Lecture 9:
Performance of Convolutional Codes
Weight enumerators of convolutional codes

- For many relevant channels $P_{Y|X}$, MAP decoding reduces to minimum distance decoding for some appropriately defined distance (e.g., Hamming distance for the BSC, or Euclidean distance for the Binary-Input AWGN channel).

- A related problem of individual interest is to determine the code weight enumerator.

- The code $\mathcal{C}$ is the set of all semi-infinite codeword sequences $c(D) = u(D)G(D)$ obtained for all $u(D) \in \mathbb{F}_2^k(D)$ (i.e., power-series inputs with non-negative powers), assuming that the initial state of the encoder is $s_0 = 0$. 

Input-output weight enumerator

- Consider an input sequence \( u(D) \) of finite Hamming weight \( i(u) = i \), generating an output sequence \( c(D) \) of finite Hamming weight \( w(c) = w \).

- To such input/output pair, we associate the monomial \( I^i W^w \).

- The input-output weight enumerating function is defined as the sum of all such monomials over all possible input/output pairs:

\[
T(I, W) = \sum_{u(D) \in \mathbb{F}_2^k(D)} \sum_{c(D) = u(D)G(D)} I^{i(u)} W^{w(c)}
\]

- Letting \( t_{i,w} \) denote the number of input/output sequence pairs with input-output weight \((i, w)\), we can write

\[
T(I, W) = \sum_{i>0} \sum_{w>0} t_{i,w} I^i W^w
\]
Computing weight enumerators

- Method to enumerate the trellis paths based on the symbolic transfer function of a weighted graph.

- Modify the state diagram by splitting state zero in two states \((0_{\text{in}} \text{ and } 0_{\text{out}})\) and by eliminating the \(0 \rightarrow 0\) transition in the original state diagram.

- Label each transition of the modified state diagram with input-output weight \((i, w)\) with the monomial label \(I^i W^w\).

- The input-output weight of any path starting from \(0_{\text{in}}\) and ending in \(0_{\text{out}}\) after an arbitrary number of transitions is obtained by the exponents of the variables \(I\) and \(W\) of the product of the labels of all the transitions traversed by the path.

- The function \(T(I, W)\) is obtained by summing over all such paths.
Algebraic general method

- Instead of applying step-by-step graph reduction, we can obtain $T(I, W)$ by solving a linear system over the formal power series in the variables $I$ and $W$.

- Define the formal input variable $X$ associated to the state $0_{\text{in}}$, the formal output variable $Y$ (associated to the state $0_{\text{out}}$, and the formal state variables $V_s$, associated to all other non-zero states $s \in \Sigma$.

- For each state $s \neq 0$, the following node flow equation holds:

$$V_s = \sum_{s'} I^{i(s', s)} W^{w(s', s)} V_{s'} + I^{i(0, s)} W^{w(0, s)} X$$

where $w(s', s)$ and $i(s', s)$ are the output and the information weights of the transition $s' \rightarrow s$ in the trellis.

- Notice that each transition $(s', s)$ of the modified state diagram is labeled by the monomial coefficient $I^{i(s', s)} W^{w(s', s)}$. 

Copyright G. Caire (Sample Lectures)
• Only existing transitions are taken into account. We may think of non-existing transitions as having monomial coefficients equal to 0.

• For the state $0_{\text{out}}$, we have the output equation

$$Y = \sum_{s'} I^{i(s',0)} W^{w(s',0)} V_{s'}$$

• We define the vector of state variables $V = (V_1, V_2, \ldots, V_{2^m-1})$, and write the system of flow equations in compact form as

$$V = VA(I, W) + XB(I, W), \quad Y = VC(I, W)$$

where $A(I, W)$ is a matrix of size $(2^m-1) \times (2^m-1)$, $B(I, W)$ is a $1 \times (2^m-1)$ row vector and $C(I, W)$ is a $(2^m-1) \times 1$ column vector of monomials in the indeterminates $I, W$.  


