

ブロック符号に対するトレリスのセクション構成について

八木 秀樹 岡田 知嗣 小林 学 平澤 茂一

早稲田大学 理工学部 経営システム工学科
〒169-8555 東京都新宿区大久保 3-4-1
tel.03-5286-3290
fax.03-5273-7215
E-mail: yagi@hirasa.mgmt.waseda.ac.jp

あらまし: H.T.Moorthy, S.Lin と G.T.Uehara はビタビ復号器の特性を活かしたトレリスの構成法を提案し, 復号時間が短縮できることを示した. 構成されるトレリスは非最簡ではあるが, 状態数が増加しても, ある1時点での最大状態数が最小トレリスのそれ以下であれば1つの並列成分を復号する速さは最小トレリスの復号時間以下であることが保証されている. しかし, この方法ではあらかじめセクション境界を決定してからでないとトレリスを構成できない.

本稿では, 最適セクション構成法に基づいて, 並列成分を多く持つと同時に, 各並列成分の復号演算量を低減するトレリスの構成法を提案する. これにより, 復号器の並列処理が効率よく実行でき, 復号時間がより短縮できることを示す.

キーワード: トレリス, 非最簡トレリス, 並列成分, セクション構成, ブロック符号, ビタビ復号器

On Sectionalization of Trellis for Block Codes

Hideki YAGI Tomotsugu OKADA Manabu KOBAYASHI Shigeichi HIRASAWA

Dep. of Industrial and Management Systems Engineering
Waseda University
3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169-8555
tel.03-5286-3290
fax.03-5273-7215
E-mail: yagi@hirasa.mgmt.waseda.ac.jp

Abstract: H.T.Moorthy, S.Lin and G.T.Uehara have proposed a trellis construction method which is based on the IC implementation of Viterbi decoder and also shown that we can speed up a decoding time of given received sequences with this trellis. Even though the trellis is nonminimal and the total number of states increases, if the maximum number of states of each time axis is equal to or smaller than that of the minimal trellis, they have proved that the decoding speed of each parallel component is faster than that of the minimal trellis. However this method can construct only trellis where the section boundaries are decided beforehand.

In this paper, we propose a new trellis sectionalization method, where the trellis has many parallel components, based on optimal sectionalization. We also show that we can effectively carry out the parallel implementation of a decoder so that we can reduce a decoding time by means of it.

key words: trellis, nonminimal trellis, parallel component, sectionalization of trellis, block code, Viterbi decoder

1 はじめに

誤り訂正符号において、2元線形ブロック符号に対する最尤復号法は復号誤り確率を最小とするが、その計算量が莫大となる問題点がある。これに対しブロック符号に対するトレリス線図（以降単にトレリスと呼ぶ）を利用し、その計算量を大幅に削減する研究が盛んに行われている[1]-[9]。トレリスを用いたビタビ復号法などの最尤復号法では符号長の増加と共に計算量が増大してしまうため、ブロック符号に対する最簡トレリスを作成し、その計算量を低減する研究も数多くなされている[2]-[5]。また A.Lafourcade らは符号長 n のブロック符号に対する n セクション最簡トレリスから、総演算量を低減するための最適セクション分割を行う手法を提案している[6]。

その一方で、H.T.Moorthy らは符号のコセット分割の考えを利用して、並列成分を多く持つ非最簡トレリスの構成法（以降 MLU 法と呼ぶ）を提案した[1]。各並列成分は最簡トレリスであり、これらが始状態 σ_0 と終状態 σ_f のみで結合したトレリスが得られる。結果的に得られるトレリスは、各並列成分に同じ構造をもち、枝につけられるラベルのみ異なる。この非最簡トレリス全体の復号にかかる演算量は最簡トレリスに比べて増加しても、各並列成分を並列に処理することにより復号時間は短縮できる。また、装置規模は最簡トレリスのものとはほぼ変わらない。この理由として

- 1 状態の総数が増加しても、ある 1 時点での最大状態数が最簡トレリスの最大状態数以下である限り、ビタビ復号器の ACS 演算器の数は同数であること。
- 2 各並列成分は最簡トレリスであり、始状態 σ_0 と終状態 σ_f のみで結合するので、各々並列して処理ができること。

