



**República Federativa do Brasil**  
Ministério da Indústria, Comércio Exterior  
e Serviços  
Instituto Nacional da Propriedade Industrial

**(11) PI 0305571-0 B1**

**(22) Data do Depósito:** 19/07/2003

**(45) Data de Concessão:** 03/01/2017



---

**(54) Título:** APARELHO E MÉTODO PARA DECODIFICAÇÃO TURBO UTILIZANDO DECODIFICADORES MAX-LOG-MAP MODIFICADOS

**(51) Int.Cl.:** H04L 1/00; H03M 13/29; H03M 13/39

**(52) CPC:** H04L 1/0055,H03M 13/2957,H03M 13/3905

**(30) Prioridade Unionista:** 19/07/2002 KR 10-2002-0042686

**(73) Titular(es):** SAMSUNG ELECTRONICS CO., LTD.

**(72) Inventor(es):** NAM-YUL YU; MIN-GOO KIM; SANG-HYUCK HA

**APARELHO E MÉTODO PARA DECODIFICAÇÃO TURBO UTILIZANDO  
DECODIFICADORES MAX-LOG-MAP MODIFICADOS**

HISTÓRICO DA INVENÇÃO

1. Campo da Invenção

[001] A presente invenção relaciona-se genericamente a um aparelho e método para a correção de erro de encaminhamento (FEC) em um sistema de comunicação digital e, em particular, e um aparelho e método para a decodificação turbo.

2. Descrição da Tecnologia Relacionada

[002] Em geral, códigos turbo são utilizados para a comunicação de dados de alta velocidade, particularmente em Evolution Data Only (1xEV-DO) ou Evolution Data and Voice (1xEV-DV). Berrou et al. propôs os códigos turbo em 1993. O codificador turbo é uma concatenação paralela de dois codificadores Recursive Systematic Convolutional (RSC) constituintes com um entrelaçador aleatório no meio. Assim, o código turbo é produzido pela codificação de bits de informação e bits de informação entrelaçada nos codificadores constituintes RSC. A decodificação turbo envolve uma concatenação serial de dois decodificadores constituintes, cada um para decodificar iterativamente, intercambiar sua informação extrínseca com o outro decodificador constituinte. Há três algoritmos aplicáveis para cada decodificador constituinte: Log-MAP, Max-Log-MAP, e Soft Output Viterbi Algorithm (SOVA).

[003] O algoritmo Log-MAP é a implementação no domínio logarítmico de um algoritmo MAP que é ótimo para decodificar uma palavra de informação em uma treliça. O algoritmo Max-Log-MAP é derivado facilmente do algoritmo

Log-MAP por uma aproximação de computação métrica. Apesar da vantagem da implementação simples quando comparado com o algoritmo Log-MAP, o algoritmo Max-Log-MAP leva a uma degradação do desempenho quando uma Proporção Sinal-a-Ruído (SNR) perfeita é possível no receptor.

[004] Para o algoritmo Log-MAP, a métrica de estado e uma Razão de Verossimilhança Logarítmica (LLR) são calculadas. A métrica de estado  $\alpha$  e  $\beta$  para um estado ( $s$  e  $s'$ ) em uma treliça no tempo de decodificação  $k$  são em uma relação recursiva expressa como:

$$\log(\alpha_k(s)) = \log\left(\sum_s \alpha_{k-1}(s') \gamma_k(s', s)\right), \log(\beta_{k-1}(s')) = \log\left(\sum_s \beta_k(s) \gamma_k(s', s)\right), \dots (1)$$

em que  $\gamma$  é uma métrica de ramo definida por um símbolo recebido em um canal. Utilizando a métrica de estado e a métrica de ramo, o LLR de um  $k^{\text{ésimo}}$  símbolo é obtido por:

$$L(\hat{u}_k) = \log\left(\frac{\sum_{(s', s) \text{ } \beta_k=0} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)}{\sum_{(s', s) \text{ } \beta_k=1} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)}\right) = \log\left(\frac{\sum_i e^{M_0(i)}}{\sum_i e^{M_1(i)}}\right)$$

$$= M_0(0) - M_1(0) + f_c$$

$$\text{onde, } f_c = \log\left(1 + \sum_{i \neq 0} e^{-(M_0(0) + M_0(i))}\right) - \log\left(1 + \sum_{i \neq 0} e^{-(M_1(0) + M_1(i))}\right), \dots (2)$$

[005] Na Equação (2),  $M_0(i)$  é uma  $i^{\text{ésima}}$  métrica em disposição por ordem decendente de métricas ( $\log(\alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s))$ ) para um símbolo de informação  $n(0$  ou  $1)$  no conjunto de estado ( $s, s'$ ) no tempo  $k$ . Portanto,  $M_0(0)$  e  $M_1(0)$  são as melhores métricas para o símbolo de informação  $1$  e  $0$  no tempo  $k$ , e  $f_c$  é um valor de correção definido pela diferença entre a melhor métrica e as outras métricas para cada símbolo de informação. Assim, o LLR é atualizado

utilizando a melhor diferença de métrica entre os símbolos de informação 0 e 1 no tempo  $k$  e o valor de correção  $f_c$ .

[006] Em resumo, o algoritmo Log-MAP gera todas as métricas de estado em uma treliça para cada decodificador constituinte pela Equação (1) e calcula o LLR de um símbolo de código na treliça utilizando sua métrica de estado pela Equação (2). Cada decodificador constituinte alimenta informação extrínseca derivada do LLR para o outro decodificador constituinte, para decodificação iterativa. Desta maneira, a decodificação turbo é efetuada.

[007] O algoritmo Max-Log-MAP é uma versão simplificada do algoritmo Log-MAP ao reduzir o cálculo de métrica de estado da Equação (1) para uma operação máxima expressa como:

$$\log \alpha_k(s) \approx \max_{s'} (\log(\alpha_{k-1}(s') \gamma_k(s', s)))$$

$$\log \beta_{k-1}(s') \approx \max_s (\log(\beta_k(s) \gamma_k(s', s))) \dots \dots (3)$$

[008] Da mesma maneira, o LLR do  $k^{\text{ésimo}}$  símbolo de decodificação é simplesmente calculado pela operação máxima. O LLR é atualizado utilizando apenas a melhor diferença métrica, supondo que  $f_c$  seja 0. Assim,

$$L(\hat{u}_k) = M_0(0) - M_1(0) \quad (4)$$

[009] Em resumo, o algoritmo Max-Log-MAP busca todas as métricas de estado na treliça por cada decodificador constituinte pela operação máxima da Equação (3) e calcula o LLR de um símbolo de código na treliça utilizando a melhor diferença métrica entre os símbolos de informação 0 e 1 pela Equação (4). Informação extrínseca derivada do LLR é alimentada ao outro decodificador constituinte, para a

decodificação iterativa. Desta maneira, a decodificação turbo é efetuada.

[0010] Um assim-chamado algoritmo Max-Log-MAP com Ganho de realimentação (FG) considera um ganho adicional derivado do LLR calculado pela Equação (4) para melhorar o desempenho de decodificação do algoritmo Max-Log-MAP. Um fator de ponderação multiplicado como o ganho de realimentação é cerca de 0,588235 e aplicado apenas à informação extrínseca de um segundo decodificador constituinte.

[0011] Como o algoritmo Log-MAP é a implementação no domínio logarítmico de um símbolo ótimo pelo algoritmo de decodificação MAP do símbolo, ele efetua tão bem quanto o algoritmo MAP. No entanto, quando o algoritmo Log-MAP é implementado em hardware, a função de  $\log(1+e^{-\Delta})$  que define cada métrica precisa ser implementado em hardware ou na forma de uma tabela de pesquisa. O algoritmo Max-Log-MAP, por outro lado, não requer qualquer tabela de pesquisa, mas efetua pior que o algoritmo Log-MAP. Os benefícios e deficiências do algoritmo Log-MAP e do algoritmo Max-Log-MAP são os seguintes:

(1) O algoritmo Log-MAP: Como ele é um algoritmo de decisão de símbolo a símbolo ótimo, ele é o melhor algoritmo de decodificação turbo. No entanto, a implementação de  $\log(1+e^{-\Delta})$  aumenta a complexidade de hardware. Ademais,  $\log(1+e^{-\Delta})$  é uma função não linear e assim uma estimativa de SNR precisa de um símbolo recebido é necessária para calcular a métrica de ramo pela qual  $\Delta$  é definida. Se a estimação de SNR envolve erros, este erro no casamento de SNR degrada acentuadamente o desempenho;

(2) O algoritmo Max-Log\_MAP: O cálculo  $\text{Log}()$  não é necessário para o cálculo de métrica porque todas as métricas são calculadas pela operação máxima. Portanto, o problema de complexidade de hardware maior conforme encontrado no algoritmo Log-MAP não é produzido. Ademais, o cálculo de métricas pela operação máxima obvia a necessidade da função não linear  $(\log(1+e^{-A}))$ , o que implica que não há nenhum problema relacionado ao erro de casamento de SNR. No entanto, como o algoritmo Max-Log\_MAP é uma aproximação do algoritmo Log-MAP, ele efetua cerca de 0,3 a 0,4 dB pior que o algoritmo Log-MAP.

[0012] Conforme é descrito acima, o algoritmo Log-MAP e o algoritmo Max-Log-MAP causam maior complexidade de hardware e degradação de desempenho como suas respectivas deficiências.

#### SINOPSE DA INVENÇÃO

[0013] Portanto, é um objeto da presente invenção fornecer um aparelho e método de decodificação turbo que desempenhem melhor que o algoritmo Max-Log-MAP na decodificação turbo.

[0014] É outro objeto da presente invenção fornecer um aparelho e método de decodificação turbo que é menos completo que o algoritmo Log-MAP.

[0015] Os objetos acima são substancialmente alcançados por um decodificador constituinte para decodificar um código turbo e um método de decodificação constituinte do mesmo. A melhor métrica e a segunda melhor métrica são calculadas para o valor de um símbolo de código recebido em um estado arbitrário de uma treliça de decodificação turbo durante a decodificação turbo do símbolo de código. A

informação extrínseca necessária para a decodificação turbo do símbolo de código é calculada. A diferença entre a informação extrínseca e uma diferença da melhor métrica-segunda melhor métrica é calculada. O LLR do símbolo de código é atualizado pela multiplicação da diferença calculada por um fator de ponderação predeterminado e decidir o valor do símbolo de código.

[0016] A informação extrínseca é calculada utilizando a diferença entre as duas métricas, o símbolo de entrada refletindo um SNR e a informação *a priori* do símbolo de entrada.

[0017] O fator de ponderação é inferior a 1 e aproximado de 1. Preferivelmente, ele é maior que 0,588235. Preferivelmente, ele é  $1/2+1/4+1/16$ .

[0018] Se o SNR pode ser estimado perfeitamente, o fator de ponderação é calculado utilizando uma função logarítmica. Se o SNR não pode ser perfeitamente estimado, o fator de ponderação é calculado utilizando uma função linear aproximada.

[0019] No decodificador constituinte para decodificar um código turbo, um primeiro somador para calcular o LLR de um símbolo de código recebido ao calcular a diferença entre a probabilidade do símbolo de código ser 1 e a probabilidade do símbolo de código ser 0 em um estado arbitrário de uma treliça de decodificação turbo durante a decodificação turbo do símbolo de código. Um segundo somador soma a informação de transmissão e a informação *a priori* do símbolo de código. Um terceiro somador calcula a diferença entre as saídas do primeiro e do segundo somadores como informação extrínseca. Um primeiro

multiplicador multiplica a saída do terceiro somador por um fator de ponderação predeterminado como um ganho de realimentação. Um calculador de valor de correção calcula um valor de correção utilizando a diferença entre a melhor métrica e a segunda melhor métrica do símbolo de código. Um quarto somador soma o valor de correção à saída do primeiro multiplicador.

[0020] O calculador de valor de correção inclui um quinto somador para calcular a diferença entre a melhor métrica e a segunda melhor métrica para um símbolo de informação 0 como o valor do símbolo de código, um sexto somador para calcular a diferença entre a melhor métrica e a segunda melhor métrica para um símbolo de informação 1 como o valor do símbolo de código, e uma tabela de pesquisa para armazenar valores de correção com base na função logarítmica para as saídas do quinto e do sexto somadores e emitir valores de correção para as saídas do quinto e do sexto somadores. O calculador de valor de correção ainda inclui um sétimo somador para calcular a diferença entre os valores de correção, um segundo multiplicador para multiplicar a saída do sétimo somador por um fator de ponderação predeterminado, um oitavo somador para calcular a diferença entre as saídas do quinto e do sexto somadores, um terceiro multiplicador para multiplicar a saída do oitavo somador pela inclinação de uma função linear aproximada da função logarítmica, e um seletor para selecionar uma das saídas do segundo e do terceiro multiplicadores de acordo com a confiabilidade do SNR do símbolo de código.

[0021] O fator de ponderação é preferivelmente  $1/2+1/4+1/16$ .

[0022] A confiabilidade do SNR é determinada de acordo como se uma estimação de SNR perfeita é ou não possível. O seletor emite o valor recebido do segundo multiplicador se uma estimação de SNR perfeita é possível, e o valor recebido do terceiro multiplicador se a estimação de SNR perfeita é impossível.

