Codificação de Canal na TV 3.0

Por Fadi Jerji

Este artigo apresenta uma introdução às técnicas de codificação de canal das tecnologias proponentes dos padrões de radiodifusão de TV Digital Terrestre (TDT), como parte do projeto TV 3.0. A análise do desempenho e complexidade dos códigos de correção de erro é destacada, e exemplos de estudos relacionados são apresentados.

Teste da camada de transporte/Foto: Fórum SBTVD

I. Introdução

Desde a publicação da chamada de propostas do projeto TV 3.0 pelo **Fórum do Sistema Brasileiro de Televisão Digital** (SBTVD) em julho de 2020, várias propostas foram recebidas dos proponentes, cada uma destas apresenta um novo candidato padrão de radiodifusão de TV Digital terrestre, do inglês *Digital Terrestrial Television Broadcasting* (DTTB) para ser considerado como o próximo padrão brasileiro de DTTB [1].

As propostas recebidas para a camada física da TV3.0 foram: o sistema *Advanced Integrated Services Digital Broadcasting-Terrestrial* (Advanced ISDB-T) pelo DiBEG, a próxima geração do sistema *Advanced Television Systems Committee* (ATSC), o ATSC 3.0 pela ETRI e ATSC, o *Digital Terrestrial Multimedia Broadcast-Advanced* (DTMB-A) pelo DTNEL e o 5G Broadcast/EnTV pela Qualcomm/Rohde & Schwarz GmbH/Kathrein.

As quatro tecnologias propostas utilizam códigos de correção de erro, do inglês *Forward Error Correction* (FEC) para detectar e corrigir erros induzidos pelo canal. O Advanced ISDB-T, o ATSC 3.0, e o DTMB-A usam os códigos verificação de paridade de baixa densidade, do inglês *Low-Density Parity-Check* (LDPC) como o FEC principal, enquanto o Quinta Geração, do inglês *fifth-generation* (5G) radiodifusão/EnTV utiliza os códigos Turbo.

A aplicação do código FEC nos bits de informação gera um conjunto de bits de paridade que pode ser enviado junto com os bits da informação permitindo a detecção pelo receptor e a correção de erros induzidos pelo canal.

Os códigos Turbo e LDPC são conhecidos por possuir desempenho de correção de erro que se aproxima ao limite teoricamente possível conhecido como o limite de Shannon. Esse limite define a taxa de dados máxima para um sinal de uma determinada potência, propagando por um canal ruidoso, onde os dados descontínuos podem ser transmitidos quase sem erros com o uso de uma largura de banda limitada [2], [3].

A implementação de um FEC de alto desempenho é de importância especial no caso da TV 3.0 considerando que um dos principais requisitos da TV 3.0 é a frequência de reuso-1. Isso significa que estações fornecendo serviços para áreas vizinhas devem possuir a capacidade de utilizar o mesmo canal de frequência de rádio, do inglês *Radio Frequency* (RF) para transmitir conteúdos diferentes, e portanto, causam alto nível de interferência, até o ponto que o receptor pode receber um sinal indesejado (ruído) que possui potência que é igual ou maior do que a potência do sinal desejado, em outras palavras, uma relação a portadora/ruído, do inglês *Carrier-to-Noise ratio* (C/N) que é menor ou igual a zero decibel (dB). Essa interferência em adição dos outros tipos de ruído eletromagnético como o ruído representado pelo canal *Additive White Gaussian Noise* (AWGN) e o canal Rayleigh, torna

o desempenho dos códigos FEC um fator de alta importância quando da escolha de um padrão sobre outro.

Neste artigo, são apresentadas as características principais dos códigos Turbo e dos códigos LDPC bem como exemplos dos processos da codificação e decodificação. Além disso, são demonstradas as métricas, os métodos da avaliação e dois exemplos de estudos anteriores. O resto deste artigo é organizado da seguinte maneira. A Seção II traz uma introdução ao processo de demodulação necessária para uma implementação eficiente de FEC, uma breve explicação dos processos de codificação e decodificação dos códigos Turbo na Seção III, as etapas de codificação e decodificação dos códigos LDPC na Seção IV, as principais métricas de avaliação da seleção de FEC na Seção V, e finalmente a conclusão na Seção VI.

