Coding for Wireless Channels

Wha Sook Jeon

Mobile Computing and Communication Lab.

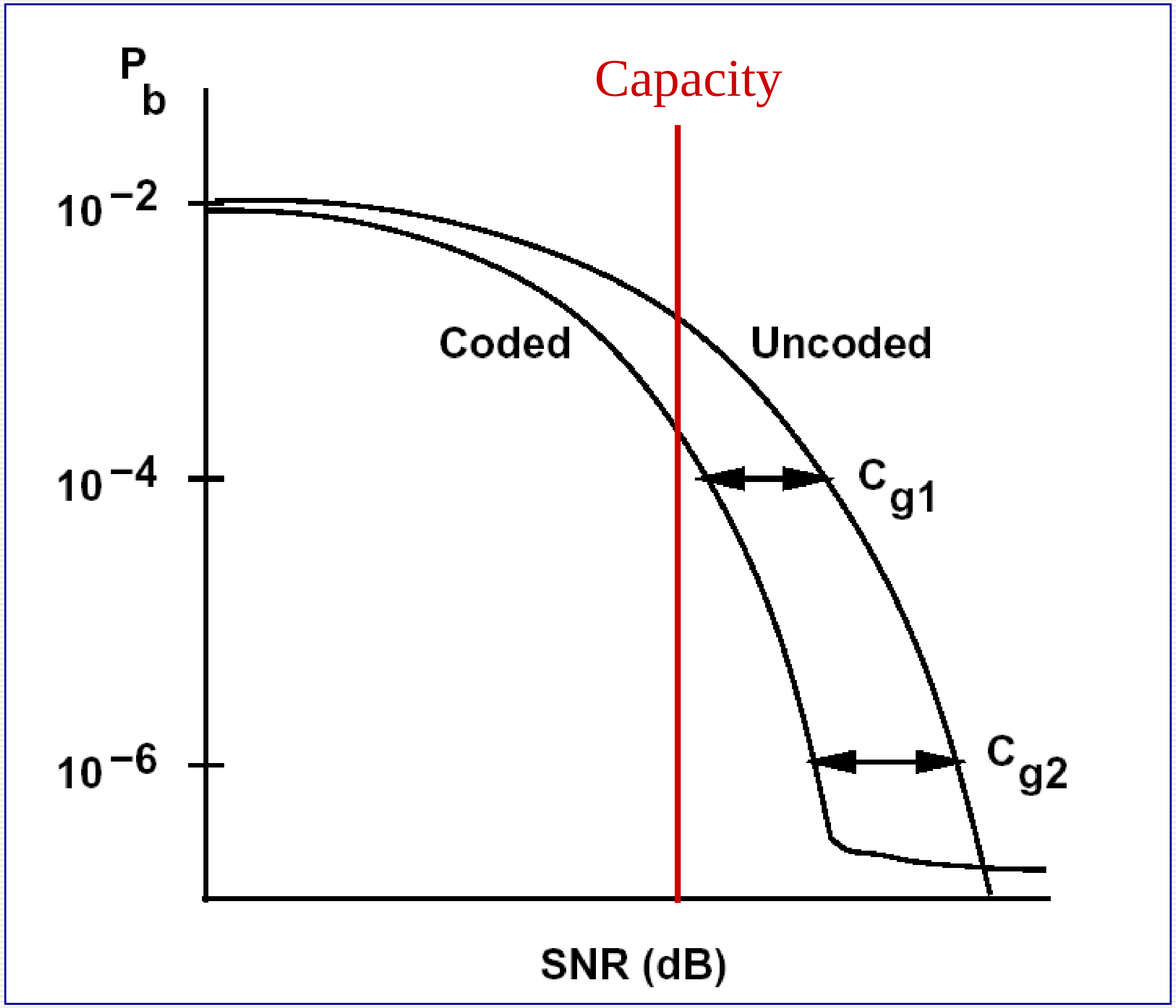
Contents

- Overview of code design
- Code design in AWGN channels
  - Linear block codes
  - Convolutional codes
  - Concatenated codes
  - Turbo codes
- Code design in fading channels
  - Combining the codes in AWGN with interleaving (diversity gain)

Overview of Code Design (1)

- Main reason to apply error correction coding in wireless systems
  - To reduce the bit error or block error probability
- Amount of error reduction provided by a given code
  - Coding gain in AWGN
  - Diversity gain in fading
- Coding gain in AWGN
  - the amount of SNR or E_b/N_0 that can be reduced under the coding technique for a given error probability
    - Sometime, negative coding gain at low SNRs, due to spreading the bit energy over multiple coded bits.
  - Capacity curve
    - It is associated with the SNR (or E_b/N_0) where the data rate of the system equals the Shannon capacity $B\log_2(1+\text{SNR})$
    - The capacity-achieving code has an error probability of zero, at rates up to capacity
    - Best performance that the practical code can achieve

Coding Gain in AWGN Channels



Overview of Code Design (2)

- Performance enhancement under the coding scheme at the cost of
  - a decrease in data rate
  - an increase in signal bandwidth
  - the increased complexity
- A joint design of the code and modulation for obtaining a coding gain without bandwidth expansion
- Codes designed for AWGN do not well work in fading channel due to the burst errors
  - Combining AWGN channel codes with interleaving
    - The interleaver spreads out the burst errors over time (time diversity)

Linear Block Codes

Binary Linear Block Codes (1)

- Linear block codes are conceptually simple codes that are basically an extension of single bit parity check codes for error detection
- (n,k) binary block code
  - A codeword of n symbols from k information bits
  - Each k bit information block is map
  - A code rate: $R_c = k/n$
  - linear if the 2^k length-n codewords of the code form a subspace **S** of the set of all binary n-tuples B_n

Modulo-2 addition

The all-zero vector is in **S**

If $S_i \in \mathbf{S}$ and $S_j \in \mathbf{S}$, then $S_i + S_j \in \mathbf{S}$

- Hamming distance between two codewords $\mathbf{C}_i$ and $\mathbf{C}_j$: d_{ij}
  - $$d_{ij} = \sum_{l=1}^{n} \left(\mathbf{C}_i(l) + \mathbf{C}_j(l) \right)$$

    where $\mathbf{C}_i(l)$ denotes the lth bit in $\mathbf{C}_i$

Binary Linear Block Codes (2)

- The weight of $\mathbf{C}_i$: $w(\mathbf{C}_i)$
  - The number of 1-bits in $\mathbf{C}_i$: $w(C_i) = \sum_{l=1}^{n} C_i(l)$
  - Hamming distance d_{0i} from the all-zero codeword
- $d_{ij} = w(\mathbf{C}_i + \mathbf{C}_j)$
- The minimum distance of code: $d_{\min} = \min_{i, i \neq 0} d_{0i}$
- Encoding operation
  - $\mathbf{U}_i = [u_{i1}, \cdots, u_{ik}]$: k information bits encoded into $\mathbf{C}_i = [c_{i1}, \cdots, c_{in}]$
  - $c_{ij} = u_{i1} g_{1j} + u_{i2} g_{2j} + \cdots + u_{ik} g_{kj} \qquad (g_{ij} = 0 \text{ or } 1)$
  - Matrix representation: $\mathbf{C}_i = \mathbf{U}_i \mathbf{G}$

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}$$

Systematic Linear Block Codes (1)

- The first k codeword symbols equal to the the information bits and remaining codeword symbols equals to the parity bits
- Generator matrix

$$\mathbf{G} = \left[\mathbf{I}_k \mid \mathbf{P}\right] = \left[\begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{matrix}\right|\left.\begin{matrix} p_{11} & p_{12} & \cdots & p_{1(n-k)} \\ p_{21} & p_{22} & \cdots & p_{2(n-k)} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{k(n-k)} \end{matrix}\right]$$

- Codeword from a systematic encoder

$$\mathbf{C}_i = \mathbf{U}_i \mathbf{G} = \mathbf{U}_i \left[\mathbf{I}_k \mid \mathbf{P}\right] = \left[\, u_{i1}, \cdots, u_{ik}, p_1, \cdots, p_{(n-k)} \,\right]$$

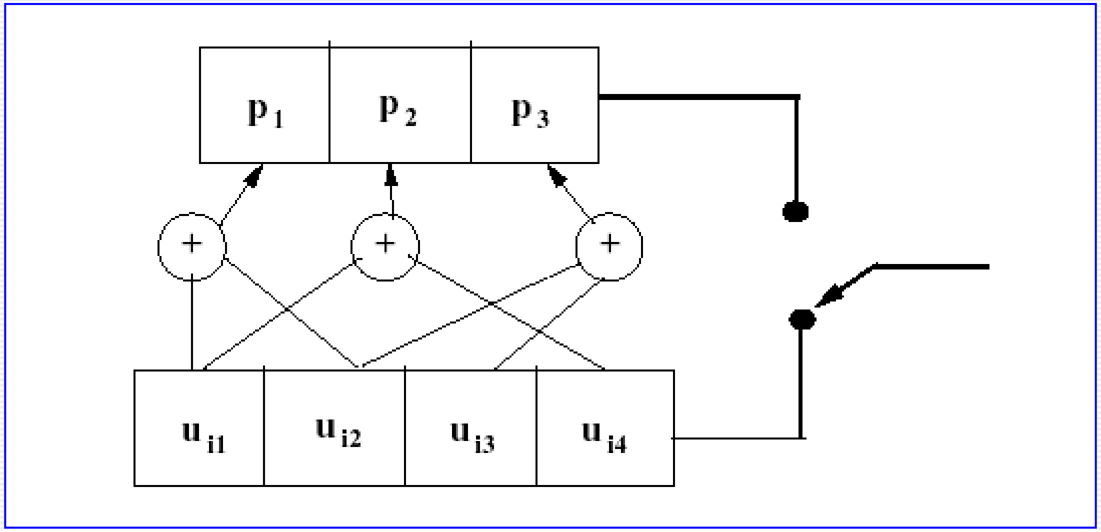
- Parity bits

$$p_j = u_{i1} p_{1j} + \cdots + u_{ik} p_{kj}, \quad j = 1, \cdots, n-k$$

Systematic Linear Block Codes (2)

- Example
  - Find the corresponding implementation for generating a (7,4) binary code with the generator matrix
  - Solution

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Systematic Linear Block Codes (3)

