UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RELATÓRIO DE ESTÁGIO

DESEMPENHO DE ESQUEMAS DE MODULAÇÃO BPSK EM CANAIS COM RUÍDO GAUSSIANO

TONY KLEBER CARVALHO SANTOS

Campina Grande, agosto de 2002

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RELATÓRIO DE ESTÁGIO

DESEMPENHO DE ESQUEMAS DE MODULAÇÃO BPSK EM CANAIS COM RUÍDO GAUSSIANO

TONY KLEBER CARVALHO SANTOS Mat: 29711103

Orientador: Prof. Dr. Marcelo Sampaio de Alencar

Banca Examinadora: Prof. Dr. José Ewerton P. de Farias
Prof. Dr. Marcelo Sampaio de Alencar

Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

Agradecimento

Waslon Terllizzie Araújo Lopes

Pela disposição em ajudar,
possibilitando a realização desse projeto.

Sumário

1-Apresentação

O presente trabalho teve por objetivo o estudo do desempenho de alguns esquemas de modulação BPSK por meio da determinação da probabilidade de erro na recepção de sinais assim modulados. Especificamente, são vistos os casos de sinais binários antipodais e ortogonais, com mesma probabilidade de erro *a priori*, transmitidos por um canal distorcido pela adição de um ruído branco gaussiano (AWGN – *additive white Gaussian noise*).

Inicialmente são apresentados os princípios do receptor ótimo, lançando mão da representação de sinais por meio de vetores e determinando seu critério de decisão. Esse critério é então aplicado ao caso de um canal com ruído gaussiano aditivo, resultando numa função de decisão utilizada para determinar a probabilidade de erro dos casos estudados.

A análise é feita por meio de simulações numéricas utilizando o MATLAB®. São apresentadas as técnicas de geração do ruído gaussiano assim como a forma de determinação da probabilidade de erro. Os anexos trazem diferentes versões do programa utilizado nas simulações realizadas, incluindo uma versão em linguagem C.

2-Princípios do Receptor Ótimo

2.1-Situação Estudada

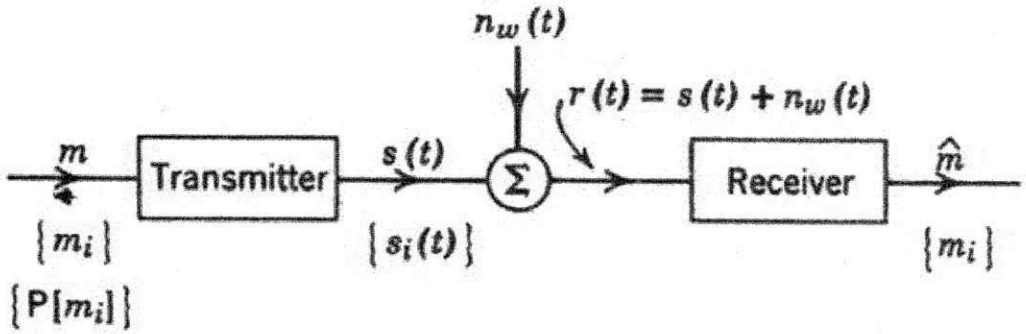

Fig 1: Comunicação por um canal com ruído branco gaussiano aditivo.

A Figura 1 ilustra a situação na qual se estudou o desempenho de alguns esquemas BPSK. Aqui, uma das formas de onda pertencente ao conjunto $\{s_i(t)\}$, i=0,1, ..., M-1, é transmitida por um canal perturbado pela adição de um ruído branco gaussiano, de modo que o sinal recebido é dado por

$$r(t) = s(t) + n_W(t).$$

Qual das formas de onda foi de fato transmitida depende da mensagem aleatória de entrada, m; quando $m=m_i$, o sinal transmitido é $s_i(t)$. Dessa forma a correspondência

$$m = m_i \Leftrightarrow s(t) = s_i(t),$$

define o transmissor. As probabilidades *a priori* $\{P[m_i]\}$ especificam a fonte.

2.2-Relação Sinais –Vetores

A fim de prosseguir na análise da situação apresentada na seção anterior, será vista uma forma de substituir a transmissão de formas de onda pela "transmissão" de vetores. Tal representação simplifica significativamente o estudo da transmissão de sinais.

2.2.1-Síntese das formas de onda

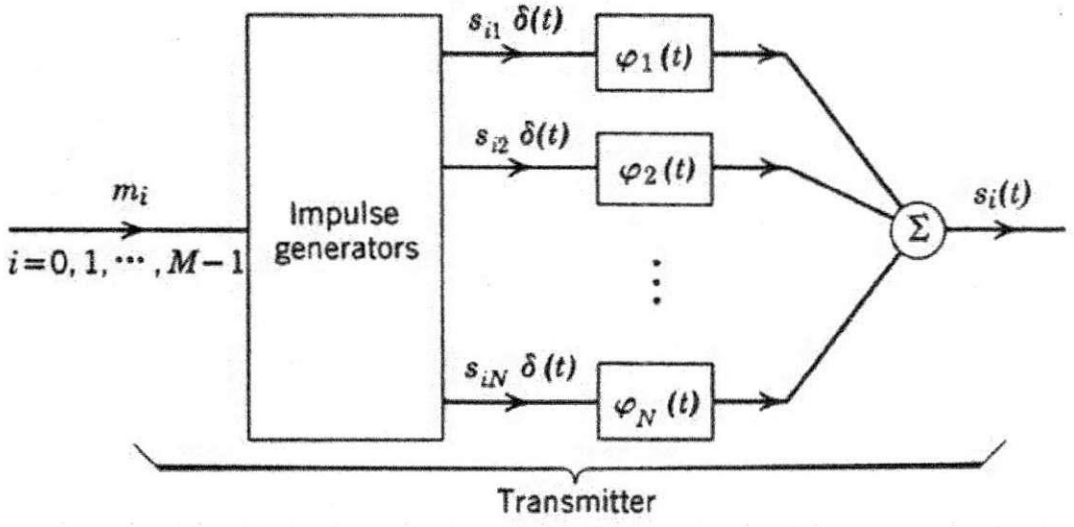

Fig 2: Síntese de sinais.

