ECEN 5682 Theory and Practice of Error Control Codes

Convolutional Code Performance

Peter Mathys

University of Colorado

Spring 2007

Performance Measures

Definition: A convolutional encoder which maps one or more data sequences of infinite weight into code sequences of finite weight is called a *catastrophic encoder*.

Example: *Encoder #5.* The binary $R = 1/2$, $K = 3$ convolutional encoder with transfer function matrix

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

has the encoder state diagram shown in Figure 15, with states $S_0 = 00$, $S_1 = 10$, $S_2 = 01$, and $S_3 = 11$.

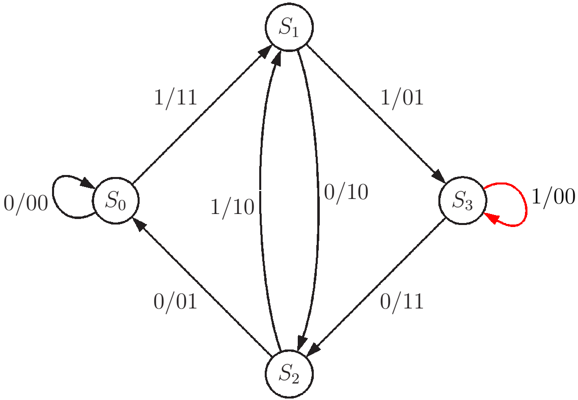


Fig.15 Encoder State Diagram for Catastrophic $R = 1/2$, $K = 3$ Encoder

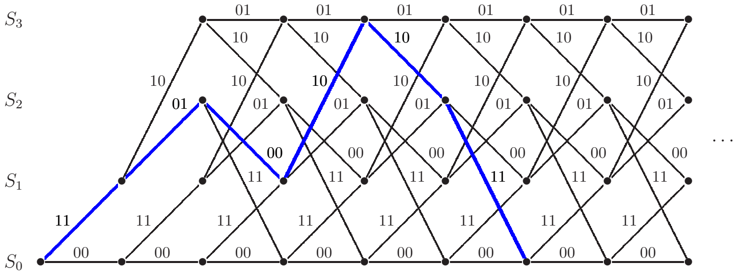


Fig.16 A Detour of Weight $w = 7$ and $i = 3$, Starting at Time $t = 0$

Definition: The *complete weight distribution* $\{A(w,i,\ell)\}$ of a convolutional code is defined as the number of detours (or codewords), beginning at time 0 in the all-zero state S_0 of the encoder, returning again for the first time to S_0 after ℓ time units, and having code (Hamming) weight w and data (Hamming) weight i.

Definition: The *extended weight distribution* $\{A(w,i)\}$ of a convolutional code is defined by

$$A(w,i) = \sum_{\ell=1}^{\infty} A(w,i,\ell)\,.$$

That is, $\{A(w,i)\}$ is the number of detours (starting at time 0) from the all-zero path with code sequence (Hamming) weight w and corresponding data sequence (Hamming) weight i.

Definition: The *weight distribution* $\{A_w\}$ of a convolutional code is defined by

$$A_w = \sum_{i=1}^{\infty} A(w,i)\ .$$

That is, $\{A_w\}$ is the number of detours (starting at time 0) from the all-zero path with code sequence (Hamming) weight w.

Theorem: The *probability of an error event (or decoding error)* P_E for a convolutional code with weight distribution $\{A_w\}$, decoded by a ML decoder, at any given time t (measured in frames) is upper bounded by

$$P_E \leq \sum_{w=d_{\text{free}}}^{\infty} A_w\, P_w(\mathcal{E})\ ,$$

where

$$P_w(\mathcal{E}) = P\{\text{ML decoder makes detour with weight } w\}\ .$$

Theorem: On a memoryless BSC with transition probability $\epsilon < 0.5$, the probability of error $P_d(\mathcal{E})$ between two detours or codewords distance d apart is given by

$$P_d(\mathcal{E}) = \begin{cases} \displaystyle\sum_{e=(d+1)/2}^{d} \binom{d}{e} \epsilon^e (1-\epsilon)^{d-e} , & d \text{ odd} , \\ \displaystyle\frac{1}{2}\binom{d}{d/2} \epsilon^{d/2} (1-\epsilon)^{d/2} + \sum_{e=d/2+1}^{d} \binom{d}{e} \epsilon^e (1-\epsilon)^{d-e} , & d \text{ even} . \end{cases}$$

