KỸ THUẬT PHÂN LỚP ĐỂ GIẢI MÃ HIỆU QUẢ MÃ LDPC TRONG HỆ THỐNG THÔNG TIN DI ĐỘNG 5G

Nguyễn Trọng Duy, Hồ Văn Khương*
Trường Đại học Bách khoa - ĐHQG TP.HCM
*Email: *hvkhuong@hcmut.edu.vn*
Ngày nhận bài: 28/01/2021; Ngày chấp nhận đăng: 05/3/2021

TÓM TẮT

Hệ thống thông tin di động thế hệ thứ 5 (5G - 5th Generation) phải đạt được 3 tiêu chí chính là băng thông rộng, độ tin cậy cao và độ trễ thấp. Mã kiểm tra chẵn lẻ mật độ thấp (LDPC - Low Density Parity Check) đã được chấp nhận cho hệ thống thông tin di động 5G vì mã LDPC gần đạt được dung lượng Shannon. Bài báo này đề xuất kỹ thuật phân lớp để giảm đáng kể thời gian giải mã và cải thiện tỷ lệ lỗi bit (BER - Bit Error Rate). Hiệu năng của kỹ thuật đề xuất được đánh giá theo nhiều thông số khác nhau như tỷ số năng lượng bit trên công suất nhiễu, độ dài từ mã và tỷ lệ mã hóa.

Từ khóa: Mã LDPC, 5G, BER, sum-product, phân lớp.

1. MỞ ĐẦU

Mã kiểm tra chẵn lẻ mật độ thấp (LDPC) lần đầu tiên được Gallager đề xuất vào đầu những năm 1960 và được MacKay & Neal xây dựng lại vào năm 1996, đã thu hút được nhiều sự quan tâm từ cả cộng đồng nghiên cứu lẫn giới công nghệ nhờ khả năng sửa lỗi đạt được gần giới hạn Shannon [1, 2]. Ngoài ra, mã LDPC cũng là một trong các loại mã sửa lỗi thuận (FEC - Forward Error Correction) được sử dụng rộng rãi nhất trong các chuẩn truyền thông như mạng cục bộ không dây (WLAN, IEEE 802.11n), mạng truy cập vô tuyến không dây (WRAN, IEEE 802.22), kỹ thuật phát video số (DVB) và hệ thống truyền hình tiên tiến (ATS - Advanced Television System) [3-8]. Trong những năm gần đây, hệ thống thông tin di động thế hệ thứ 5 (5G) được nghiên cứu, phát triển và triển khai. Mã LDPC đóng vai trò quan trọng trong giao tiếp 5G và đã được chọn cho việc mã hóa trong hệ thống thông tin di động 5G. Để hỗ trợ tương thích tốc độ và truyền dữ liệu có thể mở rộng, Dự án Đối tác Thế hệ thứ 3 (3GPP - 3rd Generation Partnership Project) đã đồng ý xem xét hai ma trận kiểm tra chẵn lẻ tương thích tốc độ, BG1 và BG2, cho mã hóa kênh [9-15]. Căn cứ vào BG1 và BG2, một số nghiên cứu đã được thực hiện trên các mã LDPC cho hệ thống thông tin di động 5G. Các mã kiểm tra chẵn lẻ mật độ thấp giả vòng (QC-LDPC - Quasi-cyclic LDPC) có nhiều ưu điểm so với các loại mã LDPC khác về hiện thực phần cứng của việc mã hóa và giải mã bằng cách sử dụng thanh ghi dịch đơn giản và các mạch luận lý.

Bài báo này có các đóng góp chính như sau:

- Hệ thống hóa các mã LDPC cho hệ thống thông tin di động 5G.
- Thực hiện mã hóa và giải mã mã LDPC dùng phần mềm Matlab.
- Mô phỏng được BER của mã LDPC theo các điều kiện vận hành khác nhau.
- Cải tiến giải thuật giải mã mã LDPC bằng kỹ thuật phân lớp để giảm thời gian giải mã và cải thiện BER.

Bài báo tiếp tục như sau: Phần 2 trình bày cấu trúc mã LPDC trong hệ thống thông tin di động 5G. Giải thuật mã hóa và giải mã được trình bày với các cải tiến trong phần 3. Tiếp theo, mô phỏng Matlab được thực hiện để kiểm tra các tính chất của mã LPDC trong phần 4. Phần này cũng cung cấp các kết quả của cả 2 loại ma trận kiểm tra chẵn lẻ của mã LPDC dành cho hệ thống thông tin di động 5G. Cuối cùng, phần 5 trình bày kết luận của nghiên cứu này.

2. CẤU TRÚC MÃ LDPC TRONG HỆ THỐNG THÔNG TIN DI ĐỘNG 5G

Công nghệ truy cập đánh dấu sự chuyển đổi trong mã hóa sửa sai thuận FEC cho 3GPP của công nghệ di động [9-15]. Trong phần này, các mã QC-LDPC được xem xét và các đặc điểm của mã QC-LDPC chuẩn cho hệ thống thông tin di động 5G được tóm tắt.

Gọi Z là kích thước của ma trận hoán vị tuần hoàn và $P_{i,j}$ là giá trị dịch chuyển. Với bất kỳ giá trị nguyên nào $P_{i,j}$, $0 \le P_{i,j} \le Z$, thì ma trận hoán vị tuần hoàn có kích thước $Z \times Z$ sẽ dịch chuyển ma trận đơn vị I có kích thước $Z \times Z$ sang phải $P_{i,j}$ lần đối với phần tử $(i, j)^{th}$ khác không trong ma trận cơ sở. Ma trận hoán vị tuần hoàn nhị phân này được ký hiệu là $Q(P_{i,j})$. Ví dụ: $Q(1)$ được cho bởi

$$Q(1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{1}$$

Để đơn giản ký hiệu thì $Q(-1)$ biểu thị ma trận rỗng (tất cả các phần tử bằng 0).

