

Códigos Turbo M -ários com Modulação M -PSK

José da Silva Barros e Renato Baldini Filho

Resumo—O objetivo deste trabalho é apresentar classes de códigos turbo M -ários. O esquema de codificação turbo consiste em dois codificadores componentes multiníveis, concatenados em paralelo via um entrelaçador, mapeados através da constelação M -PSK (M -ary Phase Shift Keying) e transmitidos pelo canal de comunicação AWGN (*Aditive White Gaussian Noise*). O esquema de decodificação turbo é formado por um algoritmo de decodificação M -ário, baseado na decodificação MAP (*máximo a posteriori*) via a decodificação iterativa. Os resultados das simulações mostram que os códigos turbo M -ário são mais eficientes que os códigos turbo binários.

Palavras-Chave—Códigos Turbo M -ário, Algoritmo de Decodificação Iterativa.

Abstract—The aim of this work is to present classes of M -ary turbo codes. The encoding scheme consists of two M -ary systematic recursive convolutional encoders separated by s -random interleaver. The decoding scheme is based on the maximum a posteriori decoding algorithm - MAP. Simulation results show that M -ary turbo codes are more efficient than the binary turbo codes.

Keywords— M -ary Turbo Codes, Iterative Decoding Algorithm.

I. INTRODUÇÃO

Desde que Ungerboeck introduziu os conceitos de códigos treliça, em 1982 [3], os códigos turbo binários apresentados por C. Berrou, A. Glavieux e P. Thitimajashima em 1993 [1], representam a mais importante descoberta na área de teoria da informação e codificação, por apresentar desempenho muito próximo ao limitante de Shannon para o canal AWGN (*Aditive White Gaussian Noise*). A estrutura entrelaçada dos códigos permite uma decodificação próxima a decodificação de máximo *a posteriori* (MAP) via decodificação iterativa.

A maioria dos códigos turbo pesquisados consideram códigos componentes binários concatenados via um entrelaçador aleatório de N bits [1], [2], [4] e [5]. Neste artigo, consideramos os códigos turbo construídos para códigos componentes não binários (ou M -ários), concatenados via um entrelaçador aleatório de N símbolos e mapeado em uma constelação apropriada M -PSK, para transmissão através de um canal AWGN. A decodificação usada para investigar o desempenho dos códigos turbo M -ários é a decodificação de máximo *a posteriori* também via decodificação iterativa.

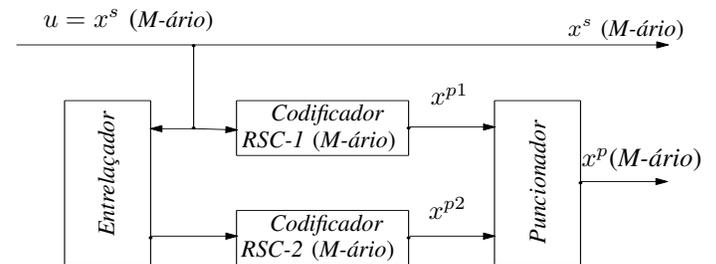
O casamento natural do tamanho da constelação com o tamanho do alfabeto, dá maior flexibilidade a combinação: complexidade de decodificação, capacidade de correção de erro, cardinalidade do alfabeto, largura de banda e mapeamento de bits para símbolos quando analisamos o desempenho do esquema turbo multinível. Assim, os esquemas

José da Silva Barros e Prof. Dr. Renato Baldini Filho, Faculdade de Engenharia Elétrica e de Computação, Departamento de Telecomunicações, UNICAMP, Campinas, Brasil, E-mails: barros@decom.fee.unicamp.br, baldini@decom.fee.unicamp.br. Este trabalho foi financiado pela FAPEAL(E.20031029439-0).

turbo M -ários se apresentam mais eficientes, com relação aos parâmetros mencionados acima, do que os esquemas turbo binários.

II. CODIFICAÇÃO TURBO M -ária

O esquema de codificação turbo M -ário é constituído por dois codificadores convolucionais geralmente idênticos, de taxa $1/2$, separados por um entrelaçador de N símbolos, juntamente com um mecanismo opcional de punção. Esses dois codificadores constituintes são sistemáticos recursivos e utilizados em paralelo, conforme a Figura 1.



- $u = x^s$ Sequência de informação
- x^{p1} Sequência de paridade do codificador RSC - 1
- x^{p2} Sequência de paridade do codificador RSC - 2
- x^p Sequência de paridade do puncionador

Fig. 1. Esquema de codificação turbo.

