Performance Comparison of Short-Length Error-Correcting Codes

Joseph J. Boutros
Talk at Nokia Bell Labs, Stuttgart

Texas A&M University at Qatar

Collaborators: J. Van Wonterghem and M. Moeneclay (Univ-Ghent), and Amira Alloum (Nokia Paris).

March 6, 2017

Thanks

Many Thanks to Dr Laurent Schmalen and his group for this invitation.

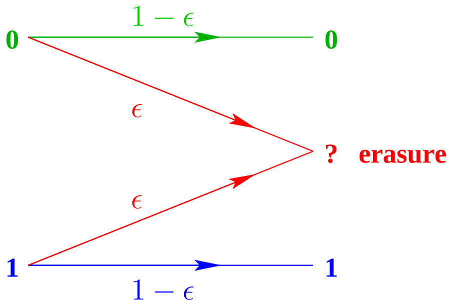
Outline of my talk

- Maximum-Likelihood (ML) Decoding over the BEC
- OSD Decoding over the BI-AWGN
- List of Codes for Short-Length Error Correction
- Performance Results

$\rightarrow$ Researchers and Engineers from all fields are welcome.

The Binary Erasure Channel (BEC)

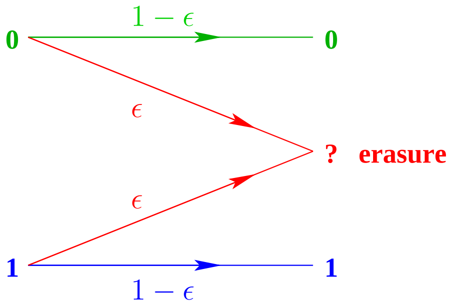
We consider the ergodic binary erasure channel (BEC). The channel input is binary and its output is ternary. Only erasures occur on this channel, no errors.



- Shannon capacity of the BEC is $C = 1 - \epsilon$ bits per channel use.
- Rate 1/2 code $\rightarrow$ $\epsilon_{max} = 1/2$. Rate 1/4 code $\rightarrow$ $\epsilon_{max} = 3/4$.

The Binary Erasure Channel (BEC)

We consider the ergodic binary erasure channel (BEC). The channel input is binary and its output is ternary. Only erasures occur on this channel, no errors.



- Shannon capacity of the BEC is $C = 1 - \epsilon$ bits per channel use.
- Rate 1/2 code $\rightarrow$ $\epsilon_{max} = 1/2$. Rate 1/4 code $\rightarrow$ $\epsilon_{max} = 3/4$.

Maximum-Likelihood Decoding over the BEC (1)

- Linear binary code $C[n,k,d]_2$ of length n, dimension k, rate $R = k/n$, and minimum Hamming distance d.
- Let G be a generator matrix of C, G is $k \times n$.
- The source vector $b = (b_1, ..., b_k) \in \mathbb{F}_2^k$.
- A codeword of C is obtained by $c = (c_1, ..., c_n) = bG \in \mathbb{F}_2^n$.
- There exists a generator matrix in systematic form $G = [I_k|P]$, in this case $c = [b \mid p]$.
- Let H be a parity-check matrix of C, H is $(n-k) \times n$.
- Any codeword of C satisfies the constraint $Hc^t = 0$, i.e. $n-k$ parity-check equations.

Maximum-Likelihood Decoding over the BEC (1)

- Linear binary code $C[n,k,d]_2$ of length n, dimension k, rate $R = k/n$, and minimum Hamming distance d.
- Let G be a generator matrix of C, G is $k \times n$.
- The source vector $b = (b_1, ..., b_k) \in \mathbb{F}_2^k$.
- A codeword of C is obtained by $c = (c_1, ..., c_n) = bG \in \mathbb{F}_2^n$.
- There exists a generator matrix in systematic form $G = [I_k|P]$, in this case $c = [b \mid p]$.
- Let H be a parity-check matrix of C, H is $(n-k) \times n$.
- Any codeword of C satisfies the constraint $Hc^t = 0$, i.e. $n-k$ parity-check equations.

