Códigos Verificadores de Paridade de Baixa Densidade

Hugerles Sales Silva[§], José Ewerton Pombo de Farias[*§] e Marcelo Sampaio de Alencar[*§]
[*]Universidade Federal de Campina Grande (UFCG), Campina Grande, Brasil
[§]Instituto de Estudos Avançados em Comunicações (Iecom)
E-mails: hugerles.silva@ee.ufcg.edu.br, {ewerton, malencar}@iecom.org.br

***Resumo*— Neste artigo são apresentados aspectos gerais sobre a construção, codificação e decodificação dos códigos verificadores de paridade de baixa densidade (LDPC – *Low-Density Parity-Check*). Essa classe de códigos, conhecida por proporcionar excelentes desempenhos quando aplicados em uma variedade de canais de comunicação, é baseada em matrizes de verificação de paridade esparsas. Quando comparado a outros esquemas de codificação, os códigos LDPC apresentam uma maior capacidade de correção de erros e um algoritmo de decodificação menos complexo. Um algoritmo foi construído no MatLab para avaliar o desempenho da taxa de erro de bit (BER – *Bit Error Rate*) para enlaces ópticos. Resultados recentemente publicados foram utilizados para validar o algoritmo.**

***Palavras-chave*— Algoritmo Soma-Produto, Códigos LDPC, Gráficos de Tanner.**

I. Introdução

CÓDIGOS para correção de erros são essenciais para transmitir informação de forma confiável em praticamente todos os sistemas de comunicações digitais [1]. Algoritmos para correção de erros utilizam informações redundantes codificadas para fazer a verificação e correção de erros no momento da recepção, podendo aumentar significativamente a eficácia na transmissão de dados. Atualmente, quase todos os sistemas de envio de informações, como telefonia digital, transmissão de dados via satélite, entre outros, possuem algum tipo de código para correção de erros [2].

Uma importante classe de códigos para correção de erros usa os códigos verificadores de paridade de baixa densidade, usualmente chamados códigos LDPC. Os códigos LDPC foram descobertos por Robert Gallager [3], em 1963, e constituem um conjunto definido a partir de matrizes de verificação de paridade esparsas que apresentam excelentes desempenhos em uma grande variedade de canais [4]. Os códigos LDPC, quando comparados a códigos clássicos, possuem maior capacidade de correção, sendo seu diferencial o mecanismo de decodificação iterativo baseado nas probabilidades das mensagens recebidas [5].

Na época de sua descoberta, os códigos LDPC não eram factíveis de implementação computacional e acabaram sendo esquecidos por anos, até que R. M. Tanner [6], na década de 1980, introduziu uma representação gráfica para códigos de blocos, intitulada grafos de Tanner, que é útil nos algoritmos de decodificação dos códigos LDPC. Mackay, na década de 1990, mostrou que esses códigos atingem um desempenho muito próximo ao limite de Shannon, quando decodificados com o algoritmo Soma-Produto (SP – *Sum-Product*) [7].

Os códigos LDPC possuem alta capacidade de correção, menores patamares de erros quando comparados aos códigos turbo e, quando decodificados com o algoritmo SP, podem atingir probabilidades de erro pequenas quando o comprimento do código aumenta [4]. Essa classe de códigos é amplamente utilizada atualmente, possuindo numerosas aplicações em vários padrões de transmissão, incluindo a segunda geração de transmissão de vídeo digital por satélite (DVB-S2 – *Digital Video Broadcasting-Satellite 2nd Generation*) [8].

Além dessa seção introdutória, o artigo está dividido em mais oito seções. A Seção II aborda a teoria relacionada aos códigos de blocos lineares, introduzindo conceitos de matriz geradora do código e matriz de verificação de paridade. A Seção III apresenta a teoria relacionada à representação gráfica de códigos LDPC e o conceito de giro de grafos. As Seções IV-VII descrevem aspectos gerais sobre os códigos LDPC, bem como a teoria para a construção, codificação e decodificação desses códigos. A Seção VIII apresenta resultados de simulações com aplicação de códigos LDPC em enlaces ópticos. Por fim, a Seção IX expõe as conclusões.

II. Códigos de Blocos Lineares

Um código de bloco binário possui k *bits* de informação. A codificação de bloco atribui à sequência de k *bits* de informação uma palavra código com n *bits* codificados, com $n > k$ [9]. A taxa de codificação de um código de bloco linear é definida por $R = k/n$.

Um código de bloco binário de tamanho M e comprimento de bloco n é um conjunto de M palavras binárias de comprimento n, chamadas palavras código. Usualmente, $M = 2^k$ para um inteiro k [10]. Em geral, é possível definir códigos de blocos sobre um alfabeto finito arbitrário. Diz-se então que esse alfabeto possui q símbolos $\{0, 1, \cdots, q-1\}$.

Um código de bloco linear possui três parâmetros: o comprimento do bloco n, o tamanho dos dados k e a distância mínima $d_{\min}$. A distância mínima é a menor distância de Hamming, que corresponde ao número de posições em que duas palavras código diferem. Um código que atende a esses requisitos é dito código de bloco $(n, k, d_{\min})$.

A. Matriz Geradora

A palavra código em um código de bloco linear (n, k) pode ser expressa por um vetor de comprimento n descrito por [9]

$$\mathbf{c} = [c_0\, c_1\, \cdots\, c_{n-1}] = \mathbf{mG}, \tag{1}$$

em que **m** é o vetor de dados de comprimento k e **G** é uma matriz de ordem $k \times n$. Devido ao fato de cada palavra código ser gerada a partir do produto do vetor de dados **m** pela matriz **G**, ela é então designada matriz geradora do código [9].

