

Códigos Turbo Quaternários

José da Silva Barros e Renato Baldini Filho

Resumo—O objetivo deste trabalho é apresentar os códigos turbo quaternários. O esquema de codificação turbo consiste em dois codificadores convolucionais sistemáticos recursivos quaternários em paralelo e separados por um entrelaçador do tipo s-aleatório. O esquema de decodificação turbo é formado por um algoritmo de decodificação iterativo quaternário, baseado na decodificação MAP (máximo a posteriori). Os resultados das simulações mostram que os códigos turbo quaternários apresentam o mesmo desempenho em relação aos códigos turbo binários com metade do tempo de decodificação.

Palavras-Chave—Códigos Turbo Quaternários, Algoritmo de Decodificação Iterativa.

Abstract—The aim of this work is to present the quaternary turbo codes. The coding scheme consists of two quaternary systematic recursive convolutional encoders separated by s-random interleaver. The decoding scheme is based on the maximum a posteriori decoding algorithm - MAP. The simulation results show that quaternary turbo codes present the same performance of binary turbo codes with half decoding time.

Keywords—Quaternary Turbo Codes, Iterative Decoding Algorithm.

I. INTRODUÇÃO

Com base nos estudos feitos sobre os códigos turbo binários [1], [2], [3] e [5], apresentamos os códigos turbo quaternários, ou seja, os códigos turbo definidos sobre os anéis de inteiros módulo-4, \mathbb{Z}_4 .

Os códigos turbo quaternários têm um esquema geral similar ao esquema apresentado para o caso binário [1]. O esquema de codificação turbo proposto é composto por dois codificadores convolucionais sistemáticos recursivos quaternários, um entrelaçador e um mecanismo opcional de puncionamento. O esquema de modulação é o 4-PSK (Quaternary Phase Shift Keying), pois esta modulação se adequa perfeitamente com as operações definidas em \mathbb{Z}_4 e o canal considerado é o AWGN (Additive White Gaussian Noise).

O esquema de decodificação é composto por um algoritmo de decodificação de máximo a posteriori (MAP), símbolo a símbolo, que utiliza os conceitos de código de treliça. Neste esquema de decodificação, existe um processo de decodificação iterativa [1] e [4] que utiliza o algoritmo MAP, em cada decodificador componente, e tem a função de obter a informação extrínseca do estágio de decodificação anterior e usá-la como informação a priori no próximo estágio de decodificação. Este processo geralmente produz uma diminuição na probabilidade de erro de símbolo (ou bit) a cada iteração.

José da Silva Barros e Prof. Dr. Renato Baldini Filho, Faculdade de Engenharia Elétrica e de Computação, Departamento de Telecomunicações, UNICAMP, Campinas, Brasil, E-mails: barros@decom.fee.unicamp.br, baldini@decom.fee.unicamp.br. Este trabalho foi parcialmente financiado pela FAPEAL(E.20031029439-0).

II. CODIFICAÇÃO TURBO QUATERNÁRIA

O esquema de codificação turbo quaternário é constituído por dois codificadores convolucionais geralmente idênticos, de taxa 1/2, separados por um entrelaçador de N símbolos, juntamente com um mecanismo opcional de puncionamento. Esses dois codificadores constituintes são sistemáticos recursivos e concatenados em paralelo de acordo com o esquema da Figura 1 e são descritos conforme as subseções a seguir.

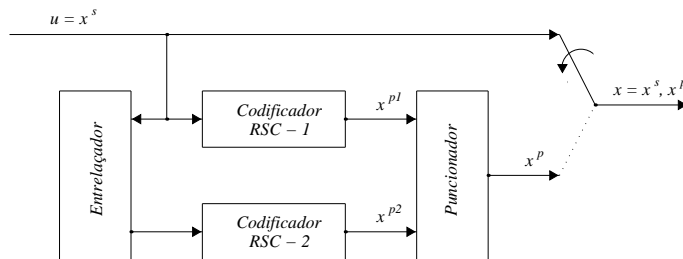


Fig. 1. Esquema de codificação turbo quaternário.