という非最簡トレリスの 2 つの性質が挙げられる[1][10]。

MLU 法ではあらかじめセクション分割を決定しておかなければならないため、セクションの長さが一定 (uniform structure) のトレリスのみを扱っている。このとき、並列成分を可能な限り多く持つ非最簡トレリスの方が、1 つの並列成分の構造は簡潔で、復号計算量も少なく済む。また、構成したトレリスが同数の並列成分を持っていたとしても、セクション境界の位置 (セクションの長さ) によって一つの並列成分にかかる復号計算量が大きく変わってしまう。

そこで本稿では、Lafourcade らの最適セクション分割の考えに基づいて、トレリスをセクション構成しつつ並列成分を多く持つトレリスの構成法を提案する。また、これにより復号時間が短縮できることを示す。

2 準備

2.1 最簡トレリス

符号長 n 、情報記号数 k の 2 元線形ブロック符号 C を考える。符号 C はセクションの数 $l, 1 \leq l \leq n$ の l セクショントレリス $T_l(S, B, L)$ で表わせる。 S は状態の集合、 B は枝の集合、 L は各枝につけられるラベル集合で

ある。深さが n のトレリスは時点 0 に始状態 σ_0 と時点 n に終状態 σ_f をそれぞれ 1 つ持つ。

第 i セクション ($i = 1, 2, \dots, l$) の終わりの時点における状態の集合を $S_i, i = 1, 2, \dots, l$ と定義する。また $S_0 = \sigma_0, S_l = \sigma_f$ とする。さらに $s_i = \log_2 |S_i|, i = 0, 1, \dots, l$ と定義される系列 (s_0, s_1, \dots, s_l) は SCP (State Complexity Profile) と呼ばれる。ある l セクショントレリスの SCP の各要素があらゆる l セクショントレリスの中で最小となるトレリスを l セクション最簡トレリスと呼ぶ。 l セクション最簡トレリスは符号のグラフ表現に適している[2][4]。

2.2 トレリス用生成行列 (TOG)

C の生成行列を G とし、 $r_j = (r_{j,1}, r_{j,2}, \dots, r_{j,n})$ を G の j 行目とする。いま任意の j に対し

$$x = \min\{t | r_{j,t} = 1\}, \quad y = \max\{t | r_{j,t} = 1\}, \quad (1)$$

としたとき r_j に対する $span(r_j), aspan(r_j)$ を

$$span(r_j) = [x, y], \quad aspan(r_j) = [x, y - 1] \quad (2)$$

と定義する。ここで、 $[,]$ は閉区間を表す。 $x = y$ の場合は $aspan(r_j) = \phi$ とする。

以下の性質を持つ生成行列をトレリス用生成 (Trellis Oriented Generator : TOG) 行列とする[11]。

1. 各行の先頭の 1 は下の行の先頭の 1 よりも早い時点にある。
2. 各行の最後尾の 1 は他の行の最後の 1 とはそれぞれ異なる。

T を C の n セクション最簡トレリスとすると、SCP の要素 s_i は TOG 行列 G の時点 i において $i \in aspan(r_j)$ を満足する G の行の数と一致する。すなわち

$$s_i = |\{j | i \in aspan(r_j)\}|. \quad (3)$$

時点 $i, 0 \leq i \leq n$ において未だ $aspan$ に入らない G の行集合を生成行列とする C の部分符号を、時点 i における future 部分符号 F_i と表す。また時点 $i, 0 \leq i \leq n$ においてすでに $aspan$ を終えた G の行集合を生成行列としてもつ C の部分符号を past 部分符号 P_i と表す。さらに、時点 i における F_i, P_i の次元数をそれぞれ f_i, p_i と定義したとき、 (f_1, f_2, \dots, f_n) を FAP (Future Active Profile)、 (p_1, p_2, \dots, p_n) を PAP (Past Active Profile) と呼ぶ。ここで任意の i に対し s_i, f_i, p_i の間には

$$k = s_i + f_i + p_i, \quad (4)$$

という関係が成り立つ。

2.3 並列成分を多く持つ非最簡トレリス構成法 (MLU 法)

