

Equalização Turbo: Fundamentos e Aplicações

Murilo Bellezoni Loiola, Renato da Rocha Lopes e João Marcos Travassos Romano

Resumo—Neste tutorial, apresentamos os equalizadores turbo e detalhamos seus princípios de funcionamento. Diferentemente dos receptores convencionais, nos quais equalização e decodificação de canal são feitas separadamente, os equalizadores turbo realizam estas duas tarefas conjuntamente através de um procedimento iterativo. Desta maneira, o equalizador pode se beneficiar da robustez introduzida pelo código corretor de erros para tentar eliminar a interferência intersimbólica de uma maneira mais efetiva. Discutiremos brevemente os principais algoritmos utilizados nos equalizadores turbo, mostrando alguns resultados de simulação. Por fim, citaremos algumas das recentes aplicações dos equalizadores turbo em sistemas de comunicações digitais.

Palavras-Chave—Equalização turbo, códigos corretores de erros, algoritmos SISO.

Abstract—In this work, we present the turbo equalizers and detail their principles. Instead of making channel equalization and channel decoding separately, as in conventional receivers, turbo equalizers jointly perform these two tasks through an iterative process. Hence, the equalizer benefit from the error correction capabilities of the codes to mitigate the intersymbol interference in a more effective way. We also present the main turbo equalization algorithms and show some simulation results. Last, we show some recent applications of turbo equalizers in digital communication systems.

Keywords—Turbo equalization, error correction codes, SISO algorithms.

I. INTRODUÇÃO

A utilização cada vez mais intensa de sistemas de comunicações digitais exige a busca por técnicas capazes de promover um aumento da taxa de transmissão de dados e uma melhora na qualidade do sinal recebido. Podemos dizer que o principal obstáculo para a concretização desses objetivos é o canal de comunicação, o meio físico pelo qual se propaga o sinal que carrega as informações. Além de introduzir ruído, o canal pode fazer com que o sinal transmitido sofra um espalhamento temporal, fazendo com que dados enviados num determinado instante interfiram com dados transmitidos em outros instantes. Este fenômeno, chamado de interferência intersimbólica (IIS), é um dos principais fatores que limitam o desempenho dos sistemas de comunicações, provocando redução da confiabilidade e/ou da taxa de transmissão. Exemplos de fatores responsáveis pelo surgimento da IIS são a propagação por múltiplos percursos, como em sistemas de comunicações móveis, e a largura de faixa limitada, como em canais telefônicos.

Para minimizar os efeitos da IIS, o receptor normalmente emprega um filtro denominado equalizador [1]. Já para aumen-

tar a imunidade do sinal transmitido ao ruído, frequentemente são utilizados códigos corretores de erros. Estes códigos acrescentam uma redundância controlada à mensagem enviada, possibilitando ao receptor detectar e/ou corrigir erros ocorridos durante a transmissão.

Em esquemas clássicos de recepção, os equalizadores não utilizam esta redundância no combate aos efeitos da IIS, sendo equalização e decodificação realizadas separadamente. Uma alternativa interessante para a melhoria significativa do desempenho do receptor surgiu com a equalização turbo [2]. Nesta nova abordagem, inspirada na decodificação dos chamados códigos turbo, equalização e decodificação são realizadas conjuntamente através de um procedimento iterativo por bloco. A idéia é fazer com que o equalizador utilize a saída do decodificador como uma informação *a priori* sobre os símbolos enviados. Desta forma, o equalizador pode aprimorar suas saídas, que serão empregadas pelo decodificador para produzir estimativas mais refinadas da mensagem transmitida, recomeçando o ciclo. É exatamente esta troca iterativa de informações entre equalizador e decodificador que faz com que os equalizadores turbo alcancem taxas de erro de bit muito menores que aquelas obtidas pelos receptores não-iterativos convencionais. Em outras palavras, a equalização turbo permite que se alcance uma certa taxa de erro de bit com uma potência de transmissão menor que a das técnicas usuais (não iterativas).

Portanto, o objetivo deste trabalho é detalhar os princípios de funcionamento dos equalizadores turbo, bem como apresentar alguns dos algoritmos mais usados neste esquema de recepção. Também discutiremos brevemente algumas aplicações dos equalizadores turbo. Assim, o restante deste tutorial está dividido da seguinte maneira: na seção II, descreveremos os fundamentos dos equalizadores turbo. Na seção III, apresentaremos os principais algoritmos de complexidade exponencial empregados como equalizador, enquanto que na seção IV exporemos alguns equalizadores de complexidade reduzida. Já na seção V, serão mostrados alguns resultados de simulações comparando alguns dos algoritmos descritos nas seções anteriores. Algumas das aplicações dos equalizadores turbo serão apresentadas na seção VI. Finalmente, na seção VII, são tecidas algumas conclusões sobre o trabalho.

