ラン長制約符号の組織的構成法 A Systematic Construction Method of Run Length Limited Codes

池谷 章, 小林 正明, 田崎 三郎*

松下電器産業株式会社, *愛媛大学工学部

Akira Iketani, Masaaki Kobayashi, Saburo Tazaki*

Matsushita Electric Industrial Co., Ltd. * Ehime University

Abstract

This paper describes a systematic construction method of Run Length Limited (RLL) codes for the purpose of getting higher recording density realized. The continuous number of a binary value in the RLL code is restricted to not less than d and not greater than k in NRZL notation by converting a data word of m -bit to a codeword of n -bit. This method comprises; a) equations for calculating the number of the codewords; b) rules for selecting codewords; c) rules for concatenating the codewords. Various kinds of fixed length or variable length RLL codes have been gotten under the conditions of $d=1, 2, \dots, 6, k=d+1, \dots, 15$ in NRZL expression, and $n=d+1, \dots, 32$.

1. まえがき

デジタル情報の記録密度を向上させる技術の一つとして、デジタル記録符号があり、より多くの情報を記録するために様々なデジタル記録符号が提案されている^{[1]~[15]}。デジタル記録符号としては、一般的に、記録ビット列における“0”及び“1”の連続ビット数を、NRZL表記の d 以上 k 以下に制約するRLL (Run Length Limited) 符号が用いられる。

RLL符号は、 m ビットのデータ語を n ビットの符号語に変換することで実現されるが、記録メディアの特性は一様でなく、それぞれのメディアに適したRLL符号について、大きく以下の4つの観点から検討が行われている。

- 1) 検出窓幅(T_w)が広いこと。
- 2) 同一2進値の最小連続ビット数(d)が大きいこと。
- 3) 同一2進値の最大連続ビット数(k)が小さいこと。
- 4) ハードウェア規模が小さいこと。

従来、RLL符号は各メディアごとに個別に開発されてきたが、上記4つの観点から、従来と同等以上の性能を有するRLL符号を組織的に構成で

きれば、今後予想される新たな記録メディアに適したRLL符号の開発が容易になる。

本稿では、新たな記録メディアに適したRLL符号を迅速に提供することを目的とし、RLL符号を組織的に構成する一手法として、以下に示す3種類の検討を行う。

- 任意の符号語長 n に対して、任意の d, k を満たす符号語数を求めるための計算式を導出する。
- 符号語同士の接続によっても上記 d, k 制約を満たし、かつ、使用可能な符号語数を選択するための符号語の接続端条件を求める。
- 具体的な固定長RLL符号と可変長RLL符号を求める。

2. 符号語数計算式の導出

本章では、NRZL則に従う n ビットよりなる 2^n 個の符号語のうち、“0”及び“1”の連続ビット数が d 以上 k 以下になる符号語数を求める計算式を導出する。

すべての2進符号語は、図1に示す2種類に大別できる。つまり、“1”で始まり“1”で終わる符号

(I)

(II)

図1 符号語の区分

Fig. 1 Classification of codewords

語(I)と, “1”で始まり“0”で終わる符号語(II)である. “0”で始まる符号語は, 上記(I)及び(II)の符号語におけるすべてのビットについて, “1”を“0”, “0”を“1”にすべて反転させた裏パターンに等しい. 従って, “1”で始まる符号語数を求められれば, 全符号語数はこれを2倍すれば得られる. 以下では, “1”で始まる符号語を例にとつて示す.

まず, $n=2N$ ($N=1, 2, \dots$)ビットの符号語について, “1”と“0”の個数が等しい場合の符号語数の計算式を導出する. この場合, 符号語における“1”及び“0”の個数はそれぞれ N である. ここで, N ビットの同一2進値を, d ビット以上 k ビット以下の小ブロックに分割する場合のブロック数を求める. この分割において, l ビット ($d \leq l \leq k$) の小ブロックの個数を n_l とすると, 式(1)が成立する.

$$\sum_{l=d}^k l \cdot n_l = N \quad (1)$$

2.1 d, k 制約を満たす“1”又は“0”の分割数

式(1)を満たすある小ブロックの組を $\{n_l\}$ とし, その総数を L_{max} と置く. 更に, 第 i 番目 ($i=1, 2, \dots, L_{max}$) の $\{n_l\}$ を $\{n_l\}_i$ と表記するものとする. このとき, $\{n_l\}_i$ に含まれる小ブロック数を $K_N(i)$ で表すと, $K_N(i)$ は式(2)で与えられ, $\{n_l\}_i$ によって N ビットを分割する場合の数 T_{Ni} は, 次式(3)で与えられる.

$$K_N(i) = \sum_{l=d}^k n_l, \quad n_l \in \{n_l\}_i \quad (2)$$

$$T_{Ni} = \frac{K_N(i)!}{n_d! n_{d+1}! \dots n_k!} \quad (3)$$

例えば, $N=16, d=2, k=4$ としたとき, 16ビットの“1”の分割例を図2に示す. この場合の“1”の分割数は $K_{16}(i)=6, n_2=3, n_3=2, n_4=1$ であるから, 式(3)により $T_{16i}=60$ 通りのパターンが存在する.