#### DESCRIÇÃO SUCINTA DOS DESENHOS

[0023] O que antecede e outros objetos, recursos e vantagens da presente invenção tornar-se-ão mais aparentes da seguinte descrição detalhada quando tomados em conjunto com os desenhos acompanhantes nos quais:

A Figura 1 é um diagrama de blocos que ilustra um exemplo de um decodificador turbo que utiliza um algoritmo modificado Max-Log-MAP de acordo com uma versão da presente invenção;

A Figura 2 é um fluxograma que ilustra um exemplo das operações para encontrar a melhor métrica  $M_n(0)$  e a segunda melhor métrica ( $M_n(1)$ ) no tempo de decodificação  $k$  de acordo com uma versão da presente invenção;

A Figura 3 é um fluxograma que ilustra um exemplo das operações para calcular um LLR e informação extrínseca para a decodificação iterativa do algoritmo Max-Log\_MAP modificado de acordo com uma versão da presente invenção;

A Figura 4 é um diagrama de blocos de um exemplo de blocos de função para encontrar simultaneamente a melhor e a segunda melhor métrica para o LLR em um tempo de decodificação arbitrário de acordo com uma versão da presente invenção;

A Figura 5 é um diagrama de blocos de um exemplo de blocos de função para produzir informação extrínseca para um símbolo de informação a um tempo de decodificação arbitrário de acordo com uma versão da presente invenção;

A Figura 6 é um diagrama de blocos de um exemplo de blocos de função para calcular um valor de correção utilizado para obter a informação extrínseca de acordo com uma versão da presente invenção;

As Figuras 7 e 8 são gráficos que ilustram exemplos do desempenho de Taxa de Erro de Bit (BER) e da Taxa de Erro de Quadro (FER) de algoritmos de decodificação turbo quando um tamanho de Pacote de Codificação (EP) é de 3864 e a velocidade de código geral é de 1/2 de acordo com uma versão da presente invenção;

As Figuras 9 e 10 são gráficos que ilustram exemplos do desempenho BER e FER de log2 MaxLogMAP, mod MaxLogMAP, MaxLogMAP com FG, e MaxLogMAP sobre iterações em  $E_b/N_o$  de 1,3 dB de acordo com uma versão da presente invenção;

As Figuras 11 e 12 são gráficos que ilustram exemplos do desempenho BER e FER de algoritmos de decodificação turbo quando o tamanho EP é de 792 e a velocidade de código efetiva é de 1/5 de acordo com uma versão da presente invenção;

As Figuras 13 e 14 são gráficos que ilustram exemplos do desempenho BER e FER sobre iterações em  $E_b/N_o$  de 0,7 dB quando o tamanho EP é de 3864 de acordo com uma versão da presente invenção;

As Figuras 15 e 16 são gráficos que ilustram exemplos do desempenho BER e FER sobre erro de casamento de SNR em  $E_b/N_o$  de 1,2 dB quando o tamanho EP é de 3864 e a

velocidade de código efetiva é de 1/2 de acordo com uma versão da presente invenção.

#### DESCRIÇÃO DETALHADA DA VERSÃO PREFERIDA

[0024] Versões da presente invenção serão aqui descritas abaixo com referência aos desenhos acompanhantes. Na descrição seguinte, funções ou construções bem conhecidas são omitidas por concisão.

[0025] A presente invenção pretende fornecer um algoritmo Max-Log-MAP aprimorado que, ao modificar a atualização LLR do algoritmo Max-Log-MAP existente, efetua apenas 0,1 dB ou menos pior que o algoritmo Log-MAP e oferece melhor desempenho de decodificação turbo que o algoritmo Max-Log-MAP e o algoritmo Max-Log-MAP com FG. Como o algoritmo Max-Log-MAP aprimorado é basicamente um algoritmo de decodificação turbo com base no algoritmo Max-Log-MAP, ele fornece, com vantagem, um ligeiro aumento na complexidade de hardware sem qualquer erro de casamento de SNR.

[0026] Os recursos da presente invenção são apresentados sucintamente conforme segue:

[0027] Para a atualização LLR em um tempo de decodificação arbitrário, a segunda melhor métrica para os símbolos de informação 0 e 1 bem como a melhor métrica são considerados. Notadamente, a segunda melhor métrica é excluída de consideração para a atualização LLR no algoritmo Max-Log-MAP existente. Será aparente de resultados de simulação descritos posteriormente que a atualização LLR da presente invenção leva a um desempenho de decodificação turbo tão bom quanto o algoritmo Log-MAP.

[0028] Se um valor de correção  $f_c$ , que é calculado

utilizando a segunda melhor métrica para os símbolos de informação 0 e 1 em um tempo de decodificação arbitrário, é definido como sendo uma função não linear, o erro de casamento SNR leva a mudanças no desempenho. Portanto,  $f_c$  é aproximado a uma função linear. Os resultados da simulação também esclarecerão que a aproximação de  $f_c$  à função linear resulta em excelente desempenho de decodificação turbo independentemente do erro de casamento SNR.

[0029] Assim, a aproximação linear de  $f_c$  será descrita de acordo com a presente invenção. Além disso, o desempenho de decodificação turbo é avaliado no caso em que  $f_c$  é definido como sua função logarítmica original e a aplicabilidade desta definição de  $f_c$  é investigada.

[0030] A Figura 1 é um diagrama de blocos que ilustra um exemplo de um decodificador turbo que utiliza um algoritmo Max-Log-MAP modificado de acordo com uma versão da presente invenção. Como foi descrito acima, o algoritmo Max-Log-MAP modificado refere-se a um algoritmo Max-Log-MAP que atualiza um LLR utilizando a melhor e a segunda melhor métrica para um símbolo de informação em um tempo de decodificação de acordo com uma versão da presente invenção.

[0031] O algoritmo Max-Log-MAP modificado é aplicado a cada decodificador constituinte (DEC1 e DEC2). Uma Controladora de Ganho de realimentação (FEC) para ponderar informação extrínseca também é aplicada a cada decodificador constituinte.

[0032] Com referência à Figura 1, o primeiro e o segundo decodificadores constituintes (DEC1 e DEC2) 101 e 104, respectivamente, derivam informação extrínseca e LLR

para um símbolo de informação utilizando o algoritmo Max-Log-MAP modificado. Isto é, os decodificadores constituintes 101 e 104 correspondem, cada um, a um dos codificadores constituintes de um codificador turbo. Um entrelaçador 102 entrelaça um sinal recebido do primeiro decodificador constituinte 101. Ao considerar o entrelaçamento dos dados entre os códigos constituintes de um código turbo, o entrelaçador 102 permuta a seqüência de dados de modo que a saída do primeiro decodificador constituinte 101 casa com a entrada do segundo decodificador constituinte 104. Um primeiro FG 103 multiplica o sinal entrelaçado por um fator de ponderação derivado da informação extrínseca calculada no primeiro decodificador constituinte 101 de acordo com o algoritmo Max-Log-MAP modificado. O fator de ponderação é um valor empírico. Ele é maior no algoritmo Max-Log-MAP do que no algoritmo Log-MAP. Considerando isto, a informação extrínseca para um símbolo de informação é multiplicada por um fator de ponderação inferior a 1, assim alcançando um desempenho melhor. O segundo decodificador constituinte 104 decodifica a saída do primeiro FGC 103. Um desentrelaçador 105 efetua o desentrelaçamento de modo que a saída do segundo decodificador constituinte 104 casa com a entrada do primeiro decodificador constituinte 101. Um segundo FGC 106 multiplica o sinal desentrelaçado por um fator de ponderação derivado da informação extrínseca calculada no primeiro decodificador constituinte 101 de acordo com o algoritmo Max-Log-MAP modificado. A saída do segundo FGC 106 é aplicada à entrada do primeiro decodificador constituinte 101.

[0033] Os somadores 107 e 108 somam a confiabilidade de transmissão e a probabilidade *a priori* (APP) de um símbolo de código recebido para gerar um LLR para um símbolo de informação utilizando a informação extrínseca derivada do segundo decodificador constituinte 105. A informação *a priori* é o LLR da probabilidade de um símbolo de informação ser 0 para a probabilidade do símbolo de informação ser 1. Em uma teoria de codificação geral, os símbolos de informação 0 e 1 são equiprováveis. Portanto, a informação *a priori* inicial é sempre 0. À medida que a decodificação iterativa prossegue, a informação extrínseca de cada decodificador constituinte é utilizada como a informação *a priori* de um símbolo de informação para o outro decodificador constituinte. Assim, a informação *a priori* não é mais 0. Um decodificador 109 decide o sinal do LLR. Se o sinal LLR é positivo, o decodificador 109 gera um símbolo de informação 0, e se o sinal LLR é negativo, ele gera um símbolo de informação 1. O valor de decisão é alimentado tanto para uma memória provisória de saída 110 como um verificador CRC 111. Em uma versão da presente invenção, a memória provisória de saída 110 pode ser uma memória para armazenar o valor de decisão 0 ou 1. O verificador CRC 111 verifica um CRC inserido *a priori* para detectar erros em um quadro de símbolos de informação decodificados.

[0034] A implementação do algoritmo Max-Log-MAP modificado nos decodificadores constituintes será agora descrita a seguir.

[0035] O algoritmo Max-Log-MAP modificado evoluiu do algoritmo Max-Log-MAP pela modificação de seu processo de atualização LLR. Assim, por simplicidade de implementação e

manutenção da insensitividade da decodificação turbo ao erro de casamento de SNR, a Eq. (3) é ainda adotada para calcular as métricas de estado  $\alpha$  e  $\beta$  para o segundo algoritmo Max-Log-MAP modificado. Assim, a Equação (2) é utilizada com o valor de correção  $f_c$  aproximado para definir o LLR para o algoritmo Max-Log-MAP modificado.

[0036] A aproximação de  $f_c$  envolve definir  $f_c$  utilizando a melhor métrica  $M_n(0)$  e a segunda melhor métrica  $M_n(1)$  para os símbolos de informação 0 e 1 entre todas as métricas  $M_n(i)$  que constituem  $f_c$  na Equação (2). Em outras palavras, o algoritmo de decodificação turbo da presente invenção atualiza um LLR em um tempo de decodificação arbitrário, considerando a segunda melhor métrica para os símbolos de informação 0 e 1, que são excluídos na atualização LLR do algoritmo Max-Log-MAP, bem como a sua melhor métrica.

[0037] Para a atualização LLR no algoritmo Max-Log-MAP modificado,  $f_c$  é aproximado como

$$f_c \approx \log(1 + e^{-(M_0^{(0)} - M_0^{(1)})}) - \log(1 + e^{-(M_1^{(0)} - M_1^{(1)})}) \quad (5)$$

[0038] Como é observado da Equação (5),  $f_c$  é definido utilizando a melhor métrica  $M_n(0)$  e a segunda melhor métrica  $M_n(1)$  para um símbolo de informação  $n$  em um tempo de decodificação. As métricas  $M_n(i)$  ( $i > 1$ ) menor que a segunda melhor métrica  $M_n(1)$  são descartadas na aproximação porque elas têm um impacto de negligência insignificante em  $f_c$ . Enquanto o algoritmo Max-Log-MAP busca todos os conjuntos de estado ( $s'$ ,  $s$ ) em uma treliça em um tempo de decodificação arbitrário e calcula a melhor métrica  $M_n(0)$  para o símbolo de informação, a métrica de atualização para cada conjunto de estado, o algoritmo Max-Log-MAP modificado

calcula a segunda melhor métrica ( $M_n(1)$ ) além da melhor métrica  $M_n(0)$ , e isso simultaneamente para não aumentar o tempo de decodificação. Para este fim, que a métrica para um estado  $s$  seja  $m(s)$ . Então  $M_n(0)$  e  $M_n(1)$  são calculados simultaneamente na maneira ilustrada na Tabela 1.

Tabela 1

- (1) inicialização:  $s=0$ ,  $M_n(0)=\text{MIN}$ ,  $M_n(1)=\text{MIN}$
- (2) encontrar  $m(s)$
- (3) se  $M_n(0) < m(s)$ ,  $M_n(1)=M_n(0)$  e  $M_n(0)=m(s)$   
caso contrário se  $M_n(1) < m(s)$ ,  $M_n(1)=m(s)$
- (4) se  $s=S-1$ , parar. Caso contrário, vá para (5)
- (5) aumentar  $s$  por 1. Vá para (2)

[0039] Na Tabela 1, MIN é um valor muito pequeno equivalente a  $-\infty$ , para a inicialização da métrica de estado e  $S$  é o número total de estados na treliça de códigos convolucionais constituintes.

[0040] A Figura 2 é um fluxograma que ilustra um exemplo de operações para calcular a melhor métrica  $M_n(0)$  e a segunda melhor métrica  $M_n(1)$  no tempo de decodificação  $k$  de acordo com uma versão da presente invenção.

