■1 群（信号・システム）- 2 編（符号理論）

6 章 ターボ符号・LDPC 符号

（執筆者：井坂元彦）［2012 年 3 月 受領］

■概要■

1993 年に提案されたターボ符号と，1960 年代の発明から 30 年以上を経て再発見された低密度パリティ検査（LDPC）符号は，符号理論研究に大きな変革をもたらし，現在では実用化・標準化されている．これらの符号に共通する事項として，簡単な符号を組み合わせた構成をとること，符号長が大きい場合でも実装可能な程度の計算量で復号が可能であること，また適切に長い符号を設計することで通信路容量（シャノン限界）に迫る特性を達成することが挙げられる．

ターボ符号は，帰還結線を有する畳込み符号化器を 2 個組み合わせたものであるが，これらの畳込み符号はターボ符号の要素符号と呼ばれる．一方，LDPC 符号は非零成分の密度が低い疎な検査行列をもつ線形符号であるが，これは単一パリティ検査符号と繰り返し符号を要素符号として構成されているととらえられる．これまでの章で述べられたとおり，上記の要素符号に対しては低複雑度の復号法が存在する．

ターボ符号や LDPC 符号は符号長が大きい場合でもサム・プロダクトアルゴリズムと総称される手法により効率的に復号することが可能である．これは受信語が与えられたもとで符号語の各シンボルに対して事後確率の周辺分布を計算することを目的としたものである．復号の操作として，各要素符号に対してシンボル単位の復号を行い，そこで得られる値をほかの要素符号の復号器で各シンボルの事前確率として用いることが繰り返される．この復号法を用いるとき，一般に周辺分布が正確に計算されるわけではないが，復号誤り確率に関しては符号長が大きければ理論的限界に迫る性能が達成される．

本章では，ターボ符号と LDPC 符号の構成及び復号法について述べる．

【本章の構成】

本章は，ターボ符号（6-1 節），LDPC 符号（6-2 節），サム・プロダクト復号（6-3 節）からなる．

■1 群 - 2 編 - 6 章

6-1 ターボ符号

（執筆者：荻原春生）[2012 年 3 月 受領]

ターボ符号（turbo code）は 1993 年にベロー（Berrou）らにより提案[1]された．それ以前には実用上十分低いといえる誤り率を実現するには，理論限界に比べ，少なくとも 2[dB] 良好な信号対雑音電力比（SNR）が要求された．ターボ符号は，現実的な復号処理量で，これを 0.5[dB] まで近づけた[1]．

ターボ符号の符号器を図 6･1 に示す．符号器には 2 個の畳込み符号器（それぞれを要素符号器と呼ぶ）が含まれている．情報系列は要素符号器 1 によりパリティビット系列 1 を生成する．また，情報系列は，ある単位（これをインタリーバサイズという）に区切られ，その内部で系列の順番を入れ替える回路であるインタリーバ（interleaver）を経由した後，要素符号器 2 によりパリティビット系列 2 を生成する．2 系列のパリティビット系列は間引きされた後に多重化（間引き多重化）され，更に情報系列と多重化され送信される．これにより生成される符号を並列連接畳込み符号（parallel concatenated convolutional code）という．要素符号器としては，再帰形の畳込み符号器が使われる．

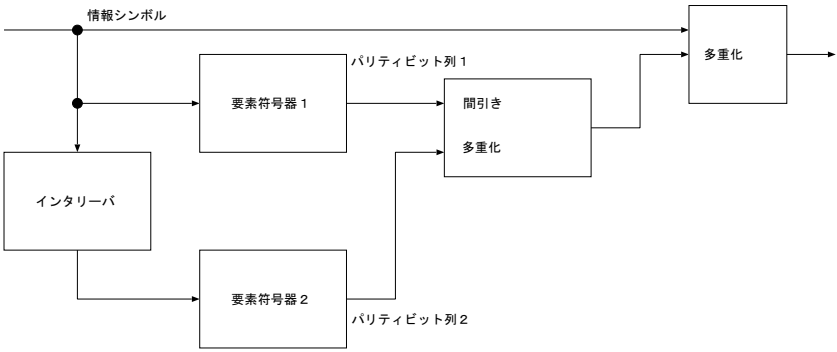

図 6･1　ターボ符号の符号器

図 6･2 に復号器の構成を示す．要素復号器 1 は，通信路からの受信信号から計算される通信路値（channel value）と要素復号器 2 から伝えられる事前値（a priori value）L_{a1} を用い（繰り返し 1 回目では，事前値は 0 である）本章 5-3 節で説明された BCJR アルゴリズム（BCJR algorithm）あるいは max log MAP アルゴリズム（max log MAP algorithm）により外部値（extrinsic value）L_{e1} を求める．それは，インタリーバで並べ替えられ，要素復号器 2 の事前値 L_{a2} として入力される．要素復号器 2 は要素復号器 1 と同様の動作により外部値 L_{e2} を求める．それは，デインタリーバを介して，要素復号器 2 の事前値となる．これを 10 回程度繰り返した後，事後値（a posteriori value）を求め，これを硬判定することにより復号結果を得る．

