UDC 621.394.147 : 681.3.041.5

自 己 同 期 符 号 に つ い て

On the Comma Free Codes

安　田　靖　彦*
Yasuhiko YASUDA

信号をブロック符号に符号化して伝送するとき，受信側では復号に先きだってビット同期およびフレーム同期をとることが必要である．このうちフレーム同期は通常フレーム同期信号を間けつ的にそう入する方法をとっているが，同期の問題を基本的立場から考え直してみると他にもいろいろな方式が考えられる．自己同期符号による同期方式はその一つであり，信号をになう符号自体がある意味で同期情報を運ぶ．したがって同期符号をそう入する必要がない．しかし自己同期符号とするためには冗長度が必要であるから，この冗長度をいかに処理するかによって自己同期符号方式のよさがきまる．本文は自己同期符号の構成の仕方，問題点等を述べたものである．

1. はじめに

信号をブロック符号に符号化して伝送する際，受信側では復号に先きだってビット同期およびワード同期（またはフレーム同期）をとることが必要である．このうちワード同期ないしフレーム同期をとる方法は2種に大別できる．

その一つはデータビットとは別に同期情報のみを運ぶ符号を用いる方法であり，他はデータ符号そのものに符号分離機能をもたせる，いいかえれば，自己同期符号を用いる方法である．そのうち前者は同期符号を連続的に送る方法と，間けつ的に送る方法とがある．連続的に送る方法において最適な符号ないし系列は PN sequence であって，その自己相関関数が同期時点以外の時点において一様に小さいことを利用する．しかしこの手法を用いるためには PN 系列の伝送のためにデータ符号を送るチャンネルとは別のチャンネルを用意するか特殊な工夫をこらしてデータチャンネルに混合する必要がある．また間けつ的にそう入する方式では同期符号をデータ符号の間に適当な間隔でそう入する．この方式での最適符号は同期符号自体の aperiodic autocorrelation ができるだけ一様に小さくなる符号が望ましく，いわゆる Barker sequence がそれである[1)2)]．これらの方式ではいずれにせよ同期符号のために余分のチャンネル容量が必要である．

データ符号そのものに符号分離機能をもたせ，同期符号を用いない方式は，余分のチャンネル容量を必要とせず，同期回復が早いなどの利点があるが，このような性質をもつ符号すなわち自己同期符号を見いだすことが問題である．

2. 自己同期符号による同期の可能性

初めに同期とは何かについて考える．1ワードないし1フレーム n ビットからできているディジタル通信路において，受信側で入ってくるパルス列を n ビットずつ区切る仕方は n とおりある．この内の一つの区切り方が正しく同期している状態であり，他の $n-1$ は同期はずれの状態である．受信者は現在の区切りが同期状態から数えて何番目であるかを知れば，少なくとも原理的には同期に入ることが可能である．したがって同期に入るために必要な情報量は

$$\log_2 n \text{ ビット} \qquad (1)$$

で与えられる．今このチャンネルが最大 R bits/sec の情報伝送速度をもつとする．また実際のデータ情報の伝送速度を R' とする．しかしこの値は同期状態に入っている場合の値であって，そうでなければ信号を復号することはできないから，伝送速度はさらに小さな値 R'' となっている．観測を開始してから同期に入るまでの時間を T_s sec とすれば

$$(R-R'')T_s \geq \log_2 n \qquad (2)$$

が成立しなければならない．一般に行なわれているように同期信号を繰り返しそう入する方法では同期回復後の伝送速度 R' は R より小さく，$R-R'$ bits/sec が無駄になる．自己同期符号を用いる方式では同期回復後の伝送速度 R' は自動的に R に等しくなり無駄を生じない．

なお，通常のディジタル伝送においてはいくつかのチャンネルを多重化する場合フレーム同期信号のみを用いワード同期信号はそう入しない．自己同期符号はワード同期に関するものであるから，時分割多重を行なう場合は別にフレーム同期信号が必要になる．事実自己同期符号の実用例[3)]では上述の方法をとっている．しかしこれでは自己同期符号を用いる意義が半減する．フレーム同期信号をそう入しないで多チャンネルを送るためには次のようにすればよい．各ワードのビット数を b，1フレームのチャンネル数を m とするとき

$$C \geq 2^{mb} \qquad (3)$$

を満足する符号語数 C の自己同期符号を用いて1フレームを一つの長さ mb ビットのワードとみなして符号化する．この方法によって多チャンネルの場合も同期信号のそう入なしに伝送することが原理的には可能である．

