

Decodificação Iterativa (Turbo) de Códigos Produto em Canais Não-Gaussianos

Daniel C. da Cunha e Jaime Portugheis

Resumo—O trabalho considera a aplicação de um algoritmo de decodificação iterativa (turbo) de códigos produto em sistemas de transmissão que utilizam dois tipos de canais não-gaussianos: o canal CDMA de salto rápido em frequência e o canal com ruído impulsivo. Para ambos os casos, propusemos uma maneira de calcular a confiabilidade das decisões tomadas após cada iteração que leva em conta as densidades de probabilidade dos canais considerados. Os resultados mostraram que os desempenhos dos sistemas melhoram com o aumento do número de iterações de decodificação, o que comprova a validade da proposta. Verificamos que, dependendo da probabilidade de ocorrência e da intensidade do ruído impulsivo, o conhecimento da relação sinal-ruído do canal pelo decodificador não proporciona ganhos adicionais de desempenho.

Palavras-Chave—Códigos produto, Decodificação iterativa (Turbo), Confiabilidade da decisão, Algoritmos SISO.

I. INTRODUÇÃO

Em [1] foi proposto um algoritmo iterativo para decodificação de códigos produto. Como o algoritmo se assemelha ao algoritmo de decodificação turbo para concatenação de códigos componentes convolucionais e um código produto pode ser considerado como concatenação de dois ou mais códigos de bloco, o termo código turbo de bloco foi utilizado. O algoritmo iterativo utiliza para a decodificação dos códigos componentes uma modificação do algoritmo de decisão suave proposto em [2] conhecido como algoritmo de Chase. A modificação introduzida é a definição de uma confiabilidade para a decisão do algoritmo de Chase, da qual é derivada a informação extrínseca necessária para a realização do algoritmo iterativo. A derivação desta confiabilidade foi feita apenas para o canal gaussiano. Com a inclusão desta confiabilidade como uma saída suave, o algoritmo modificado pode ser classificado como um algoritmo SISO (do inglês, *Soft Input Soft Output*). A aplicação do algoritmo iterativo mencionado para códigos turbo de bloco demonstrou proporcionar excelente compromisso entre desempenho e complexidade para diversos códigos, levantando a questão de como poderia ser usado em canais não-gaussianos, isto é, como gerar a confiabilidade da decisão do algoritmo SISO empregado para outros tipos de ruído.

Este artigo apresenta uma nova proposta para o cálculo aproximado da confiabilidade fornecida pelo algoritmo SISO. A aproximação heurística sugerida foi derivada especificamente para dois tipos de canais não-gaussianos: o canal FFH-CDMA

(do inglês, *Fast Frequency Hopping-Code Division Multiple Access*) e o canal com ruído impulsivo. Em ambos os casos, resultados de simulação mostraram que a aproximação sugerida é apropriada para a utilização no algoritmo iterativo turbo. A aproximação sugerida utiliza também um fator médio β para a confiabilidade das decisões do algoritmo SISO. Para este fator e para o fator α inerente ao algoritmo proposto em [1], foram considerados diferentes modelos de geração dos mesmos no intuito de otimizar o desempenho atingido. Entretanto, para alguns valores dos parâmetros do ruído impulsivo, verificamos que uma escolha antecipada dos fatores α e β do algoritmo iterativo que garanta a convergência do mesmo, pode se tornar uma tarefa bastante árdua.

O artigo está organizado nas seguintes seções: na seção II é descrito o algoritmo de decodificação turbo proposto em [1]; na seção III é apresentada a nova proposta para o cálculo da confiabilidade das decisões do algoritmo SISO utilizado no algoritmo turbo mencionado na seção anterior; na seção IV são apresentados os resultados obtidos através da aplicação da nova proposta em canais não-gaussianos. Por fim, a seção V é dedicada a conclusões e propostas de trabalhos futuros.

II. DECODIFICAÇÃO TURBO DE CÓDIGOS PRODUTO

O algoritmo de decodificação turbo inicialmente utilizado em [1], faz uso de uma confiabilidade para cada decisão $\mathbf{D} = (d_1, d_2, \dots, d_j, \dots, d_N)$ do algoritmo de Chase, que é dada por

$$\Lambda(d_j) = \ln \left(\frac{p\{e_j = +1|\mathbf{R}\}}{p\{e_j = -1|\mathbf{R}\}} \right) \quad , \quad (1)$$

onde

$$p\{e_j = +1|\mathbf{R}\} = \sum_{\mathbf{C}^i \in S_j^{+1}} p\{\mathbf{E} = \mathbf{C}^i|\mathbf{R}\} \quad e$$

$$p\{e_j = -1|\mathbf{R}\} = \sum_{\mathbf{C}^i \in S_j^{-1}} p\{\mathbf{E} = \mathbf{C}^i|\mathbf{R}\} \quad ,$$

com $p\{\}$ representando probabilidade, \mathbf{E} , a palavra transmitida e \mathbf{R} , a palavra recebida. S_j^{+1} e S_j^{-1} são subconjuntos de palavras-código, $\mathbf{C}^i = (c_1^i, c_2^i, \dots, c_j^i, \dots, c_N^i)$, tal que $c_j^i = +1$ e -1 , respectivamente.

