Intro to coding and convolutional codes

Lecture 11

Vladimir Stojanović

6.973 Communication System Design – Spring 2006

Massachusetts Institute of Technology

802.11a Convolutional Encoder

- Rate 1/2 convolutional encoder
  - Punctured to obtain 2/3 and 3/4 rate
    - Omit some of the coded bits

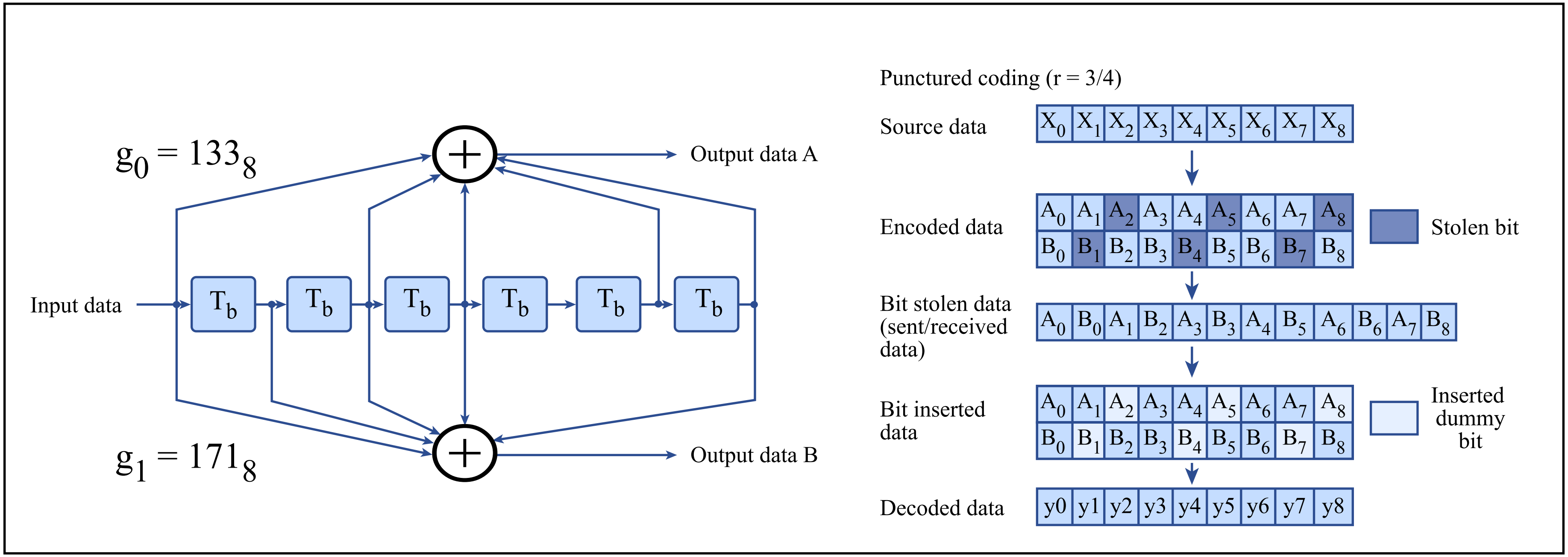


Figure by MIT OpenCourseWare.

- 64-state (constraint length K=7) code
- Viterbi algorithm applied in the decoder

What we'll cover today

- What are convolutional codes?
- How they help
- How to encode/decode them

Channel coding

- To enhance robustness, tie bits into sequences, then decide on sequences rather than individual received bits
- Encoder
  - Memory-less: translates incoming message m_k at time k into a symbol vector x_k (modulator later converts x_k to $x_k(t)$)
  - Sequential: map message bits into larger dimensionality symbols that can also depend on previous message bits through the state of the encoder
- Codewords
  - Finite (block code)
  - Semi-Infinite (tree/convolutional code)

$$m(D) = \sum_k m_k \cdot D^k$$

codeword $x(D) = \sum_k x_k \cdot D^k$

$$\{m(D)\} \xrightarrow{\text{code}} \{x(D)\}$$

- Example
  - Block code - Majority repetition binary code (0->-1-1-1, 1->+1+1+1)
    - ML decoder computes the majority polarity for the received signal
    - 1/3 bits per symbol with min.distance 2*sqrt(3)
  - Tree code – Transmit -1-1-1 if the bit has not changed, transmit +1+1+1 otherwise

