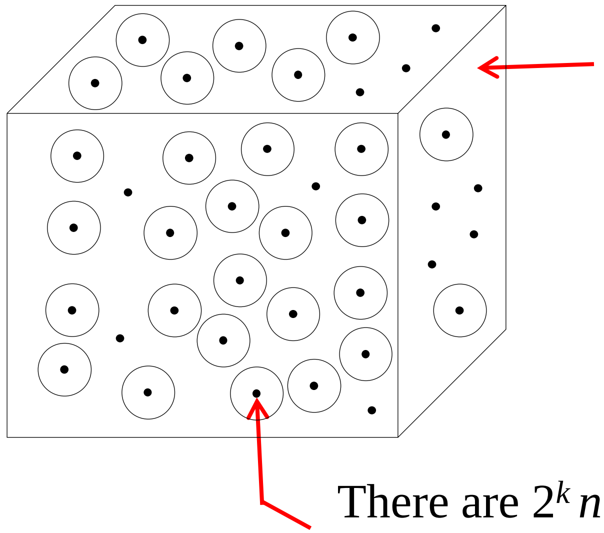
Chapter 4 Channel Coding

- 4.1 An Introduction of Channel Coding
- 4.2 Shannon's Channel Coding Theorem
- 4.3 Block Codes
- 4.4 Cyclic Codes
- 4.5 A Course Towards Decoding

§ 4.1 An Introduction of Channel Coding

- Channel Coding: map a k-dimensional message vector to an n-dimensional codeword vector, and $k < n$.
- If it is a binary channel code, there are at most 2^k n-dimensional codewords. The redundancy of $2^n - 2^k$ enables the error-correction capability of the code.



The n-dimensional binary space that can accommodate at most 2^n binary vectors.

There are 2^k n-dimensional codeword vectors filling the space.

- Codebook $\mathcal{C}$ collects all codewords. It has a cardinality of $|\mathcal{C}| = 2^k$.

§ 4.1 An Introduction of Channel Coding

- Code rate (r): A ratio of code dimension k to codeword length n, i.e., $r = \frac{k}{n}$. The redundancy is $n – k$. It underpins the efficiency in error-correction.
- Decoding:

Aim: with the received vector $\bar{y}$, we try to estimate $\bar{c}$. Let $\hat{\bar{c}}$ denote the estimation produced by the decoder. The decoding can be categorized into three cases:

Case I: $\hat{\bar{c}} = \bar{c}$, correct decoding;

Case II: $\hat{\bar{c}} \in \mathcal{C}$, but $\hat{\bar{c}} \neq \bar{c}$, decoding error;

Case III: Decoder does not produce any outcome, decoding failure.

§ 4.1 An Introduction of Channel Coding

- A channel code is a specific capacity approaching operational strategy.
- Based on the encoder structure, channel codes can be categorized into block codes and convolutional codes.

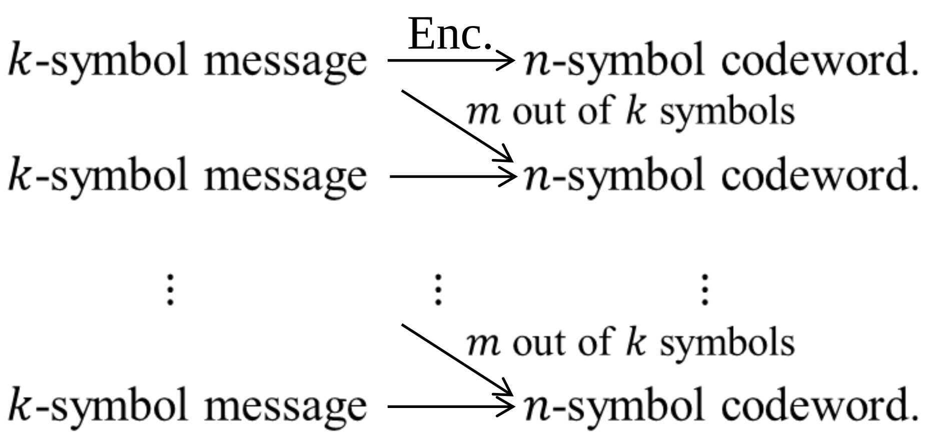
1. Block codes:

$$k\text{-symbol message} \xrightarrow{\text{Enc.}} n\text{-symbol codeword.}$$

- Encoder is memoryless and can be implemented with a combinatorial logic circuit.
- **Linear Block Code:** If $\bar{c}_i$ and $\bar{c}_j$ belong to a block code, $\bar{c}' = a \cdot \bar{c}_i + b \cdot \bar{c}_j$ also belongs to the block code. $(a, b) \in \mathbb{F}_q$ in which the block code is defined.
- Examples: **Reed-Solomon code**, algebraic-geometric code, **Hamming code**, low-density parity-check (LDPC) code.

§ 4.1 An Introduction of Channel Coding

2. Convolutional codes:



- Encoder has a memory of order *m*. It can be implemented with a sequential logic circuit.
- Examples: **Convolutional code**, **Trellis coded modulation**, **Turbo code**, Spatially-coupled LDPC code.

