Agenda

- Linear Block Codes
- Convolutional Codes
  - Codes
  - State Machine
  - Transfer Function
  - Decoding
  - Performance
- CDMA 2000
  - Pilot
Channel Coding

• Channel coding can be described as the clever use of redundancy

• Channel coding can be divided into two classes
  - Block codes
    • Binary source output sequences of length k are mapped into binary channel input sequences of length n
    • Rate k/n bits per transmission
    • (n,k) block codes
  - Convolutional codes
    • Source outputs of length k are mapped into n channel inputs, but the channel inputs depend not only on the most recent k source outputs, but also on the last (L-1)k inputs of the encoder
    • L is the constraint length
Simple Example

• Assume a simple majority rule block code where a bit is sent n times where n is an odd number

• An error occurs if \((n+1)/2\) bits out of the n bits are in error

\[
p_e = \sum_{k=(n+1)/2}^{n} \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}
\]

• Example: assume \(n = 5\) and \(\epsilon = 0.001\)

\[
p_e = \sum_{k=3}^{5} \binom{5}{k} 0.001^k (1 - 0.001)^{5-k} = 10^{-9}
\]

The channel rate is only 20% of the raw data rate, but the error rate is one million times better.
Linear Block Codes - 1

- A block code is linear if any linear combination of two codewords is a codeword
- In linear block codes the codewords form a k-dimensional subspace of an n-dimensional space
- Codewords are generated by $c = uG$
- Hamming Distance
  $$d = \min_{i \neq j} d_H(c_i, c_j)$$
- Systematic Form $G = [I_k \mid P]$
• Parity Check Matrix, $H$, where $cH^t = 0 = GH^t$

and $H = [P^t | I_k]$ if $G$ is in systematic form

• **Hamming codes** are codes $(2^m - 1, 2^m - m - 1)$ linear block codes with minimum distance 3

  - The parity check matrix, which is an $m \times (2^m - 1)$ matrix has all binary sequences of length $m$ except the all zero sequence as its columns
Linear Block Codes - 3

- Example $m = 3$ or $(7,4)$

\[ H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \]
Performance of Linear Block Codes

- Hard and Soft Decoding
- Hard Decoding \( p_e \leq (M-1)[4p(1-p)] \)

- Soft Decision
  \( p_e \leq (M-1)Q\left(\frac{d}{\sqrt{2N_o}}\right) \)
  \( d = \sqrt{2d_{\min}E} \) for orthogonal signaling
  \( d = \sqrt{4d_{\min}E} \) for orthogonal signaling

- \( E = \frac{E}{R_c} \) and \( R_c = \frac{k}{n} \)
Convolutional Codes-1

- Example of a convolutional code

\[ G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \]

Typically we assume the shift registers are loaded with zeros initially.

\[ R_c = \frac{Nk}{(N + L - 1)n} \approx \frac{k}{n} \quad \text{for large } L \]
Example of a convolutional code

\[ G = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \]

2^{(L-1)k} states
Convolutional Codes-3

- Example of a convolutional code
Convolutional Coding-4

- State Machine
The exponent of J is the number of branches spanned by the path.
The exponent of D shows the number of ones in the codeword corresponding to the path.
The exponent of N is the number of ones in the input information sequence.
Convolutional Codes-5

- Transfer Function

\[ X_c = X_a D^2 N J + X_b N J \]

\[ X_b = X_d D J + X_c D J \]

\[ X_d = X_c D N J + X_d D N J \]

\[ X_a'' = X_b D^2 J \]

\[ T(D, N, J) = \frac{X}{X} = D^5 N^3 J^2 \]

\[ T(D, N, J) = D^5 N^3 \quad + D^6 N^2 J^4 \quad + D^6 N^2 J^5 \quad + \ldots \]

One path, Three ones, Hamming distance of five
Convolutional Codes-6

- **Transfer Function**

\[T(D, N, J) = \sum_{d=\text{d}_{\text{free}}}^{\infty} a_d D^d N^d J^d f(d) g(d)\]

\[T(D) = \sum_{d=\text{d}_{\text{free}}}^{\infty} a_d D^d\]

Setting \(N=J=1\)

- **For the example**

\[T(D) = \frac{D^5 N^3 J^5}{1 - D N J - D N J^2} = \frac{D^5}{1 - 2D} = D^5 + 2D^6 + 4D^7 + \ldots\]

\[T(D) = \sum_{i=0}^{\infty} 2^i D^{5+i}\]
Convolutional Codes-7

- Decoding Convolutional Codes

Input  | 0 | 1 | 0 | 1 | 0 | 1 | 0
Encoded | 00 | 11 | 01 | 00 | 01 | 00 | 01
Rcv'd   | 00 | 11 | 11 | 00 | 01 | 00 | 01
Output  | 0 | 1 | 0 | 1 | 0 | 1 | 0
Metric  | 0 | 0 | 1 | 2 | 4 |     | Merge Paths
Metric  | 0 | 0 | 1 | 1 | 1 |     |
Convolutional Codes-8

- Performance

Where \( P_2(d) \) is the probability that a path through the trellis that is a Hamming distance, \( d \), from the all zeros path is the survivor at the \((l+1)\) stage.

