Polar Coding

Erdal Arıkan

Electrical-Electronics Engineering Department
Bilkent University
Ankara, Turkey

17 August 2022
2022 North American School of Information Theory
UCLA

Channel coding problem

- Source data $\mathbf{d} = (d_1, \dots, d_K)$
- Code word $\mathbf{x} = (x_1, \dots, x_N)$
- Received word $\mathbf{y} = (y_1, \dots, y_N)$
- Decoded data $\hat{\mathbf{d}} = (\hat{d}_1, \dots, \hat{d}_K)$
- Performance metrics
  - Probability of error $P_e = P(\hat{\mathbf{d}} \neq \mathbf{d})$
  - Code rate $R = K/N$
  - Complexity of implementation

Channel capacity theorem

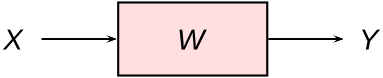
Theorem

For a broad class of channels, there exists a capacity C such that arbitrarily reliable transmission is possible at any fixed rate $R < C$.

- This theorem settles the question about the trade-off between rate and reliability.
- However, it doesn't mention complexity. We need implementable coding schemes.
- Polar codes are provably capacity-achieving with low-complexity encoding and decoding algorithms.

Channel assumptions

In the rest of this talk, we restrict attention to binary-input memoryless channels (BMCs) $W : X \to Y$.



- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$ (assume discrete),
- transition probabilities:

$$W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y}$$

Capacity with uniform channel inputs

- We will consider coding schemes for which the channel input distribution is uniform. So, the maximum achievable rate is given by

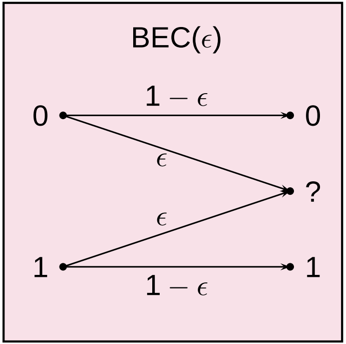
$$I(W) \triangleq I(X;Y)$$

where $I(X;Y)$ is the mutual information with X uniform on $\{0,1\}$.
- If the channel has input-output symmetry, $I(W)$ is the true capacity. Whether the channel is symmetric or not, we will still refer to $I(W)$ as capacity.
- We will use base-2 logarithms so that

$$0 \leq I(W) \leq 1.$$

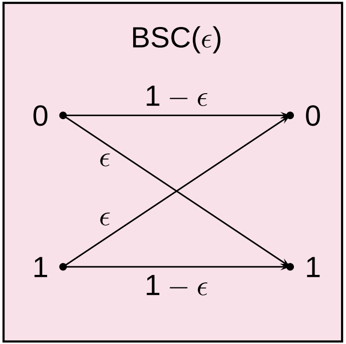
Examples of Binary Memoryless Channels

Binary Erasure Channel



$$I(W) = 1 - \epsilon$$

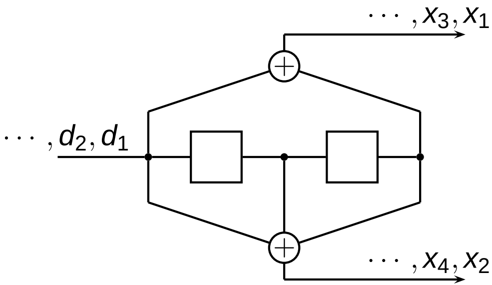
Binary Symmetric Channel



$$I(W) = 1 - H(\epsilon)$$

Convolutional codes (Elias, 1955)

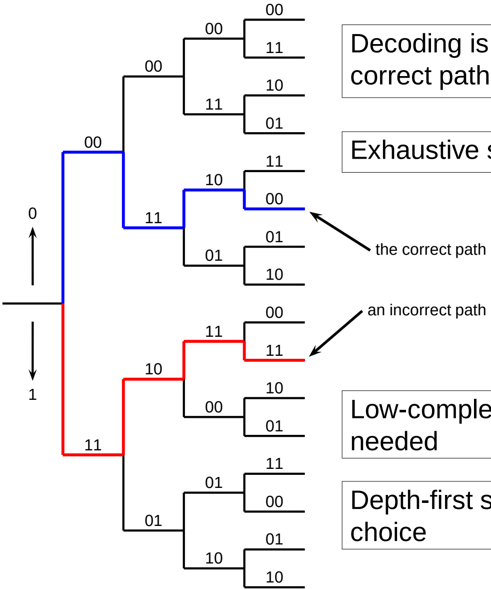
A class of linear codes that achieve channel capacity.



Encoder mapping is $\mathbf{x} = \mathbf{dG}$ where

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

Decoding as a tree-search problem



Decoding is a search problem for the correct path

Exhaustive search too complex

Low-complexity search heuristics needed

Depth-first search is a reasonable choice

Sequential decoding

- Sequential decoding is a depth-first search heuristic for decoding tree codes developed by Wozencraft (1957) that achieves symmetric capacity $I(W)$ if no limit is placed on search complexity
- However, sequential decoding has a computational cutoff rate R_0 barrier in the sense that
  - at rates $R > R_0$, the complexity of decoding the first NR source bits grows as $e^{N(R-R_0)}$
  - at rates $R < R_0$ virtually error free communication is possible at constant average complexity per decoded bit
- Sequential decoding more or less solves the coding problem for $R < R_0$.

The cutoff rate and the Bhattacharyya parameter

Definition

The cutoff rate for a binary-input channel with uniform channel input probabilities is defined as

$$R_0(W) = \log \frac{2}{1 + Z(W)},$$

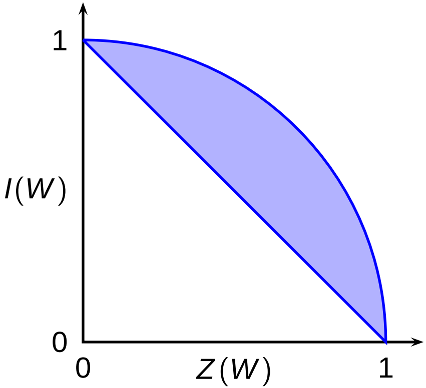
where

$$Z(W) = \sum_y \sqrt{W(y|0)W(y|1)}$$

is the Bhattacharyya parameter.

