

LDPC 符号に対する 2 元系列のメッセージを用いたビット反転復号法の改良

情報数理応用研究

5209C025-8

谷口祐樹

指導教員

後藤正幸

An Improved Bit Flipping Decoding Algorithm based on Messages of Binary Sequence for LDPC Codes

TANIGUCHI Yuki

1 序論

近年、情報通信や情報ストレージ技術が発達し、情報伝送に対する信頼性確保が不可欠である。そのため誤り訂正符号の構成法と復号法の重要性が増している。中でも、低密度パリティ検査 (LDPC) 符号 [1], [2] は、非零の要素が非常に少ない検査行列によって定義される符号で、繰り返し復号を行うことで優れた復号性能をもつことが知られている。一般に、LDPC 符号には正則 LDPC 符号と非正則 LDPC 符号のクラスがあるが、より広い符号クラスで表される非正則 LDPC 符号の方が優れた性能を示すことが知られている [5]。

一方、LDPC 符号に適用される繰り返し復号法の代表的なものとして、Bit-flipping (BF) 復号法や Belief-propagation (BP) 復号法 [2] が挙げられる。これらの繰り返し復号法はメッセージを繰り返し更新することで符号語ビットを推定する。繰り返し復号法の反復特性を解析する手法として密度発展法 [4], [5] が知られており、LDPC 符号アンサンブルに対する反復閾値を計算し、繰り返し復号の平均誤り率特性を理論的に解析することができる。

BF 復号法では、2 元のメッセージを用いて更新が行われる。BP 復号法は BF 復号法よりも優れた復号性能をもつが、メッセージ更新は実数によって行われるため、計算コストが BF 復号法よりも高く実装の際に問題となる恐れがある [3]。一方、通信路の誤り確率が十分に低い理想的な環境でも誤りが生じる現象はエラーフロアと呼ばれ、BF 復号法はエラーフロアが発生しやすいことが問題である [3]。

このエラーフロアを回避することを目的とした復号法として、メッセージに信頼度を表すビットを付加する方式である 3 ビット BF 復号法 [3] が提案されている。この方法により、エラーフロアの領域において誤り率が低減され、特に高い符号化率をもつ LDPC 符号に対しては BP 復号法に迫る復号性能をもちつつ、計算コストが少ない 2 元のメッセージで復号が可能である。しかし、3 ビット BF 復号法はその復号過程における更新ルールが複雑であるため、クラスが限定される正則 LDPC 符号の場合しか復号法が示されておらず、容易に非正則 LDPC 符号に一般化することが難しい。

本研究では、非正則 LDPC 符号のような広い符号クラスをもつ LDPC 符号でも実行でき、さらに付加したビットを有効に用いることで復号性能を向上させる新しい更新ルールに基づく 3 ビット BF 復号法を提案する。計算機シミュレーションによる実験により、提案した復号法の有効性を示す。また、従来および提案した復号法に対し密度発展法による性能解析を行い反復閾値が向上することを示す。

2 LDPC 符号と通信路モデル

LDPC 符号は、非零要素が非常に少ない M 行 N 列の検査行列 H により定義される符号である。検査行列 H の第 m 行第 n 列の要素を H_{mn} , $m \in [1, M]$, $n \in [1, N]$ とし¹, 検査行列 H に対して集合 $\mathcal{N}(m) \triangleq \{n : H_{mn} = 1\}$, $\mathcal{M}(n) \triangleq \{m : H_{mn} = 1\}$ を定義する。

検査行列 H は、タナーグラフと呼ばれる 2 部グラフで表現することができる。タナーグラフは検査行列 H の列に対応する N 個のノードをビットノード、行に対応する M 個のノードをチェックノードとし、非零の要素の位置に対応するノード同士を枝で結んだ 2 部グラフである。

ここで、各ノードと隣接する枝の本数を次数と呼び、次式のように次数分布多項式によって定義する。

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}, \quad \rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}. \quad (1)$$

