

JOHN RICHARD FITZGERALD JUNIOR

**TÓPICOS SOBRE A CARACTERIZAÇÃO DE DESEMPENHO DE
SISTEMAS DS-CDMA**

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de Mestre
em Engenharia

São Paulo
1996

JOHN RICHARD FITZGERALD JUNIOR

**TÓPICOS SOBRE A CARACTERIZAÇÃO DE DESEMPENHO DE
SISTEMAS DS-CDMA**

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de Mestre
em Engenharia

Área de Concentração:
Engenharia de Sistemas Eletrônicos

Orientador:
Prof. Dr. Paul Jean Etienne Jeszensky

Fitzgerald Junior, John Richard

Tópicos sobre a caracterização de desempenho de sistemas DS-CDMA. São Paulo. 1996
p.134.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Eletrônica - Área de Sistemas Eletrônicos.

1. CDMA 2. Espalhamento Espectral 3. Spread Spectrum I.
Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Eletrônica - Área de Sistemas Eletrônicos.

AGRADECIMENTOS

Agradeço ao Dr. Paul Jean Etienne Jeszensky pelo incentivo, ajuda e compreensão durante a orientação deste trabalho.

Agradeço á Fundação de Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pelo incentivo na forma de bolsa, durante a execução deste trabalho.

RESUMO

Este trabalho apresenta métodos de cálculo de desempenho de sistemas de comunicação multiplexados por código (CDMA-Code Division Multiple Access) através de várias abordagens com limites de validade diversos.

A ênfase do trabalho encontra-se em sistemas DS-CDMA (Direct Sequence-CDMA) com a utilização de seqüências de assinatura determinísticas. Aborda-se ainda, para efeito comparativo, um método de cálculo com seqüências aleatórias.

O desempenho é caracterizado pela probabilidade de erro de bit do sistema que é o índice de mérito aqui considerado.

ABSTRACT

This work shows methods to evaluate the performance of Code Division Multiple Access systems through many approaches with restricted validation limits.

The main body of the work concerns Direct Sequence Systems with the use of deterministic code sequences. One evaluation method is developed using random code sequences for comparative effects.

The performance is characterized by the bit error probability of the system which is the quality parameter here considered.

Índice

1 - Introdução	1
2 - Modelamento do sistema CDMA assíncrono	2
3 - Cálculo básico da probabilidade de erro de bit em sistemas CDMA	3
4 - Capacidade anti-interferência	7
5 - Abordagem de cálculo da probabilidade de erro de bit no pior caso e melhor caso	10
6 - Abordagem segundo Pursley	12
7 - Abordagem segundo Weber	18
8 - Abordagem segundo Nazari e Ziemer	22
9 - Sistemas quase-síncronos (QS-CDMA)	28
10 - Abordagem gaussiana de Lehnert para seqüências aleatórias	32
11 - Simulações	38
12 - Comparação entre os diversos métodos de cálculo da probabilidade de erro de bit / Conclusões	53
13 - Apêndices	
A - Propriedades das funções de correlação cruzada parcial no tempo	65
B - Seqüências pseudo-aleatórias lineares	68
C - Densidade espectral de potência de uma forma de onda aleatória	77
D - Média e variância do ruído branco gaussiano na saída do receptor	80
E - Alguns detalhes do espalhamento e da capacidade do sistema CDMA definido na IS-95	81
F - Listagem dos programas desenvolvidos	84
14 - Referências bibliográficas	132

GLOSSÁRIO

AMPS	-	do inglês “Advanced Mobile Phone Service”.
BPSK	-	do inglês “Binary Phase Shift Keying” e se refere uma forma de modulação digital.
C/I	-	relação potência de portadora para interferência.
CDMA	-	do inglês “Code Division Multiple Access” e se refere a uma forma de multiplexação de sinais através do uso de códigos.
CHIP	-	refere-se ao código de assinatura, representando uma unidade de informação desta.
DS	-	do inglês “Direct Sequence” e se refere a uma forma de espalhamento espectral utilizado em CDMA.
ERB	-	estação rádio base, em telefonia celular.
FDP	-	função densidade de probabilidade.
FH	-	do inglês “Frequency Hopping” e se refere a uma forma de espalhamento espectral.
FM	-	do inglês “Frequency Modulation” e se refere a uma forma de comunicação por modulação de frequência.
JAMMER	-	interferência proposital.
MUI	-	do inglês “Multiuser Interference” e se refere a interferência multi-usuário.
PCM	-	do inglês “Pulse Code Modulation” e se refere a uma forma de comunicação por codificação de pulsos.
PN	-	do inglês “Pseudo Noise”, designação para SMC.
SMC	-	Sequências de Máximo Comprimento.
SNR	-	do inglês “Signal to Noise Ratio” e se refere a relação sinal para ruído.
SS	-	do inglês “Spread Spectrum” e se refere a comunicação por espalhamento espectral.
TH	-	do inglês “Time Hopping” e se refere a uma forma de espalhamento espectral.
UM	-	unidade móvel, em telefonia celular

1 - INTRODUÇÃO

A história do desenvolvimento de sistemas SS remonta aos anos 20, com aplicações exclusivamente militares e daí sua pouca, ou nenhuma divulgação na literatura aberta, ref. [17].

O uso inicial do espalhamento espectral, baseados em documentos oficiais, ocorreu durante a 2ª Guerra Mundial em sistemas de radar, com o intuito de dificultar a interceptação de sinais transmitidos. A importância de obter um sistema anti-interferência era tanta que num determinado momento 90% dos engenheiros eletrônicos, nos USA, encontravam-se envolvidos em algum programa na área.

Em 1976 é publicado, por Robert Dixon, o primeiro livro sobre o assunto: "Spread Spectrum Systems", que é precursor de vários outros e pode ser considerado como um marco a partir do qual a discussão sobre o assunto torna-se pública. Essa discussão aberta sobre o assunto, provavelmente, originou-se das aplicações não militares de SS, assim como de um avanço tecnológico que tornou o processamento complexo de sinais realizável.

Uma definição para um sistema de espalhamento espectral que reflete as características desta técnica é:

"SS é um modo de transmissão em que o sinal ocupa uma banda superior à mínima necessária para àquela informação; o alargamento (ou espalhamento) da banda é obtido por meio de um código, que é independente da informação, e uma recepção sincronizada com o mesmo código é usada no receptor para concentrar o sinal em uma banda menor e posterior detecção".

Segundo esta definição, sistemas tradicionais de modulação/codificação como FM, PCM etc, não constituem um sistema SS.

Várias características benéficas podem ser obtidas com o espalhamento espectral, entre as quais: capacidade de rejeitar interferências, baixa probabilidade de interceptação, sistemas multiusuários multiplexados por código, sistemas de radar precisos etc.

Os meios usuais para obter-se o espalhamento espectral são:

- DS (Direct Sequence): onde uma sequência pseudo-aleatória rápida, com taxa de variação maior que a dos dados, causa transições na fase da portadora modulada com a informação.
- FH (Frequency Hopping): onde a portadora modulada é deslocada em frequência de uma maneira pseudo-aleatória.
- TH (Time Hopping): onde surtos do sinal com a informação são iniciados em instantes pseudo-aleatórios.
- Sistemas Híbridos: empregam duas, ou mais, das técnicas acima.

No trabalho que se segue é utilizado a forma de espalhamento espectral DS com receptores do tipo correlator que, embora não sejam ótimos para este sistema de transmissão, possuem desempenho próximo destes, com a vantagem da facilidade de implementação, razão pela qual é o mais utilizado na maioria dos sistemas atuais.

O objetivo do trabalho é caracterizar e simular diversos métodos existentes para o cálculo da probabilidade de erro de bit em sistemas DS assíncronos, quase-síncronos e síncronos.

2 - MODELAMENTO DO SISTEMA CDMA ASSÍNCRONO

O modelamento de sistema CDMA assíncrono adotado neste trabalho encontra-se representado na Figura 1, onde cada usuário transmite um sinal $s_i(t)$, para $i = 1, 2, \dots, K$, onde K é o número de usuários. Admite-se que na entrada do receptor tem-se todos os sinais transmitidos somados e que o efeito de todas as interferências introduzidas pelo meio de transmissão é um ruído branco gaussiano aditivo $n(t)$ de densidade espectral $N_0/2$. Este modelamento é o mais utilizado na literatura e foi introduzido por Pursley, ref. [5].

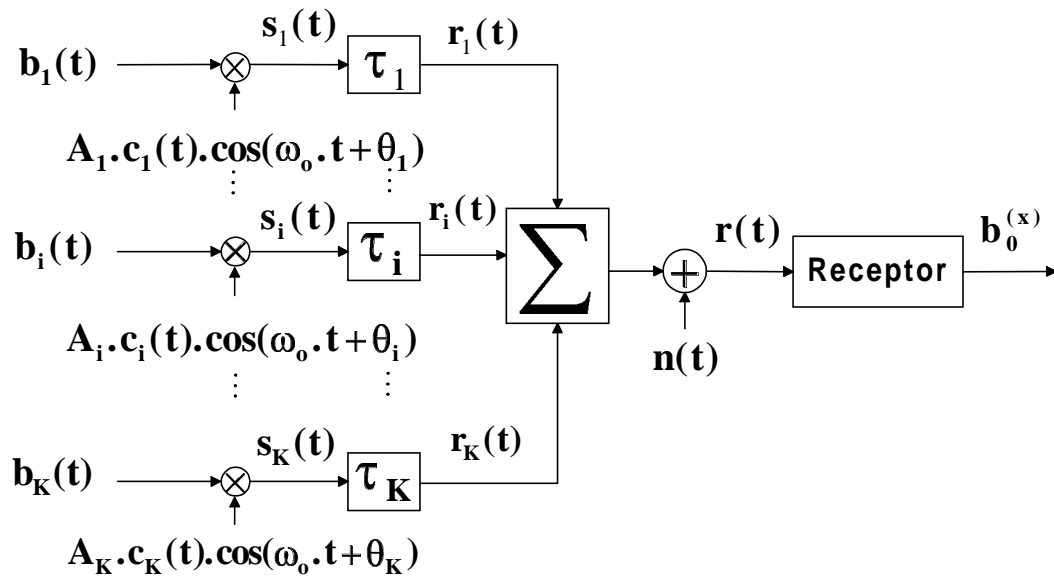


Figura 1 - Modelamento do sistema CDMA assíncrono

Cada sinal $s_i(t)$ é formado pela informação $b_i(t)$ multiplicada pelo código $c_i(t)$ e modulados convenientemente, isto é:

$$s_i(t) = A_i \cdot b_i(t) \cdot c_i(t) \cdot \cos(\omega_0 \cdot t + \theta_i) \quad (1)$$

Com:

$$b_i(t) = \pm 1 \text{ no intervalo } k \cdot T \leq t < (k+1) \cdot T \text{ com } k = \dots, -2, -1, 0, +1, +2, \dots \quad (2)$$

$$c_i(t) = \pm 1 \quad \text{no intervalo} \quad j \cdot T_c \leq t < (j+1) \cdot T_c \quad \text{com} \quad j = \dots, -2, -1, 0, +1, +2, \dots \quad (3)$$

Onde $A_i = \sqrt{2 \cdot P_i}$ com P_i representando a potência do i -ésimo sinal transmitido.

T_c é chamado de tempo de chip.

$b_i(t)$ são as informações binárias transmitidas.

$c_i(t)$ é chamado de assinatura do sinal e a forma de onda no intervalo de validade de tempo T_c é denominada de forma de onda do “chip”.

$a_x^{(y)}$ se refere ao x -ésimo elemento discreto da sequência \mathbf{a} do usuário y . Por exemplo, os elementos discretos do sinal de assinatura $\mathbf{c}_2(t)$, segunda esta representação são: $\{c_0^{(2)}, c_1^{(2)}, c_2^{(2)}, c_3^{(2)}, c_4^{(2)}, c_5^{(2)}, \dots\}$.

Quando não for necessário especificar a sequência, a representação \mathbf{a}_x indicará somente o elemento discreto da sequência \mathbf{a} .

Admite-se que o receptor consiste apenas de um demodulador (perfeitamente sincronizado) com o objetivo de trazer o sinal transmitido para a banda base, multiplicando em seguida pela sequência de espalhamento do usuário $\mathbf{c}_x(t)$ (perfeitamente sincronizado), onde x se refere ao usuário transmissor que se deseja receber. O modelo de receptor encontra-se representado na Figura 2.

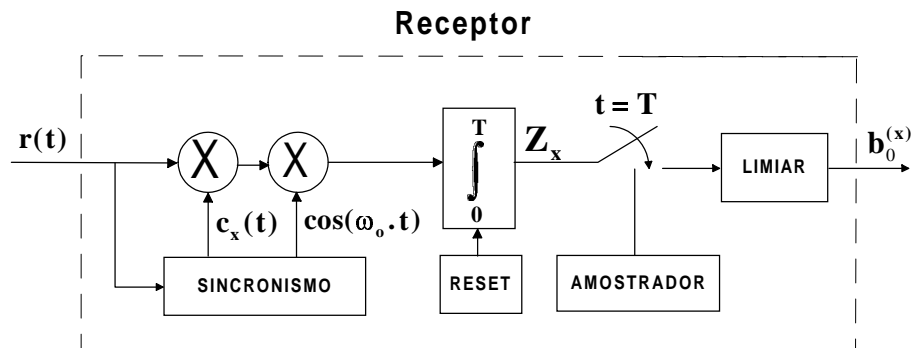


Figura 2 - Modelo de receptor para CDMA

No que se segue admite-se a relação T/T_c resulta um valor inteiro. O caso em que a relação anterior não se mantém inteira encontra-se pouco divulgado abertamente na literatura.

3 - CÁLCULO BÁSICO DA PROBABILIDADE DE ERRO DE BIT EM SISTEMAS CDMA

A ideia intuitiva do que é um sistema CDMA foi dada por Golomb, ref. [21]:

"A conversa de duas pessoas (chamadas de emissoras) que querem falar de uma só vez com outras duas pessoas (chamadas de receptoras) pode ocorrer de diversas maneiras:

- cada pessoa emissora pode falar uma de cada vez durante um intervalo de tempo determinado o que caracterizaria uma divisão no tempo.

- cada pessoa emissora pode falar em uma sala separada com a pessoa receptora o que caracterizaria uma divisão no espaço.

- cada pessoa emissora pode falar: uma com voz de tenor e outra com a voz de baixo, e o ouvido de uma pessoa receptora estar equipado com um filtro passa-baixas e o da outra com filtro passa-altas. Isto caracterizaria uma divisão de frequência do canal.

A divisão por código consiste em transmitir a informação ocupando as mesmas regiões no tempo, espaço e frequência e os receptores serem capazes de distinguir uma informação da outra sem se confundirem.

- se cada uma das pessoas emissoras falar em uma linguagem diferente e as pessoas receptoras conhecerem apenas a linguagem que estão esperando receber tem-se o equivalente do sistema de múltiplo acesso por divisão de código (CDMA).

A pessoa emissora que fala em uma linguagem que a pessoa receptora desconhece estará atrapalhando (ou interferindo); o quanto, depende da potência da voz bem como de quanto em um intervalo de uma palavra, por exemplo, a linguagem desconhecida é capaz de confundir a tal ponto que não é possível entender a palavra desejada sendo recebida."

Neste capítulo é realizado um cálculo genérico da probabilidade de erro de bit e é introduzido uma terminologia utilizada durante todo o restante do trabalho. Algumas das equações obtidas são a base do cálculo da probabilidade de erro de bit de todos os métodos abordados.

Supondo que o sinal transmitido pelo usuário i chegue no receptor com um atraso τ_i , o sinal na entrada do receptor, devido a este usuário é $r_i(t)$:

$$r_i(t) = s_i(t - \tau_i) = A_i \cdot b_i(t - \tau_i) \cdot c_i(t - \tau_i) \cdot \cos[\omega_0(t - \tau_i) + \theta_i] \quad (4)$$

Fazendo $\phi_i = \theta_i - \omega_0 \cdot \tau_i$ e considerando os K usuários na presença de um ruído $n(t)$ resulta:

$$r(t) = \sum_{i=1}^K r_i(t) + n(t) = \sum_{i=1}^K A_i \cdot b_i(t - \tau_i) \cdot c_i(t - \tau_i) \cdot \cos(\omega_0 \cdot t + \phi_i) + n(t) \quad (5)$$

O sinal na saída do correlator Z_x , no instante T , é:

$$Z_x = \int_0^T r(t) \cdot c_x(t) \cdot \cos(\omega_0 \cdot t) \cdot dt \quad (6)$$

Supondo que o receptor possui uma curva de resposta abaixo de $2 \cdot \omega_0$ (o que ocorre em implementações práticas):

$$Z_x = \text{Inf}_x + \text{Interf}_x + \eta_x \quad (7)$$

Onde:

$$\mathbf{Inf}_x = \frac{\mathbf{A}_x \cdot \mathbf{T} \cdot \mathbf{b}_0^{(x)}}{2} \quad (8)$$

$$\mathbf{Interf}_x = \frac{1}{2} \times \int_0^T \sum_{\substack{i=1 \\ i \neq x}}^K \mathbf{A}_i \cdot \mathbf{b}_i(t - \tau_i) \cdot \mathbf{c}_i(t - \tau_i) \cdot \mathbf{c}_x(t) \cdot \cos(\phi_i) \cdot dt \quad (9)$$

$$\eta_x = \int_0^T \mathbf{n}(t) \cdot \mathbf{c}_x(t) \cdot \cos(\omega_0 \cdot t) \cdot dt \quad (10)$$

Os fatores acima possuem o seguinte significado físico:

\mathbf{Inf}_x é a informação que se deseja receber.

$\mathbf{b}_0^{(x)} \in \{1, -1\}$ é o dado desejado, recebido no intervalo $[0, T]$.

\mathbf{Interf}_x é a interferência indesejável dos demais usuários, também chamada de interferência multi-usuário, a qual deseja-se minimizar.

η_x é a interferência devida ao ruído branco gaussiano. Demonstra-se no Apêndice D que η_x é uma variável aleatória gaussiana de média nula e variância $(N_0 \cdot T) / 4$.

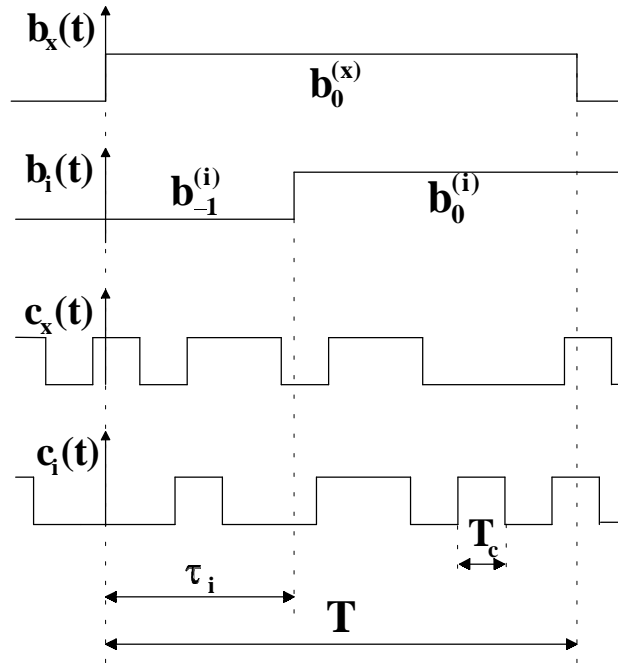


Figura 3

Definindo-se variáveis abaixo, e notando-se as formas de onda da Figura 3, consegue-se desmembrar um pouco mais a interferência **Interf_x**:

$$\mathbf{R}_{i,x}(\tau) = \int_0^T \mathbf{c}_i(t-\tau) \cdot \mathbf{c}_x(t) \cdot dt \quad (11)$$

$$\hat{\mathbf{R}}_{i,x}(\tau) = \int_{-\tau}^T \mathbf{c}_i(t-\tau) \cdot \mathbf{c}_x(t) \cdot dt \quad (12)$$

$$\mathbf{f}_{i,x}(\tau) = \mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau) \quad (13)$$

$$\hat{\mathbf{f}}_{i,x}(\tau) = \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau) \quad (14)$$

E assim:

$$\mathbf{Interf}_x = \frac{1}{2} \times \sum_{\substack{i=1 \\ i \neq x}}^K \mathbf{A}_i \cdot [\mathbf{f}_{i,x}(\tau_i) + \hat{\mathbf{f}}_{i,x}(\tau_i)] \cdot \cos(\phi_i) \quad (15)$$

$$\mathbf{Interf}_x = \frac{1}{2} \times \sum_{\substack{i=1 \\ i \neq x}}^K \mathbf{A}_i \cdot [\mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i)] \cdot \cos(\phi_i) \quad (16)$$

Onde:

- $\mathbf{R}_{i,x}(\tau_i)$ e $\hat{\mathbf{R}}_{i,x}(\tau_i)$ são conhecidas como funções de correlação cruzada parcial contínua no tempo par e ímpar, respectivamente.

- $\mathbf{b}_{-1}^{(i)}$ e $\mathbf{b}_0^{(i)}$ são dois bits de dados consecutivos de informação transmitidos pelo i-ésimo usuário no intervalo de tempo de um bit de informação do usuário x-ésimo, o qual deseja-se receber.

Da expressão (16) nota-se que uma forma de diminuir o módulo de **Interf_x** é com a escolha adequada de seqüências de código, de modo a minimizar os módulos de $\mathbf{R}_{i,x}(\tau_i)$ e $\hat{\mathbf{R}}_{i,x}(\tau_i)$ simultaneamente.

Seguindo a padronização de notação introduzida por Pursley, ref. [5], definem-se as seguintes variáveis:

$$\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) = \mathbf{T}^{-1} \cdot [\mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i)] \cdot \cos(\phi_i) \quad (17)$$

$$\gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi}) = \sum_{\substack{i=1 \\ i \neq x}}^K \mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau_i, \phi_i) \quad (18)$$

A variável $\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau_i, \phi_i)$ é chamada de interferência de múltiplo acesso normalizada na saída do x-ésimo receptor devido ao i-ésimo sinal interferente e encontra-se limitada superiormente e inferiormente por $-1 \leq \mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau_i, \phi_i) \leq 1$.

Reescrevendo \mathbf{Interf}_x e admitindo-se que todas as amplitudes, a do sinal de interesse e dos sinais interferentes, tem a mesma amplitude na entrada do receptor, isto é, $\mathbf{A}_i = \mathbf{A}$ (o problema de diferentes níveis de sinais na entrada do receptor é importante em CDMA, pois um aumento no nível do sinal desejado implica em uma melhor recepção deste, porém aumenta a interferência deste mesmo sinal nos outros receptores, prejudicando o sistema como um todo) temos:

$$\mathbf{Interf}_x = \frac{\mathbf{A} \cdot \mathbf{T}}{2} \times \gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi}) \quad (19)$$

$$\mathbf{Z}_x = \frac{\mathbf{A} \cdot \mathbf{T}}{2} \times [\mathbf{b}_0^{(x)} + \gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi})] + \eta_x \quad (20)$$

Sabendo as características da variável η_x pode-se desenvolver uma expressão básica para o cálculo da probabilidade de erro de bit \mathbf{P}_e , na saída do receptor em consideração, no caso de transmissão de mensagens equiprováveis:

$$\mathbf{P}_e = \frac{1}{2} \times \text{Prob}\{\mathbf{b}_0^{(x)} = 1 \text{ e } \mathbf{Z}_x < 0\} + \frac{1}{2} \times \text{Prob}\{\mathbf{b}_0^{(x)} = -1 \text{ e } \mathbf{Z}_x > 0\} \quad (21)$$

Devido à simetria do problema:

$$\mathbf{P}_e = \text{Prob}\{\mathbf{b}_0^{(x)} = 1 \text{ e } \mathbf{Z}_x < 0\} \quad (22)$$

Portanto:

$$\mathbf{P}_e = \text{Prob}\left\{\left[\frac{\mathbf{A} \cdot \mathbf{T}}{2} \times [1 + \gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi})] + \eta_x\right] < 0\right\} = \text{Prob}\left\{\eta_x < -\frac{\mathbf{A} \cdot \mathbf{T}}{2} \times [1 + \gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi})]\right\} \quad (23)$$

4 - CAPACIDADE ANTI-INTERFERÊNCIA

Na equação (1) o efeito da multiplicação da informação $\mathbf{b}_i(\mathbf{t})$ pelo código $\mathbf{c}_i(\mathbf{t})$ é o de espalhar no espectro a banda ocupada pela informação, por um fator \mathbf{N} . O efeito do modulador é deslocar a banda anteriormente citada para uma faixa de frequências mais altas possibilitando tanto uma transmissão como uma recepção mais eficiente quando se pensa em

uma utilização em telecomunicações (a modulação pode não ser necessária dependendo da aplicação).

A densidade espectral de potência do sinal de dados (binários) aleatórios $\mathbf{b(t)}$ com taxa $\mathbf{R = 1/T}$ bits/segundo é (ver Apêndice C):

$$S_d(f) = A^2 \cdot T \cdot \left(\frac{\sin(\pi \cdot f \cdot T)}{\pi \cdot f \cdot T} \right)^2 \quad (24)$$

Onde: $S_d(0) = A^2 \cdot T$ com $S_d(f) = 0$ em $f = \frac{i}{T}$ em $i = \pm 1, \pm 2, \pm 3, \pm 4, \dots$

A densidade espectral de potência do sinal acima, após ser multiplicado por um código binário $\mathbf{c(t)}$ de comprimento \mathbf{N} com amplitudes no conjunto $\{-1, +1\}$ e onde $\mathbf{T = N \cdot T_c}$, é dada por:

$$S_{ss}(f) = A^2 \cdot T_c \cdot \left(\frac{\sin(\pi \cdot f \cdot T_c)}{\pi \cdot f \cdot T_c} \right)^2 \quad (25)$$

Na Figura 4 ilustra-se o espalhamento em termos de densidade espectral de potência que ocorre para o caso $\mathbf{T/T_c = 2}$. Nesta figura o eixo das abcissas encontra-se normalizado em relação à frequência dos dados (foi dividido pelo fator $\mathbf{1/T}$) e o eixo das ordenadas encontra-se também normalizado tendo sido dividido por $\mathbf{A^2 \cdot T}$.

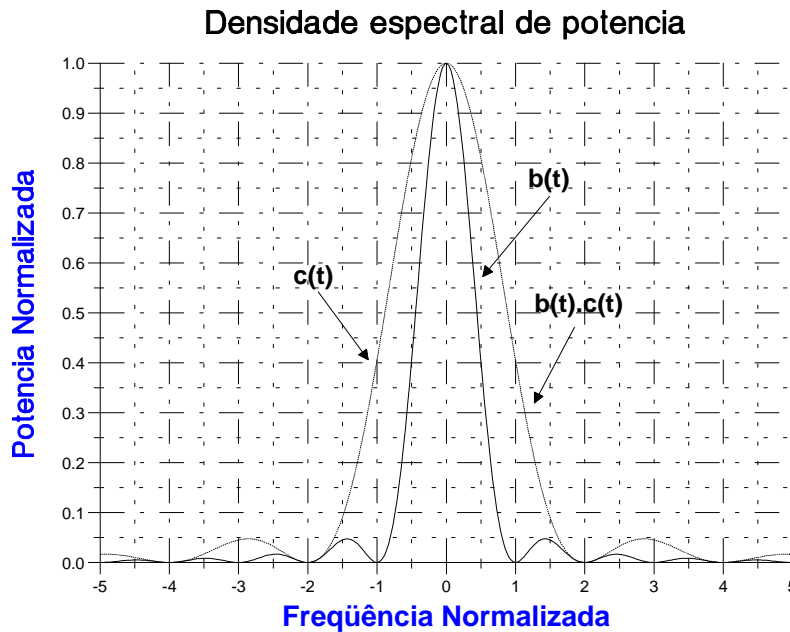


Figura 4

Seja agora este sinal espalhado sofrendo uma interferência proposital (também chamada de jammer) ou a provocada por outro usuário $\mathbf{J(t)}$. Como o sinal na entrada do receptor

$\mathbf{s}(t) = \mathbf{c}(t) \cdot \mathbf{b}(t) + \mathbf{J}(t)$ é multiplicado pelo mesmo código binário $\mathbf{c}(t)$ perfeitamente sincronizado, e neste caso $\mathbf{c}(t)^2 = \mathbf{1}$, resulta $\mathbf{r}(t) = \mathbf{b}(t) + \mathbf{c}(t) \cdot \mathbf{J}(t)$, onde o primeiro termo é a informação que deseja-se receber, que pode ser recuperada com um filtro passa-baixas; o segundo termo é a interferência espalhada no espectro. A fração da potência de interferência que passa pelo filtro passa baixas é igual ao inverso do ganho de processamento $G_p = T/T_c = N$.

Utilizando a representação geométrica de sinais, isto é, representando-se cada um dos sinais transmitidos através de um vetor n-dimensional, que é um ponto num espaço n-dimensional, resulta:

$$\mathbf{s}_i(t) = \sum_{j=1}^n s_{ij} \cdot \phi_j(t); \quad 1 \leq i \leq K; \quad 0 \leq t \leq T; \quad (26)$$

Onde: n é a dimensão do espaço.

K é o número de usuários transmitindo.

$\phi_j(t)$ com $1 \leq k \leq n$ é um conjunto de funções ortonormais que satisfazem:

$$\int_0^T \phi_l(t) \cdot \phi_m(t) dt = \begin{cases} 1 & l = m \\ 0 & l \neq m \end{cases} \quad (27)$$

Os coeficientes s_{ik} podem ser obtidos por:

$$s_{ik} = \int_0^T s_i(t) \cdot \phi_k(t) dt \quad (28)$$

$$\text{e a energia de cada sinal é } E_{s_i} = \int_0^T s_i(t)^2 dt = \sum_{k=1}^n s_{ik}^2 \quad (29)$$

Considere-se agora uma interferência adicionada ao sinal:

$$\mathbf{J}(t) = \sum_{k=1}^n \mathbf{J}_k \cdot \phi_k(t); \quad 0 \leq t \leq T; \quad (30)$$

As componentes fora desta representação não interferem no desempenho, ref. [23]. A energia efetiva (aquela representável no espaço n-dimensional) da interferência é:

$$E_j = \sum_{k=1}^n J_k^2 \quad (31)$$

O sinal recebido $\mathbf{r}(t) = \mathbf{s}_i(t) + \mathbf{J}(t)$ é correlacionado com o sinal conhecido $\mathbf{s}_i(t)$ de modo que a saída U_i no i-ésimo correlator é:

$$U_i = \int_0^T \mathbf{r}(\mathbf{t}) \cdot \mathbf{s}_i(\mathbf{t}) \cdot d\mathbf{t} = \sum_{k=1}^n (\mathbf{s}_{ik}^2 + \mathbf{J}_k \cdot \mathbf{s}_{ik}) \quad (32)$$

Da equação (32) verifica-se que um dos objetivos de uma interferência intencional é projetar uma estratégia para selecionar as componentes \mathbf{J}_k , supondo que sua energia \mathbf{E}_j é fixa, de modo a minimizar a relação sinal-ruído na saída do receptor.

5 - ABORDAGEM DE CÁLCULO DA PROBABILIDADE DE ERRO DE BIT NO PIOR CASO E MELHOR CASO

Para a abordagem do melhor e pior caso, ref. [6], retoma-se a equação (23) que devido a simetria da função densidade de probabilidade gaussiana pode ser reescrita na forma:

$$\mathbf{P}_e = \text{Prob}\left\{ \eta_x > \frac{\mathbf{A} \cdot \mathbf{T}}{2} \times [1 + \gamma_i(\mathbf{b}, \underline{\tau}, \underline{\phi})] \right\} \quad (33)$$

$$\mathbf{P}_e = \mathbf{Q}\left\{ \sqrt{\frac{\mathbf{A}^2 \cdot \mathbf{T}}{\mathbf{N}_0}} \times [1 + \gamma_i(\mathbf{b}, \underline{\tau}, \underline{\phi})] \right\} \quad (34)$$

$$\text{Onde: } \mathbf{Q}(\mathbf{x}) = \int_x^\infty \frac{1}{\sqrt{2 \cdot \pi}} \times \exp\left(-\frac{y^2}{2}\right) \cdot dy \quad (35)$$

O melhor e o pior caso são casos limites que consistem em minimizar e maximizar $\gamma_i(\mathbf{b}, \underline{\tau}, \underline{\phi})$ respectivamente, uma vez que a a função $\mathbf{Q}(\mathbf{x})$ é monotonicamente decrescente na expressão (35).

Da simetria da função $\gamma_i(\mathbf{b}, \underline{\tau}, \underline{\phi})$, calcular o pior caso consiste em:

$$\min[\gamma_i(\mathbf{b}, \underline{\tau}, \underline{\phi})] = -\max[\gamma_i(\mathbf{b}, \underline{\tau}, \underline{\phi})] = -\max[|\gamma_i(\mathbf{b}, \underline{\tau}, \underline{\phi})|] = -\sum_{\substack{i=1 \\ i \neq x}}^K \max[|\mathbf{I}_{i,x}(\mathbf{b}_i, \tau_i, \phi_i)|] \quad (36)$$

Portanto, maximizando os módulos de (17):

a) Em relação a ϕ_i :

Da característica da função $\cos(\phi_i)$ sabe-se que $\max[\cos(\phi_i)] = 1$ e $\min[\cos(\phi_i)] = -1$, permitindo escrever-se:

$$-T^{-1} \cdot \max\left[\left|\mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i)\right|\right] < \mathbf{I}_{i,x}(\mathbf{b}_i, \tau, \phi) < T^{-1} \cdot \max\left[\left|\mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i)\right|\right] \quad (37)$$

b) Em relação a $\mathbf{b}_{-1}^{(i)}$ e $\mathbf{b}_0^{(i)}$:

Definindo-se as funções de correlação cruzada par (e ímpar, respectivamente) contínuas no tempo por:

$$\mathfrak{R}_{i,x}(\tau_i) = \mathbf{R}_{i,x}(\tau_i) + \hat{\mathbf{R}}_{i,x}(\tau_i) = \int_0^T \mathbf{c}_i(t - \tau_i) \cdot \mathbf{c}_x(t) \cdot dt \quad (38)$$

$$\hat{\mathfrak{R}}_{i,x}(\tau_i) = \mathbf{R}_{i,x}(\tau_i) - \hat{\mathbf{R}}_{i,x}(\tau_i) \quad (39)$$

E levando em conta as quatro combinações possíveis de $\mathbf{b}_{-1}^{(i)}$ e $\mathbf{b}_0^{(i)}$ obtém-se:

$$\begin{aligned} \mathbf{T}^{-1} \cdot \max[|\mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i)|] &= \mathbf{T}^{-1} \cdot \max[|\mathfrak{R}_{i,x}(\tau_i)|, |\hat{\mathfrak{R}}_{i,x}(\tau_i)|] \\ &= \mathbf{T}^{-1} \cdot \max[|\mathbf{R}_{i,x}(\tau_i)| + |\hat{\mathbf{R}}_{i,x}(\tau_i)|] \end{aligned} \quad (40)$$

Finalmente:

$$-\mathbf{T}^{-1} \cdot \max[|\mathbf{R}_{i,x}(\tau_i)| + |\hat{\mathbf{R}}_{i,x}(\tau_i)|] < \mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) < \mathbf{T}^{-1} \cdot \max[|\mathbf{R}_{i,x}(\tau_i)| + |\hat{\mathbf{R}}_{i,x}(\tau_i)|] \quad (41)$$

c) Em relação a τ_i :

Demonstra-se no Apêndice A que os máximos locais e os mínimos locais de $|\mathbf{R}_{i,x}(\tau_i)| + |\hat{\mathbf{R}}_{i,x}(\tau_i)|$ ocorrem em valores de τ_i que são múltiplos inteiros de \mathbf{T}_c .

Portanto, chamando-se de $\xi_{i,x}$ o maior valor dentre todos os resultados de $|\mathbf{R}_{i,x}(\tau_i)| + |\hat{\mathbf{R}}_{i,x}(\tau_i)|$ calculados em múltiplos inteiros de \mathbf{T}_c , isto é, para $\tau_i = 0 \cdot \mathbf{T}_c, 1 \cdot \mathbf{T}_c, 2 \cdot \mathbf{T}_c, \dots, (\mathbf{N} - 1) \cdot \mathbf{T}_c$, a interferência de múltiplo acesso normalizada do i -ésimo usuário pode ser limitada superior e inferiormente por:

$$-\mathbf{T}^{-1} \cdot \xi_{i,x} < \mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) < \mathbf{T}^{-1} \cdot \xi_{i,x} \quad (42)$$

Substituindo-se os limites obtidos em (42) na equação (34) resulta para o cálculo da probabilidade de erro de bit no pior caso:

$$\mathbf{P}_e = \mathbf{Q}\left\{\sqrt{\frac{\mathbf{A}^2 \cdot \mathbf{T}}{\mathbf{N}_0}} \times [1 - \mathbf{T}^{-1} \cdot \sum_{\substack{i=1 \\ i \neq x}}^K \xi_{i,x}]\right\} \quad (43)$$

E para o cálculo da probabilidade de erro de bit no melhor caso:

$$P_e = Q\left\{\sqrt{\frac{A^2 \cdot T}{N_0}} \times [1 + T^{-1} \cdot \sum_{\substack{i=1 \\ i \neq x}}^K \xi_{i,x}]\right\} \quad (44)$$

Esta abordagem é importante, porque a obtenção de limites superiores e inferiores requer um esforço de cálculo computacional muito menor quanto comparado à outros métodos de cálculo de probabilidade de erro de bit, porém, os valores obtidos ocorrem pouco freqüentemente em casos práticos, daí a necessidade de outras abordagens, utilizando métodos estatísticos para se calcular um valor médio da probabilidade de erro de bit ou próximo desta.

6 - ABORDAGEM SEGUNDO PURSLEY

A abordagem seguida por Pursley, ref. [5], emprega métodos estatísticos e consiste em calcular **exatamente** a variância da interferência multi-usuário e admitir que esta mesma interferência tem uma distribuição gaussiana (que é a aproximação realizada nesta abordagem). Com o objetivo de obter-se uma probabilidade de erro de bit média admite-se que ϕ_i e τ_i são variáveis aleatórias uniformemente distribuídas no intervalos $[0, 2\pi]$ e $[0, T]$ respectivamente, além de $\mathbf{b}_{-1}^{(i)}$ e $\mathbf{b}_0^{(i)}$ serem duas variáveis aleatórias equiprováveis, podendo assumir os valores em $\{-1, +1\}$:

$$\text{var}[Z_x] = \text{var}[\text{Inf}_x] + \text{var}[\text{Interf}_x] + \text{var}[\eta_x] \quad (45)$$

$\text{var}[\text{Inf}_x] = 0$, porque Inf_x é um evento certo.

$\text{var}[\eta_x] = \frac{N_0 \cdot T}{4}$, conforme demonstrado no Apêndice D.

$$\text{var}[\text{Interf}_x] = \text{var}\left[\frac{A}{2} \times \sum_{\substack{i=1 \\ i \neq x}}^K [\mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i)] \cdot \cos(\phi_i)\right] = \text{var}\left[\sum_{\substack{i=1 \\ i \neq x}}^K \alpha_i \cdot \cos(\phi_i)\right]$$

Calculando a variância em relação a ϕ_i :

$$\text{var}[\text{Interf}_x] = E[\text{Interf}_x^2] - E[\text{Interf}_x]^2$$

$$\begin{aligned} \text{var}[\text{Interf}_x] &= \frac{1}{2\pi} \times \int_0^{2\pi} \left(\sum_{\substack{i=1 \\ i \neq x}}^K \alpha_i \cdot \cos(\phi_i)\right)^2 \cdot d\phi_i = \left\{ \frac{1}{2\pi} \times \int_0^{2\pi} \sum_{\substack{i=1 \\ i \neq x}}^K \alpha_i \cdot \cos(\phi_i) \cdot d\phi_i \right\}^2 \\ \text{var}[\text{Interf}_x] &= \frac{1}{2\pi} \times \sum_{\substack{i=1 \\ i \neq x}}^K \left\{ \alpha_i^2 \cdot \int_0^{2\pi} \cos^2(\phi_i) \cdot d\phi_i + \sum_{\substack{h=1 \\ h \neq i \\ h \neq x}}^K \alpha_h \cdot \cos(\phi_h) \cdot \int_0^{2\pi} \alpha_i \cdot \cos(\phi_i) \cdot d\phi_i \right\} \\ &\quad - \frac{1}{4\pi^2} \times \left\{ \sum_{\substack{i=1 \\ i \neq x}}^K \alpha_i \cdot \int_0^{2\pi} \cos(\phi_i) \cdot d\phi_i \right\}^2 \end{aligned}$$

Como os dois últimos termos são nulos devido à integração ser no período $[0, 2\pi]$:

$$\text{var}[\text{Interf}_x] = \frac{1}{2\pi} \times \sum_{\substack{i=1 \\ i \neq x}}^K \alpha_i^2 \cdot \int_0^{2\pi} \cos(\phi_i)^2 \cdot d\phi_i = \frac{1}{2\pi} \times \sum_{\substack{i=1 \\ i \neq x}}^K \alpha_i^2 \cdot \int_0^{2\pi} \left(\frac{1}{2} + \frac{1}{2} \times \cos(2\phi_i) \right) \cdot d\phi_i = \sum_{\substack{i=1 \\ i \neq x}}^K \frac{\alpha_i^2}{2}$$

Calculando a esperança de $\text{var}[\text{Interf}_x]$ em relação a $\mathbf{b}_0^{(i)}$ e $\mathbf{b}_{-1}^{(i)}$:

$$\begin{aligned} E\{\text{var}[\text{Interf}_x]\} &= \frac{A^2}{32} \times \left\{ \sum_{\substack{i=1 \\ i \neq x}}^K [\mathbf{R}_{i,x}(\tau_i) + \hat{\mathbf{R}}_{i,x}(\tau_i)]^2 \right\} + \frac{A^2}{32} \times \left\{ \sum_{\substack{i=1 \\ i \neq x}}^K [\mathbf{R}_{i,x}(\tau_i) - \hat{\mathbf{R}}_{i,x}(\tau_i)]^2 \right\} + \\ &\quad \frac{A^2}{32} \times \left\{ \sum_{\substack{i=1 \\ i \neq x}}^K [-\mathbf{R}_{i,x}(\tau_i) + \hat{\mathbf{R}}_{i,x}(\tau_i)]^2 \right\} + \frac{A^2}{32} \times \left\{ \sum_{\substack{i=1 \\ i \neq x}}^K [-\mathbf{R}_{i,x}(\tau_i) - \hat{\mathbf{R}}_{i,x}(\tau_i)]^2 \right\} \end{aligned}$$

$$E\{\text{var}[\text{Interf}_x]\} = \frac{A^2}{8} \times \sum_{\substack{i=1 \\ i \neq x}}^K [\mathbf{R}_{i,x}(\tau_i)^2 + \hat{\mathbf{R}}_{i,x}(\tau_i)^2]$$

Calculando agora a esperança em relação a τ_i , supondo que o mesmo é uma variável aleatória uniformemente distribuída no intervalo $[0, T]$:

$$E\{E\{\text{var}[\text{Interf}_x]\}\} = \frac{A^2}{8 \cdot T} \times \sum_{\substack{i=1 \\ i \neq x}}^K \sum_{m=0}^{N-1} \int_{m \cdot T_c}^{(m+1) \cdot T_c} (\mathbf{R}_{i,x}(\tau_i)^2 + \hat{\mathbf{R}}_{i,x}(\tau_i)^2) \cdot d\tau_i \quad (46)$$

Onde conforme demonstração do Apêndice A:

$$\hat{\mathbf{R}}_{i,x}(\tau) = \mathbf{C}_{i,x}(\mathbf{l}) \cdot T_c + [\mathbf{C}_{i,x}(\mathbf{l}+1) - \mathbf{C}_{i,x}(\mathbf{l})] \cdot (\tau - \mathbf{l} \cdot T_c) \quad (47)$$

$$\mathbf{R}_{i,x}(\tau) = \mathbf{C}_{i,x}(\mathbf{l}-N) \cdot T_c + [\mathbf{C}_{i,x}(\mathbf{l}-N+1) + \mathbf{C}_{i,x}(\mathbf{l}-N)] \cdot (\tau - \mathbf{l} \cdot T_c) \quad (48)$$

Substituindo as equações (47) e (48) em (46) obtém-se, após simplificações e manipulações algébricas, ref. [2]:

$$\sum_{i=1}^K \sum_{m=0}^{N-1} \int_{m \cdot T_c}^{(m+1) \cdot T_c} (\mathbf{R}_{i,x}(\tau_i)^2 + \hat{\mathbf{R}}_{i,x}(\tau_i)^2) \cdot d\tau_i = \frac{T^3}{3 \cdot N^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x} \quad (49)$$

Substituindo agora (49) em (46):

$$E\{E\{\text{var}[\text{Interf}_x]\}\} = \frac{A^2 \cdot T^2}{24 \cdot N^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x} \quad (50)$$

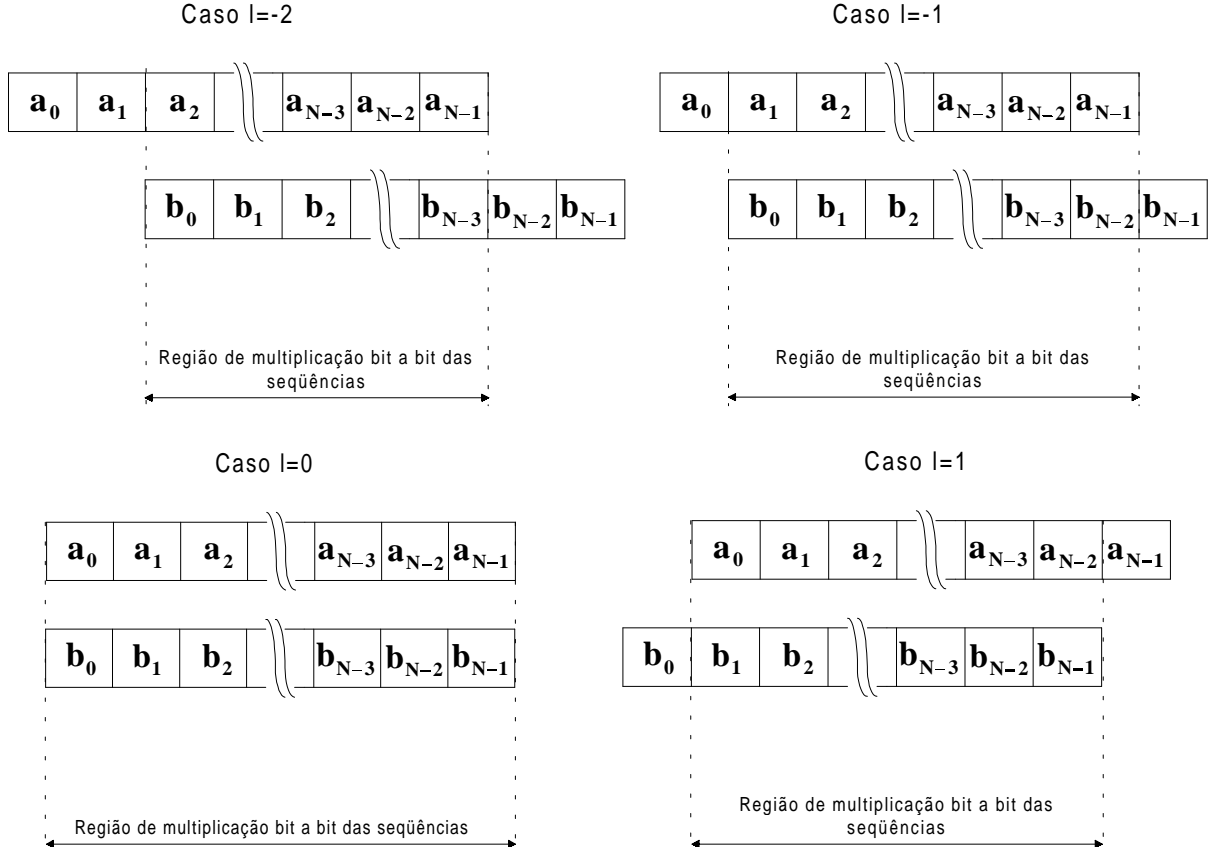
Com:

$$\beta_{i,x} = 2.N^2 + 4. \sum_{l=1}^{N-1} C_i(l).C_x(l) + \sum_{l=1-N}^{N-1} C_i(l).C_x(l+1) \quad (51)$$

Com a notação $C_{a,a}(l) = C_a$, $C_{a,b}(l)$ é conhecida como função de correlação cruzada aperiódica das seqüências e definida por, ver ref. [7]:

$$C_{a,b}(l) = \begin{cases} \sum_{i=0}^{N-1-l} a_i \cdot b_{i+l} & 0 \leq l \leq N-1 \\ \sum_{i=0}^{N-1+l} a_{i-1} \cdot b_i & 1-N \leq l < 0 \\ 0 & |l| \geq N \end{cases} \quad (52)$$

Nos esquemas abaixo verifica-se graficamente qual é o comportamento da função de correlação cruzada aperiódica para valores de $l=-2$, $l=-1$, $l=0$, $l=1$, $l=2$. Nota-se que para valores de $l > 0$ a seqüência **b** desloca-se para a esquerda (em relação a seqüência **a**) e que para valores de $l < 0$ a seqüência **b** desloca-se para a direita (em relação a seqüência **a**). Verifica-se também que a região de multiplicação bit a bit das seqüências diminui a medida que o módulo de l aumenta tornando-se nula quando $|l| \geq N$.



