IEEE 802.11 における LDPC 符号とその符号化・復号方法

LDPC codes and its encoding/decoding algorithm on IEEE 802.11

けーさん（@k3k0ma）

1 はじめに

LDPC（Low-Density Parity-Check）符号は難しい．正直よくわからん．自分で勉強し実装するまでは「LDPC ってあれでしょ？Belief Propagation でデコードするやつ」みたいな感想しか持っていなかった．実際に実装してみると，そもそもパリティ検査行列はどのように作ればいいのか，パリティ検査行列を手に入れたとしても符号語はどうやって生成すればいいのか，とデコーダの前のエンコーダの時点でつまづいてしまった．

本稿では私が LDPC 符号について勉強したこと，符号化アルゴリズムと，Sum-Product 復号アルゴリズムについてまとめる．特に符号化アルゴリズムは IEEE 802.11 における LDPC 符号に特化したアルゴリズムについてまとめている．また，Sum-Product アルゴリズムの説明では「ファクターグラフ」や「メッセージ」という聞き慣れない・使い慣れない語彙を用いずに，確率の基礎的内容のみを用いた説明を展開してみた．以下，断りなしに二元体 GF(2) 上での和や積を用いるが，これらはそれぞれビットの XOR や AND に相当する．

2 線形符号

2 元体上での (n, k) 線形符号は，k 次元（ビット）のメッセージ $\mathbf{u}$ を n 次元空間上の k 次元部分空間へ射影する符号である．この射影の際に利用される基底を並べた行列が生成行列 $\mathbf{G}$（サイズ $k \times n$）である．生成行列を用いて k 次元（ビット）のメッセージ $\mathbf{u}$（サイズ $1 \times k$）を符号化し符号語 $\mathbf{c}$（サイズ $1 \times n$）を得るには，

$$\mathbf{c} = \mathbf{uG} \tag{1}$$

というように行列 $\mathbf{G}$ の左側から $\mathbf{u}$ を掛ければよい．

受信語 $\mathbf{r}$ に誤りが存在するかどうかを確認するためにはサイズ $(n-k) \times n$ のパリティ検査行列 $\mathbf{H}$ を用いて以下の式で $n-k$ 次元のシンドローム $\mathbf{s}$ を計算する．

$$\mathbf{s} = \mathbf{rH}^T \tag{2}$$

もし受信語 $\mathbf{r}$ に誤りがあれば $\mathbf{s} \neq \mathbf{0}$ となる．これは，パリティ検査行列 $\mathbf{H}$ が任意のメッセージ $\mathbf{u}$ に対して $\mathbf{uGH}^T = \mathbf{0}$ を満たすような行列だからである．

では生成行列からパリティ検査行列はどのように得られるだろうか．生成行列 $\mathbf{G}$ に行基本変形を行い，以下のような形の新しい生成行列 $\mathbf{G}$ に変換できるとする．

$$\mathbf{G}' = [\mathbf{I}_k \ \mathbf{P}] \tag{3}$$

ここで $\mathbf{I}_k$ はサイズ $k \times k$ の単位行列であり，$\mathbf{P}$ はサイズ $k \times (n-k)$ の行列である．この生成行列 $\mathbf{G}'$ にメッセージ $\mathbf{u}$ を作用させると，符号語は $\mathbf{c} = [\mathbf{u} \ \mathbf{uP}]$ となる．後半の $\mathbf{uP}$ は次元 $n-k$ のベクトルであり，生成行列により付与された余剰次元の成分である．このような生成行列 $\mathbf{G}'$ に対するパリティ検査行列は以下のように記述される．

$$\mathbf{H}' = [\mathbf{P}^T \ \mathbf{I}_{n-k}] \tag{4}$$

つまり，生成行列 $\mathbf{G}$ に対して行基本変形を行い $\mathbf{G}'$ と $\mathbf{P}$ を得たのちに，式 (4) から検査行列を得ることができる．逆に，検査行列 $\mathbf{H}$ についても同様に行基本変形を行い $\mathbf{H}'$ と $\mathbf{P}$ が得られれば，式 (3) から生成行列が得られる．