λ_i (ρ_i) はタナーグラフの枝の総数に対する次数 i のビットノード (チェックノード) と隣接する枝の総数の比率を表す。ビットノードおよびチェックノードの次数が均一な LDPC 符号は正則 LDPC 符号と呼ばれ、不均一な場合は非正則 LDPC 符号と呼ばれる [5]。ここで d_v , d_c はそれぞれビットノード、チェックノードの最大次数である。また、符号化率 R は $R \geq R' = 1 - \int_0^1 \rho(x) dx / \int_0^1 \lambda(x) dx$ を満足し、 R' は設計符号化率を表す。チェックノードの個数は $M = N(1 - R)$ で与えられる。

2 元 LDPC 符号の符号語を $\mathbf{x} = (x_1, x_2, \dots, x_N) \in F_2^N$ とすると、 $\mathbf{x}H^T = \mathbf{0}$ をみたく。ここで、 F_2 はガロア体上の要素 $\{0, 1\}$ を表し、 T は行列の転置を表す。本研究では $(N, \lambda(x), \rho(x))$ をパラメータとする 2 元 LDPC 符号を対象とし、通信路は 2 元対称通信路 (BSC) を仮定する。各符号ビットは通信路誤り確率 p で誤りが生じ、誤りが加わった受信語を $\mathbf{y} = (y_1, y_2, \dots, y_N)$ と表し、受信側では受信語 \mathbf{y} から推定系列 $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N)$ に復号する。

3 3 ビット BF 復号法 [3]

通信路誤り確率が低い領域で復号誤り率が減少しにくくなる現象はエラーフロアと呼ばれ、タナーグラフ上の Trapping Set が原因であることが知られている [3]。3 ビット BF 復号法は受信したメッセージに信頼度を表すビットを付加して復号を行う。付加したビットを利用することで、エラーフロアの領域において誤り率を低減することが可能である。

3 ビット BF 復号法では、受信ビット y_n に 2 元のビットを付加したメッセージを復号に用いる。チェックノードからビットノードへのメッセージを $\alpha_{mn}^{(i)} = (\alpha_{mn,1}^{(i)}, \alpha_{mn,2}^{(i)}, \alpha_{mn,3}^{(i)})$, ビットノードからチェックノードへのメッセージを $\beta_{mn}^{(i)} = (\beta_{mn,1}^{(i)}, \beta_{mn,2}^{(i)}, \beta_{mn,3}^{(i)})$ と表す。ここで、 $m \in [1, M]$, $n \in [1, N]$, $H_{mn} = 1$ である。以下に 3 ビット BF 復号法の手順を示す。

[3 ビット BF 復号法]

初期設定) 各ビット位置 $n \in [1, N]$ において $\alpha_{\text{ch},n,1} := 0$, $\alpha_{\text{ch},n,2} := 1$, $\alpha_{\text{ch},n,3} := y_n$ とする。 $H_{mn} = 1$ となる (m, n) において、 $\beta_{mn,1}^{(0)} := \alpha_{\text{ch},n,1}$, $\beta_{mn,2}^{(0)} := \alpha_{\text{ch},n,2}$, $\beta_{mn,3}^{(0)} := \alpha_{\text{ch},n,3}$ とする。 $i := 1$ とし、最大繰り返し回数 I_{max} を適当な定数に設定する。

step1) $n \in [1, N]$ において以下の行処理、列処理を行う。(行処理) $m \in \mathcal{M}(n)$ において対し次式を計算する。

¹ $[a, b]$ は自然数 a から b までの集合を表す。

$$\alpha_{mn,1}^{(i)} := \prod_{n' \in \mathcal{N}(m) \setminus \{n\}} \beta_{mn',1}^{(i-1)}, \quad (2)$$

$$\alpha_{mn,3}^{(i)} := \sum_{n' \in \mathcal{N}(m) \setminus \{n\}} \beta_{mn',3}^{(i-1)} \pmod{2}, \quad (3)$$

$$\alpha_{mn,2}^{(i)} := \begin{cases} \prod_{n' \in \mathcal{N}(m) \setminus \{n\}} \beta_{mn',2}^{(i-1)}, & \text{if } \alpha_{mn,1}^{(i)} = 1; \\ \prod_{\substack{n' \in \mathcal{N}(m) \setminus \{n\} \\ \beta_{mn',1}^{(i-1)} = 0}} \beta_{mn',2}^{(i-1)}, & \text{otherwise.} \end{cases} \quad (4)$$