Uma forma conveniente de sintetizar um conjunto de sinais $\{s_i(t)\}$ no transmissor é mostrada na Figura 2. Um conjunto de N filtros é utilizado, com a resposta ao impulso do j-ésimo filtro sendo designado por $\varphi_j(t)$. Quando a entrada do transmissor é dada por m_i, o primeiro filtro é excitado por um impulso s_{i1}, o segundo por um impulso s_{i2} e assim por diante. As saídas dos filtros são somadas para formar $s_i(t)$. Assim a forma de onda transmitida é um dos M sinais

$$s_i(t) = \sum_{j=1}^{N} s_{ij}\varphi_j(t),$$

com i=0,1,..., M-1.

2.2.2-Interpretação geométrica dos sinais

A fim de facilitar a análise, é assumido que o conjunto de funções $\{\varphi_j(t)\}$ é ortonormal, assim:

$$\int_{-\infty}^{\infty} \varphi_j(t)\varphi_i(t)dt = \begin{cases} 1; & j = i \\ 0; & j \neq i \end{cases},$$

de modo que cada uma das formas de onda do transmissor é completamente especificada pelo vetor de seus coeficientes

$$s_i = (s_{i1}, s_{i2}, \ldots, s_{iN}),$$

i =0,1...,M-1.

A forma de visualizar-se os M vetores $\{s_i\}$ é, como de hábito, definindo M pontos em um espaço N-dimensional, com N eixos mutualmente perpendiculares designados por $\varphi_1, \varphi_2, \ldots, \varphi_N$. As formas de onda $\{s_i(t)\}$ dependem da escolha do conjunto $\{\varphi_j(t)\}$, no entanto sua representação geométrica depende apenas de $\{s_i\}$.

2.3-Receptor Ótimo

2.3.1-Função de decisão

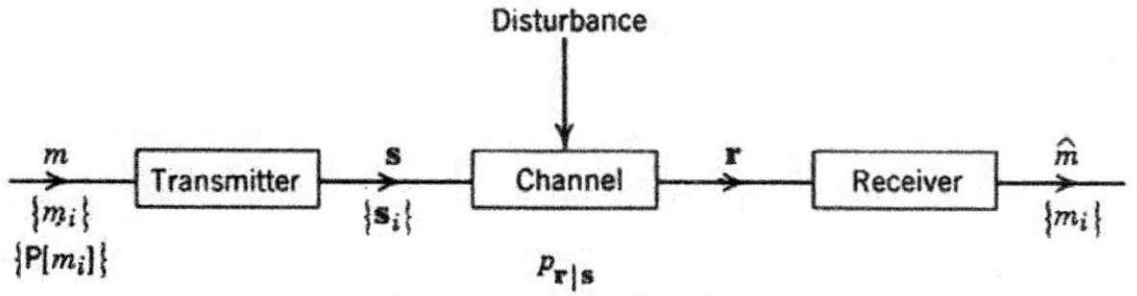

Fig 3: Sistema de comunicação vetorial.

Uma vez adotada a representação vetorial dos sinais, obtém-se o sistema de comunicação vetorial esquematizado na Figura 3. O transmissor é agora definido por um conjunto de M vetores $\{s_i\}$. Quando $m=m_i$, o vetor si é transmitido. O canal vetorial perturba a transmissão e emite um vetor aleatório

$$r_i = (r_{i1}, r_{i2}, \ldots, r_{iN}),$$

com certa probabilidade $p_{ri|si}$. O receptor ótimo seleciona

$$\hat{m} = m_k,$$

quando

$$P[m_k \mid r = \rho] > P[m_i \mid r = \rho],$$

para $i=0,1,...,M-1$, $i \neq k$. Ou seja, seleciona a mensagem mais provável dado que um certo vetor ρ foi recebido.

Pela regra de Bayes

$$P[m_i \mid r = \rho] = \frac{P[m_i].p_r(\rho \mid m_i)}{p_r(\rho)} = \frac{P[m_i].p_r(\rho \mid s = s_i)}{p_r(\rho)},$$

uma vez que o evento $m=m_i$ implica $s=s_i$.

Portanto, uma vez que $p_r(\rho)$ independe do índice i, conclui-se que o receptor ótimo seleciona a mensagem m_k quando a *função de decisão*

$$P[m_i].p_r(\rho \mid s = s_i),$$

com $i=0,1,...,M-1$ for máxima para $i=k$.

2.3.2-Regiões de decisão

A função de decisão juntamente com a representação geométrica de sinais apresentadas anteriormente, permitem introduzir o conceito de regiões de decisão.

Dado um conjunto de mensagens $\{m_k\}$, representadas pelos vetores $\{s_k\}$, num espaço de coordenadas $\{\varphi_k\}$, pode-se, para cada ponto ρ neste espaço, calcular a função de decisão. Obtém-se então um conjunto de M regiões disjuntas, chamadas *regiões de decisão*, englobando os pontos para os quais a função de decisão é máxima para cada uma das M mensagens m_k.

3-Ruído Gaussiano Aditivo

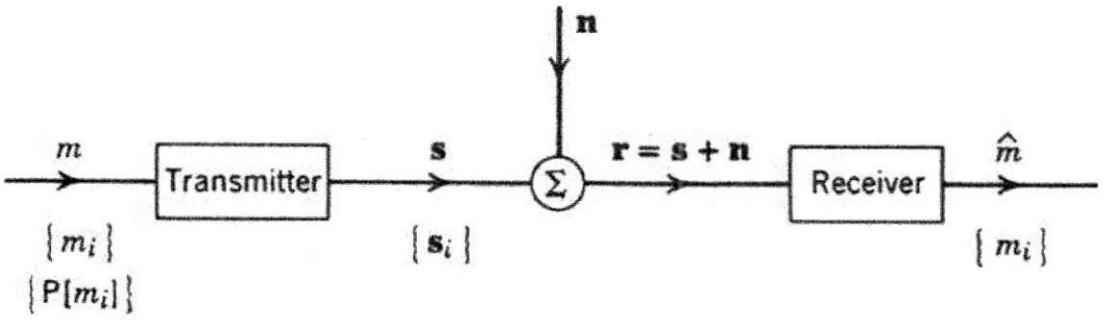

Fig 4: Sistema de comunicação vetorial com vetor ruído N-dimensional.

