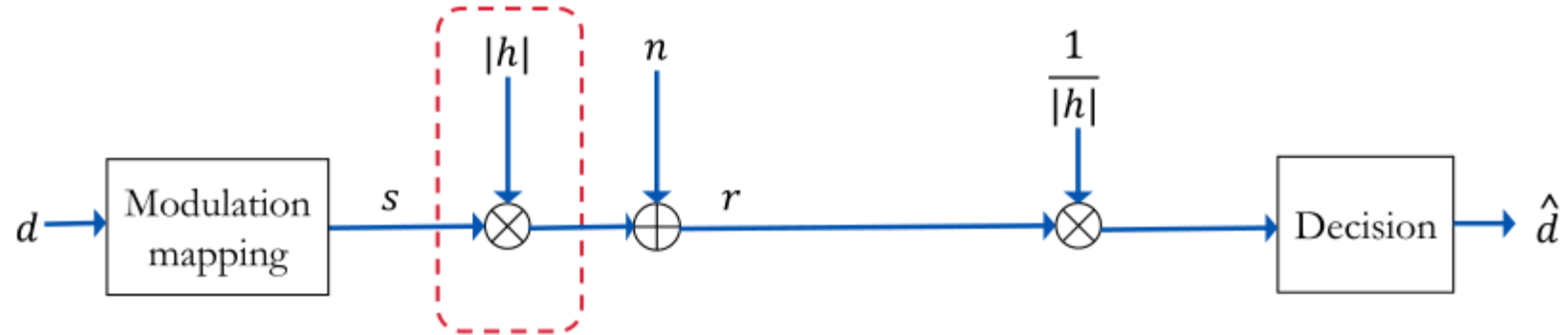
✓ 假设完美信道估计，运用匹配滤波

$$\tilde{r} = \frac{h^*}{|h|^2} y = \frac{h^*}{|h|^2} hx + \frac{h^*}{|h|^2} n = x + \tilde{n}$$



瑞利仿真框架

□ 等效基带仿真模型



□ 瑞利平衰落信道

$$h = |X + jY| \quad X \sim \mathcal{N}(0, \sigma^2 / 2) \quad Y \sim \mathcal{N}(0, \sigma^2 / 2)$$