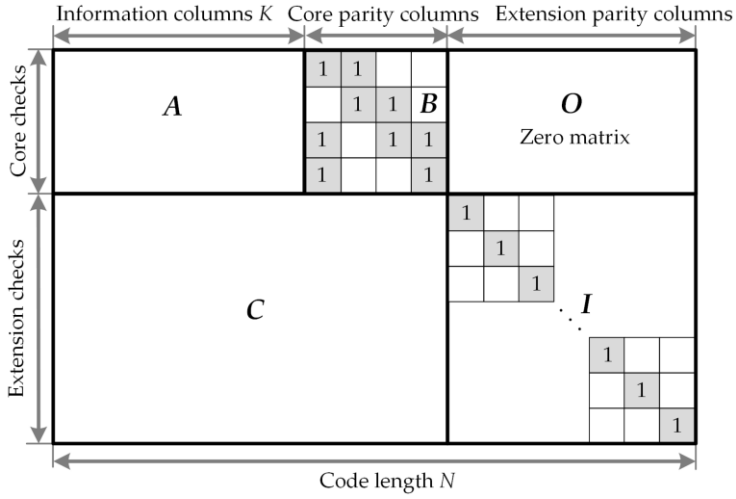
Mã QC-LDPC nhị phân có thể được đặc trưng bởi không gian rỗng của một mảng tuần hoàn thưa có cùng kích thước [9]. Ma trận kiểm tra chẵn lẻ **H** của mã QC-LDPC có thể được xác định bằng ma trận cơ sở và hệ số dịch chuyển $P_{i,j}$. Các phần tử 1 và 0 trong ma trận cơ sở được thay thế bằng một ma trận hoán vị tuần hoàn và một ma trận 0 có kích thước $Z \times Z$ tương ứng. Đối với 2 số nguyên dương m_b và n_b, với $m_b \le n_b$, thì mã QC-LDPC được biểu thị bằng mảng $m_b \times n_b$ của ma trận tuần hoàn có kích thước $Z \times Z$ trên trường GF(2):

$$\mathbf{H} = \begin{bmatrix} Q\left(P_{1,1}\right) & Q\left(P_{1,2}\right) & \cdots & Q\left(P_{1,n_b}\right) \\ Q\left(P_{2,1}\right) & Q\left(P_{2,2}\right) & \cdots & Q\left(P_{2,n_b}\right) \\ \vdots & \vdots & \ddots & \vdots \\ Q\left(P_{m_b,1}\right) & Q\left(P_{m_b,2}\right) & \cdots & Q\left(P_{m_b,n_b}\right) \end{bmatrix} \tag{2}$$

Ma trận lũy thừa của **H**, được ký hiệu là $E(\mathbf{H})$, có dạng sau:

$$E\left(\mathbf{H}\right) = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,n_b} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,n_b} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m_b,1} & P_{m_b,2} & \cdots & P_{m_b,n_b} \end{bmatrix} \tag{3}$$

Mỗi phần tử trong ma trận $E(\mathbf{H})$ được coi là một giá trị dịch chuyển. Cần lưu ý rằng ma trận kiểm tra chẵn lẻ **H** trong phương trình (2) có thể được xây dựng bằng cách khai triển ma trận lũy thừa $E(\mathbf{H})$ có kích thước $m_b \times n_b$. Quy trình này được gọi là xây dựng biểu đồ [10].



Hình 1. Cấu trúc ma trận kiểm tra chẵn lẻ cơ sở cho mã QC-LDPC [12]

Mã QC-LDPC đóng vai trò quan trọng trong truyền thông 5G và đã được chấp nhận là mô hình mã hóa kênh cho kênh dữ liệu 5G trong cuộc họp tiêu chuẩn 3GPP [10]. Hình 1 minh họa cấu trúc chung của ma trận kiểm tra chẵn lẻ cơ sở cho mã QC-LDPC. Các cột được chia thành 3 phần: cột thông tin, cột chẵn lẻ cốt lõi và cột chẵn lẻ mở rộng. Các hàng được chia thành 2 phần: hàng kiểm tra lõi và hàng kiểm tra mở rộng. Như được thể hiện trong Hình 1, ma trận cơ sở bao gồm các ma trận con, cụ thể là *A, B, O, C* và *I*. Ma trận con *A* tương ứng với các bit có tính hệ thống. Ma trận con *B* tương ứng với tập hợp bit chẵn lẻ đầu tiên và là ma trận vuông có cấu trúc đường chéo kép: cột đầu tiên của nó có trọng số là 3, trong khi ma trận con bao gồm các cột khác sau cột đầu tiên có cấu trúc đường chéo kép trên. Ma trận con *O* là một ma trận bằng không. Để hỗ trợ nhanh chóng yêu cầu lặp lại tự động kết hợp dự phòng gia tăng thì phần mở rộng dựa trên kiểm tra chẵn lẻ (PC - Parity Check) duy nhất được sử dụng để hỗ trợ tỷ lệ thấp hơn như thể hiện trong Hình 1. Ma trận con *C* tương ứng với các hàng PC, và *I* là ma trận nhận dạng tương ứng với tập bit chẵn lẻ thứ hai, tức là phần mở rộng PC. Sự kết hợp của *A* và *B* được gọi là hạt nhân, và các phần khác (*O, C* và *I*) được gọi là phần mở rộng.