Os limites das regiões de decisão dependem das probabilidades *a priori* $\{P(m_i)\}$, dos sinais $\{s_i\}$ e da definição do canal $p_{r|s}$. Será estudado o caso representado na Figura 5 no qual o canal distorce o vetor sinal simplesmente somando um vetor-ruído aleatório

$$n_i = (n_1, n_2, \ldots, n_N),$$

de modo que o vetor recebido é dado por

$$r = s + n = (s_1 + n_1, s_2 + n_2, \ldots, s_N + n_N).$$

A equação anterior implica que $r=\rho$ quando $s=s_i$ se e somente se $n=\rho-s_i$, de modo que

$$p_r(\rho \mid s = s_i) = p_n(\rho - s_i \mid s = s_i).$$

Assumindo que n e s são estatisticamente independentes

$$p_{n|s} = p_n,$$

tem-se que

$$p_n(\rho - s_i \mid s = s_i) = p_n(\rho - s_i)$$

e a função de decisão toma a forma

$$P[m_i] p_n(\rho - s_i).$$

No caso em que as N componentes de n são variáveis aleatórias gaussianas independentes, de média zero e variância σ^2,a função de densidade de probabilidade do ruído, p_n, toma a forma da função de densidade conjunta

$$p_n(\alpha) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{N}\alpha_j^2\right) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} e^{-\frac{\|\alpha\|^2}{2\sigma^2}},$$

dada pelo produto das funções de densidade individuais, em que $\{\alpha_j\}$ são as coordenadas cartesianas de α.

Vê-se então que a função de decisão passa a ser dada por

$$P[m_i] e^{-\frac{|\rho - s_i|^2}{2\sigma^2}}$$

uma vez que o fator $(2\pi\sigma^2)^{-N/2}$ independe do índice i.

É fácil perceber que maximizar a expressão acima é equivalente a minimizar

$$|\rho - s_i|^2 - 2\sigma^2 \ln P[m_i].$$

Finalmente, para o caso de haver mesma probabilidade *a priori*, a função de decisão se resume a minimizar o quadrado da distância euclidiana entre os pontos ρ e s_i

$$|\rho - s_i|^2 = \sum_{j=1}^{N} (\rho_i - s_{ij})^2$$

Assim, para um canal vetorial perturbado pela adição de um vetor ruído cuja componentes são variáveis aleatórias gaussianas independentes, e para uma fonte de mensagens com mesma probabilidade a priori, o receptor ótimo define as regiões de decisão como o conjunto de pontos mais próximos de cada vetor sinal.

4-Desempenho de Esquemas BPSK

Uma vez estabelecido o critério de decisão ótimo para a situação estabelecida em 2.1, pode-se agora estudar o desempenho de dois esquemas de modulação BPSK, a saber, antipodal e ortogonal, por meio da avaliação da probabilidade de erro na recepção desses sinais.

O limite entre as regiões de decisão para o caso de sinais antipodais está representado na Figura 5. Ele é dado pelo conjunto dos pontos eqüidistantes de s_0 e s_1, ou seja, pelo eixo φ_2. Dessa forma, um erro ocorre quando s_1 é transmitido se e somente se a componente n_1 do ruído é maior que $d/2$, em que d é a distância entre os dois sinais.

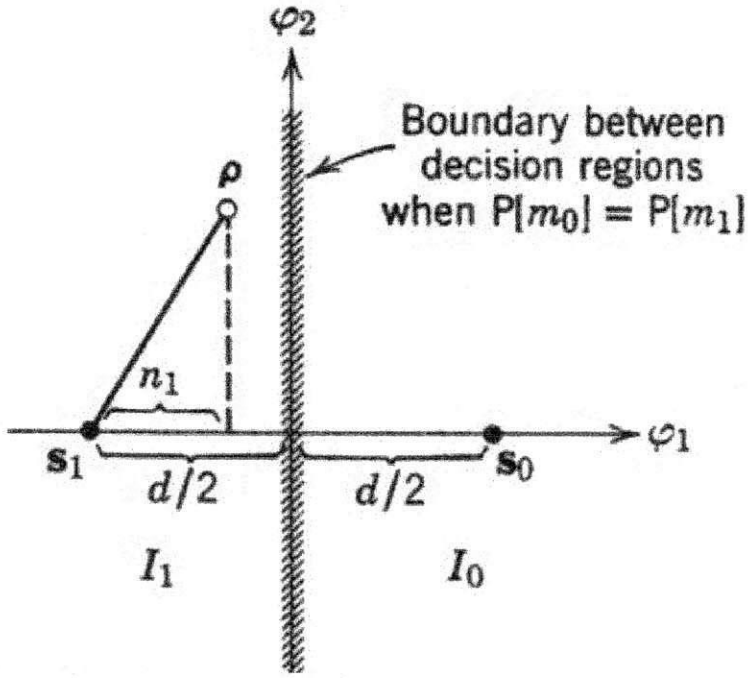

Fig 5: Sinais antipodais.

Mas n_1 é uma variável gaussiana com variância $No/2$, de modo que a probabilidade de erro, dado que a mensagem de entrada foi m_1, é

$$P[E \mid m_1] = \int_{d/2}^{\infty} \frac{1}{\sqrt{\pi No}} e^{-\frac{\alpha^2}{No}} d\alpha = P[E \mid m_2] = P[E],$$

uma vez que, por simetria, a probabilidade condicional de erro é a mesma para os dois sinais.

Definindo a função

$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-\frac{\gamma^2}{2}} d\gamma$$

e fazendo

$$\gamma = \alpha \sqrt{\frac{2}{No}},$$

tem-se

$$P[E] = Q\left(\frac{d}{\sqrt{2No}}\right)$$

O comprimento de cada vetor é dado por $\sqrt{E_s}$, em que E_s é a energia de cada sinal, de modo que para o caso antipodal d=2$\sqrt{E_s}$. Assim

$$P[E] = Q\left(\sqrt{\frac{2Es}{No}}\right).$$

Do ponto de vista da probabilidade de erro, o caso ortogonal representado na Figura 6 é equivalente ao antipodal, sendo que agora a distância entre os vetores sinais dada por d=$\sqrt{2E_s}$, de modo que