Para um código sistemático, ou seja, os *bits* de mensagem são separados dos *bits* de paridade, a palavra código é do tipo [9]

$$\mathbf{c} = [b_0\, b_1\, \cdots\, b_{n-k-1}\, m_0\, m_1\, \cdots\, m_{k-1}] = [\mathbf{b} \vdots \mathbf{m}], \tag{2}$$

em que **b** é o vetor paridade de comprimento $n - k$ e o símbolo $\vdots$ representa um particionamento na matriz. A partir da Equação (1) é possível verificar que cada *bit* da palavra código é uma combinação linear dos *bits* de mensagem.

É possível obter uma relação entre os vetores **b** e **m**. Essa relação é definida por

$$\mathbf{b} = \mathbf{mP}, \tag{3}$$

em que **P** é a matriz de paridade de dimensão $k \times (n - k)$. Nesse caso, a matriz geradora do código pode ser representada na forma sistemática

$$\mathbf{G} = [\mathbf{P} \vdots \mathbf{I}_k], \tag{4}$$

em que $\mathbf{I}_k$ é a matriz identidade de dimensão $k \times k$ [9].

B. Matriz de Verificação de Paridade

Seja **H** uma matriz contendo $n-k$ vetores base como linhas, então, uma n-upla **c** é uma palavra código se, e somente se, ela for ortogonal a todo vetor linha de **H**, ou seja,

$$\mathbf{cH}^T = \mathbf{0}. \tag{5}$$

A matriz **H** é denominada matriz de verificação de paridade do código (n, k) de dimensão $(n - k) \times n$.

A Equação (5) é uma condição necessária e suficiente para que **c** seja uma palavra do código (n, k) gerado pela matriz **G** [9]. Como a relação $\mathbf{cH}^T = 0$ é satisfeita quando **c** é ortogonal a qualquer linha de **G**, então

$$\mathbf{GH}^T = \mathbf{0}. \tag{6}$$

Um código linear, definido por uma matriz **H**, é sempre sistematizável e, nesse caso, por aplicação de operações lineares às linhas de **H**, essa pode ser escrita na forma sistemática

$$\mathbf{H} = [\mathbf{I}_{n-k} \vdots \mathbf{P}^T], \tag{7}$$

no qual $\mathbf{I}_{n-k}$ é a matriz identidade de dimensão $(n - k) \times (n - k)$ [9].

C. Detecção de Erro: Síndrome

Se, em determinado sistema de comunicação, o canal introduz ruído, então a palavra recebida é a transmitida acrescida do ruído, ou seja,

$$\mathbf{y} = \mathbf{v} + \mathbf{e}, \tag{8}$$

em que **e** é o vetor de erro, **v** é a palavra código e **y** é a palavra recebida.

Por meio da síndrome, denotada por **s**, é possível verificar se o vetor recebido **y** contém ou não erros. A síndrome pode ser obtida por meio da expressão

$$\mathbf{s} = \mathbf{yH}^T, \tag{9}$$

em que **H** é a matriz de verificação de paridade para um código LDPC (n, k) [9].

Substituindo a Equação (8) em (9) é possível verificar que

$$\mathbf{s} = \mathbf{yH}^T = \mathbf{(v + e)H}^T = \mathbf{vH}^T + \mathbf{eH}^T = \mathbf{eH}^T. \tag{10}$$

A síndrome contém alguma informação sobre o comportamento padrão de erros, embora geralmente insuficiente para o identificar sem ambiguidade. Um código (n, k) tem 2^k palavras admissíveis, em um total de 2^n palavras binárias de comprimento n. Conclui-se então que existem 2^{n-k} síndromes distintas [5].

O cálculo da síndrome é utilizado como a primeira etapa do processo de decodificação. Caso a síndrome encontrada seja nula, o vetor recebido **y** é considerado uma palavra código e admitido como vetor decodificado. Em caso contrário, admite-se que o vetor recebido contém erros, inseridos pelo canal de comunicação, e será necessária uma nova etapa de decodificação [11].

III. Grafos de Tanner

R. M. Tanner, na década de 1980, introduziu uma representação gráfica para os códigos LDPC [6]. Essa representação é útil na descrição eficiente de algoritmos de decodificação dos códigos LDPC e é conhecida por grafo de Tanner. Tanner considerou que qualquer código de bloco linear, em particular o código LDPC, podia ser representado por um grafo [12].

Os grafos de Tanner são bipartidos, possuindo dois conjuntos de nós. O grafo possui n nós de variáveis, denotados por v, e m nós de verificação, denotados por c, para as equações de paridade. O grafo de Tanner é desenhado de acordo com a seguinte regra: um nó de verificação c está conectado ao nó da variável v sempre que o elemento h_{cv} em uma matriz de paridade **H** é 1. Logo, as posições dos 1s da matriz de verificação **H** definem as interconexões entre os nós de variáveis e os nós de verificação. Em uma matriz de paridade de ordem $m \times n$, existem $m = n - k$ nós de teste e n nós de variável [12].

Considere que, para um código LDPC, a matriz de paridade **H** seja definida por

$$\mathbf{H}_{4\times 6} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{11}$$

O grafo de Tanner, associado à matriz **H**, descrita pela Matriz (11), é ilustrado na Figura 1. Os círculos representam os nós da variável, enquanto os quadrados representam os nós de teste. A partir de um grafo de Tanner, é possível obter a matriz de paridade **H** do código e vice-versa.

O grafo de Tanner possui propriedades úteis que incrementam o desempenho de correção de erros do código [12].

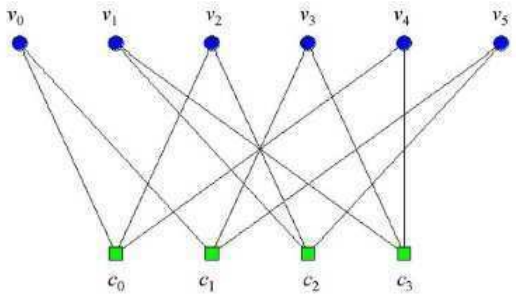

Fig. 1: Gráfico de Tanner.

