

Técnicas de Redução de Estados Aplicadas à Equalização Turbo

Miguel David Cosac e Jaime Portugheis

Resumo—O artigo considera um sistema de transmissão de informação codificada através de um canal que introduz interferência intersimbólica (IIS). No receptor, um algoritmo iterativo (equalização turbo) é utilizado para detecção da informação. É bem conhecido que a equalização turbo pode prover uma grande melhoria na taxa de erro de bit, aproximando o desempenho do sistema daquele sem IIS. Entretanto, existe um compromisso entre esta aproximação e a complexidade de implementação do receptor. O artigo analisa este compromisso através de três técnicas de redução de estados da treliça do canal: truncamento de coeficientes do canal, estimação dos coeficientes do canal e algoritmo M. Os resultados da análise mostraram que o algoritmo M possui o melhor compromisso entre desempenho e complexidade de implementação.

Palavras-Chave— Transmissão digital, Canal com Interferência Intersimbólica, Equalização Turbo, Técnicas de Redução de Estados.

Abstract—The article considers a coded data transmission system over an inter-symbol interference (ISI) channel. At the receiver, an iterative algorithm (turbo equalization) is used for information detection. It is well known that turbo equalization can yield great improvements in bit error rate, approaching the system performance to that of an ISI free channel. However, there exists a tradeoff between this approach and receiver implementation complexity. The article analyses this tradeoff by using three different reduced-state techniques on the channel trellis: truncation of channel coefficients, estimation of channel coefficients and the M algorithm. The analysis results indicate that the M algorithm has the best trade-off between performance and complexity.

Keywords—Digital transmission, Inter-symbol Interference Channels, Turbo Equalization, State-Reduction Techniques.

I. INTRODUÇÃO

Num sistema de transmissão digital codificado, a equalização é utilizada para compensar a distorção sofrida pelos sinais que carregam informação, causada pelo canal (filtros e/ou meio de transmissão). Recentemente foi comprovado em [1][2] ser possível eliminar-se completamente os efeitos de interferência intersimbólica (IIS), causados pela distorção de canal, através de um algoritmo iterativo no receptor com complexidade factível. Por analogia aos códigos turbo esse algoritmo denomina-se equalização turbo. Entretanto, existe um compromisso entre o grau de eliminação da IIS e a complexidade de implementação do receptor. A complexidade de implementação pode ser diminuída modificando-se os algoritmos de equalização e/ou os de decodificação do código de canal.

Miguel David Cosac e Jaime Portugheis, Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, Brasil, E-mails: migueldc@decom.fee.unicamp.br, jaime@decom.fee.unicamp.br.

Em [3], equalizadores lineares são utilizados para simplificar o processo iterativo de detecção. A equalização linear é considerada uma técnica determinística de redução de estados. Em [4], uma técnica probabilística baseada em seqüências de estados sobreviventes foi apresentada. Outras técnicas probabilísticas mais simplificadas foram sugeridas em [5] e [6]. Uma aplicação do algoritmo apresentado em [5] é mostrada em [7].

Neste artigo serão analisadas e comparadas três técnicas de redução de estados para o equalizador, duas delas determinísticas - truncamento e estimação dos coeficientes do canal - e uma probabilística - o algoritmo M.

O artigo está organizado como segue: na seção II é descrito um sistema de equalização turbo; na seção III, o algoritmo de decodificação é detalhado; a seção IV apresenta as técnicas de redução de estados propostas; a seção V analisa a complexidade das técnicas utilizadas. Por fim, a seção VI é dedicada às conclusões.

II. EQUALIZAÇÃO TURBO

A figura 1 mostra o diagrama de blocos de um sistema de transmissão digital utilizando equalização turbo. No diagrama, o canal é representado por seu modelo equivalente em banda básica discreto no tempo, com coeficientes h_k . A informação a ser transmitida é representada através dos símbolos binários igualmente prováveis, b_i , que assumem os valores 0 ou 1. Ela é processada pelo codificador, de taxa $R \leq 1$, que emite os símbolos codificados, c_n . Logo após, a seqüência de símbolos codificados é permutada pelo entrelaçador, I, gerando uma seqüência que é utilizada como entrada de um modulador BPSK. Os símbolos de saída do modulador, $y_n = \pm A$, são enviados através do canal com IIS e ruído aditivo gaussiano.