II. Demodulação de decisão suave

Nos sistemas de comunicação digital, os bits de informação são mapeados por um conjunto de símbolos (constelação) no lado da transmissão por um modulador, esses símbolos podem ser representados nos eixos real e imaginário, que representam os valores da quadratura (Q) e fase (I) do sinal transmitido respectivamente. Como os sinais são transmitidos por um canal ruidoso, o símbolo recebido pode sofrer um deslocamento em um dos eixos ou ambos.

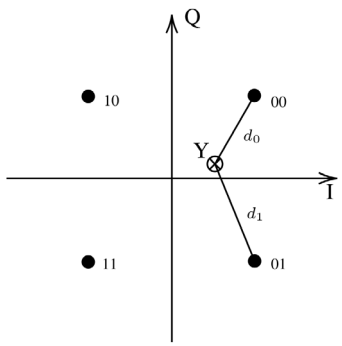


Fig. 1: Demodulação LLR de decisão suave.

Enquanto é relativamente fácil desenhar uma linha no meio da distância entre os dois símbolos, criando um diagrama de Voronoi para determinar os valores dos bits do símbolo recebido, a demodulação por decisão abrupta, do inglês *Hard Decision* (HD), não fornece informações suficientes para que os poderosos códigos FEC, como os códigos Turbo ou os códigos LDPC, decodificam a mensagem em níveis altos de ruído corretamente.

$$LLR_{bit0} = \frac{1}{\sigma^2}(d_1 - d_0) \qquad (1)$$

Demoduladores de decisão suave, do inglês *Soft Decision* (SD) estimam a verossimilhança de cada um dos bits do símbolo recebido sendo 0 ou 1. Um exemplo é o método aproximado de razão logarítmica de verossimilhança, do inglês *Log-Likelihood Ratio* (LLR), apresentada na **Fig. 1**, onde a estimação é feita pelo cálculo da distância **Euclidiana** entre o símbolo recebido (Y) e o símbolo mais próximo com o valor de bit em interesse sendo 0 d_0 e a distância Euclidiana entre o símbolo recebido e o símbolo mais próximo com o valor de bit em interesse sendo 1 d_1. O cálculo da verossimilhança deste bit sendo 0 ou 1 e feito pela aplicação da Eq. 1 onde σ2 é a variança do ruído.

Neste caso, o sinal da *LLRbit0* define a decisão do demodulador se o bit é 0 ou 1 enquanto o valor da *LLRbit0* é a confiança em essa decisão. Em outras palavras, a *LLRbit0 = −100.0* significa alta confiança que o bit recebido é 1, enquanto a *LLRbit0 = +1.3* significa baixa confiança que o bit recebido é 0. A *LLRbit0 = 0* e uma probabilidade igual desse bit sendo 0 ou 1.

III. Códigos Turbo

Os códigos Turbo foram originalmente propostos por [3] como uma forma para desenvolver c´odigos convolucionais sistemáticos recursivo, do inglês *Recursive Systematic Convolutional* (RSC) de alto desempenho pela concatenação de códigos RSC de baixo desempenho. Enquanto os códigos Turbo originais implementam RSC como o principal codificador/decodificador, qualquer código de entrada-suave saída-suave, do inglês *Soft-Input Soft-Output* (SISO) pode ser usado, incluindo os códigos LDPC. Nos códigos Turbo, a concatenação dos códigos SISO pode ser feita de maneira sequencial ou paralela.

a. Codificação

O processo de codificação de um código Turbo implementa dois codificadores SISO ou mais,

na maioria dos casos, esses codificadores são idênticos para simplificar o desenho do código Turbo. A codificação é feita pela aplicação do primeiro codificador aos bits de informação d_k e o segundo codificador a versão entrelaçada dos bits de informação, assim gerando dois conjuntos diferentes de bits de paridade Y_{1k} e Y_{2k} respectivamente. Um multiplexador é usado para concatenar os bits de informação e os dois conjuntos de bits de paridade para criar o fluxo final de bits Y_k. Uma etapa opcional de funcionamento pode ser adicionada para alcançar uma taxa de codificação especifica. A **Fig. 2** demonstra um codificador de código Turbo que utiliza dois codificadores em paralelo.