Maximum-Likelihood Decoding over the BEC (1)

- Linear binary code $C[n,k,d]_2$ of length n, dimension k, rate $R = k/n$, and minimum Hamming distance d.
- Let G be a generator matrix of C, G is $k \times n$.
- The source vector $b = (b_1, ..., b_k) \in \mathbb{F}_2^k$.
- A codeword of C is obtained by $c = (c_1, ..., c_n) = bG \in \mathbb{F}_2^n$.
- There exists a generator matrix in systematic form $G = [I_k|P]$, in this case $c = [b \mid p]$.
- Let H be a parity-check matrix of C, H is $(n-k) \times n$.
- Any codeword of C satisfies the constraint $Hc^t = 0$, i.e. $n-k$ parity-check equations.

Maximum-Likelihood Decoding over the BEC (2)

Example: Extended-Shortened BCH code $[n = 10, k = 5, d = 4]_2$.
The code has $|C| = 2^k = 32$ codewords each of length $n = 10$ bits.

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

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \mathbf{1} \end{pmatrix}$$

$b = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ )$ then $c = bG = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ )$.
Check $Hc^t = (\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ )^t$.

Maximum-Likelihood Decoding over the BEC (2)

- Let $y = (y_1 \ldots y_n)$ be the channel output. ML Decoding:

$$\hat{c} = \arg \max_c P(y|c) \quad \text{and} \quad \max_c P(y|c) = \epsilon^w (1-\epsilon)^{n-w},$$

where w is the Hamming weight of the erasure pattern.

- The ML decoder should find a unique codeword that matches the $n - w$ non-erased bits y_i.

- This codeword is solution of $Hc^t = 0$.
The decoder uses the $n - k$ parity-check equations in H to solve c.
ML Decoding over the BEC $\Longleftrightarrow$ Gaussian Elimination of H.

- If $w \leq d - 1$, all erased bits will be filled, whatever are the w positions.

- If $d \leq w \leq n - k$ (non-MDS code), erased bits may be solved for some erasure patterns.

Maximum-Likelihood Decoding over the BEC (2)

- Let $y = (y_1 \dots y_n)$ be the channel output. ML Decoding:

$$\hat{c} = \arg\max_{c} P(y|c) \quad \text{and} \quad \max_{c} P(y|c) = \epsilon^{w}(1-\epsilon)^{n-w},$$

  where w is the Hamming weight of the erasure pattern.

- The ML decoder should find a unique codeword that matches the $n - w$ non-erased bits y_i.

- This codeword is solution of $Hc^t = 0$.
  The decoder uses the $n - k$ parity-check equations in H to solve c.
  ML Decoding over the BEC $\Longleftrightarrow$ Gaussian Elimination of H.

- If $w \leq d - 1$, all erased bits will be filled, whatever are the w positions.

- If $d \leq w \leq n - k$ (non-MDS code), erased bits may be solved for some erasure patterns.

Maximum-Likelihood Decoding over the BEC (3)

- ML decoding via Gaussian elimination has an affordable complexity, at least in software applications, for a code length n as high as a thousand bits.
- The cost of solving $H\boldsymbol{c}^t = 0$ is $O(n \times (n-k)^2)$.
- Results shown at the end of this talk are obtained for a short length $n = 256$.

The Binary-Input Additive White Gaussian Noise (BI-AWGN) Channel

- A codeword c in $\mathbb{F}_2^n$ is mapped into a codeword in $\{\pm 1\}^n$, i.e. a BPSK symbol sequence $s = s(c)$, where $s_i = 2c_i - 1$, for $i = 1 \ldots n$.
- The BI-AWGN channel output is $r = s + \eta$, where $r \in \mathbb{R}^n$ and $\eta_i \sim \mathcal{N}(0, \sigma^2)$.
- Take noise variance $\sigma^2 = \frac{N_0}{2}$ and energy per bit $E_b = \frac{n}{k}$. The channel parameter is the signal-to-noise ratio E_b/N_0.

Maximum-Likelihood Decoding, known as Soft-Decision Decoding:

- The likelihood $P(r|s)$ is proportional to $\exp\left(-\frac{\|r-s\|^2}{2\sigma^2}\right)$ then

$$\hat{c} = \arg\max_c P(r|s(c)) \iff \min_c \|r - s(c)\|^2 \iff \max_c \langle r, s(c)\rangle.$$

- The cost of exhaustive decoding is 2^k metric computations!
- Near-ML reduced-complexity decoding: Ordered Statistics Decoding (OSD).

