



東南大學  
SOUTHEAST UNIVERSITY

数字通信的计算机仿真（研讨）

# 12 载波调制技术(下) ——波形仿真

讲解人：王俊波

***E-mail: [jbwang@seu.edu.cn](mailto:jbwang@seu.edu.cn)***

***Phone: 13770681926***

***QQ:308322767***

# 主要内容

---

- BPSK、差分编码的相干和非相干检测
- QPSK和OQPSK
- 连续相位调制 (MSK)
- FSK调制的相干和非相干检测

# 波形仿真概述

## □ 通信系统仿真中存在两种基本的模型：通带模型及其基带等效模型

- ✓ 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

## □ 在波形仿真中，发射信号、噪声和接收信号都由波形样本表示

- ✓ 波形级仿真通常使用每个符号的多个样本
- ✓ 优点：可以更精确的模拟通信的更多细节，主要用于信道估计、频谱分析、同步、均衡等
- ✓ 缺点：计算量大，仿真速度慢

# BPSK

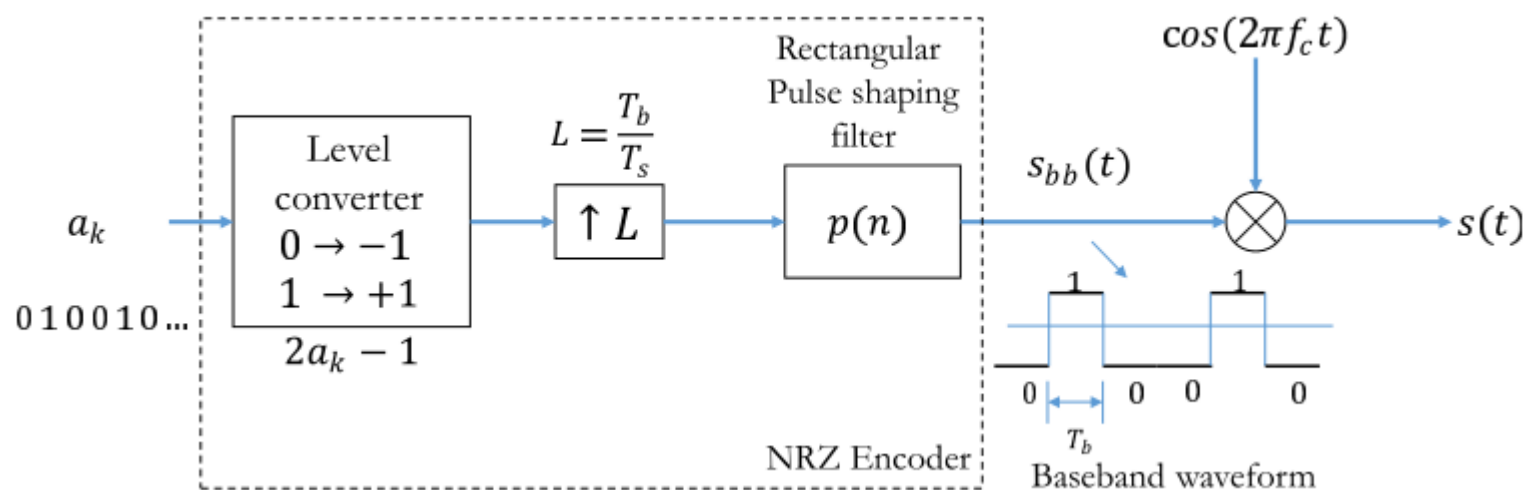
- 二进制相移键控 (BPSK) 是一种二相调制方案, 其中二进制信息中的0和1由载波信号中的两种不同相位表示

$$s_1(t) = A_c \cos(2\pi f_c t), \quad 0 \leq t \leq T_b \text{ for binary 1}$$

$$s_0(t) = A_c \cos(2\pi f_c t + \pi), \quad 0 \leq t \leq T_b \text{ for binary 0}$$

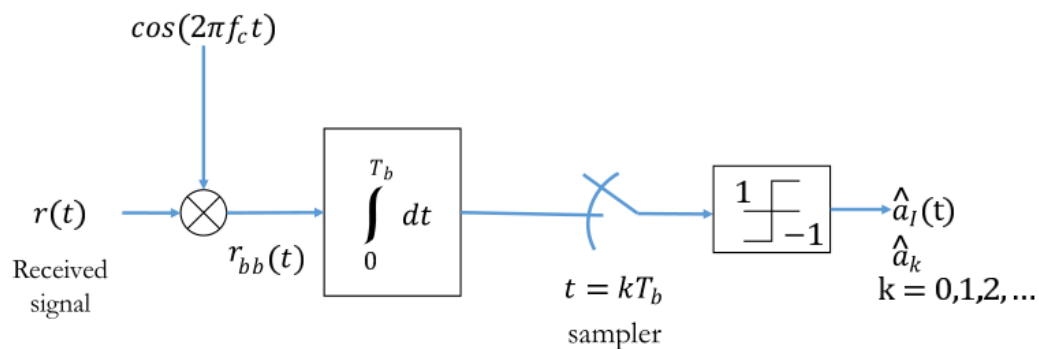
✓  $A_c$ : 正弦信号的振幅,  $f_c$ : 载波频率 (Hz),  $t$ : 以秒为单位的瞬时时间,  $t_b$ : 以秒为单位的比特周期

## □ BPSK发射机



```
01. function [s_bb,t] = bpsk_mod(ak,L)
02. % 使用BPSK调制二进制数据流
03. % 输入参数
04. % ak - 输入二进制数据流
05. % L - 过采样因子(Tsym/Tsam)
06. % 输出参数
07. % s_bb - BPSK调制信号
08. % t - 调制信号的时间矢量
09.
10. N = length(ak); %符号数
11. a = 2*ak-1; %BPSK调制
12. ai= repmat(a,1,L).'; %过采样
13. ai = ai(:).'; %串行化
14. t=0:N*L-1; %时间矢量
15. s_bb = ai; %输出基带调制信号
```

## □ BPSK接收机



## □ BER性能 $0.5\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$

```

01. function [ak_cap] = bpsk_demod(r_bb,L)
02. % 解调基带BPSK调制信号
03. %r_bb - 基带调制信号
04. %L - 过采样因子(Tsym/Tsam)
05. %ak_cap - 检测出的二进制数据流
06. x=real(r_bb); %I路
07. x = conv(x,ones(1,L));%积分L个样点 (Tsym时间)
08. x = x(L:L:end); %I路, 降采样
09. ak_cap = (x > 0).'; %门限检测

```

## □ 端到端仿真

✓ bpsk\_wfm\_sim.m

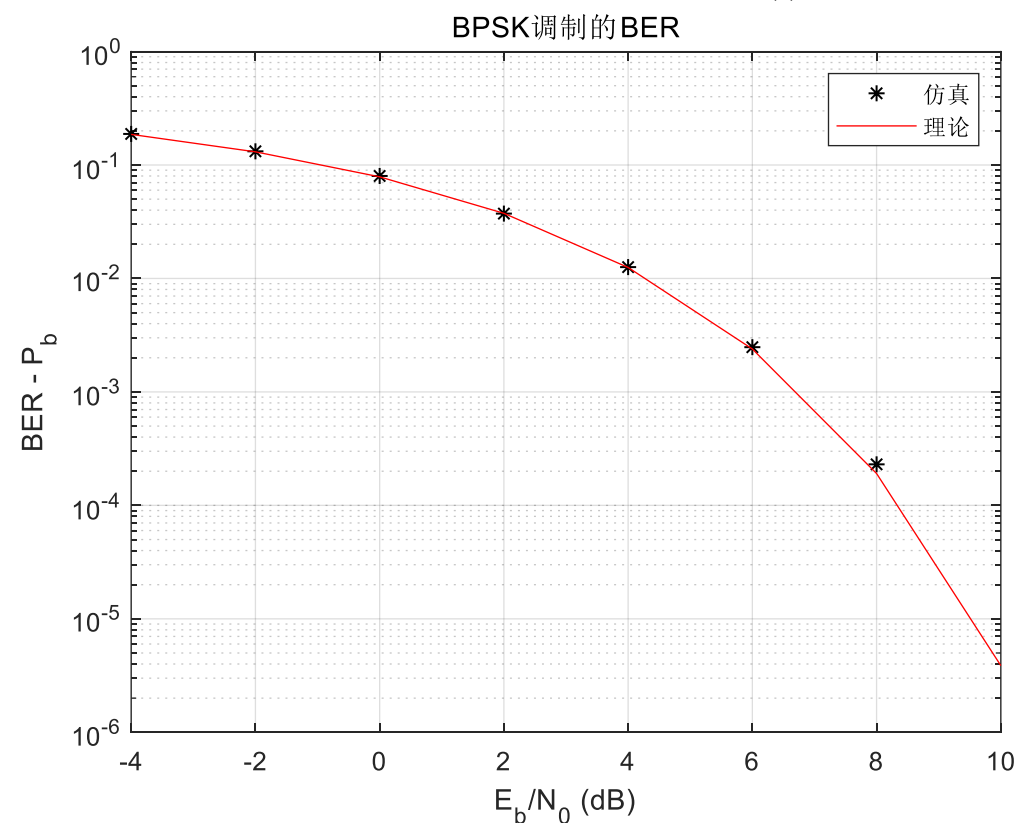
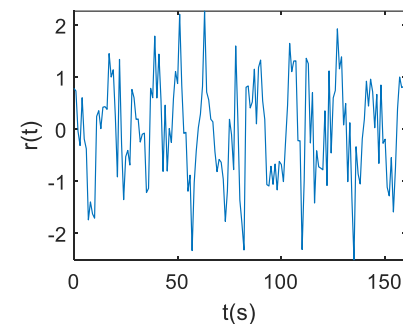
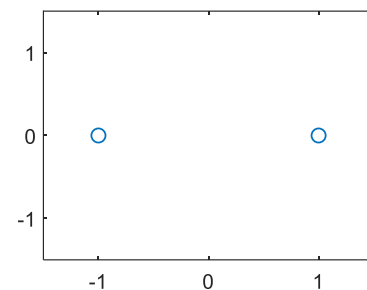
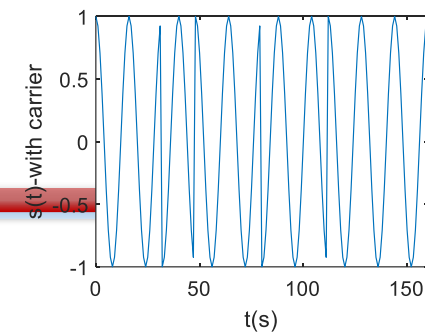
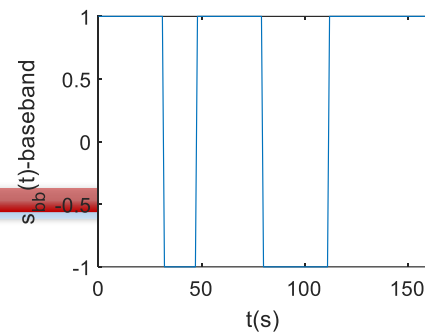
```

05. %% 参数设置
06. N=1e5 ; %发送的符号数目
07. EbN0dB = -4:2:10; % 仿真的Eb/N0的dB范围
08. L=16; %过采样因子
09. Fc=800; %载频
10. Fs=L*Fc; %采样频率
11.
12. %% 发送信号调制与发射波形
13. ak = randi([0,1],N,1)>0.5; %二进制数据流
14. [s_bb,t]= bpsk_mod(ak,L); %基带调制信号
15. s = s_bb.*cos(2*pi*Fc*t/Fs); %载波调制信号
16.
17. subplot(2,2,1);plot(t,s_bb);%基带信号
18. xlabel('t(s)'); ylabel('s_{bb}(t)-baseband');xlim([0,10*L]);
19. subplot(2,2,2);plot(t,s); %发射波形
20. xlabel('t(s)'); ylabel('s(t)-with carrier');xlim([0,10*L]);
21. %发射端星座
22. subplot(2,2,3);plot(real(s_bb),imag(s_bb),'o');
23. xlim([-1.5 1.5]); ylim([-1.5 1.5]);

```

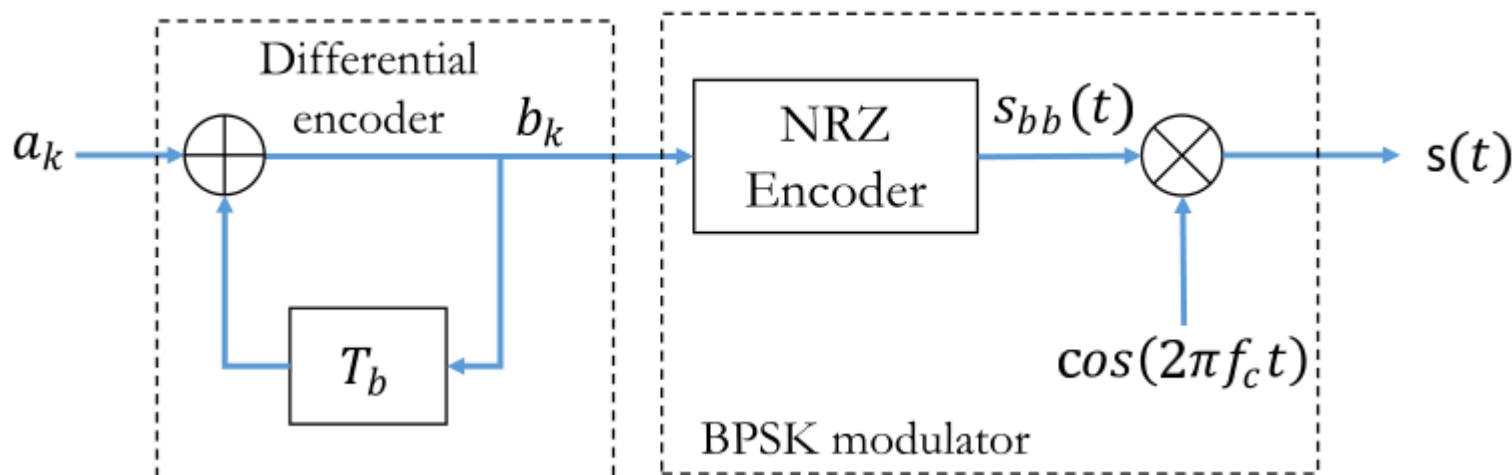
## □ bpsk\_wfm\_sim.m

```
25. %% 接收波形检测
26. EbN0lin = 10.^(EbN0dB/10);
27. BER = zeros(length(EbN0dB),1); %仿真BER存储空间
28. for i=1:length(EbN0dB)
29.     Eb=L*sum(abs(s).^2)/length(s); %信号能量
30.     N0= Eb/EbN0lin(i); %计算噪声功率谱密度
31.     n = sqrt(N0/2)*randn(1,length(s));%生成噪声
32.
33.     r = s + n;%通过噪声信道
34.     r_bb = r.*cos(2*pi*Fc*t/Fs);%载波解调
35.     ak_cap = bpsk_demod(r_bb,L);%基带相关解调
36.     BER(i) = sum(ak~=ak_cap)/N; %SER/BER统计
37.
38.     %接收信号波形
39.     subplot(2,2,4);plot(t,r);
40.     xlabel('t(s)'); ylabel('r(t)');xlim([0,10*L]);
41. end
42. theoreticalBER = 0.5*erfc(sqrt(EbN0lin));%理论BER
43. %% 结果展示
44. figure;semilogy(EbN0dB,BER,'k*');
45. hold on;semilogy(EbN0dB,theoreticalBER,'r-');
46. xlabel('E_b/N_0 (dB)'); ylabel('BER - P_b');
47. legend('仿真','理论');grid on;
48. title(['BPSK调制的BER']);
```



# 差分编码BPSK (DEBPSK) 的相干检测

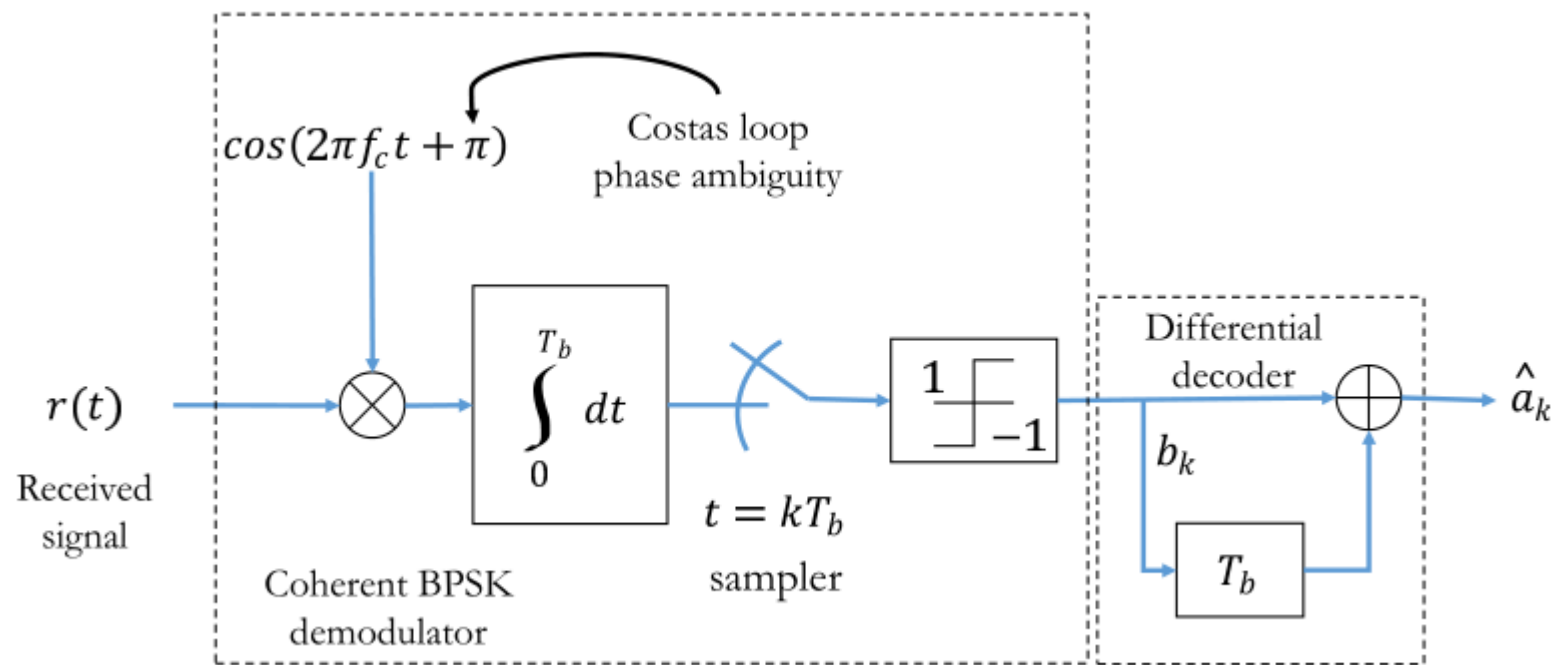
- 在相干检测中，接收机利用载波同步环导出其解调频率和相位参考
  - ✓ 这样的同步电路可能在检测到的相位中引入相位模糊度  $\phi = \hat{\theta} - \theta$ ，这可能导致解调比特中的错误判决
  - ✓ Costas环在锁定点处表现出 $\pi$ 弧度整数倍的相位模糊性，与完美载波同步期间的位相比，所有检测到的位完全反转
- 通过在BPSK调制器输入端应用差分编码和在接收端的相干解调器输出端执行差分解码，可以有效地消除相位模糊
- 调制



