

# Introdução à Equalização Turbo♦

André Fonseca dos Santos e Dayan Adionel Guimarães

Departamento de Telecomunicações – Inatel - andref@inatel.br, dayan@inatel.br

**Resumo** – Sabe-se que a interferência intersimbólica (IIS) é uma das causas principais de degradação no desempenho de sistemas de comunicação digital que operam em canais dispersivos no tempo. Sabe-se também que a equalização é a forma usual para combater os efeitos dessa IIS. Neste contexto, este tutorial aborda os principais conceitos associados ao processo conjunto de equalização e decodificação iterativa, denominado Equalização Turbo. Algumas configurações utilizadas na implementação desse processo são apresentadas, tecendo-se conclusões sobre suas vantagens e desvantagens em termos de complexidade e capacidade de redução da IIS.

**Palavras chave** – equalização turbo, decodificação iterativa (turbo), interferência intersimbólica.

**Abstract** – It is well known that inter-symbol interference (ISI) is one of the main reasons for performance degradation in digital communication systems operating in time-dispersive channels. It is also known that equalization is the usual method for combating the effects of ISI. In this context, this tutorial aims at presenting the main concepts associated to the joint process of iterative decoding and iterative equalization, called Turbo Equalization. It presents some configurations for a turbo equalization process and derives some conclusions concerning the trade-off between complexity and ISI reduction capability for these configurations.

**Key words** – turbo equalization, turbo (iterative) decoding, inter-symbol interference.

## I. INTRODUÇÃO

Devido à crescente demanda de serviços em sistemas de comunicação móvel terrestres, tem-se investido grande esforço na busca de sistemas de transmissão de alta capacidade que operam adequadamente em canais com múltiplos percursos de propagação.

Nos sistemas de comunicação móvel, tipicamente utilizam-se códigos corretores de erro para minimizar a influência do ruído AWGN (*Additive White Gaussian Noise*) e equalizadores para reduzir o efeito da interferência intersimbólica (IIS) causada pela dispersão temporal do canal. Essa dispersão pode ser causada, por exemplo, quando o canal é limitado em largura de faixa ou quando há propagação por múltiplos percursos [1].

No entanto, os processos de equalização e de decodificação são realizados normalmente de forma independente. O propósito da equalização turbo é realizar a equalização

e a decodificação de maneira conjunta e iterativa. Embora em [1] já se considere a equalização e a decodificação de forma conjunta, na equalização turbo o processo é iterativo.

O primeiro esquema de equalização turbo foi proposto em [2]. Nele, os símbolos transmitidos são codificados através de um codificador convolucional, que é tratado como um codificador externo, e o conjunto formado pelo modulador, pelo canal de comunicação e pelo demodulador é modelado como um codificador interno [2]. Na recepção, o processo de equalização turbo realiza sucessivas trocas de informação entre o decodificador interno (equalizador) e o decodificador externo (decodificador convolucional), assim como ocorre na decodificação turbo entre os decodificadores componentes [3].

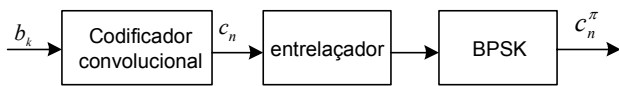
Os resultados obtidos com esse sistema mostram que a equalização turbo pode eliminar praticamente toda a influência da IIS do canal, sem a necessidade de aumento da relação sinal-ruído (RSR). Seu desempenho chega a se assemelhar ao desempenho de um sistema com codificação de canal em um canal AWGN e sem IIS [4].

Este tutorial tem como principal objetivo a apresentação dos conceitos envolvidos na equalização turbo. A Seção II apresenta o modelo discreto (no tempo) do conjunto formado pelo modulador, pelo canal com múltiplos percursos e pelo demodulador. A partir deste modelo discreto pode-se observar que um símbolo transmitido pode se sobrepor aos símbolos vizinhos, resultando em IIS. Nota-se também que este modelo discreto se comporta de forma semelhante a um codificador convolucional [2], o que permite sua caracterização como um codificador interno do sistema. Para fins de simplificação, no texto deste tutorial este modelo constituído por modulador, canal e demodulador será citado apenas como *modelo discreto do canal*. Na Seção III é apresentado o princípio de funcionamento de um equalizador turbo, no qual as iterações são apresentadas como uma cascata de módulos [2] [5] [6]. A Seção IV trata do algoritmo MAP (*Maximum a Posteriori*) símbolo-a-símbolo e sua implementação nos dois blocos do equalizador turbo: o bloco de equalização e o bloco de decodificação [4]. Na Seção V utiliza-se outra configuração do equalizador turbo, conforme apresentado em [5] e [7], onde é usado um *Cancelador de Interferência* no bloco equalizador e o algoritmo MAP no bloco de decodificação. A Seção VI faz uma referência aos critérios de parada apresentados em [12], os quais determinam o número ótimo de iterações que devem ser realizadas para que o sistema proporcione um desempenho alvo. Finalmente, na Seção VII faz-se uma análise de resultados obtidos por simulação e reportados em publicações recentes sobre o assunto, tecendo-se comentários finais sobre quesitos como complexidade e desempenho.

♦ Este trabalho foi parcialmente financiado pelo convênio nº 22.02.0431.00, celebrado entre o Inatel, a *Linear Equipamentos Eletrônicos S/A* e a *FINEP*.

**II. MODELO DISCRETO DO CANAL**

Ao longo deste tutorial o sistema de transmissão utilizado como referência é aquele apresentado na Figura 1. Primeiramente, os bits de informação  $b_k$  são codificados através de um codificador convolucional em símbolos  $c_n$ . Os símbolos  $c_n$  são embaralhados por um entrelaçador temporal (*interleaver*), usado para que erros em rajada se apresentem descorrelacionados no receptor, após a passagem pelo canal e pelo bloco de desentrelaçamento temporal (*de-interleaver*) desse receptor. Os símbolos entrelaçados são mapeados por um modulador BPSK (*Binary Phase Shift Keying*) em banda base, resultando nos símbolos  $c_n^\pi$ , que são transmitidos através do canal.



**Figura 1** - Sistema de transmissão.

Em um canal onde há propagação por múltiplos percursos, o sinal recebido é resultado da soma de infinitas réplicas (ou ecos) do sinal transmitido [8, p. 532]. Os múltiplos percursos são causados principalmente por difração, reflexão e espalhamento do sinal nos obstáculos que circundam o receptor e/ou que se localizam entre transmissor e receptor, e também por influência da atmosfera. Como é inviável modelar esse canal com infinitos percursos de propagação, principalmente por conta da dificuldade de análise matemática, pode-se admitir como aproximação que ele envolva a soma dos  $L$  percursos mais significativos. Assim, o modelo discreto de um canal com múltiplos percursos pode ser representado por uma linha de atrasos com derivações (do Inglês, *tapped delay line*) [1], como descrito pela expressão:

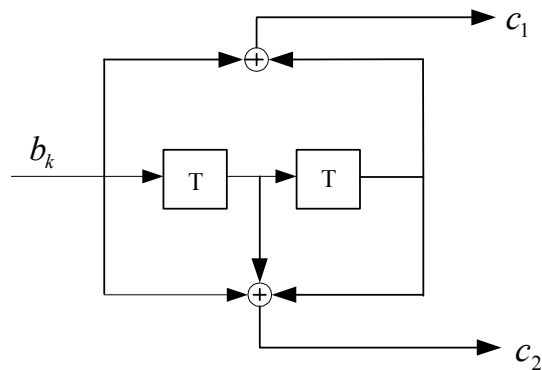
$$r_n = w_n + \sum_{l=0}^L g_l c_{n-l}^\pi = w_n + v_n \tag{1}$$

onde os coeficientes  $g_1, g_2, \dots, g_l, \dots, g_L$  representam as variações de intensidade do sinal no  $l$ -ésimo percurso de propagação, também podendo ser denominados de ganhos do canal.  $v_n$  é a  $n$ -ésima amostra de saída do canal com IIS sem a presença de ruído AWGN,  $w_n$  é a  $n$ -ésima amostra de ruído AWGN adicionado ao sinal e  $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_n \ \dots \ r_N]^T$  é o vetor que contém as amostras do sinal recebido. O número de amostras  $N$ , no vetor  $\mathbf{r}$ , é função do número de amostras operadas em cada bloco no receptor, como pode ser observado mais adiante.