3 LDPC 符号と符号化方法

LDPC 符号はおばけみたいな線形符号である．つまり，生成行列 $\mathbf{G}$ とパリティ検査行列 $\mathbf{H}$ がある．LDPC の “Low-Density: LD” 低密度というのは，パリティ検査行列 $\mathbf{H}$ が疎行列であることに由来する．中でも，パリティ検査行列 $\mathbf{H}$ の各行に存在する “1” の数が全行で同じであり，かつそれが列についても成り立つとき，そのような符号を正則（regular）LDPC 符号と呼ぶ．一方で行重みや列重みが一定でない符号を非正則（irregular）LDPC 符号と呼ぶ．このとき，各行に存在する “1” の数を行重み w_r，各列に存在する “1” の数を列重み w_c と呼ぶ．

LDPC 符号の中でも，パリティ検査行列 $\mathbf{H}$ の構成方法によっていくつか種類がある．ここでは，Gallagher 符号と，IEEE 802.11 で採用されている擬巡回（Quasi-Cyclic）LDPC 符号及び BLDPC 符号について紹介する．

3.1 Gallagher 符号

Gallagher が博士論文において利用したパリティ検査行列は，$m = n - k = 9$ かつ $n = 12$ である $(12, 3)$ 符号の一例を示すと行重み $w_r = 4$ 及び列重み $w_c = 3$ に対して，

$$\mathbf{H} = \left[\begin{array}{cccccccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right] \tag{5}$$

というように構成される．この行列の最初の w_c 行は “1” を w_r 個規則的に並べて生成する．それ以降 w_c 行ごとに，最初の w_c 行の列をランダムに入れ替えた値を利用する．生成行列は式 (3) と式 (4) の関係式から，パリティ検査行列を変形し生成行列を得る．

3.2 QC-LDPC 符号 [1]

パリティ検査行列 $\mathbf{H}$ をサイズ $L \times L$ の正方行列 $\mathbf{H}_{i,j}$ を並べた行列として定義する．例として $(4L, L)$ 符号では正方行列を行方向に 4 個，列方向に $4-1=3$ 個並べた行列である．

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \mathbf{H}_{1,2} & \mathbf{H}_{1,3} & \mathbf{H}_{1,4} \\ \mathbf{H}_{2,1} & \mathbf{H}_{2,2} & \mathbf{H}_{2,3} & \mathbf{H}_{2,4} \\ \mathbf{H}_{3,1} & \mathbf{H}_{3,2} & \mathbf{H}_{3,3} & \mathbf{H}_{3,4} \end{bmatrix} \tag{6}$$

特に $\mathbf{H}_{i,j}$ が零行列か巡回置換行列 $\mathbf{R}^n$ であるとき，この符号を QC-LDPC 符号と呼ぶ．ここで，$\mathbf{R}^n$ は

$$\mathbf{R}^n = \underbrace{\mathbf{R}\mathbf{R}\cdots\mathbf{R}}_{n}\mathbf{I}_L \tag{7}$$

であり，$\mathbf{R}$ は式 (8) のように単位行列を 1 行シフトした行列で表される．

$$\mathbf{R} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{8}$$

また $\mathbf{R}^{mL+n} = \mathbf{R}^n$ が成立する．つまり，$L=3$ においては

$$\mathbf{R}^0 = \mathbf{I} \tag{9}$$

$$\mathbf{R}^1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{10}$$

$$\mathbf{R}^2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{11}$$

$$\mathbf{R}^{3m+n} = \mathbf{R}^n \tag{12}$$

が成立する．QC-LDPC 符号のパリティ検査行列は基本行列（Base Matrix）で記述される．(nL, kL) 符号の QC-LDPC 符号に対して基本行列はサイズ $(n-k) \times n$ である．基本行列の要素を $a_{i,j}$ とすると，QC-LDPC 符号のパリティ検査行列の (i,j) ブロック $\mathbf{H}_{i,j}$ は以下のようになる．

$$\mathbf{H}_{i,j} = \begin{cases} \mathbf{R}^{a_{i,j}} & a_{i,j} \geq 0 \\ \mathbf{0} & a_{i,j} < 0 \end{cases} \tag{13}$$

たとえば基本行列が

$$\begin{bmatrix} -1 & 1 & 2 \\ 1 & 0 & -1 \\ 2 & 1 & -1 \end{bmatrix} \tag{14}$$

であり，$L=3$ であれば，そのパリティ検査行列は

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{R} & \mathbf{R}^2 \\ \mathbf{R} & \mathbf{I} & \mathbf{0} \\ \mathbf{R}^2 & \mathbf{R} & \mathbf{0} \end{bmatrix} \tag{15}$$