A. Codificador Convolucional Sistemático Recursivo

Seja \mathbb{Z}_4 um anel de inteiros comutativo com identidade multiplicativa e seja $L_r(D)$ o anel formado por polinômios racionais do tipo $q(D)/p(D)$, onde $q(D)$ e $p(D)$ são polinômios com coeficientes em \mathbb{Z}_4 , expressos pela série de Laurent [7], [9] e [6], $f(D) = \sum_{i=0}^n f_i D^i$, com $f_i \in \mathbb{Z}_4$ e $p(0) = 1$ em \mathbb{Z}_4 - chamado anel das funções realizáveis sobre \mathbb{Z}_4 .

Assim, um codificador convolucional sistemático recursivo-RSC (Recursive Systematic Convolutional), com taxa de codificação k/n sobre o anel de funções realizáveis $L_r(D)$, é um mapeamento linear expresso como

$$v(D) = u(D)G_r(D),$$

onde $G_r(D)$ é uma matriz $k \times n$ na forma sistemática

$$G_r(D) = \begin{bmatrix} I_k & X(D) \end{bmatrix},$$

e $u(D)$ é o polinômio correspondente a seqüência de informação. Observe que I_k é uma matriz identidade $k \times k$ e $X(D)$ é uma matriz $k \times (n-k)$ com elementos cuja representação é $q(D)/p(D)$ em $L_r(D)$.

Portanto, o conjunto

$$C = \{u(D)g(D) \mid u(D) \in L_r(D)^k\},$$

é um código RSC com taxa de codificação k/n sobre \mathbb{Z}_4 , onde $g(D) = G_r(D)$ é a matriz geradora com elementos em $L_r(D)$.

A Figura 2 mostra o diagrama do codificador RSC utilizado neste trabalho cuja matriz geradora é:

$$g(D) = \begin{bmatrix} 1 & \frac{2 + D + 2D^2}{1 + D + 3D^2} \end{bmatrix},$$

onde os números 2 e 3 da Figura 2 correspondem aos coeficientes dos polinômios da matriz $g(D)$.

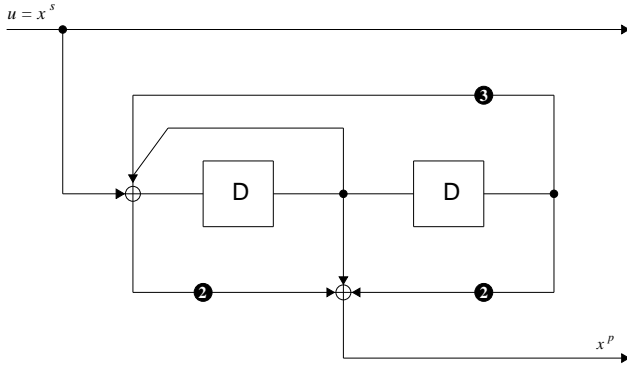


Fig. 2. Diagrama de um codificador RSC quaternário.

B. Entrelaçador

Os entrelaçadores, usados com grande frequência nos meios de comunicação, têm por objetivo espalhar erros que ocorrem em surtos (*burst*), causados pelo ruído impulsivo e pelo desvanecimento (*fading*) seletivo. Usamos em nossa simulação o entrelaçador s-aleatório [8], pois este apresenta melhores propriedades aleatórias que o entrelaçador usado no esquema turbo binário [1].

C. Puncionador

A função do puncionador é apagar periodicamente símbolos de redundância pré-selecionados dos codificadores, aumentando a taxa de codificação. Quando usamos o puncionamento, é mapeado N símbolos de informação em $2N$ símbolos do código, enquanto que, sem o puncionamento, é mapeado N símbolos de informação em $3N$ símbolos do código.

D. Modulador 4-PSK

A seqüência de entrada do modulador é composta por símbolos do alfabeto $\theta = \{0, 1, 2, 3\} \in \mathbb{Z}_4$. Assim, o sinal modulado é representado por

$$S_\theta = \left(\cos\left(\frac{\pi\theta}{2} + \frac{\pi}{4}\right), \sin\left(\frac{\pi\theta}{2} + \frac{\pi}{4}\right) \right).$$

III. ALGORITMO DE DECODIFICAÇÃO

O algoritmo de decodificação turbo de máximo *a posteriori* - MAP, utilizado pelos códigos RSC quaternários, fornece a informação *a posteriori* ou o valor máximo do logaritmo da probabilidade *a posteriori*, $L(u_k)$, definida como

$$L(u_k) = \max(\ln(p(u_k = \theta | y))), \quad (1)$$

onde a seqüência de símbolos recebida é $y = y_1 y_2 \cdots y_{k-1} y_k y_{k+1} \cdots y_N$ e $u_k = \theta \in \{0, 1, 2, 3\}$.

