Cyclic Error Correcting Codes

Grant Yang

MATH 380 Spring 2026

June 2026

Introduction

- We would like to be able to communicate over noisy channels and correct any errors that occur in transit.
- Interested in *block codes*, where we encode each k bit block (our *message*) into an n bit long *codeword*.
- Restrict ourselves to *linear* and *cyclic* codes because they have some really nice properties.
  - Famous coding schemes including Reed-Solomon and Hamming codes are linear and cyclic.
- Working over the finite field $\mathbb{F}_2$:

$\times$	**0**	**1**
0	0	0
1	0	1

$+$	**0**	**1**
0	0	1
1	1	0

- Messages can be viewed as vectors $(m_1, \ldots, m_k) \in \mathbb{F}_2^k$ or as polynomials $m_1 x^{k-1} + \cdots + m_k \in \mathbb{F}_2[x]$. Similarly for codewords $(c_1, \ldots, c_n) \in \mathbb{F}_2^n \iff c_1 x^{n-1} + \cdots + c_n \in \mathbb{F}_2[x]$.

The Quotient Ring $R = \mathbb{F}_2[x]/\langle x^n - 1\rangle$

Homework 4, §2.6 Exercise 13

Let G be a Gröbner basis. For any $f, g \in \mathbb{F}_2[x]$:

1. $\overline{f}^G = \overline{g}^G \iff f - g \in \langle x^n - 1\rangle$.
2. $\overline{f+g}^G = \overline{f}^G + \overline{g}^G$.
3. $\overline{\overline{f}^G \overline{g}^G}^G = \overline{fg}^G$.

R is the set of remainders modulo $x^n - 1$, and it is a ring:

- Addition and multiplication are defined modulo G.
- Shifting $abcd \mapsto bcda$ corresponds to multiplication by x:

$$(c_1 x^{n-1} + \cdots + c_n) \cdot x \equiv c_2 x^{n-1} + \cdots + c_1 \pmod{x^n - 1}.$$

This is why we want to use polynomials!

Ideals and Cyclic Codes

Theorem

There is a bijection between cyclic linear codes of length n and ideals of $R = \mathbb{F}_2[x]/\langle x^n - 1\rangle$, and every ideal $I \subseteq R$ is principal, i.e. $I = \langle g(x)\rangle_R$ for some $g(x) \in R$.

- Let $C \subseteq R$ be the set of valid codewords for any cyclic code.
- Linearity: For any $c, c' \in C$, we must have $c + c' \in C$.
- Cyclic: For any $c \in C$, we must have $x \cdot c \in C$.
  - Multiplication by x is a circular shift.
  - This implies $h(x) \cdot c \in C$ for any $h(x) \in R$, so C is an ideal.
- We can consider the ideal $\langle C\rangle \subseteq \mathbb{F}_2[x]$, and apply the fact that every ideal in $\mathbb{F}_2[x]$ is principal along with properties of remainders.
  - We can show the generator $g \in \mathbb{F}_2[x]$ of $\langle C\rangle \subseteq \mathbb{F}_2[x]$ satisfies $g \in R$ and also generates $C = \langle g\rangle_R$.

Hamming Distance

- Let $d_H : R \to \mathbb{N}$ be the *Hamming distance* between two codewords, the number of bits at which they differ.
- Define the *Hamming weight* of a codeword $c \in R$ to be $d_H(0, c)$.
- $d_H(c_1, c_2) = d_H(0, c_2 - c_1)$, and recall that if $c_1, c_2 \in C$, then $c_2 - c_1 \in C$, where C is the set of *valid* codewords.
- The minimum codeword separation is