3GPP đã đồng ý xem xét hai ma trận cơ sở, ký hiệu là BG1 và BG2, cho mã hóa kênh. BG1 nhắm tới mục tiêu cho độ dài khối lớn hơn và tỷ lệ mã hóa *R* cao hơn. BG2 nhắm tới mục tiêu cho độ dài khối nhỏ hơn và tỷ lệ mã hóa thấp hơn. Nếu kích thước khối ≤ 292 hoặc ≤ 3824 và $R \leq 2/3$ hoặc $R \leq 1/4$ thì ma trận cơ sở 2, BG2, của mã LDPC được sử dụng; nếu không thì ma trận cơ sở 1, BG1, của mã LDPC được sử dụng [11].

Với BG1, ma trận **H** của BG1 có kích thước $m_b \times n_b$ $(m_b = 46, n_b = 68, k_b = n_b - m_b = 22)$.

Với BG2, ma trận **H** của BG2 có kích thước $m_b \times n_b$ $(m_b = 42, n_b = 52, k_b = n_b - m_b = 10)$.

Các cột bit thông tin là ma trận có kích thước $k_b \times Z$.

Đối với các ma trận cơ sở BG1 và BG2 thì số lượng thiết kế hệ số dịch chuyển là 8. Tất cả các kích thước khác được chia thành 8 tập dựa trên tham số *a*, trong đó *a* được sử dụng để xác định kích thước nâng $Z = a \times 2^j$. Tập hợp các hệ số dịch chuyển được liệt kê trong Bảng 1.

Bảng 1. Mối quan hệ giữa ma trận lũy thừa và tập kích thước nâng

Ma trận lũy thừa	Tập kích thước nâng
Tập 1	$Z = 2 \times 2^j, j = 0,1,2,3,4,5,6,7$
Tập 2	$Z = 3 \times 2^j, j = 0,1,2,3,4,5,6,7$
Tập 3	$Z = 5 \times 2^j, j = 0,1,2,3,4,5,6$
Tập 4	$Z = 7 \times 2^j, j = 0,1,2,3,4,5$
Tập 5	$Z = 9 \times 2^j, j = 0,1,2,3,4,5$
Tập 6	$Z = 11 \times 2^j, j = 0,1,2,3,4,5$
Tập 7	$Z = 13 \times 2^j, j = 0,1,2,3,4$
Tập 8	$Z = 15 \times 2^j, j = 0,1,2,3,4$

Giá trị dịch chuyển $P_{i,j}$ có thể được tính bằng cách sử dụng hàm $P_{i,j} = f(V_{i,j}, Z)$, trong đó $V_{i,j}$ là hệ số dịch chuyển của phần tử (i, j) trong thiết kế dịch chuyển tương ứng. Hàm f được định nghĩa như sau:

$$P_{i,j} = f\left(V_{i,j}, Z\right) = \begin{cases} -1 & ,V_{i,j}, Z = -1 \\ \mathrm{mod}\left(V_{i,j}, Z\right) & , \text{ khác} \end{cases} \quad (4)$$

trong đó: mod là toán tử modulo.

3. MÃ HÓA VÀ GIẢI MÃ MÃ LDPC

3.1. Mã hóa

Thay vì sử dụng ma trận sinh **G**, mã LDPC có thể được mã hóa trực tiếp bằng ma trận kiểm tra chẵn lẻ **H** bằng cách chuyển nó thành dạng tam giác thấp và áp dụng phép thay thế ngược lại. Phương pháp mã hóa RU, được đề xuất bởi Richardson và Urbanke, là một phương pháp mã hóa có thời gian mã hóa tuyến tính cho các ma trận kiểm tra chẵn lẻ thưa. Nguyên tắc cơ bản là phép biến đổi chỉ sử dụng hoán vị hàng và cột, để định dạng lại ma trận kiểm tra chẵn lẻ **H** thành ma trận thưa. Do đó, phương pháp này có thể giảm độ phức tạp so với phương pháp nhân ma trận sinh **G**. Thuật toán RU bao gồm hai bước: bước tiền xử lý và bước mã hóa thực tế [15].

Cho từ mã $C = [s\ p_a\ p_c]$, trong đó s biểu thị phần hệ thống, được chia thành k_b nhóm gồm Z bits vì mô hình cơ sở có $k_b = n_b - m_b$ cột bit thông tin. Hơn nữa, $s = \left[s_1, s_2, \ldots, s_{k_b} \right]$, trong đó mỗi phần tử của s là một vector có độ dài Z. Các bản tin nhận được bởi bộ mã hóa được lưu trữ trong các thanh ghi được sắp xếp theo khối k_b, ký hiệu là s_i (i = 1, 2, ..., k_b), tương ứng với các khối hệ thống, trong đó mỗi khối bao gồm Z bit. Nếu bộ mã hóa được thiết kế để đọc Z bit trên mỗi chu kỳ xung nhịp thì cần k_b chu kỳ để lưu trữ tất cả các khối thông tin. Hơn nữa, chuỗi chẵn lẻ có thể được nhóm thành các tập Z bit. Giả sử rằng phần chẵn lẻ của mỗi thông tin p được chia thành 2 thành phần như sau: g = 4 bit chẵn lẻ đầu tiên

$p_a = \left[p_{a_1}, p_{a_2}, \ldots p_{a_g} \right]$ và phần còn lại gồm (m_b-g) bit kiểm tra $p_c = \left[p_{c_1}, p_{c_2}, \ldots p_{c_{m_b-g}} \right]$. Cụ thể, từ mã được mã hóa có thể được biểu thị như sau:

$$C = [s_1, s_2, \ldots, s_{k_b}, p_{a_1}, p_{a_2}, \ldots, p_{a_g}, p_{c_1}, p_{c_2}, \ldots, p_{c_{m_b-g}}] \tag{5}$$