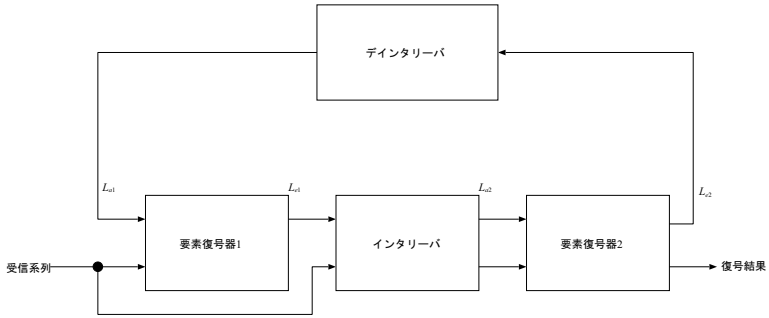

図 6･2　ターボ符号の復号器

図 6･3 は，ベローらの原論文の構成にしたがってシミュレーションを行った結果を示す．繰り返しとともに特性が改善されることが分かる．インタリーバサイズは 65536 で，単純な規則性がないように並べ替えを行っている．符号化率は 1/2，通信路は白色ガウス雑音通信路である．この符号化率で十分小さな誤り率を与える情報理論の限界（シャノン限界）は 0.2dB であり，これに 0.5dB まで近づいている．

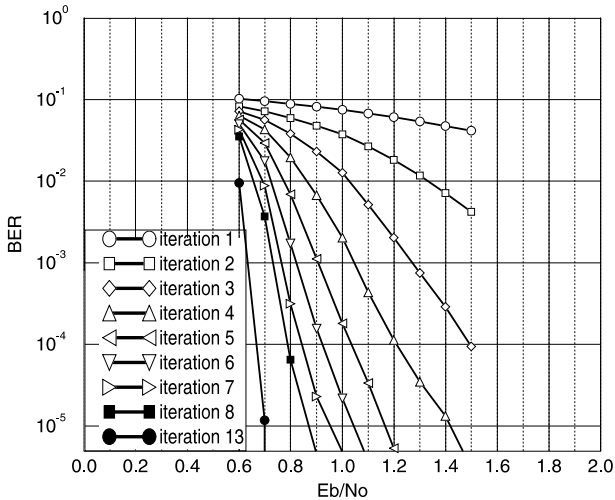

図 6･3　並列連接符号の特性：インタリーバサイズ=65536. iteration **は繰り返し回数を示す**

図 6･4 は，ベネデトー（Benedetto）らにより提案[2)]された直列連接畳込み符号（serially concatenated vonvolutional codes）の符号器構成を示す．情報系列 $\{u_k\}$（k は時点を示す）は，外部符号器（outer encoder）（外符号器とも呼ばれる）に入力され，第 1 のパリティビット系列 $\{p_k\}$ を生成する．$\{u_k\}$ と $\{p_k\}$ はインタリーブされ $\{u'_k\}$ と $\{p'_k\}$ となり，内部符号器（inner

encoder）（内符号器とも呼ばれる）に入力され第 2 のパリティビット系列 $\{q_k\}$ がつくられ，$\{u'_k\}$ と $\{p'_k\}$ とともに通信路へ送出される．受信器は，内部復号器（inner decoder）（内復号器とも呼ばれる）で，通信路値（channel value）と，情報ビットと第 1 のパリティビットの事前値（a priori value）に基づき，それらのビットの外部値（extrinsic value）を計算する．それらは，デインタリーブされ，外部復号器（outer decoder）（外復号器とも呼ばれる）の事前値として渡される．外復号器もこれらの値の外部値を計算し，インタリーブして，内復号器の事前値として渡す．これを 10 回程度繰り返した後，外復号器で，情報ビットの事後値（a posteriori value）を計算し，それを硬判定し，復号結果とする．

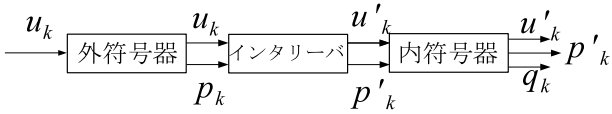

図 6･4　直列連接符号器

図 6･5 は，拘束長 3 の (7,5) 符号（数字は生成多項式 $(1 + D + D^2, 1 + D^2)$ の係数の 8 進数表示、以下同様），拘束長 6 の (45,73) 符号を用いた並列連接符号，拘束長 3 の (7,5) 符号を用いた直列連接符号で，インタリーバサイズ 1000 の場合の特性を示す．低 SNR では (7,5) 符号が，高 SNR では (45,73) 符号が良い特性を示している．直列連接符号は，高 SNR で良い特性を示している．いずれの場合もインタリーバサイズを大とすると特性は改善する[3]．並列連接/直列連接，拘束長への同様の依存性は一般に存在する．