$$P[E] = Q\left(\sqrt{\frac{Es}{No}}\right).$$

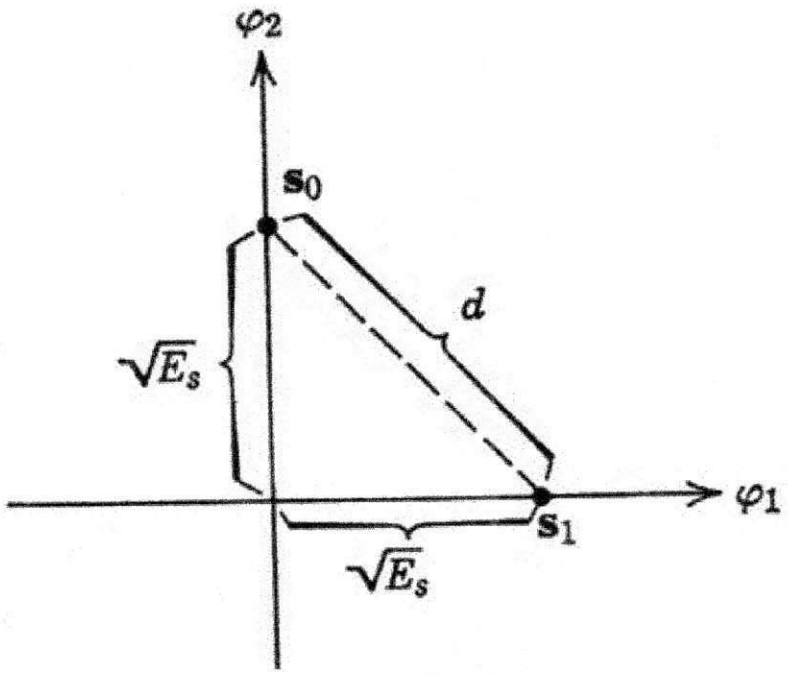

Fig 6: Sinais ortogonais

A Figura 7 apresenta a variação da probabilidade de erro em função da relação sinal ruído para os dois casos estudados.

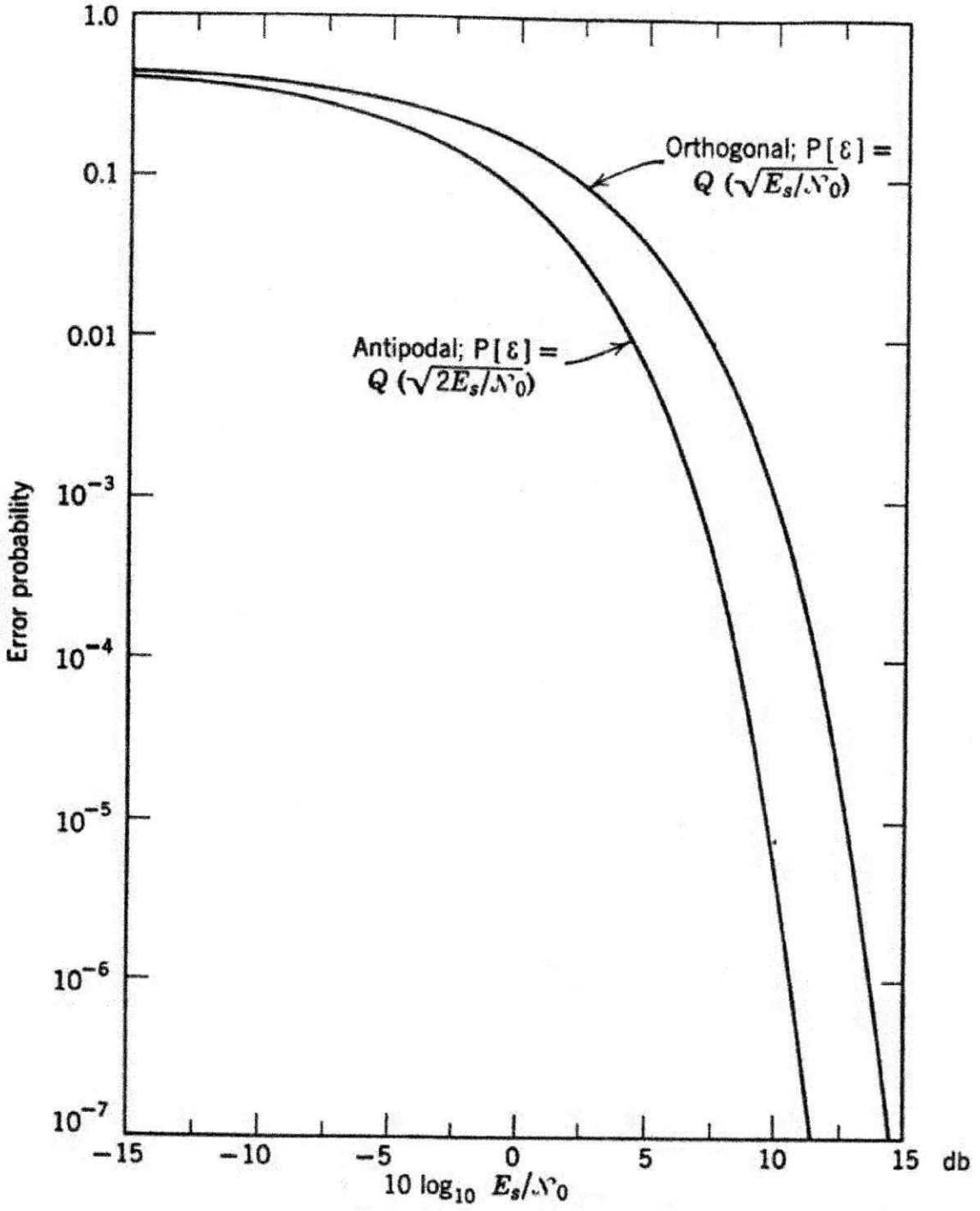

Fig 7: Probabilidade de erro para sinalização binária antipodal e ortogonal com mensagens igualmente prováveis

5-Simulação

5.1-Geração do Ruído

Nas simulações numéricas realizadas, utilizou-se, na representação do ruído gaussiano, uma variável aleatória bidimensional com função de distribuição de probabilidade normal com média zero e variância unitária. Tal variável pode ser obtida a partir de duas outras variáveis também com distribuição normal de média zero porém com variância igual à metade da desejada [3].

A função de distribuição normal com média zero e variância unitária é dada por

$$F(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}dt.$$

Várias técnicas estão disponíveis para geração de seqüências numéricas pseudo-aleatórias com distribuição normal, entre elas está o Método Polar utilizado nesse estudo. Este método fornece duas variáveis independentes normalmente distribuídas dadas duas variáveis, também independentes, com distribuição uniforme.

Sendo assim será apresentado, inicialmente, como foram obtidas as variáveis uniformemente distribuídas.