A. Giro

Um giro em um grafo de Tanner é um caminho fechado formado por l enlaces. Um giro possui comprimento l se ele é um ciclo fechado por l enlaces. Logo, o valor de l é par e seu menor valor é 4. Para $l = 4$ significa que na matriz de verificação de paridade, **H**, duas colunas possuem um 1 em comum.

A distância mínima dos códigos LDPC e os procedimentos de decodificação são influenciados pelo giro do código [12]. Em códigos LDPC é preferível o uso de grandes giros, visto que eles incrementam a distância mínima e o desempenho do algoritmo SP. Tanner mostrou que o algoritmo SP possui desempenho ótimo quando aplicado a grafos sem ciclos [6].

Diversas técnicas são propostas na literatura para se determinar o giro associado a cada nó [13-16]. Y. Mao propõe construir um esquema de árvore para cada nó da variável [17]. MacGowan propõe a determinação do giro a partir do conceito de matriz adjacente [18].

IV. Códigos LDPC Regulares e Irregulares

Os códigos LDPC foram criados por Gallager [3] na sua tese de doutorado em 1960, e por mais de 30 anos foram esquecidos pela área científica. Eles são códigos de blocos lineares [11].

Um código de bloco linear *C* de tamanho n é unicamente especificado por sua matriz geradora **G** e por sua matriz de paridade **H** [9]. Códigos LDPC são especificados em termos da sua matriz de verificação de paridade, de dimensão $(n-k)\times n$.

Os códigos LDPC podem ser divididos em regulares ou irregulares de acordo com a regularidade dos elementos das matrizes de verificação de paridade [2]. Um código LDPC regular é definido como o espaço nulo de uma matriz que possui as seguintes propriedades [3]: (1) cada linha consiste de ρ 1s; (2) cada coluna consiste de γ 1s; (3) o número de 1s em comum entre duas colunas, denotado por λ, é no máximo 1, isto é, $\lambda = 0$ ou $\lambda = 1$; (4) ρ e γ são muito pequenos quando comparados com o comprimento do código e o número de linhas em **H**.

Isso significa que em um código LDPC, cada símbolo pertencente a uma palavra código é envolvido por γ equações de paridade e cada equação de paridade envolve ρ símbolos de uma palavra código. Um código com essas características é chamado de código LDPC (ρ,γ)-regular de comprimento n. Devido às propriedades (1) e (2), uma matriz de paridade **H**, possui o peso das linhas e colunas, ρ e γ respectivamente, constantes. A propriedade (3) implica que duas linhas de **H** têm, no máximo, um 1 em comum. Essa regra é chamada de restrição linha-coluna [19]. Devido ao fato de que ρ e γ são pequenos quando comparados com o tamanho do código e com o número de linhas da matriz de paridade, então **H** possui uma pequena densidade de uns. Por essa razão, **H** é dito ser uma matriz de verificação de paridade de baixa densidade.

A densidade da matriz de verificação de paridade **H** é dada por [3]

$$r = \frac{\rho}{N} = \frac{\gamma}{M}. \tag{12}$$

Por **H** se tratar de uma matriz esparsa, suas linhas não são necessariamente linearmente independentes em um campo de Galois $GF(q)$.

A taxa R_d de um código regular é determinada por [20]

$$R_d = 1 - \frac{\rho}{\gamma}. \tag{13}$$

R_d é denominada taxa de projeto de um código LDPC. A taxa de codificação R de um código de bloco linear, definida por

$$R = k/n, \tag{14}$$

é igual à taxa de projeto se **H** possui posto completo. Para códigos de bloco, quanto maior a distância mínima, maior é a capacidade do código de detectar e corrigir erros [5].

Robert Gallager demonstrou na sua tese que em códigos regulares a distância mínima não pode aumentar mais do que logaritmicamente com o comprimento n se $\rho = 2$. No entanto, se $\rho \geq 3$ a distância mínima aumenta linearmente com n para γ e ρ constantes [20].

Se a matriz de verificação de paridade **H** possui peso variável para as linhas e colunas, diz-se então que se trata de um código LDPC irregular. Códigos LDPC irregulares possuem desempenho muito próximo do limite de Shannon [21]. Esses códigos são comumentes projetados e construídos em termos de grafos de Tanner.

Em um grafo de Tanner de um código LDPC irregular, com matriz de verificação de paridade **H**, tem-se que os nós da variável correspondem às colunas de **H**. O grau de um nó nesse grafo é definido como o número de nós conectados a ele. O grau do nó da variável é exatamente igual ao peso da correspondente coluna em **H** e o grau do nó de teste é exatamente igual ao peso da correspondente linha em **H** [3].

Um conjunto de códigos LDPC irregulares é definido a partir de distribuições de grau polinomiais $\gamma(x)$ e $\rho(x)$, para os nós de variável e para os nós de teste respectivamente, definidas por

$$\gamma(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1} \tag{15}$$

e

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1}, \tag{16}$$

em que λ_d e ρ_d denotam, respectivamente, a fração de ramos no grafo que são conectados aos nós da variável e de teste. As constantes d_v e d_c denotam, respectivamente, o grau máximo dos nós da variável e dos nós de teste.

A taxa R_d para os códigos irregulares é definida por [12]

$$R_d = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \gamma(x)dx}. \tag{17}$$

V. Construção de códigos LDPC

A construção de códigos LDPC se dá na geração da matriz **H** ou de um grafo de Tanner [11]. A primeira forma de construção requer que a matriz de verificação de paridade **H** atenda a um conjunto de requisitos previamente solicitado e a segunda maneira requer que um giro mínimo para o código seja assegurado.

Para cada conjunto de restrições solicitado, o conjunto de códigos que satisfazem as exigências impostas é elevado e torna-se necessário dispor de alguns critérios de seleção para escolher um código que apresente desempenho ótimo. Alguns critérios para avaliação de códigos LDPC são: 1) obter um desempenho próximo da capacidade; 2) obter um desempenho com baixos patamares de erro; 3) obter uma estrutura que permita uma codificação eficiente. Diferentes abordagens para a construção de códigos LDPC são descritas na literatura [4,5,11,19]. Códigos LDPC com desempenho ótimo apresentam distância mínima elevada.

