連載

世界をORする視線 (21)
第I部　通信・デジタル技術の発展
(3) コンピュータの発展：コンピュータ科学の数学的基礎（続き8）

住田　潮

（註：本稿は前回からの続きであるので，文献リストは継続し，新たに必要となる分を追加する）

1.　シャノンの第一基本定理と第二基本定理

シャノンが構想した通信システムの一般モデルは，図 1 に示すようにシンプルなものであった [12]（連載第 18 回）.

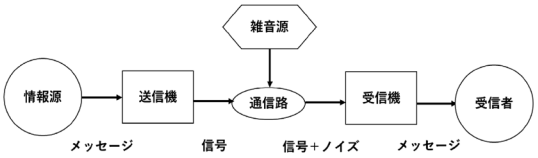

図 1　シャノンの通信一般モデル

情報源でメッセージが作成されると，送信機は，メッセージを符号化したうえで，通信可能な信号へと変換する．通信路は，送信機から受信機へと信号が通過する媒体であり，雑音源は通信路を通過する信号に対して歪みや乱れを発生させる．受信機は，雑音による誤送信の可能性を含む信号を受信し，送信機とは逆の操作を行い，可能な限り正確にメッセージを復元，その結果を受信者が受け取ることになる.

前回まで（連載第 18〜20 回），与えられた情報源に対し，どのようにして一意的に復元可能な符号化を効率的に行うかに関する研究の進展を辿ってきた．シャノンは，情報の量と価値を計る単位として情報エントロピーの概念を確立し，「ある環境で生成されるメッセージの記号出現を表わす確率変数 X とその確率分布が与えられたとき，一意的に復元可能な符号化に必要とされる平均符号語長は，X のエントロピー $H(X)$ bit 以下にはできない」という第一基本定理（情報源符号化定理）を証明したうえで，送信量が $H(X)$ bit と $H(X)+1$ bit の間に収まるシャノン符号を具体的に作成した.

その後，ファノ符号，さらに送信量の最小化を実現するハフマン符号が開発されたが，これらの符号方法は，すべて記号出現の頻度を表わす確率変数 X とその確率分布を前提知識として必要とするもので，大規模通信を行うためには極めて不便なものであった．この確率分布を必要とせず，送信記号の時系列に沿って符号化を実行し，符号列が伸びるに従って，1 符号当たりの符号語長がデータ圧縮率の下限値である 1 符号当たりの平均エントロピー $H(X)$ bit に限りなく近づくことを保証したのが LZ78 符号であった．ここまでの議論は，送信側の手続きに限定されたもので，図 1 に示すシャノンの情報理論体系の前半部分を網羅しているに過ぎず，受信側でメッセージを正確に復元するための効率性については，全く触れられていない.

送信機から受信機へと信号が通過する媒体である通信路は，通過する信号に対して，歪みや乱れを発生させる雑音源に直面する．この雑音がメッセージ復元の正確性をどう歪めるかという問題を構造的に把握するため，シャノンは，まず，通信路容量 C_0，伝送速度 R，メッセージを構成する符号に誤送信を効率的に処理するための符号を冗長的に付加した通信路符号，などの概念を導入した．さらに受信される通信路符号の集合を，正しく送信されたものを含む集合と含まない集合とに 2 分割する判定アルゴリズムを導入，受信された通信路符号が後者に属する確率を復号誤り率 $\mathcal{P}$ と定義した．受信される通信路符号は，確率 $1-\mathcal{P}$ で送信が正しく行われたと判定されるが，この判定結果は

すみた　うしお
筑波大学名誉教授
〒 305–8573　茨城県つくば市天王台 1–1–1

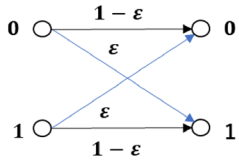

図 2　$A_1 = \{B_1 = 0, 1\}$

誤送信の可能性をも含む．一方，確率 $\mathcal{P}$ で誤送信が確実に検出され，この場合は，送信元へ当該通信路符号の再送を依頼することになる．シャノンは $\mathcal{P}$ の値が小さい通信路が良い通信路であると考え，第二基本定理（通信路符号化定理）として，$R \leq C_0$ であれば，十分な長さの冗長的符号を追加することにより，$\mathcal{P}$ をいくらでも小さくできることを証明した．本稿と次回で，おおむね文献 [67] に沿って，この第二基本定理を解説する．

2.　復号誤り率 $\mathcal{P}$ と冗長性

「誤送信を効率的に処理するための符号を冗長的に付加することにより，復号誤り率 $\mathcal{P}$ を小さくすることができる」というシャノンの発想を感覚的に知るため，まず，簡単な例を示すことから始めよう．送信記号集合 $A_1 = \{0, 1\}$ と受信記号集合 $B_1 = \{0, 1\}$ が与えられたとき，図 2 に示す通信路では，0, 1 いずれを送信しても確率 ε $(0 < \varepsilon < 1)$ で誤送信が発生する．このような雑音構造をもつ通信路は 2 元対称通信路と呼ばれ，誤送信された結果を誤って正しいものとして受諾してしまう確率は $E_{\varepsilon:1} = \varepsilon$ で与えられる．

同じ 2 元対称通信路を用いて，送信符号集合を $A_2 = \{00, 01, 10, 11\}$，受信符号集合を $B_2 = \{00, 01, 10, 11\}$ として送信を行う場合を考えると，図 3 に示すように，送信された符号が正しく復元される確率は $(1-\varepsilon)^2$ となり，誤送信受諾率は $E_{\varepsilon:2} = 1-(1-\varepsilon)^2 = \varepsilon(2-\varepsilon)$ で与えられる．

ここで，誤送信を効率的に処理するために付加される冗長的符号の例として Parity Bit を考える．送信元は，2 進数で表わされた送信符号の 1 の数が偶数であれば 0，奇数であれば 1 を送信符号の語尾に追加して送信する．ここで，偶数は 0 を含むことを注意しておく．送信符号集合 $A_2 = \{00, 01, 10, 11\}$ に対して Parity Bit を付与すると，実際に送信される通信路符号は，$A_{2:P} = \{000, 011, 101, 110\}$，その受信通信路符号集合は，$B_{2:P} = \{000, 100, 010, 001, 110, 101, 011, 111\}$ となる．

受信先では，受信信号の Parity Bit が整合性をもつ

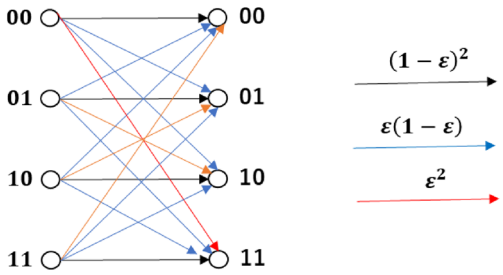

図 3　$A_2 = B_2 = \{00, 01, 10, 11\}$

表 1　Parity Bit 方式：$A_2 = \{00, 01, 10, 11\}$

通信路符号			発生確率	判断
送信符号		Parity		
c	c	c	$(1-\varepsilon)^3$	受諾
c	c	w	$(1-\varepsilon)^2\varepsilon$	棄却
c	w	c	$(1-\varepsilon)^2\varepsilon$	棄却
w	c	c	$(1-\varepsilon)^2\varepsilon$	棄却
c	w	w	$(1-\varepsilon)\varepsilon^2$	受諾
w	c	w	$(1-\varepsilon)\varepsilon^2$	受諾
w	w	c	$(1-\varepsilon)\varepsilon^2$	受諾
w	w	w	ε^3	棄却

