$SN$比の基準で性能の良い符号化率 $b/n$ の
たたみ込み符号について

福島 荘之介† 山本 博資††

†電気通信大学 大学院情報システム学研究科
〒182 東京都調布市講谷丘1-5-1
††東京大学 工学部数理工学科
〒113 東京都文京区本郷7-3-1

あらまし  符号化率 $1/n$ のたたみ込み符号に対する誤り率の Transfer function bounds を数値計算するアルゴリズム [5] を符号化率 $b/n$ のたたみ込み符号に拡張したアルゴリズムを示す。また、このアルゴリズムと Dijkstra の最小自体距離を求めるアルゴリズムを用いて計算機探索を行い、$(K,b/n) = (2,2/3), (3,2/3), (2,2/5)$ のたたみ込み符号について性能の良い符号を求める。具体的には、加法的	
白色ガウス雑音通路においてビット誤り率 $(10^{-4}, 10^{-5})$ を達成するのに必要な $SN$比が最小となるような符号を探索している。

和文キーワード たたみ込み符号

Good Convolutional Codes for rate b/n based on the criterion of SNR
Sounosuke Fukushima† Hirosuke Yamamoto††

†Graduate School of Information Systems, The University of Electro-Communications
1-5-1 Chofu-gaoka, Chofu-shi, Tokyo 182, Japan

††Department of Mathematical Engineering and Information Physics, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan

Abstract
The numerical calculation algorithm for the transfer function bounds, which was originally described for $1/n$ convolutional codes, is extended to $b/n$ convolutional codes. Furthermore, by using this algorithm and Dijkstra's algorithm to obtain the minimum free distance of a convolutional code, good convolutional codes are searched based on the criterion of minimizing the required signal-noise-ratio for a given bit error probability. Several new good convolutional codes are found for $(K,n) = (2,2/3), (3,2/3), (2,2/5)$, where $K$ is the constraint length of a convolutional code.

英文 key words convolutional code
1  はじめに

たたみ込み符号は、Victor ファデ サプロジズムを利用することにより最尤復号を行うことができ、拘束長が長くなるにつれて誤り訂正能力が高くなることが知られている。しかし、誤り訂正能力が高いたたみ込み符号を効率良く構成する方法は知られておらず、そのため、性能の良い符号は計算機探索により求められている。これは、基本的には与えられた符号化率（rate）と拘束長を持つ生成行列の全数探索である。

従来、誤り訂正能力の高いたたみ込み符号の探索には、符号間の最小自由距離が最大になるものという評価基準が用いられてきた [1][2][3]。しかし、たたみ込み符号の誤り率は式①のみにより決まるわけではないので、t 番目のたたみ込み符号が必ずしも最小の誤り率を達成することは限らない。Lee [4] は、Transfer function bounds を用いて、与えられた誤り率を達成するのに必要な S/N 比が最小となる新しいたたみ込み符号を符号化率 1/n に対して求めた。しかしながら、Lee の探索アルゴリズムは時間が必要になるため部分探索しか行っておらず、知られていない性能の良いたたみ込み符号が存在する可能性がある。

本稿では、まず符号化率 1/n のたたみ込み符号に対し Transfer function bounds を数値計算するアルゴリズム [5] を、符号化率 b/n の 2 倍たたみ込み符号に拡張したアルゴリズムを示す。さらにこのアルゴリズムと Dijkstra 法 [6] を用いた最小自由距離の計算アルゴリズムを使用して、Lee と同様の探索を符号化率 1/n および b/n のたたみ込み符号に対して行った。本探索によって新たに幾つかの性能の良いたたみ込み符号を見つけることができた。

2 符号化率 b/n のたたみ込み符号に対する Transfer function bounds を数値計算するアルゴリズム

たたみ込み符号の誤り率を求める方法として、伝達関数を用いた誤り上界式がある。この伝達関数を求めるには、符号化率が 1/n の場合、2^n-1 個の状態変数をもつ状態方程式を解く必要がある。この方法は、k が大きくなると状態変数の数が指数的に増大するため、一般に誤り上界は数値計算によって求められる。Yamamoto-Fujisawa [6] は、この数値計算を行った有効性の一例と考え、累積法 (Power Method) と状態変数の選択を組み合わせて精度良く計算するアルゴリズムを提案し、符号化率 1/n のたたみ込み符号の場合に対して詳しいアルゴリズムの記述をしている。しかし、文献 [5] ではそのアルゴリズムが一般の符号化率 b/n のたたみ込み符号に拡張可能であると指摘しているものの、詳しいアルゴリズムは記載されていない。そこで本稿ではまず、符号化率が b/n のたたみ込み符号にも適用できるように一般化した Transfer function bounds の数値計算アルゴリズムを示す。