Sequential encoder

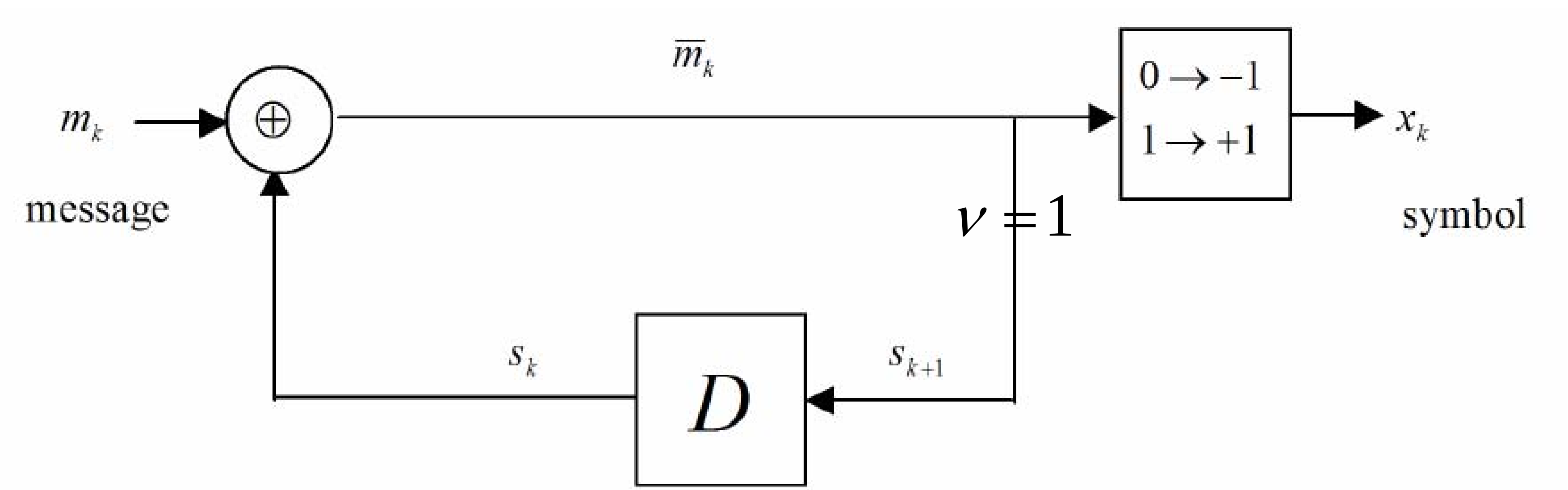
Image removed due to copyright restrictions.

- ν bits determine "state" s_k at time k
- There are 2^{ν} states
  - Encoding of bits into symbols can vary with encoder state
  - For each state encoder accepts b bits of input (m_k) and outputs a corresponding N-dimensional output vector (this is repeated once every symbol period T)
- Data rate of the encoder is $R \triangleq \frac{\log_2(M)}{T} = \frac{b}{T}$
- Block code if there is only one state (tree code otherwise)

Examples

- QAM is a block code
  - There is only one state in QAM $\nu = 0$
    - Let 1/T=2.4kHz and
    - For 4 QAM, R=2/T=2*2400=4800bps, $\bar{b} = 2/2 = 1$ bit/dimension
    - For 16 QAM, R=4/T=9600bps, $\bar{b} = 4/2 = 2$
    - For 256 QAM, R=8/T=19200bps, $\bar{b} = 8/2 = 4$

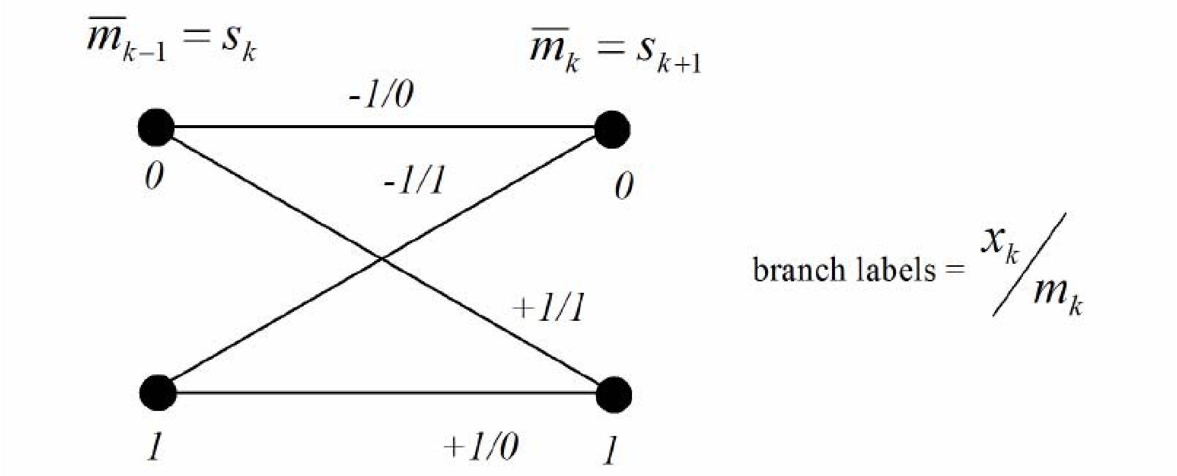
Example: Binary PAM differential encoder



- 2-states (corresponding to the possible values of the previous single-bit message)
- Example of a sequential encoder with
- Differential encoder encodes the difference modulo-M between successive message inputs to the sequential encoder
  - In binary case it only transmits a 1 on a change of a bit in a message

The Trellis

- Describes the progression of symbols within a code



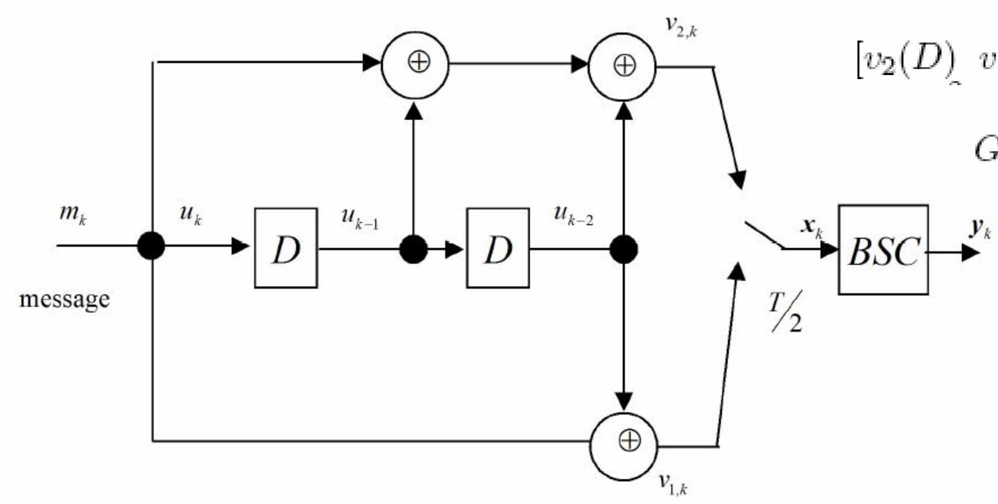
- Example – binary PAM differential encoder trellis
  - Two states in each time corresponding to the value of previously transmitted message
  - Time-invariant encoder only requires trellis representation at k and k+1
  - A trellis **branch** connects two states and corresponds to a possible input (always 2^b branches emanating from any state)
    - Each branch labeled with a channel symbol and corresponding input x_k/m_k

Example sequences for differential encoder

Image removed due to copyright restrictions.