場合は，符号部分を正しいものとして受け入れ，整合性がない場合は受信された通信路符号を棄却したうえで，その再送を送信先に依頼する．このとき，前述した判定アルゴリズムは「受信された通信路符号の Parity Bit が整合性をもつか否か」である．$x \in A_{2:P}$ に対して，この判定アルゴリズムによって受諾される受信通信路符号集合を $B_{2:PB1:Accept|x}$，棄却される受信通信路符号集合を $B_{2:PB1:Reject|x}$ と書くことにする．

ビットごとの正しい送信を c，誤送信を w と書くと，送信結果の可能な組み合わせは表 1 のようになる．$x \in A_{2:P}$ が与えられたとき，この表から色のついた行の集合として $B_{2:PB1:Accept|x}$，無色の行の集合として $B_{2:PB1:Reject|x}$ を生成できる．たとえば，$B_{2:PB1:Accept|000} = \{000, 011, 101, 110\}$，$B_{2:PB1:Reject|000} = \{001, 010, 100, 111\}$ となる．Parity Bit を判定アルゴリズムとするときは，任意の $x, y \in A_{2:P}$ に対して，$B_{2:PB1:Accept|x} = B_{2:PB1:Accept|y}$，$B_{2:PB1:Reject|x} = B_{2:PB1:Reject|y}$ が成立することを注意しておく．

通信路符号集合 $A_{2:P} = \{000, 011, 101, 110\}$ 上の任意の確率変数 X とその分布 $P_X(x) = P[X = x]$ に対して，表 1 と $B_{2:PB1:Accept|x}$ が $x \in A_{2:P}$ に依存しないことから，

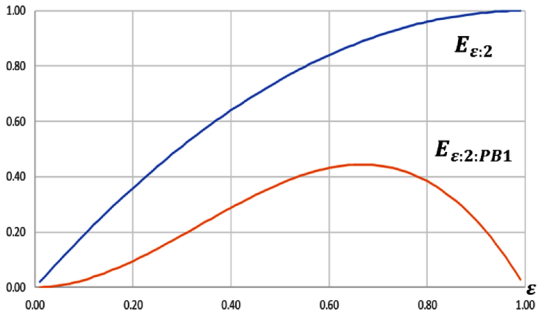

図 4　誤送信受諾率

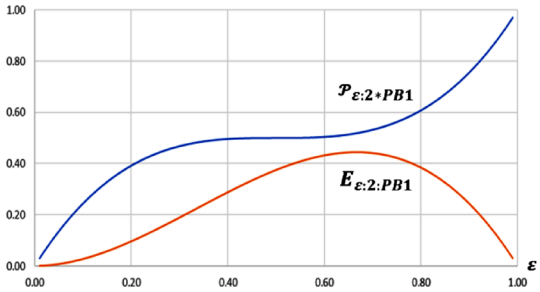

図 5　復号誤り率 vs. 誤送信受諾率

$$P(B_{2:\,PB1:\,Accept}) = \sum_{x\in A_{2:\,P}} P_X(x) P(B_{2:\,PB1:\,Accept|x}) = (1-\varepsilon)^3 + 3(1-\varepsilon)\varepsilon^2$$

を得る．同様に，この場合の復号誤り率 $\mathcal{P}_{\varepsilon:\,2:\,PB1}$ は，

$$\begin{aligned}\mathcal{P}_{\varepsilon:\,2:\,PB1} &= P(B_{2:\,PB1:\,Reject}) \\ &= 1 - P(B_{2:\,PB1:\,Accept}) \\ &= \varepsilon^3 + 3(1-\varepsilon)^2\varepsilon\end{aligned}$$

が成立する．

Parity Bit 方式を，リスク管理の観点から見てみよう．表 1 から誤送信受諾率は $E_{\varepsilon:\,2:\,PB1} = 3(1-\varepsilon)\varepsilon^2$ で与えられ，この確率で誤った情報が拡散する費用が発生する．$E_{\varepsilon:\,2:\,PB1}$ を Parity Bit を用いない場合と比べて見ると，

$$\begin{aligned}E_{\varepsilon:\,2} - E_{\varepsilon:\,2:\,PB1} &= \varepsilon(2-\varepsilon) - 3(1-\varepsilon)\varepsilon^2 \\ &= 3\varepsilon\left\{\left(\varepsilon - \frac{2}{3}\right)^2 + \frac{2}{9}\right\} > 0\end{aligned}$$

が成立，Parity Bit によってこの確率が減少していることがわかる．図 4 に，この比較をグラフで示す．2 元対称通信路の誤送信率 ε の値が 1 に近づくにつれて $E_{\varepsilon:\,2}$ は単調に 1 に近づくが，Parity Bit を導入すると $E_{\varepsilon:\,2:\,PB1}$ は 0 に近づく．これは，Parity Bit の判定アルゴリズムによってほとんどの受信結果が棄却されてしまい，そもそも受諾される受信結果が少なくなるので，誤送信受諾率も 0 に近づくためである．

一方，復号誤り率 $\mathcal{P}_{\varepsilon:\,2:\,PB1} = P(B_{2:\,PB1:\,Reject})$ は，この確率で再送に関わる費用を発生する．図 5 に，$E_{\varepsilon:\,2:\,PB1}$ と $\mathcal{P}_{\varepsilon:\,2:\,PB1}$ を比較したグラフを示す．たとえば，ε を 0.2 に抑えることができれば，誤った情報を拡散してしまう確率は 0.10 である一方，再送を要請する確率は 0.39 にも上る．シャノンは，ある条件下で，通信路符号に冗長ビットを付加することによって復号誤り率をいくらでも小さくできることを一般的な枠組で示したが，次節で論じるように，誤送信結果拡散コストと再送コストを巡るトレードオフのバランスを改善する手段を創出したのは，別の研究者であった．

冗長ビットをさらに増やす場合として，送信符号のビット列を $m \times n$ 行列で表わしてブロック化し，行と列のそれぞれについて Parity Bit を付与することを考える．$m = n = 2$ の場合，送信符号集合 $A_3 = \{(x_{11}, x_{12}, x_{21}, x_{22}) : x_{ij} = 0 \ or \ 1, i, j = 1, 2\}$ に対して Parity Bit を付与すると，実際に送信される通信路符号 $A_{3:\,P}$ は，$x_{ij} = 0 \ or \ 1, i, j = 1, 2$ として，以下のビット構造をもつ行列の集合となる．

x_{11}	x_{12}	$x_{11} \oplus x_{12}$
x_{21}	x_{22}	$x_{21} \oplus x_{22}$
$x_{11} \oplus x_{21}$	$x_{12} \oplus x_{22}$	

ここで $\oplus$ は排他的論理和 (XOR) を表わし，$0 \oplus 0 = 1 \oplus 1 = 0, 1 \oplus 0 = 0 \oplus 1 = 1$ と定義される．$A_{3:\,P}$ の受信通信路符号集合 $B_{3:\,P}$ は，$y_{ij}, b_{i\cdot}, b_{\cdot j} = 0 \ or \ 1, i, j = 1, 2$ として，以下のビット構造をもつ行列の集合になる．

y_{11}	y_{12}	$b_{1\cdot}$
y_{21}	y_{22}	$b_{2\cdot}$
$b_{\cdot 1}$	$b_{\cdot 2}$	

通信路符号がすべて正しく送信されるとき，Parity Bit の整合性は保たれるので，対応する受信通信路符号は受諾されるが，この確率は，$(1-\varepsilon)^8$ である．また，四つの送信符号部分が正しく送信されれば，たとえ Parity Bit に誤送信が発生しても判定アルゴリズムによって棄却され再送が依頼されるので，誤送信を正しいものとして受け入れてしまうことはない．したがって，誤送信を正しいと判定してしまう場合は，