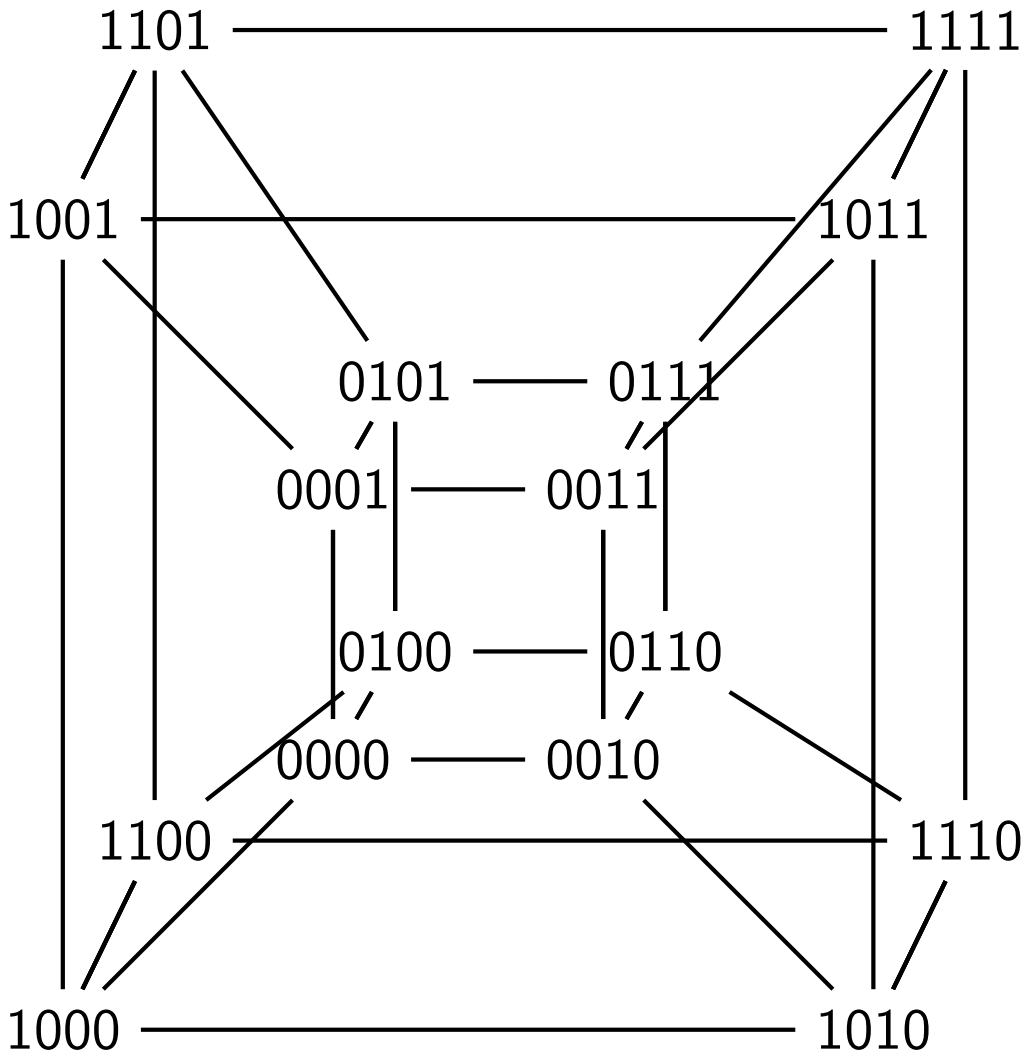
$$d = \min_{c_1, c_2 \in C} d_H(c_1, c_2) = \min_{c \in C} d_H(0, c).$$

- We can write the codeword received as the original codeword plus errors, remembering that addition is XOR:

$$c'(x) = c(x) + e(x).$$

- Any errors with Hamming weight less than $\left\lfloor \frac{d-1}{2} \right\rfloor$ are correctable.
  - We stay close to the valid codeword we originated from, as we cannot pass over the halfway point to a different codeword.

The Hamming Cube



Syndrome Decoding and Hamming Distance

The **syndrome** is the remainder

$$r(x) = \overline{c'(x)}^H = \overline{c(x) + e(x)}^H = \overline{c(x)}^H + \overline{e(x)}^H = \overline{e(x)}^H.$$

Theorem

Errors $e(x)$ with Hamming weight $\leq \left\lfloor \frac{d-1}{2} \right\rfloor$ have unique syndromes.