A. Codificador Convolucional Sistemático Recursivo

Seja \mathbb{Z}_M um anel de inteiros comutativo com identidade multiplicativa e seja $L_r(D)$ o anel formado por polinômios racionais do tipo $q(D)/p(D)$, onde $q(D)$ e $p(D)$ são polinômios com coeficientes em \mathbb{Z}_M , expressos pela série de Laurent [6], [7] e [9], $f(D) = \sum_{i=0}^n f_i D^i$, com $f_i \in \mathbb{Z}_M$ e $p(0) \neq 0$ em \mathbb{Z}_M - chamado anel das funções realizáveis sobre \mathbb{Z}_M .

Assim, um codificador convolucional sistemático recursivo RSC (*Recursive Systematic Convolutional*), com taxa de codificação k/n sobre o anel de funções realizáveis $L_r(D)$, é um mapeamento linear expresso como

$$v(D) = u(D)G_r(D),$$

onde $G_r(D)$ é uma matriz $k \times n$ na forma sistemática

$$G_r(D) = \begin{bmatrix} I_k & X(D) \end{bmatrix},$$

e $u(D)$ é o polinômio correspondente a sequência de informação. Observe que I_k é uma matriz identidade $k \times k$ e $X(D)$ é uma matriz $k \times (n-k)$ com elementos cuja representação é $q(D)/p(D)$ em $L_r(D)$.

Portanto, o conjunto

$$C = \{u(D)g(D) \mid u(D) \in L_r(D)^k\},$$

é um código *RSC* com taxa de codificação k/n sobre \mathbb{Z}_M , onde $g(D) = G_r(D)$ é a matriz geradora com elementos em $L_r(D)$.

B. Entrelaçador

Os entrelaçadores, usados com grande frequência nos meios de comunicação, têm por objetivo espalhar erros que ocorrem em surtos (*burst*), causados pelo ruído impulsivo e pelo desvanecimento (*fading*). Usamos em nossa simulação o entrelaçador *s*-aleatório [8], pois este apresenta melhores propriedades aleatórias.

C. Puncionador

A função do puncionador é apagar periodicamente símbolos de redundância pré-selecionados dos codificadores, aumentando a taxa de codificação. Quando usamos o puncionamento, são mapeados N símbolos de informação em $2N$ símbolos do código, enquanto que, sem o puncionamento, são mapeados N símbolos de informação em $3N$ símbolos do código.

D. Modulador *M*-PSK

A seqüência de entrada do modulador é composta por símbolos pertencentes ao anel de inteiros \mathbb{Z}_M . Estes símbolos são transformados através da modulação *M*-PSK em sinais do tipo

$$S_i = A \exp\left(j \frac{2\pi i}{M}\right),$$

onde i é um símbolo do alfabeto *M*-ário, $\{0, 1, \dots, M-1\} \in \mathbb{Z}_M$ e A é uma amplitude (raio) do sinal.

A relação fechada que existe entre o anel de inteiros módulo- M e a modulação *M*-PSK, justifica o ganho no desempenho do sistema turbo, referente ao perfeito casamento dos símbolos do alfabeto *M*-ário com a modulação *M*-PSK.

III. ALGORITMO DE DECODIFICAÇÃO

A função do algoritmo *MAP* é fornecer a informação *a posteriori* $L(u_k)$ na saída de cada decodificador, definida da seguinte maneira. Seja

$$L(u_k) = \max(\ln(p(u_k = \theta | y))), \quad (1)$$

o valor máximo do logaritmo natural da probabilidade *a posteriori* de um símbolo decodificado ser $u_k = \theta \in \{0, 1, 2, \dots, M-1\}$, dado que a seqüência de símbolos recebida é $y = (y_1 y_2 \dots y_{k-1} y_k y_{k+1} \dots y_N)$. Note que M é o comprimento do alfabeto de entrada e N é o comprimento da seqüência de informação.

Se as transições entre o estado prévio, $S_{k-1} = s'$, e o estado presente, $S_k = s$, são mutuamente exclusivas (isto é, apenas uma delas pode ter ocorrido na treliça referente ao codificador *RSC*), então usando a regra de *Bayes*

$$L(u_k) = \max \left(\ln \left(\frac{\sum_{(s',s)} p(S_{k-1} = s', S_k = s, y)}{p(y)} \right) \right), \quad (2)$$

onde (s', s) é o conjunto de transições do estado prévio, $S_{k-1} = s'$, para o estado presente, $S_k = s$, que pode ocorrer se o símbolo de entrada u_k for igual a θ .