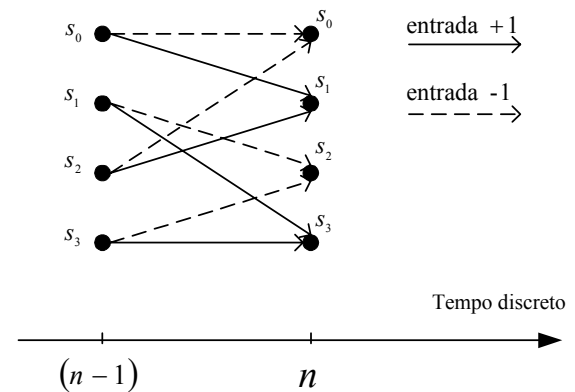
O modelo discreto do canal conforme (1) pode ser representado por uma treliça associada a uma cadeia de Markov, da mesma maneira que o codificador convolucional da Figura 2 [2]. Como exemplo, a treliça da Figura 3 representa a transição de estados de um codificador convolucional como o da Figura 2, ou de um canal com dois múltiplos percursos (canal com IIS).

Se o canal ou o codificador é composto por  $M$  atrasadores, as correspondentes treliças serão compostas por  $2^M$  estados. No exemplo em questão  $M = 2$  para o canal e para o codificador, o que resulta em uma treliça de quatro estados. Para exemplificar a formação da treliça toma-se a

transição de  $S_0$  para  $S_1$ . O estado  $S_0$  é composto pelos símbolos  $\{-1,-1\}$  na saída dos registradores da linha de atraso. Ao transmitir o símbolo  $+1$  pelo canal, o conteúdo na saída dos registradores passa a ser  $\{+1,-1\}$ , que corresponde ao estado  $S_1$ . Da mesma maneira são formadas as demais transições da treliça.



**Figura 2** - Codificador convolucional de taxa  $\frac{1}{2}$  com polinômios geradores  $g_1(D)=1+D$  e  $g_2(D)=1+D+D^2$ .



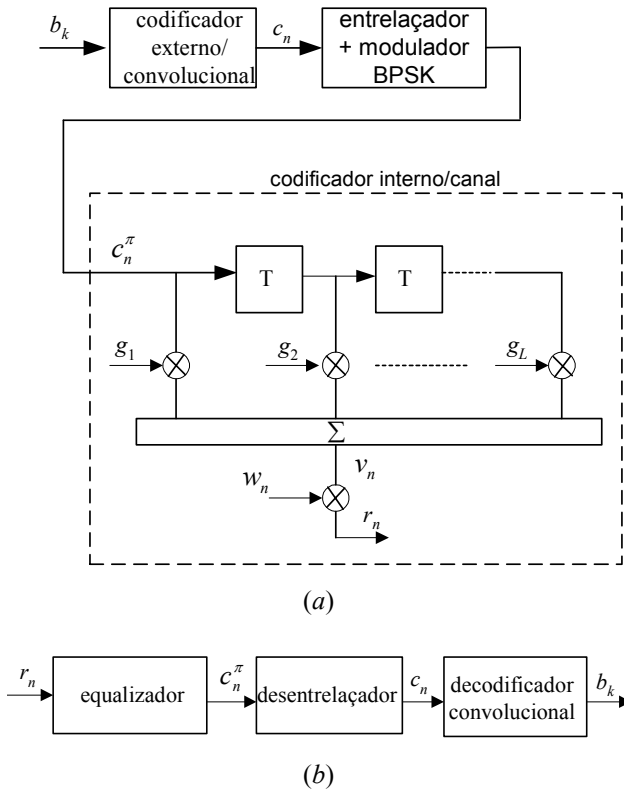
**Figura 3** - Treliça que representa as transições de estados para um codificador convolucional de comprimento 3 e de um canal com dois múltiplos percursos, onde  $S_0 = \{-1,-1\}$ ,  $S_1 = \{+1,-1\}$ ,  $S_2 = \{-1,+1\}$  e  $S_3 = \{+1,+1\}$ . Para o codificador o valor  $-1$  é trocado por 0 nos valores dos estados.

Partindo desse princípio de que o canal com IIS pode ser representado como uma treliça da mesma maneira que um codificador convolucional, para o entendimento do equalizador turbo o canal será tratado como um codificador componente do sistema de transmissão, num processo de concatenação serial, o que permitirá que a equalização e a decodificação sejam realizadas de maneira iterativa.

**III. PRINCÍPIOS DA EQUALIZAÇÃO TURBO**

A Figura 4(a) apresenta o canal como um codificador interno do sistema de transmissão e o codificador convolucional como um codificador externo. Nos sistemas convencionais de recepção a recuperação da informação é feita de acordo com a Figura 4(b). Primeiramente, é realizada a equalização do canal através de, por exemplo, um filtro empregando o algoritmo LMS (*Least Mean Square*) ou um equalizador de treliça empregando o critério MLSE (*Maximum Likelihood Sequence Estimation*) [1]. Em

seguida, os símbolos codificados  $c_n^\pi$  na saída do equalizador são reordenados pelo desentrelaçador temporal que fornece os símbolos  $c_n$  para o decodificador convolucional. Finalmente, na saída do decodificador os bits de informação  $b_k$  são estimados.



**Figura 4** - (a) Sistema de transmissão onde o canal é modelado como codificador interno. (b) Sistema de recepção convencional com decisões tomadas no equalizador e no decodificador separadamente.

A equalização e a decodificação são realizadas separadamente e o equalizador não explora a estrutura do código para realizar seu processamento. Na equalização turbo realiza-se, iterativamente, uma realimentação de informação das decisões do decodificador para o equalizador, de tal maneira que este produza decisões mais confiáveis. De maneira similar, as decisões do equalizador alimentam o decodificador, permitindo que este forneça decisões mais confiáveis. Dessa forma, o processo conjunto de equalização e decodificação turbo atingirá melhor desempenho, iteração a iteração.

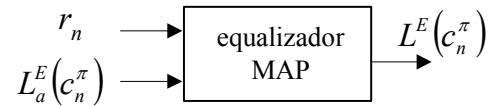
Para que as informações trocadas entre o bloco equalizador e o bloco decodificador carreguem dados de confiabilidade nas decisões, elas devem ser suaves. Por esta razão o equalizador turbo é constituído de dois blocos de entradas e saídas suaves (do Inglês, *Soft-Input Soft-Output*, SISO). O primeiro é o bloco equalizador apresentado na Figura 5. Este bloco é um equalizador que tem sua saída na forma de uma razão de verossimilhança logarítmica (do Inglês, *Log-Likelihood Ratio*, LLR) das probabilidades *a posteriori* dos símbolos transmitidos. Tal LLR vale:

$$L^E(c_n^\pi) = \ln \frac{P(c_n^\pi = +1 | \mathbf{r})}{P(c_n^\pi = -1 | \mathbf{r})} \quad (2)$$

onde  $c_n^\pi$  são os símbolos da modulação BPSK em banda-base e  $\mathbf{r}$  é o vetor de amostras recebidas do canal com IIS. A magnitude de  $L^E(c_n^\pi)$  contém a informação da confiabilidade da decisão tomada pelo equalizador. Pode-se também, a partir do sinal de  $L^E(c_n^\pi)$ , tomar uma decisão abrupta sobre os símbolos  $c_n^\pi$  da seguinte maneira:

$$\begin{aligned} \text{Se } L^E(c_n^\pi) \geq 0, & \text{ então } c_n^\pi = +1 \\ \text{Se } L^E(c_n^\pi) < 0, & \text{ então } c_n^\pi = -1 \end{aligned}$$

O critério de decisão supracitado é conhecido como critério MAP (*Maximum a Posteriori*) símbolo-a-símbolo. Este critério minimiza a probabilidade de decisão errada por cada símbolo.