図 1 に示すような符号化率が b/n のたたみ込み符号に対して、遅延素子の内容をベクトル [u_k−1, u_k−2, ..., u_k] に入力し、値を定義する。ここで、ベクトル u_k は、

\[ u_k = (u_{k,0}, u_{k,1}, ..., u_{k,b-1})^T \]

とし、T は転置を示す。符号化器の生成行列を、

\[ G_b = \begin{pmatrix} g_{b,0} & g_{b,1} & \cdots & g_{b,b-1} \\ g_{b,1} & g_{b,2} & \cdots & g_{b,b-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{b,K-1} & g_{b,K-2} & \cdots & g_{b,0} \end{pmatrix} \]

と定義すれば、出力 v1, v2, ..., v_n の各ビットは次式となる。ただし、\( n \) および \( b \) はモデル 2 加算を示す。

\[ v_j = \sum_{k=0}^{K-1} u_{k,b} g_{k,j} \]

\[ v_j = \sum_{k=0}^{K-1} u_{k,b} g_{k,j} \quad j = 1, 2, ..., n \]

したがって、出力ベクトル \( v = (v_1, v_2, ..., v_n) \) のポントは、\( u_0, u_1, ..., u_{K-1} \) の関数となり

\[ w(u_0, u_1, ..., u_{K-1}) = u_0 + u_1 + \cdots + u_n \]

\[ w(u_0, u_1, ..., u_{K-1}, u_n) = \sum_{k=0}^{K-1} \sum_{j=0}^{b-1} u_{k,b} g_{k,j} \]

とされる。

次に符号化器の状態遷移を考える。状態変数 \( u_0, u_1, u_2, ..., u_n \) に対応したノードを \( \xi_{u_0, u_1, u_2, ..., u_n} \) で表すと、\( \xi_{u_0, u_1, u_2, ..., u_n} \) は、\( \xi_{u_0, u_1, u_2, ..., u_n} \) に接続されている。したがって、整数 \( j = (j_0, j_1, ..., j_n) \) と 2 進数表現し、ベクトル \( j = (j_0, j_1, ..., j_n)^T \) と定義すれば、状態変数には

\[ \xi_{u_0, u_1, u_2, ..., u_n} = \sum_{j=0}^{2^n-1} D[j] \xi_{u_0, u_1, u_2, ..., u_n} \]

\[ \xi_{u_0, u_1, u_2, ..., u_n} = \sum_{j=0}^{2^n-1} D[j] \xi_{u_0, u_1, u_2, ..., u_n} \]

関係がある。また、終了ノード \( T(D) \) は、\( \xi_{j_0, j_1, ..., j_n} \) にベクトル \( (0, 0, 0)^T \) が入力された場合であり、

\[ T(D) = \sum_{j=0}^{2^n-1} D[j] \xi_{j_0, j_1, ..., j_n} \]

となる。
図1 異常化速度b/nのたたみ込み符号器

図2 状態遷移図

$
\frac{\partial T(D,I)}{\partial I} I=1$

を求めめるアルゴリズムを考える。開始ノードからノード$a$および$b$への伝達関数をそれぞれ

$T_a(D,I), T_b(D,I)$として、$\xi, \xi_a, \xi_b, \xi_c$を

$\xi = T_a(D,I) I=1$,  (7)
$\xi_a = T_a(D,I) I=1$,  (8)
$\xi_b = \frac{\partial T_a(D,I)}{\partial I} I=1$,  (9)
$\xi_c = \frac{\partial T_a(D,I)}{\partial I} I=1$.  (10)

と定義する。$\xi$と$\xi_b$の関係は、

$\xi_a = D^S I^S \xi_a$,  (11)

となる。また、$\xi_a$と$\xi_c$の関係は、

$\xi_a = \frac{\partial T_a(D,I)}{\partial I} I=1$
$= \frac{\partial D^S I^S T_a(D,I)}{\partial I} I=1$
$= D^S I^S \frac{\partial T_a(D,I)}{\partial I} I=1$
$= D^S \xi_a + N D^S \xi_a$.  (12)

となる。したがって、符号化率が$b/n$の場合には

$\xi_a = D^S \xi_a + D^S j u_1 \cdots u_n I=1$
$= \sum_{j=0}^{n-1} D^S j u_1 \cdots u_n I=1$
$= \sum_{j=0}^{n-1} D^S j u_1 \cdots u_n I=1$
$= \sum_{j=0}^{n-1} D^S j u_1 \cdots u_n I=1$