The Binary-Input Additive White Gaussian Noise (BI-AWGN) Channel

- A codeword c in $\mathbb{F}_2^n$ is mapped into a codeword in $\{\pm 1\}^n$, i.e. a BPSK symbol sequence $s = s(c)$, where $s_i = 2c_i - 1$, for $i = 1 \ldots n$.
- The BI-AWGN channel output is $r = s + \eta$, where $r \in \mathbb{R}^n$ and $\eta_i \sim \mathcal{N}(0, \sigma^2)$.
- Take noise variance $\sigma^2 = \frac{N_0}{2}$ and energy per bit $E_b = \frac{n}{k}$.
  The channel parameter is the signal-to-noise ratio E_b/N_0.

Maximum-Likelihood Decoding, known as Soft-Decision Decoding:

- The likelihood $P(r|s)$ is proportional to $\exp\left(-\frac{\|r-s\|^2}{2\sigma^2}\right)$ then

$$\hat{c} = \arg\max_c P(r|s(c)) \iff \min_c \|r - s(c)\|^2 \iff \max_c \langle r, s(c) \rangle.$$

- The cost of exhaustive decoding is 2^k metric computations!
- Near-ML reduced-complexity decoding: Ordered Statistics Decoding (OSD).

OSD Decoding over the BI-AWGN (1)

- The OSD algorithm: an efficient most reliable basis (MRB) decoding algorithm.
- Firstly proposed by **Dorsch** in 1974.
- Further developed by **Fang and Battail** in 1987.
- Analyzed and revived by **Fossorier and Lin** in 1995.
- Improvements to the original OSD algorithm by **Wu and Hadjicostis** in 2007.
- Our OSD implementation is based on several complexity-reduction rules, **Van Wonterghem, Alloum, Boutros, and Moeneclaey** 2016.

OSD Decoding over the BI-AWGN (2)

- For a given channel output at discrete time i, $i = 1 \dots n$, the log-likelihood ratio is

$$\log \frac{P(r_i|c_i = 0)}{P(r_i|c_i = 1)} = \frac{2}{\sigma^2} \times r_i.$$

- The hard decision yields $y = [\ b_{\mathsf{HD}} \mid p_{\mathsf{HD}}\ ]$ where

$$y_i = \begin{cases} 0 & \text{for } r_i < 0 \\ 1 & \text{for } r_i > 0 \end{cases}$$

- The confidence value of a received bit is

$$\alpha_i = |r_i|, \quad i = 1 \dots n.$$

- The OSD decoder input is:
  - The n bits y_i found by hard decision.
  - The n confidence values $\alpha_i = |r_i|$.

The OSD does not need to know the channel noise variance σ^2.

OSD Decoding over the BI-AWGN (2)

- For a given channel output at discrete time i, $i = 1 \ldots n$, the log-likelihood ratio is
$$\log \frac{P(r_i|c_i = 0)}{P(r_i|c_i = 1)} = \frac{2}{\sigma^2} \times r_i.$$
- The hard decision yields $y = [\ b_{\text{HD}} \mid p_{\text{HD}}\ ]$ where
$$y_i = \begin{cases} 0 & \text{for } r_i < 0 \\ 1 & \text{for } r_i > 0 \end{cases}$$
- The confidence value of a received bit is
$$\alpha_i = |r_i|\,, \quad i = 1 \ldots n.$$
- The OSD decoder input is:
  - The n bits y_i found by hard decision.
  - The n confidence values $\alpha_i = |r_i|$.

The OSD does not need to know the channel noise variance σ^2.