図2 16ビットの“1”の分割例($d=2, k=4$)
 Fig. 2 An Example of Partitioning
 16 Consecutive “1”s ($d=2, k=4$)

2.2 d, k 制約を満たす符号語数の計算

図1から分かるように, (I)の構成をとる符号語における“1”に関する小ブロック数は, “0”に関する小ブロック数よりも一つだけ多く, (II)の構成をとる“1”及び“0”に関する小ブロック数は等しい.

ここで, 小ブロック数 $K_N(i)$ を改めて j ($K_{Nmin} \leq j \leq K_{Nmax}$) とおき, その総数を $S_N(j)$ で表すと, (I)の構成をとる符号語数 $N_{Co}(I)$ は式(4), (II)の構成をとる符号語数 $N_{Co}(II)$ は式(5)で与えられる. なお, $S_N(j)$ は式(6)で与えられ, 式(6)における A_j は, $K_N(i)=j$ になるすべての $\{n_l\}_i$ からなる集合を表す.

$$N_{Co}(I) = \sum_{j=K_{Nmin}}^{K_{Nmax}} S_N(j) \cdot S_N(j-1) \quad (4)$$

$$N_{Co}(II) = \sum_{j=K_{Nmin}}^{K_{Nmax}} S_N(j) \cdot S_N(j) \quad (5)$$

$$S_N(j) = \sum_{\{n_l\}_i \in A_j} T_{Ni} \quad (6)$$

ただし, 明らかに $S_N(K_{Nmin}-1)=0$ である.

以上の結果により, “0”で始まる符号語数をも考慮すると, $2N$ ビットの符号語のうち, “0”及び“1”の連続ビット数が d 以上 k 以下であり, かつ, “1”と“0”の個数が等しい符号の符号語数 N_{Co} は次式で与えられる. 式(7)により, “0”と“1”の個数が等しい場合の d, k 制約された符号語数を求めることができる.

$$\begin{aligned} N_{Co} &= 2 \cdot \{N_{Co}(I) + N_{Co}(II)\} \\ &= 2 \cdot \sum_{j=K_{Nmin}}^{K_{Nmax}} S_N(j) \cdot \{S_N(j-1) + S_N(j)\} \end{aligned} \quad (7)$$

2.3 “0”と“1”の個数が等しくない場合の d, k 制約された符号語数の導出

この場合, n ビットの符号語のうち“1”を N' ビット, “0”を $n-N'$ ビットとすることで, “0”と“1”の個数が等しい場合と同様に計算できる. ただし, N' は式(8)の範囲の値である.

$$0 \leq N' \leq n \quad (8)$$

式(8)を満たす任意の N' に対して, $n-N'$ は一意に定まるので, 式(1)~(4)から, $S_{N'}(j), S_{n-N'}(j)$ が求まる. ゆえに, n ビットの符号語のうち, “1”及び“0”の連続ビット数が d 以上 k 以下の符号語数 $N_c(n)$ は, “0”で始まる符号語をも含めると, 式(9)及び(10)により与えられる.

$$N_c(n) = 2 \sum_{N=0}^n \left[\sum_{j=j_{\min}}^{j_{\max}} S_{N'}(j) \cdot \{S_{n-N'}(j-1) + S_{n-N'}(j)\} \right] \quad (9)$$

$$\begin{aligned} j_{\min} &= \min \{K_{N'_{\max}}, K_{(n-N')_{\max}}\}, \\ j_{\max} &= \max \{K_{N'_{\min}}, K_{(n-N')_{\min}}\} \end{aligned} \quad (10)$$

3. 符号語の接続条件

前節では、符号語の接続条件を考慮せずに、 n ビットの符号語内のみで d, k 制約を満たす符号語数を求めた。本節では、符号語同士を接続する場合にも d, k 制約を満たし、かつ、有効符号語数が最大になる符号語同士の接続条件がそれぞれ異なる(i) $d = 1$ の場合、(ii) $d \geq 2$ の場合についてそれぞれ示す。

なお、(i) 及び(ii) の場合共に、符号語を図3に示すL, B, Rの3ブロックに分けて考える。つまり、 l ビットの同一2進値(TB)で始まる符号語の始端Lブロックと、 r ビットの同一2進値(LB)で終わる符号語の終端Rブロック、L及びRブロックに挟まれた d, k 制約を満たす $b(=n-l-r)$ ビットの符号語の中間Bブロックである。また、図4に2つの符号語の接続図を示す。