Ma trận kiểm tra chẵn lẻ **H** của mã QC-LDPC có thể được chia thành 6 ma trận và được trình bày ở dạng sau:

$$H = \begin{bmatrix} A & B & 0 \\ C_1 & C_2 & I \end{bmatrix} \tag{6}$$

trong đó: A là ma trận có kích thước $g \times k_b$, B là ma trận có kích thước $g \times g$, C_1 là ma trận có kích thước $(m_b\text{-}g) \times k_b$ và C_2 là ma trận có kích thước $(m_b\text{-}g) \times g$. Ngoài ra, I là một ma trận đơn vị có kích thước là $(m_b\text{-}g) \times (m_b\text{-}g)$. Việc mã hóa các mã LDPC được thực hiện bằng cách sử dụng phương trình sau:

$$HC^T = 0^T \tag{7}$$

Phương trình (7) cũng có thể được biểu thị như sau:

$$\begin{bmatrix} A & B & 0 \\ C_1 & C_2 & I \end{bmatrix} \begin{bmatrix} s \\ p_a \\ p_c \end{bmatrix} = 0^T \tag{8}$$

Phương trình (8) sau đó được tách thành 2 phương trình như sau:

$$As^T + Bp_a^T + 0\,p_c^T = 0^T \tag{9}$$

$$C_1 s^T + C_2 p_a^T + I p_c^T = 0^T \tag{10}$$

Thuật toán mã hóa RU được thực hiện theo 2 bước. Trong bước đầu tiên, các bit chẵn lẻ trong phần đầu tiên được tính bằng cách giải phương trình (9). Bước thứ hai trong quá trình mã hóa bao gồm tính toán các phần chẵn lẻ p_c bằng phương trình (10).

Bước đầu tiên trong việc triển khai bộ mã hóa là xác định phần p_a. Trước tiên, phương trình (9) được viết lại ở dạng khối như sau:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,k_b} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,k_b} \\ a_{3,1} & a_{3,2} & \ddots & a_{3,k_b} \\ a_{4,1} & a_{4,2} & \cdots & a_{4,k_b} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{k_b} \end{bmatrix} + \begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 \\ 1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} p_{a_1} \\ p_{a_2} \\ p_{a_3} \\ p_{a_4} \end{bmatrix} = 0 \tag{11}$$

Sau đó, mở rộng phương trình (10) thành tập phương trình sau:

$$\sum_{j=1}^{k_b} a_{1,j} s_j + p_{a_1}^{(1)} + p_{a_2} = 0 \tag{12}$$

$$\sum_{j=1}^{k_b} a_{2,j} s_j + p_{a_1} + p_{a_2} + p_{a_3} = 0 \tag{13}$$

$$\sum_{j=1}^{k_b} a_{3,j} s_j + p_{a_3} + p_{a_4} = 0 \tag{14}$$

$$\sum_{j=1}^{k_b} a_{4,j} s_j + p_{a_1}^{(\alpha)} + p_{a_4} = 0 \tag{15}$$

trong đó $p_{a_1}^{(\alpha)}$ biểu thị phiên bản dịch chuyển theo chu kỳ thứ α (bên phải) của p_{a_1} với $0 \le \alpha \le Z$. Bằng cách cộng tất cả các phương trình trên, ta thu được kết quả sau:

$$p_{a_1} = \sum_{i=1}^{4} \sum_{j=1}^{k_b} a_{i,j} s_j \tag{16}$$

Cần lưu ý rằng việc triển khai đơn giản $a_{i,j} s_j$ có thể được thực hiện với việc sử dụng bộ dịch tuần hoàn Z bit. Vì $a_{i,j} s_j$ là một dịch vòng sang phải của s_j với hệ số dịch chuyển theo $a_{i,j}$ nên độ phức tạp phần cứng là nhỏ. Dựa trên định nghĩa

$$\lambda_i = \sum_{j=1}^{k_b} a_{i,j} s_j \ \ for\, i = 1,2,3,4 \tag{17}$$

ta có thể đạt được

$$p_{a_1} = \sum_{i=1}^{4} \lambda_i \tag{18}$$

$$p_{a_2} = \lambda_1 + p_{a_1}^{(1)} \tag{19}$$

$$p_{a_3} = \lambda_3 + p_{a_4} \tag{20}$$

$$p_{a_4} = \lambda_4 + p_{a_1}^{(1)} \tag{21}$$

Từ phương trình (17), mỗi giá trị λ_i được tính bằng cách cộng dồn tất cả các giá trị $a_{i,j} s_j$. Trong phép toán modulo 2, λ_i có được bằng cách thực hiện các phép toán XOR trên tất cả các phần tử của $a_{i,j} s_j$. Các giá trị λ_i có thể được ước tính trên mỗi chu kỳ xung nhịp trong g = 4 chu kỳ. Khối thứ nhất của các bit chẵn lẻ p_{a_1} được tính bằng cách tích lũy tất cả các giá trị λ_i. Các cặp bit chẵn lẻ còn lại có thể được lấy bằng một phương pháp có thể dễ dàng suy ra từ phương trình (19)-(21). Quá trình này có thể được thực hiện trong 2 chu kỳ xung nhịp vì có sự phụ thuộc giữa p_{a_3} và p_{a_4}. Tất cả các bit chẵn lẻ p_a trong phần chẵn lẻ đầu tiên được lưu trữ trong các thanh ghi dịch.