Caso $l=2$

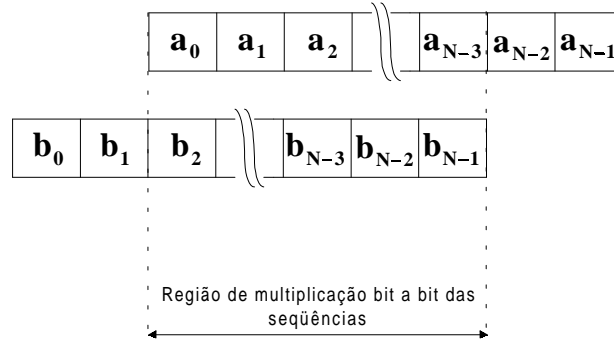


Figura 5 - Comportamento da função $C_{a,b}(l)$ para $l = -2, -1, 0, 1, 2$.

A partir das relações (45) e (50) resulta:

$$\text{var}[Z_x] = \frac{N_0 \cdot T}{4} + \frac{A^2 \cdot T^2}{24 \cdot N^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x}$$

e assim a relação sinal para ruído pode ser escrita na forma:

$$\text{SNR}_x = \frac{\frac{A \cdot T}{2}}{\sqrt{\text{var}[Z_x]}} = \left\{ \frac{1}{6 \cdot N^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x} + \frac{N_0}{A^2 \cdot T} \right\}^{-\frac{1}{2}}$$

Para um número grande de sinais interferentes, ou seqüências de comprimento relativamente grande utilizadas pelo usuários para espalhar os sinais, a distribuição da função densidade de probabilidade da interferência na saída do receptor tende para uma gaussiana, em virtude do teorema do limite central, permitindo calcular a probabilidade de erro de bit por:

$$P_e = Q[\text{SNR}_x] = Q\left[\left\{ \frac{1}{6 \cdot N^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x} + \frac{N_0}{A^2 \cdot T} \right\}^{-\frac{1}{2}}\right] \quad (53)$$

A expressão anterior pode ser simplificada ainda se considerarmos seqüências randômicas de comprimento N , pois, neste caso $\beta_{i,x} = 2 \cdot N^2$, ref. [1], resultando:

$$P_e = Q\left[\left\{ \frac{(K-1)}{3 \cdot N} + \frac{N_0}{A^2 \cdot T} \right\}^{-\frac{1}{2}}\right] \quad (54)$$

Para amplitudes dos sinais interferentes pequenas, isto é com $\mathbf{A} \rightarrow \mathbf{0}$, o termo da interferência multi-usuário tende a ser relativamente pequeno quando comparado ao termo da interferência do ruído branco gaussiano:

$$\frac{N_0}{\mathbf{A}^2 \cdot \mathbf{T}} \gg \frac{1}{6 \cdot \mathbf{N}^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x} \text{ resultando } \mathbf{P}_e = \mathbf{Q}\left[\left\{\frac{N_0}{\mathbf{A}^2 \cdot \mathbf{T}}\right\}^{\frac{1}{2}}\right] \quad (55)$$

Para amplitudes dos sinais interferentes grandes, isto é com $\mathbf{A} \rightarrow \infty$, o termo da interferência do ruído branco gaussiano tende a ser relativamente pequeno quando comparado ao termo da interferência multi-usuário:

$$\frac{N_0}{\mathbf{A}^2 \cdot \mathbf{T}} \ll \frac{1}{6 \cdot \mathbf{N}^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x} \text{ resultando } \mathbf{P}_e = \mathbf{Q}\left[\left\{\frac{1}{6 \cdot \mathbf{N}^3} \times \sum_{\substack{i=1 \\ i \neq x}}^K \beta_{i,x}\right\}^{\frac{1}{2}}\right] \quad (56)$$

A relação sinal para ruído \mathbf{SNR}_x é um parâmetro de performance muito importante pelo significado físico e o método de Pursley é freqüentemente utilizado na prática devido a quantidade relativamente pequena de esforço computacional necessário para o seu cálculo, quando comparado com outros métodos mais precisos de cálculo (por exemplo o de Nazari). Esta relação sinal para ruído é obtida a partir do cálculo da esperança na saída do receptor com respeito aos deslocamentos de fase, atrasos de sinais e a informação transmitida. Tais médias podem também ser interpretadas como médias no tempo, já que na prática estas variáveis variam lentamente no tempo podendo ser modeladas como processos estacionários aleatórios ergódicos, ref. [5].

Nas figuras 6, 7 e 8 tem-se gráficos de performance respectivamente para comprimento de seqüências de assinatura $\mathbf{N} = 31$ e número de usuários $\mathbf{K} = 2, 6$ e 20 ; comprimento de seqüências de assinatura $\mathbf{N} = 127$ e número de usuários $\mathbf{K} = 4, 10$ e 40 e finalmente comprimento de seqüências de assinatura $\mathbf{N} = 511$ e número de usuários $\mathbf{K} = 15$ e 48 . Tem-se, para efeito de comparação, também uma curva de performance do caso clássico de transmissão por modulação BPSK e curvas com o caso de seqüências aleatórias, expressa pela equação (54), que nos gráficos encontram-se representadas por curvas pontilhadas. Verifica-se que a curva correspondente ao caso de seqüências aleatórias se aproxima muito das curvas obtidas pelo método de Pursley chegando a se sobrepor sobre estas nos casos $\mathbf{N} = 31$ e $\mathbf{K} = 20$, $\mathbf{N} = 127$ e $\mathbf{K} = 40$ e $\mathbf{N} = 511$ e $\mathbf{K} = 48$.

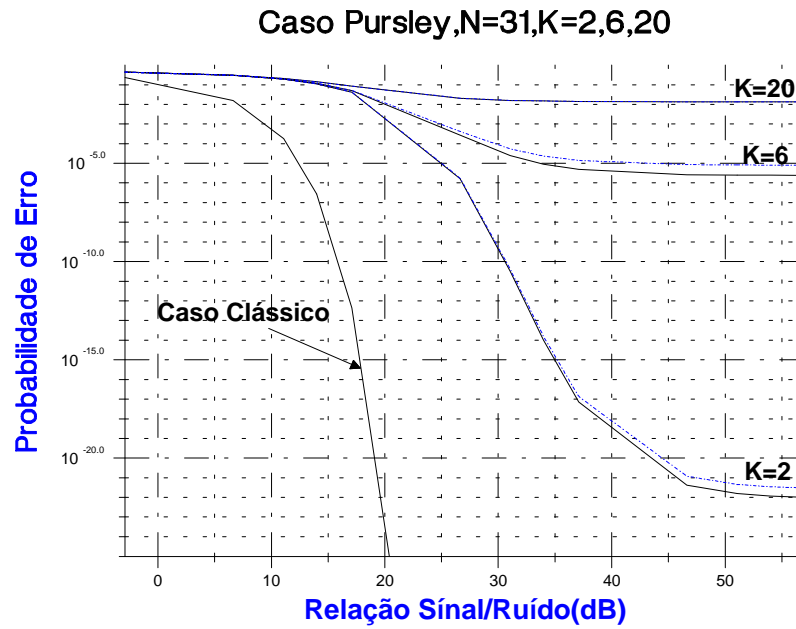


Figura 6

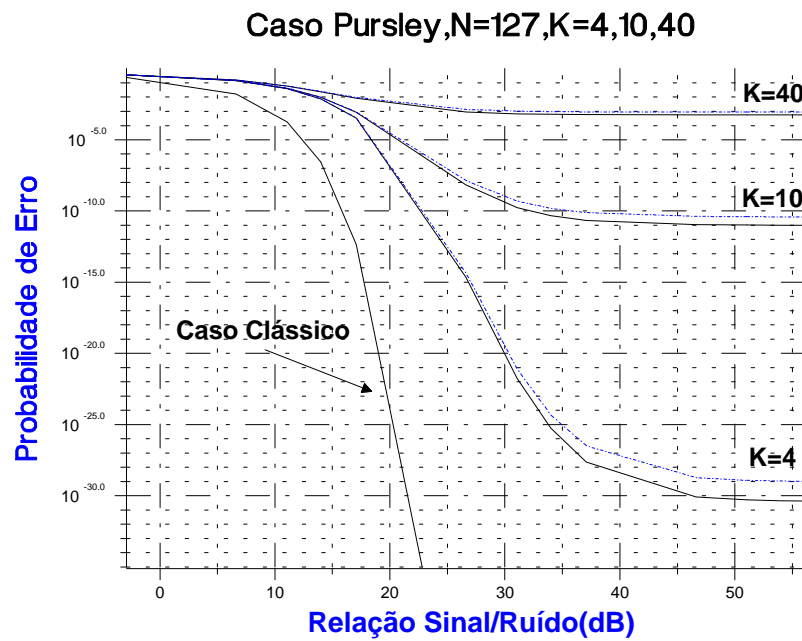


Figura 7

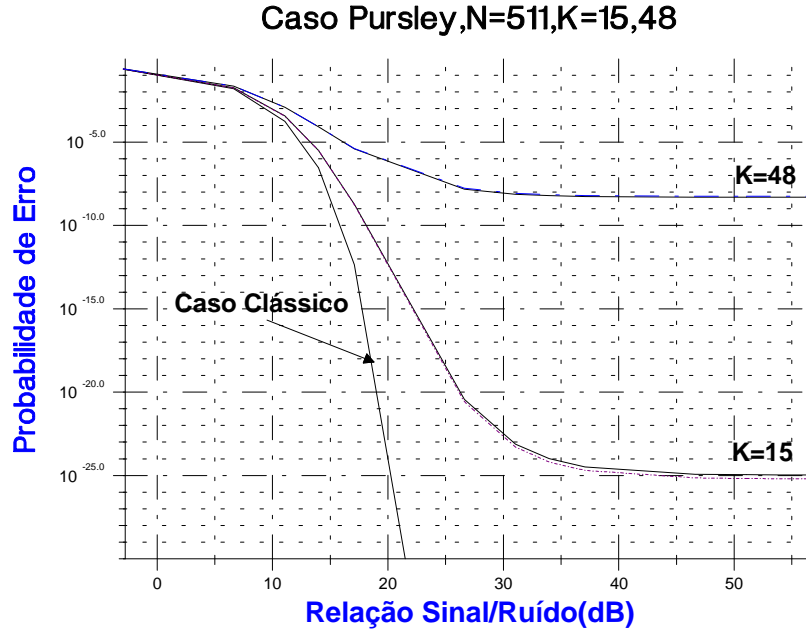


Figura 8

O exame destas figuras permite concluir que a expressão (54), válida para seqüências aleatórias, é uma boa aproximação do resultado de Pursley, dada pela expressão (53), à medida que o número de usuários aumenta.

Observa-se ainda que a probabilidade de erro de bit tende a um valor limite, dado por (56), à medida que a relação sinal para ruído cresce e este valor limite é devido à presença do ruído multiusuário.

7 - ABORDAGEM SEGUNDO WEBER

A interferência de múltiplo acesso $\gamma_i(\mathbf{b}, \tau, \phi)$, para um número grande de usuários e comprimento de seqüências de assinatura grande, pode ser modelada como uma variável aleatória gaussiana. A qualidade desta aproximação foi demonstrada como excelente por Yao, ref. [8], para períodos de código maiores que 7, para a maioria das seqüências pseudo-aleatórias e de Gold.

A abordagem de Weber, ref. [9], modela $\mathbf{I}_{i,x}(\mathbf{b}_i, \tau, \phi)$ também como uma variável aleatória gaussiana. O efeito de um usuário interferente pode então ser modelado como um ruído "broad-band" assumido como aditivo. Admitindo-se que os outros sinais interferentes são mutuamente independentes, suas densidades espectrais de potência se adicionam.

Se a razão $\eta = \mathbf{T} / \mathbf{T}_c = \text{ganho de processamento}$, que indica o espalhamento ocorrido na banda ocupada pelo sinal da informação, é grande os valores de maior significado se encontram na vizinhança da freqüência do sinal da portadora do modulador.

A densidade espectral de potência equivalente, próximo à frequência do sinal da portadora do modulador, na saída do receptor devido ao i -ésimo usuário é:

$$N_i = \frac{P_i}{R_c} = P_i \cdot T_c = \alpha_i \cdot P_o \cdot T_c \quad (57)$$

A relação sinal para ruído na saída do receptor, devida aos $K - 1$ usuários adicionais, é:

$$\left(\frac{E_b}{N_o}\right)_K = \frac{E_b}{N_o + \sum_{i=1}^{K-1} N_i} \quad (58)$$

Da equação acima nota-se que existe uma redução na relação sinal para ruído à medida que aumenta-se o número de usuários. Com a hipótese gaussiana pode-se escrever:

$$P_e = Q \left[\sqrt{2 \cdot \left(\frac{E_b}{N_o}\right)_K} \right] \quad (59)$$

Para manter a mesma probabilidade de erro de bit quando $K - 1$ usuários adicionais estão presentes igual à probabilidade de erro de bit quando um único usuário está presente, é necessário que a relação sinal para ruído de um usuário satisfaça a seguinte relação:

$$\left(\frac{E_b}{N_o}\right)_1 = \left(\frac{E_b}{N_o}\right)_K = \left[\frac{\left(\frac{E_b}{N_o}\right)_R}{1 + \sum_{i=1}^{K-1} \frac{N_i}{N_o}} \right] = \left[\frac{\left(\frac{E_b}{N_o}\right)_R}{1 + \sum_{i=1}^{K-1} \frac{\alpha_i \cdot P_o \cdot T_c}{N_o}} \right] = \left[\frac{\left(\frac{E_b}{N_o}\right)_R}{1 + \frac{T_c}{T_b} \times \left(\frac{E_b}{N_o}\right)_R \times \sum_{i=1}^{K-1} \alpha_i} \right] \quad (60)$$

Portanto, a partir da equação acima:

$$\left(\frac{E_b}{N_o}\right)_R = \frac{\left(\frac{E_b}{N_o}\right)_1}{1 - \eta^{-1} \times \left(\frac{E_b}{N_o}\right)_1 \times \sum_{i=1}^{K-1} \alpha_i} \quad (61)$$

Definindo o fator de degradação **DF** como sendo o fator multiplicativo necessário para $\left(\frac{E_b}{N_o}\right)_1$ se tornar igual a $\left(\frac{E_b}{N_o}\right)_R$:

$$DF = \frac{\left(\frac{E_b}{N_o}\right)_R}{\left(\frac{E_b}{N_o}\right)_1} = \frac{1}{1 - \eta^{-1} \times \left(\frac{E_b}{N_o}\right)_1 \times \sum_{i=1}^{K-1} \alpha_i} \quad (62)$$

Pode-se aplicar as equações obtidas para comentar um resultado interessante que ocorre no caso de todos os usuários transmitirem com potências iguais:

$$\alpha_i = 1 \text{ e } \mathbf{DF} = \frac{1}{1 - \eta^{-1} \times (\mathbf{K} - 1) \times \left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_1}$$

Nota-se que $\left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_R$ tende a $\left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_1$ quando \mathbf{K} tende a 1.

Reescrevendo a equação (58) para o caso em estudo:

$$\left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_K = \left[\frac{\left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_R}{1 + \eta^{-1} \times (\mathbf{K} - 1) \times \left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_R} \right] \quad (63)$$

Nota-se na equação acima que $\lim_{\left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_R \rightarrow \infty} \left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_K = \frac{\eta}{\mathbf{K} - 1}$ que é um limite teórico, pelo modelamento utilizado por Weber, para o máximo valor que $\left(\frac{\mathbf{E}_b}{\mathbf{N}_o}\right)_K$ pode atingir, para potências crescendo sem limite.

Observando que:

$$\mathbf{E}_b = \left(\frac{\mathbf{A}}{2}\right)^2 \cdot \mathbf{T} = \left(\frac{\mathbf{A}}{2}\right)^2 \cdot \mathbf{N} \cdot \mathbf{T}_c \quad (64)$$

$$\mathbf{N}_{o_{\text{equivalente}}} = \mathbf{N}_o + \sum_{\substack{i=1 \\ i \neq x}}^K \mathbf{N}_i = \mathbf{N}_o + \mathbf{T}_c \cdot \sum_{\substack{i=1 \\ i \neq x}}^K \left(\frac{\mathbf{A}_i}{2}\right)^2 \quad (65)$$

Dentro do modelo adotado, a equação (59) pode ser reescrita como:

$$\mathbf{Pe} = \mathbf{Q} \left(\sqrt{\frac{2 \cdot \mathbf{E}_b}{\mathbf{N}_{o_{\text{equivalente}}}}} \right) \quad (66)$$

$$\text{Onde: } \mathbf{Q}(x) = \int_x^\infty \frac{1}{\sqrt{2 \cdot \pi}} \times \exp\left(-\frac{y^2}{2}\right) \cdot dy$$

Nas figuras 9, 10 e 11 tem-se gráficos de performance respectivamente para comprimento de seqüências de assinatura $N = 31$ e número de usuários $K = 2, 6$ e 20 ; comprimento de seqüências de assinatura $N = 127$ e número de usuários $K = 4, 10$ e 40 e comprimento de seqüências de assinatura $N = 511$ e número de usuários $K = 15$ e 48 . Tem-se para efeito de comparação também uma curva de performance do caso clássico de transmissão por modulação BPSK.

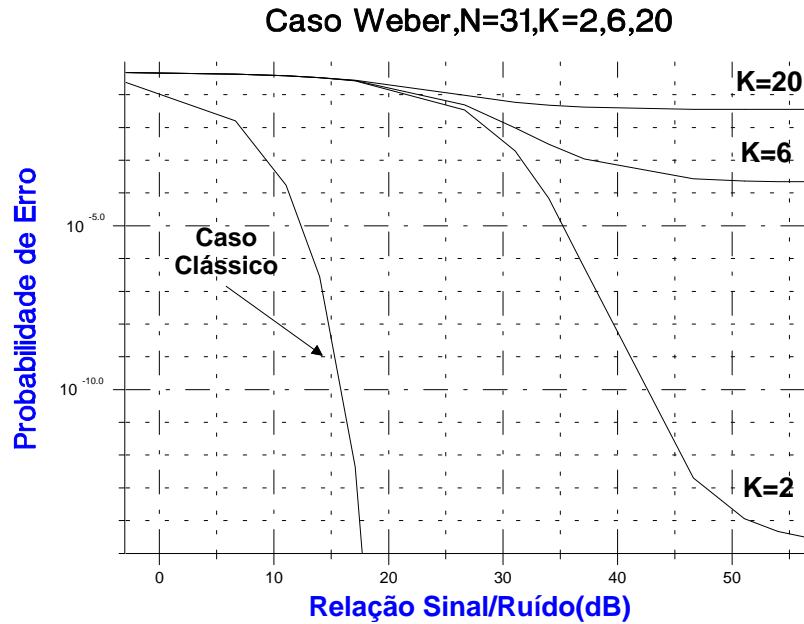


Figura 9

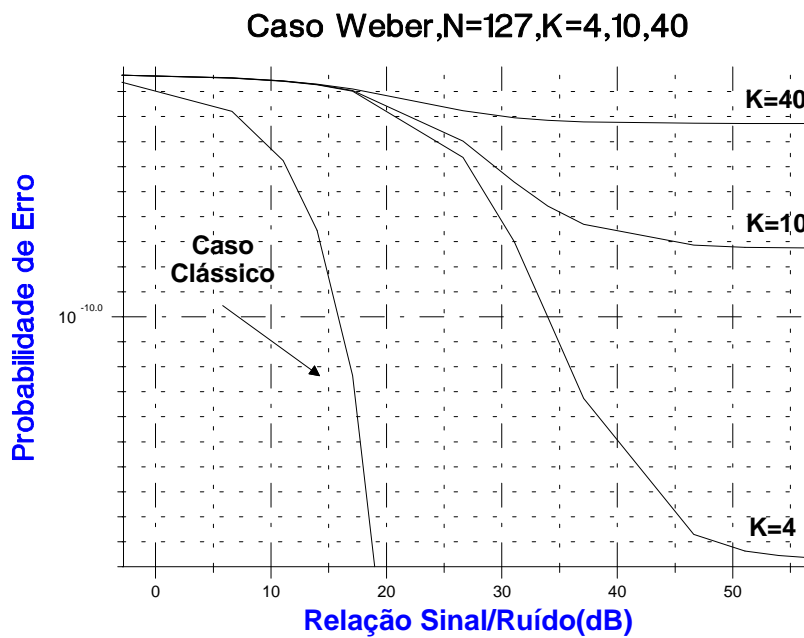


Figura 10

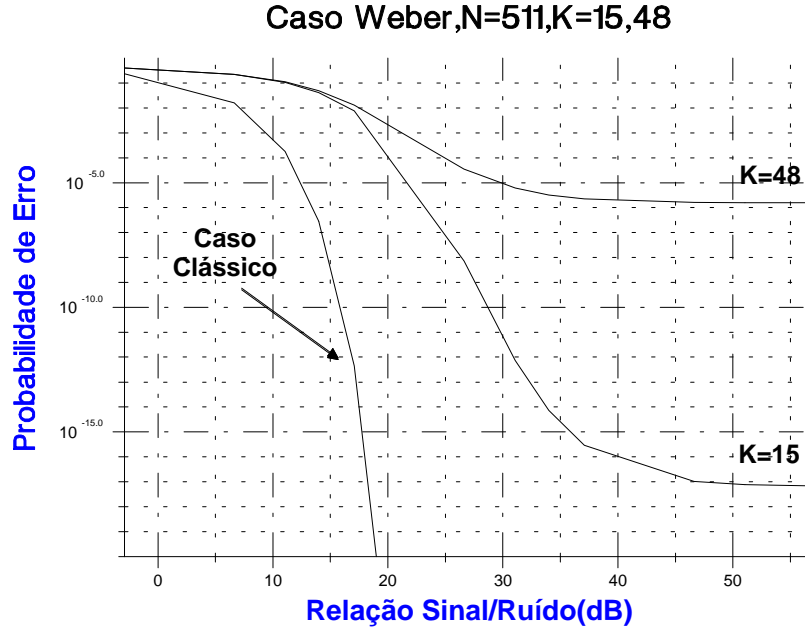


Figura 11

8 - ABORDAGEM SEGUNDO NAZARI E ZIEMER

A abordagem seguida por Nazari e Ziemer, ref. [15], é a da obtenção da função densidade de probabilidade da interferência multi-usuário na saída do correlador. Para tanto deve-se retornar a equação (16) que pode ser reescrita como:

- Se $\mathbf{b}_0^{(x)} = \mathbf{b}_{-1}^{(x)}$:

$$\mathbf{I}_{i,x}(\mathbf{b}_i, \tau, \phi) = \mathbf{T}^{-1} \cdot \mathbf{b}_0^{(x)} \cdot [\mathbf{R}_{i,x}(\tau_i) + \hat{\mathbf{R}}_{i,x}(\tau_i)] \cdot \cos(\phi_i) \quad (67)$$

Definindo-se $\mathfrak{R}_{i,x}(\tau) = \mathbf{R}_{i,x}(\tau_i) + \hat{\mathbf{R}}_{i,x}(\tau_i)$ resulta:

$$\mathbf{I}_{i,x}(\mathbf{b}_i, \tau, \phi) = \mathbf{T}^{-1} \cdot \mathbf{b}_0^{(x)} \cdot \mathfrak{R}_{i,x}(\tau) \cdot \cos(\phi_i) = \mathbf{T}^{-1} \cdot \mathbf{B}_{i,x}(\tau, \phi) \quad (68)$$

Prova-se em Pursley, ref. [6], que para formas de onda retangular do código de espalhamento:

$$\mathfrak{R}_{i,x}(\tau) = (\mathbf{T}_c - \tau + \mathbf{l}_i \cdot \mathbf{T}_c) \cdot \theta_{i,x}(\mathbf{l}_i) + (\tau - \mathbf{l}_i \cdot \mathbf{T}_c) \cdot \theta_{i,x}(\mathbf{l}_i + 1) \quad (69)$$

Com: $\theta_{i,x}(\mathbf{l}) = \mathbf{C}_{i,x}(\mathbf{l}) + \mathbf{C}_{i,x}(\mathbf{l} - \mathbf{N})$

e onde \mathbf{l}_i é a parte inteira de $\frac{\tau_i}{\mathbf{T}_c}$.

Reagrupando:

$$\mathbf{B}_{i,x}(\tau_i) = \mathbf{b}_0^{(x)} \cdot \{\theta_{i,x}(\mathbf{l}_i) \cdot \mathbf{T}_c + [\theta_{i,x}(\mathbf{l}_i + 1) - \theta_{i,x}(\mathbf{l}_i)] \cdot (\tau_i - \mathbf{l}_i \cdot \mathbf{T}_c)\} \cdot \cos(\phi_i) \quad (70)$$

- Se $\mathbf{b}_0^{(x)} \neq \mathbf{b}_{-1}^{(x)}$:

$$\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) = \mathbf{T}^{-1} \cdot \mathbf{b}_0^{(x)} \cdot [\mathbf{R}_{i,x}(\tau_i) - \hat{\mathbf{R}}_{i,x}(\tau_i)] \cdot \cos(\phi_i) \quad (71)$$

Definindo-se $\hat{\mathfrak{R}}_{i,x}(\tau) = \mathbf{R}_{i,x}(\tau_i) - \hat{\mathbf{R}}_{i,x}(\tau_i)$ resulta:

$$\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) = \mathbf{T}^{-1} \cdot \mathbf{b}_0^{(x)} \cdot \hat{\mathfrak{R}}_{i,x}(\tau) \cdot \cos(\phi_i) = \mathbf{T}^{-1} \cdot \mathbf{B}_{i,x}(\tau, \phi) \quad (72)$$

Prova-se ainda em Pursley, ref. [6], que para formas de onda retangular do código de espalhamento:

$$\hat{\mathfrak{R}}_{i,x}(\tau) = (\mathbf{T}_c - \tau + \mathbf{l}_i \cdot \mathbf{T}_c) \cdot \hat{\theta}_{i,x}(\mathbf{l}_i) + (\tau - \mathbf{l}_i \cdot \mathbf{T}_c) \cdot \hat{\theta}_{i,x}(\mathbf{l}_i + 1) \quad (73)$$

Com: $\hat{\theta}_{i,x}(\mathbf{l}) = \mathbf{C}_{i,x}(\mathbf{l}) - \mathbf{C}_{i,x}(\mathbf{l} - \mathbf{N})$

$$\mathbf{B}_{i,x}(\tau_i) = \mathbf{b}_0^{(x)} \cdot \{\hat{\theta}_{i,x}(\mathbf{l}_i) \cdot \mathbf{T}_c + [\hat{\theta}_{i,x}(\mathbf{l}_i + 1) - \hat{\theta}_{i,x}(\mathbf{l}_i)] \cdot (\tau_i - \mathbf{l}_i \cdot \mathbf{T}_c)\} \cdot \cos(\phi_i) \quad (74)$$

Calcula-se em seguida a função densidade de probabilidade supondo $\mathbf{b}_0^{(x)} = \mathbf{b}_{-1}^{(x)}$. Define-se inicialmente como interferência de múltiplo acesso normalizada $\mathbf{m}_{i,x}$:

$$\mathbf{m}_{i,x} = \frac{\mathbf{B}_{i,x}(\tau_i)}{\mathbf{b}_0^{(x)} \cdot \mathbf{T}_c} = \{\theta_{i,x}(\mathbf{l}_i) + [\theta_{i,x}(\mathbf{l}_i + 1) - \theta_{i,x}(\mathbf{l}_i)] \cdot \mathbf{u}\} \cdot \cos(\phi_i) \quad (75)$$

Onde:

$$\mathbf{u} = \frac{\tau_i - \mathbf{l}_i \cdot \mathbf{T}_c}{\mathbf{T}_c} \quad (76)$$

Adotando-se que τ_i e ϕ_i são variáveis aleatórias uniformemente distribuídas no intervalo $[0, \mathbf{T}]$ e $[0, 2\pi]$, respectivamente, conclui-se que a variável aleatória \mathbf{u} encontra-se uniformemente distribuída no intervalo $[0, 1]$.

Calcula-se a seguir a função densidade de probabilidade para os diferentes valores relativos de $\theta_{i,x}(l_i)$ e $\theta_{i,x}(l_i + 1)$.

1º Caso) $\theta_{i,x}(l_i) = \theta_{i,x}(l_i + 1)$

Calculando $\mathbf{m}_{i,x}$:

$$\mathbf{m}_{i,x} = \theta_{i,x}(l_i) \cdot \cos(\phi_i) \quad (77)$$

A função densidade de probabilidade da variável aleatória $\mathbf{m}_{i,x}$ é resultado da multiplicação da constante $\theta_{i,x}(l_i)$ pela função coseno da variável aleatória ϕ_i . Na ref. [18] prova-se que a função densidade de probabilidade de $\mathbf{m}_{i,x}$ é dada por:

$$\mathbf{f}_{\mathbf{m}_{i,x}}(\mathbf{d}) = \frac{1}{\pi \cdot |\theta_{i,x}(l_i)| \cdot [1 - (\frac{\mathbf{d}}{\theta_{i,x}(l_i)})^2]^{0.5}}, \quad |\mathbf{d}| < |\theta_{i,x}(l_i)| \quad (78)$$

$$\mathbf{f}_{\mathbf{m}_{i,x}}(\mathbf{d}) = 0, \quad \text{nas outras regiões} \quad (79)$$

Observe-se que esta função densidade de probabilidade possui singularidades em $|\mathbf{d}| = |\theta_{i,x}(l_i)|$.

2º Caso) $\theta_{i,x}(l_i) \neq \theta_{i,x}(l_i + 1)$

Definindo $\mathbf{m}_{i,x}$:

$$\mathbf{m}_{i,x} = \mathbf{Y} \cdot \cos(\phi_i) \quad (80)$$

Onde:

$$\mathbf{Y} = \theta_{i,x}(l_i) + [\theta_{i,x}(l_i + 1) - \theta_{i,x}(l_i)] \cdot \mathbf{u} \quad (81)$$

Verifica-se que a variável aleatória \mathbf{Y} encontra-se uniformemente distribuída no intervalo $[\mathbf{A}, \mathbf{B}]$ onde \mathbf{A} e \mathbf{B} são:

$$\mathbf{A} = \min[\theta_{i,x}(l_i), \theta_{i,x}(l_i + 1)] \quad (82)$$

$$\mathbf{B} = \max[\theta_{i,x}(l_i), \theta_{i,x}(l_i + 1)] \quad (83)$$

Como a variável aleatória \mathbf{Y} está sendo multiplicada por $\cos(\theta_i)$ pode-se mudar simultaneamente os sinais de \mathbf{A} e \mathbf{B} sem mudar a função densidade de probabilidade de $\mathbf{m}_{i,x}$.

Com esta constatação é possível fazer com que $\mathbf{B} > \mathbf{0}$ e $|\mathbf{B}| > |\mathbf{A}|$. Utilizando-se resultados obtidos na ref. [14]:

$$\mathbf{f}_{\mathbf{m}_{i,x}}(\mathbf{d}) = \frac{1}{\pi \cdot (\mathbf{B} - \mathbf{A})} \times \ln \left| \frac{\mathbf{B} + (\mathbf{B}^2 - \mathbf{d}^2)^{0.5}}{\mathbf{A} + (\mathbf{A}^2 - \mathbf{d}^2)^{0.5}} \right|, \quad |\mathbf{d}| < |\mathbf{A}| \quad (84)$$

$$\mathbf{f}_{\mathbf{m}_{i,x}}(\mathbf{d}) = \frac{1}{\pi \cdot (\mathbf{B} - \mathbf{A})} \times \ln \left| \frac{\mathbf{B} + (\mathbf{B}^2 - \mathbf{d}^2)^{0.5}}{\mathbf{d}} \right|, \quad |\mathbf{A}| < |\mathbf{d}| < \mathbf{B} \quad (85)$$

$$\mathbf{f}_{\mathbf{m}_{i,x}}(\mathbf{d}) = \mathbf{0}, \quad \text{nas outras regiões} \quad (86)$$

Portanto a função densidade de probabilidade da interferência normalizada $\mathbf{p1}_{\mathbf{m}_{i,x}}(\mathbf{d})$ do i-ésimo usuário na saída do receptor preparado para receber o x-ésimo sinal, correspondente ao caso $\mathbf{b}_0^{(x)} = \mathbf{b}_{-1}^{(x)}$, é igual a:

$$\mathbf{p1}_{\mathbf{m}_{i,x}}(\mathbf{d}) = \frac{1}{\mathbf{N}} \times \sum_{j=1}^{\mathbf{N}} \mathbf{f}_{\mathbf{m}_{i,x}/\mathbf{l}_j}(\mathbf{d}/\mathbf{l}_j) \quad (87)$$

Definindo-se $\mathbf{p2}_{\mathbf{m}_{i,x}}(\mathbf{d})$ como a função densidade de probabilidade da interferência normalizada correspondente ao caso $\mathbf{b}_0^{(x)} \neq \mathbf{b}_{-1}^{(x)}$ obtém-se esta substituindo-se $\theta_{i,x}(\dots)$ por $\hat{\theta}_{i,x}(\dots)$ em todas as equações utilizadas para obtenção de $\mathbf{p1}_{\mathbf{m}_{i,x}}(\mathbf{d})$

Considerando o caso usual em que a probabilidade de $\mathbf{b}_0^{(x)} = \mathbf{b}_{-1}^{(x)}$ e $\mathbf{b}_0^{(x)} \neq \mathbf{b}_{-1}^{(x)}$ é **0.5** para os dois casos, a função densidade de probabilidade da interferência normalizada do i-ésimo usuário $\mathbf{p}_{\mathbf{m}_{i,x}}(\mathbf{d})$ será:

$$\mathbf{p}_{\mathbf{m}_{i,x}}(\mathbf{d}) = \frac{1}{2} \times \mathbf{p1}_{\mathbf{m}_{i,x}}(\mathbf{d}) + \frac{1}{2} \times \mathbf{p2}_{\mathbf{m}_{i,x}}(\mathbf{d}) \quad (88)$$

A partir da equação (75) e sabendo-se que $\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) = \mathbf{T}^{-1} \cdot \mathbf{B}_{i,x}(\tau, \phi)$ podem ser estabelecidas as seguintes relações entre as funções densidade de probabilidade da interferência normalizada e desnormalizada do i-ésimo usuário:

$$\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) = \mathbf{b}_0^{(x)} \cdot \mathbf{T}^{-1} \cdot \mathbf{T}_c \cdot \mathbf{m}_{i,x} \quad (89)$$

Assumindo-se sem perda de generalidade $\mathbf{b}_0^{(x)} = \mathbf{1}$ resulta:

$$\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) = \frac{1}{\mathbf{N}} \cdot \mathbf{m}_{i,x} \quad (90)$$

Sabendo-se que multiplicando a variável aleatória \mathbf{x} , com função densidade de probabilidade $\mathbf{f}_x(\mathbf{x})$, por uma constante \mathbf{a} , resultando a variável aleatória \mathbf{y} , isto é, $\mathbf{y} = \mathbf{a} \cdot \mathbf{x}$, corresponde em termos de função densidade de probabilidade a $\mathbf{f}_y(\mathbf{y}) = \frac{1}{|\mathbf{a}|} \times \mathbf{f}_x(\frac{\mathbf{y}}{\mathbf{a}})$, tem-se :

$$\mathbf{f}_{I_{i,x}}(\underline{\mathbf{b}}_i, \underline{\tau}, \underline{\phi}) = \mathbf{N} \cdot \mathbf{f}_{m_{i,x}}(\mathbf{N} \cdot \mathbf{d}') \quad (91)$$

Com a seguinte transformação nas abcissas:

$$\mathbf{d} = \mathbf{N} \cdot \mathbf{d}' \longrightarrow \mathbf{d}' = \frac{1}{\mathbf{N}} \times \mathbf{d} \quad (92)$$

Reescrevendo a equação (18):

$$\gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi}) = \sum_{\substack{i=1 \\ i \neq x}}^K I_{i,x}(\underline{\mathbf{b}}_i, \underline{\tau}_i, \underline{\phi}_i) \quad (93)$$

Observa-se que a função densidade de probabilidade normalizada da interferência multi-usuário pode ser calculada, devido à independência entre as seqüências, através da convolução (representado pelo operador \otimes) entre as funções densidade de probabilidade das interferências de cada usuário:

$$\mathbf{f}_{\gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi})} = \bigotimes_{\substack{i=1 \\ i \neq x}}^K \mathbf{f}_{m_{i,x}} \quad (94)$$

Expandindo a equação (20) resulta:

$$\mathbf{Z}_x = \frac{\mathbf{A} \cdot \mathbf{T}}{2} + \frac{\mathbf{A} \cdot \mathbf{T}}{2} \times \gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi}) + \eta_x \quad (95)$$

Calcula-se então a probabilidade de erro de bit através da equação:

$$\mathbf{P}_e = \text{Pr obabilidade} \left(\left[\frac{\mathbf{A} \cdot \mathbf{T}}{2} \times \gamma_i(\underline{\mathbf{b}}, \underline{\tau}, \underline{\phi}) + \eta_x \right] < -\frac{\mathbf{A} \cdot \mathbf{T}}{2} \right) \quad (96)$$

Nas figuras 12, 13 e 14 tem-se gráficos de performance respectivamente para comprimento de seqüências de assinatura $\mathbf{N} = 31$ e número de usuários $\mathbf{K} = 2,6$ e 20 ; comprimento de seqüências de assinatura $\mathbf{N} = 127$ e número de usuários $\mathbf{K} = 4,10$ e 40 e finalmente comprimento de seqüências de assinatura $\mathbf{N} = 511$ e número de usuários $\mathbf{K} = 15$ e 48 . Tem-se ainda para efeito de comparação a curva de performance do caso clássico de transmissão com modulação BPSK.

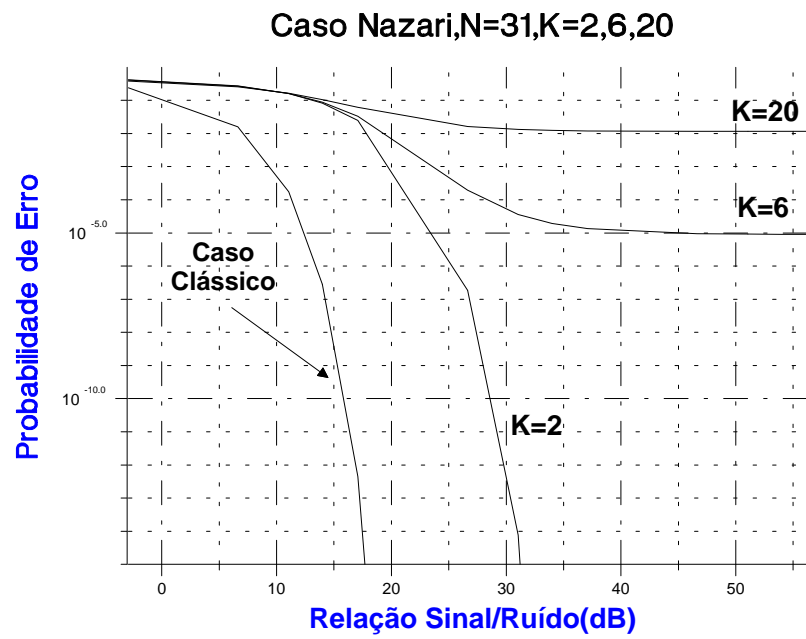


Figura 12

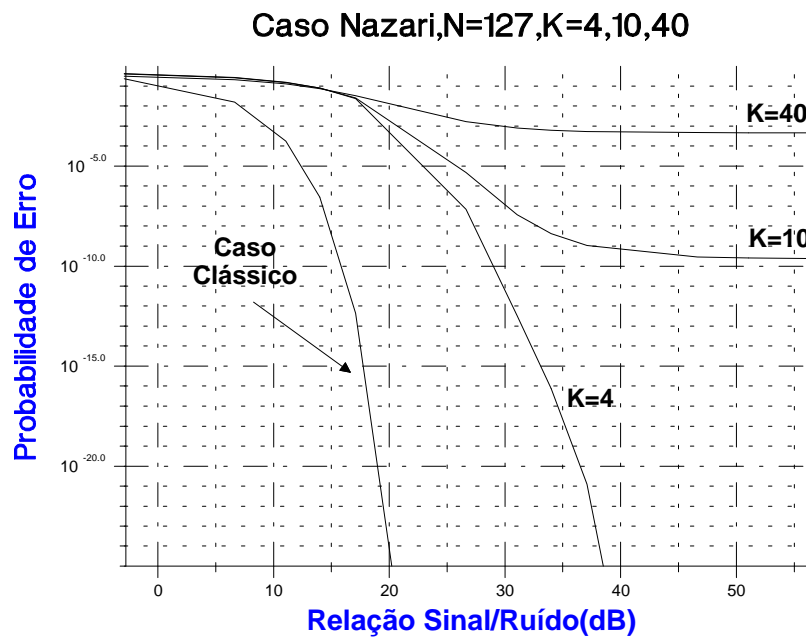


Figura 13

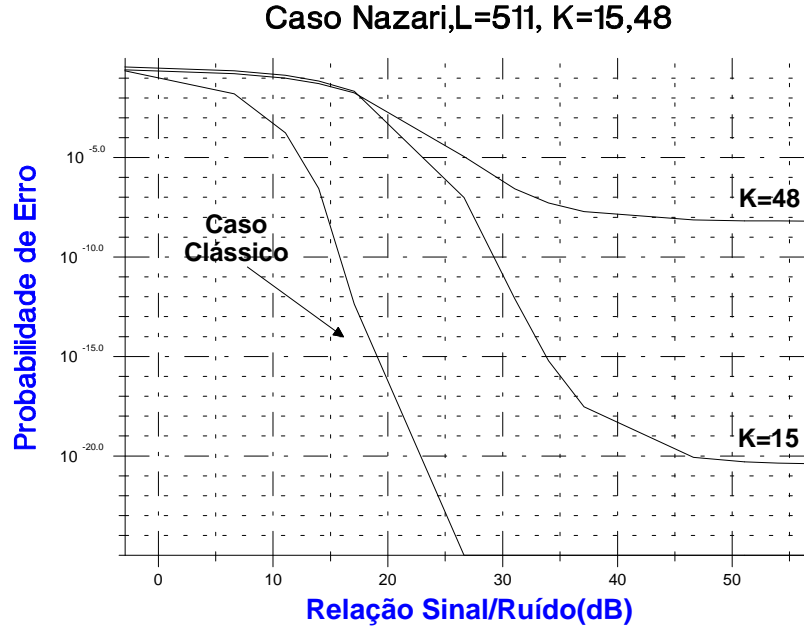


Figura 14

9 - SISTEMAS QUASE-SÍNCRONOS (QS-CDMA)

Define-se um sistema CDMA quase-síncrono como um sistema convencional CDMA onde consegue-se, através do envio de um sinal de sincronismo adicional (chamado de Master Code), manter os sinais transmitidos por todos os usuários alinhados em termos de assinaturas (portanto também de dados) dentro de determinada faixa de atrasos τ_i entre si (ver Figura 15). Considera-se nesta abordagem este atraso como estando na faixa de valores $-\gamma \cdot T_c \leq \tau_i \leq \gamma \cdot T_c$ onde $|\gamma| \leq 1$.