Como o canal é sem memória e os símbolos da seqüência recebida y são independentes entre si, substituindo $p(y)$ por $p(y)/p(y_k)$ na expressão (2) temos:

$$L(u_k) = \max \left(\ln \left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s) \right) \right), \quad (3)$$

onde $\alpha_k(s)$ e $\beta_k(s)$ são as probabilidades definidas respectivamente como:

$$\alpha_k(s) = \frac{\sum_{\text{todo } s'} \alpha_{k-1}(s') \cdot \gamma_k(s', s)}{\sum_{\text{todo } s \text{ todo } s'} \alpha_{k-1}(s') \cdot \gamma_k(s', s)} \quad \text{e} \quad (4)$$

$$\beta_{k-1}(s) = \frac{\sum_{\text{todo } s} \beta_k(s) \cdot \gamma_k(s', s)}{\sum_{\text{todo } s \text{ todo } s'} \alpha_{k-1}(s') \cdot \gamma_k(s', s)}. \quad (5)$$

Observe que $\alpha_k(s)$ e $\beta_k(s)$ são calculados recursivamente pelas equações (4) e (5) e possuem as condições iniciais.

$$\alpha_0(S_0 = s) = \beta_N(S_N = s) = \begin{cases} 1 & \text{para } s = 0 \\ 0 & \text{para } s \neq 0. \end{cases} \quad (6)$$

Com base nas mesmas hipóteses temos que a probabilidade de transição é dada por

$$\gamma_k(s', s) = p(y_k | x_k) \cdot p(u_k). \quad (7)$$

Assim, apresentamos as expressões de $\alpha_{k-1}(s')$, $\beta_k(s)$ e $\gamma_k(s', s)$, necessárias para calcular a informação *a posteriori* $L(u_k)$, na saída de cada decodificador componente.

IV. DECODIFICAÇÃO ITERATIVA TURBO *M*-ária

O processo de decodificação iterativa turbo definido sobre o anel \mathbb{Z}_M , esquematizado na Figura 2, calcula a informação extrínseca na saída do estágio de decodificação anterior, e a usa como informação *a priori* na entrada do próximo estágio de decodificação.

Sabendo que cada codificador *RSC* do esquema de codificação turbo tem taxa de codificação $1/2$, a palavra código transmitida é dada por $x_k = x_k^s x_k^p = u_k x_k^p$, e a palavra código recebida é dada por $y_k = y_k^s y_k^p$.

Assim, se o canal é sem memória, gaussiano e com modulação *M*-PSK, então, a função densidade de probabilidade é dada por

$$\begin{aligned} p(y_k | x_k) &= \frac{1}{\pi N_0} \cdot \exp \left(\frac{-\|y_k^s - x_k^s\|^2 - \|y_k^p - x_k^p\|^2}{N_0} \right) \\ &= \exp \left(\frac{-\|y_k^s - x_k^s\|^2}{N_0} \right) \cdot \gamma_k^e(s', s), \end{aligned}$$

onde: x_k^s é o símbolo sistemático da palavra código transmitida x_k ; x_k^p é o símbolo de paridade da palavra código transmitida x_k ; y_k^s é o sinal recebido correspondente ao símbolo transmitido x_k^s e y_k^p é o sinal recebido correspondente ao símbolo transmitido x_k^p .

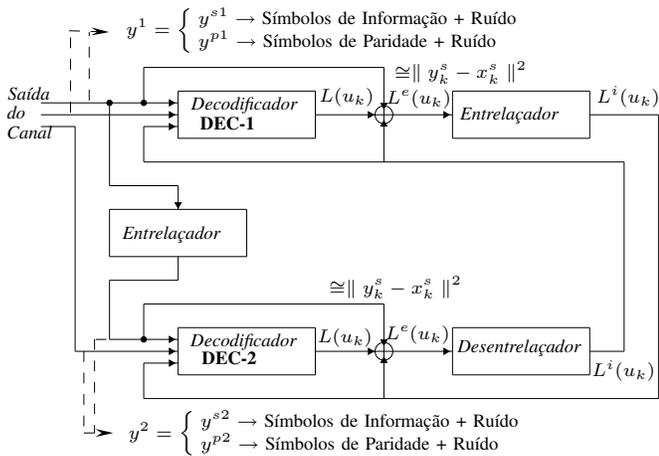


Fig. 2. Decodificação Iterativa Turbo.

Portanto, a informação *a posteriori*, $L(u_k)$, calculada através do algoritmo MAP na saída de cada decodificador componente, pode ser dividida em três termos

$$L(u_k) = L^i(u_k) + \exp\left(\frac{-\|y_k^s - x_k^s\|^2}{N_0}\right) + L^e(u_k), \quad (8)$$

onde $L^i(u_k) = \ln(p(u_k))$ é a informação intrínseca, $\exp\left(\frac{-\|y_k^s - x_k^s\|^2}{N_0}\right)$ é a informação sistemática e

$$L^e(u_k) = \ln\left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \beta_k(s) \cdot \gamma_k^e(s, s')\right), \quad (9)$$

é a informação extrínseca.