である．

3.3 BLDPC 符号 [1]

BLDPC（Block-type LDPC）符号は非正則な QC-LDPC 符号の一種である．(nL, kL) 符号な BLDPC 符号では，$m = n-k$ として，パリティ検査行列 $\mathbf{H}$ をサイズ $mL \times kL$ の情報部分 $\mathbf{H}_{\mathrm{I}}$ とサイズ $mL \times mL$ のパリティ部分 $\mathbf{H}_{\mathrm{P}}$ に分割して，$\mathbf{H} = [\mathbf{H}_{\mathrm{I}}\ \mathbf{H}_{\mathrm{P}}]$ と考える．BLDPC ではパリティ部分に次のような構造を持たせた QC-LDPC 符号の一種である．

$$\mathbf{H}_{\mathrm{P}} = \begin{bmatrix} \mathbf{R}^{b_1} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{b_2} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{R}^{b_3} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{R}^{y} & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R}^{b_m-1} & \mathbf{I} \\ \mathbf{R}^{x} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{R}^{b_m} \end{bmatrix} \tag{16}$$

ここで，$\mathbf{R}^y$ が l 番目の行ブロックに存在するとすれば以下が成立するように x, y を定める．

$$x \equiv \sum_{i=1}^{m} b_i \mod L \tag{17}$$

$$y \equiv \sum_{i=l+1}^{m} b_i \mod L \tag{18}$$

BLDPC ではパリティ部分にこのような構造を持たせることで，低計算量な符号化アルゴリズムが存在する [1].

3.4 IEEE 802.11 における BLDPC

IEEE 802.11 では BLDPC の符号化アルゴリズムをより単純にするために，b_i を次のように設定している．

$$b_i = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1 \end{cases} \tag{19}$$

また，このとき $x=1$ であり，P^y が出現する行ブロックの番号は $l \geq 2$ であるため $y=0$ が成立する．つまり，

パリティ部分 $\mathbf{H}_{\mathrm{P}}$ は以下のように単純化される．

$$\mathbf{H}_{\mathrm{P}} = \begin{bmatrix} \mathbf{R} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{I} \\ \mathbf{R} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{20}$$

次に符号化について考える．符号化では，生成行列が式 (3) のような形になっていると仮定するが，$\mathbf{P}$ がどのような行列かの具体的な値については考えない．重要なことは，$\mathbf{P}$ がどのような行列であれ，符号語 $\mathbf{c}$ について $\mathbf{H}\mathbf{c}^T = \mathbf{0}$ が成立しなければいけないことである．まず，生成行列が式 (3) のように記述できると仮定するため，符号語 $\mathbf{c}$ は kL 次元のメッセージ $\mathbf{u}$ と mL 次元の余剰情報部分 $\mathbf{p}$ に分割できる．つまり，$\mathbf{c} = [\mathbf{u}\ \mathbf{p}]$ が成立するため，$\mathbf{p}$ さえ求まれば符号語 $\mathbf{c}$ が求まる．さらにいえば，あるパリティ検査行列を持つ符号の符号化は，$\mathbf{H}\mathbf{c}^T = \mathbf{0}$ を満たすような余剰情報 $\mathbf{p}$ を探す問題と等価である．この問題を解くために，パリティ検査行列を 6 つの行列に分割する．

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ \mathbf{C} & \mathbf{D} & \mathbf{E} \end{bmatrix} \tag{21}$$

ここで各行列のサイズは，$\mathbf{A}$ はサイズ $(m-1)L \times kL$，$\mathbf{B}$ はサイズ $(m-1)L \times L$，$\mathbf{T}$ はサイズ $(m-1)L \times (m-1)L$，$\mathbf{C}$ はサイズ $L \times kL$，$\mathbf{D}$ はサイズ $L \times L$，$\mathbf{E}$ はサイズ $L \times (m-1)L$ である．つまり，情報部分 $\mathbf{H}_{\mathrm{I}}$ は $\mathbf{A}$ と $\mathbf{C}$ に分割され，パリティ部分 $\mathbf{H}_{\mathrm{P}}$ は $\mathbf{B}$，$\mathbf{T}$，$\mathbf{D}$，$\mathbf{E}$ の四つの行列に分割されている．特に，

$$\mathbf{B} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{22}$$

$$\mathbf{D} = \mathbf{R} \tag{23}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{I} \end{bmatrix} \tag{24}$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{25}$$

