Bibliografia

[1] R. J. Anderson and F. A. Petitcolas, "Information hiding: an annotated bibliography" *[online]. Available: www.petitcolas.net/fabien/steganography/bibliography/*, 1999.

[2] M. D. Swanson, M. Kobayashi,and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies" *Proc. IEEE*, v. 86, p. 1064-1087, June 1998.

[3] M. Barni, F. Bartolini and J. Fridrich, "EDITORIAL - EURASIP Journal of applied signal processing" *[online]. Available: http://www.hindawi.co.uk/open-access/asp/volume-2002/S1110865702001981.pdf*, 2002.

[4] B. Chen and G. W. Wornell, "Quantization Index Modulation: A class of provably good methods for digital watermarking and information embedding" *IEEE Trans. Inform. Theory*, v. 47, No. 4, p. 1423-1443, May 2001.

[5] F. Perez-Gonzàlez, F. Balado, and J. R. Hernández, "Performance analysis of existing and new methods for data hiding with known-host information in additive channels" *IEEE Trans. on Signal Processing*, v. 51, No. 4, p. 960-980, April 2003. Special Issue on Signal Processing for Data Hiding in Digital Media & Secure Content Delivery.

[6] J. Eggers and B. Girod, "Informed Watermarking" *Kluwer Academic Publishers*, 2002, ISBN 1-4020-7071-3.

[7] M. H. M. Costa, "Writing on dirty paper" *IEEE Trans. Inform. Theory*, v. IT-29, p. 439-441, May 1983.

[8] I. J. Cox, J. Killian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for mutimedia" *IEEE Trans. Image Processing*, v. 6, p. 1673-1687, Dec 1997.

[9] H. S. Malvar and D. A. F. Florêncio, "Improved Spread Spectrum: A new modulation Technique for robust watermarking" *IEEE Trans on Signal Processing*, v. 51, No. 4, p. 898-905, April 2003.

[10] F. Perez-Gonzàlez and F. Balado, "Quantized projection data hiding" *In Proc. of the IEEE International Conference on Image Processing (ICIP)*, Rochester (NY), USA, September 2002.

[11] B. Chen and G. W. Wornell, "Implementations of Quantization Index Modulation methods for digital watermarking and information embedding of multimedia" *J. VLSI Signal Processing Syst. Signal, Image, and Video Technol. (Special Issue on Multimedia Signal Processing)*, v. 27, p. 7-33, Feb 2001.

[12] S. Benedetto and E. Biglieri, "Principles of Digital Transmission with Wireless Applications" *Kluwer Academic / Plenum Publishers*, 1999, ISBN 0-306-45753-9.

[13] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes" *in ICC'93*,Geneva, Switzerland,p. 1064-1070, May 1993.

[14] B. Vucetic, "Turbo Codes Principles and Applications," *Kluwer Academic Publishers*, 2002, ISBN 0-7923-7868-7.

[15] L. R. Bahl, J. Cocke, F. Jelineck, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *in IEEE Trans. Inform. Theory*, vol. IT-20,pp. 284-287, 1974.

[16] J. Hagenauer and P. Hoeher, "A Viterbi algorith with soft-decision outputs and its applications" *in Proc. IEEE GLOBECOM*,Dallas, TX, pp. 47.1.1-47.1.7, 1989.

[17] R. Johannesson and K. S. Zigangirov, "Fundamentals of Convolutional Coding" *Wiley-IEEE Press*, March 1999, ISBN 0-7803-3483-3.

[18] J. Fridrich, "Methods for Tamper Detection in Digital Images," *ACM Workshop on Multimedia and Security*, Orlando, FL, October 30-31, 1999, pp. 19-23.

A
Codificador Turbo

A.1
Codificação Turbo

A.1.1
Introdução

Nesta seção pretende-se identificar os principais componentes que afetam as características e o desempenho dos codificadores/decodificadores turbo. O entendimento dos principais conceitos deste tipo de codificador permitirá a análise de emprego do mesmo em aplicações de marcação d'água digital e eventuais simulações. Para um maior detalhamento do assunto indicamos ao leitor consultar a referência [14].