Aplicando-se a regra de Bayes, assumindo palavras-código igualmente prováveis e que o canal é o AWGN (do inglês, *Additive White Gaussian Noise*), podemos manipular (1) de acordo com [1], para obter a seguinte aproximação:

$$\Lambda'(d_j) = \frac{1}{4} \left(\left| \mathbf{R} - \mathbf{C}^{-1(j)} \right|^2 - \left| \mathbf{R} - \mathbf{C}^{+1(j)} \right|^2 \right) \quad , \quad (2)$$

onde $\mathbf{C}^{+1(j)}$ e $\mathbf{C}^{-1(j)}$ são palavras-código em S_j^{+1} e S_j^{-1} , respectivamente, ambas a uma distância mínima de \mathbf{R} . $\Lambda'(d_j)$ tem o mesmo sinal de d_j e seu valor absoluto representa a confiabilidade da decisão.

Para calcularmos a confiabilidade da decisão d_j , precisamos identificar duas palavras-código $\mathbf{C}^{+1(j)}$ e $\mathbf{C}^{-1(j)}$, sendo que uma delas é \mathbf{D} , e a outra, considerada concorrente de \mathbf{D} é definida como \mathbf{C} , a palavra-código mais próxima de \mathbf{R} com $c_j \neq d_j$. Dadas as palavras \mathbf{C} e \mathbf{D} , podemos reescrever (2) da seguinte forma:

$$\Lambda'(d_j) = \frac{1}{4} \left(|\mathbf{R} - \mathbf{C}|^2 - |\mathbf{R} - \mathbf{D}|^2 \right) d_j \quad (3)$$

A eq.(3) parte da idéia de que a palavra-código concorrente \mathbf{C} , que sempre difere de \mathbf{D} pelo bit na posição j ($c_j = -d_j$), possa ser encontrada. A probabilidade de se encontrar as palavras \mathbf{C} e \mathbf{D} aumenta com o número de bits menos confiáveis da palavra recebida, que depende de qual algoritmo de Chase é utilizado. A complexidade do decodificador pode aumentar exponencialmente com este número de bits, e portanto fazer com que procuremos um compromisso entre desempenho e complexidade.

Diante desta necessidade, \mathbf{C} só é pesquisada em um subconjunto de palavras-código geradas pelo algoritmo de Chase e não no código todo. Desta maneira, em alguns casos, a palavra \mathbf{C} pode não ser encontrada para a decisão d_j , isto é, todas as palavras propostas tem o mesmo sinal na posição j . Neste caso, uma solução foi proposta por Pyndiah para computar a saída do algoritmo, dada pela seguinte equação:

$$\Lambda'(d_j) = \beta d_j \quad , \quad \beta \geq 0 \quad (4)$$

Esta aproximação foi usada para canais AWGN e se justifica por várias razões. Primeiro, o sinal de $\Lambda'(d_j)$ é igual a d_j . Segundo, se \mathbf{C} não for encontrada no subconjunto gerado pelo algoritmo de Chase, isto significa que \mathbf{C} está bem longe de \mathbf{R} em termos de distância Euclidiana. Conseqüentemente, não só a probabilidade da decisão d_j estar correta será alta, como também a confiabilidade de d_j será alta. A grandeza β , chamada fator de confiabilidade, representa o valor médio da confiabilidade das decisões d_j .

Vamos agora explicar a decodificação iterativa do código produto P , isto é, das linhas e colunas que o compõem. Considere a Fig. 1. A matriz $\Lambda(\mathbf{R})$ representa o sinal recebido

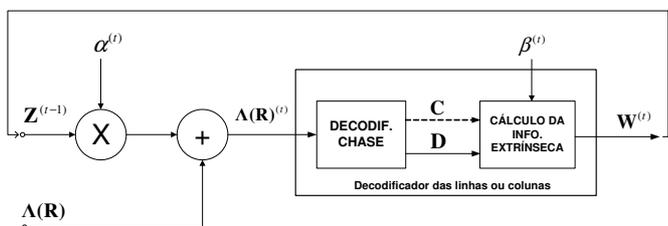


Fig. 1. Diagrama esquemático do decodificador turbo de bloco.

na entrada do decodificador turbo, sendo composta por valores de confiabilidade sobre os bits recebidos e cada elemento $\Lambda(r_j)$ é calculado da seguinte forma:

$$\Lambda(r_j) = \ln \left(\frac{p(r_j|b_j = 0)}{p(r_j|b_j = 1)} \right) \quad , \quad (5)$$

com $j \in [0, N^2 - 1]$. O primeiro decodificador componente atua sobre as linhas (ou colunas) de P usando $\Lambda(\mathbf{R})$ como entrada. A decodificação é então executada, fornecendo a saída suave desejada.