である．また，符号語の余剰情報 $\mathbf{p}$ も L 次元の $\mathbf{p}_1$ と $(m-1)L$ 次元の $\mathbf{p_2}$ に分割し，$\mathbf{c} = [\mathbf{u}\ \mathbf{p}_1\ \mathbf{p}_2]$ とする．このとき

$$\mathbf{H}\mathbf{c}^T = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ \mathbf{C} & \mathbf{D} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{u}^T \\ \mathbf{p}_1^T \\ \mathbf{p}_2^T \end{bmatrix} = \mathbf{0} \tag{26}$$

が成立するように $\mathbf{p}_1$ と $\mathbf{p}_2$ を定めれば，符号語 $\mathbf{c}$ が得られる．つまり，以下の連立方程式を解けばよい．

$$\mathbf{A}\mathbf{u}^T + \mathbf{B}\mathbf{p}_1^T + \mathbf{T}\mathbf{p}_2^T = \mathbf{0} \tag{27}$$

$$\mathbf{C}\mathbf{u}^T + \mathbf{D}\mathbf{p}_1^T + \mathbf{E}\mathbf{p}_2^T = \mathbf{0} \tag{28}$$

BLDPC の特徴的なパリティ検査行列の設計によって，この連立方程式を満たす $\mathbf{p}_1$ と $\mathbf{p}_2$ は以下の手順で計算できる．ただし，あるベクトル $\mathbf{v}$ に対して $(\mathbf{v})_i$ はその i 番目の要素を表す．

1. $\mathbf{q}_1 = \mathbf{H}_{\mathrm{I}}\mathbf{u}^T$ を計算する
2. $$(\mathbf{p}_1^T)_i = \sum_{j=0}^{m-1} (\mathbf{q}_1)_{jL+i}$$
3. $1 \le i \le (m-1)L$ について以下を計算する
$$(\mathbf{q}_2)_i = (\mathbf{q}_1)_i + (\mathbf{B}\mathbf{u}^T)_i$$
4. $0 \le j \le m-1$ について以下を計算する
$$\begin{aligned}(\mathbf{p}_2^T)_{jL+i} &= \sum_{k=0}^{j} (\mathbf{q}_2)_{kL+i} \\ &= \begin{cases} (\mathbf{q}_2)_i & j = 0 \\ (\mathbf{q}_2)_{jL+i} + (\mathbf{p}_2^T)_{(j-1)L+i} & j \ge 1 \end{cases}\end{aligned}$$

4 Sum-Product 復号

受信信号から復号するために必要な情報は，k 番目のビット c_k が「“0”っぽいのか “1”っぽいのか」という情報である．ではこの「“0”っぽいのか “1”っぽい」というのはどう定量化すればよいだろうか．LDPC 符号の Sum-Product 復号法では,「すべてのパロティ方程式が満たされていることがわかっていれば $c_k = 0$ や $c_k = 1$ である確率はどれくらいか」として，つまり以下のように定義され，ベイズの定理により変換される．ここで,「すべてのパロティ方程式が満たされている」というイベントを S として定義しており，符号化時にパリティ方程式はすべて満たされているため $P(S) = 1$ である．

$$\begin{aligned}&(c_k = 0\text{っぽさ}) = P(c_k = 0|S) \\ &= (\text{事前情報から } c_k = 0 \text{ だと思われる確率}) \\ &\times \begin{pmatrix} c_k = 0 \text{ で } c_k \text{ に関連する} \\ \text{パリティ方程式がすべて満たされる確率} \end{pmatrix} \\ &= P(c_k = 0)P(S|c_k = 0)\end{aligned} \tag{29}$$

$$
\begin{aligned}
&(c_k = 1\text{っぽさ}) = P(c_k = 1|S) \\
&= (\text{事前情報から } c_k = 1 \text{ だと思われる確率}) \\
&\times \begin{pmatrix} c_k = 1 \text{ で } c_k \text{ に関連する} \\ \text{パリティ方程式がすべて満たされる確率} \end{pmatrix} \\
&= P(c_k = 1)P(S|c_k = 1)
\end{aligned} \tag{30}
$$

ここでのパリティ方程式とは $\mathbf{Hc}^T = \mathbf{0}$ の連立方程式のことであり，しかし，パリティ方程式は複数のビットから計算される方程式であるため，$P(c_k|S)$ を計算するためには，c_k 以外の関連するビットの $P(c_k|S)$ が必要である．つまり，$P(c_k|S)$ を計算するためには，関連するビットの $P(c_k|S)$ が必要で，さらにそれを計算するためにはそれらに関連するビットの $P(c_k|S)$ が必要で，さらに... さらに... と堂々巡りを続けてしまう．結局はすべてのビットの $P(c_k|S)$ を知っておかないと，あるビットの $P(c_k|S)$ は計算できない．