3. コンマ符号

Comma Code はアルファベットの一つの文字を各ワ

* 東京大学生産技術研究所第3部

ードの特定位置におき，他の位置でこの文字を使用しないことによってワードの分離を可能にする符号であり，2進符号の一例を示せば図1のようになる[4]

符号番号	符　号
C_1	0
C_2	0 1
C_3	0 1 1
C_4	0 1 1 1
C_5	0 1 1 1 1

図 1

符号ブロックの長さを一定とすると，2進ではComma Code はできない．この場合符号長を n，アルファベットの文字の数を α とすれば最大 $(\alpha-1)^{n-1}$ 個の符号数がとれる．この一例を図2に示す．

$\alpha=3$				$n=4$			
0	1	1	1	0	2	1	1
0	1	1	2	0	2	1	2
0	1	2	1	0	2	2	1
0	1	2	2	0	2	2	2

図 2

Comma Code の欠点は与えられた α，n に対してとりうる符号数が少なく能率が悪いことである．

4. 同期可能ブロック符号

文字 0, 1, ……, $\alpha-1$ をアルファベットとする長さ n の block code を $C_{\alpha,n}$ とする．これらの符号から得られる任意の信号系列の M 個 ($M<\infty$) の相続く文字を調べれば必ず符号分離ができるとき，この $C_{\alpha,n}$ を M 次の synchronizable block code という[5]．

$C_{\alpha,n}$ の構成法は下記のとおりである．α 個の文字からなる長さ n の相異なる符号の数は α^n 個である．符号の文字を cyclic にシフトする変換 g の作る群を G_n とする．すなわち

$$X_1X_2\cdots\cdots X_n \xrightarrow{g} X_2X_3\cdots\cdots X_nX_1 \qquad (4)$$

二つの符号 W_1, W_2 に対して G_n の中に一方を他方に写す変換が存在するとき

$$W_1 = W_2 \pmod{G_n} \qquad (5)$$

とかき，W_1 と W_2 は同等 (equivalent) であると称する．同等な符号の集合を equivalence class という．この最大の符号数は明らかに n であるが，これより小さい class も存在する．

(例)

$\alpha=2$

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$n=4$

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

次に文字の K 個の連なりを Y で表わす．

$$Y=X_1X_2X_3\cdots\cdots X_K$$

このとき Y の数値 $N(Y)$ を次式で定義する．

$$N(Y)=X_1\alpha^{K-1}+X_2\alpha^{K-2}+X_{K-1}\alpha+X_k \qquad (6)$$

同じ長さの Y_1, Y_2 が同じ数値をもつならば，明らかに両者は等しい．n 個の符号をもつ equivalence class を nondegenerate class と呼び，この中の符号で最も小さい数値をもつものをその class の代表という．