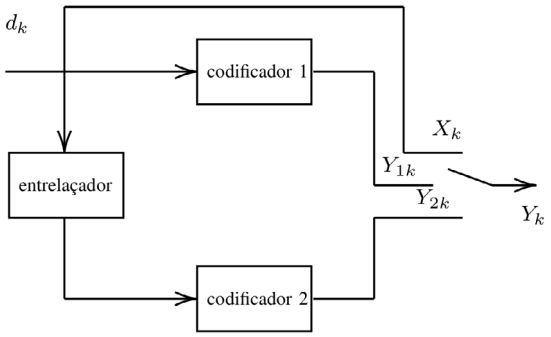


Fig. 2: Codificador de código Turbo.

b. Decodificação

Para decodificar uma mensagem codificada usando o codificador Turbo, o decodificador demonstrado na **Fig. 3** é usado. Inicialmente, os bits recebidos precisam ser de-multiplexados e os conjuntos de bits de paridade precisam ser separados. No caso de um código Turbo composto de dois codificadores RSC, uma cópia dos valores da LLR dos bits de informação e do primeiro conjunto de bits de paridade é entregue para o primeiro decodificador. Assim que o primeiro decodificador tiver produzido os seus valores extrínsecos de LLR (a sua LLR de saída menos a sua LLR da entrada), os valores extrínsecos de LLR são entrelaçados e entregues ao segundo decodificador, junto com a versão entrelaçada das LLRs dos bits de informação e o segundo conjunto de bits de paridade. Os valores extrínsecos de LLR do segundo decodificador são desentrelaçados e retornados para a entrada do primeiro decodificador. O processo é repetido até chegar ao número máximo de interações, gerando a mensagem decodificada Uk.

Essa troca de valores extrínsecos de LLR entre os decodificadores permite que decodificadores RSC de baixo desempenho tenham um desempeno de decodificação que alcança o limite teórico de Shannon.

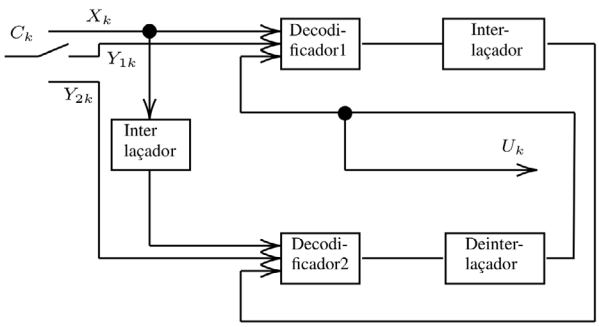


Fig. 3: Decodificador de códigos Turbo.

IV. Códigos LDPC

Os códigos LDPC foram introduzidos pelo Dr. Gallager em [4] e devido a sua alta complexidade computacional, permaneceram praticamente esquecidos ate a sua redescoberta e popularização por [5] que comprovou que códigos LDPC irregulares e com tamanho da palavra maior do que 24000 bits possuem desempenho melhor do que os códigos Turbo, e em [2] foi destacada a vantagem dos códigos LDPC de ter desempenho que alcança o limite de Shannon.

Com o avanço nas técnicas de fabricação dos microchips e na arquitetura de microprocessadores, que permitiu a decodificação dos códigos LDPC de alta complexidade em tempo real, a sua implementação em vários sistemas de comunicação digital cresceu, como a segunda geração do *Digital Video Broadcasting-Satellite- Second Generation* (DVB-S2), a segunda geração do *Digital Video Broadcasting-Terrestrial-Second Generation* (DVB-T2), ATSC 3.0 e Advanced ISDB-T [6]–[9].

a. Codificação

Para codificar uma mensagem U como na Eq. 2 usando um codificador LDPC, uma matriz esparsa de codificação G é usada como na Eq. 3. Como o nome “esparsa” sugere, a matriz G é composta principalmente de zeros com poucos 1 em cada linha e cada coluna.

$$U = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0] \tag{2}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{3}$$

A multiplicação da mensagem U pela matriz G gera a mensagem codificada que é composta de uma cópia idêntica da mensagem original além dos bits de paridade calculados como na Eq. 4. Neste caso, o código LDPC é considerado sistemático enquanto em um código LDPC não-sistemático, a mensagem original não seria concatenada com os bits de paridade, mas a multiplicação pela matriz G altera os bits originais da mensagem U.

