

1 Modulação, Demodulação e Desempenho Analítico e por Simulação Monte Carlo de Sistemas Comunicação Digitais

Utilizando o software matemático MatLab, obter:

1. Implementar e testar a função $Q(\cdot)$, dada por

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{u^2}{2}} du = \frac{1}{2} \operatorname{erfc} \left(\frac{x}{\sqrt{2}} \right)$$

- a. explorar as possibilidades de apresentação (saída gráfica) no MatLab
 - i. conversão $E_b/N_0 \rightarrow E_b/N_0_{dB}$
 - ii. uso dos comandos `legend`; `semilogy`
 - b. gerar saída gráfica comparando o $Q(x) \times \operatorname{erfc}(x)$.
2. Obter o desempenho teórico de um sistema de comunicação digital convencional com modulação BPSK em canal AWGN e Rayleigh Plano. Comparar com resultados da literatura (notas de aula e/ou ref. bibliográficas)
 3. Idem para modulação GMSK
 4. escrever uma função para o desempenho teórico (analítico) de um sistema convencional com modulação $M-PSK$ e detecção coerente. Comparar com a expressão

$$\overline{BER} = \frac{2}{k} Q \left(\sqrt{\frac{2kE_b}{N_0}} \sin \frac{\pi}{M} \right)$$

onde $k = \log_2 M$. Os resultados são os mesmos? Por que?

5. Utilizando o MatLab, resolva o exercício 1 da Lista de Exercícios 3
6. Comparação do desempenho analítico e Monte Carlo (MCS) para o **4-PSK**.
Quando não for possível obter expressões fechadas (analíticas ou analítica-iterativas) para o cálculo de desempenho de sistemas de comunicação ou mesmo quando se estiver testando a consistência de uma nova expressão analítica (exata ou aproximada, no caso de investigação de nova expressão aproximada para o desempenho de um sistema específico), então a simulação Monte Carlo tem sido um meio muito utilizado na obtenção de desempenhos. O MCS aplicado a sistemas de comunicação digitais consiste:
 - a. geração, modulação e transmissão de uma grande quantidade de bit através de um canal AWGN ou com desvanecimento;
 - b. recepção demodulação, decisão e comparação da informação com aquela enviada.
 - c. Caso haja erro nesta comparação, incrementa-se o contador de erros.
 - d. Após envio de quantidade suficiente de bit (quantidade necessária para se obter pelo menos 10 erros), calcula-se a taxa de erro de bit (\overline{BER}) ou de símbolo (\overline{SER}) ou de quadro (\overline{FER}) médias. A figura 1 ilustra o caso para 4-PSK. As amostras das componentes de ruído AWGN, em fase e quadratura, n_I e n_Q devem ser estatisticamente independentes com distribuição Gaussiana de média zero e variância σ_n^2 . Em termos de simulação, pode-se ou normalizar a variância total de ruído e ajustar a relação sinal ruído (SNR), escalonando-se o parâmetro energia do sinal (símbolo), E_s , ou vice-versa. Note que em termos de energia, em um 4-PSK tem-se:

$$\frac{E_b}{N_0} = \frac{E_s/2}{N_0}$$

isto resulta em

$$\overline{BER} \approx \frac{1}{k} \overline{SER}$$

- e. O script e as funções necessárias à implementação do diagrama da figura 1 são descritas no final. Analise os algoritmos e verifique a consistência da obtenção dos desempenhos via MCS e analítico.
- f. Modifique o algoritmo de tal forma a obter o desempenho analítico e por simulação MCS de um 16-PSK coerente.