l セクショントレリスにおいて, 符号 C に対する SCP の時点 i の要素を明示的に $s_i(C)$ と表す. また $s_{max}(C) = \max_i(s_i(C))$ とする. さらに

$$I_{max}(C) = \{i | s_i(C) = s_{max}(C)\}. \quad (5)$$

と定義する. ここで, $\dim(C') = \dim(C) - 1$ となるような C の部分符号 C' を考える. いま コセットリーダー $[C/C']$ が r_j であると仮定する. このとき時点 i において $i \in \text{aspan}(r_j)$ を満足するならば, $s_i(C') = s_i(C) - 1$ であり, そうでなければ $s_i(C') = s_i(C)$ であるのは明らかである. また, G が TOG であるので, $G' = G - r_j$ も TOG となる. ただし, $G - r_j$ は G の行から r_j を削除することを意味する. 従ってこの操作を $I_{max}(C)$ の時点すべてを aspan として含む行 j に対して行えば, 最大状態数を増やすことなく並列数が 2 の並列トレリスを作成することが可能となる.

同様の操作を繰り返すことができるように

$$R(C) = \{r_j | \text{aspan}(r_j) \supseteq I_{max}(C)\}, \quad (6)$$

と定義し, $\rho = |R(C)|$ とすると以下の定理が導かれる.

定理 1 (並列成分の存在条件 [1])

$\rho', 1 \leq \rho' \leq \rho$, に対し, $s_{max}(C') = s_{max}(C) - \rho'$ かつ $\dim(C') = \dim(C) - \rho'$ となる C の部分符号 C' が存在するための必要十分条件は, $s_i(C) > s_{max}(C)$ を満たすすべての i に対し, i を aspan に含む少なくとも $s_i(C) - s_{max}(C)$ 行が R の中にあるような, $R(C)$ の ρ' 行からなる部分行列 $R' \subseteq R(C)$ が存在することである. \square

上で述べたように, 定理 1 を満たす最大の ρ' に対し並列数 $2^{\rho'}$ の非最簡トレリスを構成することは容易である. また, このようにして構成された非最簡トレリスの最大状態空間次元数は元の最簡トレリスのそれと変わらない.

2.4 トレリスの演算

始状態 σ_0 と終状態 σ_f を持つある l セクショントレリス T と, 始状態 σ'_0 と終状態 σ'_f を持つある l' セクショントレリス T' が存在し, $\sigma_f = \sigma'_0$ のとき, トレリスに関する合成, 融合という 2 つの演算を次のように定義する. T と T' の合成 $T \circ T'$ とは T の終状態と T' の始状態を結合させた $l+l'$ セクショントレリスを生成する演算と定義する. また, T と T' の融合 $T * T'$ とは, $T \circ T'$ の始状態 σ_0 から終状態 σ'_f までの 1 つのパスを 1 つの枝に対応させた 1 セクショントレリス (始状態 σ_0 , 終状態 σ'_f) を生成する演算と定義する.

いまトレリス T と T' それぞれの表すあらゆるパスのラベルの集合が等しいとき, $T \simeq T'$ と表すと

$$T \circ T' \simeq T * T'. \quad (7)$$

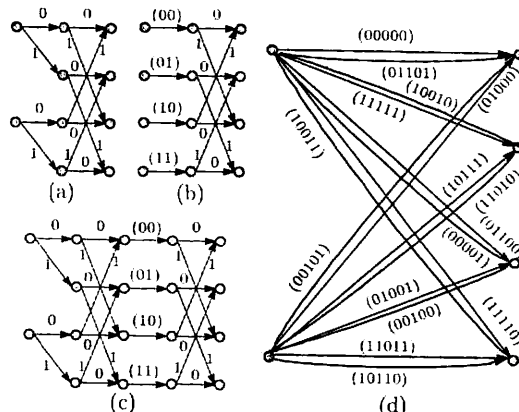


図 1: トレリスの合成・融合の例. (a) トレリス T . (b) トレリス T' . (c) 合成 $T \circ T'$. (d) 融合 $T * T'$

が成り立つ [5]. また, 合成, 融合ともそれぞれ結合則が成り立つ [5]. すなわち 3 つのトレリス T, T', T'' があるとき

$$\begin{aligned} (T \circ T') \circ T'' &= T \circ (T' \circ T'') = T \circ T' \circ T'', \\ (T * T') * T'' &= T * (T' * T'') = T * T' * T'', \end{aligned} \quad (8)$$

である.