$$U * G = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (4)$$

b. Decodificação

O processo de decodificação de uma mensagem LDPC é normalmente dividida em duas etapas, a verificação da integridade da mensagem e a correção de erros caso necessário. Para a etapa da verificação da integridade, somente os sinais das LLR são considerados, e então, ignorando a confiança do demodulador nesse sinal. Uma LLR negativa significa que o bit 1 foi recebido enquanto uma LLR positiva significa que o bit 0 foi recebido. Esses valores de bits são então multiplicados com o transpor da matriz de verificação da paridade *H* como na Eq. 5. Se o resultado desta multiplicação (o síndrome) for um vetor de zeros, então a integridade da mensagem é garantida, caso contrário, uma etapa de correção de erros será necessária.

$$H^T = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

É importante mencionar que a matriz *H* é calculada da matriz G usando a Eq. 6.

$$G * H^T = 0 \quad (6)$$

Outra maneira de representar a matriz e verificação de paridade H é usando o grafo de Tanner por [10]. A **Fig. 4** demonstra o grafo de Tanner da matriz *H* da Eq. 5, onde os valores de LLR recebidos são chamados de nós de variáveis (V1 . . . V10), enquanto as saídas da verificação da integridade são chamadas os nós de verificação (C1 . . .C4), cada linha do grafo de Tanner corresponde com um dos uns da matriz esparsa *H* conectando um nó de variável com um nó de verificação. Esse grafo permite um entendimento melhor do código LDPC e então uma construção mais fácil do código.

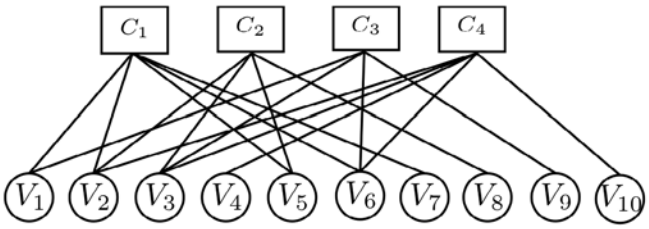


Fig. 4: Grafo de Tanner.

A etapa de correção de erros depende do algoritmo de decodificação usado. Vários algoritmos foram introduzidos na literatura, esses algoritmos variam em relação do desempenho e complexidade. O algoritmo de soma-produto, do inglês *Sum-Product Algorithm* (SPA), proposta por [11] é geralmente considerado a ter o melhor desempenho entre todos os algoritmos de decodificação, mas a sua alta complexidade torna a implementação do SPA não prática em muitos casos.

O algoritmo mínima-soma, do inglês *Min-Sum Algorithm* (MSA) por [12] é uma versão simplificada do SPA que permite uma degradação de desempenho para permitir uma implementação pratica. O algoritmo de mínima-soma normalizado, do inglês *Normalized Min-Sum Algorithm* (NMSA), o algoritmo de mínima-soma deslocado, do inglês *Offset Min-Sum Algorithm* (OMSA), e o algoritmo de correção variável, do inglês *Variable Correction Algorithm* (VCMS) são variancas do MSA desenvolvidas para minimizar a degradação de desempenho do MSA sem aumentar a sua complexidade [13]–[15].

Outros algoritmos de baixa complexidade estão presentes na literatura como o algoritmo de inversão dos bits, do inglês *Bit-Flipping Algorithm* (BFA), o algoritmo de inversão ponderada dos bits, do inglês *Weighted Bit-Flipping Algorithm* (WBF), e o algoritmo de inversão gradiente dos bits, do inglês *Gradient Bit-Flipping Algorithm* (GBF) [4], [16], [17].

A **Fig. 5** demonstra parte do processo de decodificação do GBF. O eixo da LLR demonstra os valores e os sinais das LLRs recebidas representadas por barras, aplicando a etapa da verificação da integridade para calcular os valores dos nós de verificação, pode ser visto da **Fig. 5-A** que o nó de verificação C4 está com o valor 1, que significa que

um ou mais dos nós de variável são corrompidos.