Proof: Under the Hamming distance measure, an error between two binary codewords distance d apart is made if more than $d/2$ of the bits in which the codewords differ are in error. If d is even and exactly $d/2$ bits are in error, then an error is made with probability $1/2$. QED

Note: A somewhat simpler but less tight bound is obtained by dropping the factor of $1/2$ in the first term for d even as follows

$$P_d(\mathcal{E}) \leq \sum_{e=\lceil d/2 \rceil}^{d} \binom{d}{e} \epsilon^e (1-\epsilon)^{d-e} .$$

A much simpler, but often also much more loose bound is the Bhattacharyya bound

$$P_d(\mathcal{E}) \leq \frac{1}{2} \left[4\epsilon (1-\epsilon)\right]^{d/2} .$$

Probability of Symbol Error. Suppose now that $A_w = \sum_{i=1}^{\infty} A(w,i)$ is substituted in the bound for P_E. Then

$$P_E \leq \sum_{w=d_{\text{free}}}^{\infty} \sum_{i=1}^{\infty} A(w,i)\, P_w(\mathcal{E})\,.$$

Multiplying $A(w,i)$ by i and summing over all i then yields the total number of data symbol errors that result from all detours of weight w as $\sum_{i=1}^{\infty} i\, A(w,i)$. Dividing by k, the number of data symbols per frame, thus leads to the following theorem.

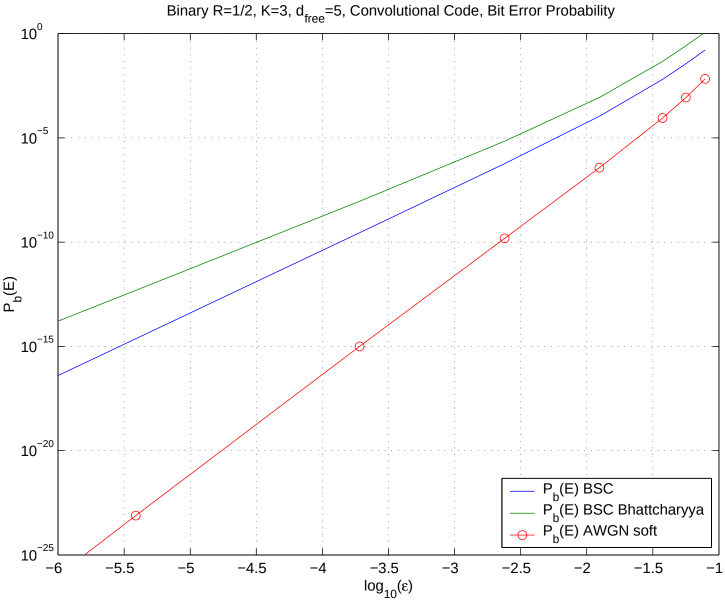
Theorem: The *probability of a symbol error* $P_s(\mathcal{E})$ at any given time t (measured in frames) for a convolutional code with rate $R = k/n$ and extended weight distribution $\{A(w,i)\}$, when decoded by a ML decoder, is upper bounded by