Some basic inequalities

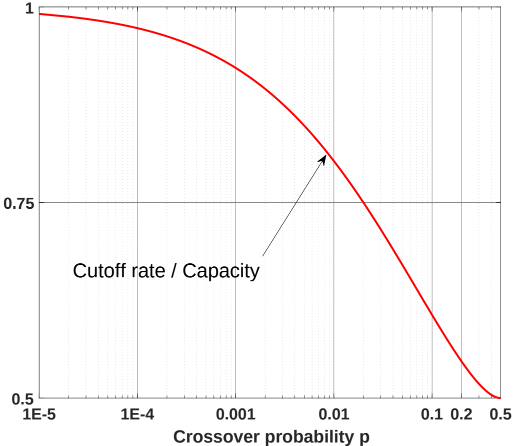
- $0 \leq R_0(W) \leq I(W)$
- $[I(W)]^2 + [Z(W)]^2 \leq 1$
- $I(W) + Z(W) \geq 1$
- These imply that either all three channel parameters are at their extremes or none.
- For proofs, please see Sect. 0.3 of the Supp. Notes.



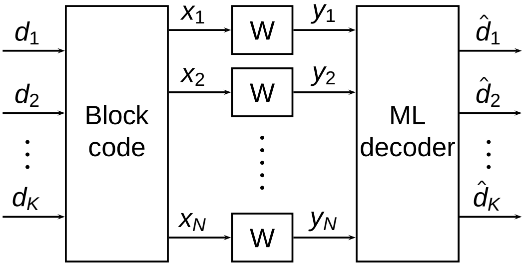
Pinsker's observation (1965)

For the binary symmetric channel (BSC) with crossover probability p,

$$\frac{R_0(p)}{I(p)} = \frac{1 - \log_2\left(1 + 2\sqrt{p(1-p)}\right)}{1 + p\log_2(p) + (1-p)\log_2(1-p)} \to 1 \quad \text{as } p \to 0.$$

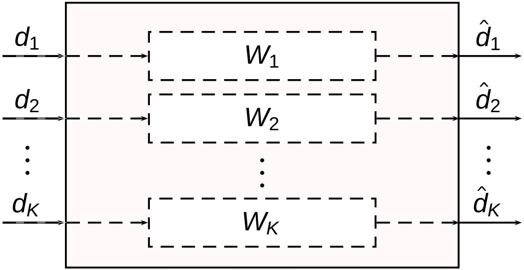


Creating near-perfect bit channels from ordinary channels - 1



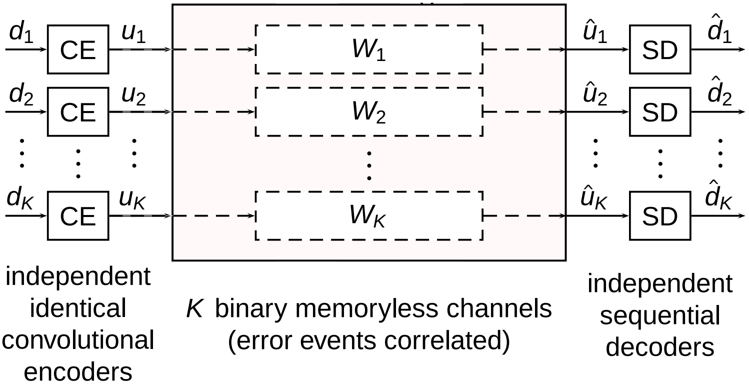
- Accept a data block $\mathbf{d} = (d_1, \ldots, d_K)$ in parallel.
- Encode $\mathbf{d}$ into a code block $\mathbf{x} = (x_1, \ldots, x_N)$ using a block code
- Send $\mathbf{x}$ across N independent copies of W and receive a channel output block $\mathbf{y} = (y_1, \ldots, y_N)$
- Decode $\mathbf{y}$ to obtain an estimate $\hat{\mathbf{d}} = (\hat{d}_1, \ldots, \hat{d}_K$ of $\mathbf{d}$
- For $K/N < I(W)$ and N large enough, a powerful block code may be found so that the probability of block error $P(\hat{\mathbf{d}} \neq \mathbf{d})$ is as small as desired.

Creating near-perfect bit channels from ordinary channels - 2



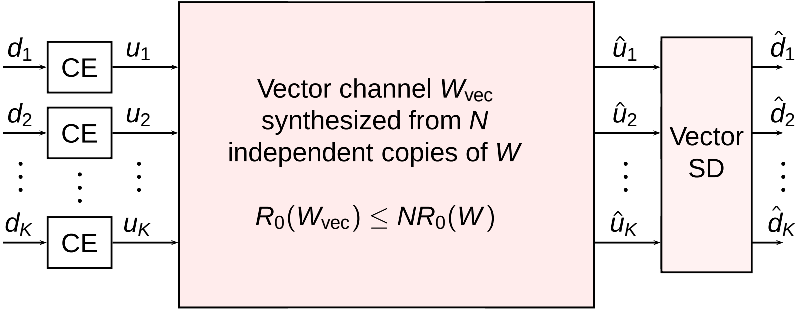
- For each i, the channel $W_i : d_i \to \hat{d}_i$ is a binary-input **memoryless** channel
- The channels $\{W_i\}$ are highly correlated among themselves but each can be encoded/decoded independently
- As long as $K/N < I(W)$, a powerful block code may be used to make each channel W_i near perfect so that $R_0(W_i) \approx 1$.