II. EQUALIZAÇÃO TURBO

Este artigo considera a transmissão de uma mensagem binária m através de um canal linear com IIS, conforme mostrado na Fig. 1. Este canal discreto equivalente pode ser visto como um “código convolucional” não-sistemático, não-binário, de taxa $R = 1$ e possivelmente variante no tempo [2]. Deste ponto de vista, codificação de canal e o próprio canal formariam um esquema de códigos serialmente concatenados.

Murilo Bellezoni Loiola, Renato da Rocha Lopes e João Marcos Travassos Romano, Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, Brasil, E-mails: {mloiola, rlopes, romano}@decom.fee.unicamp.br. Este trabalho foi financiado pela FAPESP (02/11983-0).

Conseqüentemente, o receptor deve ser formado por uma concatenação de “decodificadores”.

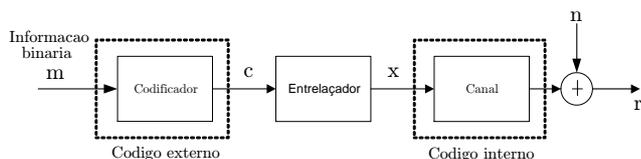


Fig. 1. Concatenação do código corretor de erros com o canal.

Conforme pode ser visto na Fig. 2, em um equalizador turbo, o equalizador “decodifica” o código interno, ou seja, o canal, enquanto o decodificador externo faz a decodificação de canal. A idéia básica da equalização turbo é fazer com que o equalizador utilize as saídas do decodificador como uma informação *a priori* sobre os símbolos transmitidos. Esta informação, também chamada de informação *extrínseca*, corresponde a uma informação incremental sobre o símbolo enviado num instante qualquer obtida através da decodificação de todos os outros símbolos da seqüência recebida.

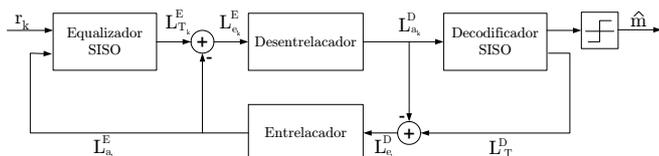


Fig. 2. Diagrama de um equalizador turbo.

Assim como nos códigos turbo, os componentes do equalizador turbo devem ser dispositivos SISO (*Soft-Input/Soft-Output*), capazes de trocar entre si decisões suaves. Estas decisões suaves são normalmente expressas por meio de LLR's (*Log-Likelihood Ratio*), definidas da seguinte maneira para uma modulação BPSK:

$$L(m_k) \triangleq \ln \frac{P(m_k = +1|\mathbf{r})}{P(m_k = -1|\mathbf{r})}, \quad (1)$$

onde m_k é o bit da mensagem no instante k e \mathbf{r} é a seqüência recebida. Convém notar que $\text{sign}(L(m_k))$ fornece a decisão MAP, \hat{m}_k , e o módulo de $L(m_k)$ indica a confiabilidade da decisão tomada estar correta. Logo, na Fig. 2, L_T representa a razão de probabilidades *a posteriori* indicada em (1), os sobrescritos “E” e “D” se referem, respectivamente, ao equalizador e ao decodificador, o subscrito “e” indica informação extrínseca, o subscrito “a”, informação *a priori* e o subscrito “k” é o índice temporal.

O funcionamento do equalizador turbo da Fig. 2 pode então ser descrito, resumidamente, da seguinte forma: na primeira iteração, o equalizador não dispõe de nenhuma informação *a priori* sobre os símbolos transmitidos, ou seja, os valores $L_{a_k}^E$ são nulos. Logo, o equalizador produz as LLR's dos bits codificados utilizando apenas a seqüência ruidosa observada \mathbf{r} . Esses valores são desentrelaçados e passados ao decodificador, que gera novas estimativas das LLR's. Em seguida, as informações extrínsecas são calculadas, subtraindo das LLR's a informação *a priori* $L_{a_k}^D$, e entrelaçadas para o correto ordenamento dos

bits, terminando a primeira iteração. Nas iterações seguintes, todo o ciclo se repete, com as estimativas da mensagem sendo refinadas a cada iteração. Vale notar que a subtração mencionada anteriormente gera a informação extrínseca pois, ao se subtrair a informação *a priori* da LLR gerada pelo decodificador, obtém-se a informação incremental produzida pelo decodificador na iteração atual [2].