OSD Decoding over the BI-AWGN: order 0

$$G = \begin{array}{c} \begin{array}{cccccccccc} \color{blue}{1} & \color{blue}{2} & \color{blue}{3} & \color{blue}{4} & \color{blue}{5} & \color{blue}{6} & \color{blue}{7} & \color{blue}{8} & \color{blue}{9} & \color{blue}{10} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

Codeword in $\mathbb{F}_2$:
$c = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ )$

Bipolar codeword:
$s = (\ +1\ \ -1\ \ \color{red}{-1}\ \ +1\ \ +1\ \ +1\ \ +1\ \ \color{red}{-1}\ \ -1\ \ +1\ )$

Received noisy word:
$r = (\ +1.91\ \ -2.64\ \ \color{red}{+0.54}\ \ +1.13\ \ +1.23\ \ +1.54\ \ +0.56\ \ \color{red}{+0.20}\ \ -0.17\ \ +1.24\ )$

Confidence values and detected word via threshold detection:
$\alpha = (\ 1.91\ \ 2.64\ \ \color{red}{0.54}\ \ 1.13\ \ 1.23\ \ 1.54\ \ 0.56\ \ \color{red}{0.20}\ \ 0.17\ \ 1.24\ )$
$y = (\ 1\ \ 0\ \ \color{red}{1}\ \ 1\ \ 1\ \ 1\ \ 1\ \ \color{red}{1}\ \ 0\ \ 1\ )$

OSD Decoding over the BI-AWGN: order 0

$$G = \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left(\begin{array}{cccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}\right) \end{array}$$

Codeword in $\mathbb{F}_2$:
$c = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ )$

Bipolar codeword:
$s = (\ +1\ \ -1\ \ -1\ \ +1\ \ +1\ \ +1\ \ +1\ \ -1\ \ -1\ \ +1\ )$

Received noisy word:
$r = (\ +1.91\ \ -2.64\ \ +0.54\ \ +1.13\ \ +1.23\ \ +1.54\ \ +0.56\ \ +0.20\ \ -0.17\ \ +1.24\ )$

Confidence values and detected word via threshold detection:
$\alpha = (\ 1.91\ \ 2.64\ \ 0.54\ \ 1.13\ \ 1.23\ \ 1.54\ \ 0.56\ \ 0.20\ \ 0.17\ \ 1.24\ )$
$y = (\ 1\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1\ )$

Sorted confidence values:
$\pi_1(\alpha) = (\ 2.64\ \ 1.91\ \ 1.54\ \ 1.24\ \ 1.23\ \ 1.13\ \ 0.56\ \ 0.54\ \ 0.20\ \ 0.17\ )$

OSD Decoding over the BI-AWGN: order 0

$$G' = \begin{array}{c} \begin{array}{cccccccccc} 2 & 1 & 6 & 10 & 5 & 4 & 7 & 3 & 8 & 9 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Codeword in $\mathbb{F}_2$:
$c = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ )$

Bipolar codeword:
$s = (\ +1\ \ -1\ \ -1\ \ +1\ \ +1\ \ +1\ \ +1\ \ -1\ \ -1\ \ +1\ )$

Received noisy word:
$r = (\ +1.91\ \ -2.64\ \ +0.54\ \ +1.13\ \ +0.17\ \ +1.54\ \ +0.56\ \ +0.20\ \ -1.23\ \ +1.24\ )$

Sorted confidence values:
$\pi_1(\alpha) = (\ 2.64\ \ 1.91\ \ 1.54\ \ 1.24\ \ 1.23\ \ 1.13\ \ 0.56\ \ 0.54\ \ 0.20\ \ 0.17\ )$

Sorted threshold detection:
$\pi_1(y) = (\ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ )$

OSD Decoding over the BI-AWGN: order 0

$$\tilde{G} = \begin{array}{c} \begin{array}{cccccccccc} 2 & 1 & 6 & 10 & 5 & 4 & 7 & 3 & 8 & 9 \end{array} \\ \left(\begin{array}{cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{array} \right) \end{array}$$

Codeword in $\mathbb{F}_2$:

$c = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ )$

Bipolar codeword:

$s = (\ +1\ \ -1\ \ -1\ \ +1\ \ +1\ \ +1\ \ +1\ \ -1\ \ -1\ \ +1\ )$

Received noisy word:

$r = (\ +1.91\ \ -2.64\ \ +0.54\ \ +1.13\ \ +0.17\ \ +1.54\ \ +0.56\ \ +0.20\ \ -1.23\ \ +1.24\ )$