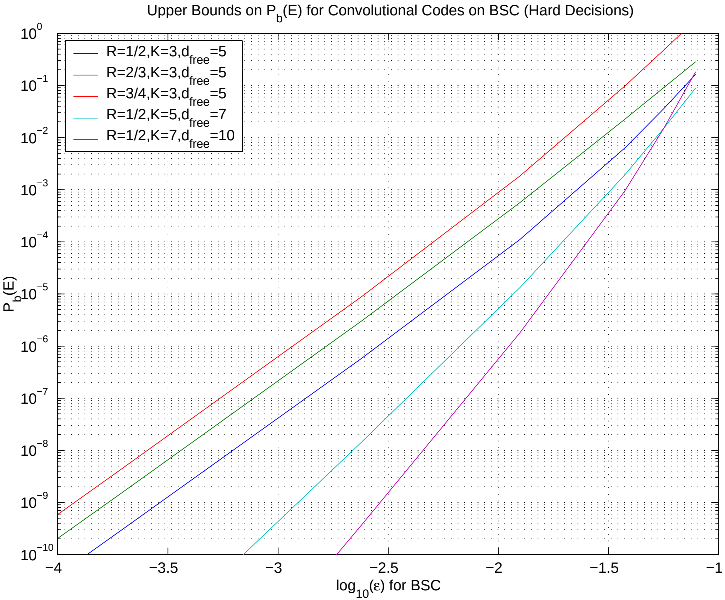
$$P_s(\mathcal{E}) \leq \frac{1}{k} \sum_{w=d_{\text{free}}}^{\infty} \sum_{i=1}^{\infty} i\, A(w,i)\, P_w(\mathcal{E})\,,$$

where $P_w(\mathcal{E})$ is the probability of error between the all-zero path and a detour of weight w.

The graph on the next slide shows different bounds for the probability of a bit error on a BSC for a binary rate $R = 1/2$, $K = 3$ convolutional encoder with transfer function matrix

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


Binary R=1/2, K=3, d_free=5, Convolutional Code, Bit Error Probability
10^0
10^−5
10^−10
10^−15
10^−20
10^−25
P_b(E)
−6
−5.5
−5
−4.5
−4
−3.5
−3
−2.5
−2
−1.5
−1
log_10(ε)
P_b(E) BSC
P_b(E) BSC Bhattcharyya
P_b(E) AWGN soft


Upper Bounds on P_b(E) for Convolutional Codes on BSC (Hard Decisions)
R=1/2,K=3,d_free=5
R=2/3,K=3,d_free=5
R=3/4,K=3,d_free=5
R=1/2,K=5,d_free=7
R=1/2,K=7,d_free=10
P_b(E)
10^0
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
10^−7
10^−8
10^−9
10^−10
−4
−3.5
−3
−2.5
−2
−1.5
−1
log_10(ε) for BSC

In[8]:= `Q = {{0, x, x*y, 0}, {x*y, 0, 0, x}, {0, 1, x^2*y, 0}, {0, 0, 0, 0}}`

Out[8]= $\{\{0, x, x\,y, 0\}, \{x\,y, 0, 0, x\}, \{0, 1, x^2\,y, 0\}, \{0, 0, 0, 0\}\}$

In[9]:= `G1 = {x^2*y, 0, 0, 0}`

Out[9]= $\{x^2\,y, 0, 0, 0\}$

In[10]:= `IQI = Inverse[IdentityMatrix[4] - Q]`

Out[10]=
$$\left\{\left\{\frac{1-x^2 y}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x+x y-x^3 y}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x y}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x^2+x^2 y-x^4 y}{1-2x^2 y-x^2 y^2+x^4 y^2}\right\},\right.$$
$$\left\{\frac{x y-x^3 y^2}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{1-x^2 y}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x^2 y^2}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x-x^3 y}{1-2x^2 y-x^2 y^2+x^4 y^2}\right\},$$
$$\left\{\frac{x y}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{1}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{1-x^2 y}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x}{1-2x^2 y-x^2 y^2+x^4 y^2}\right\},$$
$$\left.\{0, 0, 0, 1\}\right\}$$