$$b[k] = b[k-1] \oplus a[k] \quad (\text{modulo } -2)$$

# 解调与性能

## □ 解调



$$a[k] = b[k] \oplus b[k-1] \quad (\text{modulo } -2)$$

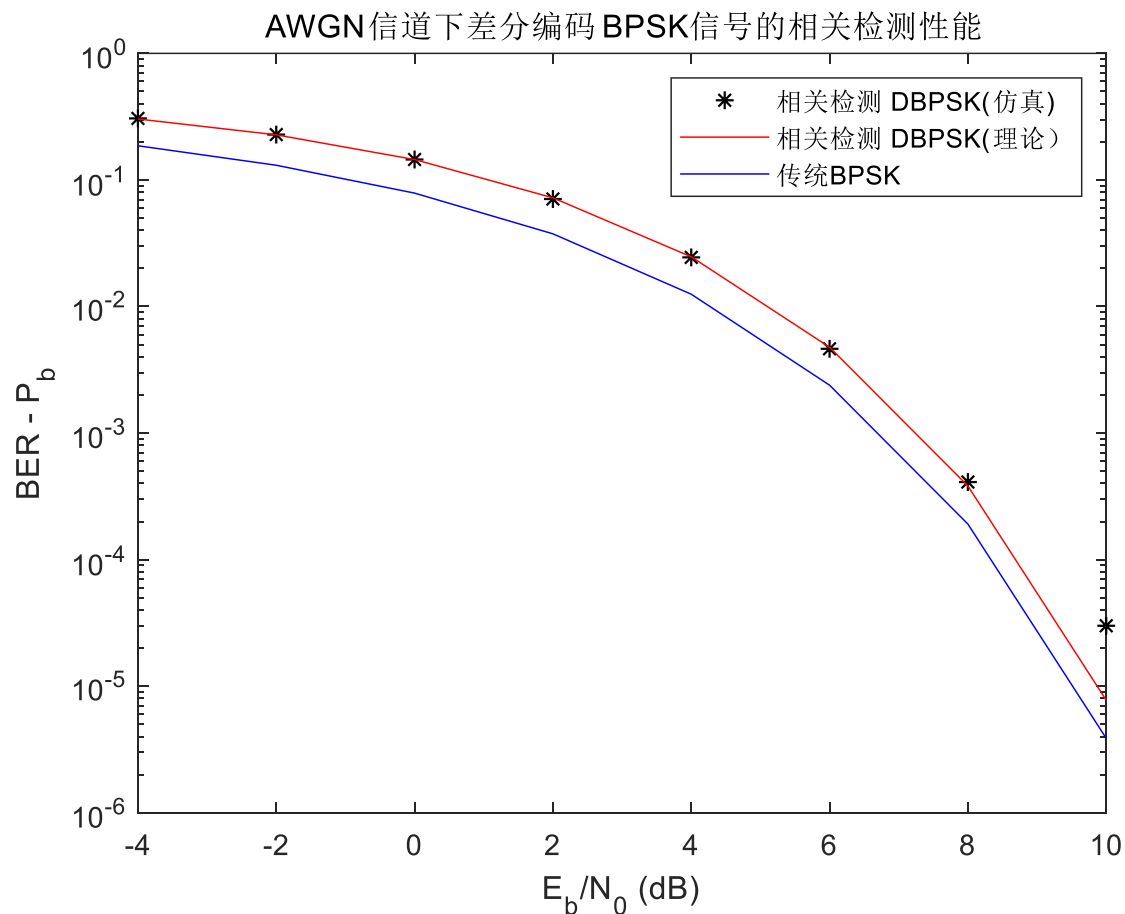
## □ 误码性能

$$P_b = \text{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right) \left[ 1 - \frac{1}{2} \text{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right) \right]$$



# DEBPSK仿真实现

## □ dbpsk\_coherent\_detection.m



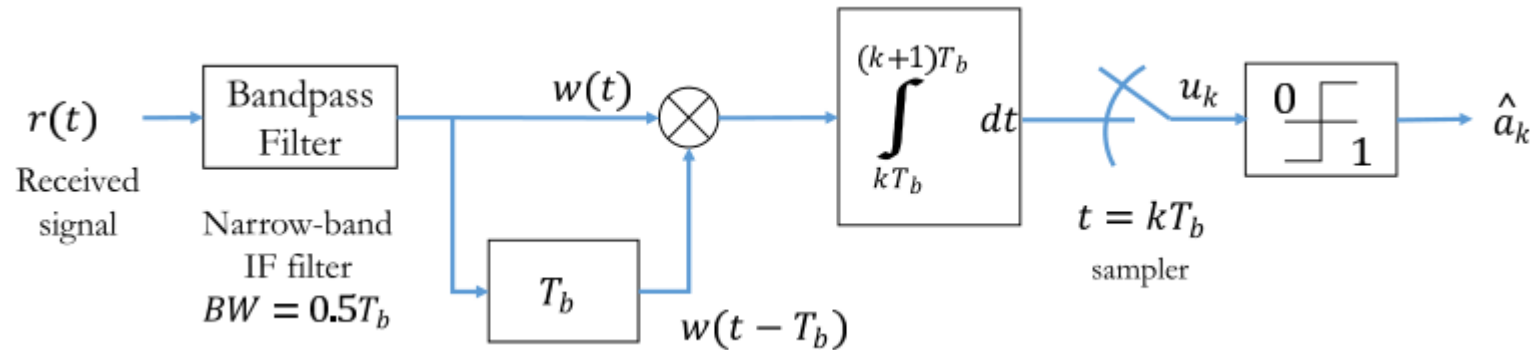
```
05. %% 参数设置
06. N=1e5; %发送的符号数目
07. EbN0dB = -4:2:10; % 仿真的Eb/N0的dB范围
08. L=16; %过采样因子
09. Fc=800; %载频
10. Fs=L*Fc; %采样频率
11.
12. EbN0lin = 10.^(EbN0dB/10);
13. EsN0dB=10*log10(1)+EbN0dB;
14. EsN0lin = 10.^(EsN0dB/10);
15. SER = zeros(length(EbN0dB),1);
16.
17. %% 发送信号调制与发射波形
18. ak = randi([0,1],N,1); %二进制数据流
19. bk = filter(1,[1 -1],ak,0); %IIR差分编码
20. bk = mod(bk,2); %模2运算
21. [s_bb,t]= bpsk_mod(bk,L); %基带BPSK调制
22. s = s_bb.*cos(2*pi*Fc*t/Fs); %载波调制
23.
24. %% 接收波形检测
25. for i=1:length(EbN0dB)
26.     Esym=sum(abs(s).^2)/length(s); %信号能量
27.     N0= Esym/EsN0lin(i); %计算功率谱密度
28.     n=sqrt(L*N0/2)*(randn(1,length(s))+1i*randn(1,length(s)));
29.     r = s + n;%通过AWGN信道
30.
31.     phaseAmbiguity=pi; %模拟Costas环导致的180°相位模糊
32.     r_bb=r.*cos(2*pi*Fc*t/Fs+phaseAmbiguity);%载波解调
33.     b_cap=bpsk_demod(r_bb,L);%基带相关解调
34.     a_cap=filter([1 1],1,b_cap); %FIR差分解码
35.     a_cap= mod(a_cap,2); %模2运算
36.     SER(i) = sum(ak~=a_cap)/N;%SER统计
37. end
38. theorySER_DPSK = erfc(sqrt(EbN0lin)).*(1-0.5*erfc(sqrt(EbN0lin)));
39. theorySER_BPSK = 0.5*erfc(sqrt(EbN0lin));
```

# 差分BPSK (DBPSK)

□ 与DEBPSK不同，差分BPSK (DBPSK) 是差分编码、非相干检测

- ✓ 相干检测要求参考信号具有准确的频率和相位信息
- ✓ 非相干检测则不需要参考信号

□ 次优接收机



□ 积分器输出

无噪声情况下

$$z = \int_{kT_b}^{(k+1)T_b} r(t)r(t - T_b)dt \quad \longrightarrow \quad z = \int_{kT_b}^{(k+1)T_b} s_k(t)s_{k-1}(t)dt = \begin{cases} E_b & , \text{if } s_k(t) = s_{k-1}(t) \\ -E_b & , \text{if } s_k(t) = -s_{k-1}(t) \end{cases}$$

✓ BER性能

$$P_b = \frac{1}{2} \exp\left(-0.76 \frac{E_b}{N_0}\right)$$

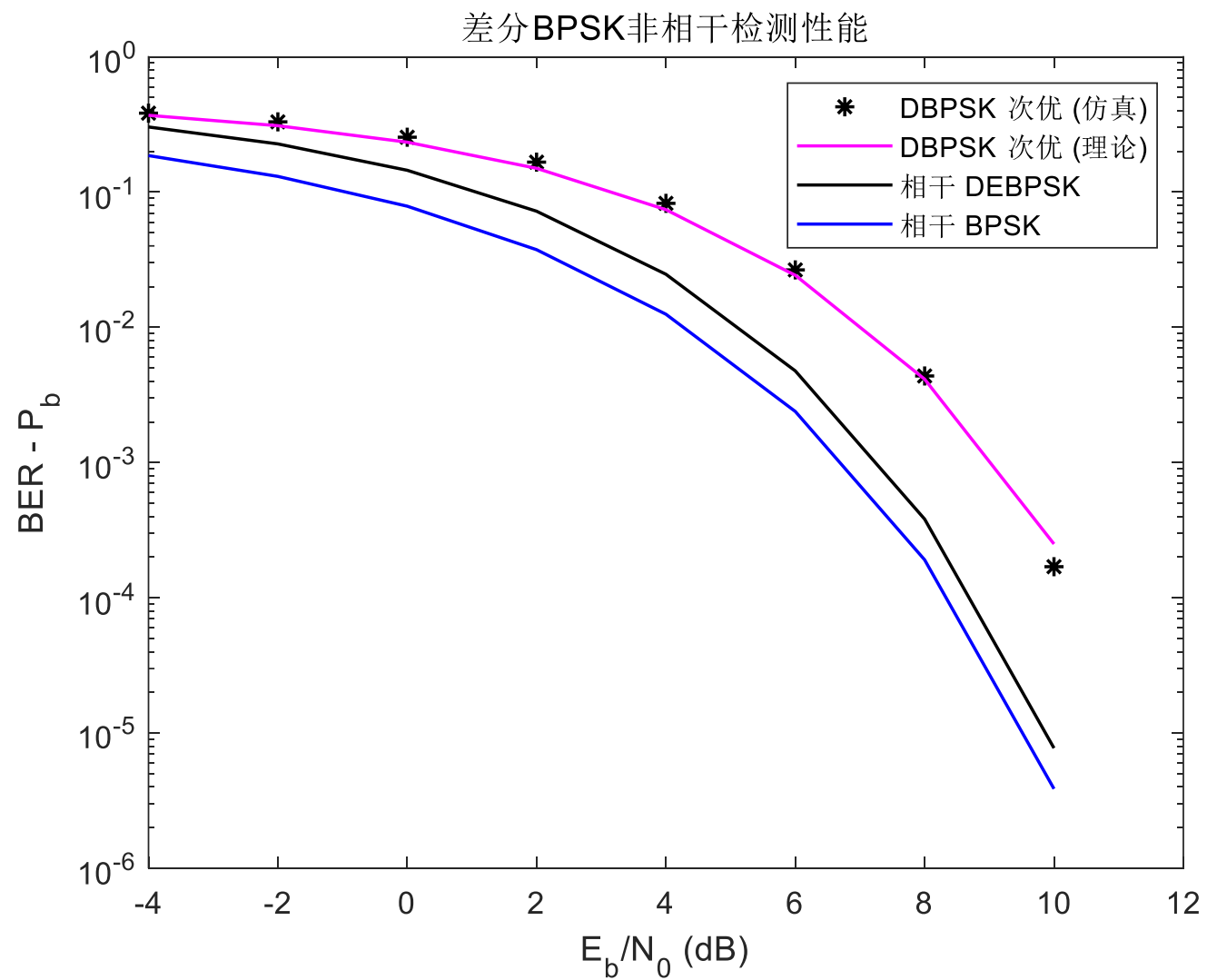
# DBPSK的仿真

## □ 程序: dbpsk\_noncoherent\_detection.m

```
05. %% 参数设置
06. N=1e5 ; %发送的符号数目
07. EbN0dB = -4:2:10; % 仿真的Eb/N0的dB范围
08. L=8; %过采样因子
09. Fc=800; %载频
10. Fs=L*Fc; %采样频率
11.
12. EbN0lin = 10.^(EbN0dB/10);
13. BER_suboptimum = zeros(length(EbN0dB),1);
14.
15. %% 发送信号调制与发射波形
16. a = randi([0,1],N,1); %二进制数据流
17. b = filter(1,[1 -1],a,0); %IIR差分编码
18. b = mod(b,2); %模2运算
19. [s_bb,t]= bpsk_mod(b,L); %基带BPSK调制
20. s = s_bb.*cos(2*pi*Fc*t/Fs); %载波调制
21.
```

```
22. %% 接收波形检测
23. for i=1:length(EbN0dB)
24.     Esym=L*sum(abs(s).^2)/length(s); %信号能量
25.     N0= Esym/EbN0lin(i); %计算功率谱密度
26.     n=sqrt(N0/2)*(randn(1,length(s))+1i*randn(1,length(s)));%noise
27.
28.     r = s + n;%通过AWGN信道
29.     p=real(r).*cos(2*pi*Fc*t/Fs);%利用带通滤波器解调到基带
30.     w0=[p zeros(1,L)];%在序列尾部补L个0
31.     w1=[zeros(1,L) p];%在序列头加L个0
32.     w = w0.*w1;%时延后相乘
33.     z = conv(w,ones(1,L));%积分
34.     u = z(L:L:end-L); % t=kTb时刻上采样
35.     a_cap = u.'<0; % 判决
36.     BER_suboptimum(i) = sum(a~=a_cap)/N;%BER统计
37. end
38. %理论BER和SER
39. theory_DBPSK_suboptimum = 0.5.*exp(-0.76*EbN0lin);
40. theory_DBPSK_coherent=erfc(sqrt(EbN0lin)).*(1-0.5*erfc(sqrt(EbN0lin)));
41. theory_BPSK_conventional = 0.5*erfc(sqrt(EbN0lin));
```

# DBPSK的仿真结果



# 正交相移键控 (QPSK)

□ QPSK是一种相位调制技术，其中两个信息位（组合为一个符号）同时被调制，选择四种可能的载波相移状态之一

□ QPSK信号定义为

$$s(t) = A \cos[2\pi f_c t + \theta_n] \quad , \quad 0 \leq t \leq T_{\text{sym}}, \quad n = 1, 2, 3, 4$$

✓ 信号相位为

$$\theta_n = (2n - 1) \frac{\pi}{4}$$

□ QPSK信号可改写为

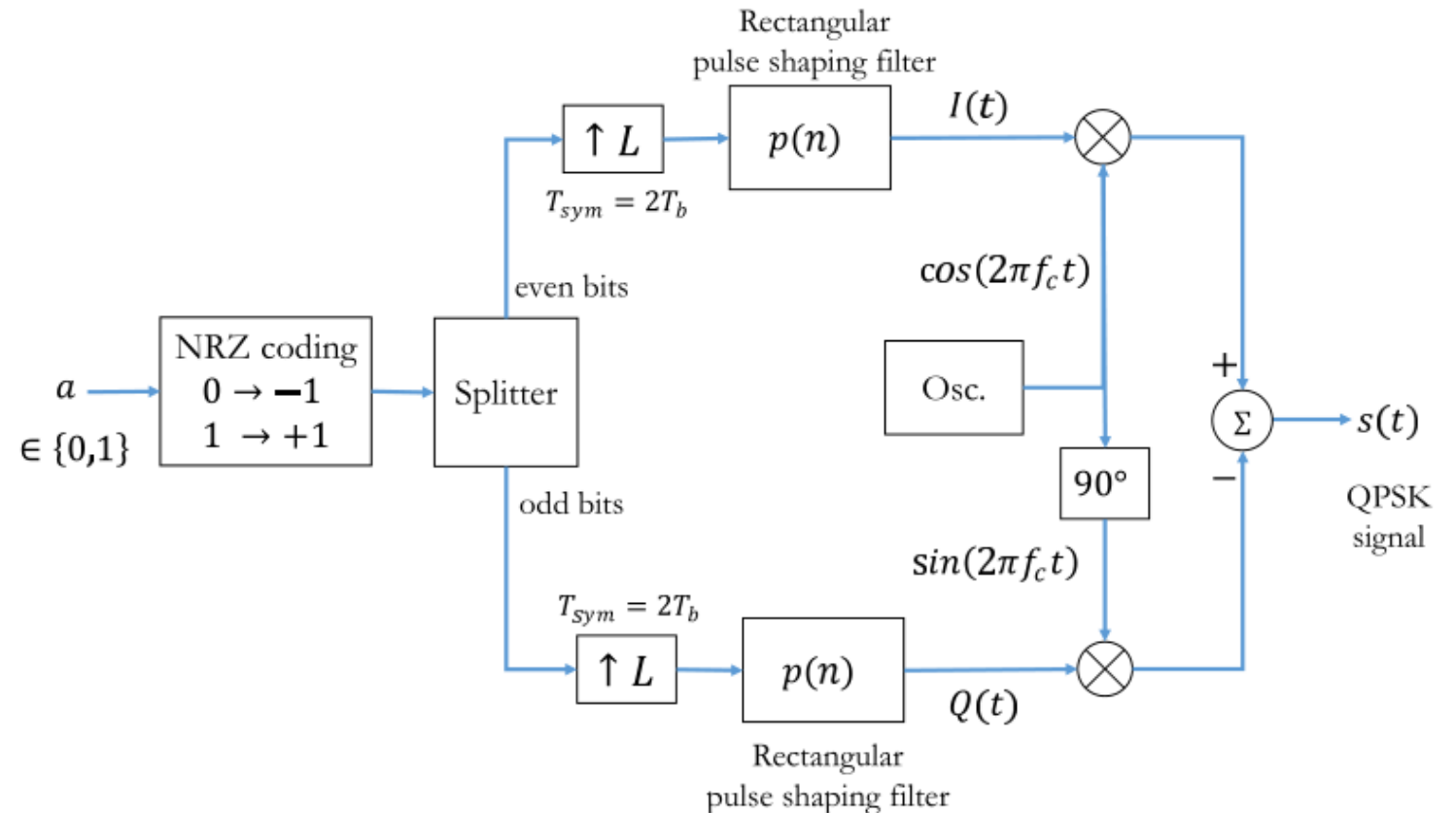
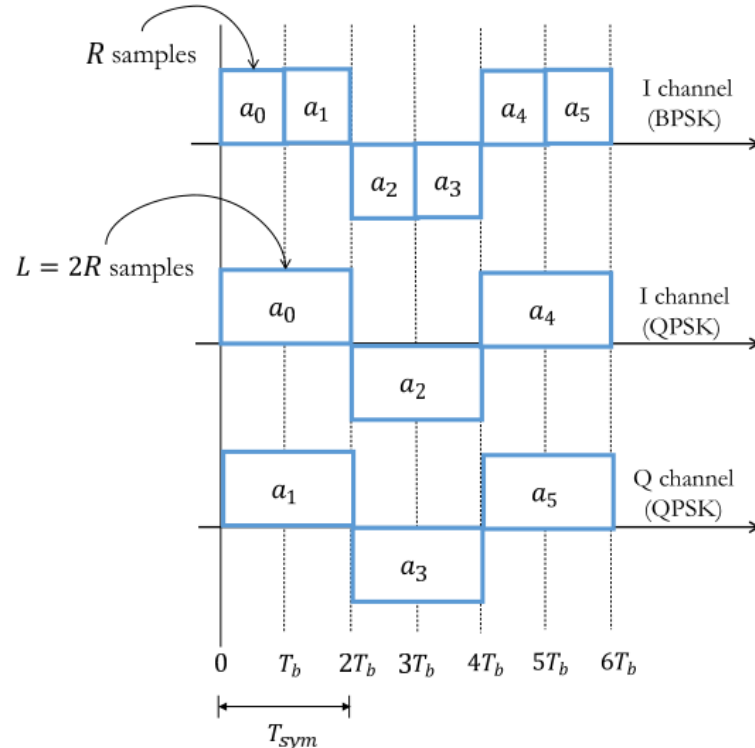
$$\begin{aligned} s(t) &= A \cos\theta_n \cos(2\pi f_c t) - A \sin\theta_n \sin(2\pi f_c t) \\ &= s_{ni}\phi_i(t) + s_{nq}\phi_q(t) \end{aligned}$$