Exploiting boosted cutoff rates



- Inner code creates K binary memoryless channels (correlated)
- Inner code operates at rate near $I(W)$
- Outer codes operate at rate near $R_0(W_i) \approx 1$
- Boosts cutoff rate from $NR_0(W)$ to near $NI(W)$

A fine point



- A data-processing theorem by Gallager states that $R_0(W_{\text{vec}}) \leq NR_0(W)$
- How do we reconcile this with achieving capacity by sequential decoding?
- $R_0(W_{\text{vec}})$ bound holds if a single vector SD is employed
- Pinsker changes the rules of the game by using multiple SDs each working on one coordinate.

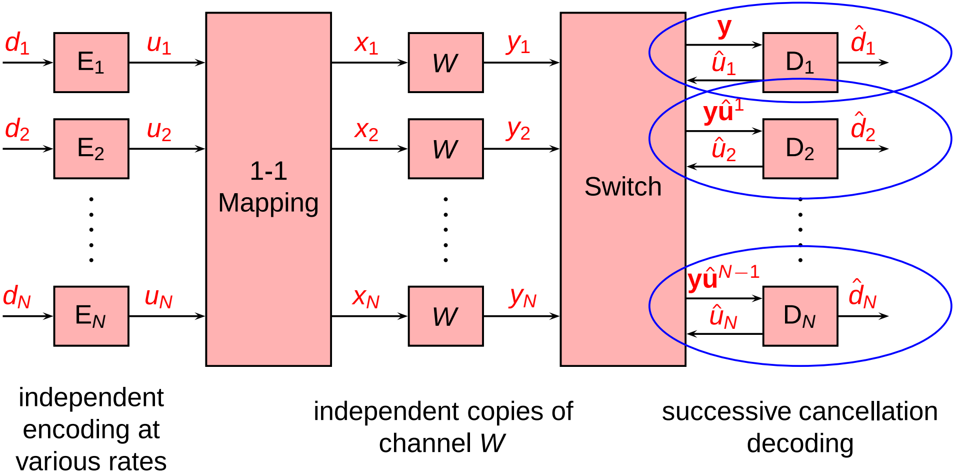
Theoretical significance of Pinsker's result

- Shannon: Arbitrarily reliable data transmission is possible at rates below capacity
- Pinsker: What Shannon says can be done at constant complexity per decoded bit.
- To quote Pinsker exactly: "For a very general class of channels operating below capacity it is possible to construct a code in such a way that the number of operations required for decoding is less than some constant that is independent of the error probability." (PPI, vol.1, no. 1, pp. 113-116, 1965.)

Problem solved?

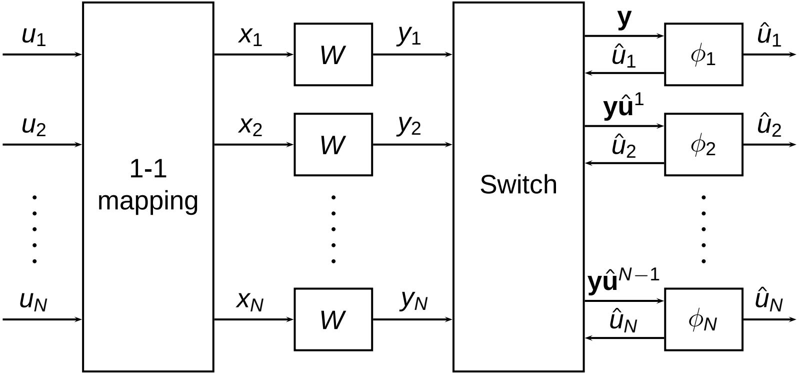
- The constant in Pinsker's scheme is extremely large, so it does not solve the coding problem in any practical sense.
- But it shows that the "cutoff rate barrier" to sequential decoding is not a fundamental one.
- Key take-aways:
  - Synthesize high quality channels from mediocre independent channels without wasting capacity
  - Correlation among synthesized channels is unavoidable but also not a problem
  - Pinsker does not exploit the correlation; can we exploit it?

Multi-level coding (Imai-Hirakawa, 1977)



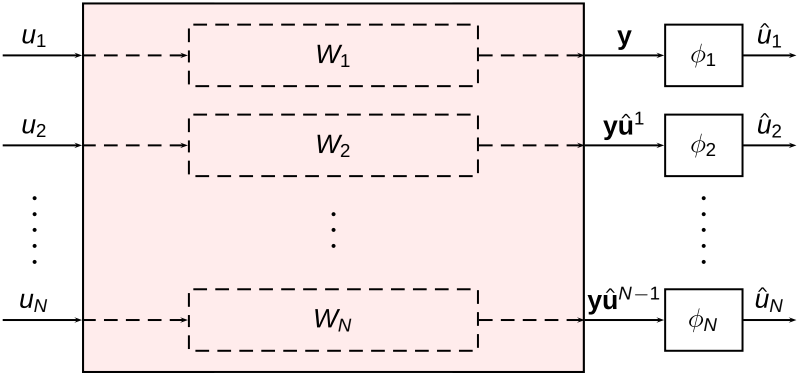
Decoder function $D_i : (\mathbf{y}, \hat{\mathbf{u}}^{i-1}) \to (\hat{d}_i, \hat{u}_i)$ where $\mathbf{y} = (y_1, \ldots, y_N)$, $\hat{\mathbf{u}}^{i-1} = (\hat{u}_1, \ldots, \hat{u}_{i-1})$

Inner core of multi-level coding



Decision function $\phi_i : (\mathbf{y}, \hat{\mathbf{u}}^{i-1}) \rightarrow \hat{u}_i$

Bit-channels created by multi-level coding



$$\mathbf{y} = (y_1, \ldots, y_N)$$
$$\hat{\mathbf{u}}^{i-1} = (\hat{u}_1, \ldots, \hat{u}_{i-1})$$

Channel $W_i : U_i \to (\mathbf{Y}, \hat{\mathbf{U}}^{i-1})$

Decision function $\phi_i : (\mathbf{y}, \hat{\mathbf{u}}^{i-1}) \to \hat{u}_i$

Capacity and cutoff rate with multi-level coding

- Multi-level coding creates bit channels $W_i : U_i \to \mathbf{Y}\mathbf{U}^{i-1}$
- Capacity is preserved

$$\sum_{i=1}^{N} I(W_i) = \sum_{i=1}^{N} I(U_i; \mathbf{Y}, \mathbf{U}^{i-1}) = I(\mathbf{U}^N; \mathbf{Y}^N) = N\, I(W)$$

- Unlike Pinsker's scheme, multi-level coding is meant to be a practical scheme
  - the bit-channels are not near perfect
  - successive cancellation decoder exploits correlation in the bit-channels
- Multi-level coding contains Pinsker's scheme as a special asymptotic case, so it is capable of boosting the cutoff rate to capacity

$$\sum_{i=1}^{N} R_0(W_i) \approx N I(W)$$

- Can this be done at low complexity?