Sorted confidence values:

$\pi_1(\alpha) = (\ 2.64\ \ 1.91\ \ 1.54\ \ 1.24\ \ 1.23\ \ 1.13\ \ 0.56\ \ 0.54\ \ 0.20\ \ 0.17\ )$

Sorted threshold detection:

$\pi_1(y) = (\ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ )$

OSD Decoding over the BI-AWGN: order 0

$$\begin{array}{c} \\ \tilde{G} = \end{array} \begin{array}{c} \begin{matrix} 2 & 1 & 6 & 10 & 5 & 4 & 7 & 3 & 8 & 9 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Codeword in $\mathbb{F}_2$:
$c = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ )$

Bipolar codeword:
$s = (\ +1\ \ -1\ \ -1\ \ +1\ \ +1\ \ +1\ \ +1\ \ -1\ \ -1\ \ +1\ )$

Sorted threshold detection:
$\pi_1(y) = (\ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ )$

Re-encoding from MRB (the 5 bits on the left, i.e. the most confident):
$\pi_1(\hat{c}) = (\ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ )$

Final codeword (order-0 OSD):
$\hat{c} = (\ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ ) = c$

OSD Decoding over the BI-AWGN: order 1

- Consider the k most confident bits on the left (MRB).
- Flip one bit out of k, i.e. add an error pattern of weight 1.
- This will generate k codeword candidates.
- From order 0 and order 1, now we have $1 + k$ codeword candidates.
- Keep the best candidate according to $\langle r, s(\hat{c}) \rangle$.

OSD Decoding over the BI-AWGN: order 2

- Consider the k most confident bits on the left (MRB).
- Flip two bits out of k, i.e. add an error pattern of weight 2.
- This will generate $\binom{k}{2} = k(k-1)/2$ codeword candidates.
- From order 0, order 1, and order 2, now we have $1 + k + k(k-1)/2$ codeword candidates.
- Keep the best candidate according to $\langle r, s(\hat{c}) \rangle$.

OSD Decoding over the BI-AWGN: order ℓ

- Consider the k most confident bits on the left (MRB).
- Flip ℓ bits out of k, i.e. add an error pattern of weight ℓ.
- This will generate $\binom{k}{\ell}$ codeword candidates.
- From order 0 up to order ℓ, now we have $\sum_{i=0}^{\ell} \binom{k}{i}$ codeword candidates.
- Keep the best candidate according to $\langle r, s(\hat{c})\rangle$.
- The complexity of OSD is $O\left(k^{\ell}\right)$.

The OSD is asymptotically optimal if $\ell \geq \min\{\lceil d/4 - 1\rceil, k\}$ (**Fossorier & Lin 1995**). The OSD order is taken to be much smaller when improvement rules are applied, e.g. skipping rule based on weighted Hamming distance or the use of multiple MRB.

List of Codes for Short-Length Error Correction

- Reed-Muller codes: The code length is $n = 2^\ell$. Take Arikan's kernel G_2 (**Arikan 2008**) and build its Kronecker product ℓ times, i.e. build $G_2^{\otimes \ell}$. Select the k rows of largest Hamming weight to get the $k \times n$ gen. matrix.

- Polar codes: As for Reed-Muller codes, k rows are selected from $G_2^{\otimes \ell}$. These rows correspond to highest mutual information channels after ℓ splittings. We used Density Evolution for the BI-AWGN channel.

- BCH codes: Standard binary primitive (n, k, t) BCH codes are built from their generator polynomial (**Blahut 2003**). An extension by one parity bit is made to get an even length.

- LDPC codes: Regular (3,6) low-density parity-check codes are built from a random bipartite Tanner graph (**Richardson & Urbanke 2008**). Length-2 cycles are avoided, the number of length-4 cycles is reduced, but no other constraint was applied to the graph construction.

List of Codes for Short-Length Error Correction

- Reed-Muller codes: The code length is $n = 2^\ell$. Take Arikan's kernel G_2 (**Arikan 2008**) and build its Kronecker product ℓ times, i.e. build $G_2^{\otimes \ell}$. Select the k rows of largest Hamming weight to get the $k \times n$ gen. matrix.

