Implementação de Códigos LDPC no GNU Radio: Estudo de Caso do Padrão G.hn

Marcos Takeda, Claudio Coutinho, Fernanda Smith, Aldebaro Klautau e Evaldo Pelaes

***Resumo*— O uso de plataformas livres é importante para as pesquisas e desenvolvimentos em telecomunicações. O GNU Radio é um dos ambientes de software livre mais consolidados na área, com diversos usos em rádio cognitivo e outras aplicações. Este artigo apresenta a implementação dos códigos LDPC (Low-Density Parity-Check) especificados pelo padrão G.hn utilizando o GNU Radio. Os códigos LDPC são atualmente a opção de codificação de canal para diversos sistemas por apresentarem performance próxima da capacidade do canal.**

***Palavras-Chave*— LDPC, GNU Radio, G.hn.**

***Abstract*— The use of free platforms is important for research and development in communications. GNU Radio is a free software environment consolidated in the area, with several uses in cognitive radio and other applications. This paper presents the implementation of LDPC (Low-Density Parity-Check) codes especified by G.hn standard using GNU Radio. The LDPC codes are currently the channel coding option for different systems, because their performance is close to the channel capacity.**

***Keywords*— LDPC, GNU Radio, G.hn.**

I. INTRODUÇÃO

Com a crescente demanda por altas velocidades de comunicação de dados, técnicas avançadas de codificação de canal vêm sendo utilizadas. Entre elas estão os códigos LDPC, que possuem alta capacidade de correção e baixa complexidade na decodificação, quando comparados com códigos Turbo. Neste artigo, será apresentada a implementação de uma técnica de codificação para as matrizes de paridade encontradas no padrão para redes domésticas de alta velocidade G.hn [1]. Estas matrizes fazem parte de uma classe de códigos chamada QC-LDPC-BC (*Quasi-Cyclic Low-Density Parity-Check Block Codes*) [2], utilizada para diminuir a complexidade da codificação e a memória necessária para o armazenamento das matrizes. Também será apresentada a implementação do decodificador padrão de códigos LDPC, que utiliza o algoritmo soma produto (SPA), tarefa facilitada com o uso das ferramentas providas pelo GNU Radio [3]. A implementação foi realizada apenas em *software*, sem a utilização da plataforma de *hardware* USRP.

II. LDPC

Os códigos LDPC foram inicialmente desenvolvidos por Gallager [4] por volta de 1960, e ficaram esquecidos até que, em 1999, foram redescobertos por MacKay [5]. A partir disso, várias pesquisas vêm sendo feitas, devido ao seu alto desempenho. Tais códigos fazem parte da classe de códigos lineares, que são representados por uma matriz de paridade ou uma matriz geradora. Normalmente, em códigos lineares, para que seja possível a decodificação com máxima verossimilhança, é necessária uma busca pela menor distância entre todas as palavras código. Com os códigos LDPC, é possível reduzir consideravelmente a complexidade do decodificador, que utiliza técnicas de *Belief Propagation*, para determinar, com maior probabilidade, os bits corretos. A principal característica de códigos LDPC é a matriz de paridade ter baixo número de valores não nulos, ou seja, uma matriz esparsa. Isso possibilita a redução da complexidade do decodificação, assim como a possibilidade de usar blocos maiores.

Marcos Takeda, Claudio Coutinho, Fernanda Smith Laboratório de Processamento de Sinais, Universidade Federal do Pará, Belém-PA, Brasil, E-mails: {marcos.takeda, claudio.filho}@itec.ufpa.br, {fesmith, aldebaro, pelaes}@ufpa.br.

A. Codificação

Um das maneiras de realizar a codificação dos códigos LDPC é usando a matriz geradora do código **G**. De posse da matriz **G**, para codificar a sequência $\hat{u}$, basta fazer $\hat{u} \cdot \mathbf{G}$ e tem-se como resultado a palavra código. Apesar de simples, este método é custoso computacionalmente, pois não leva em consideração a esparsidade da matriz.