Uma das chaves para o excelente desempenho dos equalizadores turbo reside no fato de equalizador e decodificador trocarem entre si apenas informações extrínsecas. Isto garante que cada componente SISO utilize como informação *a priori* somente valores que não foram gerados diretamente por ele próprio na iteração anterior. Isto evita que a malha de realimentação tenha *feedback* positivo, o que poderia gerar instabilidade do sistema. Outro elemento essencial nos receptores iterativos é o entrelaçador. Ele tem a função de espalhar o erro por todo o bloco de dados [2]–[4] além de promover o correto ordenamento dos bits.

Nas seções seguintes, apresentaremos alguns dos algoritmos de equalização mais empregados nos equalizadores turbo. É importante salientar que os algoritmos descritos na seção III também podem ser empregados como decodificador.

III. ALGORITMOS COM COMPLEXIDADE EXPONENCIAL

A. Algoritmo MAP (BCJR)

O algoritmo normalmente empregado na equalização turbo é o BCJR [5], que é ótimo no sentido de minimizar a probabilidade de erro de bit. Este algoritmo símbolo-a-símbolo foi proposto por Bahl *et al* [5] em 1974. Ele calcula a probabilidade *a posteriori* de cada transição de estado bem como dos bits de mensagem de um processo de Markov, dada uma seqüência ruidosa observada. De acordo com este algoritmo, (1) pode ser reescrita como

$$L(x_k) = \ln \frac{P(x_k = +1|\mathbf{r})}{P(x_k = -1|\mathbf{r})} = \ln \frac{\sum_{\{(s',s):x_k=+1\}} p(s',s,\mathbf{r})}{\sum_{\{(s',s):x_k=-1\}} p(s',s,\mathbf{r})}, \quad (2)$$

onde s' e s representam os estados da treliça nos instantes $k-1$ e k , respectivamente. Os somatórios do numerador e do denominador são realizados para todas as transições existentes entre os estados s' e s quando o bit de informação é, respectivamente, $x_k = +1$ e $x_k = -1$.

Usando as propriedades de um processo de Markov, é possível mostrar [5] que as probabilidades conjuntas $p(s',s,\mathbf{r})$ podem ser escritas como o produto de três termos independentes:

$$\begin{aligned} p(s',s,\mathbf{r}) &= p(s',\mathbf{r}_{j<k}) \cdot p(s,\mathbf{r}|s') \cdot p(\mathbf{r}_{j>k}|s) \\ &= \underbrace{p(s',\mathbf{r}_{j<k})}_{\alpha_{k-1}(s')} \cdot \underbrace{P(s|s') \cdot p(r_k|s',s)}_{\gamma_k(s',s)} \cdot \underbrace{p(\mathbf{r}_{j>k}|s)}_{\beta_k(s)}. \end{aligned} \quad (3)$$

Aqui, $\mathbf{r}_{j<k}$ representa a seqüência de símbolos recebidos do instante inicial até o instante $k-1$, enquanto que $\mathbf{r}_{j>k}$ é seqüência entre o instante $k+1$ e o final da treliça.

Ainda utilizando as propriedades de um processo de Markov, os valores de $\alpha_k(s)$ e $\beta_k(s)$ podem ser calculados

recursivamente da seguinte maneira:

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

$$\beta_{k-1}(s') = \sum_s \gamma_k(s', s) \cdot \beta_k(s). \quad (4b)$$

Considerando uma treliça de duração finita que inicia e termina no estado zero (s_0), $\alpha_k(s)$ e $\beta_k(s)$ são inicializados como segue:

$$\begin{aligned} \alpha_{inicial}(0) = 1 & \quad e \quad \alpha_{inicial}(s) = 0 \quad \forall s \neq 0 \\ \beta_{final}(0) = 1 & \quad e \quad \beta_{final}(s) = 0 \quad \forall s \neq 0 \end{aligned} \quad (5)$$

O algoritmo BCJR realiza o cálculo dos valores de α_k , indo do instante inicial ao final (recursão *forward*). Já os valores de β_k são obtidos deslocando-se do final para o início da treliça (recursão *backward*).