**Figura 5** - Equalizador MAP.

Para o cálculo da saída suave  $L^E(c_n^\pi)$ , o bloco equalizador tem como entrada o vetor de amostras recebidas do canal, com IIS,  $\mathbf{r}$ , e a LLR da informação *a priori* dos símbolos  $c_n^\pi$  dada por:

$$L_a^E(c_n^\pi) = \ln \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \quad (3)$$

Aplicando a regra de Bayes em (2), a saída suave do bloco equalizador pode ser decomposta em duas parcelas:

$$\begin{aligned} L^E(c_n^\pi) &= \ln \frac{P(c_n^\pi = +1 | \mathbf{r})}{P(c_n^\pi = -1 | \mathbf{r})} \\ &= \ln \left[ \frac{p(\mathbf{r} | c_n^\pi = +1)}{p(\mathbf{r} | c_n^\pi = -1)} \right] + \ln \left[ \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \right] \quad (4) \end{aligned}$$

A primeira parcela em (4) refere-se à *informação extrínseca* na saída do bloco equalizador, que será representada daqui em diante por  $L_{ext}^E(c_n^\pi)$ . A segunda parcela se refere à LLR das probabilidades *a priori* dos símbolos  $c_n^\pi$ , que será representada deste ponto em diante por  $L_a^E(c_n^\pi)$ . A informação extrínseca pode ser interpretada como o ganho de informação obtido sobre um símbolo no processo de equalização ou decodificação [2] ou como a LLR na saída de um decodificador SISO menos a LLR da entrada [8].

O outro bloco SISO do equalizador turbo é o decodificador mostrado na Figura 6. Este bloco é constituído de uma entrada que recebe o vetor  $\mathbf{Z}$ , que tem como elementos uma estimativa das probabilidades *a priori* dos símbolos  $c_n$ :

$$\mathbf{Z} = [P(c_1 | \mathbf{r}) P(c_2 | \mathbf{r}) \dots P(c_N | \mathbf{r})] \quad (5)$$

A partir do vetor de entrada  $\mathbf{Z}$ , o bloco decodificador calcula as LLR's dos símbolos  $c_n$  e dos bits de informação  $b_k$ .



Figura 6 - Bloco decodificador MAP.

Da mesma maneira que no bloco equalizador, a saída  $L^D(c_n)$  do bloco decodificador pode ser decomposta nas LLR's da informação extrínseca e da informação *a priori*:

$$L^D(c_n) = \ln \left( \frac{P(c_n = +1 | \mathbf{Z})}{P(c_n = -1 | \mathbf{Z})} \right) = \ln \left( \frac{p(\mathbf{Z} | c_n = +1)}{p(\mathbf{Z} | c_n = -1)} \right) + \ln \left( \frac{P(c_n = +1)}{P(c_n = -1)} \right) \quad (6)$$

onde o primeiro termo de (6) é a informação extrínseca  $L_{ext}^D(c_n)$  do símbolo  $c_n$  e o segundo termo é a informação *a priori* desse símbolo,  $L_a^D(c_n)$ .

Na equalização turbo utiliza-se a informação extrínseca  $L_{ext}^D(c_n)$  obtida pelo bloco decodificador como uma estimativa da informação *a priori* a ser operada pelo bloco equalizador. De forma análoga, a informação extrínseca obtida pelo bloco equalizador é utilizada como informação *a priori* a ser processada pelo bloco decodificador. Portanto, um equalizador turbo opera como um decodificador turbo, onde um código componente é o codificador convolucional e outro código componente pode interpretado como o canal com múltiplos percursos.

A Figura 7 ilustra o processo de equalização turbo. Através do vetor de amostras recebidas do canal,  $\mathbf{r}$ , e da informação *a priori*  $L_a^E(c_n^\pi)$  o bloco equalizador calcula sua saída suave  $L^E(c_n^\pi)$ . De acordo com (4), subtrai-se da saída suave total do bloco equalizador  $L^E(c_n^\pi)$  a informação *a priori*  $L_a^E(c_n^\pi)$  para que se transmita somente a informação extrínseca  $L_{ext}^E(c_n^\pi)$  para o bloco decodificador. A informação extrínseca  $L_{ext}^E(c_n^\pi)$  passa pelo desentrelaçador temporal e é utilizada como uma estimativa da informação *a priori* dos símbolos,  $L_a^D(c_n)$ , para o bloco decodificador. O bloco decodificador calcula sua saída suave  $L^D(c_n)$  e subtrai a parcela da informação *a priori*  $L_a^D(c_n)$  para o cálculo da informação extrínseca  $L_{ext}^D(c_n)$ , de acordo com (6). A informação extrínseca  $L_{ext}^D(c_n)$  passa pelo entrelaçador temporal e é usada como uma estimativa da informação *a priori* dos símbolos,  $L_a^E(c_n^\pi)$ , para o bloco equalizador. Note que um dos papéis do entrelaçador e do desentrelaçador no equalizador turbo é de organizar as LLR's *a priori* na entrada do bloco equalizador e do bloco decodificador respectivamente. Outra função dos processos de entrelaçamento/desentrelaçamento é a de decorrelacionar as informações trocadas pelos blocos constituintes, o que será explicado com mais detalhes no decorrer deste tutorial.

O processo descrito no parágrafo anterior se repete tantas vezes quanto necessário, até que se atinja a convergência em termos de estabilização dos valores de infor-

mação extrínseca, o que leva a uma estabilização no desempenho final da equalização turbo em termos de taxa de erro de bit ou de bloco. Pode-se também interromper o processo iterativo utilizando outros critérios, como exemplificado mais adiante, objetivando uma redução no número médio de iterações e, por conseqüência, um aumento na vazão de informação.

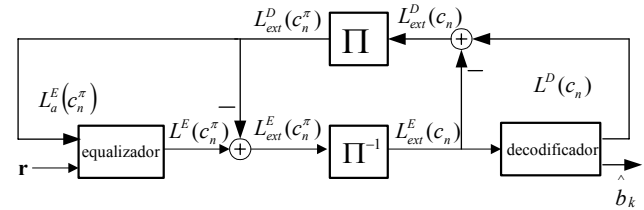


Figura 7 - Equalizador turbo.

A cada repetição (iteração) há um melhoramento da confiabilidade das informações extrínsecas dos blocos equalizador e decodificador, diminuindo a chance de erros nas decisões sobre os bits  $b_k$  ou sobre os símbolos  $c_n$  [11]. Na primeira iteração, como não há informação extrínseca disponível vinda do decodificador, a LLR *a priori* operada pelo bloco equalizador,  $L_a^E(c_n^\pi)$ , é feita igual a zero (probabilidades *a priori* admitidas iguais a 1/2) e a cada iteração a estimativa desta LLR é melhorada. Ao final das iterações pode-se realizar uma decisão abrupta sobre os bits de informação  $b_k$  ou sobre os símbolos  $c_n$ , a partir da saída suave do bloco decodificador.

O uso da informação extrínseca como estimativa da informação *a priori* dos blocos constituintes do processo de equalização turbo deve garantir que ela seja decorrelacionada em relação aos valores de informação extrínseca calculados pelo bloco anterior. Isto é feito para que um bloco não opere com valores que são resultado de decisões suaves geradas pelo próprio bloco constituinte em questão. Por exemplo, de acordo com (4), a saída suave  $L^E(c_n^\pi)$  do equalizador é composta pela informação *a priori*  $L_a^E(c_n^\pi)$  e pela informação extrínseca  $L_{ext}^E(c_n^\pi)$ . Portanto, a saída suave total  $L^E(c_n^\pi)$  do bloco equalizador não deve ser usada como estimativa da informação *a priori* para o bloco decodificador, já que a parcela  $L_a^E(c_n^\pi)$  que compõe a saída  $L^E(c_n^\pi)$  é obtida a partir da saída do bloco decodificador. Dessa forma, a saída suave  $L^E(c_n^\pi)$  do bloco equalizador torna-se correlacionada com as LLR's fornecidas pelo decodificador. Do mesmo modo, a saída  $L^D(c_n)$  do bloco decodificador apresenta certa correlação com as LLR's de saída obtidas anteriormente pelo bloco equalizador. Para evitar esta redundância é que se utiliza apenas a informação extrínseca como estimativa da entrada *a priori* do bloco seguinte.

Para que a correlação remanescente entre os valores de informação extrínseca operados pelos blocos equalizador e decodificador seja minimizada, um adequado projeto do entrelaçador temporal utilizado faz-se necessário, pois este elemento tem como papel principal decorrelacionar tais valores, além de minimizar os efeitos dos erros em rajada causados por canais com memória. Desta maneira, quanto maior é a capacidade do entrelaçador de tornar aleatórios os valores por ele operados, maior é o ganho que se tem em termos de melhoria da taxa de erro

de bit a cada iteração do equalizador turbo [4].