Since \( d \) is larger than \( d_{\text{free}} \), we can bound the first error event probability by

\[
P \leq \sum_{e}^{\infty} a_{d} P_{\text{free}}(d) \]

\[
\begin{array}{ccc}
  & \hspace{1cm} & \hspace{1cm} \\
  l-1 & l & l+1 \\
  & \hspace{1cm} & \\
  & \hspace{1cm} & \\
  & \hspace{1cm} & \\
  & \hspace{1cm} & \\
  & \hspace{1cm} & \\
\end{array}
\]
Convolutional Codes-9

- Performance of a soft decoder for BPSK

\[
P_{2}(d) = Q\left( \frac{E}{d} \sqrt{\frac{2}{N}} \right) = Q\left( \sqrt{\frac{2Ed}{N}} \right) = Q\left( \sqrt{2R} \frac{E}{d} \right)
\]

\[
P \leq \sum_{d=\text{free}}^{\infty} a_d Q\left( \sqrt{2R} \frac{E}{d} \right) \quad Q(x) = \frac{1}{2} e^{-x^2 / 2}
\]

\[
Q\left( \sqrt{2R} \frac{E}{d} \right) \leq \frac{1}{2} e^{-R \frac{dE}{N}}
\]
Convolutional Codes-10

• Performance of a soft decoder for BPSK

\[
e^{-R_c d E_b / N_o} = D^d
\]

\[
D = e^{-R_c E_b / N_o}
\]

\[
P \leq \frac{1}{2} \sum_{d=0}^{\infty} a_d D^d
\]

\[
D = e^{-R_c E_b / N_o}
\]

\[
T(D) = T(D, N, J) \bigg|_{N=J=1}
\]

\[
\frac{1}{2} T(D) = T(D, N, J) \bigg|_{N=J=1}
\]
Convolutional Codes-11

• Performance of a soft decoder for BPSK

  - To find a bound on the average number of bits in error, avg Pb(k), we note that each path through the trellis causes a certain number of input bits to be decode erroneously

  - For a general $D^d N^{f(d)} J^{g(d)}$ in the expansion of $T(D, N, J)$ there is a total of $f(d)$ non-zero input bits

$$\overline{P}_b (k) \leq \sum_{d=1}^{\infty} a f(d) P_2 (d) = \sum_{d=1}^{\infty} a f(d) Q \left( \sqrt{2R_c d} \frac{E}{N_o} \right)$$

$$\overline{P}_b (k) \leq \frac{1}{2} \sum_{d=1}^{\infty} a f(d) D^d$$

$$D = e^{-\frac{R}{c} T_N (D)} = T(D, N, J)$$
Convolutional Codes-11

- Performance of a soft decoder for BPSK

\[ T_2(D) = T(D, N, J) \mid J = 1 \]

\[ T_2(D, N) = \sum_{d = d_{\text{free}}}^{\infty} a^N \frac{f(d)}{D}^d D = e^{-R_c E_b / N_o} \]

\[ \delta T_2(D, N) \mid \frac{\delta N}{N = 1} = \sum_{d = d_{\text{free}}}^{\infty} a f(d) D^d \]

\[ \overline{P}_b(k) \leq \frac{1}{2} \frac{\delta T_2(D, N)}{\delta N} \mid N = 1, D = e^{-R_c E_b / N_o} \]
Convolutional Codes-12

- Performance of a soft decoder for BPSK – average number of bits in error

\[ \bar{P}_b \leq \frac{1}{2k} \left( \frac{\delta T (D, N)}{\delta N} \right)^2 \]

\[ N = 1, D = e^{-R_c E_b / N_o} \]
Convolutional Codes-13

- Performance of a hard decoder for BPSK

\[ P_2(d) \leq [4p(1-p)]^{1/2} \]

\[
\overline{P}_b \leq \frac{1}{2k} \frac{\delta T(D,N)}{\delta N} \quad \bigg| \quad N=1, \ D=\sqrt{4p(1-p)}
\]
Forward Link Modulator

\[ A + I\text{-Channel Pilot} + Q\text{-Channel Pilot} \rightarrow \text{Baseband Filter} \rightarrow \times \rightarrow \cos(2\pi f_c t) \rightarrow \sin(2\pi f_c t) \rightarrow \text{Baseband Filter} \rightarrow \times \rightarrow \text{A} \]
PN Code Properties

- Sequence of ones and zeros
- Periodic
- The number of zeros and the number of ones differ by exactly one
- In every period, \( \frac{1}{2} \) of the consecutive runs of ones or zeros have length one, \( \frac{1}{4} \) have length two, \( \frac{1}{8} \) have length three, etc.
- The autocorrelation function (how well a time shifted version of the code correlates to the original) is either N or -1
Generating PN Codes

- Generating polynomial:
  \[ g(x) = x^m + a_{m-1}x^{m-1} + \ldots + a_2x^2 + a_1x + 1 \]

- Shift register implementation
Short Codes

- Before transmitting, the pilot and the sync are spread by In- phase (I) and Quadrature phase (Q) PN sequences. After a run of 14 consecutive zeros in either sequence another zero is added in that sequence making a run of 15 consecutive zeros. This results in a period of $2^{15}$ for both the I and Q sequences. The sequence has a chip rate of 1.2288Mcps and a period of $2^{15}/1,228,800$cps = 26.667msec.

- The I and Q PN short codes can be generated using the linear recursions shown below

  - $PN_I(n) = PN_I(n-15) \oplus PN_I(n-10) \oplus PN_I(n-8) \oplus PN_I(n-7) \oplus PN_I(n-6) \oplus PN_I(n-2)$

  and

  - $PN_Q(n) = PN_Q(n-15) \oplus PN_Q(n-12) \oplus PN_Q(n-11) \oplus PN_Q(n-10) \oplus PN_Q(n-9) \oplus PN_Q(n-5) \oplus PN_Q(n-4) \oplus PN_Q(n-3)$. 