Trong bước thứ hai, phần p_c có thể được xác định dễ dàng dựa trên phương trình (10), trong đó ma trận C_1 và C_2 được cho bởi

$$C_1 = \begin{bmatrix} c_{1,1} & c_{2,1} & \cdots & c_{1,k_b} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,k_b} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m_b-g,1} & c_{m_b-g,2} & \cdots & c_{m_b-g,k_b} \end{bmatrix} \quad C_2 = \begin{bmatrix} c_{1,k_b+1} & c_{2,k_b+2} & \cdots & c_{1,k_b+g} \\ c_{2,k_b+1} & c_{2,k_b+2} & \cdots & c_{2,k_b+g} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m_b-g,k_b+1} & c_{m_b-g,k_b+2} & \cdots & c_{m_b-g,k_b+g} \end{bmatrix} \tag{22}$$

Khi áp dụng phương trình (10), các phần tử của p_c có thể được tính bằng các phương trình sau:

$$
\begin{aligned}
p_{c_1} &= \sum_{j=1}^{k_b} c_{1,j} s_j + \sum_{j=1}^{g} c_{1,k_b+j} p_{a_j}, \\
p_{c_2} &= \sum_{j=1}^{k_b} c_{2,j} s_j + \sum_{j=1}^{g} c_{2,k_b+j} p_{a_j}, \\
&\vdots \\
p_{c_{m_b-g}} &= \sum_{j=1}^{k_b} c_{m_b-g,j} s_j + \sum_{j=1}^{g} c_{m_b-g,k_b+j} p_{a_j}.
\end{aligned} \tag{23}
$$

Tương tự, $c_{i,j}s_j$ biểu thị sự chuyển dịch vòng của s_j với hệ số dịch chuyển được xác định bởi $c_{i,j}$ và $c_{i,k_b+j}p_{a_j}$ biểu thị sự chuyển dịch vòng của p_{a_j} với hệ số dịch chuyển được xác định bởi c_{i,k_b+j}. Ngay sau khi thu được $c_{i,j}s_j$ và $c_{i,k_b+j}p_{a_j}$, chúng có thể được sử dụng để xác định giá trị của các bit chẵn lẻ tương ứng trong phần chẵn lẻ thứ hai p_c. Bước này có thể được thực hiện trong một chu kỳ xung nhịp duy nhất. Do đó, tất cả các bit chẵn lẻ p_c có thể được thu thập trong chu kỳ xung nhịp (m_b-g). Sau đó, từ mã là sự kết hợp của thông điệp ban đầu s và hai phần chẵn lẻ được tính toán p_a và p_c.

3.2. Giải mã

Sum-product là tên chung cho một lớp thuật toán giải mã Maximum Likelihood (ML) [13]. Thuật toán sử dụng thông tin kênh truyền và các giá trị từ kênh truyền. Thuật toán tạo ra một giá trị xác suất cho mỗi bit nhận được và làm mới giá trị này sau nhiều lần lặp để tìm ước lượng cho bit đó.

Mã LDPC (N, K) là mã nhị phân được đặc trưng bởi ma trận kiểm tra chẵn lẻ thưa $\mathbf{H}_{M\times N}$ trong đó $M = N - K$ có thể được biểu diễn bằng đồ hình Tanner của các nút biến $n \in \{1,\ldots,N\}$ và các nút kiểm tra $m \in \{1,\ldots,M\}$. Biểu thị tập hợp các nút biến được kết nối với một nút kiểm tra m nào đó là $\mathrm{N}\{m\}$. Một nút biến n được kết nối với nút kiểm tra m nếu $n \in \mathrm{N}\{m\}$. Ngoài ra, tập $\mathrm{N}\{m\}\setminus n$ biểu thị tập các nút biến được kết nối với nút kiểm tra m không bao gồm n. Tương tự, tập các nút kiểm tra nối với một nút biến nào đó n được ký hiệu là $\mathrm{M}\{n\}$. Một nút kiểm tra được kết nối với nút biến n nào đó nếu $m \in \mathrm{M}\{n\}$. Tập hợp $\mathrm{M}\{n\}\setminus m$ biểu thị tập hợp các nút kiểm tra được kết nối với nút biến n loại trừ m.

Thuật toán sum-product xử lý lặp đi lặp lại các bit nhận được theo các bước nối liền nhau có thể được nhìn thấy trên đồ hình Tanner dưới dạng bước ngang tiếp theo là bước dọc để cải thiện độ tin cậy của mỗi bit được giải mã. Các thước đo độ tin cậy được tính toán của các bit ở cuối bất kỳ lần lặp giải mã nào được sử dụng làm đầu vào của lần lặp tiếp theo. Thuật toán giải mã lặp này tiếp tục cho đến khi thỏa mãn một tiêu chí dừng nào đó.

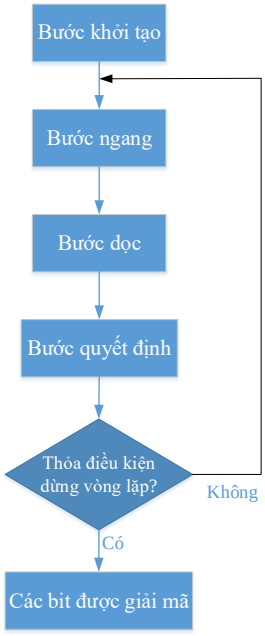
Để minh họa, hãy xét độ tin cậy của một bit đã giải mã được đo bằng xác suất posteriori $P(x_n|Y)$, $1 \le n \le N$. Sau đó, Log-Likelihood Ratio (LLR) của mỗi bit mã được tính bởi

$$
L(x_n) = \log \frac{P(x_n = 0|Y)}{P(x_n = 1|Y)} \tag{24}
$$

Trong mỗi lần lặp lại, một giá trị $r_{m\to n}$ được tính theo bước ngang tại mỗi nút kiểm tra m và được chuyển cho tất cả các nút biến n nếu $n \in \mathrm{N}\{m\}$. Tương tự, mỗi nút biến n sẽ gửi

một giá trị $q_{n\to m}$ trong bước dọc đến tất cả các nút kiểm tra *m* nếu $m\in \mathrm{M}\{n\}$.

Từ mã được ký hiệu là $X=[x_1,x_2,\ldots,x_N]$, trong đó $x_n\in\{0,1\}$. Các giá trị LLR của vector nhận được tương ứng được biểu thị bằng $Y=[y_1,y_2,\ldots,y_N]$.