self synchronizable code $C_{\alpha,n}$ はすべての nondegenerate class の代表からなっている[5]．

例　$\alpha=2$

$S_1=(\ 0\ 0\ 0\ 0\ )$

$S_2=(\ 0\ 0\ 0\ 1,\ 0\ 0\ 1\ 0,\ 0\ 1\ 0\ 0,\ 1\ 0\ 0\ 0\ )$

$S_3=(\ 0\ 0\ 1\ 1,\ 0\ 1\ 1\ 0,\ 1\ 1\ 0\ 0,\ 1\ 0\ 0\ 1\ )$

$S_4=(\ 0\ 1\ 0\ 1,\ 1\ 0\ 1\ 0\ )$

$S_5=(\ 0\ 1\ 1\ 1,\ 1\ 1\ 1\ 0,\ 1\ 1\ 0\ 1,\ 1\ 0\ 1\ 1\ )$

$S_6=(\ 1\ 1\ 1\ 1\ )$

$C_{2,4}=(\ 0\ 0\ 0\ 1,\ 0\ 0\ 1\ 1,\ 0\ 1\ 1\ 1\ )$

同様に

$C_{2,5}=(\ 0\ 0\ 0\ 0\ 1,\ 0\ 0\ 0\ 1\ 1,\ 0\ 0\ 1\ 0\ 1,$
$0\ 0\ 1\ 1\ 1,\ 0\ 1\ 0\ 1\ 1,\ 0\ 1\ 1\ 1\ 1\ )$

$C_{2,6}=(\ 0\ 0\ 0\ 0\ 0\ 1,\ 0\ 0\ 0\ 0\ 1\ 1,\ 0\ 0\ 0\ 1\ 0\ 1$
$0\ 0\ 0\ 1\ 1\ 1,\ 0\ 0\ 1\ 0\ 1\ 1,\ 0\ 0\ 1\ 1\ 0\ 1$
$0\ 0\ 1\ 1\ 1\ 1,\ 0\ 1\ 0\ 1\ 1\ 1,\ 0\ 1\ 1\ 1\ 1\ 1\ )$

$C_{\alpha,n}$ の符号数は nondegenerate class の数を求めればよく次式で与えられる．

$$f(\alpha,n)=\frac{1}{n}\sum_{d|n}\mu(d)\frac{\alpha^n}{d} \qquad (7)$$

ただし，d は n の因数であり，$\sum$ はすべての d にわたる．また $\mu(d)$ は Möbius function といわれるもので，次式で定義される．

$$\mu(d)=\begin{cases} 1 & \text{if } d=1 \\ 0 & \text{if } d \text{ has a square factor} \\ (-1)^r & \text{if } d \text{ is the product of } r \text{ distinct prime number} \end{cases} \qquad (8)$$

この式の極限は次式で与えられる．

$$\lim_{n\to\infty}\frac{nf(\alpha,n)}{\alpha^n}=1 \qquad (9)$$

$$\left.\begin{aligned} &\lim_{\alpha\to\infty}\frac{f(\alpha,n)}{\alpha^n}=a_n \\ &\text{ただし}\quad a_n=\frac{1}{n} \qquad n:\text{奇数} \\ &\qquad\frac{1}{n}\geq a_n>\frac{1}{e_n} \qquad n:\text{偶数} \end{aligned}\right\} \qquad (10)$$

同期可能とすることによって符号に冗長度を必要とする．冗長度 R はその定義から

$$R=\frac{\log\alpha^n-\log f(\alpha,n)}{\log\alpha^n}=1-\frac{\log f(\alpha,n)}{\log\alpha^n} \qquad (11)$$

$n\to\infty$ において(9)式を用いて

$$\lim_{n\to\infty}R=\lim_{n\to\infty}\frac{\log n}{n\log\alpha}=0 \qquad (12)$$

したがって十分大きな n に対しては冗長度は $1/n\log_2 n$ 程度と考えてよい．

なお同期をとるに要する時間の上限は　(13)

$$M \leqq n\alpha^{n/2}+n-1 \quad (14)$$

で与えられる.

5. コンマフリー符号

Comma free code は前記の synchronizable block code で $M=2n-1$ なる制限を付した場合に相当する. Golomb 等が始めてこの種の符号を発表したとき Comma free code の定義を次のように述べた[6]. block code $C_{\alpha,n}$ に属する任意の二つの符号 $W=W_1, W_2\cdots\cdots W_n$ および $X=X_1. X_2\cdots\cdots X_n$ の n 文字のオーバーラップ

$W_K\cdots\cdots W_nX_1\cdots\cdots X_{k-1}$ $(K=2, \cdots\cdots, n)$

のどれもが $C_{\alpha,n}$ に含まれないとき, $C_{\alpha,n}$ を Comma free code という.

Camma free code の符号数は n が奇数のときは synchronizable code のそれと等しくなり, n が偶数のときは一般にはこの値より小さくなる[7]. ある符号群が Comma free であるかどうかは Kendall および Reed が与えた図式法によるのが便利である[8]. たとえば $\alpha=4$(A, B, C, D) $n=3$ の符号群

ABA	ADA	BCB	BDD
ABB	ADB	BCC	CDA
ACA	ADC	BDA	CDB
ACB	ADD	BDB	CDC
ACC	BCA	BDC	CDD

を考えよう. まず図3に示すように, α 個の文字を行にまた n 個のシンボル位置を列に対応させた $\alpha\times n$ 行列を作る. この行列の中の各要素は 1, 0 からなり, 符号群でその文字がその位置で用いられているときは1, いなければ0とする.