✓ 该表达式表明可使用两个正交基函数： $\phi_i(t)$ 和 $\phi_q(t)$ 以及同相和正交信号点： $s_{ni}$ 和 $s_{nq}$ 表示信号

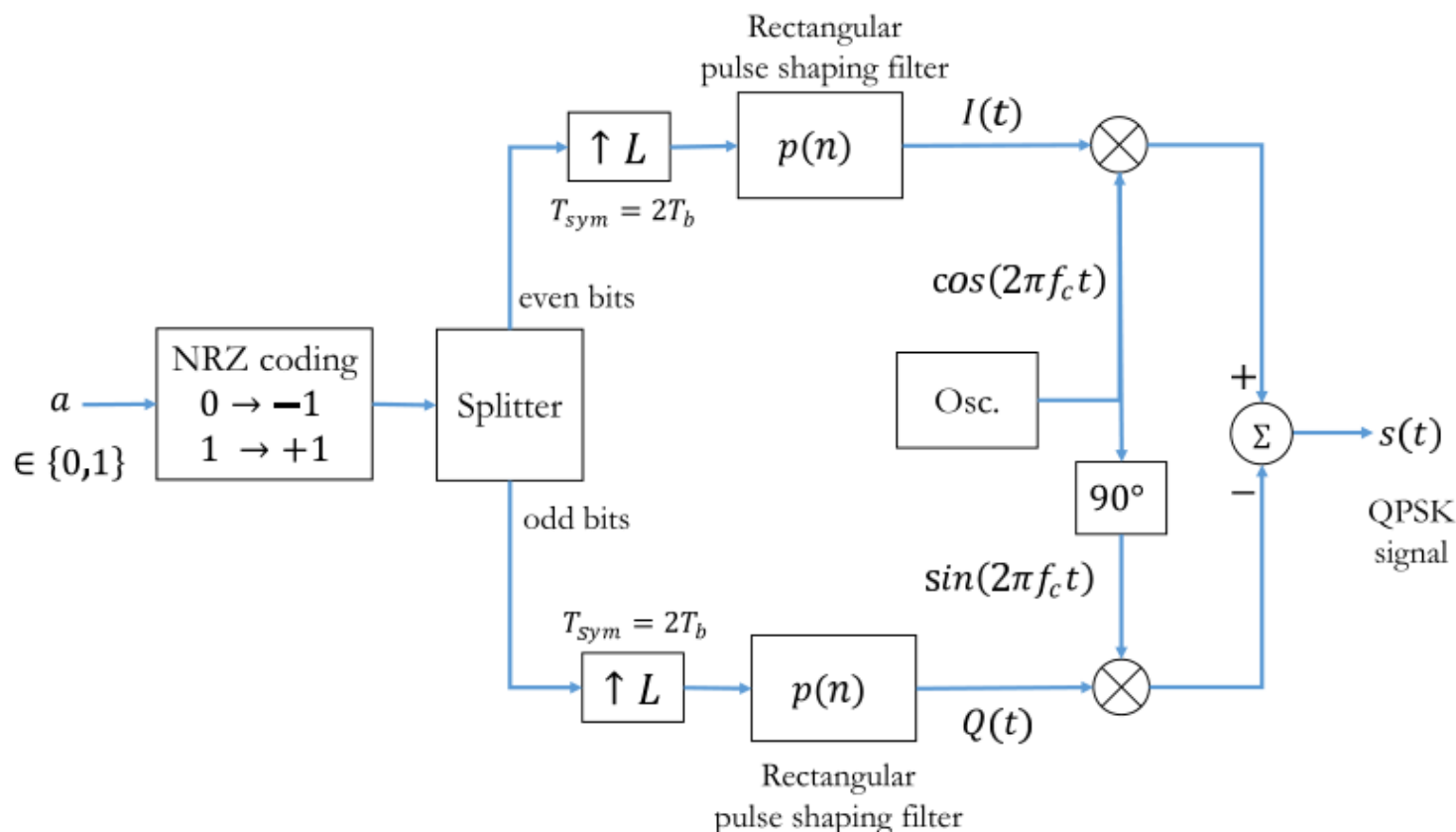
# QPSK调制

## □ QPSK发射机

- ✓ 分路器将奇偶位与生成的信息位分开
- ✓ 奇数位（正交臂）和偶数位（同相臂）的每个流以并行方式转换为NRZ格式



# QPSK调制实现



```

01. function [s,t,I,Q] = qpsk_mod(a,fc,OF)
02. % QPSK调制
03. % 输入参数
04. % a - 输入二进制数据流
05. % fc- 载频[Hz]
06. % OF - 过采样因子, fc的倍数, 至少4以上
07. % 输出参数
08. % s - QPSK载波调制信号
09. % t - 载波调制信号的时间矢量
10. % I - I路信号 (基带)
11. % Q - Q路信号 (基带)
12. L = 2*OF; %每符号的样点数目 (QPSK的每个符号2bit)
13. ak = 2*a-1; %非归零编码 0-> -1, 1->+1
14. I = ak(1:2:end); % 奇数序号数据流
15. Q = ak(2:2:end); % 偶数序号数据流
16. I= repmat(I,1,L).'; %奇数序号数据流以1/2Tb波特率
17. Q= repmat(Q,1,L).'; %偶数序号数据流以1/2Tb波特率
18. I = I(:).'; %序列化
19. Q = Q(:).'; %序列化
20.
21. fs = OF*fc; %采样率
22. t=0:1/fs:(length(I)-1)/fs; %时间点
23. iChannel = I.*cos(2*pi*fc*t);
24. qChannel = -Q.*sin(2*pi*fc*t);
25. s = iChannel + qChannel; %QPSK调制信号
26.
27. doPlot=1; %doPlot为1就画图
28. if doPlot==1,%画发送端波形
29. figure;subplot(3,2,1);plot(t,I);%I路基带波形
30. xlabel('t'); ylabel('I(t)-基带信号');xlim([0,10*L/fs]);
31. subplot(3,2,2);plot(t,Q);%Q路基带波形
32. xlabel('t'); ylabel('Q(t)-基带信号');xlim([0,10*L/fs]);
33. subplot(3,2,3);plot(t,iChannel,'r');%I路调制信号波形
34. xlabel('t'); ylabel('I(t)-载波调制信号');xlim([0,10*L/fs]);
35. subplot(3,2,4);plot(t,qChannel,'r');%Q路调制信号波形
36. xlabel('t'); ylabel('Q(t)-载波调制信号');xlim([0,10*L/fs]);
37. subplot(3,1,3);plot(t,s); %QPSK波形
38. xlabel('t'); ylabel('s(t)');xlim([0,10*L/fs]);
39. end

```

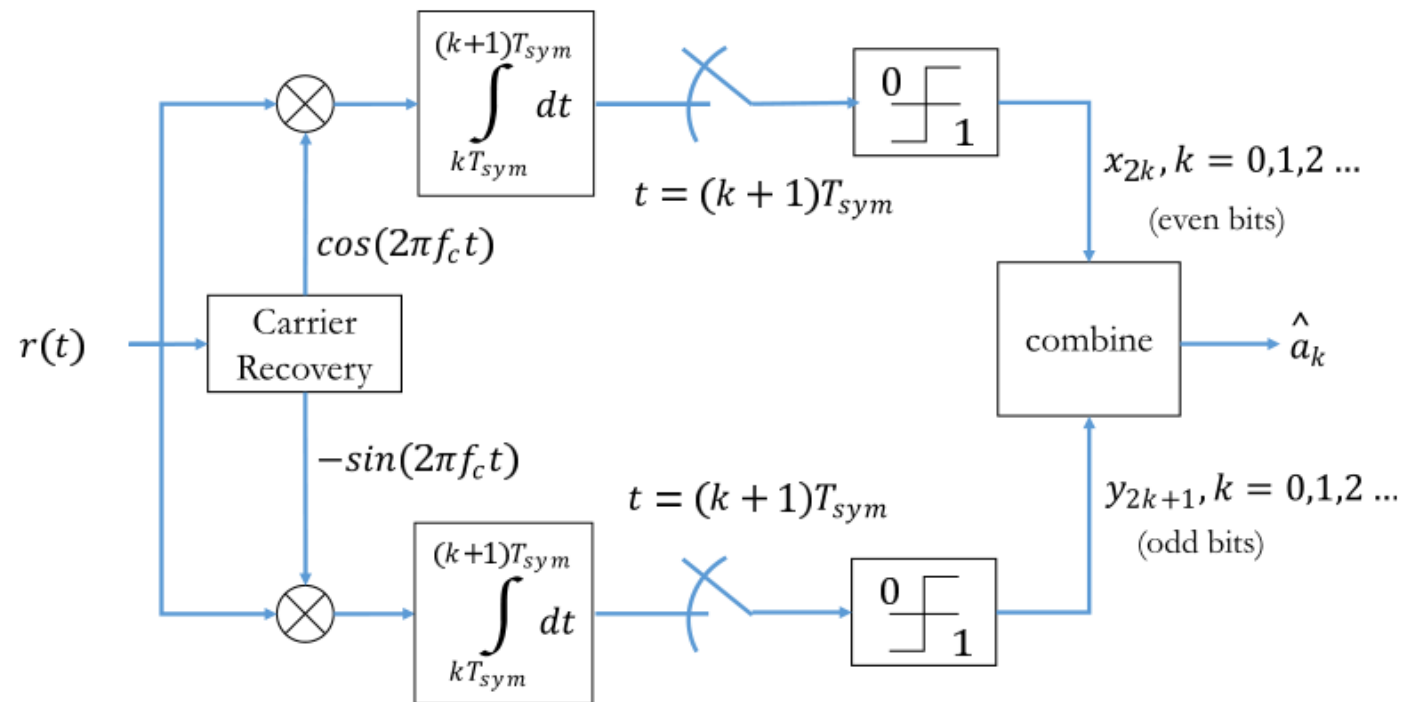
# QPSK解调

## □ QPSK接收机

- ✓ I信道和Q信道信号以与BPSK解调相同的方式单独解调
- ✓ 解调后，I信道比特和Q信道序列被组合成一个序列

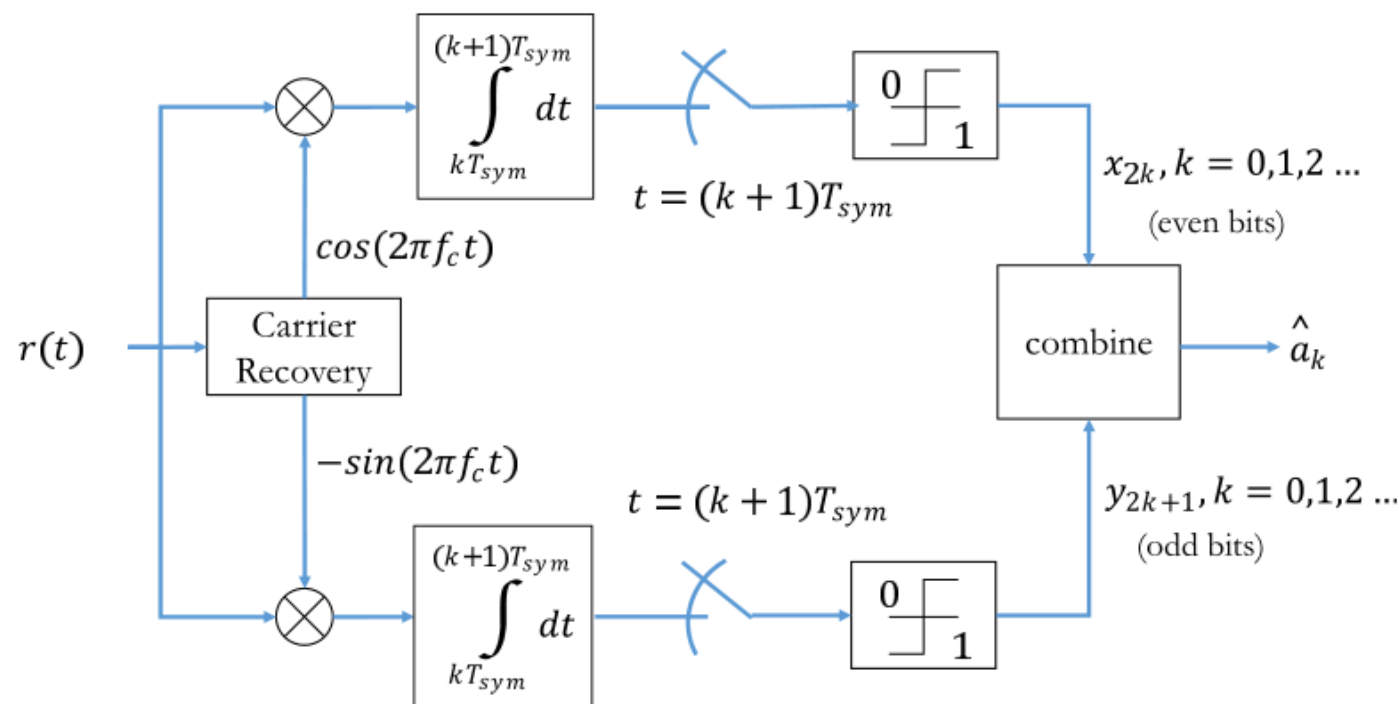
## □ AWGN信道下的性能

$$0.5\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$





# QPSK解调



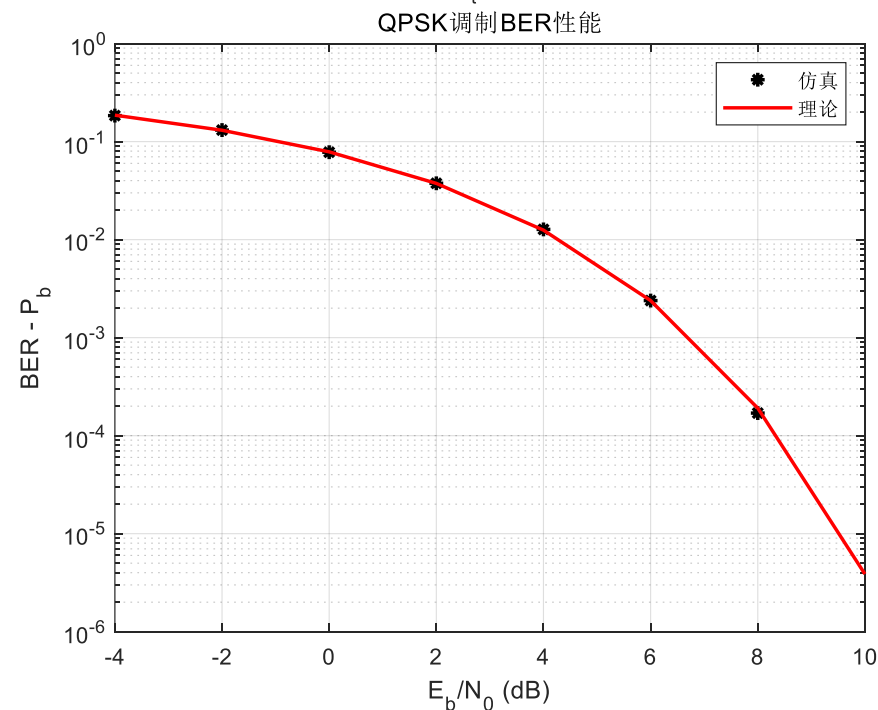
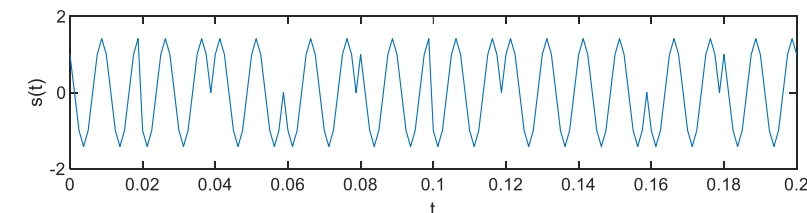
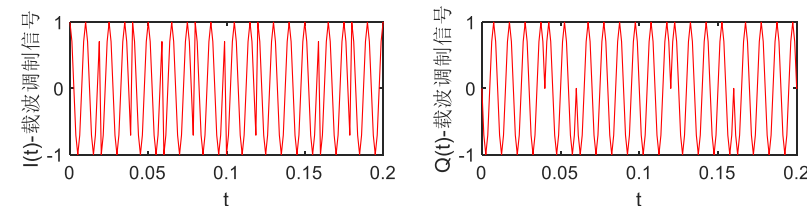
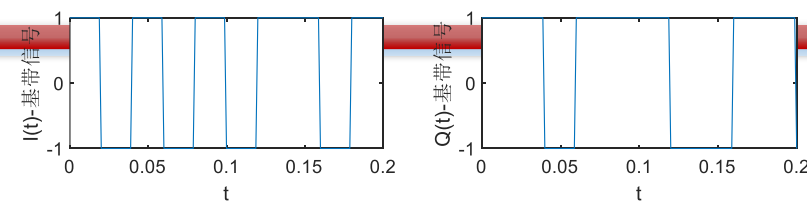
```

01. function [a_cap] = qpsk_demod(r,fc,OF)
02. % QPSK解调
03. % 输入参数
04. % r - 接收信号
05. % fc- 载频[Hz]
06. % OF - 过采样因子, fc的倍数, 至少4以上
07. % 输出参数
08. % a_cap - 检测得到的数据流
09.
10. fs = OF*fc; %采样频率
11. L = 2*OF; %一个符号周期 (两路信号) 的样点数
12. t=0:1/fs:(length(r)-1)/fs; %时间点函数
13.
14. x=r.*cos(2*pi*fc*t); %I路
15. y=-r.*sin(2*pi*fc*t); %Q路
16. x = conv(x,ones(1,L));%I路积分 (L个点, Tsym=2*Tb)
17. y = conv(y,ones(1,L));%Q路积分 (L个点, Tsym=2*Tb)
18. x = x(L:L:end) ;%I路 - 每隔Tsym取一个点
19. y = y(L:L:end); %Q路 - 每隔Tsym取一个点
20.
21. a_cap = zeros(1,2*length(x));
22. a_cap(1:2:end) = x.' > 0;
23. a_cap(2:2:end) = y.' > 0;
    
```