Note que a informação extrínseca calculada na saída do primeiro decodificador, após ser entrelaçada, é usada como informação *a priori* na entrada do segundo decodificador. E de forma análoga, a informação extrínseca calculada na saída do segundo decodificador, após ser desentrelaçada, é usada como informação *a priori* na entrada do primeiro decodificador, de acordo com o processo iterativo.

Processo Iterativo: Considere inicialmente o primeiro decodificador componente na primeira iteração. Este decodificador recebe a seqüência de canal, y^1 (ver Figura 2), e produz uma estimativa da informação *a posteriori* $L_{11}(u_k)$ dos símbolos de dados u_k , onde $k \in \{1, \dots, N\}$ e N é o comprimento da seqüência de informação. Note que nesta primeira iteração a informação *a priori* que o primeiro decodificador componente recebe é $\ln(p(u_k = \theta)) = \ln(1/M)$.

O segundo decodificador componente recebe a seqüência de canal y^2 junto com a informação extrínseca entrelaçada do primeiro decodificador componente e fornece uma estimativa da informação *a posteriori* $L_{12}(u_k)$, dos símbolos de dado u_k . Esta informação extrínseca (usada no segundo decodificador) é calculada através da expressão (9) na saída do primeiro decodificador componente.

Na segunda iteração, o primeiro decodificador componente novamente processa sua seqüência recebida de canal y^1 , mas dessa vez, ele também possui a informação *a priori* $L^i(u_k)$.

Esta informação *a priori* é fornecida pela porção extrínseca da informação *a posteriori* $L_{12}(u_k)$, calculada pelo segundo decodificador componente, na primeira iteração. Portanto, este decodificador pode produzir uma melhor informação *a posteriori* $L_{21}(u_k)$, na saída do primeiro decodificador componente, na segunda iteração.

Este processo iterativo continua e, a cada iteração, em média a taxa de erro de bit diminui para uma mesma relação sinal ruído - SNR (*signal-to-noise ratio*).

V. RESULTADOS

Neste artigo, fixamos comprimento da seqüência de informação em $N = 2000$ bits, a taxa de codificação turbo em $1/2$, o número de iterações em 8 e a energia (raio) da constelação *M-PSK* em 1, para cada sistema *M-ário* simulado. O desempenho destes sistemas serão analisados com relação a complexidade de decodificação, a capacidade de correção de erro, a cardinalidade do alfabeto e o mapeamento de bits para símbolos. Note que cada sistema *M-ário*, $M > 2$, processa $2000/\log_2 M$ símbolos e a probabilidade de erro de símbolo é dividida por $\log_2 M$ para dar origem a probabilidade de erro de bit.

A. Esquema Turbo Binário com Modulação B-PSK

Como já existem resultados na literatura ([1], [2], [4] e [5]) sobre o sistema turbo binário, apresentamos nesta seção os resultados simulados que serão usados como referência para efeito de comparação com os resultados dos sistemas turbo ternário, quaternário e 5-ário, apresentados nas próximas seções.

A Figura 3 apresenta as curvas da probabilidade $P_e(b)$ de erro de bit média versus E_b/N_0 para o esquema turbo binários que possui codificadores RSC com $g(D) = \langle 1101|1111 \rangle$ (treliça com $2^3 = 8$ estados) e $g(D) = \langle 11001|11011 \rangle$ (treliça com $2^4 = 16$ estados).

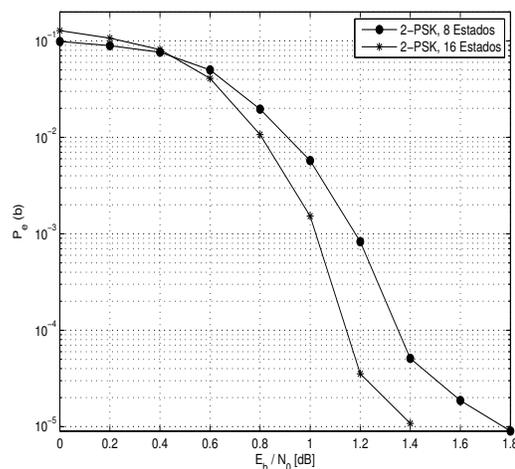


Fig. 3. Desempenho dos códigos RSC binários.

A curva de desempenho, em termos de probabilidade de erro de bit média versus E_b/N_0 , referente ao codificador cuja treliça possui 16 estados, apresenta melhor desempenho do que

a referente ao codificador de 8 estados, para $E_b/N_0 > 0.5$ dB. Por outro lado, o tempo gasto pela decodificação no esquema referente a 16 estados é de 69.35 segundos e o tempo do esquema de 8 estados é de 37.56 segundos.