O método que foi utilizado para a codificação das matrizes do padrão G.hn foi baseado no método de Richardson-Urbanke [6], modificado para as matrizes de QC-LDPC-BC [2], que utiliza a matriz de cheque paridade H para realizar a codificação. A Matriz H é constituída de submatrizes,

$$H = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,n} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m,1} & P_{m,2} & \cdots & P_{m,n} \end{bmatrix} \tag{1}$$

onde cada $P_{i,j}$ é uma matriz identidade $b \times b$ com as colunas deslocadas ciclicamente, m é o número de linhas definindo a paridade e n é o número de colunas, definindo o tamanho da palavra-código. Para acharmos a palavra código $\hat{v}$ basta usarmos a identidade $H\hat{v}^T = 0$.

B. Decodificação

A decodificação é realizada dado que são fornecidas informações probabilísticas sobre o canal. O algoritmo de Soma Produto usado é baseado em *Log Likelihood Ratios* (LLR), que são definidas por

$$LLR_i = \log\left(\frac{P(b_i = 0)}{P(b_i = 1)}\right) \tag{2}$$

O canal é classificado como *Binary Input, Soft Output* (BISO), pois recebe bits e tem como saída as probabilidades

na forma de LLRs. O algoritmo é iterativo e baseado, basicamente, em dois passos: horizontal e vertical. No passo horizontal são calculadas as LLRs que são inerentes ao código através da matriz de paridade e das informações iniciais fornecidas pelo canal. No passo vertical, a informação interna ao código é somada com a informação externa, de modo que, se a LLR total de um bit for positiva, o bit é 0, caso contrário o bit é 1. No final de cada iteração é feita uma verificação para checar se a palavra já está correta, bastando verificarmos se $\hat{w}H^T = 0$. O algoritmo como um todo necessita somente de 3 operações: soma de ponto flutuante, multiplicação de sinais (utilizando "ou exclusivo") e o *soft XOR*. Este último é definido como $\phi(\phi(x) + \phi(y))$, onde $\phi(x) = -\log(tanh(x/2))$. Visto que a função $\phi(x)$ é chamada exaustivamente durante o algoritmo e dado o alto custo computacional para calculá-la, para a implementação foi construída uma *look up table* (Tabela de consulta rápida) com valores pré-calculados.

III. GNU RADIO

A plataforma GNU Radio é um ambiente que vem sendo desenvolvido por colaboradores do projeto GNU e oferece diversas ferramentas para simulações em *software* de estruturas de telecomunicações, sendo possível manipular blocos individualmente, alterando seus valores para que se obtenha a saída desejada ou se analise as saídas geradas [7]. A vantagem que essa plataforma possui é a facilidade de construção de blocos pelo usuário e o fato de o sistema GNU Radio possuir sua própria estrutura interna de tipos de dados.

De modo geral, o funcionamento é como segue: os blocos escritos em C++ contêm o funcionamento do bloco, ou seja, que tipo de tratamento o sinal terá. Os blocos em C++ são, então, agregados em um grafo escrito em Python, responsável por realizar o fluxo de dados e permitindo assim a conexão entre os blocos [8]. Sendo assim, foi implementado no GNU Radio apenas um bloco contendo tanto o codificador quanto o decodificador LDPC. A decisão acontece na construção através de um parâmetro booleano. Após o bloco ser construído, é necessário informar qual matriz de paridade será usada e carregá-la. Após carregada a matriz, o bloco pode codificar/decodificar, recebendo *bits* ou números reais na entrada e colocando *bits* na saída.

Além disso, o código dá suporte a diferentes modulações, como BPSK, 4, 16, e 64 QAM, em blocos responsáveis por calcular as LLR necessárias para a decodificação.