- Semi-infinite series of branches starting from a known state
- To determine dmin of a code, need only find the two sequences through the trellis that have minimum separation
  - Would be the same for long period before and after the short period of divergence
  - For example, min distance between a sequence of no-changes and a single bit change would be $d_{min}^2=4=(+1-(-1))^2$
    - No gain when compared to uncoded PAM2, but that is o.k. in this case since differential encoder's purpose is to make the decoder insensitive to a sign ambiguity in transmission

A simple convolutional code

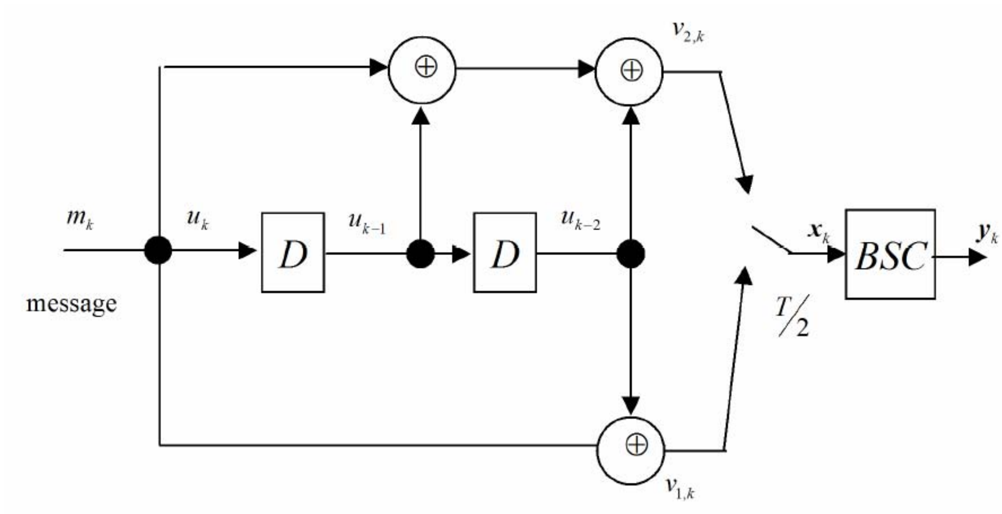


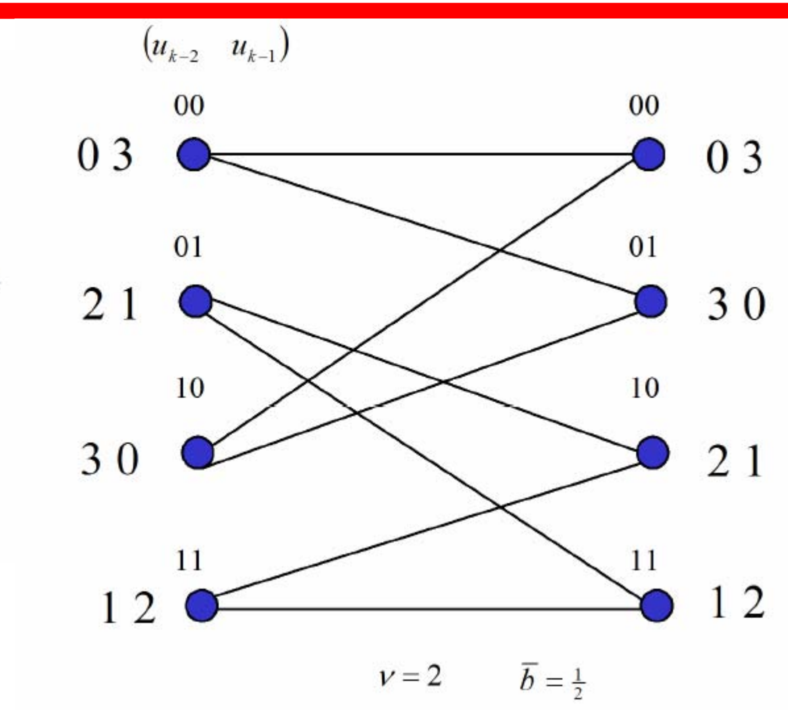
$$[v_2(D)\ \ v_1(D)] = u(D) \cdot [1 \oplus D \oplus D^2\ \ 1 \oplus D^2]$$

$$G(D) = [1 \oplus D \oplus D^2\ \ 1 \oplus D^2]$$

- G(D) – generator matrix
- Two output bits are successively transmitted through the channel (in this case binary symmetric channel with parameter p-probability of bit-error)
- Two states
- Number of dimensions is 2, thus bits/dimension=1/2

