



情報エレクトロニクス学科共通科目・2年次・夏ターム〔必修科目〕

講義「情報理論」

第13回

第8章 通信路符号化法

8.2 ハミング符号



[復習] 単一パリティ検査符号

単一パリティ検査符号

系列 $x_1x_2\cdots x_k$ を、含まれる1の個数が偶数になるように、もう一つ記号

$$c = x_1 + x_2 + \cdots + x_k$$

を付け加えた符号語 $w = x_1x_2\cdots x_k c$ に対応させる符号

情報記号 (information symbol) [2元の場合は情報ビット]

$$\text{符号語 } w = \overbrace{x_1x_2\cdots x_k} \underbrace{c}$$

(パリティ)検査記号 ([parity] check symbol) [2元の場合は(パリティ)検査ビット]

符号語のベクトル表現: $w = (x_1, x_2, \cdots, x_k, c)$



[復習]組織符号

組織符号 (systematic code)

k 個の情報記号から、 $n-k$ 個の検査記号を一定の方法で求め、付加することにより符号化される符号長 n の符号

$$w = \underbrace{x_1 x_2 \cdots x_k c_1 c_2 \cdots c_{n-k}}_n$$

(n, k) 符号

符号長 n 、情報記号数 k の組織符号



[復習]線形符号とパリティ検査方程式

線形符号 (linear code)

$c = x_1 + x_2 + \cdots + x_k$ のように、
検査記号が情報記号の線形な式で与えられる符号

パリティ検査方程式

$= 0$ という形で線形符号の符号語となるための必要十分条件を与える式(または式の組)

[単一パリティ検査符号のパリティ検査方程式]

長さ n の系列 $w = (w_1, w_2, \dots, w_n)$ が単一パリティ検査符号の符号語となるための必要十分条件は以下のパリティ検査方程式を満たすことである。

$$w_1 + w_2 + \cdots + w_{n-1} + w_n = 0$$

(符号語に含まれる1の個数が偶数)



[復習]誤りパターンとシンドローム

誤りパターン (error pattern)

送信した符号語 $w = (w_1, w_2, \dots, w_n)$ と

受信語 $y = (y_1, y_2, \dots, y_n)$ との差 $e = w + y = (w_1 + y_1, w_2 + y_2, \dots, w_n + y_n)$

- 誤りパターン $e = (e_1, e_2, \dots, e_n)$ の各ビットは以下のような値になる

$$e_i = \begin{cases} 1 & \text{(第 } i \text{ 成分に誤りが生じたとき)} \\ 0 & \text{(そうでないとき)} \end{cases}$$

- 誤りパターン e を用いると $y = w + e$ と表現できる

シンドローム (syndrome)

受信語 y をパリティ検査方程式の左辺に代入した結果 $s = y_1 + y_2 + \dots + y_n$

- 符号語 w はパリティ検査方程式を満たすので、

$$s = w_1 + e_1 + w_2 + e_2 + \dots + w_n + e_n = e_1 + e_2 + \dots + e_n$$

誤りがない $\Rightarrow s = 0$, 1個の誤り $\Rightarrow s = 1$



[復習] 2×2 水平垂直パリティ検査符号

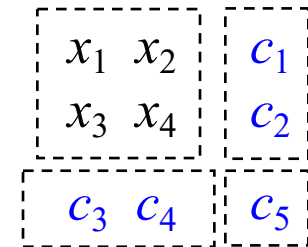
2×2 水平垂直パリティ検査符号

与えられた長さ4の系列 $x_1x_2x_3x_4$ に対し、長さ9の符号語 $(x_1, x_2, x_3, x_4, c_1, c_2, c_3, c_4, c_5)$ を、以下のように生成する符号。右図のように4個の情報ビットを 2×2 の配列に並べ、各行各列に一つずつ検査ビットを付け加える。すなわち、

$$\begin{aligned} c_1 &= x_1 + x_2 & c_2 &= x_3 + x_4 \\ c_3 &= x_1 + x_3 & c_4 &= x_2 + x_4 \end{aligned}$$