A Tabela I mostra a matriz geradora dos codificadores binários que possuem maiores distâncias D_{free} e D_{efet} [10], [8]. Observe que a distância euclidiana mínima do código turbo, D_{free} , é formada pelo peso euclidiano da seqüência de paridade do codificador *RSC-1* mais o peso euclidiano da seqüência de paridade do codificador *RSC-2* mais o peso euclidiano W_{free} , da seqüências de informação associada a D_{free} , enquanto que, a distância euclidiana efetiva D_{efet} , é a menor distância euclidiana associada ao peso efetivo W_{efet} , onde W_{efet} é o peso da seqüências de informação não nula e de menor peso euclidiano, na qual a treliça do codificador *RSC* tem [0 0] como estado inicial e final. Note que $g(D) = \begin{bmatrix} 1 & g_2(D) \\ & g_1(D) \end{bmatrix} = \langle g_1(D)|g_2(D) \rangle = \langle a_n \dots a_0 | b_n \dots b_0 \rangle$, onde a_i e b_i são os coeficientes dos polinômios $g_1(D)$ e $g_2(D)$, respectivamente.

Estados	$g(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
4	$\langle 111 101 \rangle$	20	4	14	6
8	$\langle 1101 1111 \rangle$	28	4	16	8
16	$\langle 11001 11011 \rangle$	22	6	18	10

TABELA I: Codificadores binários.

B. Esquema Turbo Ternário com Modulação 3-PSK

Nesta seção analisamos o desempenho e a complexidade do esquema turbo ternário com relação ao esquema turbo binário.

A Figura 4 apresenta as curvas da taxa de erro de bit versus E_b/N_0 para o esquema turbo ternário que possui codificadores *RSC* com $g(D) = \langle 111|102 \rangle$ (treliça com 9 estados) e $g(D) = \langle 1112|2122 \rangle$ (treliça com 27 estados), comprimento da seqüência de informação, $N = 1261$ símbolos.

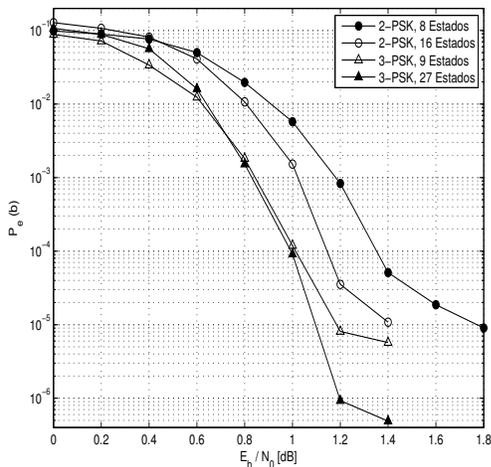


Fig. 4. Desempenho dos códigos turbo binários e ternários.

A curva de desempenho, em termos de probabilidade de erro de bit média versus E_b/N_0 , referente ao codificador cuja treliça possui 27 estados, apresenta melhor desempenho do que a curva do codificador de 9 estados, para $E_b/N_0 > 0.7$ dB.

Além disso, para $E_b/N_0 \geq 1.2$ dB a probabilidade de erro de bit do codificador de 27 estados é pelo menos 1 ordem de grandeza menor que a probabilidade do codificador de 9 estados. Por outro lado, o tempo gasto na decodificação do esquema de 27 estados, é 2.19 vezes maior do que o tempo no esquema de 9 estados.

Logo, o esquema ternário com 9 estados tem bom desempenho e baixa complexidade de decodificação. Já o esquema ternário com 27 estados tem maior complexidade de decodificação e melhor desempenho que o esquema referente a 9 estados. Note também o *erro floor* mais baixo no esquema com 27 estados.

A Figura 4 também mostra o desempenho dos esquemas binário e ternário juntos. O esquema ternário com 9 estados, tem ganhos¹ maiores do que 0.3 dB e 0.2 dB, para $p_e(b) < 10^{-2}$, quando comparado com esquemas binários com 8 e 16 estados, respectivamente. Além disso, o tempo gasto na decodificação do esquema ternário com 9 estados é 1.19 e 2.33 vezes menor que o tempo dos esquemas binários com 8 e 16 estados, respectivamente. Note que o sistema ternário processa $N = 1261$ símbolos em vez de $N = 2000$ bits do sistema binário.

Como o número de estados de cada codificador influenciam tanto no desempenho quanto na complexidade do sistema turbo, a comparação do sistema binário com o ternário não é muito justa, mas indica que os codificadores ternários proporcionam melhor desempenho.

Logo, o sistema ternário com modulação 3-PSK, apresenta melhor desempenho, menor complexidade de decodificação e ocupa menos largura de faixa que sistema binário com modulação 2-PSK. Ou seja, este sistema ternário transporta 1.58 vezes mais informação do que o sistema binário, tornando-se assim mais eficiente quando aplicado a sistemas de comunicação que possuam largura de faixa limitada. Todavia, este sistema não possui um perfeito mapeamento da fonte binária para os símbolos ternários.