5.1.1-Geração das seqüências com distribuição uniforme

Assim como no caso da distribuição normal, são vários os métodos existentes para geração de frações aleatórias, ou seja, números aleatórios reais, U_n, normalmente distribuídos entre zero e um. Pelo fato dos computadores só poderem representar um número real com uma precisão finita, essas técnicas geram na verdade inteiros X_n entre zero e algum número m. As frações aleatórias são então obtidas por

$$U_n=\frac{X_n}{m}.$$

Os métodos mais bem sucedidos para geração de números aleatórios são variações do esquema introduzido por D. H. Lehmer em 1948 [2]. Trata-se do *Método Linear Congruencial*, que baseia-se na escolha de quatro "números mágicos", a saber:

Xo, o valor inicial;	Xo ≥ 0.
a, o multiplicador;	a ≥ 0.
c, o incremento;	c ≥ 0.
m, o módulo;	m ≥ Xo, m ≥ a, m ≥ c.

A sequência de números aleatórios desejada é então obtida por

$$X_{n+1}=(aX_n+c)\bmod m,$$

em que n ≥ 0. Essa é a chamada *seqüência linear congruencial*.

Tal sequência não é sempre "aleatória" para qualquer escolha de *Xo*, *a*, *c* e *m*. Na verdade, a seqüência congruencial sempre entra em um *loop*, ou seja, trata-se de uma seqüência periódica. Pode-se mostrar [2] que o período máximo, limitado pelo valor de *m*,

pode ser obtido quando $b=a-1$ for um múltiplo de cada primo dividindo m, além de múltiplo de 4 quando m for múltiplo de 4.

Sendo assim, pode-se satisfazer tais condições simplesmente fazendo $m=2^e$, $e \geq 5$, com o multiplicador a dado por

$$a = 2^k + 1,$$

com $2 \leq k \leq e$. O incremento c deve ser um primo relativo de m, podendo-se fazer $c=1$. A relação recorrente assume então a forma

$$X_{n+1} = \left(\left(2^k + 1\right) X_n + 1\right) \bmod 2^e,$$

que foi de fato a forma utilizada na geração das duas seqüências uniformemente distribuídas necessárias à aplicação do método polar. Seqüências diferentes e independentes são geradas tão somente atribuindo-se diferentes valores ao termo inicial Xo.

5.1.2-Geração das seqüências com distribuição normal – O Método Polar

Conforme mencionado anteriormente, o método polar possibilita a obtenção de duas variáveis independentes X_1 e X_2 com distribuição de probabilidades normal, média zero e variância unitária a partir de duas variáveis independentes U_1 e U_2 com distribuição uniforme. Essa técnica pode ser resumida no seguinte algorítmo[2]:

Algorítmo P

P1: Gere duas variáveis aleatórias independentes U_1 e U_2 uniformemente distribuídas. Faça $V_1=2U_1-1$ e $V_2=2U_2-1$ (assim V_1 e V_2 estão uniformemente distribuídas entre –1 e 1).

P2: Calcule $S= V_1^2+ V_2^2$.

P3: Se $S \geq 1$ retorne para P1.

P4: Calcule X_1 e X_2 de acordo com as equações a seguir:

$$X_1 = V_1\sqrt{\frac{-2\ln S}{S}} \qquad X_2 = V_{21}\sqrt{\frac{-2\ln S}{S}}$$

Essas são as duas variáveis com distribuição gaussiana de média zero e variância unitária desejadas.

5.1.3-Geração do ruído gaussiano bidimensional

Uma vez obtidas as duas variáveis independentes X_1 e X_2 com distribuição normal pode-se gerar uma variável aleatória bidimensional. Na simulação compôs-se essa variável bidimensional como um vetor complexo cujas partes real e imaginária são dadas pelas variáveis X_1 e X_2, no MATLAB

Gauss=complex(X_1,X_2);

Como o método polar gera sequências com variância unitária, a nova variável bidimensional tem o dobro da variância das variáveis unidimensionais. É fácil demonstrar [3] que multiplicando uma sequência numérica por um certo valor sua variância fica multiplicada pelo quadrado deste valor.

Sendo assim, efetuando-se a operação

$$Gauss = \frac{Gauss}{\sqrt{2}}$$

obtém-se, enfim, uma variável gaussiana bidimensional com média zero e variância unitária utilizada na representação do ruído adicionado pelo canal de comunicação aos sinais transmitidos.

5.2-Determinação da Probabilidade de Erro

Para o cálculo da probabilidade de erro utilizou-se a analogia entre sinais e vetores apresentada anteriormente. Por simplicidade, tanto no caso antipodal como no ortogonal admitiu-se que os sinais transmitidos tinham energia unitária, sendo representados pelos vetores:

s1: (-1,0) s2: (1,0) para o esquema antipodal
s1: (0,1) s2: (1,0) para o esquema ortogonal.

5.2.1-Variação da Relação Sinal Ruído

Para se obter a faixa de valores da relação sinal ruído (SNR – *Signal Noise Rate*) desejada é suficiente variar o valor da potência do ruído adicionado ao sinal:

$$SNR = 10\log_{10}\frac{Es}{No},$$

daí

$$No = Es.10^{-\frac{SNR}{10}} = 10^{-\frac{SNR}{10}}.$$

uma vez que os sinais foram considerados como tendo energia unitária. Assim, a partir do ruído com média zero e variância unitária, pode-se obter o ruído com média zero e variância N_0 qualquer simplesmente fazendo [3]

$$N(0,No) = \sqrt{No} \times N(0,1)$$

em que $N(\mu,No)$ representa uma distribuição normal de média μ e variância N_0.

5.2.2- Detecção dos sinais

A caracterização do sinal recebido consistiu em ter o sinal enviado somado ao ruído introduzido pelo canal. A sequência de vetores representando o ruído foi adicionada a um dos vetores-sinal e em seguida ao outro. Tal procedimento assegurou a mesma probabilidade *a priori* dos sinais. Daí foi bastante contar o número de erros cometidos na detecção dos vetores-sinais transmitidos para cada valor da relação sinal ruído.

O critério de decisão pode ser representado pela Figura 8.

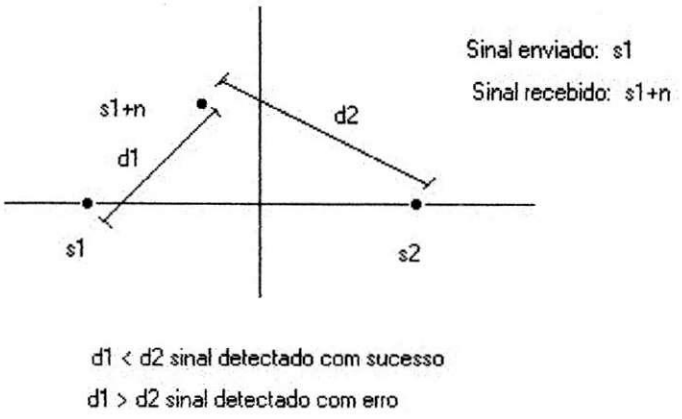

Fig 8: Critério de decisão utilizado.