さらに、検査ビットの行の1の数が偶数になるように、検査ビットの検査ビットを右隅におく。

$$c_5 = c_1 + c_2 = x_1 + x_2 + x_3 + x_4 = c_3 + c_4$$



図：水平垂直パリティ検査符号

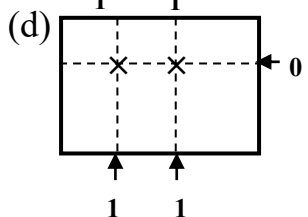
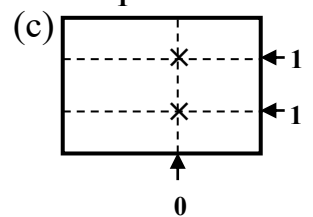
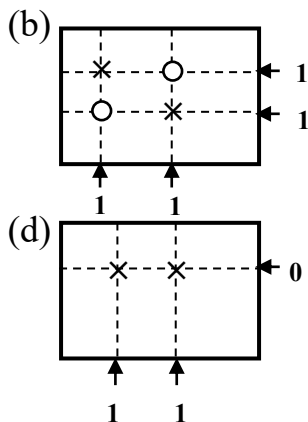
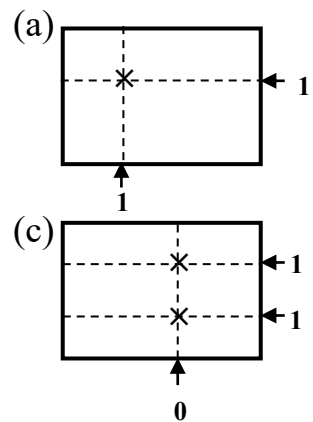
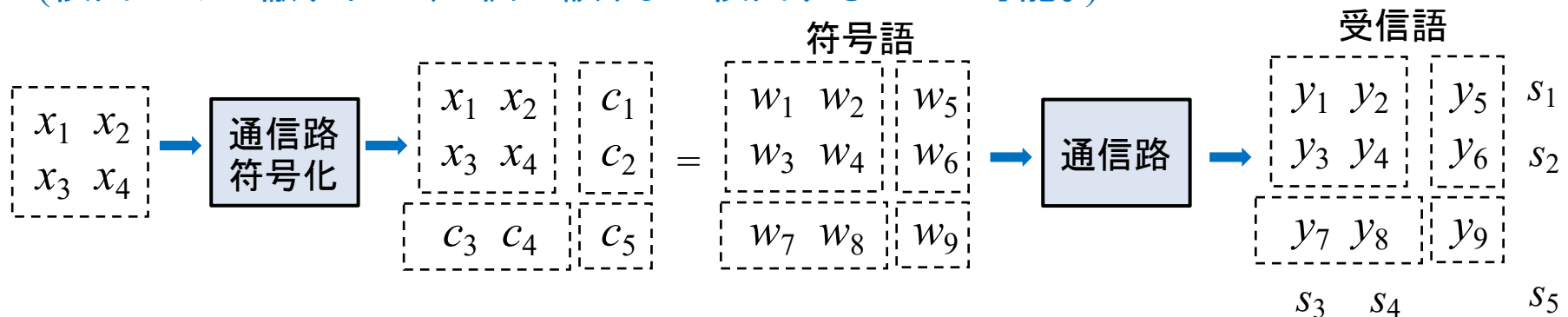


[復習] 2×2 水平垂直パリティ検査符号の誤り訂正能力

誤りの訂正まで行える 誤り(検出)訂正符号 (error-correcting code)

1個の誤りが訂正でき、同時に2個の誤りを検出することができる。

(検出のみに徹すれば、3個の誤りまで検出することが可能。)



(a) 情報ビットのみに1ビットの誤りが生じた場合

(s_1, s_2 の一方が1) and (s_3, s_4 の一方が1) and $s_5=1$

誤りが生じた行と列が特定可能 \Rightarrow 訂正可能

検査ビットのみの1ビット誤りも訂正可能

(b)(c)(d) 2ビットの誤りが生じた場合

s_1, s_2, s_3, s_4, s_5 のどれかは1

ビットの特定は不可能 \Rightarrow 検出可能、訂正不可能

図: 単一誤りの訂正と2重誤りの検出

(シンδροームが0の位置は検出できない)



(7,4)ハミング符号(1)

- 水平垂直パリティ検査符号の問題点
(9,4)符号であり、情報ビットよりも検査ビットが多く、効率がよくない。

- (7,4)ハミング符号

4個の情報ビット x_1, x_2, x_3, x_4 に対し、

$$c_1 = x_1 + x_2 + x_3$$

$$c_2 = x_2 + x_3 + x_4$$

$$c_3 = x_1 + x_2 + x_4$$

により、検査ビット c_1, c_2, c_3 を作り、

$$w = (x_1, x_2, x_3, x_4, c_1, c_2, c_3)$$

という符号語に符号化する符号

(検査ビットの作り方にはいくつかのバリエーションがあることに注意!)