Sum-Product 復号法は，この問題を繰り返し計算によって収束させ解を得る手法である．つまり，$P(c_k|S)$ のある初期値からスタートして，それらを更新していくことで真の $P(c_k|S)$ を得る．実際に $P(c_k|S)$ を求めることは式 (29) の $P(c_k = 0|S)$ と式 (30) の $P(c_k = 1|S)$ を求めることになる．実際のアルゴリズムでは，この二つの値を求める代わりに次のように対数尤度比という値を導入する．

$$
\begin{aligned}
\mathrm{LLR}(c_k|S) &= \log \frac{P(c_k = 0|S)}{P(c_k = 1|S)} \\
&= \log \frac{P(c_k = 0)P(S|c_k = 0)}{P(c_k = 1)P(S|c_k = 1)}
\end{aligned} \tag{31}
$$

もし，$\mathrm{LLR}(c_k|S) > 0$ であれば $c_k = 0$ である方が $c_k = 1$ であることよりも尤もらしい．つまり，$\mathrm{LLR}(c_k|S) > 0$ なら $c_k = 0$ と推定され，$\mathrm{LLR}(c_k|S) < 0$ なら $c_k = 1$ と推定される．

では，具体的に式 (29) や式 (30) を計算するにはどうすればいいだろうか．まず，式 (29) について考える．c_k に関連するすべてのパリティ方程式がすべて満たされる確率は，c_k に関連するパリティ方程式の一個一個が満たされる確率を掛け合わせたものである．つまり，とりあえず c_k に関連するあるパリティ方程式が満たされる確率を求めよう．ここで着目するパリティ方程式に関連するビットの集合を $\mathcal{R}_r$ としておこう．パリティ方程式が満たされるということは，「その方程式に関連する全 $|\mathcal{R}_r|$ 個のビットについて，“1” となるビットが偶数個存在する」ということに等価である．つまり，$c_k = 0$ であれば残りの $|\mathcal{R}_r| - 1$ 個のビットで “1” のビットは偶数個存在しなければそのパリティ方程式は満たされない．よって，r 番目のパリティ方程式が満たされるイベント S_r について以下が成立する．

$$
\begin{aligned}
&P(S_r|c_k = 0) \\
&= \begin{pmatrix} \text{その方程式に関連する } c_k \text{ 以外のビットのうち} \\ \text{“1” になるビットが偶数個である確率} \end{pmatrix}
\end{aligned} \tag{32}
$$

この値は i 番目のビットが “1” である確率 $p_i = P(c_i = 1|S)$ を用いて次のように計算できる．この証明は付録 A に示す．

$$
\begin{aligned}
P(S_r|c_k = 0) &= \frac{1}{2} + \frac{1}{2} \prod_{i \in \mathcal{R}_r, i \neq k} (1 - 2p_i) \\
&= \frac{1}{2} + \frac{1}{2} \prod_{i \in \mathcal{R}_r, i \neq k} \tanh\left(\frac{1}{2}\mathrm{LLR}(c_i|S)\right)
\end{aligned} \tag{33}
$$

ただし，以下の等式を利用した．

$$
\tanh\left(\frac{1}{2}\underbrace{\log\left(\frac{1-p}{p}\right)}_{\mathrm{LLR}}\right) = 1 - 2p \tag{34}
$$

この値を c_k が関連するすべての方程式で計算し，積を取ることで式 (29) が得られる．同様に $c_k = 1$ について考えれば，他のビットで “1” のビットは奇数個存在しなければいけず，そのときの確率は以下のようになる．

$$
\begin{aligned}
&P(S_r|c_k = 1) \\
&= \frac{1}{2} - \frac{1}{2} \prod_{i \in \mathcal{R}_r, i \neq k} \tanh\left(\frac{1}{2}\mathrm{LLR}(c_i|S)\right)
\end{aligned} \tag{35}
$$

