部分グラフから構成される符号化率が可変なパンクチャド低密度パリティ検査符号の構成法とその解析

情報数理応用研究　　5210C008-4　長田佳史
指導教員　後藤正幸

Analysis of Rate-Compatible Punctured Low-Density Parity-Check Codes Consisting of Two Subgraphs

OSADA Keishi

1　はじめに

近年，情報化社会における情報通信技術の発達と共に，情報伝送に対する信頼性の確保が不可欠となっており，誤り訂正符号・復号技術の果たす役割が重要になっている．誤り訂正符号の中でも低密度パリティ検査（Low-Density Parity-Check:LDPC) 符号と確率伝播型 (Belief-Propagation:BP) 復号法の組み合わせ [1], [2] は，理論的な限界に迫る高い誤り訂正能力を発揮することから近年盛んに研究が行われており，その性能解析が重要なテーマの 1 つになっている [2]-[6], [9]．LDPC 符号は確率的に構成される符号であり，個々の有限長の符号に対しその性能を理論的に評価することは難しいため，構成されうる符号の集合である符号アンサンブルに対し符号長が無限大の場合の平均的な性能を評価するアプローチが一般的である．BP 復号法の解析手法として，密度発展法 [3], [9] やガウス近似法 [5]，EXIT チャート法 [6] などが用いられている．このような解析手法を用いると性能解析のみでなく，性能の高い LDPC 符号を構成することも可能となる [4]．

一方，誤り訂正符号は符号化率と訂正能力がトレードオフの関係にあるため，通信路の状態により適切な符号化率をもつ符号を用いる必要があるが，符号化率ごとに符号を用意することは効率的ではない．そのため，符号語系列の一部をパンクチャし送信しないことで，パンクチャするビット数によって符号化率を自由に設定する方法が研究されている．この方法では，パンクチャするビットの順番を予め決めておくことで，1 つの符号化器と復号器の組を用意するだけで符号化率を可変にすることができる．LDPC 符号に対しこのようにパンクチャして得られた符号は，符号化率が可変なパンクチャド LDPC (RCP-LDPC) 符号 [7], [8], [11]-[14] と呼ばれ，必要な符号化率に応じて効果的なパンクチャドビットを選択し，BP 復号法を用いて復元することで高い性能が得られる．符号長が有限の LDPC 符号に対し性能の高い RCP-LDPC 符号を構成する多くの研究 [8], [11], [12] が行なわれており，著者らも広域な符号化率に対応した効果的なパンクチャドビットの選択法を提案している [13], [14]．

一方，符号長が無限大の場合，予め最適化された LDPC 符号の次数分布に対し，ガウス近似法 [5] を用いて各次数ごとにパンクチャするビットを導出する方法が提案されており，理論限界に近い性能を示すことが知られている [7] が，符号を符号アンサンブルで表現できない限り，このような解析手法を用いて符号を設計できない．またパンクチャドビットを復元する仕組みは消失訂正と同じ原理である [8] ため，消失訂正で効果的な手法を適用することで性能が向上する見込みがある．そこで本研究では複数の部分グラフから構成することで高い消失訂正能力を示す LDPC 符号 [10] に着目し，パンクチャドビットの復元に適した構造をもち，符号アンサンブルとして定義される新しい LDPC 符号の構成法を提案する．これにより，提案した符号のアンサンブルに対し，ガウス近似法を用いた理論解析が可能となり，必要な符号化率に応じて効果的なパンクチャ比率を決定できることを示す．ガウス近似法による数値計算結果より，提案した LDPC 符号は通常の LDPC 符号より高い性能を示すことを明らかにする．