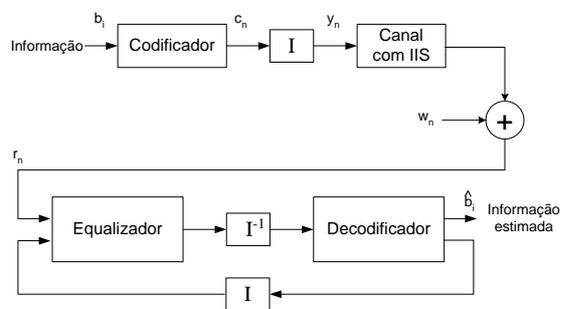


Fig. 1. Sistema de transmissão utilizando equalização turbo.

As saídas r_n do canal são dadas por

$$r_n = \sum_{k=0}^N h_k y_{n-k} + w_n \quad (1)$$

onde w_n são amostras de ruído gaussiano branco com média nula e variância σ^2 .

Os coeficientes da resposta ao impulso do canal e as amostras de ruído, w_n , são reais. A resposta ao impulso do canal é um sinal de energia que satisfaz

$$E_h = \sum_{k=0}^N h_k h_k^* = 1.$$

O modelo de canal utilizado foi extraído de [8]. É dado por

$$h_c[i] = 0,227\delta[i] + 0,46\delta[i-1] + 0,688\delta[i-2] + 0,46\delta[i-3] + 0,227\delta[i-4]. \quad (2)$$

Este canal tem sido exaustivamente estudado e outros resultados podem ser encontrados em [3] e [8]. Trata-se de um canal BI-AWGN (*Binary-Input Additive White Gaussian Noise*) e os resultados deste modelo não podem ser estendidos para um canal contínuo.

O valor de E_b/N_0 é dado por

$$\frac{E_b}{N_0} \triangleq \frac{E \left[|y_k|^2 \right] E_h}{RN_0} = \frac{A^2 E_h}{2\sigma^2 R}. \quad (3)$$

No receptor cada iteração é composta por uma passagem pelo equalizador e pelo decodificador. O algoritmo utilizado em ambos é uma generalização do algoritmo BCJR proposto em [9] e foi denominado de algoritmo SISO (Soft-Input Soft-Output) [10]. Ele atualiza as distribuições de probabilidade dos bits de informação e/ou símbolos codificados baseado na informação obtida do canal e/ou do equalizador.

Neste trabalho, para todas as simulações realizadas, utiliza-se um codificador convolucional não-recursivo e não-sistemático de taxa $R = 1/2$ cuja matriz geradora é dada por $G = [101; 111]$. Os blocos de informação têm comprimento igual a 16000 bits.

III. ALGORITMO DE DECODIFICAÇÃO SISO

No que segue, letra maiúscula indica variável aleatória (VA) e letra minúscula, a realização da VA. O texto refere-se à treliça do código, embora seja válido também para a treliça que representa a memória do canal.

O módulo SISO é um dispositivo com duas entradas e duas saídas conforme mostra a Figura 2. Ele recebe como entrada as seqüências de distribuição de probabilidade $P(u; I)$ e $P(c; I)$, gerando na saída as seqüências $P(u; O)$ e $P(c; O)$, calculadas de acordo com as respectivas entradas e com o conhecimento da seção da treliça do código.



Fig. 2. Módulo SISO.

As distribuições de probabilidade de entrada, $P(u; I)$ e $P(c; I)$ indicam, respectivamente, as probabilidades $P(u) = [P_k(u)]$ para os bits de informação e $P(c) = [P_k(c)]$ para os bits codificados, sendo $P_k(u) \triangleq P(U_k = u)$ e $P_k(c) \triangleq P(C_k = c)$.

A treliça de um código convolucional invariante no tempo é caracterizada por uma única seção, que descreve a transição entre estados da treliça nos instantes $k-1$ e k . A Figura 3 representa uma seção de treliça, em que $E^i(r)$ e $E^f(r)$ indicam estado inicial e estado final do ramo r , respectivamente.

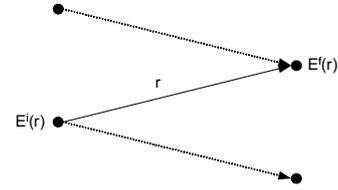


Fig. 3. Seção da treliça do codificador.

A primeira parte do algoritmo realiza o cálculo das quantidades A_k e B_k , obtidas de forma similar a α e β no algoritmo BCJR, através de recursão direta e inversa, respectivamente:

$$A_k(e) = \sum_{r: E^f(r)=e} A_{k-1} [E^i(r)] P_k [u(r); I] P_k [c(r); I], \quad (4)$$

$$k = 1, 2, \dots, L-1$$

e

$$B_k(e) = \sum_{r: E^i(r)=e} B_{k+1} [E^f(r)] P_{k+1} [u(r); I] P_{k+1} [c(r); I], \quad (5)$$

$$k = L-1, \dots, 2, 1.$$

onde L é o tamanho do bloco considerado.