In[11]:= `G = G1*IQI`

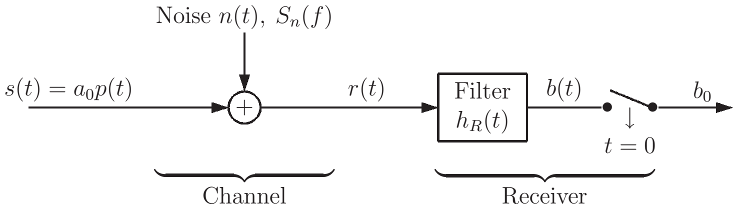
Out[11]=
$$\left\{\left\{\frac{x^2 y\,(1-x^2 y)}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x^2 y\,(x+x y-x^3 y)}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x^3 y^2}{1-2x^2 y-x^2 y^2+x^4 y^2}, \frac{x^2 y\,(x^2+x^2 y-x^4 y)}{1-2x^2 y-x^2 y^2+x^4 y^2}\right\},\right.$$
$$\left.\{0, 0, 0, 0\}, \{0, 0, 0, 0\}, \{0, 0, 0, 0\}\right\}$$

In[16]:= `G[[1, 4]]`

Out[16]=
$$\frac{x^2 y\,(x^2+x^2 y-x^4 y)}{1-2x^2 y-x^2 y^2+x^4 y^2}$$

Transmission Over AWGN Channel

The following figure shows a "one-shot" model for transmitting a data symbol with value a_0 over an additive Gaussian noise (AGN) waveform channel using pulse amplitude modulation (PAM) of a pulse $p(t)$ and a matched filter (MF) receiver. The main reason for using a "one-shot" model for performance evaluation with respect to channel noise is that it avoids intersymbol interference (ISI).



If the noise is white with power spectral density (PSD) $S_n(f) = \mathcal{N}_0/2$ for all f, the channel model is called **additive white Gaussian noise (AWGN)** model. In this case the matched filter (which maximizes the SNR at its output at $t = 0$) is

$$h_R(t) = \frac{p^*(-t)}{\int_{-\infty}^{\infty} |p(\mu)|^2 d\mu} \qquad \Longleftrightarrow \qquad H_R(f) = \frac{P^*(f)}{\int_{-\infty}^{\infty} |P(\nu)|^2 d\nu} ,$$

where * denotes complex conjugation. If the PAM pulse $p(t)$ is normalized so that $E_p = \int_{-\infty}^{\infty} |p(\mu)|^2 d\mu = 1$ then the symbol energy at the input of the MF is

$$E_s = E\Big[\int_{-\infty}^{\infty} |s(\mu)|^2 \, d\mu \Big] = E\left[|a_0|^2\right] ,$$

where the expectation is necessary since a_0 is a random variable.

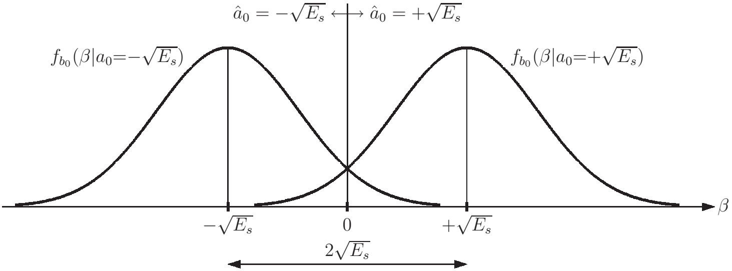
When the AWGN model with $S_n(f) = \mathcal{N}_0/2$ is used and $a_0 = \alpha$ is transmitted, the received symbol b_0 at the sampler after the output of the MF is a Gaussian random variable with mean α and variance $\sigma_b^2 = \mathcal{N}_0/2$. For antipodal binary signaling (e.g., using BPSK) $a_0 \in \{-\sqrt{E_s}, +\sqrt{E_s}\}$ where E_s is the (average) energy per symbol. Thus, b_0 is characterized by the conditional pdf's

$$f_{b_0}(\beta|a_0{=}-\sqrt{E_s}) = \frac{e^{-(\beta+\sqrt{E_s})^2/\mathcal{N}_0}}{\sqrt{\pi\mathcal{N}_0}} \,,$$

and

$$f_{b_0}(\beta|a_0{=}+\sqrt{E_s}) = \frac{e^{-(\beta-\sqrt{E_s})^2/\mathcal{N}_0}}{\sqrt{\pi\mathcal{N}_0}} \,.$$

These pdf's are shown graphically on the following slide.