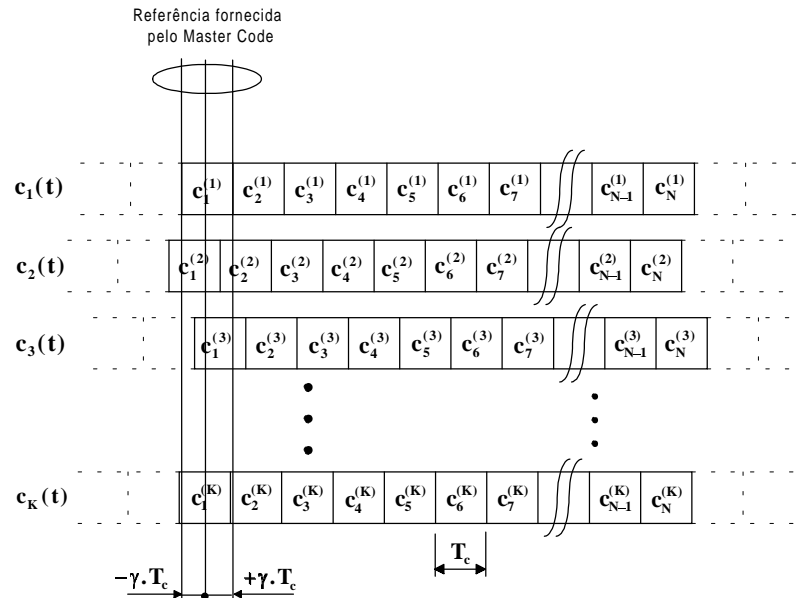


Figura 15 - Modelo de sincronismo entre seqüências de assinatura em QS-CDMA

Admite-se que a variável aleatória τ_i possui função densidade de probabilidade dada pela Figura 16, pois deve-se considerar apenas os valores de $\tau_i > 0$ na abordagem anterior.

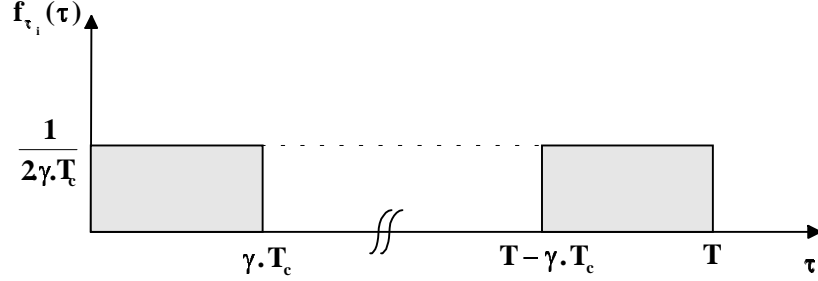


Figura 16 - Função densidade de probabilidade de τ_i

Continuando o raciocínio da abordagem de Pursley a partir da equação (46) obtemos:

$$\mathbf{E}\{\mathbf{E}\{\text{var}[\text{Interf}_x]\}\} = \frac{A^2}{16 \cdot \gamma \cdot T_c} \times \sum_{\substack{i=1 \\ i \neq x}}^K \left[\int_0^{\gamma \cdot T_c} (\mathbf{R}_{i,x}(\tau_i)^2 + \hat{\mathbf{R}}_{i,x}(\tau_i)^2) \cdot d\tau_i + \int_{T-\gamma \cdot T_c}^T (\mathbf{R}_{i,x}(\tau_i)^2 + \hat{\mathbf{R}}_{i,x}(\tau_i)^2) \cdot d\tau_i \right] \quad (97)$$

As equações (47) e (48) para este caso simplificam-se resultando:

$$\mathbf{R}_{i,x}(\tau) = \begin{cases} T_c \cdot C_{i,x}(-L) + [C_{i,x}(1-L) - C_{i,x}(-L)] \cdot \tau & \text{para } 0 \leq \tau \leq \gamma \cdot T_c \\ T_c \cdot C_{i,x}(-1) + [C_{i,x}(0) - C_{i,x}(-1)] \cdot [\tau - (L-1) \cdot T_c] & \text{para } T - \gamma \cdot T_c \leq \tau \leq T \end{cases} \quad (98)$$

$$\hat{\mathbf{R}}_{i,x}(\tau) = \begin{cases} T_c \cdot C_{i,x}(0) + [C_{i,x}(1) - C_{i,x}(0)] \cdot \tau & \text{para } 0 \leq \tau \leq \gamma \cdot T_c \\ T_c \cdot C_{i,x}(L-1) + [C_{i,x}(L) - C_{i,x}(L-1)] \cdot [\tau - (L-1) \cdot T_c] & \text{para } T - \gamma \cdot T_c \leq \tau \leq T \end{cases} \quad (99)$$

Substituindo as equações (98) e (99) em (97) obtém-se após simplificações e manipulações algébricas, ref. [24], a variância da interferência multi-usuário:

$$\mathbf{E}\{\mathbf{E}\{\text{var}[\text{Interf}_x]\}\} = \frac{A^2 \cdot T^2}{16 \cdot N^2} \times \sum_{\substack{i=1 \\ i \neq x}}^K \rho_{i,x} \quad (100)$$

Onde:

$$\begin{aligned} \rho_{i,x} = & \left(\frac{2 \cdot \gamma^2}{3} - 2 \cdot \gamma + 2 \right) \cdot C_{i,x}(0)^2 - \left(\frac{2 \cdot \gamma^2}{3} - \gamma \right) \cdot C_{i,x}(0) \cdot [C_{i,x}(1) + C_{i,x}(-1)] \\ & + \frac{\gamma^2}{3} \times (C_{i,x}(1)^2 + C_{i,x}(-1)^2 + C_{i,x}(1-L)^2 + C_{i,x}(L-1)^2) \end{aligned} \quad (101)$$

A partir de (45), calcula-se a relação sinal para ruído SNR_x na saída do receptor como:

$$\text{SNR}_x = \frac{\frac{A \cdot T}{2}}{\sqrt{\text{var}[Z_x]}} = \left\{ \frac{1}{4 \cdot N^2} \times \sum_{\substack{i=1 \\ i \neq x}}^K \rho_{i,x} + \frac{N_0}{A^2 \cdot T} \right\}^{-\frac{1}{2}} \quad (102)$$

Para um número grande de sinais interferentes, ou seqüências de comprimento relativamente grande, utilizadas pelo usuários para espalhar os sinais, a distribuição da interferência na saída do receptor tende para uma gaussiana, permitindo escrever a probabilidade de erro de bit final da informação como:

$$P_e = Q[\text{SNR}_x] = Q\left[\left\{ \frac{1}{4 \cdot N^2} \times \sum_{\substack{i=1 \\ i \neq x}}^K \rho_{i,x} + \frac{N_0}{A^2 \cdot T} \right\}^{-\frac{1}{2}}\right] \quad (103)$$

Nas figuras 17, 18 e 19 tem-se gráficos de performance respectivamente para comprimento de seqüências de assinatura $N=31$ e número de usuários $K=2,6$ e 20 ; comprimento de seqüências de assinatura $N=127$ e número de usuários $K=4,10$ e 40 e finalmente comprimento de seqüências de assinatura $N=511$ e número de usuários $K=15$ e 48 . Tem-se para efeito de comparação também uma curva de performance do caso clássico de transmissão com modulação BPSK.

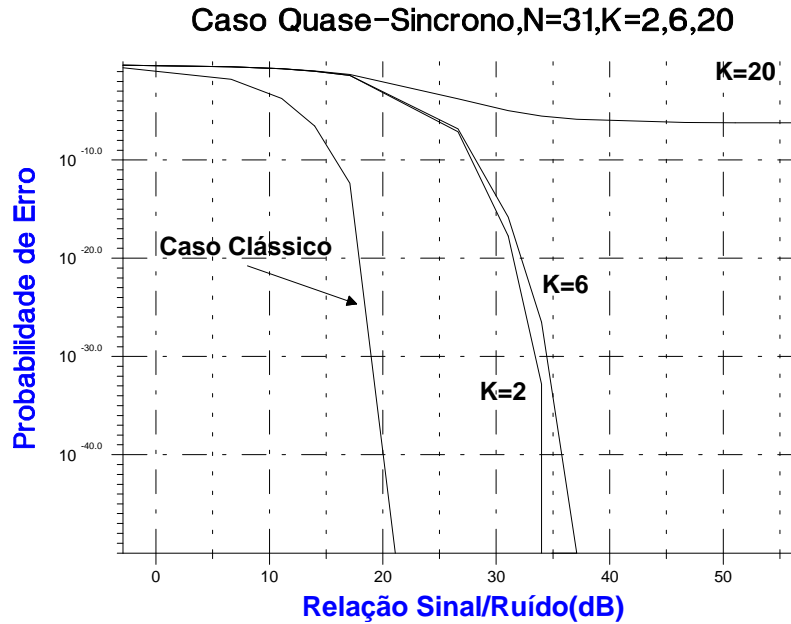


Figura 17

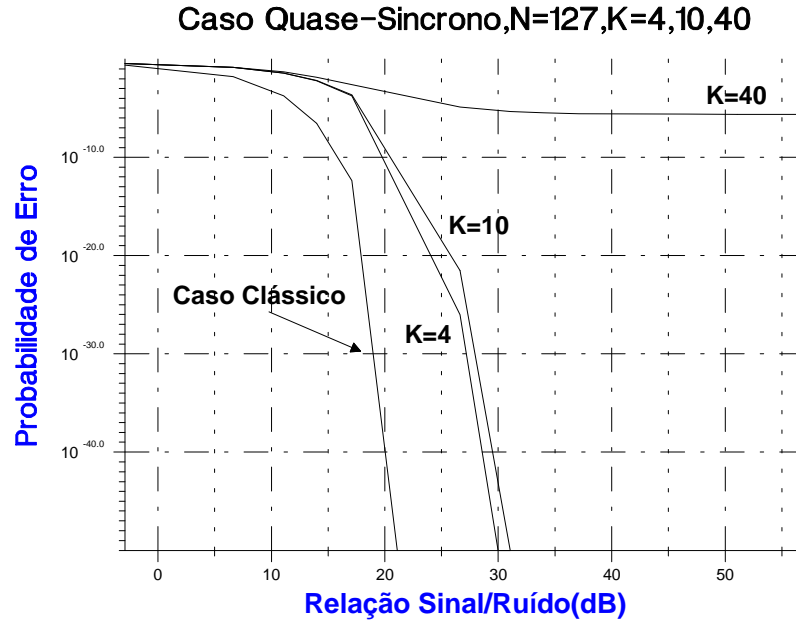


Figura 18

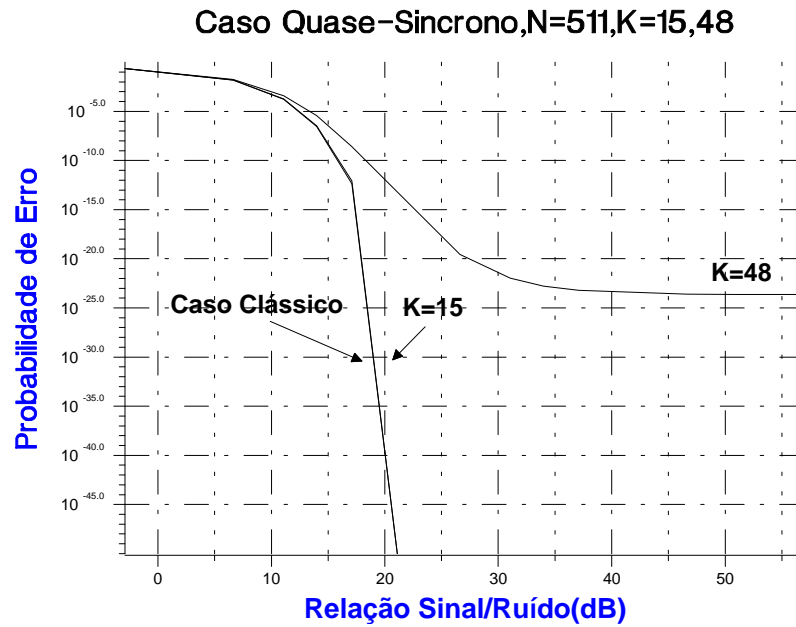


Figura 19

A performance de sistemas quase-síncronos depende prioritariamente da escolha seletiva de seqüências de assinatura com características favoráveis de correlação cruzada, ref. [16] e [11]. O critério de otimização na escolha de seqüências de uma maneira genérica é minimizar simultaneamente os módulos de $\mathbf{R}_{i,x}(\tau)$ e $\hat{\mathbf{R}}_{i,x}(\tau)$.

Em QS-CDMA τ está restrito a uns poucos décimos de um chip. Na ref. [16] demonstra-se que: $|\mathbf{R}_{i,x}(\tau)| \leq |\hat{\mathbf{R}}_{i,x}(\tau)| + 2\tau$, mostrando a dependência existente entre as duas correlações. Portanto minimizando $|\hat{\mathbf{R}}_{i,x}(\tau)|$ automaticamente minimiza-se $|\mathbf{R}_{i,x}(\tau)|$.

Na equação (100) nota-se que o ideal seria a escolha adequada de seqüências de modo a minimizar a somatória $\sum_{\substack{i=1 \\ i \neq x}}^K \rho_{i,x}$. Os métodos de procura deste valor de mínimo global empregam métodos heurísticos de pesquisa, porém devido ao tempo de execução aumentar enormemente conforme o aumento no número de usuários e do comprimento das seqüências e o fato deste mínimo global ao ser obtido ser válido somente quando os K usuários estão transmitindo, optou-se por um método prático que apresenta um solução sub-ótima, mas de aplicação simples e prática. O método utilizado para seleção das fases sub-ótimas dada uma família de seqüências consiste em:

1º Passo) Introdução da primeira seqüência da família com uma fase qualquer.

2º Passo) Introdução da segunda seqüência da família com uma fase qualquer.

3º Passo) Procura da fase ótima da última seqüência introduzida em relação às outras seqüências já introduzidas. Isto é realizado rotacionando-se esta seqüência introduzida e verificando-se a fase em que ocorre o menor valor da somatória $\sum_{\substack{i=1 \\ i \neq x}}^H \rho_{i,x}$ onde H é igual ao número de seqüências introduzidas até a execução deste passo. Achando-se a fase ótima da última seqüência introduzida esta permanece fixa em todos os passos que se sucedem.

4º Passo) Verificação se todas as seqüências da família já foram introduzidas, se não introduz-se mais uma seqüência e retorna-se ao 3º Passo.

5º Passo) Fim do algoritmo.

10 - ABORDAGEM GAUSSIANA DE LEHNERT PARA SEQÜÊNCIAS ALEATÓRIAS

A abordagem de Lehnert, ref. [22], procura caracterizar probabilisticamente a interferência multi-usuário considerando as seqüências de todos os usuários como aleatórias.

Reescrevendo a interferência multi-usuário normalizada na saída do correlator (17):

$$\mathbf{I}_{i,x}(\underline{\mathbf{b}}_i, \tau, \phi) = \mathbf{T}^{-1} \cdot [\underline{\mathbf{b}}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i)] \cdot \cos(\phi_i) = \mathbf{T}^{-1} \cdot \mathbf{B}_{i,x}(\underline{\mathbf{b}}_i, \tau) \cdot \cos(\phi_i) \quad (104)$$

Onde:

$$\mathbf{B}_{i,x}(\mathbf{b}_i, \tau) = \mathbf{b}_{-1}^{(i)} \cdot \mathbf{R}_{i,x}(\tau_i) + \mathbf{b}_0^{(i)} \cdot \hat{\mathbf{R}}_{i,x}(\tau_i) \quad (105)$$

Reescrevendo agora as equações (153) e (154) e chamando $\mathbf{S}_i = (\tau_i - \mathbf{l}_i \cdot \mathbf{T}\mathbf{c})$ resulta:

$$\mathbf{R}_{i,x}(\tau_i) = \mathbf{C}_{i,x}(\mathbf{l}_i - \mathbf{N}) \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{C}_{i,x}(\mathbf{l}_i + 1 - \mathbf{N}) \cdot \mathbf{R}_\psi(\mathbf{S}_i) \quad (106)$$

$$\hat{\mathbf{R}}_{i,x}(\tau_i) = \mathbf{C}_{i,x}(\mathbf{l}_i) \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{C}_{i,x}(\mathbf{l}_i + 1) \cdot \mathbf{R}_\psi(\mathbf{S}_i) \quad (107)$$

Substituindo-se (106), (107) e (52) em (105) resulta:

$$\begin{aligned} \mathbf{B}_{i,x}(\mathbf{b}_i, \tau) = & \left[\sum_{j=0}^{\mathbf{l}_i-1} \mathbf{b}_{-1}^{(i)} \cdot \mathbf{c}_{j-\mathbf{l}_i+\mathbf{N}}^{(i)} \cdot \mathbf{c}_j^{(x)} + \sum_{j=\mathbf{l}_i}^{\mathbf{N}-1} \mathbf{b}_0^{(i)} \cdot \mathbf{c}_{j-\mathbf{l}_i}^{(i)} \cdot \mathbf{c}_j^{(x)} \right] \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \\ & \left[\sum_{j=0}^{\mathbf{l}_i} \mathbf{b}_{-1}^{(i)} \cdot \mathbf{c}_{j-\mathbf{l}_i-1+\mathbf{N}}^{(i)} \cdot \mathbf{c}_j^{(x)} + \sum_{j=\mathbf{l}_i+1}^{\mathbf{N}-1} \mathbf{b}_0^{(i)} \cdot \mathbf{c}_{j-\mathbf{l}_i-1}^{(i)} \cdot \mathbf{c}_j^{(x)} \right] \cdot \mathbf{R}_\psi(\mathbf{S}_i) \end{aligned} \quad (108)$$

Rearranjando (108) de maneira conveniente, obtém-se:

$$\begin{aligned} \mathbf{B}_{i,x}(\mathbf{b}_i, \tau) = & \mathbf{b}_{-1}^{(i)} \cdot \sum_{j=0}^{\mathbf{l}_i-1} \mathbf{c}_{j-\mathbf{l}_i+\mathbf{N}}^{(i)} \cdot [\mathbf{c}_j^{(x)} \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{c}_{j+1}^{(x)} \cdot \mathbf{R}_\psi(\mathbf{S}_i)] + \\ & \mathbf{b}_0^{(i)} \cdot \sum_{j=\mathbf{l}_i}^{\mathbf{N}-2} \mathbf{c}_{j-\mathbf{l}_i}^{(i)} \cdot [\mathbf{c}_j^{(x)} \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{c}_{j+1}^{(x)} \cdot \mathbf{R}_\psi(\mathbf{S}_i)] + \\ & \mathbf{b}_0^{(i)} \cdot \mathbf{c}_{\mathbf{N}-\mathbf{l}_i-1}^{(i)} \cdot \mathbf{c}_{\mathbf{N}-1}^{(x)} \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{b}_{-1}^{(i)} \cdot \mathbf{c}_{\mathbf{N}-\mathbf{l}_i-1}^{(i)} \cdot \mathbf{c}_0^{(x)} \cdot \mathbf{R}_\psi(\mathbf{S}_i) \end{aligned} \quad (109)$$

Condicinando a equação acima em relação à sequência do usuário x -ésimo (sequência utilizada pelo usuário que se deseja receber) e em relação à \mathbf{l}_i definem-se as $\mathbf{N}+1$ variáveis aleatórias:

$$\mathbf{M}_j = \begin{cases} \mathbf{b}_{-1}^{(i)} \cdot \mathbf{c}_{j-\mathbf{l}_i+\mathbf{N}}^{(i)} \cdot \mathbf{c}_j^{(x)} & \mathbf{j} = 0, 1, \dots, \hat{\mathbf{l}}_i - 1 \\ \mathbf{b}_0^{(i)} \cdot \mathbf{c}_{j-\mathbf{l}_i}^{(i)} \cdot \mathbf{c}_j^{(x)} & \mathbf{j} = \hat{\mathbf{l}}_i, \hat{\mathbf{l}}_i + 1, \dots, \mathbf{N} - 2 \\ \mathbf{b}_0^{(i)} \cdot \mathbf{c}_{\mathbf{N}-\mathbf{l}_i-1}^{(i)} \cdot \mathbf{c}_{\mathbf{N}-1}^{(x)} & \mathbf{j} = \mathbf{N} - 1 \\ \mathbf{b}_{-1}^{(i)} \cdot \mathbf{c}_{\mathbf{N}-\mathbf{l}_i-1}^{(i)} \cdot \mathbf{c}_0^{(x)} & \mathbf{j} = \mathbf{N} \end{cases} \quad (110)$$

Definindo-se ainda $\mathbf{f}(\mathbf{S}) = \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{R}_\psi(\mathbf{S}_i)$ e $\mathbf{g}(\mathbf{S}) = \hat{\mathbf{R}}_\psi(\mathbf{S}_i) - \mathbf{R}_\psi(\mathbf{S}_i)$ e substituindo (110) em (109), resulta:

$$\mathbf{B}_{i,x}(\mathbf{b}_i, \tau) = \sum_{j=0}^{N-2} \mathbf{M}_j \cdot [\hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \hat{\mathbf{c}}_j^{(x)} \cdot \hat{\mathbf{c}}_{j+1}^{(x)} \cdot \mathbf{R}_\psi(\mathbf{S}_i)] + \mathbf{M}_{N-1} \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{M}_N \cdot \mathbf{R}_\psi(\mathbf{S}_i) \quad (111)$$

Definindo \mathbf{A} como o conjunto tal que $\hat{\mathbf{c}}_j^{(x)} \cdot \hat{\mathbf{c}}_{j+1}^{(x)} = 1$ onde $|\mathbf{A}|$ é a cardinalidade deste conjunto e o conjunto \mathbf{B} tal que $\hat{\mathbf{c}}_j^{(x)} \cdot \hat{\mathbf{c}}_{j+1}^{(x)} = -1$ onde $|\mathbf{B}|$ é a cardinalidade deste conjunto, pode-se reescrever (111) como:

$$\mathbf{B}_{i,x}(\mathbf{b}_i, \tau) = \sum_{j \in \mathbf{A}} \mathbf{M}_j \cdot \mathbf{f}(\mathbf{S}_i) + \sum_{j \in \mathbf{B}} \mathbf{M}_j \cdot \mathbf{g}(\mathbf{S}_i) + \mathbf{M}_{N-1} \cdot \hat{\mathbf{R}}_\psi(\mathbf{S}_i) + \mathbf{M}_N \cdot \mathbf{R}_\psi(\mathbf{S}_i) \quad (112)$$

Comparando $\hat{\mathbf{C}}_{x,x}(\mathbf{1}) = \sum_{i=0}^{N-2} \hat{\mathbf{c}}_i^{(x)} \cdot \hat{\mathbf{c}}_{i+1}^{(x)}$ com (111) e (112) conclue-se que $\hat{\mathbf{C}}_{x,x}(\mathbf{1}) = |\mathbf{A}| - |\mathbf{B}|$ e $|\mathbf{A}| + |\mathbf{B}| = N - 1$. Portanto:

$$|\mathbf{A}| = \frac{N - 1 - \hat{\mathbf{C}}_{x,x}(\mathbf{1})}{2} \quad (113)$$

$$|\mathbf{B}| = \frac{N - 1 - \hat{\mathbf{C}}_{x,x}(\mathbf{1})}{2} \quad (114)$$

Com este resultado verifica-se que o condicionamento de (109) em relação à sequência do usuário x -ésimo (sequência utilizada pelo usuário que se deseja receber) é equivalente ao condicionamento em relação a $\hat{\mathbf{C}}_{x,x}(\mathbf{1})$, $|\mathbf{A}|$ ou $|\mathbf{B}|$.

Usando a notação padronizada adotada nas ref. [14] e [22], definem-se as seguintes variáveis:

$$\mathbf{X}_i = \sum_{j \in \mathbf{A}} \mathbf{M}_j \quad (115)$$

$$\mathbf{Y}_i = \sum_{j \in \mathbf{B}} \mathbf{M}_j \quad (116)$$

$$\mathbf{P}_i = \mathbf{M}_j \quad \text{para } j = N - 1 \quad (117)$$

$$\mathbf{Q}_i = \mathbf{M}_j \quad \text{para } j = N \quad (118)$$

Com esta padronização, substituindo (112) em (18) e considerando (20), a interferência na saída no correlator composta pela interferência multi-usuário(MUI) mais a parcela devido ao ruído branco gaussiano é:

$$\begin{aligned}
\text{Interf}_x + \eta_x &= \frac{A}{2} \cdot \sum_{\substack{i=1 \\ i \neq x}}^K [X_i \cdot f(S_i) + Y_i \cdot g(S_i) + P_i \cdot \hat{R}_\psi(S_i) + Q_i \cdot R_\psi(S_i)] \cdot \cos(\phi_i) + \eta_x \\
&= \frac{A}{2} \cdot \sum_{\substack{i=1 \\ i \neq x}}^K W_i + \eta_x = \text{MUI} + \eta_x
\end{aligned} \tag{119}$$

Para o caso em estudo, onde se considera os chips com formas de onda retangulares, utilizando (149) e (150) do Apêndice A, resulta $f(S) = T_c$ e $g(S) = T_c - 2 \cdot S$. Portanto, a variável aleatória W_i pode ser reescrita como:

$$W_i = [X_i \cdot T_c + Y_i \cdot (T_c - 2 \cdot S_i) + P_i \cdot (T_c - S_i) + Q_i \cdot S_i] \cdot \cos(\phi_i) \tag{120}$$

Considerando as seqüências aleatórias as funções densidade de probabilidade das variáveis aleatórias que compõe (120) são, ref. [22]:

P_k possui uma fdp uniformemente distribuída no conjunto $[-1, +1]$

Q_k possui uma fdp uniformemente distribuída no conjunto $[-1, +1]$

X_k possui a fdp $p_{X_k}(j) = \left(\frac{|A|}{j + |A|} \right) \cdot 2^{-|A|}$ com $j \in \{-|A|, -|A| + 2, \dots, |A| - 2, |A|\}$

Y_k possui a fdp $p_{Y_k}(j) = \left(\frac{|B|}{j + |B|} \right) \cdot 2^{-|B|}$ com $j \in \{-|B|, -|B| + 2, \dots, |B| - 2, |B|\}$

$\hat{C}_{x,x}(1)$ possui a fdp $p_{\hat{C}_{x,x}(1)}(j) = \left(\frac{N-1}{j + N-1} \right) \cdot 2^{1-N}$ com $j \in \{1-N, 3-N, \dots, N-3, N-1\}$

Onde: $\binom{a}{b} = \frac{a!}{b!(a-b)!}$

Convém destacar que o cálculo de probabilidade de erro de bit feito pelo método de Nazari pode também ser obtido a partir de (119) e (120) considerando as seqüências determinísticas e as distribuições das funções densidade de probabilidade adotadas para ϕ_i e τ_i respectivamente.

Calculando a variância e a esperança de (119) e assumindo que a MUI é uma variável aleatória gaussiana demonstra-se na ref. [22] que, dependendo do alinhamento da fase ϕ_i e do atraso de chip dos sinais interferentes, a probabilidade de erro de bit, pode ser calculada por:

Tipo 1 - quando a fase da portadora e o atraso do chip dos sinais interferentes estão sincronizados com o sinal desejado:

$$P_e = Q(\text{SNR}) = \left(\frac{K-1}{N} + \frac{N_o}{A^2 \cdot T} \right)^{-\frac{1}{2}} \quad (121)$$

Tipo 2 - quando a fase da portadora está sincronizada e o atraso do chip é aleatório considerado como tendo uma fdp uniformemente distribuída em $[0, T]$:

$$P_e = Q(\text{SNR}) = Q\left(\frac{2 \cdot (K-1)}{3 \cdot N} + \frac{N_o}{A^2 \cdot T} \right)^{-\frac{1}{2}} \quad (122)$$

Tipo 3 - quando o atraso do chip está sincronizado e a fase do chip é aleatória considerada como tendo uma fdp uniformemente distribuída em $[0, 2\pi]$:

$$P_e = Q(\text{SNR}) = Q\left(\frac{K-1}{2 \cdot N} + \frac{N_o}{A^2 \cdot T} \right)^{-\frac{1}{2}} \quad (123)$$

A equação (123) é idêntica à equação (137) adiante, no limite para $A \rightarrow \infty$, mostrando que os valores assintóticos da probabilidade de erro de bit nesta abordagem coincidem com os obtidos pelo método de Weber, conforme pode-se verificar nas figuras a seguir.

Tipo 4 - quando a fase da portadora e o atraso do chip não estão sincronizados, considerados como tendo fdps uniformemente distribuídas em $[0, 2\pi]$ e $[0, T]$, respectivamente:

$$P_e = Q(\text{SNR}) = Q\left(\frac{K-1}{3 \cdot N} + \frac{N_o}{A^2 \cdot T} \right)^{-\frac{1}{2}} \quad (124)$$

A equação (124) é idêntica à equação (54) obtida pelo método de Pursley, considerando seqüências randômicas, o que era de se esperar, uma vez que os dois métodos assumem que a MUI é uma variável aleatória gaussiana e ambos calculam a variância exatamente utilizando métodos estatísticos.

Nas figuras 20, 21 e 22 tem-se gráficos de performance respectivamente para comprimento de seqüências de assinatura $N = 31$ e número de usuários $K = 2$; comprimento de seqüências de assinatura $N = 127$ e número de usuários $K = 10$ e finalmente comprimento de seqüências de assinatura $N = 511$ e número de usuários $K = 48$. Tem-se ainda para efeito de comparação a curva de performance do caso clássico de transmissão com modulação BPSK e as simulações feitas pelos métodos de Pursley com seqüências determinísticas e Weber que também assumem a hipótese de que a interferência multi-usuário tem uma distribuição gaussiana.

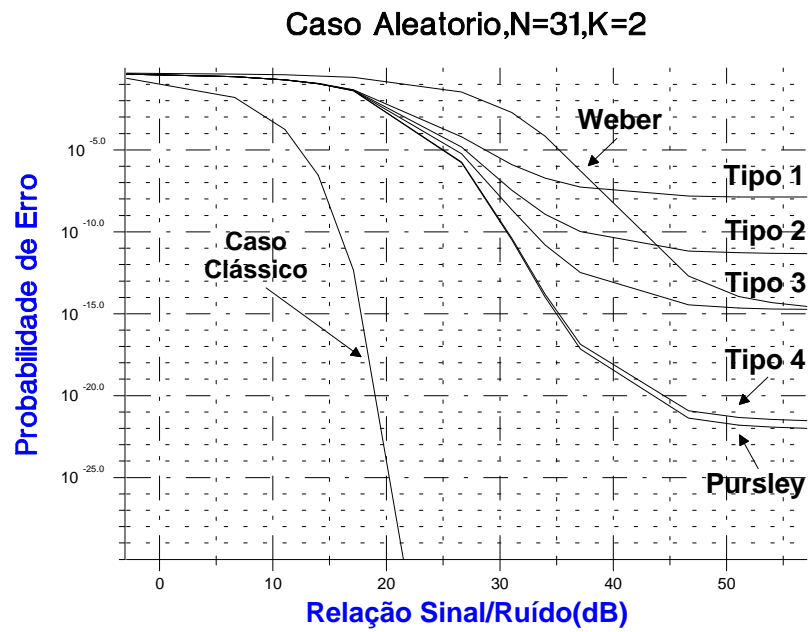


Figura 20

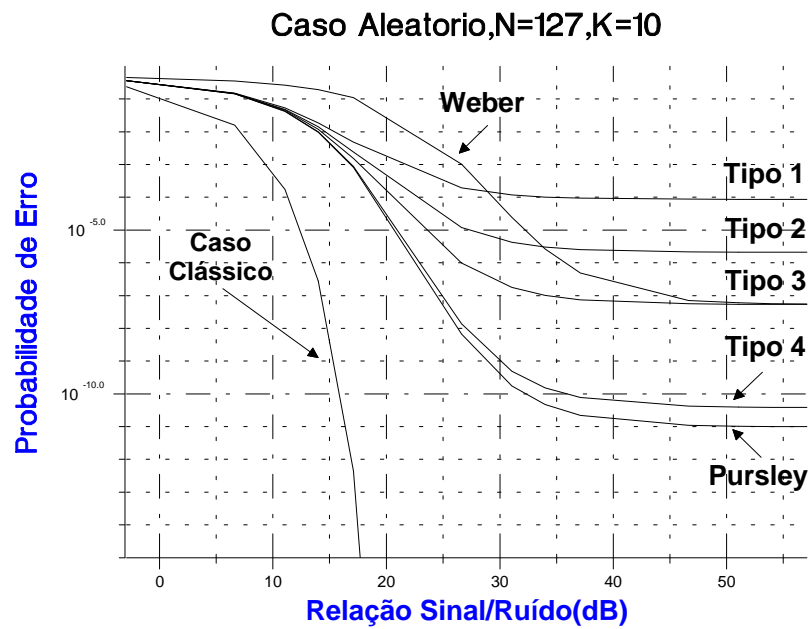


Figura 21

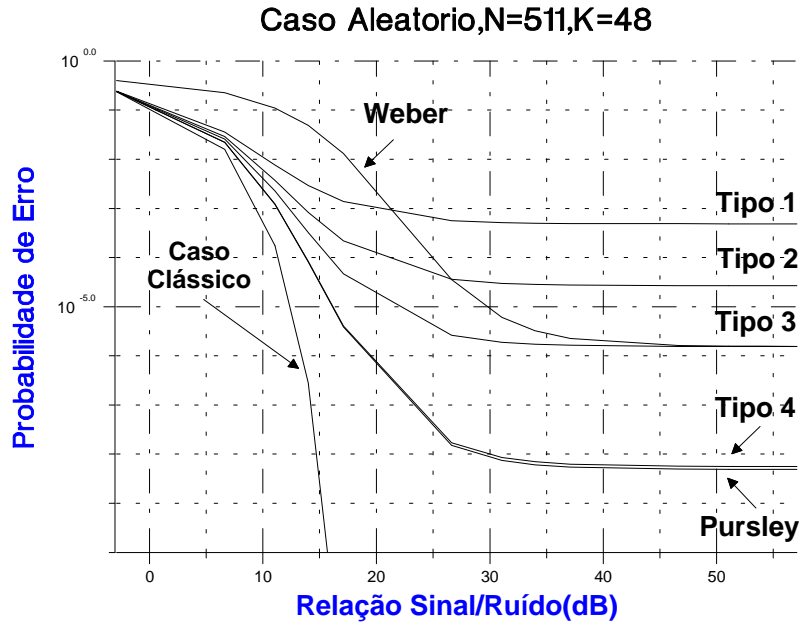


Figura 22

11 - SIMULAÇÕES

Programa utilizado para as simulações

Adotou-se para uso nas simulações o programa MATLAB V4.2b for Windows. Os programas escritos com o uso de sua linguagem interna tendem a ser menores para esta aplicação quando comparados aos escritos em outras linguagens de baixo nível (por exemplo, Linguagem C), porque o MATLAB possui funções internas que realizam operações básicas e complexas com matrizes e vetores que foram largamente utilizadas nas simulações.

Seqüências utilizadas nas simulações

Adotou-se os seguintes polinômios primitivos a partir da tabela presente na ref. [25]:

Polinômio primitivo para seqüências de comprimento L=31: $x^5 + x^2 + 1$

Polinômio primitivo para seqüências de comprimento L=127: $x^7 + x^3 + 1$

Polinômio primitivo para seqüências de comprimento L=511: $x^9 + x^4 + 1$

A partir dos polinômios primitivos selecionados geraram-se todas as seqüências de máximo comprimento existentes para seqüências com aquele comprimento, através da função ALLSMC. Isto é realizado gerando-se uma tabela de coconjuntos ciclotômicos que fornece as

dizimações necessárias para gerar todas as seqüências existentes a partir de um único polinômio primitivo. Através da função PARPREF.M buscou-se todos os pares preferenciais existentes nas seqüências de máximo comprimento. Selecionou-se o primeiro par preferencial obtido. A partir da função GOLD.M gerou-se uma família de seqüências de Gold. Escolheu-se as primeiras **K** seqüências da família para serem usadas nas simulações.

Dados usados nas simulações pelo método de Nazari

Seqüências utilizadas: Gold

Parâmetros físicos das seqüências: **A = 1** e **Tc = 1**

Parâmetros físicos do ruído branco gaussiano: **N₀ = 0.1**

Detalhes sobre a implementação do algoritmo de cálculo pelo método de Nazari

- Método de escolha dos pontos de amostragem das funções densidade de probabilidade:

No comportamento da função densidade de probabilidade definido pelas equações (78) e (79) verifica-se que existe uma singularidades no ponto $|d|=|\theta_{i,x}(I_i)|$. É importante também ressaltar que as funções de correlação cruzada periódica $\theta_{i,x}(I_i)$ e $\hat{\theta}_{i,x}(I)$ retornam sempre valores inteiros. No programa **NAZARIM** evitou-se amostrar estas funções densidade de probabilidade nos pontos em que existem singularidades. O usuário fornece como um parâmetro para o programa o número de amostras para cada unidade (para cada **d = 1**). Este parâmetro tem o nome de **amostras**.

Para evitar a amostragem nos pontos de singularidades o programa inicia a amostragem em **d = (2.amostras)⁻¹** e em seguida amostra a cada **d = (amostras)⁻¹** para a direita e para a esquerda do ponto inicial.

Um outro parâmetro fornecido pelo usuário denomina-se **limitemax** e corresponde ao limite superior (por simetria **-limitemax** corresponde ao limite inferior) dos pontos de amostragem que é o maior valor de abscissa (**d**) dentre todas as funções densidade de probabilidade definidas pelas equações (78), (79) e (84) a (86). Este parâmetro é obtido através do programa **LIMNAZAR.M** e juntamente com o parâmetro **amostras** estabelecem o passo de amostragem.

O programa **LIMNAZAR.M** cria uma tabela de correlações cruzadas periódicas pares $\theta_{i,x}(I_i)$ para todos os **i = 1,2,...,K** com **i ≠ x** e para todos **I_i = 1,2,...,N** e outra tabela de correlações cruzadas periódicas ímpares $\hat{\theta}_{i,x}(I)$ nas mesmas condições que a anterior. Determina-se o maior valor em módulo dentre as duas tabelas, e este é o limite superior do argumento da maior função densidade de probabilidade dentre todos os usuários interferentes e dados de informação.

Este limite superior e o parâmetro **amostras** estabelecem todos os pontos de amostragem das funções densidade de probabilidade definidas pelas equações (59), (60), (65), (66) e (67), bem como permitem estabelecer o passo desta amostragem.

- Algoritmo de cálculo pelo método de Nazari:

a) Compara-se $\theta_{i,x}(l_i)$ e $\theta_{i,x}(l_i + 1)$ e determina-se a fdp **p1xy** onde x é o elemento da sequência l_i e y corresponde ao usuário em análise. Estas fdps resultam de todas as combinações possíveis entre as seqüências interferentes e a do usuário casado com o receptor.

b) Compara-se $\hat{\theta}_{i,x}(l_i)$ e $\hat{\theta}_{i,x}(l_i + 1)$ e determina-se a fdp **p2xy** onde x corresponde ao usuário em análise e y é o elemento da sequência l_i .

c) Calculam-se fdps médias entre as obtidas nos itens a) e b):

$$p3y = \frac{1}{N} \times \sum_{x=1}^N p1xy \quad (125)$$

$$p4y = \frac{1}{N} \times \sum_{x=1}^N p2xy \quad (126)$$

d) Calculam-se as fdps das interferências normalizadas de cada usuário y:

$$p5ay = \frac{p3y + p4y}{2} \quad (127)$$

e) Desnormalizam-se as fdps obtidas no item d) (ver equações (91) e (92)):

$$p5cy = p5ay_{\text{desnormalizado}} \quad (128)$$

f) Reamostram-se as fdps obtidas no item e) de modo a obter a amostra na origem:

$$p5by = p5cy_{\text{reamostrado}} \quad (129)$$

g) Convoem-se todas as fdps das interferências dos usuários indesejáveis resultando então a fdp da interferência multi-usuário.

$$p6 = \bigotimes_{\substack{i=1 \\ i \neq x}}^K p5by \quad (130)$$

h) Amostra-se a fdp do ruído branco gaussiano $p_{\text{gaussiana}}$ com o mesmo passo das abcissas do item g).

i) Realiza-se a convolução entre a fdp da interferência de multi-usuário e a fdp da gaussiana obtendo a fdp das interferências na saída do receptor

$$p_{\text{interferencia total}} = p_6 \otimes p_{\text{gaussiana}} \quad (131)$$

j) Integra-se (131) para calcular a probabilidade de erro de bit.

- Estimativa do erro ocorrido durante as simulações:

As funções densidade de probabilidade definidas pelas equações (78), (79) e (84) a (86) mostram que em termos espectrais estas se distribuem-se de $-\infty$ até $+\infty$ o que obriga um número infinito de pontos para se obter uma precisão absoluta. Introduziu-se no programa uma variável de controle de nome **amostras** que determina o número de amostras em cada unidade da abcissa (para cada **d = 1**) e que desta maneira permite melhorar a precisão dos cálculos. O programa **NAZARI.M** gera após a sua execução quatro outras variáveis que permitem verificar a precisão dos cálculos obtidos; são elas: **integridade1**, **integridade2**, **integridade3** e **integridade4**. Estas variáveis correspondem à área de funções densidade de probabilidade em determinado momento dos cálculos. Se fosse possível obter uma precisão absoluta, estas áreas deveriam ter todas o valor correspondendo a **1**. Segue-se uma descrição do significado do momento em que estas variáveis são calculadas.

- A variável **integridade1** é um vetor com os valores correspondentes às áreas das fdps definidas pelas equações (78), (79) e (84) a (86).

- A variável **integridade2** é um valor correspondente à área da fdp definida pelas equações (125) e (126).

- A variável **integridade3** é um valor correspondente à área da fdp do ruído branco gaussiano.

- A variável **integridade4** é um valor correspondente à área da fdp definida pela equação (131).

Portanto, o quanto mais próximo do valor **1** estiver a variável **integridade4** mais preciso será o resultado obtido pelo programa **NAZARI.M**

- Exemplo de execução do algoritmo de cálculo pelo método de Nazari:

Nas páginas (43) até (52) encontram-se 2 exemplos de execução do programa **NAZARI.M** em situações distintas de modo a exemplificar o algoritmo de cálculo.

As seqüências utilizadas nestes exemplos são as de Gold com comprimento 3 e é analisado o caso de 3 usuários (2 sinais interferentes e um terceiro casado com o receptor). As seqüências de Gold foram geradas a partir do polinômio primitivo $f(x) = x^3 + x + 1$.