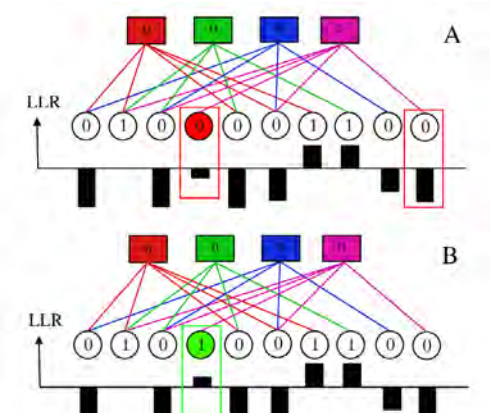


Fig. 5: Exemplo de decodificação de mensagem LDPC usando GBF.

A **Fig. 5-B** demonstra o mesmo processo de decodificação depois da aplicação da etapa de correção de erros do GBF onde o nó de variável com o menor valor de confiança (menor valor absoluta de LLR), o nó de variável V4, tem a seu sinal invertido primeiro como resultado da correção acumulativa pelo GBF, portanto, na próxima etapa de verificação da integridade, todos os nós de verificação teriam o valor 0, declarando que a mensagem foi corrigida e o processo de decodificação pode ser encerrado.

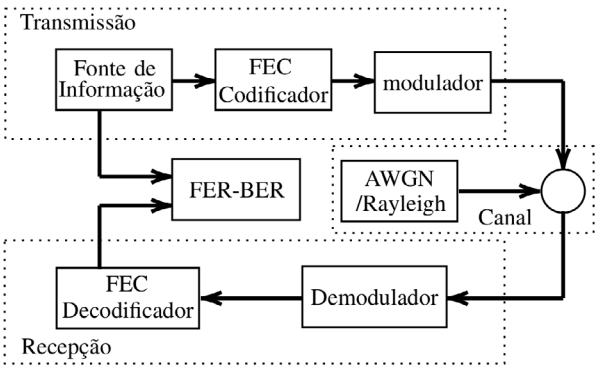


Fig. 6: Diagrama de sistema de comunicação

c. Códigos LDPC em TV 3.0

Três dos quatro proponentes estão usando LDPC como FEC principal, e para simplificar a implementação desses códigos, todos utilizam códigos LDPC da estrutura do código LDPC quase cíclico, do inglês *Quasi-cyclic LDPC Code* (QC-LDPC). Em geral, códigos LDPC de palavras longas são usados para um desempenho melhor, enquanto códigos com palavras curtas são utilizados para aplicações de baixa latência.

O *Advanced ISDB-T* utiliza códigos LDPC de dois tamanhos, uma longa palavra de 69120 bits e curta palavra de 17280 bits. Para cada tamanho, 13 taxas de codificação podem ser usadas (2/16 . . . 14/16). O ATSC 3.0 utiliza códigos LDPC de dois tamanhos, uma palavra normal de 64800 bits e curta palavra de 16200 bits. Para cada tamanho, 12 taxas de codificação podem ser usadas (2/15 . . . 13/15). Ambos o Advanced ISDB-T e o ATSC 3.0 utilizam a estrutura repetitiva acumulante irregular, do inglês *Irregular Repeat Accumulate* (IRA), para a matriz H para altas taxas de codificação e a estrutura de tipo de múltiplas bordas, do inglês *Multi-Edge Type* (MET), para as baixas taxas de codificação pelo seu desempenho superior nessas taxas [18].

No DTMB-A, o tamanho do código LDPC pode ser de 61440 bits ou 15360 bits, cinco taxas de codificação podem ser usadas (1/4, 1/3, 1/2, 2/3, 5/6). Os códigos LDPC da taxa de codificação de (1/4, 1/3) utilizam a IRA, enquanto as demais taxas implementam a forma triangular inferior aproximada [19].