If the two values of a_0 are equally likely or if a ML decoding rule is used, then the (hard) decision threshold per symbol is to decide $a_0 = +\sqrt{E_s}$ if $\beta > 0$ and $a_0 = -\sqrt{E_s}$ otherwise.

The probability of a symbol error when hard decisions are used is

$$P(\mathcal{E}|A_0{=}-\sqrt{E_s}) = \frac{1}{\sqrt{\pi\mathcal{N}_0}}\int_0^\infty e^{-(\beta+\sqrt{E_s})^2/\mathcal{N}_0}\,d\beta = \frac{1}{2}\text{erfc}\Big(\sqrt{\frac{E_s}{\mathcal{N}_0}}\Big),$$

where $\text{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-\gamma^2}d\gamma \approx e^{-x^2}$. Because of the symmetry of antipodal signaling, the same result is obtained for $P(\mathcal{E}|a_0{=}+\sqrt{E_s})$ and thus a BSC derived from an AWGN channel used with antipodal signaling has transition probability

$$\epsilon = \frac{1}{2}\,\text{erfc}\Big(\sqrt{\frac{E_s}{\mathcal{N}_0}}\Big)\,,$$

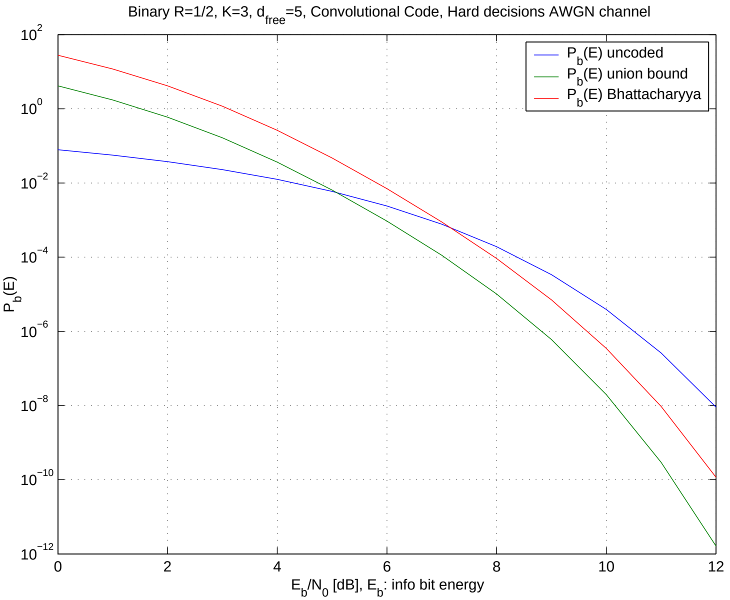
where E_s is the energy received per transmitted symbol.

To make a fair comparison in terms of signal-to-noise ratio (SNR) of the transmitted information symbols between coded and uncoded systems, the energy per code symbol of the coded system needs to be scaled by the rate R of the code. Thus, when hard decisions and coding are used in a binary system, the transition probability of the BSC model becomes

$$\epsilon_c = \frac{1}{2}\,\text{erfc}\Big(\sqrt{R\,\frac{E_s}{\mathcal{N}_0}}\Big)\,,$$

where $R = k/n$ is the rate of the code.