The desired enumerating function is given by $T(I,W) = Y/X$, after eliminating the state variables from the system of flow equations:

\[
T(I,W) = Y/X \\
= B(I,W)(I - A(I,W))^{-1}C(I,W) \\
= B(I,W) \left[ \sum_{j=0}^{\infty} A(I,W)^j \right] C(I,W)
\]

It is clear from the second line of the above equation that $T(I,W)$ is a formal power series in the two indeterminates $I$ and $W$. 
The problem with this method is that symbolic matrix inversion is a very complicated operation. Then, we can consider a truncated approximation of the weight enumerator

\[
T(I, W) \approx B(I, W) \left[ I + A(I, W) + A(I, W)^2 + \cdots + A(I, W)^N \right] C(I, W)
\]
Weight enumerator computation example

- The modified state diagram for the code $(2, 1)$ 4-state code with $G(D) = [1 + D^2, 1 + D + D^2]$ was obtained before.

- The corresponding state flow equations are

\[
\begin{align*}
V_1 &= W^2IX + IV_2 \\
V_2 &= WV_1 + WV_3 \\
V_3 &= WIV_1 + WIV_3 \\
Y &= W^2V_2
\end{align*}
\]
• In matrix from we have

$$V = V \begin{bmatrix} 0 & W & WI \\ I & 0 & 0 \\ 0 & W & WI \end{bmatrix} + X \begin{bmatrix} W^2I & 0 & 0 \end{bmatrix}$$

$$Y = V \begin{bmatrix} 0 \\ W^2 \\ 0 \end{bmatrix}$$

• By solving explicitly $T(I, W) = B(I - A)^{-1}C$ we obtain

$$T(I, W) = \frac{W^5I}{1 - 2WI}$$
• By letting $I = 1$ and by expanding the result as a series of powers of $W$, we obtain

$$T(W, 1) = W^5 + 2W^6 + 4W^7 + 8W^8 + 16W^9 + \cdots$$

showing that the code has one path of weight 5 (the free Hamming distance is 5), two paths of weight 6, 4 paths of weight 7 and so on (check this out directly on the trellis).
Free distance of a convolutional code

**Definition 27.** The lowest \( w > 0 \) for which \( t_{i,w} > 0 \) is called the free distance of the convolutional code, indicated by \( d_{\text{free}}(C) \).

- \( d_{\text{free}}(C) \) is the analogous of \( d_{\text{min}} \) for block codes.

- \( d_{\text{free}}(C) \) can be obtained from \( T(I, W) \) as

\[
T(I, W) = \sum_{w=d_{\text{free}}}^{\infty} \sum_{i=1}^{\infty} t_{i,w} I^i W^w = W^{d_{\text{free}}} f(I, W)
\]

for some rational function \( f(I, W) \) such that \( f(I, 0) \neq 0 \).
Performance bounds for MAP decoding

- We consider the block code $C_N$ obtained from a convolutional code $C$ by trellis termination.

- We wish to find tight upper bounds on the performance (probability of error) of $C_N$ under MAP decoding over BIOS channels.

- Without loss of generality, we use LLRs as canonical channel output:

$$\Lambda(y) = \log \frac{P_{Y|X}(y|x = 0)}{P_{Y|X}(y|x = 1)}$$
Examples

• Example: BSC. In this case,

\[ \Lambda(y) = (-1)^y \log \frac{1 - p}{p} \]

• Example: BI-AWGN. In this case,

\[ \Lambda(y) = \frac{4\sqrt{E_s}}{N_0} y \]
Symmetry properties

- Recall the symmetry properties:

\[ P_{\Lambda|X}(\lambda|x = 1) = P_{\Lambda|X}(-\lambda|x = 0) \]
\[ P_{\Lambda|X}(-\lambda|x = 0) = e^{-\lambda} P_{\Lambda|X}(\lambda|x = 0) \]

- The channel is fully characterized by the transition distribution (or density) function

\[ p_0(\lambda) = P_{\Lambda|X}(\lambda|x = 0) \]
MAP rule in terms of LLRs

\[ \hat{c} = \arg \max_{c \in \mathcal{C}_N} \sum_{i=0}^{N-1} \log P_{\Lambda|X}(\lambda_i^{(1)}, \ldots, \lambda_i^{(n)} | c_i) \]

\[ = \arg \max_{c \in \mathcal{C}_N} \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} \log P_{\Lambda|X}(\lambda_i^{(\ell)} | c_i^{(\ell)}) \]

\[ = \arg \max_{c \in \mathcal{C}_N} \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} \log p_0 \left( (-1)^{c_i^{(\ell)}} \lambda_i^{(\ell)} \right) \]

\[ = \arg \max_{c \in \mathcal{C}_N} \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} \log e^{-c_i^{(\ell)} \lambda_i^{(\ell)}} p_0 \left( \lambda_i^{(\ell)} \right) \]

\[ = \arg \min_{c \in \mathcal{C}_N} \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} c_i^{(\ell)} \lambda_i^{(\ell)} \]

\[ = \arg \min_{c \in \mathcal{C}_N} \lambda^T c \]
Pairwise Error Probability (PEP)