A. Códigos de Gallager

A ideia de que a matriz de verificação de paridade satisfaça um conjunto de exigências foi originalmente proposta por Gallager [3]. Os códigos de Gallager são códigos que possuem matrizes **H** esparsas, entretanto, elas não são sistemáticas [11]. Para a construção do código, Gallager estabeleceu inicialmente que k fosse um inteiro positivo maior que 1 e introduziu a notação de uma tupla (n, ρ, γ), em que n é o tamanho da palavra código, ρ e γ são, respectivamente, a quantidade de 1s em cada linha e a quantidade de 1s em cada coluna.

Para uma dada escolha de ρ e γ, Gallager [3] mostrou que uma matriz **H** de ordem $k\gamma \times k\rho$ consiste em γ submatrizes de ordem $k \times k\rho$: $\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_\gamma$. Cada linha da submatriz possui ρ 1s em cada linha e um único 1 em cada coluna. Logo, cada submatriz possui um total de $k\rho$ 1s. As outras submatrizes podem ser formadas permutando colunas de $\mathbf{H}_1$. A matriz de verificação **H** é definida como

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \mathbf{H}_3 \\ \mathbf{H}_4 \\ \vdots \\ \mathbf{H}_\gamma \end{bmatrix}. \tag{18}$$

O número total de 1s na matriz de verificação de paridade **H** é $k\rho\gamma$ e o número total de entradas é $k^2\rho\gamma$. Logo, a densidade da matriz é $1/k$. Se k é muito maior que 1, então **H** possui uma pequena densidade e a matriz é esparsa [3].

O número de 1s em comum entre duas colunas quaisquer na matriz de verificação de paridade para códigos LDPC não pode ser maior que 1. Isso depende claramente da escolha das $\gamma - 1$ permutações de colunas da submatriz de $\mathbf{H}_1$. Permutações aleatórias das colunas de $\mathbf{H}_1$, para formar outras submatrizes, resultam em uma classe de códigos de blocos lineares que contêm uma subclasse de códigos LDPC.

As $\gamma - 1$ permutações devem ser escolhidas de modo que o código gerado pela matriz **H** possua uma distância mínima elevada e seu grafo de Tanner não possua ciclos curtos, especialmente ciclos de comprimento quatro [19, 22].

No final da construção, a matriz construída atende à restrição linha-coluna, e a seguinte relação é satisfeita

$$\frac{m}{n} = \frac{\gamma}{\rho}. \tag{19}$$

A técnica de Gallager oferece baixos patamares de erro e possui baixa complexidade para implementação quando comparado à outras técnicas de construção de códigos LDPC [19].

B. Códigos de Mackay

Conjuntos de estratégias para gerar bons códigos LDPC são descritos por Mackay [7]. A complexidade de construção pelo método de Mackay é maior comparada com o algoritmo de Gallager, mesmo assim não é garantido que sempre se obterá um desempenho melhor [19]. As estratégias de construção são:

1) A matriz **H** é gerada a partir de uma matriz de zeros de dimensão $(n-k) \times n$ e aleatoriamente adicionandos ρ *bits* em cada coluna.
2) A matriz **H** é produzida ao criar aleatoriamente colunas de peso de Hamming W_ρ.
3) A matriz **H** é gerada com colunas aleatórias de peso de Hamming W_ρ e procurando uniformizar ao máximo o peso de Hamming W_γ de cada linha.
4) A matriz **H** é formada com colunas de peso de Hamming W_ρ, linhas de peso de Hamming W_γ, e não possuindo quaisquer duas colunas com mais de um 1 em comum.
5) A matriz **H** é gerada respeitando as regras anteriores, tendo como objetivo a maximização do giro do código.

Realizado esse conjunto de procedimentos, obtém-se a matriz verificação de paridade **H**.

VI. Codificação de códigos LDPC

Diferentemente dos códigos lineares, a matriz de verificação de paridade **H** de um código LDPC não é sistemática, isto é, não é possível distinguir diretamente os *bits* de mensagem dos *bits* de paridade [5]. Para a codificação, pode-se obter uma matriz geradora **G** para códigos LDPC com o método de eliminação de Gauss, com operação módulo 2 [4]. Porém, utilizando esse método, a esparsidade de **H** tende a ser perdida, descaracterizando a matriz de verificação de paridade LDPC. A matriz geradora para códigos LDPC também pode ser obtida por meio de métodos algébricos e geométricos em que a

codificação possa ser realizada por circuitos simples baseados em registradores de deslocamentos [5].

Richardson e Urbanke em 2001 [21] propuseram um método eficiente de codificação para códigos LDPC para qualquer matriz de verificação de paridade **H**, baseado em uma matriz triangular inferior aproximada, utilizando operações elementares e permutações de linhas e colunas, de modo a não afetar drasticamente o desempenho e nem a estrutura esparsa de **H**.