A Tabela II mostra a matriz geradora dos codificadores ternários que possuem maiores distâncias D_{free} e D_{efet} .

Estados	$g(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
9	$\langle 111 102 \rangle$	20.78	3.46	12.12	5.19
9	$\langle 122 202 \rangle$	20.78	3.46	13.85	6.92
27	$\langle 1112 2122 \rangle$	24.24	3.46	15.58	5.19

TABELA II: Codificadores ternários.

C. Esquema Turbo Quaternário com Modulação 4-PSK

A Figura 5 apresenta as curvas da taxa de erro de bit versus E_b/N_0 para o esquema turbo quaternário que possui codificador *RSC* com $g(D) = \langle 113|212 \rangle$ (treliça com 16 estados), comprimento da seqüência de informação, $N = 1000$ símbolos.

O codificador quaternário apresenta desempenho equivalente ao desempenho do codificador binário com 16 estados e o esquema turbo quaternário processa $N = 1000$ símbolos em vez de $N = 2000$ bits como o sistema binário.

¹Denotamos como ganho o valor absoluto da diferença entre as duas razões $E_b/N_0(1)$ e $E_b/N_0(2)$, calculada para a mesma taxa de erro de bit.

Logo, o esquema quaternário com modulação 4-PSK gasta metade do tempo de decodificação em relação ao esquema binário com modulação 2-PSK . Ou seja este esquema quaternário processa o dobro da quantidade de informação do esquema binário com o mesmo tempo de decodificação.

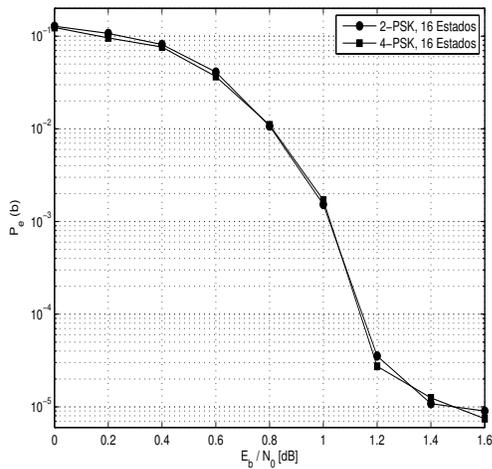


Fig. 5. Desempenho dos códigos turbo binários e quaternários.

A Tabela III mostra a matriz geradora dos codificadores quaternários que possuem maiores distâncias D_{free} e D_{efet} .

Estados	$g(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
16	$\langle 111 232 \rangle$	26.14	2.82	10.0	6
16	$\langle 113 212 \rangle$	19.31	2.82	10.0	6
16	$\langle 133 312 \rangle$	21.31	2.82	11.89	6.24

TABELA III: Codificadores quaternários.

D. Esquema Turbo 5-ário com Modulação 5-PSK

Nesta seção analisamos o desempenho, a complexidade, o mapeamento de fonte e a largura de banda do esquema 5-ário com relação ao esquema binário.

A Figura 6 apresenta a curva da taxa de erro de bit versus E_b/N_0 para o esquema turbo 5-ário que possui codificador RSC com $g(D) = \langle 111|443 \rangle$ (treliça com 25 estados), comprimento da seqüência de informação, $N = 861$ símbolos.

A curva referente ao codificador 5-ário encontra-se acima da curva do codificador binário com 16 estados. Note ainda que o esquema 5-ário, realiza o processamento com $N = 861$ símbolos, com 25 estados na treliça e com 5 transições associadas a cada estado, enquanto que o esquema binário, realiza o processamento com $N = 2000$ bits, com 16 estados na treliça e com 2 transições associadas a cada estado. O tempo gasto na decodificação do esquema 5-ário é 1.6 vezes menor, e a quantidade de informação transmitida é 2.3 vezes maior do que no esquema binário. Portanto, para um sistema que possua largura de faixa limitada, o esquema 5-ário é mais vantajoso que o esquema binário.

Codificadores M -ários que possuem 2 ou mais memórias apresentam melhor capacidade de codificação e melhor desempenho que o binário. Por outro lado, se o tamanho do alfabeto M -ário é maior do que 7, o sistema turbo torna-se impraticável

devido a complexidade de decodificação, para um codificador com 2 ou mais memórias. Por exemplo, um codificador 8-ário com 2 memórias possui $8^2 = 64$ estados e com 3 memórias possui $8^3 = 512$ estados.