2　準備

2.1　RCP-LDPC 符号と通信路モデル

2 元 LDPC 符号 [1], [2] はタナーグラフと呼ばれる疎な 2 部グラフによって定義される符号である．2 種類のノードは変数ノードとチェックノードと呼ばれ，両者は枝で連結されている．変数ノードとチェックノードの数を N, M $(N > M)$ とする．LDPC 符号の符号語を $\boldsymbol{c} = (c_1, c_2, \ldots, c_N) \in F_2^N$ とし，各符号語ビット c_n は変数ノードと 1 対 1 対応する．ここで，F_2 は 2 元ガロア体上の要素 $\{0, 1\}$ を表す．それぞれのチェックノードと連結された変数ノードに対応する符号語ビットの排他的論理和は 0 となるよう制約されている．また符号語 $\boldsymbol{c}$ は $\{0 \to +1, 1 \to -1\}$ という変換により送信系列 $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ に写像され，加法的白色ガウス雑音 (AWGN) 通信路を介して送信し，雑音 $\boldsymbol{e} = (e_1, e_2, \ldots, e_N)$ が加わった受信語 $\boldsymbol{y} = (y_1, y_2, \ldots, y_N) \in \mathcal{R}^N$ を得る．ここで $\mathcal{R}$ は実数の集合を表す．各ノードがもつ枝の本数を次数と呼び，LDPC 符号は次式で示す次数分布多項式によって定義される．

$$\lambda(x) = \sum_{i=2}^{d_\ell} \lambda_i x^{i-1}, \qquad \rho(x) = \sum_{i=2}^{d_r} \rho_i x^{i-1}. \tag{1}$$

λ_i (ρ_i) はタナーグラフの枝の総数に対する次数 i の変数ノード（チェックノード）から出る枝の総数の比率を表し，d_ℓ (d_r) は変数ノード (チェックノード) の次数の最大値を表す．ノード間の枝の連結方法は確率的に決定され，与えられた次数分布に対し得られる全ての符号の集合を符号アンサンブルと呼ぶ．次数分布は LDPC 符号の性能を大きく左右するため，密度発展法 [4] やガウス近似法 [5] などによって予め最適化された次数分布が用いられる．LDPC 符号の符号化率 R_0 は次式で定義される．

$$R_0 = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = 1 - \frac{\sum_{i=2}^{d_r} \frac{\rho_i}{i}}{\sum_{i=2}^{d_\ell} \frac{\lambda_i}{i}}. \tag{2}$$

次数 i $(2 \le i \le d_\ell)$ の変数ノードにおけるパンクチャ比率を $\pi_i^{(0)}$ $(0 \le \pi_i^{(0)} \le 1)$ とし，パンクチャド比率分布 $\pi^{(0)}(x)$ を $\pi^{(0)}(x) = \pi_2^{(0)} x + \pi_3^{(0)} x^2 + \cdots + \pi_{d_l}^{(0)} x^{d_\ell - 1}$ と表す．符号化率 R_0 の符号から次数 i の変数ノードを $\pi_i^{(0)}$ の比率でパンクチャすると，符号全体におけるパンクチャ比率 $P^{(0)}$ は次式で与えられる．

$$P^{(0)} = \frac{\sum_{i=2}^{d_\ell} \pi_i^{(0)} \frac{\lambda_i}{i}}{\sum_{i=2}^{d_\ell} \frac{\lambda_i}{i}}. \tag{3}$$

RCP-LDPC 符号を構成するにあたっては，最大でパンクチャするビット数とパンクチャする位置及び順番を決めておくことで様々な符号化率に対応して符号が構成できる．

LDPC 符号に対し BP 復号法 [1], [2] を実行するときの各ビット位置 $n = 1, 2, ..., N$ における対数尤度比 (LLR) を

$$L_n = \log \frac{\Pr(y_n | x_n = +1)}{\Pr(y_n | x_n = -1)}. \tag{4}$$

と表す．LLR の値が正の場合は +1 に，負の場合は −1 に復号する．復号過程において変数ノードからチェックノードへの LLR メッセージを v，チェックノードから変数ノードへの LLR メッセージを u とすると，復号法の更新ルールはそれぞれ以下の式となる．

$$v = \sum_{i=0}^{\alpha-1} u_i, \qquad \tanh \frac{u}{2} = \prod_{j=1}^{\beta-1} \tanh \frac{v_j}{2}. \tag{5}$$