Um método alternativo para geração de **G** é descrito em [23]. Nesse método, a palavra código em um código LDPC pode ser particionada em termos dos vetores **m** e **b** como

$$\mathbf{c} = \left[\begin{array}{c|c} \mathbf{b} & \mathbf{m} \end{array} \right], \tag{20}$$

em que **m** é o vetor de dados de comprimento k e **b** é o vetor paridade de comprimento $(n-k)$. A matriz de verificação de paridade **H** pode ser fracionada conforme apresentado na Matriz (21).

$$\mathbf{H}^T = \left[\begin{array}{c} \mathbf{H}_1 \\ \cdots \\ \mathbf{H}_2 \end{array} \right], \tag{21}$$

em que $\mathbf{H}_1$ é uma matriz quadrada de dimensão $(n-k)\times(n-k)$ e $\mathbf{H}_2$ é uma matriz retangular de dimensão $k \times (n-k)$. A transposição mostrada na Equação (21) é usada no particionamento da matriz **H** para facilitar a apresentação e para atender à relação

$$\mathbf{cH}^T = \mathbf{mGH}^T = 0. \tag{22}$$

A partir da Equação (22) pode-se obter a relação entre os vetores **b** e **m** dada por [9]

$$\mathbf{b} = \mathbf{mP}, \tag{23}$$

em que **P** é a matriz de paridade. Substituindo a Equação (23) em (22), tem-se $\mathbf{PH}_1 + \mathbf{mH}_2 = 0$ e portanto $\mathbf{P} = \mathbf{H}_2\mathbf{H}_1^{-1}$, em que $\mathbf{H}_1^{-1}$ é a matriz inversa de $\mathbf{H}_1$. Todas as operações são realizadas em GF(2).

Assim, a matriz geradora de códigos LDPC é definida por [10]

$$\mathbf{G} = \left[\begin{array}{c|c} \mathbf{P} & \mathbf{I}_k \end{array} \right] = \left[\begin{array}{c|c} \mathbf{H}_2\mathbf{H}_1^{-1} & \mathbf{I}_k \end{array} \right] \tag{24}$$

em que $\mathbf{I}_k$ é a matriz identidade de ordem $k \times k$.

Com a matriz geradora **G** definida, é possível codificar as palavras código a partir da relação

$$\mathbf{c} = \mathbf{mG} \tag{25}$$

em que **m**, a palavra de dados, é uma k-upla dos símbolos de informação a serem codificados e a n-upla **c** é a correspondente palavra código [24]. A matriz geradora **G** presente na Equação (25) está na forma sistemática e as suas linhas são linearmente independentes.

VII. Decodificação de códigos LDPC

Gallager propôs dois algoritmos para decodificação de códigos LDPC, um baseado em decisão abrupta e outro em decisão suave, ambos obtendo excelentes desempenhos no processo de decodificação [3]. O mais comum e o mais utilizado em telecomunicações é algoritmo Soma-Produto (SP) [4]. O SP baseado em decisão abrupta considera que o número de *bits* que chegam ao decodificador é finito, diferente do SP baseado em decisão suave que trabalha com a distribuição probabilística dos símbolos recebidos. O processo de decodificação por decisão suave funciona de forma iterativa.

A decodificação iterativa permite que o vetor recebido seja analisado várias vezes, até que se encontre um vetor considerado decodificado ou que seja declarado um erro, caso seja excedido o número máximo de iterações permitidas. Essa decodificação iterativa acontece com a troca de informações entre os nós de paridade e os nós de verificação, por intermédio de ligações determinadas durante a construção do grafo de Tanner [20].

Um decodificador baseado no algoritmo SP processa os símbolos recebidos iterativamente para aumentar a confiabilidade de cada símbolo decodificado. A primeira etapa de decodificação consiste em representar a distribuição de probabilidade conjunta *a posteriori* dos símbolos das palavras código transmitidas $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ condicionadas à saída do canal $\mathbf{y} = (y_1, y_2, \cdots, y_N)$. Para variáveis binárias, cada mensagem no algoritmo SP pode ser representada como o logaritmo da razão entre as duas componentes do vetor que a representa [3].

Seja X uma variável aleatória binária, o logaritmo da razão de verossimilhança (LLR – *LogLikelihood Ratio*) de X é definido por [4]

$$L(X) = \log\left(\frac{P_X(0)}{P_X(1)}\right),$$

em que $P_X(x)$ denota a probabilidade de X assumir o valor x. O sinal do logaritmo da razão de verossimilhança $L(X)$ determina qual o valor mais provável que X pode assumir e é a partir dele que é realizada a decisão. Se a variável aleatória binária X for condicionada a uma variável aleatória discreta Z, então

$$\begin{aligned} L(X|Z) &= \log\left(\frac{P_{X|Z}(0|z)}{P_{X|Z}(1|z)}\right) \\ &= \log\left(\frac{P_X(0)}{P_X(1)}\right) + \log\left(\frac{P_{Z|X}(z|0)}{P_{Z|X}(z|1)}\right) \\ &= L(X) + L(Z|X). \end{aligned} \tag{26}$$

Se os valores assumidos pela variável aleatória X forem equiprováveis, então é possível expressar a distribuição de probabilidade de X em função de $L(X)$, ou seja,

$$P_X(0) = \frac{e^{L(X)}}{e^{L(X)}+1}$$

e

$$P_X(1) = \frac{1}{e^{L(X)}+1}.$$

O que implica

$$P_X(0) - P_X(1) = \frac{e^{L(X)}-1}{e^{L(X)}+1} = \tanh\left[\frac{L(X)}{2}\right]. \tag{27}$$

Supondo que X e Z são variáveis aleatórias binárias estatisticamente independentes, o LLR da soma módulo 2 dessas variáveis é calculado por

$$P(X \oplus Z = 0) = P_{X,Z}(0,0) + P_{X,Z}(1,1) = P_X(0) \cdot P_Z(0) + P_X(1) \cdot P_Z(1) \quad (28)$$

e

$$P(X \oplus Z = 1) = P_{X,Z}(0,1) + P_{X,Z}(1,0) = P_X(0) \cdot P_Z(1) + P_X(1) \cdot P_Z(0) \quad (29)$$

Usando o mesmo argumento da Equação (27), obtém-se

$$\tanh\left[\frac{L(X \oplus Z)}{2}\right] = P(X \oplus Z = 0) - P(X \oplus Z = 1) = (P_X(0) - P_X(1))(P_Z(0) - P_Z(1)) = \tanh\left[\frac{L(X)}{2}\right] \cdot \tanh\left[\frac{L(Z)}{2}\right]. \quad (30)$$

Portanto,

$$L(X \oplus Z) = 2\tanh^{-1}\left\{\tanh\left[\frac{L(X)}{2}\right] \cdot \tanh\left[\frac{L(Z)}{2}\right]\right\}. \quad (31)$$

A Equação (31) é chamada de regra da tangente hiperbólica [4]. Essa regra é usada pelo algoritmo SP no domínio LLR para calcular as atualizações das mensagens nos nós de verificação.