の関係が成り立つ。符号化率が$1/n$の場合$N=1$であるが、符号化率が$b/n$の場合は、$N$は入力$b$のハミング重みとなる。以上から符号化率$b/n$のたたみ込み符号において、$T(D,I)$と$
\frac{\partial T(D,I)}{\partial I} I=1$

を求めるアルゴリズムを次のように記述することができる。

アルゴリズム1 (Rates b/n Element wise

Power Method for $T(D)$ and $\frac{\partial T(D,I)}{\partial I} I=1$

1. (Initialization)

$\xi_a^{(0)} \cdots 0 := 1$, and let other $\xi_u^{(0)} \cdots 0$ be zero.

Let all $\xi_u^{(0)} \cdots 0$ be zero.

$l := 0$.

2. repeat

(a) $l := l + 1$.

(b) $\xi_a^{(l)} \cdots 0 := \xi_a^{(l-1)} \cdots 0$.

(c) for $i := 1$ to $2^{K-1} - 1$ do

(i) $u_{K-2} u_{K-3} \cdots u_0 := (i)_{2^{K-1}}$.

(ii) $\xi_a^{(l)} u_{K-2} \cdots u_0 := D^S (\xi_a^{(l)} u_{K-2} \cdots u_0) I=1$
$= \sum_{j=0}^{n-1} D^S (\xi_a^{(l)} u_{K-2} \cdots u_0) I=1$
$= \sum_{j=0}^{n-1} D^S (\xi_a^{(l)} u_{K-2} \cdots u_0) I=1$
$= \sum_{j=0}^{n-1} D^S (\xi_a^{(l)} u_{K-2} \cdots u_0) I=1$
$= \sum_{j=0}^{n-1} D^S (\xi_a^{(l)} u_{K-2} \cdots u_0) I=1$

until sufficient accuracy is achieved.

3. $T(D) := \sum_{i=0}^{2^{K-1}-1} D^S(j_0 \cdots 0)\xi_a^{(i)} j_1 \cdots 0$. 

$\frac{\partial T(D,I)}{\partial I} I=1 := \sum_{j=0}^{n-1} D^S(j_0 \cdots 0)\xi_a^{(i)} j_1 \cdots 0$. 

—— 73 ——
ただし、(c) の (ii) の $2^{K-1}$ は、整数 1 の $b(K-1)$ 枚の 2 进数表現を表す。
符号化率 $1/n$ の場合と同様に、2 (c) ii, iii, を以下のように変形とすることで収束の速度を上げることができる。

(ii) $\xi_{u_{K-2}u_{K-3} \cdots u_0} := D^{(2)}(U_{K-2}U_{K-3} \cdots U_0)_{j=1}^{2^{K-1}} \sum_{j=1}^{2^{K-1}} D^{(2)}(U_{K-2}U_{K-3} \cdots U_0)_{j=1}^{2^{K-1}}$

(iii) $\gamma_{u_{K-2}u_{K-3} \cdots u_0} := D^{(2)}(U_{K-2}U_{K-3} \cdots U_0)_{j=1}^{2^{K-1}} \sum_{j=1}^{2^{K-1}} D^{(2)}(U_{K-2}U_{K-3} \cdots U_0)_{j=1}^{2^{K-1}}$

3 設計ビット誤り率において、$E_b/N_0$ を小さくする生成行列の計算機探索

符号化率 $b/n$ のたたみ込み符号の事象誤り率 $P_b$ とビット誤り率 $P_e$ の誤り上界式は、加法的白色ガウスノイズ信道 (AWGN) において、

$$P_b \leq Q \left( \frac{2d_i E_b}{N_0} \right) e^{d_i E_b / N_0} T(D, F)_{1=1, D_e=E_b/N_0}$$

(iii) $G$ がカスタストロフィックかどうか判定。カスタストロフィックなら (ii) に戻る。

(iv) $G$ の $d_f$ を求め、最小の $d_f$ と比較。最小の $d_f$ より小さいければ (ii) に戻る。

(v) 2 分探索法により、$E_b/N_0$ の最小値と最大値を与え、設計ビット誤り率に収束する $E_b/N_0$ を求める。

(vi) この $E_b/N_0$ が以前のものより小さければ、$G$ を保存。

(vii) (ii) に戻る。

カスタストロフィックな生成行列とは、有限値の誤りであっても情報系列に無限値の誤りが生じてしまう「誤り伝搬」を生じ得る生成行列である。この判定において、以下の定理を用いる。