IV. RESULTADOS

Como resultados tem-se a implementação de um codificador com complexidade linear em relação ao tamanho da palavra código, baseado em códigos em blocos LDPC quase cíclicos. Implementou-se também um decodificador LDPC padrão, com a vantagem de ter a implementação da *look up table*. Desse modo, foi feita uma simulação simples, utilizando modulação BPSK em um canal AWGN com *soft bits* na saída, ou seja, LLRs na saída do canal ao invés de bits. Foram testadas as matrizes do padrão G.hn de acordo com a Figura 1. O padrão G.hn fornece 7 matrizes de paridade, sendo uma usada somente para transmitir cabeçalhos, logo, para a transmissão de dados contínuos, são utilizadas as outras 6 matrizes. São matrizes com taxa de codificação $1/2$, $2/3$ e $5/6$, sendo que para cada taxa existem duas matrizes, uma com tamanho da palavra código grande (L) e outra com tamanho pequeno (S). No gráfico, pode-se notar que ao se aumentar a taxa de codificação, a capacidade de correção diminui, e para cada taxa, quanto maior o tamanho da palava código, maior a capacidade de corrigir erros. Por exemplo, a matriz com maior capacidade de correção é a de menor taxa $(1/2)$ e maior comprimento da palavra código (L).

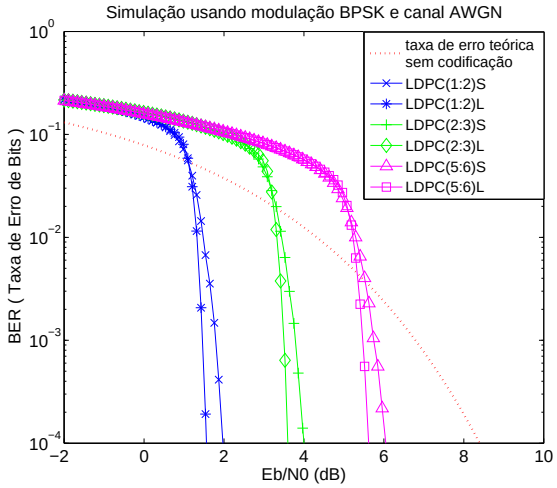

Fig. 1. BER x E_b/N_0 para as diferentes matrizes com diferentes taxas e tamanhos de blocos, utilizando BPSK em um canal AWGN.

V. CONCLUSÃO

A plataforma GNU Radio mostrou-se uma excelente ferramenta de simulação para esquemas de codificação de canal. Os blocos construídos para codificação e decodificação de LDPC em C++ foram facilmente interligados com os blocos para processamento de sinais já existentes no GNU Radio através do uso da linguagem Python. Além disso, a plataforma é livre (licença GPL), o que permite o acesso ao código fonte da plataforma e possibilita a personalização do GNU Radio de acordo com as necessidades do usuário.

REFERÊNCIAS

[1] ITU-T Recommendation G.9960, *Unified high-speed wire-line based home networking transceivers - System architecture and physical layer specification*

[2] S. Myung, K. Yang, and J. Kim, *Quasi-Cyclic LDPC Codes for Fast Encoding*, IEEE Trans. Inf. Theory, vol. 51, no. 8, pp. 2894-2900, Aug. 2005.

[3] B. Le, T. Rondeau e C. W. Bostian, *Cognitive Radio Realities*. Wiley InterScience, Maio 2007.

[4] R. G. Gallager, *Low density parity check codes*, IRE Trans. Inform Theroy, vol. IT-8, pp.21-28, Jan. 1962.

[5] D.J.C. MacKay, *Good error-correcting codes based on very sparse matrices*, IEEE Trans. Inf. Theory, vol.45, no.2, pp.399:432, 1999.

[6] T. J. Richardson and R. Urbanke, *Efficient encoding of low-density parity-check codes*, IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 638:656, Feb. 2001.

[7] D. Shen, *Entering the World of GNU Software Radio*. Agosto 2005.

[8] E. Blossom, *Exploring GNU Radio*, disponível em http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html. Acesso: Março 2012.