V. Métricas de Avaliação

Para avaliar o código FEC, vários parâmetros devem ser considerados, como o seu desempenho de correção de erros representado pela taxa de erro de bits, do inglês *Bit Error Rate* (BER) ou taxa de erro de frames, do inglês *Frame Error Rate* (FER), em relação a energia de símbolo pela densidade espectral de potência de ruído, do inglês *energy per symbol to noise power spectral density ratio* (Es/N0), ou energia de bit pela densidade espectral de potência de ruído, do inglês *Energy per bit to noise power spectral density ratio* (Eb/N0), a existência de um piso de erro e a sua complexidade de codificação e decodificação.

a. Desempenho

O desempenho da correção de erros é o fator mais importante em muitos casos de implementação, como ele pode definir a diferencia entre um sinal decodificado perfeitamente, e a falha total na recepção. Para a avaliação do desempenho de um código de correção de erros FEC, pode ser usada uma simulação de um sistema digital como na **Fig. 6**, onde dados aleatórios são gerados, codificados usando o codificador FEC, e então mapeados aos símbolos da constelação usando um modulador, isso simula o lado da transmissão. Um

simulador de canal é usado para simular ambientes diferentes, como AWGN ou o canal Rayleigh. Para simular o lado da recepção, um demodulador de saída suave e um decodificador FEC são usados. Os bits recebidos podem ser comparados aos bits transmitidos para calcular a FER ou a BER. Pela alteração do nível de ruído representado pelo valor da Es/N0 ou a Eb/N0, é possível determinar o nível de ruído que corresponde a um valor específico de FER ou BER, os valores de FER = 10−4 ou BER = 10−7 são normalmente usados.

A **Fig. 7** e **Fig. 8** são exemplos da comparação de analise de desempenho feitos por [20] e [21] respectivamente, onde os autores no primeiro analisaram a FER do código LDPC do Advanced ISDB-T e do ATSC 3.0 em relação a Es/N0, enquanto os autores no segundo analisaram o desempenho do código LDPC em comparação ao código Turbo em relação a Es/N0.

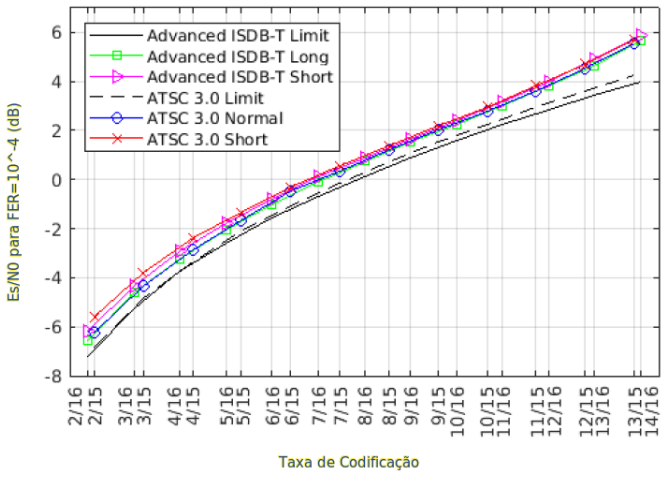
b. Complexidade

A complexidade da codificação e da decodificação são outros fatores que podem ser de alta importância no processo de seleção de um código FEC. Enquanto a complexidade da codificação em DTTB pode ser compensada até um certo ponto pelo uso de equipamentos mais complexos que consumem mais energia (ao contrario dos outros tipos dos transmissores de comunicação digital que utilizam baterias), a alta complexidade pode resultar em dispositivos mais caros, necessidade de *chip-sets* mais especializados, maior consumo de energia e maior latência na recepção.

A avaliação da complexidade dos códigos LDPC somente considerava o número de elementos não-zeros na matriz de verificação da paridade, porem, os autores de [20] demonstraram que códigos LDPC com o mesmo número de elementos não-zeros podem ter complexidades diferentes.

A **Fig. 9** e **Fig. 10** são exemplos de estudos de analise e comparação da complexidade dos códigos LDPC e os códigos Turbo feitos por [20] e [21] respectivamente. Pode ser visto que a variação da complexidade pode ser de alta ordem e não deve ser ignorada no processo de seleção de um código FEC novo.

Fig. 7: Desempeno dos códigos LDPC pelo canal AWGN para Advanced ISDB-T e ATSC 3.0 usando o decodificador SPA [20].



