24. **ERROR CONTROL CODING**

- Error detection (ARQ - Automatic Repeat Request)
- Error correction (FEC - Forward Error Correction)
- Block, convolutional codes

**BLOCK CODES**

- Message \( \{m_k\} \) \[ n \times k \text{ bits} \]
- Coder \[ \text{[}n, k\text{]} \] \( \text{rate} = \frac{k}{n} \)
- Coded output \( \{c_i\} \)
- Mod.
  - Channel
  - Demod.
  - Discrete channel.

Parity bits are related to information bits; decoder uses this "relation" to determine the message.

\[
p = \text{error probability} = \text{Prob} (\hat{c}_i \neq c_i)
\]

Binary Symmetric channel
(Hard decision)

Assume \( p = 8 \times 10^{-4} \)

Want \( \text{Prob} [\hat{m}_k \neq m_k] < 10^{-4} \)
Repetition code

\[ \begin{array}{c}
0 \rightarrow 0000 \\
1 \rightarrow 1111
\end{array} \]

1 inf. bit 2 parity bits

Decoding rule: max. likelihood, min. distance, majority logic

Hamming weight of a code word

\[ S_1 = \text{code word}_1 = 101101 \]
\[ S_2 = \text{code word}_2 = 001100 \]

\[ W[S_i] = \text{no. of 1's in code word } S_i \]

E.g. \( W[S_1] = 4 \) \( ; \) \( W[S_2] = 2 \)

If \( S_i = [s_{i1}, s_{i2}, \ldots, s_{in}] \)

\[ W[S_i] = \sum_{k=1}^{n} s_{ik} \]

Hamming weight is a proper norm

Hamming distance between code words

\[ d(S_i, S_j) = \text{no. of positions in which } S_i \text{ and } S_j \text{ differ} \]

E.g. \( d(S_1, S_2) = 2 \)

\[ d(S_i, S_j) = W(S_i \oplus S_j) \quad (g \mod 2) \]
\[ = \sum_{k=1}^{n} (s_{ik} \oplus s_{jk}) \]

E.g. \( d(S_1, S_2) = W(S_1 \oplus S_2) \)
\[ = W(100001) \]
\[ = 2 \]
- Minimum distance decoding is the decoded codeword is closest in Hamming distance to received word.

- Example: Repetition code

  \[
  \begin{array}{ccc}
  0 & 0 & 0 \\
  0 & 0 & 1 \\
  0 & 1 & 0 \\
  1 & 0 & 0 \\
  \end{array}
  \]

  \[
  \begin{array}{cccc}
  \times & \times & \times & \times \\
  0 & 0 & 0 & 1 \\
  0 & 1 & 0 & 1 \\
  1 & 0 & 0 & 1 \\
  \end{array}
  \]

  Space of received words

- \( d = 3 \) \( \Rightarrow \) can correct 1 error
  can detect 2 errors

- \( d = 2M + 1 \) \( \Rightarrow \) correct up to \( M \) errors
  \( = M + 1 \) \( \Rightarrow \) detect up to \( M \) errors

- Given no. of errors to be corrected, have to choose codewords to satisfy distance properties.

- \( k = n - 1 \)

- Rate = \( \frac{k}{n} \)
  \( 0 \rightarrow 000 \)
  \( 1 \rightarrow 111 \)
  Rate = \( \frac{1}{3} \)

- Choose \( 2^k \) codewords out of \( 2^n \) possibilities.
Pwe = word error rate

\[ P \text{we} = \Pr(2 \text{ or more bits in error in a triplet}) \]

\[ = \left(\frac{3}{2}\right) p^2 (1-p) + \left(\frac{3}{3}\right) p^3 \]

\[ = 3 p^2 - 2 p^3 \]

\[ \binom{n}{k} = \text{binomial coefficient} = \frac{n!}{k! (n-k)!} \]

For \( p = 8 \times 10^{-4} \)

\[ P \text{we} = 3 p^2 - 2 p^3 \approx 3 p^2 \]

\[ = 0.0192 \times 10^{-4} < 10^{-4} \text{ as required} \]

Prob. of undetected errors

\[ = \Pr(3 \text{ errors}) = p^3 = 512 \times 10^{-12} \]

Performance - efficiency tradeoff.

Single-Parity Check Code

Detection of single error

\[ b_1, m_1, m_2, \ldots, m_k \quad (k+1, k) \text{ code} \]

Parity bit (check bit)

Check digit added such that the Hamming weight of each codeword is even or odd

⇒ even no. of 1's (even parity)

E.g.

\[
\begin{array}{cccc}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1
\end{array}
\]

\[ b_1 = m_1 \oplus m_2 \]

or \( m_1 \oplus m_2 \oplus b_1 = 0 \)

\[ b_1, m_1, m_2 \]

Algebraic relation between parity bit and information bits.
Can detect 1 error (e.g. 110 → 100)
Can't detect 2 errors (e.g. 110 → 000)
If there are odd no. errors (e.g. 3), can we detect?
For k-bit messages, $2^{k+1}$ codewords
$$\Rightarrow \frac{1}{2} \cdot 2^{k+1} = 2^k$$ will have odd parity
Parity check excludes this half
![Even parity](image)

$d \geq 2$ as required for single-error detection

Each word has even no. of 1's

$$\text{min. distance} = \text{min wt of any nonzero code word}$$

$$\Rightarrow \text{min dist} = 2$$

**Advantage**: high efficiency $\frac{k}{k+1} \approx 1$ if $k \gg 1$

**Disadvantage**: can't correct

**Repetition code**: poor efficiency $(\frac{1}{n})$
but can correct up to $\left\lfloor \frac{n-1}{2} \right\rfloor$ errors

Want a reasonable combination of the good properties of the two types of codes

$\Rightarrow [n, k]$ codes