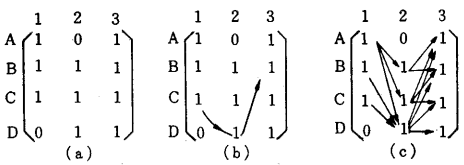


図 3

次に各符号をこの行列中の "Path" として表現する. すなわち CDA なる符号は同図（b）に示す path と1対1に対応し, 全符号は（c）図のように互に異なった "path" に対応する. 二つの符号の連らなり(a_1 a_2 a_3 b_1 $b_2\to b_3$) は図4(a)に示すように元の行列を二つ並置して左の行列の最右翼の列の "1" と右の行列の最左翼の列の "1" とをすべての組合せで接続したものに当たる. またオーバーラップした符号 (a_2 a_3 b_1) は同図の(b)ように,(a)図で a_1 b_2 b_3 の列を取り去ったときできる行列中の path で表現できる. (a_3 b_1 b_2) についても同様に(c)図のように表わせる. Comma free code の性質は(b), (c)図の path のどの一つも元の行列の path に

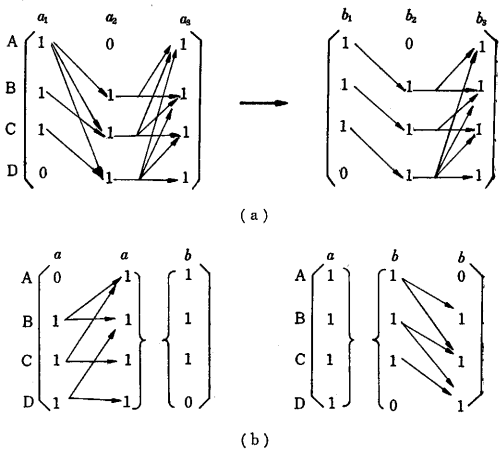


図 4

重ならないという条件に置き代わる. 上記の例では確かにこれが満たされている.

ここで行列中の "1", "0" の配置を適当に選んで, 左から右へ可能な "path" をすべて作ったとき, これに対応する符号が Comma free の特性をもつようにできる. このようにしてできた符号は明らかに一般の Comma free code の subset であり, path のとり方に関係しない. Kendall 等はこれを path invariant Comma free code と呼んだ. 図5に Comma code, Comma free code および path invariant Comma free code の符号数の比較を示す[8].

a	符号の型	n			
		3	5	7	9
2	CF	2	6	18	56
	PICF	2	4	8	16
	C	1	1	1	1
3	CF	8	48	312	2,184
	PICF	6	36	216	1,296
	C	4	16	64	256
4	CF	20	204	2,340	29,120
	PICF	16	144	1,728	20,736
	C	9	81	729	6,561
5	CF	40	624	11,160	217,000
	PICF	30	450	8,000	160,000
	C	16	256	4,096	65,536
6	CF	70	1,554	39,990	1,119,720
	PICF	54	1,152	27,648	810,000
	C	25	625	15,625	390,625

図 5

6. コンマフリー特性をもつ2進直交符号

これまでの議論は, 一般のアルファベットについて符号間距離については特別の制限をおかないで進めてきたが, ここでは実用上重要な2進の orthogonal code の中

から Comma free の特性をもつものを捜す方法を述べる．

binary orthogonal comma free code A に属する二つの符号 $a=\alpha_1,\alpha_2\cdots\cdots\alpha_N$, $b=\beta_1,\beta_2\cdots\cdots\beta_N$ の任意のオーバーラップ $\alpha_K\cdots\cdots\alpha_N$, $\beta_1\cdots\cdots\beta_{K-1}$ と A に属する任意の符号との間の Hamming の距離の最小のものを index of comma freedom P と定義する[9]．雑音の存在するチャンネルにおいてはいうまでもなく Comma freedom が大きいほど同期がとりやすい．