(列処理) $m \in \mathcal{M}(n)$ において, $\alpha_{m'n}^{(i)} = (\alpha_{m'n,1}^{(i)}, \alpha_{m'n,2}^{(i)}, \alpha_{m'n,3}^{(i)})$, $\alpha_{m''n}^{(i)} = (\alpha_{m''n,1}^{(i)}, \alpha_{m''n,2}^{(i)}, \alpha_{m''n,3}^{(i)})$ ($m', m'' \in \mathcal{M}(n) \setminus \{m\}$) を比較し, $\beta_{mn}^{(i)} = (\beta_{mn,1}^{(i)}, \beta_{mn,2}^{(i)}, \beta_{mn,3}^{(i)})$ を更新する. 以下に一例を示す².

ex.) もし次式が成り立つとき,

$$\begin{cases} \alpha_{m'n,1}^{(i)} = \alpha_{m''n,1}^{(i)} = 1; \\ \alpha_{m'n,2}^{(i)} = \alpha_{m''n,2}^{(i)} = 1; \\ \alpha_{m'n,3}^{(i)} = \alpha_{m''n,3}^{(i)} = 1, \end{cases} \quad (5)$$

$\beta_{mn,1}^{(i)} := 1, \beta_{mn,2}^{(i)} := 1, \beta_{mn,3}^{(i)} := 1$ とする. ここで, $m', m'' \in \mathcal{M}(n) \setminus \{m\}, m' \neq m''$ とする. メッセージの値がそれ以外の場合の更新ルールは表 1 に示す.

step2) 以下の step2-1), 2-2) を行う.

step2-1) $n \in [1, N]$ において, 以下のように $\hat{x}^{(i)} = (\hat{x}_1^{(i)}, \hat{x}_2^{(i)}, \dots, \hat{x}_N^{(i)})$ を求める.

case1) $\sum_{m \in \mathcal{M}(n)} \alpha_{mn,1}^{(i)} \geq 1$ の場合, 以下の case1-1), 1-2) のいずれかを実行する.

case1-1) $\sum_{m \in \mathcal{M}(n)} \alpha_{mn,2}^{(i)} \geq 1$ の場合, 次式を計算する.

$$\hat{x}_n^{(i)} := \begin{cases} 1, & \text{if } \sum_{m \in \mathcal{M}(n), \alpha_{mn,1}^{(i)}=1, \alpha_{mn,2}^{(i)}=1} \alpha_{mn,3}^{(i)} \geq |k|/2; \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

ここで, $k = \{m : m \in \mathcal{M}(n), \alpha_{mn,1}^{(i)} = 1, \alpha_{mn,2}^{(i)} = 1\}$ である³.

case1-2) case1-1) の条件を満たさない場合, 次式を計算する.

$$\hat{x}_n^{(i)} := \begin{cases} 1, & \text{if } \sum_{m \in \mathcal{M}(n), \alpha_{mn,1}^{(i)}=1} \alpha_{mn,3}^{(i)} \geq |k|/2; \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

ここで, $k = \{m : m \in \mathcal{M}(n), \alpha_{mn,1}^{(i)} = 1\}$ である.

case2) $\sum_{m \in \mathcal{M}(n)} \alpha_{mn,1}^{(i)} = 0$ の場合, 以下の case2-1), 2-2) のいずれかを実行する.

case2-1) $\sum_{m \in \mathcal{M}(n)} \alpha_{mn,2}^{(i)} \geq 1$ の場合, 次式を計算する.

$$\hat{x}_n^{(i)} := \begin{cases} 1, & \text{if } \sum_{m \in \mathcal{M}(n), \alpha_{mn,2}^{(i)}=1} \alpha_{mn,3}^{(i)} \geq |k|/2; \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

ここで, $k = \{m : m \in \mathcal{M}(n), \alpha_{mn,2}^{(i)} = 1\}$ である.

case2-2) case2-1) の条件を満たさない場合, 次式を計算する.

$$\hat{x}_n^{(i)} := \alpha_{ch,n,3} \quad (9)$$

step2-2) $\hat{x}^{(i)} H^T = 0$ または $i = I_{\max}$ のとき $\hat{x} := \hat{x}^{(i)}$ とし, 復号を終了する. それ以外のときは $i := i + 1$ とし step1) へ行く. \square

²場合分けが多いため, 本論文では省略

³集合 A に対し, $|A|$ は集合 A 内の要素数を表す.

