■1 群（信号・システム）- 2 編（符号理論）

5 章 トレリス符号

（執筆者：和田山正）［2012 年 3 月 受領］

■概要■

本章では，畳込み符号や符号化変調を含む重要な符号のクラスであるトレリス符号を紹介する．トレリス符号では，トレリスと呼ばれるラベル付きグラフの始点から終点に至るパスが符号語系列に対応する．例えば，畳込み符号では，シフトレジスタからなる符号化器の状態遷移図を時間方向に展開して得られるグラフがトレリスとなる．このクラスの符号の重要な特徴として，符号化器の状態数がそれほど大きくない場合において、トレリスの特性を利用したビタビ復号法による最尤復号が可能な点が挙げられる．ビタビ復号は，トレリス上で重み最小のパスを見つける動的計画法的アルゴリズムと見ることができる．ビタビ復号は軟判定復号とも相性が良く，例えば加法的白色ガウス通信路においては，復号計算量に比べて，比較的大きな符号化利得が得られることが知られている。そのため畳込み符号とビタビ復号の組合せは，無線 LAN や携帯電話を含む無線通信システムの信頼性向上のために広く利用されている．また近年では，畳込み符号はターボ符号の要素符号として利用され，通信路容量への接近を目指す最近の通信系においても，その活躍の場を広げている．

【本章の構成】

5-1 節では，畳込み符号の定義，符号化器，そして対応する状態遷移図とトレリス図の説明を行い，更に畳込み符号の最尤復号性能を支配するパラメータである最小自由距離の定義を与える．また，ターボ符号の要素符号として重要な再帰的組織畳込み符号について説明する．5-2 節は，ビタビ復号の詳細を例を用いて解説する．5-3 節では，ターボ復号において中心的役割を果たす BCJR アルゴリズムについて，そのアルゴリズムの詳細を述べる．BCJR アルゴリズムは，トレリス上で実行されるメッセージパッシング型アルゴリズムと見ることができる．5-4 節では，ビタビ復号とは全く異なる考え方に基づく畳込み符号の復号法である逐次復号法を紹介する．逐次復号法は，ファノメトリックなどを利用して探索空間の削減を行う木探索アルゴリズムと考えることもでき，ビタビ復号では計算量的に取り扱うことのできない状態数の多い畳込み符号の復号に適している。

■1 群 - 2 編 - 5 章

5-1 畳込み符号

（執筆者：吉川英機）[2012 年 3 月 受領]

畳込み符号は符号系列が符号器から出力される時点の情報系列だけでなく，それ以前の情報系列も依存して定まる線形符号である．この符号器は記憶素子をもつ順序回路となり，出力符号系列はその状態によって決まるので状態遷移図を用いて表現することができる．つまり，符号系列の各ビットは符号器に入力された時点の情報だけでなく近接するビットの情報を有していることになり，この時間的な相関性を利用した復号法を適用することができる．

畳込み符号の誤り訂正能力は符号器の状態数，つまり記憶素子の数にも依存する．記憶素子の数は符号系列がそれを決定する情報系列の長さを表すことになり，これが多ければ符号系列の各ビットは以前の情報ビットに関する情報をより多く含んでいることになる．記憶素子の数を m としたとき $L = m + 1$ の値を拘束長という．つまり，ある時点で出力される符号系列は現時点と m 時点前までの合計 $m+1$ 時点の情報系列の拘束を受けて定まることになる．

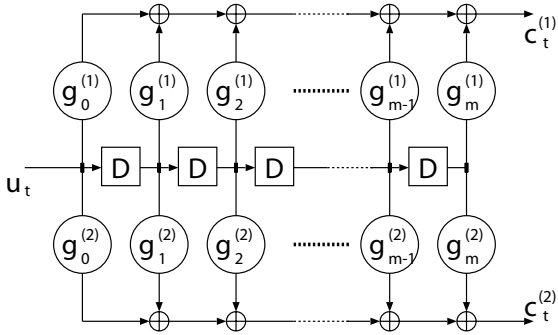

図 5･1　符号化率 1/2 の畳込み符号器（非再帰型）

ここでは 1 ビットの情報記号の入力に対して n ビットの符号系列を出力する符号化率 $1/n$ の畳込み符号を考える．$n = 2$ の畳込み符号器の例を図 5･1 に示す．ここで，D は 1 時点の遅れ演算子を表し，$g_i^{(j)}$ は結合の有無により 0 または 1 となる値である．時点 t において情報記号 u_t がこの符号器入力されるとき，出力される符号系列 $\boldsymbol{c}_t = (c_t^{(1)}, c_t^{(2)}, \ldots, c_t^{(n)})$ は

$$c_t^{(j)} = \sum_{i=0}^{m} u_{t-i} g_i^{(j)} \quad j = 1, \ldots, n \tag{5･1}$$

と表すことができ，u_i と g_i の畳込み演算で得られることが分かる．畳込み符号の生成行列を演算子 D を用いて $\boldsymbol{G} = \left(G^{(1)}(D), G^{(2)}(D), \ldots, G^{(n)}(D)\right)$，$G^{(j)} = \sum_{i=0}^{m} g_i D^i$ と表し，情報記号を $U(D) = u_0 + u_1 D + u_2 D^2 \cdots$ と表すとき，符号系列 $\boldsymbol{C}(D)$ は $\boldsymbol{C}(D) = U(D)\boldsymbol{G}(D)$ と表され，ブロック符号と同様に符号系列は情報記号に対して線形演算で得られるので線形符号の