ここでN個のベクトルをもつ任意の orthogonal code G をとり，これに各 binary symbol を $P(0)=1/2$, $P(1)=1/2$ でランダムに選んでできるN次のベクトル c を加えて次式で示す新しい orthogonal code 群を考える．

$$H=G\oplus c\oplus c^K+y \tag{15}$$

ただし c^K はランダムベクトル c の各シンボルを左に Kだけシフトしたベクトル，y は G に属する一つのベクトルの後 N-K シンボルと他のベクトルの始めの K シンボルからなるベクトルを表わす．$G\oplus y$ (y は固定) は orthogonal ベクトル群であるし，$c\oplus c^K$ はランダムベクトルであるから，H の各ベクトルもランダムベクトルである．ここで H の要素を1の代わりに $-1/N$, 0の代わりに $1/N$ と入れ換えれば H の1番目のベクトルの各シンボルの和は丁度，$G\oplus c$ の1番目のベクトルと，Kだけオーバーラップしたベクトルとの相関係数になっている．系列 $\{h_j{}^i\}$ は $P(-1/N)-P(1/N)=1/2$ のランダム系列であるから

$$S_i{}^2=\left[\sum_{j=1}^{N}h_j{}^i\right]^2 \tag{16}$$

は random variable でありその平均値は

$$E(S_i{}^2)=E\sum_{j=1}^{N}\sum_{k=1}^{N}h_j{}^i h_k{}^i=\frac{1}{N^2}\sum_{j=1}^{N}\sum_{k=1}^{N}\delta_{jk}=\frac{1}{N} \tag{17}$$

で与えられる．また分散は

$$\begin{aligned}a^2(S_i{}^2)&=E[(S_i{}^2)^2]-E(S_i{}^2)]^2\\&=E\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{m=1}^{N}\sum_{\alpha=1}^{N}h_j{}^i h_k{}^i h_m{}^i h_n{}^i-\frac{1}{N^2}\\&=\frac{2(N^2-N)}{N^4}\fallingdotseq\frac{2}{N^2}\end{aligned} \tag{18}$$

となる．したがってGの任意の符号とオーバーラップ符号の間の相関係数の期待値は $\pm1/\sqrt{N}$であり，分散はNが大きいとき急速に0に近づく．この場合の Comma feedom P_N は

$$P_N\fallingdotseq\frac{N-\sqrt{N}}{2} \tag{19}$$

で与えられる．この結果から Stiffler は 2^m 次の Hadamard行列 A_m から Comma freedom の大きい orthogonal code B_n を作り出す方法を示した[9]．すなわち

$$B_m=A_m\oplus c_m \tag{20}$$

ただし c_m は長さ 2^m-1 の P_N 系列に適当な1ビットを付加して長さ 2^m としたもので，始めのいくつかのmに対しては下記に示す系列である．

$c_4=$0 0 1 0 0 0 1 1	1 1 0 1 0 1 1 1
$c_5=$1 0 0 0 1 1 0 1	1 0 0 1 0 1 0 0
0 0 1 0 0 1 0 1	1 0 0 1 1 1 1 1
$c_6=$0 0 0 1 1 1 0 0	1 0 0 1 0 1 1 0
1 1 1 0 1 1 0 0	1 1 0 1 0 1 0 0
1 1 1 1 1 1 0 0	0 0 0 1 0 0 0 0
1 1 0 0 0 1 0 1	0 0 1 1 1 1 0 1
$c_7=$1 1 1 1 1 1 1 1	0 0 0 0 0 0 1 0
0 0 0 0 1 1 0 0	0 0 1 0 1 0 0 0
1 1 1 1 0 0 1 0	0 0 1 0 1 1 0 0
1 1 1 0 1 0 1 0	0 1 1 1 1 1 0 1
0 0 0 0 1 1 1 0	0 0 1 0 0 1 0 0
1 1 0 1 1 0 1 0	1 1 0 1 1 1 1 0
1 1 0 0 0 1 1 0	1 0 0 1 0 1 1 1
0 1 1 1 0 0 1 1	0 0 1 0 1 0 1 0

comma freedom は図6に示すとおりである．

$N=2^n$	P_N
16	2
32	6
64	14
128	34

図 6

orthogonal code の comma freedom は符号長に比例して大きくなる．

7. 固定パターンをもつ自己同期符号

一般の自己同期符号は構成がやっかいであり，装置も複雑になる．符号語に特定の同期パターンをそう入し自己同期符号とすれば，符号の生成，受信側メモリが少なくてすむ．しかし取りうる符号語数は一般の場合より小さくなる．この種の自己同期符号は Gilbert[10]が提唱し，その後，宮川，守屋[11][12]喜安[13]等が取り扱っている．

：集中方式

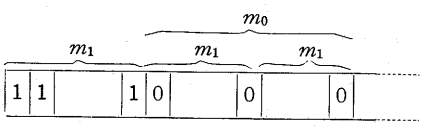
符号の長さを n とするとき先頭に $P_1\cdots\cdots P_m$ の固定パターン P を付けて残り $k=n-m$ ビットに情報をになわせる方式である．