§ 4.2 Shannon’s Channel Coding Theorem

Shannon’s Channel Coding Theorem: All rates below capacity C are achievable. For every rate $r < C$, there exists channel codes of length n and dimension nr, such that the maximum error probability $P_e \to 0$. Inversely, any such codes that realize $P_e \to 0$ must have rate $r < C$.

- Shannon’s Channel Coding Theorem demonstrates error free transmission is possible by manipulating the code rate according to the channel capacity. It is defined in the mindset of binary transmission, e.g., BPSK.
- Its proof involves the justification of achievability, i.e., if $r < C$, $P_e \to 0$, and its converse, i.e., if $P_e \to 0$, $r < C$. They require the assistance of Jointly Typical Sequences and Fano’s Inequality, respectively.

§ 4.2 Shannon's Channel Coding Theorem

- **Empirical Entropy**: Given an X sequence $X^n(x^n: x_1, x_2, \ldots, x_n)$, its empirical entropy is

$$H^*(X) = -\frac{1}{n}\log_2 P(x^n)$$

- Similarly, given two sequences $X^n(x^n: x_1, x_2, \ldots, x_n)$ and $Y^n(y^n: y_1, y_2, \ldots, y_n)$, their joint empirical entropy is

$$H^*(X,Y) = -\frac{1}{n}\log_2 P(x^n, y^n)$$

- If sequences X^n and Y^n have the i.i.d. property, i.e.

$$P(x^n) = \prod_{i=1}^{n} P(x_i) \qquad P(x^n, y^n) = \prod_{i=1}^{n} P(x_i, y_i)$$

the above empirical entropies become

$$H^*(X) = -\frac{1}{n}\sum_{i=1}^{n}\log_2 P(x_i) \qquad H^*(X,Y) = -\frac{1}{n}\sum_{i=1}^{n}\log_2 P(x_i, y_i)$$

§ 4.2 Shannon's Channel Coding Theorem

- **<u>Jointly Typical Sequences</u>:** Given $\epsilon \to 0$, x^n and y^n are jointly typical sequences if

$$|H^*(X) - H(X)| < \epsilon$$
$$|H^*(Y) - H(Y)| < \epsilon$$
$$|H^*(X,Y) - H(X,Y)| < \epsilon.$$

- ① If x^n and y^n are drawn i.i.d. as

$$P(x^n, y^n) = \prod_{i=1}^{n} P(x_i, y_i),$$

when $n \to \infty$,

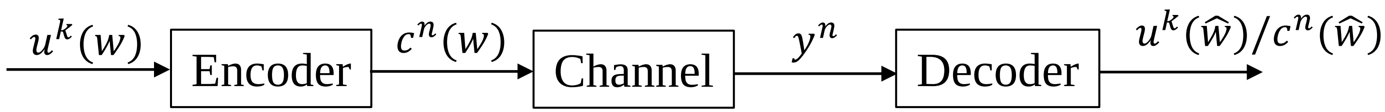
$$\Pr(x^n \text{ and } y^n \text{ are jointly typical}) \to 1.$$

② If z^n and y^n are independent, as $P(z^n, y^n) = P(z^n)\ P(y^n)$,

$$\Pr(z^n \text{ and } y^n \text{ are jointly typical}) \le 2^{-n(I(Z,Y)-3\epsilon)}.$$

§ 4.2 Shannon's Channel Coding Theorem

- **<u>Modelling and Assumptions of the Proof</u>**



- Codeword length n, dimension $k = nr$, message/codeword index w
- Decoding error probability $P(\epsilon) = \Pr(\hat{w} \neq w)$

- Assumptions (A):

A-I: A random binary code is generated as

$$P(\mathcal{C}) = \prod_{w=1}^{2^{nr}} P(c^n(w))$$

$$= \prod_{w=1}^{2^{nr}} \prod_{i=1}^{n} P(c_i(w)).$$

§ 4.2 Shannon's Channel Coding Theorem

A-II: Both the transmitter and receiver know the channel, i.e., $P(y_i|c_i(w))$, $\forall i$.

A-III: Messages (codewords of $\mathcal{C}$) are uniformly chosen for transmission as

$$P\left(u^k(w)\right) = P\left(c^n(w)\right) = \frac{1}{2^{nr}} \ .$$

A-IV: The channel is discrete memoryless, i.e.,

$$P(y^n|c^n(w)) = \prod_{i=1}^{n} P(y_i|c_i(w)) \ .$$

Therefore,

$$\begin{aligned} P(c^n(w), y^n) &= P(y^n|c^n(w))\, P\left(c^n(w)\right) \\ &= \prod_{i=1}^{n} P(y_i|c_i(w)) \cdot \prod_{i=1}^{n} P(c_i(w)) \\ &= \prod_{i=1}^{n} P(y_i, c_i(w)). \end{aligned}$$