A.1.2
Codificador Turbo

O modelo do codificador turbo é representado basicamente pela concatenação em paralelo de codificadores convolucuinais sistemáticos recursivos (RSC), denominados de codificadores componentes do código turbo, conforme ilustrado na fig. A.1.

A entrada do codificador é representada pelo vetor $\boldsymbol{c}$ de comprimento L, cujos componentes são os bits de informação. A saída do codificador turbo é composta pelos vetores $\boldsymbol{v_0}$, $\boldsymbol{v_1}$ e $\boldsymbol{v_2}$. O vetor $\boldsymbol{v_0}$ é idêntico ao vetor $\boldsymbol{c}$, e seus bits são denominados de sistemáticos. Os vetores $\boldsymbol{v_1}$ e $\boldsymbol{v_2}$ são saídas de codificadores RSC idênticos (codificadores componentes do codificador turbo), e os correspondentes bits codificados são denominados bits de paridade.

Um dos codificadores RSC componentes é precedido por um *interleaver* de comprimento L que tem a função de embaralhar a seqüência (bloco) de

entrada. Este embaralhamento é fundamental não só para a correção de erros tipo "burst", em virtude da descorrelação proporcionada entre $\boldsymbol{v_2}$ e $\boldsymbol{c}$, mas também para garantir a descorrelação entre os vetores de paridade $\boldsymbol{v_1}$ e $\boldsymbol{v_2}$, saídas de codificadores RSC idênticos. Esta última descorrelação é propriedade fundamental para o adequado funcionamento (convergência) do algoritmo de decodificação turbo que será visto adiante.

A.1.3 Codificador RSC de Referência

Para efeitos práticos de entendimento e eventual simulação, utilizaremos sempre que possível como codificador componente o mesmo codificador RSC utilizado por Berrou [13] como referência, representado na fig. A.2.

Este codificador RSC possui memória de ordem 4 e taxa $R = 1/2$, e é derivado do código convolucuinal que possui geradores $G_1 = (1, 1, 1, 1, 1)$ e $G_2 = (1, 0, 0, 0, 1)$, ou equivalentemente na representação octal, $G_1 = 37$ e $G_2 = 21$, respectivamente. A recursividade do codificador reveste-o da importante característica IIR (*infinite impulse response*), que é fundamental na distribuição de pesos das palavras códigos geradas, afetando assim diretamente o desempenho do codificador.

Desta forma, empregando ao codificador turbo, apresentado na fig. A.1, o codificador RSC componente da fig. A.2, pode-se detalhar a representação do codificador turbo conforme ilustrado na fig. A.3.

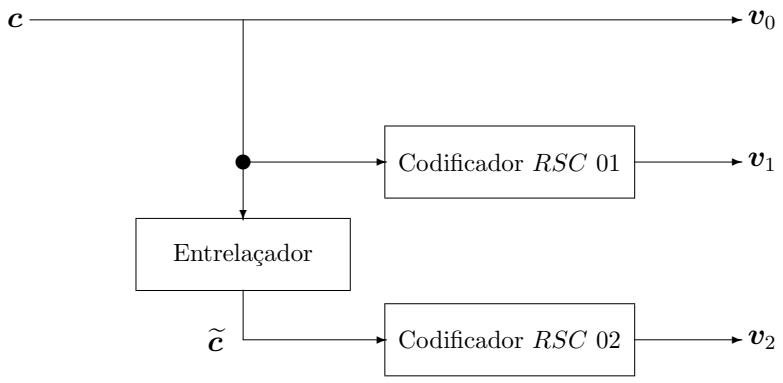

Figura A.1: Modelo do codificador turbo.

Neste caso, a taxa do codificador turbo resultante é $R = 1/3$. Aplicando a técnica de "puncturing" nos codifcadores componentes, de forma que a taxa dos mesmos seja $R = 2/3$, obtém-se um codificador turbo de taxa $R = 1/2$.

É importante comentar que o um codificador RSC define um diagrama de estados e uma correspondente treliça. As probabilidades de transição de estado e de saída (ramos da treliça) destas representações são base para os cálculos envolvidos nos algoritmos de decodificação, assim como as probabilidades de transição de canal, o qual a informação codificada é transmitida. O canal AWGN é normalmente considerado para análise, que equivalentemente no sistema de marcação d'água digital pode também ser considerado como o modelo de ataque.