Antes de se calcular 4 e 5, deve-se atribuir valores iniciais

$$A_0(e) = \begin{cases} 1, & e = E_0 \\ 0, & \text{caso contrário} \end{cases}$$

e

$$B_L(e) = \begin{cases} 1, & e = E_L \\ 0, & \text{caso contrário} \end{cases}.$$

Encontrados os valores de A_k e B_k , calcula-se, então, as distribuições de probabilidade de saída

$$P_k(u; O) = H_u \sum_{r: u(r)=u} A_{k-1} [E^i(r)] P_k [c(r); I] B_k [E^f(r)] \quad (6)$$

e

$$P_k(c; O) = H_c \sum_{r: c(r)=c} A_{k-1} [E^i(r)] P_k [u(r); I] B_k [E^f(r)], \quad (7)$$

onde H_u e H_c são constantes de normalização tais que

$$H_u \rightarrow \sum_u P_k(u; O) = 1$$

e

$$H_c \rightarrow \sum_c P_k(c; O) = 1.$$

As Eqs. 6 e 7 são versões atualizadas das distribuições de probabilidade de entrada, $P_k(u; I)$ e $P_k(c; I)$, respectivamente. Na literatura de "decodificação turbo", tais valores são chamados de *informação extrínseca* e representam a contribuição

do algoritmo SISO às distribuições de probabilidade *a priori*, $P_k(u; I)$ e $P_k(c; I)$.

Para esquemas concatenados nos quais a saída está ligada ao canal ao invés de estar ligada a outro codificador, a Eq. 7 não precisa ser calculada [10].

Como cada símbolo codificado, c_n , é formado por n_0 bits, c_n^j , $j = 1, 2, \dots, n_0$, com realização $c^j \in \{0, 1\}$, à seqüência de símbolos codificados associa-se a seqüência de distribuições de probabilidade *a priori* $P_k(c; I) = \prod_{j=1}^{n_0} P_k(c^j; I)$. As distribuições de probabilidade dos símbolos codificados podem ser representadas como produto das distribuições marginais dos bits, desde que seja utilizado um entrelaçador de bit ao invés de um entrelaçador de símbolo.

No sistema considerado na Figura 1, o primeiro módulo SISO, referente ao equalizador, está relacionado ao canal. A entrada codificada depende, neste bloco, das seqüências recebidas, r_k . Dessa forma, para se calcular $P_{1,k}[c(r); I]$, deve-se considerar a distribuição condicional

$$P_k(c; I) \propto P[R_k = r_k | Y_k = y_k] = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r_k - y_k)^2}{2\sigma^2}}. \quad (8)$$

A outra entrada deste bloco é dada pelas distribuições $P_{1,k}[u(r); I]$. Para a primeira iteração, considera-se que essa distribuição seja equiprovável. Para as demais iterações, $P_{1,k}[u(r); I]$ é gerada pelo outro SISO.

Para o segundo SISO, $P_{2,k}[c(r); I]$ é dada pela saída $P_{1,k}[u(r); O]$, após passar pelo desentrelaçador. Esse módulo recebe ainda uma distribuição equiprovável de probabilidades, $P_{2,k}[u(r); I]$, dada a característica da fonte de informação.

Resumindo, o SISO 1 calcula as distribuições de probabilidade *a posteriori* (APP) baseado nas informações do canal e o SISO 2 aplica a regra *maximum a posteriori* (MAP) para estimar o bit mais provável. Esse processo se repete a cada iteração.

É imprescindível que, para cada bloco recebido, seja realizada a terminação da treliça. Isso se faz adicionando-se N zeros à seqüência recebida, garantindo-se que a treliça retornará ao estado inicial.

IV. TÉCNICAS DE REDUÇÃO DE ESTADOS

A. Truncamento dos Coeficientes do Canal

O método de truncamento consiste em desconsiderar, no receptor, a influência das amostras de símbolos geradas em função de um ou mais *taps* de memória do canal.

Seja o canal considerado na Eq. 2. Como este canal tem quatro *taps* de memória (instantes anteriores), para modulação BPSK, tem-se, ao todo, $2^4 = 16$ estados na treliça do decodificador. Cada elemento de memória desconsiderado na recepção reduz pela metade o número de estados na treliça. Ou seja, retirando-se uma, duas ou três memórias tem-se, no decodificador, uma treliça de oito, quatro e dois estados, respectivamente.