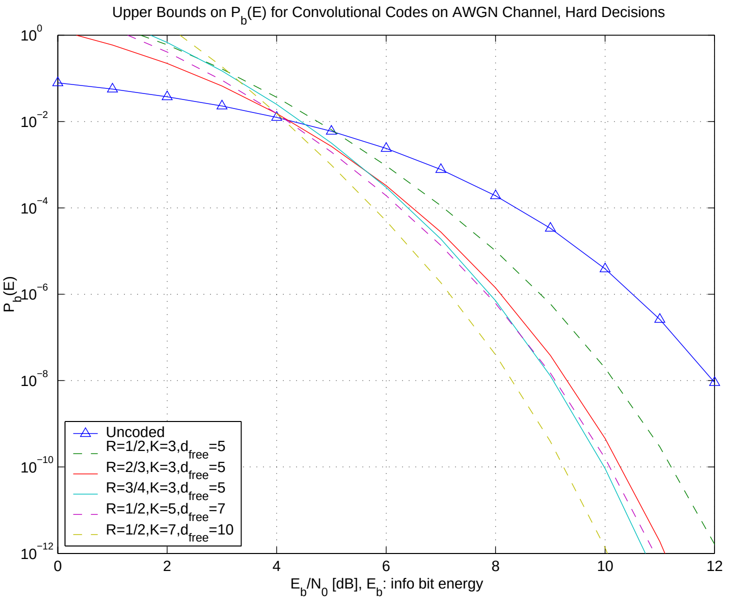
The figure on the next slide compares $P_b(\mathcal{E})$ versus $E_b/\mathcal{N}_0$ for an uncoded and a coded binary system. The coded system uses a $R = 1/2$ $K = 3$ convolutional encoder with $\mathbf{G}(D) = \begin{bmatrix} 1 + D^2 & 1 + D + D^2 \end{bmatrix}$.


Binary R=1/2, K=3, d_free=5, Convolutional Code, Hard decisions AWGN channel
P_b(E) uncoded
P_b(E) union bound
P_b(E) Bhattacharyya
P_b(E)
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
0
2
4
6
8
10
12
E_b/N_0 [dB], E_b: info bit energy

Definition: **Coding Gain.** Coding gain is defined as the reduction in $E_s/\mathcal{N}_0$ permissible for a coded communication system to obtain the same probability of error ($P_s(\mathcal{E})$ or $P_B(\mathcal{E})$ as an uncoded system, both using the same average energy per transmitted information symbol.

Definition: **Coding Threshold.** The value of $E_s/\mathcal{N}_0$ (where E_s is the energy per transmitted information symbol) for which the coding gain becomes zero is called the coding threshold.

The graphs on the following slide show $P_b(\mathcal{E})$ (computed using the union bound) versus $E_b/\mathcal{N}_0$ for a number of different binary convolutional encoders.


Upper Bounds on $P_b(E)$ for Convolutional Codes on AWGN Channel, Hard Decisions
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
$P_b(E)$
0
2
4
6
8
10
12
E_b/N_0 [dB], E_b: info bit energy
Uncoded
$R=1/2, K=3, d_{free}=5$
$R=2/3, K=3, d_{free}=5$
$R=3/4, K=3, d_{free}=5$
$R=1/2, K=5, d_{free}=7$
$R=1/2, K=7, d_{free}=10$

Soft Decisions and AWGN Channel

Assuming a memoryless channel model used without feedback, the ML decoding rule after the MF and the sampler is: Output code sequence estimate $\hat{\mathbf{c}} = \mathbf{c}_i$ iff i maximizes

$$f_{\mathbf{b}}(\beta|\mathbf{a}{=}\mathbf{c}_i) = \prod_{j=0}^{N-1} f_{b_j}(\beta_j|a_j{=}c_{ij}) \,,$$

over all code sequences $\mathbf{c}_i = (c_{i0}, c_{i1}, c_{i2}, \ldots)$ for $i = 0, 1, 2, \ldots$.