✓ 为了使信道增益为1, σ 设为 $\sqrt{0.5}$

MPSK在Rayleigh信道下的性能仿真

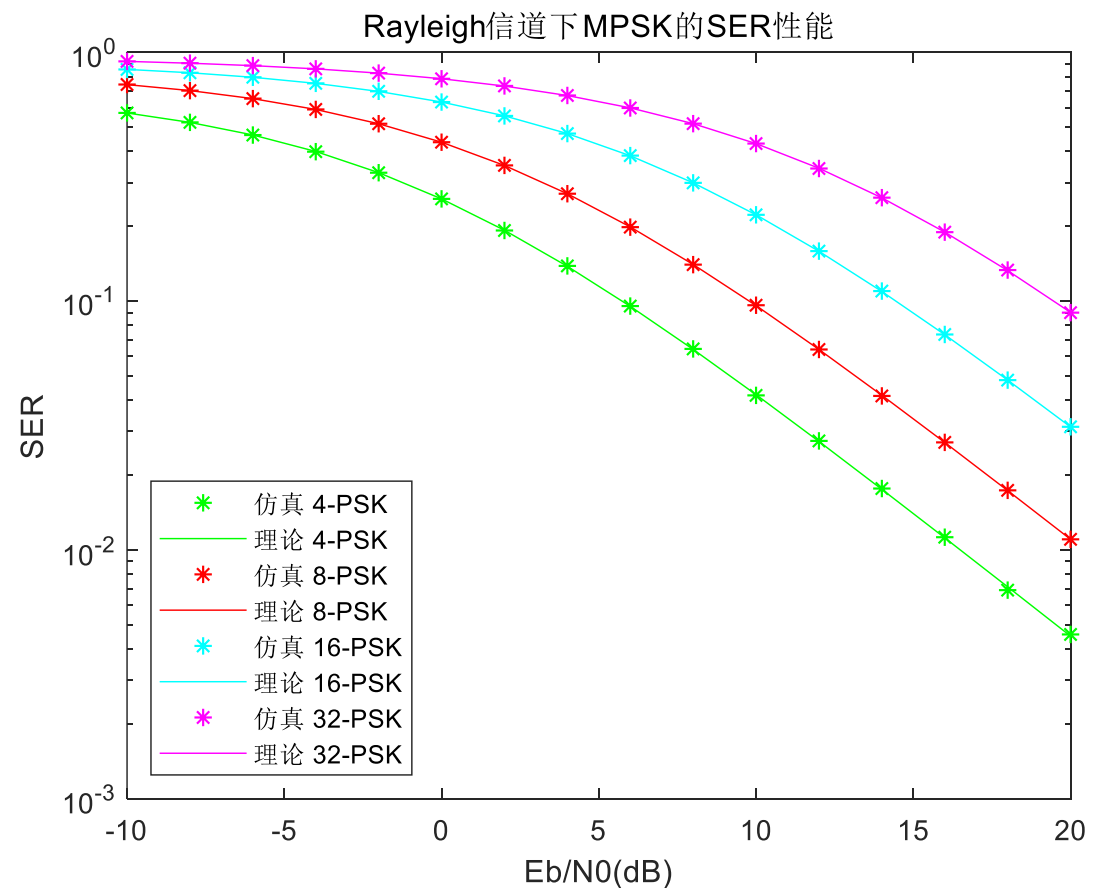
□ 性能理论公式

$$P_s(E) = \left(\frac{M-1}{M}\right) \left\{ 1 - \sqrt{\frac{g_{\text{PSK}} \bar{\gamma}_s}{1 + g_{\text{PSK}} \bar{\gamma}_s}} \left(\frac{M}{(M-1)\pi} \right) \right. \\ \left. \times \left[\frac{\pi}{2} + \tan^{-1} \left(\sqrt{\frac{g_{\text{PSK}} \bar{\gamma}_s}{1 + g_{\text{PSK}} \bar{\gamma}_s}} \cot \frac{\pi}{M} \right) \right] \right\}$$

$$g_{\text{PSK}} \triangleq \sin^2 \pi / M.$$

□ 仿真: simMPSKoverRayleigh.m

```
10. %% 仿真
11. SER_theory=zeros(length(arrayofM),length(EbN0dB));
12. SER_sim=zeros(length(arrayofM),length(EbN0dB));
13.
14. for i = 1:length(arrayofM)
15.     M=arrayofM(i);
16.     k=log2(M);
17.     EsN0dB = 10*log10(k)+EbN0dB; %计算Es/N0的dB值
18.     a=randi([1,M],1,nSym);%生成1~M的符号值
19.     [s,ref]=mpsk_modulator(M,a);
20.
21.     for j=1:length(EsN0dB)
22.         h=1/sqrt(2)*(randn(1,nSym)+1i*randn(1,nSym));%瑞丽信道
23.         hs=abs(h).*s;
24.         r = add_awnn_noise(hs,EsN0dB(j));%通过AWGN信道
25.         y=r./abs(h);
26.         acap= mpsk_detector(M,y);
27.         SER_sim(i,j)=sum((a~=acap))/nSym; %计算仿真SER
28.         %计算理论SER: 性能公式为《Digital Communication over Fading Channels》第2版 (8.113)
29.         g = sin(pi/M).^2;
30.         temp=sqrt(g*10^(EsN0dB(j)/10)/(1+g*10^(EsN0dB(j)/10)));
31.         SER_theory(i,j)=(M-1)/M*(1-temp*M/((M-1)*pi)*(pi/2+atan(temp*cot(pi/M))));
32.     end
33. end
```



莱斯平衰落

- 瑞丽信道适合无直达径的场景。若存在直达径，信号包络为莱斯分布
- 若直达径的到达角AoA为 θ ，相位为 ϕ ，最大多普勒频率为 f_D ，衰落过程可以描述为

$$h(t) = \sqrt{\frac{K\Omega}{K+1}} e^{j2\pi f_D \cos \theta t + \phi} + \sqrt{\frac{\Omega}{K+1}} g(t)$$

- ✓ $g(t)$ 是窄带复高斯过程， K 为莱斯K因子
- ✓ 第一项为直达径，第二项散射径，两者功率之比为 $K = \frac{A^2}{S^2}$
- 对于仿真，给定 K 因子，莱斯平衰落可以生成为

$$h = |X + jY| \quad X, Y \sim \mathcal{N}(\mu, \sigma^2)$$

$$\mu = g_1 = \sqrt{\frac{K}{2(K+1)}} \quad \sigma = g_2 = \sqrt{\frac{1}{2(K+1)}}$$