以上の式 (7), (8) より, n セクショントレリスから合成, 融合を用いて l セクショントレリスを構成することをセクション構成 (sectionalization) するという.

図 1 にトレリスの演算の例を示す [6]. ここで, (a) を T , (b) を T' で表す 2 つの 2 セクショントレリスとすると, (c) は 2 つのセクショントレリスの合成 $T \circ T'$ である. また, (d) は 2 つのセクショントレリスを融合した $T * T'$ を表す. 2 セクショントレリスに対し, 合成演算を行うと 4 セクショントレリスが, これらに対し融合演算を行うと, セクション長 5 の 1 セクショントレリスが得られる.

2.5 最適セクション構成法 (LV 法)

A.Lafourcade と A.Vardy は Dijkstra アルゴリズムを利用し, 目的関数に対し最適であるセクションの境界を持つトレリスのセクション構成法 (以降 LV 法と呼ぶ) を提案した [6].

ビタビ復号法を対象とした場合の目的関数について考える. トレリス T に対するビタビ復号に必要な総計算量は, 各枝のメトリックを算出する計算量 $M(T)$ とその枝メトリックを Add-Compare-Select (ACS) する計算量 $D(T)$ の合計であり, その値を目的関数 $F(T) = M(T) + D(T)$ とする.

このような目的関数を最小とするトレリスのセクション境界を決定する LV アルゴリズムを以下に示す. ただし, 初期トレリスとして用いる n セクション最簡トレリスを $T_1 \circ T_2 \circ \dots \circ T_n$ と表す (T_i は i 番目のユニットセクショントレリスを表す).

Step1) n セクショントレリスを n 個のユニットトレリス T_1, T_2, \dots, T_n に分け, $T_1^{min} = T_1, i = 2$ とする.

Step2) 式(9)~(11)の最小値を求める.

$$F(T_{i-1}^{min}) + F(T_i), \quad (9)$$

$$F(T_{j-1}^{min}) + F(T_j * T_{j+1} * \dots * T_{i-1} * T_i), \quad (10)$$

$$\text{for } j = 2, 3, \dots, i-2, i-1,$$

$$F(T_1 * T_2 * \dots * T_{i-1} * T_i). \quad (11)$$

Step3) step2) において最小値となる式に対応する以下の操作を行う.

$$(9) \rightarrow T_i^{min} = T_{i-1}^{min} \circ T_i,$$

$$(10) \rightarrow$$

$$T_{j-1}^{min} = T_{j_{min}-1}^{min} \circ (T_{j_{min}} * \dots * T_{i-1} * T_i),$$

$$(11) \rightarrow T_i^{min} = T_1 * T_2 * \dots * T_{i-1} * T_i,$$

ただし j_{min} は式(10)において最小値となる j を表す. $i = n$ なら, T_n^{min} が最適セクション構成されたトレリスである. $i < n$ なら $i = i+1$ とし, Step2)へ.

目的関数を $F(T) = M(T) + D(T)$ とすると, 最適セクション構成されたセクション境界に関し, 次の定理が成り立つ.

定理 2 (最適セクションの境界 [6])

FAP と PAP において, 第 i 要素の変化を $\Delta f_i = f_i - f_{i+1}, \Delta p_i = p_i - p_{i-1}$ と表す. i 時点において $(\Delta f_i, \Delta p_i) = (1, 1)$ となるとき, i 時点は必ずセクションの境界となる. \square

これをトレリスの動的性 (dynamics) という.

3 並列非最簡トレリスのセクション構成法

MLU 法ではセクションの長さを一定にしないことも可能ではあるが, 並列数を多く持つ非最簡トレリスを作成するに先立ち, あらかじめセクション境界を決定しておくかなければならない. 符号長が n の符号に対し 2^{n-1} 通りのセクション構成の中から, 復号時間が短縮できるものを選択するのは現実的ではないため, 限られたパターンのセクション構成から復号時間のなるべく短いトレリスを選択することになる ([1] ではセクションの長さが一定のトレリスのみを扱っている).