表 1. 3 ビット BF 復号法の列処理における更新ルール. チェックノードからのメッセージを $\alpha_{m'n}^{(i)} = a_1, \alpha_{m''n}^{(i)} = a_2$, 受信語を y , 更新するメッセージを $\beta_{mn}^{(i)} = m$ としたときのそれぞれの更新結果を示す.

a_1	a_2	y	m	a_1	a_2	y	m	a_1	a_2	y	m
000	000	0	010	010	100	1	010	100	101	0	010
000	000	1	011	010	101	0	011	100	101	1	011
000	010	0	010	010	101	1	101	100	110	0	110
000	010	1	000	010	110	0	110	100	110	1	110
000	011	0	000	010	110	1	100	100	111	0	101
000	011	1	011	010	111	0	101	100	111	1	101
000	100	0	100	010	111	1	111	101	101	0	101
000	100	1	010	011	011	0	000	101	101	1	111
000	101	0	011	011	011	1	101	101	110	0	100
000	101	1	101	011	100	0	100	101	110	1	100
000	110	0	110	011	100	1	010	101	111	0	111
000	110	1	100	011	101	0	011	101	111	1	111
000	111	0	101	011	101	1	101	110	110	0	110
000	111	1	111	011	110	0	110	110	110	1	110
010	010	0	100	011	110	1	100	110	111	0	010
010	010	1	000	011	111	0	101	110	111	1	011
010	011	0	010	011	111	1	111	111	111	0	111
010	011	1	011	100	100	0	110	111	111	1	111
010	100	0	100	100	100	1	100				

4 提案手法

4.1 背景

3 ビット BF 復号法は BF 復号法を実行したときに発生するエラーフロアに対してビットを付加することにより, エラーフロア領域での復号誤り率を低減することが可能である. しかし, 3 ビット BF 復号法の列処理は表 1 に示したようにビットノードの次数が 3 の符号の場合にしか提案されていない. 一方, 正則 LDPC 符号よりも非正則 LDPC 符号の方が優れた性能をもつといわれている [5]. そのため本研究では, 非正則 LDPC 符号に対しても実行可能な 3 ビット BF 復号法を提案する.

4.2 提案復号法

3 ビット BF 復号法で用いる全てのメッセージは長さ 3 の $\{0, 1\}$ の系列であることから, 各メッセージの 1, 2 ビット目を 2 進数で表された信頼度と見なすことが可能である. そこで BP 復号法の更新式を利用し, 列処理における更新では各メッセージの 1, 2 ビット目に対して 2 進演算を行う. 演算上 $\alpha_{mn,3}^{(i)} = 0, \alpha_{mn,3}^{(i)} = 1$ をもつメッセージは別々に演算を行う. 更新の際に用いるメッセージを $\mu_0, \mu_1, \mu = (\mu(0)\mu(1))_2$ と表し, 演算時の桁上がりに対応するためビットノードの最大次数に応じた桁数をもつとする. ここで, 演算時に必要な桁数は $\lceil \log_2(d_v(2^2 - 1)) \rceil$ である. 以下に提案復号法を示す. なお, step1) の行処理, step2) は 3 ビット BF 復号法と同様である.

[提案復号法]

(列処理) $m \in \mathcal{M}(n), b \in \{0, 1\}$ において, 次式を計算する.

$$\mu_b := \sum_{\substack{m' \in \mathcal{M}(n) \setminus \{m\} \\ \alpha_{m'n,3}^{(i)} = b}} (\alpha_{m'n,1}^{(i)} \alpha_{m'n,2}^{(i)})_2, \quad (10)$$

$$\mu := \begin{cases} \mu_0 - \mu_1, & \text{if } \mu_0 \geq \mu_1; \\ \mu_1 - \mu_0, & \text{otherwise.} \end{cases} \quad (11)$$

式 (11) を計算後, $\mu := \min\{\mu, (11)_2\}$ を計算し, 次式のように μ を再び $\{0, 1\}$ の系列と表記する.