Trellis for convolutional code example





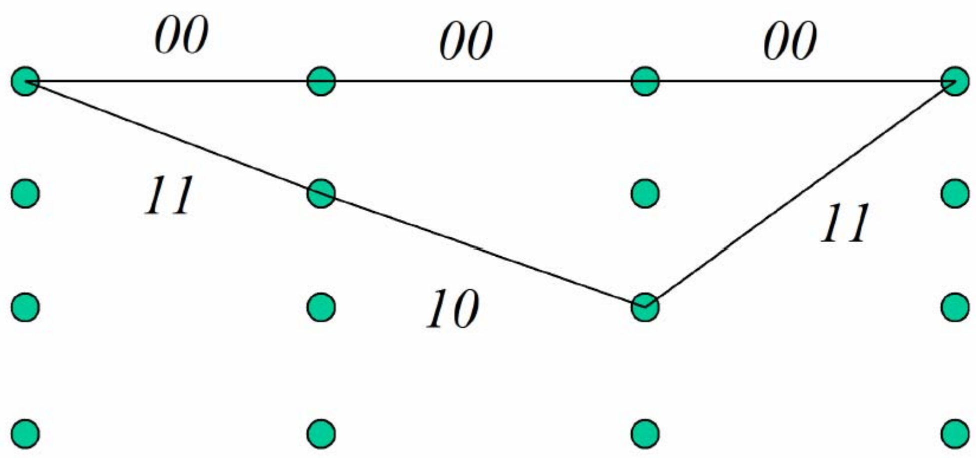
- The branches are not labeled – the convention is that the upper branch from each state corresponds to input bit of "0" while lower branch corresponds to input "1"
- The outputs transmitted for each state are listed in modulo-4 notation to the left of each state (leftmost – upper branch, rightmost – lower branch)

Distance between sequences

- The ML detector simply chooses the sequence of transitions through the trellis that **differs** least in the trellis-path bits [v2(D), v1(D)], from the received 2-dimensional sequence y(D)
- Term “differs” depends on the definition of “distance” between sequences
- Hamming distance – number of bit positions in which two sequences differ
- Euclidean distance – physical distance in which the received signal differs from the “expected” level for that bit/symbol

Hamming distance

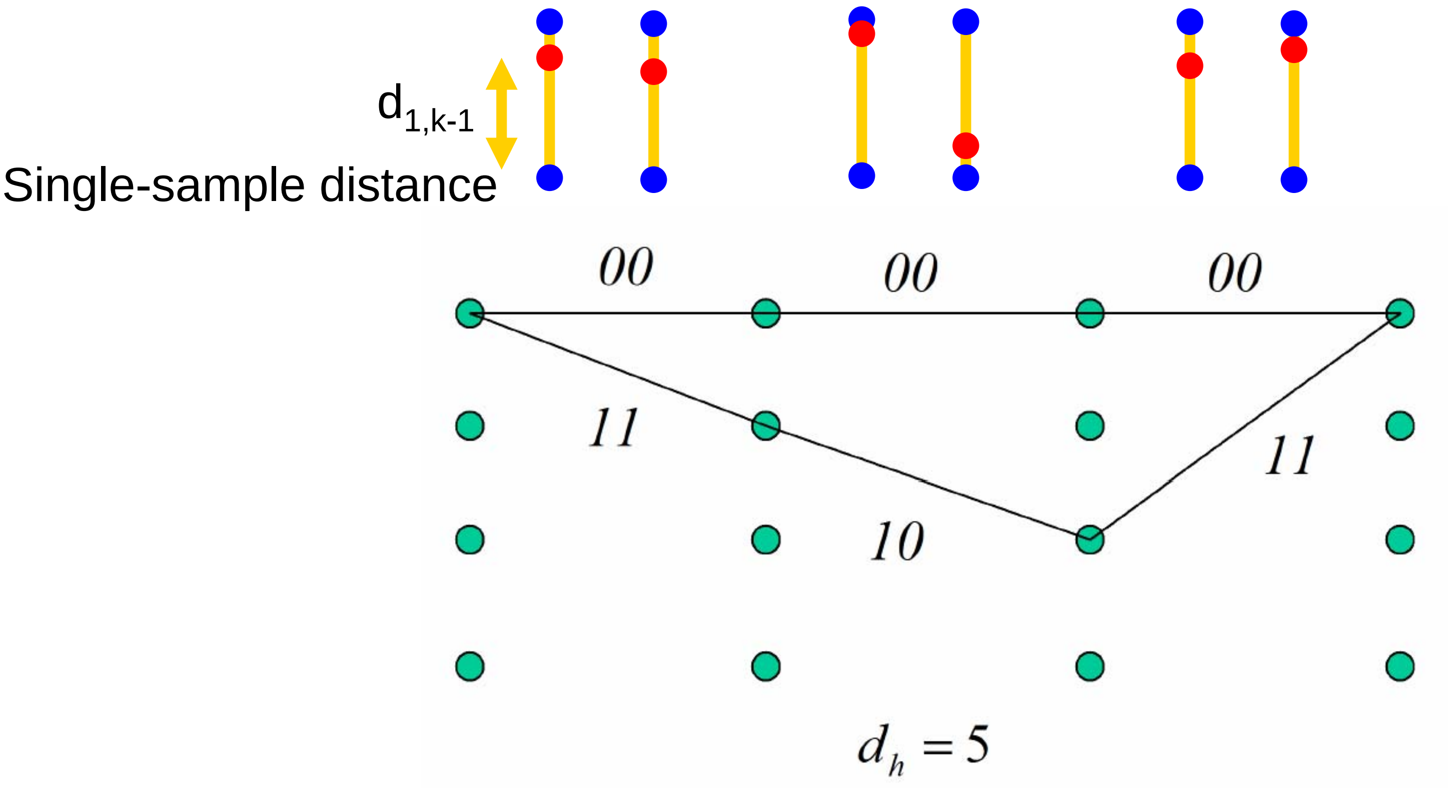
- Minimum Hamming distance illustration for the convolutional code example



$$d_h = 5$$

- Two sequences above differ in 5 bit positions
  - At least 3 bit errors must occur in the BSC before these two sequences could be confused
  - Thus the probability of detecting the erroneous sequence ~p^3, which for p<0.5 means convolutional code has improved the probability of error significantly (at the cost of half the bit-rate of uncoded transmission)
- Hamming weight is defined as Hamming distance between a codeword and the zero sequence $w_H(v(D))=d_H(v(d),0)$ (i.e. the number of “ones” in the codeword)