① 送信符号の部分で少なくとも 1 bit の誤送信が

表 2　2 × 2 Block Parity における誤送信受諾

c c c c w w c w	c c c w w c w w	c w w w w c w c
w w c w w c c c	c c c w c w w c	c w w c w w c c
w c w w w c c w	c w w c w w c c	w c w w w c c w
c w w c c c c w	w c w c w w w w	w w c c w w w c
w c w c c c w c	c w w w c w w w	w w c w c w c w
w c w w c w c c		
w w c c c c w w		

表 3　誤通信の数に基づく表 2 のブロックの分類

	含まれる w の数				合計
	3	4	5	6	
第1列	4	0	0	0	4
第2列	0	4	0	2	6
第3列	0	0	4	0	4
第4列	0	1	0	0	1
合計	4	5	4	2	15

発生

② それを相殺するように行・列の Parity Bit で誤送信が発生

の場合に限られる．これらの場合を表 2 に示す．第 1 列には，送信符号に一つの誤送信が発生し，それを打ち消すように Parity Bit にも誤送信が発生，誤送信を正しいと判断してしまう場合を列挙してある．以下，同様に，第 2 列には送信符号に二つの誤送信が発生する場合，第 3 列には送信符号に三つの誤送信が発生する場合，第 4 列には送信符号の全てについて誤送信が発生する場合，を示している．

表 2 のブロック群を含まれる w の数で整理すると，表 3 を得る．この表 3 に基づいて誤送信受諾率 $E_{\varepsilon:\,2:\,BP}$ を求めると，

$$E_{\varepsilon:\,2:\,BP}= 4\varepsilon^3(1-\varepsilon)^5+5\varepsilon^4(1-\varepsilon)^4+4\varepsilon^5(1-\varepsilon)^3 +2\varepsilon^6(1-\varepsilon)^2 \tag{2.1}$$

となる．図 6 に誤送信受諾率 $E_{\varepsilon:\,2}$，$E_{\varepsilon:\,2:\,PB1}$，$E_{\varepsilon:\,2:\,BP}$ をグラフで表わす．冗長 bit を通常の Parity Bit から Block Parity 方式へ増大させることにより，誤送信受諾率が著しく減少することがわかる．

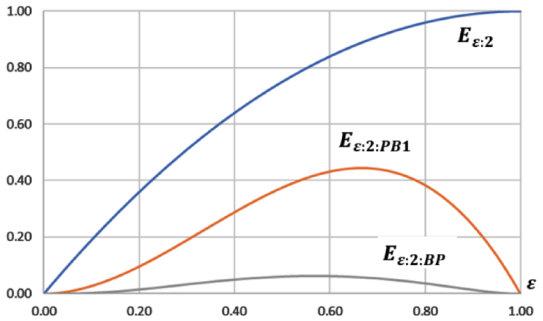

図 6　誤送信受諾率

以上より，2×2 Block Parity における復号誤り率 $\mathcal{P}_{\varepsilon:\,2:\,BP}$ を求めると，

$$\mathcal{P}_{\varepsilon:\,2:\,BP}= P(B_{2:\,BP:\,Reject}) = 1-P(B_{2:\,BP:\,Accept}) = 1-\left\{(1-\varepsilon)^8+E_{\varepsilon:\,2:\,BP}\right\}$$

を得る．図 7 に，$E_{\varepsilon:\,2:\,PB1}, E_{\varepsilon:\,2:\,BP}, \mathcal{P}_{\varepsilon:\,2:\,PB1}$, $\mathcal{P}_{\varepsilon:\,2:\,BP}$ をグラフで示す．1 Parity Bit を Block Parity Bits へと増やすことによって，$B_{\varepsilon:\,2:\,BP:\,Accept}$ が急激に縮小し $E_{\varepsilon:\,2:\,BP}$ が激減すると同時に，その分，復号誤り率 $\mathcal{P}_{\varepsilon:\,2:\,BP}$ が急激に増加している．これは，誤送信結果拡散コストと再送コストを巡るトレードオフに関し，前者を縮小させ後者を増大させていることを意味する．$\varepsilon = 0.2$ のとき $E_{\varepsilon:\,2:\,BP} = 0.0145$，$\mathcal{P}_{\varepsilon:\,2:\,BP} = 0.8177$ であり，誤送信結果が拡散する割合は 1.5%弱であるのに比べ，80%を超える送信通信路符号が再送されることになる．

この原因は，Parity Bit 方式によって誤送信を検出することはできても，訂正することはできないことに起因する．誤送信を検出後，誤りが訂正できれば，すべての受信を受諾し，再送コストを 0 とすることができる．ただし，すべての誤りを正しく訂正できるとは限らず，どれだけ誤送信受諾率を抑え，それによって誤送信結果が拡散するコストを制御できるかが鍵となる．

3.　ハミング符号：誤送信検出・訂正符号

1950 年，送信された符号語の成分に生じた 1 個までの誤りを検出・訂正できる符号を開発したのは，アメリカの AT&T ベル研究所（当時）の研究者であったリチャード・ハミング (Richard W. Hamming) [77] である．1915 年，シカゴで生まれたハミングは，技術者になることを志したが，1929 年の大恐慌後のアメリカでは奨学金が著しく削減されており，ようやくのことでシカゴ大学 (University of Chicago) から奨学金を

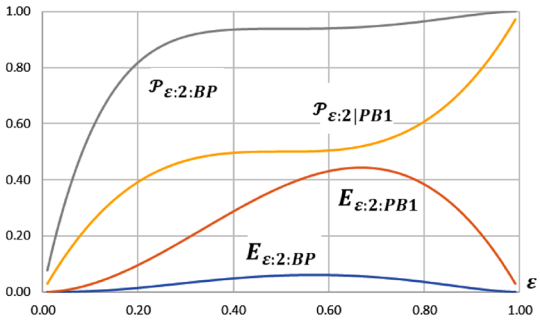

図 7　1 Parity Bit vs Block Parity Bits

得たが，当時のシカゴ大学には工学部がなかった．このことは，コンピュータ・通信科学にとって僥倖であった．仕方なく数学を専攻したハミングは，1937 年に学部を卒業後，1939 年，ネブラスカ大学 (University of Nebraska) で修士号を修め，1942 年にはイリノイ大学アーバナ・シャンペーン校 (University of Illinois at Urbana-Champaign) から博士号を授与された．そのまま 2 年間，イリノイ大学で数学講師を務めた後，1944 年，ルイビル大学 (University of Louisville) で数学助教授の職を得た．第 2 次世界大戦が戦われている中，1945 年，ロスアラモス研究所 (Los Alamos Laboratory) に招請されてマンハッタン計画 (Manhattan Project) に参画，IBM の計算機を使って物理学者が要請するさまざまな方程式の数値解を計算する仕事に従事した．後に，何やらわからぬままに核爆弾の起爆確率を計算させられたと知ってショックを受けたと述懐している．

終戦後の 1946 年，ハミングは弾力性理論の研究を担当する研究者としてベル研究所に就職した．そこで，シャノンとオフィスを共有することがあったと伝えられている．ベル研究所では，マンハッタン計画での経験を買われて，本来の研究はそっちのけで同僚が必要とする複雑な数値計算を引き受けることで忙しかった．