$$\beta_{mn,1}^{(i)} := \mu(0), \beta_{mn,2}^{(i)} := \mu(1), \quad (12)$$

$$\beta_{mn,3}^{(i)} := \begin{cases} 0, & \text{if } \mu_0 > \mu_1; \\ 1, & \text{if } \mu_0 < \mu_1; \\ \alpha_{ch,n,3}, & \text{otherwise.} \end{cases} \quad (13)$$

\square

上記の復号法は、ビットノードの度数に依存せずに更新を行うことが可能である。また、2進演算を用いることでBP復号法の更新式に近い処理を行うことができているため、復号性能の向上が期待できる。

5 密度発展法による性能解析

密度発展法 [4], [5] は LDPC 符号アンサンブル上での繰り返し復号法の平均誤り確率を計算する反復計算法である。反復計算過程で交換する各メッセージに更新される確率を繰り返し計算する。密度発展法では、通信路と復号法の対称性より全ゼロの符号語を送り、符号長 N は N とし、LDPC 符号のグラフはループが存在しないものと仮定する。

ここで、 $\mathcal{U} = \{(11)_2, (10)_2, (01)_2, (00)_2\}$ を定義する。提案復号法の繰り返し $i (\geq 1)$ 回目において、メッセージ $\alpha_{mn}^{(i)}$ のうち、1,2 ビット目を2進数で表した値 $(\alpha_{mn,1}^{(i)}, \alpha_{mn,2}^{(i)})_2$ が u 、かつ3ビット目を $q_{b,u}^{(i)}$ が b をとる確率を $q_{b,u}^{(i)}$ ($u \in \mathcal{U}$) とし、 $q_b^{(i)} = \sum_{u \in \mathcal{U}} q_{b,u}^{(i)}$ とする。提案復号法の繰り返し $i (\geq 0)$ 回目において、メッセージ $\beta_{mn}^{(i)}$ のうち、1,2 ビット目を2進数で表した値 $(\beta_{mn,1}^{(i)}, \beta_{mn,2}^{(i)})_2$ が u 、かつ3ビット目を $p_{b,u}^{(i)}$ が b をとる確率を $p_{b,u}^{(i)}$ ($u \in \mathcal{U}$) とし、 $p_b^{(i)} = \sum_{u \in \mathcal{U}} p_{b,u}^{(i)}$ とする。ただし、 $p_{1,(01)_2}^{(0)} = p$, $p_{0,(01)_2}^{(0)} = 1 - p$, $p_{1,u'}^{(0)} = 0$ ($u' \in \mathcal{U} \setminus \{(01)_2\}$), $p_{0,u'}^{(0)} = 0$ ($u' \in \mathcal{U} \setminus \{(01)_2\}$) とする。提案復号法のある列処理において、メッセージ $\alpha_{mn}^{(i)}$ のうち、1,2 ビット目が u 、かつ3ビット目が b をとる数を $a_{b,u}$ とする。

このとき、提案復号法で更新するメッセージ $\alpha_{mn}^{(i)}$ がそれぞれの値をとる確率は $b \in \{0, 1\}$ において、以下のように計算される。

$$q_{b,(11)_2}^{(i)} = \sum_{k=2}^{d_c} \rho_k \sum_{\substack{t=0, \\ t \bmod 2=b}} \binom{k-1}{t} (p_{0,(11)_2}^{(i-1)})^{k-1-t} (p_{1,(11)_2}^{(i-1)})^t, \quad (14)$$

$$q_{b,(10)_2}^{(i)} = \sum_{k=2}^{d_c} \rho_k \sum_{\substack{t=0, \\ t \bmod 2=b}} \binom{k-1}{t} \times \left\{ (p_{0,(11)_2}^{(i-1)} + p_{0,(10)_2}^{(i-1)})^{k-1-t} (p_{1,(11)_2}^{(i-1)} + p_{1,(10)_2}^{(i-1)})^t - (p_{0,(11)_2}^{(i-1)})^{k-1-t} (p_{1,(11)_2}^{(i-1)})^t \right\}, \quad (15)$$

