

# SISTEMA DE COMUNICAÇÃO TX-→CANAL-→RX

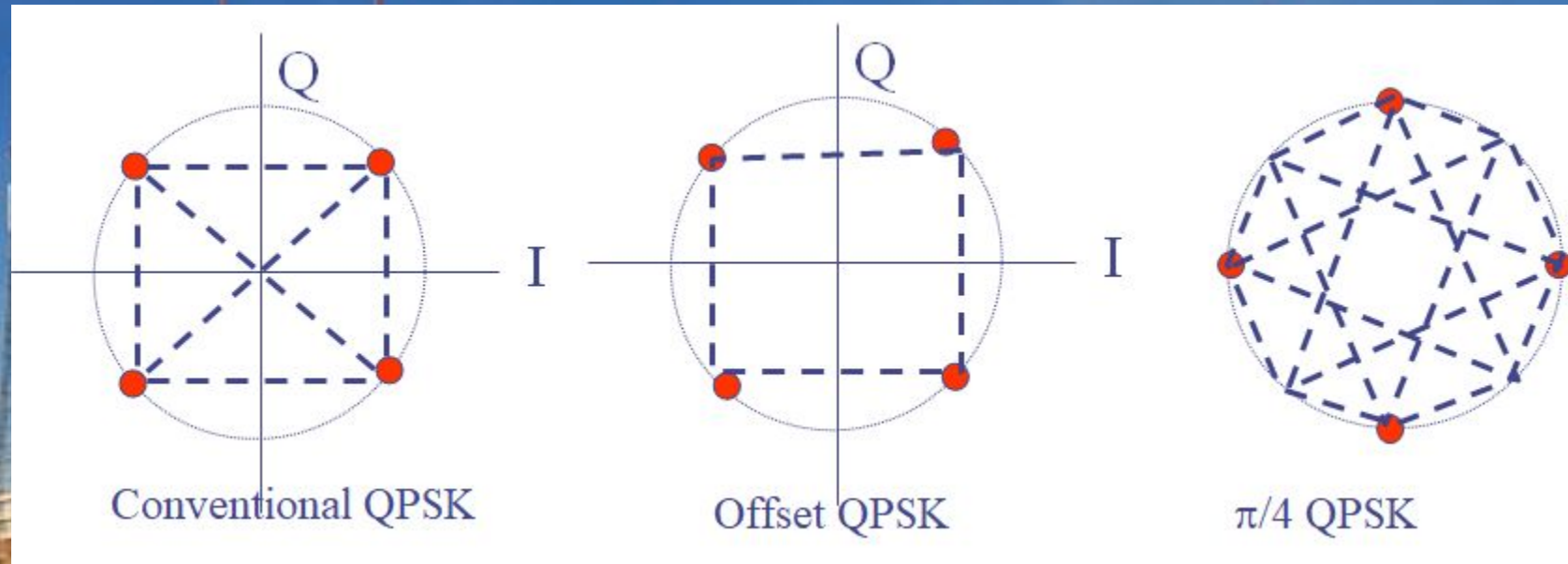
Cléber Werlang, Lucas P. Siqueira

## Quadrature Phase Shift Keying (QPSK)

O QPSK faz uso de uma propriedade de ortogonalidade dos sinais que faz com que se forem transmitidos simultaneamente em um canal, dois sinais PSK defasados de  $90^\circ$  (seno e cosseno) é possível detectar cada um independentemente um do outro. Trata-se portanto de um esquema de modulação com 4 estados nas fases  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  e  $270^\circ$ .

O que significa que **na mesma banda de um sinal PSK** é possível transmitir uma **taxa de dados duas vezes maior** utilizando modulação QPSK. A modulação QPSK deu origem a uma família de esquemas que inclui o DQPSK,  $\pi/4$  QPSK e OQPSK entre outros.