O termo $\gamma_k(s', s)$ nas equações (4) depende da saída atual do canal e, conforme pode ser observado na equação (3), é expresso em função de $P(s|s')$ e da probabilidade condicional $p(r_k|s', s)$ [6], [7]:

$$P(s|s') = P(x_k) = \begin{cases} \frac{e^{L_{\alpha_k}}}{1+e^{L_{\alpha_k}}} & \text{para } x_k = 1 \\ \frac{1}{1+e^{L_{\alpha_k}}} & \text{para } x_k = -1 \end{cases} \quad (6)$$

e

$$p(r_k|s', s) = \exp\left(-\frac{1}{2\sigma_n^2} \cdot \left|r_k - \sum_{i=0}^{\mu} h_i(t) \cdot x_{k-i}\right|^2\right). \quad (7)$$

onde L_{α_k} é a informação *a priori* produzida pelo decodificador, $h_i(t)$, $i = 0 \dots \mu$, são os coeficientes do canal, σ_n^2 é a variância do ruído aditivo branco e gaussiano de média nula e x_{k-i} são os símbolos associados à transição entre os estados s e s' .

A saída do decodificador MAP pode, portanto, ser reescrita da seguinte forma:

$$L(x_k) = \ln \frac{\sum_{\{(s', s): m_k = +1\}} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)}{\sum_{\{(s', s): m_k = -1\}} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)}. \quad (8)$$

As referências [5], [7] apresentam uma descrição detalhada do algoritmo BCJR.

B. Algoritmo Log-MAP

O algoritmo BCJR da seção anterior apresenta alta complexidade computacional uma vez que requer o cálculo de um grande número de exponenciais e multiplicações. O BCJR pode, ainda, apresentar instabilidade devido à representação numérica com precisão finita. Uma forma de diminuir a carga computacional e a sensibilidade numérica do algoritmo MAP consiste em realizar todas as operações no domínio logarítmico. O algoritmo assim obtido é chamado de Log-MAP [8], [9].

As expressões resultantes do algoritmo Log-MAP podem ser calculadas mais facilmente através da utilização do *logaritmo Jacobiano*, definido como

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_1 - \delta_2|}). \quad (9)$$

O termo $\ln(1 + e^{-|\delta_1 - \delta_2|})$ tende a zero à medida que a diferença entre δ_1 e δ_2 aumenta. Logo, ao invés de calcularmos essa função para cada valor de δ_1 e δ_2 , podemos consultá-la em uma tabela unidimensional construída antecipadamente. Em [8], verificou-se que apenas 8 valores de $|\delta_1 - \delta_2|$ na faixa de 0 a 5 são suficientes para um excelente desempenho do algoritmo. A partir de (9), a expressão $\ln(e^{\delta_1} + \dots + e^{\delta_n})$ pode ser calculada recursivamente supondo que $\delta = \ln(e^{\delta_1} + \dots + e^{\delta_{n-1}})$ é conhecido. Assim, considerando $\Delta = e^{\delta_1} + \dots + e^{\delta_{n-1}} = e^\delta$, tem-se que

$$\begin{aligned} \ln(e^{\delta_1} + \dots + e^{\delta_n}) &= \ln(\Delta + e^{\delta_n}) \\ &= \max(\ln \Delta, \delta_n) + f_c(|\ln \Delta - \delta_n|) \\ &= \max(\delta, \delta_n) + f_c(|\delta - \delta_n|). \end{aligned} \quad (10)$$

onde $f_c(|\delta - \delta_n|) = \ln(1 + e^{-|\delta - \delta_n|})$. É importante mencionar que o algoritmo Log-MAP, calculado através do logaritmo Jacobiano, é totalmente equivalente ao BCJR [8].

C. Algoritmo Max-Log-MAP

Ainda considerando o comportamento de $\ln(1 + e^{-|\delta_1 - \delta_2|})$, uma aproximação razoável é obtida simplesmente ignorando este termo. Então, o logaritmo Jacobiano é escrito como

$$\ln(e^{\delta_1} + e^{\delta_2}) \approx \max(\delta_1, \delta_2) \quad (11)$$

e o algoritmo assim obtido é conhecido como Max-Log-MAP [8], [9].

Como consequência da aproximação feita em (11), o Max-Log-MAP considera a cada instante apenas dois caminhos da treliça do canal: o melhor para um bit transmitido $x_k = +1$ e o melhor para $x_k = -1$, sendo que um deles é o caminho de máxima verossimilhança. Desta maneira, as decisões abruptas produzidas pelo Max-Log-MAP são idênticas às produzidas pelo algoritmo de Viterbi [8]. Já o MAP e o Log-MAP levam em consideração todos os caminhos da treliça para calcular as LLR dos bits transmitidos e suas decisões podem diferir daquelas produzidas pelo algoritmo de Viterbi.