- Polar codes: As for Reed-Muller codes, k rows are selected from $G_2^{\otimes \ell}$. These rows correspond to highest mutual information channels after ℓ splittings. We used Density Evolution for the BI-AWGN channel.

- BCH codes: Standard binary primitive (n, k, t) BCH codes are built from their generator polynomial (**Blahut 2003**). An extension by one parity bit is made to get an even length.

- LDPC codes: Regular (3,6) low-density parity-check codes are built from a random bipartite Tanner graph (**Richardson & Urbanke 2008**). Length-2 cycles are avoided, the number of length-4 cycles is reduced, but no other constraint was applied to the graph construction.

List of Codes for Short-Length Error Correction

- Reed-Muller codes: The code length is $n = 2^\ell$. Take Arikan's kernel G_2 (**Arikan 2008**) and build its Kronecker product ℓ times, i.e. build $G_2^{\otimes \ell}$. Select the k rows of largest Hamming weight to get the $k \times n$ gen. matrix.

- Polar codes: As for Reed-Muller codes, k rows are selected from $G_2^{\otimes \ell}$. These rows correspond to highest mutual information channels after ℓ splittings. We used Density Evolution for the BI-AWGN channel.

- BCH codes: Standard binary primitive (n, k, t) BCH codes are built from their generator polynomial (**Blahut 2003**). An extension by one parity bit is made to get an even length.

- LDPC codes: Regular (3,6) low-density parity-check codes are built from a random bipartite Tanner graph (**Richardson & Urbanke 2008**). Length-2 cycles are avoided, the number of length-4 cycles is reduced, but no other constraint was applied to the graph construction.

List of Codes for Short-Length Error Correction

- Reed-Muller codes: The code length is $n = 2^\ell$. Take Arikan's kernel G_2 (**Arikan 2008**) and build its Kronecker product ℓ times, i.e. build $G_2^{\otimes \ell}$. Select the k rows of largest Hamming weight to get the $k \times n$ gen. matrix.

- Polar codes: As for Reed-Muller codes, k rows are selected from $G_2^{\otimes \ell}$. These rows correspond to highest mutual information channels after ℓ splittings. We used Density Evolution for the BI-AWGN channel.

- BCH codes: Standard binary primitive (n, k, t) BCH codes are built from their generator polynomial (**Blahut 2003**). An extension by one parity bit is made to get an even length.

- LDPC codes: Regular (3,6) low-density parity-check codes are built from a random bipartite Tanner graph (**Richardson & Urbanke 2008**). Length-2 cycles are avoided, the number of length-4 cycles is reduced, but no other constraint was applied to the graph construction.

Joint Decoding of Codes with CRC

I. Tal and A. Vardy (2011):
Cyclic redundancy check (CRC) code to improve list decoding of polar codes.

Our Approach:

- Let G be the $k \times n$ generator matrix of C.
- Let G_{CRC} be the $(k-m) \times k$ generator matrix of the CRC code.
- Joint OSD decoding is based on the following generator matrix:

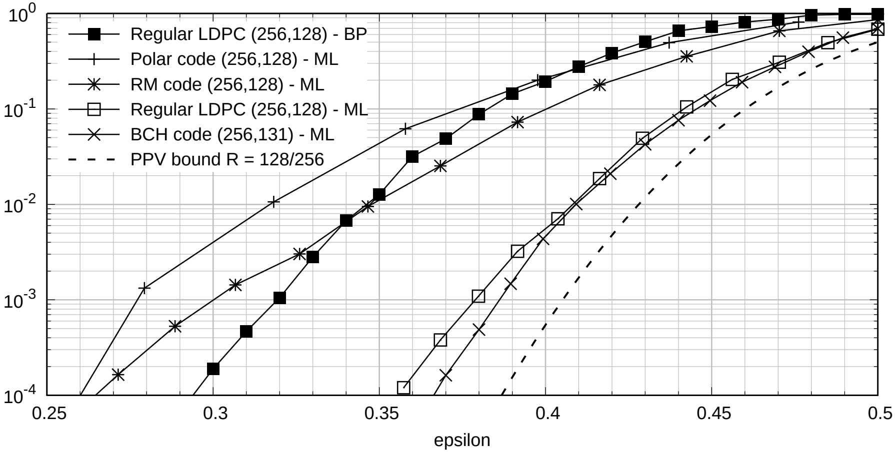
$$G_{CRC} \times G.$$

- We considered $m = 16$ redundancy bits and the CRC-CCITT code with generator polynomial