If the mapping $0 \rightarrow -1$ and $1 \rightarrow +1$ is used so that $c_{ij} \in \{-1, +1\}$ then $f_{b_j}(\beta_j|a_j{=}c_{ij})$ can be written as

$$f_{b_j}(\beta_j|a_j{=}c_{ij}) = \frac{e^{-(\beta_j - c_{ij}\sqrt{E_s})^2/\mathcal{N}_0}}{\sqrt{\pi\mathcal{N}_0}} \,.$$

Taking (natural) logarithms and defining $v_j = \beta_j/\sqrt{E_s}$ yields

$$\begin{aligned}
\ln f_{\mathbf{b}}(\beta|\mathbf{a}{=}\mathbf{c}_i) &= \ln \prod_{j=0}^{N-1} f_{b_j}(\beta_j|a_j{=}c_{ij}) = \sum_{j=0}^{N-1} \ln f_{b_j}(\beta_j|a_j{=}c_{ij}) \\
&= -\sum_{j=0}^{N-1} \frac{(\beta_j - c_{ij}\sqrt{E_s})^2}{\mathcal{N}_0} - \frac{N}{2}\ln(\pi\mathcal{N}_0) \\
&= -\frac{E_s}{\mathcal{N}_0}\sum_{j=0}^{N-1}(v_j^2 - 2\,v_j\,c_{ij} + c_{ij}^2) - \frac{N}{2}\ln(\pi\mathcal{N}_0) \\
&= \frac{2E_s}{\mathcal{N}_0}\sum_{j=0}^{N-1} v_j\,c_{ij} - \left(\frac{|\beta|^2 + NE_s}{\mathcal{N}_0} + \frac{N}{2}\ln(\pi\mathcal{N}_0)\right) \\
&= K_1 \sum_{j=0}^{N-1} v_j\,c_{ij} - K_2\,,
\end{aligned}$$

where K_1 and K_2 are constants independent of the codeword $\mathbf{c}_i$ and thus irrelevant for ML decoding.

Example: Suppose the convolutional encoder with $\mathbf{G}(D) = \begin{bmatrix} 1 & 1+D \end{bmatrix}$ is used and the received data is

$$\mathbf{v} = \text{-}0.4, \text{-}1.7,\ \ 0.1, 0.3,\ \ \text{-}1.1, 1.2,\ \ 1.2, 0.0,\ \ 0.3, 0.2,\ \ \text{-}0.2, 0.7,\ \ \ldots$$

Soft Decisions versus Hard Decisions

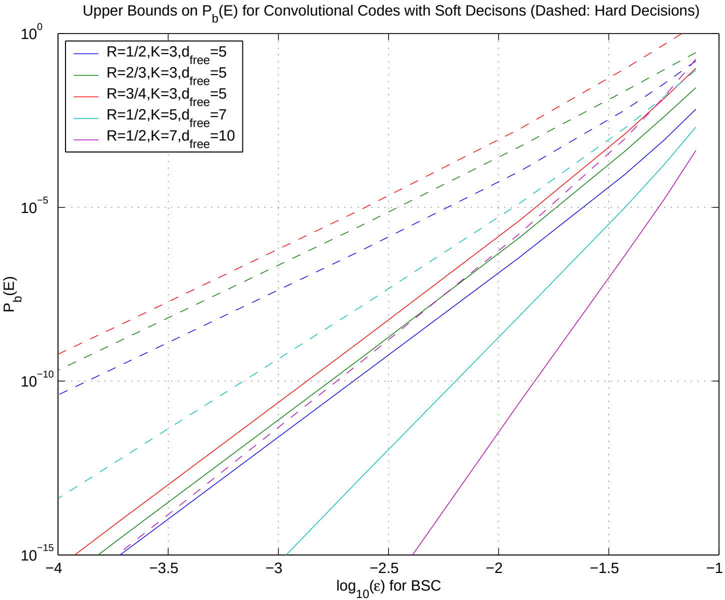
To compare the performance of coded binary systems on a AWGN channel when the decoder performs either hard or soft decisions, the energy E_c per coded bit is fixed and $P_b(\mathcal{E})$ is plotted versus ϵ of the hard decision BSC model where $\epsilon = \frac{1}{2}\text{erfc}\big(\sqrt{E_c/\mathcal{N}_0}\big)$ as before. For soft decisions the expression

$$P_w(\mathcal{E}) = \frac{1}{2}\,\text{erfc}\Big(\sqrt{\frac{w\,E_c}{\mathcal{N}_0}}\Big)$$

is used for the probability that the ML decoder makes a detour with weight w from the correct path. Thus, for soft decisions with fixed SNR per code symbol

$$P_b(\mathcal{E}) \le \frac{1}{2k} \sum_{w=d_{free}}^{\infty} D_w\,\text{erfc}\Big(\sqrt{\frac{w\,E_c}{\mathcal{N}_0}}\Big)\,.$$

Examples are shown on the next slide.