A retirada de um ou mais *taps* de memória altera a resposta impulsiva do canal. Essa alteração leva a uma diferença considerável entre os símbolos do alfabeto transmitido através do canal e os símbolos do novo alfabeto reconhecido pelo receptor descasado. Para os casos onde são retirados dois e três

elementos de memória, há uma degradação de desempenho de tal forma que o receptor passa a decidir por símbolos errados em quase 50 por cento das vezes, independentemente do aumento de E_b/N_0 , tornando-se, assim, impraticável a sua aplicação.

A Figura 4 mostra os resultados da simulação do sistema implementado para um receptor descasado com 8 estados, referentes ao caso onde apenas um elemento de memória é retirado. Na mesma figura estão também as curvas para o canal AWGN e para o sistema de equalização turbo sem redução de estados. Nota-se que, apesar do descasamento entre transmissor e receptor, há um ganho razoável de codificação da primeira à décima iteração, indicando que o princípio turbo é verificado nesse sistema.

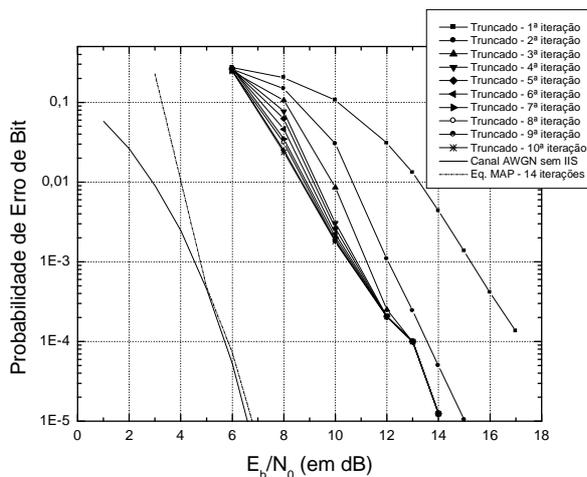


Fig. 4. Desempenho utilizando o algoritmo truncado para 8 estados.

B. Estimação dos Coeficientes do Canal

O método de estimação de coeficientes do canal consiste no uso de um modelo para a densidade espectral de potência do canal cujo número de estados é menor do que o modelo original. Considerando a forma da Eq. 2, decidimos utilizar um processo de Média Móvel, denotado $MA(q)$, onde q é a ordem do processo.

O processo MA é representado por uma equação de diferenças linear, dada por

$$x[n] = \sum_{k=0}^q b[k]u[n-k] \quad (9)$$

onde $b[0] = 1$ e $b[k]$ são os coeficientes do processo a serem estimados. Como o processo de entrada, $u[n]$, corresponde aos símbolos da modulação, pode-se obter uma representação desse modelo na forma de um diagrama de treliça invariante no tempo. A densidade espectral de potência (PSD) do processo é dada por

$$P_{MA}(f) = \sigma^2 \left| \sum_{k=0}^q b[k] \exp(j2\pi fk) \right|^2. \quad (10)$$

Uma vez escolhido o modelo, a próxima etapa consiste em aproximar a PSD do canal por uma PSD do processo $MA(q)$, $1 \leq q \leq 3$. Para uma escolha dos coeficientes do processo

que garanta uma boa aproximação, utiliza-se como critério, a minimização do erro quadrático médio, ε , dado por

$$\varepsilon = (S(f) - P_{MA}(f))^2 \quad (11)$$

onde $S(f)$ é a PSD do canal da Eq. 2.

Considerando agora L amostras das funções de densidade espectral envolvidas, a minimização do erro, ε , é equivalente a encontrar os coeficientes $b[i]$, $1 \leq i \leq 3$, e σ^2 que resolvem o sistema

$$\frac{\partial}{\partial b[i]} \left[\sum_{l=1}^L \left(S(f_l) - \sigma^2 \left| \sum_{k=0}^q b[k] \exp(j2\pi f_l k) \right|^2 \right)^2 \right] = 0, \quad 1 \leq i \leq 3 \text{ e}$$

$$\frac{\partial}{\partial \sigma^2} \left[\sum_{l=1}^L \left(S(f_l) - \sigma^2 \left| \sum_{k=0}^q b[k] \exp(j2\pi f_l k) \right|^2 \right)^2 \right] = 0 \quad (12)$$

A Tabela I apresenta os valores dos coeficientes $b[n]$ e de σ^2 para $q = 1, 2$ e 3 , encontrados como solução do sistema da Eq. 12, bem como os respectivos erros quadráticos médios.