$$g(x) = x^{16} + x^{12} + x^5 + 1.$$

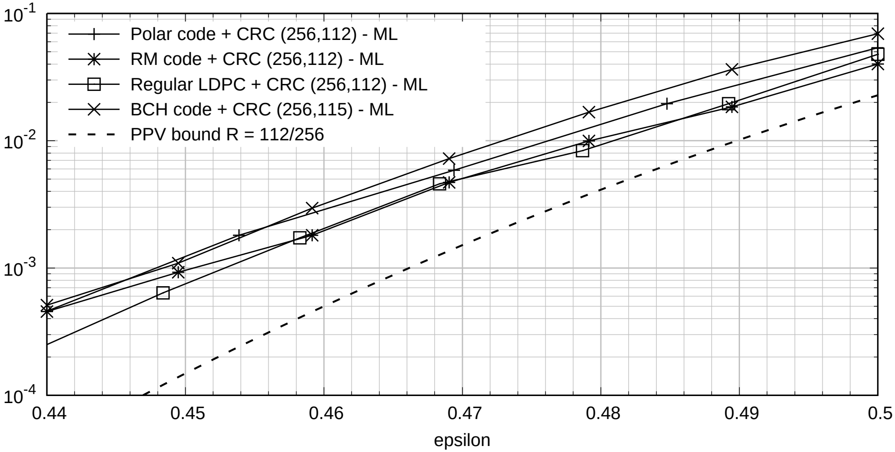
- The CRC will scramble the original matrix G making any code C look like a random code.

Linear Binary Codes over the BEC, No CRC



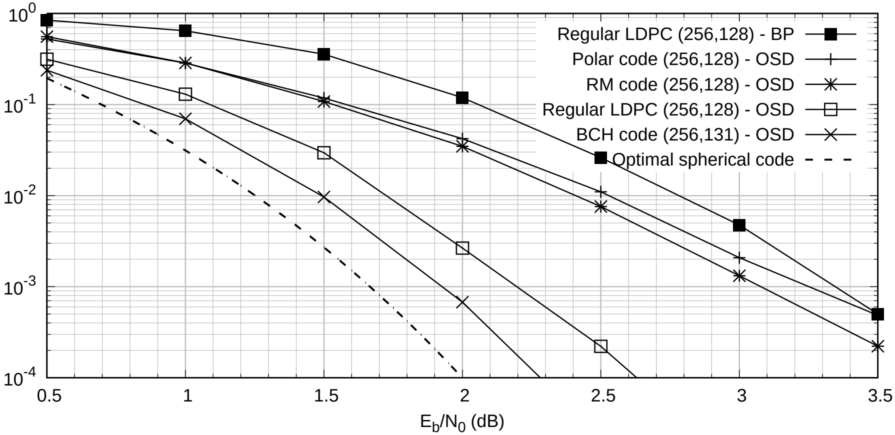
Van Wonterghem, Alloum, Boutros, Moeneclaey, 2016

Linear Binary Codes over the BEC, 16-bit CRC



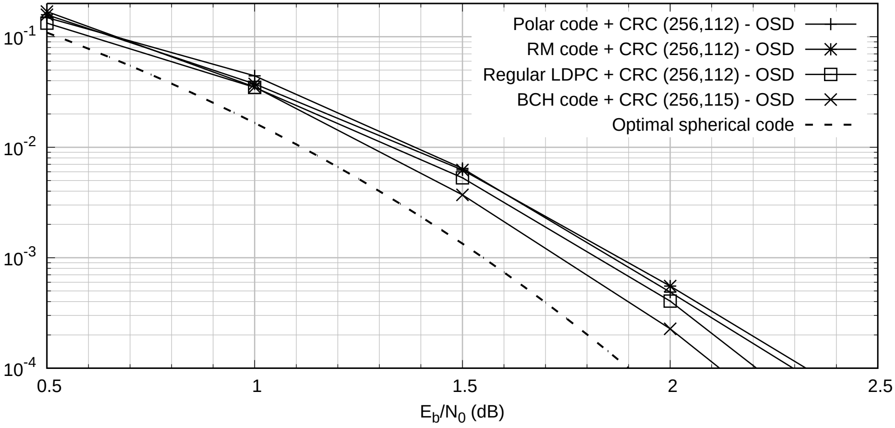
Van Wonterghem, Alloum, Boutros, Moeneclaey, 2016