# Variações do QPSK



# QPSK MODEM

- MODULAÇÃO: QPSK
- BANDA: PASSANTE
- PULSO: NORMALIZADO
- ADIÇÃO DO RUÍDO BRANCO GAUSSIANO NO CANAL



## Quadrature Phase Shift Keying (QPSK)

Na modulação QPSK, a informação carregada pelo sinal transmitido está contida na fase. O sinal QPSK transmitido é descrito por :

$$S_i(t) = \sqrt{\frac{2E}{T}} \cos\left[(2i-1)\frac{\pi}{4}\right] \cos 2\pi f_c t - \sqrt{\frac{2E}{T}} \sin\left[(2i-1)\frac{\pi}{4}\right] \sin 2\pi f_c t$$

Onde T é o período de símbolo, E é a energia de símbolo,  $i = 1, 2, 3, 4$  e  $f_c$  é a frequência da portadora. Isso produz as quatro fases  $\pi/4, 3\pi/4, 5\pi/4$  e  $7\pi/4$  conforme necessário.

## Quadrature Phase Shift Keying (QPSK)

O símbolo QPSK é formado por 2 bits codificados em código Gray, de forma que somente um bit é modificado na transição para um símbolo adjacente. Tem-se:

$$T = n \tau$$

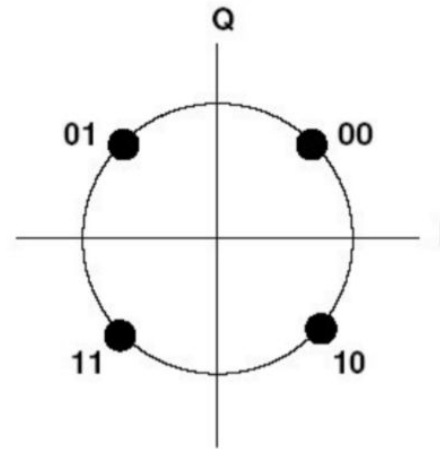
Onde  $n = 2$  bits por símbolo,  $\tau$  é o período do bit e  $E_b$  é a energia por bit. Neste caso, mais bits por símbolo, ou mais bits por Hz, são transmitidos. Este é um dos recursos utilizados para melhorar a eficiência espectral da banda de frequência disponível.

$$E = n E_b$$

# Quadrature Phase Shift Keying (QPSK)

O sinal QPSK transmitido pode ser representado por uma constelação que mostra os símbolos enviados.

$$\left( \pm \sqrt{\frac{E_s}{2}}, \pm \sqrt{\frac{E_s}{2}} \right).$$



## QPSK - BER (Taxa de erro de bit)

Embora o QPSK possa ser visto como uma modulação quaternária, é mais fácil vê-lo como duas portadoras de quadratura modulados independentemente. O BPSK é usado em ambos os portadores e eles podem ser demodulados independentemente. A probabilidade de erro de bit para o QPSK é a mesma que para BPSK:

$$P_b = Q \left( \sqrt{\frac{2E_b}{N_0}} \right).$$



## QPSK - Taxa de erro de símbolo

A taxa de erro de símbolo é dada por:

$$\begin{aligned} P_s &= 1 - (1 - P_b)^2 \\ &= 2Q\left(\sqrt{\frac{E_s}{N_0}}\right) - \left[Q\left(\sqrt{\frac{E_s}{N_0}}\right)\right]^2. \end{aligned}$$

QPSK usa o dobro da energia (uma vez que dois bits são transmitidos simultaneamente)



## QPSK - Taxa de erro de símbolo

Se a relação sinal-ruído for alta (como é necessário para sistemas QPSK práticos) então a fórmula pode ser aproximada:

$$P_s \approx 2Q \left( \sqrt{\frac{E_s}{N_0}} \right).$$

# Configuração inicial

```
clc;
clear;
close all;

% Definicao dos parametros
Fs = 44100; % Frequencia das amostras
Fc = 4000; % Frequencia trasmisor
Ts = 2.3*(10^-3); % Tempo de simbolo
Nb = 90000; % Numero de bits de entrada
beta = 3/100; % Fator de atenuacao
Ns = round(Fs*Ts); % Numero de amostras
Tsam = 1/Fs; % Tempo de amostra
T = 0:Tsam:(Ns-1)*Tsam; % Tempo total
mho = 23*10^-5; % Tempo de delay
N = 0; % Ruido inicial
```