[0041] Com referência à Figura 2, o estado de treliça e a melhor e a segunda melhor métrica para os símbolos de informação 0 e 1 são fixados para valores iniciais no tempo de decodificação  $k$  na etapa 200 conforme indicado por (1) na Tabela 1. Na etapa 202, uma métrica para o símbolo de informação  $n$  (0 ou 1) é calculada, aumentando o estado por 1 a cada vez. Assim, a operação da Figura 2 é o processo de encontrar o estado atual  $s$ . A métrica calculada é comparada com a melhor métrica existente para o símbolo de informação  $n$  na etapa 204. Se a métrica atual é maior que a melhor

métrica existente, o procedimento vai para a etapa 206. Caso contrário, ele vai para a etapa 208. Na etapa 206, a métrica atual é fixada como a melhor métrica e a melhor métrica existente é fixada como a segunda melhor métrica.

[0042] Por outro lado, na etapa 208, a métrica atual é comparada com a segunda melhor métrica existente. Se a métrica atual é maior que a segunda melhor métrica existente, a segunda melhor métrica é atualizada para a métrica atual na etapa 210. Se a métrica atual é igual ou inferior à segunda melhor métrica existente na etapa 206 ou 210, ou na etapa 208, o procedimento vai para a etapa 212.

[0043] Na etapa 212, é determinado se o estado atual é o último estado. Se for, o procedimento termina. Se não for, o estado é aumentado em 1 na etapa 214.

[0044] Desta maneira, a métrica melhor e a segunda melhor  $M_n(0)$  e  $M_n(1)$  são obtidas ao mesmo tempo em um tempo de decodificação arbitrário. Utilizando essas métricas, o valor de correção  $f_c$  é aproximado como a Equação (5).

[0045] No entanto, a aproximação não linear de  $f_c$  na Equação (5) afeta o desempenho de decodificação de acordo com o valor absoluto de um símbolo de entrada no decodificador turbo. Isto é, o receptor não pode estimar um SNR preciso, resultando em um erro de casamento de SNR. Como resultado, se um símbolo de entrada de decodificador é modificado, o desempenho de decodificação turbo também é modificado. Portanto, é necessário aproximar  $f_c$  da função logarítmica para uma função linear.

[0046] Será feita agora uma descrição de uma aproximação da função logarítmica comparada com uma função linear.

[0047] Com uma representação de  $f_c$  pela função logarítmica como no algoritmo Log-MAP, ou por uma tabela de pesquisa correspondente à função logarítmica, o erro de casamento de SNR muda  $E_s/N_0$  que é multiplicado por um símbolo de entrada de decodificador apesar de um SNR constante para o símbolo de entrada, que, de forma assombrosa, muda o desempenho da decodificação turbo. Para manter o desempenho de decodificação independentemente do valor do símbolo de entrada, a função logarítmica precisa ser modificada. A Equação (6) oferece uma aproximação para a função logarítmica.

$$l(x) = \log(1+e^{-x}) \approx -Kx+c \quad x>0, K>0, c>0 \quad (6)$$

[0048] Uma função tendo uma métrica como fator precisa ser linear com relação à métrica, para alcançar um LLR na maneira que oferece um desempenho de decodificação independentemente da mudança de um símbolo de entrada. Se  $f_c$  muda de uma maneira não linear dependendo da métrica variar com o valor do símbolo de entrada,  $f_c$  também muda não linearmente com relação ao LLR de acordo com o símbolo de entrada variável apesar do mesmo SNR. Portanto, um desempenho constante não é assegurado.

[0049] Na aproximação expressa como Eq (6), a constante  $c$  é desprezível. Ela é recuada pela aproximação de  $l(x)$  como uma função de primeira ordem com a constante  $c$  porque  $f_c$  é definida pela diferença entre as funções  $l(x)$  tendo métricas para os símbolos de informação 0 e 1 como seus fatores.

[0050] Esta aproximação é grosseira. Devido a erros causados pela aproximação grosseira, o algoritmo Max-Log-MAP modificado da presente invenção desempenha pior que o

algoritmo Max-Log<sub>MAP</sub> modificado com  $l(x)$  definido como uma função logarítmica. No entanto, definir  $l(x)$  como uma função não linear poderá levar a um bom desempenho se a estimação SNR perfeita for assegurada, mas o desempenho de decodificação é modificado quando o erro de casamento de SNR muda o valor do símbolo de entrada.

[0051] Através da aproximação, o LLR é atualizado no algoritmo Max-Log-MAP modificado por

$$L(\hat{u}_k) = M_0(0) - M_1(0) + f_c$$

$$\text{em que } f_c = K(M_0(0) - M_0(1)) + K(M_1(0) - M_1(1)) \quad (7)$$

[0052] A segunda melhor métrica  $M_n(1)$  é calculada ao mesmo tempo que a melhor métrica  $M_n(0)$  pela Equação (7) no algoritmo de aproximação.

[0053] Agora, os fatores de ponderação aplicados à informação extrínseca serão descritos. A informação extrínseca sobre um símbolo de informação pode ser obtida utilizando um LLR do processo de atualização do LLR do algoritmo Max-Log-MAP modificado. Como o algoritmo Max-Log-MAP produz informação extrínseca através de aproximação repetida, a informação extrínseca tem um valor relativamente grande comparado com a informação extrínseca no algoritmo Log-MAP. Para reduzir este impacto, a informação extrínseca para o símbolo de informação é multiplicada por um fator de ponderação. No algoritmo Max-Log-MAP convencional com FG, um fator de ponderação predeterminado, por exemplo, 0,588235, é multiplicado pela informação extrínseca do segundo decodificador constituinte para cada iteração. No entanto, no algoritmo Max-Log-MAP modificado, o valor de correção  $f_c$  que reflete a segunda

melhor métrica está envolvido no LLR e assim um fator de ponderação para a informação extrínseca precisa ser mais próximo de 1 que  $f_c$ . Considerando um fator de ponderação  $W_f$ , a informação extrínseca é formada como:

$$\begin{aligned} L_e(u_k) &= W_f((M_0(0) - M_1(0) + f_c) - (L_c y_k + La(\hat{u}_k))) \\ &= W_f((M_0(0) - M_1(0)) - (L_c y_k + La(\hat{u}_k))) + f_c' \end{aligned}$$

em que  $f_c' = W_f(\log(1 + e^{-(M_0(0) - M_0(1))}) - \log(1 + e^{-M_1(0) - M_1(1)}))$

ou

$$f_c' = -K'(M_0(0) - M_0(1)) + K'(M_1(0) - M_1(1)) \quad (8)$$

[0054] Na Equação (8),  $K' = K \cdot f \cdot L_c y_k$  é um sinal que reflete uma confiabilidade de canal na entrada do decodificador turbo e  $La(\hat{u}_k)$  é a informação *a priori* do símbolo de informação atual. A fórmula é produzida ao subtrair a informação extrínseca da diferença entre a melhor métrica e a segunda melhor métrica e depois acrescentar um novo valor de correção  $f_c'$  à diferença resultante. Doravante,  $f_c'$  é denominado de valor de correção.

[0055] A descrição seguinte é feita para definir um LLR e a informação extrínseca para a decodificação iterativa no algoritmo Max-Log-MAP modificado.

[0056] A Figura 3 é um fluxograma que ilustra um exemplo das operações para calcular o LLR para um símbolo de informação e a informação extrínseca utilizada para a decodificação iterativa no algoritmo Max-Log-MAP modificado de acordo com uma versão da presente invenção.

[0057] Com referência à Figura 3, uma métrica de ramo  $\gamma$  é calculada para uma transição de estado arbitrária em uma treliça na etapa 400 e as métricas de estado  $\alpha$  e  $\beta$  são atualizadas para todos os conjuntos de estado  $(s, s')$  em

relação à transição de estado na etapa 402. Na etapa 404, a melhor métrica  $M_n(0)$  e a segunda melhor métrica  $M_n(1)$  são encontradas ao mesmo tempo para obter o LLR no procedimento da Figura 2, atualizando a métrica de estado. O LLR é calculado utilizando a diferença entre  $M_n(0)$  e  $M_n(1)$ , a entrada de decodificador com um SNR considerado nela, e a informação *a priori* de um símbolo de informação pela Equação (8) na etapa 406. Esta etapa é efetuada nos blocos de função 601, 602 e 603 ilustradas na Figura 5. Na etapa 408, a informação extrínseca é multiplicada por um fator de ponderação  $W_f$ , que é efetuado no bloco de função 604 ilustrado na Figura 5.

[0058] O valor de correção  $f_c'$  é escolhido como um de dois valores definidos na Equação (8), dependendo de se a função logarítmica é aproximada ou não a uma função linear. Se o receptor pode efetuar a estimação de SNR perfeita,  $f_c'$  é escolhido como a função logarítmica original. Caso contrário, ela é escolhida como a função linear aproximada. Assim, se a estimação SNR perfeita é possível na etapa 410, o procedimento vai para a etapa 412, e se ela é impossível, o procedimento vai para a etapa 414. A função logarítmica é escolhida quando os blocos de função 701, 702, 703, 705 e 707, ilustrados na Figura 6, emitem FLAG como 0, enquanto a função linear é escolhida quando os blocos de função 701, 702, 704, 706 e 708, ilustrados na Figura 6, emitem FLAG como 1.

[0059] A Figura 4 é um diagrama de blocos de um exemplo de blocos de função para encontrar a melhor métrica e a segunda melhor métrica em relação a um LLR em um tempo de

decodificação arbitrário de acordo com uma versão da presente invenção.

[0060] Com referência à Figura 4, a linha sólida em negrito denota a seção para encontrar a segunda melhor métrica, isto é, os blocos de função 511 a 514. Portanto, os outros blocos de função 501, 502 e 503 operam de acordo com o algoritmo Max-Log-MAP. Esses blocos de função atualizam as métricas para todos os estados da treliça, aumentando o índice de um estado por 1 a cada vez. Aqui, o sinal SEL0 é 0 para o primeiro estado e 1 para os estados seguintes. O sinal SEL1 é 0 zero para o primeiro e segundo estados e 1 para os estados seguintes.

[0061] Os blocos de função 502, 503, 511, 513 e 514 são seletores para emitir a entrada na porta 0 se um sinal selecionado for 0 e a entrada na porta 1 se o sinal selecionado for 1. Os blocos de função 501 e 512 são seletores para emitir 1 se o sinal na porta a for inferior a um sinal na porta b, e 0 se o sinal na porta a é igual ou maior que o sinal na porta b.

[0062] A Figura 5 é um diagrama de blocos de um exemplo de blocos de função para gerar informação extrínseca a respeito de um símbolo de informação em um tempo de decodificação arbitrário de acordo com uma versão da presente invenção.

[0063] Com referência à Figura 5, um primeiro somador 601 emite a diferença entre a melhor métrica para 0 e 1 como informação LLR a respeito de um símbolo de informação. Um segundo somador 602 soma a informação de transmissão e a informação *a priori* de um símbolo recebido. Um terceiro somador 603 subtrai a soma recebida do segundo somador 602

da informação LLR recebida do primeiro somador 601. A saída do terceiro somador 603 é informação extrínseca definida no algoritmo Max-Log-MAP existente. Um multiplicador 604 multiplica a informação extrínseca por um fator de ponderação, como é feito no algoritmo Max-Log-MAP existente com FG. Se o fator de ponderação é 1, isto efetua o algoritmo Max-Log-MAP. Um quarto somador 605 acrescenta o valor de correção  $f_c'$  obtido pelos blocos de função ilustrados na Figura 6 à saída do multiplicador 604. Assim, a informação extrínseca final para o algoritmo Max-Log-MAP modificado é obtida.

[0064] Isto é, a informação extrínseca é obtida para o algoritmo Max-Log-MAP modificado ao ainda utilizar o multiplicador 604 associado ao fator de ponderação  $W_f$  e o somador 605 associado ao valor de correção  $f_c'$ , comparado com o algoritmo Max-Log-MAP. Assim, comparado com o algoritmo Max-Log-MAP com FG, o somador 605 é ainda utilizado.

[0065] A Figura 6 é um diagrama de blocos de um exemplo de blocos de função para calcular o valor de correção  $f_c'$  para utilização no cálculo da informação extrínseca de acordo com uma versão da presente invenção.

[0066] Com referência à Figura 6, o primeiro somador 701 calcula a diferença entre a melhor métrica e a segunda melhor métrica para um símbolo de informação 0 e o segundo somador 702 calcula a diferença entre a melhor métrica e a segunda melhor métrica para um símbolo de informação 1. Uma tabela de pesquisa (LUT) 703 encontra valores de correção da função logarítmica definida na Equação (8) utilizando as diferenças. O terceiro somador 707 calcula a diferença

entre os valores de correção. O primeiro multiplicador 707 multiplica a diferença por um fator de ponderação, assim decidindo um valor de correção final.

[0067] O quarto somador 704 calcula a diferença entre as saídas do primeiro e do segundo somadores 701 e 702. O segundo multiplicador 706 multiplica a diferença por um valor de inclinação, assim decidindo um valor de correção aproximado a uma função linear.