ここで，変数ノードが接続するチェックノード数は α 個，チェックノードが接続する変数ノード数は β 個であり，u_0 は初期値，u_i $(1 \le i \le \alpha - 1)$，v_j $(1 \le j \le \beta - 1)$ はそれぞれ変数ノード，チェックノードへの入力メッセージである．

符号語ビット x_n をパンクチャした場合，$L_n = 0$ であり，送信したビットが +1 か −1 か判定できないため，BP 復号法の実行過程においてパンクチャドビットのメッセージを非零に更新することで送信ビットに更新する（復元する）必要がある．(5) 式における u の更新から，パンクチャドビットの位置に対応する変数ノード (パンクチャド変数ノード) v と接続するチェックノードのうち 1 つも v 以外のパンクチャド変数ノードと接続していないとき v は復元される．

符号長が有限の LDPC 符号に対し性能の高い RCP-LDPC 符号を構成する手法として，パンクチャドビットが復元されるまでの BP 復号法の繰り返し回数を小さくすることが良く知られている [8]．しかしこの手法では多くのビットをパンクチャできないため，一部を確定的に構成することで多くの符号語ビットがパンクチャ可能で，なおかつ効率的な符号化が可能な性能を併せ持つ E^2RC 符号が提案されている [11]．この符号は高い符号化率では良い性能を発揮するが，元の符号の性能が悪く，符号化率が低い場合は確率的に構成する LDPC 符号よりも性能が劣ることが知られている．これら符号長が有限の個々の符号と BP 復号法の組み合わせに対し，性能を理論的に評価することは困難である．

2.2 ガウス近似法による性能解析と符号構成法

LDPC 符号・BP 復号法の解析手法の 1 つにガウス近似法 [5] があり，符号長が無限大の符号アンサンブルに対し符号の平均性能を評価することができる．ガウス近似法では，BP 復号法でやりとりするメッセージの確率密度関数を正規分布 $N(m, \delta^2)$ で近似する．正規分布で近似されたメッセージ τ の確率密度関数は以下のように表される．

$$f(\tau) = \frac{1}{\sqrt{2\pi\delta^2}} \exp\Big\{ -\frac{(\tau - m)^2}{2\delta^2} \Big\}. \tag{6}$$

また，密度発展法 [4] よりメッセージ τ のもとで対称条件 $f(\tau) = f(-\tau)e^{\tau}$ が成り立つことが知られており，そのためガウス近似法では分散 δ^2 と平均 m が $\delta^2 = 2m$ を満足する．これにより，メッセージ v，u の確率密度関数は，それぞれの分布の平均 m のみを計算する漸化式で容易に表現することができる．

BP 復号法の繰り返し ℓ 回目に更新されるメッセージ v（次数 i），u（次数 j）の平均 $m_{v,i}^{(\ell)}, m_{u,j}^{(\ell)}$ は次式のように求められる．

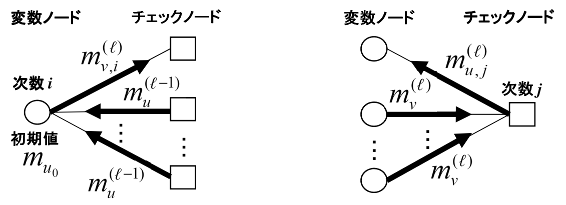
$$m_{v,i}^{(\ell)} = m_{u_0} + (i-1) m_u^{(\ell-1)}. \tag{7}$$

$$m_{u,j}^{(\ell)} = \phi^{-1}\bigg(1 - \Big[1 - \sum_{i=2}^{d_\ell} \lambda_i \phi(m_{v,i}^{(\ell)})\Big]^{j-1}\bigg). \tag{8}$$

ここで $\phi(\tau)$ は

$$\phi(\tau) = \begin{cases} e^{0.0564\tau^2 - 0.4856\tau}, & 0 \le \tau < 0.867861; \\ e^{-0.4527\tau^{0.86} + 0.0218}, & 0.867861 \le \tau < 10; \\ \sqrt{\frac{\pi}{\tau}} e^{-\frac{\pi}{4}} (1 + \frac{1}{14\tau} - \frac{3}{2\tau}), & 10 \le \tau, \end{cases}$$