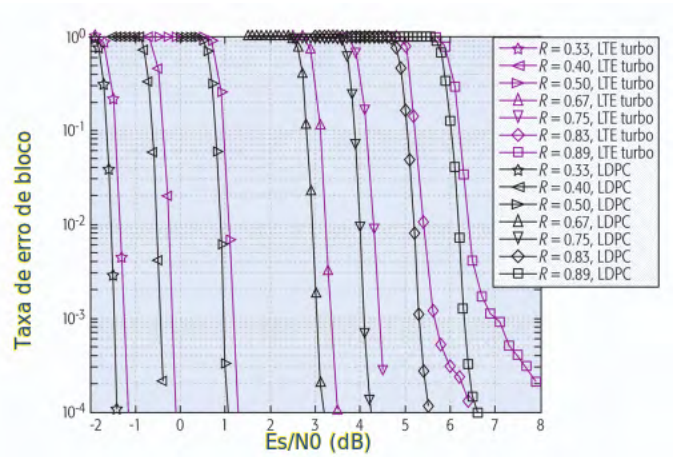


Fig. 8: Desempenho dos códigos LDPC usando o decodificador SPA e códigos LTE Turbo usando o decodificador Log- MAP pelo canal AWGN [21].

VI. Conclusão

Este artigo apresentou uma visão geral dos métodos de codificação do canal pospostos pelas tecnologias proponentes no projeto TV 3.0. A tecnologia selecionada deve utilizar um FEC que garanta a integridade das informações considerando a topologia e os pré-requisitos específicos. Além disso, o código FEC do padrão selecionado dever ter um desempenho aceitável em diferentes modelos de canal de comunicação e uma baixa complexidade de decodificação para facilitar a implementação nos dispositivos dos consumidores.

Referências

[1] B. D. T. T. S. Forum. (2020, July) TV 3.0 project. [Online]. Available: https://forumsbtvd.org.br/tv3 0/, accessed on 17/07/2022

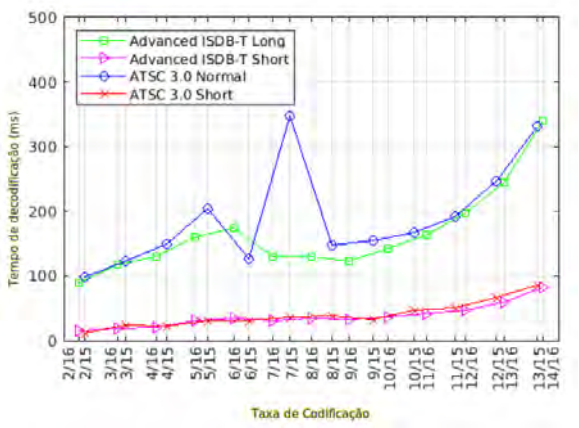


Fig. 9: Complexidade medida do ISDB-T avançado e ATSC 3.0 usando o decodificador SPA [20].

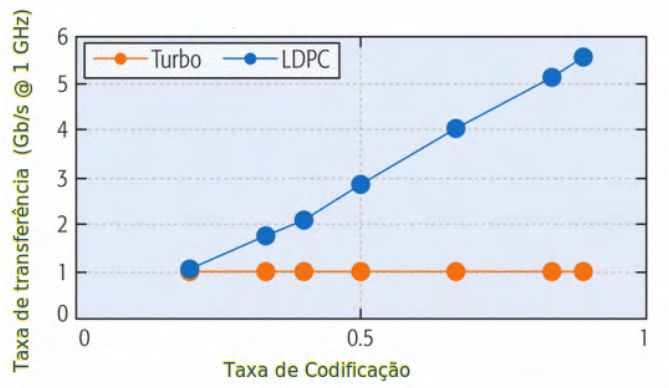


Fig. 10: Taxa de transferência dos decodificadores de LDPC e Turbo para taxas de codificação diferentes [21].

[2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes,"Electronics Letters, vol. 33, no. 6, pp. 457–458, Mar 1997.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit errorcorrecting coding and decoding: Turbo-codes. 1," in Proceedings of ICC '93 - IEEE International Conference on Communications, vol. 2, 1993, pp. 1064–1070 vol.2.

[4] R. Gallager, "Low-density parity-check codes," IRE Transactions on Information Theory, vol. 8, no. 1, pp. 21–28, Jan 1962.

[5] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," IEEE Transactions on Information Theory, vol. 45, no. 2, pp. 399–431, Mar 1999.