[0068] Um dos valores de correção definidos na Equação (8) é escolhido de acordo com o sinal FLAG. Para a escolha da função logarítmica, o seletor 708 seleciona a entrada na porta 0. Pelo contrário, para a escolha da função linear, o seletor 708 seleciona a entrada na porta 1. O primeiro caso requer um LUT, enquanto o último caso simplesmente requer somadores e um multiplicador. Notadamente, quando FLAG é 0, a estimação SNR perfeita precisa ser assegurada no receptor. Esta estrutura ilustrada na Figura 6 é ainda implementada em hardware para o algoritmo Max-Log-MAP modificado. Se o fator de ponderação  $W_f$  e o valor  $K'$  podem ser expressos como expoentes de 2, os multiplicadores das Figuras 5 e 6 podem ser implementados como simples seletores de bit ou somadores incluindo eles.

[0069] Para avaliar o desempenho de decodificação turbo do algoritmo Max-Log-MAP modificado de acordo com a presente invenção, simulações foram efetuadas sob as condições seguintes.

[0070] Todas as simulações foram de pontos flutuantes e o desempenho da decodificação foi avaliado em termos de Taxa de Erro de Bit (BER) e de Taxa de Erro de Quadro (FER). Para investigar o impacto do erro de casamento de

SNR, o desempenho de decodificação com relação ao recuo  $E_b/N_0$  também foi avaliado. Um codificador turbo de velocidade de 1/5 conforme fornecido por CDMA 2000 1xEV-DV foi utilizado e uma operação de Código Turbo Quase Complementar (QCTC) foi efetuada para converter uma velocidade de código geral para ser um valor outro que não 1/5. O tamanho de quadro é um dos tamanhos EP conforme definidos na especificação 1xEV-DV. O esquema de modulação utilizado foi BPSK e um canal AWGN foi suposto. Para a decodificação turbo, o número máximo de iterações de decodificação foi 8. BER e FER foram medidos ao executar as simulações até 50 erros de quadro terem sido produzidos.

[0071] O fator de ponderação  $W_f$  e o valor  $K'$  são definidos empiricamente. Como as iterações de decodificação para a decodificação turbo é geralmente uma operação de decodificação sub-ótima, não uma decodificação de probabilidade máxima, há uma probabilidade do desempenho ser degradado durante a decodificação iterativa. A simulação do erro de casamento do SNR revelou que o desempenho melhor é atingido para um recuo  $E_b/N_0$  de cerca de -1 dB para nenhum recuo  $E_b/N_0$ . Isto é porque a degradação do desempenho possivelmente gerado durante as iterações de decodificação é compensado pela ponderação de -1 dB errônea. Assim, o fator de ponderação  $W_f$  é dado empiricamente por

$$W_f \approx -1 \text{ dB} = 0,79432 = 1/2 + 1/4 + 1/16 \quad (9)$$

[0072] Ao representar  $W_f$  como a soma dos expoentes de 2, a multiplicação pelo fator de ponderação é implementada com facilidade no hardware.

[0073]  $K'$  na Equação (8) é o produto de uma inclinação  $K$  na Equação (4) e o fator de ponderação  $W_f \cdot K$  na Equação (6) é definido como a inclinação média de uma linha tangente da função  $l(x) = \log(1 - e^{-x})$ . Portanto,

$$-K = \frac{1}{a} \int_0^a \frac{1}{1 - e^{-x}} dx = \frac{\log(1 + e^{-a}) - \log 2}{a}$$

em que  $a$  é fixado a um valor significativo máximo. Se  $a$  for maior que cerca de 9,  $l(x)$  é inferior a  $10^{-1}$ . Assim, para  $a$  como 9,  $K$  é determinado por

$$K = \left. \frac{\log(1 + e^{-a}) - \log 2}{a} \right|_{a=9} = 0.7700 \approx \frac{1}{13}$$

(11)

[0074] Algumas simulações revelam que definir  $-K$  como a Equação (11) leva a um desempenho excelente.  $K'$  na Equação (11) também pode ser expresso como

$$K' = K \cdot W_f = 1/16 \quad (12)$$

$K'$  pode ser obtido simplesmente pela seleção de bit que é uma implementação de hardware simplificada da multiplicação.

[0075] Os resultados da simulação da aproximação e da não aproximação serão comparados com referência às Figuras 7 a 16. As Figuras 7 e 8 ilustram o desempenho de decodificação turbo em termos de BER e de FER para um tamanho EP de 3.864 e uma velocidade de código geral (R) de  $\frac{1}{2}$ . Nas Figuras 7 e 8, LogMAP denota o algoritmo Log-MAP, log2 MaxLogMAP denota o algoritmo Max-Log-MAP utilizando  $f_c$  definido como a função logarítmica  $l(x)$ , mod MaxLogMAP denota o algoritmo Max-Log-MAP utilizando  $f_c$  definido como uma função de primeira ordem aproximada, MaxLogMAP com FG

denota o algoritmo Max-Log-MAP com FG, e MaxLogMAP denota o algoritmo Max-Log-MAP existente. Como é ilustrado,  $\log 2$  MaxLogMAP é aproximado de LogMAP no desempenho de decodificação, mas este desempenho não é assegurado no caso de erro de casamento de SNR.  $\text{mod}$  MaxLogMAP desempenha cerca de 0,1 dB pior que LogMAP no desempenho de decodificação, mas este desempenho não é assegurado no caso de erro de casamento de SNR,  $\text{mod}$  MaxLogMAP desempenha meramente cerca de 0,1 dB pior que LogMAP a um FER DE  $10^{-2}$ , enquanto ele desempenha cerca de 0,5 dB melhor que MaxLogMAP com FG.  $\text{Mod}$  MaxLogMAP desempenha constantemente independentemente do erro de casamento de SNR.

[0076] As Figuras 9 e 10 ilustram os desempenhos BER e FER de  $\log 2$  MaxLogMAP,  $\text{mod}$  MaxLogMAP, MaxLogMAP com FG, e MaxLogMAP sobre iterações com  $E_b/N_0=1,3$  dB. É observado das Figuras 9 e 10 que  $\log 2$  MaxLogMAP tem o melhor desempenho sobre iterações.  $\text{mod}$  MaxLogMAP não desempenha melhor que MaxLogMAP com FG, mas alcança o desempenho FER de MaxLogMAP com FG sobre 8 iterações apenas com 7 iterações.

[0077] As Figuras 11 e 12 ilustram o desempenho BER e FER para um tamanho EP de 792 e uma velocidade de código efetiva de 1/5. De modo similar ao caso em que o tamanho EP é de 3.864, não há mudança nas classificações de desempenho dos cinco algoritmos. Ainda assim, comparado com o caso em que o tamanho EP é de 3.864,  $\text{mod}$  MaxLogMAP desempenha cerca de 0,1 dB melhor que MaxLogMAP com FG.

[0078] As Figuras 13 e 14 ilustram o desempenho BER e FER dos cinco algoritmos quando  $E_b/N_0=0,7$  dB e o tamanho EP=792, e as Figuras 15 e 16 ilustram o desempenho BER e FER dos cinco algoritmos para o EP de tamanho=3.864, a

velocidade de código efetiva=1/2, e o erro de casamento de SNR em  $E_b/N_0$  de 1,2 dB, isto é, quando erros equivalentes a um recuo  $E_b/N_0$  são gerados na estimação SNR de um símbolo de entrada de decodificador sob a suposição de que a estimação SNR perfeita é alcançada quando o recuo  $E_b/N_0$  é 0. Como é ilustrado, mod MaxLogMAP desempenha irrespectivamente do erro de casamento de SNR porque  $l(x)$  é aproximado a uma função de primeira ordem. Entretanto,  $\log_2$  MaxLogMAP exibe um desempenho variável de acordo com o erro de casamento de SNR porque  $l(x)$  é definido como uma função  $\log()$  não linear e  $f_c$  varia não linearmente dependendo da mudança de uma métrica na função  $\log()$ . Ainda assim, a variação  $f_c$  não é grande comparada com LogMAP. Portanto, até onde a estimação SNR dentro de cerca de -6 dB a +6 dB é garantida,  $\log_2$  MaxLogMAP pode ser utilizado como um algoritmo de decodificação turbo.

[0079] Observa-se das simulações que o algoritmo Max-Log-MAP modificado desempenha apenas cerca de 0,1 dB pior que o algoritmo Log-MAP independentemente do tamanho de EP, significando que este desempenho é melhor que aquele do algoritmo Max-Log-MAP (com ou sem FG). Apesar de alguns erros na estimação SNR de um símbolo de entrada, o algoritmo Max-Log-MAP modificado possui desempenho excelente independentemente dos erros de estimação SNR, que é aparente dos resultados da simulação.

[0080] Como foi descrito acima, o algoritmo Max-Log-MAP modificado desempenha melhor que o algoritmo Max-Log-MAP com pequena adição de hardware quando comparado com o algoritmo Max-Log-MAP e uma estrutura simplificada quando comparado com o algoritmo Log-MAP. Portanto, o algoritmo

Max-Log-MAP modificado é aplicável a um decodificador de canal em um terminal móvel para UMTS e HSDPA bem como um decodificador de canal para um sistema e terminal de CDMA2000 1xEV-DV. Ele desempenha vantajosamente e de modo excelente com uma estrutura simplificada.

[0081] Embora a invenção tenha sido mostrada e descrita com referência a certas versões da mesma, será compreendido por aqueles habilitados na tecnologia que várias mudanças na forma e nos detalhes poderão ser nela feitas sem desviar do espírito e escopo da invenção conforme definido pelas reivindicações apensas.

**REIVINDICAÇÕES**

1. Método para decodificar um código turbo em um sistema de comunicação digital, caracterizado pelo fato de compreender as etapas de:

(1) calcular uma melhor métrica e uma segunda melhor métrica das métricas que são somas das métricas de estado e uma métrica de ramo para um símbolo de informação recebido em uma treliça de decodificação turbo em um ponto no tempo;

(2) calcular uma diferença (601) entre uma melhor métrica para um símbolo de informação ser 0 e uma melhor métrica para o símbolo de informação ser 1;

(3a) calcular uma diferença (701) entre a melhor métrica e uma segunda melhor métrica para o símbolo de informação ser 0;

(3b) calcular uma diferença (702) entre a melhor métrica e uma segunda melhor métrica para o símbolo de informação ser 1;

(4) calcular a diferença (704) entre a diferença calculada na etapa (3a) e a diferença calculada na etapa (3b) e multiplicar a diferença calculada (704) calculada na etapa (4) por um fator de ponderação; e

(5) atualizar uma Razão de Verossimilhança Logarítmica (LLR) do símbolo de informação utilizando a diferença (601) calculada na etapa de (2) e um produto obtido na etapa de (4) e decidir um valor do símbolo de informação com base no LLR atualizado,

em que a melhor métrica é uma métrica máxima dentre as métricas para o símbolo de informação, e a segunda métrica é uma métrica máxima dentre as métricas restantes para o símbolo de informação, exceto para a melhor métrica.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de compreender ainda a etapa de calcular informação extrínseca com base no LLR atualizado, uma confiabilidade de uma Proporção Sinal a Ruído (SNR), e uma informação *a priori* do símbolo de informação.

3. Método, de acordo com a reivindicação 1, caracterizado pelo fato de o fator de ponderação ser expresso por

$$\text{Fator de ponderação} = k \cdot W_f$$

em que  $W_f$  é inferior a 1 e próximo de 1 e  $K$  é a inclinação média de linhas tangentes de uma função logarítmica  $l(x) = \log(1+e^{-x})$ .

4. Método, de acordo com a reivindicação 3, caracterizado pelo fato de  $W_f$  ser maior que 0,588235.

5. Método, de acordo com a reivindicação 3, caracterizado pelo fato de o fator de ponderação ser derivado de uma função linearizada com base na inclinação média das linhas tangentes da função logarítmica, a função logarítmica sendo representada pela diferença (704) calculada na etapa (4).

6. Método, de acordo com a reivindicação 3, caracterizado pelo fato de a inclinação média das linhas tangentes ser um inteiro entre 0 e 9.

7. Aparelho para decodificar um código turbo em um sistema de comunicação digital, caracterizado pelo fato de compreender:

um primeiro somador (601) para calcular uma diferença entre uma melhor métrica para um símbolo de informação ser 1 e uma melhor métrica para o símbolo de informação ser 0 em uma treliça de decodificação turbo em um ponto no tempo;

um segundo somador (602) para somar informação de transmissão e informação *a priori* do símbolo de informação;

um terceiro somador (603) para calcular uma diferença entre as saídas do primeiro e do segundo;

um primeiro multiplicador (604) para multiplicar a saída do terceiro somador por um primeiro fator de ponderação;

uma calculadora de valor de correção compreendendo um quinto somador (701) para calcular uma diferença entre a melhor métrica e a segunda melhor métrica para o símbolo de informação ser 0 e um sexto somador (702) para calcular a diferença entre a melhor métrica e a segunda melhor métrica para o símbolo de informação ser 1, em que a calculadora de valor de correção calcula um valor de correção utilizando uma diferença entre a saída do quinto somador e a saída do sexto somador; e

um quarto somador (605) para somar o valor de correção à saída do primeiro multiplicador,

em que a melhor métrica é uma métrica máxima dentre as métricas para o símbolo de informação, e a segunda métrica é uma métrica máxima dentre as métricas restantes para o símbolo de informação, exceto para a melhor métrica.