As regiões de decisão englobavam os pontos mais próximos do sinal em questão que do outro. No caso do esquema antipodal a fronteira de decisão era o eixo das ordenadas. No caso ortogonal, a bissetriz dos quadrantes ímpares.

Um erro na detecção é cometido quando o ruído é intenso o suficiente para deslocar o sinal transmitido de sua região de decisão.

5.2.3- Resultados Obtidos

Os resultados obtidos na simulação podem ser ilustrados pela Figura 9. Nela estão representados os gráficos da probabilidade de erro em função da relação sinal ruído para sinais antipodais e ortogonais, obtidos com a utilização do programa prob_erro2.m apresentado no anexo 2. O gráfico mostra, por exemplo, que a configuração antipodal tem um desempenho superior em 3 dB em relação à ortogonal.

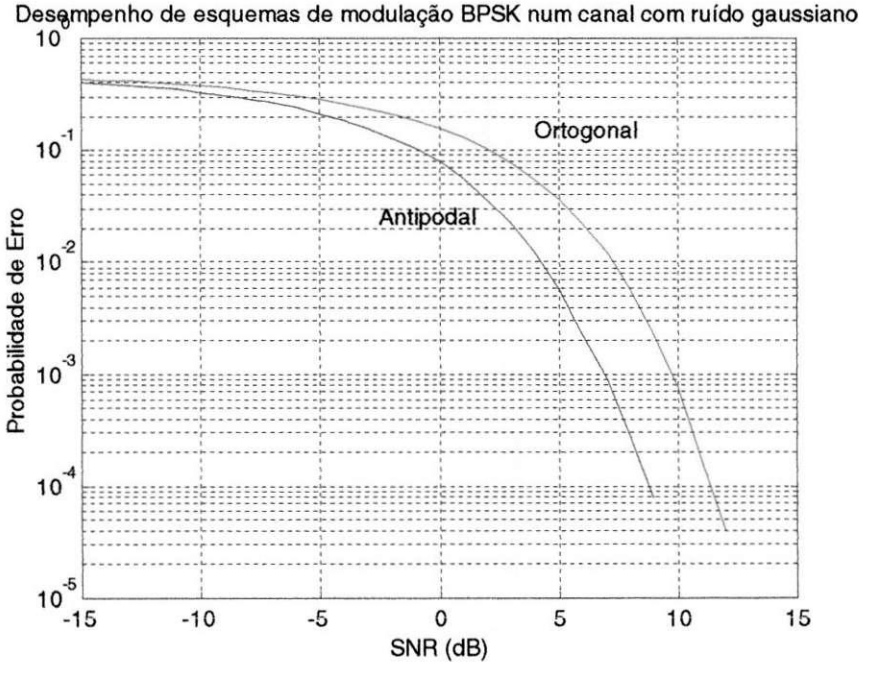

Fig 9: Resultado do programa prob_erro2.m

6-Conclusões

O trabalho apresentado neste relatório possibilitou um estudo mais aprofundado do desempenho de sistemas de modulação BPSK que aquele realizado nas disciplinas da graduação de Engenharia Elétrica. Isso porque além de expor a formulação matemática necessária a este estudo, mostrou meios para se realizar uma análise computacional desse desempenho.

Tal análise tem como virtude a necessidade de se considerar modelos matemáticos que ilustrem a situação física, ponderando-se quais simplificações podem ser utilizadas e que algorítmos ou métodos servem para representar esses modelos num ambiente computacional.

Entre os resultados apresentados foi visto que a transmissão de sinais *M*-ários por um canal perturbado pela adiçao de ruido Gaussiano aditivo define regiões de decisão simples, dadas pelos pontos mais próximos dos vetores sinais representando cada mensagem. Foi visto ainda como o método polar pode ser utilizado para gerar uma variável gaussiana apropriada para modelar o ruído adicionado por esse canal.

Como um estudo complementar ao que foi apresentado, pode-se sugerir uma análise comparativa do desempenho de diferentes constelações *M*-árias em canal perturbado por ruído AWGN buscando-se uma configuração ótima para essa situação.

7-Referências Bibliográficas

[1] Wozencraft, John M. e Jacobs, Irwin Mark; "Principles of Communication Engineering", John Wiley & Sons Inc., 1967

[2] Knuth, Donald E., "The Art of Computer Programming", Addison-Wesley Publishing Company, 1969

[3] Papoulis, Athanasios, "Probability, Random Variables ans Stochastic Processes", McGraw-Hill, 1965