である．また各メッセージの更新は図 2 のようになる．



(a) $m_{v,i}^{(\ell)}$ の更新　　(b) $m_{u,j}^{(\ell)}$ の更新

図 2．メッセージの更新

2.1 節より，符号をパンクチャした場合パンクチャド変数ノードにおけるメッセージの初期値は 0 であるため，繰り返し ℓ 回目におけるメッセージ v, u がそれぞれ復元されていない確率 $e^{(\ell)}, \epsilon^{(\ell)}$ を次式のように求めることができる [7]．

$$e^{(\ell)} = \sum_{j=2}^{d_\ell} \lambda_j \pi_j^{(0)} \big(\epsilon^{(\ell-1)}\big)^{j-1}. \tag{9}$$

$$\epsilon^{(\ell)} = \sum_{s=2}^{d_r} \rho_s \big(1 - (1 - e^{(\ell)})^{s-1}\big). \tag{10}$$

以上の (7)–(10) 式から，繰り返し ℓ 回目におけるメッセージ u の平均 $m_u^{(\ell)}$ は次式ようになる．

$$\begin{aligned} m_u^{(\ell)} = \ & \sum_{s=2}^{d_r} \rho_s \phi^{-1}\bigg(1 - \frac{1}{(1-e^{(\ell)})^{s-1}}\Big[1 - \sum_{j=2}^{d_\ell} \\ & \times\Big\{\lambda_j \pi_j^{(0)} \sum_{i=0}^{j-1} {}_{j-1}\chi_i^{(\ell)} \phi(i m_u^{(\ell-1)}) + \lambda_j (1 - \pi_j^{(0)}) \\ & \times \sum_{i=0}^{j-1} {}_{j-1}\chi_i^{(\ell)} \phi(i m_u^{(\ell-1)} + m_{u_0})\Big\}\Big]^{s-1}\bigg). \end{aligned} \tag{11}$$

ここで ${}_{j-1}\chi_i^{(\ell)}$ は ${}_{j-1}\chi_i^{(\ell)} = {}_{j-1}C_i (\epsilon^{(\ell-1)})^{j-1-i} (1 - \epsilon^{(\ell-1)})^i$ であり，(11) 式で $m_u^{(\ell)}$ を繰り返し計算することで，復号が成功する AWGN 通信路における誤差分布の標準偏差の上限である閾値 σ^* を求めることができる．

定義 1 σ^* は繰り返し回数 $\ell \to \infty$ において，$m_u^{(\ell)} \to \infty$ を満たす AWGN 通信路における誤差分布の標準偏差 σ の上限値である．□

ガウス近似法は与えられた次数分布に対して符号の性能を解析するだけでなく，性能が高くなるような最適な次数分布や各次数ごとのパンクチャド比率分布を求めることも可能である．パンクチャド比率分布の各 $\pi_j^{(0)}$ は目的のパンクチャ率 $P^{(0)}$ 毎に $m_u^{(\ell)}$ を利用して線形計画法から決定でき，$m_u^{(\ell)}, \ell = 0, 1, \cdots$，が単調増加するという制約条件のもとで目的関数 $P^{(0)}$ を最大化するような各 $\pi_j^{(0)}$ を求める．

3 提案する LDPC 符号の構成法と解析法

ここでは複数の部分グラフから構成することで高い消失訂正能力を示す LDPC 符号に注目し，パンクチャド変数ノードの復元に適した構造を持ち，なおかつ符号アンサンブルとして定義できる LDPC 符号の構成法を提案する．またその符号に対しガウス近似法による理論評価を適用し，効果的なパンクチャ比率の決定法を示す．符号アンサンブルとして表現することにより，ガウス近似法による解析結果から符号の性能を理論的に評価することができる．

