L16

Wednesday, September 28, 2022 1:21 PM

* View a codeword as a vector

$$\vec{c} = (c_0, c_1, \dots, c_{n-1})$$

Construction 1

* $\vec{c} = G\vec{m}$ in $GF(p)$ or in $GF(p^m)$

$$G \triangleq \begin{bmatrix} I \\ P \end{bmatrix} \qquad P: (n-k) \times k$$

* View a codeword as a polynomial

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

$$\deg(c(x)) \leq n-1$$

Construction 2

* A polynomial-based construction of construction

$$m(x) = m_0 + m_1 x + m_2 x + \cdots + m_{k-1} x^{k-1}$$

$$\deg(m(x)) \leq k-1$$

$$c(x) = m(x) \cdot g(x) \quad \text{where } \deg(g(x)) = n-k$$

* $g(x)$ is called the generator polynomial

Corollary: Construction 2 is a subclass

Corollary: Construction 2 is a strict subclass of Construction 1.

proof: $C_0 = m_0 \cdot g_0$

$C_1 = m_1 \cdot g_0 + m_0 \cdot g_1$

$C_i = \sum_{j=0}^{\min(i,k)} m_j \cdot g_{i-j}$ for $i = 0, \cdots, n-1$

However, the design freedom is $g_0, \cdots, g_{n-k}$, which is much smaller than the design freedom $P: (n-k) \times k$.

In practice, how do we encode?

Soln #1: By multiplication.

$C(x) = m(x) \cdot g(x)$

Soln #2: By division. (Much more popular)

$\boxed{\text{Every codeword is a multiple of } g(x).}$

$$\Rightarrow \overbrace{\boxed{p_0 \mid p_1 \mid \cdots \mid p_{n-k-1} \mid m_0 \mid m_1 \mid \cdots \mid m_{k-1}}}^{n \text{ coordinates}}$$

$$\Rightarrow \underbrace{\boxed{p_0 \mid p_1 \mid \cdots \mid p_{n-k-1}} \boxed{m_0 \mid m_1 \mid \cdots\cdots \mid m_{k-1}}}_{C(x)}$$

$$\underbrace{m_0 \; m_1 \; \cdots \; m_{k-1}}_{\text{last } k \text{ coordinates}}$$

That is, we only need to compute the $p_0, \cdots, p_{n-k}$ values such that the resulting $C(x)$ is a multiple of $g(x)$

$\Longrightarrow$ By long division

$$g(x) \,\big)\!\overline{\; m_{k-1} \;\; m_{k-2} \;\; \cdots \; m_0 \; \overbrace{0\;0\;0\;0\;0}^{n-k \text{ zeros}}} \to -d(x)$$