ターボ符号は 7 章 7-4 節で説明するように，第 3 世代携帯電話でデータ通信時に使われている．

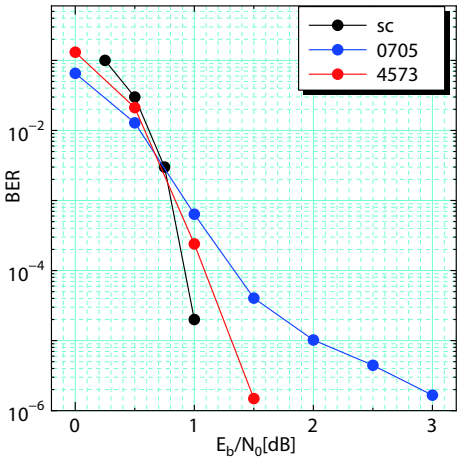

図 6･5　並列連接 (7,5) 符号,(45,73) 符号, 直列連接 (7,5) 符号 (SC) の特性．インタリーバサイズ=1000

■1 群 - 2 編 - 6 章

6-2 LDPC 符号

（執筆者：渋谷智治）[2012 年 3 月 受領]

6-2-1 はじめに

LDPC（Low-Density Parity Check，低密度パリティ検査）符号とは，疎な検査行列により定義される線形符号の一クラスである．サム・プロダクト（sum-product）アルゴリズムに基づく反復復号法（iterative decoding）（サム・プロダクト復号法）と組み合わせることより，シャノン（Shannon）限界に迫る高い復号性能を達成する．

LDPC 符号の研究は 1960 年代初頭のギャラガー[4]（Gallager）による博士論文にまで遡る．しかしながら，それに先立つ 1950 年代から 1960 年代にかけては，現在でも実用上重要な BCH 符号や RS 符号が開発され，その後の代数的符号理論の隆盛の基礎が築かれた時期でもあった．このため，LDPC 符号が再び注目を集めるには，1990 年代初頭のターボ符号（turbo code）[1]の大成功と，それを契機とする，確率推論に基づく誤り訂正符号の見直しの気運を待たなければならなかった．その後は，マッカイ[5]（MacKay）による LDPC 符号の再発見を通じて，確率推論（probabilistic inference）[6]や機械学習（machine learning）[7]，統計力学（statistical mechanics）[8] といった応用数理の諸分野の独自の進展と LDPC 符号の研究とが有機的に結びつき，大きな研究領域[9]を形成して今日に至っている．

また，近年では実用化に関する研究開発も長足の進歩を遂げ，イーサネット[10]や衛星通信[11]などにおける誤り訂正方式として標準化されるなど，LDPC 符号はターボ符号と並んで今後の誤り訂正符号の主役として期待されている．

なお，本項では GF(2) 上の 2 元 LDPC 符号のみを考える．

6-2-2 LDPC 符号とその表現

GF(2) 上の $m \times n$ 行列 $H = [h_{ij}]$ を検査行列とする線形符号 C を考える．H が疎行列（sparse matrix）*のとき C を LDPC 符号と呼ぶ．特に H の各行及び各列の重みがそれぞれ一定値であるとき，C を正則 LDPC 符号（regular LDPC code）と呼ぶ．正則 LDPC 符号以外の LDPC 符号を非正則 LDPC 符号（irregular LDPC code）と呼ぶ．

LDPC 符号の構成や復号法，復号性能などについて検討する際には，検査行列 H から一意に定まる二部グラフ（bipartite graph）を考えることがしばしば有用である．H の各行及び各列に対応する節点をそれぞれ p_i $(i = 1, 2, \ldots, m)$, c_j $(j = 1, 2, \ldots, n)$ で表し，二つの節点集合 $V_c \triangleq \{p_1, p_2, \ldots, p_m\}$, $V_b \triangleq \{c_1, c_2, \ldots, c_n\}$ を考える．更に $V \triangleq V_c \cup V_b$ を節点集合，$E \triangleq \{(p_i, c_j) \in V_c \times V_b \mid h_{ij} = 1\}$ を枝集合とする二部グラフ $\Gamma = (V, E)$ を考える．Γ を線形符号 C の H に関するタナー（Tanner）グラフと呼ぶ．また，V_c, V_b の節点 p_i, c_j をそれぞれ検査ノード（check node），ビットノード（bit node）と呼ぶ．図 6･6 に，符号長 6 のある線形符号の検査行列の例と，それに関するタナーグラフを示す．なお，図 6･6 では検査ノードを□，ビットノードを○で表している．