- Parity check matrix
  - is used to decode linear block codes with generator matrix **G**
  - Parity-check matrix **H** corresponding to $\mathbf{G}=[\mathbf{I}_k|\mathbf{P}]$

$$\mathbf{H} = \left[\mathbf{P}^T \middle| \mathbf{I}_{n-k} \right]$$

  - Since $\mathbf{G}\mathbf{H}^{\mathrm{T}} = \mathbf{0}_{k,n-k}$ (an all-zero k x (n-k) matrix),

$$\mathbf{C}_i\mathbf{H}^T = \mathbf{U}_i\mathbf{G}\mathbf{H}^T = \mathbf{0}_{n-k}$$

- Syndrome testing
  - **R**: the received codeword resulting from transmission of cordword **C**
  - **R** = **C**+**e**, where e is the error vector
  - Syndrome of R: $\mathbf{S} = \mathbf{R}\mathbf{H}^T$
  - The syndrome is a function only of the error pattern **e**

$$\mathbf{S} = \mathbf{R}\mathbf{H}^T = (\mathbf{C} + \mathbf{e})\mathbf{H}^T = \mathbf{C}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{0}_{n-k} + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T$$

Cyclic Codes (1)

- Linear block codes
- Generator polynomial

$$g(X) = g_0 + g_1 X + \cdots + g_{n-k} X^{n-k}$$

- Message polynomial

$$u(X) = u_0 + u_1 X + \cdots + u_{k-1} X^{k-1}$$

- Codeword

$$c(X) = u(X) g(X) = c_0 + c_1 X + \cdots + c_{n-1} X^{n-1}$$

  – A valid codeword for a cyclic code with generating polynomial $g(X)$ if and only if $g(X)$ divides $c(X)$ with no remainder,

$$\frac{c(X)}{g(X)} = q(X)$$

Cyclic Codes (2)

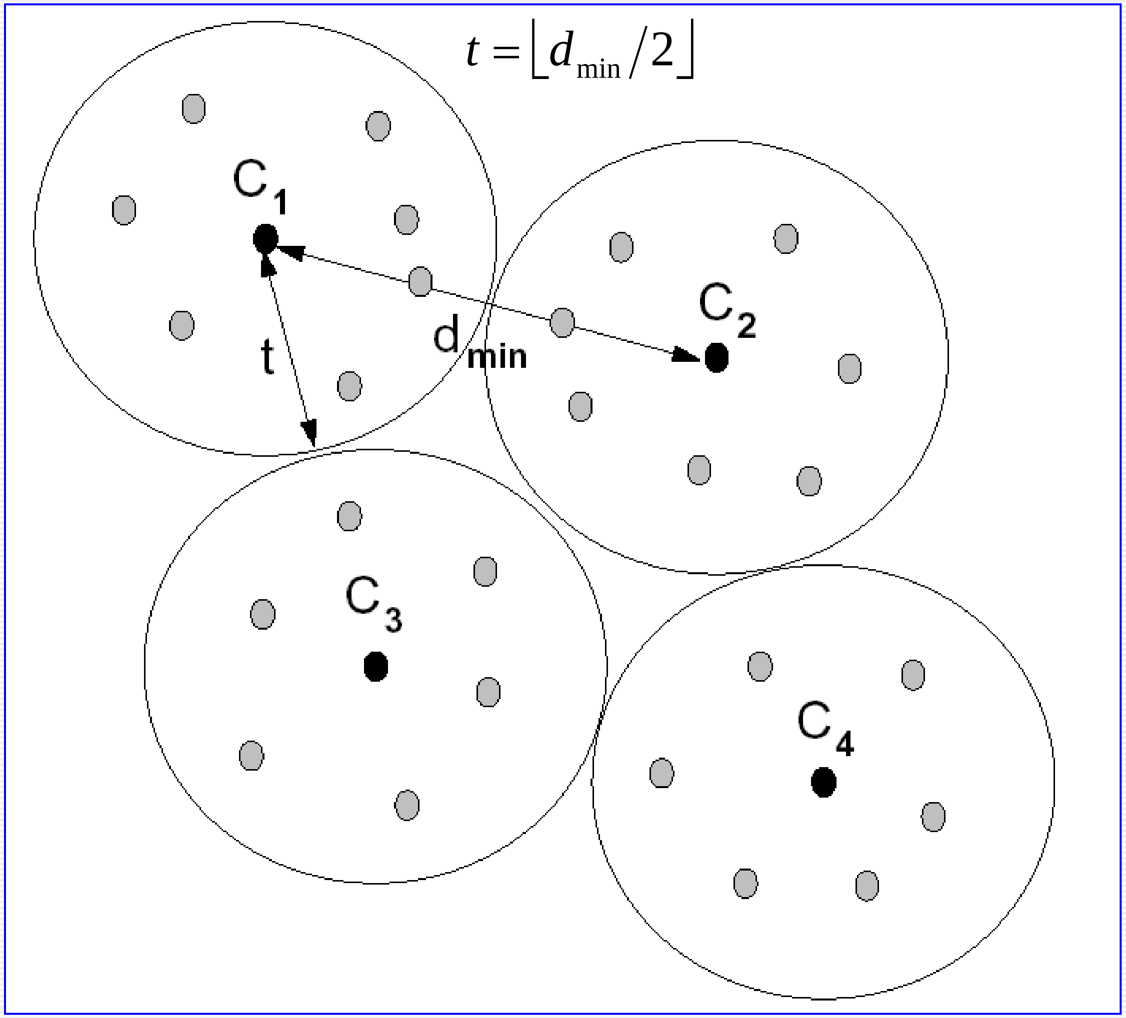
- A cyclic code can be put in systematic form
  - Multiplying the message polynomial by X^{n-k}
  - Dividing $X^{n-k}\,u(X)$ by $g(X)$ to get the remainder polynomial $p(X)$
  - Adding $p(X)$ to $X^{n-k}\,u(X)$
  - Then, the codeword is $c(X) = X^{n-k}\,u(X) + p(X)$

Hard Decision Decoding (HDD) (1)

- Each code symbol is demodulated individually as 0 or 1
  - This form of demodulation removes information that can be used by the channel decoder
- Hard decision decoding
  - Minimum distance decoding based on Hamming distance
    - Pick $\mathbf{C}_j$ s.t. $d\left(\mathbf{C}_j, \mathbf{R}\right) \le d\left(\mathbf{C}_i, \mathbf{R}\right) \;\; \forall i \ne j$
    - If there is more than one codeword with the minimum distance, one of these are randomly chosen
  - Maximum likelihood decoder chooses the codeword $\mathbf{C}_j$
    - $\mathbf{C}_j = \arg\max_i \; p\left(\mathbf{R} \mid \mathbf{C}_i\right), \quad i = 0, \cdots, 2^k - 1$
  - The minimum distance criterion is equivalent to the maximum likelihood criterion in an AWGN channel
    - since most probable error event in AWGN is the event with the minimum number of errors needed to produce the received codeword

Hard Decision Decoding (HDD) (2)

Maximum likelihood decoding



- The decoder can correct up to *t* errors
- The decoder can detect all error patterns of d_{min}-1

Probability of Error for HDD in AWGN (1)

- A received codeword may be decoded in error if it contains more than t errors. Since the bit errors in a codeword occur independently on an AWGN channel,

$$P_e \le \sum_{j=t+1}^{n} \binom{n}{j} p^j (1-p)^{n-j} \quad \text{: upper bound}$$

  - p corresponds to the error probability associated with uncoded modulation for the given energy per codeword symbol
  - When a codeword symbols are sent via a coherent BPSK modulation,

$$p = Q(\sqrt{2E_c/N_0})$$

  - Powerful block codes with a large number of parity bits reduce the energy per symbol ($E_c = k\,E_b$/n)
    - The error probability in demodulating the codeword symbol is increased.
    - At high SNR, the high correction capability compensates for this reduction.
    - At low SNR, a higher probability than uncoded modulation (negative coding gain)

Probability of Error for HDD in AWGN (2)

- At high SNR, the most likely way to make a codeword error is to mistake a codeword for one of its nearest neighbors
  - Lower bound: one nearest neighbor at distance d_{min}

$$\sum_{j=t+1}^{d_{\min}} \binom{d_{\min}}{j} p^j (1-p)^{d_{\min}-j} \le P_e$$

  - Upper bound: all of the other 2^k-1 codewords are at distance d_{min}

$$P_e \le \left(2^k - 1\right) \sum_{j=t+1}^{d_{\min}} \binom{d_{\min}}{j} p^j (1-p)^{d_{\min}-j}$$

Probability of Error for HDD in AWGN (3)

- A tighter upper bound
  - The probability of decoding the all-zero codeword as the jth codeword with weight w_j: $p(w_j)$

$$p(w_j) \le \left[4p(1-p)\right]^{w_j/2}$$

  - Since the probability of decoding error is upper bounded by the probability of mistaking the all-zero codeword for any other codeword,

$$P_e \le \sum_{j=1}^{2^k-1} \left[4p(1-p)\right]^{w_j/2}$$

  - A simple , slightly looser bound by using d_{min} instead of the individual weights

$$P_e \le \left(2^k-1\right)\left[4p(1-p)\right]^{d_{\min}/2}$$

Soft Decision Decoding (SDD)

- Soft decision decoding
  - The distance between the received symbol and the transmitted constellation point (output from the demodulator) is used in the channel decoder
  - For BPSK, if the *j*th symbol of the transmitted codeword is a 1, the received symbol from the demodulator is $r_j = \sqrt{E_c} + n_j$; if it is a 0, $r_j = -\sqrt{E_c} + n_j$
- The decoder forms a correlation metric $C(\mathbf{R}, \mathbf{C}_\mathrm{i})$ for a received codeword $\mathbf{R}=[r_1,\ldots, r_\mathrm{n}]$ and each codeword $\mathbf{C}_\mathrm{i}=(c_{i1},\ldots, c_{in})$, and chooses the codeword $\mathbf{C}_\mathrm{i}$ with the highest correlation metric.

$$C(R, C_i) = \sum_{j=1}^{n} (2c_{ij} - 1) r_j$$

if c_{ij}=0, $2c_{ij}$ -1= -1

if c_{ij}=1, $2c_{ij}$ -1=1

  - At very high SNR, if C_i is transmitted, $C(\mathrm{R}, \mathrm{C}_i) \approx n\sqrt{E_c}$