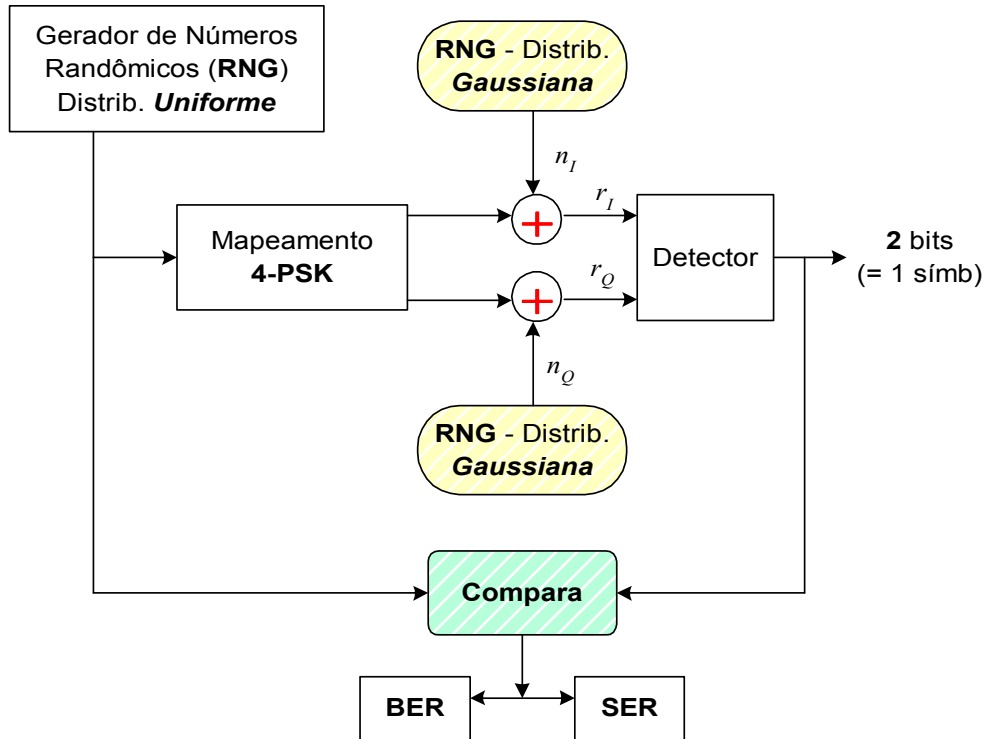


Fig.1. Diagrama em blocos de um sistema $M = 4PSK$ em canal AWGN para simulação Monte Carlo.

g. Através dos resultados MCS confirme a tendência de desempenho do PSK em função de M e a relação entre \overline{BER} e \overline{SER} .

7. Densidade Espectral de Potência (PSD) - MoDem.

A PSD é dada pela transformada de Fourier do sinal transmitido (ou recebido) no domínio do tempo. Para o MSK com demodulação coerente (desvio de freq. ajustado para $\Delta f = \frac{1}{4T_b}$, resulta:

$$s_{MSK}(t) = A \left[\cos\left(\pm\pi\frac{t}{2T_b}\right) \cos(2\pi f_c t) - \sin\left(\pm\pi\frac{t}{2T_b}\right) \sin(2\pi f_c t) \right]$$

a. Confirme na literatura que a PSD para o MSK (binário) é dada por

$$s_{MSK}(t) \xrightarrow{Fourier} S_{MSK}(f) = \frac{4P_c T_s [1 + \cos 2\pi (f - f_c) T_s]}{\pi^2 [1 - 4T_s^2 (f - f_c)^2]^2} = \frac{8P_c T_b [1 + \cos 4\pi (f - f_c) T_b]}{\pi^2 [1 - 16T_b^2 (f - f_c)^2]^2}$$

De forma análoga, pode-se obter a PSD para qualquer tipo de modulação. Para QPSK e O-QPSK resulta na mesma PSD:

$$S_{QPSK}(f) = 2P_c T \left[\frac{\sin 2\pi (f - f_c) T_b}{2\pi (f - f_c) T_b} \right]^2$$

b. Plote as PSD não filtradas teóricas para QPSK e MSK com freq normalizada de $(f - f_c) T_b$. Confirme que

i. o primeiro nulo espectral ocorre em

$$\begin{cases} 0,75 & \rightarrow MSK \\ 0,5 & \rightarrow QPSK \end{cases}$$

ii. energia dos lóbulos laterais é menor no MSK

8. A partir do $s_{MSK}(t)$ plote a PSD via $\mathfrak{F}\{s_{MSK}(t)\}$, comparando com a curva teórica dada anteriormente.

1.1 Scripts e Funções utilizadas

1.1.1 Script para desempenho teórico e Monte Carlo - 4PSK

```
SNRindB1=0:2:10; SNRindB2=0:0.1:10;
for i=1:length(SNRindB1),
[ps,pe]=cm_sm32(SNRindB1(i)); % simulated bit and symbol error rates
sml_d_bit_err_prb(i)=pe;
sml_d_symbol_err_prb(i)=ps;
end;
for i=1:length(SNRindB2),
SNR=exp(SNRindB2(i)*log(10)/10); % signal to noise ratio
theo_err_prb(i)=Qf(sqrt(2*SNR)); % theoretical bit error rate
end;
% Plotting commands follow
semilogy(SNRindB1,sml_d_bit_err_prb,'*');
hold
semilogy(SNRindB1,sml_d_symbol_err_prb,'o');
semilogy(SNRindB2,theo_err_prb);
```