Hình 2. Giải thuật giải mã sum-product

Quá trình giải mã sử dụng thuật toán sum-product có thể được thực hiện theo các bước liên tiếp như Hình 2.

Bước khởi tạo: Các giá trị ban đầu của LLR có thể nhận được từ đầu ra của bộ giải điều chế y_n. Các giá trị ban đầu này được sử dụng làm $q_{n\to m}$ của lần lặp đầu tiên cho bước cập nhật nút kiểm tra (Bước ngang).

Bước ngang: Bước ngang tại nút kiểm tra *m* được dành riêng để xử lý các giá trị đến từ các nút biến $q_{n\to m}$ để tính toán các giá trị trả lời $r_{m\to n}$ cho mọi $n\in \mathrm{N}\{m\}$. Vì vậy, đối với mỗi nút kiểm tra *m*:

$$r_{m\to n}=\left(\prod_{n'\in \mathrm{N}(m)\backslash n}\operatorname{sgn}(q_{n'\to m})\right)\times 2\tanh^{-1}\left(\prod_{n'\in \mathrm{N}(m)\backslash n}\tanh\left(\frac{|q_{n'\to m}|}{2}\right)\right) \tag{25}$$

Bước dọc: Bước dọc tại nút biến *n* được dành riêng để xử lý các giá trị đến từ các nút kiểm tra $r_{m\to n}$ để tính toán các giá trị trả lời $q_{n\to m}$ cho mọi $m\in \mathrm{M}\{n\}$. Vì vậy, đối với mỗi nút biến *n*:

$$q_{n\to m}=y_n+\sum_{m'\in M(n)\backslash m} r_{m'\to n}(x_n) \tag{26}$$

Bước quyết định: Đối với mỗi nút biến, các giá trị LLR được cập nhật theo

$$L(x_n)=y_n+\sum_{m\in \mathrm{M}(n)} r_{m\to n}(x_n) \tag{27}$$

Các giá trị LLR được áp dụng cho quyết định cứng để quyết định về giá trị có thể có của x_n là 1 nếu $L(x_n) < 0$ và 0 nếu ngược lại. Syndrome $H\hat{x}^T$ sau đó được tính toán và kiểm tra để quyết định giải mã thành công nếu Syndrome bằng 0 hoặc tiến hành lặp lại tiếp theo nếu điều kiện Syndrome không được thỏa mãn. Quá trình này tiếp tục cho đến khi từ mã được giải mã thành công hoặc số lần lặp tối đa đã hết.

3.3. Kỹ thuật phân lớp đề xuất

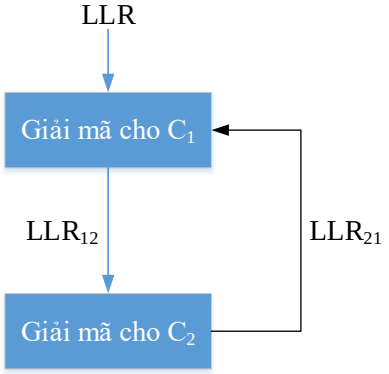
Để cải thiện hiệu năng giải mã của thuật toán sum-product, nhóm tác giả đề xuất kỹ thuật phân lớp. Kỹ thuật này tăng tính bảo mật cho quyết định của bit x_n. Ưu điểm của kỹ thuật đề xuất là hệ số hiệu chỉnh làm giảm tổn thất hiệu năng và độ phức tạp của việc giải mã. Trong kỹ thuật này, chúng ta xem lớp đầu tiên là một tập hợp các nút biến có giá trị thấp của thông tin nội tại y_n của bit x_n.

Đối với mỗi lần lặp, chúng ta tính toán nút kiểm tra và nút biến trong một lớp. Việc giải mã sau đó diễn ra tuần tự. Điều này có nghĩa là ta sẽ tập hợp một số hàng của ma trận kiểm tra chẵn lẽ thành một lớp và thực hiện bước dọc trong giải thuật giải mã sum-product. Ma trận **H** sẽ được phân lớp thành như sau:

$$\mathbf{H} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{N-K} \end{bmatrix} \tag{28}$$

trong đó mỗi khối hàng trong ma trận **H** là một lớp.

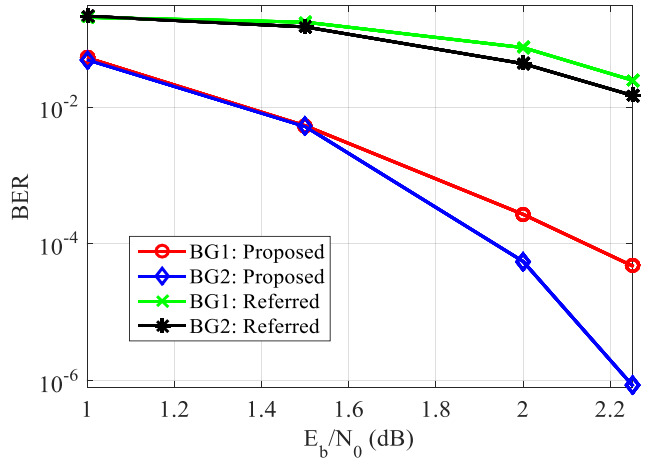
Hình 3 minh họa sơ đồ giải mã của kỹ thuật phân lớp đề xuất khi có 2 lớp trong đó: C_1 là từ mã được mã hóa từ H_1 và C_2 là từ mã được mã hóa từ H_2. Việc giải mã C_1 sẽ sử dụng LLR trong vòng lặp đầu tiên, sau đó sẽ sử dụng LLR_{21} sau khi đã cập nhật cột.