TABELA I

PARÂMETROS DO PROCESSO MA PARA $q = 1, 2$ E 3 .

q	b[1]	b[2]	b[3]	σ^2	Erro Quadrático Médio
1	1	-	-	0,614399	0,798357
2	1,29762	1,6	-	0,223906	0,193058
3	1,23886	1,23886	1	0,205577	0,008145

Os resultados de simulação do sistema para as aproximações referentes a $q = 1$ e $q = 2$, dois e quatro estados, respectivamente, não foram bons e, tal qual no método anterior, apresentam taxa de erro de decodificação alta e insensibilidade à variação da razão sinal-ruído.

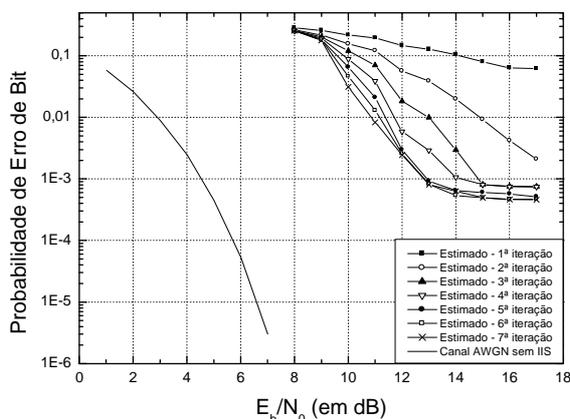


Fig. 5. Desempenho utilizando estimação dos coeficientes para 8 estados.

Para a aproximação do espectro pelo processo MA(3), os resultados foram relativamente melhores. A Figura 5 mostra os resultados das simulações desse sistema. Pode ser observado nesse gráfico que o sistema implementado, apesar de melhorar o desempenho através das iterações, converge rapidamente

para um ponto de saturação, que se encontra próximo a uma probabilidade de erro igual a 5×10^{-4} .

O erro quadrático médio da PSD da resposta impulsiva para um receptor descasado (truncamento de coeficientes) com 8 estados é igual a 0,10065. A estimação dos coeficientes do canal, embora tenha alcançado um erro relativo cerca de dez vezes menor que o erro do sistema truncado para oito estados, obteve um desempenho pior.

A Figura 6 mostra que a PSD estimada se aproxima daquela da Eq. 2 para frequências menores que 0,23, divergindo bastante a partir desse ponto, o que eleva o erro relativo entre as duas curvas.

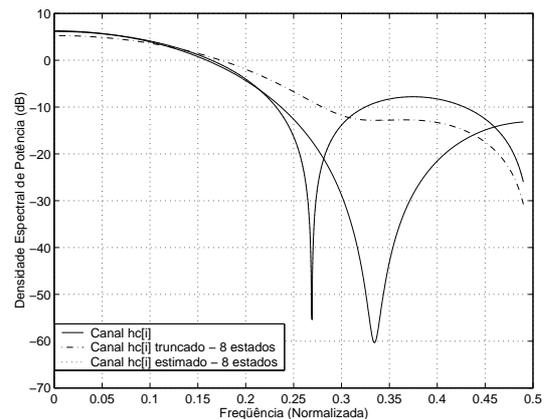


Fig. 6. Comparação entre as PSDs real, truncada e estimada.

C. Algoritmo M

O algoritmo M foi apresentado em [5] como uma variante do algoritmo BCJR original. É um algoritmo de busca reduzida que elimina o cálculo de parte das probabilidades de transição de estados na treliça.

Esse algoritmo considera o fato de que algumas componentes dos vetores α e β do BCJR, similares aos vetores A_k e B_k do SISO utilizado, possuem valores muito pequenos, se comparadas às componentes mais significativas. Pode-se, assim, removê-las dos cálculos sem perda expressiva de desempenho.

O algoritmo M mantém os M estados com maior probabilidade de ocorrência em cada seção da treliça. Assim, no cálculo de A_k e B_k , são considerados apenas os M maiores valores de A_{k-1} e B_{k+1} . Pela necessidade de efetuar comparações para determinar os M maiores valores, o algoritmo M efetua uma quantidade de cálculos superior à dos dois métodos descritos anteriormente. Entretanto, comparado ao equalizador turbo que utiliza todos os estados da treliça de canal, o algoritmo M reduz consideravelmente a quantidade de cálculos, como será visto na seção V.