$(P_1,\ P_2,\cdots\cdots,\ P_m,\ x_1,\ x_2,\cdots\cdots x_k)$

符号語数の点で最適な同期パターンの一つは(111……10) である．

：分散方式

$P_1,\cdots\cdots,\ P_m$ の m 個の固定ビットを図のように，先頭に“1”をm_1 個，次に“0”を m_1-1 個おきに m_0 個おく．



ただし

$$\left.\begin{aligned} m_1&=\left[\frac{m+1}{2}\right]\\ m_0&=m-m_1 \end{aligned}\right\} \tag{21}$$

とする．

このとき同期パターンの長さ γ は次式のようになる．

$$\gamma=\left[\frac{m^2}{4}\right]+1 \tag{22}$$

符号語数 G は一般に

$$G\leq 2^k \tag{23}$$

となるが，宮川等は上式で等合が成立する場合を完全自己同期符号，他の場合を擬完全自己同期符号と呼んでいる．

完全自己同期符号の条件は明らかに

$$\left.\begin{aligned} &\text{集中方式のとき}\quad k\leq m-1\\ &\text{分散方式のとき}\quad n\leq 2\gamma-1\\ &\text{すなわち}\quad k\leq\frac{m^2}{2}-m+1\\ &\qquad\qquad\leq n-\sqrt{2(n-1)} \end{aligned}\right\} \tag{24}$$

となる．

完全自己同期符号の符号数は一般の自己同期符号にくらべ著しく小さい．そこで Gilbert 等は同期パターンの長さを(24)式で与えられる限界より小さくし，その結果，情報ビットに制限をもうけることによって，全体としての符号語数の増大が得られることを示した．これが擬完全自己同期符号である．これらの符号の符号語数は前述の人々によって得られている．一般に集中方式より分散方式の方が符号語数が大きく，擬完全自己同期符号においては n を大きくするにつれて，一般の自己同期符号と同程度の冗長度になることが示されている．

ここで述べた自己同期符号は1ワードに1フレームを対応させると，従来から取り扱われている同期信号そう入によるフレーム同期方式と非常に類似していることがわかる．したがって逆にこの考え方をフレーム同期方式に応用することも可能であろう．ただしデータビットが 1/2 の確率でオンオフするランダムビットで符号誤りも考慮に入れると，自己同期符号で先頭におくのに適当な (1 1 1……1) ないし (1 1……1 0) なるパターンは最悪のパターンである．

8. 自己同期誤り訂正群符号

先に述べたように自己同期符号を構成するためには冗長度が必要である．一方誤り訂正符号を構成するときにも冗長度がいる．したがって，もし誤り訂正符号とするための冗長度を自己同期のためにそのまま利用できれば，余分の冗長度が必要でなくなり，きわめて望ましい符号が得られる．J. J. Stiffler はこの観点に立って，誤り訂正群符号から自己同期特性をもつものを捜す方法および誤り訂正巡回符号に自己同期特性を持たせる方法を求めた[14]．以下その手法を述べる．

(i) 群符号の自己同期特性

まず次の定義をもうける．

定義：長さ n の符号群 S から任意の二つの符号 $a_1\ a_2\cdots a_n$，$b_1\ b_2\cdots b_n$ を取る．

二つの符号の n 字のオーバーラップ

$$a_{r+1},\ a_{r+2}\cdots\cdots a_n\ b_1\ b_2\cdots\cdots b_r$$

が符号群 S の中に含まれないとき，この符号群 S は位置 r において同期可能という．

符号群がすべての位置 ($r=1, 2, \cdots\cdots, n-1$) で同期可能ならば，コンマフリー符号である．

今 S を群符号に限定する．群符号は符号語 (0…0) を必ず含む．したがってそのままではコンマフリーとはならない．そこで2進ベクトル $\bar{c}$ をとり，$\bar{c}$ に対する S の coset を作ると，誤り訂正能力を変えずに新しい符号群ができる．これをコンマフリーにする可能性を調べる．