Probability of Error for SDD in AWGN

- Assume that the all-zero codeword C_0 is transmitted.
  - To correctly decode **R**, $C(\mathbf{R}, \mathbf{C}_0) > C(\mathbf{R}, \mathbf{C}_i)$ for all i ($\neq 0$)
  - $C(\mathbf{R}, \mathbf{C}_i)$ is an Gaussian random variable with mean $\sqrt{E_c}(n-w_i)-\sqrt{E_c}w_i$ and variance $nN_0/2$.
  - The probability $P_e(\mathbf{C}_i) = p(C(\mathbf{R}, \mathbf{C}_0) < C(\mathbf{R}, \mathbf{C}_i))$ is equal to the probability that a Gaussian random variable with mean $-2w_i\sqrt{E_c}$ and variance nN_0 is larger than 0

$$P_e(C_i) = Q\left(\frac{2w_i\sqrt{E_c}}{\sqrt{nN_0}}\right) = Q\left(\sqrt{\frac{2w_i}{n}}\sqrt{2w_i\gamma_b R_c}\right) \approx Q\left(\sqrt{2w_i\gamma_b R_c}\right)$$

  - By union bound: $P_e \le \sum_{i=1}^{2^k-1} P_e(C_i) = \sum_{i=1}^{2^k-1} Q\left(\sqrt{2w_i\gamma_b R_c}\right)$
  - Simplification by noting that $w_i > d_{min}$: $P_e \le \left(2^k-1\right) Q\left(\sqrt{2\gamma_b R_c d_{\min}}\right)$

Common Linear Block Codes (1)

- Binary Block Codes
  - Hamming (n, k) code
    - redundant bits $m = n - k$
    - $n = 2^m - 1$, $k = 2^m - m - 1$
    - $d_{min} = 3$, $t = 1$ (not powerful)
    - Perfect code: $P_e = \sum_{j=t+1}^{n} \binom{n}{j} p^j (1-p)^{n-j}$
  - Golay and Extended Golay
    - Golay (23,12): $d_{min} = 7$, $t = 3$
    - Extended Golay (24,12): adding a single parity bit to Golay (23,12)
      - Error capability is not changed between two codes
      - Simple implementation (the bit rate is half the code rate)

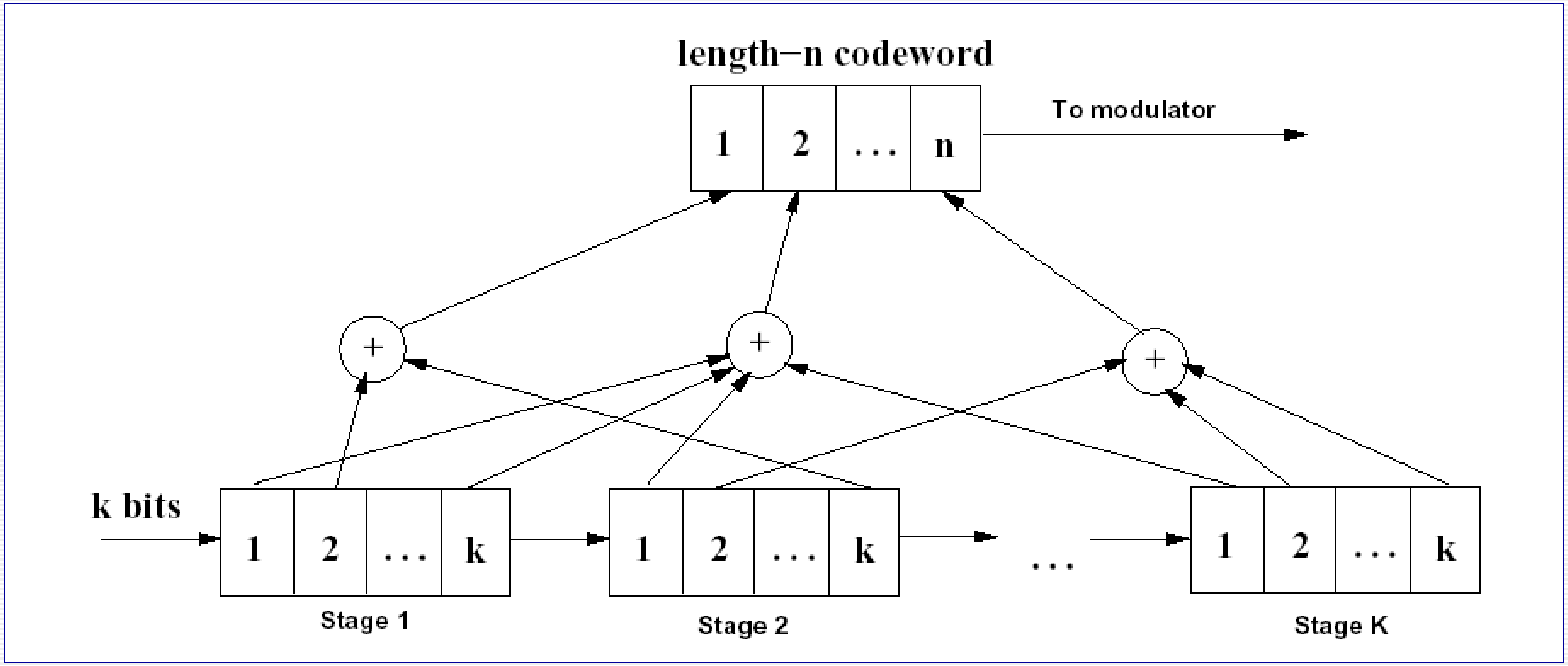
Common Linear Block Codes (2)

- Binary Block Codes
  - BCH (n, k) code
    - Cyclic code
    - Outperform all other block codes with the same n, k at moderate and high SNRs
- Nonbinary Block Codes
  - Reed Solomon code
    - Similar to the binary codes in that it has K information symbols mapped into codeword of length N
    - Each symbol of a codeword is not binary but is chosen from a nonbinary alphabet of size q

Convolutional Code

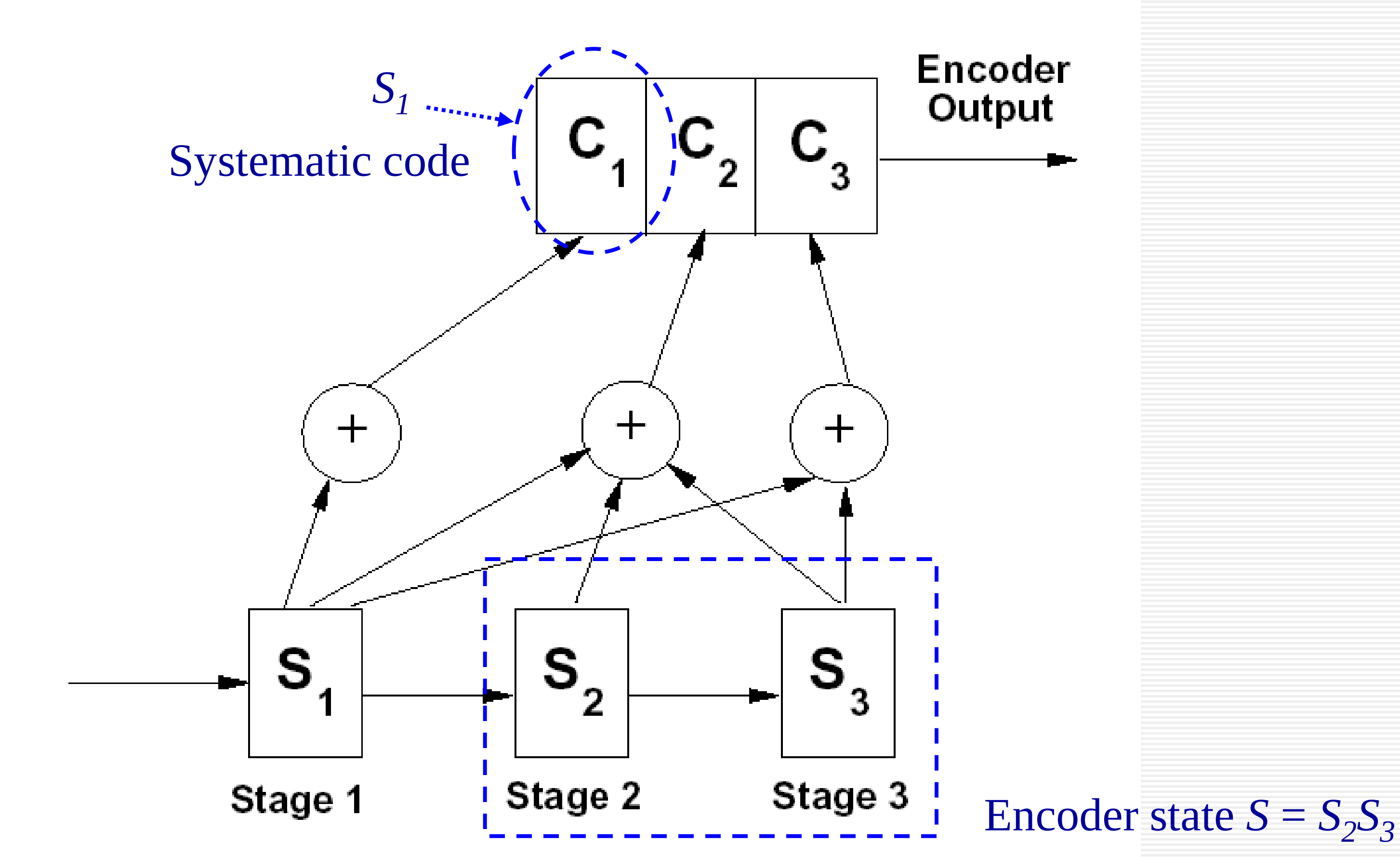
Convolutional Encoder

- The encoder generates a codeword of length *n* for *k*-bit input sequence
  - a shift register: *K* stages with *k* bits per stage (*k*-bits shift at a time)
  - *n* binary addition operator
  - Constraint length: *kK* bits