1947 年のある金曜日，ハミングは長く複雑な計算を週末の間に実行させるべく計算機を設定してから帰宅した．月曜日に研究所に出勤すると，計算の早い段階で誤作動が起きたことが判明し，それ以後のすべての計算が無駄になったことを知った．ハミングは，Parity Bit によって誤作動の発生がわかるのであれば，それがどの箇所で起きたかを知り訂正する手立てもあるはずだと考え，この研究に取り組むようになる．1950 年，その成果を論文 “Error detecting and error correcting codes”（誤送信検出・訂正符号）として Bell System Technical Journal 4 月号に発表，コンピュータ・通信科学の分野で，シャノンと並び称される名声を確立した．以下，その概略を解説する．

まず，$\{0,1\}$ 上での四則演算 $\oplus, \ominus, \otimes, \oslash$ を『通常の整数の演算結果を 2 で割った余り』と定義することから始める．具体的には，以下のとおりに定義される．

$\oplus$	0	1
0	0	1
1	1	0

$\ominus$	0	1
0	0	1
1	1	0

$\otimes$	0	1
0	0	0
1	0	1

$\oslash$	0	1
0		0
1		1

たとえば，$1 \oplus 1$ は，$1+1=2$ を 2 で割った余り 0 となる．この $\oplus$ は，前述した排他的論理和 (XOR) と一致する．同様に，$0-1=-1=(-1)\times 2+1$ と書けるので，$0 \ominus 1 = 1$ となる．$\oplus$ と $\ominus$ が全く同等であることを注意しておく．乗法と除法は，通常の整数の演算と一致する．一般的に，四則演算が定義される集合を体と呼ぶ．

2 以上の整数 m に対して，0 と 1 を要素にもつ m 次元行ベクトルの集合を $\hat{F}_2^m$，そこから $\mathbf{0}$ ベクトルを除去した集合を $F_2^m = \hat{F}_2^m \setminus \{\mathbf{0}\}$ と定義する．以下，$\hat{F}_2^m$ 上での四則演算は，$\oplus, \ominus, \otimes, \oslash$ に従うものとする．いま，送信符号が $\boldsymbol{v_I} \in F_2^{n-m}$ として与えられているとする．ここで，$n = 2^m - 1$ である．このとき，m Parity bits からなるベクトル $\boldsymbol{v_P} \in \hat{F}_2^m$ を付加し，送信通信路符号 $\boldsymbol{v} = [\boldsymbol{v_I}, \boldsymbol{v_P}]$ を生成することを考える．

今後，前者を情報ベクトル，後者を Parity ベクトルと呼ぶことにする．ハミングが目指したのは，$\boldsymbol{v}$ を送信した際，誤送信が 1 ヶ所でしか起こらないとして，それを検出し訂正できるように $\boldsymbol{v_P}$ を定める方式を確立することにあった．

m 次元行ベクトルの集合を $R^{\boldsymbol{m}}$，$m \times n$ 行列の集合を $R^{m\times n}$ と表わす．ハミングは，まず，F_2^m のすべての要素を列ベクトルとしてもつ行列 $\boldsymbol{H}$ をハミング Parity 検査行列と定義した．便宜上，m 次元単位行列を右側に置くと定め，

$$\boldsymbol{H} = [\boldsymbol{W}, \boldsymbol{I_m}] \in R^{m\times n} \tag{3.1}$$

と表わすことにする．ここで，$\boldsymbol{W} \in R^{m\times(n-m)}$ は単位ベクトルを除いた F_2^m の要素を横に辞書式順序[1]の昇順にしたがって並べた行列，$\boldsymbol{I_m}$ は m 次元単位行列である．$m = 3$ の場合は以下のようになる．

$$\boldsymbol{H} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

ハミングの着想は，送信通信路符号を $\boldsymbol{H}$ の零空間に限定することにあった．すなわち，送信通信路符号の集合 A_H を

$$A_H = \{\boldsymbol{v} \in F_2^n : \boldsymbol{v}\boldsymbol{H}^T = \boldsymbol{0}\} \tag{3.3}$$

と限定したのである．ここで，生成行列と呼ばれる行列 $\boldsymbol{G}$ を

$$\boldsymbol{G} = \left[\boldsymbol{I_{n-m}}, \boldsymbol{W^T}\right] \in R^{(\boldsymbol{n-m})\times\boldsymbol{n}} \tag{3.4}$$

と定義すると，次の定理が成立する．

定理 3.1

$\boldsymbol{v} = [\boldsymbol{v_I}, \boldsymbol{v_P}] \in F_2^n$ とする．このとき，
$\boldsymbol{v} \in A_H \Longleftrightarrow$ ある $\boldsymbol{u} \in F_2^{n-m}$ が存在して，$\boldsymbol{v} = \boldsymbol{uG}$

[証明]

$\boldsymbol{v} \in A_H$ とすると，(3.1), (3.3) 式より，

$$\boldsymbol{v}H^T = [\boldsymbol{v_I}, \boldsymbol{v_P}] \begin{bmatrix} \boldsymbol{W}^T \\ \boldsymbol{I_m} \end{bmatrix} = \boldsymbol{v_I}\boldsymbol{W}^T \oplus \boldsymbol{v_P} = \boldsymbol{0}$$

となり，これより

$$\boldsymbol{v_P} = \boldsymbol{0} \ominus \boldsymbol{v_I}\boldsymbol{W}^T = \boldsymbol{0} \oplus \boldsymbol{v_I}\boldsymbol{W}^T = \boldsymbol{v_I}\boldsymbol{W}^T$$

を得る．したがって，$\boldsymbol{u} = \boldsymbol{v_I}$ とおくと (3.4) 式より，

$$\boldsymbol{uG} = \boldsymbol{v_I}\boldsymbol{G} = \left[\boldsymbol{v_I}, \boldsymbol{v_I}\boldsymbol{W}^T\right] = [\boldsymbol{v_I}, \boldsymbol{v_P}] = \boldsymbol{v}$$

となって，$\Longrightarrow$ が成立する．

逆に任意の $\boldsymbol{u} \in F_2^{n-m}$ に対して $\boldsymbol{v} = \boldsymbol{uG}$ とすると，

$$\begin{aligned} \boldsymbol{v}\boldsymbol{H}^T = (\boldsymbol{uG})\,\boldsymbol{H}^T &= \boldsymbol{u}\left[\boldsymbol{I_{n-m}}, \boldsymbol{W^T}\right] \begin{bmatrix} \boldsymbol{W}^T \\ \boldsymbol{I_m} \end{bmatrix} \\ &= \boldsymbol{u}\left[\boldsymbol{W^T} \oplus \boldsymbol{W^T}\right] \end{aligned}$$

が成立する．ここで，$\boldsymbol{W^T} \oplus \boldsymbol{W^T}$ は要素がすべて 0 の行列となるので $\boldsymbol{v}\boldsymbol{H}^T = \boldsymbol{0}$，よって定理が証明された．□

送信符号集合を A とすると，$\boldsymbol{v_I} \in A$ に対して $\boldsymbol{v} = \boldsymbol{v_I}\boldsymbol{G}$ と変換すると，定理 3.1 より $\boldsymbol{v} \in A_H \subset F_2^n$ が成立する．これを送信する際，ノイズ発生による誤送信発生の可能性を考慮し，$\boldsymbol{e} \in \hat{F}_2^n$ として，$\boldsymbol{r} = \boldsymbol{v} + \boldsymbol{e} \in F_2^n$ が受信されたとしよう．誤送信が高々 1 ヶ所でしか発生しないと仮定すると，発生しない場合は $\boldsymbol{e} = \boldsymbol{0}$，第 j 要素で発生した場合は $\boldsymbol{e}$ は n 次元の第 j 単位ベクトルとなる．