8. Aparelho, de acordo com a reivindicação 7, caracterizado pelo fato de a calculadora de valor de correção compreender:

uma tabela de pesquisa (703) para armazenar valores de correção para as saídas do quinto e do sexto somadores;

um sétimo somador (705) para calcular uma diferença entre os valores de correção;

um segundo multiplicador (707) para multiplicar a saída do sétimo somador por um segundo fator de ponderação;

um oitavo somador (704) para calcular uma diferença entre as saídas do quinto e do sexto somadores;

um terceiro multiplicador (706) para multiplicar a saída do oitavo somador por um valor de inclinação de uma função linear aproximada de uma função logarítmica; e

um seletor (708) para selecionar uma das saídas do segundo e do terceiro multiplicadores com base em uma confiabilidade da proporção sinal a ruído (SNR) do símbolo de informação.

9. Aparelho, de acordo com a reivindicação 8, caracterizado pelo fato de se o segundo fator de ponderação e o valor de inclinação da função linear puderem ser expressos como expoentes de 2, cada um dos primeiro, segundo e terceiros multiplicadores é implementado como um seletor de bit.

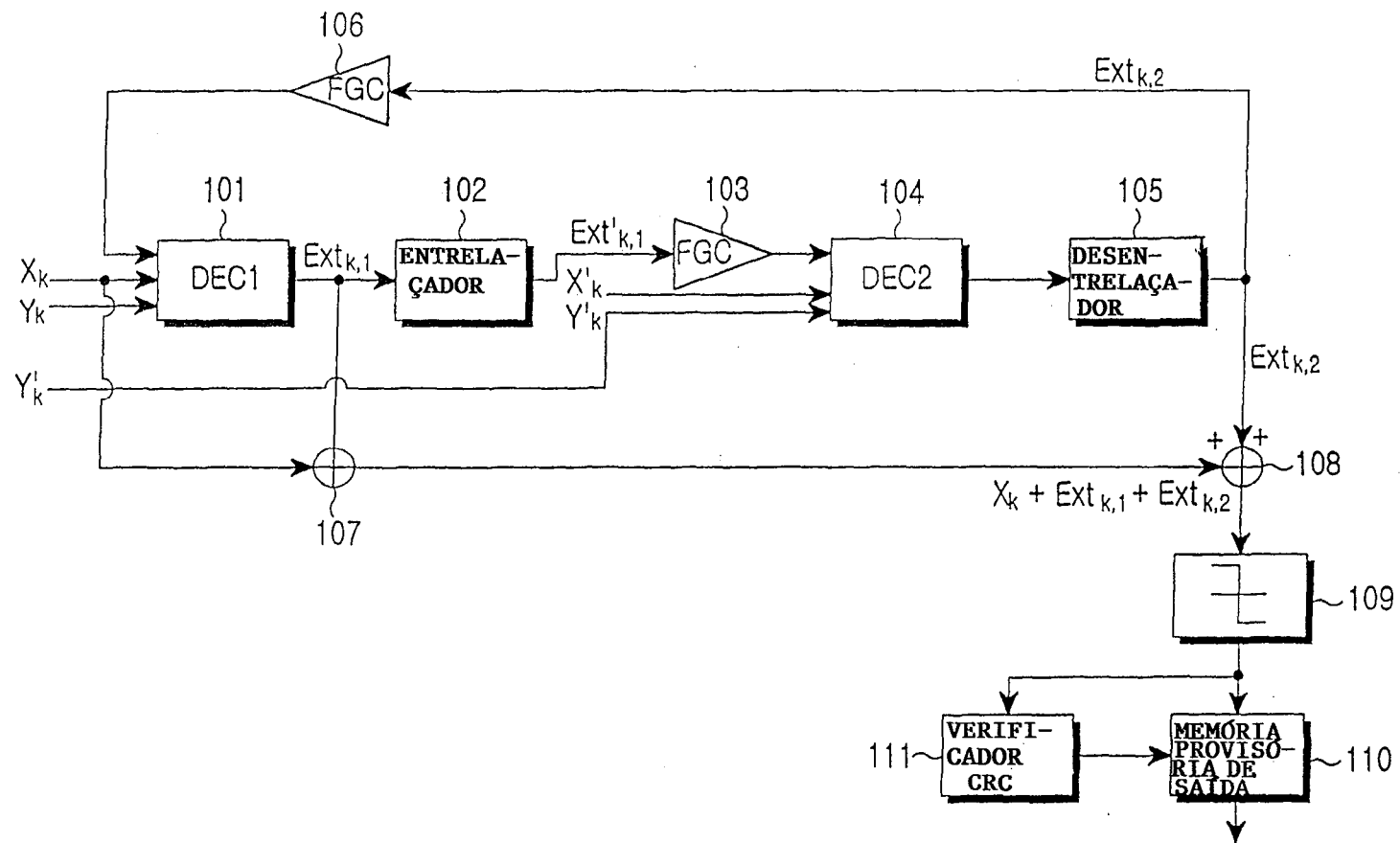


FIG. 1

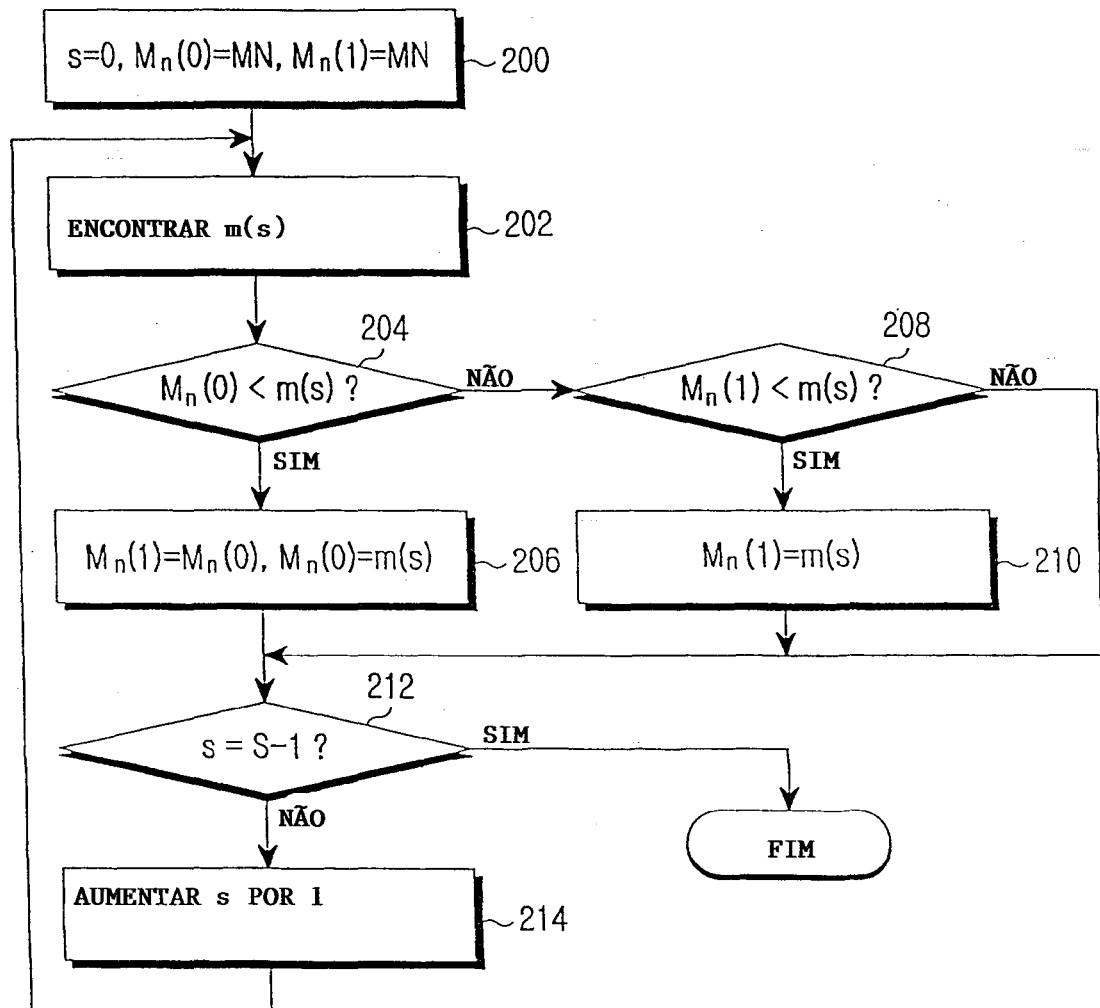


FIG.2

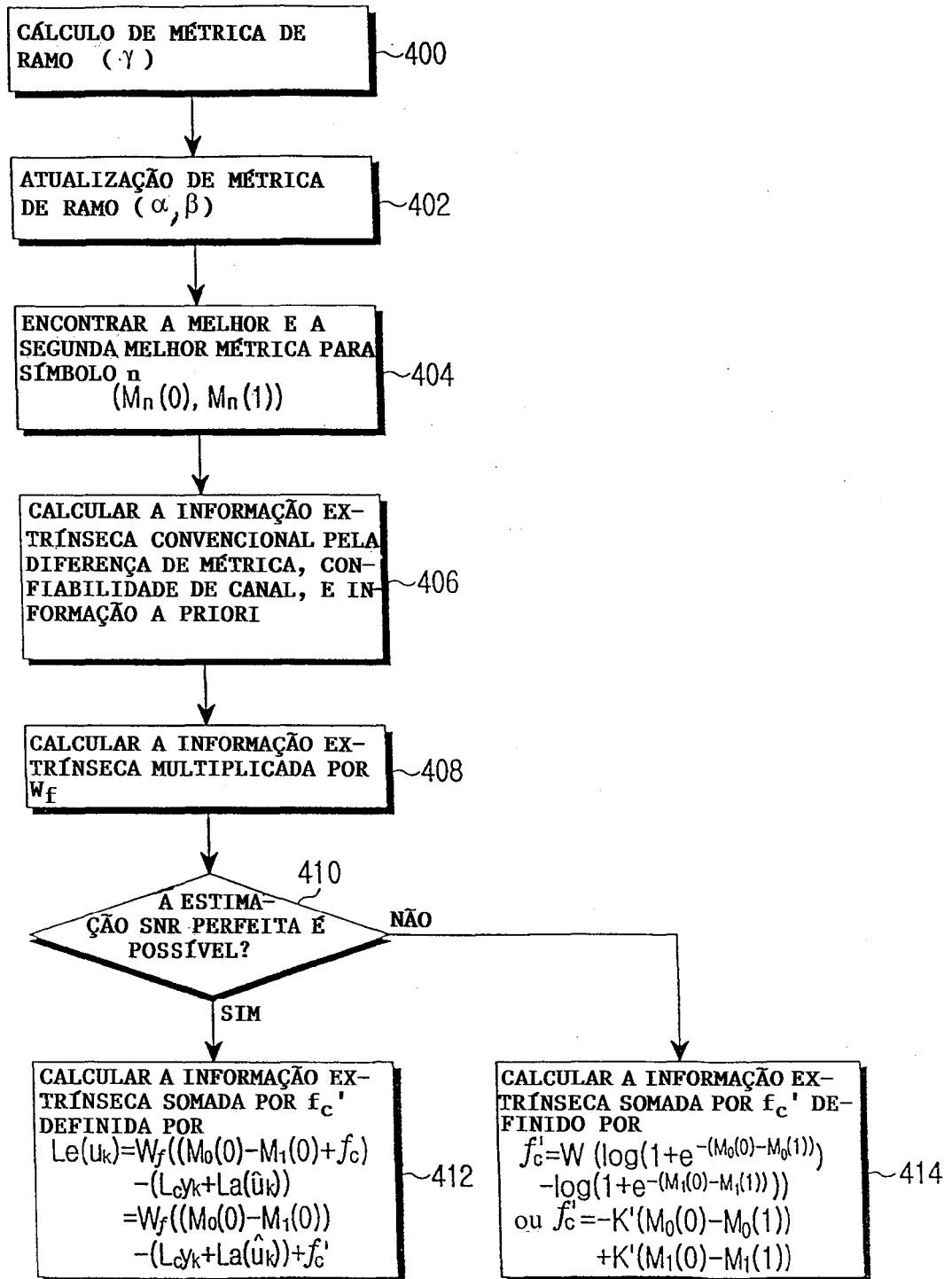


FIG.3

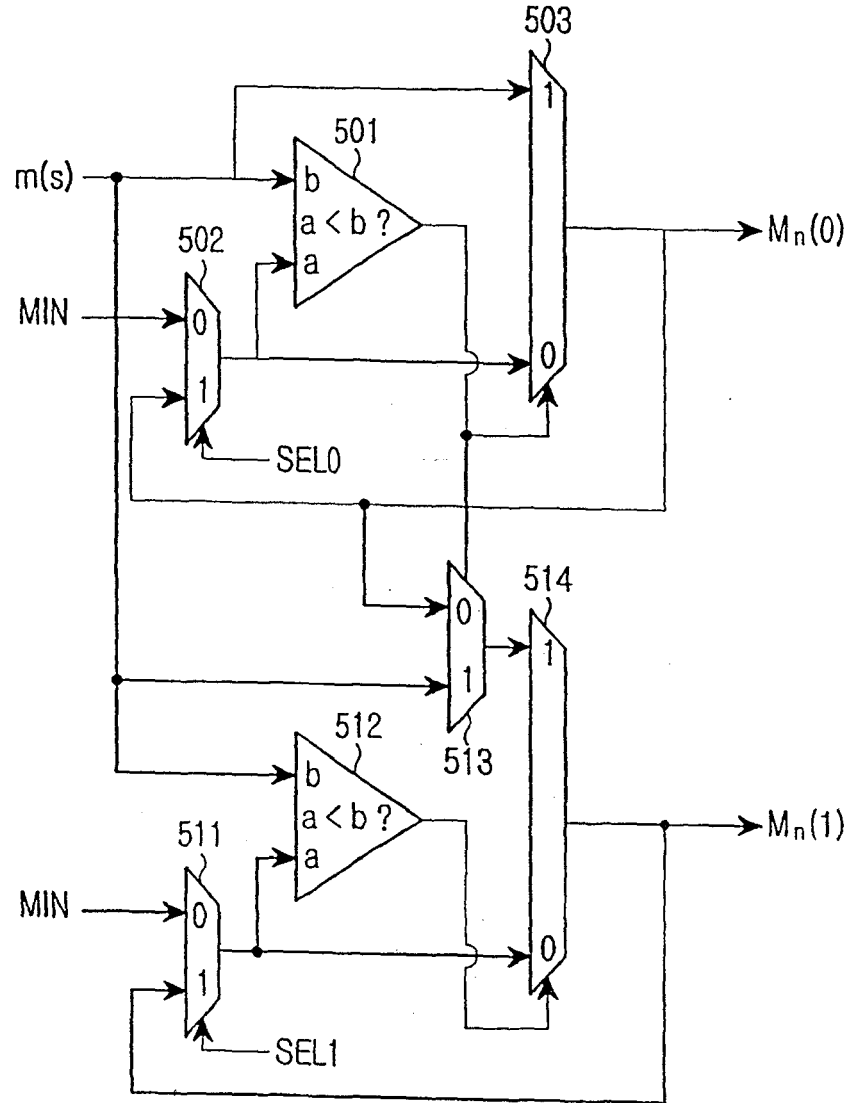


FIG. 4

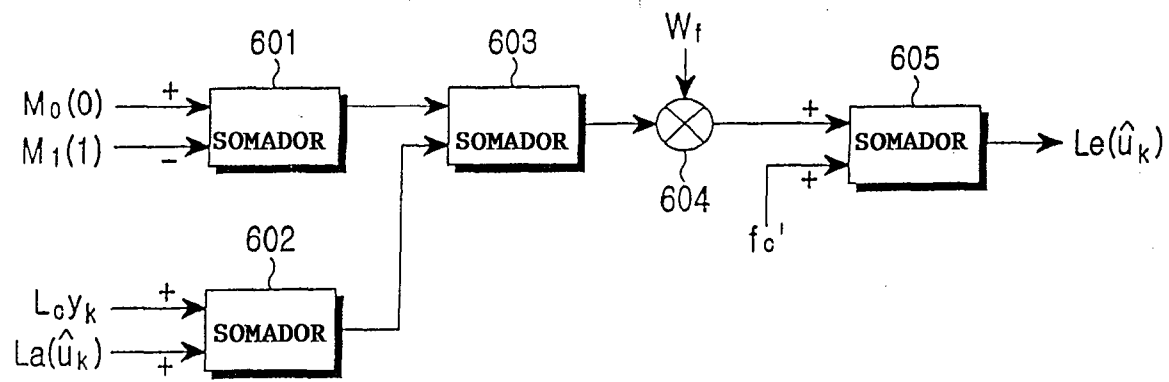


FIG.5

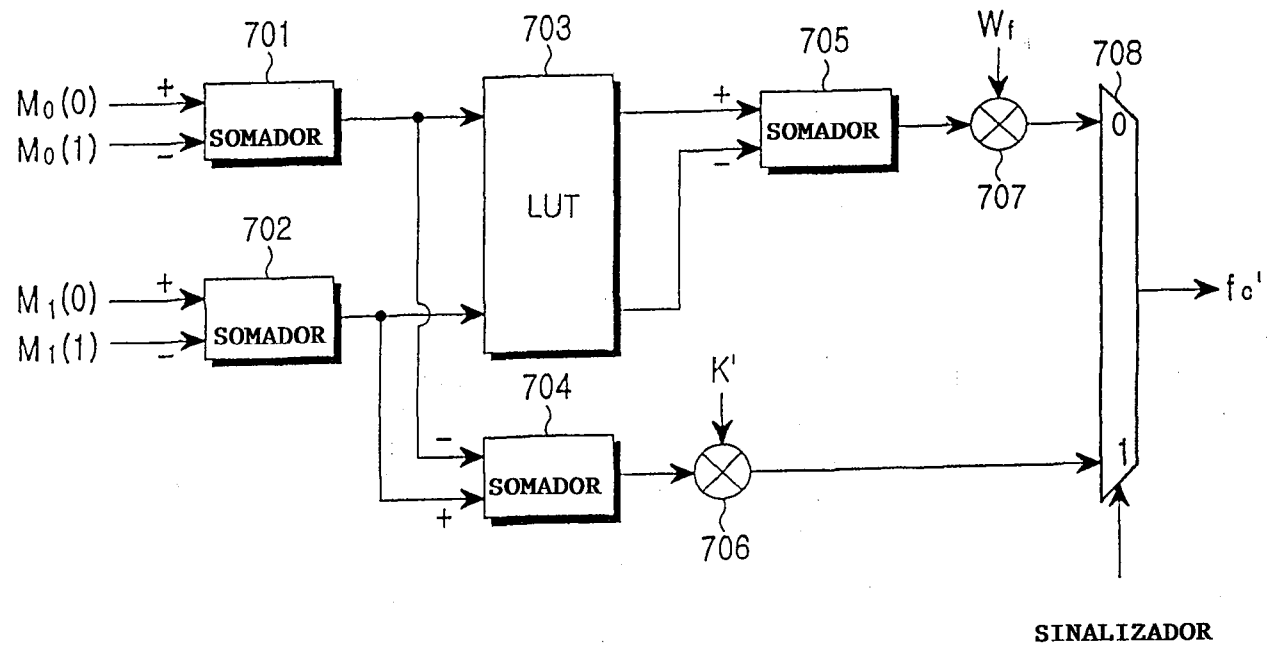
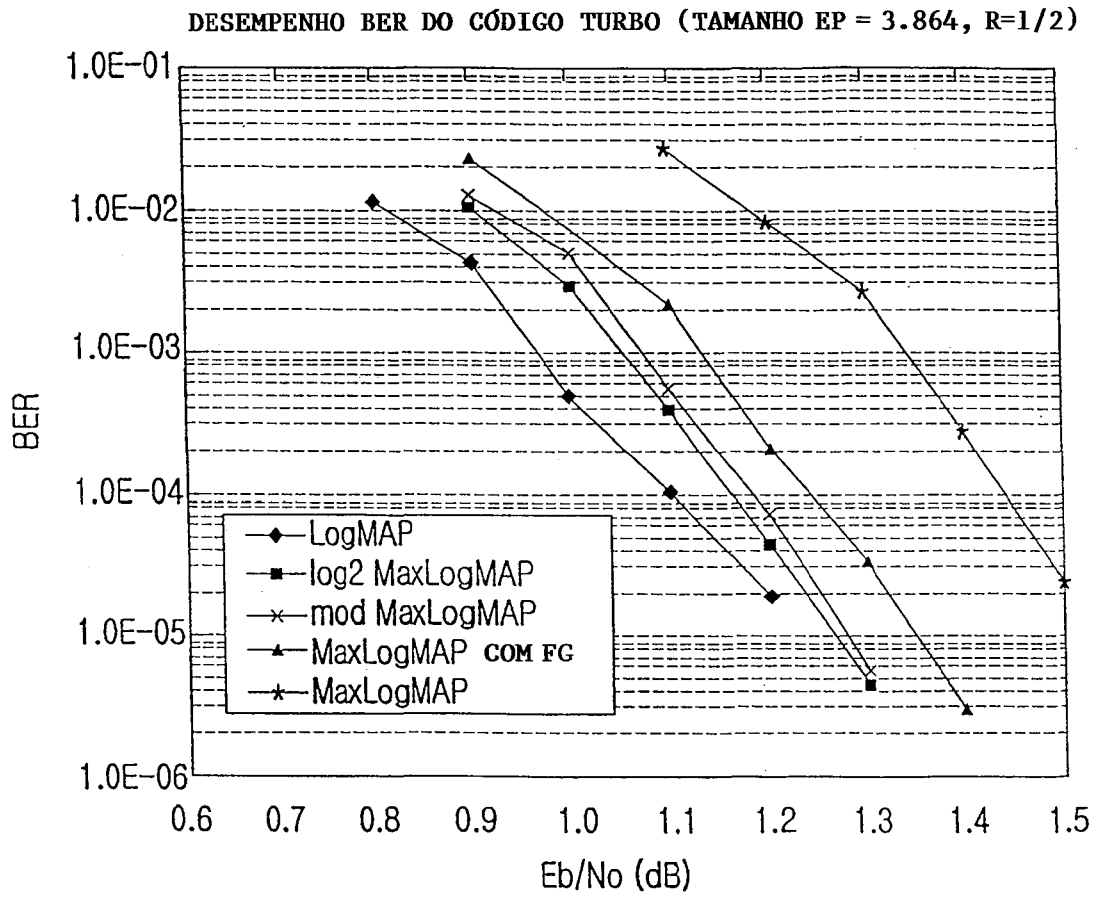
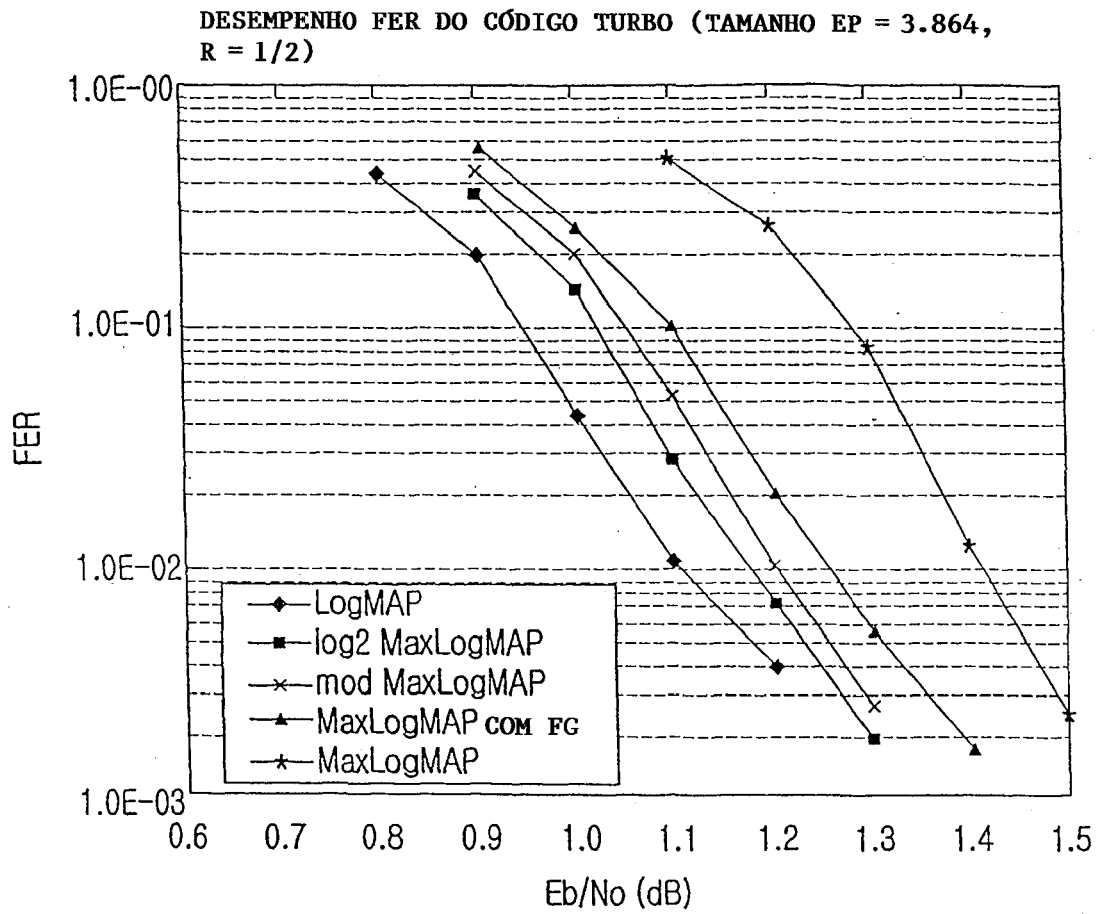