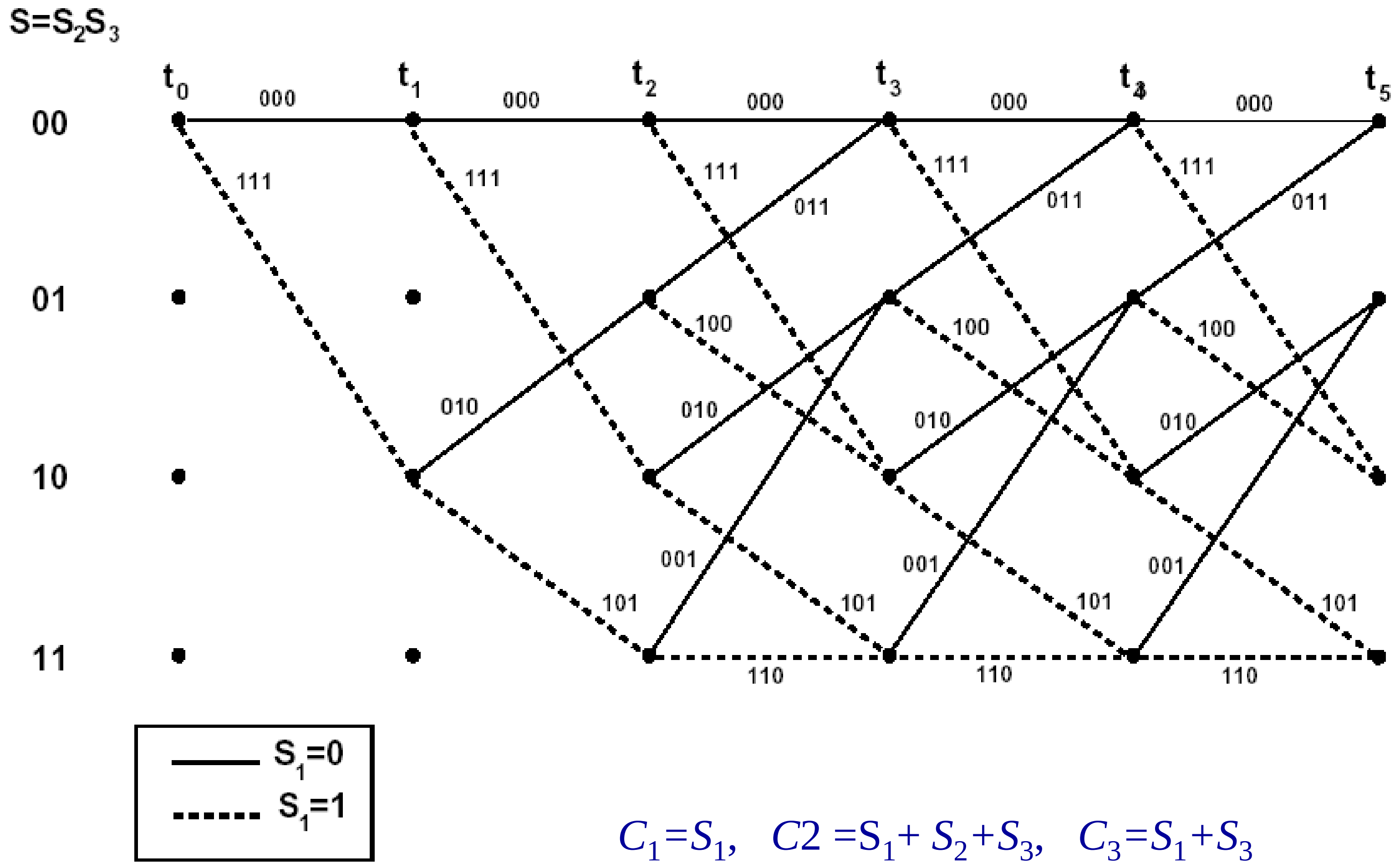


Trellis Diagram (1)

- the most common characterization of a convolutional code
- Convolutional encoder example (n=3, k=1, K=3)



Trellis Diagram (2)



$C_1 = S_1$, $C2 = S_1 + S_2 + S_3$, $C_3 = S_1 + S_3$

Maximum Likelihood Decoding (1)

- For a received sequence **R**, the decoder decides that coded symbol sequence **C*** was transmitted if

$$p\left(\mathbf{R} \mid \mathbf{C}^*\right) \geq p\left(\mathbf{R} \mid \mathbf{C}\right) \quad \forall \mathbf{C}$$

- For an AWGN channel, which noise affects each symbol independently, and for a convolutional code of rate 1/*n* and a path of length *L* through the trellis

$$p\left(\mathbf{R} \mid \mathbf{C}\right) = \prod_{i=0}^{L-1} p\left(R_i \mid C_i\right) = \prod_{i=0}^{L-1} \prod_{j=1}^{n} p\left(R_{ij} \mid C_{ij}\right)$$

$$\log p\left(\mathbf{R} \mid \mathbf{C}\right) = \sum_{i=0}^{L-1} \log p\left(R_i \mid C_i\right) = \sum_{i=0}^{L-1} \sum_{j=1}^{n} \log p\left(R_{ij} \mid C_{ij}\right)$$

- Branch metric: $B_i = \sum_{j=1}^{n} \log p\left(R_{ij} \mid C_{ij}\right)$

Maximum Likelihood Decoding (2)

- HDD
  - If **R** and **C** are *N* symbols long and differ in *d* places, and *p* is a symbol error probability in demodulation

$$p\left(\mathbf{R}\,\middle|\,\mathbf{C}\right) = p^{d}(1-p)^{N-d}$$

$$\log p\left(\mathbf{R}\,\middle|\,\mathbf{C}\right) = -d\log\frac{1-p}{p} + N\log(1-p)$$

  - Note that $p < 0.5$.
  - When *d* is minimized, $p(\mathbf{R}|\mathbf{C})$ is maximized.
  - The coded sequence C with minimum Hamming distance to the received sequence R corresponds to the maximum likelihood decoding.

Maximum Likelihood Decoding (3)

- SDD
  - For example, if the C_{ij} is sent via BPSK over an AWGN channel with a 1 mapped to $\sqrt{E_c}$ and a 0 mapped to $-\sqrt{E_c}$,

$$R_{ij} = \sqrt{E_c}\left(2C_{ij} - 1\right) + n_{ij}$$

    - n_{ij} is a Gaussian noise with mean zero and variance σ^2

$$p\left(R_{ij} \middle| C_{ij}\right) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{\left(R_{ij} - \sqrt{E_c}\left(2C_{ij} - 1\right)\right)^2}{2\sigma^2} \right]$$

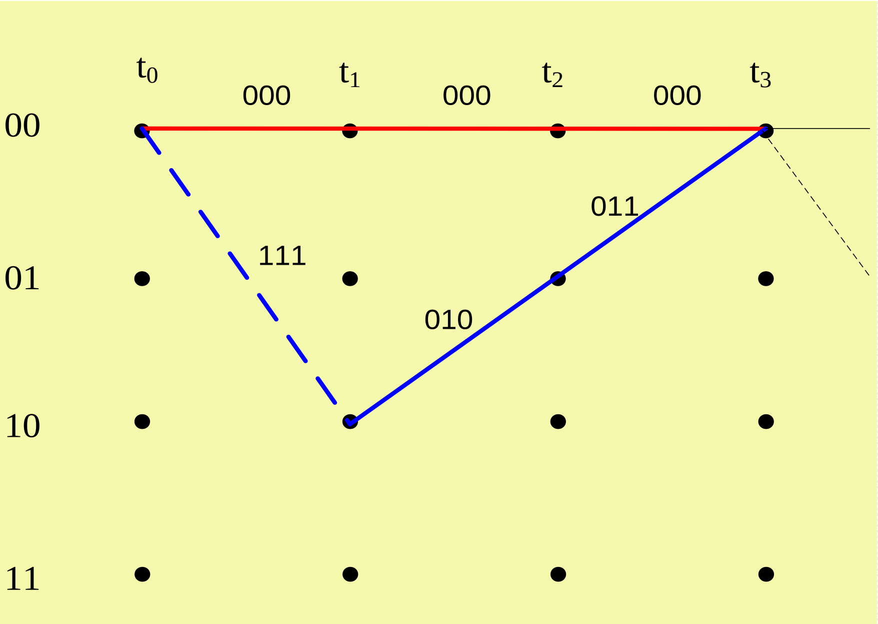
  - The equivalent branch metric obtains the same maximum likelihood output

$$\mu_i = \sum_{j=1}^{n} R_{ij}\left(2C_{ij} - 1\right)$$

Maximum Likelihood Decoding (4)

Example

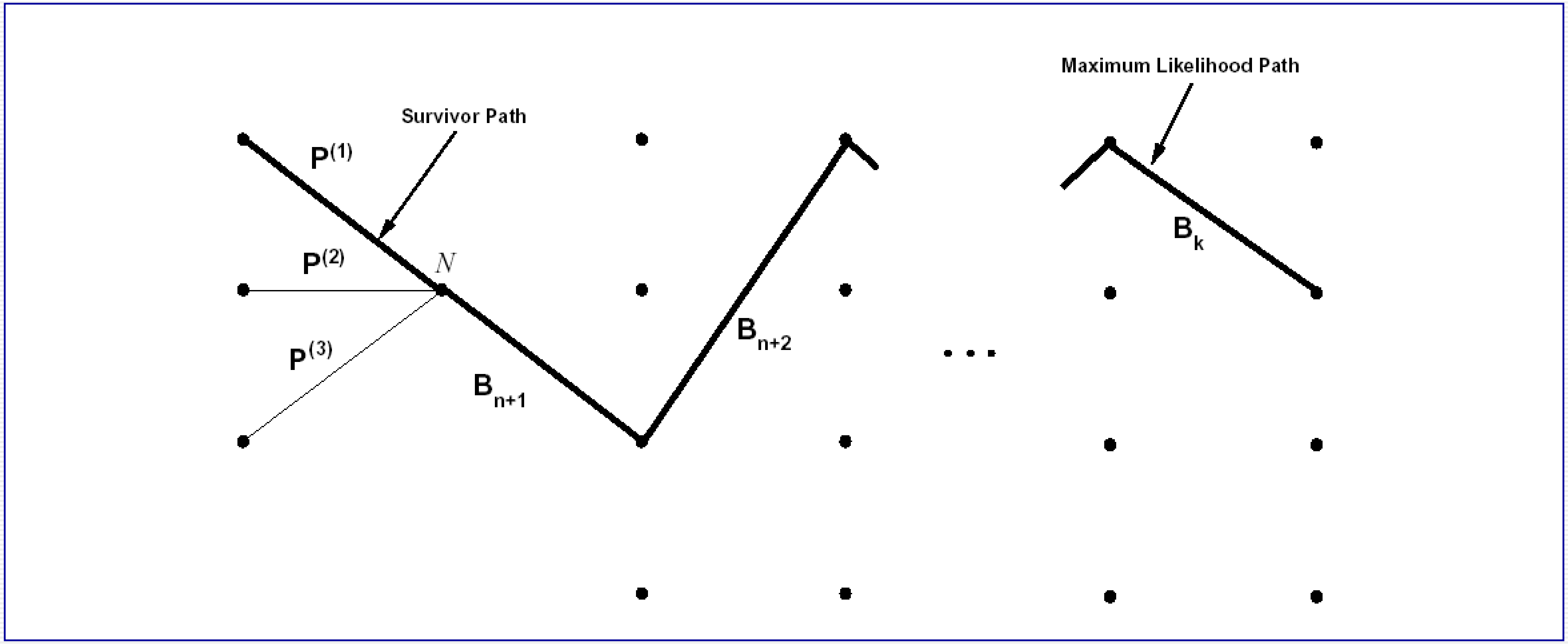
($\mathbf{C}_0$=000000000, $\mathbf{C}_1$=111010011)