図3 符号語の構造

Fig.3 Structure of a codeword

3.1 $d=1$ の場合の接続条件

この場合、孤立する“0”及び“1”も d 制約を満たすから、符号語同士の接続による d 制約違反は起こり得ず、考慮の必要はない。また、 k 制約を満たす符号語同士の接続に関して、先行符号語のRブロックが $r_1(\leq k)$ ビットの2進値(LB)、先行符号語に続く後続符号語のLブロックが $l_2(\leq k)$ ビットの2進値(TB)とすると、LBとTBが互いに異なる2進値の場合には k 制約を満たす。従って、符号語同士を接続しても k 制約を満たすための条件は、先行符号語のLBと後続符号語のTBが同じ2進値の場合に、有効符号語数を最大にする条件を求めればよい。

図4 符号語の接続

Fig. 4 Concatenation of codewords

各符号語はいずれも先行、後続双方になり得る。従って、任意の符号語を接続しても k 制約を満たすための条件は、図3における符号語のLブロックを構成するビット数 l と、Rブロックを構成するビット数 r に関する条件に一般化できる。従って、これらの l と r に関して、まず、式(11)が成り立たなければならない。

$$r + l \leq k \quad (11)$$

ここで、 l 及び r をそれぞれ式(12)で示す範囲にあるものとする。

$$1 \leq l \leq x, 1 \leq r \leq k - x \quad (12)$$

ただし、

$$1 \leq x \leq k - 1 \quad (13)$$

である。このときの条件を満たす符号語数 C は、式(9)を用いて式(14)となる。

$$C = \sum_{l=1}^x \sum_{r=1}^{k-x} N_c(n-l-r) \quad (14)$$

式(14)は、図3におけるBブロックが $b=n-l-r$ ビットであり、Bブロック内では完全に d, k 制約を満たすから、 b ビットで d, k 制約を満たす符号語数は、一組の l 及び r に対して、 $N_c(b)$ となる。また、式(14)には、 $b=0$ かつL及びRブロックを構成する2進値が同一の符号語、及び、Bブロックが同一の2進値のみで構成され、かつ、L及びRブロックは共にBブロックとは異なる同一の2進値よりなる符号語も含まれる。

ところで、L及びRブロックに関しては、式(12)及び式(13)、Bブロックに関して完全に d, k 制約を満たすように制約を課した符号語については、符号語同士の接続による k 制約違反は起こらない。したがって、これらの符号語は単独でデータ語に対応できる。

一方、マップを切り換え法を用いて、複数の符号語を一つのデータ語に対応させることも可能である。すなわち、式(12)で制約されない接続端を有する符号語を使用することである。この場合問題になるのは、先行符号語のRブロックと後続符

号語のLブロックが同一の2進値で構成されている場合の k 制約違反である。

これは、同一の2進値であるがゆえの問題であるから、先行符号語のRブロックと後続符号語のLブロックが同一2進値であれば、後続符号語を裏パターンに反転させればよい。すなわち、式(15)に示す接続端を有する符号語は、その裏パターンと一対でデータ語に対応させ、上述の接続則を適用することにより、符号語として用いることができる。

$$x+1 \leq l \leq k, 1 \leq r \leq k-x \quad (15)$$

以上により、マップ切り換え法で条件を満たす符号語数 $C_1(x)$ は式(14)を含み、式(16)で与えられる。

$$\begin{aligned} C_1(x) &= \sum_{l=1}^x \sum_{r=1}^{k-x} N_c(n-l-r) + \frac{1}{2} \sum_{l=x+1}^k \sum_{r=1}^{k-x} N_c(n-l-r) \\ &= \frac{1}{2} \sum_{r=1}^{k-x} \left\{ \sum_{l=1}^x N_c(n-l-r) + \sum_{l=r}^k N_c(n-l-r) \right\} + Y \end{aligned} \quad (16)$$

ただし、式(16)において、 Y は式(17)で与えられる。

$$\begin{cases} Y=0: & k < n+x \\ Y=1: & k \geq n+x \end{cases} \quad (17)$$

式(17)は、 n ビットすべてが同一の2進値よりなる符号語を使用する場合、同一2進値が連続する最大値は $n+x$ ビットとなることによる。

以上示した符号語の接続端条件、データ語と符号語の対応関係、及び、符号語同士の接続則をまとめると表1のようになる。なお、表1における LB は第1符号語の最終ビットを、 $F=0$ は $l \leq x$ 、 $F=1$ は $l > x$ を、“—”印は参照しなくてもよいことを表す。

表1 符号語の接続則とデータ語との対応
Table 1 Rules of codeword concatenation and dataword-codeword mapping