形となっている．線形符号であればブロック符号と同様に生成行列と検査行列を定義することができる．また，畳込み符号器は m ビット前までの情報記号を記憶しているので，その 2^m 個の状態によって出力される符号系列が定まる．例として，拘束長 $L = 3$，$n = 2$ の符号器，及び状態遷移図を図 5･2 に示す．この生成行列 $\boldsymbol{G}$ と検査行列 $\boldsymbol{H}$ は演算子 D を用いて，$\boldsymbol{G} = \left(1 + D + D^2, 1 + D^2\right)$，$\boldsymbol{H} = \left(1 + D^2, 1 + D + D^2\right)$ と表され，係数をとして 2 を法とする演算とすればブロック符号と同様に $\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$ が成り立つ．

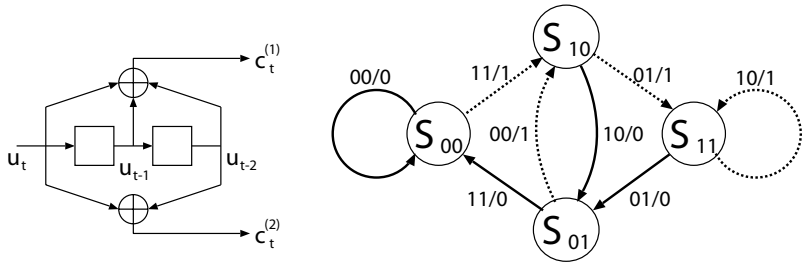

図 5･2　拘束長 $L = 3$ の畳込み符号の符号器とその状態遷移図の例

この状態遷移図において $S_{u_{t-1}u_{t-2}}$ は符号器の状態を表し，各状態間を遷移するときの線に割り当てられている記号は入力される情報記号，及び出力される符号系列を $c_{1,t}, c_{2,t}/u_{1,t}$ として表し，実線は情報記号が 0，破線は情報記号が 1 に対応している．

ここで，図 5･2 の状態遷移図の各状態を縦に，時刻 t を横に並べると図 5･3 のような格子状の図で表すことができる．これをトレリス線図と呼ぶ．トレリス線図において，各状態を結ぶ線を枝と呼び，枝を連ねてできる道をパスと呼ぶ．実線及び破線の枝はそれぞれ情報記号が 0 及び 1 の場合に通る枝を示し，このときに符号器から出力される符号系列を各枝に割り当てており，太線は初期状態を S_{00} として符号器に 10100 を入力したときにたどるパスを示している．ブロック符号と比べた畳込み符号の特長はトレリス線図が単純な形で表現する

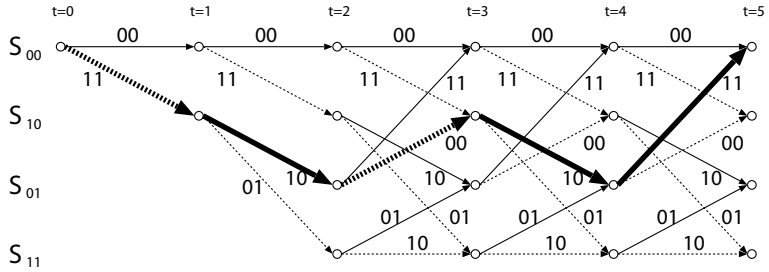

図 5･3　図 5･2 の符号器のトレリス線図

ことができることにあり，後述するビタビ復号を容易に適用することができる．

図 5･3 のトレリス線図において，S_{00} から分岐して再び同じ状態に戻るには 3 本の枝を通る必要がある．したがって，拘束長とは任意の状態から分岐して再び元の状態に戻るために必要な枝数ということもできる．畳込み符号の誤り性能はブロック符号と同様にすべての情報系列に対する符号系列間の最小距離で決まるが，ブロック符号のようにブロック単位で符号化が終端せず，逐次的な符号化が行われるため符号長を任意に設定することができる．

そこで，二つの情報系列 $\boldsymbol{u} = (u_1, u_2, u_3, \ldots), \boldsymbol{v} = (v_1, v_2, v_3, \ldots)$ に対する符号語を $\boldsymbol{c}^{(u)}, \boldsymbol{c}^{(v)}$ とする．このとき，二つの符号系列のハミング距離の最小値 $d_f = \min_{\boldsymbol{u}\neq\boldsymbol{v}} d_H\left(\boldsymbol{c}^{(u)}, \boldsymbol{c}^{(v)}\right)$ を畳込み符号の最小自由距離という．図 5･2 に示す符号器より畳込み符号においてもすべて 0 の符号語が存在し，フロック符号と同様に符号系列の最小ハミング重みが最小自由距離となる．拘束長 3 の畳込み符号は図 5･3 に示すように最上段 $00\cdots0$ から分岐して再び元の状態に戻るパスのなかで最小ハミング重みをもつパスは 111011 であり，この最小自由距離は 5 となり，これは符号器に入力する情報源記号数を増やしてもパス間の距離は変わらないことも確認できる．一般に拘束長を増やして適切な符号器を構成すれば性能の良い畳込み符号を得ることができる．しかし，距離構造を状態遷移図などから解析的に解けるのは拘束長が短い場合に限られており，拘束長が長い畳込み符号の距離構造を理論的に求めるのは容易ではない．