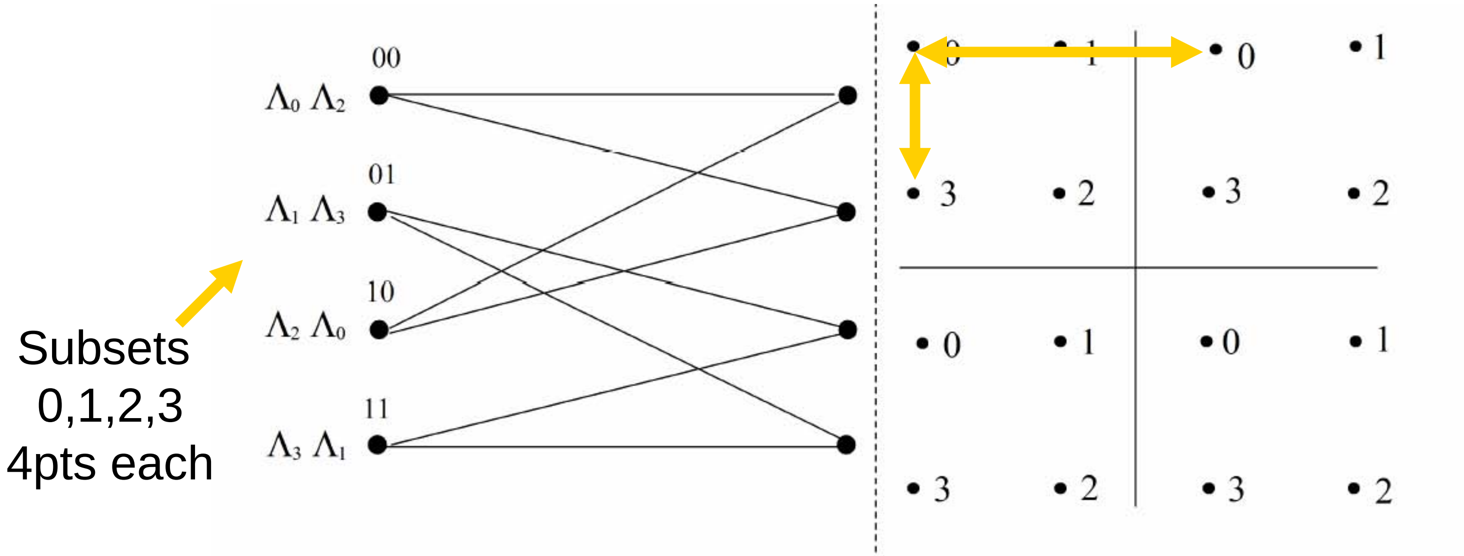
Euclidean distance



- $d_E^2 = d_{1,k-1}^2 + d_{2,k-1}^2 + d_{1,k}^2 + d_{2,k}^2 + d_{1,k+1}^2 + d_{2,k+1}^2$
- More suitable for noise characterization
  - For example AWGN channel

Trellis codes – Euclidean distance example

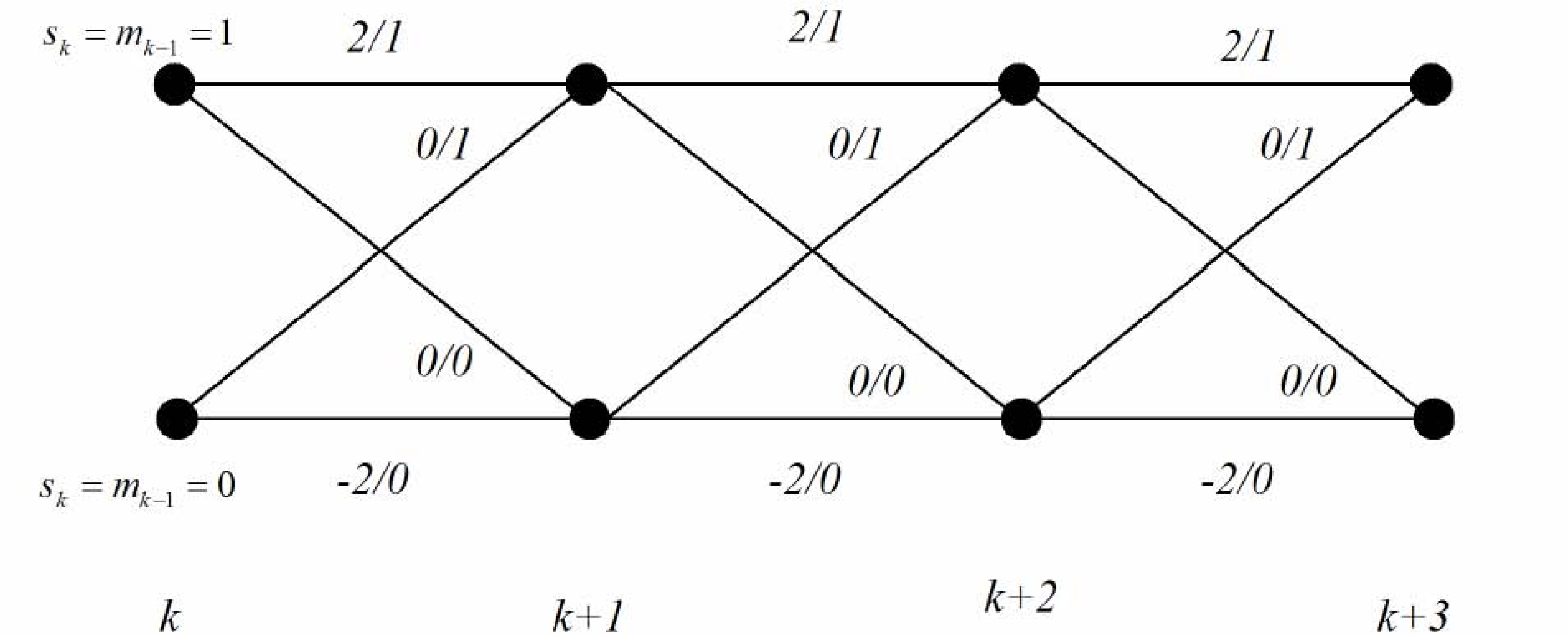
- Also use sequential encoding
  - Expand the constellation instead of reducing the data rate