A entrada do t -ésimo decodificador componente é dada por

$$\Lambda(\mathbf{R})^{(t)} = \Lambda(\mathbf{R}) + \alpha^{(t)} \mathbf{Z}^{(t-1)} \quad , \quad (6)$$

onde t é um índice que representa o passo de decodificação. Como cada iteração compreende a atuação dos dois decodificadores componentes (linha e coluna), o total de passos será sempre o dobro do total de iterações. As matrizes $\mathbf{Z}^{(t-1)}$ e $\mathbf{W}^{(t)}$ representam a informação extrínseca calculada pelo decodificador a cada passo. A diferença entre elas, apesar de numericamente iguais, é que a primeira é a extrínseca calculada no passo anterior e a segunda é a saída do decodificador no passo atual. Sendo assim, a informação extrínseca obedece a uma estrutura recursiva e depende da confiabilidade gerada por cada decodificador componente.

De acordo com o método heurístico discutido por Colavolpe, Ferrari e Raheli [3], a confiabilidade da decisão é descrita da seguinte forma :

$$L(a_j) = F z_j + w_j \quad , \quad (7)$$

onde F é um fator multiplicativo. Trazendo (7) para o nosso contexto, temos que :

$$\Lambda(d_j) = \alpha z_j + w_j \quad , \quad (8)$$

ou matricialmente,

$$\Lambda(d_j)^{(t)} = \alpha^{(t)} \mathbf{Z}^{(t-1)} + \mathbf{W}^{(t)} \quad (9)$$

A grandeza α , chamada fator de escala, é usada para reduzir o efeito da informação extrínseca no decodificador nos primeiros passos de decodificação.

Substituindo (6) em (9) e considerando que o valor constante de $\Lambda(\mathbf{R})$ não altera as decisões, temos que :

$$\mathbf{W}^{(t)} = \Lambda(d_j)^{(t)} - \Lambda(\mathbf{R})^{(t)} \quad (10)$$

A eq. (10) diz que a informação extrínseca gerada por cada decodificador componente é dada pela diferença entre a confiabilidade da decisão fornecida pelo algoritmo de Chase e a confiabilidade da nova entrada deste decodificador.

III. PROPOSTA HEURÍSTICA PARA GERAR A CONFIABILIDADE

De maneira similar a (4), iremos definir a confiabilidade da decisão utilizando um fator de confiabilidade β da seguinte forma:

$$\Lambda'(d_j) = \beta \Phi(d_j) \quad , \quad (11)$$

onde $\Phi(d_j)$ é definida como o logaritmo da razão de verossimilhança (do inglês, *Log-Likelihood Ratio*) da decisão d_j .

Vamos agora fazer alguns comentários sobre a proposta e os canais aqui abordados. Para o sistema que utiliza canal AWGN e modulação BPSK (do inglês, *Binary Phase Shift Keying*), podemos definir $\Phi(d_j)$ da seguinte forma:

$$\Phi(d_j) = \ln \left(\frac{p(d_j|e_j = +\sqrt{E_b})}{p(d_j|e_j = -\sqrt{E_b})} \right) \quad , \quad (12)$$

onde E_b é a energia média por bit de informação. A eq. (12) nos diz que $\Phi(d_j)$ tem o mesmo sinal de d_j e conseqüentemente $\Lambda'(d_j)$ também. Da mesma forma que em (4), o valor absoluto de $\Lambda'(d_j)$ indica quão confiável é a decisão d_j . É possível mostrar que (11) pode ser reduzida a (4), a menos do fator multiplicativo $2\sqrt{E_b}/\sigma^2$ quando aplicada a um canal AWGN de variância σ^2 .

Em relação ao canal com ruído impulsivo, podemos abordá-lo da mesma forma que o canal AWGN visto que a modulação empregada também é BPSK. Entretanto, a facilidade de simplificação de (12) não ocorre pelo fato da função densidade de probabilidade do canal com ruído impulsivo ser mais complexa. Para o canal FFH-CDMA, também podemos utilizar a proposta definida, porém com algumas particularidades do sistema, conforme veremos na seção IV.

Esta proposta também inclui uma nova maneira de se obter os fatores α e β anteriormente mencionados. Em [1], os parâmetros α e β foram escolhidos de maneira pré-determinada nos intervalos $[0,0;1,0]$ e $[0,2;1,0]$, respectivamente. Em [4], foi proposta uma adaptação a cada passo da iteração para α e uma regra de progressão linear para β . Neste trabalho, mantivemos uma progressão linear para β e utilizamos uma regra segundo uma progressão logarítmica para α , como foi sugerido em [5]. Esta nova escolha dos fatores α e β demonstrou em geral levar a um desempenho superior quando comparada a uma escolha pré-determinada dos mesmos.

IV. APLICAÇÕES

A. Canal FFH-CDMA

Consideraremos um sistema FFH-CDMA codificado, onde a comunicação de cada grupo de usuários com a estação rádio base (ERB) é realizada a uma taxa de R_b bits/s, através do compartilhamento de um mesmo canal de largura de banda igual a W Hz. Os componentes deste sistema são expostos em [6].