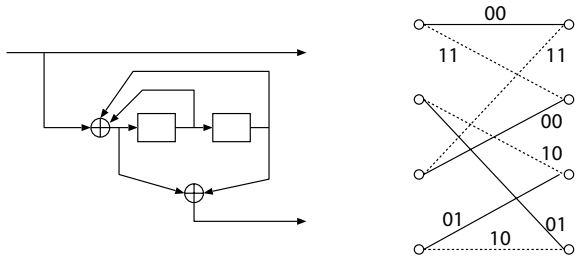

図 5･4　拘束長 3 の再帰型組織畳込み符号の例

畳込み符号は符号器の構造によって組織符号と非組織符号を構成することができる．図 5･2 は非組織畳込み符号であるが，この生成行列 $G(D)$ の各項を $1 + D + D^2$ で割れば $[1, (1 + D^2)/(1 + D + D^2)]$ となり，これを生成行列とする組織畳込み符号は図 5･4 で与えられ，情報記号と検査記号が分離できる形の符号となる．組織畳込み符号の最小自由距離は非組織符号と比べると小さくなるが，図 5･4 のように帰還を有する再帰的な構造の符号器によって構成すると非組織符号と同じ符号が構成できる．図 5･4 に示すトレリス線図は図 5･3 の同様の構造になっており，最小自由距離が 5 になることが分かる．再帰的組織畳込み符号は高 SN 比におけるビット誤り特性は非組織符号と比べて悪くなるが，ターボ符号の要素符号として用いる場合に，容易に短縮化することができるので効率の良い符号を構成することができる．また，符号化率を柔軟に変更することができることからトレリス符号化変調にも適用されている．一方，非組織符号の符号器は帰還がなく非再帰的であり，同じ拘束長をもつ組織符号と比べると最小自由距離が大きく高 SN 比において性能の良い符号を構成することができる．

■1 群 - 2 編 - 5 章

5-2 ビタビ復号

（執筆者：吉川 英機）［2012 年 3 月 受領］

ビタビ（Viterbi）復号[1]は畳込み符号に対して最ゆう復号を容易に実現できる復号法として実用上重要であり，また，軟判定復号を容易に適用することができるので優れた誤り性能を得ることができる．まず，ビタビ復号アルゴリズムで用いるゆう度関数について述べる．ゆう度関数は送信された長さ N の符号系列 $\boldsymbol{c} = (c_1, c_2, \ldots, c_N)$ に対して受信系列 $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$ を受信する確率 $P(\boldsymbol{r}|\boldsymbol{c})$ を表しており，通信路が無記憶であれば $P(\boldsymbol{r}|\boldsymbol{c}) = \prod_{i=1}^{N} P(r_i|c_i)$ となる．最ゆう復号では $P(\boldsymbol{r}|\boldsymbol{c})$ が最大となる系列を復号結果とするので，大小関係が保たれていれば $P(\boldsymbol{r}|\boldsymbol{c})$ の値を用いる必要はなく，この値の対数値を用いた対数ゆう度 $\log P(\boldsymbol{r}|\boldsymbol{c})$ を用いると，

$$\log P(\boldsymbol{r}|\boldsymbol{c}) = \log \prod_{i=1}^{N} P(r_i|c_i) = \sum_{i=1}^{N} \log P(r_i|c_i) \tag{5·2}$$

のように積の演算は和の演算となり計算量を削減できる．ビタビ復号では $\log P(r_i|c_i)$ の値ではなく，この値に比例するメトリックと呼ばれる量の累積値を用いて最ゆう系列を判定する．

では，実際にどのような値をメトリックとして用いることができるであろうか．まず，反転確率 p の 2 元対称通信路における硬判定復号ではハミング距離がメトリックとなることを示す．長さ N の符号系列 $\boldsymbol{c}$ と受信系列 $\boldsymbol{r}$ のハミング距離を t とする．つまり，$\boldsymbol{c}$ を送信したときに通信路上で t 個の誤りが生じた結果が $\boldsymbol{r}$ となって受信されたと仮定する．このときの対数ゆう度関数は $\log P(\boldsymbol{r}|\boldsymbol{c}) = \log p^t(1-p)^{N-t} = -t \log \frac{1-p}{p} + N \log(1-p) = -At - B$ と表すことができる．ここで，$0 < p < 0.5$ とすれば A は正の定数である．つまり，対数ゆう度はハミング距離の一次単調減少関数になっているので，最大ゆう度の符号系列に判定することは受信系列とのハミング距離が最小の符号系列が送られたと判定することと同等である．

次に，AWGN 通信路における軟判定復号の場合を考える．この場合，$c_i \in \{0, 1\}$ に対する BPSK 変調信号として $x_i = \pm 1$ を送信するものとし，受信側ではガウス分布に従う連続値 y_i を受け取ると仮定する．このとき，y_i の確率密度関数が $p(y_i|x_i) = A \exp\left[-B(y_i - x_i)^2\right]$ で与えられる無記憶通信路で表される．ここで，A, B は x_i, y_i に依存しない定数であるから，対数ゆう度関数は次式のように表すことができる．

$$\log P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{i=1}^{N} \left(\log A + \log B - (y_i - x_i)^2\right) = -\sum_{i=1}^{N} d_E(y_i, x_i)^2 + K \tag{5·3}$$

$$= 2\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} (y_i^2 + x_i^2) + K = 2\sum_{i=1}^{N} x_i y_i + K' \tag{5·4}$$