3.1 部分グラフから構成する LDPC 符号

提案する符号はチェックノードを共有し 2 つの部分グラフから構成され，2 つの変数ノードの集合を V_1, V_2 とする．ここで V_1, V_2 は同じ変数ノードの次数分布 $\lambda(x)$ をもつとする．次数 i のチェックノードから V_1 へは次数 1 の枝で接続され，V_2 へは次数 $i-1$ の枝で接続される．以上の議論より，V_1, V_2 の変数ノードの数の比率は $\sum_{i=2}^{d_r}\frac{\rho_i}{i} : 1-\sum_{i=2}^{d_r}\frac{\rho_i}{i}$ であることがわかる．また変数ノード V_1, V_2 をもつ部分グラフのそれぞれのチェックノードの次数分布 $\rho_1(x), \rho_2(x)$ は次式で得られる．

$$\rho_1(x)=\sum_{i=2}^{d_r}\rho_{1,i}x^{i-1},\quad \rho_2(x)=\sum_{i=2}^{d_r}\rho_{2,i}x^{i-1}. \tag{12}$$

ここで $\rho_{1,i}=\frac{\rho_i}{i}\Big/\sum_{j=2}^{d_r}\frac{\rho_j}{j}$, $\rho_{2,i}=\frac{\rho_i(i-1)}{i}\Big/\sum_{j=2}^{d_r}\frac{\rho_j(j-1)}{j}$ である．この符号を全体としてみると，通常の LDPC 符号と同じ次数分布 $\lambda(x), \rho(x)$ をもつ．

提案する符号に対しパンクチャドビットの復元を考える場合，以下の 2 つの補題が成り立つ．

補題 1 V_1 の変数ノードのみをパンクチャした場合，パンクチャド変数ノードは全て繰り返し 1 回で復元できる．

証明 符号の構造と (5) 式より明らかである． □

補題 2 V_1 の全てと V_2 の 1 部の変数ノードをパンクチャした場合，V_1 にのみパンクチャド変数ノードをもつチェックノードを用いると繰り返し 1 回でこれらのパンクチャド変数ノードは復元できる．

証明 補題 1 の議論より明らかである． □

3.2 提案する符号に対する性能解析手法

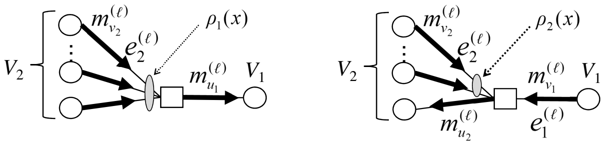
前節で述べた符号に対しガウス近似法を適用することで性能解析を行い，効果的なパンクチャ方法を示す．この場合，従来の符号アンサンブルに対するガウス近似法の反復 l 回目におけるメッセージの平均 $m_v^{(\ell)}, m_u^{(\ell)}$ とそれらのメッセージが復元されていない確率 $e^{(\ell)}, \epsilon^{(\ell)}$ を V_1 側と V_2 側の 2 種類に分けて考え，それぞれ $m_{v_1}^{(\ell)}, m_{u_1}^{(\ell)}, e_1^{(\ell)}, \epsilon_1^{(\ell)}$ と $m_{v_2}^{(\ell)}, m_{u_2}^{(\ell)}, e_2^{(\ell)}, \epsilon_2^{(\ell)}$ とする．$e_k^{(\ell)}(k=1,2), \epsilon_1^{(\ell)}, \epsilon_2^{(\ell)}$ はそれぞれ次式のように求めることができる．

$$e_k^{(\ell)}=\sum_{j=2}^{d_\ell}\lambda_j\pi_{k,j}^{(0)}\big(\epsilon_k^{(\ell-1)}\big)^{j-1}. \tag{13}$$

$$\epsilon_1^{(\ell)}=\sum_{s=2}^{d_r}\rho_{1,s}\big(1-(1-e_2^{(\ell)})^{s-1}\big). \tag{14}$$

$$\epsilon_2^{(\ell)}=\sum_{s=2}^{d_r}\rho_{2,s}\big(1-(1-e_1^{(\ell)})(1-e_2^{(\ell)})^{s-2}\big). \tag{15}$$