ANEXO 1

```
%                          Programa prob_erro.m

%             Este programa determina a probabilidade de erro na detecção
de sinais
%        BPSK (antipodal e ortogonal) com mesma probabilidade a priori
transmitidos por
%        um canal submetido a um ruído gaussiano.

%        Os "loops for" tornam o programa mais lento mas permitem um
entendimento mais
%        simples pela correspondência com outras liguagens de programação.
clear;
%Geração das sequências com distribuição de probabilidade uniforme

m=2^15;                              % Definição do módulo
a=2^5+1;                             % Definição do multiplicador
u1(1)=8;                             % Valor inicial da primeira sequência
u2(1)=13;                            % Valor inicial da  segunda sequência
for n=1:m-1
   u1(n+1)=mod(a*u1(n)+1,m);         % Cálculo das sequências congruenciais com
   u2(n+1)=mod(a*u2(n)+1,m);         % incremento c=1
end
uu1=u1/m;                            % Sequências com distribuição uniforme
entre
uu2=u2/m;                            % zero e um.

%Geração das sequências com distribuição de probabilidades normal
%Método Polar

v1=2*uu1-1;
v2=2*uu2-1;
s=v1.^2+v2.^2;
k=(s<1 & s~=0);                      % vetor lógico (zeros e uns) para
selecionar os valores
s=s(k);                              % necessários ao prosseguimento do
algorítmo. Corresponde
v1=v1(k);                            % ao passo P3 do algorítmo P.
v2=v2(k);
x1=v1.*sqrt(-2*log(s)./s);           % Sequências com distribuição gaussiana de
média zero e
x2=v2.*sqrt(-2*log(s)./s);           % variância unitária.
r=2*length(x1);

% Geração da variável aleatória bidimensional "gauss" com distriguição
gaussiana.

for n=1:r/2
   gauss(n)=complex(x1(n),x2(n));
end
gauss=gauss/2^.5;                    %corrige a variância de 2 para 1

% Cálculo da probabilidade de erro

db1=-15;                             % variação em dB da relação sinal ruído
Es/No
db2=15;
```

```
% Configuração antipodal com sinais de energia unitária

z1=complex(-1,0);
z2=complex(1,0);
for SNR=db1:db2
   nerro=0;
   for n=1:r/2
      No=10^(-SNR/10);           % Variação da Potência do ruído de acordo
com o valor
      lgauss(n)=gauss(n)*No^.5;  % de SNR (a energia do sinal é mantida
fixa).
   end
   for n=1:r/2
      if(abs(z1+lgauss(n)-z2)<abs(lgauss(n))) nerro=nerro+1;
      end                        % O ruído é somado ao sinal (vetor) z1 e
calcula-se
   end                           % o número de erros cometidos na
"detecção".
   for n=1:r/2
      if(abs(z2+lgauss(n)-z1)<abs(lgauss(n))) nerro=nerro+1;
      end                        % O ruído é somado ao sinal (vetor) z2 e
calcula-se
   end                           % o número de erros cometidos na
"detecção".
   perro1(SNR-db1+1)=nerro/r;    % A probabilidade de erro é tão somente a
razão entre
end                              % o número total de erros e a quantidade
de
                                 % valores da sequência representativa do
ruído

% Configuração ortogonal com sinais de energia unitária

z1=complex(0,1);
z2=complex(1,0);
for SNR=db1:db2
   nerro=0;
   for n=1:r/2
      No=10^(-SNR/10);
      lgauss(n)=gauss(n)*No^.5;
   end
   for n=1:r/2
      if(abs(z1+lgauss(n)-z2)<abs(lgauss(n))) nerro=nerro+1;
      end
   end
   for n=1:r/2
      if(abs(z2+lgauss(n)-z1)<abs(lgauss(n))) nerro=nerro+1;
      end
   end
   perro2(SNR-db1+1)=nerro/r;
end

% Gráfico da Probabilidade de Erro versus Relação Sinal Ruído (entre os
valores db1
% e db2 selecionados) para as configurações antipodal e ortogonal.

t=db1:db2;
semilogy(t,perro1,t,perro2,'r'),grid on;  %Probabilidade de erro em escala
logarítmica
```

```
title('Desempenho de esquemas de modulação BPSK num canal com ruído
gaussiano');
xlabel('SNR (dB)');
ylabel('Probabilidade de Erro');
text(-2.5,.025,'Antipodal');
text(4,.15,'Ortogonal')
```

Anexo 2

```
%                          Programa prob_erro2.m

%            Este programa determina a probabilidade de erro na detecção
de sinais
%       BPSK (antipodal e ortogonal) com mesma probabilidade a priori
transmitidos por
%       um canal submetido a um ruído gaussiano.

%       Os "loops for" encontrados na versão original "prob_erro.m" foram,
em sua
%       maioria, substituidos por comandos equivalentes de modo a dar mais
velocidade
%       à execução do programa.
clear;
%Geração das sequências com distribuição de probabilidade uniforme

m=2^15;                            % Definição do módulo
a=2^5+1;                           % Definição do multiplicador
u1(1)=8;                           % Valor inicial da primeira sequência
u2(1)=13;                          % Valor inicial da  segunda sequência
for n=1:m-1
   u1(n+1)=mod(a*u1(n)+1,m);       % Cálculo das sequências congruenciais com
   u2(n+1)=mod(a*u2(n)+1,m);       % incremento c=1
end
uu1=u1/m;                          % Sequências com distribuição uniforme
entre
uu2=u2/m;                          % zero e um.

%Geração das sequências com distribuição de probabilidades normal
%Método Polar

v1=2*uu1-1;
v2=2*uu2-1;
s=v1.^2+v2.^2;
k=(s<1 & s~=0);                    % vetor lógico (zeros e uns) para
selecionar os valores
s=s(k);                            % necessários ao prosseguimento do
algorítmo. Corresponde
v1=v1(k);                          % ao passo P3 do algorítmo P.
v2=v2(k);
x1=v1.*sqrt(-2*log(s)./s);         % Sequências com distribuição gaussiana de
média zero e
x2=v2.*sqrt(-2*log(s)./s);         % variância unitária.
r=2*length(s);

% Geração da variável aleatória bidimensional "gauss" com distriguição
gaussiana.

gauss=complex(x1,x2);
gauss=gauss/2^.5;                  % corrige a variância de 2 para 1

% Cálculo da probabilidade de erro

db1=-15;                           % variação em dB da relação sinal ruído
Es/No
```

```
db2=15;

% Configuração antipodal com sinais de energia unitária

z1=complex(-1,0);
z2=complex(1,0);

SNR=db1:db2;
No=10.^(-SNR/10);                % Variação da Potência do ruído de acordo
com o valor
lgauss=(gauss.')*No.^.5;         % de SNR (a energia do sinal é mantida
fixa).

k1=(abs(z1+lgauss-z2)<abs(lgauss)); % O ruído é somado ao sinais (vetores)
z1 e z2
k2=(abs(z2+lgauss-z1)<abs(lgauss)); % calcula-se o número de erros
cometidos
nerro1=sum(k1)+sum(k2);             % na "detecção".
perro1=nerro1/r;                 % A probabilidade de erro é tão somente a
razão entre
                                 % o número total de erros e a quantidade
de
                                 % valores da sequência representativa do
ruído

% Configuração ortogonal com sinais de energia unitária

z1=complex(0,1);
z2=complex(1,0);
k1=(abs(z1+lgauss-z2)<abs(lgauss));
k2=(abs(z2+lgauss-z1)<abs(lgauss));
nerro2=sum(k1)+sum(k2);
perro2=nerro2/r;

% Gráfico da Probabilidade de Erro versus Relação Sinal Ruído (entre os
valores db1
% e db2 selecionados) para as configurações antipodal e ortogonal.

t=db1:db2;
semilogy(t,perro1,t,perro2,'r'),grid on;  %Probabilidade de erro em escala
logarítmica
title('Desempenho de esquemas de modulação BPSK num canal com ruído
gaussiano');
xlabel('SNR (dB)');
ylabel('Probabilidade de Erro');
text(-2.5,.025,'Antipodal');
text(4,.15,'Ortogonal')
```