$$q_{b,(01)_2}^{(i)} = \sum_{k=2}^{d_c} \rho_k \sum_{\substack{s_2+t_2 \geq 1 \\ s_1+t_1 < k-1 \\ (t_1+t_2) \bmod 2=b \\ 0 \leq s_1, s_2, t_1, t_2 \leq k-1}} \binom{k-1}{s_1, s_2, t_1, t_2} \times \left\{ (p_{0,(11)_2}^{(i-1)} + p_{0,(10)_2}^{(i-1)})^{s_1} (p_{1,(11)_2}^{(i-1)} + p_{1,(10)_2}^{(i-1)})^{t_1} \times (p_{0,(01)_2}^{(i-1)})^{s_2} (p_{1,(01)_2}^{(i-1)})^{t_2} - (p_{0,(11)_2}^{(i-1)} + p_{0,(10)_2}^{(i-1)})^{s_1+s_2} \times (p_{1,(11)_2}^{(i-1)} + p_{1,(10)_2}^{(i-1)})^{t_1+t_2} \right\}. \quad (16)$$

式 (14)–(17) で用いるメッセージ $\beta_{mn}^{(i)}$ がそれぞれの値をとる確率は以下のように計算される。

$$p_{b,u}^{(i)} = p \sum_{k=2}^{d_v} \lambda_k \sum_{\substack{0 \leq a_{\kappa, \tau} \leq k-1 \\ \kappa \in \{0,1\}, \tau \in \mathcal{U} \\ \sum_{\kappa} \sum_{\tau} a_{\kappa, \tau} = k-1 \\ \sum_{0,b,u}}} (k-1)! \prod_{\tilde{\kappa} \in \{0,1\}} \prod_{\tilde{\tau} \in \mathcal{U}} \frac{(q_{\tilde{\kappa}, \tilde{\tau}}^{(i)})^{a_{\tilde{\kappa}, \tilde{\tau}}}}{a_{\tilde{\kappa}, \tilde{\tau}}!} \\ + (1-p) \sum_{k=2}^{d_v} \lambda_k \sum_{\substack{0 \leq a_{\kappa, \tau} \leq k-1 \\ \kappa \in \{0,1\}, \tau \in \mathcal{U} \\ \sum_{\kappa} \sum_{\tau} a_{\kappa, \tau} = k-1 \\ \sum_{1,b,u}}} (k-1)! \prod_{\tilde{\kappa} \in \{0,1\}} \prod_{\tilde{\tau} \in \mathcal{U}} \frac{(q_{\tilde{\kappa}, \tilde{\tau}}^{(i)})^{a_{\tilde{\kappa}, \tilde{\tau}}}}{a_{\tilde{\kappa}, \tilde{\tau}}!} \quad (18)$$

ここで、条件 $g_{b_1, b_2, u}$ ($b_1, b_2 \in \{0, 1\}$) は次式で与えられる。

$$g_{b_1, b_2, u} \Rightarrow \begin{cases} (-1)^{b_2} \times \left(\sum_{u_0 \in \mathcal{U}} u_0 \times (a_{0, u_0}) - \sum_{u_1 \in \mathcal{U}} u_1 \times (a_{1, u_1}) \right) \\ \geq u - (-1)^{b_1} \times (01)_2, & \text{if } u = (11)_2; \\ (-1)^{b_2} \times \left(\sum_{u_0 \in \mathcal{U}} u_0 \times (a_{0, u_0}) - \sum_{u_1 \in \mathcal{U}} u_1 \times (a_{1, u_1}) \right) \\ = u - (-1)^{b_1} \times (01)_2, & \text{otherwise,} \end{cases} \quad (19)$$

したがって、メッセージ $\beta_{mn}^{(i)}$ の平均誤り確率 $p_1^{(i)}$ は、

$$p_1^{(i)} = \sum_{u \in \mathcal{U}} p_{1,u}^{(i)}, \quad (20)$$

となる。また反復閾値は、

$$p^* = \sup \{ p : 0 \leq p \leq 0.5, \lim_{i \rightarrow \infty} p_1^{(i)} = 0 \}, \quad (21)$$