§ 4.2 Shannon's Channel Coding Theorem

Achievability Proof

- Generate a random binary code of length *n* rate *r* as A-I. The codebook $\mathcal{C}$ is

$$\mathcal{C} = \begin{bmatrix} c_1(1) & c_2(1) & \cdots & c_n(1) \\ \vdots & \vdots & \cdots & \vdots \\ c_1(w) & c_2(w) & \cdots & c_n(w) \\ \vdots & \vdots & \cdots & \vdots \\ c_1(2^{nr}) & c_2(2^{nr}) & \cdots & c_n(2^{nr}) \end{bmatrix}$$

They are codewords

$$P(\mathcal{C}) = \prod_{w=1}^{2^{nr}} \prod_{i=1}^{n} P(c_i(w))$$

- Based on A-III,

$$P\big(c^n(w)\big) = \prod_{i=1}^{n} P(c_i(w)) = \frac{1}{2^{nr}}.$$

- With received vector y^n , the decoder estimates codeword $c^n(\widehat{w})$ such that
  - $c^n(\widehat{w})$ and y^n are jointly typical sequences.
  - There is no other codeword $c^n(v)$ such that $c^n(v)$ and y^n are jointly typical sequences.

§ 4.2 Shannon's Channel Coding Theorem

- The decoding error probability is

$$P(\epsilon) = \sum_{\mathcal{C}} \underline{P(\mathcal{C})}\ \underline{P_e(\mathcal{C})}$$

Prob. of a particular code $\mathcal{C}$

Error prob. of the code $\mathcal{C}$

$$P_e(\mathcal{C}) = \frac{1}{2^{nr}} \sum_{w=1}^{2^{nr}} \underline{P_{e,w}(\mathcal{C})}$$

Error prob. of a particular codeword $c^n(w) \in \mathcal{C}$

$$P(\epsilon) = \frac{1}{2^{nr}} \sum_{\mathcal{C}} \sum_{w=1}^{2^{nr}} P(\mathcal{C}) P_{e,w}(\mathcal{C})$$

- Due to symmetry of code construction, we know

$$\frac{1}{2^{nr}} \sum_{w=1}^{2^{nr}} P_{e,w}(\mathcal{C}) = P_{e,1}(\mathcal{C})$$

- Hence,

$$\begin{aligned} P(\epsilon) &= \sum_{\mathcal{C}} P(\mathcal{C})\ P_{e,1}(\mathcal{C}) \\ &= \underline{P_{e,1}} \end{aligned}$$

Average (over all codebooks) error prob. of codeword $c^n(1)$

§ 4.2 Shannon's Channel Coding Theorem

- Let E_w denote the event that codeword $c^n(w)$ (X^n) and y^n (Y^n) are jointly typical sequences.

$$P(\epsilon) = P_{e,1}$$
$$= \Pr(E_1^C \cup E_2 \cup E_3 \cup \cdots \cup E_{2^{nr}})$$
$$\leq \Pr\left(E_1^C\right) + \sum_{w=2}^{2^{nr}} \Pr(E_w)$$

Based on ①, where $n \to \infty$, $\Pr\left(E_1^C\right) \leq \epsilon$.

Based on ②, $\Pr(E_w) \leq 2^{-n(I(X,Y)-3\epsilon)}$.

- Therefore,

$$P(\epsilon) \leq \epsilon + \sum_{w=2}^{2^{nr}} 2^{-n(I(X,Y)-3\epsilon)}$$
$$= \epsilon + (2^{nr} - 1) \cdot 2^{-n(I(X,Y)-3\epsilon)}$$
$$< \epsilon + 2^{3n\epsilon} 2^{-n(I(X,Y)-r)}$$
$$= \epsilon + 2^{-n(I(X,Y)-3\epsilon-r)}$$

§ 4.2 Shannon's Channel Coding Theorem

- If n is sufficiently large and $r < I(X,Y) - 3\epsilon$,

$$P(\epsilon) \leq 2\epsilon,$$

  the decoding error probability can be arbitrarily small.

- Choose $P(c_i(w))$ to be the distribution that maximizes $I(X,Y)$ as

$$C = \max_{P(c_i(w))}\{I(X,Y)\},$$

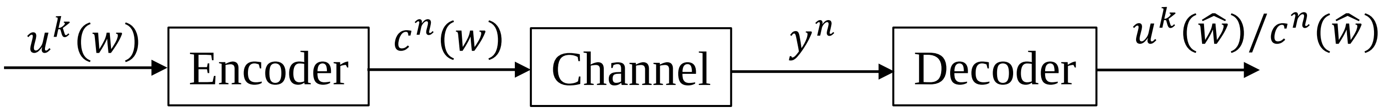
  the above conclusion implies if $r < C$, the decoding error probability $P(\epsilon)$ can be arbitrarily small. <u>Achievability Proof Ends</u>

Remark: The achievability proof is founded on <u>random code construction</u>, <u>large codeword length</u> and <u>ideal codeword symbol distributions</u>. They become the features of capacity approaching (achieving) codes, i.e. Turbo codes, LDPC codes and Polar codes.

§ 4.2 Shannon's Channel Coding Theorem

- **Converse of Shannon's Channel Coding Theorem**

  If $P(\epsilon) \to 0$, $r \leq C$.