しかし, セクション構成を行いつつ並列成分を生成することができれば, より効率良く復号することができるトレリスを構成することが可能となると考えられる. そこで, 本節では並列成分を多く持ち, セクション長を可変にするセクション構成法を非最簡トレリスに適用する手法を提案する.

3.1 提案アルゴリズム

符号 C のトレリスに対し各並列成分を並列にビタビ復号を行うとき, その復号時間は 1 つの並列成分のビタビ復号を行う時間と, 最後に各並列成分の生き残りパス同士を比較・選択する時間の合計となる [1]. そこで, 本節では計算量として 1 並列成分をビタビ復号する計算量を目的関数とするセクション構成を行うアルゴリズムを考える.

まず, あるトレリス T に対する SCP を $SCP(T)$ と表す. 同様に PAP, FAP についても $PAP(T), FAP(T)$ と表すことにする. LV 法と同様に, まず n セクション最簡トレリスを各セクションの長さが 1 のユニットトレリス $T_i, i = 1, 2, \dots, n$, に分解し, これを初期トレリスとする.

以下に提案アルゴリズムを示す.

提案アルゴリズム

Step1) 符号 C の n セクション最簡トレリスを T とし, C に対する TOG 行列を G とする. $i = 2, h = 1, T_1^{min} = T_1$ とする.

Step2) $SCP(T)$ に対し式(6)を満たす $R(C)$ を求め $\rho = |R(C)|$ とする. また定理 1 を満足する最大の ρ' と R' を求め, $G' = G - R'$ とする. G' を TOG 行列とする C の部分符号を C' とし, これに対応する C の 1 並列成分であるトレリスを T' とする. また, T' に対し $SCP(T')$ を作成する. さらに, $PAP(T'), FAP(T')$ を求め, これに対応する $(\Delta f_i, \Delta p_i)$ を計算する.

Step3) もし $i = n+1$ ならば T' が求める並列部分トレリスであり, 残りの $2^{\rho'} - 1$ 個の並列成分と結合し, 終了 (他の並列成分は TOG より容易に求まる).

Step4) $h = 1$ あるいは $\rho' = 0$ ならば Step6)へ. そうでなければ T_{i-1}^{min} と同じセクション境界を持つ T' の $i-1$ 時点までの部分トレリス T'_{i-1}^{min} について $F(T'_{i-1}^{min})$ を計算する.

Step5) $G = G', C = C', T = T', T_{i-1}^{min} = T'_{i-1}^{min}$ とする.

Step6) 式(12)~(14)の最小値を求める (ただし $h > 1$ のときは式(14)は求めない).

$$F(T_{i-1}^{min}) + F(T_i), \quad (12)$$

$$F(T_{j-1}^{min}) + F(T_j * T_{j+1} * \dots * T_{i-1} * T_i), \quad (13)$$

$$\text{for } j = h+1, h+2, \dots, i-2, i-1$$

$$F(T_1 * T_2 * \dots * T_{i-1} * T_i). \quad (14)$$

Step7) step6) において最小値となる式に対応する以下の操作を行う (ただし $h > 1$ のときは式(14)に対応

する操作は行わない)。

$$(12) \rightarrow T_i^{min} = T_{i-1}^{min} \circ T_i,$$

$$(13) \rightarrow T_i^{min} = T_{j_{min}-1}^{min} \circ (T_{j_{min}} * \dots * T_{i-1} * T_i),$$

$$(14) \rightarrow T_i^{min} = T_1^{min} * T_2 * \dots * T_{i-1} * T_i.$$

ただし j_{min} は式 (13) において最小値となる j を表す。

Step8) $i = n$ もしくは $(\Delta f_i, \Delta p_i) = (1, 1)$ の場合, $T = T_i^{min} \circ T_{i+1} \circ \dots \circ T_n$. $h = i, i = i + 1$ とし, step2) へ. そうでなければ $i = i + 1$ とし, Step6) へ.