A. Descrição do Algoritmo Soma-Produto

Na descrição do algoritmo SP, denota-se por $L(q_{n\to m})$ a mensagem enviada do nó de verificação c_n para o nó de paridade v_m e esta corresponde ao logaritmo da razão de verossimilhança da variável c_n, e por $L(r_{m\to n})$ a mensagem enviada do nó v_m para o nó c_n [4]. O algoritmo SP no domínio LLR é descrito a seguir.

1) **Início** - Cada função $p(y_n|x_n)$ passa ao seu único vizinho um LLR inicial

$$L(y_n|x_n) = \log\left(\frac{p(y_n|x_n = 0)}{p(y_n|x_n = 1)}\right) \quad (32)$$

para os nós de variável adjacentes. Então, todos nós de variável passam para seus vizinhos a mensagem

$$L(q_{n\to m}) = L(y_n|x_n). \quad (33)$$

2) **Atualização para nós de verificação de paridade** - O m-ésimo nó de verificação de paridade recebe as mensagens $L(q_{n\to m})$, em que $n \in N(m)$, e atualiza as mensagens $L(r_{m\to n})$ segundo a seguinte equação

$$L(r_{m\to n}) = 2\tanh^{-1}\left[\prod_{n' \in N(m)\setminus n} \tanh\left(\frac{L(q_{n'\to m})}{2}\right)\right]. \quad (34)$$

3) **Atualização para nós de variável** - O n-ésimo nó de variável recebe as mensagens $L(r_{m\to n})$, em que $m \in M(n)$, e atualiza as mensagens $L(q_{n\to m})$ segundo a seguinte equação:

$$L(q_{n\to m}) = L(y_n|x_n) + \sum_{m' \in M(n)\setminus m} L(r_{m'\to n}). \quad (35)$$

4) **Terminação** - O decodificador determina a informação a *posteriori* total sobre o símbolo n usando a soma das mensagens transmitidas por todos os nós de verificação a ele conectados.

$$D_n = L(y_n|x_n) + \sum_{m \in M(n)} L(r_{m\to n}). \quad (36)$$

A decisão $\hat{x}$ do vetor $D = (D_1, D_2, \cdots, D_N)$ é dada por

$$\hat{x}_n = \begin{cases} 1 & \text{se} \quad D_N \geq 0, \\ 0 & \text{se} \quad D_N < 0. \end{cases} \quad (37)$$

O algoritmo é encerrado quando a condição $\hat{\mathbf{x}}\mathbf{H}^T = 0$ ou um número máximo de iterações pré-definido é alcançado [4].

VIII. Aplicações de Códigos LDPC em Enlaces Ópticos

A. Algoritmo para Simulação dos Enlaces

A Figura 2 ilustra o diagrama de blocos do sistema implementado no MatLab para a realização das simulações e obtenção dos resultados. Nesse diagrama, a fonte de informação gera os *bits* a serem transmitidos. Esta fonte é caracterizada por um alfabeto, por uma taxa de informação e por uma distribuição de probabilidade de emissão de sequências de *bits*.

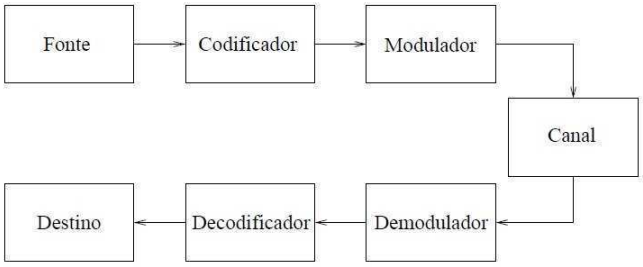

Fig. 2: Diagrama de blocos para o sistema implementado.

O codificador LDPC transforma uma sequência de k *bits* de informação em uma sequência codificada de n *bits*, denominadas palavra-código, em que $n > k$. A quantidade de *bits* redundantes introduzidos na sequência de informação é portanto $n - k$. A taxa de codificação é definida como $R = k/n$. Neste bloco, a matriz geradora do código LDPC é construída com base em um conjunto de exigências impostas sobre a matriz de verificação de paridade $\mathbf{H}$. Os k *bits* de informação são codificados utilizando a técnica de Gallager, levando em consideração o valor da redudância introduzida e a dimensão da matriz de verificação de paridade.

Os *bits* na saída do codificador LDPC entram em um modulador que tem a função de convertê-los em formas de onda analógicas para a transmissão. Neste bloco, os *bits* são agrupados e mapeados em símbolos complexos que variam a amplitude, fase ou frequência de uma senóide. Esses símbolos determinam os pontos da constelação de cada subportadora de acordo com o tipo de modulação empregada.

O canal de comunicação é o meio físico usado para se enviar o sinal do transmissor ao receptor. O meio de propagação

utilizado neste trabalho é a fibra óptica. Neste trabalho, os sinais propagantes na fibra são corrompidos de forma aleatória por ruído aditivo gaussiano branco (AWGN – *Additive White Guassian Noise*).

O demodulador converte a sequência de formas de ondas corrompidas pelas imperfeições do canal em uma sequência de símbolos. O processo de detecção pode produzir símbolos errôneos, devido à perturbação ruidosa. A sequência errônea chega ao decodificador LDPC que utiliza o algoritmo SP com um determinado número de iterações pré-estabelecido para realizar a decodificação. As curvas de desempenho de taxa de erro de *bit* (BER – *Bit Error Rate*) em função de E_b/N_0 para o cenário óptico simulado na Seção VIII-C são obtidas logo após a etapa de decodificação.