F	LB	接続則	データ語との対応
0	—	そのまま接続	単独
1	0 1	そのまま接続 裏パターンで接続	裏パターンと一対

表1に示す符号語同士の接続則は、次のように非常に簡潔に表現できる。

$$U_1 = F \cdot LB$$

ここに、 U_1 は後続符号語を裏パターンにする場

合に1になる値であり、“ $\cdot$ ”は論理積を表す。

3.2 $d \geq 2$ の場合の接続条件

この場合も $d = 1$ の場合と同様にして、式(18)で示す接続端条件を満たす符号語は単独でデータ語に対応させ、式(19)に示す接続端条件を満たす符号語は、裏パターンと一対でデータ語に対応させる。このときの有効符号語数 C_2' は式(20)で与えられる。

$$d \leq l \leq x, d \leq r \leq k-x \quad (18)$$

$$x+1 \leq l \leq k, d \leq r \leq k-x \quad (19)$$

$$C_2' = \sum_{r=d}^{k-x} \left\{ \sum_{l=d}^x N_c(n-l-r) + \frac{1}{2} \sum_{l=x+1}^k N_c(n-l-r) \right\} \quad (20)$$

ところで、 $d \geq 2$ の場合、Bブロックで d, k 制約を満たせば、L及びRブロックでは接続することで d 制約を満たせばよい。したがって、 d 制約に関しては、 $l+r \geq d$ が成り立てばよい。ここで、 l と r をそれぞれ式(21)で示す範囲の値とする。

$$l \geq z, r \geq d-z, 1 \leq z \leq d-1 \quad (21)$$

式(21)で示す範囲内の値を有する符号語同士の接続により、先行符号語のRブロックと後続符号語のLブロックが互いに異なる2進値のときは d 制約、同一2進値のときは k 制約が問題となる。

先行符号語のRブロックのビット数を r_1 、後続符号語のLブロックのビット数を l_2 とすると、符号語の接続端条件及び接続則は次のようにして求まる。

(I.1) 式(22)が成立する場合

$$r_1 \leq d-1, l_2 \geq z \quad (22)$$

a) 先行符号語のRブロックと後続符号語のLブロックが互いに異なる2進値の場合

この場合、接続部に d 制約違反が起こる。後続符号語を裏パターンにすることで d 制約を満たすが、 k 制約が不明。 k 制約を満たすためには、式(23)が成立しなければならない。

$$r_1 + l_2 \leq k \quad (23)$$

式(23)において、 r_1 の最大値は $d-1$ であるから、 l_2 は式(24)の範囲であればよい。

$$z \leq l_2 \leq k-d+1 \quad (24)$$

b) 先行符号語のRブロックと後続符号語のLブロックが同一の2進値である場合

$r_1 + l_2 \geq d$ であり、式(23)が成り立てば k 制約を満たす。従って、この場合の r_1 と l_2 とに関する制約は式(23)が成り立つことである。

(I.2) $r_1 \leq d-2$, $l_2 \geq d-1$, $1 \leq z \leq d-1$ の場合

この場合も、(I.1)の場合と同様にして、
 $d - z \leq r_1 \leq k - d + 1$ (25)
 を満たす。

(II) 式(26)が成り立つ場合

$$r_1 \geq d, l_2 \geq d \quad (26)$$

この場合、先行符号語のRブロックと後続符号語のLブロックが同一の2進値であれば、後続符号語を裏パターンにすれば、 d, k 制約を満たす。以上のことから、符号語の接続端条件は式(27)で表され、これらは裏パターンと一対でデータ語に対応させる。

式(27)を満たす n ビットの符号語数 C_2 は、式(28)で与えられる。

$$z \leq l \leq k - d + 1, \quad d - z \leq r \leq k - d + 1, \quad 1 \leq z \leq d - 1 \quad (27)$$

数値計算により、式(20)と式(28)による符号語数

$$C_2 = \frac{1}{2} \sum_{l=z}^{k-d+1} \sum_{r=d-z}^{k-d+1} N_c(n-l-r) \quad (28)$$

を比較すると、式(28)によって得られる符号語数のほうが多く、より効率の良いRLL符号を構成できる。したがって、 $d \geq 2$ の場合、符号語の接続端条件は式(27)で与えられ、これらの制約を課せられた符号語は裏パターンと一対で、マップ切り換えでデータ語に対応させる。