□

step1) では n セクショントレリスを n 個のユニットセクショントレリスに分解し, それらを T_1, T_2, \dots, T_n とする. また, 初期値として, $i = 2, h = 1, T_1^{min} = T_1$ とする. step2) において, 並列成分 T' の抽出を行う. step3) で $i = n + 1$ のとき, step2) から得られる T_{n+1}^{min} と同じセクション境界を持つ T' がもつめる並列成分であるので, 繰り返しを終了する. step4), step5) において, step2) で新たな並列成分 T' が抽出できるのであれば, 目的関数値 $F(T_i^{min}), G, C, T, T_i^{min}$ を更新する. step6), step7) でセクション構成を行う. step8) で $(\Delta f_i, \Delta p_i) \neq (1, 1)$ である場合, step6) のセクション構成の過程へ戻り, そうでない場合は step2) の並列要素の抽出過程へもどる.

4 評価基準と結果

3 節で示した提案アルゴリズムを用いて, 符号長 32 の拡大 (ex)BCH 符号と符号長 32, 64 の RM 符号についてセクション構成した結果を示す. 拡大 BCH 符号の生成行列は列置換したものを使用する [3][11]. この時の評価基準は復号の時間として, 1 つの並列成分について復号するのに必要な計算量と, それぞれの並列成分で復号した $2^{p'}$ 個の生き残りパスを比較・選択する計算量 $2^{p'} - 1$ を加えたものとする. ここで, 減算や比較演算は加算と等価な演算量として総演算回数を求める.

メトリックを求める演算量 $M(T)$ や ACS 演算量 $D(T)$ は以下のようにして求められる. ある時点 h から時点 h' までの 1 セクショントレリス $T_{h,h'}$ に対し, セクションの長さが $m = h' - h$ の時, メトリックを算出する演算量 $M(T_{h,h'})$ は Gray 符号に基づけば

$$\begin{aligned} M(T_{h,h'}) &= (m - 1) + (2^{m-1} - 1) \\ &= 2^{m-1} + m - 2, \end{aligned} \quad (15)$$

で求まる [6].

同様に ACS 手続きを行う演算量 $D(T_{h,h'})$ は

$$\begin{aligned} D(T_{h,h'}) &= |B_{h'}| - |S_{h'}| \\ &= 2^{s_0+f_0-f_{h'}} - 2^{s_{h'}}, \end{aligned} \quad (16)$$

$$\begin{aligned} D(T_{h,h'}) &= 2|B_{h'}| - |S_{h'}| \\ &= 2^{s_h+f_h-f_{h'+1}} - 2^{s_{h'}}, \end{aligned} \quad (17)$$

表 1: 符号長 $n = 32$ の exBCH 符号, RM 符号の復号演算量

(n, k) 符号	MLU 法				提案法
	4-sec	8-sec	16-sec	32-sec	
(32,6)RM _{5,1}	558	136	132	188	126
(32,11)exBCH	782	466	638	1,072	432
(32,16)RM _{5,2}	1,078	582	1,318	1,080	579
(32,21)exBCH	17,018	5,094	7,878	8,956	2,920
(32,26)RM _{5,3}	4,992	1,336	956	1,010	897

表 2: 符号長 $n = 32$ の exBCH 符号, RM 符号の並列成分数 ρ'

(n, k) 符号	MLU 法				提案法
	4-sec	8-sec	16-sec	32-sec	
(32,6)RM _{5,1}	3	2	2	2	2
(32,11)exBCH	4	4	4	4	4
(32,16)RM _{5,2}	3	3	2	3	3
(32,21)exBCH	4	3	2	2	3
(32,26)RM _{5,3}	1	1	1	1	1

で求まる. ここで S_i は時点 i における状態の集合, B_i は S_i に集まる枝の集合である. 式 (16) は第 1 セクションの枝メトリックについて比較する過程に要する演算量であり, 式 (17) は第 2 セクション以降のメトリックの ACS 手続きに必要な演算量である. 第 1 セクションの復号過程では状態メトリックは 0 なので, 枝メトリックの比較のみが必要となる.