- **Fano's inequality**

  Over a DMC, given a code of rate r with the input message uniformly distributed, let $P(\epsilon) = \Pr(\hat{w} \neq w)$,

$$H(c^n|y^n) \leq 1 + P(\epsilon) \cdot nr.$$

Proof: Extending the Fano's inequality into vector domain,

$$H(c^n|y^n) \leq H\big(P(\epsilon)\big) + P(\epsilon)\log(2^{nr} - 1)$$
$$\leq 1 + P(\epsilon) \cdot nr.$$

Note: The 2nd inequality is realized with $n \to \infty$.

§ 4.2 Shannon's Channel Coding Theorem

Converse Proof

- Based on A-III, input messages are uniformly distributed.

$$H\left(u^k(w)\right) = \log 2^{nr} = nr.$$

- Since

$$H\left(u^k(w)\right) = H\left(u^k(w)|y^n\right) + I(u^k(w), y^n)$$

where

$$\mathrm{H}\left(u^k(w)\middle|y^n\right) = H(c^n(w)|y^n)$$

and based on Data Processing Inequality,

$$I\left(u^k(w), y^n\right) \leq I(c^n(w), y^n).$$

we have

$$nr = H\left(u^k(w)\right) \leq H(c^n(w)|y^n) + I(c^n(w), y^n).$$

§ 4.2 Shannon's Channel Coding Theorem

- Applying Fano's Inequality

$$H(c^n(w)|y^n) \leq 1 + P(\epsilon) \cdot nr.$$

- Over DMC and input being independent

$$I(c^n(w), y^n) = \sum_{i=1}^{n} I(c_i(w), y_i)$$
$$= n \cdot C.$$

Therefore,

$$nr \leq 1 + P(\epsilon)nr + nC$$
$$r \leq P(\epsilon)r + \frac{1}{n} + C$$

With $n \to \infty$ and $P(\epsilon) \to 0$, $r \leq C$.

Converse Proof Ends

§ 4.3 Block Codes

- All block codes are defined by their codeword length *n*, dimension *k* and the minimum Hamming distance *d*. A block code is often denoted as an (*n*, *k*, *d*) code.
- Code rate: $r = \frac{k}{n}$.
- Encoding of a linear block code can be written as:

$$\boxed{\bar{c} = \bar{u} \cdot \mathbf{G}}$$

$\bar{u}$ — *k*-dimensional message vector.

G — a generator matrix of size $k \times n$. It defines the legal space among all *n*-dimensional vectors.

$\bar{c}$ — *n*-dimensional codeword vector.

Linear block code:

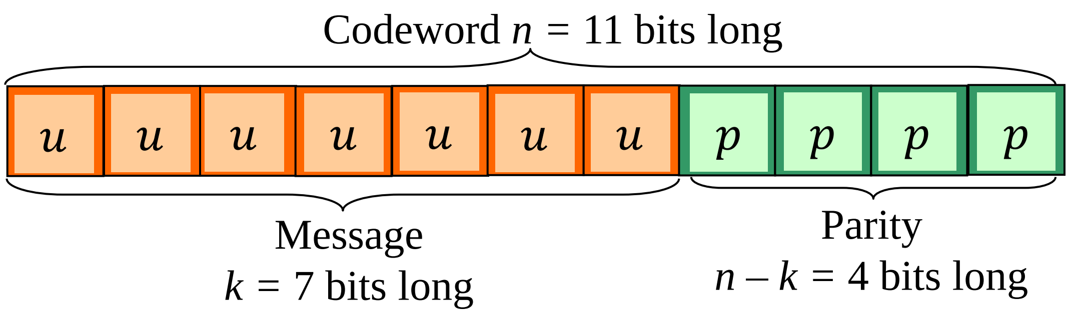
$$\bar{c}_1 = \bar{u}_1 \cdot \mathbf{G}$$

$$\bar{c}_2 = \bar{u}_2 \cdot \mathbf{G}$$

$$(\bar{u}_1 + \bar{u}_2) \cdot \mathbf{G} = (\bar{c}_1 + \bar{c}_2) \in \mathcal{C}$$

§ 4.3 Block Codes

Hamming Distance



u = message bits
p = parity-check bits

The Hamming Distance between any two codewords is the total number of positions where the two codewords differ.

1	0	0	1	1	1	0	1
0	0	1	1	0	1	0	0

The total number of positions where these two codewords differ is 4. Therefore, the Hamming distance is 4.

Weight: Given a vector, its weight is the number of nonzero positions.

1	0	0	1	1	1	0	1

The weight of the vector is 5.