A partir da execução do programa **PARPREF.M** selecionou-se as seguintes seqüências preferenciais:

seqüência preferencial 1 = { **1 1 1 0 1 0 0** }

seqüência preferencial 2 = { **1 0 0 1 0 1 1** }

Estas seqüências preferenciais são capazes de gerar 9 códigos de Gold (7 resultantes de todas as operação XOR entre todas as combinações de fase possíveis entre as duas seqüências e mais 2 do próprio par preferencial). Através do programa **GOLD.M** selecionou-se as seguintes 3 seqüências de Gold:

seqüência Gold 1 = { **0 1 1 1 1 1 1** }

seqüência Gold 2 = { **0 0 1 0 0 0 1** }

seqüência Gold 3 = { **0 0 0 0 1 1 0** }

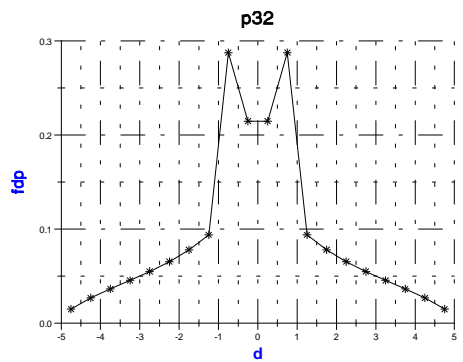
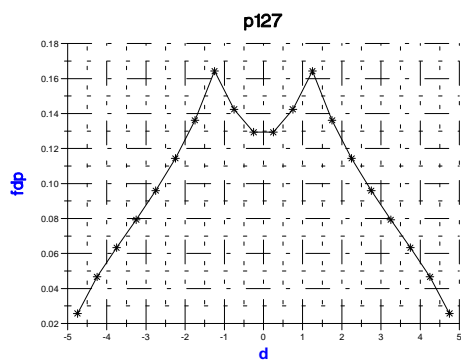
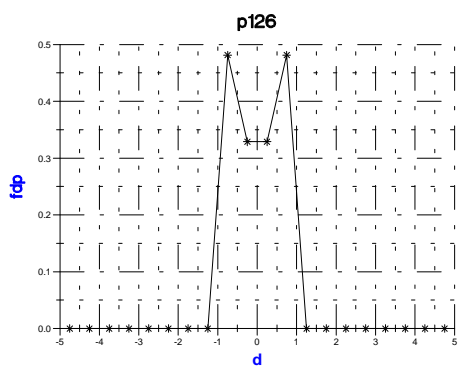
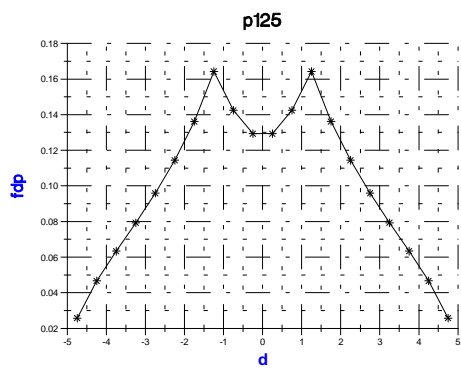
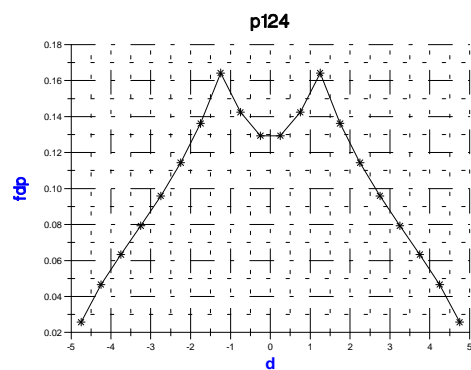
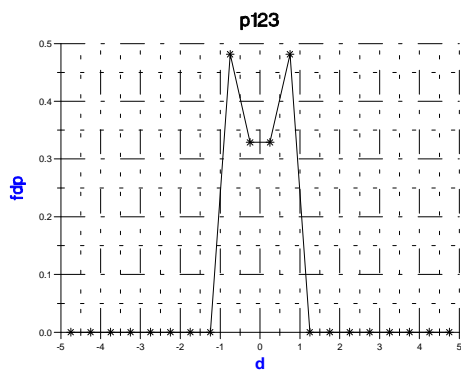
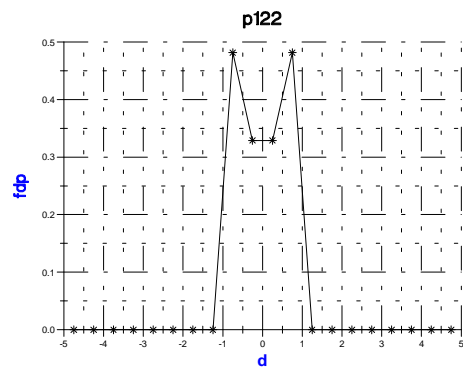
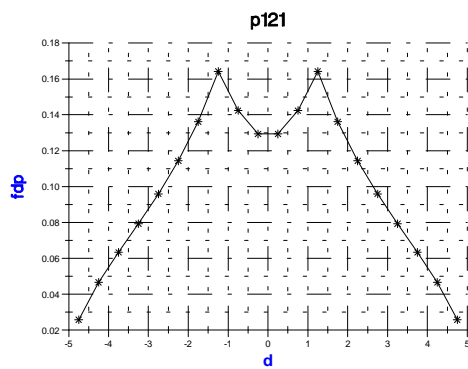
Como parâmetros globais para execução do programa foram utilizados: tempo de chip $T_c = 1$, densidade espectral de potência do ruído branco gaussiano $N_0 = 0.1$, número do usuário que se encontra perfeitamente sincronizado com o receptor igual ao da primeira seqüência (seqüência Gold 1) e amplitude de pico dos sinais na entrada do receptor $A = 1$. Nas matrizes abaixo tem-se as correlações cruzadas onde as colunas correspondem aos valores I_i e a linha ao respectivo usuário.

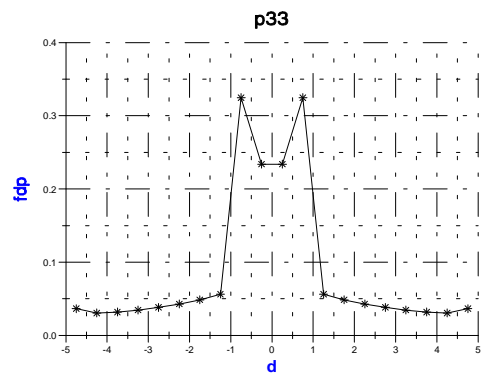
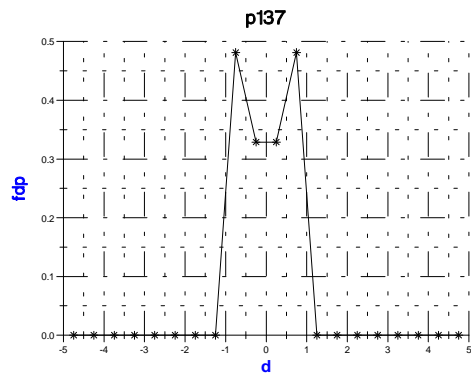
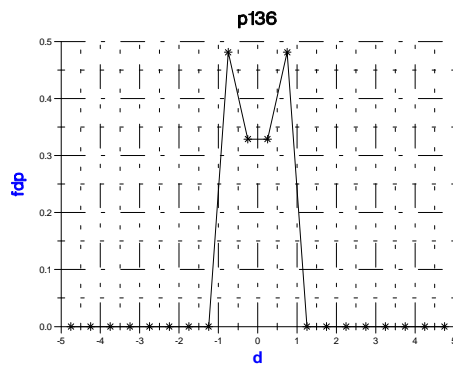
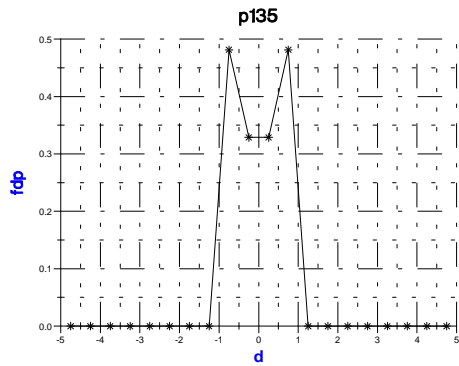
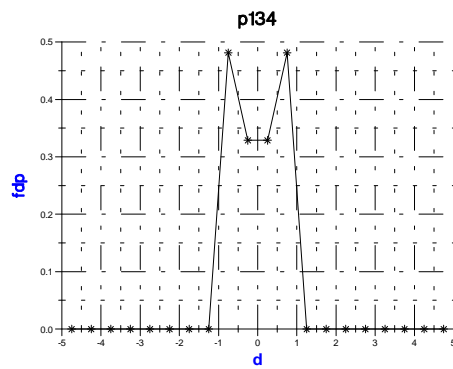
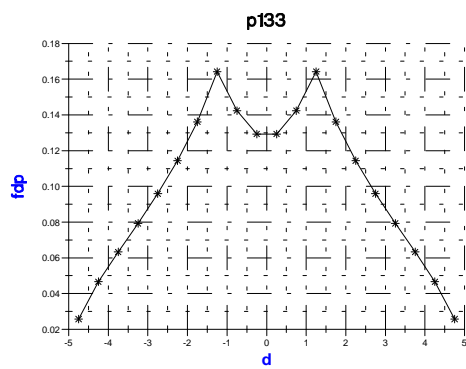
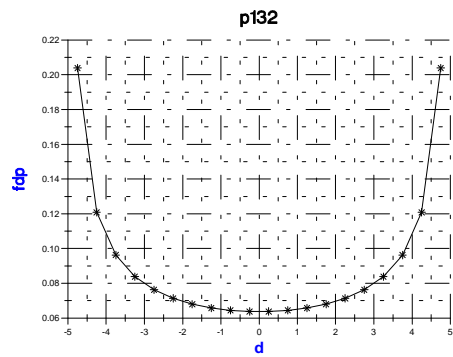
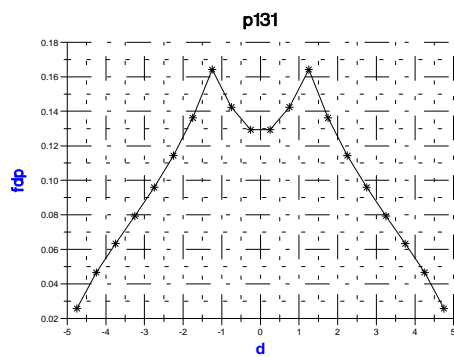
$$\theta_{i,x}(I_i) = \begin{bmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 7 \\ -5 & -1 & -1 & -1 & -5 & -1 & -1 \\ -1 & -5 & -5 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (132)$$

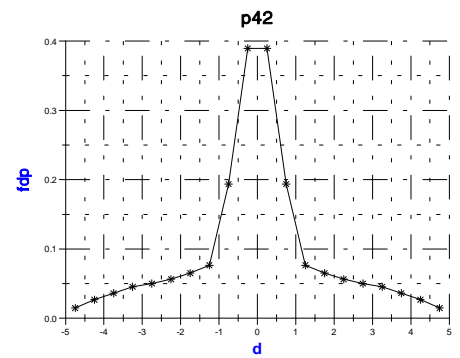
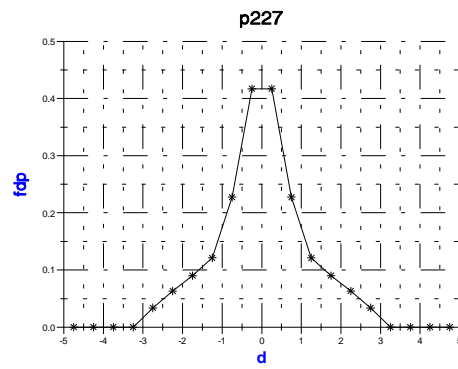
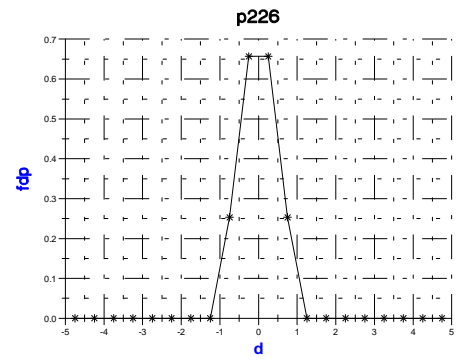
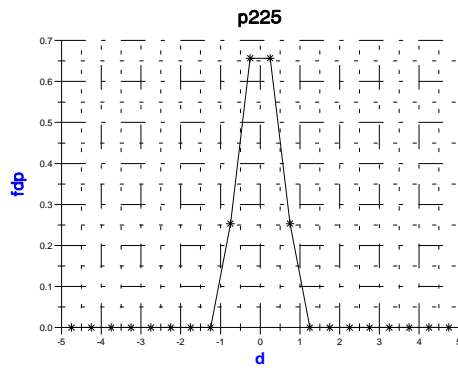
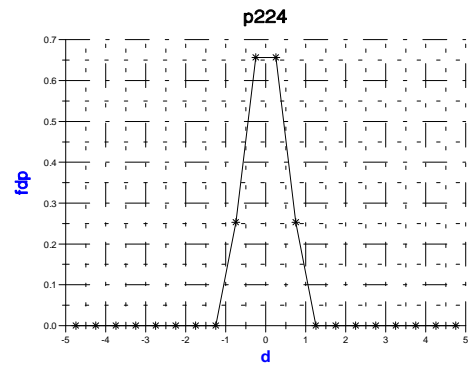
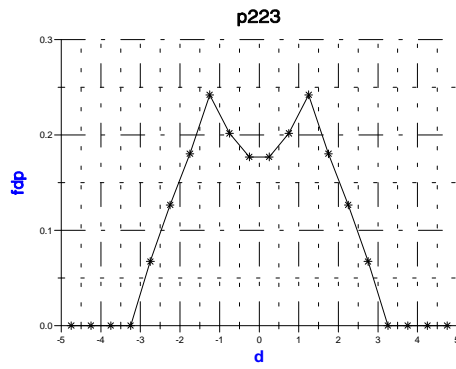
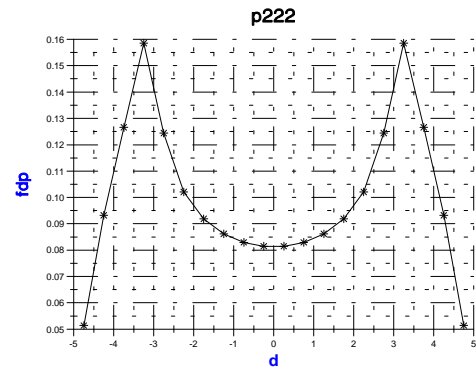
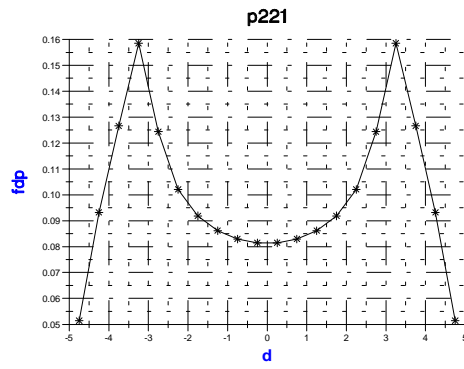
$$\hat{\theta}_{i,x}(I_i) = \begin{bmatrix} 5 & 3 & 1 & -1 & -3 & -5 & -7 \\ -3 & -5 & -3 & -1 & 1 & -1 & 1 \\ -3 & -1 & -3 & -5 & -3 & -1 & 1 \end{bmatrix} \quad (133)$$

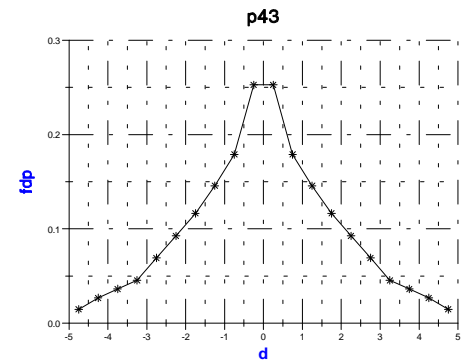
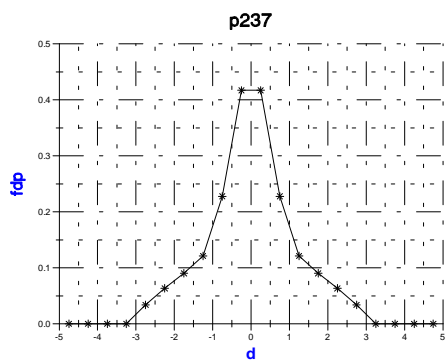
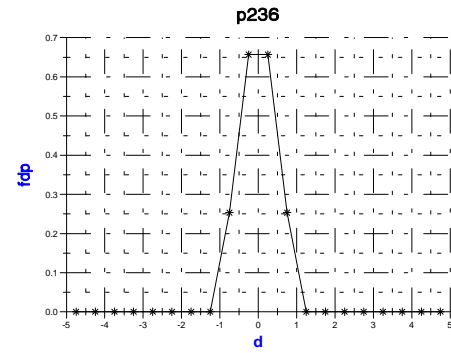
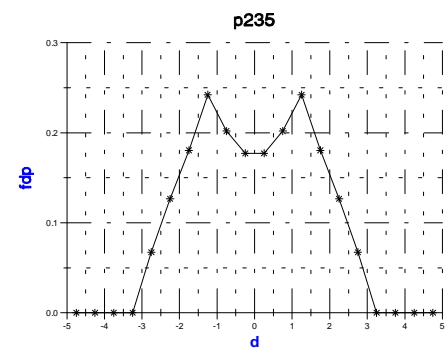
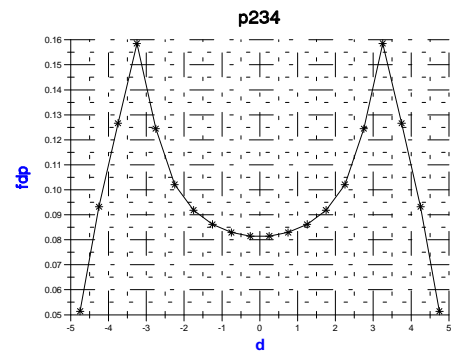
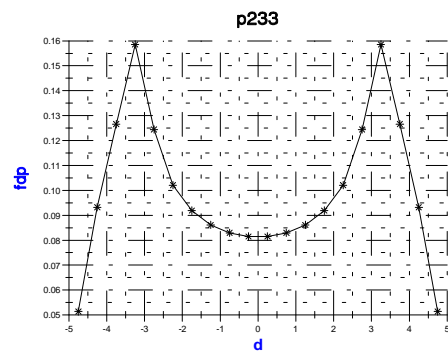
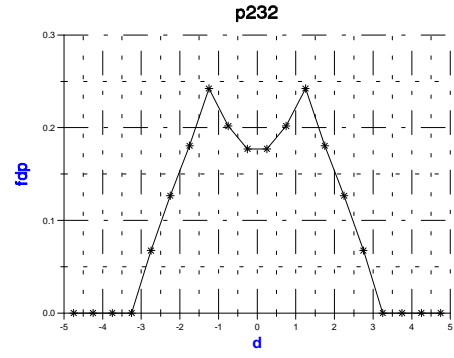
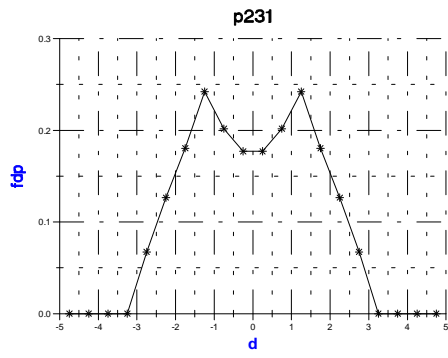
Através do programa **LIMNAZAR.M** obtém-se o parâmetro **limitemax=5** que neste pequeno exemplo pode também ser obtido através de inspeção das matrizes (132) e (133), lembrando que a primeira linha das matrizes correspondem a autocorrelações e portanto não devem ser consideradas.

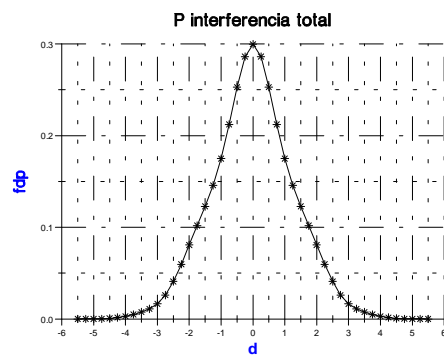
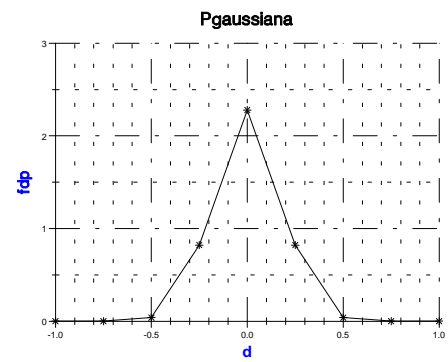
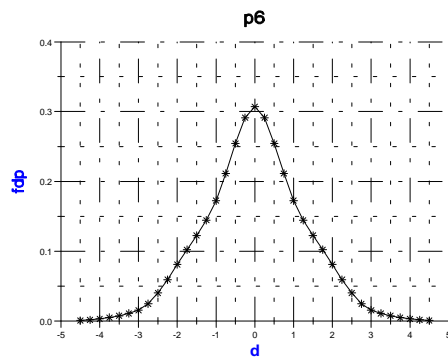
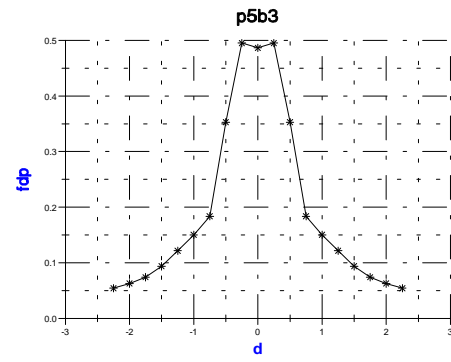
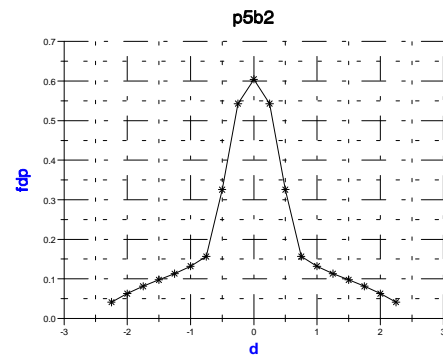
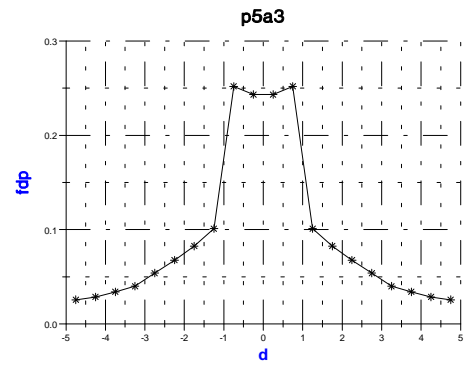
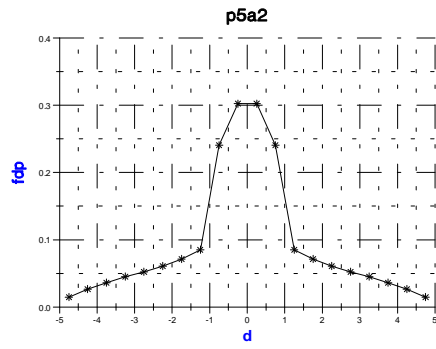
Nas páginas (43), (44), (45), (46) e (47) encontra-se uma primeira simulação executada com **2 amostras por unidade** e nas páginas (48), (49), (50), (51) e (52) encontra-se uma segunda simulação executada com **100 amostras por unidade**. Na primeira simulação visualiza-se os pontos de amostragem perfeitamente através do símbolo “*” embora os mesmos estejam conectados entre si por linhas retas, e no caso da segunda simulação utilizou-se como símbolo

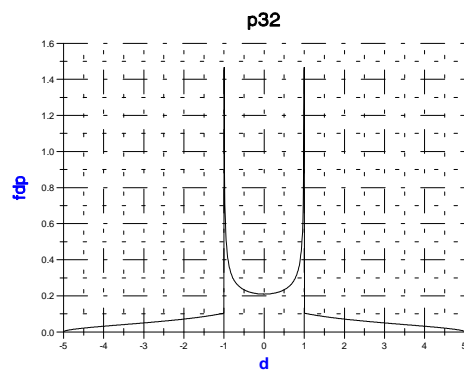
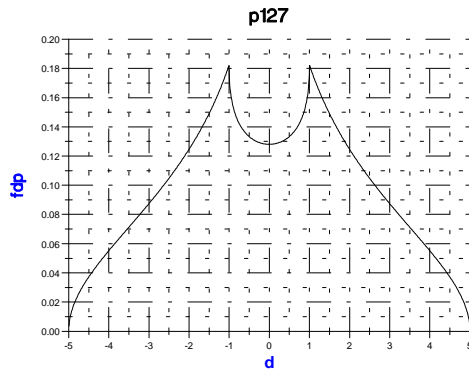
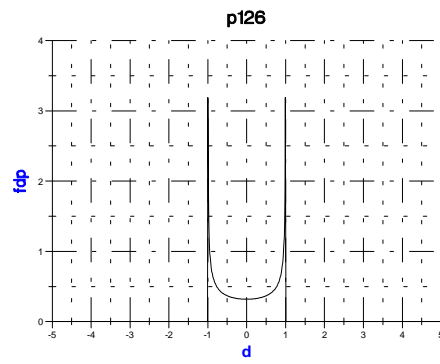
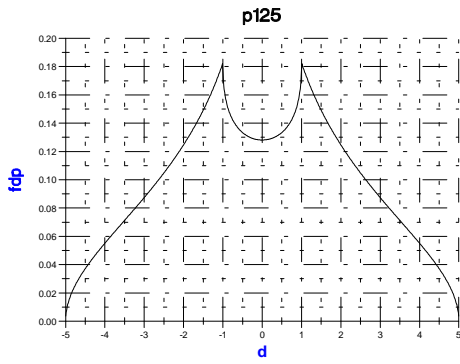
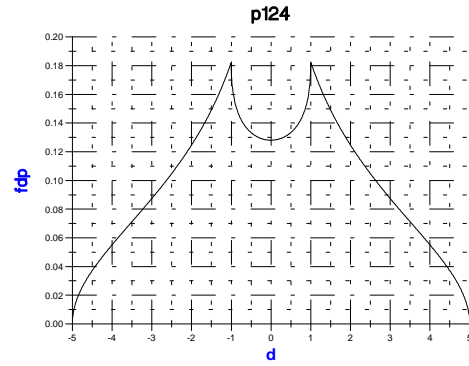
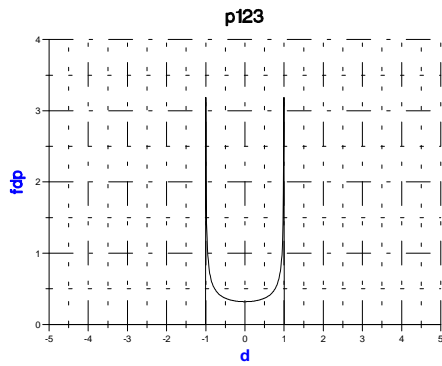
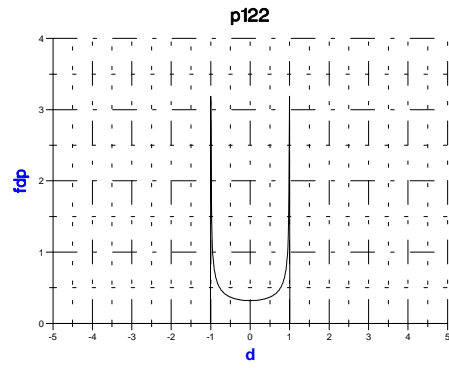
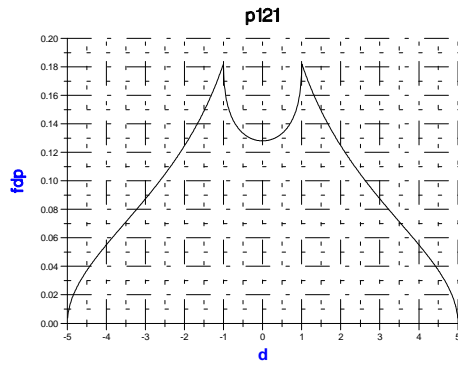


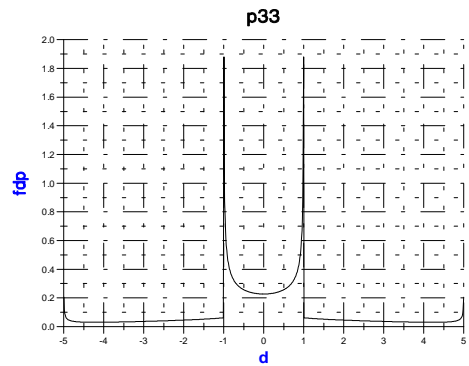
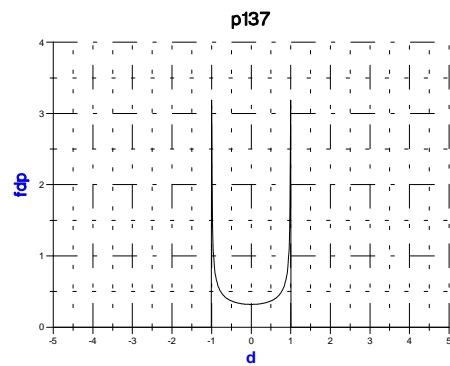
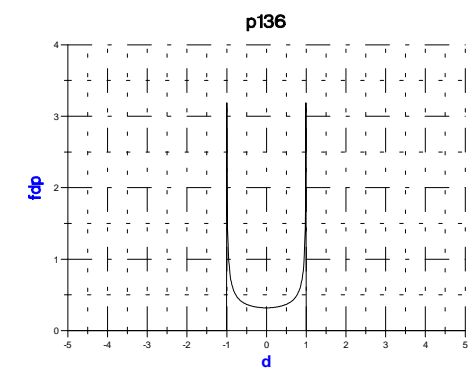
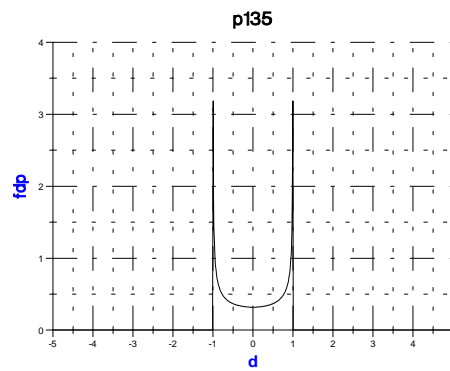
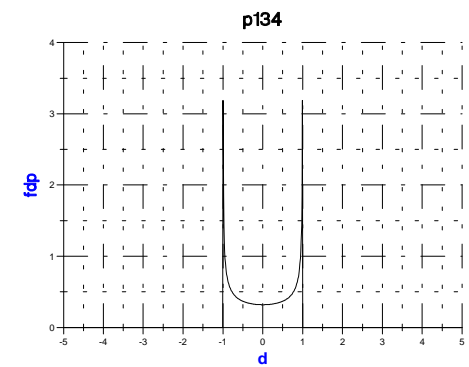
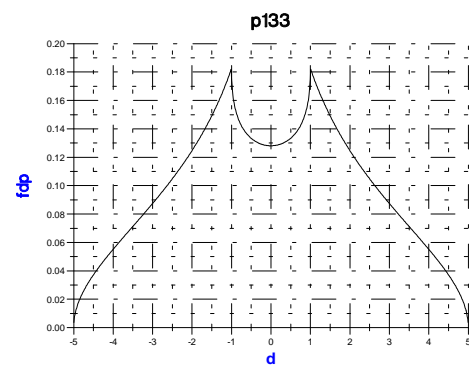
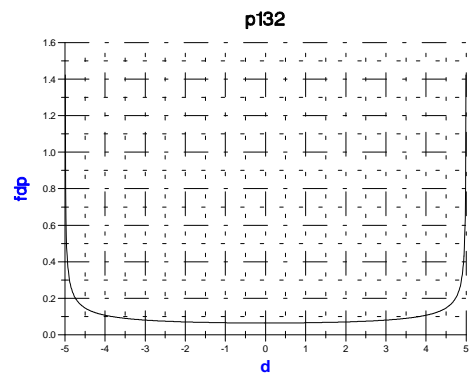
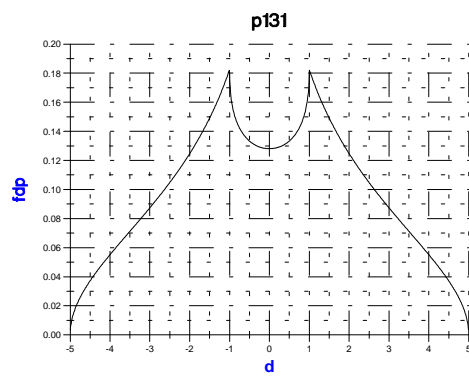


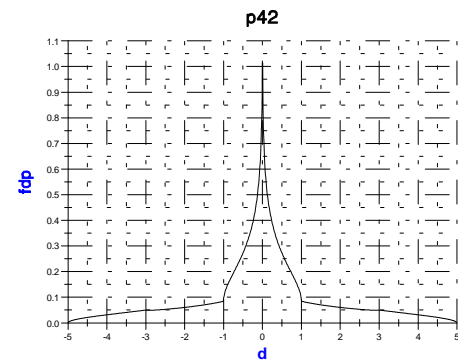
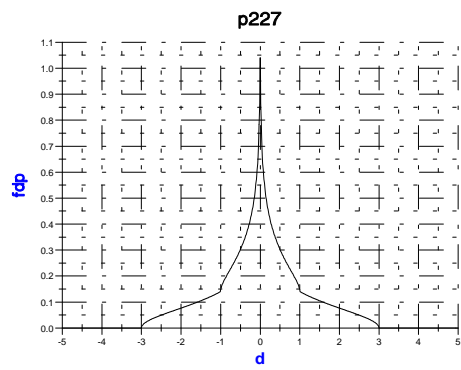
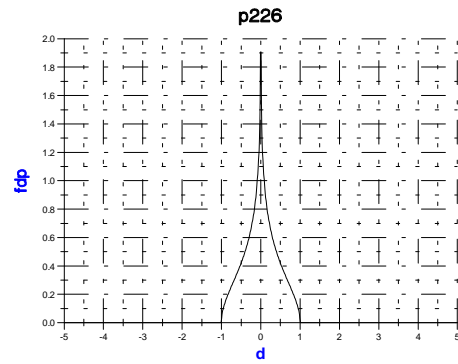
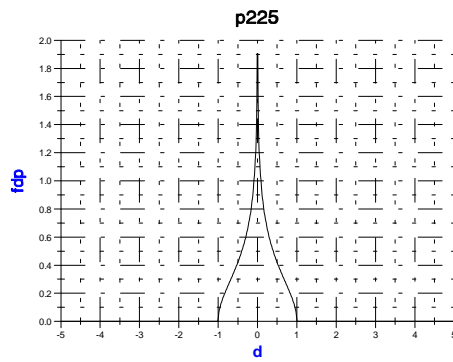
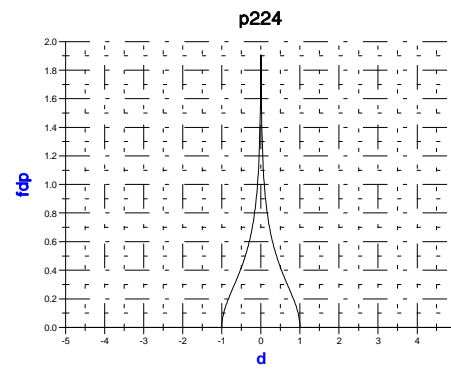
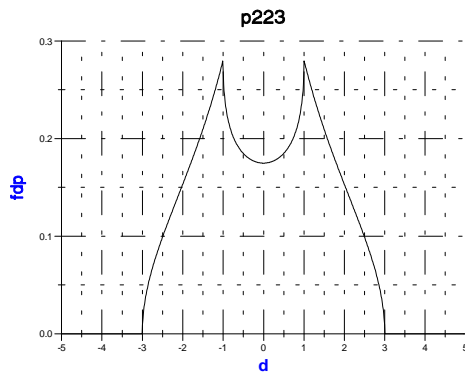
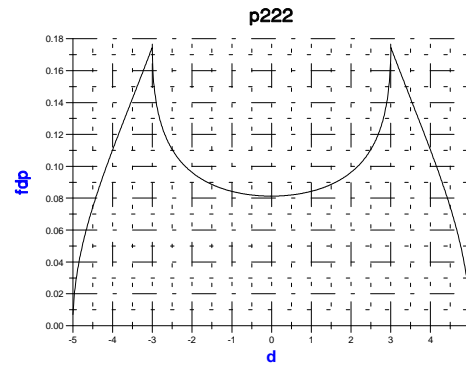
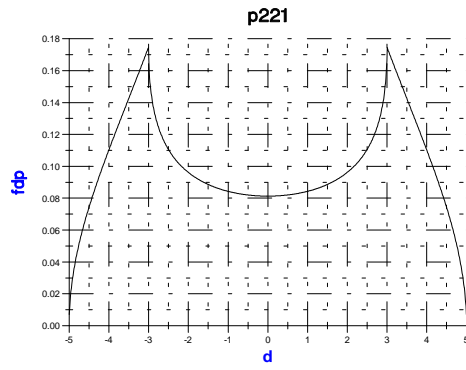


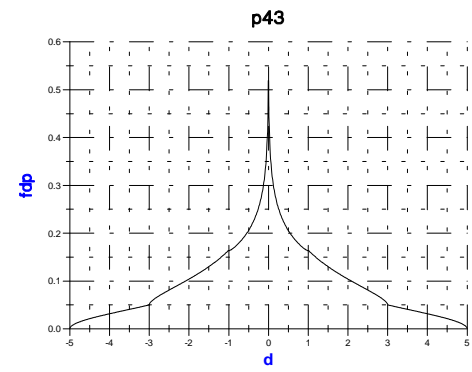
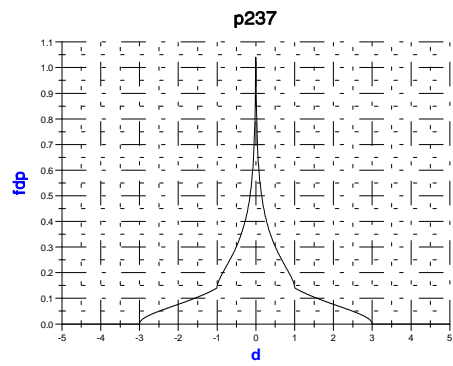
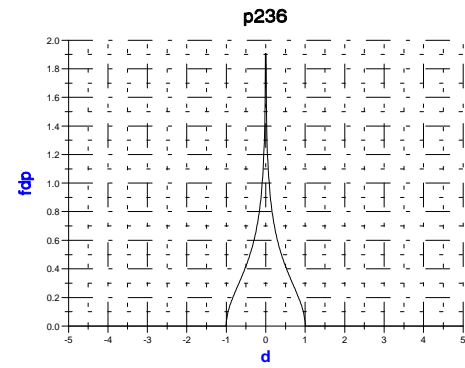
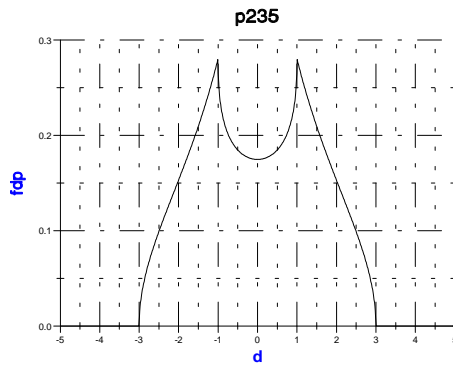
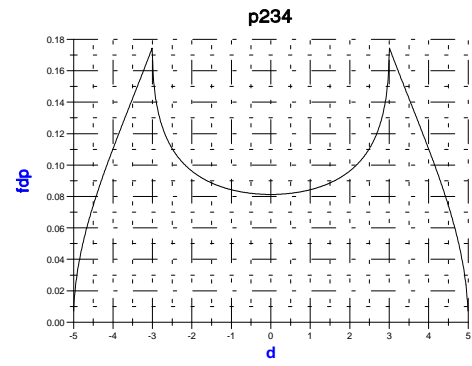
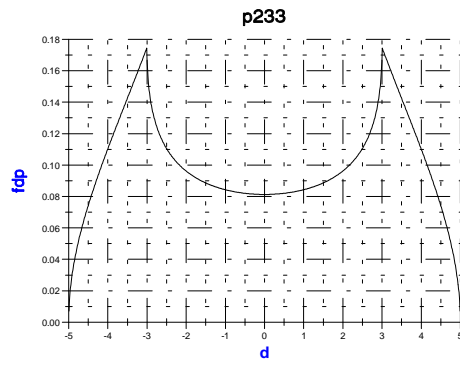
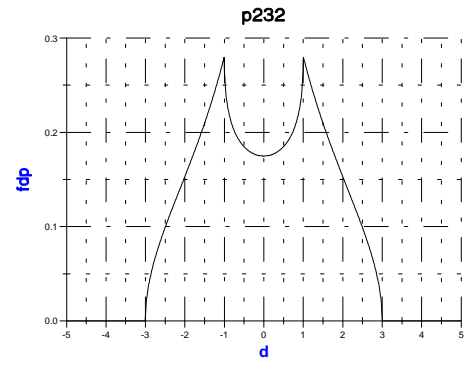
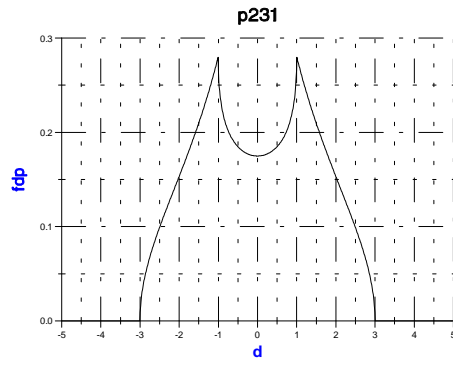


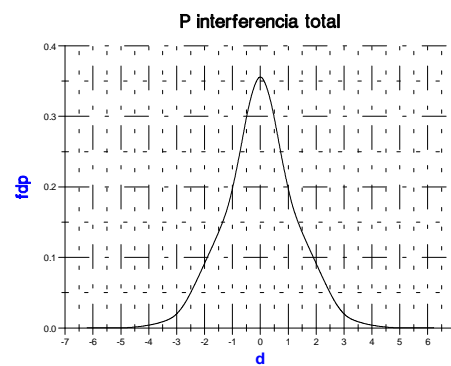
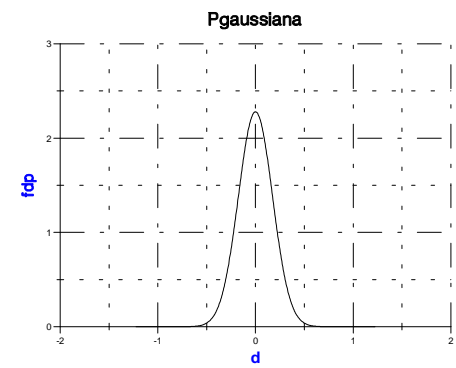
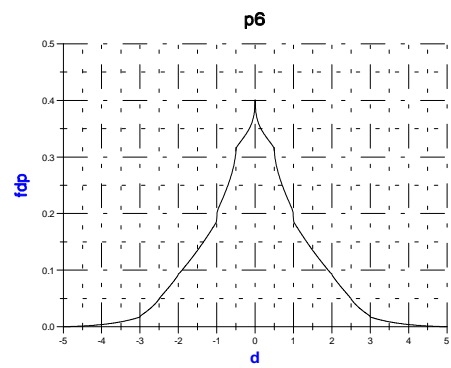
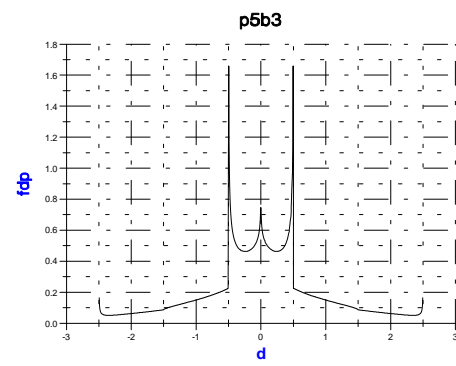
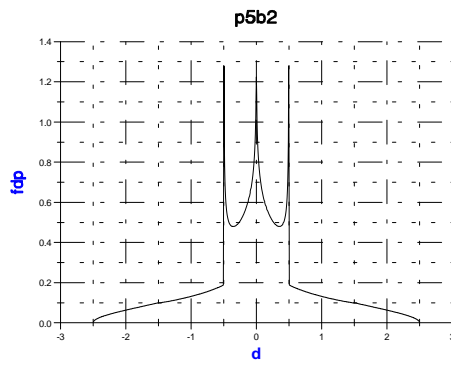
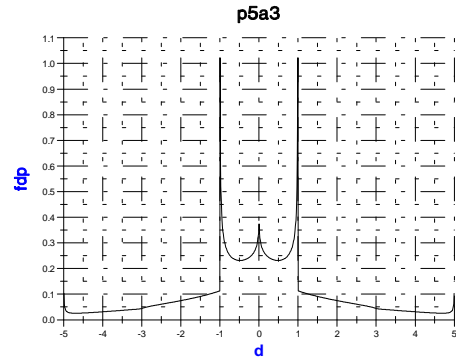
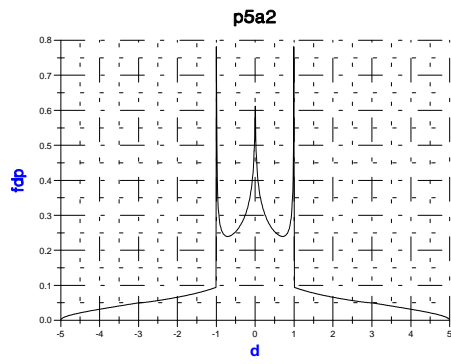












um ponto, mas devido à proximidade entre os pontos os mesmos tendem a aparecerem no gráfico como linhas. Na Tabela 1 encontram-se os valores da Probabilidade de Erro de Bit e as integridades obtidas durante as duas simulações. Nota-se que apesar da diferença de quase 2 décadas entre as quantidades de amostras efetuadas, as probabilidades de erro de bit obtidas encontram-se muito próximas entre si.