Se as transições entre o estado prévio, $S_{k-1} = s'$, e o estado presente, $S_k = s$, são mutuamente exclusivas (isto é, apenas uma delas pode ter ocorrido na treliça referente ao codificador RSC), então

$$L(u_k) = \max \left(\ln \left(\frac{\sum_{(s',s)} p(S_{k-1} = s', S_k = s, y)}{p(y)} \right) \right), \quad (2)$$

onde (s', s) é o conjunto de transições do estado prévio, $S_{k-1} = s'$, para o estado presente, $S_k = s$, que pode ocorrer se o símbolo de entrada u_k for igual a θ .

Usando o fato de que os símbolos da seqüência recebida y são independentes entre si, esta seqüência y pode ser dividida em três partes: a seqüência recebida antes da transição presente $y_1^{k-1} = y_1 y_2 \cdots y_{k-1}$; a palavra código transmitida associada a transição presente $y_k^k = y_k$, e a seqüência recebida depois da transição presente $y_{k+1}^N = y_{k+1} y_{k+2} \cdots y_N$.

Com base na divisão da seqüência y , podemos escrever:

$$\begin{aligned} p(s', s, y) &= p(S_{k-1} = s', S_k = s, y) \\ &= p(s', s, y_1^{k-1}, y_k, y_{k+1}^N) \end{aligned} \quad (3)$$

que pode ser decomposta como o produto de três probabilidades, $\bar{\alpha}_{k-1}(s')$, $\gamma_k(s', s)$ e $\bar{\beta}_k(s)$, onde: $\bar{\alpha}_{k-1}(s') = p(s', y_1^{k-1})$ é a probabilidade de estar no estado prévio s' no tempo $k-1$ e ter recebido a seqüência de canal y_1^{k-1} ; $\gamma_k(s', s) = p(\{y_k, s\} | s')$ é a probabilidade de receber a seqüência de canal y_k e estar no estado presente s , no tempo k , dado que se estava no estado prévio s' , no tempo $k-1$ e, $\bar{\beta}_k(s) = p(y_{k+1}^N | s)$, é a probabilidade de receber a seqüência futura de canal y_{k+1}^N , dado que se está no estado presente s , no tempo k .

Como o canal é sem memória e os símbolos da seqüência recebida y são independentes entre si, usando a regra de Bayes temos que:

$$p(s', s, y) = \bar{\alpha}_{k-1}(s') \cdot \gamma_k(s', s) \cdot \bar{\beta}_k(s). \quad (4)$$

Finalmente, substituindo a expressão (4) na expressão (2), obtemos

$$L(u_k) = \max \left(\ln \left(\frac{\sum_{(s',s)} \bar{\alpha}_{k-1}(s') \cdot \gamma_k(s', s) \cdot \bar{\beta}_k(s)}{p(y)} \right) \right) \quad (5)$$

onde

$$\bar{\alpha}_k(s) = \sum_{\text{todo } s'} \bar{\alpha}_{k-1}(s') \cdot \gamma_k(s', s) \quad (6)$$

é a expressão recursiva direta e

$$\bar{\beta}_{k-1}(s') = \sum_{\text{todo } s} \bar{\beta}_k(s) \cdot \gamma_k(s', s) \quad (7)$$

é a expressão recursiva reversa.

Supondo que a treliça começa no estado inicial $S_0 = 0$, e que o codificador sempre volta ao estado $S_N = 0$, as condições iniciais e finais para estas recursões são, respectivamente:

$$\bar{\alpha}_0(S_0 = s) = \bar{\beta}_N(S_N = s) = \begin{cases} 1 & \text{para } s = 0 \\ 0 & \text{para } s \neq 0. \end{cases} \quad (8)$$