ここで $\pi_{1,j}^{(0)}$, $\pi_{2,j}^{(0)}$ はそれぞれ V_1, V_2 における次数 j の変数ノードのパンクチャ比率であり，$\pi_j^{(0)}=\pi_{1,j}^{(0)}+\pi_{2,j}^{(0)}$ が成り立つ．$m_{u_1}^{(\ell)}, m_{u_2}^{(\ell)}$ の算出方法は (16), (17) 式と図 4 に示す通りである．ここで，${}_{j-1}\chi_{k,i}^{(\ell)}, k=1,2$, は ${}_{j-1}\chi_{k,i}^{(\ell)}={}_{j-1}C_i(\epsilon_k^{(\ell-1)})^{j-1-i}(1-\epsilon_k^{(\ell-1)})^i$ である．なお，チェックノードの次数分布が 2 種類 $(\rho_1(x), \rho_2(x))$ 存在することに注意されたい．



(a) $m_{u_1}^{(\ell)}$ の更新　(b) $m_{u_2}^{(\ell)}$ の更新

図 4．メッセージの更新

符号の構造から，$m_{u_1}^{(\ell)}$ は V_2 の変数ノードのメッセージ ($m_{v_2}^{(\ell)}$ と $e_2^{(\ell)}$) から計算され，$m_{u_2}^{(\ell)}$ は V_1 の変数ノードのメッセージ ($m_{u_1}^{(\ell)}$ と $e_1^{(\ell)}$) と V_2 の変数ノードのメッセージ ($m_{u_2}^{(\ell)}$ と $e_2^{(\ell)}$) から計算される．

(16), (17) 式により $m_{u_1}^{(\ell)}$, $m_{u_2}^{(\ell)}$ を反復毎に計算することで，$m_u^{(\ell)}=\sum_{i=2}^{d_r}\frac{\rho_i}{i}\times m_{u_1}^{(\ell)}+(1-\sum_{i=2}^{d_r}\frac{\rho_i}{i})\times m_{u_2}^{(\ell)}$ を求めることができる．よって従来の符号アンサンブルに対するガウス近似法と同様に，閾値 σ^* を求めることができる．

各 $\pi_j^{(0)}$ は目的のパンクチャ率 $P^{(0)}$ 毎に線形計画法を解くことで決定できるが，提案する LDPC 符号では補題 1, 2 に注目し，V_1 から優先的にパンクチャしていくことを考える．そこでパンクチャ比率の決定法において以下の場合分けを行い，線形計画法を解く際の制約条件を追加する．

$$\begin{cases}0\le\pi_{1,j}^{(0)}\le\displaystyle\sum_{i=2}^{d_r}\frac{\rho_i}{i}, \pi_{2,j}^{(0)}=0, & \text{if } P^{(0)}\le\displaystyle\sum_{i=2}^{d_r}\frac{\rho_i}{i};\\ \pi_{1,j}^{(0)}=\displaystyle\sum_{i=2}^{d_r}\frac{\rho_i}{i}, \sum_{i=2}^{d_r}\frac{\rho_i}{i}\le\pi_{2,j}^{(0)}\le 1, & \text{otherwise,}\end{cases} \tag{18}$$

このとき，少ない繰り返し回数で多くのパンクチャド変数ノードが復元可能であることが示されているため，復号性能の向上が期待できる．

4 数値計算

4.1 数値計算条件

提案した LDPC 符号とパンクチャ比率の決定法の有効性を検証するために数値計算を行った．数値計算には次数分布 $\lambda(x)=0.25105x+0.30938x^2+0.00104x^3+043853x^9, \rho(x)=0.63676x^6+0.36324x^7$, 符号化率 0.5 の LDPC 符号を用いた [7]．ここで，V_1 の比率は $\sum_{i=2}^{d_r}\frac{\rho_i}{i}=0.1363$ である．また通信路には AWGN 通信路を仮定し，比較対象として従来の符号アンサンブルに対してガウス近似法を適用した結果を用いた．