表2 固定長符号

Table 2 Examples of Fixed Length RLL Codes

n	$d=1$			$d=2$			$d=3$		
	k	m	T_w	k	m	T_w	k	m	T_w
2	2	1	0.50	—	—	—	—	—	—
3	3	2	0.67	3	1	0.33	—	—	—
4	3	3	0.75	4	2	0.50	—	—	—
5	3	4	0.80	4	2	0.40	5	1	0.20
6	4	5	0.83	4	3	0.50	6	2	0.33
7	4	6	0.86	5	4	0.57	6	2	0.29
8	4	7	0.88	8	5	0.63	7	3	0.38
9	4	8	0.89	5	5	0.56	9	4	0.44
10	4	9	0.90	6	6	0.60	7	4	0.40
11	4	10	0.91	8	7	0.64	8	5	0.46
12	4	11	0.92	5	7	0.58	7	5	0.42
13	5	12	0.92	6	8	0.62	8	6	0.46
14	5	13	0.93	8	9	0.64	7	6	0.43
15	5	14	0.93	6	9	0.60	8	7	0.47
16	5	15	0.94	6	10	0.63	12	8	0.50

表2に、上記計算式を用いて符号語数を算出して得た符号の構成例を示す。なお、 $d \geq 2$ のRLL符号に対する(I)及び(II)で示した符号語同士の接続は次のように非常に簡単な論理式で与えられる。ただし、 U_2 は後続符号語を裏パターンにするとき1となり、 LB は第1符号語の最終ビットの値、 E は先行符号語のRブロックに関して、 $r_1 < d$ のとき $E=0$, $r_1 \geq d$ のとき $E=1$ とする値、 F は後続符号語のLブロックに関して、 $l_2 < d$ のとき $F=0$, $l_2 \geq d$ のとき $F=1$ とする値である。ただし、“ $\oplus$ ”は排他的論理和、“ $\cdot$ ”は論理積、“ $\neg$ ”は否定を表す。

$$U_2 = \overline{LB \oplus E \cdot F}$$

3. 可変長符号

同一の d, k 制約及び T_w/T を、最も短い符号長で実現する方法は、可変長符号を用いることである^[14]。これは、基本ビット長 n_{min} の符号語 N_i と、ビット長が n_{min} の i ($1 \leq i \leq i_{max}$) 倍である $i \cdot n_{min}$ ビットの符号語 N_i を用い、基本のビット長 m_{min} のデータ語 M_i は、それぞれ n_i ビットの符号語に対応させる。こうすることにより、 T_w/T は式(29)のように一定となる。

$$\frac{T_w}{T} = \frac{m_i}{n_i} = \frac{i \cdot m_{min}}{i \cdot n_{min}} = \frac{m_{min}}{n_{min}} = \text{一定} \quad (29)$$

一般的に、同一の $d, k, T_w/T$ を満たす固定長、つまり $i_{max}=1$ の符号語長 L_F と、可変長 ($i_{max} \geq 2$) の符号語長の最大値 $i_{max} \cdot n_{min}$ を比較すると、式(30)の関係が成り立つことが知られている^[11]。

$$L_F \geq i_{max} \cdot n_{min} \quad (30)$$

前節までで示した符号語同士の接続則と符号語の接続端条件は、符号語長に依存しない。従って、これらは、可変長符号にもそのまま適用でき、可変長符号にも式(27)で示される接続端条件が有効である。本章では、式(27)の条件を満たす $d \geq 2$ の可変長符号を求める。

可変長符号において有効な符号語は $i \cdot n_{min}$ ビット独自の符号語である。 $i \cdot n_{min}$ ビット独自の符号語とは、 $i \cdot n_{min}$ ビットより短い符号語同士の接続によって生じない符号語をいう。

例えば、 $i=2$ の場合、 n_{min} ビットの符号語数を V_1 とするとき、 n_{min} ビットの符号語同士の、定められた接続則に従って接続した結果の $2n_{min}$ ビットのビット列の数は V_1^2 となる。この V_1^2 個の $2n_{min}$ ビットのビット列は、いずれも、L, Rブロックの接続端条件を満たし、かつ、Bブロック内では d, k 制約を満たしているから、 $2n_{min}$ ビットの V_2 個の符号語に含まれる。従って、 $2n_{min}$ ビット独自の符号語数 V_0_2 は式(31)で与えられる。

$$Vo_2 = V_2 - V_1^2 \quad (31)$$

$i \cdot n_{min}$ ビットの符号語に $i \cdot n_{min}$ ビット独自の符号語が含まれる場合、 n_{min} ビットの符号語だけでビット変換できないとき、 n_{min} ビットの整数倍符号長の独自の符号語を用いて、式(29)の条件を満たしつつ不足する符号語を補う。