O canal FFH-CDMA pode ser modelado por ML funções densidade de probabilidade, $p(R_{nl}|m)$, $0 \leq n, m \leq M-1$, $0 \leq l \leq L-1$, para cada símbolo M -ário de entrada do modulador FFH [7]. Vale ressaltar que M é o número de frequências da modulação MFSK (do inglês, *M-ary Frequency Shift Keying*) e L , o número de chips utilizados nos saltos em frequência. Cada função $p(R_{nl}|m)$ é dada por :

$$p(R_{nl}|m) = \sum_{k=0}^J \frac{B(k; J, \mu)}{k + d + \delta_{nm}} \exp\left(-\frac{R_{nl}}{k + d + \delta_{nm}}\right), \quad (13)$$

onde $\mu = 1/M$, δ_{nm} é o delta de Kronecker, J é o número de usuários interferentes, $B(k; J, \mu) = \binom{J}{k} \mu^k (1 - \mu)^{J-k}$ e d é definido como o inverso da relação sinal-ruído por chip, dado por:

$$d = \frac{N_o}{E_c} = \frac{L}{KR_p} \left(\frac{E_b}{N_o}\right)^{-1}, \quad (14)$$

onde R_p é a taxa do código produto, $K = \log_2 M$ é o número de bits por símbolo da modulação MFSK e N_o , a densidade unilateral do ruído.

Para este tipo de canal, adaptamos o algoritmo de decodificação turbo, iniciando pela matriz $\Lambda(\mathbb{R})$, onde \mathbb{R} representa o sinal na saída do canal. Após o cálculo de $p(R_{nl}|m)$, obtemos a função densidade conjunta $p(\mathcal{R}|m)$, onde \mathcal{R} representa cada uma das matrizes FFH-CDMA que compõem a matriz \mathbb{R} [6]:

$$p(\mathcal{R}|m) = \prod_{n=0}^{M-1} \prod_{l=0}^{L-1} p(R_{nl}|m) \quad . \quad (15)$$

O nosso objetivo é extrair um valor que possa ser utilizado como medida de informação de canal nos algoritmos SISO a serem utilizados. Assim sendo, vamos definir esta medida, ρ_u , como:

$$\rho_u = A \ln \left(\frac{p(\mathcal{R}|b_u = 0)}{p(\mathcal{R}|b_u = 1)} \right) \quad , \quad (16)$$

onde $p(\mathcal{R}|b_u = 0)(p(\mathcal{R}|b_u = 1))$ é a função densidade de probabilidade de transição de \mathcal{R} dado que o bit transmitido b_u assume o valor 0(1) e A é uma constante arbitrária.

A partir de agora vamos definir $p(\mathcal{R}|b_u)$ em função de $p(\mathcal{R}|m)$. Seja $p(m, \mathcal{R})$ a função densidade conjunta tal que:

$$p(m, \mathcal{R}) = p(b_0, b_1, \dots, b_u, \dots, b_{K-1}, \mathcal{R}) \quad (17)$$

onde $\{b_0, b_1, \dots, b_u, \dots, b_{K-1}\}$ é a seqüência de K bits que compõem o símbolo m . Podemos dizer que:

$$p(b_0 = 0, \mathcal{R}) = \sum_{b'} p(b_0 = 0, b', \mathcal{R}) \quad , \quad b' = b_1, \dots, b_u, \dots, b_{K-1}$$

$$p(b_0 = 0, \mathcal{R}) = \sum_{b'} p(\mathcal{R}|b_0 = 0, b') p(m) \quad ,$$

onde o somatório considera os símbolos m para os quais $b_0 = 0$. Logo,

$$p(\mathcal{R}|b_0 = 0) = \frac{p(b_0 = 0, \mathcal{R})}{p(b_0 = 0)} = \frac{\sum_{b'} p(\mathcal{R}|b_0 = 0, b') p(m)}{p(b_0 = 0)} \quad .$$

Então,

$$p(\mathcal{R}|b_0 = 0) = \frac{\sum_{b'} p(\mathcal{R}|m) \frac{1}{M}}{\frac{1}{2}} = \sum_{b'} p(\mathcal{R}|m) \frac{2}{M} \quad . \quad (18)$$

Sendo assim, a função densidade de probabilidade $p(\mathcal{R}|b_u = 0)$ fica definida como :

$$p(\mathcal{R}|b_u = 0) \triangleq \sum_{r=0}^{\left(\frac{M}{2}\right)-1} p(\mathcal{R}|m) \frac{2}{M} \quad , \quad (19)$$

onde $m = r + \lfloor \frac{r}{2^u} \rfloor 2^u$ e $u \in [0, K-1]$. Conseqüentemente, a função densidade $p(\mathcal{R}|b_u = 1)$ é obtida da seguinte forma :

$$p(\mathcal{R}|b_u = 1) \triangleq \sum_{r=0}^{\left(\frac{M}{2}\right)-1} p(\mathcal{R}|m) \frac{2}{M} \quad , \quad (20)$$

onde $m = (M-1) - (r + \lfloor \frac{r}{2^u} \rfloor 2^u)$ e $u \in [0, K-1]$.