# QPSK仿真结果

## □ qpsk\_wfm\_sim.m

```
05. %% 参数设置
06. N=1e5 ;           %发送的符号数目
07. EbN0dB = -4:2:10; %仿真的Eb/N0的dB范围
08. fc=100;          %载频[Hz]
09. OF =8;           %过采样因子 fs=OF*fc
10.
11. EbN0lin = 10.^(EbN0dB/10);
12. BER = zeros(length(EbN0dB),1);
13.
14. a = randi([0,1],N,1); %二进制数据流
15. [s,t] = qpsk_mod(a,fc,OF); %QPSK调制
16.
17. for i=1:length(EbN0dB)
18.     Eb=OF*sum(abs(s).^2)/(length(s)); %计算信号能量
19.     N0= Eb/EbN0lin(i); %噪声功率谱
20.     n = sqrt(N0/2)*(randn(1,length(s))); %生成噪声
21.     r = s + n; %通过AWGN信道
22.     a_cap = qpsk_demod(r,fc,OF); %QPSK解调
23.     BER(i) = sum(a~=a_cap.)/N; %BER统计
24. end
25. theoreticalBER = 0.5*erfc(sqrt(EbN0lin)); %理论BER
--
```



# OQPSK与QPSK的比较

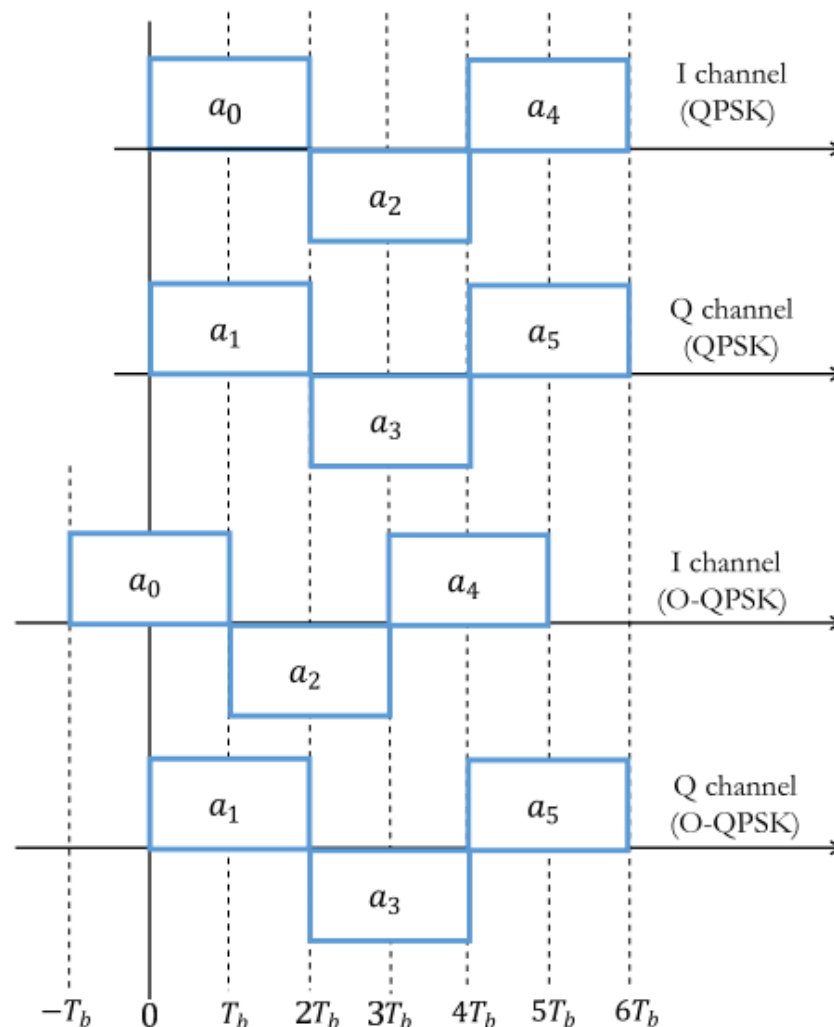
## □ QPSK的问题

- ✓ 在QPSK中, 由于I和Q信道的重合对准, 载波相位在每个符号边界处改变
- ✓ 如果I-通道或Q-通道分量中的任何一个改变符号, 则发生 $\pm 90^\circ$ 的相移; 如果两个部件都出现符号变化, 则会导致 $180^\circ$ 的相移
- ✓ QPSK中 $180^\circ$ 的相移会降低其恒定包络特性, 并且在通过带限滤波器时会产生带外辐射等有害影响

## □ OQPSK的优势

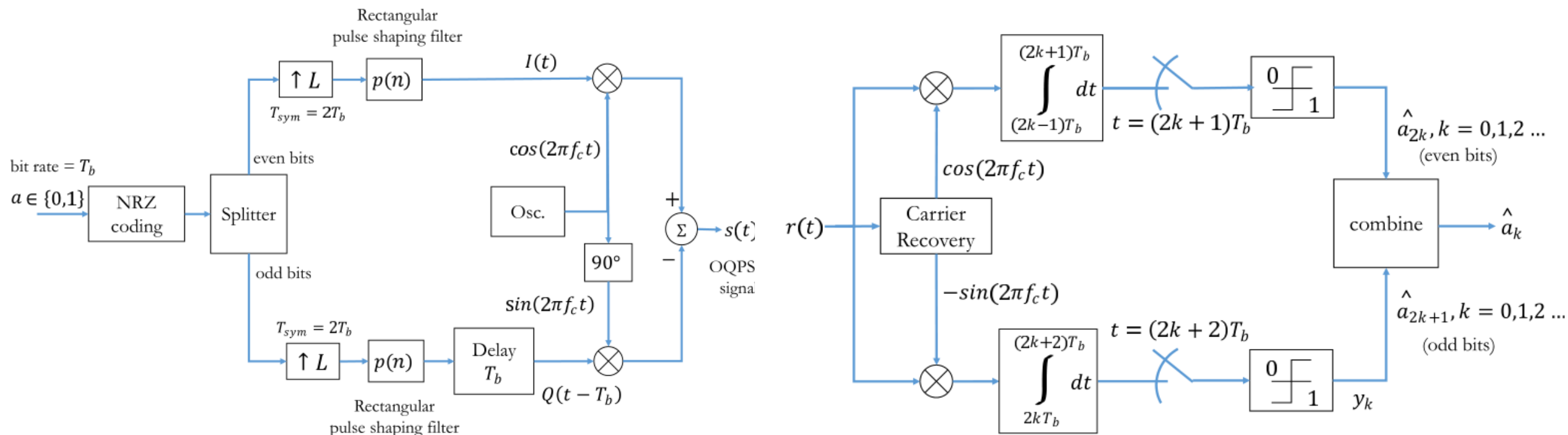
- ✓ 在OQPSK中, 正交分量不能同时改变状态, 这是因为分量仅在符号周期的中间改变状态 (由于Q信道中的半符号偏移)。这样就消除了 $180^\circ$ 的相位偏移, 并且相位变化限制在 $0^\circ$ 或 $\pm 90^\circ$ 每秒钟
- ✓ 与QPSK相比, 当带宽受限时, OQPSK的频谱保持不变。此外, 当受到相位抖动时, OQPSK的性能优于QPSK

## □ 由于OQPSK与QPSK仅在正交比特流的时间对准方面不同, 因此QPSK和OQPSK具有相同的功率谱密度和相同的误码率性能



# 偏置四相相移键控 (offset QPSK/OQPSK)

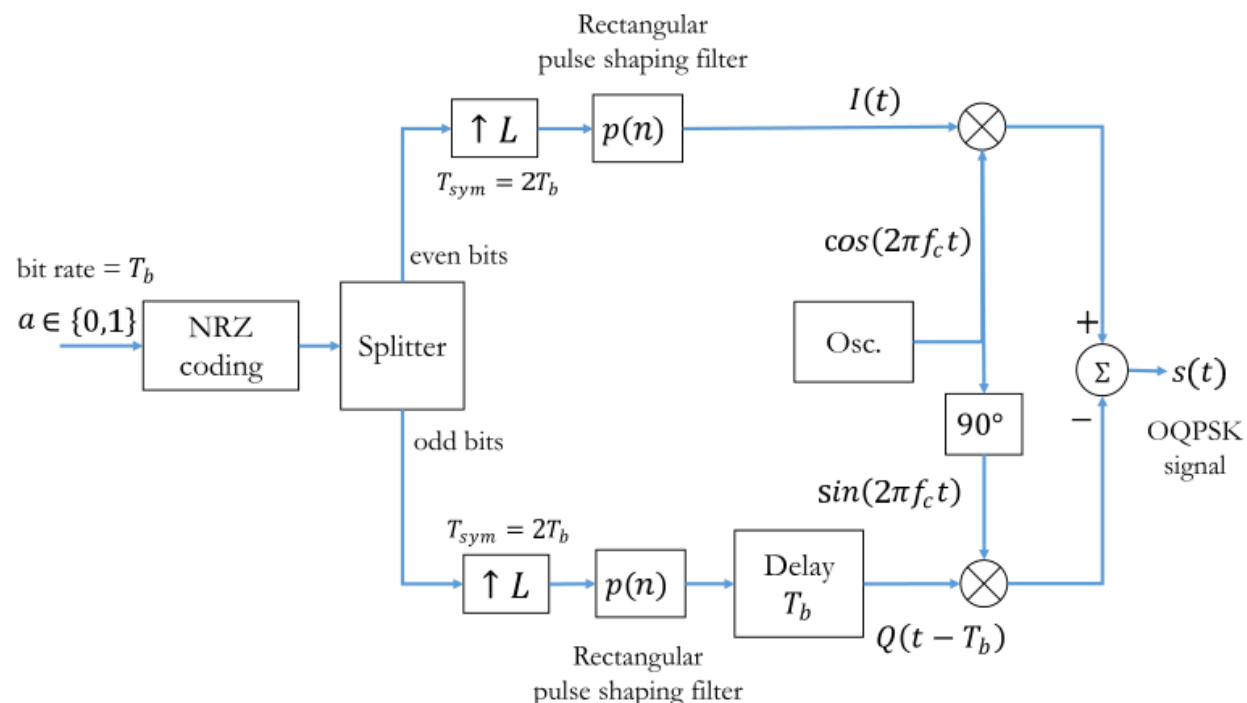
- Offset QPSK与QPSK基本相同，只是I信道和Q信道上的正交载波信号是交错的（其中一个延时的）



- Offset QPSK的信号可以表示为

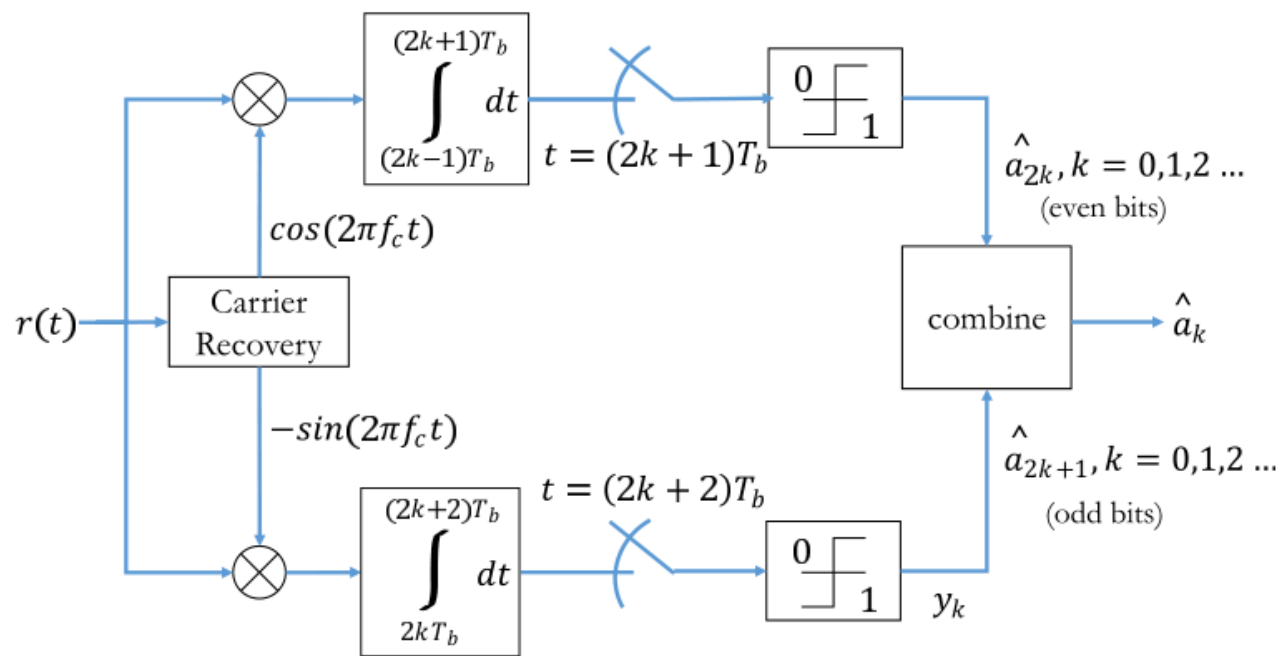
$$s(t) = I(t) \cos(2\pi f_c t) - Q\left(t - \frac{T_{sym}}{2}\right) \sin(2\pi f_c t)$$

# OQPSK调制实现



```
01. function [s,t,I,Q] = oqpsk_mod(a,fc,OF)
02. % OQPSK调制
03. % 输入参数
04. % a - 输入二进制数据流
05. % fc- 载频[Hz]
06. % OF - 过采样因子, fc的倍数, 至少4以上
07. % 输出参数
08. % s - QPSK载波调制信号
09. % t - 载波调制信号的时间矢量
10. % I - I路信号 (基带)
11. % Q - Q路信号 (基带)
12. L = 2*OF; %每符号的样点数目 (QPSK的每个符号2bit)
13. ak = 2*a-1; %非归零编码 0-> -1, 1->+1
14. I = ak(1:2:end);% 奇数序号数据流
15. Q = ak(2:2:end);% 偶数序号数据流
16. I= repmat(I,1,L).'; %奇数序号数据流以1/2Tb波特率
17. Q= repmat(Q,1,L).'; %偶数序号数据流以1/2Tb波特率
18. I = [I(:); zeros(L/2,1)].'; %I路后补半个符号周期的0
19. Q = [zeros(L/2,1); Q(:)].'; %Q路延迟半个符号周期
20.
21. fs = OF*fc; %采样函数
22. t=0:1/fs:(length(I)-1)/fs; %时间点
23. iChannel = I.*cos(2*pi*fc*t);
24. qChannel = -Q.*sin(2*pi*fc*t);
25. s = iChannel + qChannel; %OQPSK调制信号
26.
27. doPlot=1; %doPlot为1就画图
28. if doPlot==1,%画发送端波形
29. figure;subplot(3,2,1);plot(t,I);%I路基带波形
30. xlabel('t'); ylabel('I(t)-基带信号');xlim([0,10*L/fs]);
31. subplot(3,2,2);plot(t,Q);%Q路基带波形
32. xlabel('t'); ylabel('Q(t)-基带信号');xlim([0,10*L/fs]);
33. subplot(3,2,3);plot(t,iChannel,'r');%I路调制信号波形
34. xlabel('t'); ylabel('I(t)-载波调制信号');xlim([0,10*L/fs]);
35. subplot(3,2,4);plot(t,qChannel,'r');%Q路调制信号波形
36. xlabel('t'); ylabel('Q(t)-载波调制信号');xlim([0,10*L/fs]);
37. subplot(3,1,3);plot(t,s); %QPSK波形
38. xlabel('t'); ylabel('s(t)');xlim([0,10*L/fs]);
39. end
```