m_{min}/n_{min} 変換では、 n_{min} ビット長の符号語数 V_1 は式(32)を満たさなければならない。

$$V_1 \geq 2^{m_{min}} \quad (32)$$

逆に、式(32)が成り立たなければ、 n_{min} ビットの符号語だけでは m_{min}/n_{min} 変換できないので、不足する u_i 個分を $2n_{min}$ ビット独自の符号語で補うが、 $2n_{min}$ ビット独自の符号語のみで補うには、式(33)が成り立たなければならない。ただし、 V_2 は $2n_{min}$ ビット長の符号語数、 Vo_2 は $2n_{min}$ ビット長独自の符号語数である。

$$u_i = 2^{m_{min}} - V_1 > 0 \quad (33)$$

$$Vo_2 = V_2 - V_1^2 = u_i \cdot 2^{m_{min}} \quad (34)$$

一般的に、 $i \cdot n_{min}$ ビット独自の符号語数 Vo_i は、式(35)により与えられる。

$$Vo_i = V_i - A \cdot \sum Vo_1^{q_1} \cdot Vo_2^{q_2} \cdots Vo_{i-1}^{q_{i-1}} \quad (35)$$

ただし、 V_i は $i \cdot n_{min}$ ビットの符号語数であり、また

$$A = \frac{\left(\sum_{j=1}^{i-1} q_j \right)!}{q_1! q_2! \cdots q_{i-1}!} \quad (36)$$

$$\sum_{j=1}^{i-1} j q_j = i \quad (37)$$

である。

ここで、式(38)が成り立てば、 $i \cdot n_{min}$ ビット独自の符号語を用いることで、可変長符号を構成できる。

$$u_i = u_{i-1} \cdot 2^{m_{min}} - Vo_i < 0 \quad (38)$$

式(35)及び式(38)は計算機を用いて容易に求めることができる。

ところで、式(35)、式(38)の計算の結果、可変長符号が構成できるという結論が得られた場合でも、実際には正しく復号できない場合がある。以下、その原因と対応について記す。

$i \cdot n_{min}$ ビット独自の符号語を構成する $j \cdot n_{min}$ ($1 \leq j \leq i-1$) ビットの符号語のうち、 $j \cdot n_{min}$ ビット独自の符号語に含まれないような符号語があり、それが $i \cdot n_{min}$ ビットの符号語のLブロックにもRブロックのいずれにも存在し得るとき、 $j \cdot n_{min}$ ビットの符号語と $i \cdot n_{min}$ ビットの符号語との接続によって生じるビット列の $i \cdot n_{min}$ ビットの部分に、 $i \cdot n_{min}$ ビット独自の符

号語が現れる可能性がある。このような場合には、復号誤りが生じる。

例えば、 $d=2$, $k=7$ のとき、 $n_{min}=3$ では次の3個の符号語がある。

$$A:100, B:110, C:111 \quad (39)$$

$2n_{min}=6$ ビットの符号語は12であり、そのうち6ビット独自の符号語は3個あるが、それらを構成する3ビットの符号語は式(39)の $A, B, \bar{B}, C, \bar{C}$ のいずれかである ($\bar{B}$ は B の裏パターン、 $\bar{C}$ は C の裏パターンを表す)。しかし、 $3n_{min}=9$ ビット独自の符号語を構成する6ビットの符号語の中には、6ビットの12個の符号語には含まれない111111なるビット列があり、これは、9ビットの符号語のLブロック、Rブロックの何れにも現れている。例えば、

$$\left. \begin{aligned} \alpha &= 100111111 \\ \beta &= 111111001 \end{aligned} \right\} \quad (40)$$

このとき、式(39)の A と式(40)の β との接続によって得られる12ビットのうち、先頭の9ビット、つまり、式(41)の下線を施した部分に式(40)の α が現れている。このような場合には、本来、 A に対応するデータ語が復号されなければならないのに、 α に対応するデータ語が復号され、復号誤りを起こす。

$$A\beta: \underline{100111111}001 \quad (41)$$

図5及び式(27)により、 n_{min} ビットすべて1の符号語がある場合、すべてのビットが1である $2n_{min}$ ビットは、式(42)が成り立つときには $2n_{min}$ ビット独自の符号語にはならないが、 $3n_{min}$ ビット以上の符号語のLブロックとRブロックのいずれにも現れる。

$$\left. \begin{aligned} 2 \cdot n_{min} + 2 \cdot (d-1) &> k \\ 2 \cdot n_{min} &\leq k - d + 1 \end{aligned} \right\} \quad (42)$$

一般的に、 $i \cdot n_{min}$ ビットすべてが1の符号語があり、 $(i+1) \cdot n_{min}$ ビットの”1”が $(i+1) \cdot n_{min}$ ビット独自の符号語でないのは、式(43)が成立する場合である。

$$\left. \begin{aligned} (i+1) \cdot n_{min} + 2 \cdot (d-1) &> k \\ (i+1) \cdot n_{min} &\leq k - d + 1 \end{aligned} \right\}$$