A.1.4
Decodificação de Códigos RSC

A intenção desta seção não é descrever o detalhamento completo dos algoritmos de decodificação de códigos RSC, mas sim evidenciar seus principais conceitos e aspectos para compreensão do algoritmos de decodificação iterativa de códigos turbo que será visto na próxima sub-seção.

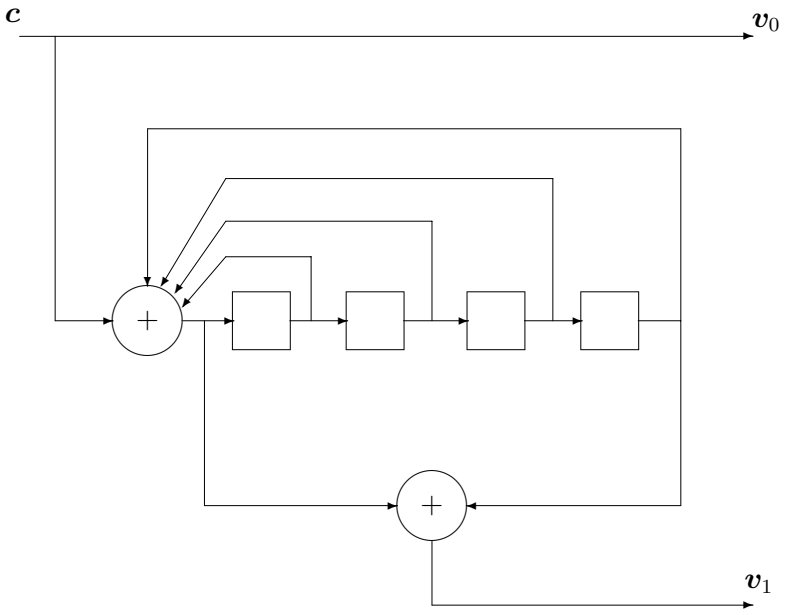

Figura A.2: Codificador RSC.

O objetivo do algoritmo de decodificação é calcular ou estimar a razão de verosemelhança (*loglikehood ratio* - LLR)do bit c_l:

$$\Lambda(c_l) = \log \frac{Pr\{c_l = 1|\boldsymbol{r}\}}{Pr\{c_l = 0|\boldsymbol{r}\}} \tag{A-1}$$

onde $\boldsymbol{r}$ representa o vetor recebido correspondente aos vetor $\boldsymbol{v} = (\boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2})$ transmitido. Assim, uma vez calculado o valor desta razão, é possível realizar a estimativa $\hat{c}_l$ do bit c_l como:

$$\hat{c}_l = \begin{cases} 1, & \Lambda(c_l) \geqslant 0; \\ 0, & \Lambda(c_l) < 0. \end{cases} \tag{A-2}$$