Assim como qualquer processo iterativo, o processo de equalização turbo também pode ser representado como uma linha de módulos (do Inglês, *pipe-line*), como ilustrado na Figura 8, onde cada módulo representa uma iteração do equalizador. Cada módulo possui duas entradas: o vetor de amostras recebidas do canal,  $\mathbf{r}$ , e a informação *a priori* do bloco equalizador,  $L_a^E(c_n)$ . Na saída de um módulo tem-se a informação extrínseca do bloco decodificador, que é utilizada como estimativa da informação *a priori* para o próximo módulo (próxima iteração). A saída de cada módulo possui maior confiabilidade em relação à saída do módulo anterior [9] [10]. Ao final das iterações, a partir da saída suave do último módulo é realizada uma decisão abrupta em relação aos bits de informação  $b_k$  ou em relação aos símbolos  $c_n$ . A representação em módulos foi inserida neste texto com o objetivo de facilitar o entendimento do processo de equalização turbo.

Neste ponto merecem ser comentadas algumas questões relacionadas ao tempo que o equalizador turbo precisa para obter convergência. Duas medidas de tempo podem ser aqui consideradas: a *latência* e o *atraso*. O atraso é inversamente proporcional à taxa de transmissão e diretamente proporcional ao tamanho do bloco codificado, pois de forma independente do tipo de decodificação, há que se esperar pela recepção de todos os símbolos associados a uma palavra-código para que a estimação dessa palavra-código se inicie<sup>1</sup>. Já a latência está associada ao tempo de processamento necessário para que, depois recebida uma palavra-código completa, o primeiro bit estimado válido de informação esteja disponível na saída do equalizador turbo. Este tempo depende do número de iterações (número de módulos) no processo de equalização turbo e também da complexidade dos algoritmos utilizados nos blocos componentes (bloco de equalização e bloco de decodificação). Nesse sentido, um dos grandes desafios apresentados ao desenvolvimento de novos processos de equalização turbo está relacionado com o objetivo de redução da latência, assim como ocorre com os códigos turbo.

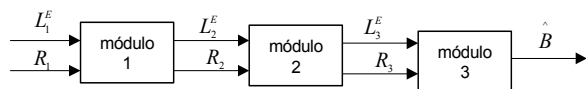


Figura 8 - Equalizador turbo composto por módulos.

Os primeiros sistemas de equalização turbo tratados em [2], [11] e [13] são constituídos de blocos equalizador e decodificador que utilizam o algoritmo ótimo de decodificação MAP, chamado BCJR [14] e que possui uma complexidade computacional exponencialmente crescente com o número de estados das treliças que representam o canal e o codificador. Outras alternativas que simplificam o algoritmo BCJR, como o algoritmo *Log-MAP* que opera no domínio logarítmico, sem perda de desempenho em relação ao BCJR, e o algoritmo sub-ótimo *Max-Log-MAP*

<sup>1</sup> Esta afirmação é válida para códigos de bloco e para outros esquemas de codificação de canal que operem com blocos de símbolos codificados, por exemplo, quando se utiliza um código convolucional com terminação na sua treliça de forma que ao final da seqüência de informação de comprimento definido o estado do codificador retorne ao estado nulo.

[12]. Ainda no intuito de simplificar o algoritmo BCJR, podem-se utilizar técnicas de redução de estados da treliça que representa o canal [15]. Em [5] é utilizada uma alternativa sub-ótima através de um *Cancelador de Interferências* no bloco de equalização. Esta alternativa apresenta significativa redução de complexidade de implementação em relação ao algoritmo BCJR. Outra alternativa para o bloco equalizador é apresentada em [16], onde é usado no equalizador um filtro que procura minimizar o erro médio quadrático (*MMSE - Minimum Mean Square Error*), entre o sinal transmitido e o estimado.

Nas próximas seções, optou-se por utilizar apenas o algoritmo BCJR e o Cancelador de Interferências como componentes do processo de equalização turbo. Formas alternativas podem ser consultadas em [12], [16] e [15], apenas para citar alguns exemplos.

#### IV. ALGORITMO BCJR

O algoritmo BCJR, que implementa o critério de decisão MAP símbolo-a-símbolo, foi proposto em 1974 por Bahl, Cocke, Jelinek e Raviv [14]. Para o entendimento do algoritmo tomam-se os símbolos<sup>2</sup>  $b_k$  que passarão por um sistema que é modelado por uma treliça como a da Figura 9. Esta treliça pode representar um modelo de Markov, como um canal com múltiplos percursos ou um codificador convolucional. O algoritmo BCJR tem como objetivo calcular a LLR

$$L(b_k) \equiv \ln \left( \frac{P(b_k = +1 | \mathbf{r})}{P(b_k = -1 | \mathbf{r})} \right) \quad (7)$$

Aplicando o teorema de Bayes  $P(a, b) = P(a|b)P(b)$  em (7), obtém-se:

$$L(b_k | \mathbf{r}) = \ln \left( \frac{p(b_k = +1, \mathbf{r})}{p(b_k = -1, \mathbf{r})} \right) \quad (8)$$

Por meio da Figura 9 pode-se fazer uma interpretação sobre os valores operados na expressão (8). A probabilidade de  $b_k$  ser +1 é a probabilidade de ocorrer as transições representadas pelas linhas cheias na Figura 9, enquanto a probabilidade de  $b_k$  ser -1 é a probabilidade de ocorrer as transições representadas pelas linhas tracejadas.

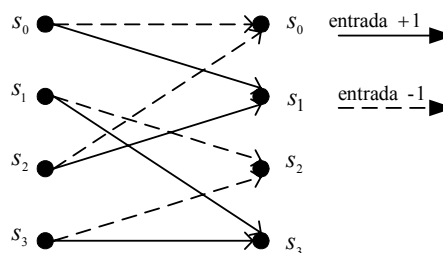


Figura 9 - Modelo de Markov representado por uma treliça.

Como o conjunto das probabilidades de transições é mutuamente exclusivo, a equação (8) pode ser expressa da seguinte maneira [17]:

<sup>2</sup> Deve-se atentar para a re-utilização da notação  $b_k$ , agora de forma mais abrangente, podendo estar associada a bits de informação ou a símbolos ao longo do sistema de transmissão e de recepção, dependendo de onde e como se aplica o algoritmo BCJR.

$$L(b_k) = \ln \left( \frac{\sum_{b_k=+1} p(S_{k-1}, S_k, \mathbf{r})}{\sum_{b_k=-1} p(S_{k-1}, S_k, \mathbf{r})} \right) \quad (9)$$

onde  $S_k$  refere-se ao estado que a treliça se encontra no instante de tempo presente e  $S_{k-1}$  se refere ao estado que a treliça se encontrava no instante de tempo anterior. Note que o numerador na parte da direita de (9) representa as probabilidades das transições causadas pelo símbolo +1 (linhas contínuas da Figura 9), enquanto que o denominador se refere aos símbolos -1.

Ainda referindo-se à expressão (9) pode-se mostrar que [17]:

$$p(S_{k-1}, S_k, R) = p(r_{j>k} | s) p(\{r_k, s\} | s') p(s', r_{j<k}) \quad (10)$$

que, numa notação alternativa, pode ser re-escrita como:

$$p(S_{k-1}, S_k, R) = \beta_k(S_k) \gamma_k(S_{k-1}, S_k) \alpha_{k-1}(S_{k-1}) \quad (11)$$

onde  $s$  se refere ao estado que a treliça se encontra no instante de tempo presente e  $s'$  ao instante de tempo anterior. O termo  $\alpha_{k-1}$  em (11) se refere à função densidade da probabilidade que a treliça estava no estado anterior  $s'$  e recebeu uma sequência passada  $r_{j<k}$ . O valor de  $\alpha$  pode ser calculado através da equação [17]:

$$\alpha_k(S_k) = \sum_{S_{k-1}} \gamma_k(S_{k-1}, S_k) \alpha_{k-1}(S_{k-1}) \quad (12)$$

O termo  $\beta_k$  em (11) se refere à função densidade da probabilidade de receber uma sequência futura  $r_{j>k}$ , dado o estado atual  $s$  da treliça. O valor de  $\beta_k$  pode ser calculado através de [17]:

$$\beta_k(S_k) = \sum_{S_{k+1}} \gamma_k(S_{k-1}, S_k) \beta_{k+1}(S_{k+1}) \quad (13)$$

A Figura 10 exemplifica o cálculo de  $\alpha$  e  $\beta$ , onde se deseja calcular os valores de  $\alpha_k$  e  $\beta_k$  do estado 0. A figura apresenta as possíveis transições que terminam e partem do estado 0 e os respectivos cálculos de  $\alpha_k$  e  $\beta_k$  a partir das equações (12) e (13). Os valores de  $\alpha$  são obtidos partindo do início para o fim da treliça de acordo com (12). Os valores de  $\beta$  são obtidos partindo do fim para o início da treliça através de (13).