これらをまとめると、

$$k - 2d + 2 < (i+1) \cdot n_{min} \leq k - d + 1 \quad (43)$$

式(43)が成り立つときの l 及び r の制約条件は、式(44)及び式(45)となる ($(i+2) \cdot n_{min}$ ビット以上の符号語のみ適用)。

$$z \leq l \leq (i+1) \cdot n_{min} - 1, \quad d - z \leq r \leq k - d + 1 \quad (44)$$

$$z \leq l \leq k - d + 1, \quad d - z \leq r \leq (i+1) \cdot n_{min} - 1 \quad (45)$$

図5 “1”の連続個数

Fig. 5 Maximum Number of Consecutive “1”s

式(35), 式(38), 式(43), 式(44)及び式(45)を用いて求めた可変長符号を表3に示す. 表3中の T_{wmax} は T_w/T の理論限界^[11]を示す. また, 表3の注欄の“*”印は, 式(43)が成り立ち, 接続端条件を式(44)として求めたものであることを示す. なお, 以上示した可変長のRLL符号に関しても, 符号語同士の接続則は, 上記 U_2 で与えられ, 極めて容易に実現できる.

表3 可変長RLL符号の構成例

Table 3 Examples of Variable Length RLL Codes

No	d	k	$i_{\max}$	$m_{\min}$	$n_{\min}$	T_w	T_{wmax}	注
1	2	3	3	2	5	0.40	0.41	
2	2	4	2	6	11	0.55	0.55	
3	2	7	5	2	3	0.67	0.67	*
4	2	8	2	2	3	0.67	0.68	
5	3	5	6	2	5	0.40	0.41	
6	3	8	4	2	4	0.50	0.52	
7	3	12	2	8	15	0.53	0.55	
8	4	6	4	2	7	0.29	0.32	
9	4	9	5	2	5	0.40	0.43	*
10	4	10	4	3	7	0.43	0.44	*
11	5	10	4	2	6	0.33	0.36	*
12	5	13	4	3	8	0.38	0.39	*
13	5	18	6	2	5	0.40	0.40	*
14	6	12	4	2	7	0.29	0.33	*
15	6	14	2	4	13	0.31	0.34	
16	6	16	4	2	6	0.33	0.35	

NRZL表記の (d, k) はNRZI表記の $(d-1, k-1)$ と等価であるから, 表3における No.4 のRLL符号は, NRZI表記の $(1, 7)$ 符号^[12]と等価であり, No.6のRLL符号は, NRZI表記の $(2, 7)$ 符号^[13]~^[15]と等価である.

このように, 本稿で述べた構成法により, 従来それぞれ別個に開発された優れたRLL符号と等価なRLL符号を実現できることがわかる. 加えて, 従来構成が困難であった最大符号語長が20を越えるようなRLL符号についても, 符号長が短い場合

と同一の方法で, RLL符号を構成できる.

なお, 本稿で示した可変長RLL符号の場合, $i \cdot n_{\min}$ ビット独自の符号語の個数は少なく, 従って, $i_{\max} \cdot n_{\min}$ が長くなっても, 復号に要する回路規模が非現実的になることはない. 例えば, 表3の No.3 のRLL符号の最大符号語長は15ビットであるが, 使用する符号語数は全部で16(裏パターンは除く)である. この内訳は, 符号語長15の符号語数は4, 符号語長12の符号語数は3, 符号語長9の符号語数は3, 符号語長6の符号語数は3, 符号語長が3の符号語数は3である.

また, 表3の No.6 のRLL符号の最大符号語長は16ビットであるが, 使用する符号語数は全部で13(裏パターンは除く)であり, No.16 のRLL符号の最大符号語長は24ビットにもなるが, 使用する符号語は全部で40(裏パターンは除く)である. このように, 最大符号語長が長くなっても, 検出しなければならない符号語の個数は非常に少なく, 単純なロジックで実現可能である.

5. むすび

デジタル記録機器に適したデジタル変調符号を求めることを目的として, 記録媒体上で0及び1の連続ビット数を d 以上 k 以下に制約する, ラン長制約符号の組織的な構成方法について検討した. この結果次のような結果が得られた.

- (1) d, k 制約されたRLL符号の符号語数の計算式を導出した.
- (2) 有効符号語数が最大になる符号語の最適接続条件及び符号語同士の接続則を, $d=1$ 及び $d \geq 2$ の場合についてそれぞれ求め, 極めて簡潔な論理式で実現できることを明らかにした.
- (3) (1)及び(2)の条件に基づいて組織的にRLL符号を構成し, 従来よく知られているNRZI表記の $(1, 7)$ 符号や $(2, 7)$ 符号と等価なRLL符号を実現できることを明らかにすると共に, NRZL表記で $(d, k, m_{\min}, n_{\min}, i_{\max}) = (2, 7, 2, 3, 5), (3, 12, 8, 15, 2), (5, 18, 2, 5, 6), (6, 16, 2, 6, 5)$ などの新たな可変長RLL符号が得られた.