N	K	Amostras/ Unidade	$\theta \max_{i,x}$	Amostras $2.\theta \max_{i,x} \cdot$ Unidade	Integridade 3	Integridade 4	Probabilidade de erro de bit
7	3	2	5	20	1	0.8496	4.4437E-3
7	3	100	5	1000	1	0.9838	4.6107E-3

Tabela 1

12 - COMPARAÇÃO ENTRE OS DIVERSOS MÉTODOS DE CÁLCULO DA PROBABILIDADE DE ERRO DE BIT / CONCLUSÕES

A Tabela 2 contém os parâmetros e resultados das integridades obtidos durante as simulações pelo método de Nazari para vários comprimentos de seqüências e número de usuários.

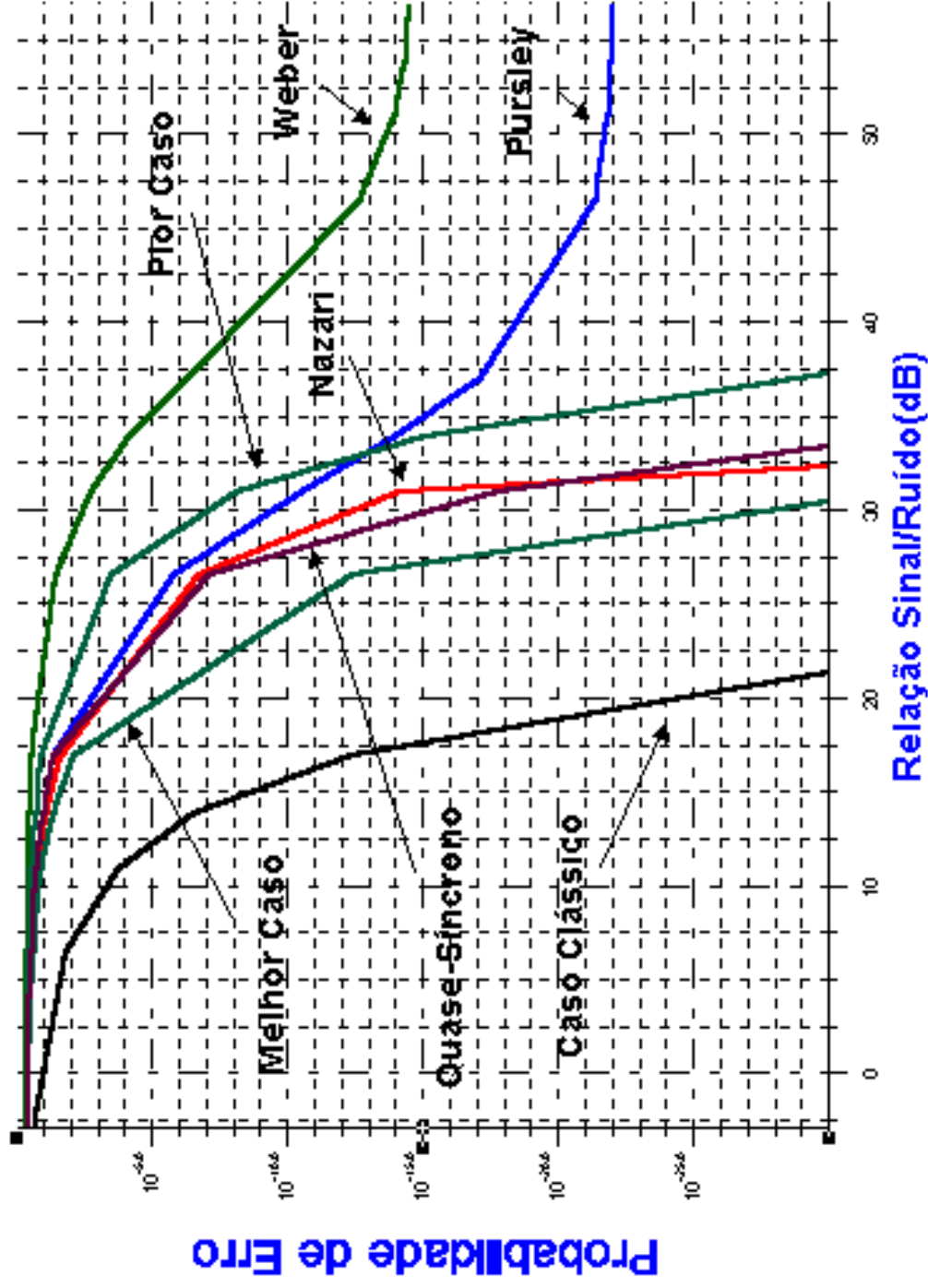
N	K	Amostras/ Unidade	$\theta \max_{i,x}$	Amostras $2.\theta \max_{i,x} \cdot$ Unidade	Integridade 3	Integridade 4
31	2	84	15	2520	1	0.9948
31	6	84	15	2520	1	0.9727
31	20	84	15	2520	1	0.8990
127	4	31	41	2542	1	0.9794
127	10	31	41	2542	1	0.9391
127	40	31	41	2542	1	0.7542
511	15	14	91	2548	1	0.8663
511	48	14	91	2548	1	0.6073

Tabela 2

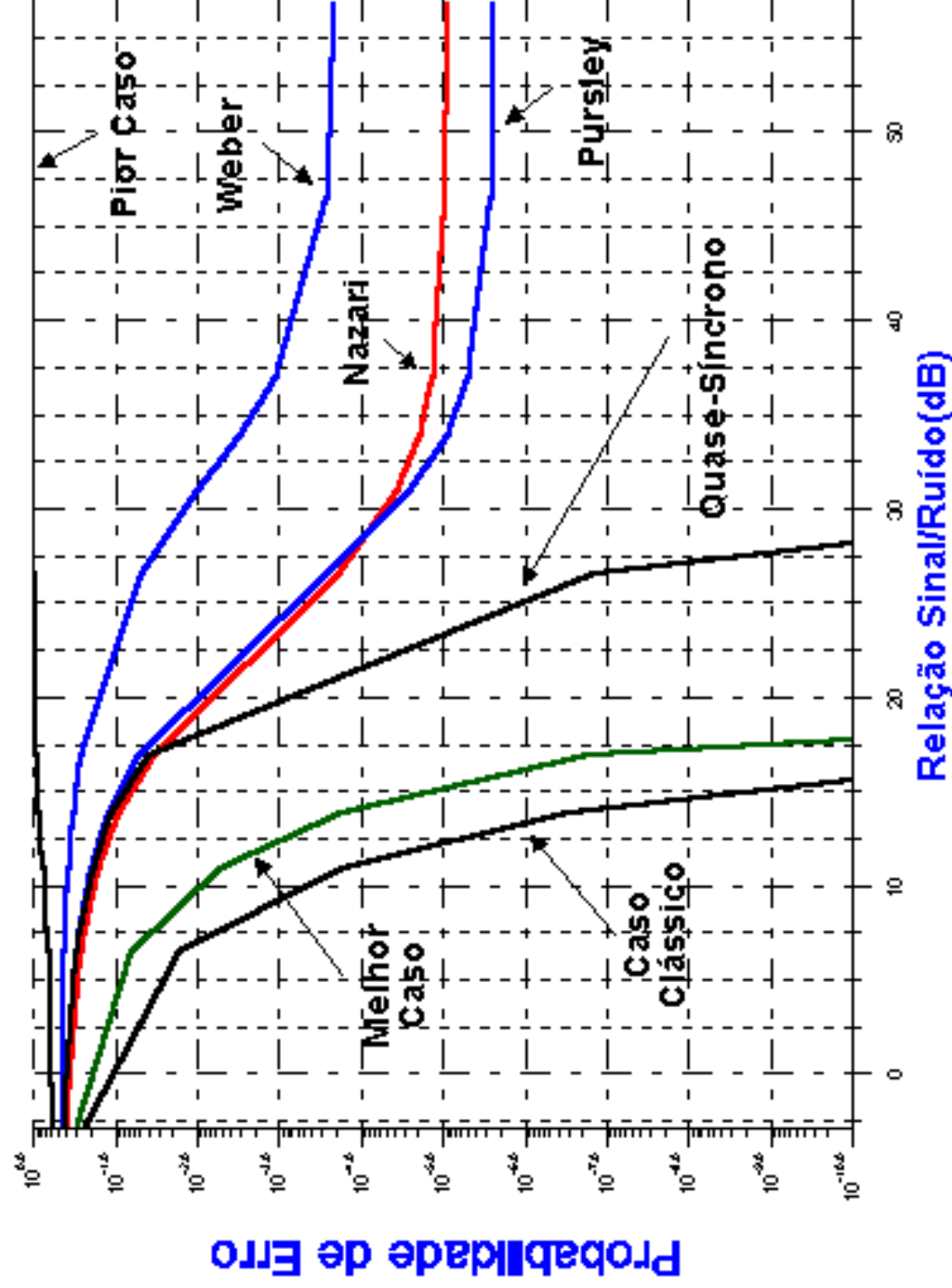
Os resultados obtidos para a probabilidade de erro de bit, nas várias situações ensaiadas, são apresentados nos gráficos das páginas (54) a (61).

Uma análise mais detalhada dos resultados evidencia que:

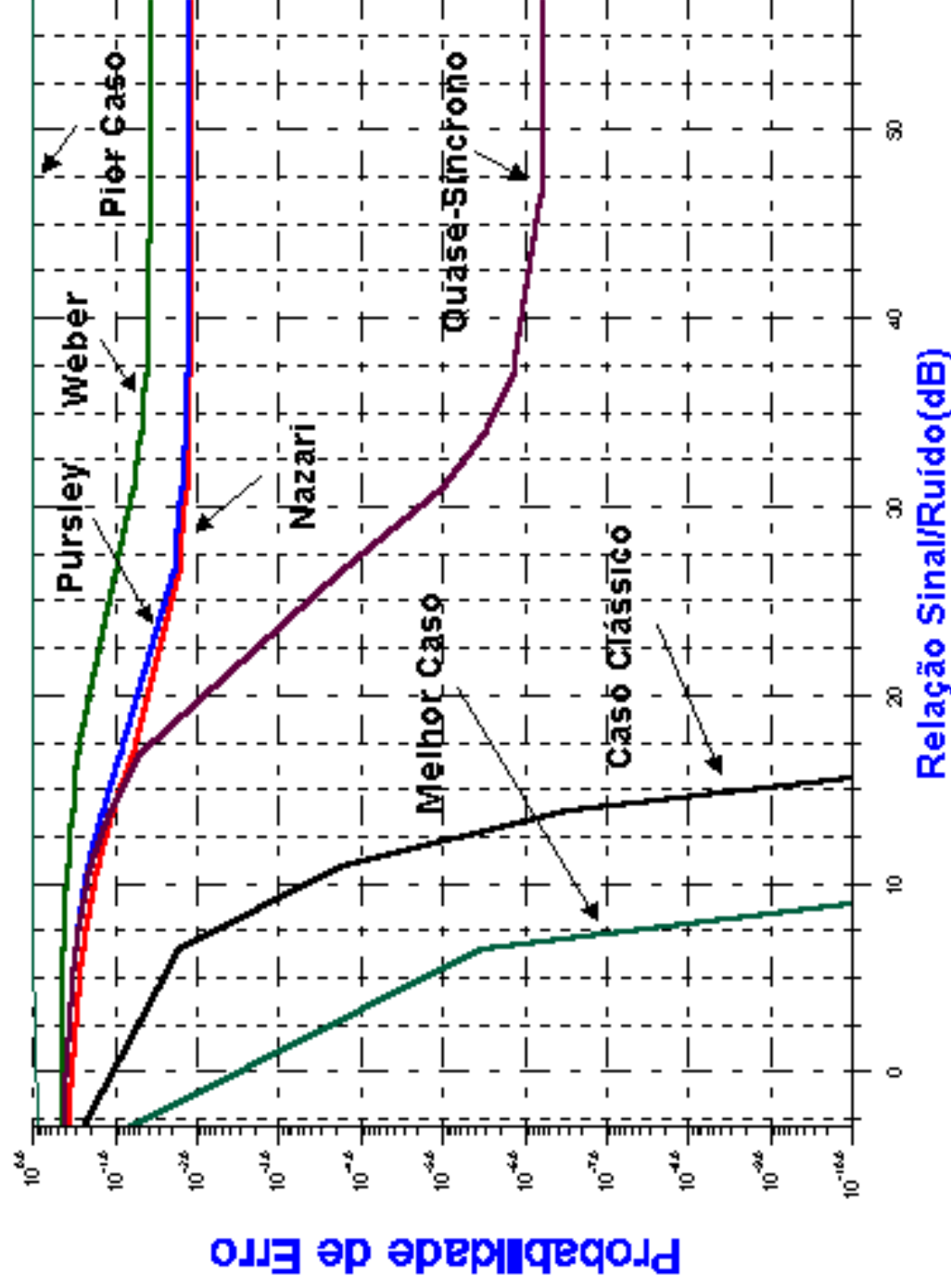
Caso: $N=31$, $K=2$



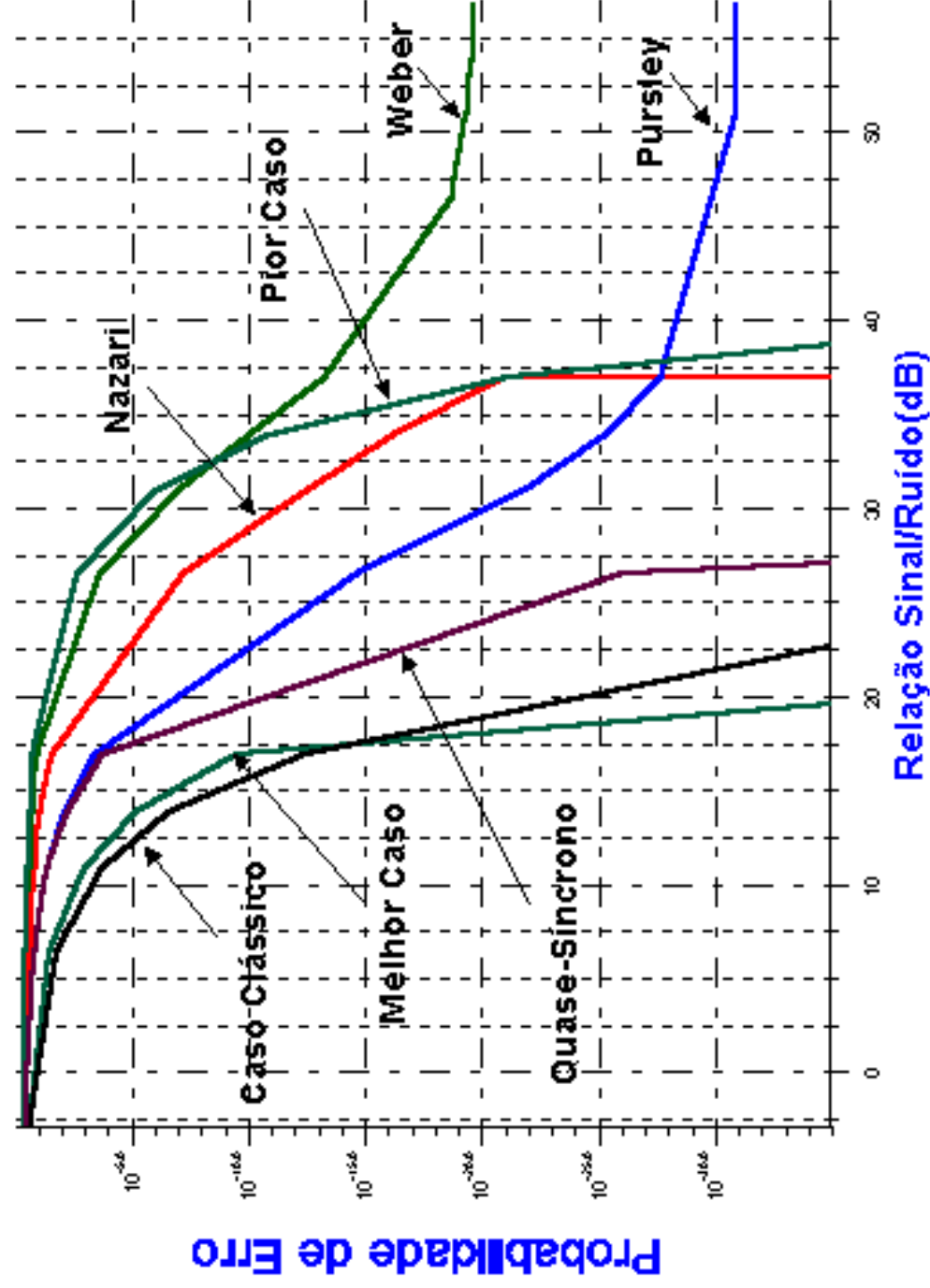
Caso: $N=31$, $K=6$



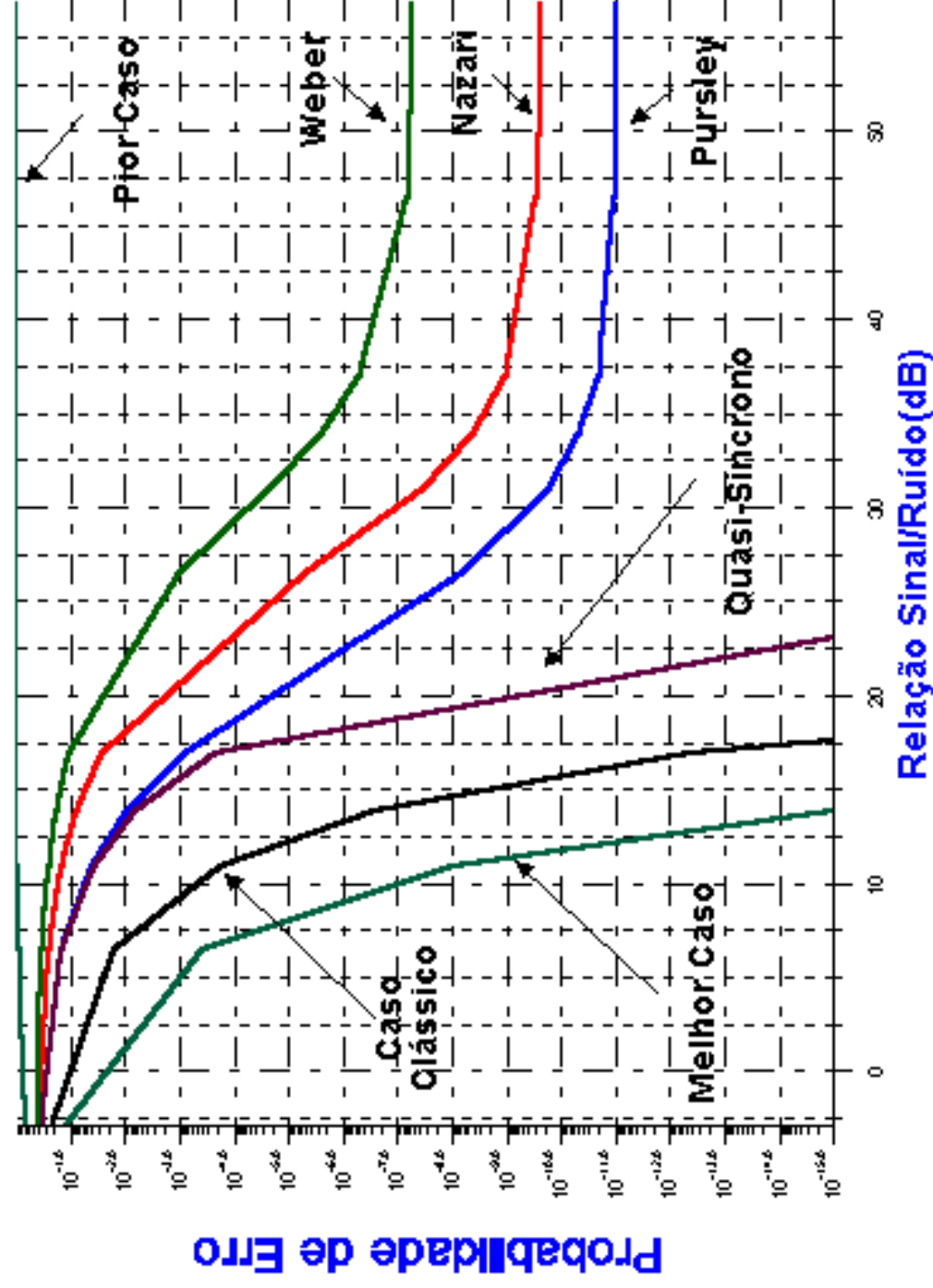
Caso: $N=31$, $K=20$



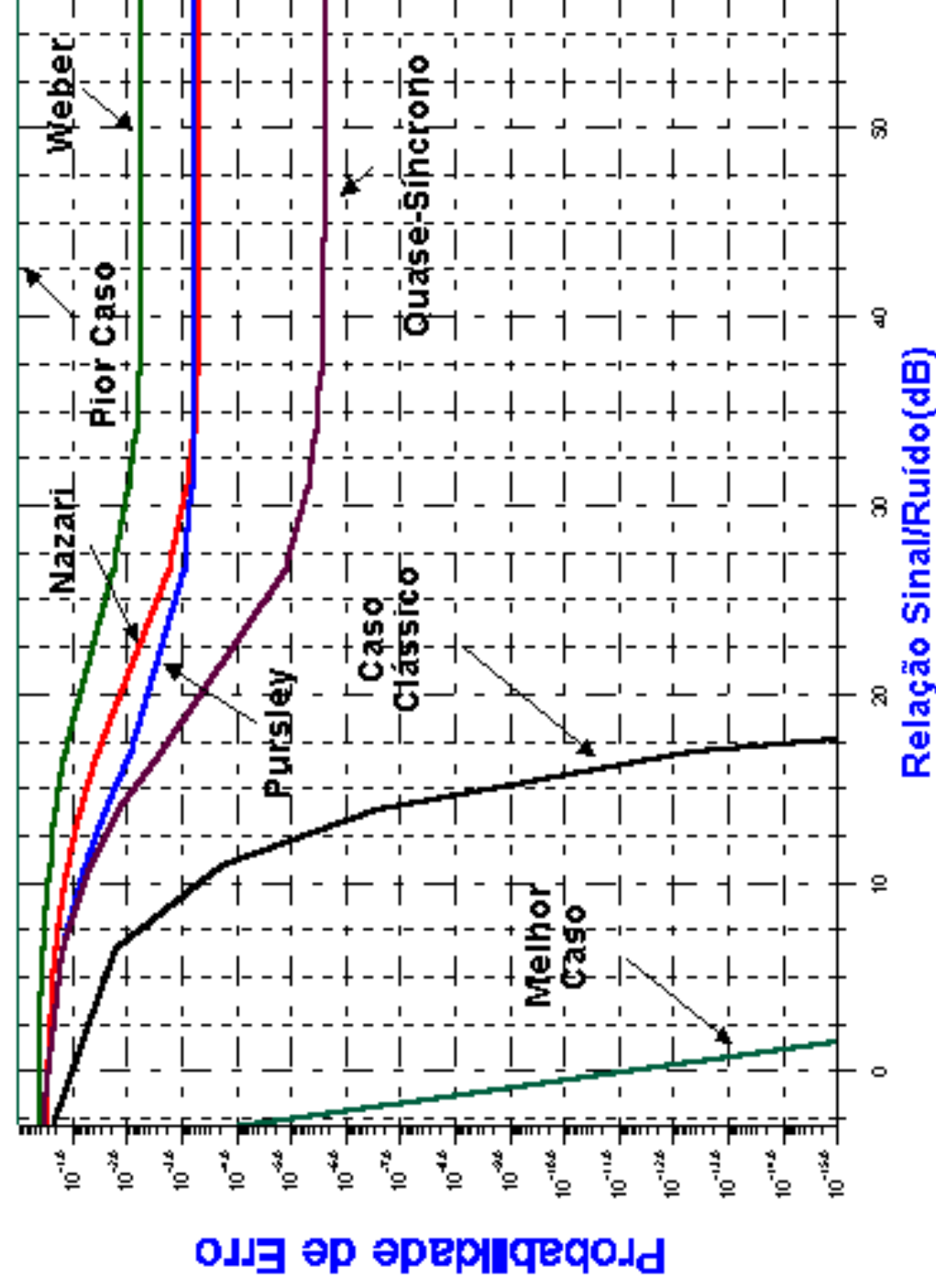
Caso: $N=127$, $K=4$



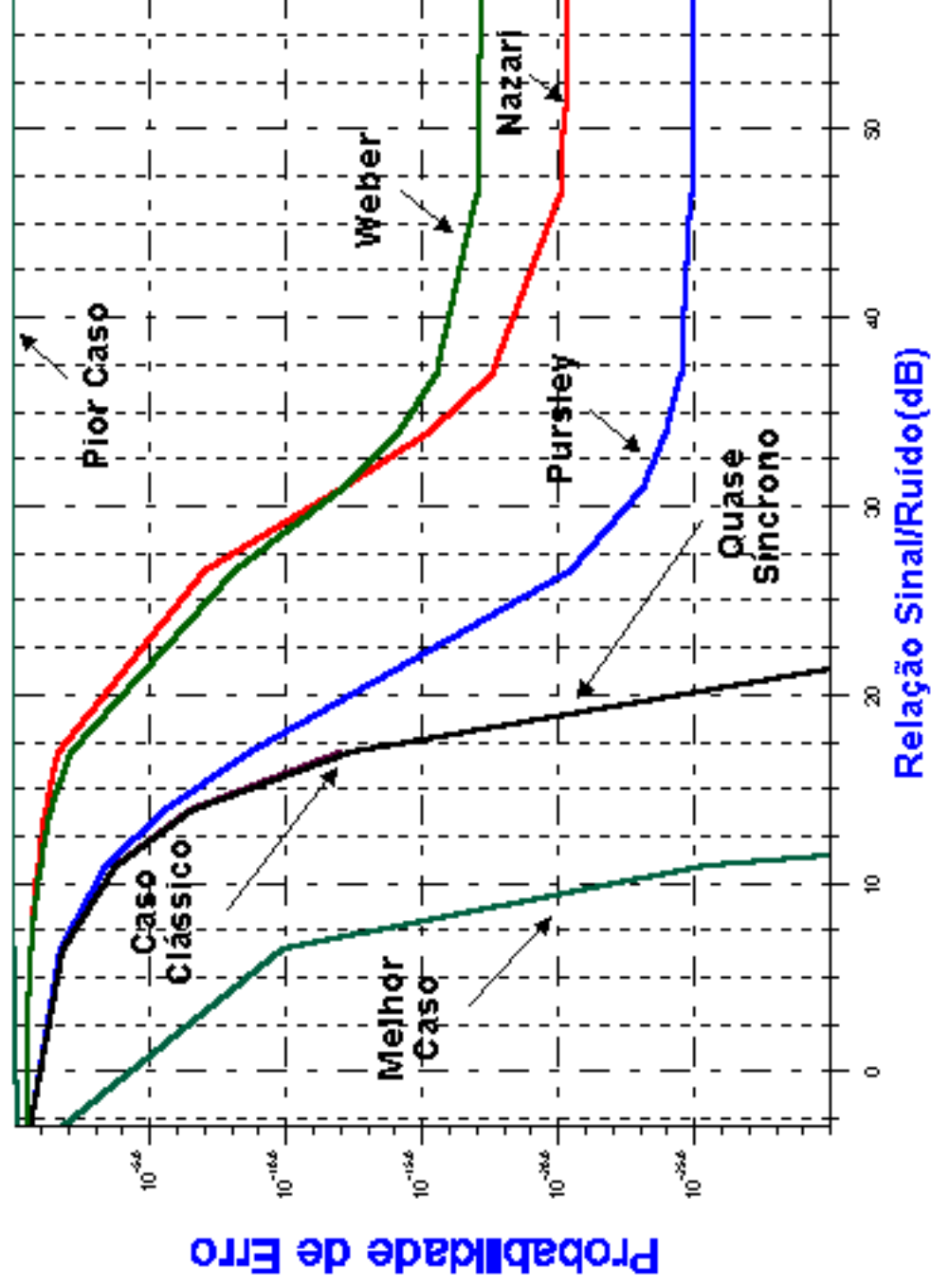
Caso: $N=127$, $K=10$



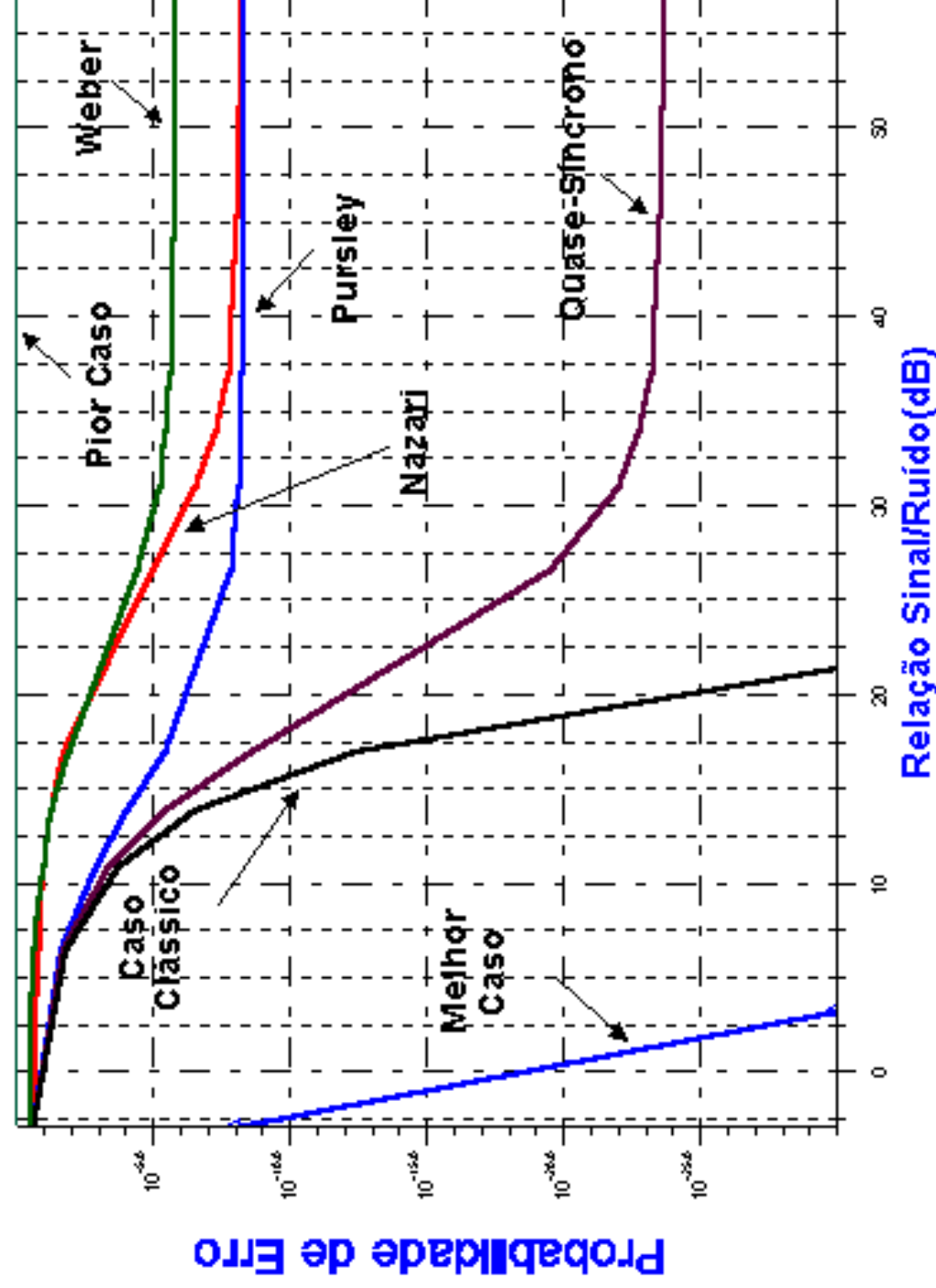
Caso: $N=127$, $K=40$



Caso: $N=511$, $K=15$



Caso: $N=511$, $K=48$



- o método de Weber é o mais conservativo em todas as situações ensaiadas neste trabalho e este fato sugere utilizá-lo em uma primeira abordagem pois alia a segurança à simplicidade dos cálculos.

- o método de Nazari é o mais preciso do ponto de vista teórico. Do ponto de vista computacional este método é também o mais complexo em termos de implementação e o que exige maior tempo de execução para obtenção de resultados. A necessidade de amostrar grande quantidade de funções densidade de probabilidade com descontinuidades cuja região de existência aumenta proporcionalmente ao período das seqüências tem como consequência direta introduzir um erro na área destas fdps (área esta que idealmente deveria ser unitária) que se propaga à medida que as convoluções (quantidade proporcional ao número de usuários) que fazem parte do método ocorrem; tal fato se evidencia através dos valores obtidos pela variável Integridade 4. Cálculos numéricos sugerem que, dentro de certos limites, os resultados do método de Nazari podem ser “desnormalizados” levando-se em conta esta integridade final resultante. A Figura 23 mostra o efeito desta propagação de erros para seqüências de comprimento $N=511$ e número de usuários $K=15$ e $K=48$ onde a Integridade 4 obtida possuía os valores **0.8663** e **0.6073**, respectivamente. Multiplicando-se as curvas pelo inverso de suas integridades, fator este que garante que a área da fdp da interferência total na saída do correlator possui área unitária, obtém-se os gráficos da Figura 24.

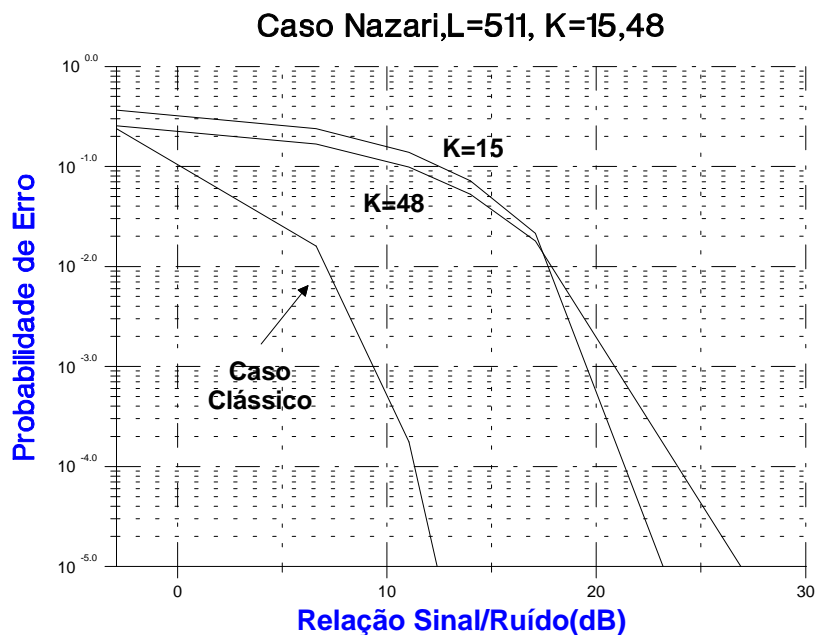


Figura 23

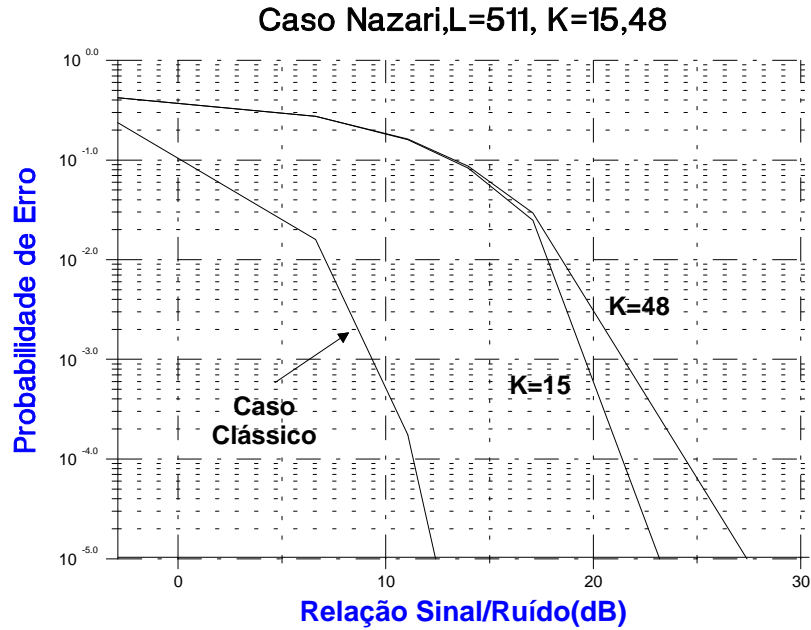


Figura 24

- o método de Pursley não é exato para um pequeno número de usuários por modelar a MUI como sendo uma variável aleatória gaussiana. Verifica-se nos gráficos obtidos que os resultados de Pursley e de Nazari se aproximam à medida em que se aumenta o número de usuários já que neste caso a aproximação gaussiana é mais apropriada.

- o estudo de pior e melhor casos só oferece resultados interessantes para um número pequeno de usuários; nos exemplos simulados em seqüências de comprimento $N = 31$ e número de usuários $K = 2$ e para seqüência de comprimento $N = 127$ e número de usuários $K = 4$. Nestes casos verifica-se inclusive a precisão teórica do método de Nazari pois as curvas obtidas são as únicas que encontram-se delimitadas pelas curvas de melhor e pior caso, que do ponto de vista teórico e computacional são absolutamente precisas. No restante dos casos simulados a faixa de valores que estas curvas delimitam é muito extensa, razão pela qual deixam de ter interesse de uso prático.

- o caso quase-síncrono, que segue literalmente a abordagem de Pursley, apresenta os melhores resultados em termos de performance, ou seja, menor probabilidade de erro de bit para determinada relação sinal/ruído. Isto é devido as seqüências estarem com atrasos controlados entre si e estas terem sido escolhidas com suas fases sub-otimizadas. Desta forma o resultado supera os demais não pelo método, mais sim pelo sistema empregado.

- o caso clássico $P_e = Q(\sqrt{2E_b/N_0})$ foi acrescentado em todos os gráficos apenas como um paradigma de comparação. Em qualquer um dos métodos de cálculo, considerando-se um único usuário este será o limite de desempenho esperado.

- os valores assintóticos da P_e do método de Weber e do de Pursley estão relacionados entre si pois:

Reescrevendo a equação (54):

$$P_e = Q\left[\left\{\frac{(K-1)}{3.N} + \frac{N_0}{A^2.T}\right\}^{\frac{1}{2}}\right] \quad (134)$$

Fazendo $A \rightarrow \infty$ obtém-se:

$$P_e = Q\left[\sqrt{\frac{3.N}{K-1}}\right] \quad (135)$$

Reescrevendo a equação (59):

$$P_e = Q\left[\sqrt{2.\left(\frac{E_b}{N_0}\right)_K}\right] = Q\left[\sqrt{2.\left(\frac{A^2.T}{N_0 + \sum_{i=1}^{K-1} A^2.T_c}\right)}\right] = Q\left[\sqrt{2.\left(\frac{T}{\frac{N_0}{A^2} + \sum_{i=1}^{K-1} T_c}\right)}\right] \quad (136)$$

Fazendo $A \rightarrow \infty$ obtém-se:

$$P_e = Q\left[\sqrt{2.\left(\frac{T}{(K-1).T_c}\right)}\right] = Q\left[\sqrt{\frac{2.N}{K-1}}\right] \quad (137)$$

Portanto, assintoticamente quando $A \rightarrow \infty$ existe uma diferença entre os valores obtidos entre os resultados de Nazari e Weber, pois dividindo-se os argumentos das equações (135) e (137) obtém-se:

$$\frac{\text{argumento de } Q(x) \text{ Pursley}}{\text{argumento de } Q(x) \text{ Weber}} = \sqrt{\frac{3}{2}} \quad (138)$$

Assim, por exemplo, no caso $L = 127$ e $N = 10$ tem-se pelo método de Pursley $P_e(\text{Pursley}) = 10^{-11}$. Consultando-se uma tabela de valores de $Q(x)$ tem-se $x_{\text{Pursley}} = 6.70$; assim $x_{\text{Weber}} = 6.70 \times \sqrt{3/2} = 5.47$ e novamente consultando uma tabela com valores de $Q(x)$ tem-se $P_e(\text{Weber}) = 2.2 \times 10^{-8}$, resultado bastante próximo do observado no gráfico.

- o fato de nas situações $N = 127$, $K = 4$ e $N = 31$, $K = 2$ alguns resultados serem inconsistentes com a abordagem de pior caso deve ser debitado a não aplicabilidade da aproximação gaussiana nestes casos, conforme já observado anteriormente.

13 - APÊNDICES

A - PROPRIEDADES DAS FUNÇÕES DE CORRELAÇÃO CRUZADA PARCIAL NO TEMPO:

Define-se função de correlação cruzada parcial no tempo par e ímpar respectivamente por:

$$\mathbf{R}_{i,x}(\tau) = \int_0^{\tau} \mathbf{c}_i(t - \tau) \cdot \mathbf{c}_x(t) \cdot dt \quad (139)$$

$$\hat{\mathbf{R}}_{i,x}(\tau) = \int_{\tau}^T \mathbf{c}_i(t - \tau) \cdot \mathbf{c}_x(t) \cdot dt \quad (140)$$

Reescrevendo as equações para o sinal da assinatura de uma forma genérica:

$$\mathbf{c}_i(t) = \sum_{j=0}^{N-1} \mathbf{c}_j^{(i)} \cdot \Pi(t - \mathbf{j} \cdot \mathbf{T}_c) \cdot \psi(t - \mathbf{j} \cdot \mathbf{T}_c) \quad (141)$$

Onde: $\Pi(t) = \begin{cases} 1 & \text{para } 0 \leq t \leq \mathbf{T}_c \\ 0 & \text{caso contrario} \end{cases}$ e $\psi(t)$ é a forma de onda do chip normalizado que satisfaz à relação $\mathbf{T}_c^{-1} \cdot \int_0^{\mathbf{T}_c} \psi^2(t) \cdot dt = 1$

Utilizando (141) a equação (140) pode ser reescrita como:

$$\hat{\mathbf{R}}_{i,x}(\tau) = \sum_{j=0}^{N-1-l} \mathbf{c}_j^{(i)} \cdot \mathbf{c}_{j+l}^{(x)} \cdot \int_{\tau}^{\mathbf{T}_c} \psi(t) \cdot \psi(t - \tau') \cdot dt + \sum_{j=0}^{N-1-l'} \mathbf{c}_j^{(i)} \cdot \mathbf{c}_{j+l'}^{(x)} \cdot \int_0^{\tau'} \psi(t) \cdot \psi(t + \mathbf{T}_c - \tau') \cdot dt \quad (142)$$

Onde:

l é a parte inteira de τ / \mathbf{T}_c representada matematicamente por $\lfloor \tau / \mathbf{T}_c \rfloor$ e $l' = l + 1$

Na equação (142) o primeiro termo encontra-se evidenciado na Figura (25) onde as regiões pontilhadas correspondem às regiões onde ocorre a multiplicação dos chips. Para estas regiões a multiplicação dos chips é evidenciada pela Figura (26). O segundo termo da equação (142) encontra-se evidenciado pela Figura (28) onde as regiões pontilhadas correspondem às regiões onde ocorre a multiplicação dos chips, como no caso anterior. Para estas regiões a multiplicação dos chips é evidenciada agora na Figura (27).