によって定められる。

6 シミュレーション結果と考察

6.1 シミュレーション条件

提案の有効性を示すため、計算機シミュレーションによる評価を行った。まず正則 LDPC 符号に対し提案復号法と従来復号法として3ビットBF (3BF) 復号法、BF 復号法、BP 復号法を実行したときの復号性能を示す。さらに非正則 LDPC 符号に対し提案復号法とBF 復号法、BP 復号法を実行したときの復号性能を示す。ここで符号1, 2は符号化率 $R = 0.5$ 、符号3は符号化率 $R = 0.7$ で以下のような異なる次数分布をもち、符号1は正則 LDPC 符号、符号2, 3は非正則 LDPC 符号を表す。

符号1) $\lambda(x) = x^2$, $\rho(x) = x^5$.

符号2) $\lambda(x) = 0.2997x + 0.1149x^2 + 0.5854x^5$, $\rho(x) = x^6$.

符号3) $\lambda(x) = 0.18x + 0.64x^2 + 0.18x^5$, $\rho(x) = x^9$.

通信路には誤り確率 p のBSCを仮定し、符号長 $N = 1000$, $I_{\max} = 100$ とし、符号に対し各復号法を 10^7 回実行するか、もしくは復号が200回失敗するまでシミュレーションを行った。縦軸に復号後のビット誤り確率 (BER)、横軸にBSCの通信路誤り確率 p をとる。

6.2 シミュレーション結果と考察

6.2.1 正則 LDPC 符号

図1に復号性能を検証するため、正則 LDPC 符号 (符号1) を用いたときの3BF復号法と提案復号法の復号結果を示す。比較としてBF復号法、BP復号法の復号結果も同時に示す。

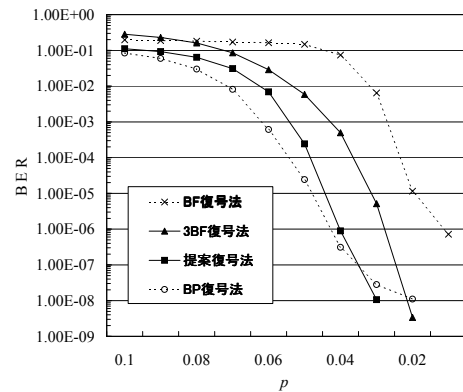


図1. 符号1を用いて各復号法を実行したときの復号結果

$$q_{b,(00)_2}^{(i)} = \sum_{k=2}^{d_c} \rho_k \sum_{\substack{s_2+t_2 \geq 1 \\ s_1+t_1 < k-1 \\ (t_1+t_2) \bmod 2 = b \\ 0 \leq s_1, s_2, t_1, t_2 \leq k-1}} \binom{k-1}{s_1, s_2, t_1, t_2} \times \left\{ (p_{0,(11)_2}^{(i-1)} + p_{0,(10)_2}^{(i-1)} + p_{0,(01)_2}^{(i-1)})^{s_1} (p_{1,(11)_2}^{(i-1)} + p_{1,(10)_2}^{(i-1)} + p_{1,(01)_2}^{(i-1)})^{t_1} \right. \\ \left. \times (p_{0,(00)_2}^{(i-1)})^{s_2} (p_{1,(00)_2}^{(i-1)})^{t_2} - (p_{0,(11)_2}^{(i-1)} + p_{0,(10)_2}^{(i-1)})^{s_1} (p_{1,(11)_2}^{(i-1)} + p_{1,(10)_2}^{(i-1)})^{t_1} \times (p_{0,(01)_2}^{(i-1)})^{s_2} (p_{1,(01)_2}^{(i-1)})^{t_2} \right\}. \quad (17)$$

図1より、BF復号法、BP復号法を実行したときに生じるエラーフロアの領域で3BF復号法と提案復号法がともに復号誤り率が低減しており、復号性能が向上していることがわかる。また提案復号法が3BF復号法よりも高い復号性能をもつことから、付加したビットをより有効に利用することができたといえる。

6.2.2 非正則 LDPC 符号

図2, 3に非正則 LDPC 符号 (符号 2, 3) を用いたときの提案復号法の復号結果を示す。比較としてBF復号法、BP復号法の復号結果も同時に示す。非正則である符号 2, 3 に対しては、3BF復号法は実行することができないことに注意されたい。