FIG. 6



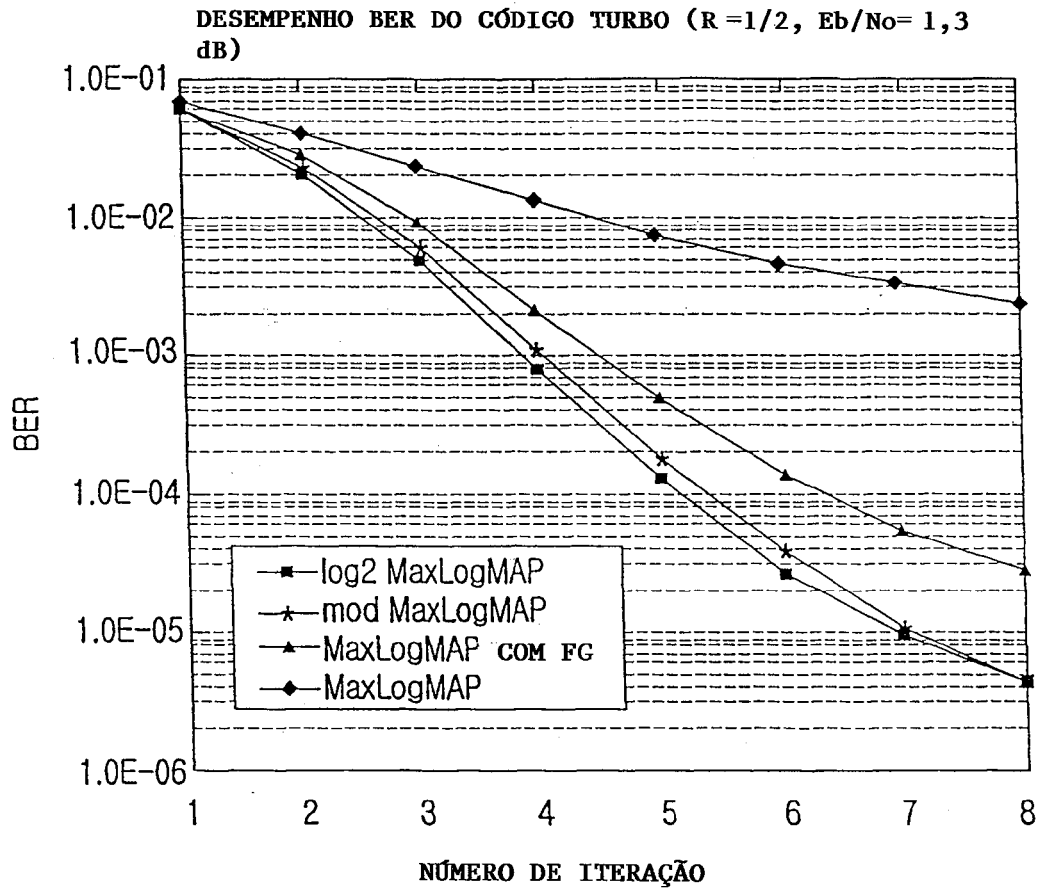
DESEMPENHO BER DO CÓDIGO TURBO (TAMANHO EP = 3.864, R=1/2)

FIG.7



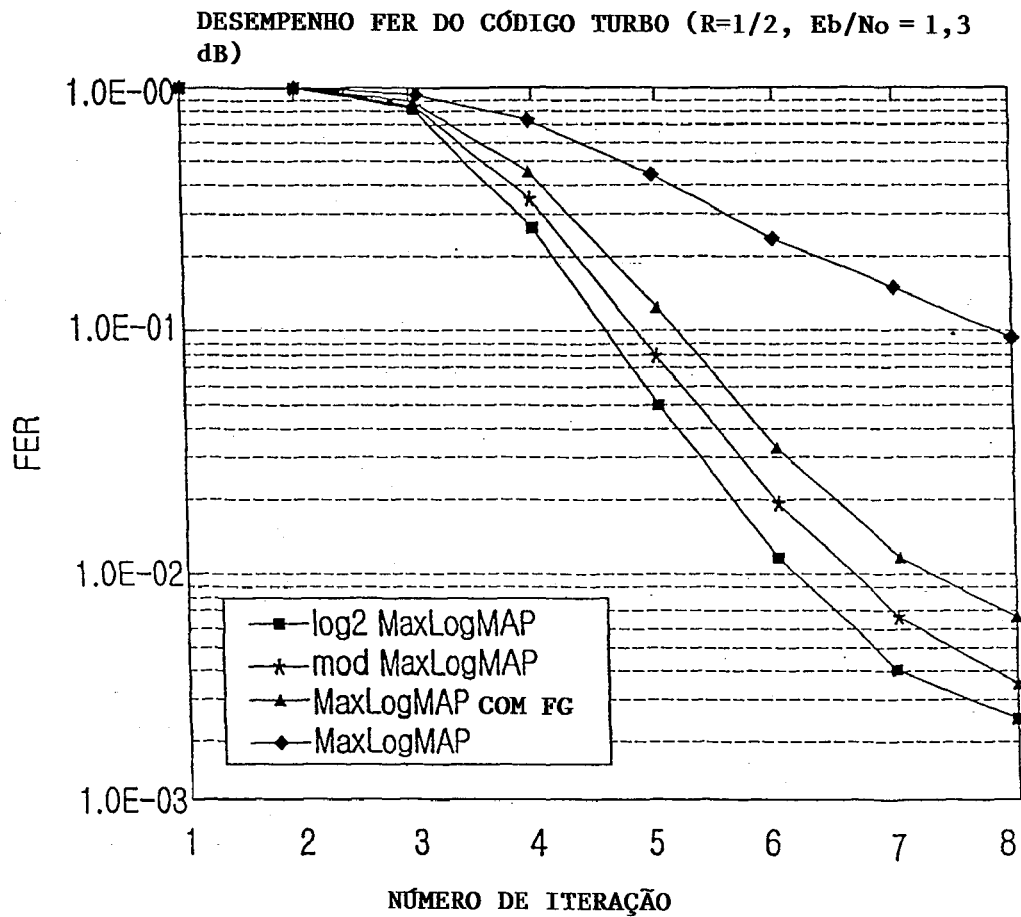
DESEMPENHO FER DO CÓDIGO TURBO (TAMANHO EP = 3.864,  
R = 1/2)

FIG.8



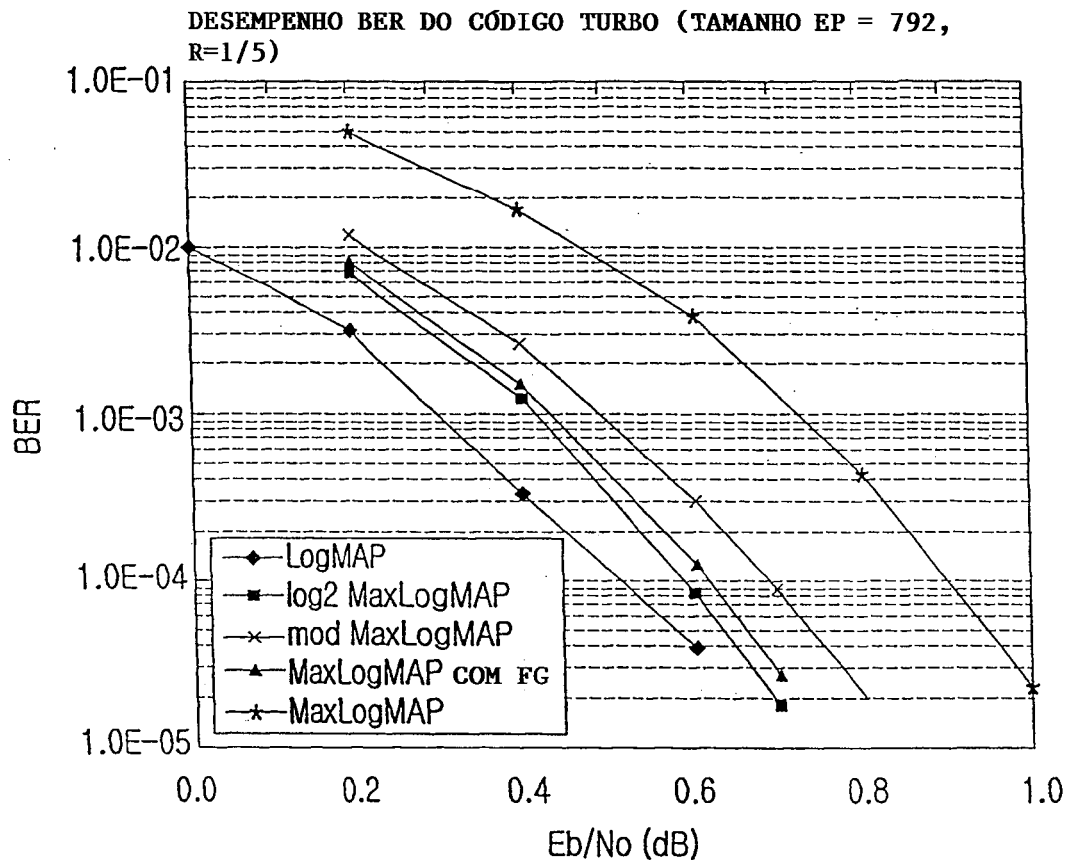
DESEMPENHO BER DO CÓDIGO TURBO POR ITERAÇÕES (TAMANHO EP = 3.864,  $R=1/2$ ,  $E_b/N_0=1,3$  dB)

FIG.9



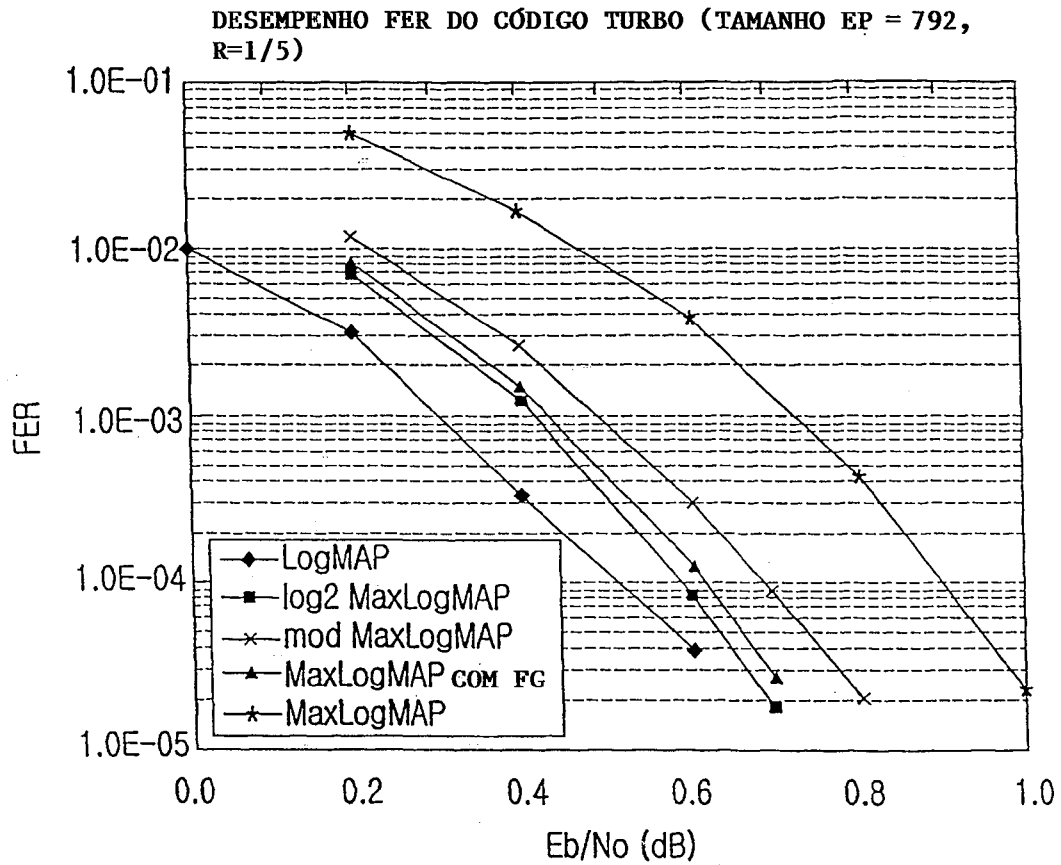
DESEMPENHO FER DO CÓDIGO TURBO POR ITERAÇÕES (TAMANHO EP = 3.864, R=1/2, Eb/No = 1,3 dB)