- Example 4-state Ungerboeck trellis code
  - b=3 bits per symbol (redundant, extra pointsin the constellation over the minimum needed for transmission)
  - 16 points (for 16 QAM) is double the 8points needed for uncoded 8SQ QAM transmission)
  - Intra subset minimum distance increases by 3dB
  - Two input bits select one of 4 pts in a subset
  - One input bit enters the encoder to choose a branch

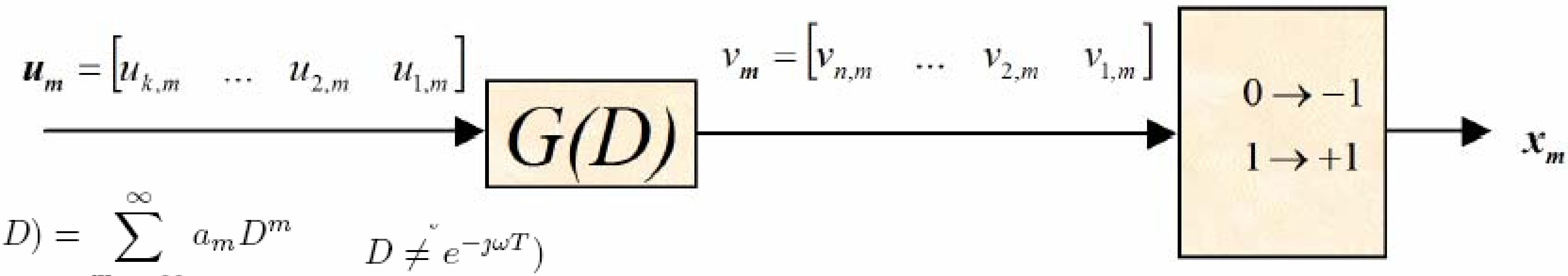
Channel as encoder

- Sometimes, can use channel with memory as a sequential encoder
  - Example is 1+D partial response channel
  - The closest two sequences are d_{min}^2=8 apart, not d_{min}^2=4 as with symbol-by-symbol detection (for PAM2)



  - Still need the sequence decoder (e.g. Viterbi decoder) to obtain the ML estimate of the received sequence

Generator and Parity matrices



$$a(D) = \sum_{m=-\infty}^{\infty} a_m D^m \qquad D \neq e^{-j\omega T}$$

$$F[D] \triangleq \left\{ a(D) \;\middle|\; a(D) = \sum_{m=0}^{\nu} a_m D^m \;,\; a_m \in F,\; \nu \geq 0,\; \nu < \infty,\; \nu \in \mathcal{Z} \right\}$$

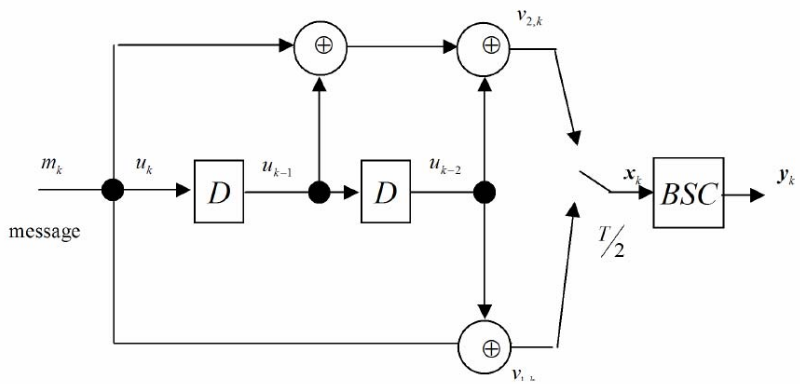
$$F_r(D) \triangleq \left\{ c(D) \;\middle|\; c(D) = \frac{a(D)}{b(D)} \;,\; a(D) \in F[D],\; b(D) \neq 0,\; b(D) \in F[D] \text{with } b_0 = 1 \right\}$$

$$C(G) \triangleq \{ v(D) \mid v(D) = u(D) \cdot G(D) \;,\; u(D) \in F_r(D) \}$$

$$C(G) = \{ v(D) \mid v(D) H^*(D) = 0 \} \quad v(D) H^*(D) = 0 \quad G(D) H^*(D) = 0.$$