ここで (n, k) 群符号 S の生成行列を G とする．すなわち

$$\bar{\xi}=G\bar{x}\quad\text{ただし}\ \bar{\xi}\in s \tag{25}$$

$\bar{x}$：要素 k の任意ベクトル

符号群を $\bar{c}$ によって作られる S の coset とすれば，符号群の任意の符号語 $\bar{a}$ は

$$\bar{a}=G\,\bar{x}\oplus\bar{c} \tag{26}$$

と表わせる．次に G と二つの行列 G_1(m 行 k 列)，G_2 (n-m 行 k 列) に分け

$$G=\begin{bmatrix}G_1\\ G_2\end{bmatrix} \tag{27}$$

とかく．同様に $\bar{c}$ を $\bar{c}_1$ (m要素)，$\bar{c}_2$ (n-m 要素) に分け

$$c=\begin{bmatrix}\bar{c}_1\\ \bar{c}_2\end{bmatrix} \tag{28}$$

とおく．$\bar{z}$，$\bar{y}$ を k 要素の列ベクトルとして $\bar{\alpha}$，$\bar{\beta}$ を

$$\left.\begin{aligned} \bar{\alpha}&=G_1\,\bar{z}\oplus c_1\\ \bar{\beta}&=G_2\,\bar{y}\oplus\bar{c}_2 \end{aligned}\right\} \tag{29}$$

と定義すれば，ベクトル $\begin{bmatrix}\bar{\beta}\\ \bar{\alpha}\end{bmatrix}$ は二つの符号の任意のオーバーラップを表わす．これを使って任意の符号語 $\bar{a}$ と，任意のオーバーラップ語との要素の不一致数は次のベクトルの“1”の数として求まる．

$$\begin{aligned}\overline{W}&=\bar{a}+\begin{bmatrix}\bar{\beta}\\ \bar{c}_m\end{bmatrix}+\begin{bmatrix}\bar{o}_{n-m}\\ \bar{\alpha}\end{bmatrix}\\ &=\begin{bmatrix}G & \begin{matrix}G_2 & O_{n-m}\\ O_m & G_1\end{matrix}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\\ \bar{z}\end{bmatrix}\oplus\bar{c}\oplus\begin{bmatrix}\bar{o}_2\\ \bar{o}_1\end{bmatrix}\end{aligned} \tag{30}$$

ただし，$\bar{o}_m$ は m 要素がすべて 0 のベクトル，O_m は要素がすべて 0 の m 行 k 列の行列を表わす．

coset が位置 m において同期不能ならばある $3k$ 次の2進ベクトル $\bar{v}$ があって

$$M\bar{v}=\bar{b} \tag{31}$$

が成立する．

ただし

$$M=\begin{bmatrix} G & \begin{matrix} G_2 & O_{n-m} \\ O_m & G_1 \end{matrix} \end{bmatrix}$$

$$\bar{b}=\bar{c}\oplus\begin{bmatrix}\bar{c}_2\\ \bar{c}_1\end{bmatrix}$$

したがって coset が位置 m において同期可能であることは(31)式を満足するベクトル$\bar{c}$が存在しないことである．そのためには

$$\mathrm{rank}(M)<\mathrm{rank}(M,\bar{b}) \tag{32}$$

がいえればよい．今 M のある1次従属な行ベクトルの番号の集合を s_j とするとき

$$\sum_{i\in s_j} b_i=1 \tag{33}$$

ならば(32)式が成立する．

次にこのような s_j が存在するための条件は G の null space H に次式を満たす三つのベクトル $\bar{h}$, $\bar{h}^{(1)}$, $h^{(2)}$ が存在することである．

$$\left.\begin{aligned} h_i^{(1)}&=\begin{cases} h_{i+n+m} & i=1,2,\cdots\cdots,m \\ 0 & i=m+1,\cdots\cdots,n \end{cases} \\ h_i^{(2)}&=\begin{cases} 0 & i=1,2,\cdots\cdots,m \\ h_{i-m} & i=m+1,\cdots\cdots,n \end{cases} \end{aligned}\right\} \tag{34}$$