Vamos agora explicar como é feito o mapeamento dos bits transmitidos neste sistema para melhor compreender como são calculados os valores de entrada do decodificador turbo. Considere que a matriz do código produto foi construída

a partir de códigos componentes de comprimento N . Cada conjunto de K bits das colunas que compõem esta matriz é mapeado em um símbolo da modulação M -ária usada, gerando uma matriz de dimensões $\frac{N}{K} \times N$. Em seguida, cada símbolo é convertido em uma matriz de dimensões $M \times L$. A saída do canal será então formada pela matriz $\mathbb{R} = [\mathcal{R}^{vi}]$, tal que $v \in [0, \frac{N}{K} - 1]$ e $i \in [0, N - 1]$.

Assim podemos calcular a matriz $\Lambda(\mathbb{R})$ de entrada do decodificador turbo, no qual cada elemento, Λ_{ji} , assumirá o valor $\Phi(\mathcal{R})$, ou seja,

$$\Lambda_{ji} = \ln \left(\frac{p(\mathcal{R}|b_{ji} = 0)}{p(\mathcal{R}|b_{ji} = 1)} \right) \quad , \quad (21)$$

onde $j = vK + u$, com $v \in [0, \frac{N}{K} - 1]$, $u \in [0, K - 1]$ e $i \in [0, N - 1]$.

Durante o processo de decodificação das linhas e colunas que compõem o código produto, o cálculo descrito até (21) também é executado, entretanto no lugar de \mathcal{R} usaremos a matriz \mathcal{D} , que difere de \mathcal{R} por não considerar o ruído introduzido pelo canal. O restante do algoritmo procede da mesma maneira, isto é, após a decisão tomada pelo algoritmo de Chase, calculamos a confiabilidade da seguinte forma:

$$\Lambda(d_{ji}) = \beta \Phi(\mathcal{D}) \quad , \quad (22)$$

onde

$$\Phi(\mathcal{D}) = \ln \left(\frac{p(\mathcal{D}|b_{ji} = 0)}{p(\mathcal{D}|b_{ji} = 1)} \right) \quad . \quad (23)$$

Note que b_{ji} se refere aos bits que compõem toda a matriz produto, enquanto b_u faz referência apenas aos bits que compõem um determinado símbolo da modulação M -ária.

A informação extrínseca e os novos valores de entrada para o próximo decodificador componente seguem as equações (6) e (10) da seção II.

As Figs. 2 e 3 ilustram o desempenho de sistemas FFH-CDMA que utilizam um código produto $P_1 = (400, 196, 16)$ composto por dois códigos $C_1 = (20, 14, 4)$ e um código produto $P_2 = (225, 196, 4)$ composto por dois códigos $C_2 = (15, 14, 2)$. Vale salientar que os códigos de paridade, C_2 , são decodificados através do algoritmo de Wagner por não possuírem capacidade de correção.

Nos resultados apresentados utilizamos os fatores pré-determinados $\alpha_1 = [0, 0; 0, 2; 0, 3; 0, 5; 0, 7; 0, 9]$ e $\beta_1 = [0, 2; 0, 4; 0, 6; 0, 8; 1, 0; 1, 0]$, onde cada valor é usado nos dois passos de uma iteração. Podemos observar nas figuras que a cada iteração ocorre uma melhora de desempenho do sistema. Entretanto, a partir da segunda iteração não há melhora significativa de desempenho para ambos os casos. Não atribuímos este comportamento à proposta de confiabilidade sugerida neste trabalho, mas sim ao fato de que um mapeamento inadequado do código produto sobre a modulação M -ária do sistema FFH-CDMA foi utilizado. A importância de um mapeamento adequado foi ressaltada em [8]. A inadequação do mapeamento fica ainda mais comprovada se compararmos a distância do desempenho obtido com a taxa de corte dos sistemas, como mostram as figuras. De qualquer maneira, a melhora de desempenho com o aumento do número de iterações indica a viabilidade da proposta feita neste trabalho.

As Figs. 2 e 3 também comparam os sistemas utilizando α e β pré-determinados com a nova proposta (α logarítmico, β linear). Verifica-se que a nova regra para α e β possui desempenho no mínimo tão bom quanto o atingido pela regra de valores pré-determinados.