# OQPSK解调



```

01. function [a_cap] = oqpsk_demod(r,N,fc,OF)
02. % OQPSK解调
03. % 输入参数
04. % r - 接收信号
05. % N -发送符号数目
06. % fc- 载频[Hz]
07. % OF - 过采样因子, fc的倍数, 至少4以上
08. % 输出参数
09. % a_cap - 检测得到的数据流
10.
11. fs = OF*fc; %采样频率
12. L = 2*OF; %一个符号周期 (两路信号) 的样点数
13. t=0:1/fs:(length(r)-1)/fs; %时间点函数
14.
15. x=r.*cos(2*pi*fc*t); %I路
16. y=-r.*sin(2*pi*fc*t); %Q路
17. x = conv(x,ones(1,L));%I路积分 (L个点, Tsym=2*Tb)
18. y = conv(y,ones(1,L));%Q路积分 (L个点, Tsym=2*Tb)
19. x = x(L:L:end-L); %I路 - 每隔Tsym取一个点
20. y = y(L+L/2:L:end-L/2); %Q路 - 每隔Tsym取一个点, 时延半个符号周期
21.
22. a_cap = zeros(N,1);
23. a_cap(1:2:end) = x.' > 0;
24. a_cap(2:2:end) = y.' > 0;
25.

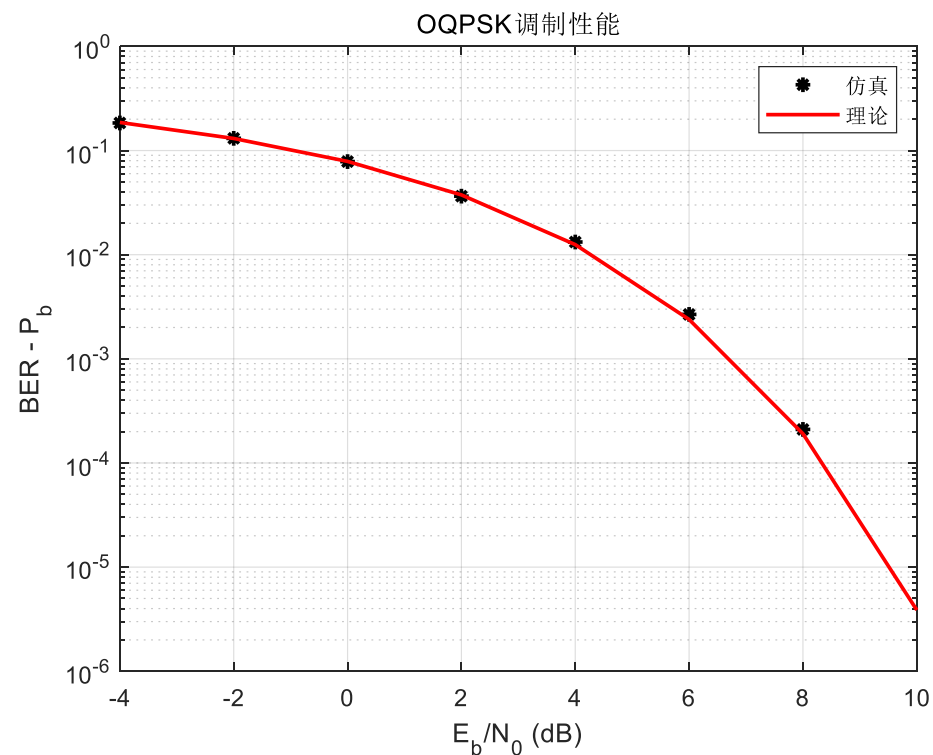
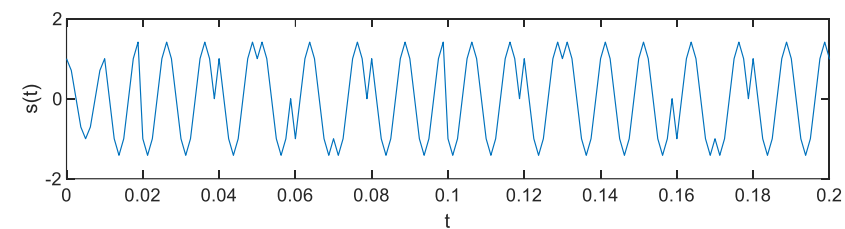
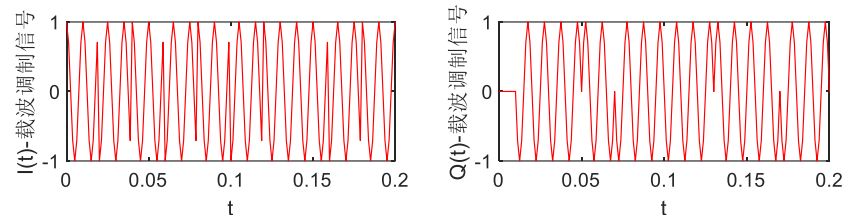
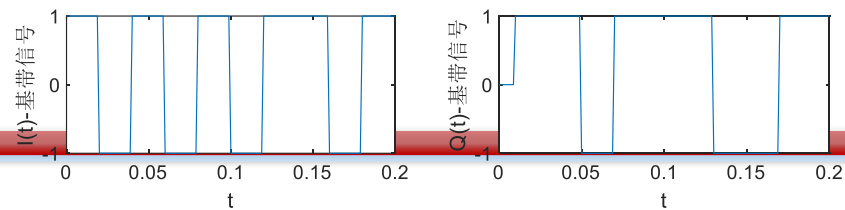
```



# OQPSK仿真

## □ oqpsk\_wfm\_sim.m

```
05. %% 参数设置
06. N=1e5 ; %发送的符号数目
07. EbN0dB = -4:2:10; %仿真的Eb/N0的dB范围
08. fc=100; %载频[Hz]
09. OF =8; %过采样因子 fs=OF*fc
10.
11. EbN0lin = 10.^(EbN0dB/10);
12. BER = zeros(length(EbN0dB),1);
13.
14. a = randi([0,1],N,1); %二进制数据流
15. [s,t] = oqpsk_mod(a,fc,OF); %OQPSK 调制
16.
17. for i=1:length(EbN0dB)
18.     Eb=OF*sum(abs(s).^2)/(length(s)); %计算信号能量
19.     N0= Eb/EbN0lin(i); %噪声功率谱
20.     n = sqrt(N0/2)*(randn(1,length(s))); %生成噪声
21.     r = s + n; %通过AWGN信道
22.     a_cap = oqpsk_demod(r,N,fc,OF); %QPSK解调
23.     BER(i) = sum(a~=a_cap)/N; %BER统计
24. end
25. theoreticalBER = 0.5*erfc(sqrt(EbN0lin)); %理论BER
26.
27. %% 结果展示
28. figure; semilogy(EbN0dB, BER, 'k*', 'LineWidth', 1.5); %simulated BER
29. hold on; semilogy(EbN0dB, theoreticalBER, 'r-', 'LineWidth', 1.5);
30. title('OQPSK调制性能');
31. xlabel('E_b/N_0 (dB)'); ylabel('BER - P_b');
32. legend('仿真', '理论');
```



# 连续相位调制 (CPM)

- 传统的PSK或FSK在传输信号中引入了突变的相位不连续性
  - ✓ 突然的相位不连续会产生旁瓣，并可能对附近的通道造成干扰
  - ✓ 为无线电通信选择的调制方案应该产生频谱有效的信号（最小化副瓣效应），以便在分配的频带内可以为更多用户服务，这是非常理想的
  - ✓ 相位不连续性在通过功率效率高的非线性放大器时也会引入不希望的频率旁瓣
- 解决思路
  - ✓ 限制发送信号的相位变化范围，如使用OQPSK可以在旁瓣电平上取得一些改善
  - ✓ 如果载波相位可以从一个符号连续到另一个符号，则可以降低副瓣电平
- 在CPM中，载波的相位从一个符号逐渐变为另一个符号。瞬时相位取决于先前符号的相位，因此CPM是一种带记忆的调制
  - ✓ 由于一个符号的相位依赖于先前符号的相位，因此CPM的调制和解调更为复杂
  - ✓ 利用Viterbi算法实现的类MLSE（极大似然序列估计）技术，可以很好地利用CPM调制信号的相位轨迹



# 连续相位频移键控调制 (CPFSK)

□ 连续相移键控 (CPFSK) 是频移键控 (FSK) 技术的一种变体, 它消除了相位不连续性

- ✓ 在传统的FSK调制中, 瞬时载波频率在两个频率之间切换
- ✓ 由于两个频率之间的突然切换, 传统FSK的相位不连续

□ 在CPFSK, 发射机信号定义为

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos [2\pi f_c t + \theta(t)]$$

- ✓  $E_b$  为每比特平均能量,  $T_b$  为比特周期,  $f_c$  为载频

- ✓  $\theta(t)$  为相位演化, 其导数引起CPFSK的瞬时角频移  $\theta(t) = \theta(0) + \frac{\pi h}{T_b} \int_0^t b(t) dt$

- ✓  $\theta(0)$  是  $t=0$  之前的累积相位历史

- ✓ 因子  $h$  为调制指数, 是频率偏差的度量 ( $h=1$  对应于二进制CPFSK,  $h=0.5$  对应于最小移位键控 (MSK) 调制)

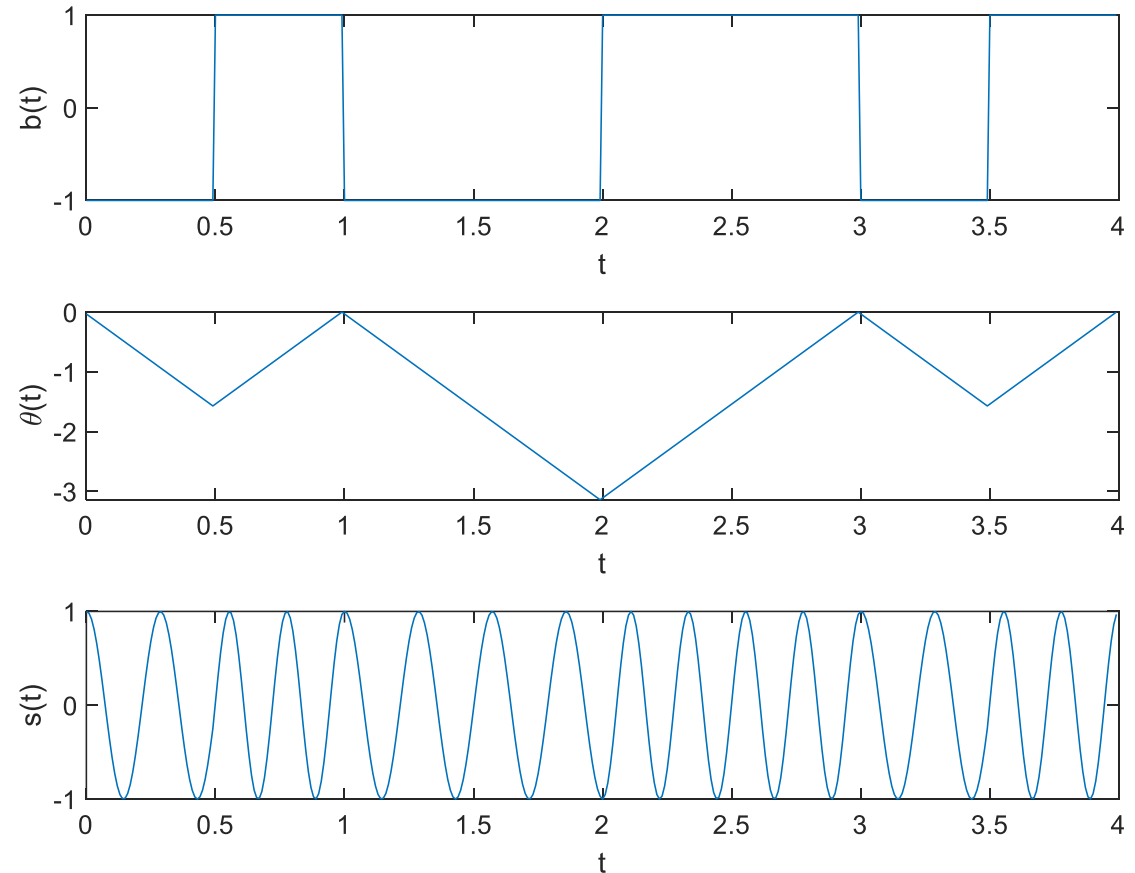
- ✓  $b(t) \in (\pm 1)$  是表示二进制信息序列  $a$  的波形,  $b=+1$  表示  $a=0$ ,  $b=-1$  表示  $a=1$

# CPFSK波形的仿真实现

## □ 程序: cpfsk.m

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_c t + \theta(t)] \quad \theta(t) = \theta(0) + \frac{\pi h}{T_b} \int_0^t b(t) dt$$

```
01. clear all;clc;close all
02. rng default
03.
04. L=50;           %过采样因子
05. Tb=0.5;        %比特周期[秒]
06. fs=L/Tb;       %采样频率[Hz]
07. fc=2/Tb;       %载频
08. N = 8;         %发送的比特数
09. h=0.5;         %调制指数
10. b=2*(rand(N,1)>0.5)-1;%随机信息序列 +1/-1
11. b=repmat(b,1,L).'; %过采样
12. b=b(:).';      %序列化
13. theta= pi*h/Tb*filter(1,[1 -1],b,0)/fs;%FIR积分滤波器
14. t=0:1/fs:Tb*N-1/fs; %时间矢量
15. s = cos(2*pi*fc*t + theta); %CPFSK signal
16.
17. subplot(3,1,1);plot(t,b);xlabel('t');ylabel('b(t)');
18. subplot(3,1,2);plot(t,theta);xlabel('t');ylabel('\theta(t)');
19. subplot(3,1,3);plot(t,s);xlabel('t');ylabel('s(t)');
```



□ 注意：相位轨迹中存在相位转移模糊度问题（+pi和-pi是一样的），因此2进制CPFSK无法使用相位信息接收

# 最小移位键控 (MSK)

## □ 最小移位键控是一种真正的CPM调制技术

- ✓ MSK调制提供了通信工程师所喜爱的所有期望质量-包括恒定的包络、与QPSK和OQPSK相比非常紧凑的频谱以及良好的误码率性能

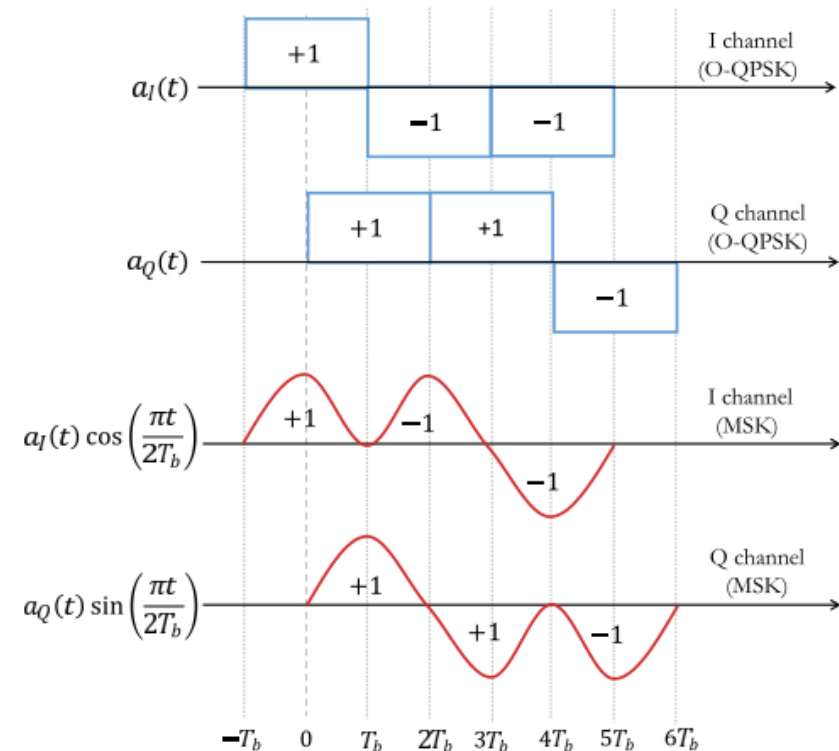
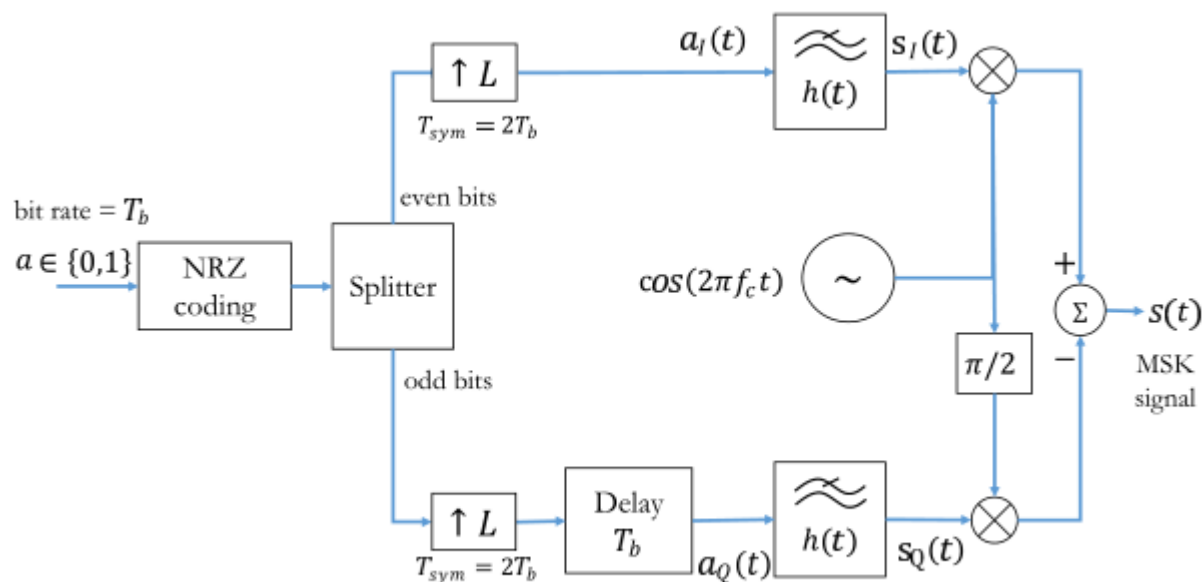
$$\begin{aligned} s(t) &= \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_c t + \theta(t)] & \theta(0) = 0, h=0.5 \\ \theta(t) &= \theta(0) + \frac{\pi h}{T_b} \int_0^t b(t) dt & \longrightarrow \\ s(t) &= \sqrt{\frac{2E_b}{T_b}} \cos\left[\theta(0) \pm \frac{\pi}{2T_b} t\right] \cos(2\pi f_c t) - \sqrt{\frac{2E_b}{T_b}} \sin\left[\theta(0) \pm \frac{\pi}{2T_b} t\right] \sin(2\pi f_c t) \\ &= s_I(t) \cos(2\pi f_c t) - s_Q(t) \sin(2\pi f_c t), \quad 0 \leq t \leq 2T_b \\ s_I(t) &= \pm \sqrt{\frac{2E_b}{T_b}} \cos\left[\frac{\pi}{2T_b} t\right], \quad -T_b \leq t \leq T_b \\ s_Q(t) &= \pm \sqrt{\frac{2E_b}{T_b}} \sin\left[\frac{\pi}{2T_b} t\right], \quad 0 \leq t \leq 2T_b \\ \longrightarrow & s(t) = \sqrt{\frac{2}{T_b}} a_I(t) \cos\left[\frac{\pi}{2T_b} t\right] \cos(2\pi f_c t) - \sqrt{\frac{2}{T_b}} a_Q(t) \sin\left[\frac{\pi}{2T_b} t\right] \sin(2\pi f_c t) \end{aligned}$$