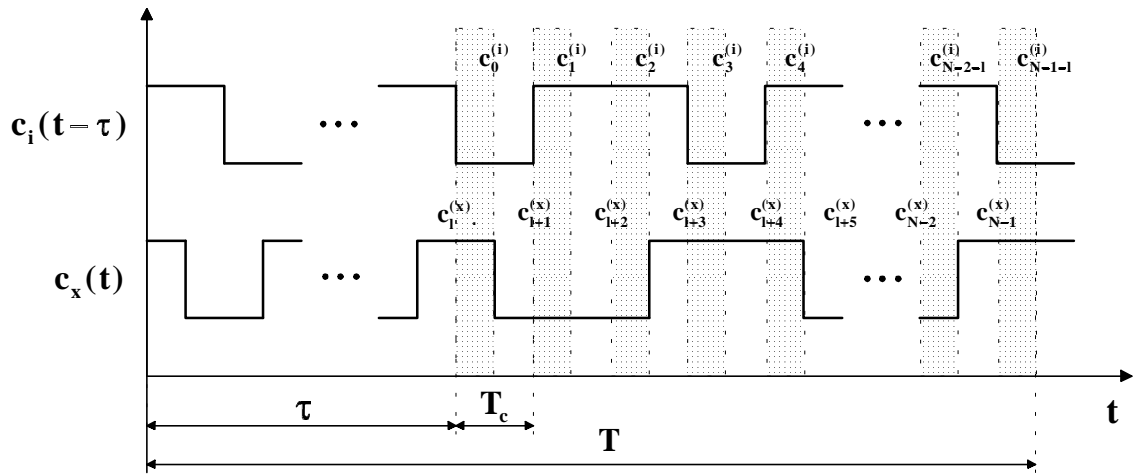


Figura 25

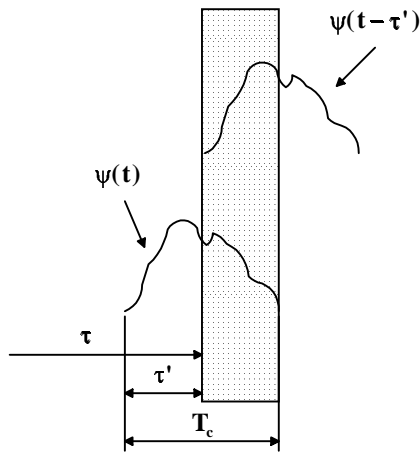


Figura 26

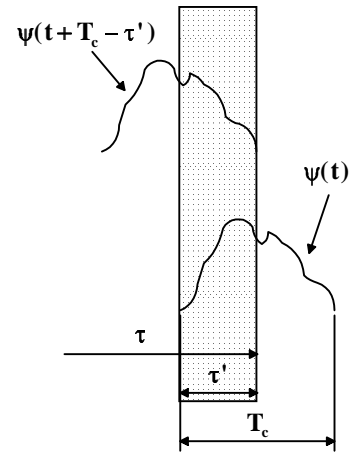


Figura 27

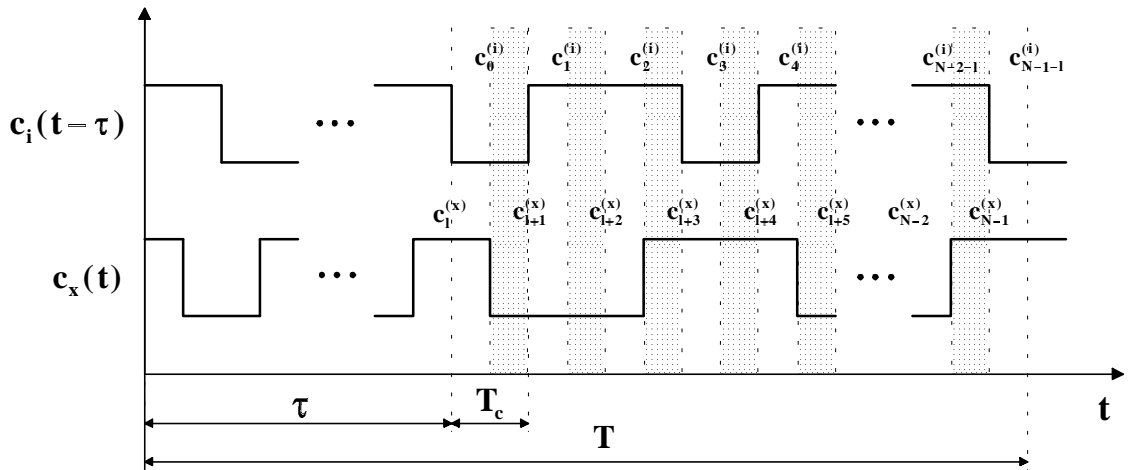


Figura 28

Definindo as funções:

$$\mathbf{R}_{\psi}(s) = \int_0^s \psi(t) \cdot \psi(t + T_c - s) \cdot dt \quad (143)$$

$$\hat{\mathbf{R}}_{\psi}(s) = \int_s^{T_c} \psi(t) \cdot \psi(t - s) \cdot dt \quad (144)$$

Onde:

$\mathbf{R}_{\psi}(s)$ é chamada de função de autocorrelação aperiódica do chip par.

$\hat{\mathbf{R}}_{\psi}(s)$ é chamada de função de autocorrelação aperiódica do chip ímpar.

A equação (142) pode ser reescrita como:

$$\hat{\mathbf{R}}_{i,x}(\tau) = C_{i,x}(l) \cdot \hat{\mathbf{R}}_{\psi}(\tau - l \cdot T_c) + C_{i,x}(l+1) \cdot \mathbf{R}_{\psi}(\tau - l \cdot T_c) \quad (145)$$

De forma análoga obtém-se:

$$\mathbf{R}_{i,x}(\tau) = C_{i,x}(l-N) \cdot \hat{\mathbf{R}}_{\psi}(\tau - l \cdot T_c) + C_{i,x}(l+1-N) \cdot \mathbf{R}_{\psi}(\tau - l \cdot T_c) \quad (146)$$

Para formas de onda quadrada:

$$\mathbf{R}_{\psi}(s) = s \quad (147)$$

$$\hat{\mathbf{R}}_{\psi}(s) = T_c - s \quad (148)$$

Substituindo (147) e (148) em (145) e (146), obtém-se:

$$\hat{\mathbf{R}}_{i,x}(\tau) = C_{i,x}(l) \cdot T_c + [C_{i,x}(l+1) - C_{i,x}(l)] \cdot (\tau - l \cdot T_c) \quad (149)$$

$$\mathbf{R}_{i,x}(\tau) = C_{i,x}(l-N) \cdot T_c + [C_{i,x}(l-N+1) + C_{i,x}(l-N)] \cdot (\tau - l \cdot T_c) \quad (150)$$

Claramente durante o intervalo de $[l \cdot T_c, (l+1) \cdot T_c]$ o termo $(\tau - l \cdot T_c)$ varia linearmente na faixa de valores $[0, T_c]$ para quaisquer valores de l e τ . Portanto as equações (149) e (150) são cada uma nestes intervalos resultados da soma de uma constante por uma função linear que varia no intervalo $[0, (C_{i,x}(l+1) - C_{i,x}(l)) \cdot T_c]$ ou $[0, (C_{i,x}(l-N+1) - C_{i,x}(l-N)) \cdot T_c]$, para o caso das funções $\hat{\mathbf{R}}_{i,x}(\tau)$ e $\mathbf{R}_{i,x}(\tau)$, respectivamente. Desta constatação pode-se

concluir que os máximos e mínimos locais para o caso em que a forma de onda do chip tem a forma de onda quadrada ocorrem em valores de τ que são múltiplos inteiros de T_c .

A escolha da forma de onda quadrada em todo o trabalho se justifica pelo uso desta em toda a literatura, bem como na maior parte das implementações práticas. O seu uso implica na ocupação de uma faixa ampla do espectro de frequências que dependendo da aplicação pode ser indesejável. Neste caso pode-se utilizar outras formas de onda para o chip da assinatura, como por exemplo, as formas de onda de Nyquist com a qual consegue-se uma limitação da faixa do espectro de frequências utilizado.

B - SEQÜÊNCIAS PSEUDO-ALEATÓRIAS LINEARES

As seqüências pseudo-aleatórias lineares são importantes pois podem ser facilmente geradas e possuem algumas propriedades desejáveis idênticas à das seqüências aleatórias de interesse, vide ref. [10].

Propriedades de seqüências pseudo-aleatórias:

Dentre as seqüências pseudo-aleatórias destaca-se no uso prático as SMC- Seqüências de Máximo Comprimento, por possuírem uma teoria bem desenvolvida e propriedades convenientes. As SMC são implementáveis com registradores de deslocamento, genericamente como indicado abaixo:

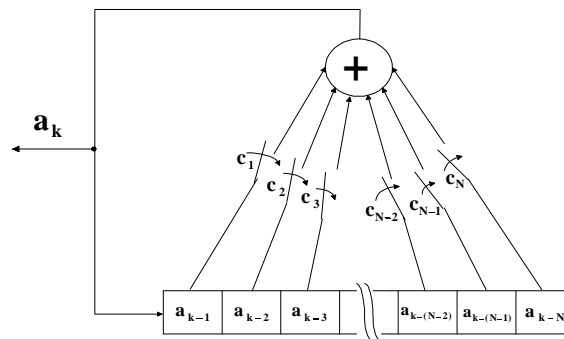


Figura 29 - Implementação genérica de um registrador de deslocamento

Da Figura 29:
$$a_k = \sum_{i=1}^N a_{k-i} \cdot c_i$$

Definindo a função geradora como sendo a saída no tempo, pode-se escrever:

$$G(x) = a_0 \cdot x + a_1 \cdot x^2 + a_2 \cdot x^3 + \dots = \sum_{k=0}^{\infty} a_k \cdot x^k = \{a_0, a_1, a_2, \dots\}$$

Onde $\mathbf{a}_k \cdot \mathbf{x}^k$ denota que a saída no k-ésimo intervalo de tempo é dada por $\mathbf{a}_k \in \{0,1\}$

E assim:

$$\mathbf{G}(\mathbf{x}) = \sum_{k=0}^{\infty} \mathbf{a}_k \cdot \mathbf{x}^k = \sum_{k=0}^{\infty} \sum_{i=1}^N \mathbf{c}_i \mathbf{a}_{k-i} \cdot \mathbf{x}^{k-i} = \sum_{i=1}^N \mathbf{c}_i \cdot \mathbf{x}^i \cdot \sum_{k=0}^{\infty} \mathbf{a}_{k-i} \cdot \mathbf{x}^{k-i} = \sum_{i=1}^{\infty} \mathbf{c}_i \cdot \mathbf{x}^i \cdot \left[\mathbf{a}_{-i} \cdot \mathbf{x}^{-i} + \mathbf{a}_{-i+1} \cdot \mathbf{x}^{-i+1} + \dots + \mathbf{a}_{-1} \cdot \mathbf{x}^{-1} + \mathbf{G}(\mathbf{x}) \right]$$

$$\mathbf{G}(\mathbf{x}) = \frac{\sum_{i=1}^N \mathbf{c}_i \cdot \mathbf{x}^i \cdot (\mathbf{a}_{-i} \cdot \mathbf{x}^{-i} + \mathbf{a}_{-i+1} \cdot \mathbf{x}^{-i+1} + \dots + \mathbf{a}_{-1} \cdot \mathbf{x}^{-1})}{\sum_{i=1}^N \mathbf{c}_i \cdot \mathbf{x}^i} = \frac{\mathbf{g}(\mathbf{x})}{\mathbf{f}(\mathbf{x})}$$

Na equação acima nota-se que o numerador da função geradora depende das condições iniciais dos registradores de deslocamento e o denominador de sua realização física; $\mathbf{f}(\mathbf{x})$ é denominado de polinômio, ou função, característico da implementação.

O grau do numerador da função geradora é pelo menos uma vez menor que o grau do denominador, isto é:

$$\deg\{\text{num}[\mathbf{G}(\mathbf{x})]\} < \deg\{\text{den}[\mathbf{G}(\mathbf{x})]\}$$

Supondo que as condições iniciais são $\mathbf{a}_{-N} = 1$ e $\mathbf{a}_{-N+1} = \mathbf{a}_{-N+2} = \dots = \mathbf{a}_{-1} = 0$ resulta $\mathbf{g}(\mathbf{x}) = 1$ e portanto:

$$\mathbf{G}(\mathbf{x}) = \frac{1}{\mathbf{f}(\mathbf{x})}$$

A divisão longa indicada resulta na sequência de saída do registrador de deslocamento.

Exemplo:

Considerando a Figura 30:

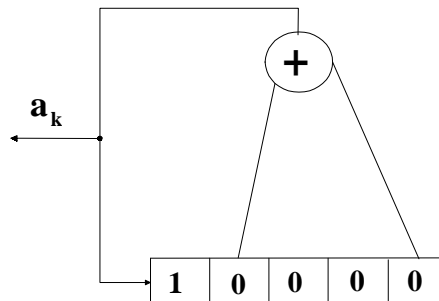


Figura 30

Observa-se que:

$$\mathbf{c}_1 = \mathbf{c}_3 = \mathbf{c}_4 = \mathbf{0} \text{ e } \mathbf{c}_2 = \mathbf{c}_5 = \mathbf{1}$$

$$\mathbf{a}_{-5} = \mathbf{a}_{-4} = \mathbf{a}_{-3} = \mathbf{a}_{-2} = \mathbf{0} \text{ e } \mathbf{a}_{-1} = \mathbf{1}$$

$$\mathbf{a}_k = \mathbf{a}_{k-2} + \mathbf{a}_{k-5}$$

$$\mathbf{G}(\mathbf{x}) = \frac{1}{\mathbf{x}^5 + \mathbf{x}^2 + 1}, \quad \text{pois } \mathbf{g}(\mathbf{x}) = 1 \text{ e } \mathbf{f}(\mathbf{x}) = \mathbf{x}^5 + \mathbf{x}^2 + 1.$$

Realizando a divisão longa obtém-se a saída no tempo:

$$\begin{array}{r} 1 \\ \underline{1 + \mathbf{x}^2 + \mathbf{x}^5} \\ \mathbf{x}^2 + \mathbf{x}^5 \\ \underline{\mathbf{x}^2 + \mathbf{x}^4 + \mathbf{x}^7} \\ \mathbf{x}^4 + \mathbf{x}^5 + \mathbf{x}^7 \\ \underline{\mathbf{x}^4 + \mathbf{x}^6 + \mathbf{x}^9} \\ \mathbf{x}^5 + \mathbf{x}^6 + \mathbf{x}^7 + \mathbf{x}^9 \\ \underline{\mathbf{x}^5 + \mathbf{x}^7 + \mathbf{x}^{10}} \\ \mathbf{x}^6 + \mathbf{x}^9 + \mathbf{x}^{10} \\ \underline{\mathbf{x}^6 + \mathbf{x}^8 + \mathbf{x}^{11}} \end{array} \quad \begin{array}{r} \underline{1 + \mathbf{x}^2 + \mathbf{x}^5} \\ 1 + \mathbf{x}^2 + \mathbf{x}^4 + \mathbf{x}^5 + \mathbf{x}^6 + \dots \end{array}$$

Seqüência de saída: $\{1, 0, 1, 0, 1, 1, 1, \dots\}$.

Dizimação de seqüências:

Dada uma seqüência \mathbf{a} e sendo \mathbf{i} a ordem de um elemento desta, representado por \mathbf{a}_i , a operação $\{\mathbf{a}_{r_i}\}$ indica uma dizimação \mathbf{r} da seqüência \mathbf{a} . Esta operação gera uma outra seqüência composta dos elementos $\{\mathbf{a}_r, \mathbf{a}_{r.2}, \mathbf{a}_{r.3}, \mathbf{a}_{r.4}, \mathbf{a}_{r.5}, \dots\}$.

Seqüências de Máximo Comprimento (SMC):

O número máximo de combinações para as realimentações dos conteúdos de um registrador de deslocamento de dimensão \mathbf{N} é $2^{\mathbf{N}}$; portanto, devido à saída ser resultante de uma função linear deste conteúdo o comprimento (período) máximo de uma seqüência produzida pelo sistema da Figura 29 seria $2^{\mathbf{N}}$. Entretanto o caso em que o conteúdo de todos os estágios é 0 não é considerado uma vez que para a configuração proposta o registrador de deslocamento permaneceria indefinidamente neste estado. Portanto, o comprimento da maior seqüência que pode ser obtida utilizando a configuração proposta é $\mathbf{p} = 2^{\mathbf{N}} - 1$. A estas dá-se o nome de seqüências de máximo comprimento.

Relação entre a função geradora e o período da seqüência:

Teorema 1: Se a seqüência gerada por $G(x) = \frac{g(x)}{f(x)}$ tem período p com $a_{-N} = 1$ e $a_{-N+1} = a_{-N+2} = \dots = a_{-1} = 0$ então $f(x)$ divide $1 + x^p$.

Para a seqüência ter período p a equação abaixo tem que ser válida:

$$\begin{aligned} \frac{1}{f(x)} &= a_0 + a_1 \cdot x^1 + \dots + a_{p-1} \cdot x^{p-1} + x^p \cdot (a_0 + a_1 \cdot x^1 + \dots + a_{p-1} \cdot x^{p-1}) + \dots \\ &= (a_0 + a_1 \cdot x^1 + \dots + a_{p-1} \cdot x^{p-1}) \cdot (1 + x^p + x^{2p} + \dots) \end{aligned}$$

$$\text{Como: } (1 + x^p + x^{2p} + \dots) = \frac{1}{(1 + x^p)}$$

Resulta:

$$\frac{1}{f(x)} = \frac{a_0 + a_1 \cdot x^1 + \dots + a_{p-1} \cdot x^{p-1}}{1 + x^p} \longrightarrow \frac{1 + x^p}{f(x)} = a_0 + a_1 \cdot x^1 + \dots + a_{p-1} \cdot x^{p-1}$$

O teorema acima se mantém para um $g(x)$ qualquer desde que o numerador não tenha fatores em comum com $f(x)$, pois se houver, haverá um cancelamento de termos, fazendo com que a seqüência gerada tenha um período menor que p . Dependendo do termo cancelado, pode-se ter seqüências de períodos diferentes, conforme o estado inicial do registrador de deslocamento. Os polinômios característicos que geram SMC são irredutíveis, isto é, não podem ser fatorados. Como exemplo de um polinômio irredutível tem-se o polinômio obtido da figura 30, $f(x) = x^5 + x^2 + 1$, pois este não pode ser fatorado e como um exemplo de polinômio redutível tem-se o polinômio $f(x) = x^5 + x + 1 = (1 + x + x^2) \cdot (1 + x^2 + x^3)$. Claramente este polinômio gera seqüências de comprimento 2, 3 e 5, para as condições iniciais dadas pelos polinômios $g(x) = 1 + x^2 + x^3$, $g(x) = 1 + x + x^2$ e $g(x) = 1$ respectivamente.

O fato do polinômio característico ser irredutível não assegura que este gere uma SMC. Para que o polinômio característico irredutível gere uma SMC ele deve também ser primitivo, isto é, não pode dividir $(1 + x^m)$ para todo $m < 2^n - 1$, caso contrário, pode-se ter seqüências de comprimento m , conforme o Teorema 1.

Pode-se finalmente afirmar:

" Todo polinômio primitivo de grau n gera uma SMC de comprimento $2^n - 1$ "

Teorema 2: Se $p = 2^n - 1$ é primo, todo polinômio irreduzível de grau n gera uma SMC.

Teorema 3: O recíproco de um polinômio irreduzível é um polinômio irreduzível.

Para se obter o polinômio recíproco $f_r(x)$ do polinômio $f(x)$ basta calcular:

$$f_r(x) = x^n \cdot \frac{1}{f(x)}$$

Teorema 4: Qualquer sequência que é a soma em **mod 2** das sequências geradas pelas funções características $f_1(x)$ e $f_2(x)$ pode ser gerada por um registrador de deslocamento de função característica $f(x) = f_1(x) \cdot f_2(x)$.

Propriedades das sequências de máximo comprimento:

Dentre as propriedades que as SMC possuem destacam-se:

- Em um período da sequência, o número de '1's excede o número de '0's de uma unidade. Esta propriedade é denominada de balanceamento.
- Em um período da sequência, $1/2$ dos agrupamentos de bits de mesmo valor tem comprimento 1, $1/4$ tem comprimento 2, $1/8$ tem comprimento 3 e assim por diante.
- A função de autocorrelação periódica $\theta_{x,x}(l)$ possui os seguintes valores:

$$\theta_{x,x}(l) = \begin{cases} N & l = 0, N, 2N, \dots \\ -1 & \text{caso contrario} \end{cases}$$

Onde:

$$\theta_{a,b}(l) = \sum_{i=0}^{N-1} a_i \cdot b_{i+l}$$

Assim a função de autocorrelação das SMCs apresenta-se como uma característica favorável, facilitando sincronização. Infelizmente as SMCs não possuem boas características de correlação cruzada. O obtenção de SMCs se justifica pois dentro de uma família de sequências se encontram pares de sequências denominados de “pares preferenciais” que possibilitam a geração de sequências de Gold com boas propriedades de autocorrelação e propriedades de correlação cruzada controláveis.

Exemplo das funções de autocorrelação e correlação cruzada de duas SMC:

O programa abaixo, desenvolvido no MATLAB V4.2c for WINDOWS calcula a correlação cruzada de duas seqüências:

```
function saída=corcsmc(seqüencia1,seqüencia2)
% function saída=corcsmc(seqüencia1,seqüencia2)
% Calcula a função de correlação cruzada entre duas seqüências de máximo
% comprimento(isto e' importante pois a rotina se utiliza da propriedade de
% periodicidade da seqüência para o cálculo) durante um determinado interva-
% lo. As duas seqüências fornecidas devem ter o mesmo período. O cálculo é
% efetuado somente em um período.

% Verifica o período das seqüências
comprimento=length(sequencia1);
intervalo=comprimento;

% Transforma cada vetor no equivalente bipolar
for i=1:comprimento,
    if sequencia1(i)==0,
        seq1(i)=1;
    else
        seq1(i)=-1;
    end
    if seqüencia2(i)==0,
        seq2(i)=1;
    else
        seq2(i)=-1;
    end
end

% Realiza a operação de correlação cruzada entre os vetores seq1 e seq2
for i=1:intervalo,
    saída(i)=0;
    for j=1:comprimento,
        saída(i)=saída(i)+seq1(j)*seq2(j);
    end
    % Rotaciona o vetor seq2 uma posição para a direita
    buffer=seq2(comprimento);
    for j=comprimento:-1:2,
        seq2(j)=seq2(j-1);
    end
    seq2(1)=buffer;
end
```

Calculando a autocorrelação da SMC gerada pela função característica $f_1(x) = x^5 + x^2 + 1$:

```
> seqüência=regdesl([0 1 0 0 1],[0 0 0 0 1],31);
> corcsmc(polariza(seqüência),polariza(seqüência))
```

ans =

Columns 1 through 12

```
31 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

Columns 13 through 24

```
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

Columns 25 through 31

```
-1 -1 -1 -1 -1 -1 -1
```

>

Na Figura 31 temos o gráfico da autocorrelação durante um período para uma seqüência de comprimento $N = 31$. Os pontos obtidos pelo programa encontram-se representados pelo simbolo “*”. Considerando seqüências de assinatura com formas de onda quadrada e os resultados das equações (153) e (154), o restante dos pontos estão representados pelos segmentos de reta que unem os pontos obtidos.

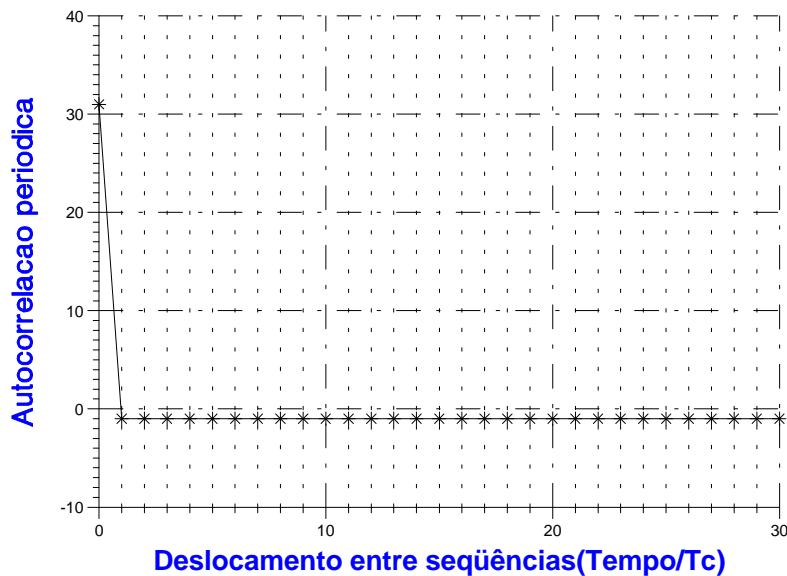


Figura 31

Calculando agora a correlação cruzada de duas SMC geradas pela funções características $f_1(x) = x^5 + x^2 + 1$ e $f_2(x) = x^5 + x^4 + x^3 + x^2 + 1$.

```
> seqüência1=regdesl([0 1 0 0 1],[0 0 0 0 1],31);
> seqüência2=regdesl([0 1 1 1 1],[0 0 0 0 1],31);
> corcsmc(polariza(seqüência1),polariza(seqüência2))
```

ans =

Columns 1 through 12

```
-1    7    7   -1    7   -1   -1   -9   -1   -1   -1   -1
```

Columns 13 through 24

```
-9   -1    7   -1   -9   -9    7   -1    7    7   -1   -1
```

Columns 25 through 31

```
-1   -9    7   -9    7    7   -1
```

>

Na Figura 32 temos o gráfico de correlação cruzada entre duas seqüências durante um período de comprimento $N = 31$. Os pontos obtidos pelo programa encontram-se representados pelo símbolo “*”. Considerando seqüências de assinatura com formas de onda quadrada e os resultados das equações (153) e (154), o restante dos pontos estão representados pelos segmentos de reta que unem os pontos obtidos.

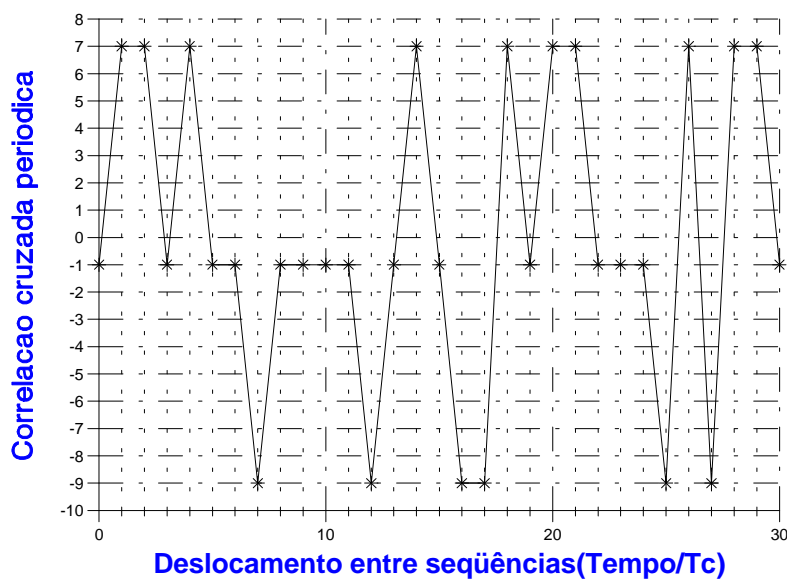


Figura 32

Pares preferenciais:

Dentre as SMCs existem pares de seqüências denominadas "pares preferenciais" em que a função de correlação cruzada periódica possui apenas três valores dados por:

$$\mathbf{R}_{i,k} = \begin{cases} -\mathbf{p}(n) \\ -1 \\ +\mathbf{p}(n) - 2 \end{cases} \quad \text{Onde: } \mathbf{p}(n) = \begin{cases} 1 + 2^{\frac{n+1}{2}} & \mathbf{p}/\mathbf{n} \text{ ímpar} \\ 1 + 2^{\frac{n+2}{2}} & \mathbf{p}/\mathbf{n} \text{ par} \end{cases}$$

Para se encontrar um par preferencial **b** e **b'** as seguintes condições são suficientes:

- O período do registrador deve ser diferente de quatro ou de um múltiplo de 4.
- Uma seqüência **b'** pode ser obtida a partir de uma dizimação α da seqüência **b** onde:

$$\alpha = 2^k + 1 \text{ ou } \alpha = 2^{2^k} - 2^k + 1$$

- O máximo divisor comum deve possuir os valores:

$$\text{mdc}(n, k) = 1 \quad \mathbf{p}/\mathbf{n} \text{ ímpar e } \text{mdc}(n, k) = 2 \quad \mathbf{p}/\mathbf{n} = 2 \pmod{4}$$

Seqüências de Kasami

Existem dois diferentes grupos de seqüências de KASAMI denominados de conjunto pequeno e conjunto grande. A sua importância reside no fato de ser uma classe mais geral que permite obter, por exemplo, as seqüências de Gold.

- Conjunto Pequeno de Seqüências de Kasami:

Seja **f(x)** um polinômio característico de grau **n** par e seja $\mathbf{q} = 2^{\frac{n}{2}} + 1$.

Como **q** divide **p**, $\{\mathbf{a}_{qi}\}$ tem período $\frac{\mathbf{p}}{\mathbf{q}} = 2^{\frac{n}{2}} - 1$, e é uma SMC com polinômio característico

$\mathbf{f}_q(\mathbf{x})$ de grau $\frac{\mathbf{n}}{2}$.

O conjunto pequeno de Kasami é obtido com todas as seqüências de comprimento $2^n - 1$ que podem ser obtidas utilizando condições iniciais diferentes para **f(x)**, **f_q(x)**. A correlação cruzada periódica entre qualquer par de seqüências obtida, pode ter somente 3 valores: **-1**, $-1 - 2^{\frac{n}{2}}$ e $-1 + 2^{\frac{n}{2}}$

- Conjunto Grande de Seqüências de Kasami:

Engloba o conjunto pequeno de Kasami e as seqüências de Gold.

Considere o polinômio característico $f(x).f_q(x)$ obtido no item anterior. Seja $r = 2^{\frac{n+2}{2}} + 1$. Seja $\{a_{ri}\}$ com polinômio característico $f_r(x)$. O conjunto grande de Kasami é obtido com todas as seqüências de comprimento $2^n - 1$ utilizando diferentes condições iniciais de $f(x).f_q(x).f_r(x)$.

A correlação cruzada entre qualquer par de seqüências tem somente 5 valores: -1 , $-1 + 2^{\frac{n}{2}}$, 0 , $-1 + 2^{\frac{n}{2}+1}$ e $-1 - 2^{\frac{n}{2}+1}$.

Seqüências de Gold

Todas as combinações possíveis de fases num período $p = 2^n - 1$ do par preferencial constituem um conjunto de códigos de Gold. O número de combinações possíveis é $2^n + 1$.

Os códigos de Gold obtidos podem ser divididos em 3 grupos conforme a quantidade de '1's e '0's num período:

1º Grupo) O número de '1's é maior que o número de '0's em uma unidade. Este grupo contitue então em um conjunto de seqüências balanceadas. Cerca de metade das seqüências de Gold obtidas com N ímpar pertencem a este grupo.

2º Grupo) A seqüência possui um excesso de '1's num período.

3º Grupo) A seqüência possui um excesso de '0's num período.

Conforme a idealização de um código pseudo-aleatório, as seqüências do 1º Grupo são as que possuem as melhores características para uso em comunicação por espalhamento espectral.

C - DENSIDADE ESPECTRAL DE POTÊNCIA DE UMA FORMA DE ONDA ALEATÓRIA

Seja uma seqüência de pulsos, representado na figura abaixo, com as seguintes características:

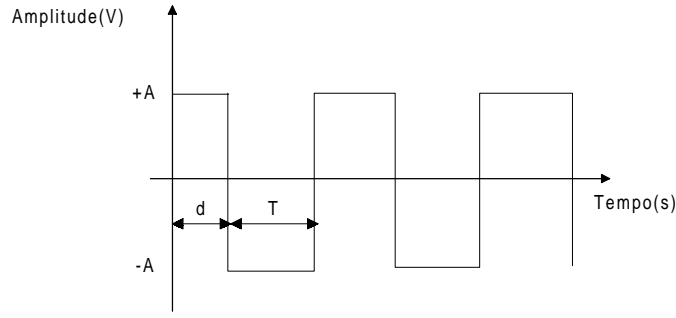


Figura 33 - Sinal de dados aleatório binário $x(t)$

- 1 - Cada pulso tem a forma de onda quadrada com uma duração de tempo T e uma amplitude aleatória de valor $\pm A$.
- 2 - A probabilidade da amplitude ser $+A$ é $1/2$ e de ser $-A$ é $1/2$.
- 3 - As amplitudes dos pulsos são estatisticamente independentes.
- 4 - O atraso d existente no início do primeiro pulso é uma variável aleatória uniformemente distribuída no intervalo $[0, T]$.

Em qualquer instante de tempo o sinal possui apenas um entre dois valores $\pm A$ com igual probabilidade e portanto a média possui o valor $E[x(t)] = 0$ e a variância o valor $E[x(t)^2] = A^2$.

Assumindo dois valores de instante de tempo t_1 e t_2 tais que $0 < t_1 < t_2 < T$ nota-se:

Com $0 < d < t_1$ ou $t_2 < d < T$, implica em t_1 e t_2 estarem no intervalo de tempo em que a forma de onda quadrada é constante e portanto o produto $x(t_1) \cdot x(t_2) = A^2$. Na situação $t_1 < d < t_2$ implica em t_1 e t_2 estarem no intervalo de tempo antes e após a transição da forma de onda quadrada o que implica em $x(t_1) \cdot x(t_2) = \pm A^2$ com equiprobabilidade de ocorrência.

A variável aleatória d encontra-se uniformemente distribuída no intervalo $[0, T]$ o que implica em $P[0 < d < t_1 \text{ ou } t_2 < d < T] = 1 - \frac{(t_2 - t_1)}{T}$ e $P[t_1 < d < t_2] = \frac{(t_2 - t_1)}{T}$.

A função de autocorrelação pode ser escrita na forma:

$$\begin{aligned}
 R_{xx}(t_1, t_2) &= E[X(t_1) \cdot X(t_2)] \\
 &= E[X(t_1) \cdot X(t_2) \mid 0 < d < t_1 \text{ ou } t_2 < d < T] \cdot P[0 < d < t_1 \text{ ou } t_2 < d < T] + \\
 &\quad E[X(t_1) \cdot X(t_2) \mid t_1 < d < t_2] \cdot P[t_1 < d < t_2] \\
 &= A^2 \cdot \left[1 - \frac{(t_2 - t_1)}{T} \right] + A^2 \cdot \frac{1}{2} \cdot \frac{(t_2 - t_1)}{T} - A^2 \cdot \frac{1}{2} \cdot \frac{(t_2 - t_1)}{T} \\
 &= A^2 \cdot \left[1 - \frac{(t_2 - t_1)}{T} \right]
 \end{aligned}$$

Para generalizar este resultado, ref. [19], para valores arbitrários de t_1 e t_2 verifica-se que $R_{xx}(t_1, t_2) = R_{xx}(t_2, t_1)$ e que $R_{xx}(t_1, t_2) = 0$ quando $|t_2 - t_1| > T$. Também $R_{xx}(t_1 + kT, t_2 + kT) = R_{xx}(t_1, t_2)$ e consequentemente:

$$R_{xx}(t_1, t_2) = \begin{cases} A^2 \cdot \left[1 - \frac{|t_2 - t_1|}{T} \right], & |t_2 - t_1| < T \\ 0, & \text{caso contrario} \end{cases}$$

A densidade espectral de potência obtém-se usando a transformada de Fourier na equação acima (teorema de Wiener-Khinchin), o que resulta, a partir de tabelas com dados matemáticos conhecidos, ref. [20]:

$$S_{xx}(f) = A^2 \cdot T \left[\frac{\sin(\pi \cdot f \cdot T)}{\pi \cdot f \cdot T} \right]^2$$

Graficamente a densidade espectral de potência normalizada está representada na figura adiante, onde o eixo das abcissas foi multiplicado por T e o eixo das ordenadas foi dividido por $A^2 \cdot T$, abaixo:

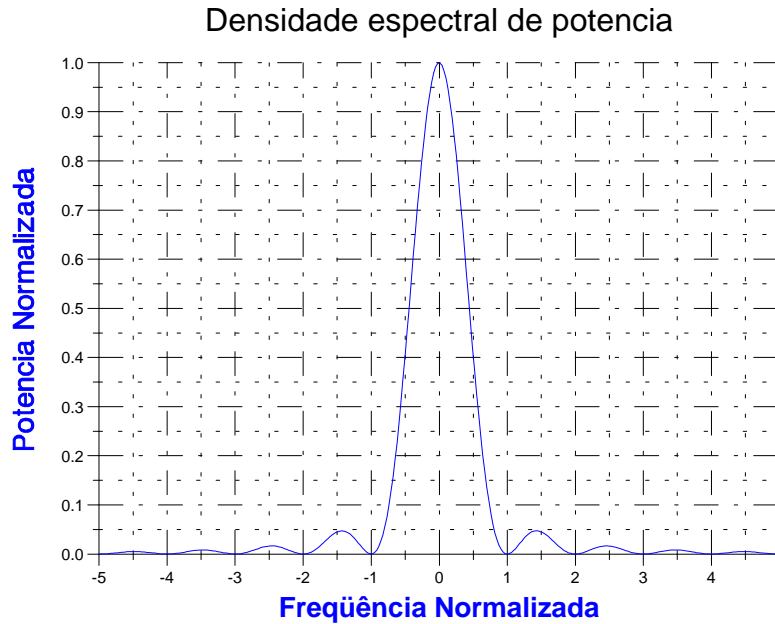


Figura 34

No lóbulo principal que se estende de $-\frac{1}{T}$ até $+\frac{1}{T}$ encontra-se **90%** da potência total do sinal.

D - MÉDIA E VARIÂNCIA DO RUÍDO BRANCO GAUSSIANO NA SAÍDA DO RECEPTOR

Neste apêndice será calculada a média e a variância na saída de um receptor tendo como entrada um ruído branco gaussiano de média 0 e variância $N_0 / 2$.

Utilizando (10):

$$\text{var}\{\eta_x\} = \text{var}\left\{\sum_{j=0}^{N-1} c_x^{(j)} \cdot \int_{j.T_c}^{(j+1).T_c} n(t) \cdot \cos(\omega_0 \cdot t) \cdot dt\right\}$$

$$\text{Com: } c_x(t) = \sum_{j=0}^{N-1} c_x^{(j)} \cdot \Pi(t - j \cdot T_c)$$

Sabendo-se que:

$$\text{var}\{\eta_x\} = E\{\eta_x^2\} - E\{\eta_x\}^2$$

$$E\{\eta_x\} = \sum_{j=0}^{N-1} c_x^{(j)} \cdot \int_{j.T_c}^{(j+1).T_c} E[n(t)] \cdot \cos(\omega_0 \cdot t) \cdot dt$$

Como $E\{n(t)\} = 0$ conclui-se que $E\{\eta_x\} = 0$.

A variância de η_x é igual a:

$$\text{var}\{\eta_x\} = \text{var}\left\{\sum_{j=0}^{N-1} c_x^{(j)} \cdot \int_{j.T_c}^{(j+1).T_c} n(t) \cdot \cos(\omega_0 \cdot t) \cdot dt\right\}$$

A partir da seguinte propriedade:

$$E\left[\int_{j.T_c}^{(j+1).T_c} n(t) \cdot \cos(\omega_0 \cdot t) \cdot dt \cdot \int_{j.T_c}^{(j+1).T_c} n(u) \cdot \cos(\omega_0 \cdot u) \cdot du\right] = \frac{N_0 \cdot T_c}{4}$$

Conclue-se que a variância de η_x é igual a uma somatória de variâncias e portanto:

$$\text{var}\{\eta_x\} = \frac{N_0 \cdot T}{4}$$

E - ALGUNS DETALHES DO ESPALHAMENTO E DA CAPACIDADE DO SISTEMA CDMA DEFINIDO NA IS - 95

O objetivo deste apêndice é o de relacionar este trabalho com o sistema CDMA padronizado no IS-95. A ref. [3] introduz ao conceito de telefonia móvel celular convencional. As referências [4], [12] e [13] apresentam os conceitos básicos sobre telefonia móvel celular baseado na técnica CDMA. A ref. [26] fornece mais detalhes sobre o sistema IS-95 e o resumo a seguir explana apenas os aspectos mais importantes desta última para estabelecer uma conexão com este trabalho.

Espalhamento Espectral:

Antes da transmissão o canal de tráfego reverso e o canal de acesso são espalhados utilizando-se um código de espalhamento longo. O código longo é obtido pelo produto, módulo 2, de uma máscara e o estado dos registradores de deslocamento. Os registradores de deslocamento possuem o seguinte polinômio característico:

$$f(x) = x^{42} + x^{35} + x^{33} + x^{31} + x^{27} + x^{26} + x^{25} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{10} + x^7 + x^6 + x^5 + x^3 + x^2 + x + 1$$

Espalhamento espectral em quadratura:

Após o espalhamento realizado no item anterior os canais de tráfego reverso e de acesso são espalhados novamente, agora em quadratura. As seqüências utilizadas para este espalhamento são as seqüências pilotos pseudo-aleatórias I e Q, usadas também nos canais diretos. Estas seqüências possuem **32767** ($2^{15}-1$) chips e um zero é inserido após 14 zeros consecutivos, resultando em seqüências de comprimento **32768**. Os polinômios correspondentes das SMC são, respectivamente:

$$p_I(x) = x^{15} + x^{13} + x^9 + x^8 + x^7 + x^5 + 1$$

$$p_Q(x) = x^{15} + x^{12} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^3 + 1$$

Existe um alinhamento entre estas duas seqüências (fase inicial) de modo a estabelecer uma referência de tempo.

Sinais da Conexão da Estação Base para Estação Móvel:

O padrão especifica formas de onda com:

Combinação de divisão na freqüência: o espectro utilizável para telefonia celular é dividido em canais com bandas de **1.25 MHz**. Um canal seria inicialmente utilizado e novos canais serão fornecidos conforme a demanda aumente.

Divisão por código pseudo-aleatório: os códigos pseudo-aleatórios são utilizados para distinguir os sinais recebidos na estação móvel pelas diferentes estações base. Todos os sinais

CDMA são transmitidos com um par de códigos pseudo-aleatórios em quadratura. As diferentes células, e setores, são distinguidos por um deslocamento no tempo básico. Os códigos gerados são SMC de comprimento **32768**. A taxa de chip é **1.2288** MHz ou **128** vezes a taxa de informação de **9600** bps. Os sinais transmitidos por uma célula num mesmo canal de rádio CDMA utilizam o mesmo deslocamento de tempo no código PN. São distinguíveis no receptor da estação móvel através do uso de um código binário ortogonal baseado nas funções de Walsh/Hadamard (são gerados 64 códigos distintos, com 64 bits por símbolo e perfeitamente ortogonais entre si). A informação a ser transmitida é codificada convoluionalmente em um codificador de “constraint length” igual à 9 e dois símbolos binários são gerados para cada bit de informação. Os símbolos binários são entrelaçados para combater o desvanecimento de Rayleigh. Para fornecer privacidade cada canal de tráfego é entrelaçado com o endereço do usuário que é também uma sequência pseudo-aleatória. O sinal piloto é transmitido a um nível maior do que os outros tipos de sinais, não carrega informações e consiste somente de um par de códigos PN em quadratura. Detalhes do sincronismo e informações do sistema são transmitidos para a estação móvel pela estação base no canal de sincronismo. Uma vez que seja recebido o sinal de sincronismo a estação móvel seleciona um dos canais de paging para receber outras informações do sistema e as informações de paging propriamente ditas.

Sinais da Conexão: Estação Móvel para Estação Base:

Esta conexão consiste da mesma modulação com sequência pseudo-aleatória que a do item anterior. As diferenças básicas são as de que informação a ser transmitida é codificada convoluionalmente em um codificador de “constraint length” igual a 9 e 3 símbolos binários são gerados para cada bit de informação. A informação codificada é então entrelaçada. Cada grupo de 6 bits (portanto 64 combinações possíveis) é utilizado para selecionar um dos diferentes símbolos do codificador que é transmitido. Cada canal de CDMA reverso pode ter até 62 canais de tráfego e até 32 canais de acesso para paging.

Resumidamente pode-se pois concluir:

- da ERB para a UM (forward link): o sinal é espalhado por duas seqüências PN, uma para cada componente de quadratura, na taxa de **1.2288** Mbits/s; é espalhado por um código de Walsh específico para cada canal e na mesma taxa do PN anterior; finalmente, é espalhado por uma seqüência longa (mascarada de forma distinta em cada caso), agora na taxa de dados (**19.2** Kbits/s devido ao código corretor utilizado).

