L18 Block Codes

18.1 Block Code Review

- (n, k) notation ... turn 2^k k-tuples $\overline{m}_i$ into 2^k n-tuple codewords $\overline{U}_i$
- 2^n possible n-tuples $\overline{V}_i$ but only $\overline{U}_i$ (2^k) are valid
- ideally I can detect $2^n - 2^k$ error sequences
- and correct up to 2^{n-k} error sequences
- Systematically generate $\overline{U}_i$ from messages

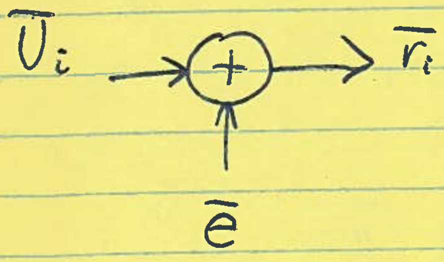
$$\overline{U}_i = \overline{m}_i \overline{G}$$

generator matrix ... formed from some basis vectors $\overline{V}$

- from $\overline{G}$ can form parity check matrix $\overline{H}^T$
- i.e. $\overline{H}$ is a re-arrangement of $\overline{G}$ such that

$$\overline{G}\,\overline{H}^T = 0 \qquad \therefore \quad m_i \overline{G}\,\overline{H}^T = \overline{U}_i \overline{G}\,\overline{H}^T = 0$$

- and errors?



$$\bar{r}_i \bar{H}^T = (\bar{U}_i \oplus \bar{e})\bar{H}_T = \bar{e}\bar{H}^T = \bar{S}$$

$\bar{S}$: syndrome can be used to identify errors

how do we use syndrome for error correction?

18.2 Making Codes

- Correcting errors starts with a code $\bar{U}$
- How do you come up with $\bar{U}$?
- Many approaches...

1. Select k you want to handle (depends on details of application)

2. What $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ do you want?

   Use appropriate bound to estimate the n you will need

   e.g. Hamming bound

$$2^{n-k} \geq \left[1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}\right]$$

(useful for high rate codes. $R = \frac{k}{n} > 0.5$)

for low-rate codes try Plotkin bound: $d_{min} \le \frac{n \cdot 2^{k-1}}{2^k - 1}$

... now you know n

3. Come up with 2^k n-tuple codewords

$\overline{U} \rightarrow$ make sure they satisfy d_{min}, all zeros vector, $\underbrace{\overline{U}_i + \overline{U}_j = \overline{U}_k}_{\text{linear subspace property}}$

4. Make $\overline{G}$ and $\overline{H}^T$ ($\overline{G}$ is a basis for the subspace $\overline{U}$, use some row-reduction method to construct it)

generator matrix

parity check matrix

5. Figure out which $\bar{e}$ (error vectors) you'll be scanning for ... you can only handle 2^{n-k}, not all possibilites, so you must be selective

for this you use the...

18.3 Standard Array

a) lay out your codewords

$\overline{U}_1 \quad \overline{U}_2 \; \ldots \; \overline{U}_i \; \ldots \; \overline{U}_{2^k}$

b) select $\bar{e}_2$ NOT IDENTICAL to any vector in first row and create new row with $\bar{e}_2$ added to your $\bar{U}$'s

$$\bar{U}_1+\bar{e}_2 \quad \bar{U}_2+\bar{e}_2 \;\cdots\; \bar{U}_i+\bar{e}_2 \;\cdots\; \bar{U}_{2^k}+\bar{e}_2$$

$\bar{e}_2$ can be arbitrary, but you are best to choose the most probable vector (e.g. one with smallest Hamming weights to start... more likely to have few errors than lots)

c) now choose another $\bar{e}_3$, make it smallest weight and a value that does not appear in the 1st 2 rows

$$\bar{U}_1+\bar{e}_3 \quad \bar{U}_2+\bar{e}_3 \cdots \bar{U}_i+\bar{e}_3 \cdots \bar{U}_{2^k}+\bar{e}_3$$

d) ... keep going until you have created an array of

2^n possible values ... 2^k columns, so must have 2^{n-k} rows (rows are called cosets)

- thus you will be able to correct for 2^{n-k} errors

that is ~~you~~ you have...

$$\bar{U}_1 \qquad \bar{U}_2 \cdots \qquad \bar{U}_i \quad \cdots \quad \bar{U}_{2^k}$$

$$\boxed{\bar{V}_1'} \qquad \boxed{\bar{V}_2'} \cdots \qquad \boxed{\bar{V}_i'} \quad \cdots \quad \boxed{\bar{V}_{2^k}'}$$

decoding regions — different received sequences that get mapped back to my codewords

- you could correct for more than d_{min} errors

  e.g. (6, 3) code in book with $d_{min} = 4$ ($t = 1$) can correct for $2^{n-k} = 2^{6-3} = 2^3 = 8$ errors

- in a 6-tuple there are only 6 1-error possibilities, throw out 1 for all zeros (i.e. no error) "error" so you have one other error you can look for ... say 010001

18.4 ... Continuing

... with our legitimate errors selected using the STANDARD ARRAY we can form a syndrome LUT

$$\bar{S} = \bar{e}\,\bar{H}^T$$

6

ERROR	SYNDROME
$\bar{e}_1$	$\bar{S}_1$
$\vdots$	$\vdots$
$\bar{e}_{2^{n-k}}$	$\bar{S}_{2^{n-k}}$

7 Now when $\overline{r}$ arrives we calculate

$$\overline{S} = \overline{r}\,\overline{H}_T$$

look up $\overline{e}_i$ and get $\overline{U} = \overline{r} \oplus \overline{e}_i$