受信側で $\boldsymbol{r}$ に右側から $\boldsymbol{H}^T$ を掛ける変換を施すと

$$\boldsymbol{r}\boldsymbol{H}^T = (\boldsymbol{v} + \boldsymbol{e})\,\boldsymbol{H}^T = \boldsymbol{v}\boldsymbol{H}^T + \boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{0} + \boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T$$

となる．この結果，$\boldsymbol{e}\boldsymbol{H}^T$ が $\boldsymbol{0}$ ベクトルである場合は，$\boldsymbol{r} \in A_H$ となり，そのまま $\boldsymbol{v_I}$ を受諾する．$\boldsymbol{e}\boldsymbol{H}^T$ がハミング Parity 検査行列 $\boldsymbol{H}$ の第 j 列と一致した場合は，$\boldsymbol{e}$ は n 次元の第 j 単位ベクトルであることを意味し，そこで誤送信が発生しているので，$\boldsymbol{v}$ の第 j 要素の 0-1 を反転させたうえで，$\boldsymbol{v_I}$ を受諾する．以上を，ハミング符号化・復号アルゴリズムとしてまとめておく．

ハミング符号化・復号アルゴリズム

[1] 情報ベクトル $\boldsymbol{v_I} \in A$ に対し，送信通信路ハミング符号 $\boldsymbol{v} = \boldsymbol{v_I}\boldsymbol{G} \in A_H$ を生成し，それを送信
[2] 受信通信路符号 $\boldsymbol{r}$ に 復号変換 $\boldsymbol{r}\boldsymbol{H}^T$ を実行
[3] $\boldsymbol{r}\boldsymbol{H}^T = \boldsymbol{0}$ であれば，$\boldsymbol{v_I}$ をそのまま受諾
[4] $\boldsymbol{r}\boldsymbol{H}^T \neq \boldsymbol{0}$ であれば，$\boldsymbol{r}\boldsymbol{H}^T$ はハミング Parity 検査行列 $\boldsymbol{H}$ のどれかの列ベクトルと一致する；第

[1] 辞書式順序：$\boldsymbol{a} = [a_1, \cdots, a_n], \boldsymbol{b} = [b_1, \cdots, b_n] \in R^{\boldsymbol{n}}$ に対し，$a_0 = b_0$ としたとき，ある $j \in \{1, \cdots, n\}$ が存在して，$a_i = b_i, i = 0, \cdots, j-1; a_j < b_j$ が成立するとき，$\boldsymbol{b}$ は辞書式順序で $\boldsymbol{a}$ より大きいと呼び，$\boldsymbol{a} \prec_L \boldsymbol{b}$ と書く．ここで，L は Lexicographically larger（辞書式で大きい）の頭文字である．

j 列と一致した場合は，$\boldsymbol{v}$ の第 j 要素の 0-1 を反転させたうえで受諾

ハミング符号化・復号アルゴリズムは，誤送信が高々 1 ヶ所でしか発生しない場合，すなわち，$\boldsymbol{e}=\boldsymbol{0}$ か $\boldsymbol{e}$ が n 次元単位ベクトルである場合は，正しい情報ベクトルを受信できるが，後述するように，2 ヶ所以上で誤送信が発生すれば，誤った情報ベクトルを正しいものとして受信してしまう．これを具体的に見るため，(3.2) 式に与えられた $\boldsymbol{H}$ を用いて，$\boldsymbol{CaseI}\sim\boldsymbol{IV}$ まで，ハミング符号化・復号アルゴリズムの四つの数値例を示す．

[1] $\boldsymbol{v_I}=[1,1,0,1]$

$$\boldsymbol{v}=\boldsymbol{v_I}\boldsymbol{G}=[1,1,0,1]\begin{bmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{bmatrix}$$

$$=[1,1,0,1,0,0,1]$$

CaseI

[2] $\boldsymbol{e}=[0,0,1,0,0,0,0]$ の誤送信が発生

$\boldsymbol{r}=\boldsymbol{v}+\boldsymbol{e}=[1,1,1,1,0,0,1]$

$$\boldsymbol{rH^T}=\boldsymbol{r}\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}^{\boldsymbol{T}}=[1,1,0]$$

[3] $\boldsymbol{rH^T}$ は $\boldsymbol{H^T}$ の 3 列目と同じなので，$\boldsymbol{r}=[1,1,1,1,0,0,1]$ を $\boldsymbol{r}=[1,1,0,1,0,0,1]$ と訂正して $[1,1,0,1]$ を受諾（誤送信の訂正に成功）

CaseII

[2] $\boldsymbol{e}=[1,0,1,0,0,0,0]$ の誤送信が発生

$\boldsymbol{r}=[0,1,1,1,0,0,1]$

$$\boldsymbol{rH^T}=\boldsymbol{r}\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}^{\boldsymbol{T}}=[1,0,1]$$

[3] $\boldsymbol{rH^T}$ は $\boldsymbol{H^T}$ の 2 列目と同じなので，$\boldsymbol{r}=[0,1,1,1,0,0,1]$ を $\boldsymbol{r}=[0,0,1,1,0,0,1]$ と訂正して $[0,0,1,1]$ を受諾（誤送信を受諾）

CaseIII

[2] $\boldsymbol{e}=[0,0,1,0,0,1,0]$ の誤送信が発生

$\boldsymbol{r}=[1,1,1,1,0,1,1]$

$$\boldsymbol{rH^T}=\boldsymbol{r}\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}^{\boldsymbol{T}}=[1,0,0]$$

[3] $\boldsymbol{rH^T}$ は $\boldsymbol{H^T}$ の 5 列目と同じなので，$\boldsymbol{r}=[1,1,1,1,0,1,1]$ を $\boldsymbol{r}=[1,1,1,1,1,1,1]$ と訂正して $[1,1,1,1]$ を受諾（誤送信を受諾）

CaseIV

[2] $\boldsymbol{e}=[0,0,0,0,1,0,1]$ の誤送信が発生

$\boldsymbol{r}=[1,1,0,1,1,0,0]$

$$\boldsymbol{rH^T}=\boldsymbol{r}\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}^{\boldsymbol{T}}=[1,0,1]$$

[3] $\boldsymbol{rH^T}$ は $\boldsymbol{H^T}$ の 2 列目と同じなので，$\boldsymbol{r}=[1,1,0,1,1,0,0]$ を $\boldsymbol{r}=[1,0,0,1,0,0,0]$ と訂正して $[1,0,0,1]$ を受諾（誤送信を受諾）

$\boldsymbol{rH}^T\neq\boldsymbol{0}$ の場合，$\boldsymbol{rH}^T\in F_2^n$ となるので，$\boldsymbol{rH}^T$ は必ず $\boldsymbol{H}^T$ のどこかの列と一致し，$\boldsymbol{r}$ の該当箇所を 1 ヶ所だけ訂正したうえで受諾することになる．したがって，ハミング符号化・復号アルゴリズムでは，送信されたすべての通信路符号を受諾することになる．ここで問題となるのは，受諾した情報ベクトルの結果が正しい確率である．

① $\boldsymbol{rH}^T=\boldsymbol{0}$ の場合，誤送信は発生しておらず，正しい情報ベクトルが受信される

② 誤送信が 1 ヶ所で発生した場合，$\boldsymbol{rH}^T$ は $\boldsymbol{H^T}$ の発生箇所に該当する列ベクトルと一致し，訂正が適正に行われ，正しい情報ベクトルが受信される ($\boldsymbol{CaseI}$)