Voltando à expressão (5), verificamos que, se utilizarmos o divisor $p(y)$, o algoritmo não converge para o limitante de Shannon [5]. Diante disso, usamos o artifício de retirar da seqüência recebida y o elemento de ordem k . Isto é, substituindo $p(y)$ por $p(y)/p(y_k)$ na expressão (5) temos:

$$L(u_k) = \max \left(\ln \left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s) \right) \right)$$

onde $\alpha_k(s)$ e $\beta_k(s)$ são as novas probabilidades modificadas definidas respectivamente como:

$$\alpha_k(s) = \frac{\sum_{\text{todo } s'} \alpha_{k-1}(s') \cdot \gamma_k(s',s)}{\sum_{\text{todo } s} \sum_{\text{todo } s'} \alpha_{k-1}(s') \cdot \gamma_k(s',s)} \quad (9)$$

$$\beta_{k-1}(s) = \frac{\sum_{\text{todo } s} \beta_k(s) \cdot \gamma_k(s',s)}{\sum_{\text{todo } s} \sum_{\text{todo } s'} \alpha_{k-1}(s') \cdot \gamma_k(s',s)}. \quad (10)$$

Observe que $\alpha_k(s)$ e $\beta_k(s)$ são calculados recursivamente pelas equações (9) e (10) e possuem as mesmas condições iniciais que seus complementos naturais, dados em (8) para $\bar{\alpha}_k(s)$ e $\bar{\beta}_k(s)$, respectivamente.

Com base nas mesmas hipóteses temos que a métrica do ramo é dada por

$$\gamma_k(s',s) = p(y_k | x_k) \cdot p(u_k).$$

Assim, apresentamos as expressões finais de $\alpha_{k-1}(s')$, $\beta_k(s)$ e $\gamma_k(s',s)$, necessárias para calcular a informação *a posteriori* $L(u_k)$, na saída de cada decodificador componente.

IV. DECODIFICAÇÃO ITERATIVA TURBO QUATERNÁRIA

O processo de decodificação iterativa turbo sobre \mathbb{Z}_4 , esquematizado na Figura 3, calcula a informação extrínseca na saída do estágio de decodificação anterior, e a usa como informação *a priori* na entrada do próximo estágio de decodificação.

Sabendo que cada codificador *RSC* do esquema de codificação turbo tem taxa de codificação 1/2, a palavra código transmitida é dada por $x_k = x_k^s x_k^p = u_k x_k^p$, e a palavra código recebida é dada por $y_k = y_k^s y_k^p$.

Assim, se o canal é sem memória, gaussiano e com modulação 4-PSK, então, a função densidade de probabilidade é dada por

$$\begin{aligned} p(y_k | x_k) &= \frac{1}{\pi N_0} \cdot \exp \left(\frac{-\|y_k^s - x_k^s\|^2}{N_0} \right) \cdot \exp \left(\frac{-\|y_k^p - x_k^p\|^2}{N_0} \right) \\ &= \exp \left(\frac{-\|y_k^s - x_k^s\|^2}{N_0} \right) \cdot \gamma_k^e(s',s), \end{aligned}$$

onde: x_k^s é o símbolo sistemático da palavra código transmitida x_k ; x_k^p é o símbolo de paridade da palavra código transmitida x_k ; y_k^s é o sinal recebido correspondente ao símbolo transmitido x_k^s e y_k^p é o sinal recebido correspondente ao símbolo transmitido x_k^p .

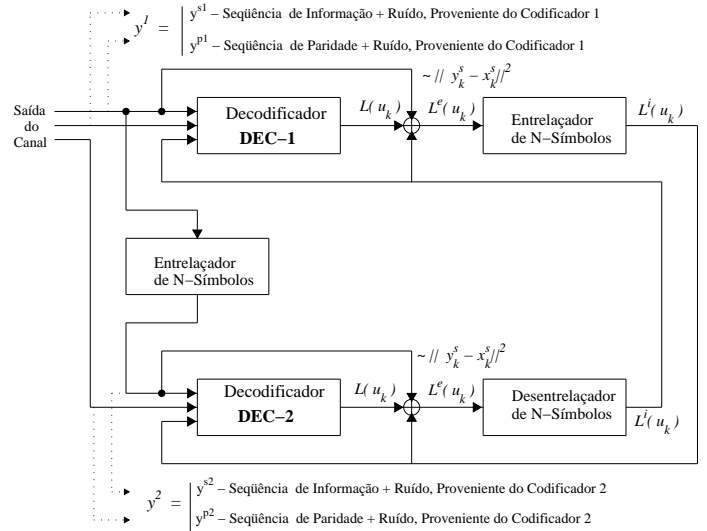


Fig. 3. Esquema de decodificação iterativa turbo.