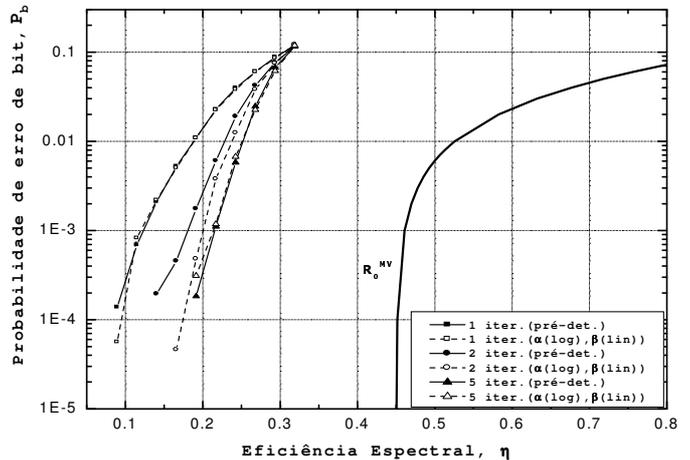


Fig. 2. Curvas de desempenho do sistema FFH-CDMA com parâmetros $M = 32, L = 6, E_b/N_0 = 25$ dB, utilizando o código P_1 de taxa $r = 0,49$. R_0^{MV} representa a taxa de corte do sistema.

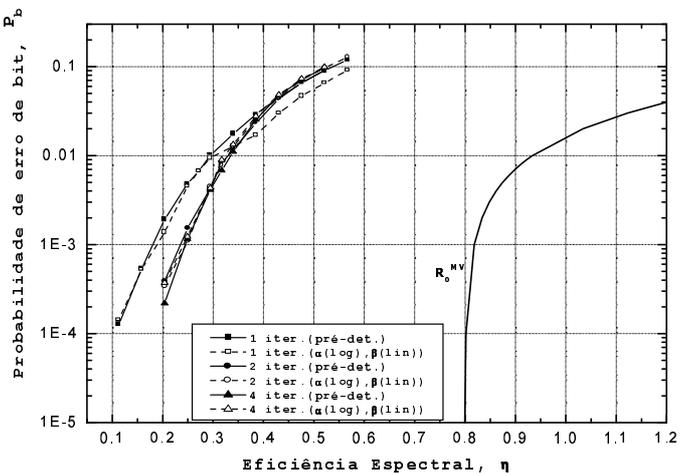


Fig. 3. Curvas de desempenho do sistema FFH-CDMA com parâmetros $M = 32, L = 6, E_b/N_0 = 25$ dB, utilizando o código P_2 de taxa $r = 0,87$. R_0^{MV} representa a taxa de corte do sistema.

B. Canal com ruído impulsivo

Vamos considerar a modelagem matemática deste tipo de canal de acordo com [9]. O sinal recebido r_j é descrito como:

$$r_j = e_j + w_j + i_j \quad , \quad (24)$$

onde e_j é o símbolo transmitido a partir de uma modulação BPSK, w_j é o ruído AWGN com média zero e variância σ_w^2 e i_j é o ruído impulsivo. Considere que o ruído impulsivo seja o produto entre um processo gaussiano e um processo de Bernoulli da seguinte forma:

$$i_j = B_j g_j \quad , \quad (25)$$

onde B_j é o processo de Bernoulli, ou seja, uma seqüência i.i.d. de zeros e uns, com $Pr(B_j = 1) = p_b$ e g_j é o ruído AWGN com média zero e variância $\sigma_i^2 = \gamma \sigma_w^2$. Denomina-se p_b a probabilidade de ocorrência do ruído impulsivo e γ a intensidade do mesmo. Sendo assim, o ruído visto pelo receptor vale:

$$n_j = w_j + B_j g_j \quad . \quad (26)$$

A probabilidade de transição do canal é dada por

$$p(r_j|e_j) = (1 - p_b) \frac{1}{\sqrt{2\pi}\sigma_w} e^{-\frac{(r_j - e_j)^2}{2\sigma_w^2}} + p_b \frac{1}{\sqrt{2\pi}\sqrt{\sigma_w^2 + \sigma_i^2}} e^{-\frac{(r_j - e_j)^2}{2(\sigma_w^2 + \sigma_i^2)}} \quad . \quad (27)$$

Uma vez conceituado o canal com ruído impulsivo, precisamos processar o sinal na saída do canal e adequá-lo à entrada do decodificador turbo, calculando a seguinte equação:

$$\Lambda(r_j) = \ln \left(\frac{p(r_j|e_j = +\sqrt{E_b})}{p(r_j|e_j = -\sqrt{E_b})} \right) \quad . \quad (28)$$

Durante o processo de decodificação, a confiabilidade da decisão $\Lambda'(d_j)$ é calculada de acordo com (11) e (12).

As Figs. 4, 5 e 6 ilustram o desempenho de um sistema que utiliza um código produto $P_3 = (576, 144, 64)$ composto por dois códigos $C_3 = (24, 12, 8)$. Na Fig. 4 consideramos um canal com $p_b = 0,01$ e $\gamma = 10$ e que o decodificador possui conhecimento da Relação Sinal-Ruído (RSR) do canal. Podemos observar que a cada iteração ocorre uma melhora significativa de desempenho do sistema. Isto novamente comprova a viabilidade da proposta aqui apresentada. Da figura podemos observar que o desempenho do sistema dista de $\sim 1,5$ dB da taxa de corte do canal para $P_b = 10^{-4}$. A figura também mostra a comparação dos sistemas utilizando α e β pré-determinados e segundo a nova proposta. Podemos verificar que a nova regra para α e β permite que seja obtido um maior ganho de codificação, notadamente nas iterações intermediárias (confronte as curvas da 2ª iteração). Para este canal foram também realizadas simulações onde o decodificador não possui conhecimento da RSR. Os resultados destas simulações mostraram que o conhecimento da RSR leva a um desempenho ligeiramente superior. Vale ressaltar que em ambos os casos, com e sem conhecimento da RSR, o algoritmo convergiu.