Esta estimativa é denominada de "hard". O valor $\Lambda(c_l)$ é denominado de

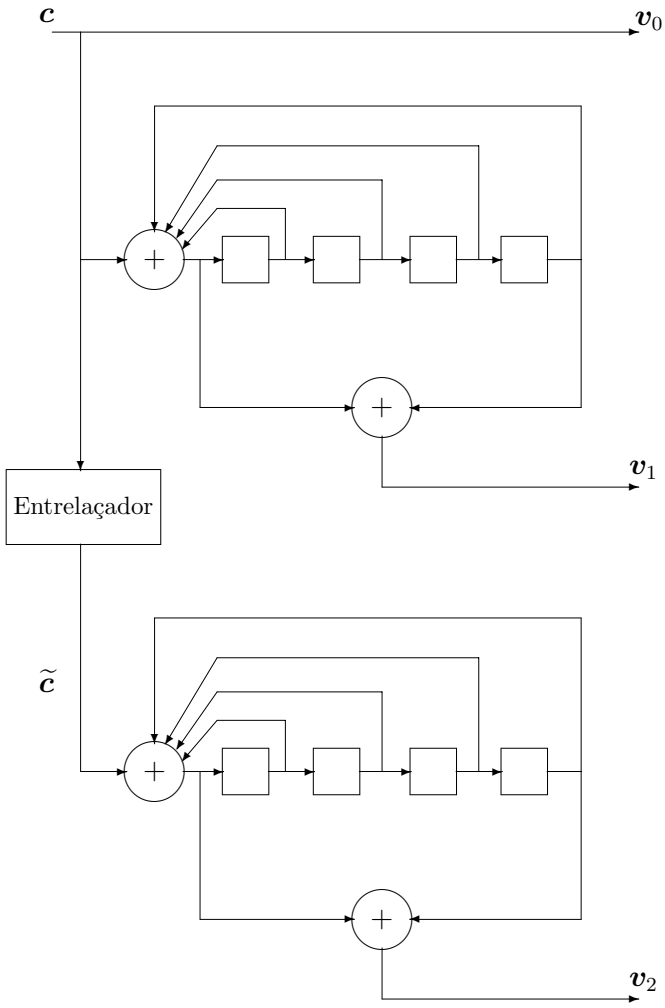

Figura A.3: Modelo do codificador turbo empregando o codificador RSC componente de referência.

"Software Information" (SI) do processo de decodificação.

Os principais algoritmos já desenvolvidos para cálculo de $\Lambda(c_l)$ são o MAP (*"Maximum a Posteriori"*) [15] e o SOVA (*"Software Output Viterbi Algorithm"*) [16], e suas derivações e aproximações. O algoritmo MAP é preciso, contudo é mais complexo e dependente de precisa caracterização do canal.

O primeiro passo importante para a decodificação iterativa, dada por Berrou [13], foi identificar componentes aditivas em $\Lambda(c_l)$ de características distintas. $\Lambda(c_l)$ pode ser decomposta como a adição de três parcelas:

$$\Lambda(c_l) = \Lambda_a + \Lambda_c + \Lambda_e \tag{A-3}$$

Λ_a é a componente função das probabilidades a priori dos bits transmitidos.

$$\Lambda_a = \log \frac{Pr\{c_l = 1\}}{Pr\{c_l = 0\}} \tag{A-4}$$

Para uma distribuição equiprovável obtém-se Λ_a=0. Λ_c é função dos bits sistemáticos recebidos pelo decodificador e da característica do canal, e a componente Λ_e é a componente extrínseca (informação extrínseca) que é função dos bits de redundância recebidos das probabilidades a priori. A conceitual descorrelação de Λ_e com os bits sistemáticos recebidos e também com os bits redundantes recebidos do correspondente codificador RSC, de um codificador turbo, possibilitou a implementação da decodificação iterativa, e o desenvolvimento do conceito SISO ("Soft Input - Soft Output") para a decodificação turbo.

A.1.5 Decodificação Turbo

O decodificador turbo é composto basicamente por dois módulos idênticos de decodificadores RSC em uma configuração iterativa, conforme representado na Figura A.4. Os interleavers são necessários para ordenar componentes das seqüências (blocos) de entrada de cada módulo de decodificação, conforme o embaralhamento realizado, ou não, na entrada dos codificadores componentes do codificador turbo.

Onde $\boldsymbol{r_0}$, $\boldsymbol{r_1}$ e $\boldsymbol{r_2}$ representam os vetores recebidos correspondentes aos vetores $\boldsymbol{v_0}$, $\boldsymbol{v_1}$ e $\boldsymbol{v_2}$ transmitidos, RSC_1 e RSC_2 representam os módulos

de decodificação correspondentes aos codificadors componentes do codificador turbo, Λ_{1e} e Λ_{2e} representam os valores da informação extrínseca obtidos na saída dos módulos de decodificação RSC_1 e RSC_2, respectivamente, e Λ_1 e Λ_2 representam os valores da LLR ($\Lambda(c_l)$) obtidos na saída dos módulos de decodificação RSC_1 e RSC_2, respectivamente.