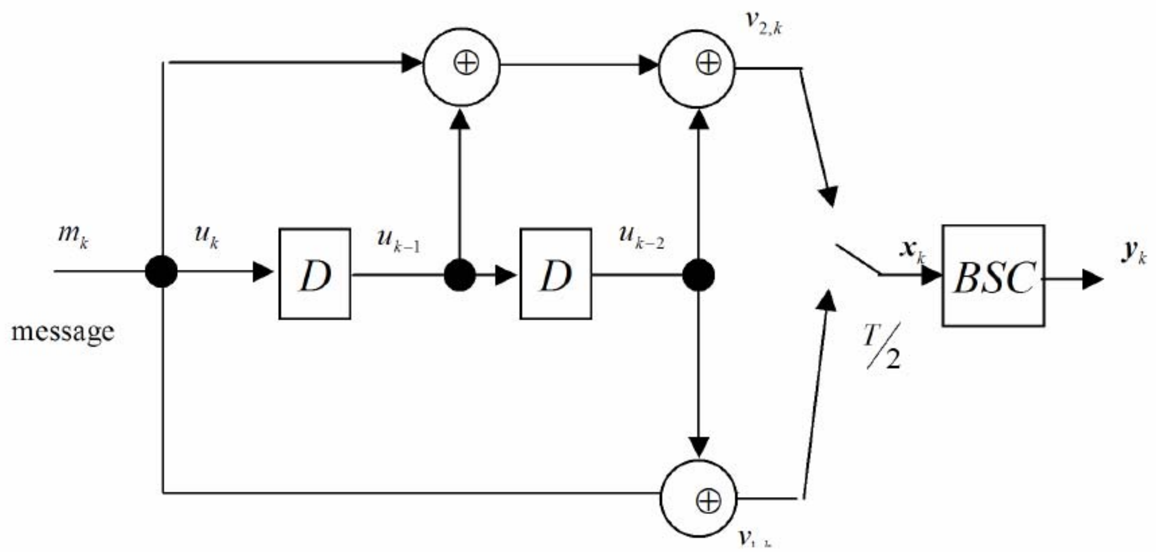
- G(D) can be any k x n matrix with entries in Fr(D) and rank k
- H(D) is parity matrix (n-k x n matrix with rank n-k)
  - When used as a generator, describes a **dual code** (all codewords in dual code orthogonal to codewords in original code)
- Code rate r=k/n

Constraint length (ν)



- Log$_2$ of the number of states of a convolutional encoder
- The number of D flip-flops in the obvious realization
- Often used as a measure of complexity of a convolutional code
- The complexity of a convolutional code is the minimum constraint length over all equivalent encoders
- An encoder is said to be minimal if the complexity equals the constraint length

Example, repeated



$$
\begin{aligned}
v_2(D) &= (1 + D + D^2)u_1(D) \\
v_1(D) &= (1 + D^2)u_1(D)
\end{aligned}
$$

$$G(D) = \begin{bmatrix} 1 + D + D^2 & 1 + D^2 \end{bmatrix}$$

$$H(D) = \begin{bmatrix} 1 + D^2 & 1 + D + D^2 \end{bmatrix}$$

Systematic code → $G(D) = \begin{bmatrix} 1 & \dfrac{1 + D^2}{1 + D + D^2} \end{bmatrix}$

8-state Ungerboeck encoder example

Image removed due to copyright restrictions.

$$G(D)H^*(D) = 0$$

$$\begin{aligned} 0 &= h_3 + h_2 \cdot D \\ 0 &= h_3 \cdot D^2 + h_2 + D \cdot h_1 \end{aligned}$$

$h_3 = D^2$, $h_2 = D$, and $h_1 = 1 + D^3$

$$H(D) = \begin{bmatrix} D^2 & D & 1 + D^3 \end{bmatrix}$$

Trellis diagram for 8-state code

Image removed due to copyright restrictions.

8-state Ungerboeck code with feedback

Image removed due to copyright restrictions.

$$G(D) = \begin{bmatrix} 1 & 0 & \frac{D^2}{1+D^3} \\ 0 & 1 & \frac{D}{1+D^3} \end{bmatrix}$$

$$0 = h_3 + h_1 \cdot \frac{D^2}{1+D^3}$$

$$0 = h_2 + h_1 \cdot \frac{D}{1+D^3}$$

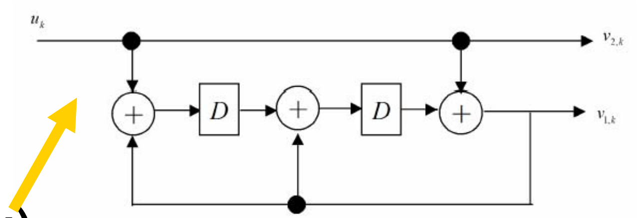
$$h_3 = D^2,\ h_2 = D, \text{ and } h_1 = 1 + D^3$$

$$H(D) = \begin{bmatrix} D^2 & D & 1+D^3 \end{bmatrix}$$

Same code as previous 8-state example with no feedback!