式 (5·3) より 2 乗ユークリッド距離の負値がメトリックとなり，受信系列との 2 乗ユークリッド距離が最小となる符号系列に判定すればよいことになる．また，$y_i^2 + x_i^2$ は各時点において一定の値となることからこれを定数とおくことで式 (5·3) より式 (5·4) が得られる．第 1 項は送信系列 $\boldsymbol{x}$ と受信系列 $\boldsymbol{y}$ の相関関数となるので，最ゆう復号とは受信系列との相関値が最大となる送信系列が送られたと判定することであるともいえる．

ここで，状態数 M のトレリス線図を用いる復号手順の概要を述べる．状態 S_i から状態 S_j に遷移する枝に割り当てられている送信記号と受信記号から得られる時点 t の枝メトリックを $\gamma_t(S_j, S_i)$, $i, j \in \{0, \ldots, M-1\}$ と表すと，ゆう度関数はトレリス線図の各状態のメトリックが最大になるように $\gamma_t(S_j, S_i)$ を加算していくことで得られるから，ビタビ復号法は次式のように時点 t, 状態 S_i におけるパスメトリック $\alpha_t(S_i)$ を繰り返し求めるアルゴリズムとなる．

$$\alpha_t(S_i) = \max_{S_j} \left\{ \alpha_{t-1}(S_j) + \gamma_t(S_j, S_i) \right\} \tag{5･5}$$

この演算によって図 5･5 に示すように各状態につながるパスが一本ずつ生き残ることになる．なお，ゆう度差が 0（$\alpha_{t-1}(S_k) + \gamma_t(S_k, S_j) = \alpha_{t-1}(S_j) + \gamma_t(S_i, S_j)$）のときは無作為に選択する．

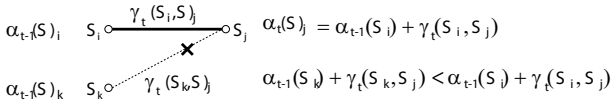

図 5･5　パスメトリックの計算と生き残りパスの選択

トレリス線図上で生き残るパスの本数は符号器の状態数に等しくなるが，符号器の初期状態と最終状態をすべて 0 の状態に定めておくことで最大ゆう度となるパスを一本に確定できる．式 (5･5) の演算は各状態において「加算・比較・選択」といった単純処理で実現することができる．ただし，この演算はトレリス線図の状態の数だけ必要となるので，拘束長 L の符号に適用すると式 (5･5) の演算は 2^{L-1} 回必要となり，復号の計算量は指数関数的に増大する．

以下，図 5･2 の符号器で生成される符号系列に対して図 5･3 に示す 4 状態のトレリス線図を用いて復号アルゴリズムを適用した例を示す．情報記号系列 01110100 を入力したとき符号系列は 0011011001001011 となり，これを送信して受信側では誤りが 4 箇所に生じた 0001011101101011 を受信系列としたとき，メトリックとしてハミング距離を用いて上記の三つの操作を繰り返した結果を図 5･6 に示す．ここで，丸の中の数値は累積ハミング距離を表しており，各状態の選択操作において実線は生き残った枝，破線は切り捨てた枝を表している．生き残りパスをたどると送信系列 0011011001001011 が復号できることが確認できる．

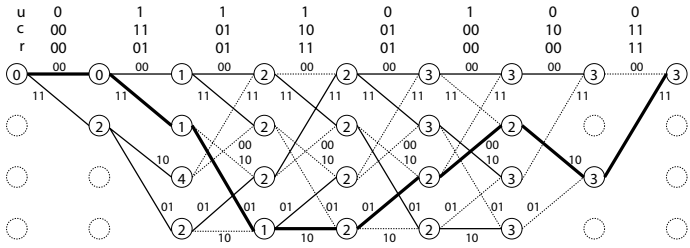

図 5･6　ビタビ復号の例

■1 群 - 2 編 - 5 章

5-3 BCJR アルゴリズム

（執筆者：吉川 英機）［2012 年 3 月 受領］

時点 j において送信記号 c_j が送られる確率を事前確率 $P(c_j)$ と表すとき，受信系列 $\boldsymbol{r}$ を受け取った下で c_j が送られたと判定する条件つき確率を事後確率 $P(c_j|\boldsymbol{r})$ と表す．受信側で事後確率が最大となる記号が送られたと判定する復号法を最大事後確率（maximum a posteriori probability: MAP）復号法といい，ビット誤り率を最小にする最適な復号法となる[2]．