D. Algoritmo de Viterbi com Saída Suave

O SOVA (*Soft Output Viterbi Algorithm*) [10], é uma variante do algoritmo de Viterbi que produz não apenas a seqüência mais verossímil como também a probabilidade *a posteriori*, ou valor de confiabilidade, de cada símbolo decodificado.

Para que o SOVA forneça estimativas das LLR's para cada símbolo decodificado, deve-se acrescentar ao esquema clássico um estágio responsável pelo cálculo das probabilidades. Assim, para cada estado pertencente a cada um dos caminhos sobreviventes, observa-se o caminho competidor até o ponto onde eles inicialmente divergiram. A diferença entre as métricas dos dois caminhos é usada para atualizar a "confiabilidade" dos símbolos que diferiram. Se, para um símbolo qualquer, a mesma decisão foi feita em ambos os caminhos, então a probabilidade *a posteriori* não é atualizada.

Uma diferença entre o BCJR e o SOVA reside no fato que os estados estimados pelo algoritmo de Viterbi com saída

suave devem formar um caminho factível através da treliça enquanto que os estados estimados pelo algoritmo BCJR não estão, necessariamente, conectados.

As referências [9], [10] apresentam discussões detalhadas sobre o SOVA, incluindo pseudo-códigos para a implementação do algoritmo.

E. Algoritmos de Redução de Estados

Os algoritmos desta família procuram reduzir um pouco a complexidade do BCJR através da eliminação de alguns estados da treliça do canal. Esses algoritmos consideram o fato de que alguns valores de α e β no BCJR são muito pequenos, podendo ser removidos sem grande perda de desempenho. Dentre os vários algoritmos de redução de estados, o mais conhecido é o algoritmo *M* [11].

O algoritmo *M* mantém os *M* estados com maior probabilidade em cada seção da treliça. Assim, no cálculo de α_k e β_{k-1} , são considerados apenas os *M* maiores valores de α_{k-1} e β_k . A redução de complexidade não é tão grande pois, embora apenas *M* ramos sejam considerados para o cálculo de (8), os valores de α e β são calculados para todos os estados e, somente depois de comparados, são descartados os estados com menores valores.

IV. ALGORITMOS COM COMPLEXIDADE REDUZIDA

O grande inconveniente dos algoritmos apresentados na seção anterior é que a complexidade computacional cresce exponencialmente com o comprimento do canal. O problema fica ainda maior uma vez que na equalização turbo, equalização e decodificação são realizadas várias vezes para um mesmo bloco de dados. Surge então a necessidade de se procurar soluções mais simples. Muitos esforços têm sido dedicados a esse respeito e esquemas utilizando equalizadores lineares [3], [4], [12] têm sido aplicados com sucesso. A seguir, discutiremos brevemente estes equalizadores SISO lineares.

A. Cancelamento de Interferência

Os equalizadores SISO lineares propostos em [3], [4] se baseiam na estrutura mostrada na Fig. 3. Nesta estrutura, o

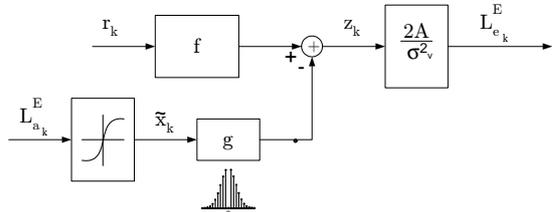


Fig. 3. Cancelamento de interferência.

signal recebido num instante *k*, r_k , é filtrado por um filtro linear *f*, cuja saída contém IIS residual. Para cancelá-la, podemos utilizar a informação *a priori*, $L_{a_k}^E$, proveniente do decodificador, para produzir estimativas dos símbolos interferentes.

Uma vez que a saída do equalizador será usada para estimar o símbolo transmitido no instante *k*, a influência deste símbolo não deve ser cancelada. Isto é feito restringindo o coeficiente

central de *g* a zero. Assim, a saída do equalizador no instante *k* não depende de $L_{a_k}^E(x_k)$ e, portanto, z_k pode ser usado para produzir informações extrínsecas.

O equalizador proposto em [3] considera que as estimativas dos símbolos interferentes estão corretas. Portanto, *f* é escolhido como o filtro casado ao canal. Já o equalizador apresentado em [4] é escolhido segundo o critério MMSE e depende da informação *a priori*. Conseqüentemente, seus coeficientes devem ser computados para cada símbolo transmitido, resultando num equalizador variante no tempo e cuja complexidade computacional é quadrática em relação ao comprimento de *f*. Ainda em [4], é proposto um outro equalizador, chamado de equalizador híbrido (ou SE - *switched equalizer*). Este equalizador, que possui complexidade linear, chaveia o filtro *f* entre um filtro casado e um filtro MMSE, dependendo da qualidade das informações *a priori* recebidas do decodificador.