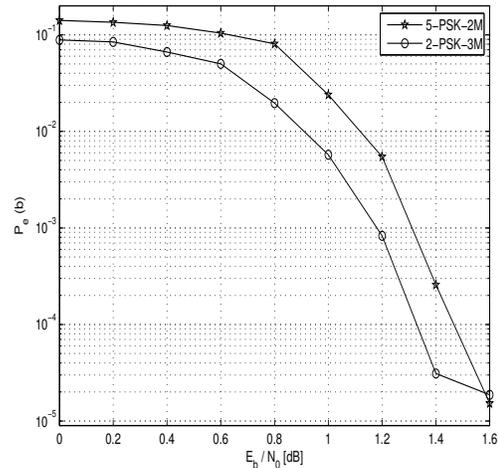


Fig. 6. Desempenho dos códigos turbo binários e 5-ários.

A Tabela IV mostra a matriz geradora dos codificadores 5-ários que possuem maiores distâncias D_{free} e D_{efet} .

Estados	$g(D)$	D_{efet}	W_{efet}	D_{free}	W_{free}
25	$\langle 111 213 \rangle$	21.54	3.07	12.31	6.16
25	$\langle 111 443 \rangle$	17.73	3.07	12.03	3.52
25	$\langle 144 242 \rangle$	17.56	3.35	13.20	3.82

TABELA IV: Codificadores 5-ários.

Verificaremos agora o desempenho do sistema turbo M -ário com relação a variação do número de iterações, do comprimento da seqüência de informação e do número de estados da treliça referente ao codificador RSC.

A Figura 7 mostra que as curvas referentes a 8ª, 7ª, 6ª, 5ª, 4ª, 3ª, 2ª e 1ª iteração alcançam a probabilidade de erro de bit de 10^{-5} nas taxas de 1.19, 1.20, 1.28, 1.38, 1.49, 1.79, 2.40 e 4.20 dB, com tempos de decodificação de 21.71, 19.38, 17.00, 14.70, 12.33, 9.95, 7.61 e 5.25 segundos, respectivamente.

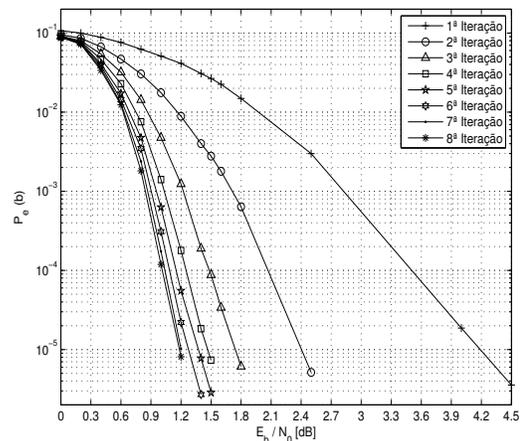


Fig. 7. Processo iterativo turbo ternário.

Note que existe um determinado valor para o número de iterações tal que, se aumentarmos este valor, aumentamos em consequência disto o tempo gasto na decodificação, mas não obtemos um ganho de codificação adicional significativo para o esquema turbo.

A Figura 8 mostra que as curvas na 8ª iteração referentes aos comprimentos das seqüências de informação de $N = 8000$, $N = 4000$, $N = 2000$, $N = 1000$ e $N = 500$ bits, alcançam a probabilidade de erro de bit de 10^{-5} nas taxas de 0.80, 0.99, 1.19, 1.54 e 2.00 dB, com tempos de decodificação de 141.70, 50.93, 21.91, 10.02 e 4.89 segundos, respectivamente.

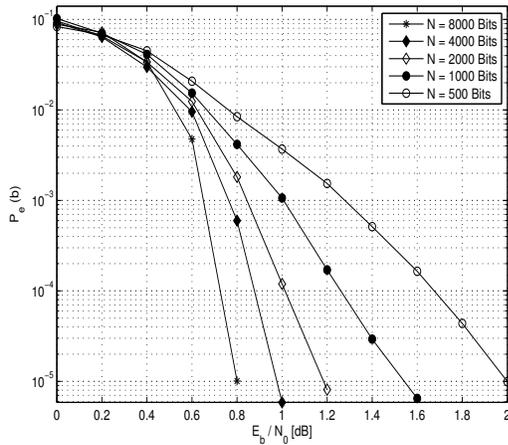


Fig. 8. Variação do comprimento da seqüência de informação do sistema ternário.

Note que existe um determinado valor para o comprimento da seqüência de informação que, a partir deste valor, se aumentarmos o comprimento da seqüência de informação, aumentamos em consequência disto o tempo gasto na decodificação, mas o ganho de codificação obtido com esse aumento é muito pequeno.