§ 4.3 Block Codes

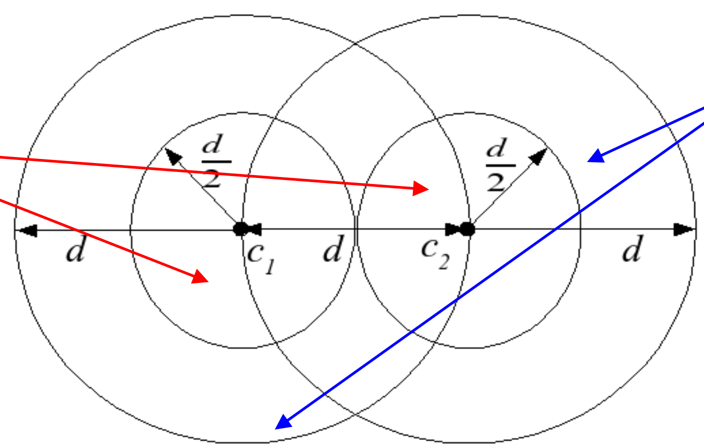
The Minimum Hamming Distance and Error-Correction Capability

The minimum Hamming distance: for any two codewords $\bar{c}_i$ and $\bar{c}_j$ picked up from the codebook $\mathcal{C}$, the minimum Hamming distance d is defined as:

$$d = \min_{(\bar{c}_i, \bar{c}_j) \in \mathcal{C}} \{d_{\mathrm{Ham}}(\bar{c}_i, \bar{c}_j)\}.$$

- In general, a block code can correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors, where $\lfloor x \rfloor$ means that x is rounded down to the nearest integer, e.g., $\lfloor 2.5 \rfloor = 2$.
- A block code can **detect** d – 1 errors.

A block code can ***correct*** received words with up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.



A block code can ***detect*** up to d – 1 errors

- For a linear block code, $d = \min\{\text{weight}\,(\bar{c}_j), \bar{c}_j \neq 0\}$.

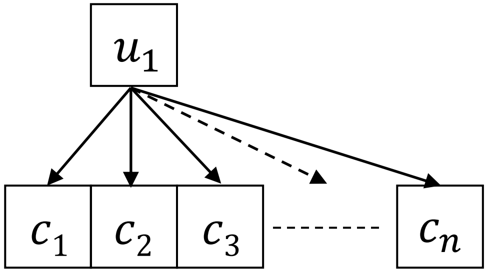
§ 4.3 Block Codes

Repetition Codes

A repetition encoder takes a **single** message bit and gives a codeword that is the message bit repeated *n* times, where *n* is an **odd** number

A message bit **0** will be encoded to give the codeword **0000...000**
A message bit **1** will be encoded to give the codeword **1111...111**

- This is the simplest type of error-correcting code as it only has **two codewords**
- We can easily see that it has a minimum Hamming distance $d = n$
- It is an $(n, 1, n)$ block code



The generator matrix of the code is simply

$$\mathbf{G} = [1\ 1\ 1\ 1\ \ldots\ 1]$$

§ 4.3 Block Codes

Repetition Codes

To recover the transmitted codeword of a repetition code, a simple decoder known as a **Majority Decoder** can be used

1. The number of 0s and 1s in the received word are counted.
2. If the number of 0s > number of 1s (i.e., a majority) , then the message bit was a 0. Else if the number of 1s > number of 0s, then the message bit was a 1.

***Example 4.1*:** Say our message bit was a 1 and it was encoded by the (5, 1, 5) repetition code. The codeword will be $\bar{c}$ = (11111).

- If after transmission we receive the word $\bar{r}$ = (10011), then the number of 1s > number of 0s and so the majority decoder decides that the original message was 1.
- However, if we receive the word $\bar{r}$ = (00011) then the number of 0s > number of 1s and the majority decoder **incorrectly** decides that the original message was 0.

In general, a (n, 1, n) repetition code can correct up to $\frac{n-1}{2}$ errors.

§ 4.3 Block Codes

Repetition Codes

The Great Wall

§ 4.3 Block Codes

Hamming Codes

- Single-error-correcting codes.
- Given any positive integer $m \geq 3$, its

  $n = 2^m - 1$

  $k = 2^m - m - 1$

  $d = 3$

- ***Example 4.2* :** Given $m = 3$, the generator matrix of the (7, 4, 3) Hamming code is

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

The codewords can be generated by $\bar{c} = \bar{u} \cdot \mathbf{G}$.

This code can correct 1 error.

§ 4.3 Block Codes

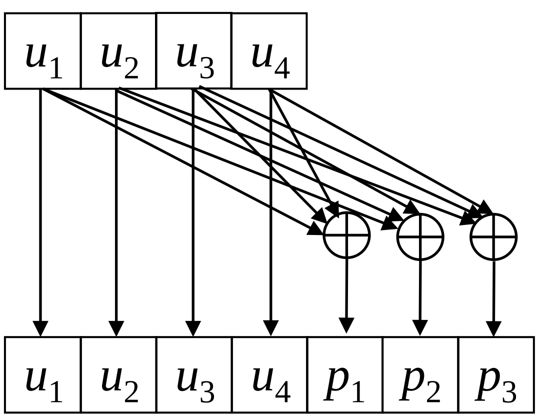
Notice that only 16 of 128 possible sequences of length 7 bits are used for transmission.