# Configuração da portadora

```
% Geração de pulso, bits e símbolos
p = sin(pi/Ts*T); % pulso
norm = 1/sqrt(1/Fs*sum(p.^2)); % Normalização do pulso
P = norm*p; % pulso normalizado
B = randi([0 1],1,Nb); % Bits aleatorios
S = zeros(1,length(B)/2); % Símbolos

% Mapeamento de bits dentro dos simbolos da QSPK pelo metodo de gray
for n = 1:length(B)/2
    if (B(2*n-1) == 0) && (B(2*n) == 0)
        S(n) = 1+1i;
    elseif (B(2*n-1) == 1) && (B(2*n) == 0)
        S(n) = -1+1i;
    elseif (B(2*n-1) == 1) && (B(2*n) == 1)
        S(n) = -1-1i;
    elseif (B(2*n-1) == 0) && (B(2*n) == 1)
        S(n) = 1-1i;
    end
end

end

% Geração do pulso
PtI = zeros(Nb/2,length(P));
PtQ = zeros(Nb/2,length(P));
```



# Configuração da portadora

```
for n = 1:length(B)/2
    PtI(n,1:length(P)) = P*real(S(n)); % matriz com diferentes pulsos
    PtQ(n,1:length(P)) = P*imag(S(n));
end

PtI1 = reshape(vec2mat(PtI,1),[1,(length(S)*length(P))]);
PtQ1 = reshape(vec2mat(PtQ,1),[1,(length(S)*length(P))]);

Tc = 0:Tsam:Ts*length(S); % Tempo de sinal da transmissora

% Modulação da transmissora
CsI = sqrt(2).*cos(2*pi*Fc*Tc); % Parte real
CsQ = sqrt(2).*-sin(2*pi*Fc*Tc); % Parte Imaginaria
CsI = CsI(1:length(PtI1));
CsQ = CsQ(1:length(PtQ1));
SgI = CsI.*PtI1;
SgQ = CsQ.*PtQ1;

Tx = SgI+SgQ; % Sinal transmitido
```

# Configuração do canal

```
Mho = round(Fs*mho);  
for n = Mho+1:Ns*Nb/2  
    H(n) = (Tx(n)*(sqrt(1-beta^2))+Tx(n-Mho)*beta+N);  
end  
  
% SNR  
SNR_dB = 0:0.1:10;  
SNR_lin = 10.^(SNR_dB/10);  
Es = sum(abs(H.^2))./length(H); % Energia do sinal transmitido  
Eb = Es*length(P)/2; % Energia por bit  
  
for n = 1:length(SNR_lin)  
    No = Eb/SNR_lin(n);  
  
    % Adição do ruído branco gaussiano no canal  
    WGN = sqrt(No/2).*randn(1,length(H));  
  
    Tr = H+WGN; % Sinal transmitido
```

# Configuração do receptor

```
% Demodulação
RsI = CsI.*Tr; % Parte real
RsQ = CsQ.*Tr; % Parte imaginaria

% Match filtragem
MfI = conv(RsI,P);
MfQ = conv(RsQ,P);

% Amostragem
SamI = MfI(100:Ns:end);
SamQ = MfQ(100:Ns:end);

% Construção dos símbolos
SymI = sign(SamI(1:1:end));
SymQ = 1i*sign(SamQ(1:1:end));
Sym = SymI+SymQ;
```