* LDPC 符号の場合，検査行列 H における非零要素の個数が符号長の定数倍程度であるとき，H を疎行列とみなすことが多い．

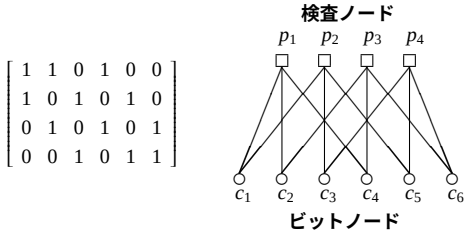

図 6･6　検査行列とタナーグラフ

以下の説明では，タナーグラフにおいて検査ノード p_i に隣接するビットノードの添え字集合を A_i，また，ビットノード c_j に隣接する検査ノードの添え字集合を B_j で表すものとする．すなわち，$A_i = \{j \mid h_{ij} = 1\}$, $B_j = \{i \mid h_{ij} = 1\}$ である．

6-2-3 メッセージ・パッシング復号法

LDPC 符号の復号法には様々なバリエーションが存在するが，それらの多くは，タナーグラフ上の隣接するノード間でメッセージ（message）と呼ばれる数値を交換しながら送信ビットや送信符号語を推定するメッセージ・パッシング（message passing, MP）復号法として解釈できる．

MP 復号法では，検査ノード p_i $(i = 1, 2, \ldots, m)$ からそれに隣接するビットノード c_j $(j \in A_i)$ に伝達されるメッセージ $m_{ij}(\alpha)$ $(\alpha \in \mathrm{GF}(2))$，及び，ビットノード c_j $(j = 1, 2, \ldots, n)$ からそれに隣接する検査ノード p_i $(i \in B_j)$ に伝達されるメッセージ $m_{ji}(\alpha)$ $(\alpha \in \mathrm{GF}(2))$ の 2 通りのメッセージを考える．また，各メッセージは，メッセージの伝達先以外の隣接するノードからもたらされたメッセージを用いて計算される．すなわち，$m_{ij}(\alpha)$ は $m_{ki}(\alpha_k)$ $(k \in A_i \setminus \{j\}, \alpha_k \in \mathrm{GF}(2))$ から，また，$m_{ji}(\alpha)$ は $m_{\ell j}(\alpha)$ $(\ell \in B_j \setminus \{i\})$ から計算される．タナーグラフ上におけるメッセージの伝達方向と，メッセージの計算におけるメッセージ間の依存関係を図 6･7 に示す．

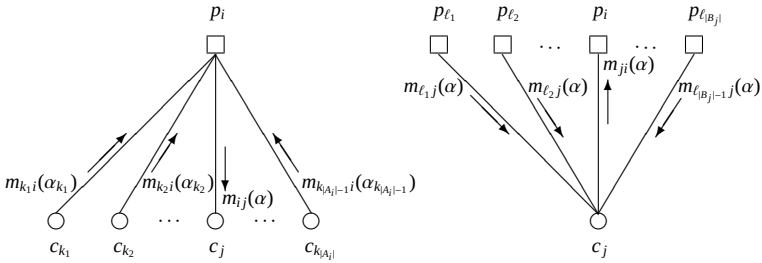

図 6･7　メッセージの計算におけるメッセージ間の依存関係

MP 復号法では，上で述べたメッセージ・パッシングに基づいてメッセージ $m_{ij}(\alpha)$ の更新と $m_{ji}(\alpha)$ の更新を交互に繰り返し，c_j の尤度とメッセージ $\{m_{ij}(\alpha) \mid i \in B_j\}$ に基づいて各送信ビットの推定値 $\hat{c}_j$ を決定する．なお，送信符号語の推定 $\hat{\boldsymbol{c}} = (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n)$ が $H\hat{\boldsymbol{c}}^T = \boldsymbol{0}$ を満たした（$\hat{\boldsymbol{c}}$ が符号語となった）時点で，メッセージの更新を止めて $\hat{\boldsymbol{c}}$ を復号結果として出力することにより，復号性能をそれほど悪化させることなくメッセージの更新回数を抑制す

ることができる．

なお，メッセージの具体的な更新式や送信ビットの推定値の決定方法，MP 復号法の詳細な手順などについては，本章 6-3 節を参照のこと．

6-2-4 LDPC 符号の性能評価

LDPC 符号を MP 復号法により復号した際の性能を評価する方法は大きく分けて二つある．一つは，計算機シミュレーションによって個々の LDPC 符号の復号誤り率を直接求める方法である．そしてもう一つは，符号長を無限大にしたときの漸近的な性能を，LDPC 符号の符号アンサンブル（ensemble of codes）に対する平均値として求める方法である．密度発展法（density evolution）[12] [本章 6-3 節 参照] や EXIT チャート（EXtrinsic Information Transfer Chart）法[13] は後者に属する代表的な手法であり，符号アンサンブルに対する MP 復号法の平均的な復号性能を，メッセージの更新式や符号アンサンブルのパラメータから解析的あるいは数値的に求める方法である．