- Consider two codewords $c, c' \in C_N$ and consider the (already defined!) pairwise error event $\{c \rightarrow c'\}$, this is the event that (for the time being we neglect the ties)

$$\lambda^T c > \lambda^T c', \text{ given that } c \text{ is transmitted}$$

- The probability of such event, called PEP, is given by

$$P(c \rightarrow c') \triangleq \mathbb{P} \left( \lambda^T c > \lambda^T c' \bigg| X = c \right)$$
Uniform error property

**Theorem 14.** For a linear code over a BIOS channel, we have

\[ P(c \rightarrow c') = P(0 \rightarrow c \oplus c') \]

Proof:

- The pairwise error event is given by

\[ 0 > \nabla^T (c' - c) \]

with respect to the product measure \( P(\lambda|c) = \prod_i P_{\lambda|X}(\lambda_i|c_i) \).
• The components for which \( c_i = c'_i \) do not contribute to the metric difference \( \lambda^T (c' - c) \).

• The components for which \( c_i = 0 \) and \( c'_i = 1 \) contribute for a term \( +\lambda_i \).

• The components for which \( c_i = 1 \) and \( c'_i = 0 \) contribute for a term \( -\lambda_i \).

• Define the product measure \( P(\lambda|0) = \prod_i P_{\Lambda|X}(\lambda_i|0) \) and notice that

\[
P(\lambda|0) = \prod_i P_{\Lambda|X}((-1)^{c_i}\lambda_i|c_i)
\]

• Letting \( c'' = c' \oplus c \) (over the binary field), it follows that

\[
P(0 > \lambda^T (c' - c)|X = c) = P(0 > \lambda^T c''|X = 0) = P(0 \rightarrow c \oplus c')
\]
• **Corollary:** the PEP depends only on the Hamming distance between $c$, $c'$ (equivalently, on $w(c \oplus c')$).

• **Corollary:** the union bound (UB) on the average probability of error is given by

\[
P_e = \mathbb{P} \left( \bigcup_{c \neq 0: c \in C_N} \{0 \rightarrow c\} \right) \\
\leq \sum_{c \neq 0: c \in C_N} P(0 \rightarrow c) \\
= \sum_{w = d_{\text{min}}(C_N)}^{NN} A_w P_w
\]

where $P_w$ is the PEP for a codeword of weight $w$. 

Evaluating the PEP

- In some cases, \( P_w \) can be computed exactly.

- **Example: BSC.** In this case, for odd \( w \)

\[
P_w = \mathbb{P}\left( \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} c_i^{(\ell)} x_i^{(\ell)} < 0 \left| X = 0 \right. \right)
\]

\[
= \mathbb{P}\left( \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} c_i^{(\ell)} (-1)^{Y_i^{(\ell)}} \log \frac{1-p}{p} < 0 \left| X = 0 \right. \right)
\]

\[
= \mathbb{P}\left( \sum_{i=1}^{w} (-1)^{Y_i} < 0 \left| X_1 = 0, \ldots, X_w = 0 \right. \right)
\]

\[
= \sum_{e > w/2} \binom{w}{e} p^e (1-p)^{w-e}
\]

(for even \( w \) we need to randomize on the “tie” case and add the term \( \frac{1}{2} \binom{w}{w/2} p^{w/2} (1-p)^{w/2} \)).
• Example: BI-AWGN. In this case, we have

\[
P_w = \mathbb{P} \left( \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} x_i^{(\ell)} \lambda_i^{(\ell)} < 0 \bigg| X = 0 \right)
\]

\[
= \mathbb{P} \left( \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} c_i^{(\ell)} \left( \sqrt{E_s} + Z_i^{(\ell)} \right) < 0 \right)
\]