Hình 3. Thuật toán giải mã đề xuất với kỹ thuật phân lớp

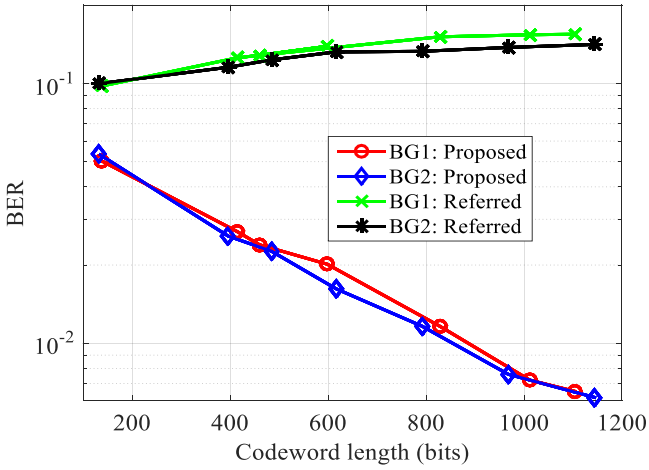
4. KẾT QUẢ MINH HỌA

Phần này trình bày các kết quả mô phỏng cho hệ thống truyền thông được điều chế BPSK (Binary Phase Shift Keying) với nhiễu Gaussian trắng cộng (Additive White Gaussian Noise - AWGN). Mã LDPC sử dụng 2 loại ma trận kiểm tra chẵn lẻ, BG1 và BG2, cho hệ thống thông tin di động 5G. Giải thuật mã hóa và giải mã đã được trình bày trong phần 3.



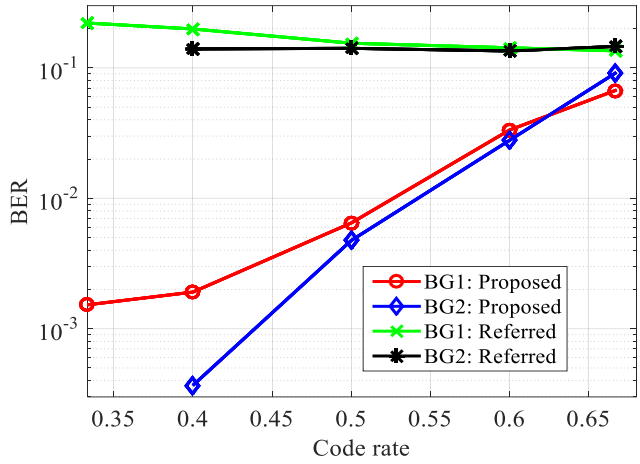
Hình 4. BER theo E_b/N_0

Hình 4 trình bày BER theo tỷ số năng lượng bit trên nhiễu E_b/N_0 với độ dài từ mã là 1024 bit, tỷ lệ mã hóa là 1/2, giải mã với 20 vòng lặp. Hình này cho thấy BER giảm đáng kể khi E_b/N_0 tăng. Đặc biệt, thuật toán giải mã với kỹ thuật phân lớp đề xuất (được ký hiệu trên hình là "Proposed") đã làm cho BER giảm nhanh hơn nhiều so với thuật toán giải mã truyền thống (được ký hiệu trên hình là "Referred") khi E_b/N_0 tăng. Ngoài ra, kỹ thuật phân lớp đề xuất đã cải thiện độ tin cậy đáng kể so với khi không sử dụng kỹ thuật này.



Hình 5. BER theo độ dài từ mã

Hình 5 trình bày BER theo độ dài từ mã với E_b/N_0 = 1,5 dB, tỷ lệ mã hóa là 1/2, giải mã với 20 vòng lặp. Hình này cho thấy BER giảm đáng kể khi độ dài từ mã tăng đối với giải thuật giải mã với kỹ thuật phân lớp đề xuất trong khi đó BER tăng không đáng kể khi độ dài từ mã tăng đối với giải thuật giải mã truyền thống. Điều này cho thấy hiệu quả của kỹ thuật đề xuất trong việc cải thiện độ tin cậy truyền tin. Hơn nữa, kỹ thuật phân lớp đề xuất đã làm giảm đáng kể BER so với khi không sử dụng kỹ thuật này.



Hình 6. BER theo tỷ lệ mã hóa

Hình 6 trình bày BER theo tỷ lệ mã hóa với E_b/N_0 = 1,5 dB và giải mã với 20 vòng lặp. Hình này cho thấy BER giảm đáng kể khi tỷ lệ mã hóa tăng đối với giải thuật giải mã với kỹ thuật phân lớp đề xuất trong khi đó BER giảm không đáng kể khi tỷ lệ mã hóa tăng đối với giải thuật giải mã truyền thống. Tuy nhiên, đối với mọi giá trị của tỷ lệ mã hóa thì kỹ thuật phân lớp đề xuất đều đạt BER nhỏ hơn đáng kể so với khi không sử dụng kỹ thuật này. Điều này cho thấy hiệu quả của kỹ thuật đề xuất trong việc cải thiện độ tin cậy truyền tin.

Sử dụng công cụ đo thời gian thực của Matlab, chúng tôi đã khảo sát thời gian giải mã của thuật toán giải mã với E_b/N_0 = 1 dB, giải mã với 20 vòng lặp, ma trận kiểm tra chẵn lẻ BG2 với Z = 52, 100 khối bit truyền. Kết quả đo được khi không sử dụng và sử dụng kỹ thuật phân lớp được trình bày lần lượt trên các Hình 7 và 8. Các hình này cho thấy thời gian giải mã giảm đáng kể từ khoảng 45,383 giây xuống còn 27,811 giây khi sử dụng kỹ thuật phân lớp.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
BPSK_nrldpc_sim	1	45.383 s	41.258 s	

Hình 7. Thời gian giải mã khi không sử dụng kỹ thuật phân lớp. "Function Name" = "Tên hàm" dùng để tính thời gian thực hiện. Trong hình này thì tên hàm là BPSK_nrldpc_sim. "Calls" là số hàm cần thực hiện để đo thời gian. "Total Time" là tổng thời gian thực hiện.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
BPSK_nrldpc	1	27.811 s	0.160 s	

Hình 8. Thời gian giải mã khi sử dụng kỹ thuật phân lớp. "Function Name" = "Tên hàm" dùng để tính thời gian thực hiện. Tên hàm là BPSK_nrldpc. "Calls" là số hàm cần thực hiện để đo thời gian. "Total Time" là tổng thời gian thực hiện.