The parity bits are calculated by

$$p_1 = u_1 \oplus u_3 \oplus u_4$$

$$p_2 = u_1 \oplus u_2 \oplus u_3$$

$$p_3 = u_2 \oplus u_3 \oplus u_4$$



The encoding can be written as

$$\bar{c} = \bar{u} \cdot \mathbf{G},$$

and

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

$\bar{u}$	$\bar{c}$
0000	0000 000
0001	0001 101
0010	0010 111
0011	0011 010
0100	0100 011
0101	0101 110
0110	0110 100
0111	0111 001
1000	1000 110
1001	1001 011
1010	1010 001
1011	1011 100
1100	1100 101
1101	1101 000
1110	1110 010
1111	1111 111

Remark: This is a **systematic encoding** as the message symbols appear in the codeword.

§ 4.4 Cyclic Codes

- A cyclic code is a block code which has the property that cyclically shifting a codeword results in another codeword

- Cyclic codes have the advantage that simple encoders can be constructed using shift registers and low complexity decoding algorithms exist to decode them

- An (n, k) cyclic code is constructed by first choosing a generator polynomial $g(x)$ and multiplying this by a message polynomial $m(x)$ to generate a codeword polynomial $c(x)$ as

$$c(x) = u(x) \cdot g(x)$$

$$u(x) = u_0 + u_1 x + \cdots + u_{k-1} x^{k-1}$$

$$g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k}$$

$$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

§ 4.4 Cyclic Codes

Cyclic Hamming Code

- The (7, 4, 3) Hamming code is also a cyclic code that can be constructed using the generator polynomial $g(x) = x^3 + x^2 + 1$.
- ***Example 4.3***: To encode the binary message 1010, we first write it as the message polynomial $u(x) = x^3 + x$ and then multiply it with $g(x)$ modulo-2

$$\begin{aligned} c(x) &= u(x)g(x) \\ &= (x^3 + x)(x^3 + x^2 + 1) \\ &= x^6 + x^5 + x^4 + x^3 + x^3 + x \qquad [(x^3 + x^3) \bmod 2 = 2x^3 \bmod 2 = 0] \\ &= x^6 + x^5 + x^4 + x \end{aligned}$$

This codeword polynomial corresponds to 1 1 1 0 0 1 0. However, notice that the first 4 bits of this codeword are not the same as the original message 1010.

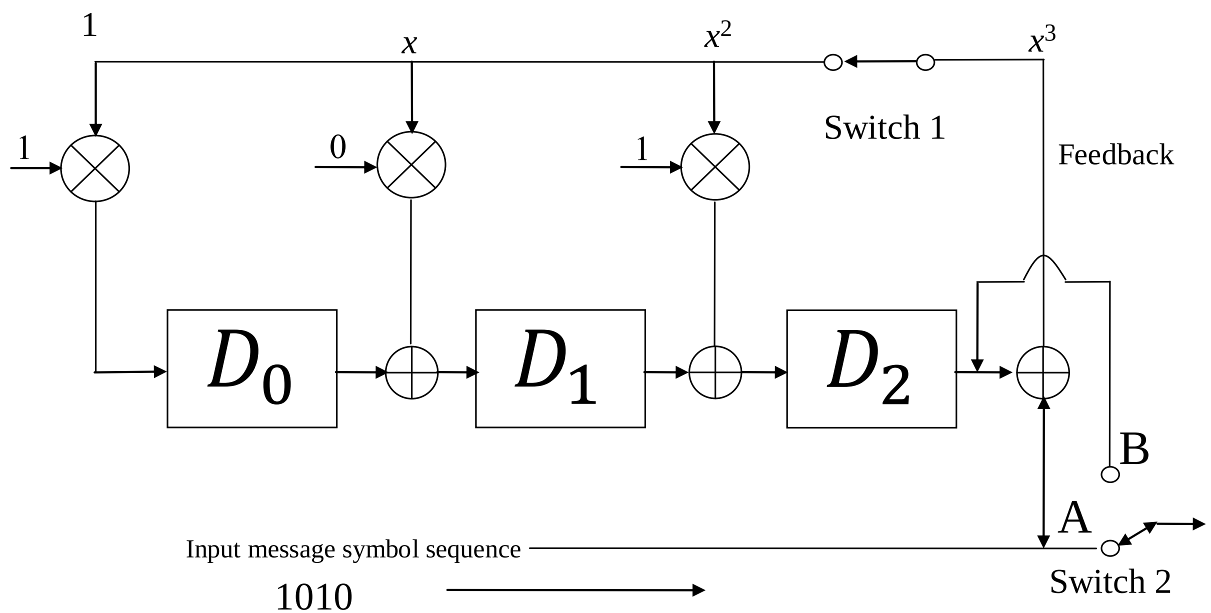
- This is an example of a **non-systematic code**.

Remark: Systematic encoding and non-systematic encoding only change the mapping between message and codeword, not the codebook.