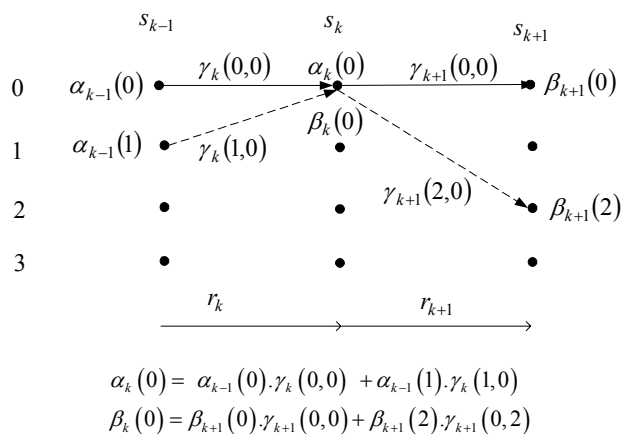


Figura 10 - Cálculo de  $\alpha$  e  $\beta$  [17].

Caso seja forçado, através de transmissão de zeros, que a treliça inicie e termine no estado  $S_0$ ,  $\alpha$  e  $\beta$  podem ser inicializados como as expressões [12]:

$$\begin{aligned} \alpha_{inicial}(0) &= 1 \text{ e } \alpha_{inicial}(s) = 0 \quad \forall s \neq 0 \\ \beta_{final}(0) &= 1 \text{ e } \beta_{final}(s) = 0 \quad \forall s \neq 0 \end{aligned} \quad (14)$$

Quando o estado inicial e o final não forem conhecidos,  $\alpha_{inicial}$  e  $\beta_{final}$  são inicializados com 1 para todos os estados [4]. De posse dos valores de  $\alpha$  e  $\beta$ , resta calcular os valores de  $\gamma$  para que se obtenha o valor estimado por (11), para todas as transições possíveis, e resta também determinar  $L(b_k)$  através de (9).

Um receptor ideal seria aquele que utilizasse o algoritmo BCJR em uma supertreliça que englobaria o codificador, o entrelaçador e o canal [4]. No entanto, a modelagem dessa treliça seria extremamente complexo, assim como o uso do algoritmo BCJR. O equalizador turbo trata de maneira separada a treliça do canal e a treliça do decodificador. Portanto, o algoritmo BCJR pode ser aplicado ao equalizador e ao decodificador. Para ambos os casos, o cálculo de  $\alpha$  e  $\beta$  é executado da mesma maneira. O que irá diferenciar o emprego do algoritmo BCJR nos blocos equalizador e decodificador é a maneira de calcular o valor de  $\gamma$ . Estas observações são mais bem exploradas adiante.

#### IV.1 ALGORITMO BCJR APLICADO AO BLOCO EQUALIZADOR

Nesta seção é apresentado o algoritmo BCJR aplicado ao bloco equalizador. Os valores de  $\alpha$  e  $\beta$  são calculados a partir das equações (12) e (13), respectivamente. Para os cálculos envolvendo (12) e (13) é necessário calcular o valor de  $\gamma$ . De (10) e (11) tem-se:

$$\gamma_n(s', s) = p(\{r_n, S_n = s\} | S_{n-1} = s') \quad (15)$$

Aplicando a regra de Bayes modificada  $P[(a,b)|c] = P[a|(b,c)]P(b|c)$  em (15), tem-se:

$$\gamma_n(s', s) = p(r_n | \{s, s'\}) P(s | s') \quad (16)$$

O primeiro termo do lado direito de (16),  $p(r_n | \{s, s'\})$ , é a função densidade de probabilidade de recepção de uma amostra  $r_n$  do canal, dado que ocorreu uma transição  $\{s, s'\}$ . A partir do modelo discreto do canal representado por (1), observa-se que esta função densidade de probabilidade é dada por uma distribuição Gaussiana com média  $v_n$ , que é o sinal com IIS sem adição de ruído, e com variância  $\sigma^2$  devido à adição do ruído AWGN  $w_n$ . Portanto, este termo é dado por:

$$p(r_n | v_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - v_n)^2}{2\sigma^2}\right) \quad (17)$$

O segundo termo do lado direito de (16),  $P(s | s')$ , é a probabilidade da treliça se encontrar no estado  $s$ , dado que ela estava no estado  $s'$ . Esta probabilidade é governada pela probabilidade *a priori* do símbolo de entrada  $c_n^\pi$ , já que é este símbolo que irá definir qual transição ocorrerá na treliça do canal. A partir desses resultados, a expressão (16) pode ser re-escrita como:

$$\gamma(s', s) = P(c_n^\pi) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - v_n)^2}{2\sigma^2}\right) \quad (18)$$

onde  $P(c_n^\pi)$  é a probabilidade *a priori* dos símbolos  $c_n^\pi$ , estimada através da informação extrínseca do decodificador,  $L_{ext}^D(c_n^\pi)$ . Como a informação extrínseca  $L_{ext}^D(c_n^\pi)$  é usada como estimativa da informação *a priori*  $P(c_n^\pi)$ , ela pode ser escrita como [2]:

$$L_{ext}^D(c_n^\pi) \approx \ln \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \quad (19)$$

A partir da equação (19) obtém-se a expressão da probabilidade *a priori*  $P(c_n^\pi)$  em função de  $L_{ext}^D(c_n^\pi)$ :

$$P(c_n^\pi = c) \cong \frac{\exp(c L_{ext}^D(c_n^\pi))}{1 + \exp(L_{ext}^D(c_n^\pi))} \quad (20)$$

onde  $c \in \{0, 1\}$  para o cálculo de  $P(c_n^\pi = +1)$  ou  $P(c_n^\pi = -1)$ , respectivamente. Aplicando (20) em (18) obtém-se finalmente o valor de  $\gamma$  para o bloco equalizador:

$$\gamma(s', s) = \frac{\exp(c L_{ext}^D(c_n^\pi))}{1 + \exp(L_{ext}^D(c_n^\pi))} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - v_n)^2}{2\sigma^2}\right) \quad (21)$$

A partir de (21) pode-se notar o ganho de informação no processo iterativo do equalizador turbo. Na primeira iteração não há informação extrínseca  $L_{ext}^D(c_n^\pi)$  do decodificador realimentada para o equalizador. Sendo assim, os símbolos  $c_n^\pi$  são admitidos equiprováveis e  $P(c_n^\pi)$  é considerada igual a  $1/2$ . Nas próximas iterações o valor de  $L_{ext}^D(c_n^\pi)$  é usado para o cálculo de  $\gamma$  conforme (21). A cada iteração há um aumento na confiabilidade da informação extrínseca do decodificador,  $L_{ext}^D(c_n^\pi)$ , e, conseqüentemente, o valor de  $\gamma$  é aprimorado.

A inicialização do algoritmo BCJR admite que os estados inicial e final da treliça sejam desconhecidos. Desta maneira, os valores de  $\alpha$  e  $\beta$  são inicializados com o valor 1. Em uma implementação prática, os valores de  $\alpha$  e  $\beta$  convergem rapidamente para zero, o que pode gerar algumas instabilidades numéricas no algoritmo. Em [18] é usada uma normalização de  $\alpha$  e  $\beta$  que elimina essas instabilidades.