- HDD
  - **R**=100110111
  - $M_0 = \sum_{i=0}^{2}\sum_{j=1}^{3} \log P\left(R_{ij} \middle| C_{ij}\right)$
    $= 6\log p + 3\log(1-p)$
  - $M_1 = 4\log p + 5\log(1-p)$
- SDD
  - R=(0.8, -0.35, -0.15, 1.35, 1.22, -0.62, 0.87. 1.08, 0.91)
  - $M_0 = \sum_{i=0}^{2} \mu_i = \sum_{i=0}^{2}\sum_{j=1}^{3} R_{ij}(2C_{ij} - 1) = \sum_{i=0}^{2}\sum_{j=1}^{3} -R_{ij} = -5.11$
  - $M_1 = 1.91$

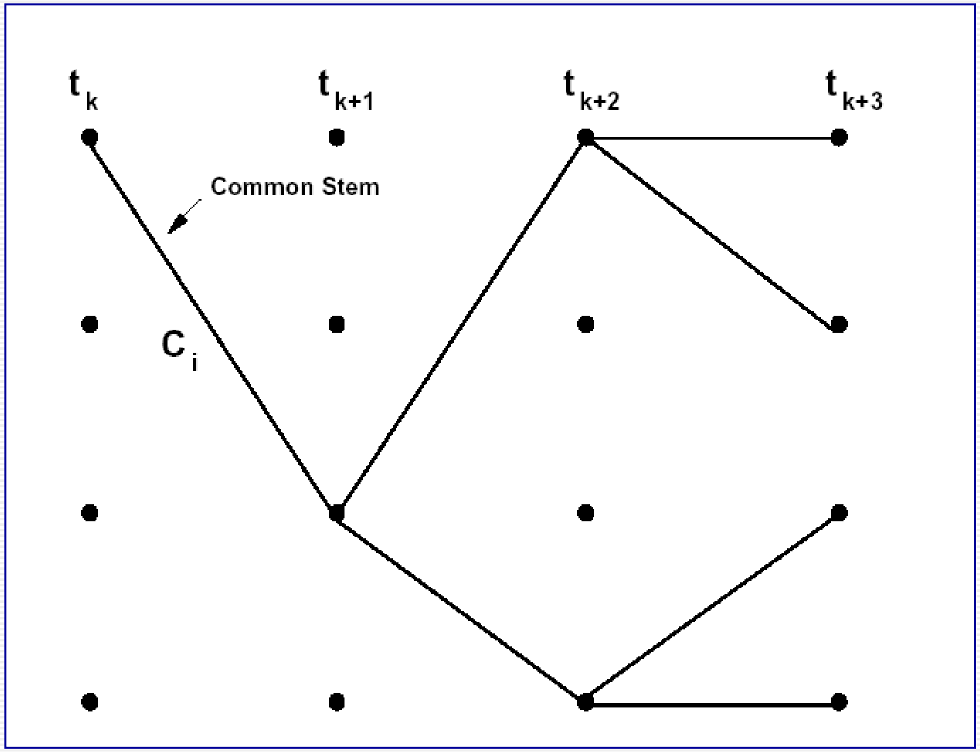
Viterbi Algorithm (1)

- discards all paths entering a given node N except the *supervisor path*, which is the path with the largest partial path metric up to that node



Viterbi Algorithm (2)

- The decoder can ouput a codeword symbol $\mathbf{C}_i$ associated with the common stem when all of the supervisor paths at a stage can be traced back to the common stem.
- Modification for avoiding a random decoding delay
  - the most likely branch *n* stages back is decided upon based on the partial path metrics up to a stage

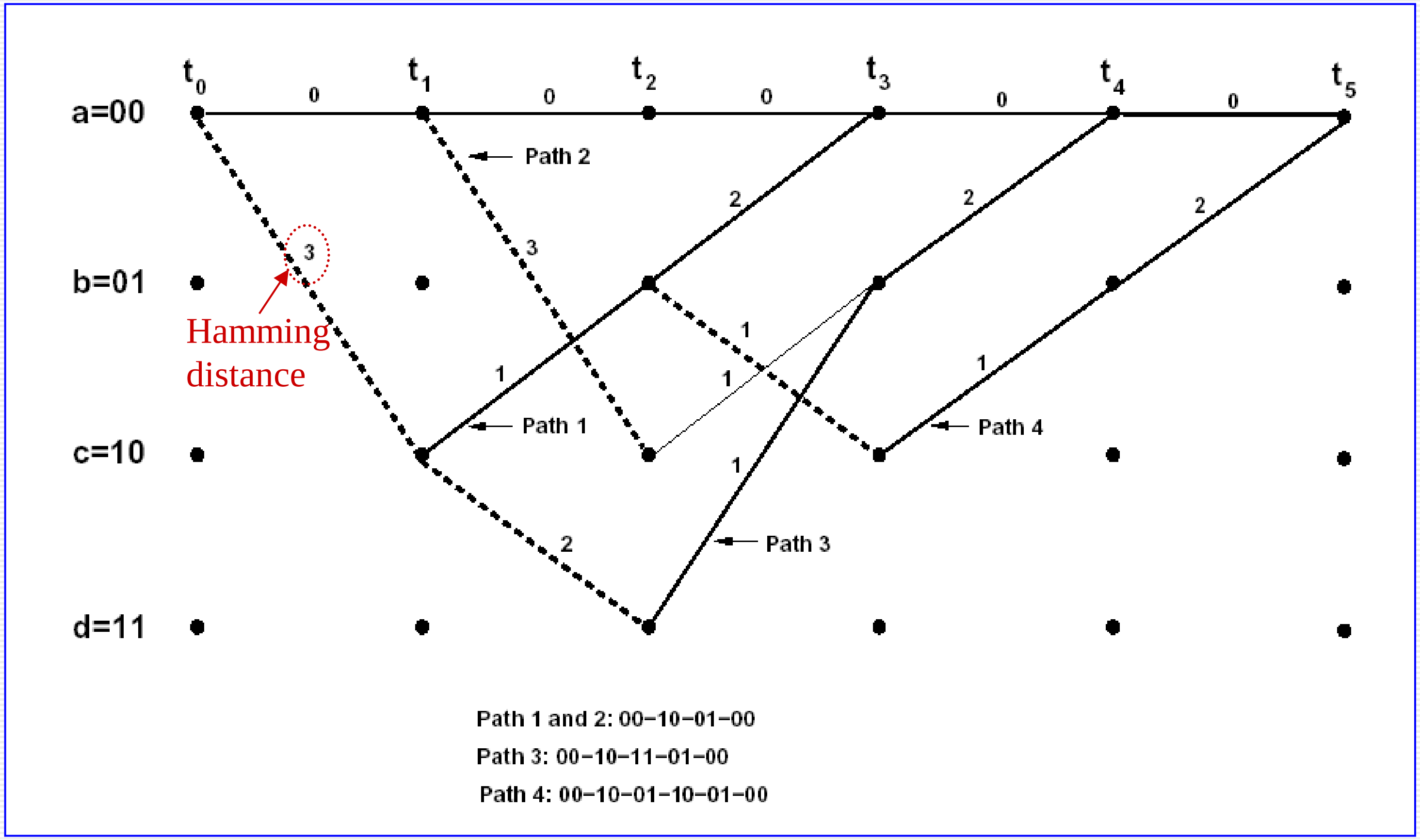


Distance Property (1)

- Minimum free distance, d_f
  - is defined as the minimum Hamming distance of all paths through the trellis to all-zero path
- Error correction capability of a convolutional code
  - is obtained in the same manner as for block codes
  - The code can correct t errors, where $t = \lfloor d_f / 2 \rfloor$
- To find the minimum free distance path,
  - We must consider all paths that diverge from the all-zero state and then remerge with the state.

Distance Property (2)

- Example



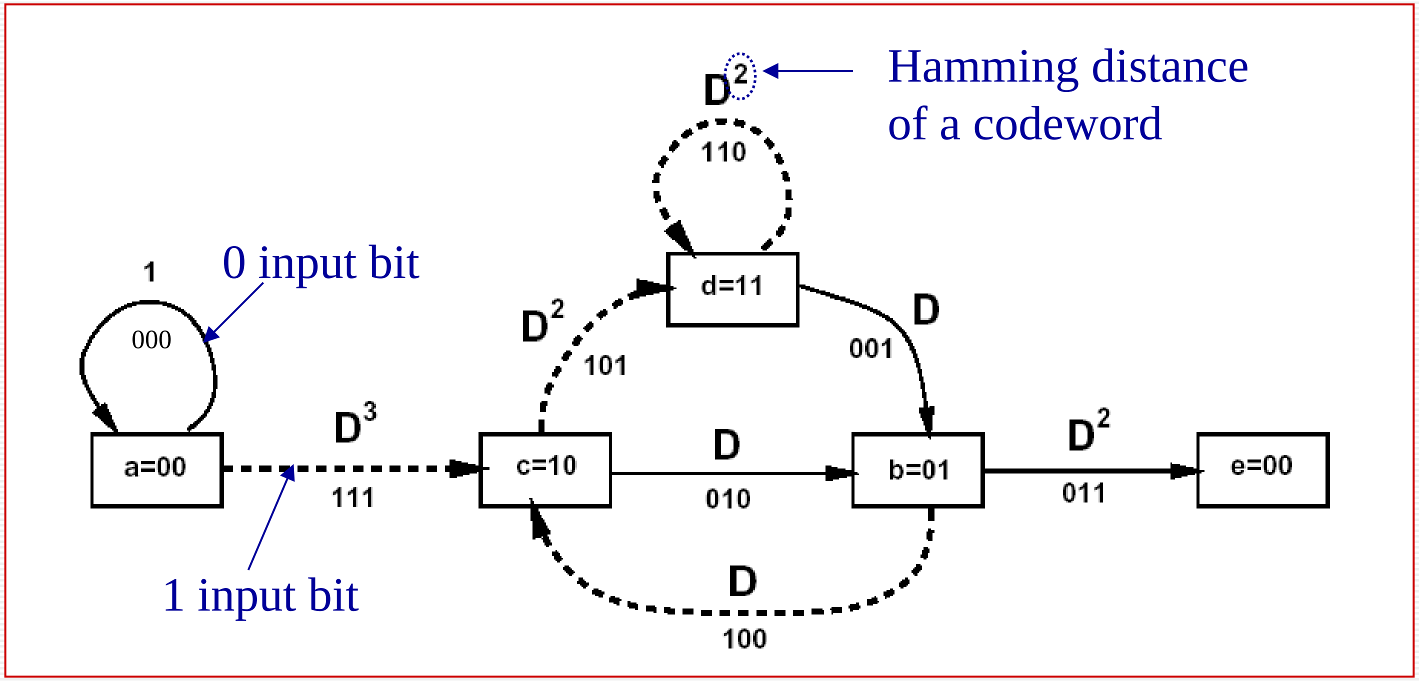
Path distance: 6 (Path1, Path2), 8 (Path3), 8 (Path4)