$a_I(t)$  和  $a_Q(t)$  分别是I信道和Q信道中的随机信息序列

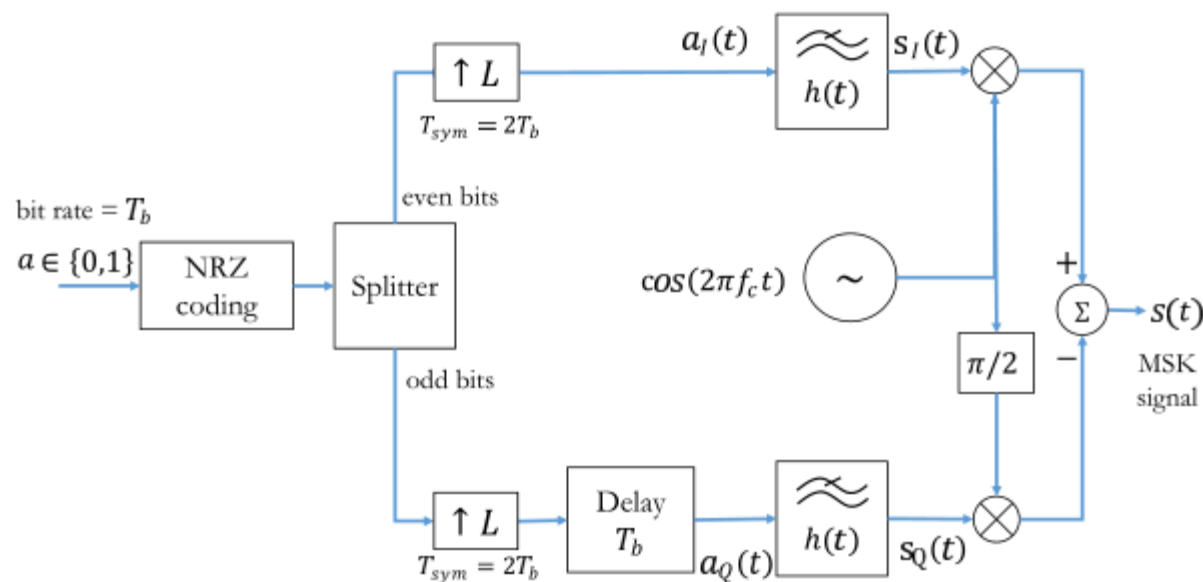
# MSK调制器的结构

- MSK调制器的实现形式有很多种，其中有一种将MSK看作OQPSK的一种特例，因此调制器结构类似
  - ✓ 同相臂和正交臂上的脉冲整形功能使用相同低通滤波器来实现

$$h(t) = \begin{cases} \sin\left(\frac{\pi}{2T_b}t\right), & 0 \leq t \leq 2T_b \\ 0, & \text{otherwise} \end{cases}$$



# MSK调制器的实现



```

01. function [s,t,I,Q] = msk_mod(a,fc,OF)
02. % MSK调制
03. % 输入参数
04. % a - 输入二进制数据流
05. % fc- 载频[Hz]
06. % OF - 过采样因子, fc的倍数, 至少4以上
07. % 输出参数
08. % s - QPSK载波调制信号
09. % t - 载波调制信号的时间矢量
10. % I - I路信号 (基带)
11. % Q - Q路信号 (基带)
12. ak = 2*a-1; %NRZ编码 0-> -1, 1->+1
13. ai = ak(1:2:end); % 奇数序号数据流
14. aq = ak(2:2:end); % 偶数序号数据流
15.
16. L=2*OF; %符号采样率 Tsym=2xTb
17. %上采样和序列化 I路和Q路
18. ai = [ai zeros(length(ai),L-1)]; ai=ai.';ai=ai(:);
19. aq = [aq zeros(length(aq),L-1)]; aq=aq.';aq=aq(:);
20.
21. ai = [ai(:) ; zeros(L/2,1)].'; % 矢量后补半个符号周期的0
22. aq = [zeros(L/2,1); aq(:)].'; % 时延半个周期
23.
24. %构造低通滤波器
25. Fs=OF*fc;
26. Ts=1/Fs;
27. Tb = OF*Ts;
28. t=0:Ts:2*Tb;
29. h = sin(pi*t/(2*Tb));%低通滤波器
30. % 两路信号通过低通滤波器
31. I = filter(h,1,ai);%I路基带信号
32. Q = filter(h,1,aq);%Q路基带信号
33.
34. t=(0:1:length(I)-1)*Ts;
35. iChannel = I.*cos(2*pi*fc*t);
36. qChannel = Q.*sin(2*pi*fc*t);
37. s = iChannel-qChannel; %带通MSK调制信号
    
```

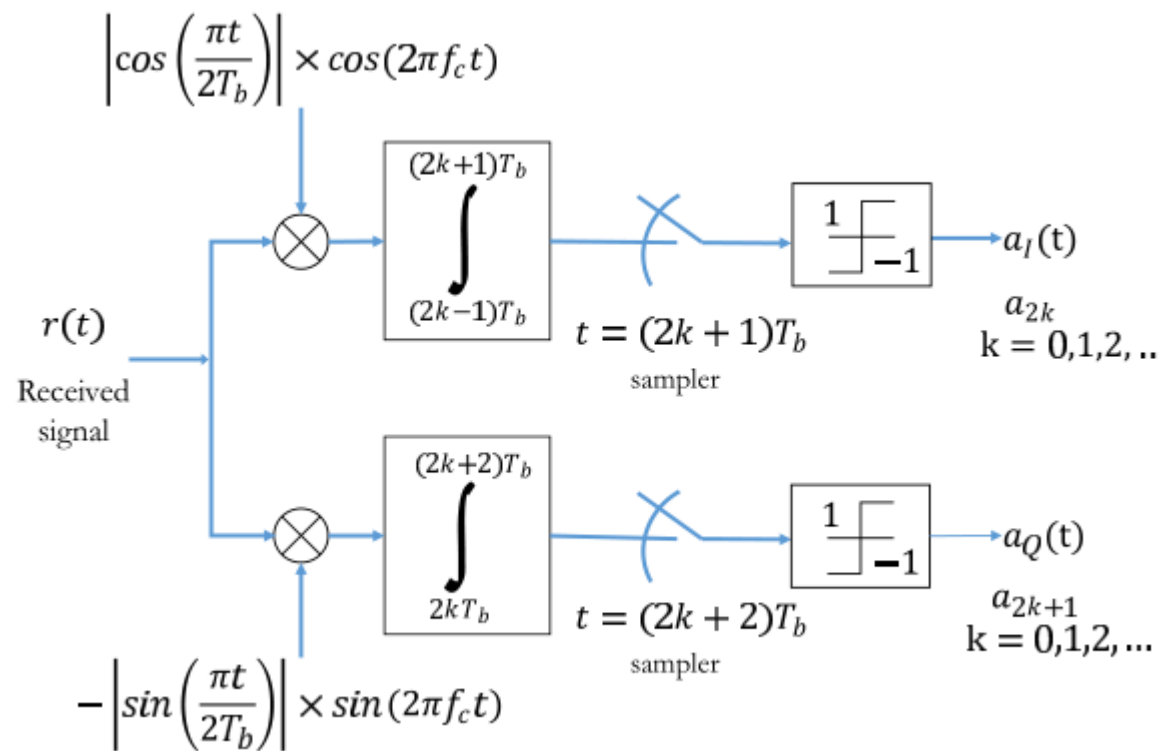
# MSK解调器结构

## □ MSK解调器

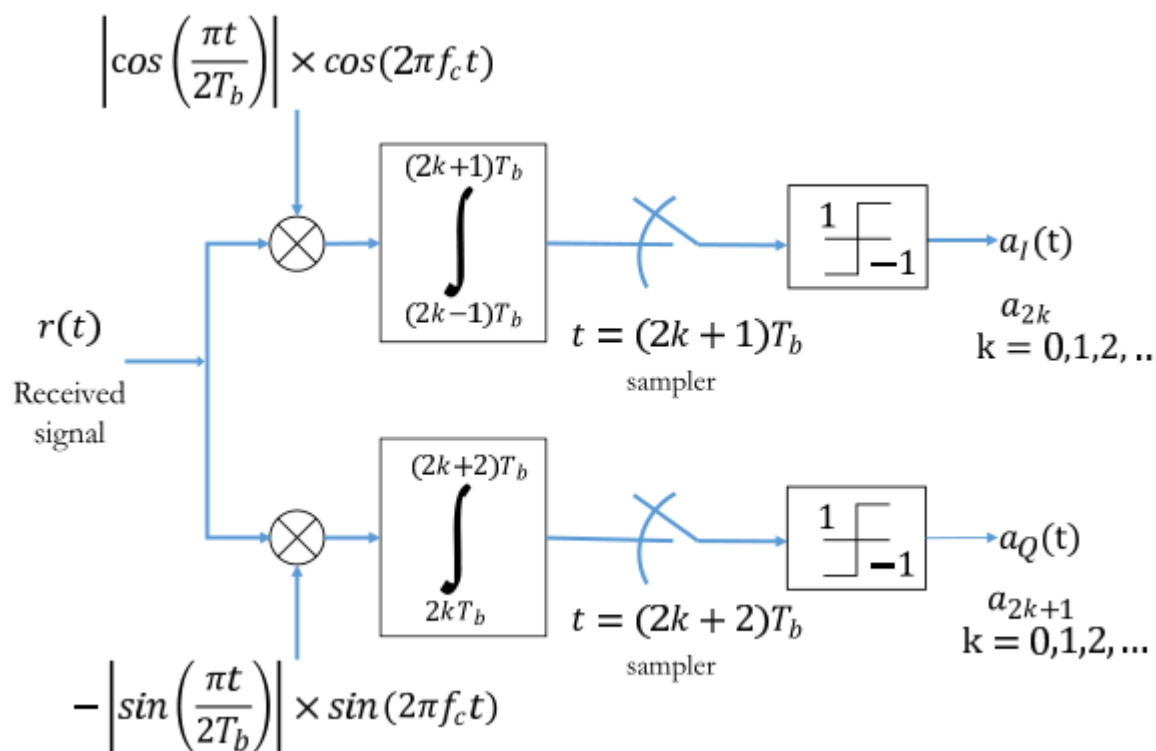
- ✓ 同相和正交分支中的信号乘以半周期函数的绝对值
- ✓ 其余部分与OQPSK接收机类似

□ MSK调制可以解释为在I信道和Q信道上有两个类似BPSK的调制器，唯一的区别是MSK的半周期正弦成形的半符号移位，因此，MSK在AWGN信道下与BPSK的性能一样

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



# MSK解调器实现



```

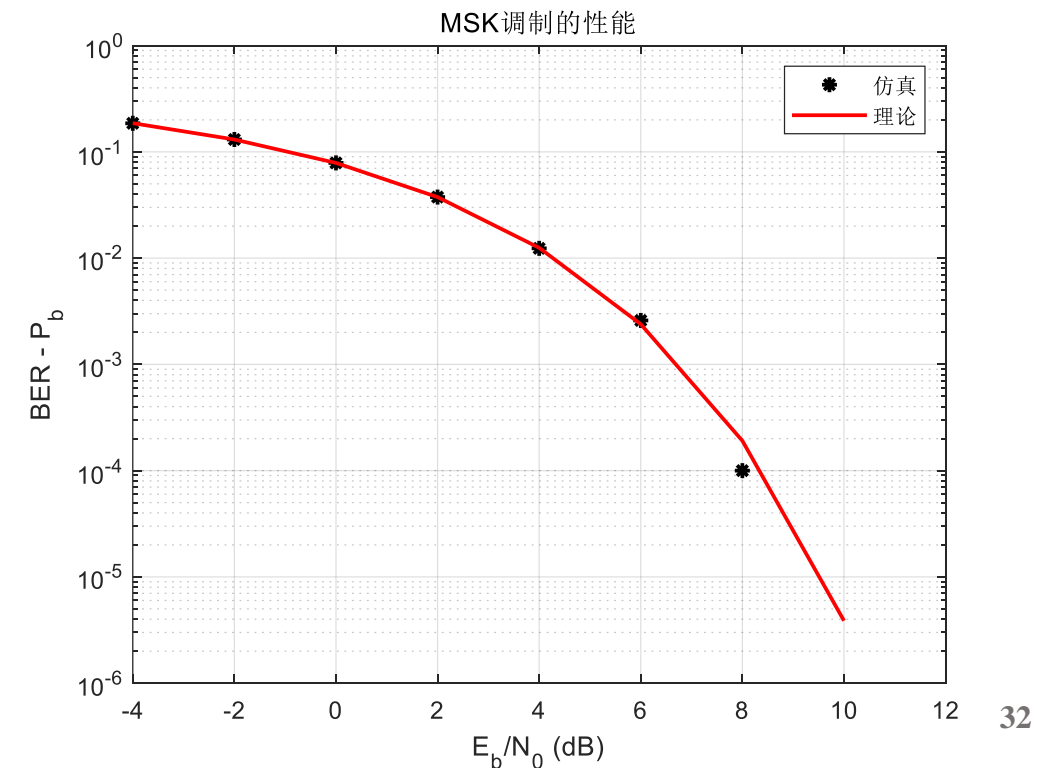
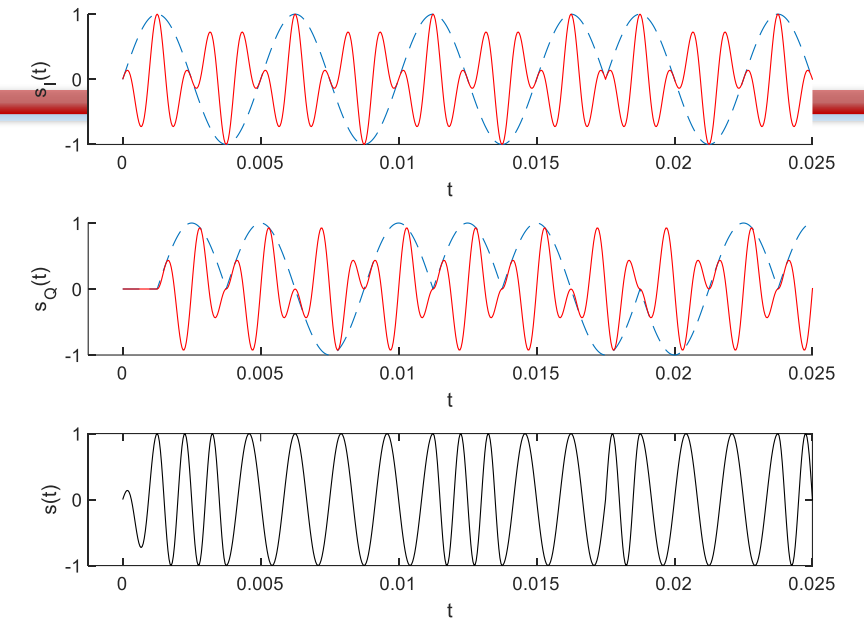
01. function [a_cap] = msk_demod(r,N,fc,OF)
02. % MSK解调
03. % 输入参数
04. % r - 接收信号
05. % N - 发送符号数目
06. % fc- 载频[Hz]
07. % OF - 过采样因子, fc的倍数, 至少4以上
08. % 输出参数
09. % a_cap - 检测得到的数据流
10.
11. L = 2*OF; %一个符号周期的采样数
12. %cosine and sine functions for half-sinusoid shaping
13. Fs=OF*fc;
14. Ts=1/Fs;
15. Tb = OF*Ts;
16. t=(-OF:1:length(r)-OF-1)/Fs;
17. x=abs(cos(pi*t/(2*Tb)));
18. y=abs(sin(pi*t/(2*Tb)));
19.
20. u=r.*x.*cos(2*pi*fc*t);
21. v=-r.*y.*sin(2*pi*fc*t);
22. iHat = conv(u,ones(1,L));%I路积分 (L个点, Tsym=2*Tb)
23. qHat = conv(v,ones(1,L));%Q路积分 (L个点, Tsym=2*Tb)
24. iHat= iHat(L:L:end-L);%I路 - 每隔Tsym取一个点
25. qHat= qHat(L+L/2:L:end-L/2);%Q路 - 每隔Tsym取一个点, 时延半个符号周期
26.
27. a_cap = zeros(N,1);
28. a_cap(1:2:end) = iHat > 0;
29. a_cap(2:2:end) = qHat > 0;
    
```



# MSK仿真结果

## □ msk\_wfm\_sim.m

```
05.  %% 参数设置
06.  N=1e5 ;           %发送的符号数目
07.  EbN0dB = -4:2:10; %仿真的Eb/N0的dB范围
08.  fc=800;          %载频[Hz]
09.  OF =32;           %过采样因子 fs=OF*fc
10.
11.  EbN0lin = 10.^(EbN0dB/10);
12.  BER = zeros(length(EbN0dB),1);
13.
14.  a = randi([0,1],N,1); %二进制数据流
15.  [s,t] = msk_mod(a,fc,OF); %MSK调制
16.
17.  for i=1:length(EbN0dB),
18.      Eb=OF*sum(abs(s).^2)/(length(s)); %计算信号能量
19.      N0= Eb/EbN0lin(i); %噪声功率谱密度
20.      n = sqrt(N0/2)*(randn(1,length(s))); %生成噪声
21.      r = s + n;
22.      a_cap = msk_demod(r,N,fc,OF);
23.      BER(i) = sum(a~=a_cap)/N; %统计
24.  end
25.  theoreticalBER = 0.5*erfc(sqrt(EbN0lin)); %理论BER
26.
27.  %% 结果展示
28.  figure; semilogy(EbN0dB,BER,'k*','LineWidth',1.5); hold on; %simulated BER
29.  semilogy(EbN0dB,theoreticalBER,'r-','LineWidth',1.5); %theoretical BER
30.  set(gca,'XLim',[-4 12]); set(gca,'YLim',[1E-6 1E0]); set(gca,'XTick',-4:2:12);
31.  title('MSK调制的性能');
32.  xlabel('E_b/N_0 (dB)'); ylabel('BER - P_b');
33.  legend('仿真','理论'); grid on;
```