Upper Bounds on P_b(E) for Convolutional Codes with Soft Decisons (Dashed: Hard Decisions)
R=1/2,K=3,d_free=5
R=2/3,K=3,d_free=5
R=3/4,K=3,d_free=5
R=1/2,K=5,d_free=7
R=1/2,K=7,d_free=10
10^0
10^−5
10^−10
10^−15
P_b(E)
−4
−3.5
−3
−2.5
−2
−1.5
−1
log_10(ε) for BSC

Coding Gain for Soft Decisions

To compare the performance of uncoded and coded binary systems with soft decisions on a AWGN channel, the energy E_b per information bit is fixed and $P_b(\mathcal{E})$ is plotted versus the signal-to-noise ratio (SNR) $E_b/\mathcal{N}_0$. For an uncoded system

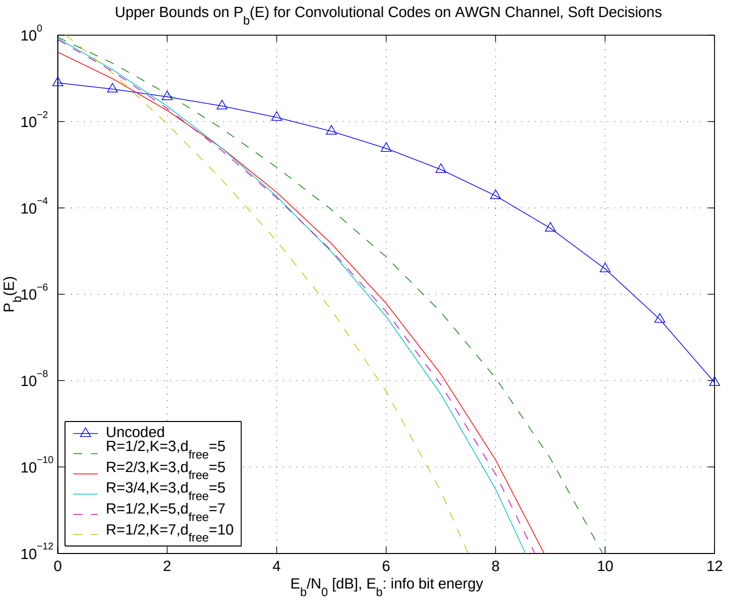
$$P_b(\mathcal{E}) = \frac{1}{2}\,\mathrm{erfc}\Big(\sqrt{\frac{E_b}{\mathcal{N}_0}}\Big)\,, \qquad \text{(uncoded)}\,.$$

For a coded system with soft decision ML decoding on a AWGN channel

$$P_b(\mathcal{E}) \le \frac{1}{2k} \sum_{w=d_{free}}^{\infty} D_w\,\mathrm{erfc}\Big(\sqrt{\frac{wR\,E_b}{\mathcal{N}_0}}\Big)\,,$$

where $R = k/n$ is the rate of the code.

Examples are shown in the graph on the next slide.


Upper Bounds on P_b(E) for Convolutional Codes on AWGN Channel, Soft Decisions
P_b(E)
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
0
2
4
6
8
10
12
E_b/N_0 [dB], E_b: info bit energy
Uncoded
R=1/2,K=3,d_free=5
R=2/3,K=3,d_free=5
R=3/4,K=3,d_free=5
R=1/2,K=5,d_free=7
R=1/2,K=7,d_free=10