A redução de estados, em geral, degrada o desempenho do sistema. Entretanto, o número médio de estados sobreviventes tende para valores baixos à medida que se aumenta E_b/N_0 e a quantidade de iterações [6]. Dessa forma, a retirada de estados da treliça com baixas probabilidades de ocorrência não obrigatoriamente diminui o desempenho do sistema.

Embora o Algoritmo M possa controlar o número de estados sobreviventes, a simulação desse método considerou $M = 8$,

para fins de comparação com os demais métodos apresentados. Os resultados desse sistema são mostrados na Figura 7.

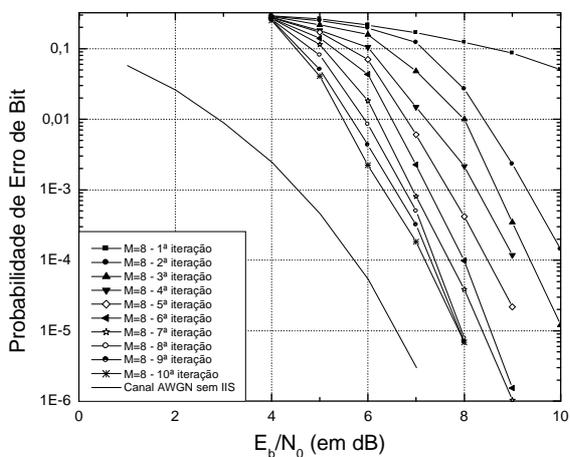


Fig. 7. Desempenho utilizando o algoritmo M para M=8 estados.

Para uma probabilidade de erro de 10^{-4} e 10 iterações, a diferença de desempenho desse sistema em relação ao canal sem IIS é de aproximadamente 1,5 dB. Com 5 iterações, a diferença sobe para quase 3 dB.

D. Comparação entre os Métodos

A Figura 8 mostra uma comparação entre os métodos com três iterações. Para uma probabilidade de erro igual a 10^{-4} , a equalização turbo sem redução de estados possui um ganho de 2,5 dB em relação ao algoritmo M. Entre os métodos de redução de estados, o algoritmo M obteve o melhor desempenho, alcançando um ganho por volta de 3,5 dB em relação ao truncamento dos coeficientes. O método de estimação dos coeficientes não conseguiu sequer atingir o patamar de erro considerado.

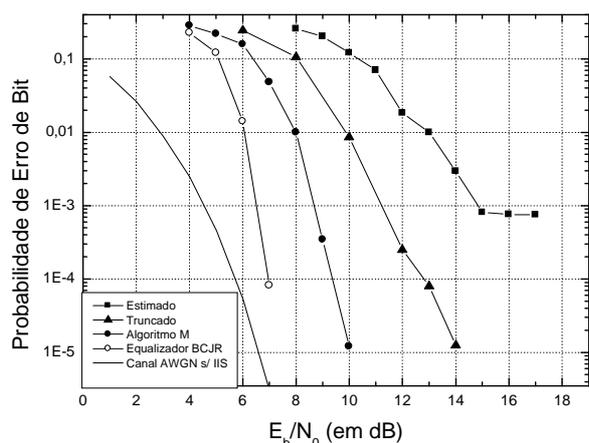


Fig. 8. Comparação entre os métodos para 3 iterações e 8 estados.

O desempenho para sete iterações é mostrado na Figura 9. Pode-se observar que o algoritmo M atinge uma perda de desempenho de apenas 2 dB em relação à um canal livre de IIS, para uma probabilidade de 10^{-4} . O desempenho do método de truncamento é pior, com perda de 5 dB em relação ao do algoritmo M para uma mesma probabilidade de erro.

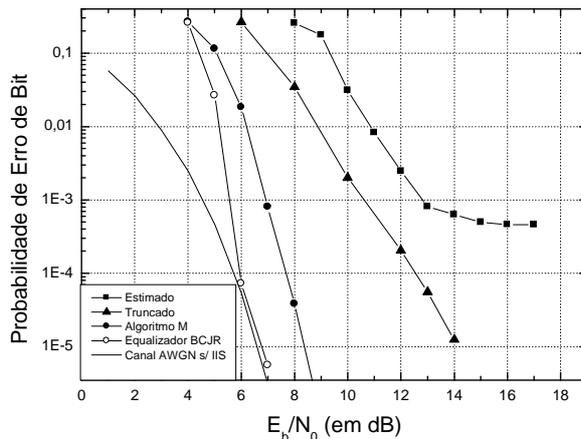


Fig. 9. Comparação entre os métodos para 7 iterações e 8 estados.