B. Modelo de Relação Sinal-Ruído Óptica

O modelo de relação sinal-ruído óptica (OSNR – *Optical Signal-to-Noise Ratio*) utilizado nas simulações da Seção VIII-C é dado por meio da fórmula

$$\text{OSNR} = \frac{E_b}{N_0} + 10\log_{10}\frac{V \cdot R \cdot R_s}{B_{\text{ref}}}, \qquad (38)$$

em que B_{ref} é a largura de faixa de referência do ruído, cujo valor utilizado foi 12,5 GHz, correspondente a 0,1 nm de largura de linha para a região de comprimento de onda de 1550 nm; R_s é a taxa dos símbolos transmitidos, R é a taxa do código LDPC e V corresponde ao número de *bits* associados a cada símbolo modulado [20]. A definição de OSNR na Fórmula (38) é aplicada para sistemas ópticos com qualquer formato de modulação, taxa de transmissão e tecnologia de amplificação [20].

C. Simulação de Enlaces Ópticos

Na literatura são descritas inúmeras aplicações de códigos LDPC em diversos sistemas de comunicações. Atualmente, tem-se destacado em diversos trabalhos a aplicação desses códigos em sistemas ópticos [20]. Nesta seção, simulações foram realizadas com o objetivo de reproduzir e agregar resultados ao sistema descrito em [20].

Em [20] os autores descrevem a aplicação de códigos LDPC em sistemas ópticos operando a 32 G*bit*/s com o objetivo de melhorar o desempenho do sistema de transmissão frente ao ruído AWGN. Nesse sistema considera-se que os dados são transmitidos a uma taxa de 32 G*bit*/s em um canal AWGN com densidade espectral de potência $N_0/2$ e são modulados utilizando a técnica BPSK com taxa de símbolos (R_s) de 32 Gbaud. O processo de codificação dos dados é realizado utilizando um código LDPC (3, 6)-regular construído por meio da técnica de Gallager, com taxa de projeto R_d = 1/2 e matriz de verificação de paridade $\mathbf{H}$ de ordem 15×36. A taxa de codificação dos dados é dada por $R = K/N \approx 0,583$, satisfazendo a condição que $R > R_d$. A decodificação dos dados é realizada com o algoritmo SP utilizando oito iterações. Os resultados da BER em função de E_b/N_0, para o sistema com e sem codificação descrito, utilizando as técnicas de modulação BPSK, QPSK e 16-QAM, podem ser vistos na Figura 3.

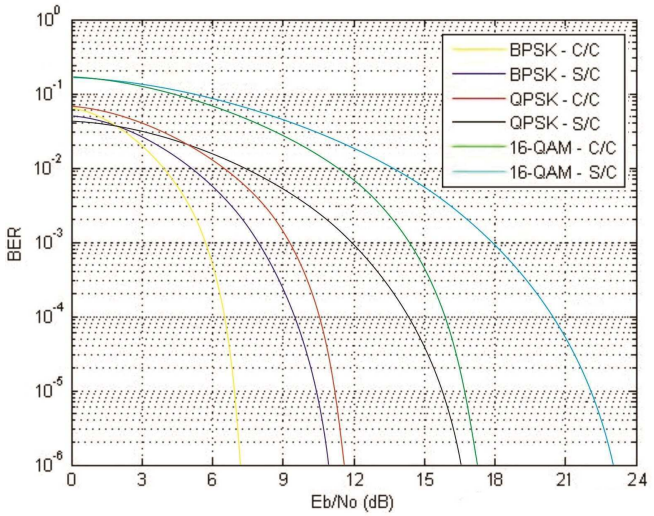

Fig. 3: Curva de desempenho: BER em função de E_b/N_0, para o enlace utilizando as técnicas BPSK, QPSK e 16-QAM com R_d = 1/2.

Na Figura 3, o desempenho obtido para o sistema codificado é superior ao sistema sem codificação para os três esquemas de modulação simulados. Nas simulações realizadas, ao se utilizar o esquema BPSK, um ganho de codificação de 3,7 dB é obtido para uma BER de 10^{-6}, para oito iterações do algoritmo SP. O ganho obtido para esse sistema é muito próximo de 3,6 dB, valor obtido por A. Leven em [20], evidenciando então que os resultados obtidos na simulação do sistema são coerentes. Para os esquemas de modulação de alta eficiência espectral, como o QPSK e o 16-QAM, percebe-se que maiores ganhos de codificação são obtidos em comparação ao BPSK, porém em contrapartida uma maior OSNR mínima é requerida para que o sinal possa ser recuperado pelo receptor e, consequentemente, menor o alcance do sistema. Para uma BER de 10^{-6}, o ganho obtido para o esquema QPSK é de 4,8 dB e para o 16-QAM de 5,4 dB. A Tabela I apresenta os valores de OSNR obtidos para o sistema simulado, em que OSNR_1, OSNR_2 e OSNR_3 referem-se, respectivamente, à OSNR para os esquemas BPSK, QPSK e 16-QAM. Para um dado valor de E_b/N_0, maiores valores de OSNR são obtidos para a técnica 16-QAM.

IX. Conclusões

Neste artigo foi apresentada a teoria matemática relacionada aos códigos LDPC, que são baseados em decodificação suave e proporcionam um ganho de codificação elevado quando comparados aos códigos de gerações anteriores. Resultados recentemente publicados foram reproduzidos e novas contribuições foram agregadas ao trabalho analisado. No enlace simulado, o desempenho para o sistema, quando a codificação LDPC foi aplicada, mostrou-se superior ao sistema sem codificação. Ao utilizar o esquema BPSK, um ganho de codificação de 3,7 dB foi obtido para uma BER de 10^{-6}, com oito iterações do algoritmo SP, com diferença de 0,1 dB do valor obtido por A. Leven em [20].

TABELA I: Valores de OSNR para o sistema simulado descrito em [20].