§ 4.4 Cyclic Codes

Systematic Cyclic Hamming Code

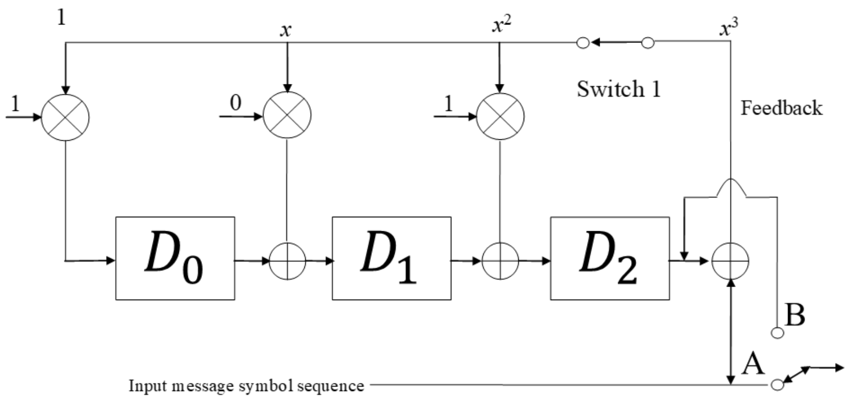
- Encoding of a systematic cyclic Hamming code can be performed by shift-registers.



An encoder for the systematic (7, 4, 3) cyclic Hamming code

1. For the first $k = 4$ message bits, switch 1 is closed and switch 2 is in position A
2. After the first 4 message bits have entered, switch 1 is open, switch 2 is in position B and the contents of memory elements are read out giving the parity-check bits

§ 4.4 Cyclic Codes



Example 4.4: Let the message be $\bar{u} = (u_1, u_2, u_3, u_4)$, the shift register computes

Input	Registers (left to right)		
u_1	u_1	0	u_1
u_2	$u_1 \oplus u_2$	u_1	$u_1 \oplus u_2$
u_3	$u_1 \oplus u_2 \oplus u_3$	$u_1 \oplus u_2$	$u_2 \oplus u_3$
u_4	$u_2 \oplus u_3 \oplus u_4$	$u_1 \oplus u_2 \oplus u_3$	$u_1 \oplus u_3 \oplus u_4$

Update of the shift registers:

$$Feedback = D_2 \oplus Input$$
$$D_2' = D_1 \oplus 1 \cdot Feedback$$
$$D_1' = D_0 \oplus 0 \cdot Feedback$$
$$D_0' = 1 \cdot Feedback$$

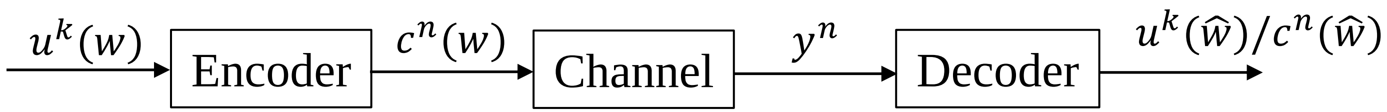
Hence, $p_1 = u_1 \oplus u_3 \oplus u_4$

$p_2 = u_1 \oplus u_2 \oplus u_3$

$p_3 = u_2 \oplus u_3 \oplus u_4$

This is equivalent to the systematic encoding of ***Example 4.2***.

§ 4.5 A Course Towards Decoding



- Given a received word y^n, decoding aims to recover codeword $c^n(w)$ (or message $u^k(w)$), yielding its estimation $(c^n(\hat{w}))$(or $u^k(\hat{w})$).
- Error-Correction starts from error-detection.
- The **Parity-Check Code**: for each binary message, a parity-check bit is added so that there are an even number of 1s in each codeword.

  If k = 3 then there are 8 possible messages. The eight codewords will be:

000 → 000**0**	100 → 100**1**
001 → 001**1**	101 → 101**0**
010 → 010**1**	110 → 110**0**
011 → 011**0**	111 → 111**1**

When there are odd number of 1, the decoder (detector) knows error has been introduced.

§ 4.5 A Course Towards Decoding

Parity-Check Matrix

- A primitive thought: given a received word $\bar{r}$, we can search the whole codebook and find the codeword (message) that has the smallest Hamming distance to $\bar{r}$. But even for a binary code, this has a complexity of $O(2^k)$. This process is called the maximum likelihood (ML) decoding.
- Alternatively, we can utilize the algebraic structure of the code, which is often told by the parity-check matrix **H**.
- A parity-check matrix **H** is defined as the **null space** of the generator matrix **G**, i.e., the inner product of the two matrices results in an all-zero matrix, $\mathbf{G}\mathbf{H}^T = \mathbf{0}$ (T is the transpose of the matrix)
- When a codeword is multiplied by the parity-check matrix, it should result in an all-zero vector, i.e.,

$$\bar{c} \cdot \mathbf{H}^T = \bar{u} \cdot \mathbf{G} \cdot \mathbf{H}^T = \underline{0}.$$

Syndrome vector.

- If $\hat{\bar{c}} \cdot \mathbf{H}^T = 0$, it implies $\hat{\bar{c}}$ is a valid codeword.

§ 4.5 A Course Towards Decoding

- If the generator matrix is of the form $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$, where $\mathbf{I}_k$ is a $k \times k$ identity matrix and $\mathbf{P}$ is a parity matrix, the parity-check matrix is in the form of $\mathbf{H} = [\mathbf{P}^T \mid \mathbf{I}_{n-k}]$.