"Desta forma no sentido ERB-UM tem-se um espalhamento equivalente à combinação de uma seqüência PN e um código de Walsh, num dado canal de uma ERB específica. Como a fonte de geração do sinal é única (ERB), o sistema pode ser considerado praticamente síncrono e ortogonal. Observe-se ainda que cada bit de dado (mascarado) está sendo espalhado por 64 bits da combinação equivalente acima. Face ao exposto conclue-se que apenas existe a ortogonalidade no receptor da UM para sinais provenientes de uma mesma ERB; para os sinais provenientes de outras ERBs a influência deve ser considerada como randômica e caracterizável pela correlação cruzada parcial (ou aperiódica)."

- da UM para a ERB (reverse link): o sinal é espalhado por duas seqüências PN, uma para cada componente de quadratura, na taxa de **1.2288** Mbits/s; é espalhado por um código de

Walsh específico e variável (dependente da informação), numa taxa 4 vezes inferior à do PN anterior e $32/3$ vezes superior ao dos dados (28.8 Kbits/s devido ao código corretor utilizado); finalmente, é espalhado por uma sequência longa (mascarada de forma distinta para cada caso), agora na mesma taxa da sequência PN.

"Desta forma no sentido UM-ERB tem-se um espalhamento equivalente à combinação de uma sequência PN com uma sequência longa e mascarada. Como as fontes de geração dos sinais são múltiplas (UM's), o sistema é assíncrono e, evidentemente, não ortogonal. Face ao exposto conclue-se que não existe ortogonalidade no receptor da ERB para sinais provenientes das UMs e a influência deve ser considerada como randômica e caracterizável pela correlação cruzada parcial (ou aperiódica)."

Determinação da Capacidade do Sistema IS-95:

A relação entre as potências transmitida e interferente, denotada por C/I em analogia com os sistemas analógicos, pode ser escrita no caso CDMA como:

$$\frac{C}{I} = \frac{R \cdot E_b}{W \cdot N_0}$$

onde:

R = taxa de transmissão

E_b = energia por bit

W = largura da banda ocupada pelo sinal

N₀ = densidade espectral de potência equivalente dos sinais interferentes

Assumindo que N usuários encontram-se transmitindo dentro da célula, a interferência para um determinado será **C.(N-1)**. Levando em consideração que a potência recebida de todos os sinais interferentes são iguais na entrada do receptor resulta:

$$\frac{C}{I} = \frac{R \cdot E_b}{R \cdot E_b \cdot (N-1)} = \frac{1}{N-1}$$

Portanto:

$$N-1 = \frac{W}{R} \cdot \frac{1}{\frac{E_b}{N_0}}$$

Observe-se então que esta abordagem é essencialmente a de Weber, já exposta anteriormente.

Equação Completa da Capacidade:

Da equação anterior, considerando-se **N >> 1**, e os demais fatores, abaixo detalhados, tem-se:

$$N = \frac{W}{R} \times \frac{1}{\frac{E_b}{N_0}} \times \frac{1}{d} \times F \times G$$

onde:

F = fator de reuso de frequência = **0.60**

$\frac{1}{d}$ = ciclo de atividade vocal $1/0.35 = 2.86$

G = ganho devido ao uso de antenas setorizadas com abertura de **120° = 2,55**

Assumindo os seguintes valores:

R = **9600** bps; **W** = **1.25** MHz e $\frac{E_b}{N_0} = 7.0$ dB resulta uma capacidade de **N = 99** canais.

Considerando que o sistema pode ser setorizado em **33** canais por setor e admitindo-se uma perda de **2%**, o tráfego total por setor será de **24.6** erlangs. Este tráfego, por sua vez, significará **73.8** erlangs na célula e portanto **738** erlangs para uma banda total de **12.5** MHz.

Para efeito de comparação seja o sistema FDMA-AMPS na mesma banda total de **12.5** MHz e com uma largura de canal de **30** KHz. Tem-se assim **417** canais no total. Com um fator de reuso de frequências na proporção **7:1** ter-se-á **59** canais vocais por célula. Como alguns destes canais devem ser utilizados para controle considere-se **57** canais para o escoamento de tráfego telefônico por célula. Setorizando a célula estarão disponíveis **19** canais por setor e, com a mesma hipótese de perda anterior, isto implicará num tráfego total de **12.3** erlangs no setor e portanto **36.9** erlangs na célula.

*Comparando-se os resultados acima o sistema CDMA apresenta um ganho em termos de tráfego escoado de **20:1** em relação ao sistema FDMA-AMPS.*

F - LISTAGEM DOS PROGRAMAS DESENVOLVIDOS

Na Tabela 3 encontra-se a relação de dependência entre as rotinas desenvolvidas neste trabalho. Esta relação especifica quais sub-rotinas são chamadas por quais rotinas principais. Por exemplo: A rotina ALLSMC.M, responsável pela geração de seqüências de máximo comprimento ciclicamente distintas, chama as rotinas COCOSETS.M, REGDESL.M e DIZIMA.M durante a sua execução.

<u>ROTINA PRINCIPAL</u>	<u>SUB-ROTINAS</u>
ALLSMC.M	COCOSETS.M, REGDESL.M, DIZIMA.M
BAB.M	CAB.M
CAB.M	
CABP.M	
CLEAN.M	LINHA.M
COCOSETS.M	
CORCSMC.M	
DIZIMA.M	
GOLD.M	XORSEQ.M
KASAMIS.M	REGDESL.M, DIZIMA.M, XORSEQ.M
LEHNERT.M	
LIMNAZAR.M	MAXMIN.M
LINHA.M	
MAXMIM.M	
NAZARIM	OAB.M, OABC.M, MAXMIN.M
OAB.M	CAB.M
OABC.M	CAB.M
PARPREF.M	COCOSETS.M, REGDESL.M, DIZIMA.M, CABP.M
PIJ.M	
PIJSUBOT.M	PIJ.M
PIORCASO.M	LINHA.M, RKI.M, RKIC.M
POLARIZA.M	
PRIMO.M	
PURSLEY.M	LINHA.M, BAB.M
QSCDMA.M	PIJ.M
REGDESL.M	
RFI.M	
RFIC.M	
RKI.M	CAB.M, RFIC.M, RFI.M
RKIC.M	CAB, RFIC.M, RFI.M
SAMPLE1.M	
XORSEQ.M	

Tabela 3 - Relações entre rotinas utilizadas nas simulações

Rotina ALLSMC.M: Gera todas seqüências de máximo comprimento ciclicamente distintas.

function [saída]=allsmc(polinômio)

% SINTAXE: [saída]=allsmc(polinômio)

%

% DESCRIÇÃO: Gera todas seqüências de máximo comprimento ciclicamente distintas.

%

% PARÂMETROS DE ENTRADA:

```

%
% polinômio - vetor com os n+1 coeficientes do polinômio de grau n. Portanto,
% dado o polinômio  $1+a_1.x^1+a_2.x^2+...+a_n.x^n$  o vetor deve ser
% [1 a1 a2 ... an].
%
% PARÂMETROS DE SAÍDA:
%
% saída - matriz onde cada linha contém seqüências de máximo comprimento
% e cada coluna é um elemento da seqüência.
%
% EXEMPLO: Por exemplo, para gerar todas seqüências de máximo comprimento a
% partir do polinômio primitivo de grau 6  $f(x)=1+X^2+X^3+X^5+X^6$ ,
% digitar:
%
% allsmc([1 0 1 1 0 1 1]);
%

% Obtém o grau do polinômio
grau=length(polinômio)-1;

% Calcula o período das seqüências
período=2^grau-1;

% Gera um vetor com as realimentações do registrador de deslocamento a partir
% do polinômio característico
for i=1:grau,
    realimentação(i)=polinômio(i+1);
end

% Gera a tabela de coconjuntos ciclotômicos
tabela=cocosets(primo(período),período);

% Obtém a dimensão dos coconjuntos ciclotômicos
[linhas,colunas]=size(tabela);

% Gera um vetor com condições iniciais tais que a função geradora é igual ao in-
% verso da função característica
tamreg=length(realimentação);
for i=1:tamreg,
    inicial(i)=0;
end
inicial(tamreg)=1;

% Obtém uma das SMC a partir do polinômio primitivo fornecido
seqpri=regdesl(realimentação,inicial,período);

% Realiza as dizimações e obtém todas as SMC possíveis

```

```

for i=1:linhas,
    seqdiz=dizima(seqpri,tabela(i,1),período);
    for j=1:length(seqdiz)
        saída(i,j)=seqdiz(j);
    end
end
end

```

Rotina BAB.M: Calcula a função $\beta_{i,x}$ entre duas seqüências que é utilizada durante o cálculo da probabilidade de erro de bit pelo método de Pursley. A função é definida pela equação:

$$\beta_{i,x} = 2 \cdot N^2 + 4 \cdot \sum_{l=1}^{N-1} C_i(l) \cdot C_x(l) + \sum_{l=1-N}^{N-1} C_i(l) \cdot C_x(l+1)$$

```

function [saída]=bab(seqa,seqb)
% SINTAXE: [saída]=bab(seqa,seqb)
%
% DESCRIÇÃO:      Calcula a função Bab entre duas seqüências não polarizadas seqa e
%                  seqb.
%
% PARÂMETROS DE ENTRADA:
%
% seqa            -   vetor com a primeira seqüência. O vetor deve conter a seqüên-
%                  cia durante um período.
% seqb            -   vetor com a segunda seqüência. O vetor deve conter a seqüên-
%                  cia durante um período.
%
% PARÂMETROS DE SAÍDA:
%
% saída           -   um escalar com o resultado da função Bab entre duas seqüências
%                  e cada coluna é um elemento da seqüência.
%
% EXEMPLO:        Supondo que se queira calcular a função Bab entre as seqüências seqa
%                  seqa e seqb abaixo:
%
%                  seqa = [1 1 1 0 0 1];
%                  seqb = [0 1 1 0 1 1];
%
%                  Chama-se a função:
%
%                  bab(seqa,seqb);
%
% Obtém o período da seqüência

```

```

N=length(seqa);

% Inicializa o resultado com a constante
saída=2*N^2;

% Calcula a parte variável da função Bab
for l=1:N-1,
    buffer=cab(l,seqa,seqa);
    saída=saída+4*buffer*cab(l,seqb,seqb)+buffer*cab(l+1,seqb,seqb);
end

for l=1-N:0,
    saída=saída+cab(l,seqa,seqa)*cab(l+1,seqb,seqb);
end

```

Rotina CAB.M: Calcula a função de correlação cruzada aperiódica entre duas seqüências seqa e seqb com deslocamento l. A função é definida pela equação:

$$C_{a,b}(l) = \begin{cases} \sum_{i=0}^{N-1-l} a_i \cdot b_{i+l} & 0 \leq l \leq N-1 \\ \sum_{i=0}^{N-1+l} a_{i-l} \cdot b_i & 1-N \leq l < 0 \\ 0 & |l| \geq N \end{cases}$$

```

function [saída]=cab(l,seqa,seqb)
% SINTAXE: [saída]=cab(l,seqa,seqb)
%
% DESCRIÇÃO:      Calcula a função de correlação cruzada aperiódica entre duas
%                  seqüências seqa e seqb com deslocamento l.
%
% PARÂMETROS DE ENTRADA:
%
% seqa            -   vetor com a primeira seqüência. O vetor deve conter a seqüên-
%                      cia durante um período.
% seqb            -   vetor com a segunda seqüência. O vetor deve conter a seqüên-
%                      cia durante um período.
%
% PARÂMETROS DE SAÍDA:
%
% saída           -   um escalar com a correlação cruzada aperiódica entre as duas
%                      seqüências.
%

```



```

% EXEMPLO:      Supondo que se queira calcular a correlação cruzada aperiódica entre
%               as seqüências seqa e seqb com deslocamento l=2:
%
%               seqa = [1 1 1 0 0 1];
%               seqb = [0 1 1 0 1 1];
%
%               Chama-se a função:
%
%               cab(2,seqa,seqb)
%
%
% Obtém o período da seqüência
N=length(seqa);

% Inicializa o resultado com zero
saída=0;

% Calcula a correlação cruzada aperiódica
if (l>=0)&(l<=N-1),
    saída=sum(seqa((0:N-1-l)+1).*seqb((0:N-1-l)+1+1));
elseif (l<0)&(l>=1-N),
    saída=sum(seqa((0:N-1+l)+1-l).*seqb((0:N-1+l)+1));
end

```

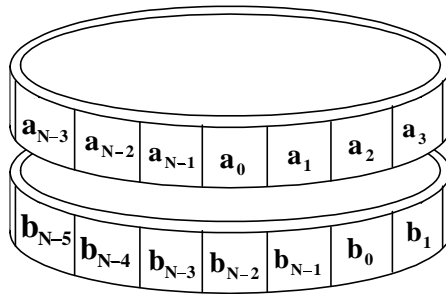
Rotina CABP.M: Calcula a função de correlação cruzada periódica entre duas seqüências para $l = 0, 1, \dots, N-1$. A função é definida pela equação:

$$\theta_{a,b}(l) = \sum_{i=0}^{N-1} a_i \cdot b_{i+l}$$

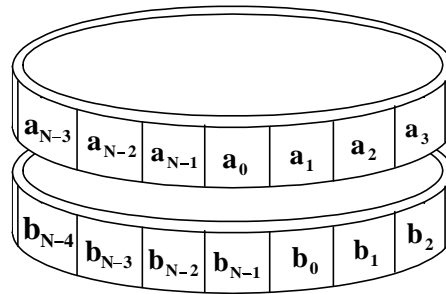
Nos esquemas adiante representados verifica-se visualmente qual é o comportamento da função de correlação cruzada periódica para valores de $l=-2, l=-1, l=0, l=1, l=2$. Verifica-se que para valores de $l > 0$ a seqüência **b** desloca-se para a esquerda (em relação a seqüência **a**) e que para valores de $l < 0$ a seqüência desloca-se para a direita (em relação a seqüência **a**). A região de multiplicação bit a bit das seqüências é constante e de período **N**. A representação facilita a visualização do comportamento da função já que as seqüências são periódicas.

Atenção: Esta rotina implementa a mesma operação que **CORCSMC.M**. A única diferença entre as rotinas é que as seqüências utilizadas como argumento de **CABP.M** não são polarizadas e as seqüências utilizadas como argumento de **CORCSMC.M** são polarizadas.

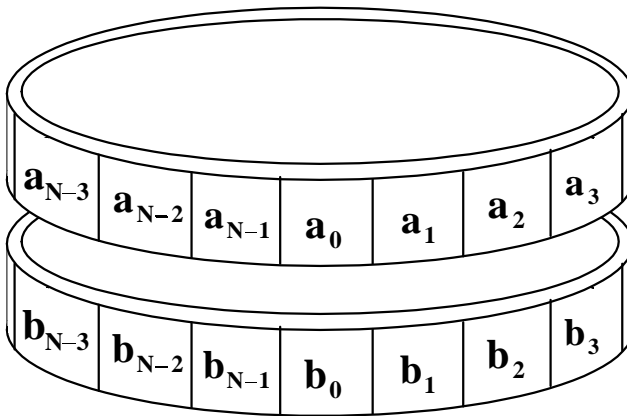
Caso $l=-2$



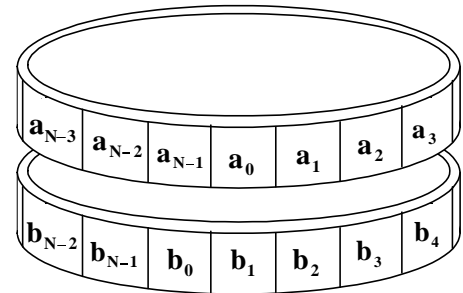
Caso $l=-1$



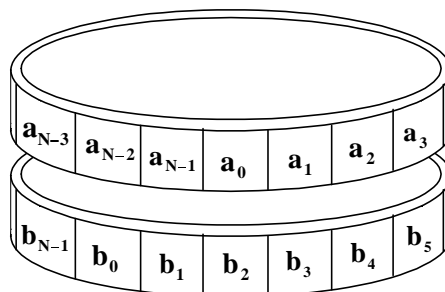
Caso $l=0$



Caso $l=1$



Caso $l=2$



```
function saída=cabp(seqa,seqb)
% SINTAXE: saída=cabp(seqa,seqb)
%
% DESCRIÇÃO:      Obtém um vetor com todos os valores de correlação cruzada periódica
%                  entre as duas seqüências num período.
%
% PARÂMETROS DE ENTRADA:
```

```

%
% seqa      -      vetor com a primeira seqüência. O vetor deve conter a seqüên-
%                  cia durante um período.
% seqb      -      vetor com a segunda seqüência. O vetor deve conter a seqüên-
%                  cia durante um período.
%
% PARÂMETROS DE SAÍDA:
%
% saída      -      um vetor com todos os valores de correlação cruzada periódica
%                  entre as duas seqüências em um período.
%
% EXEMPLO:    Supondo que se queira calcular a correlação cruzada periódica entre
%              as seqüências seqa e seqb abaixo:
%
%              seqa=[1 0 1 1 1 0 0];
%              seqb=[1 1 0 1 0 0 1];
%
%              Chama-se a função:
%
%              cabp(seqa,seqb);
%
%
% Verifica o período das seqüências
comprimento=length(seqa);

% Transforma cada vetor no equivalente bipolar
seq1=(-1).^seqa;
seq2=(-1).^seqb;

% Realiza a operação de correlação cruzada entre os vetores seq1 e seq2
for i=1:comprimento,
    saída(i)=sum(seq1.*seq2);
% Rotaciona o vetor seq2 uma posição para a direita
seq2=[seq2(length(seq2)) seq2(1:length(seq2)-1)];
end

```

Rotina CLEAN.M: Gera uma matriz de saída que não contém duplicações de linhas da matriz de entrada levando em consideração inclusive a fase. A matriz deve ter no mínimo duas linhas, senão ocorre um erro de execução do programa. Para realizar a procura de seqüências iguais independente da fase é implementado o método da exaustão, ou seja, compara-se todas as fases possíveis de uma determinada seqüência (a que queremos determinar se existe uma outra igual) com todas as seqüências existentes. Claramente esta rotina tende a aumentar exponencialmente o tempo de execução em função do aumento do número e comprimento das seqüências.

```

function [saída]=clean(entrada)
% SINTAXE: [saída]=clean(entrada)
%
% DESCRIÇÃO:      Gera uma matriz de saída que não contém duplicações de linhas da
%                  matriz de entrada levando em consideração inclusive a fase.
%                  A matriz deve ter no mínimo duas linhas, senão ocorre um erro de
%                  execução do programa.
%
% PARÂMETROS DE ENTRADA:
%
% entrada      -      matriz com linhas de entrada.
%
% PARÂMETROS DE SAÍDA:
%
% saída        -      matriz que não contém duplicações de linhas da matriz entrada
%
% EXEMPLO:      Seja a matriz de entrada abaixo onde claramente a primeira linha é
%                  igual a segunda linha (de um deslocamento de 1):
%
%                  entrada=[ 1 0 1 0
%                             0 1 0 1
%                             1 1 0 0 ];
%
%                  Chama-se a função:
%
%                  saída=clean(entrada);
%
%                  Obtém-se:
%
%                  saída=[ 1 0 1 0
%                           1 1 0 0 ]
%
% Obtém o tamanho da matriz de entrada
[linhas,colunas]=size(entrada);

% Gera um vetor de tamanho igual ao número de linhas da matriz de entrada
% e seta todos os elementos com o valor 1; desta forma durante a busca as
% que já forem identificadas como duplicadas serão setadas com o valor 0.
% O programa de busca não procura as linhas cuja correspondência neste ve-
% tor estiver setado para 0. Desta forma o tempo de busca de linhas dupli-
% cadas diminui significativamente.
for linhatual=1:linhas,
    indice(linhatual)=1;
end

% Rotina principal de busca de linhas duplicadas

```

```

for linhatual=1:linhas-1,
linhatual=linhatual
% Verifica se a linha é duplicada
if indice(linhatual)==1,
    vetoratual=linha(entrada,linhatual);
    for linhabusca=linhatual+1:linhas,
        if indice(linhabusca)==1,
            vetorbusca=linha(entrada,linhabusca);
            achei=0;
            fase=0;
            while (achei==0)&(fase<colunas),
% Verifica se as duas linhas são iguais
                if min(vetoratual==vetorbusca)~=0,
                    achei=1;
                end
% Rotaciona um dos vetores que estão sendo comparados
                if achei==0,
                    buffer=vetoratual(colunas);
                    for coluna=colunas:-1:2,
                        vetoratual(coluna)=vetoratual(coluna-1);
                    end
                    vetoratual(1)=buffer;
                end
                fase=fase+1;
            end
            if achei==1,
                indice(linhabusca)=0;
            end
        end
    end
end
end
end

% Gera a matriz de saída
linhasaída=0;
for linhatual=1:linhas,
    if indice(linhatual)==1,
        linhasaída=linhasaída+1;
        for coluna=1:colunas,
            saída(linhasaída,coluna)=entrada(linhatual,coluna);
        end
    end
end
end

```

Rotina COCOSETS.M: Retorna uma matriz que contém as dizimações necessárias para gerar a partir de um polinômio primitivo todas as seqüências de máximo comprimento distintas possíveis. Genericamente pode-se dizer que esta matriz contém os coconjuntos ciclotômicos.

```
function [saída]=cocosets(primo,p)
% SINTAXE: [saída]=cocosets(primo,p)
%
% DESCRIÇÃO:      Retorna uma matriz que contém as dizimações necessárias para gerar
%                  a partir de um polinômio primitivo todas as seqüências de máximo
%                  comprimento distintas possíveis. Genericamente pode-se dizer que esta
%                  matriz contém os coconjuntos ciclotômicos.
%
% PARÂMETROS DE ENTRADA:
%
% primo          -   vetor com os primos em relação a p (comprimento da seqüência).
%                  Para facilitar a obtenção deste vetor utiliza-se a função
%                  primo com o parâmetro p.
% p              -   comprimento da seqüência.
%
% PARÂMETROS DE SAÍDA:
%
% saída          -   matriz onde para cada linha, cada coluna corresponde às dizima-
%                  ções que geram a mesma seqüência com uma fase diferente.
%
% EXEMPLO:       Para gerar todos os coconjuntos ciclotômicos de comprimento 7 (a par-
%                  tir portanto de um polinômio primitivo de grau 3) chama-se a função:
%
%                  saída=cocosets(primo(7),7);
%
%                  Obtém-se:
%
%                  saída=[ 1 2 4
%                          3 5 6 ]
%
%                  Portanto, somente as dizimações de 1 e de 3 geram seqüências de má-
%                  ximo comprimento distintas.
%
%
% Cria a tabela utilizada para verificar os coconjuntos ciclotômicos
limite=fix(log(p)/log(2));
for j=0:limite,
    for i=1:length(primo),
        resultado1=2^j*primo(i);
        resultado2=resultado1-fix(resultado1/p)*p;
        if(resultado2==0)
```

```

    resultado2=p;
end
tabela(i,j+1)=resultado2;
end
end

% Sorteia cada linha da tabela obtida
tabela=(sort(tabela'))';

% Copia a primeira coluna da tabela em um vetor
for i=1:length(primo),
    pricol(i)=tabela(i,1);
end

% Cria um vetor que informa quais linhas não são repetidas
vetoridx=0;
for i=1:length(primo),
    aprocurar=pricol(i);
    encontrados=find(tabela==aprocurar);
    if encontrados(1)==i,
        vetoridx=vetoridx+1;
        vetor(vetoridx)=i;
    end
end

% Baseado no vetor formado no passo anterior gera uma matriz com os coconjuntos
% ciclotômicos.
linha=1;
for i=1:vetoridx,
    for j=0:limite,
        saída(linha,j+1)=tabela(vetor(i),j+1);
    end
    linha=linha+1;
end
end

```

Rotina CORCSMC.M: Calcula a função de correlação cruzada periódica entre duas seqüências para $\mathbf{l} = \mathbf{0}, \mathbf{1}, \dots, \mathbf{N} - \mathbf{1}$. A função é definida pela equação:

$$\theta_{a,b}(\mathbf{l}) = \sum_{i=0}^{N-1} \mathbf{a}_i \cdot \mathbf{b}_{i+l}$$

Atenção: Ler funcionamento da rotina, bem como diferenças de funcionamento, na explicação da rotina **CABP.M**.

```
function saída=corcsmc(seqa,seqb)
```

```

% SINTAXE: function saída=corcsms(seqa,seqb)
%
% DESCRIÇÃO:      Calcula a função de correlação cruzada periódica entre duas se-
%                  quências com uma das seqüências sofrendo todos os deslocamentos
%                  de fase possíveis (que é igual ao comprimento da seqüência).
%
% PARÂMETROS DE ENTRADA:
%
% seqa            -      vetor com a primeira seqüência. O vetor deve conter a seqüên-
%                  cia (polarizada) durante um período.
% seqb            -      vetor com a segunda seqüência. O vetor deve conter a seqüên-
%                  cia (polarizada) durante um período.
%
% PARÂMETROS DE SAÍDA:
%
% saída           -      vetor com os valores da função de correlação cruzada periódica entre
%                  duas seqüências
%
% EXEMPLO:        Supondo que se queira calcular a função de correlação cruzada pe-
%                  riódica entre as seqüências seqa e seqb abaixo:
%
%                  seqa=[1,-1,1,1,1,-1,-1];
%                  seqb=[1,1,-1,1,-1,-1,1];
%
%                  Chama-se a função:
%
%                  corcsms(seqa,seqb)
%
% Verifica o período das seqüências
comprimento=length(seqa);
intervalo=comprimento;

seq1=seqa;
seq2=seqb;

% Realiza a operação de correlação cruzada entre os vetores seq1 e seq2
for i=1:intervalo,
    saída(i)=0;
    for j=1:comprimento,
        saída(i)=saída(i)+seq1(j)*seq2(j);
    end
% Rotaciona o vetor seq2 uma posição para a direita
buffer=seq2(comprimento);
for j=comprimento:-1:2,
    seq2(j)=seq2(j-1);
end

```



```
seq2(1)=buffer;
end
```

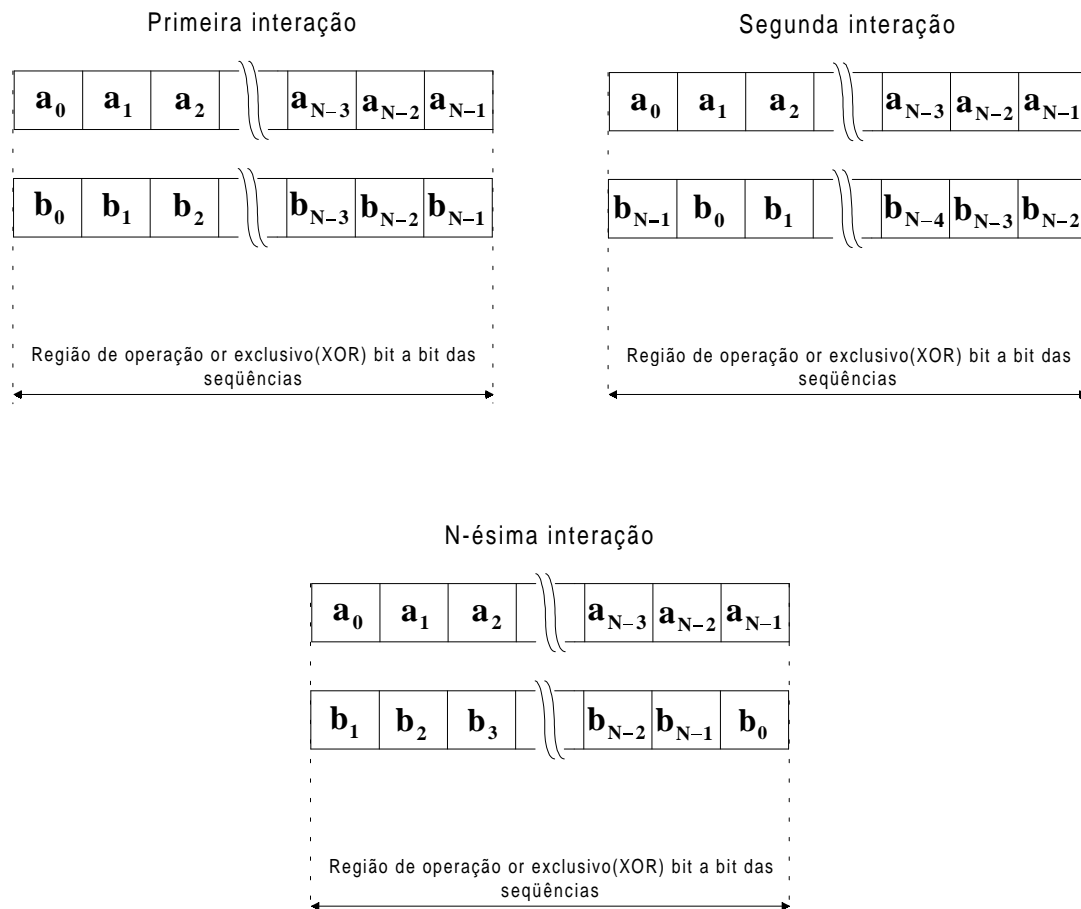
Rotina DIZIMA.M: Dizima uma seqüência pelo número dado pelo dizimador.

```
function [saída]=dizima(seqüência,dizimador,interações)
% SINTAXE: [saída]=dizima(seqüência,dizimador,interações)
%
% DESCRIÇÃO:      Dizima uma seqüência pelo número dado pelo dizimador.
%
% PARÂMETROS DE ENTRADA:
%
% seqüência      -      vetor com a seqüência. O vetor deve conter a seqüência durante
%                      período.
% dizimador      -      escalar com o dizimador.
% interações     -      período da seqüência obtida pela dizimação.
%
% PARÂMETROS DE SAÍDA:
%
% saída          -      vetor com a seqüência obtida pela dizimação
%
% EXEMPLO:       Dizimar de 3 a seqüência [1 0 1 1 1 0 0] por um período (no caso 7):
%
%                      dizima([1 0 1 1 1 0 0],3,7)
%
% Obtém o período da seqüência
período=length(seqüência);

% Realiza a dizimação. Utiliza-se um índice que vai se movimentando pela se-
% qüência de uma quantidade indicada pelo dizimador e que ao ultrapassar um
% período se utiliza das propriedades de periodicidade para se reinicializar.
contador=1;
índice=1;
while contador<=interações,
saída(contador)=seqüência(índice);
índice=índice+dizimador;
if(índice>período)
índice=índice-fix(índice/período)*período;
end
contador=contador+1;
end
```

Rotina GOLD.M: Gera seqüências de Gold a partir de seqüências preferenciais (geradas por um par preferencial). O programa realiza a operação “or exclusivo” entre duas seqüências preferenciais resultando uma terceira seqüência chamada de seqüência de Gold. Em seguida a segunda seqüência é rotacionada uma posição para a direita e a operação “or exclusivo” é novamente realizada. O processo se repete por N vezes no total. A rotina retorna com todas as N seqüências obtidas mais as seqüências preferenciais que pertencem também ao conjunto de seqüências de Gold.

A figura abaixo exemplifica o processo realizado pelo programa durante a primeira, segunda e n-ésima interação.



```
function saída=gold(seqüência1,seqüência2)
% SINTAXE: saída=gold(seqüência1,seqüência2)
%
% DESCRIÇÃO:      Gera seqüências de Gold a partir de seqüências preferenciais (ge-
%                  radas por um par preferencial).
%
% PARÂMETROS DE ENTRADA:
%
% seqüência1 -     vetor com a primeira das seqüências preferenciais.
```

```

% seqüência2 -      vetor com a segunda das seqüências preferenciais.
%
% PARÂMETROS DE SAÍDA:
%
% saída      -      matriz com as seqüências geradas pelo par preferencial.
%

% Obtém o período da seqüência
período=length(seqüência1);

% Obtém todas as seqüências de Gold através da operação OR exclusivo entre as
% duas seqüências defasando uma das seqüências um chip de cada vez durante um
% período
saída=xorseq(seqüência1,seqüência2,período);

% Obtém o número de seqüências e o comprimento das seqüências obtidas no passo
% anterior
[linhas,colunas]=size(saída);

% Acrescenta as seqüências geradas pelo par preferencial
for coluna=1:colunas,
    saída(linhas+1,coluna)=seqüência1(coluna);
    saída(linhas+2,coluna)=seqüência2(coluna);
end

```

Rotina KASAMIS.M: Gera as seqüências de Kasami do grupo pequeno a partir de um polinômio primitivo de grau par.

```

function saída=kasamis(polinômio)
% SINTAXE: saída=kasamis(polinômio)
%
% DESCRIÇÃO:      Gera as seqüências de Kasami do grupo pequeno a partir de um poli-
%                  nômio primitivo de grau par.
%
% PARÂMETROS DE ENTRADA:
%
% seqüência1 -      vetor com os coeficientes do polinômio primitivo  $1+a_1.x+a_2.x^2+\dots+a_n.x^m$  na seguinte ordem [1 a1 a2 ...an].
%
% PARÂMETROS DE SAÍDA:
%
% saída      -      matriz com as seqüências de Kasami do grupo pequeno.
%
% EXEMPLO:      Para gerar as seqüências de Kasami a partir do polinômio primitivo
%                 $f(X)=1+X+X^4$  digita-se:

```

```

%
%
%
% Obtém o grau do polinômio
grau=length(polinômio)-1;

% Gera um vetor com o conteúdo inicial do registrador de deslocamento
for i=1:grau,
    inicial(i)=0;
end
inicial(grau)=1;

% Gera um vetor com as realimentações do registrador de deslocamento a partir
% do polinômio característico
for i=1:grau,
    realimentação(i)=polinômio(i+1);
end

% Obtém uma seqüência de máximo comprimento a partir do polinômio primitivo
seqpri=regdesl (realimentação,inicial,2^grau-1);

% Calcula o valor de q que é a dizimação necessária que deve sofrer a seqüência
% gerada pelo polinômio primitivo
q=2^(grau/2)+1;

% Realiza a dizimação q da seqüência gerada pelo polinômio primitivo
seqq=dizima(seqpri,q,2^grau-1);

% Retorna uma matriz em que cada linha é uma seqüência obtida pela operação XOR
% entre as seqüências seqpri e seqq, onde seqq encontra-se deslocada para a di-
% reita de (número da linha-1) posições;
saída=xorseq(seqpri,seqq,2^grau-1);

% Obtém o número de linhas e colunas da matriz de saída
[linhas,colunas]=size(saída);

% Adiciona a seqüência gerada pelo polinômio primitivo à matriz de saída
for coluna=1:colunas,
    saída(linhas+1,coluna)=seqpri(coluna);
end

```

Rotina LEHNERT.M: Calcula e integra a função densidade de probabilidade pelo método de Lehnert.

```

function [saida,limite1,limite2]=lehnert(configuração,seqüências,tipo)
% SINTAXE: [saida,limite1,limite2]=lehnert(configuração,seqüências,tipo)
%
% DESCRIÇÃO:      Calcula e integra a função densidade de probabilidade pelo método
%                  de Lehnert
%
% PARÂMETROS DE ENTRADA:
%
% configuração-    vetor com 4 elementos elementos [Tc No usuário Ampli] onde Tc
%                  é o tempo de um chip, No é uma constante (No/2 corresponde a
%                  variância do ruído branco gaussiano), usuário determina qual
%                  dos vetores dentre as seqüências é utilizada pelo receptor e
%                  Ampli é um vetor com a amplitude de cada um dos sinais inter-
%                  ferentes na entrada do receptor.
% seqüências -     vetor com as seqüências utilizadas pelos usuários. Possui nú-
%                  mero de linhas correspondendo ao número de usuários e número
%                  de colunas correspondendo ao comprimento das seqüências.
%                  Todas as seqüências devem estar polarizadas.
% tipo            - escalar que indica o tipo de abordagem conforme o texto e abaixo:
%                  1 - indica abordagem do Tipo 1
%                  2 - indica abordagem do Tipo 2
%                  3 - indica abordagem do Tipo 3
%                  4 - indica abordagem do Tipo 4
%
% PARÂMETROS DE SAÍDA:
%
% pe              - probabilidade de erro.
%
%
% Obtém parâmetros de configuração
Tc=configuração(1);
No=configuração(2);
usuário=configuração(3);
A=configuração(4);
%
% Obtém o número e o período das seqüências
[K,N]=size(seqüências);
%
% Calcula o período de um dado
T=N*Tc;
%
if tipo==1,
% Tipo 1 - conforme texto do trabalho
SNR=sqrt(inv((K-1)/(N)+No/(A^2*T)));
elseif tipo==2,
% Tipo 2 - conforme texto do trabalho
SNR=sqrt(inv((2/3)*(K-1)/(N)+No/(A^2*T)));

```

```

elseif tipo==3,
% Tipo 3 - conforme texto do trabalho
SNR=sqrt(inv((1/2)*(K-1)/(N)+No/(A^2*T)));
elseif tipo==4,
% Tipo 4 - conforme texto do trabalho
SNR=sqrt(inv((1/3)*(K-1)/(N)+No/(A^2*T)));
end

% Obtém a probabilidade de erro através da aproximação gaussiana
saida=0.5*erfc(SNR/sqrt(2));

```

Rotina LIMNAZAR.M: Obtém o limite máximo (por simetria o mínimo é o negativo deste valor) dentre todas as funções densidade de probabilidade obtidas durante a execução da programa NAZARIM.

```

function [limmax]=limnazar(seqüências,usuário)
% SINTAXE: [limmax]=limnazar(seqüências,usuário)
%
% DESCRIÇÃO:      Obtém o limite máximo (por simetria o mínimo é o negativo deste
%                  valor) dentre todas as funções densidade de probabilidade obtidas
%                  durante a execução da programa NAZARIM.
%
% PARÂMETROS DE ENTRADA:
%
% seqüências -      família de seqüências (não polarizadas) utilizadas pelos usuá-
%                  rios
% usuário      -      vetor indicando qual é o usuário receptor preparado para re-
%                  ceber o sinal desejado. Indicação da linha do receptor.
%
% PARÂMETROS DE SAÍDA:
%
% limmax      -      passo a ser utilizado em todos os cálculos
%
% Obtém o tamanho da família e o comprimento das seqüências
[K,N]=size(seqüências);

% Gera uma matriz com todos os Oix e Oixc. Cada linha corresponde ao resultado
% obtido para um usuário interferente
k=0;
for j=1:K,
if j~=usuário,
k=k+1;
for i=1:N,
Oix(k,i)=oab(i,linha(seqüências,j),linha(seqüências,usuário));

```

```

    Oixc(k,i)=oabc(i,linha(seqüências,j),linha(seqüências,usuário));
end
end
end

% Obtém o valor máximo (em módulo) e o mínimo (em módulo) de todas as fdps pares
[máximo1,mínimo1,caso1]=maxmin(Oix);

% Obtém o valor máximo (em módulo) e o mínimo (em módulo) de todas as fdps ímpa-
% res
[máximo2,mínimo2,caso2]=maxmin(Oixc);

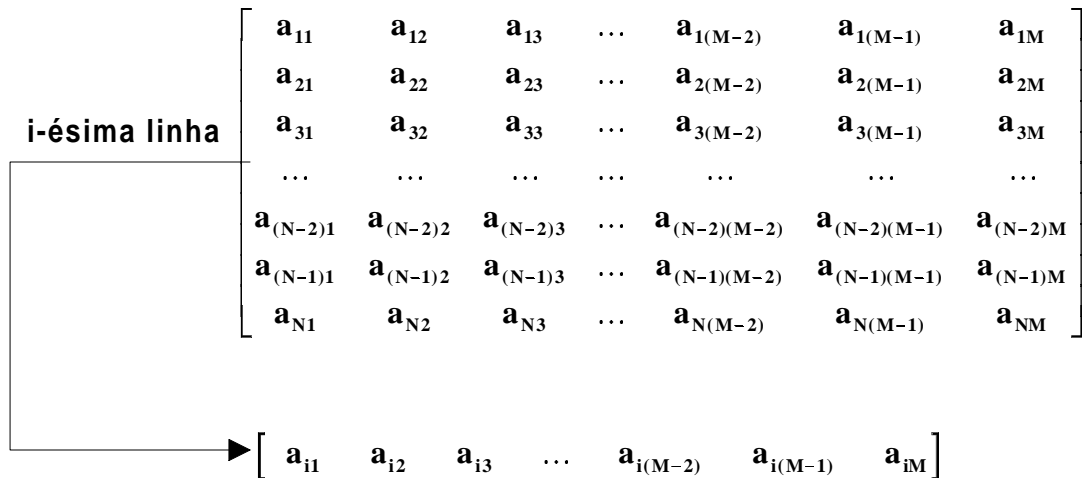
% Obtenho o limite máximo das fdps (e consequentemente o mínimo por simetria)
maxmax1=max(max(abs(máximo1)));
maxmax2=max(max(abs(máximo2)));
maxmax(j)=max([maxmax1 maxmax2]);
minmin=-maxmax;

end

limmax=max(maxmax);

```

Rotina LINHA.M: Obtém um vetor que é a linha de uma matriz. Dada uma matriz com **N** linhas e **M** colunas a obtenção da *i*-ésima linha é representado pela figura adiante:



```

function saída=linha(matriz,linha)
% SINTAXE: saída=linha(matriz,linha)
%
% DESCRIÇÃO:    Obtém um vetor que é a linha de uma matriz.
%

```

```

% PARÂMETROS DE ENTRADA:
%
% matriz      -      matriz de entrada.
% linha       -      número da linha da matriz que se deseja obter.
%
% PARÂMETROS DE SAÍDA:
%
% saída       -      vetor com a linha da matriz.
%
% EXEMPLO:      Para se obter a segunda linha da matriz abaixo:
%
%                               matriz=[ 0 1 0 1 0
%                                       1 1 1 0 0
%                                       0 1 1 0 0 ];
%
%                               Chama-se a função:
%
%                               saída=linha(matriz,2)
%
%                               Obtém-se:
%
%                               saída=[ 1 1 1 0 0 ]
%
%
% Obtém o número de linhas e colunas da matriz
[linhas,colunas]=size(matriz);

% Copia a linha da matriz no vetor saída
saída=matriz(linha,1:colunas);

```

Rotina MAXMIN.M: Obtém os limites máximos e mínimos de todas as funções densidade de probabilidade. Utilizado pela rotina que calcula a probabilidade de erro de bit pelo método de Nazari.

```

function [máximo,mínimo,caso]=maxmin(Oixm)
% SINTAXE: [máximo,mínimo,caso]=maxmin(Oixm)
%
% DESCRIÇÃO:      Obtém os limites máximos e mínimos de todas as funções densidade
%                  de probabilidade. Utilizado pela rotina que calcula a probabilidade de
%                  erro de bit pelo método de Nazari.
%
% PARÂMETROS DE ENTRADA:
%
% Oixm          -      matriz de entrada.
%

```



```

% PARÂMETROS DE SAÍDA:
%
% máximo    -    matriz que contém o valor máximo da função densidade de pro-
%                babilidade da interferência devido a um bit da seqüência in-
%                terferente.
% mínimo    -    matriz que contém o valor mínimo da função densidade de pro-
%                babilidade da interferência devido a um bit da seqüência in-
%                terferente.
% caso      -    matriz com escalares que indicam em qual dos casos os valores
%                armazenados nas matrizes máximo e mínimo foram obtidos. O es-
%                calar 1 indica que o caso obtido foi em que o bit atual e o
%                consecutivo são iguais e o escalar 2 indica que o caso obtido
%                foi em que o bit atual e o consecutivo são diferentes.
%
%
% Obtém o número de usuários e o comprimento das seqüências
[K,N]=size(Oixm);

for i=1:K,
    for l=1:N,
        % Garante a periodicidade do cálculo
        if l==N,
            A=Oixm(i,N);
            B=Oixm(i,1);
        else
            A=Oixm(i,l);
            B=Oixm(i,l+1);
        end
        % Verifica em qual caso (qual função densidade de probabilidade) se encaixam os
        % valores obtidos no item anterior e obtém os limites mínimo e máximo das fun-
        % ções densidade de probabilidade
        if(A==B),
            caso(i,l)=1;
            máximo(i,l)=abs(A);
            mínimo(i,l)=-máximo(i,l);
        else
            caso(i,l)=2;
            mínimo(i,l)=min(A,B);
            máximo(i,l)=max(A,B);
            if abs(máximo(i,l))<abs(mínimo(i,l)),
                buffer=máximo(i,l);
                máximo(i,l)=-mínimo(i,l);
                mínimo(i,l)=-buffer;
            end
        end
    end
end
end