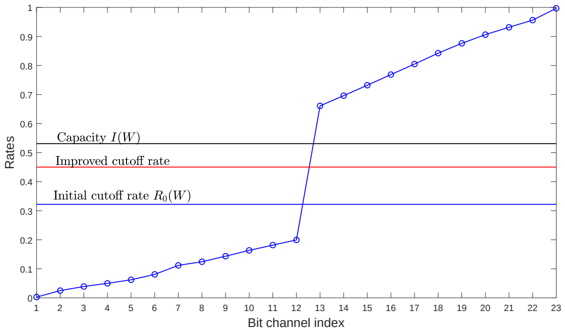
Multi-level coding with dual Golay code

Let the mapper be $\left[\begin{array}{c|c} I_{12} & 0 \\ \hline P & I_{11} \end{array}\right]$ where P is a parity-check matrix for a systematic dual Golay code,

$$P = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Cutoff rate improvement with dual Golay code

- There is a marked improvement in cutoff rate
- ... but comes at a high cost in system complexity.
- Capacities are polarizing!
- Can we take advantage of polarization?



Here, W is a BSC with $\epsilon = 0.1$ and capacity $I(W) = 0.531$ bits.

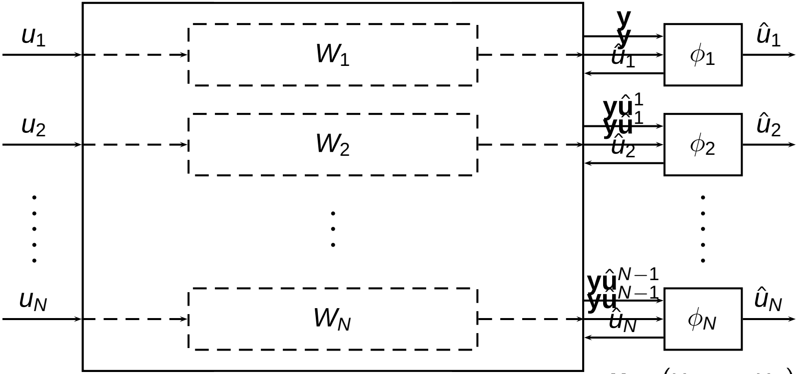
Research goal

- Goal: Seek a family of transforms $\{G_N\}$ indexed by transform size N so that, as N becomes large, the bit channels created by multi-level coding with the transforms $\{G_N\}$ polarize in the sense that
  - a fraction $I(W)$ of the capacities $\{I(W_i)\}$ approach 1,
  - a fraction $1 - I(W)$ of the capacities $\{I(W_i)\}$ approach 0.
- Motivation
  - Optimal rate allocation in multi-level coding is too complex
  - Rely on polarization and focus on schemes that use 0-1 rate allocation
  - Eliminates outer codes altogether
  - Should work if the transforms are sufficiently strong (as Pinsker's result shows)

Transform coding

K inputs
carry uncoded data

$N - K$ inputs
carry frozen bits



$\mathbf{y} = (y_1, \ldots, y_N)$

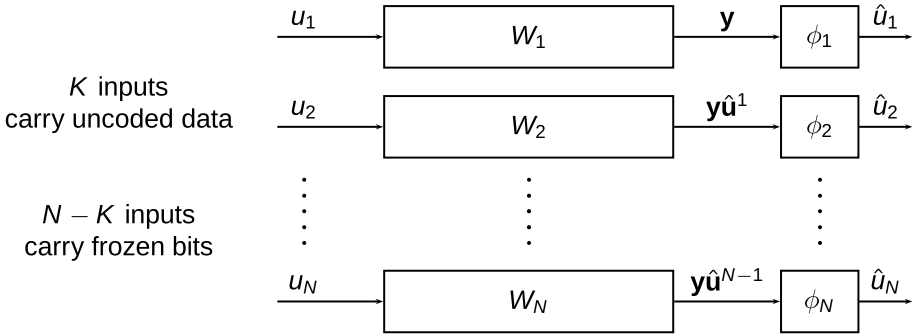
$\hat{\mathbf{u}}^{i-1} = (\hat{u}_1, \ldots, \hat{u}_{i-1})$

Pick the best K channels
to carry uncoded data

Channel $W_i : U_i \to (\mathbf{Y}, \hat{\mathbf{U}}^{i-1})$

Decision function $\phi_i : (\mathbf{y}, \hat{\mathbf{u}}^{i-1}) \to \hat{u}_i$

Transform coding system model



Channel $W_i : U_i \to (\mathbf{Y}, \hat{\mathbf{U}}^{i-1})$

Decision function $\phi_i : (\mathbf{y}, \hat{\mathbf{u}}^{i-1}) \to \hat{u}_i$

Transform code definition

A transform code is specified by a set of parameters $(N, K, \mathbf{G}_N, A)$ where

- N is the code block length, with $N = 2^n$ for some $n \geq 1$,
- K is the code dimension,
- $\mathbf{G}_N$ is a 1-1 transform (an invertible $N \times N$ matrix)
- A is a data index set, $A \subset \{1, 2, \dots, N\}$ with size $|A| = K$.

The rate of such a transform code is defined as $R = K/N$.