Implementations/Systematic encoders

- Two kinds of implementations most used
  - Feedback-free implementations
    - Often enumerated in code tables
  - Systematic (possibly with feedback)
    - All inputs directly passed to the output, with remaining n-k outputs being reserved as "parity" bits ($v_{n-i}(D)=u_{k-i}(D)$ for i=0,…,k-1)
    - When feedback is used, always possible to determine a systematic implementation



- Conversion to a systematic encoder

$$u'(D) = \frac{1}{1+D+D^2} \cdot u(D)$$

u'(D) can take on all possible causal sequences, just as can u(D), so this is simply a relabeling of the relationship of input to output sequences

$$v'(D) = u'(D)G(D) = u(D)\frac{1}{1+D+D^2}G(D) = u(D)\left[1 \quad \frac{1+D^2}{1+D+D^2}\right]$$

$$G_{sys} = G_{1:k}^{-1}G = [I \; G_{1:k}^{-1}G_{k+1:n}]$$

$$G_{sys} = [I_k \;\; h(D)] \qquad H_{sys} = [h^T(D) \;\; I_{n-k}] \qquad G(D) = [G_{1:k}(D) \; G_{k+1:n}(D)]$$

Catastrophic encoder

- Catastrophic encoder is the one for which at least one codeword with finite Hamming weight corresponds to an input of infinite Hamming weight
  - Since the set of all possible codewords is also a set of all possible error events, this is the same as saying a finite number of decoding errors in a sequence could lead to an infinite number of input bit errors – clearly a catastrophic event
- Catastrophic encoder test
  - An encoder is non-catastrophic if an only if the GCD of the determinants of all the k x k submatrices of G(D) is a nonnegative power of D (i.e. D^{delta} delta>=0)
- A non-catastrophic encoder always exists for any code
  - A systematic encoder can never be catastrophic (why?)
  - Also possible to find first a minimal, non-catastrophic encoder and then convert it to systematic encoder

Coding gain – comparison

- Method one – Bandwidth expansion
  - R=b/T, P=Ex/T (fix, R, P and T while allow W (bandwidth) to increase with n)
  - Simply compare d_{min} values of convolutionally coded systems with PAM2 at the same data rate R
    - Convolutional code system has 1/b_bar more W than uncoded system, and at fixed P and T this means that Ex_bar is reduced to b_bar*Ex_bar
      - Hence coded minimum distance is then dfree*b_bar*Ex_bar and coding gain is gamma=$10\log_{10}$(b_bar*d_{free}) – listed in coding tables
  - Somewhat unfair since assumes more bandwidth is available for free

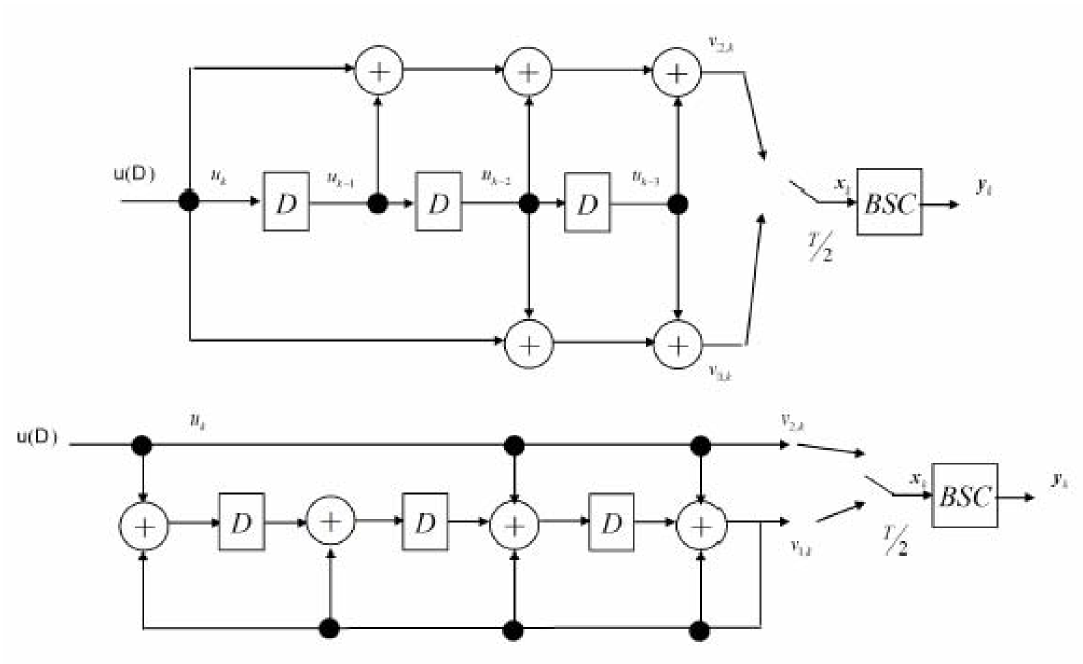
Coding gain – comparison

- Method two – Data-Rate (Energy) Reduction
  - Fix P and W (positive frequencies bandwidth)
  - For the coded system fixed W leads to fixed value 1/(b_bar*T)
    - Leads to data rate reduction by a factor b_bar (R_{code}=b_bar*2W)
  - The squared distance increases to d_{free}*Ex
    - But could have used a lower-speed uncoded system with 1/b_bar more energy per dimension for PAM2 transmission
  - Thus, the ratio of squared distance improvement is still b_bar*dfree (i.e. the coding gain)
  - This method of comparison is more fair since we don't assume any bandwidth expansion

Codes from tables

- Example r=1/2 table
  - [17 13]
  - $G(D)=[D^3+D^2+D+1 \;\; D^3+D+1]$

Figure removed due to copyright restrictions.



Readings

- Chapters 8,10 (parts related to convolutional codes)
- [1] G. Forney, Jr. "Convolutional codes I: Algebraic structure," *IEEE Transactions on Information Theory,* vol. 16, no. 6, pp. 720-738, 1970.
- [2] A. Viterbi "Convolutional Codes and Their Performance in Communication Systems," *IEEE Transactions on Communications,* vol. 19, no. 5, pp. 751-772, 1971.