\[
= \mathbb{P} \left( \sum_{i=0}^{N-1} \sum_{\ell=1}^{n} c_i^{(\ell)} Z_i^{(\ell)} < -w \sqrt{E_s} \right)
\]

\[
= \mathbb{P} \left( Z_w < -w \sqrt{E_s} \right)
\]

where \( Z_w \sim \mathcal{N}(0, wN_0/2) \), since it is given by the sum of \( w \) i.i.d. Gaussian RVs \( \sim \mathcal{N}(0, N_0/2) \).
• Finally, we have

\[ P_w = Q\left(\sqrt{\frac{w}{N_0} \cdot \frac{2E_s}{R}}\right) = Q\left(\sqrt{w \cdot \frac{2E_b}{N_0}}\right) \]

where we define the received energy per information bit \( E_b = \frac{E_s}{R} \).

• Use the \( Q(x) \) function approximation for large argument:

\[ Q(x) \leq \frac{1}{2} e^{-x^2/2} \]

and plot \( P_e \) versus \( E_b/N_0 \) in dB on a log-10 scale.

• Asymptotic (power) coding gain in dB:

\[ \gamma_{\text{dB}} = 10 \log_{10}(d_{\text{free}}R) \]
Bounding the PEP

- We consider a general technique to bound the PEP known as the Chernoff bound. The Bhattacharyya bound already seen is a special case.

- Let $\Lambda_j \sim p_0(\lambda)$ denote i.i.d. RVs distributed as the LLRs when the all-zero codeword is transmitted. Then, we can write

$$P_w \leq \Pr \left( \sum_{j=1}^{w} \Lambda_j \leq 0 \right) = \mathbb{E} \left[ 1 \left\{ \sum_{j=1}^{w} \Lambda_j \leq 0 \right\} \right]$$

$$\leq \min_{s \geq 0} \mathbb{E} \left[ e^{-s \sum_{j=1}^{w} \Lambda_j} \right] = \min_{s \geq 0} \Phi_{\Lambda}^w(s)$$

where we define the moment generating function (or Laplace transform) of the cdf of $\Lambda_j$ as

$$\Phi_{\Lambda}(s) = \mathbb{E} \left[ e^{-s \Lambda_j} \right] = \int_{-\infty}^{\infty} e^{-sz} p_0(z) dz$$
Chernoff Union Bound

• If we have the weight enumerator \( A(W) = \sum_{w=d_{\text{min}}(C_N)}^{nN} A_w W^w \) of the code, then we obtain

\[
P_e \leq \min_{s \geq 0} \sum_{w=d_{\text{min}}(C_N)}^{nN} A_w \Phi^w_A(s) = \min_{s \geq 0} A(\Phi_A(s))
\]

• Example: BSC. In this case, one can show that, indeed, the optimal value is \( s \) is equal to 1/2, and we obtain

\[
\Phi_A(1/2) = \sqrt{4p(1-p)}.
\]
• Example: BI-AWGN. In this case, \( \Lambda_j \sim \mathcal{N}\left(\frac{4E_s}{N_0}, \frac{8E_s}{N_0}\right) \). It follows that

\[
\Phi_\Lambda(s) = e^{-\frac{4E_s}{N_0} s(1-s)}
\]

• Minimizing w.r.t. \( s \geq 0 \), we find again that the optimal \( s \) is \( 1/2 \), therefore

\[
\Phi_\Lambda(1/2) = e^{-\frac{E_s}{N_0}}.
\]
Using $T(I, W)$ for error bounding

- Consider the case of $N \gg \nu_{\max}$ (relevant for applications), such that $R \approx k/n$ and $d_{\min}(C_N) = d_{\text{free}}(C)$.

- Recall that $T(I, W)$ enumerates all simple error events paths, i.e., those paths leaving state 0 at time $i = 0$ and merging to state 0 after some steps, without passing through state zero before.

- **Observation 1**: all non-zero code words in the trellis-terminated code $C_N$ can be obtained by concatenating, in any possible way, shifted simple error events separated by all-zero segments.

- **Observation 2**: consider a codeword $c = c' \oplus c''$, such that the paths of $c'$ and $c''$ diverge from the all-zero path in non-overlapping segments. Then, $c$ can be removed from the UB. In fact, we have that

\[
\{0 \to c\} \subseteq \{0 \to c'\} \cup \{0 \to c''\}
\]
• Eliminating all such “composite” error events, it follows that it is sufficient to count only the elementary error events, formed by the simple error events and by their shifts.