上記の条位によってコンマフリー符号を捜す手間は大幅に減る．

(ii) 自己同期巡回符号

巡回行列 $(n,\ k)$ のnull space H の generator H は一般に次式で与えられる[15]．

$$H=\begin{pmatrix} \alpha_1 & \alpha_2\cdots\cdots\alpha_{k+1} & 0\cdots\cdots\cdots 0 \\ 0 & \alpha_1 \quad \alpha_k & \alpha_{k+1}\cdots\cdots 0 \\ \vdots & \vdots & \\ 0 & 0\cdots\cdots 0 & \alpha_1,\alpha_2\cdots\cdots\alpha_{k+1} \end{pmatrix} \tag{35}$$

$$\alpha_1,\ \alpha_{k+1}\neq 0$$

これを前述の結果と結合して次の定理を得る．

定理：2進巡回符号 $(n,\ k)$ は $|r|\leq n-k-1$ を満足するすべての位置 r において同期可能である．

〔証明〕

まず $m\leq n-k-1$ とする．$\bar{h}$ として H の最初の行ベクトルをとれば $\bar{h}^{(1)}=O\epsilon H$ であり，$\bar{h}^{(2)}$ は H の$(m+1)$行目のベクトルによって，(34)式が満たされる．

一方

$$\bar{c}=\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix} \tag{36}$$

とおけば

$$b_i=\begin{cases}1 & i=1,\ n-m+1\\ 0 & \text{その他の } i\end{cases} \tag{37}$$

となるから $k+1<n-m+1$ すなわち $m\leq n-k-1$ に対し

$$\sum_{i=1}^{n} h_i b_i=\sum_{i=1}^{k+1} h_i b_i=\alpha_1=1 \tag{38}$$

同様に H の最後の行について考えると

$$n-m\leq n-k-1$$

についても同じことがいえる．

また次の定理が証明できる．

定理：$n-k\leq r\leq k$ なる位置 r においては同期不能，したがって上記二つの定理を結合することによって，$(n,\ k)$ 巡回符号は $k\leq(n-1)/2$ なるかぎり常にコンマフリー符号となし得ること．逆に $k>(n-1)/2$ のときはコンマフリーとなし得ないことがわかる．

9. むすび

以上自己同期符号について概略を述べた．もともと理論的興味から出発した研究であるが，最近は space communication（とくに deep space communication）．あるいはデータ伝送などに対する応用を見いだしている．この観点からすれば J. J. Stiffler が行なっているように誤り訂正のために必要な冗長度を自己同期特性のために共用することが望ましい方向であろうと思われる．

(1967年7月28日受理)

文　献

1) R. H. Barker: Group Synchronization of Binary Digital Systems, Communication Theory by W. Jackson, Butterworth's Scientific Publications.
2) M. W. Willard: Optimum Code Patterns for PCM Synchronization, Proc. of 1962 National Telemetering Conference No. 5-5.
3) R. M. Jaffe: Digilock Telemetry System for the Air Force Special Weapons Center's Blue Scout Jr. IRE Trans. on SET Mar. 1962, pp. 44-49.
4) R. A. Scholtz: Codes with Synchronization Capability, IEEE Trans. on IT Apr. 1966, pp. 135-142.
5) W. L. Eastman and S. Even: On Synchronizable and PSK Synchronizable Block Codes; IEEE Trans. on IT Oct. pp. 351-356.
6) S. W. Golomb, B. Gordon and L. R. Welch: Comma-Free Codes, Canad. J. Mathematics; Vol. 10, pp. 202-209.
7) W. L. Eastman: On the Construction of Comma Free Codes, IEEE Trans. on IT Apr. 1965, pp. 263-267.
8) W. R. Kendall and I. S. Reed: Path-Invariant Comma-Free Codes, IRE Trans. on IT, Oct. 1962, pp. 350-355.
9) J. J. Stiffler: Synchronization of Telemetry Codes, IRE. Trans. on SET June 1962, pp. 112-117.
10) E. N. Gilbert: Synchronization of Binary Message, IRE. Trans. IT-6.
11) 宮川，守屋：Check Bit を有する Comma-Free 符号の構成法，信学会，インホ理論研資，1966-7-18.
12) 宮川，守屋：固定検査点を有する自己同期符号の総数に関する一考察，信学会，インホ理論研資，1967-4-27.
13) 喜安，丸岡：同期信号につき2進符号について，信学会インホ理論研資，1967-2-24.
14) J. J. Stiffler: Comma-Free Error Correcting Codes, IEEE Trans. on IT Jan. 1965, pp. 107-112.
15) W. W. Peterson: Error Correcting Codes, M. J. T. Press. Cambridge, Mass and John Wiley & Sons, New York 1961.