Para se empregar o algoritmo BCJR é necessário que os coeficientes  $g_i$  do modelo discreto do canal sejam conhecidos. Em [19] é apresentado um algoritmo MAP que, além de equalizar o canal, também estima os coeficientes  $g_i$ . Como exemplo de aplicação do algoritmo BCJR no bloco equalizador, a Figura 11 apresenta a treliça de um canal com os coeficientes [0,407 ; 0,815 ; 0,407]. Os valores associados às transições dessa treliça são os valores de  $v_n$  utilizados na equação (21).

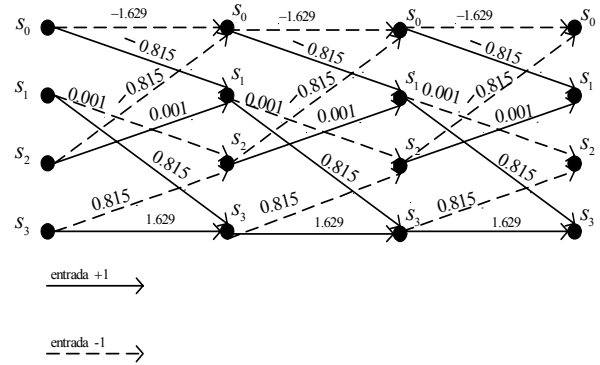
#### IV.2 ALGORITMO BCJR APLICADO AO BLOCO DECODIFICADOR

Nesta secção é apresentado o algoritmo BCJR aplicado ao bloco de decodificação. Os valores de  $\alpha$  e  $\beta$  são calculados a partir das equações (12) e (13), da mesma

maneira que para o bloco equalizador. Para o cálculo de  $\gamma$  utiliza-se agora a expressão:

$$\gamma_n(s', s) = p(z_n | \{s, s'\}) \cdot P(s | s') \quad (22)$$

onde  $z_n$  são os elementos do vetor  $\mathbf{Z}$  definido em (5). O primeiro termo do lado direito de (22) é governado pelos símbolos de saída  $c_n$  do codificador convolucional. O segundo termo é governado pelos bits de informação  $b_k$  na entrada do codificador convolucional.



**Figura 11** - Treliça para canal com os coeficientes  $g_1=0,407$ ,  $g_2=0,815$  e  $g_3=0,407$ . As linhas tracejadas representam transições causadas por entradas  $-1$  no canal e as linhas contínuas por entradas  $+1$ . De [4].

Como apresentado na Seção III, o bloco decodificador tem como entrada o vetor  $\mathbf{Z}$  constituído das probabilidades referentes aos símbolos  $c_n$ . Portanto, o valor de  $\gamma$  para o bloco decodificador é dado pela expressão:

$$\gamma(s_i, s_j) = P(c_1 | \mathbf{r}) P(c_2 | \mathbf{r}) P(b_k) \quad (23)$$

onde  $c_1$  e  $c_2$  são, respectivamente, o primeiro e o segundo símbolo de saída para as transições da treliça do codificador convolucional, como na Figura 2. O termo  $P(b_k)$  é a probabilidade *a priori* dos bits de informação. De início esta informação não é disponível e, portanto, os bits são considerados identicamente distribuídos e o valor de  $P(b_k)$  é igual a  $1/2$ . As probabilidades  $P(c_1 | \mathbf{r}) P(c_2 | \mathbf{r})$  de (23) são retiradas diretamente do vetor  $\mathbf{Z}$ , este obtido a partir da informação extrínseca do bloco equalizador,  $L_{ext}^E(c_n)$ :

$$P(c_n | \mathbf{r}) \cong \frac{\exp(c L_{ext}^E(c_n))}{1 + \exp(L_{ext}^E(c_n))} \quad (24)$$

onde  $c \in \{0, 1\}$  para os símbolos 0 e 1, respectivamente.

O algoritmo a seguir apresenta um resumo da equalização turbo empregando o algoritmo MAP BCJR. São usadas as seguintes convenções:  $\pi^{-1}(x)$  significa que  $x$  passou pelo desentrelaçador temporal,  $\pi(x)$  significa que  $x$  passou pelo entrelaçador temporal e  $MAP(x)$ , significa que  $x$  foi processado pelo algoritmo MAP no bloco equalizador ou no bloco decodificador:

### Equalização turbo com uso do algoritmo BCJR

*Entrada:* vetor de amostras na saída do canal  $\mathbf{r}$  e o número de iterações  $p$ .

*Inicialização:*  $L_a^E(c_n^\pi) = 0$

*Para cada iteração  $p$  faça:*

$$L^E(c_n^\pi) = \text{MAP}(L_a^E(c_n^\pi), \mathbf{r})$$

$$L_{\text{ext}}^E(c_n^\pi) = L^E(c_n^\pi) - L_a^E(c_n^\pi)$$

$$L_{\text{ext}}^E(c_n) = \pi^{-1}(L_{\text{ext}}^E(c_n^\pi))$$

$$L^D(c_n) = \text{MAP}(L_{\text{ext}}^E(c_n))$$

$$L_{\text{ext}}^D(c_n) = L^D(c_n) - L_{\text{ext}}^E(c_n)$$

$$L_{\text{ext}}^D(c_n^\pi) = \pi(L_{\text{ext}}^D(c_n))$$

$$L_a^E(c_n^\pi) = L_{\text{ext}}^D(c_n^\pi)$$

*fim das iterações*

*cálculo de  $L(b_k)$*

*Saída:*  $b_k$  = decisão abrupta sobre  $L(b_k)$

### V. EQUALIZADOR TURBO COM CANCELADOR DE INTERFERÊNCIA

A abordagem usada até aqui neste tutorial utiliza o algoritmo BCJR no bloco equalizador e no bloco decodificador. Existem alguns inconvenientes neste tipo de abordagem. Para o emprego do algoritmo BCJR no bloco equalizador é necessário que haja conhecimento total do canal para que o mesmo seja modelado como uma treliça. Além disso, para canais com grande dispersão temporal e modulações de alta ordem, o número de estados que modelam a treliça do bloco equalizador pode tornar o uso de um equalizador BCJR proibitivo computacionalmente.

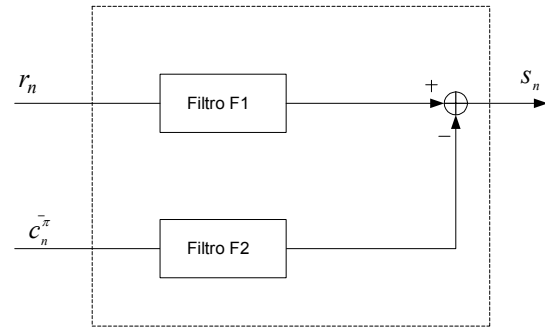
Em [5] é proposto um novo tipo de equalização turbo, onde o bloco equalizador de treliça é trocado por um *Cancelador de interferência* [20] [21]. O bloco equalizador deste sistema é apresentado na Figura 12. O filtro F1 é um filtro casado e, portanto, ele maximiza a relação sinal-ruído em sua saída, com pena do aumento de IIS. O filtro F2 é um filtro usado para estimar a IIS presente no canal e na saída do filtro F1. Tendo estimado a IIS, ela é subtraída da saída do filtro casado, eliminando-a sem diminuição da relação sinal-ruído [5] [6].

O *Cancelador de Interferência* da Figura 13 tem duas entradas: a entrada do filtro F1, que é o vetor de amostras recebidas do canal  $\mathbf{r}$ , e a média dos símbolos  $c_n^\pi$  como entrada do filtro F2. A média do símbolo  $c_n^\pi$  é estimada através da informação extrínseca  $L_{\text{ext}}^D(c_n^\pi)$  do decodificador da seguinte maneira:

$$\begin{aligned} \overline{c_n^\pi} &= E\{c_n^\pi\} = P(c_n^\pi = +1)1 + P(c_n^\pi = -1)-1 \\ &= \frac{\exp(L_{\text{ext}}^D(c_n^\pi))}{1 + \exp(L_{\text{ext}}^D(c_n^\pi))} + \frac{-1}{1 + \exp(L_{\text{ext}}^D(c_n^\pi))} \quad (25) \\ &= \tanh\left(\frac{L_{\text{ext}}^D(c_n^\pi)}{2}\right) \end{aligned}$$

Na primeira iteração, como não há informação extrínseca do decodificador,  $L_{\text{ext}}^D(c_n^\pi)$ , disponível, o equalizador usado pode ser um equalizador convencional como um filtro empregando o algoritmo LMS (*Least Mean Square*).