この値を c_k が関連するすべての方程式で計算し，積を取ることで式 (30) が得られる．

アルゴリズムで実際に計算したい値は式 (31) であり，これは対数領域であるから，式 (31) は次のように計算できる．

$$
\begin{aligned}
&\mathrm{LLR}(c_k|S) = \log \frac{P(c_k = 0)}{P(c_k = 1)} \\
&+ \sum_{j \in \mathcal{C}_k} \log \frac{\frac{1}{2} + \frac{1}{2}\prod_{i \in \mathcal{R}_j, i \neq k} \tanh\left(\frac{1}{2}\mathrm{LLR}(c_i|S)\right)}{\frac{1}{2} - \frac{1}{2}\prod_{i \in \mathcal{R}_j, i \neq k} \tanh\left(\frac{1}{2}\mathrm{LLR}(c_i|S)\right)} \\
&= \log \frac{P(c_k = 0)}{P(c_k = 1)} \\
&+ 2\sum_{j \in \mathcal{C}_k} \tanh^{-1}\left(\prod_{i \in \mathcal{R}_j, i \neq k} \tanh\left(\frac{1}{2}\mathrm{LLR}(c_i|S)\right)\right)
\end{aligned} \tag{36}
$$

これが LDPC 符号の復号アルゴリズムの Sum-Product 復号法の更新式である．ただし，$\mathcal{C}_i$ は i 番目のビットが関連するパリティ方程式の添字の集合であり，さらに式変形において以下の等式を利用した．

$$
2\tanh^{-1}(x) = \log\left(\frac{1+x}{1-x}\right) \tag{37}
$$

4.1 線形領域 Sum-Product

そもそもなぜ対数尤度比でないといけないのだろうか．別に対数を取らずに尤度比で計算してもいいのではないだろうか．対数を取らないで尤度比の領域で計算するアルゴリズムは次の通りである．

$$\mathrm{LR}(c_k|S) = \frac{P(c_k=0)}{P(c_k=1)} \times \times \prod_{j\in\mathcal{C}_k} g\left(\prod_{i\in\mathcal{R}_r, i\neq k}\left(1-\frac{2}{1+\mathrm{LR}(c_k|S)}\right)\right) \tag{38}$$

ただし，$g(x)=(1+x)/(1-x)$ である．このとき，ビットの判定は $\mathrm{LR}(c_k|S)>1$ で $c_k=0$ であり，$\mathrm{LR}(c_k|S)<1$ で $c_k=1$ と判定する．この更新式の方が対数尤度比を計算する更新式よりも数倍高速である．また，この更新式は単に対数領域を線形領域へ変換しているだけであるため，誤り率の劣化は生じない．

A パリティ方程式が満たされる確率

以下のパリティ方程式を考える．

$$c_k + \underbrace{\sum_{i\in\mathcal{R}_r, i\neq k} c_i}_{A} = 0 \tag{39}$$

パリティ方程式を満たすためには，$A=0$ なら $c_k=0$，$A=1$ なら $c_k=1$ である必要がある．そして，$A=0$ とは c_k 以外の $\mathcal{R}_r$ に含まれるビットの “1” の数が偶数個であり，$A=1$ では奇数個である．つまり，$P(S_r|c_k)$ を求めるには “1” のビットが偶数個や奇数個である確率を求めればよい．この問題を考えるために，次のような式を考える．ただし，p_r は i 番目のビットが “1” である確率である．

$$\prod_{i\in\mathcal{R}_r, i\neq k}((1-p_i)+p_i t) \tag{40}$$

この式を展開したとき，t^n の係数は “1” が n 個現れる確率となる．$t=1$ と $t=-1$ を代入した式を比べると，両者は奇数項の符号が異なるだけである．つまり，“1” が偶数個現れる確率は $t=1$ と $t=-1$ の和の半分であり，

$$\frac{1}{2}\left(\prod_{i\in\mathcal{R}_r, i\neq k}((1-p_i)+p_i 1^n) + \prod_{i\in\mathcal{R}_r, i\neq k}((1-p_i)+p_i(-1))\right) = \frac{1}{2}+\frac{1}{2}\prod_{i\in\mathcal{R}_r, i\neq k}(1-2p_i) \tag{41}$$

同様の議論で，“1” が奇数個現れる確率は

$$\frac{1}{2}\left(\prod_{i\in\mathcal{R}_r, i\neq k}((1-p_i)+p_i 1^n) + \prod_{i\in\mathcal{R}_r, i\neq k}((1-p_i)+p_i(-1))\right) = \frac{1}{2}-\frac{1}{2}\prod_{i\in\mathcal{R}_r, i\neq k}(1-2p_i)$$

参考文献

[1] Seho Myung, Kyeongcheol Yang and Jaeyoel Kim, “Quasi-cyclic LDPC codes for fast encoding,” IEEE Transactions on Information Theory, vol. 51, no. 8, pp. 2894–2901, Aug. 2005.