事後確率は $P(c_j|r_j) = P(c_j)P(r_j|c_j)/P(r_j)$ のように事前確率とゆう度で得られ，すべての送信記号の事前確率が等しく，無記憶対称通信路であれば $P(c_j) = P(r_j)$ より，$P(c_j|r_j) = P(r_j|c_j)$ となるから MAP 復号法は最ゆう復号法と等価となる．しかし，事前確率が等確率でない場合は最ゆう復号は最適ではないので，MAP 復号と比べると誤り性能は少し劣化する．ここでは，拘束長 L の畳込み符号で符号化された受信系列に対してシンボル単位で MAP 復号を実現する BCJR アルゴリズムについて述べる．時点 j における受信シンボルを y_j，長さ N の受信シンボル系列 $\boldsymbol{y} = y_1y_2\cdots y_N$ としたときの時点 j における 2 元送信情報 $u_j \in \{+1, -1\}$ の事後確率 $P(u_j|\boldsymbol{y})$ を考える．トレリス線図では u_j は時点 $j-1$ の状態 s_{j-1} から時点 j の状態 s_j，$s_j \in \{S_0, \ldots, S_{2^{L-1}}\}$ に遷移する枝に割り当てられるから u_j を (s_{j-1}, s_j) に対応づけると

$$P(u_j|\boldsymbol{y}) = P(\boldsymbol{y})^{-1}P(u_j, \boldsymbol{y}) = P(\boldsymbol{y})^{-1} \sum_{U(s_{j-1},s_j)=u_j} P(s_{j-1}, s_j, \boldsymbol{y}) \tag{5･6}$$

と表すことができる．ただし，$U(s_{j-1}, s_j)$ は s_{j-1} から s_j に遷移する枝に割り当てられている情報記号を表す．ここで，$\boldsymbol{y}_1^{j-1}$ は $\boldsymbol{y}$ の部分系列を $y_1y_2\cdots y_{j-1}$ と表すと，無記憶通信路であれば，u_j と $\boldsymbol{y}$ の結合確率は $P(s_{j-1}, s_j, \boldsymbol{y}) = P(s_{j-1}, \boldsymbol{y}_1^{j-1})P(s_j, y_j|s_{j-1})P(\boldsymbol{y}_{j+1}^N|s_j)$ のように分けることができて，それぞれを $P(s_j, \boldsymbol{y}_1^{j-1})$ を $\alpha(s_{j-1})$，$P(\boldsymbol{y}_{j+1}^N)$ を $\beta(s_j)$，及び，$P(s_j, y_j|s_{j-1}) = \gamma(s_{j-1}, s_j)$ とおくと，次式のように $\alpha(s_j)$ は前方向，$\beta(s_{j-1})$ は後方向の反復演算のかたちで表現できる．

$$\alpha(s_j) = \sum_{s_{j-1}} \gamma(s_{j-1}, s_j)\alpha(s_{j-1}) \tag{5･7}$$

$$\beta(s_{j-1}) = \sum_{s_j} \gamma(s_j, s_{j-1})\beta(s_j) \tag{5･8}$$

上式は最初と最後の状態が既知であれば図 5･7(a) に示すような反復演算により各時点の確率を得ることができることを示している．通常は $\alpha_0(s_0 = S_0) = \beta_N(s_N = S_0) = 1$, $\alpha_0(s_0 \neq S_0) = \beta(s_N \neq S_0) = 0$ として符号器の最初と最後の状態がすべて 0 となるように定めておく．

MAP 復号法では時点 j における送信情報 u_j の判定には対数ゆう度比（log-likehood ratio: LLR）を用いる．これは，式 (5･6) で表される結合確率の比の対数値であり，

$$L(u_i|\boldsymbol{y}) = \log \frac{P(u_j = +1, y_j)}{P(u_j = -1, y_j)} = \log \frac{P(y_j|u_j = +1)}{P(y_j|u_j = -1)} + \log \frac{P(u_j = +1)}{P(u_j = -1)} \tag{5･9}$$

のような式で与えられるので[4]，各時点 j において $L(u_j|\boldsymbol{y})$ が正の値であれば $u_j = +1$ に，負の値であれば $u_j = -1$ と判定でき，また，この絶対値 $|L(u_i|\boldsymbol{y})|$ は復号結果に対する信頼度情

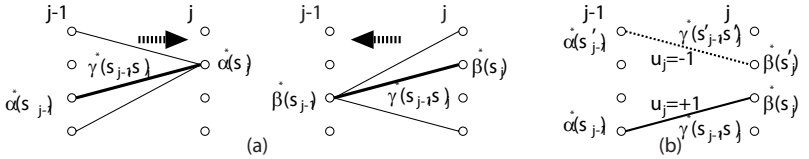

図 5･7　BCJR アルゴリズムにおける双方向演算

報となる．この第 1 項は通信路によって定まる値であり通信路に関する情報を与え，これを $L(y_j|u_j)$ と表す．また，第 2 項は事前確率の対数比となるので事前情報を与え，これを $L(u_j)$ と表す．つまり，LLR は通信路値 $L(y_j|u_j)$ と事前値 $L(u_j)$ の和で与えられることが分かる．

ここで，記号反転確率 q の 2 元対称通信路上の硬判定復号を仮定すると，$u_j = +1$ としたときは $L(y_j|u_j) = \log((1-q)/q)$ で与えられ，通信路値のみで定まり送信情報とは無関係の値となる．また，AWGN 通信路上を一定の振幅 a の振幅の信号を送信する場合，式 (5･9) の第 1 項は，$\log\left(K\exp\left[-E_b/N_0(y_j-a)^2\right]\right) - \log\left(K\exp\left[-E_b/N_0(y_j+a)^2\right]\right) = 4aE_b/N_0 \cdot y_j$ となるので通信路の信号対雑音電力比で与えられるから，送信記号には無関係な通信路値となる．