[6] DVB, "Part 1: DVB-S2, ETSI EN 302 307-1 V1.4.1," Digital Video Broadcasting, FRANCE, Standard, Nov. 2014.

[7] ——, "DVB-T2, ETSI EN 302 755 V1.4.1," Digital Video Broadcasting, FRANCE, Standard, Jul. 2015.

[8] ATSC, "Physical Layer Protocol Standard, A/322:2017," The Advanced Television Systems Committee, Washington, DC, USA, Standard, Jun. 2017.

[9] T. Shitomi, "Fixed reception performance of fdm-based transmission system for advanced ISDB-T," SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING, vol. 6, no. 0, p. 11, 2021. [Online]. Available: http://revistaeletronica.set.org.br/index.php/ijbe/article/view/202

[10] R. Tanner, "A recursive approach to low complexity codes," IEEE Transactions on Information Theory, vol. 27, no. 5, pp. 533–547, Sep 1981.

[11] F. R. Kschischang, B. J. Frey, and H. . Loeliger, "Factor graphs and the sum-product algorithm," IEEE Transactions on Information Theory, vol. 47, no. 2, pp. 498–519, Feb 2001.

[12] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," IEEE Transactions on Communications, vol. 47, no. 5, pp. 673–680, May 1999.

[13] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," IEEE Transactions on Communications, vol. 50, no. 3, pp. 406–414, March 2002.

[14] ——, "Density evolution for two improved bp-based decoding algorithms of ldpc codes," IEEE Communications Letters, vol. 6, no. 5, pp. 208–210, May 2002.

[15] C. Chen, Y. Xu, H. Ju, D. He, W. Zhang, and Y. Zhang, "Variable correction for min-sum ldpc decoding applied in atsc3.0," in 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), June 2018, pp. 1–5.

[16] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," IEEE Transactions on Information Theory, vol. 47, no. 7, pp. 2711–2736, Nov 2001.

[17] F. Jerji and C. Akamine, "Gradient bit-flipping ldpc decoder for atsc 3.0," in 2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Jun 2019.

[18] S. Myung, S. Park, K. Kim, J. Lee, S. Kwon, and J. Kim, "Offset and normalized min-sum algorithms for atsc 3.0 ldpc decoder," IEEE Transactions on Broadcasting, vol. 63, no. 4, pp. 734–739, Dec 2017.

[19] T. Richardson and R. Urbanke, "Efficient encoding of low-density paritycheck codes," IEEE Transactions on Information Theory, vol. 47, no. 2, pp. 638–656, 2001.

[20] F. Jerji and C. Akamine, "Advanced isdb-t and atsc 3.0 ldpc codes performance and complexity comparison," IEEE Transactions on Broadcasting, vol. 68, no. 1, pp. 254–262, 2022.

[21] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5g new radio," IEEE Communications Magazine, vol. 56, no. 3, pp. 28–34, 2018.

Fadi Jarji é graduado em Engenharia Elétrica, com ênfase em Engenharia de Computadores e Automação pela Universidade Albaath, Síria (2010) revalidada pela Universidade Federal do Rio de Janeiro (Engenharia Eletrônica e de Computação) (2017) e mestrado em Engenharia Elétrica e Computação pela Universidade Presbiteriana Mackenzie (2019) com distinção e louvor. Professor no curso de Engenharia Elétrica (2019-2021) e atualmente é cursado do curso de doutorado do Programa de Pós-Graduação em Engenharia Elétrica e Computação (PPGEEC) da Universidade Presbiteriana Mackenzie e participa de projetos realizados pelo Laboratório de TV digital da Universidade Presbiteriana Mackenzie. Como linha de pesquisa, tem atuado principalmente em áreas como inteligência artificial e codificação de canal como: redes neurais artificiais, aprendizagem de máquina e decodificação dos códigos de correção de erros. Nessas áreas têm publicado artigos científicos em principais congressos internacionais, bem como em revistas especializadas. Possui experiência na área de Engenharia Elétrica, com ênfase em sistemas embarcados, arquitetura de computadores, computação paralela, lógica reconfigurável, comunicação digital, redes de computadores, rádio definido por software e automação industrial

Contato: fadi.jerji@gmail.com