Proof.

- Let $e_1 \neq e_2$ be errors with weight $\leq \left\lfloor \frac{d-1}{2} \right\rfloor$.
- Assume for contradiction that $\overline{e_1}^H = \overline{e_2}^H$. Then $\overline{e_1 - e_2}^H = 0$.
- This means $e_1 - e_2$ is a valid nonzero codeword.
- However, $\text{weight}(e_1 - e_2) \leq \text{weight}(e_1) + \text{weight}(e_2) \leq d - 1$.
- This contradicts the fact that the minimum weight of a nonzero codeword is d.
- Thus, every error with weight $\leq \left\lfloor \frac{d-1}{2} \right\rfloor$ has a unique syndrome.

$\mathbb{F}_2$ Vector Spaces vs. $\mathbb{F}_2[x]$-modules

$$\text{Messages} \longrightarrow \text{Codewords} \longrightarrow \text{Syndromes}$$

$$0 \longrightarrow \mathbb{F}_2^k \xrightarrow{G} \mathbb{F}_2^n \xrightarrow{H} \mathbb{F}_2^{n-k} \longrightarrow 0$$

$$0 \longrightarrow \langle g \rangle_R \hookrightarrow R \twoheadrightarrow R/\langle g \rangle_R \longrightarrow 0$$

Hamming(7,4): Characteristics

- $n = 7, k = 4$. $R = \mathbb{F}_2[x]/\langle x^7 - 1\rangle$.
- Generator $g(x) = x^3 + x^2 + 1$.
- Minimum codeword separation of $d = 3$.
- Can correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor = 1$ bit flip.

$e(x)$	0	1	x	x^2	x^3	x^4	x^5	x^6
$\overline{e(x)}^H$	0	1	x	x^2	x^2+1	x^2+x+1	$x+1$	x^2+x

Table: Syndromes for $H = \{g(x)\} = \{x^3 + x^2 + 1\}$.

Hamming(7,4): Worked Example

Working with Gröbner basis $H = \{g(x)\} = \{x^3 + x^2 + 1\}$.

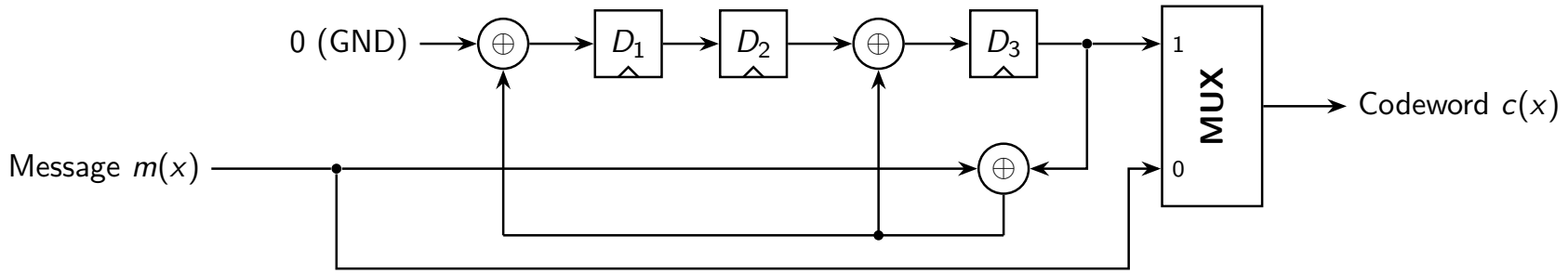
- **Message:** $1010 \implies m(x) = x^3 + x$.
- **Encode:** $c(x) = m(x)g(x) = x^6 + x^5 + x^4 + x \implies 1110010$.
- Receive $c'(x) = x^5 + x^4 + x \implies \underline{\mathbf{0}}110010$.
- **Decode:** Syndrome $\overline{c'(x)}^H = x^2 + x$.
- Lookup table: $x^2 + x \mapsto e(x) = x^6$.
- Recover original codeword $c(x) = c'(x) + x^6 \implies 1110010$.