式 (5･9) において，組織符号で符号化された送信系列を仮定すると，受信シンボル y_j の情報記号に対応する記号を $y_j^{(u)}$，検査記号に対応する記号を $y_j^{(p)}$ としたとき，$\boldsymbol{y}_j = (y_j^{(u)}, y_j^{(p)})$ と表すことができるので $P(\boldsymbol{y}_j|u_j) = P(y_j^{(u)}|u_j)P(y_j^{(p)}|u_j)$ より，$L(u_j|\boldsymbol{y}_j) = L(y_j^{(u)}|u_j) + L(u_j) + L(y_j^{(p)}|u_j)$ のように変形できる．ここで第 3 項は符号系列のパリティ検査部に対する対数ゆう度比となっており外部情報（extrinsic information），または外部値と呼ばれ，後述されるターボ符号では他方の復号器における事前情報として利用することで復号性能を向上させている．

MAP 復号は式 (5･7)(5･8) で示される演算に大きな計算量を必要とするが，ビタビ復号法と同様に対数領域の演算により計算量を削減できる．Max-Log MAP 復号法は $\log\sum_{i=1}^{n} e^{a_i} \simeq \max(a_1, a_2, \ldots, a_n)$ の近似式を適用した復号法であり，式 (5･7)(5･8) の反復演算に適用して，

$$\alpha^*(s_j) = \log\alpha(s_j) = \log\left(\sum_{s_{j-1}} e^{\log(\gamma(s_{j-1},s_j)\alpha(s_{j-1}))}\right) \simeq \max_{s_{j-1}}\left(\alpha^*(s_{j-1}) + \gamma^*(s_{j-1}, s_j)\right) \quad (5\cdot10)$$

$$\beta^*(s_{j-1}) = \log\beta(s_{j-1}) = \log\left(\sum_{s_j} e^{\log(\gamma(s_j,s_{j-1})\beta(s_j))}\right) \simeq \max_{s_j}\left(\beta^*(s_j) + \gamma^*(s_j, s_{j-1})\right) \quad (5\cdot11)$$

のように総和を最大値選択の演算で近似する．ここで，$\gamma^*(s_{j-1}, s_j) = \log P(y_j|s_j, s_{j-1})P(s_j|s_{j-1}) = \log P(y_j|s_j, s_{j-1}) + \log P(u_j)$ と表せるので，事前値となる $\log P(u_j)$ を 0 とした場合，式 (5･10) において $\alpha^*(s_j)$ はビタビ復号法における前方向のパスメトリック，$\log P(y_j|s_{j-1}, s_j)$ は s_j から s_{j-1} につながる枝メトリックに相当する．また，$\beta^*(s_j)$ は時点 N から 0 に向かう後方向ビタビ復号におけるパスメトリックに相当する．この近似によって復号操作は簡単になるものの低 SN 比において性能劣化が生じる[5]．そこで，関数 $F(x) = \log(1+e^x)$ を導入し，$a_1 > a_2 > \cdots > a_n$ となる値に対して $\log\left(\sum_{i=1}^{n} e^{a_i}\right) = a_1 + F(a_1 - a_2 + \cdots + F(a_{n-2} - a_{n-1} + F(a_n)))$ の演算に基づいて LLR を計算するのが Log-MAP 復号法である[3]．実際には a_1, a_2 のみを考慮して

も MAP 復号法と性能はほとんど変わらないので，$\max(a_1, a_2) + F(|a_1 - a_2|)$ の第 2 項を式 (5･10)(5･11) の右辺に加えることで MAP 復号法と同等の誤り性能を維持しつつ，計算量の削減を実現できる．式 (5･6)-(5･11) より対数領域演算によって LLR は次式で与えられる．

$$L(u_j|\boldsymbol{y}) = \max_{u_j=+1}\left(\alpha^*(s_{j-1}) + \gamma^*(s_{j-1}, s_j) + \beta^*(s_j)\right) - \max_{u_j=-1}\left(\alpha^*(s'_{j-1}) + \gamma^*(s'_{j-1}, s'_j) + \beta^*(s'_j)\right) \quad (5\text{･}12)$$

つまり，対数領域における MAP 復号は双方向にビタビ復号アルゴリズムを実行することで $\alpha^*(s_j)$,$\beta^*(s_j)$ を求め，図 5･7(b) に示すように各時点において式 (5･12) を適用することで LLR を得るアルゴリズムといえる．LLR は復号結果 u_j を出力するだけでなくその信頼度情報を与えるので軟出力と呼ばれ，連接符号など複数の復号器を用いる場合にはこれを効果的に利用して復号特性を改善することができる．MAP 復号や max-log-MAP 復号では軟出力を得るためには式 (5･10)(5･11) の双方向演算が必要となるが，これをビタビ復号と同様に前方向演算のみで得るのが軟出力ビタビアルゴリズム（soft output Viterbi algorithm: SOVA）である[4]．