Input bit sequence: 10000 (Path1), 01000 (Path2), 11000 (Path3), 10100 (Path4)

Minimum free distance: 6; $t = 3$

State Diagram

- The state diagram represents possible transitions from the all-zero state to the all-zero state



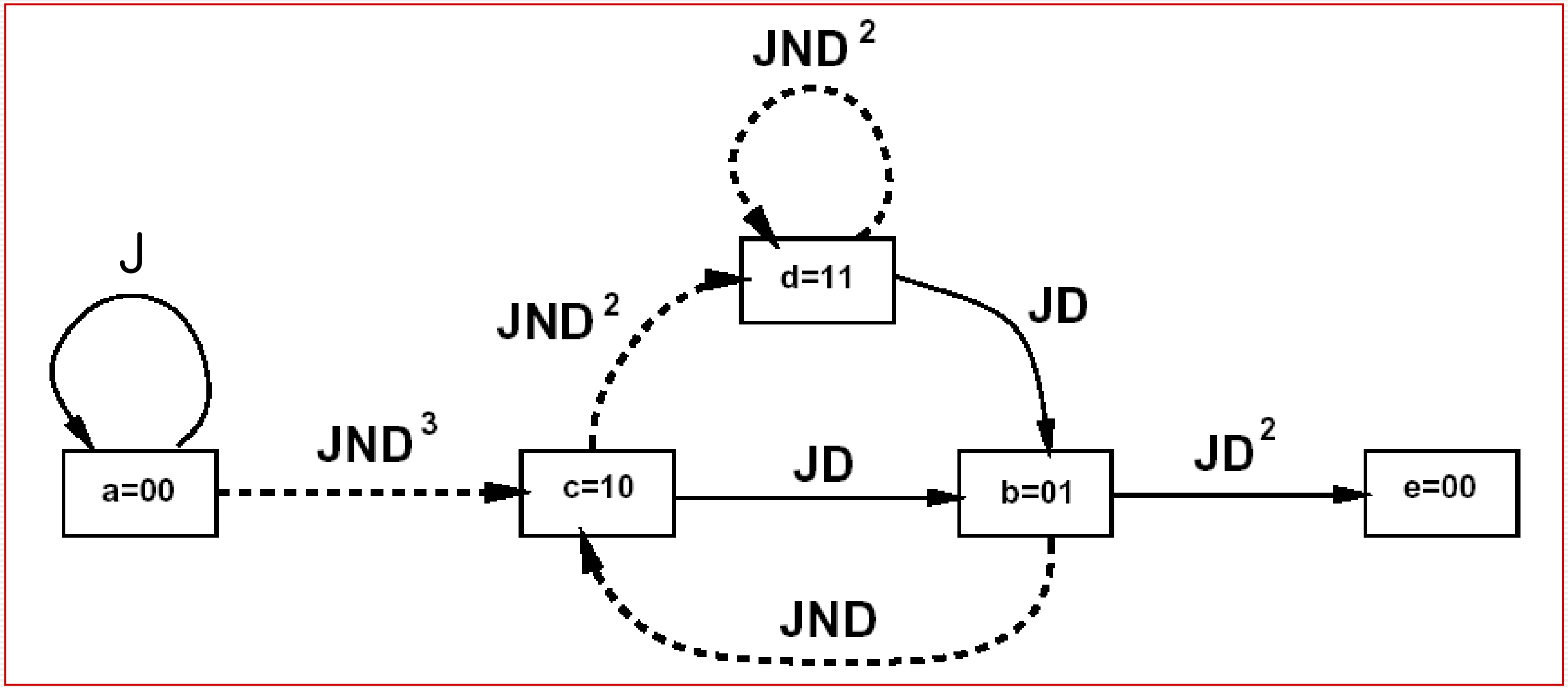
Transfer Function

- The transfer function $T(D)$ describes the paths from state a to state e
  - $X_c = D^3 X_a + D X_b, \quad X_b = D X_c + D X_d, \quad X_d = D^2 X_c + D^2 X_d, \quad X_e = D^2 X_b$
  - $$T(D) = \frac{X_e}{X_a} = \frac{D^6}{1-2D^2} = D^6 + 2D^8 + 4D^{10} + \cdots$$

    a path with minimum distance 6

    4 paths of distance 10
  - a convenient shorthand for enumerating the number and corresponding Hamming distance of all paths that diverge and remerge with the all-zero path
- Extended state diagram
  - *J*: is introduced to every branch (its exponent is the number of branches in any path from state *a* to state *e*)
  - *N*: is introduced on all branch transitions associated with a 1 input bit

Extended state diagram and Transfer Function



$$X_c = JND^3 X_a + JNDX_b$$

$$X_b = JDX_c + JDX_d$$

$$X_d = JND^2 X_c + JND^2 X_d$$

$$X_e = JD^2 X_b$$

$$T(D,N,J) = \frac{J^3 ND^6}{1 - JND^2(1+J)}$$

$$= J^3 ND^6 + J^4 N^2 D^8 + J^5 N^2 D^8 + J^5 N^3 D^{10} + \cdots$$

Distance 6, length 3, one bit error

Error Probability for Convolutional Code (1)

- The error probability can be obtained by first assuming that the all-zero sequence is transmitted and then determining the probability that the decoder decides as a different sequence.
- SDD
  - For an AWGN channel using coherent BPSK modulation with energy $E_c = R_c E_b$, the probability of mistaking the all-zero sequence with a sequence Hamming distance d away is

$$P_2(d) = Q\left(\sqrt{\frac{2E_c}{N_0}d}\right) = Q\left(\sqrt{2\gamma_b R_c d}\right)$$

Error Probability for Convolutional Code (2)

- SDD
  - By the union bound: $P_e \le \sum_{d=d_f}^{\infty} a_d \, Q\left(\sqrt{2\gamma_b R_c d}\right)$

    where a_d is the number of paths with distance d
  - Since $Q\left(\sqrt{2\gamma_b R_c d}\right) \le e^{-\gamma_b R_c d}$, $P_e \le T(D)\Big|_{D=e^{-\gamma_b R_c}}$ $\qquad T(D) = \sum_{d=d_f}^{\infty} a_d \, D^d$
  - The bit error probability
    - When $T(D,N) = \sum_{d=d_f}^{\infty} a_d D^d N^{f(d)}$, $\qquad P_b \le \sum_{d=d_f}^{\infty} a_d \, f(d) Q\left(\sqrt{2\gamma_b R_c d}\right)$

      where $f(d)$ denotes the number of bit errors with a path of distance d from the all-zero path
    - Therefore, $P_b \le dT(D,N)/dN\Big|_{N=1, D=e^{-\gamma_b R_c}}$

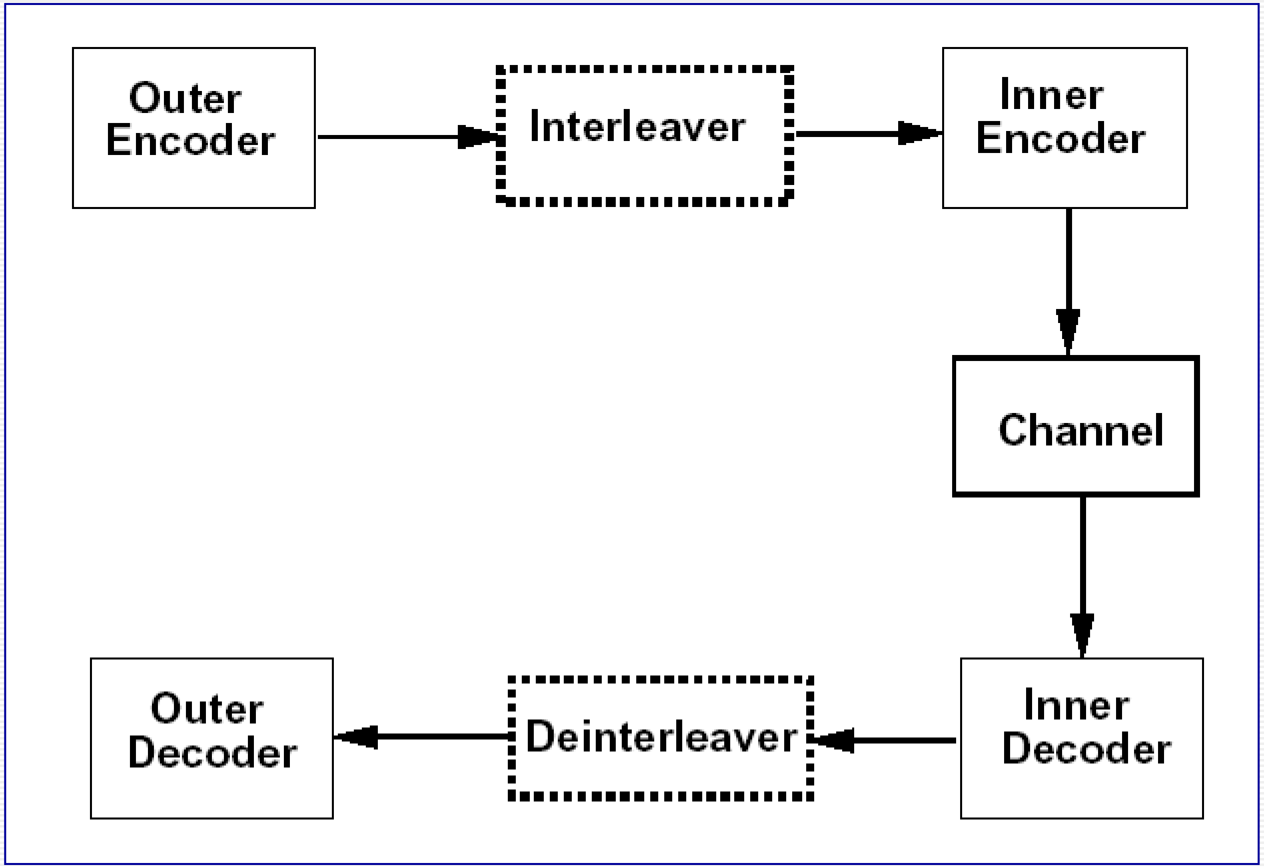
Some Other Codes

- Concatenated Codes
- Turbo Codes
- Low-Density Parity Check Codes
- Coded Modulation