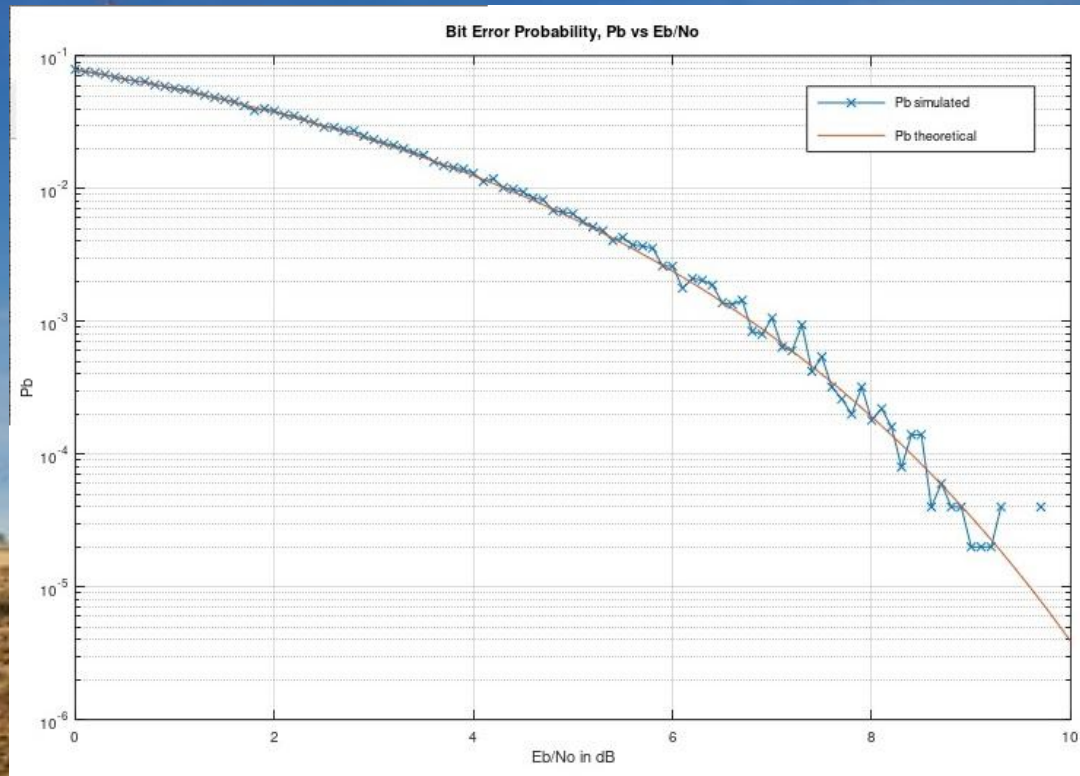
# Configuração do receptor

```
% Mapeamento de símbolos para os bits
R = zeros(1,Nb);
for m = 1:Nb/2
    if Sym(m) == 1+1i
        R(2*m-1) = 0;
        R(2*m) = 0;
    elseif Sym(m) == -1+1i
        R(2*m-1) = 1;
        R(2*m) = 0;
    elseif Sym(m) == -1-1i
        R(2*m-1) = 1;
        R(2*m) = 1;
    elseif Sym(m) == 1-1i
        R(2*m-1) = 0;
        R(2*m) = 1;
    end
end

Be = sum(R~=B); % Número de erros no bits recebidos
Pb_simulated(n) = Be/length(B); % Probabilidade de erro de bit
Pb_theoretical(n) = qfunc(sqrt(2*SNR_lin(n))); % Probabilidade de erro de bit
end
```



# Gráfico da BER



# Bibliografia

- [http://feihu.eng.ua.edu/NSF\\_TUES/Modulation.ppt](http://feihu.eng.ua.edu/NSF_TUES/Modulation.ppt) , Acessado em 30/11/19
- <https://www.rfwireless-world.com/Terminology/QPSK-vs-SQPSK.html> , Acessado em 30/11/19
- [https://www.mathworks.com/QPSK\\_modem.m](https://www.mathworks.com/QPSK_modem.m) , Acessado em 30/11/19