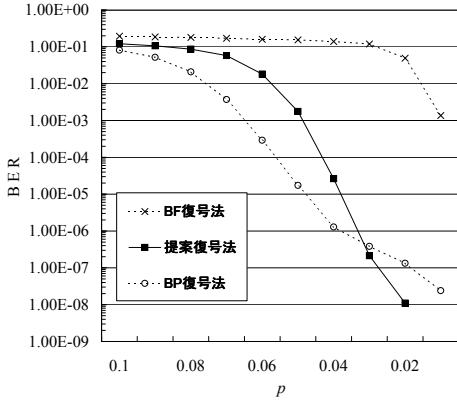


図 2. 符号 2 を用いて各復号法を実行したときの復号結果

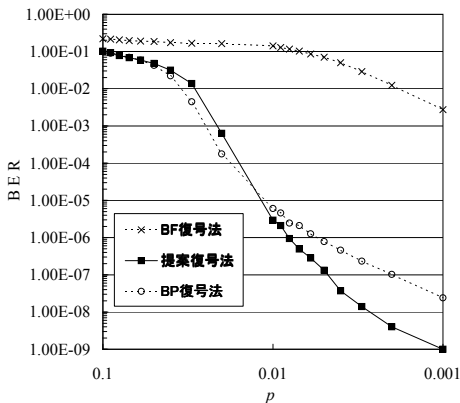


図 3. 符号 3 を用いて各復号法を実行したときの復号結果

図2より、非正則 LDPC 復号法を用いた場合でも同様にエラーフロア領域での復号誤り率が低減していることがわかる。また図3より、符号化率が高い符号に対して復号を行うとき、通信路誤り確率が高い領域でもBP復号法に近い復号性能を示すことがわかる。

7 密度発展法による数値計算結果と考察

式 (21) を用いて計算した提案復号法の反復閾値を表 2 に示す。比較のため、BF復号法、3BF復号法、BP復号法の反復閾値も同時に示した。

表 2. 各復号法を実行したときの反復閾値 (符号 1)

	BF 復号法	3BF 復号法	提案復号法	BP 復号法
p^*	0.039	0.050	0.056	0.080

表 2 より、3BF 復号法は BF 復号法より反復閾値が向上していることを確認できる。さらに、提案復号法が 3BF 復号法よりも反復閾値が向上していることから、付加したビットを有効に利用しているといえる。また、提案復号法は BP 復号法よりも低い反復閾値を示しているが、図 3 のように BER が低いときの復号性能が近いことから、付加したビットがエラーフロア領域での性能向上に繋がったと考えられる。

また、提案復号法の復号性能についての検証を行うため、密度発展法を用いて計算した繰り返しごとのメッセージ $\beta_{mn}^{(i)}$ の割合を 3BF 復号法、提案復号法のそれぞれに対し計算する。その結果、提案復号法では i を大きくするに従い、 $\beta_{mn}^{(i)}$ が 110 をとる割合が増加していることを確認した。そのため提案復号法の方が反復の過程でより信頼度の高いメッセージを判別し、それらのメッセージを優先的に更新に用いることができているといえる。以上のことから、復号性能が向上したと考えることができる。

8 まとめと今後の課題

本研究では従来、正則 LDPC 符号に対してのみ示されていた 3 ビット BF 復号法に対し、非正則 LDPC 符号でも実行可能であると共に、付加したビットで表される信頼度を有効に用いる 3 ビット BF 復号法を提案した。シミュレーション及び密度発展法により、提案した復号法は 3 ビット BF 復号法と同様にエラーフロア領域における復号性能が向上することを示した。今後の課題として、BP 復号法のような更新に実数を用いる復号法に対しても、3 ビット BF 復号法の着想を利用できるような更新ルールを決定する必要がある。

参考文献

- [1] R.G. Gallager, *Low density parity check codes*, MIT Press, 1963.
- [2] D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
- [3] S.K. Planjery, D. Declercq, S.K. Chilappagari, and B. Vasic, "Multilevel decoders surpassing belief propagation on the binary symmetric channel," *Proc. 2010 IEEE International Symposium on Information Theory.*, pp. 769–773, Austin, TX, Jun. 2010.
- [4] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [5] T. Richardson and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.