Encoding of a transform code

Given a data block $\mathbf{d} = (d_1, \ldots, d_K)$, a transform code with parameters $(N, K, \mathbf{G}_N, A)$ encodes $\mathbf{d}$ into a code block $\mathbf{x} = (x_1, \ldots, x_N)$ in two steps:

- constructs a transform input block by $\mathbf{u} = (u_1, \ldots, u_N)$ by setting

$$\mathbf{u}_A \stackrel{\Delta}{=} (u_i : i \in A) = \mathbf{d}, \qquad \mathbf{u}_{A^c} \stackrel{\Delta}{=} (u_i : i \in A^c) = \mathbf{0}.$$

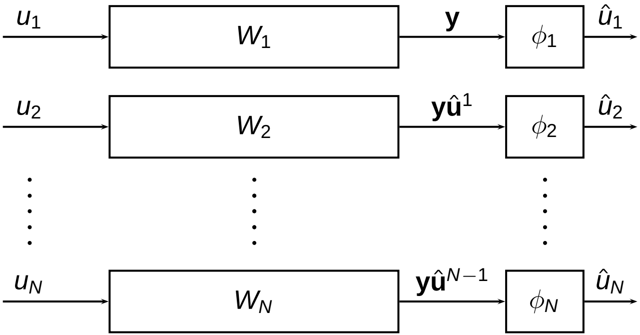
- computes the code block $\mathbf{x}$ by computing the transform of $\mathbf{u}$,

$$\mathbf{x} = \mathbf{u}\mathbf{G}_N.$$

Encoding and decoding a transform code

Subset A of inputs carry uncoded data

Subset A^c of inputs carry frozen bits



Channel $W_i : U_i \to (\mathbf{Y}, \hat{\mathbf{U}}^{i-1})$

Decision function $\phi_i : (\mathbf{y}, \hat{\mathbf{u}}^{i-1}) \to \hat{u}_i$

Decoding a transform code by successive cancelation

Given a channel output $\mathbf{y} = (y_1, \ldots, y_N)$, we consider a decoder that

- first produces an estimate $\hat{\mathbf{u}} = (\hat{u}_1, \ldots, \hat{u}_N)$ of the transform input block by $\mathbf{u} = (u_1, \ldots, u_N)$ by successively computing

$$\hat{u}_i = \phi_i(\mathbf{y}, \hat{\mathbf{u}}^{i-1}), \quad i = 1, \ldots, N,$$

  using a sequence of decision functions $\phi_1, \ldots, \phi_N$,
- then generates an estimate $\hat{\mathbf{d}} = (\hat{d}_1, \ldots, \hat{d}_K)$ of the data block $\mathbf{d}$ by setting,

$$\hat{\mathbf{d}} = \hat{\mathbf{u}}_A.$$

- The decoder knows the frozen bits $\mathbf{u}_{A^c}$ and generates an estimate so that $\hat{\mathbf{u}}_{A^c} = \mathbf{u}_{A^c}$.
- Thus, a decision error $\hat{\mathbf{d}} \neq \mathbf{d}$ occurs if and only if $\hat{\mathbf{u}}_A \neq \mathbf{u}_A$.

Probability of decoding error

- For the probability of error analysis we introduce a joint ensemble $(\mathbf{U}, \mathbf{X}, \mathbf{Y}, \hat{\mathbf{U}})$ of random vectors corresponding to the signals $(\mathbf{u}, \mathbf{x}, \mathbf{y}, \hat{\mathbf{u}})$ in the system.
- We are interested in finding a tractable bound on the block error probability

$$P_e \stackrel{\Delta}{=} P(\hat{\mathbf{U}}_A \neq \mathbf{U}_A) = P(\hat{\mathbf{U}} \neq \mathbf{U}).$$

- We will use a well-known genie argument for this purpose.

Genie-aided decoder argument

Regular decoder	Genie-aided decoder
$\hat{u}_1 = \phi_1(\mathbf{y})$	$\tilde{u}_1 = \phi_1(\mathbf{y})$
$\hat{u}_2 = \phi_2(\mathbf{y}, \hat{u}_1)$	$\tilde{u}_2 = \phi_2(\mathbf{y}, u_1)$
$\hat{u}_3 = \phi_3(\mathbf{y}, \hat{u}_1, \hat{u}_2)$	$\tilde{u}_3 = \phi_3(\mathbf{y}, u_1, u_2)$
$\vdots \quad \vdots$	$\vdots \quad \vdots$
$\hat{u}_N = \phi_N(\mathbf{y}, \hat{u}_1, \ldots, \hat{u}_{N-2}, \hat{u}_{N-1})$	$\tilde{u}_N = \phi_N(\mathbf{y}, u_1, \ldots, u_{N-2}, u_{N-1})$

- A block error is said to occur in the genie-aided decoder if $\tilde{u}_i \neq u_i$ for any i.
- Genie prevents error propagation but has no effect on the first error.
- If the genie-aided decoder decides the entire block **u** correctly, so does the regular decoder.
- Block error probabilities are the same.

Probability of block error for transform coding

Block error probability P_e for a transform code with data index set A is bounded as

$$P_e = P\left(\bigcup_{i \in A} \left\{ U_i \neq \phi_i(\mathbf{Y}, \mathbf{U}^{i-1}) \right\} \right)$$
$$\leq \sum_{i \in A} P\left(U_i \neq \phi_i(\mathbf{Y}, \mathbf{U}^{i-1}) \right)$$

where the first line uses the genie argument, and the second line the union bound.

• No longer need to worry about error propagation in the analysis. May safely study bit-channels with perfect decision-feedback.