多くの MP 復号法では，通信路を定めるパラメータ（加法的白色ガウス通信路における S/N 比や 2 元対称通信路における誤り確率など）のある値を境にして，復号誤り率が急激に変化する現象が観測される．この値は反復閾値（threshold）などと呼ばれ，LDPC 符号を設計するうえで重要なパラメータの一つとなっている．密度発展法や EXIT チャート法を用いることによって，符号アンサンブルに対する反復閾値を効率的に評価することができる．

6-2-5 LDPC 符号の構成

本章 6-2-2 項で述べた，検査行列から 2 部グラフへの対応を逆に用いると，与えられた 2 部グラフをタナーグラフとする検査行列が得られる．そこで，2 部グラフにおける枝と節点との接続に関する条件（代表的なものに次数分布（degree distribution）による条件[12]などがある）を定め，これを満たす 2 部グラフのアンサンブルを構成すると，2 部グラフから得られる検査行列を通じて線形符号のアンサンブルを構成することができる．

密度発展法によると，与えられた次数分布に対して定まる符号アンサンブルに対して，反復閾値を比較的容易に計算できる．そこで，次数分布を適切に調整することによって，反復閾値の意味で優れた LDPC 符号，特に非正則 LDPC 符号[14]のアンサンブルを構成することができる．

ここで，符号長が十分大きな場合には，符号アンサンブルにおける個々の符号の MP 復号法に対する性能とそのアンサンブル平均とが大きく乖離する確率は非常に小さくなるため，密度発展法による符号アンサンブルの設計は具体的な符号の構成にも有効である．一方，実用的な符号長では，符号アンサンブルの平均的な性能と個々の符号の性能との乖離が大きくなるため，符号の構成に一層の工夫を凝らす必要がある．

タナーグラフに短いループ，特に長さ 4 のループが多数含まれるとき，MP 復号法の性能劣化が著しいことが知られている．そこで，長さ 4 のループを含まないタナーグラフを具体的に構成することによって MP 復号法の性能劣化を抑えた個々の LDPC 符号を構成する方法が数多く提案されている．代表的なものに，有限幾何・射影幾何に基づく符号[15] 組合せデザイン（combinatorial design）に基づく符号[16], Ramanujan グラフ（Ramanujan graph）に基づく符号[17]，擬巡回符号（quasi-cyclic code）や配列（array）符号の応用による符号[18] などが

ある．なお，これらの符号のなかにはエラーフロア (error floor) 現象が発生しやすいものもあるが[19]，エラーフロア現象の抑制にも効果的な手法として PEG (ProgressiveEdge-Growth)[20] と呼ばれるタナーグラフの構成方法も提案されている．

代数的に構成された LDPC 符号の符号化が比較的容易であるのに対し，ランダムに構成された LDPC 符号の符号化は一般に多くの計算量を必要とする．これに対し，検査行列が疎行列であることを利用して符号化の計算量を削減する手法が提案されている[21]．

■1 群 - 2 編 - 6 章

6-3 サム・プロダクト復号

（執筆者：渋谷智治）［2012 年 3 月 受領］

6-3-1 線形符号の復号における周辺分布の計算問題

GF(2) 上の $m \times n$ 行列 $H = [h_{ij}]$ を検査行列とする 2 元線形符号 C を考える．H に対して，集合 A_i $(i = 1, 2, \ldots, m)$, B_j $(j = 1, 2, \ldots, n)$ をそれぞれ $A_i \triangleq \{j \mid h_{ij} \neq 0\}$, $B_j \triangleq \{i \mid h_{ij} \neq 0\}$ で定めると，受信語 $\boldsymbol{r}$ を受け取ったときの送信符号語 $\boldsymbol{c} \in C$ の事後確率（a posteriori probability）は

$$P(\boldsymbol{c}|\boldsymbol{r}) = \kappa \prod_{i=1}^{m} \delta\Big(\sum_{j' \in A_i} c_{j'}, 0 \Big) \prod_{j=1}^{n} P(r_j|c_j) \tag{6·1}$$

で表される．ただし，送信符号語は一様に生起すると仮定する．また，通信路は定常無記憶であると仮定し，通信路の遷移確率を $P(r|c)$ で表すものとする．更に，κ は $\sum_{\boldsymbol{c} \in \mathrm{GF}(2)^n} P(\boldsymbol{c}|\boldsymbol{r}) = 1$ となるための正規化の定数であり，$\delta(a, b)$ は $a = b$ のとき 1，$a \neq b$ のとき 0 となる関数である．