4.2 数値計算結果および考察

各手法で得られた $P^{(0)}$ 毎の各 $\pi_j^{(0)}$ と σ^* を表 1, 2 に示す．

表 1．数値計算結果 (通常の LDPC 符号)

$P^{(0)}$	0	0.10	0.1363	0.20	0.30	0.40
$\pi_2^{(0)}$	0	0.1330	0.1750	0.2484	0.3719	0.4879
$\pi_3^{(0)}$	0	0.0403	0.0727	0.1286	0.2013	0.2944
$\pi_4^{(0)}$	0	0	0	0	0	0
$\pi_{10}^{(0)}$	0	0.1464	0.1762	0.2305	0.3277	0.3988
σ^*	0.9420	0.8654	0.8341	0.7756	0.6701	0.5281

$$
\begin{aligned}
m_{u_1}^{(\ell)} = & \sum_{s=2}^{d_r} \rho_{1,s} \phi^{-1} \Bigg(1 - \frac{1}{(1-e_2^{(\ell)})^{s-1}} \Big[1 - \sum_{j=2}^{d_\ell} \big\{ \lambda_j \pi_{2,j}^{(0)} \sum_{i=0}^{j-1} {}_{j-1}\chi_{2,i}^{(\ell)} \phi(i m_{u_2}^{(\ell-1)}) \\
& + \lambda_j (1-\pi_{2,j}^{(0)}) \sum_{i=0}^{j-1} {}_{j-1}\chi_{2,i}^{(\ell)} \phi(i m_{u_2}^{(\ell-1)} + m_{u_2,0}) \big\} \Big]^{s-1} \Bigg). \qquad (16) \\
m_{u_2}^{(\ell)} = & \sum_{s=2}^{d_r} \rho_{2,s} \phi^{-1} \Bigg(1 - \frac{1}{(1-e_2^{(\ell)})^{s-2}} \Big[1 - \sum_{j=2}^{d_\ell} \big\{ \lambda_j \pi_{2,j}^{(0)} \sum_{i=0}^{j-1} {}_{j-1}\chi_{2,i}^{(\ell)} \phi(i m_{u_2}^{(\ell-1)}) \\
& + \lambda_j (1-\pi_{2,j}^{(0)}) \sum_{i=0}^{j-1} {}_{j-1}\chi_{2,i}^{(\ell)} \phi(i m_{u_2}^{(\ell-1)} + m_{u_2,0}) \big\} \Big]^{s-2} \frac{1}{(1-e_1^{(\ell)})} \Big[1 - \sum_{j=2}^{d_\ell} \big\{ \lambda_j \pi_{1,j}^{(0)} \sum_{i=0}^{j-1} {}_{j-1}\chi_{1,i}^{(\ell)} \\
& \times \phi(i m_{u_1}^{(\ell-1)} + \lambda_j (1-\pi_{1,j}^{(0)}) \sum_{i=0}^{j-1} {}_{j-1}\chi_{1,i}^{(\ell)} \phi(i m_{u_1}^{(\ell-1)} + m_{u_1,0}) \big\} \Big] \Bigg). \qquad (17)
\end{aligned}
$$

表 2．数値計算結果 (提案した符号)

$P^{(0)}$	0	0.10	0.1363	0.20	0.30	0.40
$\pi_2^{(0)}$	0	0.1301	0.1363	0.2451	0.3721	0.4881
$\pi_3^{(0)}$	0	0.0480	0.1363	0.1363	0.2009	0.2939
$\pi_4^{(0)}$	0	0	0.1363	0.1363	0.1363	0.1363
$\pi_{10}^{(0)}$	0	0.1363	0.1363	0.2205	0.3276	0.3988
σ^*	0.9420	0.8697	0.8392	0.7824	0.6758	0.5366