情報ビットが4個であるから、符号語は $2^4 = 16$ 個

表:(7,4)ハミング符号

x_1	x_2	x_3	x_4	c_1	c_2	c_3
0	0	0	0	0	0	0
1	0	0	0	1	0	1
0	1	0	0	1	1	1
1	1	0	0	0	1	0
0	0	1	0	1	1	0
1	0	1	0	0	1	1
0	1	1	0	0	0	1
1	1	1	0	1	0	0
0	0	0	1	0	1	1
1	0	0	1	1	1	0
0	1	0	1	1	0	0
1	1	0	1	0	0	1
0	0	1	1	1	0	1
1	0	1	1	0	0	0
0	1	1	1	0	1	0
1	1	1	1	1	1	1



(7,4)ハミング符号(2)

- 符号語を $w=(w_1, w_2, \dots, w_7)$ する。

(7,4)ハミング符号のパリティ検査方程式

$$w_1 + w_2 + w_3 + w_5 = 0$$

$$w_2 + w_3 + w_4 + w_6 = 0$$

$$w_1 + w_2 + w_4 + w_7 = 0$$

受信語 $y=(y_1, y_2, \dots, y_7)$ に対するシンドローム

$$s_1 = y_1 + y_2 + y_3 + y_5$$

$$s_2 = y_2 + y_3 + y_4 + y_6$$

$$s_3 = y_1 + y_2 + y_4 + y_7$$

誤りパターンを $e=(e_1, e_2, \dots, e_7)$ とすると

$$s_1 = e_1 + e_2 + e_3 + e_5$$

$$s_2 = e_2 + e_3 + e_4 + e_6$$

$$s_3 = e_1 + e_2 + e_4 + e_7$$

$$y_i = w_i + e_i$$

表: 単一誤りに対するシンドローム

誤りパターン							シンドローム		
e_1	e_2	e_3	e_4	e_5	e_6	e_7	s_1	s_2	s_3
1	0	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	1	1	1
0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	1	1
0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0

シンドロームのパターンから
1個の誤りパターンが判る!

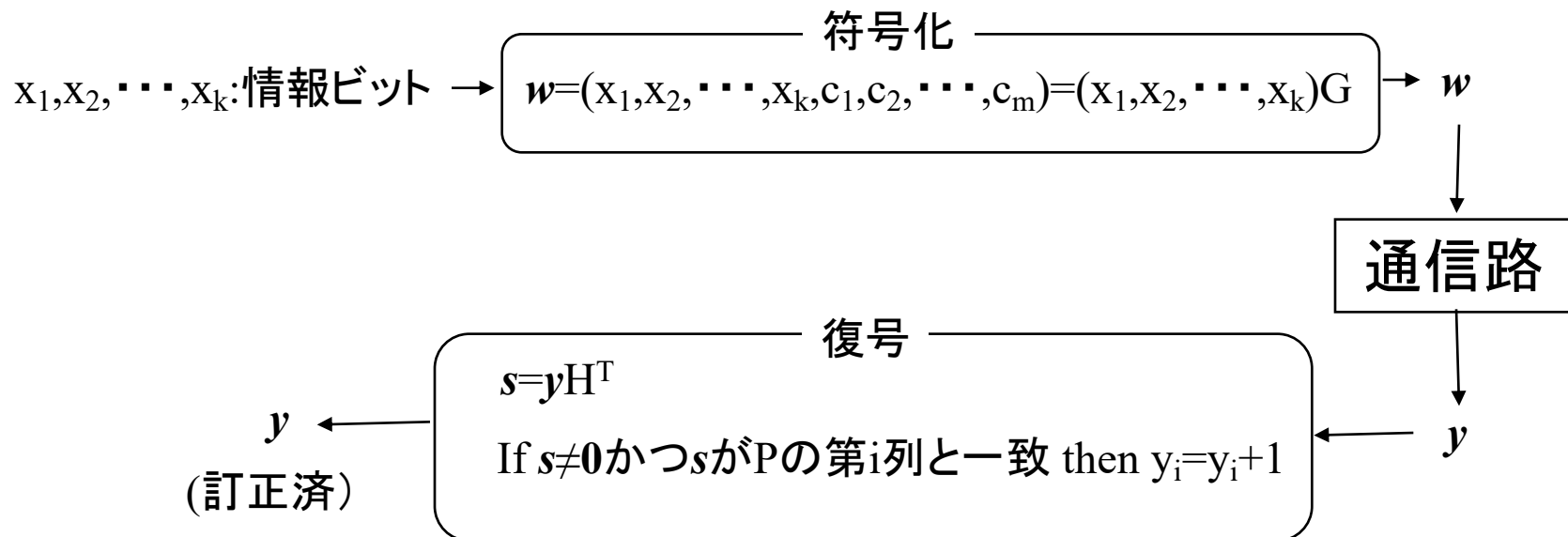


ハミング符号の符号化と復号

検査行列 $H = \begin{bmatrix} P & E_m \end{bmatrix}$
($m \times n$)