ここで，事後確率 $P(\boldsymbol{c}|\boldsymbol{r})$ の送信ビット c_j に関する周辺分布（marginal distribution）：

$$P(c_j = \alpha|\boldsymbol{r}) \triangleq \sum_{\boldsymbol{c} \in \mathrm{GF}(2)^n,\, c_j = \alpha} P(\boldsymbol{c}|\boldsymbol{r}) \qquad (\alpha \in \mathrm{GF}(2)) \tag{6·2}$$

に対し，c_j の推定値 $\hat{c}_j$ を，$P(c_j = 0|\boldsymbol{r}) \geq P(c_j = 1|\boldsymbol{r})$ のとき $\hat{c}_j = 0$, $P(c_j = 0|\boldsymbol{r}) < P(c_j = 1|\boldsymbol{r})$ のとき $\hat{c}_j = 1$ で定める復号を考える．この復号は $\hat{c}_j \neq c_j$ となる確率を最小にするという意味で最適であるが，式 (6·2) の計算には一般に符号長 n の指数関数に比例する計算量を要する．サム・プロダクト（sum-product）復号法とは，周辺分布 $P(c_j = \alpha|\boldsymbol{r})$ をサム・プロダクトアルゴリズム（sum-product algorithm）により効率的に評価し，送信ビット c_j を推定する復号法である．

6-3-2 サム・プロダクトアルゴリズム

(1) サム・プロダクトアルゴリズム

図 6·8 に，式 (6·1) の事後確率に対する周辺分布 $P(c_j = \alpha|\boldsymbol{r})$ $(\alpha \in \mathrm{GF}(2))$ を評価するサム・プロダクトアルゴリズムを示す．

(2) ファクターグラフ

式 (6·1) の右辺を関数 $\delta_i \triangleq \delta(\sum_{j' \in A_i} c_{j'}, 0)$ $(i = 1, 2, \ldots, m)$ 及び $P_j \triangleq P(r_j|c_j)$ $(j = 1, 2, \ldots, n)$ による $P(\boldsymbol{c}|\boldsymbol{r})$ の因子分解とみなす．更に，変数 $c_1, c_2, \ldots, c_n$，及び関数 $\delta_1, \delta_2, \ldots, \delta_m$ と $P_1, P_2, \ldots, P_n$ とをラベルとする節点を考え，変数 c_j がある関数の変数であるとき，かつそのときに限り，変数を表す節点と関数を表す節点との間に枝を結んでできるグラフを考える．このグラフを $P(\boldsymbol{c}|\boldsymbol{r})$ のファクターグラフ（factor graph）という．図 6·9 に式 (6·1) のファクターグラフを示す．なお，図 6·9 では，変数を表す節点を ○，関数を表す節点を □ で表している．また，節点 c_j は節点 δ_i $(i \in B_j)$ 及び P_j と，節点 δ_i は節点 c_j $(j \in A_i)$ と隣接している．

ステップ 1　$h_{ij} = 1$ を満たす i, j に対して $m_{ij}(\alpha) = 1$ $(\alpha \in \mathrm{GF}(2))$ とする．

ステップ 2　$h_{ij} = 1$ を満たす i, j に対して $m_{ji}(\alpha)$ を次式で更新する．

$$m_{ji}(\alpha) \leftarrow P(r_j|\alpha) \prod_{i' \in B_j \setminus \{i\}} m_{i'j}(\alpha) \qquad (\alpha \in \mathrm{GF}(2)). \tag{6･3}$$

ステップ 3　$h_{ij} = 1$ を満たす i, j に対して $m_{ij}(\alpha)$ を次式で更新する．

$$m_{ij}(\alpha) \leftarrow \sum_{\substack{\alpha_{j'} \in \mathrm{GF}(2) : j' \in A_i \setminus \{j\} \\ \text{s.t } \sum_{j' \in A_i \setminus \{j\}} \alpha_{j'} \in \alpha}} \prod_{j' \in A_i \setminus \{j\}} m_{j'i}(\alpha_{j'}) \qquad (\alpha = \mathrm{GF}(2)). \tag{6･4}$$

ステップ 4　すべての $m_{ij}(\alpha)$ が収束するまでステップ 2– 4 を繰り返す．収束したら $Q_j(\alpha)$ $(j = 1, 2, \ldots, n)$ を次式で求める．ただし，κ_j は $Q_j(0) + Q_j(1) = 1$ となるように定める．

$$Q_j(\alpha) \leftarrow \kappa_j P(r_j|\alpha) \prod_{i' \in B_j} m_{i'j}(\alpha) \qquad (\alpha \in \mathrm{GF}(2)).$$