FIG.10



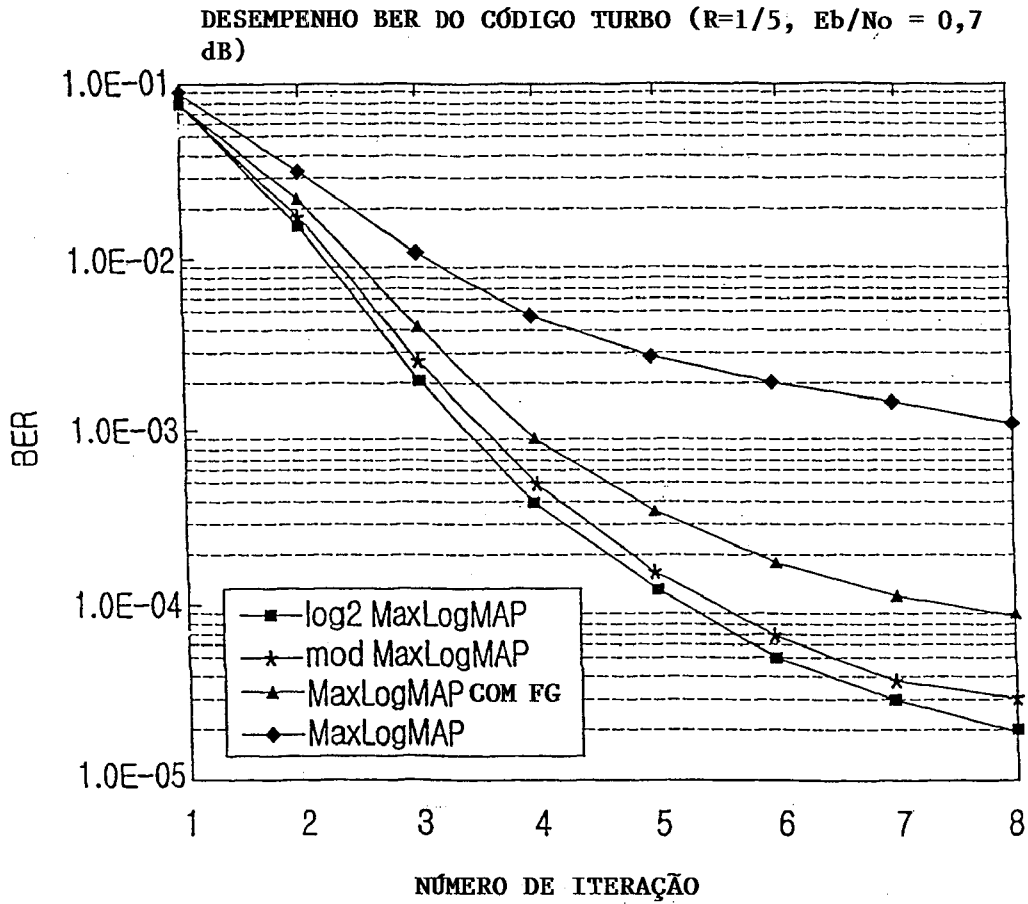
DESEMPENHO BER DO CÓDIGO TURBO (TAMANHO EP = 792,  
R=1/5)

FIG.11



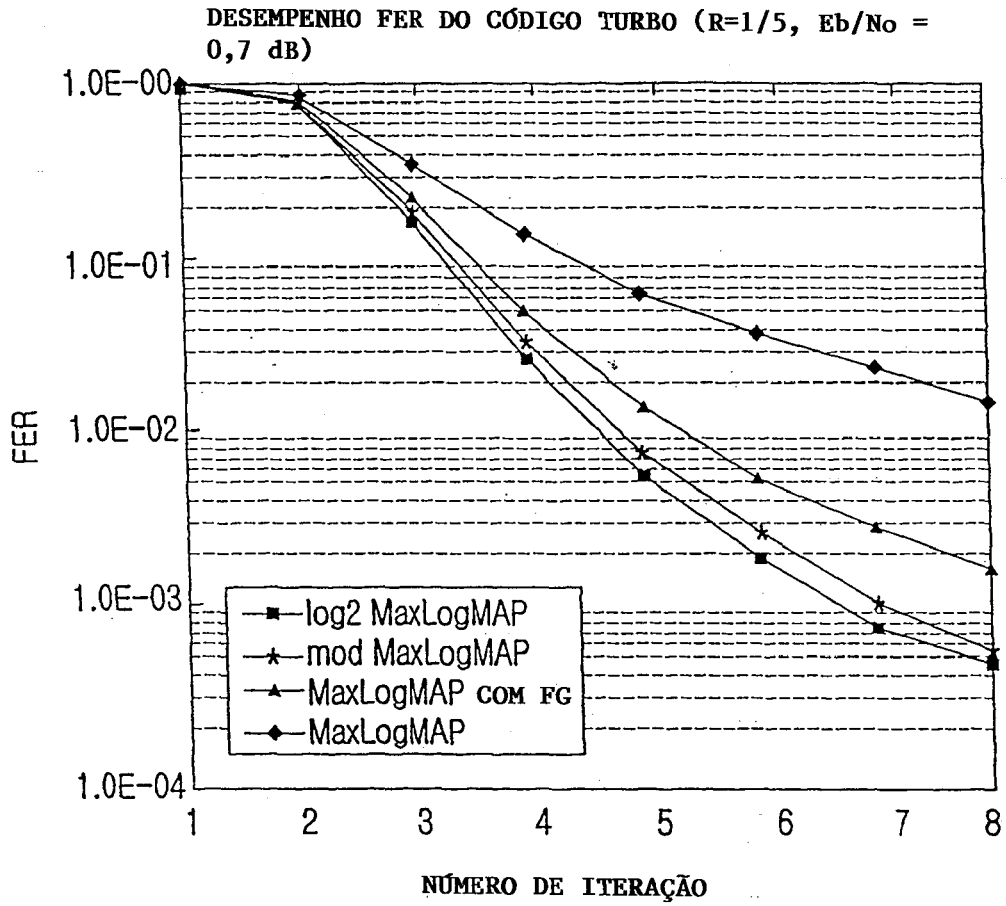
DESEMPENHO FER DO CÓDIGO TURBO (TAMANHO EP = 792,  
R=1/5)

FIG.12



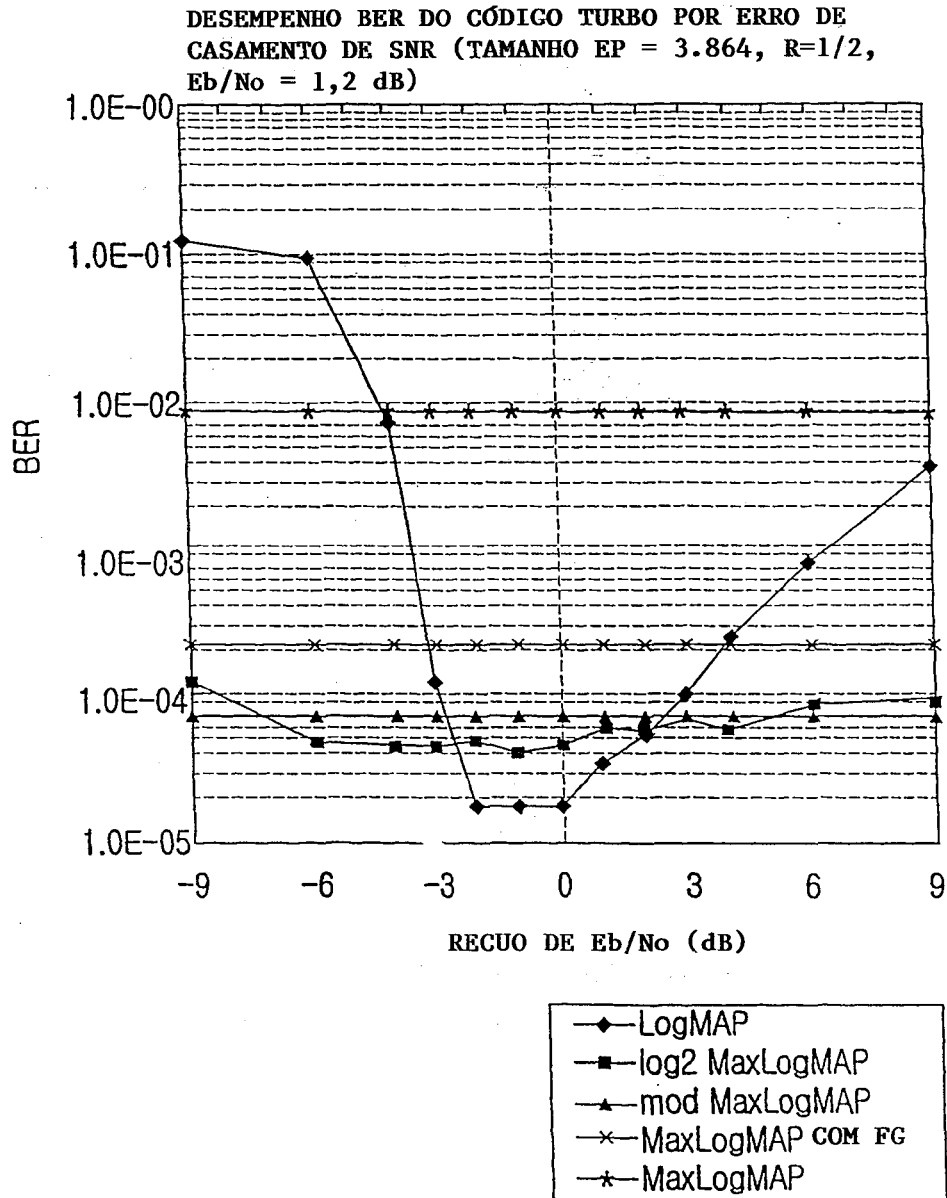
DESEMPENHO BER DO CÓDIGO TURBO POR ITERAÇÕES  
(TAMANHO EP = 792,  $R=1/5$ ,  $E_b/N_0 = 0,7$  dB)

FIG.13



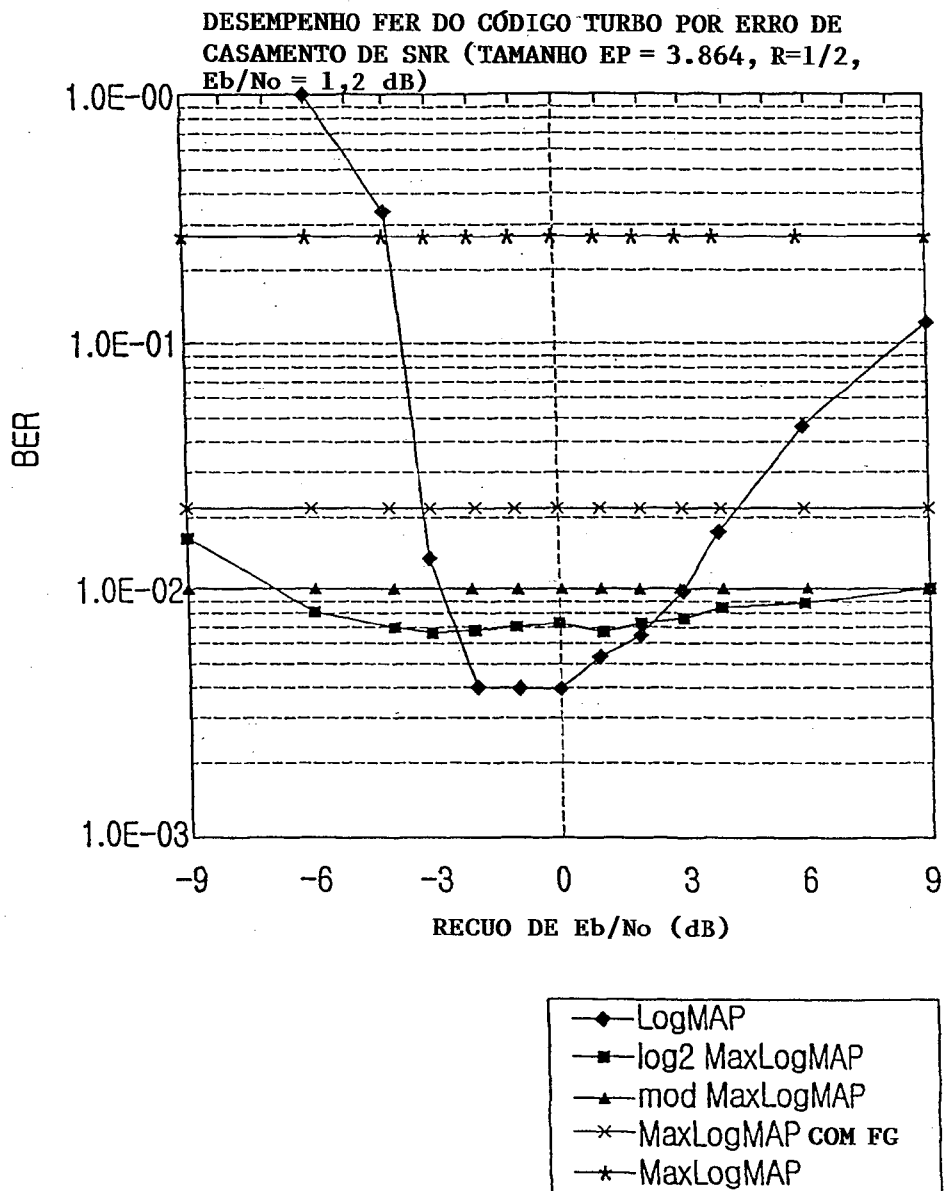
DESEMPENHO FER DO CÓDIGO TURBO POR ITERAÇÕES  
(TAMANHO EP = 792, R=1/5, Eb/No = 0,7 dB)

FIG.14



DESEMPENHO BER DO CÓDIGO TURBO SOBRE RECULO DE Eb/No (TAMANHO EP = 3.864, Eb/No = 1,2 dB)

FIG.15



**DESEMPENHO FER DO CÓDIGO TURBO SOBRE RECULO DE Eb/No (TAMANHO EP = 3.864, Eb/No = 1,2 dB)**

FIG.16