この結果, 記録メディアの特徴を活かし, より一層高密度記録に貢献し得るRLL符号の構成が容易になった.

ただ, RLL符号だけで高密度記録を実現できるものではなく, メディアの特性に適した信号検出方式, 並びに誤り訂正符号との整合をも考慮したシステム開発も重要である.

例えば, d が大きくなると逆に T_w が小さくなり,

クロック再生用PLL(Phase Locked Loop)の精度向上とともに、帯域が広がることによって増加する等化雑音を低減するための、例えば Partial Response^[16]等の信号検出方式についても検討する必要がある。

更に、ラン長制限符号の符号語長が長くなると、ラン長制限符号の一つの符号語の誤りが複数のデータ語の誤りに伝搬する場合がある。このような誤り伝搬の影響を低減するような符号語とデータ語の対応関係の検討とともに、シャプリングなどによって一つのラン長制限符号の符号語に対応する複数のデータ語が、一つの誤り訂正符号の符号語に集中しないようなことも考慮する必要がある。

システム全体の誤りを低減するために、信号検出方式並びに誤り訂正符号の構成を含めて、個々の記録メディアに適したラン長制限符号を見出すことが今後の課題である。

最後に、日頃ご指導、ご鞭撻をいただく当社ディスプレイデバイス研究所 山光長寿郎所長その他関係各位に深謝する。

文献

- [1] 伊藤陽之助, 西村一敏, “ディジタル磁気記録の変復調方式”, 日経エレクトロニクス, 1978年12月11号, pp.126-164, Dec. 1978.
- [2] 渡辺彰三, “大容量磁気ディスクIBM3370/3380と, 開発進む各社対抗機種”, 日経エレクトロニクス, 1981年8月3日号, pp.106-119, Aug. 1981.
- [3] M. Artigas, “8/16-A New Channel Coding for Digital VTR”, 12th International Television Symposium and Technical Exhibition, Program of Equipment Innovations Sections, pp. 261-280, Jun. 1981.
- [4] 池谷章, 山光長寿郎, 末定邦雄, 小倉一郎, “RLL符号の一般的構成法”, テレビジョン学会全国大会予稿集, pp.153-154, Jul. 1984.
- [5] 田崎三郎, 大沢寿, “高密度記録における符号化方式の動向”, 信学誌, Vol. 68, No. 12, pp. 1301-1306, Dec. 1985.
- [6] 池谷章, 山光長寿郎, 末定邦雄, 小倉一郎, “ディジタル変調方式の組織的構成法”, National Technical Report, Vol.31, No.6, pp.91-97, Dec. 1985.
- [7] 田崎三郎, 原田浩二, 山田芳郎, 大沢寿, “(2, 7; m, n) 適応型固定長記録符号の諸特性”, 信学論(C), Vol. J69-C, No. 5, pp.653-661, May 1986.
- [8] 池谷章, 山光長寿郎, “可変長RLL符号の一構成法”, 信学技報, MR86-41, Dec. 1986.
- [9] P. A. Franaszek, “Sequence-State Methods for Run-Length-Limited Coding”, IBM J. Res. Develop., pp.376-383, Jul. 1970.
- [10] K. A. S. Immink, “A Generalized Method for Encoding and Decoding Run-Length - Limited Binary Sequences”, IEEE Trans., Vol. IT-29, No. 5, pp.751-754, Sept. 1983.
- [11] D. T. Tang and L. R. Bahl, “Block Codes for a Class of Constrained Noiseless Channel”, Information and Control, 17, pp.436-461, 1970.
- [12] T. Horiguchi et al.: “An Optimization of Modulation Codes in Digital Recording”, IEEE Trans. Magn., Vol. MAG-12, No. 6, pp. 740-742, Nov. 1976.
- [13] 古川輝雄, 田中邦麿, “ディジタル変調方式の一評価”, 信学技報, MR81-14, Sept. 1981.
- [14] M. Cohn and G. V. Jacoby, “Run-Length Reduction of 3PM Code via Look-Ahead Technique”, IEEE Trans. Magn., Vol. MAG-18, No. 6, pp. 1253-1255, Nov. 1982.
- [15] 木泰治, 松島宏, 加藤三三男, 竹内宏壮: “ディジタル磁気記録変調方式の一検討”, 信学技報, MR82-20, Oct. 1982.
- [16] E. R. Kretzmer, “Generalization of a Technique for Binary-data Transmission”, IEEE Trans. Vol. COM-14, No.1, pp.67-68, Feb. 1966.