図 6･8　サム・プロダクトアルゴリズム

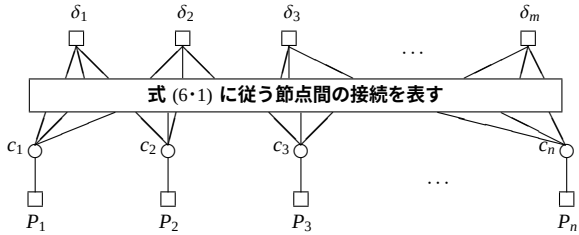

図 6･9　$P(\boldsymbol{c}|\boldsymbol{r})$ のファクターグラフ

ファクターグラフが木である（ループが存在しない）とき，サム・プロダクトアルゴリズムは収束し，更にアルゴリズムの出力 $Q_j(\alpha)$ は真の周辺分布 $P(c_j = \alpha|\boldsymbol{r})$ を与えることが知られている．一方，ファクターグラフが木でないときにはサム・プロダクトアルゴリズムの収束は保証されず，また，例え収束しても $Q_j(\alpha)$ が真の周辺分布を与えるとは限らない[6, 7]．

6-3-3 サム・プロダクト復号法

(1) サム・プロダクト復号法

サム・プロダクトアルゴリズムでは，反復の終了に収束判定が必要であるが，これにはやや複雑な計算を要する．そこで実際の復号では，まずステップ 4 において $m_{ij}(\alpha)$ が収束したか否かにかかわらず $Q_j(\alpha)$ を求めるものとする．更に，$Q_j(0) \geq Q_j(1)$ ならば $\hat{c}_j = 0$, $Q_j(0) < Q_j(1)$ ならば $\hat{c}_j = 1$ として，推定ビットの系列 $\hat{\boldsymbol{c}} \triangleq (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n)$ を定め，$H\hat{\boldsymbol{c}}^T = \boldsymbol{0}$，すなわち，$\hat{\boldsymbol{c}}$ が符号語となった時点でステップ 2– 4 の反復を停止する．これにより，復号性能をそれほど劣化させることなく反復回数を抑制できる．また，サム・プロダクトアルゴリズムの最大反復回数を制限することによって，一定の反復にもかかわらず $\hat{\boldsymbol{c}}$ が符号語とならない場合でもアルゴリズムを停止する．このようにして得られる反復復号法をサム・プロダクト復号法と

いう．

また，$m_{ij}(\alpha)$, $m_{ji}(\alpha)$ を式 (6･3), (6･4) により更新するサム・プロダクト復号法を確率領域サム・プロダクト復号法（sum-product decoding in probabilistic domain）と呼ぶのに対し，これらの対数比 $\log \frac{m_{ij}(0)}{m_{ij}(1)}$ 及び $\log \frac{m_{ji}(0)}{m_{ji}(1)}$ に対する更新式を用いたサム・プロダクト復号法は対数領域サム・プロダクト復号法（sum-product decoding in logarithm domain）と呼ばれる[22]．

(2) マックス・プロダクト復号

確率領域サム・プロダクト復号法における $m_{ij}(\alpha)$ の更新式 (6･3) に現れる和の計算は，サム・プロダクト復号法の実現に必要な計算量の大きな部分を占める．この和の計算を最大値の探索に置き換えた復号法をマックス・プロダクト（max-product）復号法と呼ぶ．ファクターグラフが木である場合には，マックス・プロダクト復号法による復号結果は式 (6･1) を最大にする符号語に一致する．また，畳込み符号に対するマックス・プロダクト復号法はビタビ復号法（Viterbi decoding）として知られている．

6-3-4 メッセージ・パッシング復号法

$m_{ij}(\alpha)$, $m_{ji}(\alpha)$ をファクターグラフ上の節点から節点へ送られるメッセージとみなし，式 (6･3), (6･4) に代わるメッセージの更新式を様々に与えることによって得られる反復復号法を総称してメッセージ・パッシング復号法と呼ぶ [本章 6-2 節 参照]．サム・プロダクト復号法やマックスプロダクト復号法のほかにも，これらの簡単化や通信路に適した修正などによって得られる以下のような MP 復号法が知られている．

(1) ビット・フリッピング（bit flipping）復号法

各回の反復において，満たされていないパリティ検査式に関与する信頼度の低いビットの推定値を反転することによって，最終的にすべてのパリティ検査式が満たされることを目指す MP 復号をビット・フリッピング復号法（bit-filliping decoding）と呼ぶ．反転するビットの個数や反転の条件などに関して様々なバリエーションが存在する． 復号性能はサム・プロダクト復号には遠く及ばないが，実装が簡単で高速な処理が可能なことから，実用上は重要な復号法である．

(2) ピーリング（peeling）復号法

サム・プロダクト復号法の更新式を 2 元消失通信路に対して書き直すと，ピーリング復号法（peeling decoding）と呼ばれる MP 復号法が得られる．これは，線形連立方程式において，未知変数が一つである方程式から順に解を決定することによってすべての解を得る逐次解法に等しい．したがって，復号の途中で未知変数が二つ以上の方程式のみが残った場合，それ以上復号を進めることができなくなる．このときの未知変数の組は停止集合（stopping set）と呼ばれ，停止集合の要素数の最小値は二元消失通信路における LDPC 符号の性能を評価するうえで重要なパラメータとされる．