B. Equalizador com Realimentação Suave

O SFE (*Soft-Feedback Equalizer*), proposto em [12], [13] e apresentado na Fig. 4, é uma estrutura SISO de cancelamento de interferência que possui algumas características similares às de um DFE (*Decision-Feedback Equalizer*). Diferentemente dos equalizadores descritos na seção IV-A, que utilizam apenas a informação *a priori* fornecida pelo decodificador para cancelar toda a interferência sobre o símbolo z_k , o SFE combina as informações extrínsecas produzidas por ele próprio com as informações *a priori* fornecidas pelo decodificador para obter estimativas mais confiáveis dos símbolos interferentes.

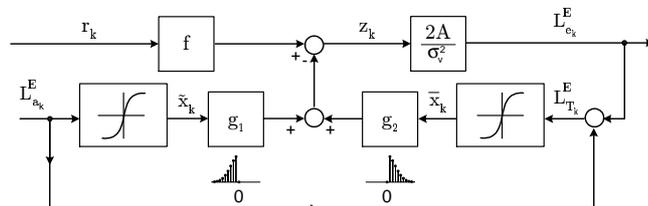


Fig. 4. Equalizador com realimentação suave.

No SFE, o filtro g_1 é estritamente anticausal e o filtro g_2 é estritamente causal. Logo, a saída do equalizador num instante *k*, z_k , não depende da informação *a priori* gerada pelo decodificador no instante *k*. Assim, z_k pode ser usada para produzir informações extrínsecas. Como mostrado em [12], [13], os coeficientes do SFE são invariantes no tempo, levando a uma complexidade computacional por símbolo proporcional ao número de coeficientes do filtro *f*.

Ainda é possível mostrar [12], [13] que o SFE se reduz a outros equalizadores bastante conhecidos, dependendo da confiabilidade de $L_{a_k}^E$ e $L_{T_k}^E$. Assim, quando $L_{a_k}^E$ e $L_{T_k}^E$ são pouco confiáveis, nenhum cancelamento de interferência é feito e o SFE se reduz a um equalizador MMSE linear *f*. Já quando $L_{a_k}^E$ é pouco confiável e $L_{T_k}^E$ tem um valor alto, o SFE se reduz a um DFE convencional, tentando cancelar apenas IIS pós-cursora. Por fim, quando as informações *a priori* $L_{a_k}^E$ apresentam uma confiabilidade alta, o SFE se reduz ao equalizador da Fig. 3.

V. COMPARAÇÃO DE DESEMPENHO

Nesta seção, apresentamos um estudo comparativo do desempenho de alguns dos equalizadores turbo apresentados nas seções III e IV. Em todas as simulações foi considerada a transmissão de 10^6 bits de informação para cada relação sinal-ruído. Os entrelaçadores foram gerados aleatoriamente e os bits da mensagem foram codificados pelo código convolucional sistemático recursivo de taxa 1/2 e geradores (7,5). Os bits codificados foram mapeados para os símbolos da constelação BPSK antes da transmissão e o ruído foi considerado como aditivo, branco, gaussiano, de média nula e de potência definida pela relação sinal-ruído. A decodificação foi realizada pelo algoritmo Log-MAP e o receptor possuía perfeito conhecimento do canal e da potência do ruído.

A Fig. 5 mostra os resultados obtidos após 15 iterações turbo quando blocos de bits de informação de tamanhos variados são transmitidos pelo canal $0,23 + 0,46z^{-1} + 0,688z^{-2} + 0,46z^{-3} + 0,227z^{-4}$. Nesta figura, além do BCJR e do SFE, traçamos a curva do equalizador MMSE proposto em [4]. Como podemos notar, todos os equalizadores turbo simulados atingem a curva do canal AWGN, removendo toda a IIS a partir de uma certa relação sinal-ruído.