SOVA は式 (5･10) で示す max-log-MAP 復号の前方向演算と同じであり，これは $\alpha^*(s_j) = \alpha^*(s_{j-1}) + \log P(y_j|x_j) + \log P(x_j)$ と書ける．各時点で事前値となる $\log P(x_j)$ を省略して枝メトリックとなる $\log P(y_j|x_j)$ のみ加算し，パスメトリック $\alpha^*(s_j)$ が最大となる一本の生き残りパスを求めるのがビタビ復号である．SOVA では事前値もパスメトリックに加えたうえで最ゆうパスとの比較で生き残れなかったパスとのメトリック差を記録して復号ビットの信頼度に利用している．この軟出力は時点 k におけるメトリック差を Δ_k，時点 j の復号結果を u_j，生き残れなかったパスの復号結果を u'_j としたとき，$L(u_j|\boldsymbol{y}) = u_j \cdot \min_{u_j \neq u'_j} \Delta_k$ で与えられる．

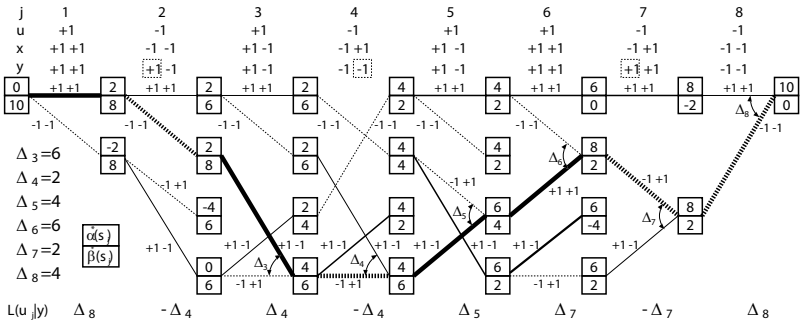

図 5･8　SOVA における軟出力 (LLR) の計算例

図 5･8 は式 (5･4) で示す $x_i y_i$ を各枝のメトリックとした SOVA の適用例である．図 5･4 に示す畳込み符号の入力を 01010011 とし，その符号系列 $\boldsymbol{x}$ を送信して，3 ビットの誤りが生じた受信系列 $\boldsymbol{y}$ に対して復号を実行している．ただし，$\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}$ は 0 に対して +1, 1 に対して −1 に変換している．この図では前方向の生き残りパスのみを表示しているが，各状態の上下の数

値は前方向，及び後方向のビタビ復号を実行したときのパスメトリック $\alpha^*(s_j), \beta^*(s_j)$ を示す．例えば，時点 2 では $u_2 = -1$ に判定され，$u_2' = +1$ となるパスとの最小メトリック差は Δ_4 となり，$L(u_2|\boldsymbol{y}) = -\Delta_4$ となる．この時点に式 (5･12) を適用すると $(2+0+6)-(2+0+8) = -2$ となり，Max-log-MAP 復号と SOVA の軟出力は一致することが分かる（ここでは事前値 0）．

■1 群 - 2 編 - 5 章

5-4 逐次復号法

（執筆者：大橋 正良）[2012 年 3 月 受領]

5-4-1 逐次復号の原理

逐次復号（Sequential Decoding）は，ビタビ復号のようなトレリスを用いる復号が困難な拘束長の長い畳込み符号に対して，木探索の手法によって近似的に最尤復号を実現しようとする復号法である．逐次復号の歴史は古く，1960 年代に畳込み符号が議論されていた時期から研究が行われていた[8, 9]．

逐次復号の原理を図 5･9 に示す．これは符号化率 $r = 1/2$ の畳込み符号を想定した木構造に展開された符号語の部分集合である．枝 1 本が情報 1 ビット，符号化系列 2 ビットに対応している．枝の深さが n ならば，木は 2^n 本の枝に分岐する．逐次復号では，受信系列と，候補の符号系列間の尤度（メトリック）を求め，それが十分に確からしい符号系列（パス）と判断される場合，その探索木を一本づつ伸ばして探索する手法をとる．ここでは簡単に，受信系列と想定符号語間のメトリックをハミング距離とし，ビットが合致したら +1，不一致なら −1 を与える．更に後述するが，異なる長さをもつ複数候補パスの蓄積メトリックを公正に評価する目的で 1 枝当たり −0.5 のバイアスを課してみる．

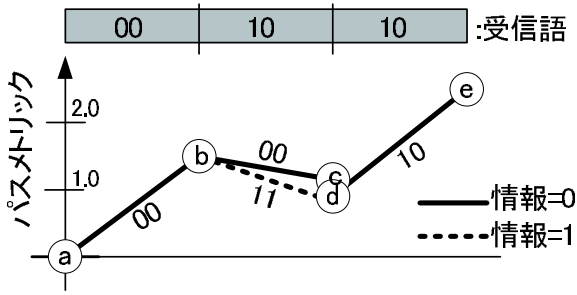

図 5･9　逐次復号の原理

まず (a) から試しに 1 本の枝（情報 0）を伸ばす．このときの符号語 00 は受信語 00 と全部一致したので，メトリックは 0 から 1.5 となる．次に伸びたノード (b) から 1 本の枝（情報 0）を伸ばす．このとき 1 ビットが不一致なので (c) のメトリックは −0.5 下がり，積算値が 1.0 となる．このため，ノード (b) から別の符号語を試してみると (d) でも逆の不一致となり，積算値が同じく 1.0 となる．ほかに候補もないので，このまま (d) を伸ばすと (e) でメトリックが順調に伸びたので，このまま探索が進められることになる．このように逐次復号では，候補となる符号系列はそれぞれ長さが異なっているので，これを補うべく未復号部分の平均評価量に相当するバイアスをかけたメトリックを定めて，木探索を行う必要がある．逐次復号では枝当たり次式で定義される，**ファノメトリック**（Fano metric）と呼ばれる尺度が適用される．

$$c(y^{(j)}, w^{(j)}) = \log\left(\frac{P_{y^{(j)}\to w^{(j)}}}{P(w^{(j)})}\right) - r \tag{5･13}$$