Na Fig. 5 consideramos um canal com $p_b = 0,33$ e $\gamma = 1000$, onde o decodificador desconhece a RSR do canal. Os fatores α e β utilizados neste caso seguem a nova proposta (α logarítmico, β linear). Podemos observar que a partir da 3ª iteração o algoritmo de decodificação diverge, fazendo com que o desempenho do sistema se situe muito longe da taxa de corte. Para este canal foram também realizadas simulações

onde o decodificador conhece a RSR, cujos resultados mostraram divergência semelhante a partir da 3ª iteração.

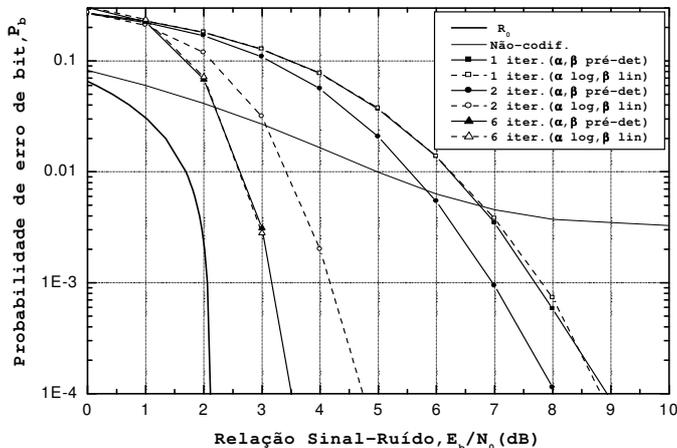


Fig. 4. Curva $P_b \times E_b/N_0$ para um sistema com ruído impulsivo $p_b = 0,01$ e $\gamma = 10$. R_0 representa a taxa de corte do sistema.

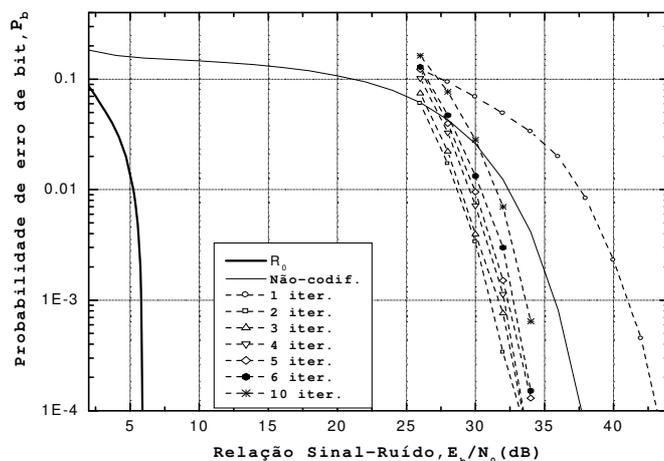


Fig. 5. Curva $P_b \times E_b/N_0$ para um sistema com ruído impulsivo $p_b = 0,33$ e $\gamma = 1000$. R_0 representa a taxa de corte do sistema.

Para finalizar o conjunto de canais com ruído impulsivo analisados, temos na Fig. 6 o desempenho do sistema que utiliza o canal com $p_b = 0,33$ e $\gamma = 10$, onde o decodificador desconhece a RSR do canal. Neste caso também são utilizados os fatores α logarítmico e β linear. Podemos observar que o desempenho do sistema satura por volta da 10ª iteração. Para $P_b = 10^{-3}$, este desempenho se situa bem próximo da taxa de corte (~ 1 dB). De maneira similar ao canal com $p_b = 0,33$ e $\gamma = 1000$, o conhecimento da RSR pelo decodificador mostrou divergência do algoritmo a partir da 3ª iteração. Para este caso, foram feitas tentativas de se obter fatores α e β que evitassem a divergência do algoritmo, porém não houve sucesso.