• Enumerating all shifted elementary error events is very easy. In fact, if there are $t_{i,w,\ell}$ paths with Hamming output weight $w$, information weight $i$ and length $\ell$, there are at most $\max\{N - \ell + 1, 0\}t_{i,w,\ell}$ elementary error events in the trellis of $C_N$.

• Since $\max\{N - \ell + 1, 0\} \leq N$ for all $\ell$, a very simple Chernoff UB on the word error probability (WER) for $C_N$ is given by

$$P_e \leq \min_{s \geq 0} NT(1, \Phi_\Lambda(s))$$

• Notice that the WER depends roughly linearly on the trellis length $N$. 
Bound on the bit-error rate (BER)

- A more meaningful performance measure at the output of a convolutional decoder is the BER, denoted by $P_b$, i.e., the probability of error for information bits.

- Recall that the dimension of $C_N$ is given by $k(N - \nu_{\text{max}})$.

- On the other hand, each elementary error event with information weight $i$ yields $i$ information bits in error.

- The average fraction of bit in error (BER) is bounded by

$$P_b \leq \frac{N}{k(N - \nu_{\text{max}})} \sum_{w=d_{\text{free}}}^{nN} i \ t_{i,w} P_w$$
• For large $N$, the coefficient in front of the sum tends to $1/k$. Therefore, for large $N$ we have

$$P_b \leq \frac{1}{k} \sum_{w=d_{\text{free}}}^{\infty} it_{i,w} P_w \leq \min_{s \geq 0} \frac{1}{k} \frac{\partial}{\partial I} T(I, \Phi_{\Lambda}(s)) \bigg|_{I=1}$$
Tighter union bounds for the BI-AWGN channel

• Let’s go back to the UB without Chernoff bound relaxation. We have

\[ P_e \leq \sum_{w=d_{\text{free}}}^{\infty} \left( N \sum_{i>0} t_{i,w} \right) Q \left( \sqrt{\frac{2wE_s}{N_0}} \right) \]

• The problem with the above expressions is that the summation is over an infinite number of terms.

• We make use of Craig’s integral expression for the Gaussian tail function \( Q(z) \) (for \( z \geq 0 \)):

\[ Q(z) = \frac{1}{\pi} \int_{0}^{\pi/2} \exp \left( -\frac{z^2}{2 \sin^2 \phi} \right) d\phi \]
• Then, the WER bound can be written as

\[ P_e \leq \frac{N}{\pi} \int_0^{\pi/2} T \left( 1, e^{-\frac{E_s}{N_0 \sin^2 \phi}} \right) d\phi \]

• Similarly, the BER bound can be written as

\[ P_b \leq \frac{1}{k\pi} \int_0^{\pi/2} \frac{\partial}{\partial I} T \left( I, e^{-\frac{E_s}{N_0 \sin^2 \phi}} \right) \bigg|_{I=1} d\phi \]

• The integrals with respect to \( \phi \) can be easily computed numerically by using the following Gauss-Tchebyshev quadrature rule of order \( \nu \).
Denote the integrand by $g(\sin^2(\phi))$. Then, we can write

$$\frac{1}{\pi} \int_0^{\pi/2} g(\sin^2(\phi))d\phi = \frac{1}{2\pi} \int_{-1}^{1} g(x^2) \frac{dx}{\sqrt{1 - x^2}}$$

$$\approx \frac{1}{2\nu} \sum_{i=0}^{\nu-1} g\left(\cos^2\left(\frac{2i + 1}{2\nu}\pi\right)\right)$$

$$\nu \text{ odd} \quad \frac{1}{\nu} \sum_{i=0}^{(\nu-1)/2 - 1} g\left(\cos^2\left(\frac{2i + 1}{2\nu}\pi\right)\right)$$
• A looser but simpler bound can be obtained by noticing that \( \sin^2 \phi \) is increasing for \( \phi \in [0, \pi/2] \) and attains its maximum value 1 for \( \phi = \pi/2 \).

• It is immediate to see that if we replace the integrand function with with its maximum, i.e., its value at \( \phi = \pi/2 \), we obtain an upper bound that coincides with the Chernoff union bound improved by a factor 1/2.
End of Lecture 9