$$d(x) \to 0$$

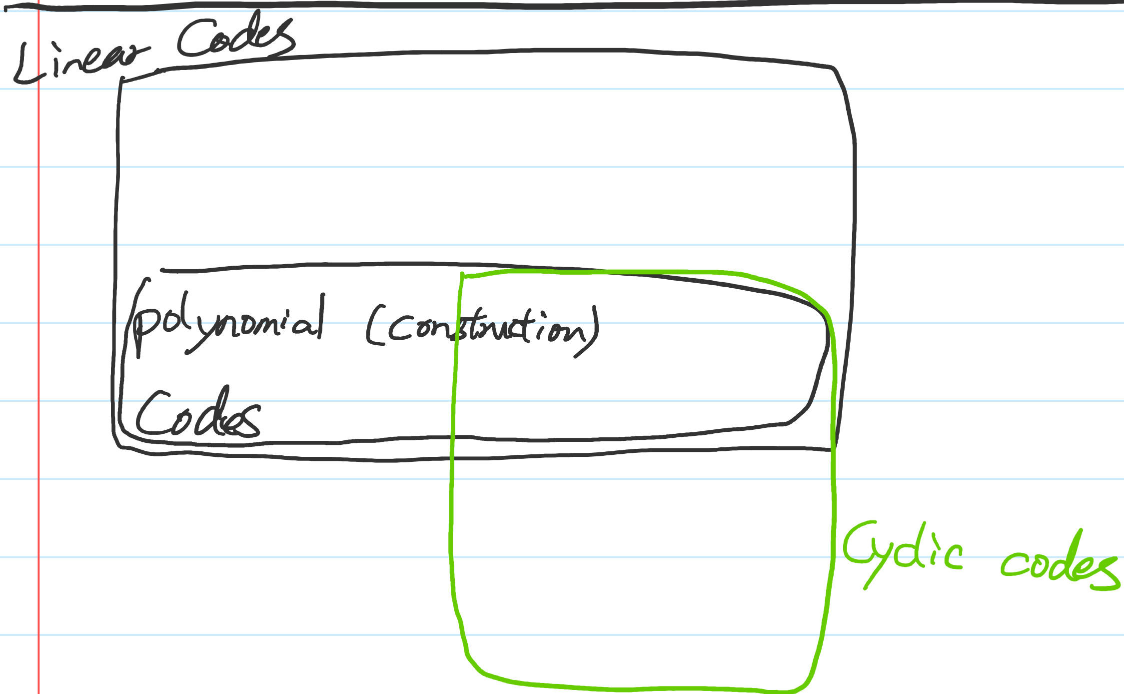
that is

$$d(x) = x^{n-k} \cdot m(x) \mod g(x)$$

then $x^{n-k} \cdot m(x) - d(x)$ will be a multiple of $g(x)$ with the highest k coefficients being exactly $m_{k-1}, \cdots, m_0$. $\Longrightarrow$ Systematic Code

$m_{k-1}, \cdots, m_0 \Longrightarrow$ Systematic Code

m_{k-1}	m_{k-2}	$\cdots$	m_0	$-d_{n-k-1}$	$-d_{n-k-2}$	$\cdots$	$-d_0$



A code $\mathbb{C}$ is cyclic

$\Rightarrow \forall (c_0, c_1, \cdots, c_{n-1}) \in \mathbb{C}$, we also have

$$(c_1, c_2, \cdots, c_{n-1}, c_0) \in \mathbb{C}$$

(i.e. the cyclically-shifted version is also a codeword.

Construction of a linear cyclic code.

Step 1: Fix the finite field size

$GF(p^m)$ (each coordinate of the n-dim codeword is in $GF(p^m)$)

Step 2: Choose the codeword length n.

Step 3: Choose $g(x)$ to be a non-trivial factor of $x^n - 1$, i.e.

$$g(x) \cdot h(x) = x^n - 1.$$

Usually choose the $g(x)$ with the leading coeff $= 1$

Step 4: $\deg(g(x)) = r \leq n-1$ ($\Rightarrow g_r \neq 0$)

Set $k = n - r$.

$$m(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}$$

then $C(x) = m(x) \cdot g(x)$ is cyclic.

Theorem: This the only way to construct a linear cyclic code. That is, any linear cyclic code can be constructed by a special $g(x)$ that divides $x^n - 1$

Corollary: $g_0 \neq 0$. since $g(x)$ divides $x^n - 1$

Discussion #1: Lemma: if $g(x)$ is a factor of $x^n - 1$, then the codebook is cyclic.