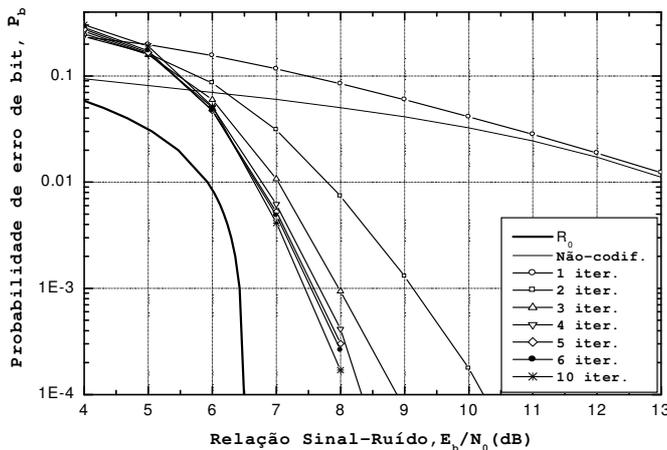


Fig. 6. Curva $P_b \times E_b/N_0$ para um sistema com ruído impulsivo $p_b = 0,33$ e $\gamma = 10$. R_0 representa a taxa de corte do sistema.

V. CONCLUSÕES

Este artigo apresentou um estudo sobre a aplicação de um algoritmo de decodificação turbo em sistemas de transmissão que utilizam canais não-gaussianos, mais precisamente o canal FFH-CDMA e o canal com ruído impulsivo. Nosso estudo se concentrou em propor uma maneira de se calcular a confiabilidade da decisão dada pelo algoritmo SISO componente e testar o desempenho do algoritmo de decodificação iterativa.

De uma maneira geral, os resultados obtidos aqui para os dois tipos de canais não-gaussianos comprovaram a eficiência da proposta. Para o canal com ruído impulsivo, o conhecimento da RSR do canal pelo decodificador nem sempre se mostrou vantajoso e, em alguns casos, trouxe agregada a divergência do algoritmo após as primeiras iterações.

A idéia da geração dos fatores α e β através de uma progressão logarítmica e linear, respectivamente, trouxe bons resultados em alguns casos. Também a variação desses fatores a cada passo de decodificação se mostrou mais eficiente que a variação por iteração. Entretanto, dependendo do canal, do código e da modulação utilizados, enfim, da aplicação empregada, esta forma de gerar α e β não nos garante a convergência do algoritmo. Isto indica que a busca por esses fatores, de maneira a otimizar o desempenho do sistema, pode eventualmente ser uma tarefa bastante árdua e tediosa.

Pelo conhecimento dos autores, quase não existem trabalhos que analisem a convergência e o desempenho do algoritmo iterativo proposto em [1] para canais não-gaussianos. Mesmo o método descrito em [10], no qual tanto α quanto β são calculados de maneira adaptativa, dependendo da estatística do canal e da informação extrínseca, apenas o canal AWGN é considerado. Sendo assim, a análise de convergência e a aplicação deste algoritmo em canais não-gaussianos é uma questão que permanece em aberto. Resultados adicionais de desempenho para os canais considerados neste trabalho são mostrados em [11].

AGRADECIMENTOS

Os autores gostariam de agradecer aos revisores pelos comentários e sugestões que proporcionaram uma melhor qualidade ao trabalho.

REFERÊNCIAS

- [1] R. Pyndiah, "Near Optimum Decoding of Product Codes: Block Turbo Codes," *IEEE Trans. Comm.*, vol. 46, n.8, pp. 1003-1010, Ago. 1998
- [2] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, n.1, pp. 170-182, Jan. 1972
- [3] G.Colavolpe, G.Ferrari e R.Raheli, "Extrinsic information in iterative decoding: a unified view," *IEEE Trans. Comm.*, vol. 49, n.12, pp. 2088-2094, Dez. 2001.
- [4] A. Picart e R. Pyndiah, "Adapted iterative decoding of product codes," *Global Telecommunications Conference - Globecom'99*, 1999.
- [5] D. A. Guimarães, *Tese de Doutorado.*, submetida para defesa em maio de 2003.
- [6] G.A.de Deus Jr. e J.Portugheis, "Sistemas FFH-CDMA codificados - Parte II: Critério de projeto," *CD do XIX Simp. Bras. de Telecomunicações*, Fortaleza, CE, Set. 2001.
- [7] O. Yue, "Maximum Likelihood Combining for Noncoherent and Differentially Coherent Frequency-Hopping Multiple Access Systems," *IEEE Trans. Inform. Theory*, vol. IT-28, n.4, pp. 631-639, Jul. 1982.
- [8] G.A.de Deus Jr. e J.Portugheis, "Coding for FFH-CDMA Systems," *Proceedings of the 2002 IEEE International Symposium on Information Theory*, Lausanne, Switzerland, p. 336, June 30- July 5, 2002.
- [9] M. Ghosha, "Analysis of the effect of impulse noise on multicarrier and single carrier QAM systems," *IEEE Trans. Comm.*, vol. 44, n.2, pp. 145-147, Fev. 1996.
- [10] P.A.Martin e D.P.Taylor, "On Adaptive Reduced-Complexity Iterative Decoding," *In IEEE GLOBECOM'00 Conf.*, pp. 772-776, San Francisco, CA, USA, Dez. 2000.
- [11] D. C. da Cunha, "Decodificação Iterativa (Turbo) de Códigos Produto em Canais Não-Gaussianos," *Dissertação de Mestrado*, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Campinas-SP, Brasil, 2003.