Portanto, a informação *a posteriori*, $L(u_k)$, calculada através do algoritmo *MAP* na saída de cada decodificador componente, pode ser dividida em três termos

$$L(u_k) = L^i(u_k) + \exp \left(\frac{-\|y_k^s - x_k^s\|^2}{N_0} \right) + L^e(u_k), \quad (11)$$

onde $L^i(u_k) = \ln(p(u_k))$ é informação intrínseca, $\exp \left(\frac{-\|y_k^s - x_k^s\|^2}{N_0} \right)$ é a informação sistemática, e

$$L^e(u_k) = \ln \left(\sum_{(s',s)} \alpha_{k-1}(s') \cdot \beta_k(s) \cdot \gamma_k^e(s',s) \right), \quad (12)$$

é a informação extrínseca. Este termos serão usados no processo de decodificação iterativo como a seguir.

Processo Iterativo: Considere inicialmente o primeiro decodificador componente na primeira iteração. Este decodificador recebe a seqüência de canal, y^1 (ver Figura 3), e produz uma estimativa da informação *a posteriori* $L_{11}(u_k)$ dos símbolos de dados u_k , onde $k \in \{1, \dots, N\}$ e N é o comprimento da seqüência de informação. Note que nesta primeira iteração a informação *a priori* que o primeiro decodificador componente recebe é $\ln(p(u_k = \theta)) = \ln(1/4)$.

O segundo decodificador componente recebe a seqüência de canal y^2 junto com a informação extrínseca entrelaçada do primeiro decodificador componente e fornece uma estimativa da informação *a posteriori* $L_{12}(u_k)$, dos símbolos de dado u_k . Esta informação extrínseca¹ (usada no segundo decodificador) é calculada através da expressão (12) na saída do primeiro decodificador componente.

Na segunda iteração, o primeiro decodificador componente novamente processa sua seqüência recebida de canal y^1 , mas dessa vez, ele também possui a informação *a priori* $L^i(u_k)$.

¹A informação extrínseca calculada na saída do primeiro decodificador, após ser entrelaçada, é usada como informação *a priori* na entrada do segundo decodificador. E de forma análoga, a informação extrínseca calculada na saída do segundo decodificador, após ser desentrelaçada, é usada como informação *a priori* na entrada do primeiro decodificador e assim por diante.

Esta informação *a priori* é fornecida pela porção extrínseca da informação *a posteriori* $L_{12}(u_k)$, calculada pelo segundo decodificador componente, na primeira iteração. Portanto, este decodificador pode produzir uma melhor informação *a posteriori* $L_{21}(u_k)$, na saída do primeiro decodificador componente, na segunda iteração.

Este processo iterativo continua e, a cada iteração, em média a taxa de erro de bit diminui para uma mesma relação sinal ruído - *SNR* (*signal-to-noise ratio*).

V. RESULTADOS

As curvas da Figura 4 apresentam o desempenho dos esquemas turbo binários e quaternários, respectivamente. A curva do esquema binário foi simulada para um comprimento da seqüência de informação de $N = 2000$ bits, matriz geradora do codificador $RSC \ g(D) = \left[1 \ \frac{1+D^4}{1+D+D^2+D^3+D^4} \right]$, com 16 estados na treliça, modulação 2-PSK, taxa de codificação turbo igual a 1/2 e 8 iterações na decodificação. O tempo gasto para decodificar esta seqüência binária na 1ª iteração foi de 67,697 segundos.

Já no esquema quaternário o comprimento da seqüência de informação é de $N = 2000$ bits ou 1000 símbolos, matriz geradora do codificador $RSC \ g(D) = \left[1 \ \frac{2+D+2D^2}{1+D+3D^2} \right]$, com 16 estados na treliça, modulação 4-PSK, taxa de codificação turbo igual a 1/2 e 8 iterações na decodificação. O tempo gasto para decodificar esta seqüência de 1000 símbolos na 1ª iteração foi de 38,806 segundos.

De acordo com a Figura 4 e os parâmetros acima observamos que o desempenho do esquema binário e do quaternário são equivalentes, mas o tempo gasto na decodificação do esquema binário é 1,57 vezes maior que no esquema quaternário. Isto ocorre porque o esquema binário (2000 bits) realiza as operações com o dobro da quantidade de elementos usadas no esquema quaternário (1000 símbolos). Quando os esquemas binário e quaternário operam com seqüências pequenas, 8 bits e 4 símbolos, respectivamente; estes esquemas gastam o mesmo tempo para realizar a decodificação na 1ª iteração. Mas para seqüências de comprimento 10000 bits (no binário) e 5000 símbolos (no quaternário), o tempo gasto pelo esquema binário é de 2 vezes o tempo gasto pelo quaternário na 1ª iteração.