Tổng hợp các kết quả ở trên cho thấy, kỹ thuật phân lớp đã giảm đáng kể cả về tỷ lệ lỗi bit và thời gian giải mã. Điều này cho thấy ưu điểm của kỹ thuật phân lớp đề xuất.

5. KẾT LUẬN

Bài báo này đã đề xuất kỹ thuật phân lớp để giảm thời gian giải mã và tỷ lệ lỗi bit cho mã LDPC ứng dụng trong hệ thống thông tin di động 5G. Nhiều kết quả đã chứng minh các ưu điểm của kỹ thuật đề xuất bằng cách so sánh thuật toán giải mã sum-product khi sử dụng và khi không sử dụng kỹ thuật này theo nhiều thông số khác nhau như tỷ lệ năng lượng bit trên nhiễu, độ dài từ mã, tỷ lệ mã hóa.

Bài báo này đã hiện thực thuật toán mã hóa và giải mã cho mã LDPC ứng dụng trong hệ thống thông tin di động 5G sử dụng phần mềm Matlab. Hướng phát triển tiếp theo là đánh giá hiệu năng của mã LDPC trong điều kiện vận hành thực tế của hệ thống thông tin di động 5G để đánh giá toàn diện khả năng triển khai của kỹ thuật phân lớp đề xuất.

TÀI LIỆU THAM KHẢO

1. Gallager R. - Low-density parity-check codes, IRE Transactions on Information Theory **8** (1) (1962) 21-28.
2. MacKay D. J. C. and Neal R. M. - Near Shannon limit performance of low density parity check codes, Electronics Letters **33** (6) (1997) 457-458.
3. Yuhai S., Chunjiang L. and Ming Y. - The application of LDPC code in ABS-S system, in Proc. International Forum on Information Technology and Applications, Chengdu, China (2009) 159-162.
4. Zhu K. and Wu Z. - Comprehensive study on CC-LDPC, BC-LDPC and polar code, in Proc. IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Seoul, South Korea (2020) 1-6.
5. Sun Y., Karkooti M. and Cavallaro J. R. - High throughput, parallel, scalable LDPC encoder/decoder architecture for OFDM systems, in Proc. IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software, Richardson, TX, USA (2006) 39-42.
6. de Fez I., Fraile F., Belda R. and Guerri J. C. - Analysis and evaluation of adaptive LDPC AL-FEC codes for content download services, IEEE Transactions on Multimedia **14** (3) (2012) 641-650.
7. Wang Y., Ueng Y., Peng C. and Yang C. - A low-complexity LDPC decoder architecture for WiMAX applications, in Proc. International Symposium on VLSI Design, Automation and Test, Hsinchu, Taiwan (2011) 1-4.
8. Tsatsaragkos I. and Paliouras V. - A reconfigurable LDPC decoder optimized for 802.11n/ac applications, IEEE Transactions on Very Large Scale Integration (VLSI) Systems **26** (1) (2018) 182-195.
9. Li H., Bai B., Mu X., Zhang J. and Xu H. - Algebra-assisted construction of quasi-cyclic LDPC codes for 5G new radio, IEEE Access **6** (2018) 50229-50244.
10. https://www.cohere-technologies.com/wp-content/uploads/2017/06/R1-1700859.pdf
11. Yasotharan H. and Carusone A. C. - A flexible hardware encoder for systematic low-density parity-check codes, in Proc. IEEE International Midwest Symposium on Circuits and Systems, Cancun, Mexico (2009) 54-57.
12. N. T. B. Tram, N. T. Tuy, and L. Hanho - Efficient QC-LDPC encoder for 5G new radio, Electronics **8** (6) (2019) 1-15.

13. Emran A. A. and Elsabrouty M. - Simplified variable-scaled min sum LDPC decoder for irregular LDPC codes, in Proc. IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA (2014) 518-523.
14. Liang T., Zhang P., Liu C. and Liu J. - Efficient encoding of quasi-cyclic low-density parity-check codes, in Proc. IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China (2018) 1189-1193.
15. Roberts M. K., Mohanram S. S., and Shanmugasundaram N. - An improved low complex offset min-sum based decoding algorithm for LDPC codes, Mobile Networks and Applications **24** (6) (2019) 1848-1852.

ABSTRACT

STRATIFYING TECHNIQUE FOR DECODING EFFICIENTLY LDPC CODES IN 5G MOBILE COMMUNICATION SYSTEM

Nguyen Trong Duy, Ho Van Khuong*
Ho Chi Minh City University of Technology, VNU-HCM
*Email: *hvkhuong@hcmut.edu.vn*

The 5^{th} Generation (5G) mobile communication system must meet three main criteria: wide bandwidth, high reliability and low latency. Low density parity check (LDPC) code was adopted for the 5G mobile communication system because the LDPC code reaches closely to the Shannon capacity. This paper proposes a stratifying technique to significantly reduce the decoding time and improve the bit error rate (BER). The performance of the proposed technique is evaluated in different system parameters such as energy-per-bit to noise power ratio, codeword length and code rate.

Keywords: LDPC code, 5G, BER, sum-product, stratified decoding.