表 1, 2 の結果から各手法で得られた σ^* を比較すると，パンクチャした全ての $P^{(0)}$ において提案手法の σ^* の値の方が従来手法よりも大きくなっていることがわかる．そのため提案手法は比較手法に比べて復号性能が優れているといえる．これは提案した LDPC 符号の構造と各 $\pi_j^{(0)}$ の決定法から，少ない繰り返し回数で全てのパンクチャド変数ノードが復元できるためだと考えられる．さらに $P^{(0)} = 0$ において同じ σ^* であることからパンクチャをしないとき，提案した符号は従来の LDPC 符号と同じ性能を示すことがわかる．また $P^{(0)}$ が大きくなるにつれて各 $\pi_j^{(0)}$ も単調に増加しており，符号化率が可変な符号を構成できることを確認した．$\pi_4^{(0)}$ が常に 0 もしくは 0.1363 と制約条件に依存した値になったのは，ρ_4 が極めて小さいためだと考えられる．

次に表 1 で得られたパンクチャ比率を提案した LDPC 符号に適用して得られる σ^* を表 3 に示す．

表 3．数値計算結果 (表 1 のパンクチャ率で提案した符号)

$P^{(0)}$	0	0.10	0.1363	0.20	0.30	0.40
σ^*	0.9420	0.8694	0.8388	0.7817	0.6758	0.5365

表 2, 3 を比較すると，全ての $P^{(0)}$ において表 2 の σ^* の方が表 3 の σ^* 以上の値が得られていることから，提案した符号に合わせて解析したパンクチャ比率の決定法の有効性が示すことができたといえる．

5　今後の課題とまとめ

本研究では，部分グラフから構成するパンクチャド変数ノードの復元に適した構造をもつ LDPC 符号の構成法とその符号アンサンブルに対するガウス近似法を用いた性能解析法を提案した．また得られた解析手法から，効果的なパンクチャ比率の決定法を提案し，数値計算結果からその有効性を示した．

今後の課題は，より高い復号性能を示す RCP-LDPC 符号の構成法を考案し，その符号アンサンブルに対しての性能解析法を示すことである．

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] T. J. Richardson and R. L. 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.

[4] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inform. Theory*, Vol. 47, No. 2, pp. 619–637, Feb. 2001.

[5] S. -Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inform. Theory*, Vol. 47, No. 2, pp. 657–670, Feb. 2001.

[6] S. Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, Vol. 49, No. 10, pp. 1727–1737, Oct. 2001.

[7] J. Ha, J. Kim, and S. W. McLaughlin, “Rate-compatible puncturing of low-density parity-check codes,” *IEEE Trans. Inform. Theory*, Vol. IT-50, No. 11, pp. 2824–2836, Nov. 2004.

[8] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, “Rate-compatible punctured low-density parity-check codes with short block lengths,” *IEEE Trans. Inform. Theory*, Vol. 52, No. 2, pp. 728–738, Feb. 2006.

[9] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*, MIT Press, 2008.

[10] G. Hosoya, T. Matsushima, and S. Hirasawa, “A combined matrix ensemble of low-density parity-check codes for correcting a solid burst erasure,” *IEICE Trans. Fundamentals*, Vol. E91-A, No. 10, pp. 2765–2778, Oct. 2008.

[11] J. Kim, A. Ramamoorthy, and S. W. McLaughlin, “The design of efficiently-encodable rate-compatible LDPC codes,” *IEEE Trans. Commun.*, Vol. 57, No. 2, pp. 365–375, Feb. 2009.

[12] M. El-Khamy, J. Hou, and N. Bhushan, “Design of rate-compatible structured LDPC codes for hybrid ARQ applications,” *IEEE Journal on selected areas in Commun.*, Vol. 27, No. 6, pp. 965–973, Aug. 2009.

[13] 長田佳史, 寺本賢一, 細谷剛, 後藤正幸, “高符号化率までパンクチャ可能な LDPC 符号に関する一考察,” 電子情報通信学会技術研究報告 *IT*, Vol. 110, No. 137, pp. 95–100, July 2010.

[14] 長田佳史, 細谷剛, 後藤正幸, “検査行列の構造を利用した LDPC 符号のパンクチャ法に関する一考察,” 第 10 回情報科学技術フォーラム, Vol. 1, pp. 141–146, Sep. 2011.