Nas próximas iterações o *Cancelador de Interferência* utiliza os valores de  $L_{\text{ext}}^D(c_n^\pi)$  para eliminar a IIS. A cada iteração o valor de  $L_{\text{ext}}^D(c_n^\pi)$  se torna mais confiável e, conseqüentemente, o desempenho do *Cancelador de Interferência* se aproxima de seu desempenho ótimo. Este desempenho ótimo é alcançado quando os símbolos  $c_n^\pi$  são conhecidos. Nesta situação, toda IIS é eliminada sem diminuição da relação sinal-ruído [5].



**Figura 12** - Bloco equalizador utilizando cancelador de interferência.

Os coeficientes dos filtros F1 e F2 apresentados em [5] são calculados a partir do critério de minimização do erro médio quadrático (do Inglês, *Mean Square Error*, MSE) entre  $s_n$  e  $c_n^\pi$ :

$$MSE = E\{|s_n - c_n^\pi|^2\} \quad (26)$$

Em [5] e [6] os coeficientes dos filtros F1 e F2 são calculados de maneira adaptativa através do algoritmo LMS (*Least Mean Square*), onde no início da transmissão é usada uma sequência de treinamento e em seguida o equalizador entra no modo de decisão direta [8]. Em [6] também é apresentado um algoritmo de adaptação dos coeficientes  $g(n)$ , baseado no algoritmo RLS (*Recursive Least Squares*), para canais variantes no tempo.

### VI. CRITÉRIOS DE PARADA

O número de iterações realizadas para que se obtenha um desempenho desejado na equalização turbo não é sempre o mesmo para todos os blocos processados. Para alguns blocos, apenas algumas iterações são suficientes para que o canal esteja praticamente livre de erros. Os chamados critérios de parada apresentados em [11] determinam quando o número de iterações em um bloco já é suficiente. Desta forma, o uso de critérios de parada otimiza o processo de equalização turbo diminuindo a latência do sistema, aumentando a vazão de dados transmitidos. Nesta seção são apresentados resumidamente dois dos critérios de parada abordados em [11]: o da *observação de decisões* abruptas e o da *função risco*.

#### VI.1. OBSERVANDO DECISÕES ABRUPTAS

Este critério consiste em observar, a cada iteração, a saída abrupta dos bits de informação do decodificador. Quando a saída não se alterar por duas iterações consecutivas, o equalizador pára com o processo iterativo. Este critério, apesar de alcançar bons resultados, torna necessário que se faça uma iteração a mais no bloco processado para que se observe que a saída não foi alterada.



**VI.2. FUNÇÃO RISCO**

Partindo do princípio que as LLR's medem a confiabilidade das decisões, este critério se baseia em observar o valor das LLR's dos bits de informação em cada iteração. Quando as LLR's estiverem com valores altos, o risco da decisão tomada se torna pequeno e o processo iterativo pode ser encerrado.

Calculando as probabilidades  $P(b_k | \mathbf{r})$  a partir das LLR's dos bits  $b_k$ , a função risco é dada por:

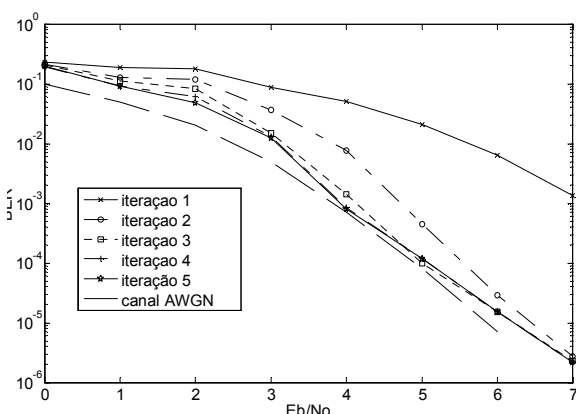
$$R = \frac{1}{L} \left( \sum_{\substack{k=1 \\ L(u_k > 0)}}^L P(b_k = -1 | \mathbf{r}) + \sum_{\substack{k=1 \\ L(u_k < 0)}}^L P(b_k = +1 | \mathbf{r}) \right) \quad (27)$$

onde  $L$  é o tamanho do bloco de informação. A função  $R$  deve estar abaixo de um determinado limiar para se obter um taxa de erro aceitável. Em [12] é dado um valor prático de  $5 \times 10^{-5}$  para este limiar.

**VII. COMENTÁRIOS FINAIS**

Neste tutorial foi apresentado que a equalização turbo realiza a equalização e a decodificação de canal de maneira iterativa. O equalizador turbo é composto de um bloco equalizador e um bloco decodificador, ambos com entradas e saídas suaves. As saídas suaves do bloco equalizador e do bloco decodificador são dadas na forma de LLR's e possuem a informação da confiabilidade das decisões estimadas pelos respectivos blocos. Para evitar uma redundância na realimentação de informação durante as iterações, deve-se subtrair as LLR's da saída de um bloco do equalizador turbo (equalizador ou decodificador) da LLR da entrada. Desta maneira, somente a informação extrínseca de um bloco será usada como informação *a priori* do bloco seguinte.

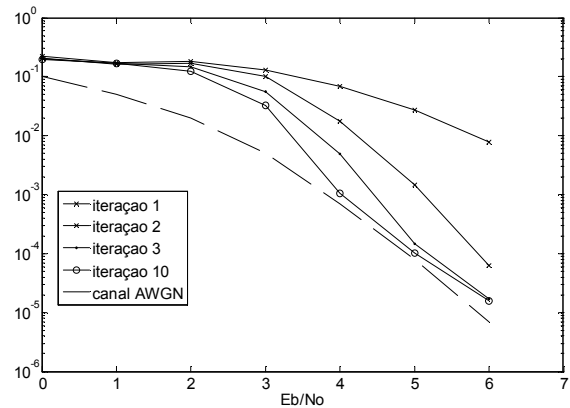
A Figura 13 apresenta os resultados obtidos por simulação computacional de um equalizador turbo operando em um canal com coeficientes  $g_{-1} = 0,407$ ,  $g_0 = 0,815$  e  $g_1 = 0,407$ .



**Figura 13** - Desempenho ( $BER \times E_b/N_0$ ) de um equalizador turbo em um canal com coeficientes  $g_{-1} = 0,407$ ,  $g_0 = 0,815$  e  $g_1 = 0,407$ .

A Figura 14 apresenta os resultados obtidos por simulação computacional de um equalizador turbo operando em um canal com coeficientes  $g_{-1} = 0,671$ ;  $g_0 = 0,5$ ;  $g_1 = 0,387$ ;  $g_2 = 0,316$  e  $g_3 = 0,224$ . Nas simulações apresentadas aqui foi simulada a transmissão de blocos de 512 bits

que foram codificados por um codificador convolucional de taxa  $1/2$ , como o da Figura 3. Os 1024 símbolos na saída do codificador foram embaralhados por um entrelaçador *S-random*, com  $S = 16$ , e transmitidos através do canal com IIS. Na recepção foi utilizado um equalizador turbo com cinco iterações empregando o algoritmo BCJR no bloco equalizador e no bloco decodificador.



**Figura 14** - Desempenho ( $BER \times E_b/N_0$ ) de um equalizador turbo em um canal com coeficientes  $g_{-1} = 0,671$ ;  $g_0 = 0,5$ ;  $g_1 = 0,387$ ;  $g_2 = 0,316$  e  $g_3 = 0,224$ .

Por meio das figuras 13 e 14 nota-se que existe um ganho considerável da primeira iteração para a segunda iteração do processo de equalização turbo, enquanto não há decréscimos significativos de taxa de erro de bit (do Inglês, *Bit Error Rate*, BER) nas próximas iterações. Nota-se também que o desempenho do equalizador turbo possui um limitante dado por um sistema codificado em um canal com ruído AWGN, sem a presença de IIS. Este limitante vem a confirmar que o equalizador turbo pode proporcionar aproximadamente o mesmo desempenho daquele obtido em um canal AWGN com codificação, sem a presença de IIS.