Logo, como o esquema turbo quaternário (com a modulação 4-PSK) ocupa a metade da largura de faixa e gasta a metade do tempo de decodificação em relação ao esquema turbo binário (com modulação 2-PSK), ou seja, para a mesma largura de faixa, o esquema turbo quaternário pode fornecer o dobro da quantidade de informação fornecida no esquema turbo binário com o mesmo tempo de decodificação. Podemos concluir que o uso do esquema turbo quaternário em um sistema de comunicação que possua largura de faixa limitada será mais vantajoso que o esquema turbo binário.

VI. CONCLUSÃO

Neste artigo, apresentamos a construção de um esquema de codificação e decodificação turbo quaternário, analisando a contribuição dada pelos componentes do esquema de codificação para que o esquema de decodificação forneça

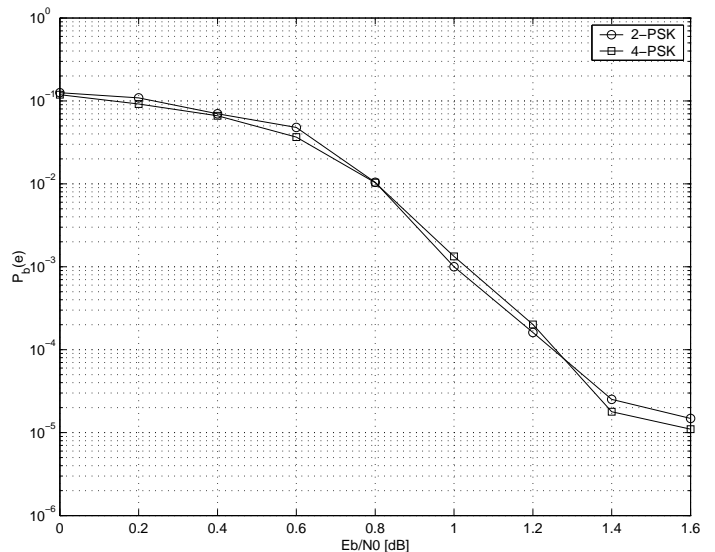


Fig. 4. Desempenho da $p_b(e) \times E_b/N_0$ dos esquema binário e quaternário para 2000 bits e 1000 símbolos, 2-PSK e 4-PSK, respectivamente e taxa 1/2, entrelaçador s-aleatório, 8 iterações e 16 estados na treliça para ambos.

o máximo de confiabilidade sobre os símbolos decodificados. Em seguida, fizemos a implementação do algoritmo de decodificação obtendo como resultados curvas para a probabilidade de erro, nas quais notamos que os esquemas binários e quaternários são equivalentes em relação ao desempenho e o esquema quaternário é capaz de transmitir o dobro da informação transmitida pelo esquema binário com o mesmo tempo de decodificação.

REFERÊNCIAS

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," Proc. Int. Conf. Comm., pp. 1064-1070, May 1993.
- [2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," IEEE Trans. Inform. Theory, vol. 42, pp. 429-445, Mar. 1996.
- [3] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," Proc. Globecom. pp. 1298-1303, 1994.
- [4] Jason P. Woodard and Lajos Hanzo, "Comparative Study of Turbo Decoding Techniques: An Overview," IEEE Trans. Veh. Technol., Vol. 49, n. 6, pp. 2208-2233, Nov. 2000.
- [5] W. E. Ryan, "A Turbo Code Tutorial," Proc. IEEE Globecom'98, 1998.
- [6] Rolf Johannesson, "Some Structural Properties of Convolutional Codes over Rings," IEEE Trans. Inform. Theory, pp. 839-845, Mar. 1998.
- [7] M. F. Atiyah and I. G. MacDonald, "Commutative Algebra," Reading, MA: Addison - Wesley, 1969.
- [8] Chris Heegard and Stephen B. Wicker, "Turbo Coding," Kluwer Academic Publishers pp. 10-63.
- [9] F. Fagnini and S. Zampieri, "System-Theoretic Properties of Convolutional Codes Over Rings," IEEE Trans. Inform. Theory, vol. 47, N.º. 6, Sept. 2001, pp. 2256-2274.