6-3-5 密度発展法

MP 復号におけるメッセージ $m_{ij}(\alpha)$, $m_{ji}(\alpha)$ を確率変数としたとき，メッセージの更新に伴ってその密度関数も変化する．密度発展法（density evolution）[12] では，ファクターグラフが木であるという仮定のもとで，密度関数の変化をメッセージの更新回数 ℓ に関する漸化式により記述する．更に，通信路パラメータ s により定まるメッセージの初期分布 $P(s)$ とこ

の漸化式から，ℓ 回目の更新後のメッセージの密度関数を計算する手法を与える．その結果，MP 復号法の復号誤り率のアンサンブル平均が s と ℓ の関数 $\overline{P}_e(s,\ell)$ として得られる．

ここで $\lim_{\ell\to\infty}\overline{P}(s,\ell)$ を考えると，反復回数を十分大きくした際に MP 符号法の復号誤り率が 0 となる通信路パラメータ s（反復閾値（threshold））を定めることができる．多くの MP 復号法では，この値を境にして復号誤り率が急激に変化する現象が観測されるため，反復閾値は LDPC 符号を設計するうえで重要なパラメータの一つとなっている．また，ビット・フリッピング復号法やサム・プロダクト復号法などの MP 復号法では，反復閾値の導出は比較的容易である[4, 12]．

■参考文献

1) C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” Proc. IEEE Int. Commun. Conf., pp.1064–1070, 1993.
2) S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding,” IEEE Trans. Inf. Theory, vol.44, no.3, pp.909–926, 1998.
3) S. Benedetto and G. Montorsi, “Unveiling turbo codes: some results on parallel concatenated coding schemes,” IEEE Trans. Inf. Theory, vol.42, no.2, pp.409–428, 1996.
4) R.G. Gallager, “Low density parity check code,” Research Monograph series, Cambridge, MIT Press, 1963
5) D.J.C. MacKay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inf. Theory, vol.45, no.2, pp.399–431, Mar. 1999.
6) J. Pearl: “Probabilistic inference and expert systems”, Morgan Kaufmann (1988)
7) B.J. Frey, “Graphical models for machine learning and digital communication,” MIT Press, 1998.
8) 西森秀稔，“スピングラス理論と情報統計力学,” 新物理学選書, 岩波書店，1999.
9) 文部科学省 科学研究費補助金・特定領域研究，“ 情報統計力学の深化と展開（DEX-SMI）,” 領域代表者:樺島祥介（東京工業大学）, 研究期間: 平成 18 年度～平成 21 年度.
10) 802.3an-2006, “IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications,” 2006.
11) ETSI EN 302 307 V1.1.2 (2006-06), “ European Standard (Telecommunications series) Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services,” News Gathering and other broadband satellite applications, 2006.
12) T. Richardson and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” IEEE Trans. Inf. Theory, vol.47, no.2, pp.619–637, Feb. 2001.
13) S. ten Brink, “Convergence behaviour of iterative decoded parallel concatenated codes,” IEEE Trans. Commun., vol.49, no.10, pp.1727–1737, Oct. 2001.
14) M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, “Improved low-density parity-check codes using irregular graphs,” IEEE Trans. Inf. Theory, vol.47, no.2, pp.585–598, Feb. 2001.
15) Y. Kou, S. Lin, and M.P.C. Fossorier, “Low-density parity-check codes based on finite geometries: a rediscovery and new results,” IEEE Trans. Inf. Theory, vol.47, no.7, pp.2711–2736, Nov. 2001.
16) S.J. Johnson and S.T. Weller, “Regular low-density parity-check codes from combinatorial designs,” Proc. of Inf. Theory Workshop 2001, pp.90–92, Sep. 2001.

17) J. Rosenthal and P.O. Vontobel, “Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis,” Prof. of 38th Allerton Conf. on Commun., Control and Computing, pp.248–257, Oct. 2000.
18) J.L. Fan, “Constrained coding and soft iterative decoding,” Springer, 2001.
19) D.J.C. MacKay and M.C. Davey, “Evaluation of Gallager codes for short block length and high rate applications,” Codes, Systems, and Graphical Models, Springer-Verlag, New York, pp.113–130, 2001.
20) X.Y. Hu, E. Eleftheriou, and D.M. Arnold, “Regular and irregular progressive edge-growth Tanner graphs,” IEEE Trans. Inf. Theory, vol.51, no.7, pp.386–398, Jan. 2005.
21) T.J. Richardson and R.L. Urbanke, “Efficient encoding of low-density parity-check codes,” IEEE Trans. Inf. Theory, vol.47, no.2, pp.638–656, Feb. 2001.
22) 和田山正，“低密度パリティ検査符号とその復号法,” トリケップス，2002.