# 相位转移特性研究

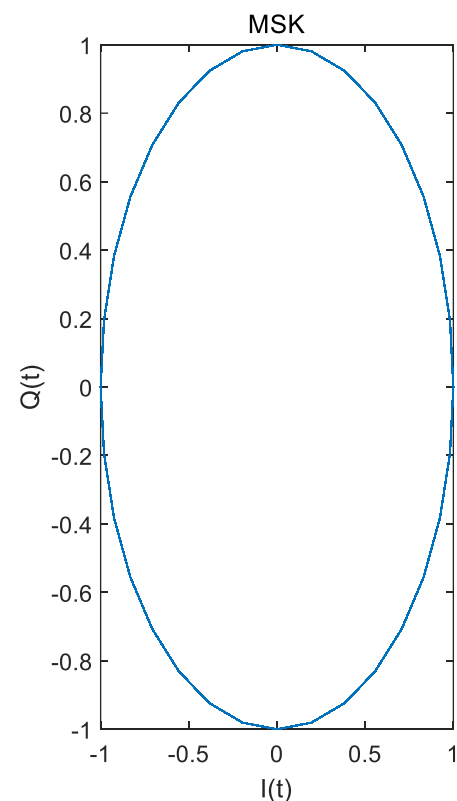
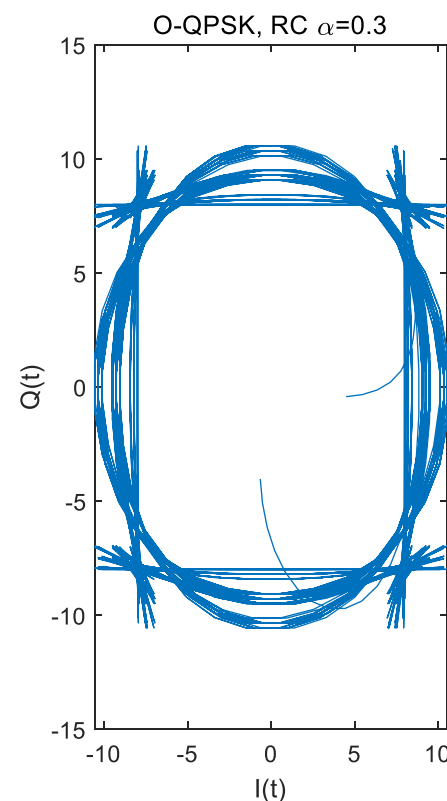
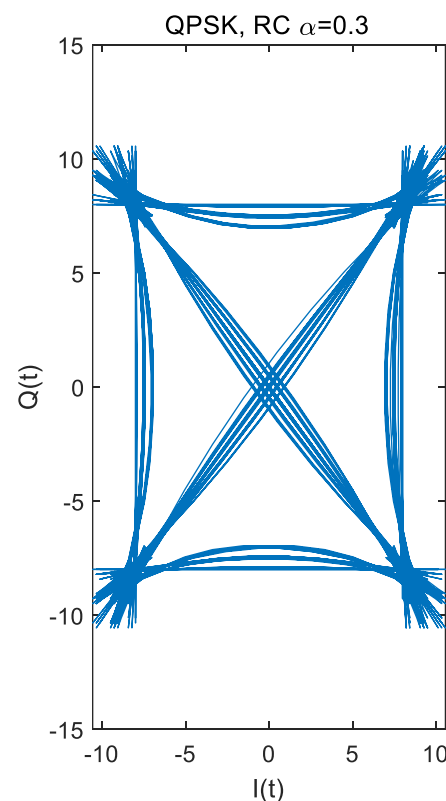
- 利用星座图可以方便地研究QPSK方案和MSK的不同变体的相变特性
- 在实际应用中，基带调制波形通过脉冲整形滤波器以消除符号间干扰（ISI）现象，这里选择升余弦滤波器

$$P(f) = \begin{cases} T_{sym}, & |f| \leq \frac{1-\alpha}{2T_{sym}} \\ \frac{T_{sym}}{2} \left[ 1 + \cos \left( \frac{\pi T_{sym}}{\alpha} \left[ |f| - \frac{1-\alpha}{2T_{sym}} \right] \right) \right], & \frac{1-\alpha}{2T_{sym}} \leq f \leq \frac{1+\alpha}{2T_{sym}} \\ 0, & |f| \geq \frac{1+\alpha}{2T_{sym}} \end{cases}$$

- 生成调制波形（如QPSK，O-QPSK，MSK方案），并将它们通过升余弦滤波器
  - ✓ 绘制星座图，通过星座点之间的转移轨迹，分析调制方案的特性

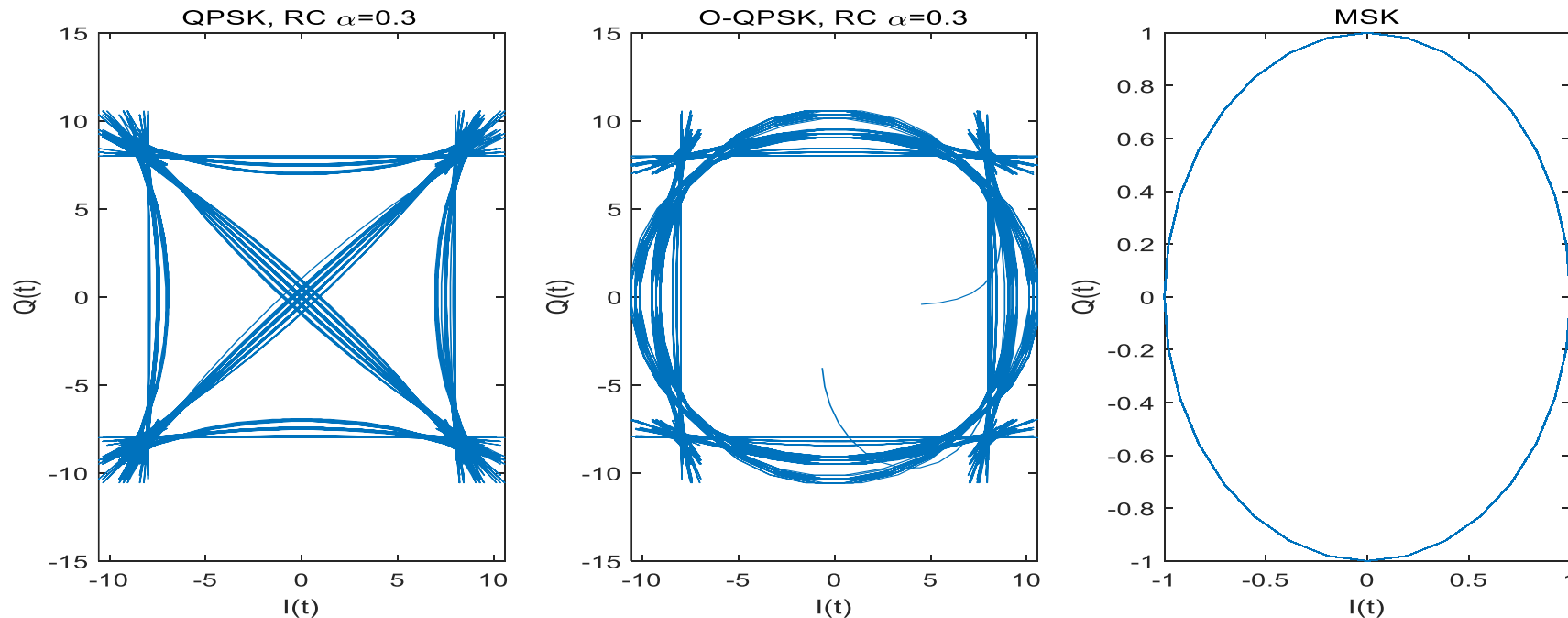
## □ constellations\_plots.m

```
05. %% 参数设置
06. N=1000;%发送的符号数目, 数量足够即可
07. fc=10;%载频
08. L=8;%过采样因子
09.
10. a = randi([0,1],N,1); %二进制数据流
11.
12. %% 使用 QPSK,OQPSK 和 MSK 调制信号
13. [s_qpsk,t_qpsk,I_qpsk,Q_qpsk]=qpsk_mod(a,fc,L);
14. [s_oqpsk,t_oqpsk,I_oqpsk,Q_oqpsk]=oqpsk_mod(a,fc,L);
15. [s_msk,t_msk,I_msk,Q_msk] = msk_mod(a,fc,L);
16.
17. %构造升余弦滤波器
18. alpha = 0.3;% 升余弦滤波器alpha参数
19. Nsym = 10;% 滤波器的符号跨度
20. rcPulse = raisedCosineFunction(alpha,L,Nsym);%升余弦滤波器函数
21.
22. %不同调制信号的I路和Q路信号通过成型滤波器
23. iRC_qpsk = conv(I_qpsk,rcPulse,'same'); %成型滤波后的I路QPSK信号
24. qRC_qpsk = conv(Q_qpsk,rcPulse,'same'); %成型滤波后的Q路QPSK信号
25. iRC_oqpsk = conv(I_oqpsk,rcPulse,'same');%成型滤波后的I路OQPSK信号
26. qRC_oqpsk = conv(Q_oqpsk,rcPulse,'same');%成型滤波后的Q路OQPSK信号
27.
28. %% 画星座
29. subplot(1,3,1);plot(iRC_qpsk,qRC_qpsk); %QPSK星座
30. title('QPSK, RC \alpha=0.3');xlabel('I(t)');ylabel('Q(t)');
31. subplot(1,3,2);plot(iRC_oqpsk,qRC_oqpsk); %OQPSK星座
32. title('O-QPSK, RC \alpha=0.3');xlabel('I(t)');ylabel('Q(t)');
33. subplot(1,3,3);plot(I_msk(20:end-20),Q_msk(20:end-20)); %MSK信号星座
34. title('MSK');xlabel('I(t)');ylabel('Q(t)');
```



# 星座图的发现

- 常规QPSK具有 $180^\circ$ 相变，因此需要具有高Q因子的线性放大器
- Offset-QPSK的相变限制为 $90^\circ$ （消除了 $180^\circ$ 的相变）
- MSK是连续的相位调制，因此，当符号发生变化时（如星座图中的平滑圆周所指示），不会发生突然的相变。因此，频带受限的MSK信号将不会遭受任何包络变化

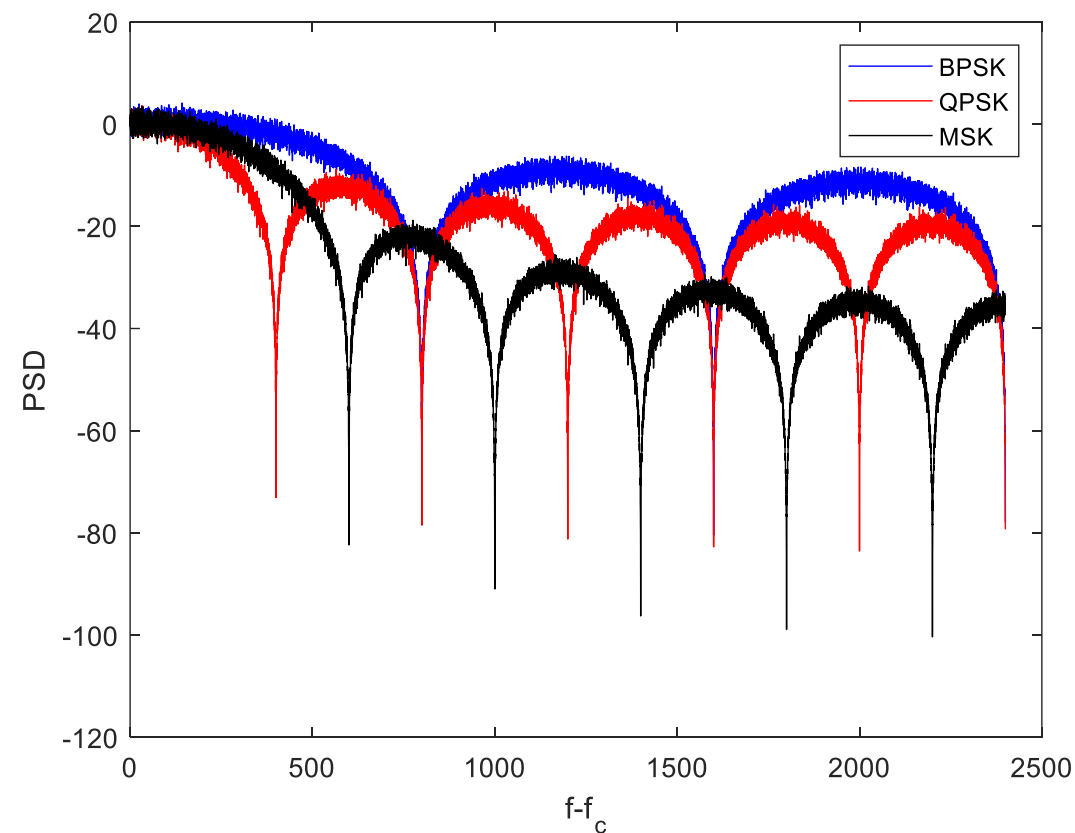


# 功率谱密度

- 功率谱密度 (PSD) 是信号在频域中的功率强度的量度
- 比较各种调制方案的PSD在理解其频谱特性以及决定选择满足给定通信系统要求的调制方案方面起着至关重要的作用

✓ bpsk\_qpsk\_msk\_psd.m

```
01. %比较几种带通信号的功率谱密度
02. clear all;clc;close all
03. rng default
04.
05. N=100000;%发送的符号数目
06. Fc=10;%载频
07. OF=8;%过采样因子
08. Fs = Fc*OF;%采样频率
09.
10. a = randi([0,1],N,1); %二进制数据流
11.
12. [s_bb,t]= bpsk_mod(a,OF); %BPSK基带
13. s_bpsk = s_bb.*cos(2*pi*Fc*t/Fs);%载波调制后的BPSK
14. s_qpsk = qpsk_mod(a,Fc,OF); %传统QPSK
15. s_msk = msk_mod(a,Fc,OF); %MSK信号
16.
17. %计算并绘制PSD
18. figure
19. plotWelchPSD(s_bpsk,Fs,Fc,'b'); hold on;
20. plotWelchPSD(s_qpsk,Fs,Fc,'r');
21. plotWelchPSD(s_msk,Fs,Fc,'k');
22. legend('BPSK','QPSK','MSK');xlabel('f-f_c'),ylabel('PSD');
```



# 频移键控 (FSK)

- 在二进制FSK中，二进制数据使用频率为 $f_1$ 和 $f_2$ 的两个波形传输，它们从载波频率 $f_c$ 的中心稍微偏移了 $\Delta f$

$$f_1 = f_c + \frac{\Delta f}{2}$$

$$f_2 = f_c - \frac{\Delta f}{2}$$

- ✓ 调制指数：由频率间隔 $\Delta f$ 和比特率 $T_b$ 相乘

$$h = \Delta f T_b$$

$$S_{BFSK}(t) = \begin{cases} A \cos[2\pi f_1 t + \phi_1] = A \cos\left[2\pi \left(f_c + \frac{\Delta f}{2}\right) t + \phi_1\right], & 0 \leq t \leq T_b, \text{ for binary 1} \\ A \cos[2\pi f_2 t + \phi_2] = A \cos\left[2\pi \left(f_c - \frac{\Delta f}{2}\right) t + \phi_2\right], & 0 \leq t \leq T_b, \text{ for binary 0} \end{cases}$$

$$S_{BFSK}(t) = \begin{cases} A \cos\left[\left(2\pi f_c + \frac{\pi h}{T_b}\right) t + \phi_1\right], & 0 \leq t \leq T_b, \text{ for binary 1} \\ A \cos\left[\left(2\pi f_c - \frac{\pi h}{T_b}\right) t + \phi_2\right], & 0 \leq t \leq T_b, \text{ for binary 0} \end{cases}$$

# BFSK检测的正交条件

- 正交性的条件可以通过找到两个波形之间相对于两个频率之间的频率偏移  $\Delta f$  或相对于调制指数  $h$  的相关性来获得

$$R(f_1, f_2) = R(\Delta f) = \int_0^{T_b} \cos(2\pi f_1 t + \phi_1) \cos(2\pi f_2 t + \phi_2) dt = 0$$
$$\cos(\phi_1 + \phi_2) \frac{\sin(2\pi[f_1 + f_2]T_b)}{2\pi[f_1 + f_2]} + \sin(\phi_1 + \phi_2) \left[ \frac{\cos(2\pi[f_1 + f_2]T_b) - 1}{2\pi[f_1 + f_2]} \right]$$
$$+ \cos(\phi_1 - \phi_2) \frac{\sin(2\pi[f_1 - f_2]T_b)}{2\pi[f_1 - f_2]} + \sin(\phi_1 - \phi_2) \left[ \frac{\cos(2\pi[f_1 - f_2]T_b) - 1}{2\pi[f_1 - f_2]} \right] = 0$$

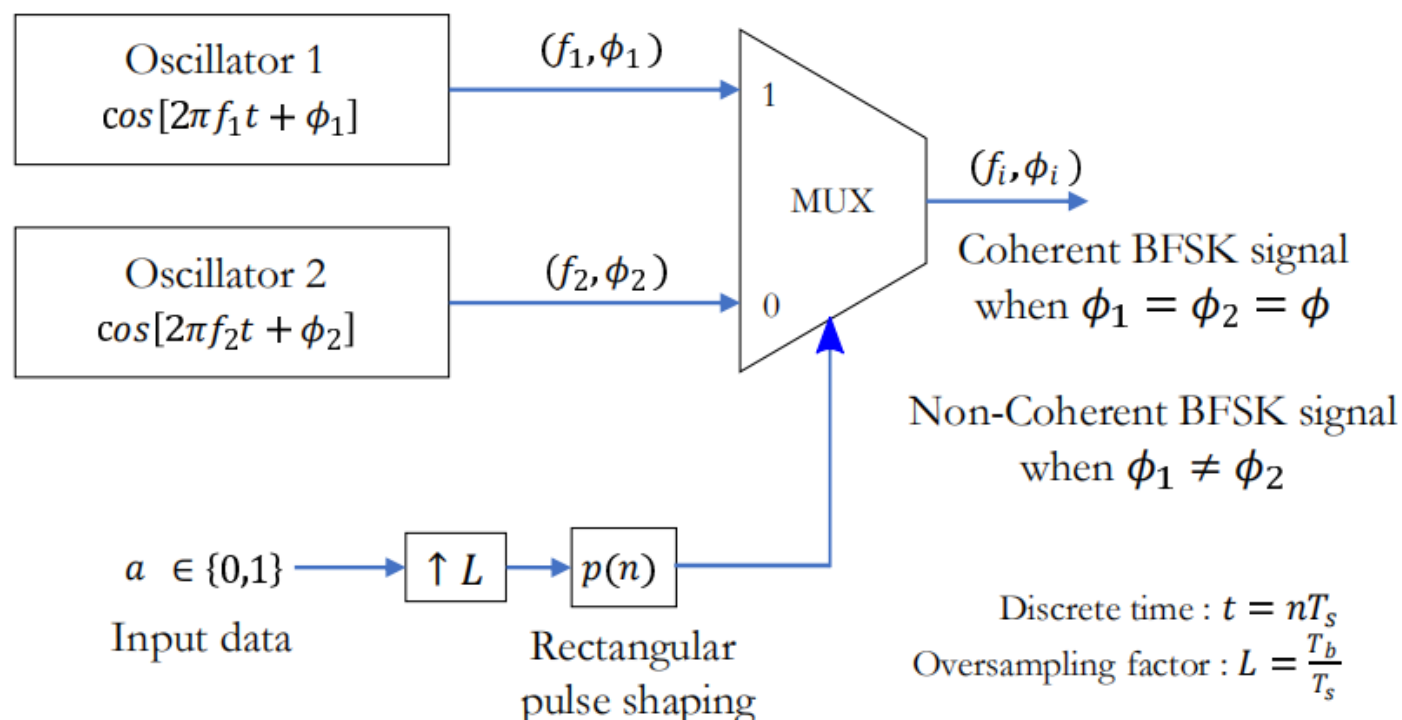
- 非相干BFSK  $\phi_1 \neq \phi_2$

$$\Delta f = f_1 - f_2 = \frac{n}{T_b}, \quad n = 1, 2, 3, \dots \quad (\Delta f)_{\min} = \frac{1}{T_b} \Rightarrow h_{\min} = 1 \quad 0.5 e^{-E_b/(2N_0)}$$