O segundo passo importante dado por Berrou [13] foi propor a atualização das probalilidades a priori na entrada de cada módulo de decodificação, em um processo iterativo, obtida considerando $\Lambda_{2a} = \Lambda_{1e}$, para o segundo decodificador, e $\Lambda_{1a} = \Lambda_{2e}$, no primeiro decodificador. Como se percebe da equação A-4, da atualização de Λ_a obtém-se diretamente um refinamento das probabilidades a priori. Após um determinado número de iterações, estima-se o bit c_l a partir de Λ_2, conforme equação A-2.

A convergência desta algoritmo ainda não foi demonstrada teoricamente, mas experimentalmente já é largamente empregado, inclusive em aplicações

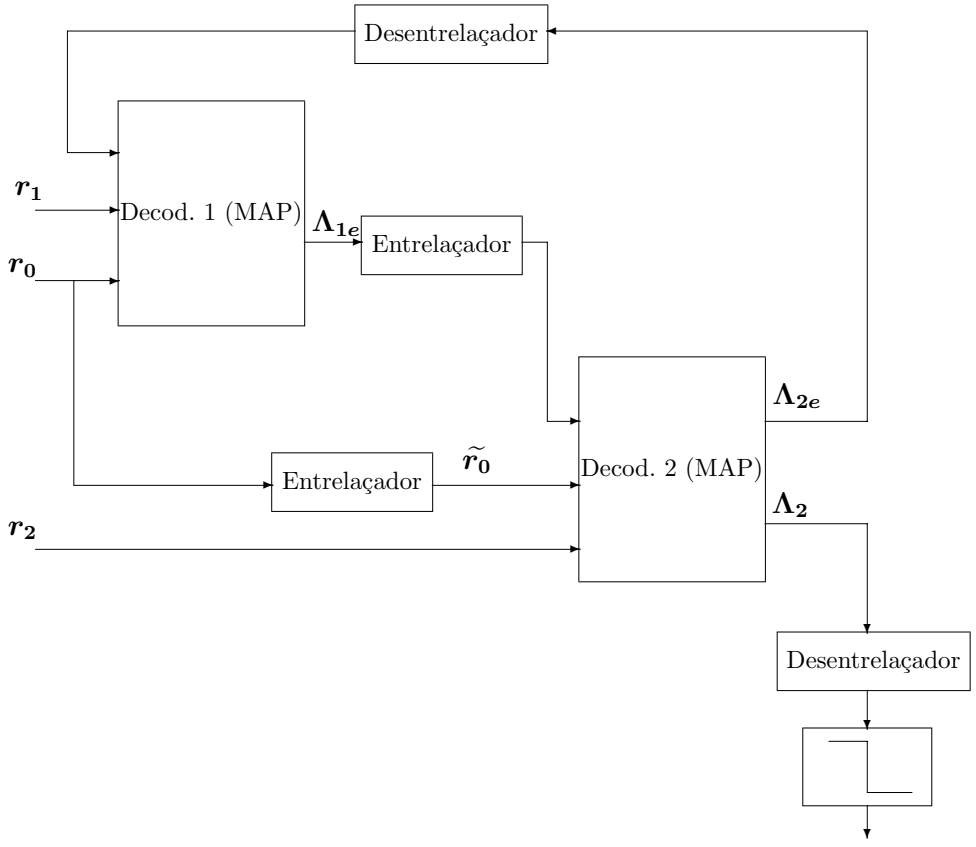

Figura A.4: Decodificador turbo.

críticas. O número de iterações dependerá dos diversos parâmetros do codificador e do canal. Vale comentar que o conceito do algoritmo turbo de decodificação iterativa pode ser estendido para o caso de codificadores componentes não idênticos, e codificadores componentes concatenados em série.

A.1.6
Ganho de Codificação

Na fig. A.5 está traçada a curva de desempenho (p_e x E_b/N_0) para uma implementação de referência (Figura A.3) do código turbo, juntamente com a curva de desempenho de um sistema de comunicação BPSK sem codificação.