③ 誤送信が情報ベクトルの 2 ヶ所以上のみで発生した場合，ハミング符号化・復号アルゴリズムでは高々 1 ヶ所の訂正しか行わないので，誤った情報ベクトルが受信される ($\boldsymbol{CaseII}$)

④ 誤送信が情報ベクトルの 1 ヶ所と Parity ベクトルの 1 ヶ所以上で発生した場合，後者の発生箇所に該当する単位ベクトルが，$\boldsymbol{H^T}$ の前者の該当箇所の列ベクトルに加算されるので，その結果が同じ $\boldsymbol{H^T}$ の列ベクトルになることはない；したがって，情報ベクトルの誤送信は訂正されないままになり，誤った情報ベクトルが受信される ($\boldsymbol{CaseIII}$)

⑤ 誤送信が Parity ベクトルの 2 ヶ所以上のみで発生した場合，単位ベクトルの和は単位ベクトルとはならないので，$\boldsymbol{rH}^T$ は $\boldsymbol{H^T}$ の情報ベクトルに対応する列ベクトルのどれかと一致する；したがって，情報ベクトルの該当箇所に誤った 0-1 変換が施されることになり，誤った情報ベクトルが受信される ($\boldsymbol{CaseIV}$)

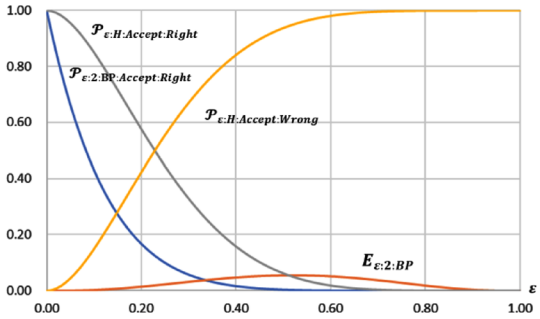

図 8　Block Parity 方式 vs. ハミング方式

表 4　Block Parity 方式 vs. ハミング符号

ε	Block Parity 方式			ハミング方式	
	送信成功確率	誤通信受諾確率	再送確率	送信成功確率	誤通信受諾確率
0.00	1.00000	0.00000	0.00000	1.00000	0.00000
0.01	0.92274	0.00000	0.07725	0.99797	0.00203
0.02	0.85076	0.00003	0.14921	0.99214	0.00786
0.03	0.78374	0.00010	0.21616	0.98291	0.01709
0.04	0.72139	0.00022	0.27839	0.97062	0.02938
0.05	0.66342	0.00041	0.33617	0.95562	0.04438
0.06	0.60957	0.00069	0.38974	0.93822	0.06178
0.07	0.55958	0.00105	0.43937	0.91873	0.08127
0.08	0.51322	0.00151	0.48527	0.89741	0.10259
0.09	0.47025	0.00206	0.52768	0.87452	0.12548
0.10	0.43047	0.00272	0.56681	0.85031	0.14969

以上より，ハミング符号化・復号アルゴリズムでは，誤送信が高々 1 ヶ所で発生した場合にのみ正しい情報ベクトルが受信され，2 ヶ所以上で誤送信が発生した場合は，誤った情報ベクトルが受信されることがわかる．これにより，以下の定理が成立する．

定理 3.2

2 以上の整数 m に対して $n=2^m-1$ とし，長さ $n-m$ の情報ベクトルを，ハミング符号化・復号アルゴリズムに基づいて，2 元対称通信路を通して送信するとする．このとき，情報ベクトルが正しく受信される確率 $\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Right}$ と，誤って受信される確率 $\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Wrong}$ は，次の式で与えられる．

$$\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Right}=(1-\varepsilon)^n+n\varepsilon(1-\varepsilon)^{n-1}$$
$$\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Wrong}=1-\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Right}$$

$m=3, n=7$ の場合，情報ベクトルの長さは $n-m=4$ となる．このとき，Block Parity 方式の結果である $\mathcal{P}_{\boldsymbol{\varepsilon}:\,2:\,BP:\,Accept:\,Right}=(1-\varepsilon)^8$ と (2.1) 式の $E_{\varepsilon:\,2:\,BP}$ を，ハミング方式の結果である $\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Right}$ と $\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Wrong}$ に対比した結果を，図 8 にグラフで示す．$E_{\varepsilon:\,2:\,BP}$ と $\mathcal{P}_{\varepsilon:\,H:\,Accept:\,Wrong}$ は，ともに，誤送信の結果が拡散してしまうリスクを表わしている．Block Parity 方式の場合は

$$\mathcal{P}_{\varepsilon:\,2:\,BP}=1-(\mathcal{P}_{\boldsymbol{\varepsilon}:\,2:\,BP:\,Accept:\,Right}+E_{\varepsilon:\,2:\,BP})$$

の確率で，再送に必要となるコストが隠されていることに注意する必要がある．この点を明確にするため，表 4 にそれぞれの確率を示してある．$0<\varepsilon\leq 0.05$ 程度であれば，ハミング方式の優位性は明らかである．

1 個の誤送信を検出・訂正できるのであれば，複数個の誤送信を検出・訂正できるアルゴリズムを追い求めることは自然である．これが実現できれば，より大きな ε の値にも対応できることになる．この方向での研究のアイデアは，必要となる情報ベクトルの数 $|A|$ は応用環境で決まるので，送信通信路ハミング符号 $\boldsymbol{v}\in A_H$ の次元を拡大することによって，A_H の要素間の距離を離すことができるという点にある．

具体的には，$\boldsymbol{x},\boldsymbol{y}\in F_2^n$ のハミング距離 $d_H(\boldsymbol{x},\boldsymbol{y})$ を二つのベクトルの異なる要素の数，すなわち，

$$d_H(\boldsymbol{x},\boldsymbol{y})=|\{j\colon x_j\neq y_j, j=1,\cdots,n\}| \quad (3.5)$$

と定義する．いま，$i=1,\cdots,|A|$ に対し，情報ベクトル $\boldsymbol{v_I}(i)\in A$ に対応して，送信通信路ハミング符号 $\boldsymbol{v}(j)\in A_H\subset F_2^n$ が生成されるとする．ここで，A_H に属するベクトル間の最小距離 $d(A_H)$ を

$$d(A_H)\min_{\substack{1\leq i,j\leq n\\ i\neq j}} d_H(\boldsymbol{v}(i),\boldsymbol{v}(j)) \quad (3.6)$$

と定義すると，この値は送信通信路符号に生じた誤りを何個まで訂正できるかの指標となる．すなわち，$\boldsymbol{v_I}(i)\in A$ から生成された $\boldsymbol{v}(i)\in A_H\subset F_2^n$ が送信された結果，誤送信の可能性を含めて $\boldsymbol{r}=\boldsymbol{v}+\boldsymbol{e}\in F_2^n$ が受信されたとする．すると，少なくとも $d(A_H)$ 個の誤送信が発生しない限り $\boldsymbol{r}\notin A_H$ とはならないので，これによって $d(A_H)-1$ 個までの誤送信が検出できることになる．

さらに，$\boldsymbol{v}(i)\in A$ を中心とするハミング距離で半径 t の球 $S(\boldsymbol{v}(i))$ を

$$S(\boldsymbol{v}(i))=\{\boldsymbol{x}\in F_2^n\colon d_H(\boldsymbol{x},\boldsymbol{v}(i))\leq t\} \quad (3.7)$$