Linear Binary Codes over the BI-AWGN, No CRC



Van Wonterghem, Alloum, Boutros, Moeneclaey, 2016

Linear Binary Codes over the BI-AWGN, 16-bit CRC

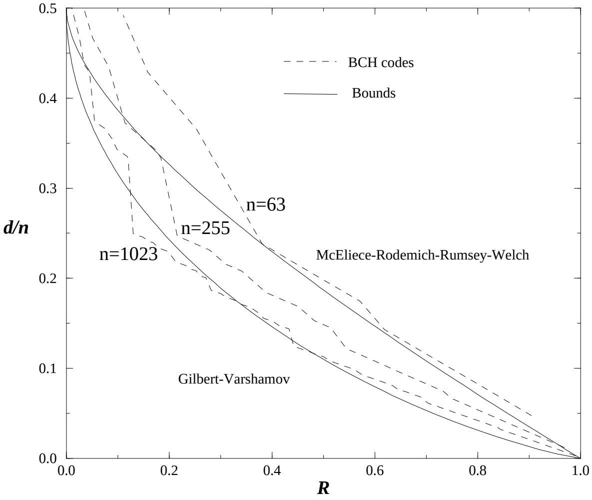


Van Wonterghem, Alloum, Boutros, Moeneclaey, 2016

Conclusions

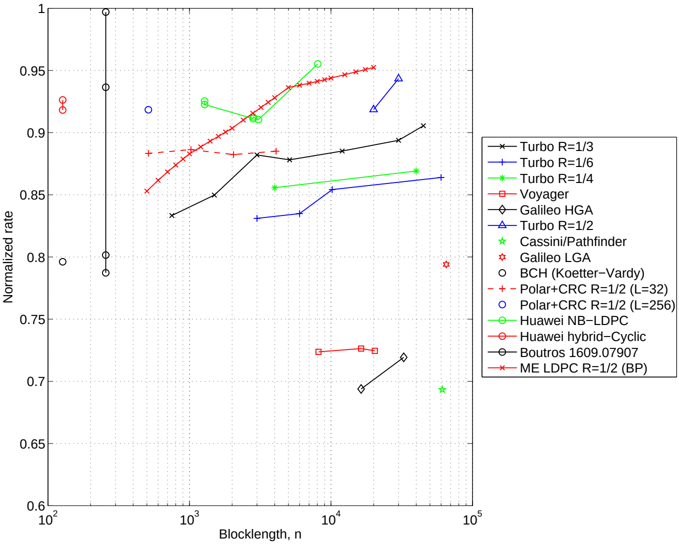
- A universal optimal/near-optimal decoder was used: the ML decoder for the BEC (via Gaussian elimination) and the OSD soft-decision decoder for the binary-input AWGN channel.
- BCH code outperforms Reed-Muller, Polar, and LDPC codes on both channels.
- Under CRC with joint decoding, the different codes lie much closer together and the choice of a good error-correcting code is not so critical.
- More details are found in our paper: "Performance Comparison of Short-Length Error-Correcting Codes", by J. Van Wonterghem, A. Alloum, J.J. Boutros, and M. Moeneclaey, *IEEE SCVT 2016*, Belgium, Nov. 2016.

BCH Minimum Distance versus Bounds



Result from J.J. Boutros, Techniques Modernes de Codage, ENST Paris, 1998

Normalized Rates of Codes over BI-AWGN, $P_e = 10^{-4}$



Result from Yury Polyanskiy, October 2016 (private communication)