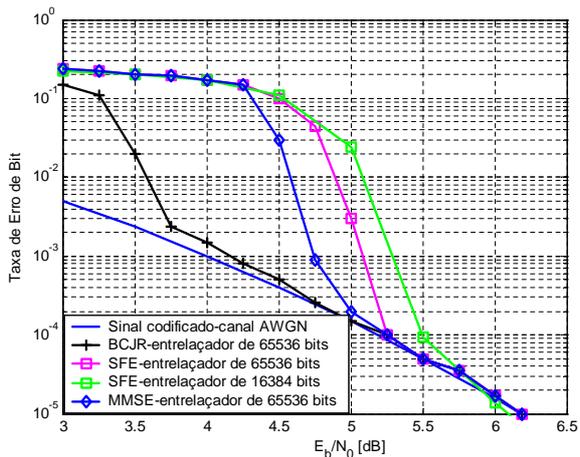


Fig. 5. Desempenho de equalizadores turbo com vários tamanhos de entrelaçador

No segundo cenário de simulação, consideramos a transmissão de blocos de 1024 bits de informação pelo canal $0,23 + 0,42z^{-1} + 0,52z^{-2} + 0,52z^{-3} + 0,42z^{-4} + 0,23z^{-5}$. Na Fig. 6 estão as curvas obtidas após a 16ª iteração para o BCJR, o Log-MAP, o SFE e o SE. Pela Fig. 6, vemos que a degradação de desempenho do SFE em relação ao BCJR e ao Log-MAP é de aproximadamente 3 dB para uma BER de 10^{-3} . Já o SE tem uma perda de cerca de 7 dB em relação ao BCJR.

VI. APLICAÇÕES

Além das aplicações usuais em sistemas de comunicações móveis, os equalizadores turbo também vêm sendo explorados recentemente em sistemas de gravação magnética. Nestes sistemas, o sinal recebido tradicionalmente passa por um equalizador de resposta parcial (ERP), cuja função é transformar

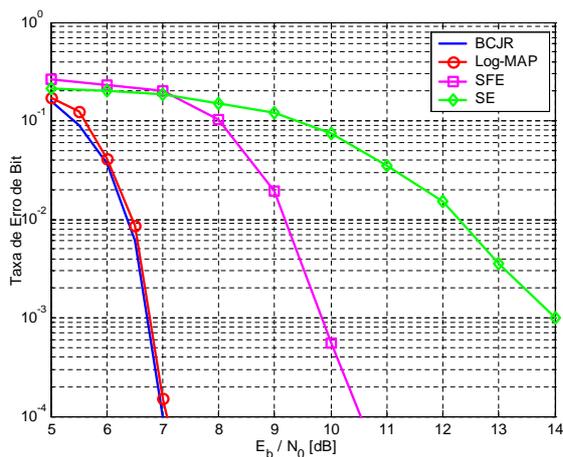


Fig. 6. Desempenho de alguns equalizadores turbo para o canal $0,23 + 0,42z^{-1} + 0,52z^{-2} + 0,52z^{-3} + 0,42z^{-4} + 0,23z^{-5}$.

a resposta ao impulso do canal em uma resposta curta e pré-estabelecida, de forma que um equalizador de máxima verossimilhança possa ser usado. Entretanto, o ERP é subótimo, causando, entre outros, amplificação de ruído. Uma alternativa é remover o ERP e substituí-lo por um equalizador turbo. Evidentemente, sem o ERP o canal de gravação magnética fica excessivamente longo, tornando inviável o uso do BCJR. Assim, em [14], o SFE é usado no lugar do BCJR, alcançando altas densidades de gravação e sem perda de desempenho. A Fig. 7 mostra a complexidade (número de adições e multiplicações) dos algoritmos BCJR e SFE para um canal de gravação magnética Lorentziano com densidade $D = 3$.

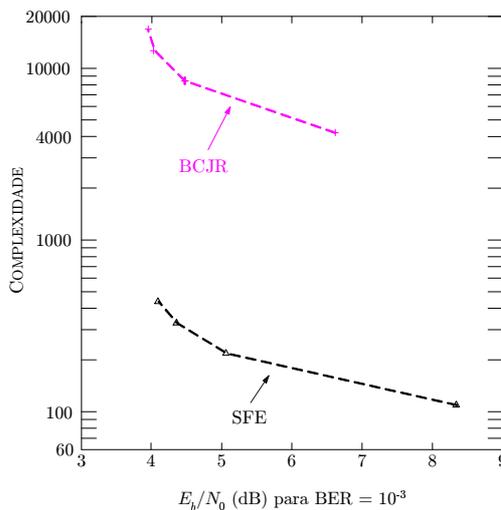


Fig. 7. Relação complexidade-desempenho para um canal de gravação magnética.