表 1, 2, 3, 4 に数値計算結果を示す. ここで, n は符号長, k は情報記号数, RM _{m,r} はオーダ数 r , 符号長 2^m の RM 符号を表している. 表 1 では MLU 法を利用することができるセクション長一定の 4, 8, 16, 32 セクショントレリスと, 提案法によりセクション構成されたトレリスの比較を行った結果を示す. また, 表 3 にはセクション長 8, 16, 32, 64 のトレリスと提案法によりセクション構成されたトレリスの比較を行った結果を示す. 表 1 と表 3 には 1 並列成分の復号演算量と各並列成分の生き残りパスを比較した演算量の合計を計算量として, また表 2 と表 4 には並列成分の次元数 ρ' を示す. ただし, 符号長 32 の RM 符号の 2 セクショントレリスと, 符号長 64 の RM 符号の 2, 4 セクショントレリスについては, 復号計算量が莫大になってしまうため割愛する. 表 1 と表 3 より, 符号長 32 の拡大 BCH 符号と符号長 32, 64 の RM 符号のそれぞれについて, 復号時間を短縮することが可能であることが分かる.

5 考察

4 節の表 1, 3 より, 符号長 32 の拡大 BCH 符号と符号長 32, 64 の RM 符号のそれぞれについて, MLU 法で構成できる uniform structure のトレリス 4 パターンよりも, 提案法を用いてセクション構成したトレリスは復号時間を短縮することが可能であることがわかる.

また表 2 から, 1 つの符号に対し単に並列成分を多く持つように構成すればよいわけではないことは明白である. (32,6)RM 符号, (32,21)exBCH 符号では, 4 セクショントレリスの並列数が最大となっているが, 復号計算量は他と比べて最大になってしまうことに注目されたい. つまり, セクション境界の選択と並列成分の多さは相互に 1 つの並列成分の計算量に大きく影響を与え, これらの

表 3: 符号長 $n = 64$ の RM 符号の復号演算量

(n, k) 符号	MLU 法				提案法
	8-sec	16-sec	32-sec	64-sec	
(64,7)RM _{6,1}	1,132	364	367	368	342
(64,22)RM _{6,2}	3,022	5,062	7,762	13,544	2,816
(64,42)RM _{6,3}	69,758	81,890	75,498	100,096	62,642
(64,57)RM _{6,4}	42,656	8,560	4,868	4,570	4,247

表 4: 符号長 $n = 64$ の RM 符号の並列成分数 ρ'

(n, k) 符号	MLU 法				提案法
	8-sec	16-sec	32-sec	64-sec	
(64,7)RM _{6,1}	3	3	3	3	3
(64,22)RM _{6,2}	6	5	5	5	6
(64,42)RM _{6,3}	3	4	3	3	3
(64,57)RM _{6,4}	1	1	1	1	1

選び方の重要性が大きいことを示唆している。

本稿の提案法では、トレリスの動的性について注目し、 $(\Delta f_i, \Delta p_i) = (1, 1)$ となる時点 i までセクション構成をして、並列数を増やすステップにもどっている。これは定理 2 より、これらの時点 i に必ずセクションの境界を持つことが保証されているからである。逆に、 $(\Delta f_i, \Delta p_i) \neq (1, 1)$ の場合は時点 i をセクション境界とするのが良いかどうかの判断を行うことができない。従ってこのような時点をセクション境界として並列成分を抽出しても、結果として得られるトレリスが復号時間を短縮できる可能性は低い。

また、今回は MLU 法の結果と比較するために符号長が 2 のべき乗となる符号を計算したが、提案方法は符号長が素数のブロック符号にも適用できる。

さらに、1 並列成分の復号に必要な計算量を低減するために、提案アルゴリズムで T_n^{\min} まで求めて並列成分を抽出した後、 T_n^{\min} が持つセクション境界のみについて、step6), step7) を繰り返し行うという手法も考えられる。

6 まとめと今後の課題

本稿では並列成分を多く持ち、セクション構成を行いつつ非最簡トレリスを構成法する手法を提案した。またその結果、復号時間を短縮することが可能であることを示した。

復号器を装置化する際には、最簡トレリスのみでなく、並列成分を並列処理するための、コセット分割を利用した非最簡トレリスの重要性が、今後ますます増してくると思われる。本稿の提案アルゴリズムは装置を設計する際に、選択の幅を大きく広げることができると考えられる。