```

Rotina NAZARI.M: Calcula e integra a função densidade de probabilidade pelo método de Nazari. A seção relativa às simulações possui uma descrição bem detalhada, inclusive com exemplos da execução deste programa.

```
function[pe,integridade1,integridade2,integridade3,integridade4]=nazari(configuração,seqüências,amostras,limitemax,inicio)
```

```
% SINTAXE:
```

```
% [pe,integridade1,integridade2,integridade3,integridade4]=nazari(configuração,seqüências,  
% amostras,limitemax,inicio)
```

```
%
```

```
% DESCRIÇÃO:      Calcula e integra a função densidade de probabilidade pelo método de Nazari
```

```
%
```

```
% PARÂMETROS DE ENTRADA:
```

```
%
```

```
% configuração-   vetor com 4 elementos elementos [Tc No usuário Ampli] onde Tc  
%                é o tempo de um chip, No é uma constante (No/2 corresponde a  
%                variância do ruído branco gaussiano), usuário determina qual  
%                dos vetores dentre as seqüências é utilizada pelo receptor e  
%                Ampli é um vetor com a amplitude de cada um dos sinais inter-  
%                ferentes na entrada do receptor.
```

```
% seqüências -    vetor com as seqüências utilizadas pelos usuários. Possui nú-  
%                mero de linhas correspondendo ao número de usuários e número  
%                de colunas correspondendo ao comprimento das seqüências.  
%                Todas as seqüências devem estar polarizadas.
```

```
% amostras -      quantidade de amostras no intervalo de uma unidade.
```

```
% limitemax -     determina qual o limite máximo de todas as funções densidade  
%                de probabilidade. A partir do parâmetro amostras o programa  
%                determina qual o passo a ser utilizado durante todo o processo  
%                de amostragem. A função LIMNAZAR.M deve ser utilizada para  
%                se determinar este parâmetro.
```

```
% inicio -        variável que pode possuir os valores 1 ou 0. Previne qualquer  
%                falta de energia que possa ocorrer bem como permite que o  
%                usuário interrompa os cálculos podendo continuar em outro mo-  
%                mento sem necessitar executar todos os cálculos novamente.  
%                Durante a primeira execução do programa deve ser inicializado  
%                sempre com o valor 1 e em cada nova execução inicializa-  
%                do com o valor 0.
```

```
%
```

```
% PARÂMETROS DE SAÍDA:
```

```
%
```

```
% pe -            probabilidade de erro de bit pelo método de Nazari.
```

```
% integridade1-   vetor com o valor das integridades de cada uma das funções  
%                densidade de probabilidade dos sinais interferentes (menos a  
%                causada pelo ruído branco gaussiano).
```

```

% integridade2 -      valor da integridade da função densidade de probabilidade da
%                    interferência obtida após a convolução das funções densidade
%                    de probabilidade dos sinais interferentes (menos a causada pe-
%                    lo ruído branco gaussiano) .
% integridade3 -      valor da integridade da função densidade de probabilidade do
%                    ruído branco gaussiano.
% integridade4 -      valor da integridade da função densidade de probabilidade ob-
%                    tida após a convolução das funções densidade de probabilidade
%                    da interferência com a do ruído branco gaussiano.
%

```

```

% Obtém parâmetros de configuração

```

```

Tc=configuração(1);

```

```

No=configuração(2);

```

```

usuário=configuração(3);

```

```

Ampli=configuração(4);

```

```

% Obtém o número e o período das seqüências

```

```

[K,N]=size(seqüências);

```

```

% Calcula o período de um dado

```

```

T=N*Tc;

```

```

% Verifica se é a primeira vez que o programa está sendo executado, se sim cria

```

```

% as variáveis necessárias para o funcionamento correto do programa

```

```

if inicio==1,

```

```

    etapa=1;

```

```

    i=1;

```

```

    Oix=[];

```

```

    Oixc=[];

```

```

    eval(['save posição etapa i Oix Oixc']);

```

```

end

```

```

eval(['load posição']);

```

```

posi=i;

```

```

% Gera uma matriz com todos os Oix e Oixc. Cada linha corresponde ao resultado

```

```

% obtido para um usuário interferente

```

```

if etapa==1,

```

```

    counter=1;

```

```

    total1=(K-posi+1)*N

```

```

    t0=clock;

```

```

    for i=posi:K,

```

```

        for l=1:N,

```

```

            Oix(i,l)=oab(l,linha(seqüências,i),linha(seqüências,usuário));

```

```

            Oixc(i,l)=oabc(l,linha(seqüências,i),linha(seqüências,usuário));

```

```

            if counter==10,

```

```

total1=total1-1
etime(clock,t0)
t0=clock;
counter=1;
else
total1=total1-1;
counter=counter+1;
end
eval(['save posição etapa i Oix Oixc']);
end
end
etapa=2;
posi=1;
eval(['save posição etapa i Oix Oixc']);
end

% Obtém o valor máximo (em módulo) e o mínimo (em módulo) de todas as fdps pares
[máximo1,mínimo1,caso1]=maxmin(Oix);

% Obtém o valor máximo (em módulo) e o mínimo (em módulo) de todas as fdps ímpa-
% res
[máximo2,mínimo2,caso2]=maxmin(Oixc);

% Calcula as abcissas e salva no arquivo abcissai.mat
passo=1/amostras;
amostragem=2*round(limitemax/passo);
abcissa=passo*([1:amostragem/2]-0.5);
abcissa=[-fliplr(abcissa) abcissa];
eval(['save abcissai abcissa']);

% Obtém todas as fdps para o caso Oix (dois dados subsequentes iguais)
if etapa==2,
counter=1;
total2=(K-posi+1)*N
t0=clock;
for i=posi:K,
inicio=0;
for l=1:N,
A=mínimo1(i,l);
B=máximo1(i,l);
if caso1(i,l)==1,
% Se Oix(l)=Oix(l+1) obtém fdp correspondente
cte=pi*abs(B);
for j=1:amostragem,
d=abcissa(j);
if abs(d)<abs(B),
fdp(j)=inv(cte*sqrt(1-(d/B)^2));

```

```

else
    fdp(j)=0;
end
end
else
    % Se Oix(l)~=Oix(l+1) obtém fdp correspondente
    cte=inv(pi*(B-A));
    for j=1:amostragem,
        d=abcissa(j);
        if abs(d)<abs(A),
            fdp(j)=cte*log(abs((B+sqrt(B^2-d^2))/(A+sqrt(A^2-d^2))));
        elseif (abs(A)<abs(d))&(abs(d)<B)
            fdp(j)=cte*log(abs((B+sqrt(B^2-d^2))/d));
        else
            fdp(j)=0;
        end
    end
end
end

if inicio==0,
    interfil=fdp;
    inicio=1;
else
    interfil=interfil+fdp;
end

clear fdp;
if counter==10,
    total2=total2-1
    etime(clock,t0)
    t0=clock;
    counter=1;
else
    total2=total2-1;
    counter=counter+1;
end
end
interfil=inv(N)*interfil;
arquivo=['i' int2str(i)];
eval(['save ' arquivo ' interfil']);
clear interfil;
eval(['save posição etapa i Oix Oixc']);
end
etapa=3;
posi=1;
eval(['save posição etapa i Oix Oixc']);
end

```

```

% Obtém todas as fdps para o caso Oixc (dois dados subsequentes diferentes)
if etapa==3,
    counter=1;
    total3=(K-posi+1)*N
    t0=clock;
    for i=posi:K,
        inicio=0;
        for l=1:N,
            A=mínimo2(i,l);
            B=máximo2(i,l);
            if caso2(i,l)==1,
                % Se Oixc(l)=Oixc(l+1) obtém fdp correspondente
                cte=pi*abs(B);
                for j=1:amostragem,
                    d=abcissa(j);
                    if abs(d)<abs(B),
                        fdp(j)=inv(cte*sqrt(1-(d/B)^2));
                    else
                        fdp(j)=0;
                    end
                end
            else
                % Se Oixc(l)~=Oixc(l+1) obtém fdp correspondente
                cte=inv(pi*(B-A));
                for j=1:amostragem,
                    d=abcissa(j);
                    if abs(d)<abs(A),
                        fdp(j)=cte*log(abs((B+sqrt(B^2-d^2))/(A+sqrt(A^2-d^2))));
                    elseif (abs(A)<abs(d))&(abs(d)<B)
                        fdp(j)=cte*log(abs((B+sqrt(B^2-d^2))/d));
                    else
                        fdp(j)=0;
                    end
                end
            end
        end
    end

    if inicio==0,
        interfil=fdp;
        inicio=1;
    else
        interfil=interfil+fdp;
    end

    clear fdp;
    if counter==10,
        total3=total3-1

```

```

etime(clock,t0)
t0=clock;
counter=1;
else
total3=total3-1;
counter=counter+1;
end
end
interfil=inv(N)*interfil;
arquivo=['i' int2str(i) 'c'];
eval(['save ' arquivo ' interfil']);
clear interfil;
eval(['save posição etapa i Oix Oixc']);
end
etapa=4;
eval(['save posição etapa i Oix Oixc']);
end

```

% Calcula a função densidade de probabilidade da interferência do i-ésimo usuário
 % rio

```

for i=1:K,
arquivo=['i' int2str(i)];
eval(['load ' arquivo]);
interfi=0.5*interfil;
eval(['load ' arquivo ' c']);
interfi=interfi+0.5*interfil;
arquivo=['int' int2str(i)];
eval(['save ' arquivo ' interfi']);
clear interfi;
end

```

% Desnormalização das funções densidade de probabilidade da interferência do
 % i-ésimo usuário

```

cte=(Ampli*T)/(2*N);
abcissa=abs(cte)*abcissa;
passo=abs(cte)*passo;

```

% Calcula as novas abcissas cujos valores correspondentes nas ordenadas serão
 % obtidas por interpolação

```

numcol=length(abcissa);
nabcissa=passo*[1:(numcol/2)-1];
nabcissa=[-fliplr(nabcissa) 0 nabcissa];
eval(['save nabcissa nabcissa']);

```

% Por interpolação obtém o ponto de amostragem na origem e em novas posições

```

for i=1:K,
arquivo=['int' int2str(i)];

```

```

eval(['load ' arquivo]);
interfi=inv(cte)*interp1(abcissa,interfi,nabcissa);
% Teoricamente a integral da fdp tem que ser igual a 1. A implementação com-
% putacional e as singularidades presentes nas fdp introduzem erros que alte-
% ram o resultado obtido
arquivo=['int' int2str(i)];
eval(['save ' arquivo ' interfi']);
integridade1(i)=passo*sum(interfi);
clear interfi;
end
abcissa=nabcissa;

% Calcula a função densidade de probabilidade da interferência na saída do cor-
% relator
inicio=0;
for i=1:K,
    if i~=usuário,
        arquivo=['int' int2str(i)];
        eval(['load ' arquivo]);
        if inicio==0,
            interf=interfi;
            inicio=1;
        else
            interf=passo*conv(interf,interfi);
        end
    end
end
interf=abs(interf);
eval(['save interf interf']);
integridade2=passo*sum(interf);
eval(['save passo passo']);

% Calcula as abcissas e salva no arquivo abcissa.mat
clear abcissa;
numcol=length(interf);
abcissa=passo*[1:numcol/2];
abcissa=[-fliplr(abcissa) 0 abcissa];
eval(['save abcissa abcissa']);

% Calcula a função densidade de probabilidade do ruído branco gaussiano norma-
% lizado
var=No*T/4;
abcissag=passo:passo:7*var;
abcissag=[-fliplr(abcissag) 0 abcissag];

numcol=length(abcissag);
eval(['save abcissag abcissag']);

```



```

for i=1:numcol,
    gaussian(i)=inv(var*sqrt(2*pi))*exp(-(abscissag(i)^2)/(2*var^2));
end
eval(['save gaussian gaussian']);
integridade3=sum(gaussian*passo);

taminter=length(interf)
tamgauss=length(gaussian)

% Calcula a convolução das fdp das interferências com a fdp do ruído branco
% gaussiano normalizado
interff=passo*conv(interf,gaussian);
eval(['save interff interff']);
integridade4=passo*sum(interff);

% Calcula as abscissas e salva no arquivo abscissaf.mat
numcol=length(interff);
abscissaf=passo*[1:numcol/2];
abscissaf=[-fliplr(abscissaf) 0 abscissaf];
eval(['save abscissaf abscissaf']);

% Calcula a probabilidade de erro de bit através de uma integração definida na
% função densidade de probabilidade.
inicinteg=(Ampli*T)/2;
reginteg=find(abscissaf>=inicinteg);
tamreginteg=length(reginteg);
if tamreginteg==0,
    pe=0;
else
    erroalg=((interff(reginteg(1))+interff(reginteg(1)-1))/2)*(abscissaf(reginteg(1))-inicinteg);
    pe=passo*sum(interff(reginteg))+erroalg;
end

```

Rotina OAB.M: Calcula a função de correlação cruzada periódica par $\theta_{a,b}(\mathbf{l})$ entre duas seqüências. A função é definida pela equação:

$$\theta_{a,b}(\mathbf{l}) = C_{a,b}(\mathbf{l}) + C_{a,b}(\mathbf{l} - \mathbf{N})$$

```

function [saída]=oab(l,seqa,seqb)
% SINTAXE: [saída]=oab(l,seqa,seqb)
% -----
%
% DESCRIÇÃO:      Calcula a função de correlação cruzada periódica par entre

```

```

%                                duas seqüências.
% -----
%
% PARÂMETROS DE ENTRADA:
%
% seqa      -      seqüência a. Esta seqüência deve estar polarizada.
% seqb      -      seqüência b. Esta seqüência deve estar polarizada
% l         -      deslocamento relativo entre as seqüências, em termos de tempo
%                                equivale a l.Tc, onde Tc é o tempo de um chip.
%
% PARÂMETROS DE SAÍDA:
%
% saída     -      correlação cruzada periódica par entre as duas seqüências
%
%
% Obtém o número e o período das seqüências
[K,N]=size(seqa);

saída=cab(l,seqa,seqb)+cab(l-N,seqa,seqb);

```

Rotina OABC.M: Calcula a função de correlação cruzada periódica ímpar $\hat{\theta}_{a,b}(l)$ entre duas seqüências. A função é definida pela equação:

$$\hat{\theta}_{a,b}(l) = C_{a,b}(l) - C_{a,b}(l - N)$$

```

function [saída]=oabc(l,seqa,seqb)
% SINTAXE: [saída]=oabc(l,seqa,seqb)
% -----
%
% DESCRIÇÃO:      Calcula a função de correlação cruzada periódica ímpar entre
%                                duas seqüências.
% -----
%
% PARÂMETROS DE ENTRADA:
%
% seqa      -      seqüência a. Esta seqüência deve estar polarizada.
% seqb      -      seqüência b. Esta seqüência deve estar polarizada
% l         -      deslocamento relativo entre as seqüências, em termos de tempo
%                                equivale a l.Tc, onde Tc é o tempo de um chip.
%
% PARÂMETROS DE SAÍDA:
%
% saída     -      correlação cruzada periódica ímpar entre as duas seqüências

```

```
%
% Obtém o número e o período das seqüências
[K,N]=size(seqa);

saída=cab(1,seqa,seqb)-cab(1-N,seqa,seqb);
```

Rotina PAPREF.M: Obtém todos os pares preferenciais a partir de um polinômio primitivo. Como a função de correlação cruzada periódica entre pares preferenciais possui apenas três valores determinísticos (ver Apêndice B), o programa procura os pares preferenciais utilizando o método da exaustão, isto é, calcula todas as correlações cruzadas periódicas entre todas as combinações possíveis de todas as seqüências e verifica quais seqüências satisfazem às condições de um par preferencial.

```
function [pares,saída]=parpref(polinômio)
% SINTAXE: [pares,saída]=parpref(polinômio)
%
% DESCRIÇÃO:      Obtém todos os pares preferenciais a partir de um polinômio pri-
%                  mitivo.
%
% PARÂMETROS DE ENTRADA:
%
% polinômio      -   vetor com os n+1 coeficientes do polinômio de grau n. Portan-
%                  to, dado o polinômio  $1+a_1.x^1+a_2.x^2+...+a_n.x^n$  o vetor deve
%                  ser [1 a1 a2 ... an].
%
% PARÂMETROS DE SAÍDA:
%
% pares          -   matriz contendo para cada linha, na primeira e na segunda co-
%                  lunas, a indicação das linhas da matriz saída que contém um par
%                  preferencial.
% saída          -   matriz contendo todas as seqüências de máximo comprimento ci-
%                  clicamente distintas. Cada linha contém uma seqüência. As seqüên-
%                  cias obtidas não estão polarizadas. Para extrair uma seqüência da
%                  matriz pode-se utilizar o programa LINHA.M.
%
% EXEMPLO:       Para gerar a partir do polinômio  $f(X)=1+X^2+X^3$  as matrizes com as
%                  seqüências de máximo comprimento e a que indica quais são os pares
%                  preferenciais digitar:
%
%                  [pares,saída]=parpref([1 0 1 1])
%
%
% Obtém o grau do polinômio
grau=length(polinômio)-1;
```

```

% Calcula o período das seqüências
período=2^grau-1;

% Gera um vetor com as realimentações do registrador de deslocamento a partir
% do polinômio característico
for i=1:grau,
    realimentação(i)=polinômio(i+1);
end

% Gera a tabela de coconjuntos ciclotômicos
tabela=cocosets(primo(período),período);

% Obtém a dimensão dos coconjuntos ciclotômicos
[linhas,colunas]=size(tabela);

% Gera um vetor com condições iniciais tais que a função geradora é igual ao
% inverso da função característica.
tamreg=length(realimentação);
for i=1:tamreg,
    inicial(i)=0;
end
inicial(tamreg)=1;

% Obtém uma das seqüências de máximo comprimento a partir do polinômio primi-
% tivo fornecido
seqpri=regdesl(realimentação,inicial,período);

% Realiza as dizimações e obtém todas as seqüências de máximo comprimento pos-
% síveis
for i=1:linhas,
    seqdiz=dizima(seqpri,tabela(i,1),período);
    for j=1:length(seqdiz)
        saída(i,j)=seqdiz(j);
    end
end

% Obtém o tamanho da matriz que contém as seqüências de máximo comprimento
[linhas,colunas]=size(saída);

linhapares=0;

% Calcula os 3 possíveis valores de correlação cruzada existentes entre pares
% preferenciais
valor1=-1;
valor2=-1+2^((grau+1)/2);
valor3=-1-2^((grau+1)/2);

```

```

% Procura pelo método de exaustão os pares preferenciais
for linhatural=1:linhas-1,
    for linhabusca=linhatural+1:linhas,
        correlação=cabp(linha(saída,linhatural),linha(saída,linhabusca));
        tvalor1=length(find(correlação==valor1));
        tvalor2=length(find(correlação==valor2));
        tvalor3=length(find(correlação==valor3));
        if (tvalor1+tvalor2+tvalor3)==colunas,
            linhapares=linhapares+1;
            pares(linhapares,1)=linhatural;
            pares(linhapares,2)=linhabusca;
        end
    end
end
end

```

Rotina PIJ.M: Calcula a função pix entre duas seqüências não polarizadas seqa e seqb.

```

function [saída]=pij(seqa,seqb,gama)
% SINTAXE: [saída]=pij(seqa,seqb,gama)
%
% DESCRIÇÃO:      Calcula a função pij entre duas seqüências não polarizadas seqa e
%                  seqb.
%
% PARÂMETROS DE ENTRADA:
%
% seqa            -      vetor com a primeira seqüência. O vetor deve conter a seqüên-
%                  cia polarizada durante um período.
% seqb            -      vetor com a segunda seqüência. O vetor deve conter a seqüên-
%                  cia polarizada durante um período.
% gama            -      faixa de deslocamento no tempo entre seqüências definido como
%                  sendo -gama<=tal<=+gama,
%
% PARÂMETROS DE SAÍDA:
%
% saída           -      um escalar com o resultado da função pix entre as duas seqüências
%
%
% Obtém o período da seqüência
N=length(seqa);

% Inicializa as constantes
A=2-2*gama+2*(gama^2)/3;
B=gama*(1-2*gama/3);
C=gama^2/3;

```

```

c0=cab(0,seqa,seqb);
c1=cab(1,seqa,seqb);
c_1=cab(-1,seqa,seqb);
cN_1=cab(N-1,seqa,seqb);
c1_N=cab(1-N,seqa,seqb);

% Calcula pix para o par de seqüências desejado
saida=A*c0^2+B*c0*(c1+c_1)+C*(c1^2+c_1^2+c1_N^2+cN_1^2);

```

Rotina PIJSUBOT.M: Ordena as fases de uma família de seqüências de modo a minimizar a somatória dos pix. Esta solução não é ótima mas subótima. Maiores detalhes se encontram no texto.

```

function [saida]=pijsubot(familia,gama)
% SINTAXE: [saida]=pijsubot(familia,gama)
%
% DESCRIÇÃO:      Ordena as fases de uma família de seqüências de um modo a minimi-
%                  zar a somatória dos pix. Esta solução não é ótima mas subótima.
%
% PARÂMETROS DE ENTRADA:
%
% familia      -      família de seqüências(polarizadas)
% gama        -      faixa de deslocamento no tempo entre seqüências definido como
%                  sendo -gama<=tal<=+gama,
%
% PARÂMETROS DE SAÍDA:
%
% saida        -      a família de seqüências com as fases subotimizadas
%
%
% Obtém o tamanho da família e o período das seqüências
[K,N]=size(familia);

saida=[];

for i=1:K;
    seq_analisada=linha(familia,i);
    if i==1,
        seq_subotima=seq_analisada;
    else
        % Obtém o tamanho da família e o comprimento das seqüências já subotimizadas
        [K2,N2]=size(saida);
        for l=1:N,
            pij_analisada=0;

```

```

for j=1:K2,
    pij_analisada=pij_analisada+pij(seq_analisada,linha(saida,j),gama);
end
% Verifica se a fase analisada é melhor que a fase anterior
if l==1,
    pij_subotimo=pij_analisada;
    seq_subotima=seq_analisada;
elseif pij_analisada<pij_subotimo,
    pij_subotimo=pij_analisada;
    seq_subotima=seq_analisada;
end
% Rotaciona a seqüência analisada
seq_analisada=[seq_analisada(N) seq_analisada(1:N-1)];
end
end
saida=[saida
        seq_subotima];
end

```

Rotina PIORCASO.M: Calcula a probabilidade de erro de bit no pior caso e no melhor caso.

```

function [pe_pior,pe_melhor]=piorcaso(configuração,seqüências)
% SINTAXE: [pe_pior,pe_melhor]=piorcaso(configuração,seqüências)
%
% DESCRIÇÃO:      Calcula a probabilidade de erro de bit no pior caso e no melhor caso
%
% PARÂMETROS DE ENTRADA:
%
% configuração-   vetor com 4 elementos [Tc No usuário Ampli] onde Tc é o tempo
%                  de um chip, No é uma constante (No/2 corresponde a variância
%                  do ruído branco gaussiano), usuário determina qual dos vetores
%                  dentre as seqüências é utilizada pelo receptor e Ampli é um
%                  vetor com a amplitude de cada um dos sinais interferentes na
%                  entrada do receptor.
% seqüências -    vetor com as seqüências utilizadas pelos usuários. Possui nú-
%                  mero de linhas correspondendo ao número de usuários e número
%                  de colunas correspondendo ao comprimento das seqüências.
%                  Todas as seqüências devem estar polarizadas.
%
% PARÂMETROS DE SAÍDA:
%
% pe_pior        -   probabilidade de erro de bit no pior caso.
% pe_melhor      -   probabilidade de erro de bit no melhor caso.
%
% Obtém parâmetros de configuração

```

```

Tc=configuração(1);
No=configuração(2);
usuário=configuração(3);
A=configuração(4);

% Obtém o número e o período das seqüências
[K,N]=size(seqüências);

% Calcula o período de um dado
T=N*Tc;

% Obtém a seqüência do receptor
seqüência_usuário=linha(seqüências,usuário);

% Procura o caso em que ocorre a maior interferência entre as seqüências (por
% simetria obtém-se a menor interferência)
Interf=0;
for i=1:K,
    if i~=usuário,
        seqüência_interf=linha(seqüências,i);
        máximo=0;
        for j=0:N-1,

cálculo=abs(Rki(j*Tc,Tc,seqüência_interf,seqüência_usuário))+abs(Rkic(j*Tc,Tc,seqüência_i
nterf,seqüência_usuário));
            if cálculo>máximo,
                máximo=cálculo;
            end
        end
        Interf=Interf+máximo;
    end
end

% Calcula o melhor caso e o pior caso
pe_pior=1/2*erfc((2*A/No*(1-Interf/T))/sqrt(2));
pe_melhor=1/2*erfc((2*A/No*(1+Interf/T))/sqrt(2));

```

Rotina POLARIZA.M: Polariza uma matriz ou vetor. Para uma seqüência **a** com elementos no conjunto {0,1} mapeia os elementos para o conjunto {1,-1} obtendo uma seqüência **b**. Supondo que um elemento qualquer da matriz possa ser representado pelo símbolo **a_{xy}** onde **a** é o elemento em si e **x** e **y** a linha e coluna em que se encontra posicionado o elemento na matriz, a rotina implementa a função abaixo em cada elemento da matriz:

$$\mathbf{b}_{xy} = (-1)^{a_{xy}}$$


```

function saída=polariza(matriz)
% SINTAXE: saída=polariza(matriz)
%
% DESCRIÇÃO:      Polariza uma matriz ou vetor. Para seqüências com elementos no
%                  conjunto {0,1} mapeia os elementos para o conjunto {1,-1}.
%
% PARÂMETROS DE ENTRADA:
%
% matriz          -      matriz ou seqüência de entrada a ser polarizada.
%
% PARÂMETROS DE SAÍDA:
%
% saída           -      matriz ou seqüência de saída polarizada.
%
% EXEMPLO:        Para se polarizar a matriz abaixo:
%
%                  matriz=[ 0 1 0 1 0
%                           1 1 1 0 0
%                           0 1 1 0 0];
%
%                  Chama-se a função:
%
%                  saída=polariza(matriz)
%
%                  Obtendo-se:
%
%                  saída=[ 1 -1 1 -1 1
%                          -1 -1 -1 1 1
%                          1 -1 -1 1 1]
%
saída=(-1).^matriz;

```

Rotina PRIMO.M: Retorna um vetor com todos os números primos em relação a variável de entrada n. Se retornar um vetor de comprimento n-2 então n é primo.

```

function saída=primo(n)
% SINTAXE: saída=primo(n)
%
% DESCRIÇÃO:      Retorna um vetor com todos os números primos em relação a n. Se
%                  retornar um vetor de comprimento n-2 então n é primo.
%
% PARÂMETROS DE ENTRADA:
%

```

```

% n          -      escalar cujos números primos em relação a este se deseja obter.
%
% PARÂMETROS DE SAÍDA:
%
% saída      -      vetor com todos os números primos em relação a n.
%

% Loop principal do programa
stackidx=0;
saídaidx=0;
if(n>2)
    for i=2:n-1,
        % Divide n por i e verifica o resto. Se o resto é zero guarda este valor num
        % stack. Se o resto for diferente de zero verifica se este número não é divi-
        % sível por um dos valores do stack, se não então é primo.
        flag=1;
        resto=n-fix(n/i)*i;
        if resto==0,
            stackidx=stackidx+1;
            stack(stackidx)=i;
            flag=0;
        else
            for j=1:stackidx,
                resto=i-fix(i/stack(j))*stack(j);
                if resto==0,
                    flag=0;
                end
            end
        end
        if flag==1,
            saídaidx=sádaidx+1;
            saída(sádaidx)=i;
        end
    end
end
end

```

Rotina PURSLEY.M: Calcula e integra a função densidade de probabilidade pelo método de Pursley.

```

function [saída]=pursley(configuração,seqüências)
% SINTAXE: [saída]=pursley(configuração,seqüências)
%
% DESCRIÇÃO:      Calcula e integra a função densidade de probabilidade pelo método de
Pursley
%

```

```

% PARÂMETROS DE ENTRADA:
%
% configuração-      vetor com 4 elementos elementos [Tc No usuário Ampli] onde Tc
%                   é o tempo de um chip, No é uma constante (No/2 corresponde a
%                   variância do ruído branco gaussiano), usuário determina qual
%                   dos vetores dentre as seqüências é utilizada pelo receptor e
%                   Ampli é um vetor com a amplitude de cada um dos sinais inter-
%                   ferentes na entrada do receptor.
% seqüências -      vetor com as seqüências utilizadas pelos usuários. Possui nú-
%                   mero de linhas correspondendo ao número de usuários e número
%                   de colunas correspondendo ao comprimento das seqüências.
%                   Todas as seqüências devem estar polarizadas.
%
% PARÂMETROS DE SAÍDA:
%
% pe                -      probabilidade de erro de bit pelo método de Pursley.
%
%
% Obtém parâmetros de configuração
Tc=configuração(1);
No=configuração(2);
usuário=configuração(3);
A=configuração(4);

% Obtém o número e o período das seqüências
[K,N]=size(seqüências);

% Calcula o período de um dado
T=N*Tc;

% Calcula a relação sinal para ruído
seqüência_usuario=linha(seqüências,usuário);
SNR=0;
for i=1:K,
    if i~=usuário,
        SNR=SNR+bab(linha(seqüências,i),seqüência_usuario);
    end
end

SNR=1/(6*N^3)*SNR;
SNR=1/sqrt(SNR+No/(A^2*T));

% Obtém a probabilidade de erro de bit através da aproximação gaussiana
saída=0.5*erfc(SNR/sqrt(2));

```

Rotina OSCDMA.M: Calcula e integra a função densidade de probabilidade para o caso quase-síncrono.

```
function [saida]=qscdma(configuração,seqüências,gama)
% SINTAXE: [saida]=qscdma(configuração,seqüências,gama)
%
% DESCRIÇÃO:      Calcula e integra a função densidade de probabilidade para o caso
%                  quase-síncrono.
%
% PARÂMETROS DE ENTRADA:
%
% configuração-   vetor com 4 elementos elementos [Tc No usuário Ampli] onde Tc
%                  é o tempo de um chip, No é uma constante (No/2 corresponde a
%                  variância do ruído branco gaussiano), usuário determina qual
%                  dos vetores dentre as seqüências é utilizada pelo receptor e
%                  Ampli é um vetor com a amplitude de cada um dos sinais inter-
%                  ferentes na entrada do receptor.
% seqüências -    vetor com as seqüências utilizadas pelos usuários. Possui nú-
%                  mero de linhas correspondendo ao número de usuários e número
%                  de colunas correspondendo ao comprimento das seqüências.
%                  Todas as seqüências devem estar polarizadas.
% gama           - faixa de deslocamento no tempo entre seqüências definido como
%                  sendo -gama<=tal<=+gama,
%
% PARÂMETROS DE SAÍDA:
%
% pe             -   probabilidade de erro.
%
%
% Obtém parâmetros de configuração
Tc=configuração(1);
No=configuração(2);
usuário=configuração(3);
A=configuração(4);
%
% Obtém o número e o período das seqüências
[K,N]=size(seqüências);
%
% Calcula o período de um dado
T=N*Tc;
%
% Calcula a relação sinal para ruído
seqüência_usuario=linha(seqüências,usuário);
SNR=0;
for i=1:K,
    if i~=usuário,
```



```

% -----
%
% Da figura percebe-se que n=3 (portanto período=7), o polinômio caracterís-
% tico é f(X)=1+X^2+X^3, que é primitivo e gera seqüências de máximo com-
% primento. Para simular este circuito com as condições iniciais do desenho
% durante 7 interações (um período), digitar:
%
%                               regdesl([0 1 1],[0 0 1],7)
%
%
% Obtém o tamanho do registrador de deslocamento
comprimento=length(realimentação);

% Carrega um vetor com estado corrente (atual) do registrador de deslocamento
for i=1:comprimento,
    estado(i)=inicial(i);
end

% Calcula a saída do registrador de deslocamento primeiro verificando as rea-
% limentações e gerando as saídas conforme os estado atual do conteúdo do re-
% gistrador e depois atualizando o estado atual deste.
for i=1:interações,
    saída(i)=0;
    for j=1:comprimento,
        if realimentação(j)==1,
            if estado(j)==1,
                if saída(i)==0,
                    saída(i)=1;
                else
                    saída(i)=0;
                end
            end
        end
    end
end

% Desloca o estado corrente uma posição para a direita
for j=comprimento:-1:2,
    estado(j)=estado(j-1);
end
estado(1)=saída(i);
end

```

Rotina RFLM: Calcula a função de autocorrelação aperiódica do chip par $\mathbf{R}_\psi(\mathbf{s})$ entre 2 seqüências para chips de assinatura com formas de onda retangulares retornando:

$$\mathbf{R}_\psi(\mathbf{s}) = \mathbf{s}$$

```

function [saída]=rfi(argumento)
% SINTAXE: [saída]=rfi(argumento)
%
% DESCRIÇÃO:      Calcula a função de autocorrelação aperiódica do chip par entre
%                  2 seqüências.
%
% PARÂMETROS DE ENTRADA:
%
% argumento -      atraso relativo entre dois chips
%
% PARÂMETROS DE SAÍDA:
%
% saída      -      autocorrelação aperiódica do chip par.
%

% Atualmente esta função está setada para formas de ondas retangulares
saída=argumento;

```

Rotina RFIC.M: Calcula a função de autocorrelação aperiódica do chip ímpar $\hat{\mathbf{R}}_{\psi}(s)$ entre 2 seqüências para chips de assinatura com formas de onda retangulares retornando:

$$\hat{\mathbf{R}}_{\psi}(s) = \mathbf{T}_c - s$$

```

function [saída]=rfic(argumento,Tc)
% SINTAXE: [saída]=rfic(argumento,Tc)
%
% DESCRIÇÃO:      Calcula a função de autocorrelação aperiódica do chip ímpar entre
%                  2 seqüências.
%
% PARÂMETROS DE ENTRADA:
%
% argumento -      atraso relativo entre dois chips
%
% PARÂMETROS DE SAÍDA:
%
% saída      -      autocorrelação aperiódica do chip ímpar.
%

% Atualmente esta função está setada para formas de ondas retangulares
saída=Tc-argumento;

```

Rotina RKLM: Calcula a função de correlação cruzada parcial contínua no tempo par entre duas seqüências. Esta função é definida pela equação:

$$\mathbf{R}_{i,x}(\tau) = \int_0^{\tau} \mathbf{c}_i(t - \tau) \cdot \mathbf{c}_x(t) \cdot dt$$

```
function [saída]=rki(tal,Tc,seqa,seqb)
% SINTAXE: [saída]=rki(tal,Tc,seqa,seqb)
%
% DESCRIÇÃO:      Calcula a função de correlação cruzada parcial contínua no
%                  tempo par entre as seqüências seqa e seqb.
%
% PARÂMETROS DE ENTRADA:
%
% tal             -      deslocamento no tempo entre as seqüências seqa e seqb.
% Tc              -      tempo de um chip.
% seqa            -      vetor com a primeira seqüência. O vetor deve conter a seqüên-
%                  cia durante um período.
% seqb            -      vetor com a segunda seqüência. O vetor deve conter a seqüên-
%                  cia durante um período.
%
% PARÂMETROS DE SAÍDA:
%
% saída           -      escalar com o resultado da operação de correlação cruzada par-
%                  cial contínua par.
%
N=length(seqa);
l=fix(tal/Tc);
argumento=tal-l*Tc;

saída=cab(l-N,seqa,seqb)*Rfic(argumento,Tc)+cab(l+1-N,seqa,seqb)*Rfi(argumento);
```

Rotina RKIC.M: Calcula a função de correlação cruzada parcial contínua no tempo ímpar entre duas seqüências. Esta função é definida pela equação:

$$\hat{\mathbf{R}}_{i,x}(\tau) = \int_{\tau}^T \mathbf{c}_i(t - \tau) \cdot \mathbf{c}_x(t) \cdot dt$$

```
function [saída]=rkic(tal,Tc,seqa,seqb)
% SINTAXE: [saída]=rkic(tal,Tc,seqa,seqb)
%
% DESCRIÇÃO:      Calcula a função de correlação cruzada parcial contínua no
%                  tempo ímpar entre as seqüências seqa e seqb.
%
% PARÂMETROS DE ENTRADA:
%
% tal             -      deslocamento no tempo entre as seqüências seqa e seqb.
```



```

% Tc      -      tempo de um chip.
% seqa    -      vetor com a primeira seqüência. O vetor deve conter a seqüên-
%            cia durante um período.
% seqb    -      vetor com a segunda seqüência. O vetor deve conter a seqüên-
%            cia durante um período.
%
% PARÂMETROS DE SAÍDA:
%
% saída    -      escalar com o resultado da operação de correlação cruzada par-
%            cial contínua ímpar.
%

N=length(seqa);
l=fix(tal/Tc);
argumento=tal-l*Tc;

saída=cab(l,seqa,seqb)*Rfic(argumento,Tc)+cab(l+1,seqa,seqb)*Rfi(argumento);

```

Rotina WEBER.M: Calcula e integra a função densidade de probabilidade pelo método de Weber

```

function [saída]=weber(configuração,seqüências)
% SINTAXE: [saída]=weber(configuração,seqüências)
%
% DESCRIÇÃO:      Calcula e integra a função densidade de probabilidade pelo método de
% Weber
%
% PARÂMETROS DE ENTRADA:
%
% configuração-   vetor com 4 elementos elementos [Tc No usuário Ampli] onde Tc
%                  é o tempo de um chip, No é uma constante (No/2 corresponde a
%                  variância do ruído branco gaussiano), usuário determina qual
%                  dos vetores dentre as seqüências é utilizada pelo receptor e
%                  Ampli é um vetor com a amplitude de cada um dos sinais inter-
%                  ferentes na entrada do receptor.
% seqüências -    vetor com as seqüências utilizadas pelos usuários. Possui nú-
%                  mero de linhas correspondendo ao número de usuários e número
%                  de colunas correspondendo ao comprimento das seqüências.
%                  Todas as seqüências devem estar polarizadas.
%
% PARÂMETROS DE SAÍDA:
%
% pe            -      probabilidade de erro de bit pelo método de Weber.
%

```

```

% Obtém parâmetros de configuração
Tc=configuração(1);
No=configuração(2);
usuário=configuração(3);
A=configuração(4);

% Obtém o número e o período das seqüências
[K,N]=size(seqüências);

% Calcula o período de um dado
T=N*Tc;

% Calcula a relação sinal para ruído
Eb=A^2*N*Tc;
Neq=No;
for i=1:K-1,
    Neq=Neq+A^2*Tc;
end
SNR=sqrt(2*Eb/Neq);

% Obtém a probabilidade de erro de bit através da aproximação gaussiana
saída=0.5*erfc(SNR/sqrt(2));

```

Rotina XORSEQ.M: Realiza a operação XOR entre duas seqüências.

```

function saída=xorseq(seqa,seqb,interações)
% SINTAXE: saída=xorseq(seqa,seqb,interações)
%
% DESCRIÇÃO:      Realiza a operação XOR entre duas seqüências.
%
% PARÂMETROS DE ENTRADA:
%
% seqa           -   vetor com a primeira seqüência. O vetor deve conter a seqüên-
%                   cia durante um período.
% seqb           -   vetor com a segunda seqüência. O vetor deve conter a seqüên-
%                   cia durante um período.
% interações     -   inicializar com o período das seqüências.
%
% PARÂMETROS DE SAÍDA:
%
% saída          -   matriz em que cada linha é uma seqüência obtida pela operação
%                   XOR entre as seqüências seqa e seqb. A cada linha uma nova se-
%                   qüência é obtida rotacionando-se uma das seqüências de entra-
%                   da uma posição para a direita.
%

```

```

% Obtém o comprimento das seqüências
comprimento=length(seqa);

% Copia a seqb para um buffer
seqüência=seqb;

% Realiza a operação XOR entre as duas seqüências
for i=1:interações,
    for j=1:comprimento,
        A=seqa(j);
        B=seqüência(j);
        saída(i,j)=(((~A)&B)|(A&(~B)));
    end
% Rotaciona a seqüência uma posição para a direita
ultimo=seqüência(comprimento);
for k=comprimento:-1:2,
    seqüência(k)=seqüência(k-1);
end
seqüência(1)=ultimo;
end

```

14 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Paul Jean E. Jeszensky, "Notas de Aula do Curso de Comunicação por Espalhamento Espectral", Departamento de Engenharia Eletrônica, Universidade de São Paulo, 1991.
- [2] Paul Jean E. Jeszensky, "Notas de Aula do Curso de Comunicação por Espalhamento Espectral: Uma motivação para o estudo de seqüências de códigos", Departamento de Engenharia Eletrônica, Universidade de São Paulo, 1992.
- [3] V. H. MacDonald, "Advanced Mobile Phone Service - The Cellular Concept", Bell System Technical Journal, January 1979, pp 15-41.
- [4] William C. Y. Lee, "Overview of Cellular CDMA", IEEE Transactions on Vehicular Technology, Vol. 40, Nº 2, May 1991, 291-302.
- [5] Michael B. Pursley, "Performance Evaluation for Phase-Coded Spread Spectrum Multiple-Access Communication - Part I: System Analysis", IEEE Transactions on Communications, Vol. COM-25, Nº 8, August 1977, pp. 795-799.
- [6] Michael B. Pursley, "Spread-Spectrum Multiple-Access Communications", New Concepts in Multi-User Communication, Nato Advanced Study Institutes Series, 1981, pp. 139-199.
- [7] Michael B. Pursley and D. V. Sarwate, "Performance Evaluation for Phase-Coded Spread Spectrum Multiple-Access Communication - Part II: Code Sequence Analysis", IEEE Transactions on Communications, Vol. COM-25, Nº 8, August 1977, pp. 800-803.
- [8] Kung Yao, "Error Probability of Asynchronous Spread Spectrum Multiple Access Communication Systems", IEEE Transactions on Communications, Vol. COM-25, Nº 8, August 1977, pp. 803-809.
- [9] Charles L. Weber, Gaylord K. Huth and Bartus H. Batson, "Performance Considerations of Code Division Multiple-Access Systems", IEEE Transactions on Vehicular Technology, Vol. VT-30, Nº 1, February 1981, pp. 3-10.
- [10] Dilip V. Sarwate and Michael B. Pursley, "Crosscorrelation Properties of Pseudorandom and Related Sequences", Proceedings of the IEEE, Vol. 68, No 5, May 1980, pp. 593-619.
- [11] Michael B. Pursley and H. F. A. Roefs, "Numerical Evaluation of Correlation Parameters for Optimal Phases of Binary Shift-Register Sequences", IEEE Transactions on Communications, Vol. COM-27, Nº 10, October 1979, pp. 1597-1604.
- [12] Klein S. Gilhousen, Irwin M. Jacobs, Roberto Padovani, Andrew J. Viterbi, Lindsay A. Weaver Jr. and Charles E. Wheatley III, "On the Capacity of a Cellular CDMA System", IEEE Transactions on Vehicular Technology, Vol. 40, Nº 2, May 1991, pp. 303-312.

- [13] Raymond L. Pickholtz, Laurence B. Milstein and Donald L. Schilling, "Spread Spectrum for Mobile Communications", IEEE Transactions on Vehicular Technology, Vol. 40, Nº 2, May 1991, pp. 313-322.
- [14] James S. Lehnert and Michael B. Pursley, "Error Probabilities for Binary Direct-Sequence Spread-Spectrum Communications with Random Sequence Signatures", IEEE Transactions on Communications, Vol. COM-35, No 1, January 1987, pp. 87-98.
- [15] Nersi Nazari and Rodger E. Ziemer, "Computationally Efficient Bounds for the Performance of Direct-Sequence Spread Spectrum Multiple-Access Communications Systems in Jamming Environments", IEEE Transactions on Communications, Vol. 36, Nº 5, May 1988, pp. 577-587.
- [16] Riccardo De Gaudenzi, Carlos Elia and Roberto Viola, "Bandlimited Quasi-Synchronous CDMA: A Novel Satellite Access Technique for Mobile and Personal Communication Systems", IEEE Journal on Selected Areas in Communications, Vol. 10, Nº 2, February 1992, pp. 328-343.
- [17] Robert A. Scholtz, "The Spread Spectrum Concept", IEEE Transactions on Communications, Vol. COM-25, No 8, August 1977, pp. 748-755.
- [18] Athanasios Papoulis, "Probability, Random Variables, and Stochastic Processes", New York, McGraw-Hill Book Co., Second Edition, 1982
- [19] K.Sam.Shanmugan and A.M.Breipohl, "Random Signals: Detection, Estimation and Data Analysis", John Wiley & Sons, 1988
- [20] Murray R. Spiegel, "Manual de Fórmulas e Tabelas Matemáticas", McGraw-Hill
- [21] Skwirzynski, J.K. (editor), New Concepts in Multi-User Communication, NATO Advanced Study Institutes Series, Sijthoff Noordhoff International Publishers, 1981; sessão devida a Solomon W. Golomb, "Correlation Properties of Periodic and Aperiodic Sequences, and Applications to Multi-User Systems"
- [22] Robert K.Morrow,Jr and James S. Lehnert,"Bit-to-Bit Dependence in Slotted DS/SSMA Packet Systems with Random Signature Sequences", IEEE Transactions on Communications, Vol. 37, Nº 10, October 1989, pp. 1052-1061.
- [23] John G. Proakis, "Digital Communications", McGraw-Hill Book Co., 1989
- [24] Paul Jean E. Jeszensky e John R.Fitzgerald Jr., "Probabilidade de erro de bit em sistemas Spread Spectrum Quase-Sincronos", submetido ao TELEMÓ-96 a ser realizado em 7/1996.
- [25] W.W.Peterson, "Error-Correcting Codes", Cambridge, MIT Press, Second Edition, 1972.

[26] "An Overview of the Application of Code Division Multiple Access (CDMA) to Digital Cellular Systems and Personal Cellular Networks", submitted to TIA TR45.5 Subcommittee; 21/05/1992; Qualcomm Incorporated Document Number EX60-10010.