• We recognize $P\left(U_i \neq \phi_i(\mathbf{Y}, \mathbf{U}^{i-1}) \right)$ as the probability of error for uncoded transmission across the ith bit-channel

$$W_i : U_i \to (\mathbf{Y}, \mathbf{U}^{i-1}).$$

Maximum-likelihood (ML) decision functions

In order to obtain the tightest possible upper bound in

$$P_e \leq \sum_{i \in A} P\left(U_i \neq \phi_i(\mathbf{Y}, \mathbf{U}^{i-1})\right)$$

we choose the decision functions ϕ_i as ML decision functions:

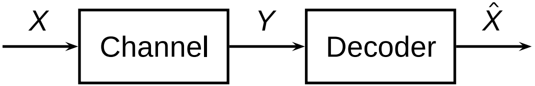
$$\phi_i(\mathbf{y}, \mathbf{u}^{i-1}) = \begin{cases} 0, & \text{if } W_i(\mathbf{y}, \mathbf{u}^{i-1}|0) \geq W_i(\mathbf{y}, \mathbf{u}^{i-1}|1) \\ 1, & \text{if } W_i(\mathbf{y}, \mathbf{u}^{i-1}|0) < W_i(\mathbf{y}, \mathbf{u}^{i-1}|1) \end{cases}$$

where W_i is the ith synthetic bit-channel.

Then, $P\left(U_i \neq \phi_i(\mathbf{Y}, \mathbf{U}^{i-1})\right)$ is the probability of error under ML decoding for uncoded transmission across the bit-channel

$$W_i : U_i \rightarrow (\mathbf{Y}, \mathbf{U}^{i-1}).$$

Bhattacharyya bound for uncoded transmission



Consider sending a single random bit X over a binary-input channel W and deciding it using the ML rule

$$\hat{X} = \begin{cases} 0, & \text{if } W(y|0) \geq W(y|1) \\ 1, & \text{if } W(y|0) < W(y|1) \end{cases}$$

Then, we have

$$P(\hat{X} \neq X) \leq Z(W) \triangleq \sum_{y} \sqrt{W(y|0)W(y|1)}$$

where $Z(W)$ is the Bhattacharyya parameter.

Proof of the Bhattacharyya bound

$$
\begin{aligned}
P(\hat{X} \neq X) &= P(X=0)P(\hat{X}=1|X=0) + P(X=1)P(\hat{X}=0|X=1) \\
&= \frac{1}{2} \sum_{y:W(y|1)>W(y|0)} W(y|0) + \frac{1}{2} \sum_{y:W(y|1)\leq W(y|0)} W(y|1) \\
&\leq \frac{1}{2} \sum_{y:W(y|1)>W(y|0)} W(y|0) \sqrt{\frac{W(y|1)}{W(y|0)}} + \frac{1}{2} \sum_{y:W(y|1)\leq W(y|0)} W(y|1) \sqrt{\frac{W(y|0)}{W(y|1)}} \\
&\leq \frac{1}{2} \sum_{\text{all } y} W(y|0) \sqrt{\frac{W(y|1)}{W(y|0)}} + \frac{1}{2} \sum_{\text{all } y} W(y|1) \sqrt{\frac{W(y|0)}{W(y|1)}} \\
&= \sum_{y} \sqrt{W(y|1)W(y|0)} \\
&= Z(W)
\end{aligned}
$$

Block error probability for transform coding

Combining the above results, the block error probability for transform coding under successive cancellation decoding is bounded as

$$P_e \leq \sum_{i \in A} Z(W_i)$$

Our next task is to seek sequences of transform codes that have rates approaching $I(W)$ and for which the above bound on P_e converges to zero.

Polar transform

- A sequence of transforms of size $N = 2^n$, $n \geq 1$, defined recursively as follows

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \qquad \mathbf{G}_N = \begin{bmatrix} \mathbf{G}_{N/2} & \mathbf{0} \\ \mathbf{G}_{N/2} & \mathbf{G}_{N/2} \end{bmatrix} = \mathbf{G}_2^{\otimes n}.$$

- We call $\mathbf{G}_N$ a polar transform of size N
- Recursive form leads to low-complexity encoding and decoding methods, as well as provably capacity-achieving explicit code constructions

Polar code definition

A polar code is specified by a set of parameters $(N, K, \mathbf{G}_N, A)$ where

- N is the code block length, with $N = 2^n$ for some $n \geq 1$,
- K is the code dimension,
- $\mathbf{G}_N$ is the polar transform defined by

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \qquad \mathbf{G}_N = \begin{bmatrix} \mathbf{G}_{N/2} & \mathbf{0} \\ \mathbf{G}_{N/2} & \mathbf{G}_{N/2} \end{bmatrix} = \mathbf{G}_2^{\otimes n},$$

- A is a data index set, $A \subset \{1, 2, \ldots, N\}$ with size $|A| = K$.

The rate of such a polar code is defined as $R = K/N$.

Encoding of a polar code

Given a data block $\mathbf{d} = (d_1, \ldots, d_K)$, a polar code with parameters $(N, K, \mathbf{G}_N, A)$ encodes $\mathbf{d}$ into a code block $\mathbf{x} = (x_1, \ldots, x_N)$ in two steps:

- constructs a transform input block by $\mathbf{u} = (u_1, \ldots, u_N)$ by setting

$$\mathbf{u}_A \triangleq (u_i : i \in A) = \mathbf{d}, \qquad \mathbf{u}_{A^c} \triangleq (u_i : i \in A^c) = \mathbf{0}.$$

- computes the code block $\mathbf{x}$ by computing the polar transform of $\mathbf{u}$,

$$\mathbf{x} = \mathbf{u}\mathbf{G}_N.$$

Preview of results

Polar codes are a class of linear block codes for binary-input memoryless channels with

- block length $N = 2^n$ for any $n \geq 1$,
- rate $R = K/N$ for any integer $1 \leq K \leq N$
- encoding complexity $N \log N$
- decoding complexity $N \log N$
- construction complexity roughly $O(N)$
- probability of block error goes to zero roughly as $2^{-\sqrt{N}}$ for any fixed rate $R < I(W)$ (no error floor)

Polar transform of size two

- Polar transform of size-2 is given by the matrix

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

- This transform takes an input $\mathbf{u} = (u_1, u_2)$ to an output $\mathbf{x} = (x_1, x_2)$ such that