今後の課題として挙げられるのは、1 並列成分に対する目的関数を最小とするようなトレリス構成法の開発である。提案アルゴリズムでは、並列成分を抽出するためにはセクションの境界の配置に制限が伴う。しかしセクション構成が、1 並列成分の復号の計算量低減の大きな鍵を握っているため、目的関数を最小とするようなより適切な時点にセクション境界を配置させつつ、並列成分をもたせる均衡点を追及する必要がある。

また、本稿の提案方法は任意の線形ブロック符号に適用できるが、実際にどのような性質の符号に関して有効に働くかを明らかにする必要がある。その時、どこまで計算量が低減できるかについて、その限界を示すことが望ましい。

SCP 作成の操作や並列性の次元数算出は TOG 行列から容易に求めることができるので、提案アルゴリズムに必要な計算量はセクション構成に必要な計算量 $O(n^2)$ が大部分を占める。しかし符号長、情報記号数の増加とともに新たな SCP 作成、並列成分抽出のステップへ戻る回数が増える。そこでこれらも考慮した提案アルゴリズムに必要な計算量の上限を示すことも今後の課題である。

謝辞: 著者の一人八木は、研究を進めるにあたり、日頃よりご討論、ご助言を頂いている早稲田大学 平澤研究室の各氏に深く感謝する。また、本研究の一部は文部省科学研究費 (奨励研究 A11750338)、早稲田大学特定課題研究助成費 99A-205 の助成によるものである。

参考文献

- [1] H.T.Moorthy, S.Lin, and G.T.Uehara. "Good trellises for IC implementation of Viterbi decoders for linear block codes," *IEEE Trans. Commun.*, vol.45, no.1, pp.52-63, Jan. 1997.
- [2] D.J.Muder, "Minimal trellises for block codes," *IEEE Trans. Infom. Theory*, vol.34, pp.1049-1053, Sept. 1988.
- [3] T.Kasami, T.Takata, T.Fujiwara, and S.Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes," *IEEE Trans. Infom. Theory*, vol.39, pp.242-245, 1996.
- [4] R.J.McEliece. "On the BCJR trellis for linear block codes," *IEEE Trans. Infom. Theory*, vol.42, no.4, pp.1072-1092, July 1996.
- [5] A.Lafourcade, and A.Vardy, "Lower bounds on trellis complexity of block codes," *IEEE Trans. Infom. Theory*, vol.41, no.6, pp.1938-1945, Nov. 1995.
- [6] A.Lafourcade, and A.Vardy, "Optimal sectionalization of a trellis," *IEEE Trans. Infom. Theory*, vol.42, no.3, pp.689-703, May 1996.
- [7] T.Fujiwara, H.Yamamoto, T.Kasami, and S.Lin, "A trellis-based recursive maximum-likelihood decoding algorithm for binary linear block codes," *IEEE Trans. Infom. Theory*, vol.44, no.2, pp.714-729, Mar. 1998.
- [8] A.Vardy, and Y.Be'ery, "Maximum-likelihood soft decision decoding of BCH codes," *IEEE Trans. Infom. Theory*, vol.40, pp.546-557 Mar. 1994.
- [9] J.K.Wolf, "Efficient maximum likelihood decoding of linear block codes," *IEEE Trans. Infom. Theory*, vol.IT-24, pp.76-80, Jan. 1978.
- [10] M.Bóo, F.Argüello, J.D.Bruguera, R.Doallo, and E.L.Zapata, "High-performance VLSI architecture for the Viterbi algorithm," *IEEE Trans. Commun.*, vol.45, no.2, no.2, pp.168-173, Feb. 1997.
- [11] S.Lin, T.Kasami, T.Fujiwara, and M.Fossorier, *Trellises and Trellis-Based Decoding Algorithms for Linear Block Codes*, Kluwer Academic Publishers, MA, 1998.
- [12] 今井 秀樹, 符号理論, 電子通信情報学会, 1990.
- [13] 平澤 茂一, 西島 利尚, 符号理論入門, 情報数理シリーズ, 培風館, 1999.