生成行列 $G = \begin{bmatrix} E_k & P^T \end{bmatrix}$
($k \times n$)

(P : $m \times k$ 行列, E_m : $m \times m$ 単位行列, $n = k + m$)



各ビットを並列に処理することが可能 \rightarrow 並列符号器、並列復号器



生成行列

- (7,4)ハミング符号の符号語 w は情報記号のみで表すと

$$w = (x_1, x_2, x_3, x_4, x_1 + x_2 + x_3, x_2 + x_3 + x_4, x_1 + x_2 + x_4)$$

という形で書ける。ここで、

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{matrix} \leftarrow x_1 \\ \leftarrow x_2 \\ \leftarrow x_3 \\ \leftarrow x_4 \end{matrix}$$

という行列を考えれば、符号語 w を

$$w = x G$$

と書くことができる。ただし、 $x = (x_1, x_2, x_3, x_4)$ である。

生成行列 (generator matrix)

k 個の情報記号からなるベクトル x をかけたとき、
対応する符号語が生成されるような行列

(n, k) 線形符号の生成行列は $k \times n$ 行列となる。



検査行列

- (パリティ)検査行列 (parity check matrix)

(7,4)ハミング符号のパリティ検査方程式の係数行列 H

$$H = \begin{array}{cccc|ccc} x_1 & x_2 & x_3 & x_4 & c_1 & c_2 & c_3 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

$$H^T = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

これを用いれば、パリティ検査方程式は

$$w H^T = 0$$

(H^T : H の転置行列、 0 : 全成分が0のベクトル)

と書ける。

(n, k)線形符号のパリティ検査行列は $(n-k) \times n$ 行列

- 検査行列 H を用いれば、(7,4)ハミング符号のシンδροームの計算式は

$$s = y H^T$$

と書ける。ここに s は $s = (s_1, s_2, s_3)$ であり、**シンδροームパターン**または単に

シンδροームと呼ばれる。 $s = (w + e)H^T = wH^T + eH^T = eH^T$



一般のハミング符号

表: 単一誤りに対するシンドローム

誤りパターン							シンドローム		
e_1	e_2	e_3	e_4	e_5	e_6	e_7	s_1	s_2	s_3
1	0	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	1	1	1
0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	1	1
0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0

検査行列

$$\left(\begin{array}{cc|cc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right)$$

単一誤りに対する
シンドロームの行

検査行列の列

すべて異なる計 $2^m - 1$ 行
= 0以外の m 次元2元ベクトル (m : 検査ビット数)

一般のハミング符号

$$\text{検査行列} \left\{ \underbrace{p_1 p_2 \cdots p_k}_{0\text{以外の } m\text{次元2元ベクトル}(n=2^m-1)} \mid e_1 e_2 \cdots e_m \right\}_m$$

符号長: $n = 2^m - 1$
 情報ビット数: $k = 2^m - 1 - m$
 検査ビット数: $n - k = m$



ちょっと休憩





ハミング距離とハミング重み(1)

2つの n 次元ベクトル $u=(u_1, u_2, \dots, u_n)$, $v=(v_1, v_2, \dots, v_n)$ の間のハミング距離 $d_H(u, v)$

$$\stackrel{\text{def}}{\Leftrightarrow} d_H(u, v) = \sum_{i=1}^n \delta(u_i, v_i) \quad \text{ただし} \quad \delta(u, v) = \begin{cases} 0 & \text{if } u=v \\ 1 & \text{if } u \neq v \end{cases}$$

$d_H(u, v)$ は u と v の対応する位置にある成分の対のうち、互いに異なるものの数

ハミング距離は距離の3公理を満たす。

距離の3公理

任意の n 次元ベクトル v_1, v_2, v_3 に対して以下のことが成り立つ。

- (i) $d_H(v_1, v_2) \geq 0$ であり、等号が成立するのは $v_1 = v_2$ のときに限る。
- (ii) $d_H(v_1, v_2) = d_H(v_2, v_1)$
- (iii) $d_H(v_1, v_2) + d_H(v_2, v_3) \geq d_H(v_1, v_3)$ (三角不等式)