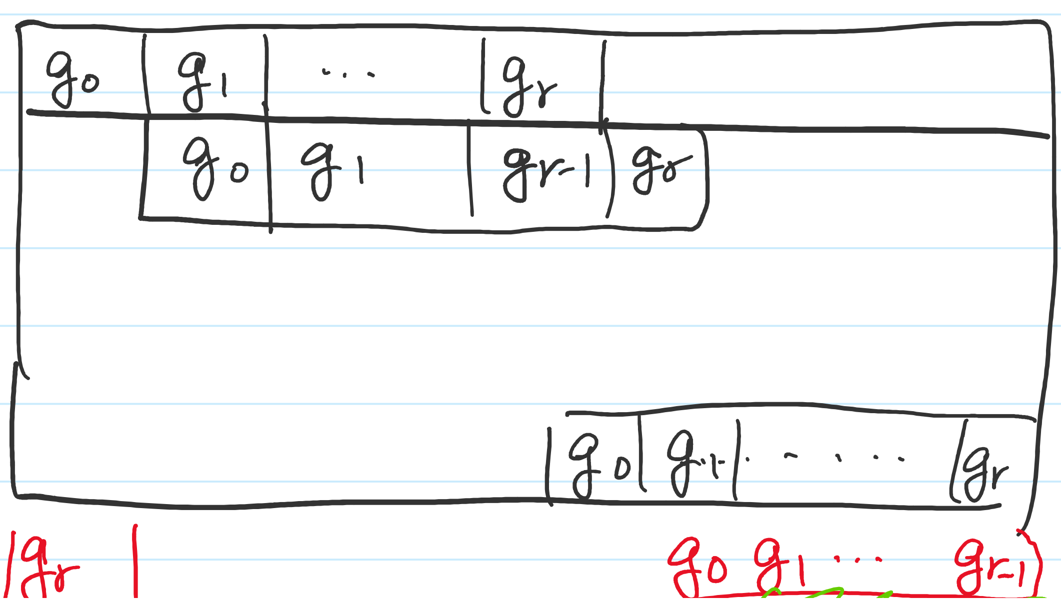
---

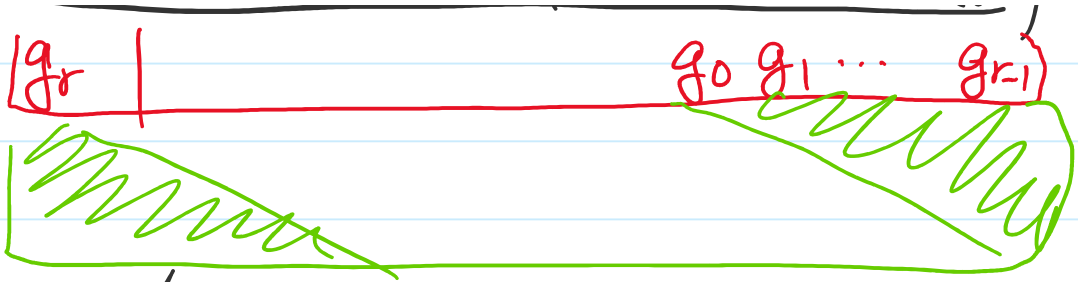
let us switch back to

$G: k \times n$

If we choose $m_0 = 1$, and all other $m_i = 0$.

then we have a codeword



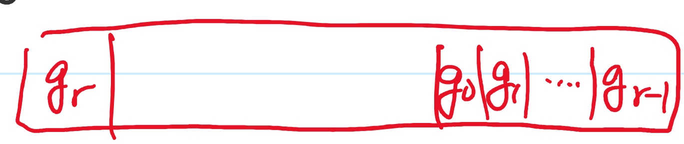


Repeatedly, we choose $m_i=1$ and all other $m_j=0$, we have k-shifted codewords. Clearly it forms a basis of the codebook because of the diagonal structure, and $g_0 \neq 0$.

* It is almost cyclic. (when shifting at most k times)

* Question: Is continuously shifting them still in the codebook?

* Question: Can we find $(m_0 + m_1 x + \cdots + m_{k-1} x^{k-1})$ such that $m(x) \cdot g(x)$

gives us 

Answer: Observation #1.

We know $g(x) \cdot h(x) = x^n - 1$

We know $g(x) \cdot h(x) = x^n - 1$

$\deg(g(x)) = n-k = r$

$\deg(h(x)) = k$

WLOG: $g_r = 1$ the $h_k = 1$ leading coefficients are 1

$k+r-1 = n-1$

Observation #2:

$\boxed{} \to g_r \cdot x^0 + g_0 x^k + g_1 x^{k+1} + \cdots g_{r-1} x^{k+r-1}$

proof: $= x^k \cdot g(x) \mod x^n - 1$

by long division

$$\begin{array}{r|l} & \qquad\qquad\qquad\qquad\qquad\qquad g_r \\ \hline x^n - 1 & g_r x^n + \cdots\cdots \qquad + g_0 \cdot x^k \\ & g_r x^n \qquad\qquad\qquad\qquad\qquad - g_r \\ \hline & \qquad g_{r-1} x^{n-1} + \cdots + g_0 x^k \qquad + g_r \end{array}$$

Combine the above, we set $\because \deg(h(x)) = k$

$m(x) = (x^k \mod h(x)) = x^k - h(x)$

$\Rightarrow m(x) \cdot g(x) = \left[x^k - h(x)\right] \cdot g(x)$

$= x^n - 1$

k

$$= x^k \cdot g(x) - \overbrace{\boxed{g(x) \cdot h(x)}}^{= x^n - 1}$$

$$= \underbrace{x^k \cdot g(x) \bmod x^n - 1}_{\text{the cyclically shifted version}}$$

Similar reasons show that all n shifted versions are in the codebook.

I.e. we can find $\tilde{m}(x) g(x) = c_{\text{shifted}}(x)$.

$\Rightarrow$ The codebook is cyclic.

The generator matrix of a cyclic code with generator polynomial $g(x)$ is

$$\underbrace{\begin{bmatrix} g_0 & \cdots & g_r & & \\ & \ddots & & \ddots & \\ & & g_0 & \cdots & g_r \end{bmatrix}}_{n} \Bigg\} k$$

Q: What is the parity check matrix?

Ans: Recall $g(x) \cdot h(x) = x^n - 1$ and $\deg(h(x)) = k$