定理 1 [7, 定理 11.1] 符号化率 $b/n$ のたたみ込み符号の生成行列 $G$ が基本生成行列である必要十分条件は、$G$ の $r$ 階の $b \times b$ 小行列の行列式 $\Delta_r$ で $\Delta_r \neq 0$ となるものの最大公約数が定数となることである。■

定理 2 [7, 定理 11.2] 生成行列がカスタストロフィックにならないための必要十分条件は、その生成行列が基本生成行列であることがある。■

(iv) の $d_f$ の最小値は、無駄な探索を減らすためのもので、従来知られている最大の $d_f[3]$ から 1 を減じた値を用いた。

4 探索結果

と考えられる。前節の手法は、特定の \( d_f \) 値以上の全数探索であるため部分探索で探索されなかった符号が見つかなかったため、この結果を表1に示す。

次に符号化率が \( b/n \) のたまに符号化を \( (K, b/n) \) が(2,2/3),(3,2/3),(2,5/6)で探索した。この結果、同じ最大誤り率 \( P_e \) を達成するのに必要な \( E_b/N_0 \) が、従来の符号[3]よりも小さい良い符号を求めることができた。探索によって求めた符号は、従来の符号と比べて0.1～0.3dB 利得が高いため、この結果を従来の符号[3]と比較し、表2に示す。また、新しい符号の \( E_b/N_0 \) 対ビット誤り率のグラフを図3～図5に示す。

<table>
<thead>
<tr>
<th>( K )</th>
<th>( d_f )</th>
<th>( G ) (total)</th>
<th>( E_b/N_0 ) 1.0E-6</th>
<th>( E_b/N_0 ) 1.0E-5</th>
<th>( E_b/N_0 ) 1.0E-4</th>
<th>( E_b/N_0 ) 1.0E-3</th>
<th>( E_b/N_0 ) 1.0E-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>7 5</td>
<td>6.796</td>
<td>4.427</td>
<td>5</td>
<td>1.0, 2* (6), 3* (5)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>7 7 5</td>
<td>6.736</td>
<td>4.410</td>
<td>5</td>
<td>1.0, 2* (6), 3* (5)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3 1 5 5</td>
<td>7 5 5</td>
<td>6.093</td>
<td>4.345</td>
<td>10</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7 7 5 5</td>
<td>7 5 5</td>
<td>6.768</td>
<td>4.027</td>
<td>10</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7 7 7 5 5</td>
<td>7 7 5 5</td>
<td>6.899</td>
<td>4.015</td>
<td>15</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7 7 7 7 5 5</td>
<td>7 7 7 5 5</td>
<td>6.796</td>
<td>4.027</td>
<td>15</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7 7 7 7 7 5 5</td>
<td>7 7 7 7 5 5</td>
<td>6.805</td>
<td>3.992</td>
<td>17</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7 7 7 7 7 7 5 5</td>
<td>7 7 7 7 7 5 5</td>
<td>6.285</td>
<td>3.981</td>
<td>21</td>
<td>5, 6, 7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7 7 7 7 7 7 7 5 5</td>
<td>7 7 7 7 7 7 5 5</td>
<td>6.385</td>
<td>3.981</td>
<td>21</td>
<td>5, 6, 7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7 7 7 7 7 7 7 7 5 5</td>
<td>7 7 7 7 7 7 7 5 5</td>
<td>6.385</td>
<td>3.981</td>
<td>21</td>
<td>5, 6, 7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7 7 7 7 7 7 7 7 7 5 5</td>
<td>7 7 7 7 7 7 7 7 5 5</td>
<td>6.385</td>
<td>3.981</td>
<td>21</td>
<td>5, 6, 7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7 7 7 7 7 7 7 7 7 7 5 5</td>
<td>7 7 7 7 7 7 7 7 5 5</td>
<td>6.385</td>
<td>3.981</td>
<td>21</td>
<td>5, 6, 7</td>
<td></td>
</tr>
</tbody>
</table>

1. *a*是Minimizes required \( E_b/N_0 \) for desired BER ≤ 1.0E-6
2. *b*是Minimizes required \( E_b/N_0 \) for desired BER ≤ 1.0E-5
3. Time is CPU time on EWS(2MHz).

- 図3 新しいたまに符号化符号 \( K = 2, b/n = 2/3 \) の \( E_b/N_0 \) 対ビット誤り率
- K=2, b/n=2/3
  - Found by Daut [1982]

- 図4 新しいたまに符号化符号 \( K = 2, b/n = 2/5 \) の \( E_b/N_0 \) 対ビット誤り率
- K=2, b/n=2/5
  - Found by Daut [1982]
参考文献