Anexo 3

```
/*                    proberro.cpp

 Este programa determina a probabilidade de erro na detecção de sinais
 BPSK (antipodal e ortogonal) com mesma probabilidade a priori transmitidos
 por um canal submetido a um ru¡do gaussiano.  */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <complex.h>
void antipodal(complex *gauss,int r);
void ortogonal(complex *gauss,int r);
double perro1[31],perro2[31];
void main(void)
{
FILE *p;
char ch;
int m,a;
int r,n,u1[128],u2[128];
double
z1,z2,uu1[128],uu2[128],s[128],v1[128],v2[128],x1[128],x2[128],ss[128];
complex *gauss;
r=0;
m=128;       /* Definição do módulo */
a=33;        /* Definição do multiplicador */
u1[0]=8;     /* Valor inicial da primeira sequência */
u2[0]=13;    /* Valor inicial da segunda sequência */

/* Geração das sequências com distribuição de probabilidade uniforme */
/* M,todo Linear Congruencial */
for(n=0;n<(m-1);++n) {u1[n+1]=(a*u1[n]+1)%m;
                u2[n+1]=(a*u2[n]+1)%m;}

/* Sequ^ncia uniformemente distribu¡da entre zero e um*/

for(n=0;n<m;++n) {uu1[n]=(double)u1[n]/m;
                uu2[n]=(double)u2[n]/m;}

/* Geração das sequências com distribuição de probabilidades normal */
/* M,todo Polar*/
for(n=0;n<m;++n) {v1[n]=2*uu1[n]-1;
                v2[n]=2*uu2[n]-1;
                s[n]=v1[n]*v1[n]+v2[n]*v2[n];}
p=fopen("ruido.txt","w+");
if(!p) {printf("\nOcorreu um erro na abertura do arquivo.");
      printf("\nO programa ser  finalizado.");
      exit(1);}

/* Imprime no arquivo ruido.txt os valores v lidos das sequências com
   distribuição normal. Equivale ao passo P3 do algorítmo P. */

for(n=0;n<m;++n){if(s[n]<1 & s[n]!=0)
               {x1[n]=v1[n]*sqrt(-2*log(s[n])/s[n]);
                x2[n]=v2[n]*sqrt(-2*log(s[n])/s[n]);
                fprintf(p,"%lf ",x1[n]);
```

```
                fprintf(p,"%lf ",x2[n]);}
                else x1[n]=x2[n]=0;}
rewind(p);

/* A quantidade de valores v lidos nas duas sequˆncias , contado
   e armazenado na vari vel r */

while((ch=getc(p))!=EOF) if(ch==' ') ++r;

/* Vari vel aleatória bidimensional "gauss" com distribuição gaussiana
   , lida a partir dos valores impressos em ruido.txt */

gauss=(complex *)calloc((r/2),sizeof(complex));
rewind(p);
for(n=0;n<(r/2);++n) {fscanf(p,"%lf",&z1);
                      fscanf(p,"%lf",&z2);
                      gauss[n]=complex(z1,z2);}
fclose(p);

/* Corrige a varifncia de 2 para 1 */

for(n=0;n<(r/2);++n) gauss[n]=gauss[n]/sqrt(2);

/* As funções a seguir calculam as probabilidades de erro e as armazenam
nas variveis globais perro1 e perro2 */
antipodal(gauss,r);
ortogonal(gauss,r);
p=fopen("proberro.txt","w");
if(!p) {printf("\nOcorreu um erro na abertura do arquivo. O programa ser
finalizado");
      exit(1);}
/* Escreve o resultado das probabilidades de erro calculadas pelas funções
   antipodal e ortogonal no arquivo proberro.txt */

fprintf(p,"Configuração Antipodal\n");
fprintf(p,"SNR          Probabilidade de Erro\n");
for(n=0;n<31;++n)
   fprintf(p,"%d dB     %lf \n",n-15,perro1[n]);

fprintf(p,"Configuração Ortogonal\n");
fprintf(p,"SNR          Probabilidade de Erro\n");
for(n=0;n<31;++n)
   fprintf(p,"%d dB     %lf \n",n-15,perro2[n]);
fclose(p);
}

/* C lculo da probabilidade de erro */
/* Configuração antipodal com sinais de energia unitária */

void antipodal(complex *gauss,int r)
{
 complex *lgauss;
 double No;
 int n,SNR,nerro;
 complex  z1=complex(-1,0),z2=complex(1,0);
 lgauss=(complex *)calloc((r/2),sizeof(complex));
 for(SNR=-15;SNR<16;++SNR)
 { nerro=0;
   /* Variação da potência do ru¡do de acordo com o valor
      de SNR (a energia do sinal , mantida fixa*/
```

```
   No=(double)pow(10,(double)-SNR/10);
   for(n=0;n<r/2;++n)                /*O ruído , somado aos
sinais(vetores)*/
      lgauss[n]=gauss[n]*sqrt(No); /*e calcula-se o nÉmero de erros      */
   for (n=0;n<r/2;++n)               /*cometidos na detecção
*/
      if (abs(z1+lgauss[n]-z2)<abs(lgauss[n]))  ++nerro;
   for (n=0;n<r/2;++n)
      if (abs(z2+lgauss[n]-z1)<abs(lgauss[n]))  ++nerro;
   perro1[SNR+15]=(double) nerro/r;  /*A probabilidade de erro , a razão
*/
 }    /*entre o número de erros e a quantidade de valores da sequência
*/
  free(lgauss);   /* representativa de ruídos                          */
}

/* Configuração antipodal com sinais de energia unitária */

void ortogonal(complex *gauss,int r)
{
 complex *lgauss;
 double No;
 int n,SNR,nerro;
 complex  z1=complex(0,1),z2=complex(1,0);
 lgauss=(complex *)calloc((r/2),sizeof(complex));
 for(SNR=-15;SNR<16;++SNR)
 { nerro=0;
   No=(double) pow(10,(double)-SNR/10);
   for(n=0;n<r/2;++n)
      lgauss[n]=gauss[n]*sqrt(No);
   for (n=0;n<r/2;++n)
      if (abs(z1+lgauss[n]-z2)<abs(lgauss[n]))  ++nerro;
   for (n=0;n<r/2;++n)
      if (abs(z2+lgauss[n]-z1)<abs(lgauss[n]))  ++nerro;
   perro2[SNR+15]=(double) nerro/r;
 }
  free(lgauss);
}
```