Outros sistemas de equalização turbo vêm sendo propostos. Além das modificações realizadas no bloco equalizador apresentadas na Seção III, outros blocos decodificadores têm sido empregados. Em [22] é apresentado um sistema que realiza de maneira combinada a equalização e a decodificação turbo utilizando códigos turbo convolucionais [3]. Em [23] é apresentado um sistema que também realiza, conjuntamente, a equalização e a decodificação turbo utilizando códigos turbo de bloco. Resultados obtidos por equalizadores turbo utilizando os códigos convolucional, turbo convolucional e turbo de bloco podem ser encontrados em [24]. Em [25] é apresentado um novo esquema de equalização turbo, onde é realizada uma decodificação turbo parcial de códigos produto de paridade simples multidimensionais no bloco decodificador. A abordagem apresentada em [25] possui baixa complexidade e supera o desempenho das abordagens utilizadas em [24] quando se utilizam altas taxas de codificação.

**VIII. REFERÊNCIAS**

[1] J. G. Proakis, "Digital Communications," third edition. New York: McGraw-Hill, 1995.  
 [2] C. Douillard, M. Jezequel, C. Berrou, A Picart, P. Didier, and A. Glavieux, "Iterative Correction of Intersymbol Interference: Turbo Equalization," in

- European trans. on telecomm., vol. 6, pp. 507–511, Sep-Oct 1995.
- [3] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon Limit Error-Correcting and Decoding: Turbo-Codes,” presented at the international conference on communication (ICC), pp. 1064-1070, IEEE, May 1993.
- [4] Ralf Koetter, A. C. Singer and Michael Tüchler, “Turbo Equalization,” IEEE Signal Processing Magazine, invited paper, 2003.
- [5] A. Glavieux, C. Laot, and J. Labat, “Turbo Equalization Over a Frequency Selective Channel,” in Proc. Of the Intern. Symposium on Turbo codes, Brest, France, pp. 96–102, September 1997.
- [6] C. Laot, A. Glavieux and J. Labat, “Turbo Equalization: Adaptive Equalization and Channel Decoding Jointly Optimized,” IEE Journal on Selected Areas in Communications, vol.19, n°9, September, 2001
- [7] M. Tüchler, R. Koetter, and A. Singer, “Turbo Equalization: Principles and New Results,” IEEE Trans. on Comm., vol. 50, pp. 754–767, May 2002.
- [8] S. Haykin, “Communication Systems,” 4<sup>th</sup> edition: John Wiley and Sons, Inc, New York, 2001.
- [9] S. T. Brink, “Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes,” IEEE Transactions on communications, vol.49, n°10, October 2001.
- [10] R. Otnes, T. Micheal, “EXIT Chart Analysis Applied To Adaptive Turbo Equalization,” presented at the Nordic Signal Processing Symposium, NORSIG, 2002, Hurlgruten Tromso-Trondheim, Norway, October 4-7 2002.
- [11] G. Bauch, H. Khorram and J. Hagenauer, “Iterative Equalization and Decoding in Mobile Communication Systems,” presented at the 2<sup>nd</sup> EPMCC’97 and 3<sup>rd</sup> ITG-Fachtagung Telecomm, Mobile Communication, Bonn, Germany, Oct. 1997.
- [12] G. Bauch and V. Franz, “A Comparison of Soft-in/Soft-out Algorithms for Turbo detection,” presented at the intern. conf. on telecomm., pp. 259–263, June 1998.
- [13] Anastasopoulos and K. Chugg, “Iterative Equalization/Decoding for TCM for Frequency-Selective Fading Channels,” Conf. Record 31th Asilomar Conf. On Signals, Systems & Comp., vol. 1, pp. 177–181, Nov 1997.
- [14] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” IEEE Transactions on Information Theory, vol. IT-20, pp. 284-287, March, 1974.
- [15] M. D. Cosac, “Técnicas de Redução de Estados Aliadas à Equalização Turbo,” Dissertação de Mestrado: Universidade Estadual de Campinas - UNICAMP, Campinas, SP, 2004.
- [16] M. Tüchler, R. Koetter, and A. Singer, “Iterative Correction of ISI via Equalization and Decoding using Priors,” Proc. Intern. Symp. on Information Theory, Sorrento, Italy, pp. 194, July 2000.
- [17] L. Hanzo, T. H. Liew, and B. L. Yeap, “Turbo Coding, Turbo equalization and Space-Time Coding for Transmission Over Fading Channels,” John Wiley and Sons, Inc 2002.
- [18] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proceedings of the IEE, vol.77, n°2, February 1989.
- [19] M. D. Linda, B. C. Iain, and P. Hoeher, “Joint Map Equalization and Channel Estimation for Frequency-Selective and Frequency-Flat Fast Fading Channels,” IEEE Transactions on Communications, vol. 49, n°12, December 2001.
- [20] A. Gersho and T. Lim, “Adaptive cancellation of intersymbol interference for data transmission,” Bell Syst. Tech. J., vol. 60, n° 11, pp.1997-2021, Nov.1981.
- [21] M. S. Muller and J. Salz, “A unified theory of data-aided equalization,” Bell Syst. Tech. J., vol. 60, n° 9, pp.2023-2038, Nov.1981.
- [22] D. Raphaeli, Y. Zarai, “Combined Turbo Equalization And Turbo Decoding,” Proc. Global Telecommunications Conf., Phoenix , AZ, Nov. 3-8, pp. 639-641, 1997.
- [23] M. Noorbakhsh and K. Mohamed-Pour, “Combined Turbo Equalization and Block Turbo Decoding,” IEE proceedings. Communications, vol. 150, n°3, pp.149-152, 2003.
- [24] B. L. Yeap, T. H. Liew, J. Hamorsky, and L. Hanzo, “Comparative Study of Turbo Equalization Schemes Using Convolutional, Convolutional Turbo, and Block-Turbo Codes,” IEE Transactions on Wireless Communications, vol. 1, N°2, April 2002.
- [25] A. F. Santos, “Um Esquema de Equalização Turbo Aplicando Decodificação Turbo de Códigos Produto de Paridade Simples Multidimensionais,” Dissertação de Mestrado: Instituto Nacional de Telecomunicações-INATEL, Santa Rita do Sapucaí, MG, Março de 2005.

#### SOBRE OS AUTORES

**André Fonseca dos Santos** nasceu em Pouso Alegre, em 05 de abril de 1980. Possui os títulos: *Engenheiro Eletricista* (Inatel, 2003) e *Mestre em Telecomunicações* (Inatel, 2005). Em 2003 estagiou na área de calibração de equipamentos de telecomunicações no CPqD. *Telecom & IT Solutions*. Desde janeiro de 2005 faz parte do corpo de engenheiros do INATEL, onde trabalha em um projeto financiado pela FAPEMIG que visa o desenvolvimento de uma Unidade Receptora e Conversora (URC) para sinais de televisão digital. Em 2003 ministrou aulas no INATEL na área de controle e automação e atualmente é professor na área de comunicação digital. Suas pesquisas incluem aspectos sobre televisão digital, sistemas OFDM, sincronismo, equalização turbo, decodificação turbo, especificamente equalização turbo conjunta com decodificação turbo.

**Dayan Adionel Guimarães** nasceu em Carrancas, MG, em 01 de março de 1969. Possui os títulos: *Técnico em Eletrônica* (ETE “FMC”, 1987), *Engenheiro Eletricista* (Inatel, 1994), *Especialista em Engenharia de Comunicação de Dados* (Inatel, 2003), *Especialista em Administração* com ênfase em Gerência de RH (FAI, 1996), *Mestre em Engenharia Elétrica* (Unicamp, 1998) e *Doutor em Engenharia Elétrica* (Unicamp, 2003). De 1988 a 1993 desenvolveu sensores e equipamentos para instrumentação industrial e controle e também foi Supervisor de Produção e Supervisor de Engenharia de Produtos na SENSE Sensores e Instrumentos. Desde de janeiro de 1995 é Professor do Inatel onde, por oito anos, foi responsável pela estrutura que dá apoio às atividades de ensino prático nas áreas de Telecomunicações, Eletrônica e Eletrotécnica. É membro do corpo editorial da revista *Telecomunicações* do Inatel. Suas pesquisas incluem aspectos gerais sobre transmissão digital e sistemas de comunicação móvel, especificamente sistemas CDMA Multiplicadora e esquemas de codificação para canais com desvanecimento, especificamente códigos turbo de bloco.