Recitation 8

Today:
1. Convolutional Codes
2. Hamming Codes, if time permits.

Announcement: Quiz 1 Review
Office Hours

Recall:
- Elias, 1955 - use of shift registers to add redundancy
- Block Codes: Segment data stream into k-length fixed "blocks"
  - Encode blocks into length-\(n\) codewords
  \[ \text{Rate} \frac{k}{n} \]
- Convolutional Codes: Encode entire data stream (possibly infinite) into single codeword
  - Encoder outputs \(n\)-bits for every \(k\)-input bits - rate \(\frac{k}{n}\)

Example: Consider rate-\(\frac{1}{2}\) encoder shown below:

Assuming encoder initialized to all-zeros, encode the input sequence

\[ x = (1, 0, 1, 1, 0) \]

<table>
<thead>
<tr>
<th>Time</th>
<th>(x[i])</th>
<th>Current State</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>000</td>
<td>100</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>100</td>
<td>010</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>010</td>
<td>101</td>
<td>01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>101</td>
<td>110</td>
<td>01</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>110</td>
<td>011</td>
<td>01</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>011</td>
<td>001</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>001</td>
<td>000</td>
<td>11</td>
</tr>
</tbody>
</table>

Effective rate = \(\frac{5}{14} < \frac{1}{2}\) - code rate
- Moral: Encode longer bit sequences!

Use commas to separate bits output at same time
- Output: \(y = (11, 01, 01, 01, 11, 01, 11)\)

End!
Linear: If inputs $x_1$ and $x_2$ yield outputs $y_1$ and $y_2$ respectively, then input $(x_1 + x_2)$ yields output $(y_1 + y_2)$ is also a codeword.

Impulse Response: (aka "generator sequence") $\frac{g(t)}{j}$

(eq) for our previous encoder, taps at

$g(1) = (1011)$
$g(2) = (1101)$

Example 2: Use the generator sequences to encode the message in (eq)

$X = (10110)$

Sol:

Define D-transforms:

$X(D) = (1 + D^2 + D^3)$ ~ $(10110)$
$G^{(0)}(D) = (1 + D^2 + D^3)$ ~ $(1011)$
$G^{(1)}(D) = (1 + D + D^3)$ ~ $(1101)$

Then,

$Y^{(0)}(D) = X(D) \cdot G^{(0)}(D) = (1 + D^2 + D^3)(1 + D^2 + D^3) = 1 + \cdots + D^4 + \cdots + D^8 = (100010)$

and

$Y^{(1)}(D) = X(D) \cdot G^{(1)}(D) = (1 + D^2 + D^3)(1 + D + D^3) = 1 + D + D^2 + D^3 + D^4 + D^5 + D^6 = (111111)$

Then

$\frac{Y^{(0)}}{j} = (1000101)$
$\frac{Y^{(1)}}{j} = (1111111)$

and thus

$\frac{Y}{j} = (11, 01, 01, 01, 11, 01, 11)$, as above.
Constraint Length $K$:

- max number of output bits that can be affected by any input bit.

For $m$ memory elements, each bit in input data sequence can affect at most $(m+1)$ bits, hence length of $g = (m+1)$; i.e. $m$ determines extent to which an input bit directly affects the output data streams.

$$K = 1 + \text{max } m$$

~ not universal definition, but used in military & industry!

max $m$ ~ maximal memory order ~ length of longest input shift register.

State Diagram:

For the encoder in example 1 with $g^{(10)} = (1011)$ and $g^{(1)} = (1101)$,
use the state diagram to encode $x = (10110)$ as before.

Solution:

Eight states: (000) ~ (111)

0, 01, 10, 11

$$\bar{y} = (11, 01, 01, 01, 11)$$

then input gees back to 000.
**Systematic Codes:**

An \((m,k)\) linear code systematic if 'first' \(k\)-bits are uncoded data bits, so that

The generator matrix \( G = \begin{bmatrix} I & P \end{bmatrix} \) (where \( I \) is the identity matrix)

The parity check matrix \( H = \begin{bmatrix} P^T & I \end{bmatrix} \)

Thus, \( GH^T = \begin{bmatrix} I & P \end{bmatrix} \begin{bmatrix} P \\ I \end{bmatrix} = P + P = \emptyset \)

\( s \), for any codeword \( \hat{c} = mG \),

\[ \hat{c}H^T = mGH^T = m(GH^T) = \emptyset \]

**Syndrome:**

Given received codeword \( \hat{r} \),

syndrome is \( \hat{s} = \hat{r}H^T \)

If \( \hat{s} = 0 \), no error.

If single-bit error,

\[ \hat{r} = \hat{c} + \hat{e}_i \]  where \( \hat{e}_i = \begin{bmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \)  8th position

\( \hat{s} = \hat{c}H^T = (\hat{c} + \hat{e}_i)H^T = \hat{e}_i H^T \)  error in 8th bit position
Example: Hamming Code.

H.C. - \((2^m - 1, 2^m - m - 1)\) linear block code, \(d_{\text{min}} = 3\).
- Parity check matrix \(H\) contains all non-zero \(m\)-bit binary vectors.

For \(m = 3\), \(n = 7\), \(k = 4\) \(\Rightarrow (7, 4)\) HC.

\[\begin{align*}
\text{Suppose, given generator matrix } G_{4 \times 4} &= \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 \\
\end{bmatrix} \\
I & \quad P
\end{align*}\]

**Construct parity check matrix** \(H\),

\[H = [P^T; I] = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\end{bmatrix}\]

Note matrix contains all non-zero 3-bit vectors!

**Q:** If received vector \(r = (1 \ 0 \ 1 \ 1 \ 1 \ 0)\), decode original sent message.

Syndrome \(s = r^H = (1 \ 0 \ 1 \ 1 \ 1 \ 0) \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\end{bmatrix} = (0 \ 1 \ 0)\)

\( \text{sixth bit position} \)

Thus, corrected \(\tilde{r} = (1 \ 0 \ 1 \ 1 \ 1 \ 0)\) \(\text{corrected.}\)

Original message \(= (1 \ 0 \ 1)\) since systematic!