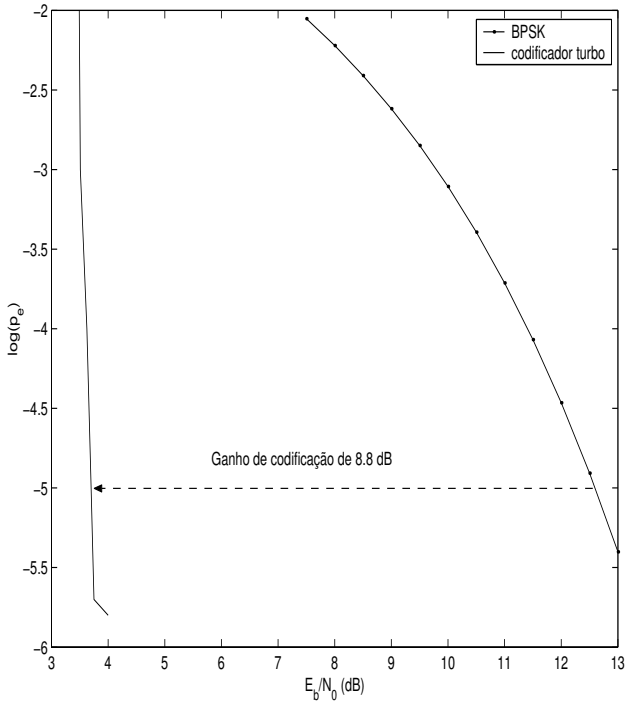

Figura A.5: Ganho de codificação para o código turbo em relação ao sistema BPSK.

Dada uma probabilidade de erro, a redução de E_b/N_0 obtida quando empregada a codificação é denominado de ganho de codificação. Observando a Figura A.5, dado $p_e = 10^{-5}$, obtém-se um ganho de codificação de 8.8dB, quando empregada a codificação turbo. Nestas condições ($p_e = 10^{-5}$ e $R = 1/2$), o limite de Shannon para a capacidade do sistema é $E_b/N_0 \approx 0dB$. Assim, o desempenho deste codificador turbo está a $0.7dB$ deste limite. Este resultado apresentado inicialmente por Berrou [13] representou um forte avanço

no desenvolvimento de códigos corretores de erros, promovendo uma grande flexibilidade nos compromissos (tamanho da antena, alcance, potência do transmissor, sensibilidade do receptor, peso, etc.) de projeto de um sistema de comunicações.

A.1.7
Principais Parâmetros de Projeto de Codificadores Turbo

Consideraremos nesta subseção os principais parâmetros de projeto que condicionam o desempenho do codificador turbo. São eles: comprimento (L) do interleaver; tipo de interleaver; número iterações na decodificação e o algoritmo de decodificação. Para analisar e ilustrar a influência destes parâmetros, replicou-se os gráficos obtidos em [14] nas Figuras A.6 a A.9

O efeito do comprimento do *interleaver* no desempenho do codificador turbo é observado na Figura A.6. Notar que o incremento do comprimento do interleaver melhora significativamente o desempenho do codificador.

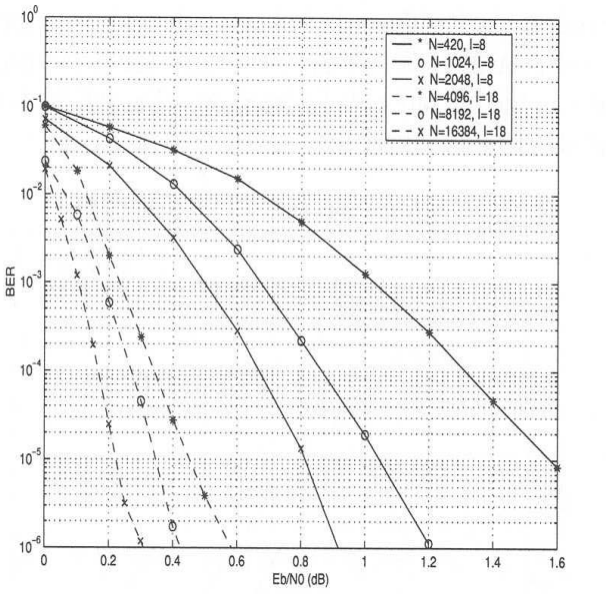

Figura A.6: Efeito do comprimento do interleaver.