V. COMPLEXIDADE

Um aspecto relevante, quando se trata de algoritmos de equalização baseados no algoritmo BCJR, é o esforço computacional exigido no processamento dos blocos recebidos.

A complexidade é definida neste trabalho como a quantidade de operações de adição e multiplicação realizadas por símbolo recebido por iteração.

O equalizador BCJR realiza os cálculos das grandezas A_k , B_k e $P_k(U; O)$ (Eqs. 4, 5 e 6). Para cada seção da treliça, há um total de $3.2^m \cdot 2^N$ multiplicações e $3.2^{N-1}(2^m - 1)$ adições, onde 2^m é o tamanho do alfabeto da constelação de sinais e N , como definido anteriormente, é o comprimento da resposta ao impulso do canal.

No casos de truncamento e estimação dos coeficientes do canal, o número de operações exigido é idêntico à de um algoritmo BCJR com número de estados reduzido.

Já no caso do algoritmo M, como o número de estados da treliça do canal que serão utilizados é uma variável aleatória, a complexidade precisa ser caracterizada através de um valor médio (μ) do número de ramos sobreviventes.

A Figura 10 mostra dois exemplos onde há variação na posição dos A_k e B_k eliminados, onde apenas dois estados dos quatro iniciais são mantidos, isto é, $M = 2$.

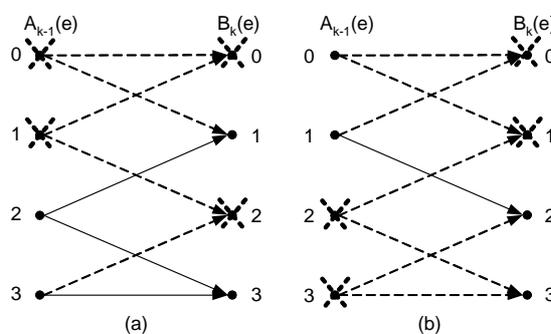


Fig. 10. Exemplo de diferença na quantidade de ramos da treliça considerados no cálculo de A_k , B_k e $P_k(U; O)$ para o algoritmo M.

Na Figura 10(a), como os valores de $A_{k-1}(0)$, $A_{k-1}(1)$, $B_k(0)$ e $B_k(2)$ são zerados, no cálculo de $P_k(U; O)$ são

considerados três ramos (linhas contínuas). Os ramos pontilhados são eliminados. Na Figura 10(b), zerando-se $A_{k-1}(2)$, $A_{k-1}(3)$, $B_k(0)$ e $B_k(1)$, apenas um ramo é considerado em $P_k(U; O)$, todos os demais são eliminados. De maneira geral, o valor de μ pode ser obtido considerando as combinações possíveis de posições de A_k e B_k .

É importante observar que, logo após A_k e B_k serem calculados, há um processo de comparação e ordenamento de seus valores para todos os estados. Essa rotina, para fins de comparação de complexidade, envolve poucos cálculos e não é considerada neste trabalho.

Para 16 estados e $M = 8$, tanto A_k quanto B_k possui, ao todo, $\frac{16!}{8!(16-8)!}$ possíveis combinações. O número médio de ramos sobreviventes, nesse caso, é 10, ou 31,25% do total de ramos.

A Tabela II mostra o volume de operações necessárias para os diversos métodos estudados.

TABELA II

VOLUME DE OPERAÇÕES DE ADIÇÃO E MULTIPLICAÇÃO POR SÍMBOLO RECEBIDO POR ITERAÇÃO.

Método	Adições	Multiplicações
Equalizador BCJR	$3 \cdot 2^{N-1} (2^m - 1)$	$3 \cdot 2^m \cdot 2^N$
Truncamento	$3 \cdot 2^{N-1} (2^m - 1)$	$3 \cdot 2^m \cdot 2^N$
Estimação	$3 \cdot 2^{N-1} (2^m - 1)$	$3 \cdot 2^m \cdot 2^N$
Algoritmo M	$3 \cdot 2^m \frac{\mu}{2}$	$2^m (2^N + 2\mu)$

Para o caso em que o número de estados é reduzido de 16 para 8 estados, a redução no volume de operações em relação ao equalizador BCJR é mostrada na Tabela III. Considera-se modulação binária, isto é, $m = 1$.