と定義し，$d(A_H)-1$ を 2 で割った際の最大整数として $t=\lfloor\{d(A_H)-1\}/2\rfloor$ と設定すると，$S(\boldsymbol{v}(i))$ は $\boldsymbol{v}(i)\in A_H$ に高々 t 個までの誤送信が発生した際に得られる受信通信路符号の集合となる．また，

$$2t+1=2\lfloor\{d(A_H)-1\}/2\rfloor+1$$
$$\leq\{d(A_H)-1\}+1=d(A_H)$$

となるので，

$$S(\boldsymbol{v_I}(i)) \cap S(\boldsymbol{v_I}(j)) = \phi, i \neq j \qquad (3.8)$$

が成立する．したがって，受信通信路符号について $\boldsymbol{r} \in S(\boldsymbol{v}(i))$ であれば，ハミング距離で最短にある送信通信路符号 $\boldsymbol{v}(i)$ が特定できることになる．

以上より，複数個の誤送信を検出・訂正するアルゴリズムを確立する問題は，

① 符号語数 $|A|$ と通信路符号長が与えられたとき，送信通信路符号 A_H を $d(A_H)$ が最大になるように定めること

② 受信通信路符号 $\boldsymbol{r}$ に対し，$\boldsymbol{r} \in S(\boldsymbol{v}(i))$ を満たす $i \in \{1, \cdots, n\}$ を決定し，見つかれば $\boldsymbol{r}$ を $\boldsymbol{v}(i)$ で置き換えて受諾；$\boldsymbol{r} \notin A_H$ と判定されれば何もしないか，再送を依頼すること

という問題に帰着する．こうした誤り訂正符号に関する研究は 1960 年代に注目を集め，現在では符号理論と呼ばれる独立した研究分野を構成している．興味ある読者は，文献 [78] を参照されたい．

次回は，これまで述べてきた受信側の理論を特殊な事例として含むシャノンの第二基本定理を，より一般的な枠組で紹介する．

参考文献

[1] H. Goldstine, The Computer from Pascal to von Neumann, Princeton University Press, 1972.（末包良太，米口肇，犬伏茂之訳，『復刊　計算機の歴史―パスカルからノイマンまで―』，共立出版，2016.）

[2] S. McCartney, The Triumphs and Tragedies of the World's First Computer, Walker, 1999.（日暮雅通訳，『エニアック―世界最初のコンピュータ開発秘話―』，パーソナルメディア，2001.）

[3] 坂村健，『痛快！　コンピュータ学』，集英社，1999（文庫版 2002）.

[4] 竹内伸，『実物でたどるコンピュータの歴史―石ころからリンゴへ―』，東京理科大学出版センター(編)，東京書籍，2012.

[5] 小田徹，『コンピュータ開発のはてしない物語―起源から驚きの近未来まで―』，技術評論社，2016.

[6] Wikipedia, Francois Viéte, https://enwikipedia.org/wiki/Francois_Viéte（2021 年 12 月 14 日閲覧）

[7] E. T. Bell, *Men of Mathematics Volume 1*, Simon & Schuster, 1937.（田中勇・銀林浩訳，『数学をつくった人びと上』，東京図書，1976.）

[8] Wikipedia, René Descartes, https://en.wikipedia.org/wiki/René_Descartes（2021 年 12 月 21 日閲覧）

[9] E. T. Bell, *Men of Mathematics Volume 2*, Simon & Schuster, 1937.（田中勇・銀林浩訳，『数学をつくった人びと下』，東京図書，1976.）

[10] Wikipedia, George Boole, https://en.wikipedia.org/wiki/George_Boole（2021 年 12 月 14 日閲覧）

[11] P. J. Nahin, *The Logician and the Engineer: How George Boole and Claude Shannon Created the Information Age*, Princeton University Press, 2012.（松浦俊輔訳，『0 と 1 の話―ブール代数とシャノン理論―』，青土社，2013.）

[12] J. Soni and R. Goodman, *A Mind at Play: How Claude Shannon Invented the Information Age*, Simon & Schuster, 2017.（小坂恵理訳，『クロード・シャノン―情報時代を発明した男―』，筑摩書房，2019.）

[13] Wikipedia, Claude Shannon, https://en.wikipedia.org/wiki/Claude_Shannon（2021 年 12 月 20 日閲覧）

[14] Wikipedia, Alan Turing, https://en.wikipedia.org/wiki/Alan_Turing（2021 年 12 月 20 日閲覧）

[15] B. J. Copeland,*Turing: Pioneer of the Information Age*, Oxford University Press, 2012.（服部桂訳，『チューリング―情報時代のパイオニア―』，NTT 出版，2013.）

[16] A. Hodges, *Alan Turing: The Enigma*, Princeton University Press, 2014.（土屋俊・土屋希和子訳，『エニグマ―アラン・チューリング伝―』，勁草書房，2015.）

[17] 高岡詠子，『チューリングの計算理論入門―チューリング・マシンからコンピュータへ―』，講談社，2014.

[18] Wikipedia, Galileo Galilei, https://en.wikipedia.org/wiki/Galileo_Galilei（2021 年 12 月 21 日閲覧）

[19] Wikipedia, Nicolaus Copernicus, https://en.wikipedia.org/wiki/Nicolaus_Copernicus（2021 年 12 月 21 日閲覧）

[20] Wikipedia, Marin Mersenne, https://en.wikipedia.org/wiki/Marin_Mersenne（2021 年 12 月 21 日閲覧）

[21] Wikipedia, Isaac Beeckman, https://en.wikipedia.org/wiki/Isaac_Beeckman（2022 年 1 月 2 日閲覧）

[22] Wikipedia, Adrien Baillet, https://en.wikipedia.org/wiki/Adrien_Baillet（2022 年 1 月 2 日閲覧）

[23] Wikipedia, Elisabeth of the Palatinate, https://en.wikipedia.org/wiki/Elisabeth_of_the_Palatinate（2022 年 1 月 2 日閲覧）

[24] Wikipedia，エリーザベト・フォン・デア・プファルツ (1618-1680), https://ja.wikipedia.org/wiki/ エリーザベト・フォン・デア・プファルツ (1618-1680)（2022 年 1 月 2 日閲覧）

[25] 有賀暢迪，“合理力学の一例としての衝突理論　1720–1730 年”，科学哲学科学史研究，**6**, pp. 17–37, 2012.

[26] Wikipedia，ソディの 6 球連鎖，https://ja.wikipedia.org/wiki/ソディの 6 球連鎖（2022 年 1 月 4 日閲覧）

[27] Wikipedia, Thorold Gosset, https://en.wikipedia.org/wiki/Thorold_Gosset（2022 年 1 月 4 日閲覧）

[28] 寒川町ガイド，https://samukawaguide.blogspot.com/2019/12/6.html（2022 年 1 月 4 日閲覧）

[29] Wikipedia, Gottfried Wilhelm Leibniz, https://en.wikipedia.org/wiki/Gottf_ried_Wilhelm_Leibniz（2022 年 1 月 4 日閲覧）

[30] Wikipedia, Christina, Queen of Sweden, https://en.wikipedia.org/wiki/Christina,_Queen_of_Sweden（2022 年 1 月 4 日閲覧）

[31] Wikipedia, Isaac Newton, https://en.wikipedia.org/wiki/Isaac_Newton（2022 年 1 月 4 日閲覧）

[32] 向井茂，“不変式の話，”数学セミナー連載，2005 年 12 月号，2006 年 1，2，4 月号.