Example 4.5: Taking the (7, 4, 3) Hamming code in ***Example 4.2***

$\mathbf{I}_4$, $\mathbf{P}$

$$\mathbf{G} = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

The parity-check matrix is

$\mathbf{P}^T$, $\mathbf{I}_{n\text{-}k} = \mathbf{I}_{7\text{-}4} = \mathbf{I}_3$

$$\mathbf{H} = \left[\begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

- Dual code property

	Generator matrix	Parity-check matrix
Code	$\mathbf{G}_{k\times n}$	$\mathbf{H}_{(n-k)\times n}$
Dual code	$\mathbf{G}_{(n-k)\times n}$	$\mathbf{H}_{k\times n}$

§ 4.5 A Course Towards Decoding

- Note that

$$\mathbf{G} \cdot \mathbf{H}^T = \left[\mathbf{I}_k \vdots \mathbf{P}_{k\times(n-k)}\right] \cdot \begin{bmatrix} \mathbf{P}_{k\times(n-k)} \\ \cdots\cdots\cdots \\ \mathbf{I}_k \end{bmatrix}$$
$$= \left[\mathbf{P}_{k\times(n-k)} + \mathbf{P}_{k\times(n-k)}\right]$$
$$= [0]_{k\times(n-k)}.$$

For a pair of dual codes, their codewords are generated by $\bar{c} = \bar{u} \cdot \mathbf{G}$, $\bar{c}^{\perp} = \bar{u}' \cdot \mathbf{H}$, where $\bar{u} \in \mathbb{F}_q^k$, $\bar{u}' \in \mathbb{F}_q^{n-k}$.

Then

$$\bar{c} \cdot (\bar{c}^{\perp})^T = (\bar{u} \cdot \mathbf{G}) \cdot (\mathbf{H}^T \cdot (\bar{u}')^T)$$
$$= \bar{u} \cdot \mathbf{G} \cdot \mathbf{H}^T \cdot (\bar{u}')^T$$
$$= 0.$$

G and **H** define two orthogonal vector spaces (of the same length).

- **H** can be constituted by $n-k$ linearly independent codewords of an $(n, n-k)$ code.
- **G** can be constituted by k linearly independent codewords of an (n, k) code.

§ 4.5 A Course Towards Decoding

Example 4.5: Decoding of (7, 4, 3) Hamming code.

$$\mathbf{H} = \left[\begin{array}{cccc:ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Assume the transmittal codeword is

$$\bar{c} = (0\ 1\ 0\ 1\ 1\ 1\ 0).$$

The received word is

$$\bar{r} = \bar{c} + \bar{e} = (0\ 1\ 0\ 1\ 0\ 1\ 0).$$

($\bar{e} = (0\ 0\ 0\ 0\ 1\ 0\ 0)$ is the error pattern.)

The syndrome is

$$\bar{r} \cdot \mathbf{H}^T = (\bar{c} + \bar{e}) \cdot \mathbf{H}^T\ .$$

§ 4.5 A Course Towards Decoding

The syndrome is

$$\bar{r} \cdot \mathbf{H}^T = (\bar{c} + \bar{e}) \cdot \mathbf{H}^T$$

$$= \bar{c} \cdot \mathbf{H}^T + \bar{e} \cdot \mathbf{H}^T$$

$$= \bar{0} + (0\ 0\ 0\ 0\ 1\ 0\ 0) \cdot \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= (1\ 0\ 0)$$

=> Column-4 of **H**. (Row-4 of $\mathbf{H}^T$)

=> $c_4 = r_4 + 1 = 1$.

=> $\hat{c} = (\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ )$.

§ 4.5 A Course Towards Decoding

Singleton Bound: Given an (n, k) linear block code with minimum Hamming distance d, we have

$$d \leq n - k + 1.$$

Proof:

- For the code, its parity-check matrix $\mathbf{H}_{(n-k)\times n}$ can be written as

  $$\mathbf{H} = [\,\bar{h}_1, \bar{h}_2, \dots, \bar{h}_n].$$

  Given a minimum weight codeword $\bar{c}$, it has a support of $\{i_1, i_2, \dots, i_d\}$. Moreover,

  $$c_{i_1} \cdot \bar{h}_{i_1}^T + c_{i_2} \cdot \bar{h}_{i_2}^T + \cdots + c_{i_d} \cdot \bar{h}_{i_d}^T = \bar{0}$$

  Hence, there are **at least** d column of **H** are linearly dependent.
- For **H**, its row rank equals to its column rank.
  Hence, there are **at most** $n - k$ linearly independent columns in **H**. That says any $n - k + 1$ columns of **H** are linearly dependent.
- Therefore,

  $$d \leq n - k + 1.$$

- Otherwise if $d > n - k + 1$, the minimum Hamming distance of the code will not be d.

Remark: If a code with $d = n - k + 1$, it is a maximum distance separable (MDS) code.

References:

[1] Elements of Information Theory, by T. Cover and J. Thomas.