ハミング距離とハミング重み(2)

n 次元ベクトル v のハミング重みまたは重み $w_H(v)$ $\stackrel{\text{def}}{\Leftrightarrow} w_H(v) = d_H(v, \mathbf{0})$

$w_H(v)$ は v の0でない成分の数

ハミング距離はハミング重みを用いて次のように表せる。

$$d_H(u, v) = w_H(u - v)$$

(例) 符号語 w を送り t 個の誤りが生じて $y = w + e$ が受信された場合

$$d_H(w, y) = w_H(e) = t$$



最小距離と誤り訂正能力(1)

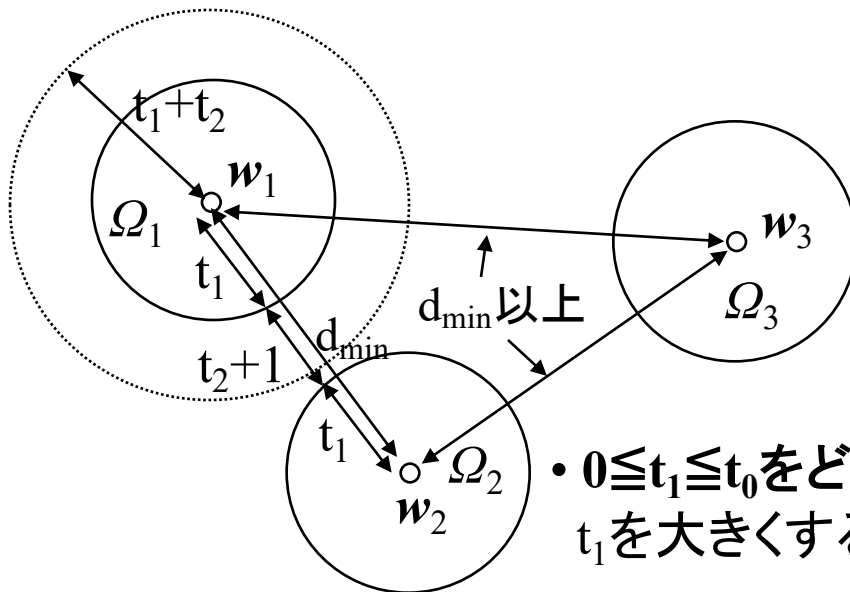
符号Cの**最小ハミング距離**または**最小距離**(minimum distance) d_{\min}

def

$$\Leftrightarrow d_{\min} = \min_{u \neq v, u, v \in C} d_H(u, v)$$

限界距離復号法

式 $d_{\min} \geq 2t_1 + 1$ を満たす整数 t_1 を定め、 t_1 以下の誤り訂正を行う復号法



- t_1 の最大値 $t_0 = \lfloor (d_{\min} - 1) / 2 \rfloor$ を **誤り訂正能力** という。

- $t_2 = d_{\min} - 2t_1 - 1$ とおけば $t_1 + 1 \leq t \leq t_1 + t_2$ 個の誤りは訂正はできないが検出は可能

- $0 \leq t_1 \leq t_0$ をどのように選ぶかは重要な問題
 t_1 を大きくする \rightarrow 正しく復号される確率は増大するが
 誤って復号される確率も増大
 検出さえできれば、再送要求などの救済措置が可能



最小距離と誤り訂正能力(2)

【例】 $d_{\min}=5$ の符号による誤りの訂正と検出

t_1	訂正可能な誤り	訂正できないが検出可能な誤り
0	—	1~4個
1	1個	2~3個
2	2個	—

線形符号の最小距離 = 0でない符号語のハミング重みの最小値
最小ハミング重みまたは重み

$$\text{何故ならば } d_{\min} = \min_{u \neq v, u, v \in C} d_H(u, v) = \min_{u \neq v, u, v \in C} w_H(u - v) = \min_{w \in C, w \neq 0} w_H(w)$$

[ハミング符号] 最小距離 $d_{\min}=3$ 、誤り訂正能力 $t_0=1$

(例) (7,4)ハミング符号の場合 最小距離 d_{\min} = 最小ハミング重み=3

[水平垂直パリティ検査符号] 最小距離 $d_{\min}=4$ 、誤り訂正能力 $t_0=1$

単一誤り訂正・2重誤り検出符号

(single-error-correcting/double-error-detecting code; SEC/DED符号)