TABELA III

VOLUME DE OPERAÇÕES DE ADIÇÃO E MULTIPLICAÇÃO POR SÍMBOLO RECEBIDO POR ITERAÇÃO.

Método	Adições	Multiplicações	Total	Redução (%)
Equalizador MAP	48	192	240	-
Truncamento	24	96	120	50
Estimação	24	96	120	50
Algoritmo M	30	104	134	44

Há uma redução de complexidade de aproximadamente 50% dos métodos de truncamento e estimação dos coeficientes do canal em relação ao equalizador BCJR. A redução obtida pelo algoritmo M é de 44%.

VI. CONCLUSÕES

A estimação dos coeficientes do canal apresentou um desempenho pobre. Uma justificativa pode ser a própria escolha do modelo. A PSD desse canal apresenta um pico negativo estreito. Como mostrado em [11], processos MA de ordem pequena não são adequados para aproximar este tipo de comportamento espectral. Neste caso, seria melhor usar um modelo ARMA ou AR [11]. Entretanto, a escolha do processo MA se deve à necessidade de representação do modelo na forma de uma treliça invariante no tempo. Outra justificativa pode estar na falta de informação sobre a fase da resposta impulsiva, já que somente a amplitude desta é considerada na aproximação da PSD do canal.

Embora o volume de cálculos do algoritmo M seja ligeiramente maior que o dos demais métodos, ele sempre apresentou o melhor compromisso entre desempenho e complexidade.

Em [3], diferentes métodos de equalização linear foram considerados para a simplificação do equalizador turbo. Resultados de desempenho foram obtidos para um bloco de informação de 32.000 bits utilizando um codificador convolucional sistemático e recursivo. Comparações de desempenho destes resultados com os mostrados neste trabalho são descritas em [2]. Para uma taxa de erro igual a 10^{-4} e três iterações, o desempenho do algoritmo M aproxima-se ao do equalizador linear MMSE, o método de truncamento equivale ao do equalizador linear MMSE aproximado e o método de estimação de coeficientes supera o do equalizador MMSE DFE.

AGRADECIMENTOS

Os autores agradecem ao Professor Amauri Lopes por sugestões que elevaram a qualidade deste trabalho.

REFERÊNCIAS

- [1] C. Douillard, M. Jésumél, A. Picart, P. Didier and A. Glavieux, "Iterative Correction of Intersymbol Interference: Turbo-Equalization" *European Transactions on Telecommunications*, v. 6, n. 5, p. 507-511, September-October 1995.
- [2] M. D. Cosac, "Técnicas de Redução de Estados Aplicadas à Equalização Turbo", Tese de Mestrado, Fevereiro 2004.
- [3] R. Koetter, M. Tüchler and A. C. Singer, "Turbo Equalization: Principles and New Results" *IEEE Transactions on Communications*, v. 50, n. 5, p. 754-767, May 2002.
- [4] G. Ferrari, G. Colavolpe and R. Raheli, "Reduced-State BCJR-Type Algorithms" *IEEE Journal on Selected Areas in Communications*, v. 19, n. 5, p. 848-859, May 2001.
- [5] V. Franz and J. B. Anderson, "Concatenated Decoding with a Reduced-Search BCJR Algorithm" *IEEE Journal on Selected Areas in Communications*, v. 16, n. 2, p. 186-195, February 1998.
- [6] H. Freire and J. Portugheis, "Algoritmo H-BCJR de Busca Reduzida para Código Turbo" *XVIII Simpósio Brasileiro de Telecomunicações*, Gramado, RS-Brasil, Setembro 2000.
- [7] C. Fragouli, N. Al-Dhahir, S. N. Diggavi and W. Turin, "Prefiltered Space-Time M-BCJR Equalizer for Frequency-Selective Channels" *IEEE Transactions on Communications*, v. 50, n. 5, p. 742-753, May 2002.
- [8] J. G. Proakis, "Digital Communications", McGraw-Hill, 3ª edição, 1995.
- [9] F. Jelinek, L. R. Bahl, J. Cocke and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate" *IEEE Transactions on Information Theory*, n. 4, p. 284-287, March 1974.
- [10] G. Montorsi, S. Benedetto, D. Divsalar and F. Pollara, "Soft-Input Soft-Output Modules for the Construction and Distributed Iterative Decoding of Code Networks" *European Transactions on Telecommunications Special Issue*, v. 9, n. 2, p. 165-172, March-April 1998.
- [11] S. M. Kay, "Modern Spectral Estimation: Theory and Application", Prentice Hall, 1ª edição, 1988.