Hamming(7,4): Systematic Encoding

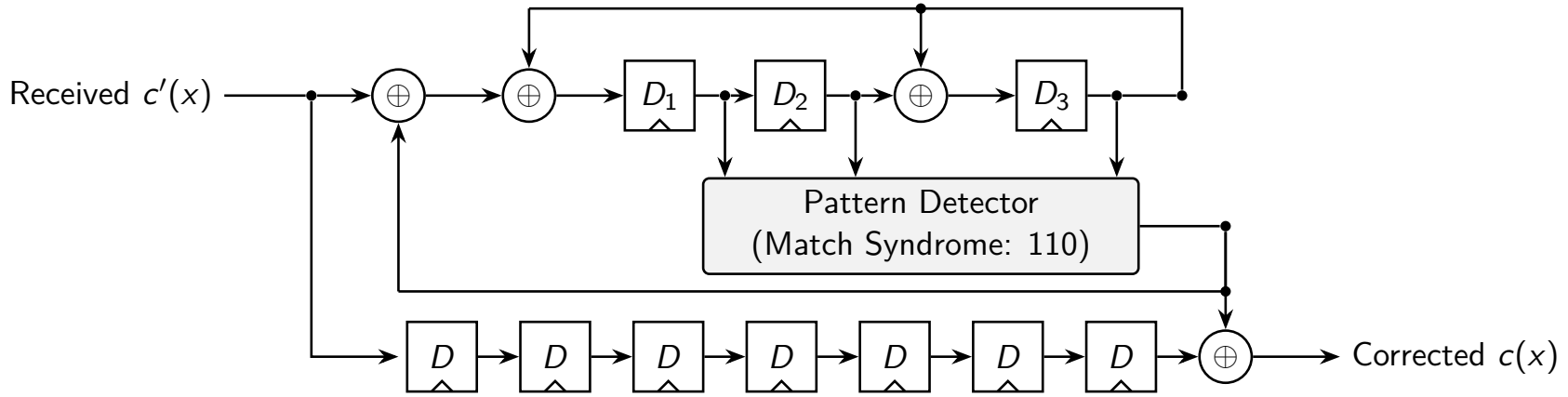
Systematic Encoding: $c(x) = x^{n-k}m(x) - \overline{x^{n-k}m(x)}^{H}$.

- **Message:** $1010 \implies m(x) = x^3 + x$.
- $x^3 m(x) = x^6 + x^4$. Remainder modulo g is 1.
- **Encode:** $c(x) = x^6 + x^4 + 1 \implies \underline{1010}001$
  - 3 parity bits appended.
  - The first k bits of the codeword matches the message!
- Receive $c'(x) = x^4 + 1 \implies \underline{\mathbf{0}}010001$.
- **Decode:** Syndrome $\overline{c'(x)}^{H} = x^2 + x$.
- Lookup table: $x^2 + x \mapsto e(x) = x^6$.
- Recover $c(x) = c'(x) + x^6 = \underline{1010}001$.

Systematic Encoder & Meggitt Decoder



Mux selects raw message (0) for first $k = 4$ clock cycles, then selects shift register output (1) for $n - k = 3$ cycles (parity bits).



Linear Feedback Shift Register (LFSR) implements multiplication by x in the ring $\mathbb{F}_2[x]/\langle x^3 + x^2 + 1\rangle$.

References

[1] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer International Publishing, 2015. ISBN: 9783319167213. DOI: 10.1007/978-3-319-16721-3. URL: http://dx.doi.org/10.1007/978-3-319-16721-3.

[2] David A. Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*. Springer-Verlag, 2005. ISBN: 0387207066. DOI: 10.1007/b138611. URL: http://dx.doi.org/10.1007/b138611.

[3] Rohan Ramkumar and Grant Yang. *Information Theory 1 Course Notes*. https://students.washington.edu/granty29/2025/Info_Theory_1_Notes.pdf. Lecture notes from Dr. Anu Aiyer's course at Harker. 2025.