Uma outra aplicação na qual a equalização turbo pode ter desempenho superior ao das técnicas convencionais é na minimização dos efeitos do ruído impulsivo. Este fenômeno, comum em vários sistemas de comunicações digitais, degrada significativamente o sinal recebido. Em [15], é desenvolvido

um equalizador bayesiano cego próprio para ambientes com este tipo de ruído. Já em [16], é proposto um equalizador *fuzzy* iterativo para mitigar os efeitos do ruído impulsivo em canais PLC.

VII. CONCLUSÕES

Neste trabalho apresentamos os conceitos básico da equalização turbo. Neste novo paradigma, equalização e decodificação são realizadas conjuntamente através de um procedimento iterativo. Com isso, o equalizador pode se beneficiar da capacidade de correção de erros do código para remover a IIS de uma maneira mais efetiva, obtendo taxas de erro de bit bem inferiores àquelas alcançadas pelos receptores não-iterativos usuais. Os principais equalizadores de complexidade exponencial foram apresentados, bem como alguns equalizadores de complexidade reduzida. Por fim, citamos algumas outras aplicações recentes dos equalizadores turbo em sistemas de comunicações digitais.

AGRADECIMENTOS

Os autores gostariam de agradecer a Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP - pelo apoio financeiro.

REFERÊNCIAS

- [1] E. A. Lee e D. G. Messerschmit, *Digital Communication*, 2^a ed. Kluwer Academic Press, 1994.
- [2] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, e A. Glavieux, "Iterative Correction of Intersymbol Interference: Turbo-Equalization," *European Transactions on Telecommunications*, vol. 6, n^o. 5, p. 507–511, Setembro-Outubro 1995.
- [3] C. Laot, A. Glavieux e J. Labat, "Turbo Equalization: Adaptive Equalization and Channel Decoding Jointly Optimized," *IEEE Journal on Selected Areas in Communications*, vol. 19, n^o. 9, p. 1744–1752, Setembro 2001.
- [4] M. Tüchler, R. Koetter e A. C. Singer, "Turbo Equalization: Principles and New Results," *IEEE Transactions on Communications*, vol. 50, n^o. 5, p. 754–766, Maio 2002.
- [5] L. R. Bahl, J. Cocke, F. Jelinek, e J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, vol. 20, n^o. 2, p. 284–287, Março 1974.
- [6] M. C. Valenti, "Iterative Detection and Decoding for Wireless Communications," Tese de Doutorado, Virginia State University (EUA), Julho 1999.
- [7] W. E. Ryan, "A Turbo Code Tutorial," acessado em junho 2003, <http://www.ece.arizona.edu/ryan/turbo2c.pdf>.
- [8] P. Robertson, E. Villebrun, e P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," *Proceedings IEEE International Conference on Communications. ICC'95*, Seattle, Estados Unidos, p. 1009–1013, Junho 1995.
- [9] J. P. Woodard e L. Hanzo, "Comparative Study of Turbo Decoding Techniques: An Overview," *IEEE Transactions on Vehicular Technology*, vol. 49, n^o. 6, p. 2208–2233, Novembro 2000.
- [10] J. Hagenauer e P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," *Proceedings IEEE Global Telecommunications Conference. GLOBECOM'89*, vol. 3, p. 1680–1686, Novembro 1989.
- [11] V. Franz e J. B. Anderson, "Concatenated Decoding with a Reduced Search BCJR Algorithm," *IEEE Journal on Selected Areas on Communications*, vol. 16, n^o. 2, p. 186–195, Fevereiro 1998.
- [12] R. R. Lopes e J. R. Barry, "Soft-Output Decision-Feedback Equalization with a Priori Information," *Proceedings IEEE Global Telecommunications Conference, GLOBECOM'03*, vol. 3, p. 1705–1709, Dezembro 2003.
- [13] R. R. Lopes, "Iterative Estimation, Equalization and Decoding," Tese de Doutorado, Georgia Institute of Technology (EUA), Atlanta, Julho 2003.
- [14] E. Chesnutt, R. R. Lopes e J. R. Barry, "Beyond PRML: Linear-Complexity Turbo Equalization Using the Soft-Feedback Equalizer," *Proceedings IEEE International Magnetics Conference*, 2005.
- [15] X. Wang e R. Chen, "Blind Turbo Equalization in Gaussian and Impulsive Noise," *IEEE Transactions on Vehicular Technology*, vol. 50, n^o. 4, p. 1092–1105, Julho 2001.
- [16] M. B. Loiola, "Técnicas Autodidatas e Soluções de Baixa Complexidade para Equalização e Estimação Turbo," Dissertação de Mestrado, FEEC - UNICAMP, Campinas, Fevereiro 2005.