ただし $P_{y^{(j)}\to w^{(j)}}$ ：通信路シンボル $y^{(j)} \to w^{(j)}$ の遷移確率
$P(w^{(j)})$ ：出力シンボル $w^{(j)}$ の生起確率
$r = k/n$ ：符号化率

最初のノードからの符号化系列に沿った枝メトリックの和がパスメトリック $c(\hat{y})$ を与える．

5-4-2 代表的なアルゴリズムと復号特性

（1）ファノ（Fano）アルゴリズム

ファノアルゴリズム（Fano algorithm）[8] は，パスメトリックの閾値を用いて適応的に探索範囲をコントロールしつつ，探索木に沿って探索ポイントを前後させて探索を繰り返すことによって最適パスを見いだす．図 5･9 で例えば閾値を 0 とすると，このアルゴリズムはパスメトリックが正の範囲で探索ポイントを動かして，探索を尽くす．どうしてもこの閾値を超える候補パスがない場合に，ある数値 Δ だけ閾値を下げて探索範囲を広げ，これを繰り返す．うまくパスメトリックが伸びる符号語系列が見出せて，パスメトリックが順調に伸びると閾値を Δ 上昇させる．

ファノアルゴリズムは，雑音が少なく，順調に探索が進む間は効率が良いが，雑音が増加するにつれ，探索に要する計算回数が急速に増加する特徴をもつ．

（2）スタックアルゴリズム

スタックアルゴリズム（Stack algorithm）は，Jelinek[10]によって提案されたアルゴリズムで，Fano アルゴリズムでは不可避である同じノードの繰り返しのトレースをスタックと呼ばれるストレージを用いて回避する手法である．すなわち，探索で木の枝が伸ばされるごとに，ノード情報を，そのパスメトリック値でソートしてスタックに格納する．次のステップでは，スタック中から最もパスメトリック値の大きなノードが取り出され，探索木が伸張される．ストレージが必要となる代わりに高雑音下でも比較的計算回数が少なく抑えられる．

（3）逐次復号器の復号特性

逐次復号のパフォーマンスは，誤ったパスを選択することによって生じる見逃し誤りと，復号に要する探索回数が許容回数を上回ってそれ以上の復号が不可能となって生じるオーバフローに起因する誤りの二つがある．

見逃し誤りの発生確率はビタビ復号時の特性に準じるが，オーバフローが生じる場合には，これ以上の復号を断念するか，もしくは復号器のリセット/再起動/再同期などを講じる必要があり，これが復号特性を支配する主要因になっている．

（4）逐次復号に適した符号

逐次復号では，ビタビ復号と同様，見逃し誤りを避ける観点から符号語間の距離ができるだけ離れた最小自由距離が大きい符号が求められる．逐次復号では拘束長の長い符号を使うのは容易なので，この距離は比較的大きく確保できる．

一方で，バッファオーバフローを避ける視点からは，木探索の初期段階で，探索の深さが

浅いうちに正しいパスと誤ったパスの符号間距離が速やかに増加してゆく符号が望ましい．このような特性をもつ符号は最適列距離（ODP; Optimum Distance Profile）符号と呼ばれ，計算機探索によって最適符号が求められている[11)]．

■参考文献

1) A.J. Viterbi, “Convolutional codes and their performance in communication systems,” IEEE Trans. Commun., vol.19, pp.751–772, 1971.
2) L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” IEEE Trans. Inf. Theory, vol.20, pp.284–287, 1974.
3) H.Wang, H.Yang, and D. Yang, “Improved Log-MAP decoding algorithm for turbo-like codes,” IEEE Commun. lett., vol.16, pp.186–188, 2006.
4) J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and covolutional codes,” IEEE Trans. Inf. Theory, vol.42, pp.429–445, 1996.
5) P. Robertson, E.Villebrun, and P.Hoeher, “A comparison of optimal and suboptimal MAP decoding algorithms operation in the log domain,” Proc. of IEEE Int. Conf. on Commun., vol.2, pp.1009–1013, Jun. 1995.
6) 西村芳一, “ディジタル・エラー訂正技術入門, ” CQ 出版社, 2004.
7) S. Lin and D.J. Costello, Jr., Error Control Coding, Second Edition, Pearson Prentice Hall, 2004.
8) R.M. Fano, “A heuristic discussion of probabilistic decoding,” IEEE Trans. Inf. Theory, vol.9, pp.64–74, Apr. 1963.
9) G.D. Forney Jr., “Convolutional code III : sequential decoding,” Inf. Control, vol.25, pp.267–297, July 1974.
10) F. Jelinek, “A fast sequential decoding algorithm using a stack,” IBM J. Res. Dev., vol.13, pp.675–685, 1969.
11) J.L. Massey and D.J. Costello, “Nonsystematic convolutional codes for sequential decoding in space applications,” IEEE Trans. Commun., vol.19, pp.806–813, Oct. 1971.