[33] 日本医学会ホームページ，https://jams.med.or.jp/news/013.html（2022 年 2 月 4 日閲覧）

[34] Wikipedia, Vannevar Bush, https://en.wikipedia.org/wiki/Vannevar_Bush（2022 年 2 月 25 日閲覧）

[35] Britanica, William-Thomson-Baron-Kelvin, https://www.britannica.com/biography/William-Thomson!-Baron-Kelvin（2022 年 3 月 6 日閲覧）

[36] Wikipedia, Hannibal Ford, https://en.wikipedia.org/wiki/Hannibal_Ford（2022 年 3 月 6 日閲覧）

[37] Wikipedia, Joseph Fourier, https://en.wikipedia.org/wiki/Joseph_Fourier（2022 年 3 月 6 日閲覧）
[38] Wikipedia, ユトランド沖海戦, https://ja.wikipedia.org/wiki/ユトランド沖海戦（2022 年 3 月 6 日閲覧）
[39] Wikipedia, Mark I Fire Control Computer, https://en.wikipedia.org/wiki/Mark_I_Fire_Control_Computer（2022 年 3 月 7 日閲覧）
[40] Wikipedia, Bell Labs, https://en.wikipedia.org/wiki/Bell_Labs（2022 年 4 月 7 日閲覧）
[41] Wikipedia, Thornton Carle Fry, https://en.wikipedia.org/wiki/Thornton_Carle_Fry（2022 年 4 月 7 日閲覧）
[42] Wikipedia, Schön scandal, https://en.wikipedia.org/wiki/Schön_scandal（2022 年 4 月 7 日閲覧）
[43] Wikipedia, ヘンドリック・シェーン, https://ja.wikipedia.org/wiki/ヘンドリック・シェーン（2022 年 4 月 7 日閲覧）
[44] Wikipedia, ジョン・フォン・ノイマン, https://ja.wikipedia.org/wiki/ジョン・フォン・ノイマン（2022 年 4 月 29 日閲覧）
[45] Wikipedia, ヘルマン・ワイル, https://ja.wikipedia.org/wiki/ヘルマン・ワイル（2022 年 4 月 29 日閲覧）
[46] Wikipedia, 第二次世界大戦, https://ja.wikipedia.org/wiki/第二次世界大戦（2022 年 5 月 31 日閲覧）
[47] Wikipedia, フランクリン・ルーズベルト, https://ja.wikipedia.org/wiki/フランクリン・ルーズベルト（2022 年 4 月 30 日閲覧）
[48] Wikipedia, ウォーレン・ウィーバー, https://ja.wikipedia.org/wiki/ウォーレン・ウィーバー（2022 年 5 月 3 日閲覧）
[49] Wikipedia, ジェイムス・コナント, https://ja.wikipedia.org/wiki/ジェイムス・コナント（2022 年 5 月 3 日閲覧）
[50] Wikipedia, ロバート・オッペンハイマー, https://ja.wikipedia.org/wiki/ロバート・オッペンハイマー（2022 年 5 月 3 日閲覧）
[51] Wikipedia, Homer Dudley, https://en.wikipedia.org/wiki/Homer_Dudley（2022 年 4 月 7 日閲覧）
[52] Wikipedia, SIGSALY, https://ja.wikipedia.org/wiki/SIGSALY（2022 年 5 月 31 日閲覧）
[53] Wikipedia, ワンタイムパッド, https://ja.wikipedia.org/wiki/ワンタイムパッド（2022 年 5 月 3 日閲覧）
[54] 釜賀一夫, 藤原邦樹, 吉村昭, “座談会日本陸軍暗号はなぜ破られなかったか,” 歴史と人物―太平洋戦争シリーズ―, 昭和 60 年冬号, 1985.
[55] Wikipedia, Harry Nyquist, https://en.wikipedia.org/wiki/Harry_Nyquist（2022 年 5 月 7 日閲覧）
[56] Wikipedia, Ralph Hartley, https://en.wikipedia.org/wiki/Ralph_Hartley（2022 年 5 月 7 日閲覧）
[57] Wikipedia, ニコラ・レオナール・サディ・カルノー, https://ja.wikipedia.org/wiki/ニコラ・レオナール・サディ・カルノー（2022 年 6 月 6 日閲覧）
[58] Wikipedia, ジェームズ・プレスコット・ジュール, https://ja.wikipedia.org/wiki/ジェームズ・プレスコット・ジュール（2022 年 6 月 6 日閲覧）
[59] Wikipedia, ユリウス・ロベルト・フォン・マイヤー, https://ja.wikipedia.org/wiki/ユリウス・ロベルト・フォン・マイヤー（2022 年 6 月 6 日閲覧）
[60] Wikipedia, ヘルマン・フォン・ヘルムホルツ, https://ja.wikipedia.org/wiki/ヘルマン・フォン・ヘルムホルツ（2022 年 6 月 6 日閲覧）
[61] Wikipedia, ルドルフ・クラウジウス, https://ja.wikipedia.org/wiki/ルドルフ・クラウジウス（2022 年 6 月 6 日閲覧）
[62] Wikipedia, ジェームズ・クラーク・マクスウェル, https://ja.wikipedia.org/wiki/ジェームズ・クラーク・マクスウェル（2022 年 6 月 6 日閲覧）
[63] Wikipedia, ルートヴィッヒ・ボルツマン, https://ja.wikipedia.org/wiki/ルートヴィッヒ・ボルツマン（2022 年 6 月 6 日閲覧）
[64] Wikipedia, エントロピー, https://ja.wikipedia.org/wiki/エントロピー（2022 年 6 月 6 日閲覧）
[65] Wikipedia, カルノーの定理_（熱力学）, https://ja.wikipedia.org/wiki/カルノーの定理_(熱力学)（2022 年 6 月 7 日閲覧）
[66] Wikipedia, ウィラード・ギブズ, https://ja.wikipedia.org/wiki/ウィラード・ギブズ（2022 年 6 月 7 日閲覧）
[67] 植松友彦,『イラストで学ぶ情報理論の考え方』, 講談社, 2012.
[68] Wikipedia, アルフレッド・ヴェイル, https://ja.wikipedia.org/wiki/アルフレッド・ヴェイル（2022 年 6 月 7 日閲覧）
[69] Wikipedia, Robert Fano, https://en.wikipedia.org/wiki/Robert_Fano（2022 年 8 月 4 日閲覧）
[70] Wikipedia, Shannon-Fano coding, https://en.wikipedia.org/wiki/Shannon-Fano_coding（2022 年 8 月 4 日閲覧）
[71] Wikipedia, David A. Huffman, https://en.wikipedia.org/wiki/David_A._Huffman（2022 年 8 月 4 日閲覧）
[72] Wikipedia, Abraham Lempel, https://en.wikipedia.org/wiki/Abraham_Lempel（2022 年 8 月 4 日閲覧）
[73] Wikipedia, Jacob Ziv, https://en.wikipedia.org/wiki/Jacob_Ziv（2022 年 8 月 4 日閲覧）
[74] J. Ziv and A. Lempel, “A Universal Algorithm for Sequential Data Compression,” *IEEE Transactions on Information Theory*, **23**, pp 337–343, 1977.
[75] J. Ziv and A. Lempel, “Compression of Individual Sequences via Variable-Rate Coding,” *IEEE Transactions on Information Theory*, **24**, pp 530–536, 1978.
[76] 正畠宏一, 高木重定, 折口壮志, 鶴田祥一郎, 鈴木徹也, “モノからコトへ―新たなる循環経済の形成―,” 日本 LCA 学会誌, **14**, pp. 173–177, 2018.
[77] Wikipedia, Richard Hammin, https://en.wikipedia.org/wiki/Richard_Hamming（2022 年 12 月 20 日閲覧）
[78] 植松友彦,『代数系と符号理論』, コロナ社, 2010.