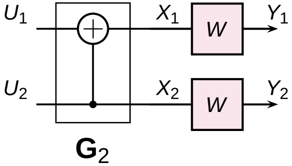
$$\mathbf{x} = \mathbf{u}\mathbf{G}_2 = (u_1 \oplus u_2, u_2),$$

  where $\oplus$ denotes addition in the binary field $\mathbb{F}_2$.
- The bit-channels created are given by

$$W_1 : U_1 \to (Y_1, Y_2), \qquad W_2 : U_2 \to (Y_1, Y_2, U_1)$$

Pictorial representation of polar transform of size two

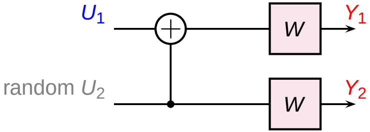
Combine two independent copies of a binary-input channel W



and split to create two bit-channels

$$W_1 : U_1 \to (Y_1, Y_2)$$
$$W_2 : U_2 \to (Y_1, Y_2, U_1)$$

The first bit-channel W_1



$$W_1 : U_1 \to (Y_1, Y_2)$$

Transition probabilities

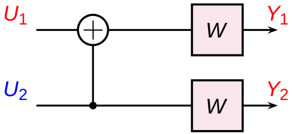
$$W_1(y_1, y_2|u_1) = \sum_{u_2 \in \{0,1\}} \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2)$$

Capacity

$$I(W_1) = I(U_1; Y_1, Y_2)$$

We will often denote this channel as W^-.

The second bit-channel W_2



$$W_2 : U_2 \to (Y_1, Y_2, U_1)$$

Transition probabilities

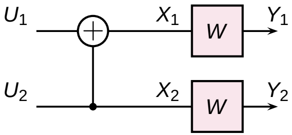
$$W_2(y_1, y_2, u_1 | u_2) = W(y_1 | u_1 \oplus u_2) W(y_2 | u_2)$$

Capacity

$$I(W_2) = I(U_2; Y_1, Y_2, U_1)$$

We will often denote this channel as W^+.

Capacity conserved but redistributed unevenly



- Conservation:

$$I(W^-) + I(W^+) = 2I(W)$$

- Motion towards extremes:

$$I(W^-) \leq I(W) \leq I(W^+)$$

with equality iff $I(W)$ equals 0 or 1.

Bhattacharyya parameter evolution

Lemma

For the channels in the size-2 transform $(W, W) \to (W^-, W^+)$, *we have*

$$Z(W^-) \leq 2Z(W) - \big[Z(W)\big]^2$$

and

$$Z(W^+) = \big[Z(W)\big]^2.$$

Equality holds if and only if W *is a binary erasure channel. (Note that extreme channels are special instances of a BEC.)*

Proof: Section 2.2.2 of the *Notes*.

Thus, the Bhattacharyya parameter is compressed by the polar transform:

$$Z(W^-) + Z(W^+) \ \leq \ 2Z(W).$$

Special case: BEC

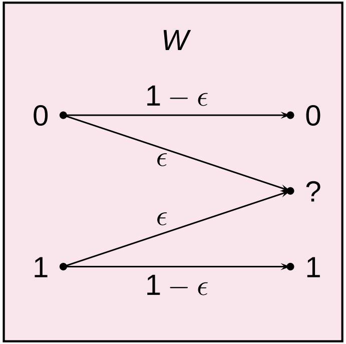
If W is a BEC with erasure probability ϵ, then W^- and W^+ are also BECs with erasure probabilities

$$\epsilon^- \triangleq 2\epsilon - \epsilon^2$$

and

$$\epsilon^+ \triangleq \epsilon^2,$$

respectively.



Special case: BEC W^-

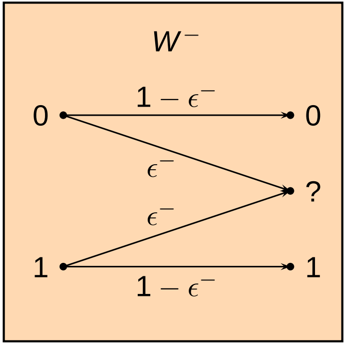
If W is a BEC with erasure probability ϵ, then W^- and W^+ are also BECs with erasure probabilities

$$\epsilon^- \triangleq 2\epsilon - \epsilon^2$$

and

$$\epsilon^+ \triangleq \epsilon^2,$$

respectively.



Special case: BEC W^+

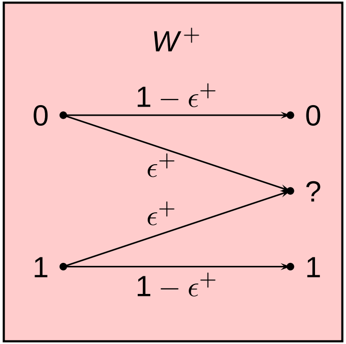
If W is a BEC with erasure probability ϵ, then W^- and W^+ are also BECs with erasure probabilities

$$\epsilon^- \triangleq 2\epsilon - \epsilon^2$$

and

$$\epsilon^+ \triangleq \epsilon^2,$$

respectively.



Guaranteed progress towards extreme channels

- For any binary-input channel W, we have

$$Z(W) - Z(W^+) \;=\; Z(W) - \left[Z(W)\right]^2 \;=\; Z(W)(1 - Z(W)).$$

  So, progress towards extremes is zero if and only if $Z(W) = 0$ or $Z(W) = 1$.
- Since the extremes of $I(W)$ and $Z(W)$ occur together, we can conclude that

$$I(W^+) - I(W) \;\equiv\; I(W) - I(W^-)$$

  is zero if and only if $I(W) = 0$ or $I(W) = 1$.
- Thus, size-2 polar transform makes “guaranteed progress” towards extremes, unless W is already extreme.