Concatenated Code (1)

- Two levels coding
  - An inner code is designed to remove most of the errors introduced by the channel
  - An outer code is a less powerful code that reduces an error probability when the received coded bits have a relatively low probability of error (since most errors are corrected by the inner code)
- Effective in correcting error bursts
  - At low SNRs, Viterbi decoding of a convolutional code tends to have burst errors
- Common in wireless channels
  - Inner code: convolutional code
  - Outer code: Reed Solomon code
  - The inner and outer codes are separated by an interleaver
- Very low error probability with less complexity than a single code with the same error probability performance

Concatenated Code (2)

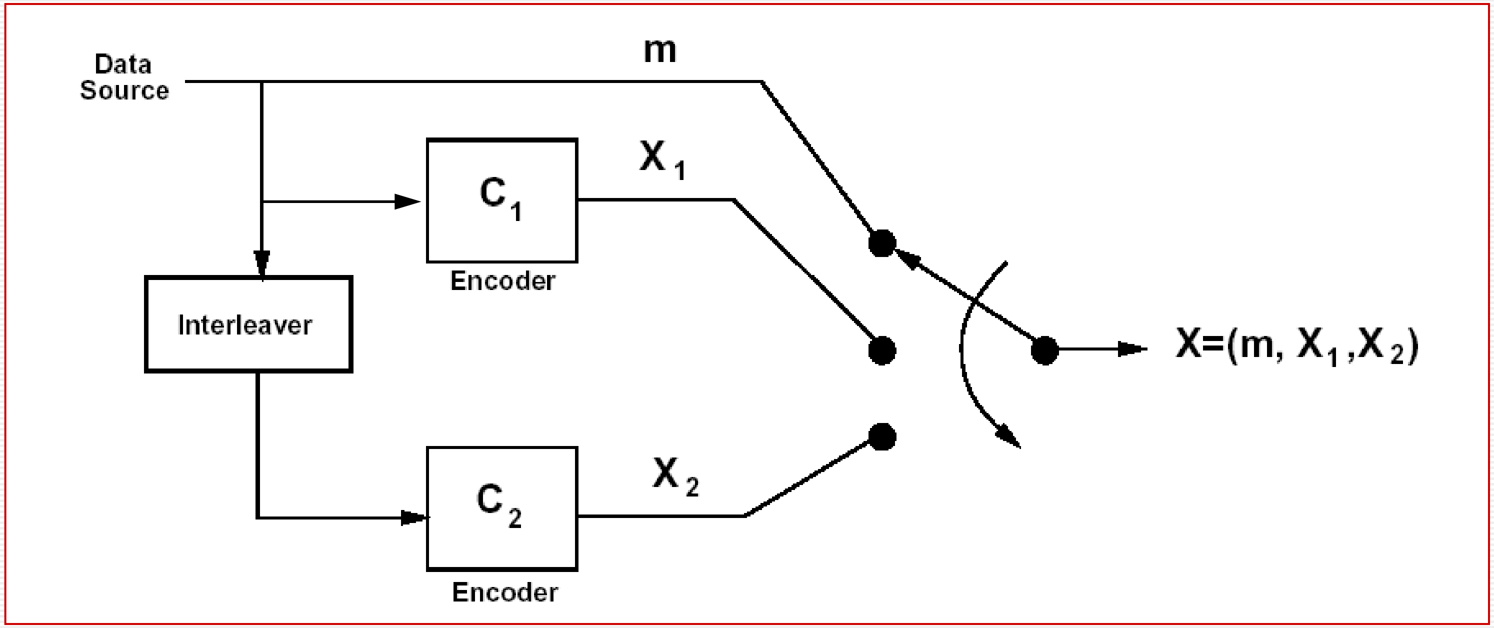


Turbo Codes (1)

- Turbo codes was introduced in 1993 in a landmark paper by Berrou, Glavieux, and Thitimajshima.
- Powerful codes that achieve performance close to the Shannon limit. (within a fraction of a decibel of a Shannon capacity on AWGN channel)
- Two key components
  - Parallel concatenated encoding
  - Iterative, graph-based decoding

Turbo Codes (2)

- Parallel concatenated (turbo) encoder
  - Two parallel convolutional codes separated by an interleaver
  - A systematic code: the *m* information bits are transmitted as a part of the codeword

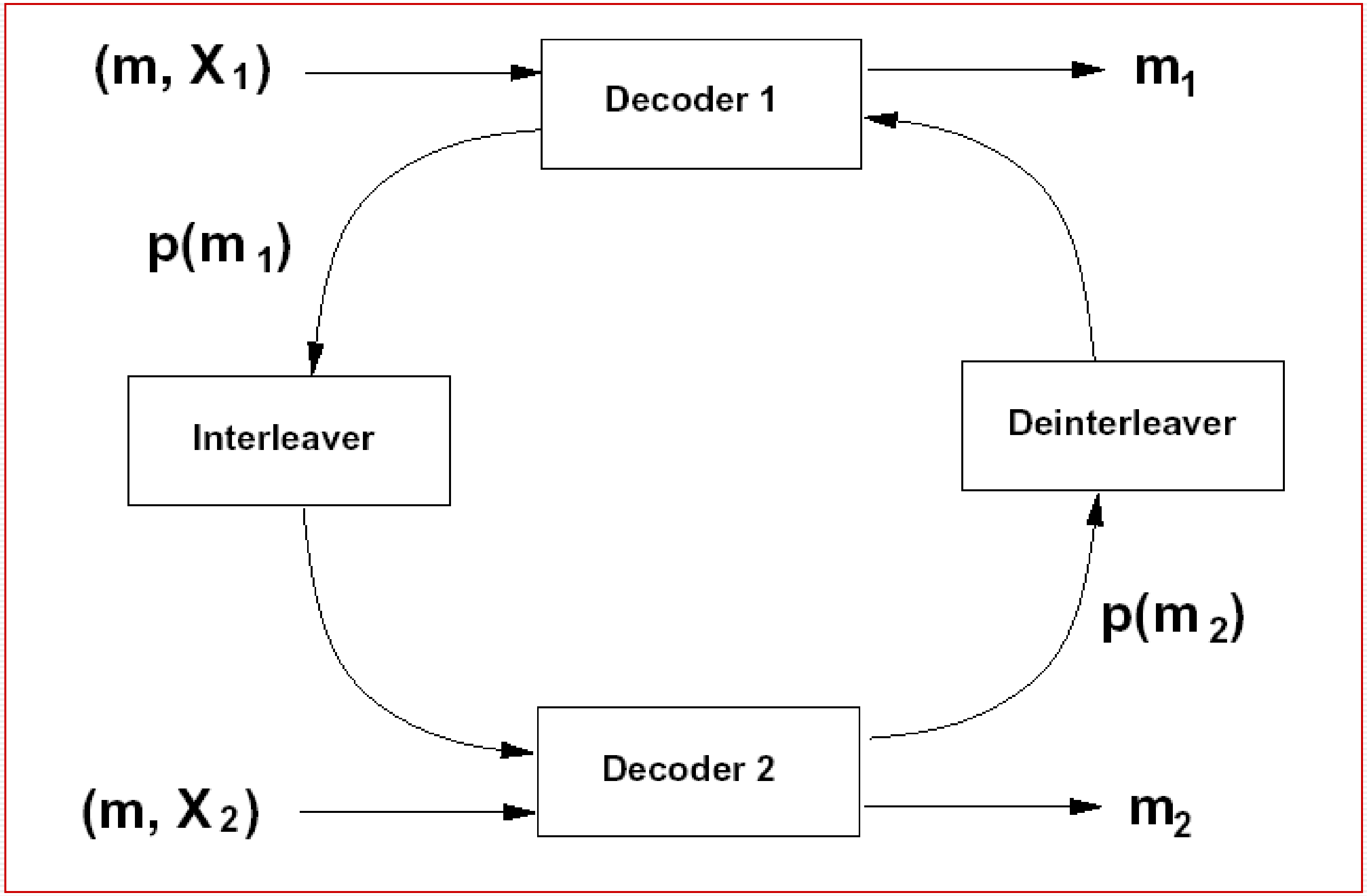


Turbo Codes (3)

- Iterative decoder
  - Decoder 1 generates a soft decision in the form of a probability measure $p(m_1)$ on the information bits based on the received codeword (m, X_1).
  - The probability measure is either a maximum posteriori probability or soft output Viterbi algorithm (which attaches a reliability indicator to the VA hard decision outputs).
  - operates an iterative manner with the two decoders alternately updating their probability measures.
  - Ideally, $m = m_1 = m_2$
  - The stopping condition for turbo decoding is not well-defined: there are many case in which the decoding dose not converge.