A Figura 9 mostra que as curvas na 8ª iteração do esquema binário referentes a 4, 8 e 16 estados, alcançam a probabilidade de erro de bit de 2×10^{-5} nas taxas de 1.3, 1.6 e 1.9 dB, com tempos de decodificação de 69.35, 37.56 e 21.00 segundos, respectivamente, e que as curvas da 8ª iteração do esquema

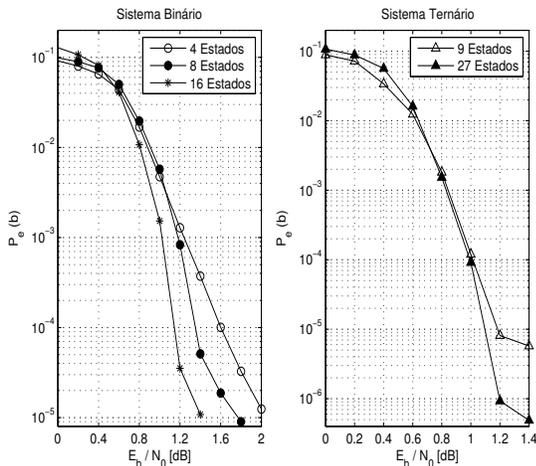


Fig. 9. Variação do número de estados.

ternário referentes a 9 e 27 estados, alcançam a probabilidade de erro de bit de 6×10^{-6} nas taxas de 1.4 e 1.11 dB, com tempos de decodificação de 23.21 e 55.02 segundos, respectivamente. Note que existe um determinado valor para o número de estados da treliça do codificador RSC, que, a partir deste valor, o tempo gasto na decodificação é grande tornando assim o algoritmo de decodificação iterativo impraticável.

VI. CONCLUSÃO

Nesta seção analisamos as curvas da taxa de erro de bit versus E_b/N_0 referente as simulações realizadas para os códigos turbo multiníveis definidos sobre os anéis e campos de inteiros módulo- M .

No sistema M -ário definido sobre o anel de inteiros \mathbb{Z}_M , existe um número reduzido de polinômios para a matriz geradora do codificador RSC, pois nem todo elemento do anel \mathbb{Z}_M possui inverso multiplicativo. Por outro lado, alguns anéis possuem um perfeito mapeamento de fonte quando comparado com o sistema binário, como os anéis \mathbb{Z}_4 , \mathbb{Z}_8 , \mathbb{Z}_{16} e etc.

No sistema M -ário definido sobre o campo \mathbb{Z}_M , existe um número bem maior de polinômios para a matriz geradora do codificador RSC, mas não existe um perfeito mapeamento da fonte binária nos símbolos de \mathbb{Z}_M .

As curvas simuladas mostram que os códigos turbo multiníveis apresentam bom desempenho com baixa relação sinal ruído e, em geral, quanto maior o tamanho do alfabeto maior será a eficiência do sistema turbo M -ário, pois 1 símbolo transmitido corresponde a $\log_2 M$ bits transmitidos no sistema M -ário. Além disso, para cada sistema turbo M -ário existe uma faixa de valores para os parâmetros (tamanho do alfabeto, comprimento da seqüência de informação, número de iterações, taxa de codificação e codificador RSC) para os quais o sistema turbo apresenta melhor desempenho.

Os resultados simulados mostram que o esquema turbo ternário, que possuem codificadores componentes com 9 estados, apresentam melhor desempenho com menor complexidade de decodificação que os demais esquemas turbo M -ários analisados.

REFERÊNCIAS

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," Proc. Int. Conf. Comm., pp. 1064-1070, May 1993.
- [2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," IEEE Trans. Inform. Theory, vol. 42, pp. 429-445, Mar. 1996.
- [3] G. Ungerboeck, "Channel coding with multilevel/phase signals," IEEE Trans. Inform. Theory, pp. 55-67, Jan. 1982.
- [4] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," Proc. Globecom. pp. 1298-1303, 1994.
- [5] W. E. Ryan, "A Turbo Code Tutorial," Proc. IEEE Globecom'98, 1998.
- [6] M. F. Atiyah and I. G. MacDonald, "Commutative Algebra," Reading, MA: Addison - Wesley, 1969.
- [7] Rolf Johannesson, "Some Structural Properties of Convolutional Codes over Rings," IEEE Trans. Inform. Theory, pp. 839-845, Mar. 1998.
- [8] Chris Heegard and Stephen B. Wicker, "Turbo Coding," Kluwer Academic Publishers pp. 10-63.
- [9] F. Fagnini and S. Zampieri, "System-Theoretic Properties of Convolutional Codes Over Rings," IEEE Trans. Inform. Theory, vol. 47, N.º. 6, Sept. 2001, pp. 2256-2274.
- [10] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," IEEE Trans. Comm., vol. 44, N.º. 5, May. 1996, pp. 591-600.