1.1.2 Função cm_sm32.m

```
function [ps,pe]=cm_sm32(snr_in_dB)
% [ps,pe]=cm_sm32(snr_in_dB)
% CM_SM32 finds the probability of bit error and symbol error for the
% given value of snr_in_dB, signal to noise ratio in dB.
N=10000;
E=1; % energy per symbol
snr=10^(snr_in_dB/10); % signal to noise ratio
sigma=sqrt(E/snr)/2; % noise variance
% the signal mapping
s00=[1 0]; s01=[0 1]; s11=[-1 0]; s10=[0 -1];
% generation of the data source
for i=1:N,
temp=rand; % a uniform random variable between 0 and 1
if (temp<0.25), % with probability 1/4, source output is "00"
dsource1(i)=0;
dsource2(i)=0;
elseif (temp<0.5), % with probability 1/4, source output is "01"
dsource1(i)=0;
dsource2(i)=1;
elseif (temp<0.75), % with probability 1/4, source output is "10"
dsource1(i)=1;
dsource2(i)=0;
```

```

else % with probability 1/4, source output is "11"
dsource1(i)=1;
dsource2(i)=1;
end;
end;
% detection and the probability of error calculation
numofsymbolerror=0;
numofbiterror=0;
for i=1:N,
% the received signal at the detector, for the ith symbol, is:
n(1)=gngauss(sigma);
n(2)=gngauss(sigma);
if ((dsource1(i)==0) & (dsource2(i)==0)),
r=s00+n;
elseif ((dsource1(i)==0) & (dsource2(i)==1)),
r=s01+n;
elseif ((dsource1(i)==1) & (dsource2(i)==0)),
r=s10+n;
else
r=s11+n;
end;
% The correlation metrics are computed below
c00=dot(r,s00);
c01=dot(r,s01);
c10=dot(r,s10);
c11=dot(r,s11);
% The decision on the ith symbol is made next
c_max=max([c00 c01 c10 c11]);
if (c00==c_max),
decis1=0; decis2=0;
elseif (c01==c_max),
decis1=0; decis2=1;
elseif (c10==c_max),
decis1=1; decis2=0;
else
decis1=1; decis2=1;
end;
% increment the error counter, if the decision is not correct
symbolerror=0;
if (decis1~=dsource1(i)),
numofbiterror=numofbiterror+1;

```

```

symbolerror=1;
end;
if (decis2~=dsource2(i)),
numofbiterror=numofbiterror+1;
symbolerror=1;
end;
if (symbolerror==1),
numofsymbolerror = numofsymbolerror+1;
end;
end;
ps=numofsymbolerror/N; % since there are totally N symbols
pb=numofbiterror/(2*N); % since 2N bits are transmitted

```

1.1.3 Função qf.m: função implementada no item 1.

1.1.4 Função gngauss.m:

```

function [gsrv1,gsrv2]=gngauss(m,sgma)
% [gsrv1,gsrv2]=gngauss(m,sgma)
% [gsrv1,gsrv2]=gngauss(sgma)
% [gsrv1,gsrv2]=gngauss
% GNGAUSS generates two independent Gaussian random variables with mean
% m and standard deviation sgma. If one of the input arguments is missing
% it takes the mean as 0, and the standard deviation as the given parameter.
% If neither mean nor the variance is given, it generates two standard
% Gaussian random variables.
if nargin == 0,
m=0; sgma=1;
elseif nargin == 1,
sgma=m; m=0;
end;
u=rand; % a uniform random variable in (0,1)
z=sgma*(sqrt(2*log(1/(1-u))))); % a Rayleigh distributed random variable
u=rand; % another uniform random variable in (0,1)
gsrv1=m+z*cos(2*pi*u);
gsrv2=m+z*sin(2*pi*u);

```

Bibliografia

- [1] J. Proakis and M. Salehi, *Contemporary Communication Systems Using MATLAB*. PWS Publishing Company, 1997.
- [2] K. Feher, *Wireless Digital Communications - Modulation and Spread Spectrum Applications*. Prentice Hall PTR, 1995.
- [3] S. Sampei, *Applications of Digital Wireless Technologies to Global Wireless Communications*. Upper Saddle River, NJ: Prentice-Hall, Inc, 1997.