E_b/N_0	0	3	6	9	12	15	18	21	24	27
OSNR_1	1,74	4,74	7,74	10,74	13,74	16,74	19,74	22,74	25,74	28,74
OSNR_2	4,74	7,74	10,74	13,74	16,74	19,74	22,74	25,74	28,74	31,74
OSNR_3	7,76	10,76	13,76	16,76	19,76	22,76	25,76	28,76	31,76	34,76

AGRADECIMENTOS

Os autores agradecem à Universidade Federal de Campina Grande (UFCG), ao Instituto de Pesquisas Avançadas em Comunicações (Iecom), a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) e a Coordenadoria de Pós-Graduação em Engenharia Elétrica (Copele) pelo suporte dado à pesquisa.

REFERÊNCIAS

[1] R. E. Blahut. *Theory and Practice of Error Control Codes*. 2ª edição: Reading, MA: Addison-Wesley, 1987.

[2] S. Lin e D. J. C. Júnior. *Error Control Coding*. 2ª edição: Pearson Prentice Hall, 2011.

[3] R. G. Gallager. *Low-Density Parity-Check Codes*. 2ª edição: Cambridge, MA: MIT Press, 1963.

[4] M. M. Vasconcelos. *Decodificação Iterativa de Códigos Baseados em Matrizes de Verificação de Paridade Esparsas*. Dissertação de Mestrado – Universidade Federal de Pernambuco – UFPE, Recife, Brasil, 2006.

[5] F. J. F. Nazareth. *Estudo de Códigos LDPC (Low-Density Parity-Check): Matrizes Regulares e Irregulares*. Dissertação de Mestrado – Universidade Federal do Rio de Janeiro – UFRJ, Rio de Janeiro, Brasil, 2010.

[6] R. M. Tanner. *A Recursive Approach to Low Complexity Codes*. IEEE Transactions on Information Theory, vol. 27, nº 5, p. 533-547, Setembro 1981.

[7] D. J. C. Mackay e R. M. Neal. *Near Shannon Limit Perfomance of Low-Density Parity-Check Codes*. Eletronic Letters, vol. 32, nº 6, p. 1645-1646, Setembro 1996

[8] H. Hussien *et al*. *Performance Study on Implementation of DVB-S2 Low-Density Parity-Check Codes on Additive White Gaussian Noise Channel and Rayleigh Fading Channel*. IEEE International Conference on Electronics Design, Systems and Applications (ICEDSA), vol. 1, nº 30, p. 179-182, Novembro 2012.

[9] R. E. Blahut. *Algebraic Codes for Data Transmission*. 3ª edição: Cambridge University Press, 2003.

[10] R. E. Blahut. *Algebraic Codes on Lines, Planes and Curves*. 3ª edição: Cambridge University Press, 2008.

[11] F. P. Magalhães. *Análise de Desempenho de Algoritmos de Decodificação para Códigos LDPC Regulares*. Dissertação de Mestrado – Intituto Nacional de Telecomunicações – Inatel, Santa Rita do Sapucaí, Brasil, 2009.

[12] I. B. Djordjevic, M. Arabaci e L. L. Minkiv. *Next Generation FEC for High-Capacity Communication in Optical Transport Networks*. Journal of Lightwave Technology, vol. 27, nº 16, p. 3518-3530, Agosto 2009.

[13] X. Wu, X. You e C. Zhao. *An Efficient Girth-Locating Algorithm for Quasi-Cyclic LDPC Codes*. IEEE International Symposium on Information Theory, vol. 31, nº 3, p. 817-820, Julho 2006.

[14] X. Wu, X. You e C. Zhao. *A Necessary and Sufficient Condition for Determining the Girth of Quasi-Cyclic LDPC Codes*. IEEE Transactions on Communications, vol. 56, nº 6, p. 854-857, Junho 2008.

[15] T. R. Halford, K. M. Chugg e A. J. Grant. *Which Codes Have 4-Cycle-Free Tanner Graphs?* IEEE International Symposium on Information Theory, vol. 31, nº 3, p. 871-875, Julho 2006.

[16] B. Bollobas e E. Szemerdi. *Girth of Sparse Graphs*. Journal of Graph Theory, vol. 39, nº 1, p. 194-200, Março 2002.

[17] Y. Mao e A. H. Banihashemi. *A Heuristic Search for Good Low-Density Parity-Check Codes at Short Block Lenghts*. IEEE Internation Conference on Communication, vol. 1, nº 3, p. 41-44, Junho 2001.

[18] J. A. McGowan and R. C. Williamson. *Loop Removal from LDPC Codes*. IEEE Transactions on Information Theory, vol. 24, nº 2, p. 230-233, Abril 2003.

[19] L. F. Santos. *Decodificadores de Baixa Complexidade para Códigos LDPC Q-ários*. Dissertação de Mestrado – Universidade Estadual de Campinas – UNICAMP, Campinas, Brasil, 2014.

[20] A. Leven and L. Schmalen. *Status and Recent Advances on Forward Error Correction Technologies for Lightwave Systems*. Journal of Lightwave Technology, vol. 32, nº 16, p. 2735-2750, Agosto 2014.

[21] T. Richardson, M. Shokrollahi e R. Urbanke. *Design of Capacity Approaching Irregular Low-Density Parity-Check Codes*. IEEE Transactions on Information Theory, vol. 47, nº 3, p. 638-656, Fevereiro 2001.

[22] C. Berrou *et al*. *Computing the Minimum Distance of Linear Codes by the Error Impulse Method*. IEEE GLOBECOM, vol. 2, nº 1, p. 1017-1020, Novembro 2002.

[23] S. Haykin. *Communication Systems*. 4ª edição: John Willey and Sons: EUA, 2002.

[24] R. P. Júnior *et al*. *Fundamentos Algébricos e Geométricos dos Códigos Corretores de Erro*. 1ª edição: Editora da Universidade Estadual de Campinas – UNICAMP, 2006.