MPSK在莱斯信道下的性能仿真

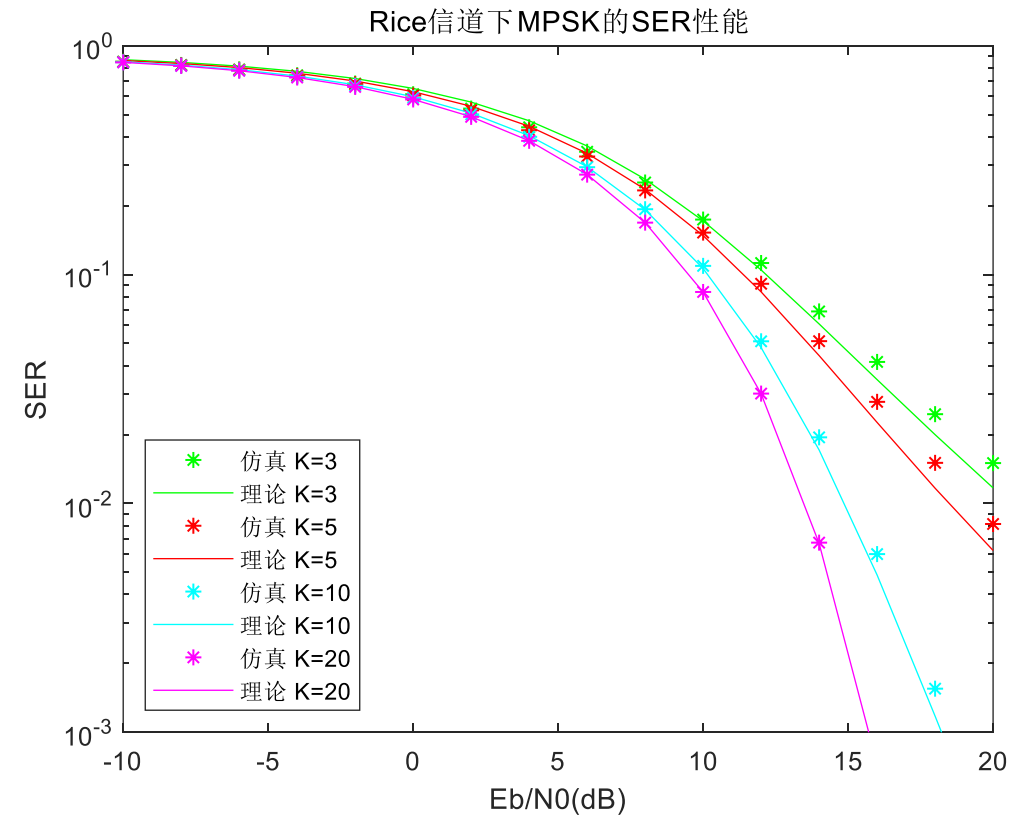
□ 性能理论公式

$$P_s(E) = \frac{1}{\pi} \int_0^{(M-1)\pi/M} \frac{(1+K) \sin^2 \theta}{(1+K) \sin^2 \theta + g_{\text{PSK}} \bar{\gamma}_s} \times \exp\left(-\frac{K g_{\text{PSK}} \bar{\gamma}_s}{(1+K) \sin^2 \theta + g_{\text{PSK}} \bar{\gamma}_s}\right) d\theta$$

$$g_{\text{PSK}} \triangleq \sin^2 \pi/M.$$

□ 仿真: simMPSKoverRice.m

```
12. %% 仿真
13. SER_theory=zeros(length(K_dB),length(EbN0dB));
14. SER_sim=zeros(length(K_dB),length(EbN0dB));
15.
16. for i = 1:length(K_dB)
17. % M=arrayofM(i);
18. k=log2(M);
19. EsN0dB = 10*log10(k)+EbN0dB; %计算Es/N0的dB值
20. a=randi([1,M],1,nSym);%生成1~M的符号值
21. [s,ref]=mpsk_modulator(M,a);
22.
23. K = 10.^(K_dB(i)/10); %K factor in linear scale
24. g1 = sqrt(K/(2*(K+1))); g2 = sqrt(1/(2*(K+1)));
25.
26. for j=1:length(EsN0dB)
27. h = (g2*randn(1,nSym)+g1)+1i*(g2*randn(1,nSym)+g1);
28. hs=abs(h).*s;
29. r = add_awgn_noise(hs,EsN0dB(j));%通过AWGN信道
30. y=r./abs(h);
31. acap= mpsk_detector(M,y);
32. SER_sim(i,j)=sum((a~=acap))/nSym; %计算仿真SER
33. %计算理论SER: 性能公式为《Digital Communication over Fading Channels》第2版 (8.114)
34. gamma_s=10^(EsN0dB(j)/10);
35. g = sin(pi/M).^2;
36. fun = @(x) ((1+K)*sin(x).^2)/((1+K)*sin(x).^2+g*gamma_s)...
37. *exp(-K*g*gamma_s./((1+K)*sin(x).^2+g*gamma_s));
38. SER_theory(i,j) = (1/pi)*integral(fun,0,pi*(M-1)/M);
39. end
40. end
```



课后作业

□ 仿真MQAM，绘制BER性能曲线，BER理论公式为

$$P_b = \frac{2}{\sqrt{M} \log_2 \sqrt{M}} \times \sum_{k=1}^{\log_2 \sqrt{M}} (1-2^{-k}) \sum_{i=0}^{\sqrt{M}-1} \left\{ (-1)^{\left\lfloor \frac{i2^{k-1}}{\sqrt{M}} \right\rfloor} \left(2^{k-1} - \left\lfloor \frac{i2^{k-1}}{\sqrt{M}} + \frac{1}{2} \right\rfloor \right) Q \left((2i+1) \sqrt{\frac{6 \log_2 M}{2(M-1)} \frac{E_b}{N_0}} \right) \right\}$$
$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp(-t^2/2) dt$$

✓ Cho, K., and Yoon, D., "On the general BER expression of one- and two-dimensional amplitude modulations", *IEEE Trans. Commun.*, vol. 50, no. 7, July 2002, pp. 1074-1080.

课后作业

- **Demo3: 在Demo2的基础上, 实现可根据配置, 进行MQAM、MPSK和MPAM三种调制的仿真**

有问题，随便问！