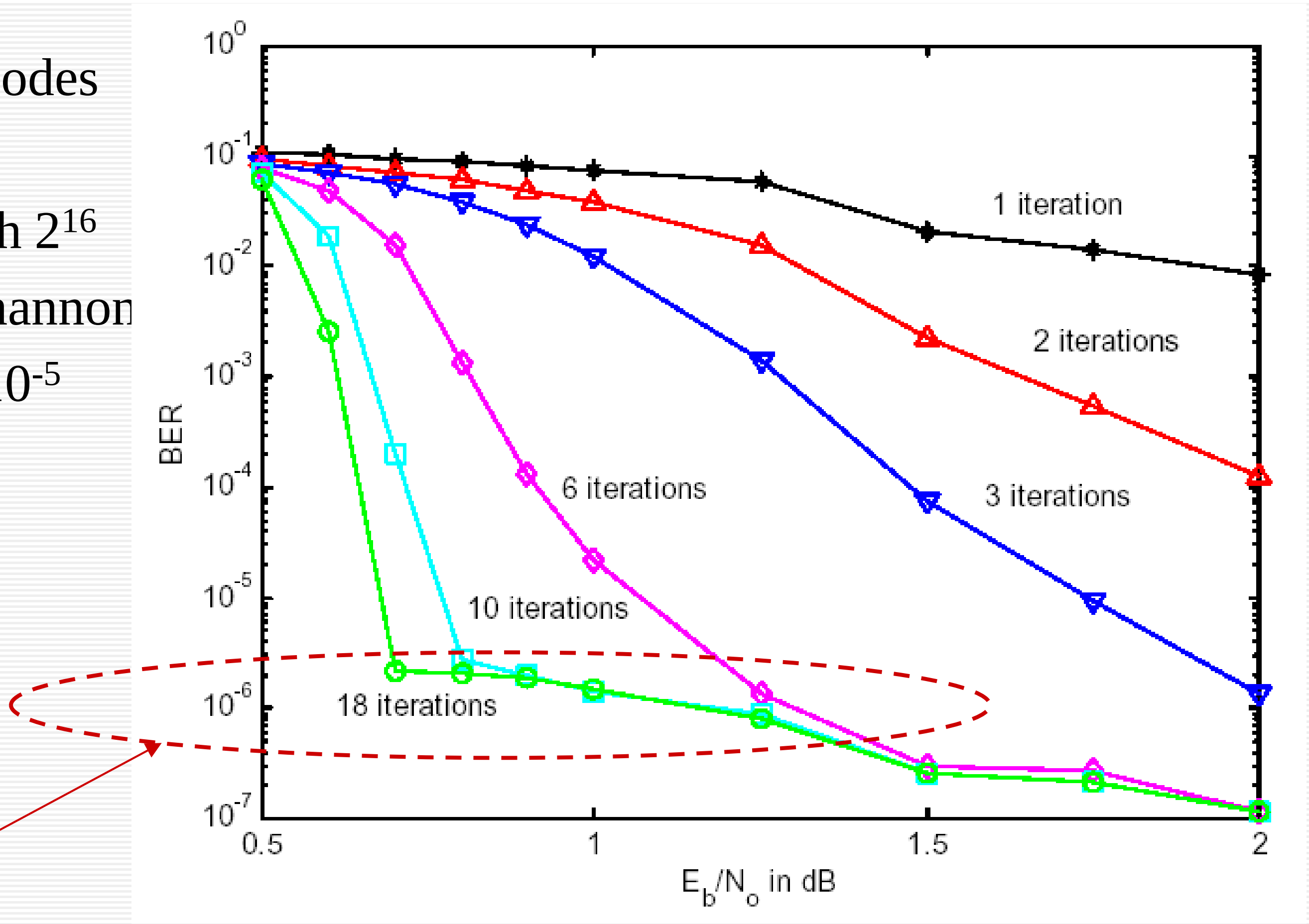
Turbo Codes (4)

Turbo Decoder



Turbo Codes (5)

- Simulation
  - Convolutional codes (rate 1/2, K=5)
  - Interleaver depth 2^{16}
  - 0.5 dB of the Shannon capacity at $P_b=10^{-5}$



Error floor

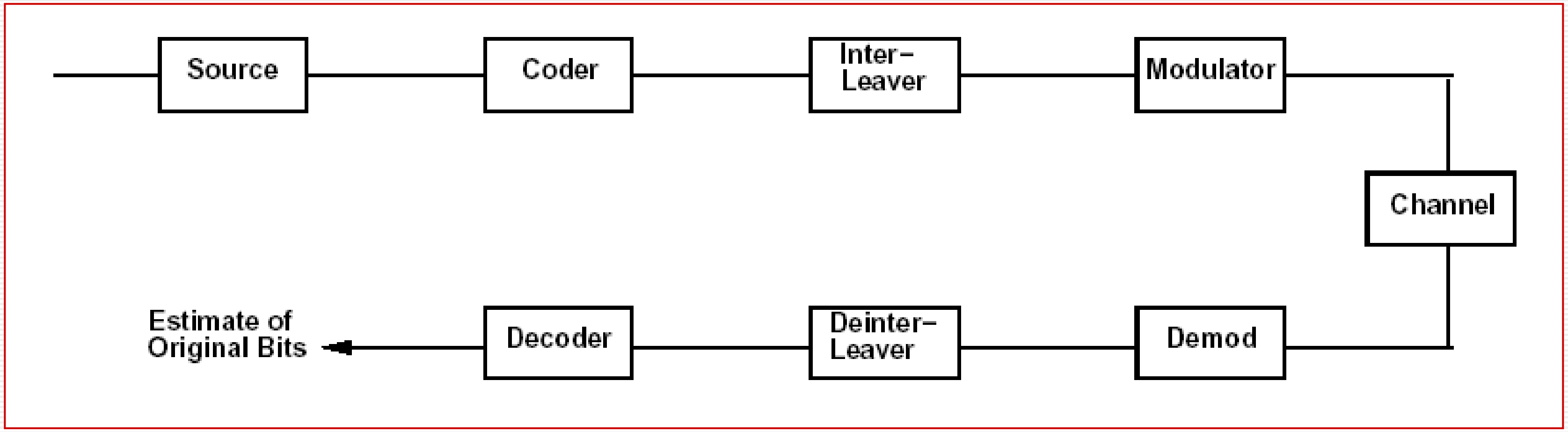
Low-Density Parity Check Code

- A (d_v, d_c) regular binary LDPC: a linear block code with a particular structure for the parity check matrix **H** with d_v 1s in each column and d_c 1s in each row.
- When the codeword length is long, LDPC codes achieve performance close to the Shannon limit
- LDPC codes have relatively high encoding complexity and low decoding complexity, whereas Turbo codes tends to have low encoding complexity and high decoding complexity.

Interleaving

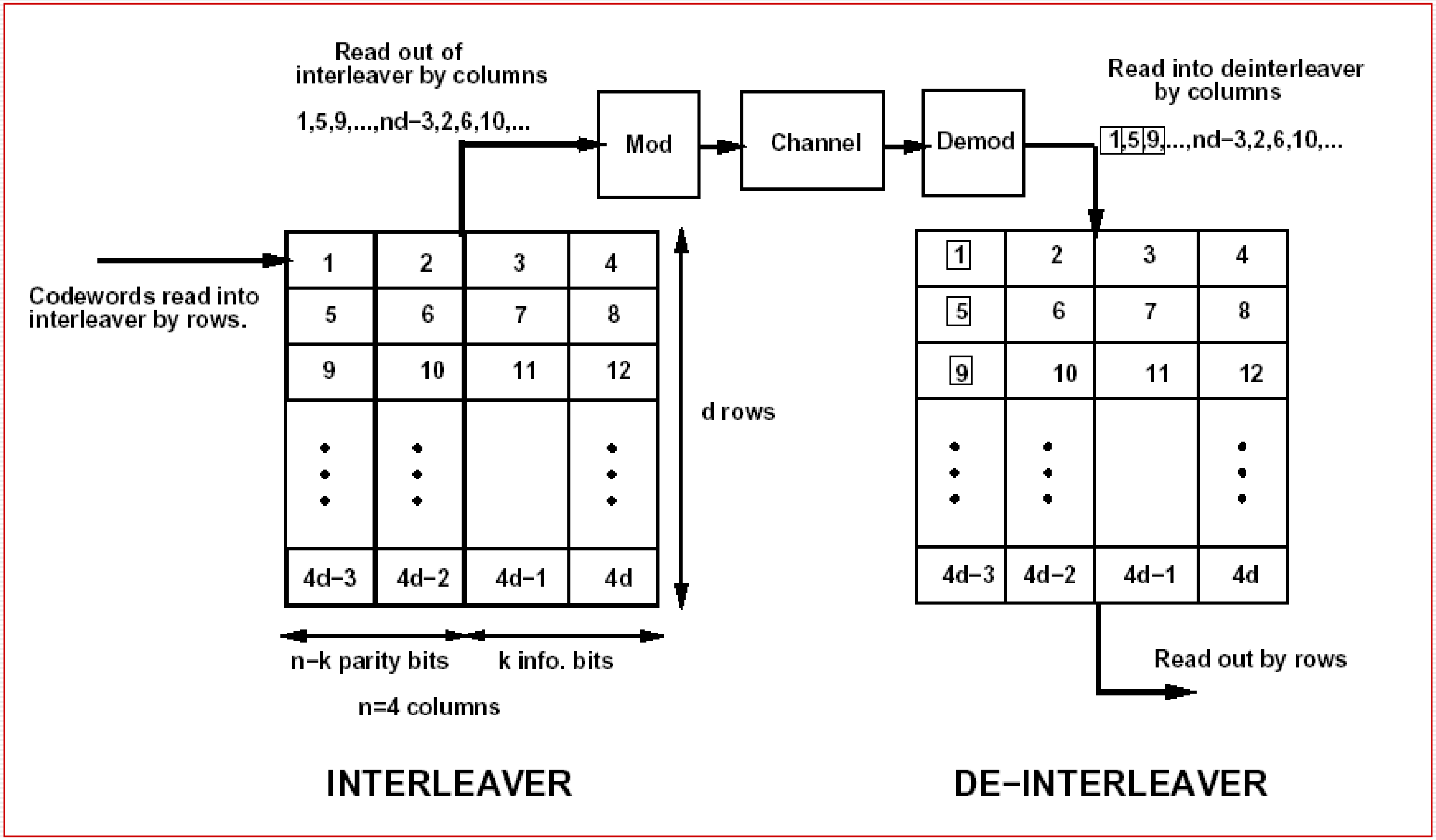
Coding with Interleaving for Fading Channels

- Codes designed for AWGN channels can exhibit worse performance in fading than an uncoded system
- To mitigate the effects of error bursts in fading channel, coding is typically combined with interleaving.
  - Interleaver: spreading out error bursts due to deep fades
  - Channel decoder: error correction over the spread error
- Slow fading channels require large interleaver



Block Coding with Interleaving (1)

an (n, k) block code



Block Coding with Interleaving (2)

- Code symbols in the same codeword are separated by *d*-1 other symbols
- Symbols in the same codeword experience approximately independent fading if $dT_s > T_c \approx 1/B_D$ (*deep interleaving*)
  - T_s: duration of a codeword symbol
  - T_c: channel coherence time
  - B_D: channel Doppler spread