STANFORD

Lecture S8B

Convolutional Code Performance Calculation

January 29, 2026

JOHN M. CIOFFI

Hitachi Professor Emeritus (recalled) of Engineering

Instructor EE379A – Winter 2026

Announcements & Agenda

- Announcements

- Today
  - Code Performance Analysis

This is really review, but also extension of P_e and $\overline{P}_b$ analysis from Chapter 1 and early lectures.

It's the same basic formulas, Just finding the minimum distance And nearest neighbors is more Elaborate

Typically, these are tabulated for designers, so the designer need not repeat this type of analysis. However, it's not hard should you need it someday for a new untabulated code.

Code/Decoder Performance Analysis

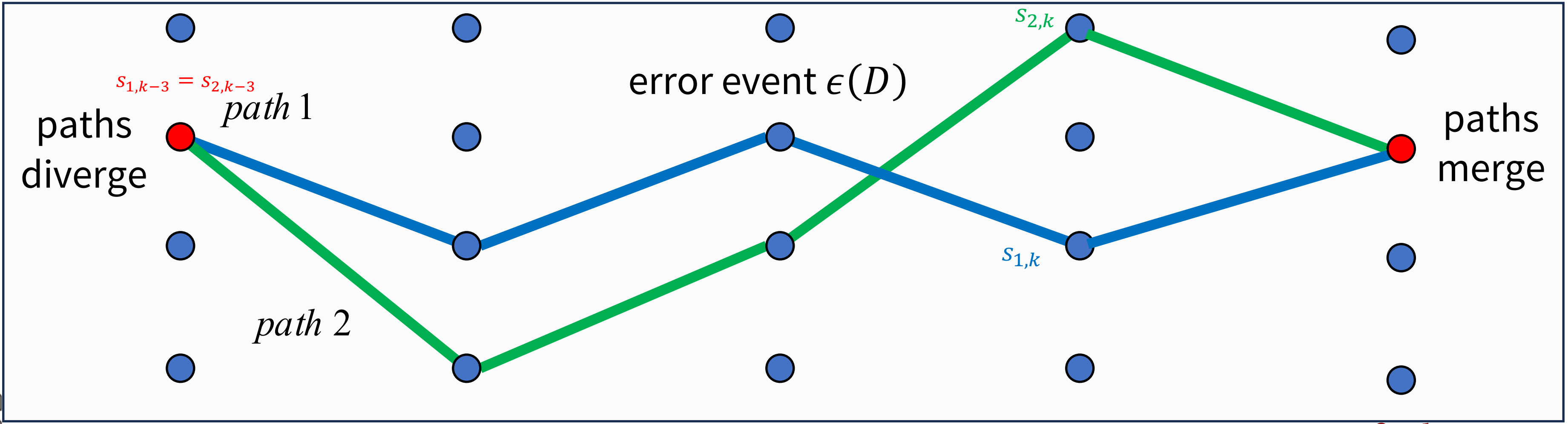
Section 7.2

MLSD Error Events

- MLSD is ML (maximum likelihood) – and error occurs if wrong symbol (codeword) chosen, P_e .
  - MLSD's add-compare-select is basic machine-learning element (a ReLU), so ML 2 ways 😀.

- The symbol is an entire codeword, which theoretically is infinite-length for CC's.
  - An error → error event $\epsilon(D) = x(D) - \hat{x}(D)$; $\epsilon_x(D)$ with $\epsilon_y(D)$ are difference between input/output.
  - For binary codes, the subtraction is binary (mod-2 or "xor").

- So, $\epsilon(D)$'s probability counts either at the time it begins (or ends, the two are equivalent) in P_e .
  - All the corresponding input-bit errors are counted as occurring at that time, so $\bar{P}_b \geq P_e$.



Minimum distance → distance spectrum

- Union bound includes all the distances:

$$P_{e,BSC} \leq \sum_{d=d_{free}}^{\infty} N_d \cdot \left[\sqrt{4p(1-p)}\right]^d \qquad P_{e,AWGN} \leq \sum_{d=d_{free}}^{\infty} N_d \cdot Q\left(\frac{d}{2\sigma}\right)$$

- For individual bit errors, really need counting function $N(b,d)$ - this differs from N_b but targets same goal:

$$\bar{P}_{b,BSC} \leq \frac{1}{k} \cdot \sum_{d=d_{free}}^{\infty} \sum_{i=1}^{\infty} i \cdot N(i,d) \cdot \left[\sqrt{4p(1-p)}\right]^d \qquad \bar{P}_{b,AWGN} \leq \frac{1}{k} \cdot \sum_{d=d_{free}}^{\infty} \underbrace{\sum_{i=1}^{\infty} i \cdot N(i,d)}_{N_b(d)} \cdot Q\left(\frac{d}{2\sigma}\right)$$

- Matlab finds N_d and $N_b\ (d)$ for convolutional codes (e.g., 8-state $r = 2/3$ code) – need to divide by k.