- 相干BFSK  $\phi_1 = \phi_2$

$$\Delta f = f_1 - f_2 = \frac{n}{2T_b}, \quad n = 1, 2, 3, \dots \quad (\Delta f)_{\min} = \frac{1}{2T_b} \Rightarrow h_{\min} = 0.5 \quad Q\left(\sqrt{E_b/N_0}\right)$$

# BFSK调制与实现

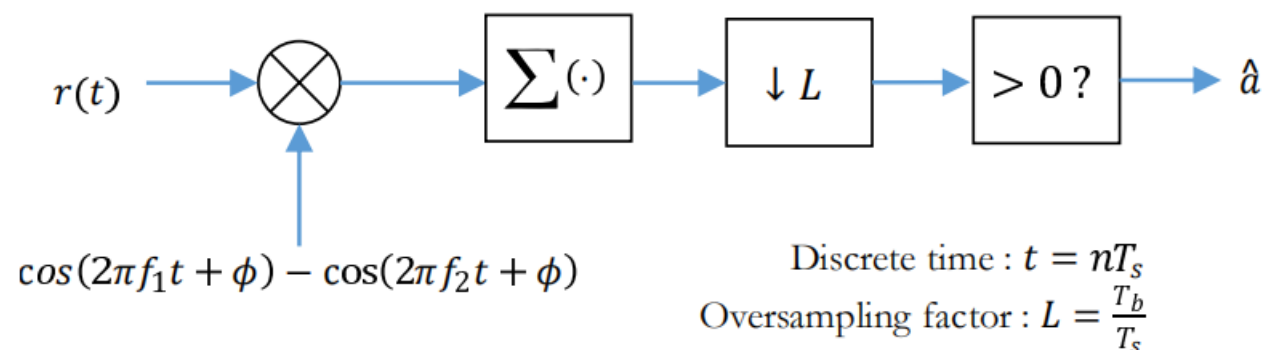


```

01. function [s,t,phase,at] = bfsk_mod(a,Fc,Fd,L,Fs,fsk_type)
02. % 使用BFSK调制二进制数据流
03. % 输入参数
04. % a: 输入二进制数据流
05. % Fc: 载频
06. % Fd: 距离Fc的频率偏移
07. % L: 比特周期的采样数目
08. % Fs: 采样频率
09. % fsk_type: 'COHERENT' (默认) or 'NONCOHERENT' FSK
10. % 输出参数
11. %s - BFSK调制信号
12. %t - 调制信号的时间点矢量
13. %phase - 调制器初始相位, 适用于相干检测.
14. % at 输入数据的数据波形
15.
16. phase=0;
17. at = kron(a,ones(1,L)); %数据转成波形
18. t = (0:1:length(at)-1)/Fs; %时间点矢量
19. if strcmpi(fsk_type,'NONCOHERENT')
20.     c1 = cos(2*pi*(Fc+Fd/2)*t+2*pi*rand);%带随机相位的载波1
21.     c2 = cos(2*pi*(Fc-Fd/2)*t+2*pi*rand);%带随机相位的载波2
22.     %注意: 非相干, 无相位信息返回
23. else
24.     phase=2*pi*rand;%随机相位 [0,2pi)
25.     c1 = cos(2*pi*(Fc+Fd/2)*t+phase);%带随机相位的载波1
26.     c2 = cos(2*pi*(Fc-Fd/2)*t+phase);%带随机相位的载波2
27. end
28. s = at.*c1 +(-at+1).*c2; %BFSK信号
    
```



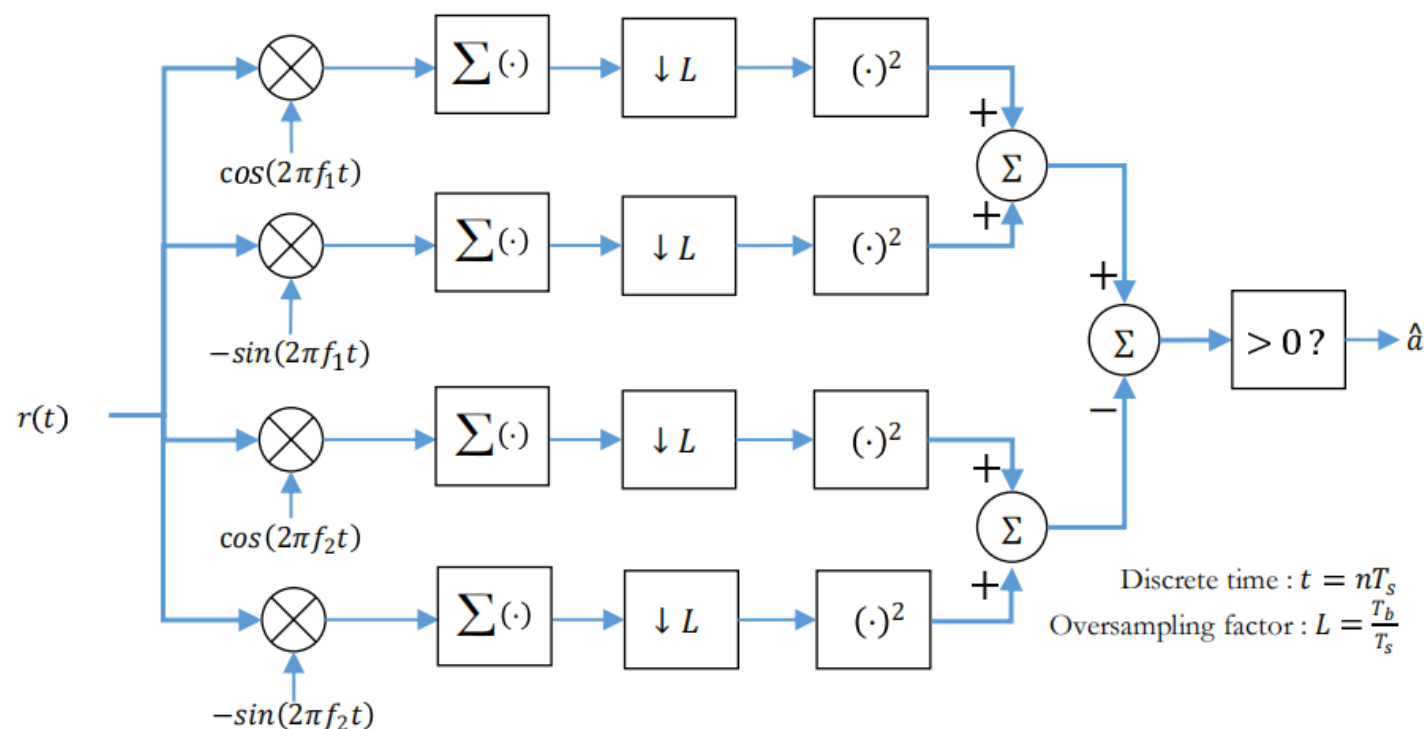
# BFSK相干解调与实现



```
01. function a_cap = bfsk_coherent_demod(r,Fc,Fd,L,Fs,phase)
02. % BFSK的相干解调
03. % 输入参数
04. % r: 接收信号
05. % Fc: 载频
06. % Fd: 距离Fc的频率偏移
07. % L: 比特周期的采样数目
08. % Fs: 采样频率
09. % phase: 调制器初始相位
10. % 输出参数
11. % a_cap - 检测得到的数据流
12.
13. t = (0:1:length(r)-1)/Fs; %时间矢量
14. x = r.*(cos(2*pi*(Fc+Fd/2)*t+phase)-cos(2*pi*(Fc-Fd/2)*t+phase));
15. y = conv(x,ones(1,L)); %积分
16. a_cap = y(L:L:end)>0;%采样并检测
17. end
```



# BFSK非相干解调与实现



```

01. function a_cap = bfsk_noncoherent_demod(r,Fc,Fd,L,Fs)
02. % BFSK的非相干解调
03. % 输入参数
04. % r: 接收信号
05. % Fc: 载频
06. % Fd: 距离Fc的频率偏移
07. % L: 比特周期的采样数目
08. % Fs: 采样频率
09. % 输出参数
10. % a_cap - 检测得到的数据流
11.
12. t = (0:1:length(r)-1)/Fs;
13. F1 = (Fc+Fd/2);
14. F2 = (Fc-Fd/2);
15.
16. %4个解调载波信号
17. p1c = cos(2*pi*F1*t);
18. p1s = -sin(2*pi*F1*t);
19. p2c = cos(2*pi*F2*t);
20. p2s = -sin(2*pi*F2*t);
21.
22. %积分
23. r1c = conv(r.*p1c,ones(1,L));
24. r1s = conv(r.*p1s,ones(1,L));
25. r2c = conv(r.*p2c,ones(1,L));
26. r2s = conv(r.*p2s,ones(1,L));
27.
28. %下采样
29. r1c = r1c(L:L:end);
30. r1s = r1s(L:L:end);
31. r2c = r2c(L:L:end);
32. r2s = r2s(L:L:end);
33.
34. %平方相加
35. x = r1c.^2 + r1s.^2;
36. y = r2c.^2 + r2s.^2;
37.
38. a_cap=(x-y)>0; %判决
    
```

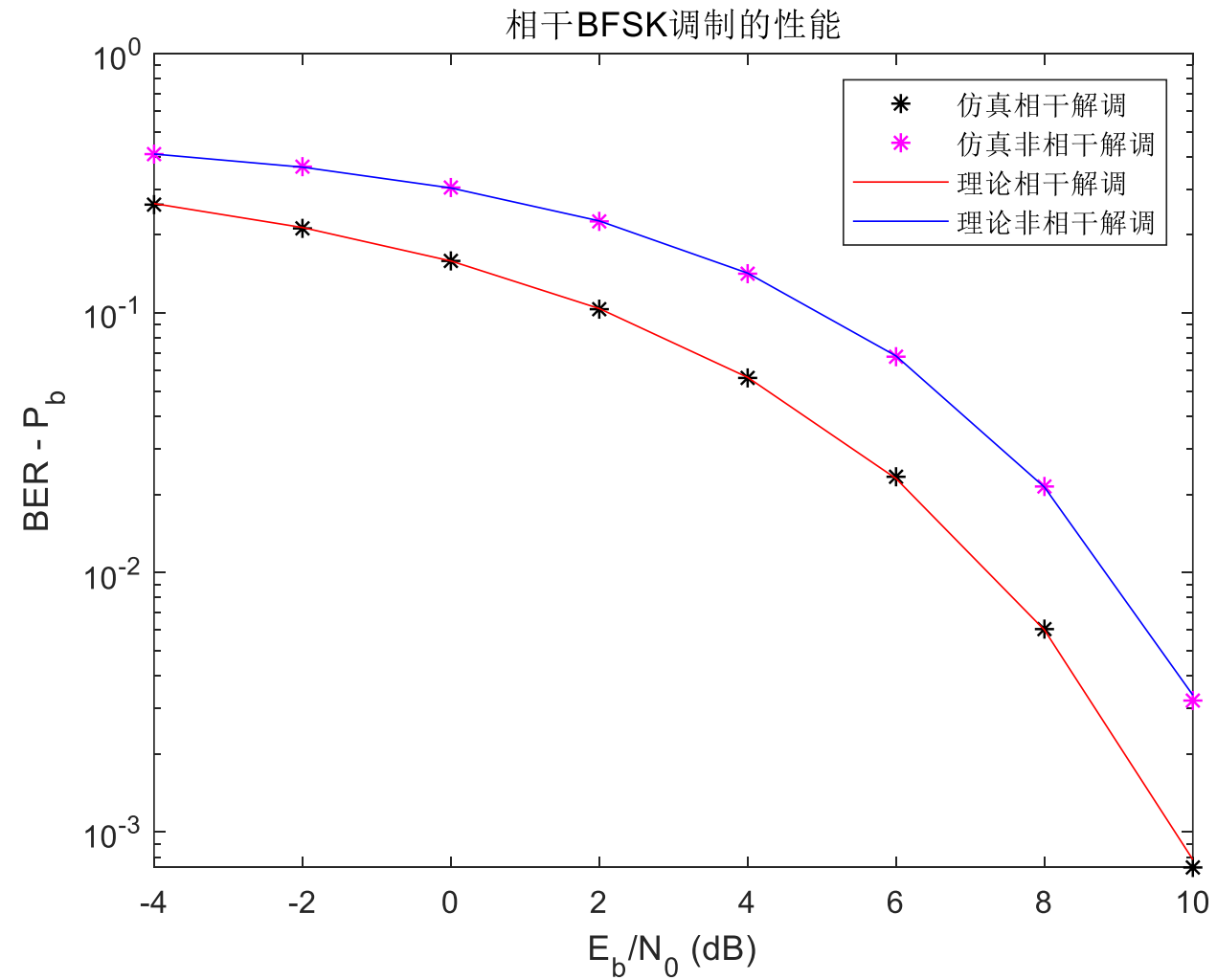
# BFSK的性能仿真

## □ bfsk\_wfm\_sim.m

```
05. %% 参数设置
06. N=100000;          %发送比特数目,
07. EbN0dB = -4:2:10; %仿真的Eb/N0的dB范围
08. Fc = 400;          %载频
09. fsk_type = 'COHERENT'; %COHERENT/NONCOHERENT FSK generation at Tx
10. h = 1;             %调制指数, 对相干FSK, h最小为0.5或0.5的倍数
11.                 %      对非相干FSK, h最小为1或为1的倍数
12. L = 40;             %过采样因子
13. Fs = 8*Fc;          %采样频率
14. Tb = L/Fs;          %比特周期
15. Fd = h/Tb;          %频率偏移
16.
17. EbN0lin = 10.^(EbN0dB/10);
18. BER_coherent = zeros(length(EbN0dB),1);
19. BER_nonCoherent = BER_coherent;
```

```
21. %% 仿真
22. a = randi([0,1],1,N); %二进制数据流
23. [s,t,phase]=bfsk_mod(a,Fc,Fd,L,Fs,fsk_type); %BFSK调制
24.
25. for i=1:length(EbN0dB)
26.     Eb=L*sum(abs(s).^2)/(length(s)); %计算每比特信号能量
27.     N0= Eb/EbN0lin(i); %噪声功率谱密度
28.     n = sqrt(N0/2)*(randn(1,length(s))); %生成噪声
29.     r = s + n;
30.
31.     if strcmpi(fsk_type,'COHERENT')
32.         a_cap1 = bfsk_coherent_demod(r,Fc,Fd,L,Fs,phase); %相干检测
33.         a_cap2 = bfsk_noncoherent_demod(r,Fc,Fd,L,Fs); %非相干检测
34.         BER_coherent(i) = sum(a~=a_cap1)/N; %相干BER统计
35.         BER_nonCoherent(i) = sum(a~=a_cap2)/N; %非相干BER统计
36.     end
37.     if strcmpi(fsk_type,'NONCOHERENT')
38.         a_cap2 = bfsk_noncoherent_demod(r,Fc,Fd,L,Fs); %非相干检测
39.         BER_nonCoherent(i) = sum(a~=a_cap2)/N; %非相干BER统计
40.     end
41. end
42. coherent = 0.5*erfc(sqrt(EbN0lin/2)); %相干检测理论BER
43. nonCoherent = 0.5*exp(-EbN0lin/2); %非相干检测理论BER
44.
```

# 仿真结果



# 课后作业

---

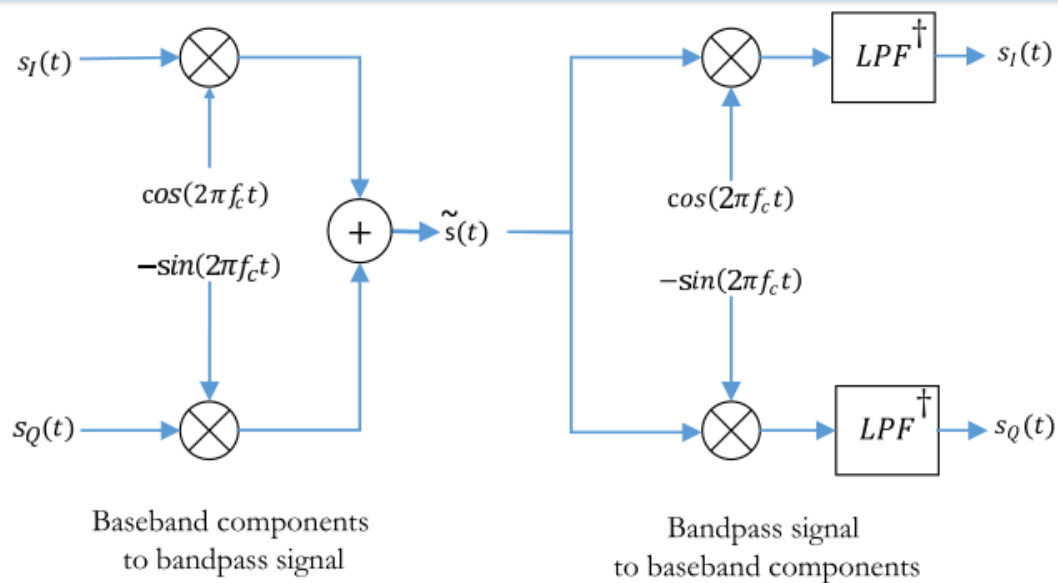
- 修改QPSK调制与解调程序，实现8PSK的调制与解调
- 绘制相干和非相干FSK信号的PSD（载频 $F_c=400$ ，调制指数 $h=1$ ）

# 课后作业

□ 在Demo3的基础上，增加发送载波搬移和接收载波搬移，两个功能可以函数实现

- ✓ `function SigBandPass = TxCarrierShift(TxSig,Fs,Fc)`
  - %功能：发送载波调制，也即基带搬移载频
  - %输入：TxSig 待发送的基带信号
  - % Fs 采样率 Fc 载波频率
  - %输出：SigBandPass 载波调制信号（实信号）
- ✓ `function SigBaseband = RxCarrierShift(RxSig,Fs,Fc)`
  - %功能：接收载波解调，也即载频搬移到基带
  - %输入：RxSig 接收到的载波调制信号
  - % Fs 采样率 Fc 载波频率
  - %输出：SigDeFc 解调后的等效基带复信号

# 载波搬移



在发送端，载波搬移需要将 I、Q 两路数据搬移至载波；在接收端，载波搬移需要将载波上的信号恢复成 I、Q 两路数据。设 I、Q 两路数据分别为  $x_i(t)$  和  $x_q(t)$ ，载波为  $f_c$ ，则发送端载波搬移公式如式2.1所示：

$$x(t) = x_i(t) \cos 2\pi f_c t - x_q(t) \sin 2\pi f_c t \quad (2.1)$$

接收端载波搬移公式为

$$x_i(t) = \text{Filter}(x(t) \cos 2\pi f_c t) \quad (2.2)$$

$$x_q(t) = \text{Filter}(-x(t) \sin 2\pi f_c t) \quad (2.3)$$

其中 Filter 表示低通滤波器，由于成型滤波器可以看作低通滤波器，因此在此部分可以省略该步骤。

# 有问题，随便问！