O efeito do tipo de interleaver no desempenho do codificador turbo é observado na Figura A.7.

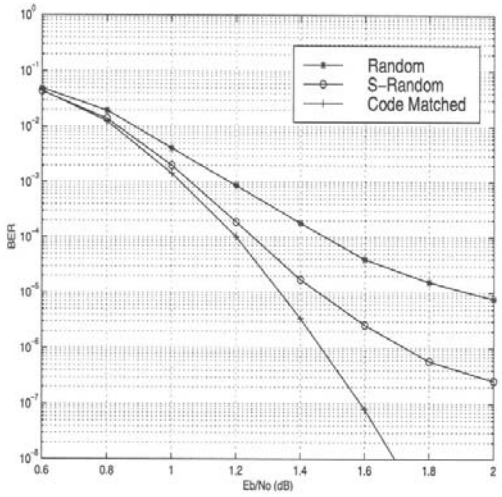

Figura A.7: Efeito do tipo do interleaver.

O efeito do número de iterações no desempenho do codificador turbo é observado na Figura A.8. Notar que o incremento do número de iterações na decodificação melhora significativamente o desempenho do codificador.

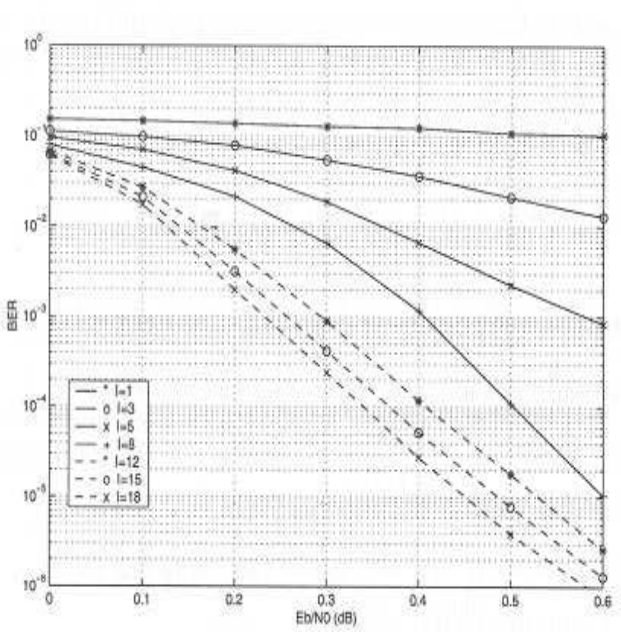

Figura A.8: Efeito do número de iterações.

O efeito do algoritmo no desempenho do codificador turbo é observado na Figura A.9. Observar contudo que a escolha do algoritmo deverá atender diversos requisitos, como a sua complexidade e outros, conforme já comentado.

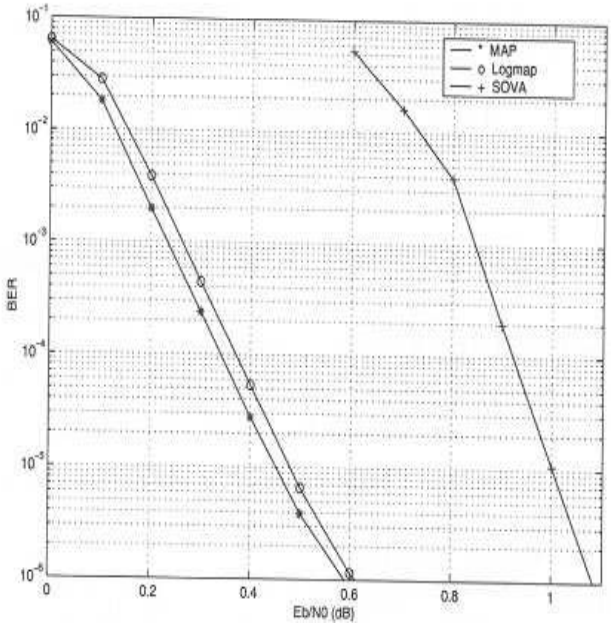

Figura A.9: Efeito do tipo do algoritmo.