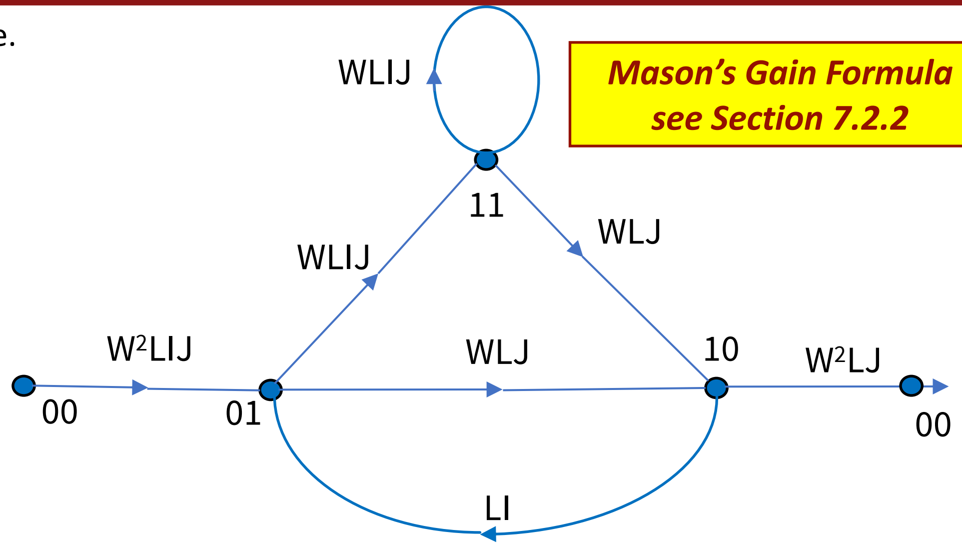
tmin=poly2trellis([3 2], [2 5 5; 3 2 1])
distspec(tmin,5) =
dfree: 4
weight: [1 11 108 417 1857] % $N_b\ (d \in [4:8]))$
event: [1 5 24 71 238] % $N_e\ (d \in [4:8])$

d	$N_b\ (d)$	$N_e(d))$
4	1	1
5	**11**	5
6	**108**	**24**
7	417	71
8	1857	238

Ouch! – next-to-nearest can dominate the union-bound sum

Transfer Function Analysis – example 4-state r=1/2

- Transfer function redraws trellis as single-time state machine.
- Each branch has multivariate transfer function:
  - W^d collects distance from all zeros as exponent,
  - L^l collects length as exponent (each branch is L),
  - I^i collects input errors w.r.t. all zeros as exponent,
  - J^j collects number of subsymbol differences.



Mason's Gain Formula see Section 7.2.2

$$T(W,L,I,J) = \frac{W^5 \cdot L^3 \cdot I \cdot J^3}{1 - WLIJ \cdot (L+1)}$$

$$= W^5 \cdot L^3 \cdot I \cdot J^3 \cdot \left[1 + WLIJ \cdot (L+1) + \left(WLIJ \cdot (L+1)\right)^2 + \cdots\right]$$

- Only 1 error event has $d_{free} = 5$:
  - length is 3, with
  - 1 error event (N_5= **1**) ,
  - 1 input bit error ($N(1{,}5)$ = **1**),
  - $N(2{:}\infty, 5)$**=0** **,** N_b (5) $= 1$.
- **2** error events have $d = 6$:
  - lengths are 5 and 6,
  - both have 2 input bit errors (so Nb (d=6) = 4), &
  - N_6= **2.**
  - $N(2{,}6)$=2; $N(1{,}6)$**=0 ;** $N(3{:}\infty, 6)$**=0**
  - $2 \cdot N(2{,}6) = 4$.

```
>> t=poly2trellis(3, [7 5])
   numInputSymbols: 2
  numOutputSymbols: 4
         numStates: 4
        nextStates: [4×2 double]
           outputs: [4×2 double]
>> distspec(t,4)
   dfree: 5
  weight: [1 4 12 32]
   event: [1 2 4 8]
```

End Lecture S8B