Polar transform of size four

- Polar transform of size-4 is given by the matrix

$$\mathbf{G}_4 = \begin{bmatrix} \mathbf{G}_2 & \mathbf{0} \\ \mathbf{G}_2 & \mathbf{G}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

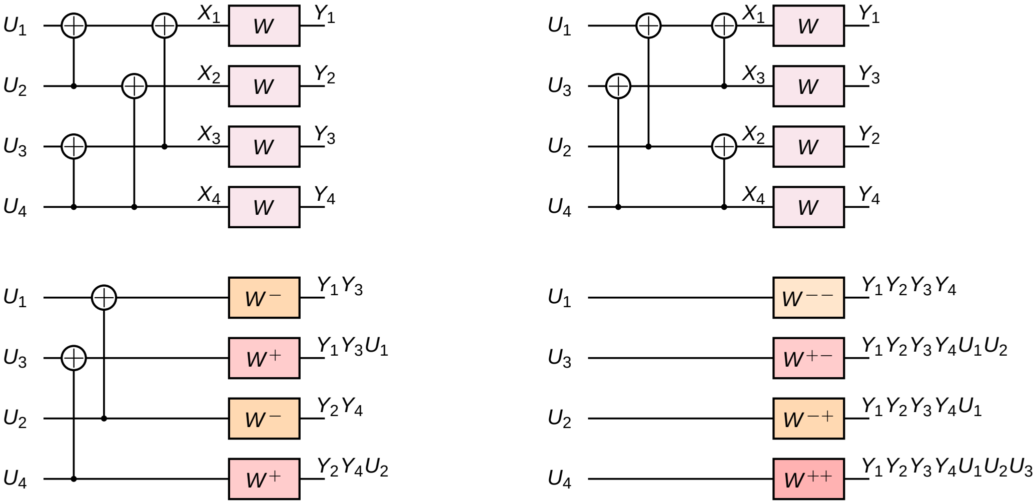
- This transform takes an input $\mathbf{u} = (u_1, u_2, u_3, u_4)$ to an output $\mathbf{x} = (x_1, x_2, x_3, x_4)$ such that

$$\mathbf{x} = \mathbf{u}\mathbf{G}_4 = (u_1 \oplus u_2 \oplus u_3 \oplus u_4, u_2 \oplus u_4, u_3 \oplus u_4, u_4).$$

- The bit-channels created are given by the size-4 polar transform are

$$W_1 : U_1 \to \mathbf{Y}, \quad W_2 : U_2 \to (\mathbf{Y}, U^1), \quad W_3 : U_3 \to (\mathbf{Y}, \mathbf{U}^2), \quad W_4 : U_4 \to (\mathbf{Y}, \mathbf{U}^3).$$

Disentangling the size-4 polar transform



Summary for size-4 polar transform

- Polar transform of size-4 creates four channels
  - $W_1 : U_1 \to (\mathbf{Y}, \mathbf{U}^0)$ is equivalent to $W^{--} = (W^-)^-$
  - $W_2 : U_1 \to (\mathbf{Y}, \mathbf{U}^1)$ is equivalent to $W^{-+} = (W^-)^+$
  - $W_3 : U_1 \to (\mathbf{Y}, \mathbf{U}^2)$ is equivalent to $W^{+-} = (W^+)^-$
  - $W_4 : U_1 \to (\mathbf{Y}, \mathbf{U}^3)$ is equivalent to $W^{++} = (W^+)^+$
- Thus, the extremization of channel capacity continues:

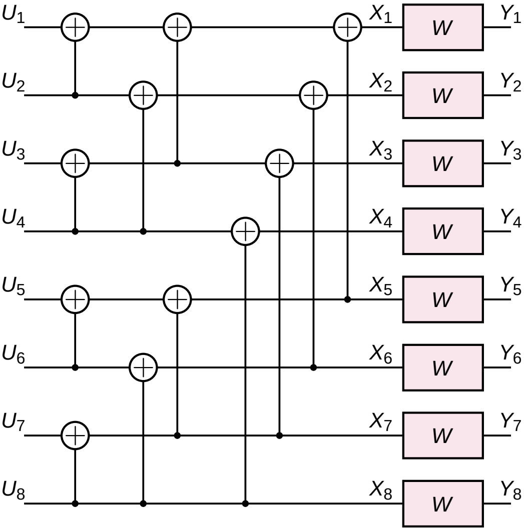
$$I(W^{--}) \leq I(W^-) \leq I(W^{-+}), \qquad I(W^{+-}) \leq I(W^+) \leq I(W^{++}),$$

  while there is conservation of capacity

$$I(W^{--}) + I(W^{-+}) + I(W^{+-}) + I(W^{++}) = 2I(W^-) + 2I(W^+) = 4I(W).$$

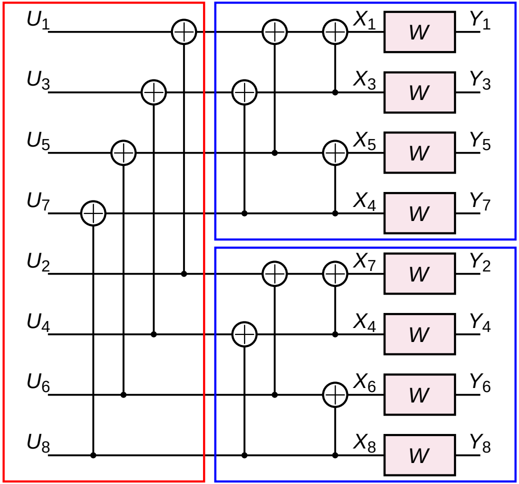
- Likewise, the Bhattacharyya parameters continue polarizing.

Size 8 construction



Rewire size-8 transform (using bit-reversal permutation)

- Rewiring is equivalent to applying a bit-reversal permutation to the wires without changing the logical relations
- Notice that the size-8 transform after rewiring contains two independent copies of the size-4 transform
- Like copies of the bit-channels created by the size-4 transform are combined using 4 independent copies of a size-2 transform



Summary for size-8 polar transform

Polar transform of size-8 creates eight bit-channels

- $W_1 \equiv W^{---} = (W^{--})^-$
- $W_2 \equiv W^{--+} = (W^{--})^+$
- $W_3 \equiv W^{-+-} = (W^{-+})^-$
- $W_4 \equiv W^{-++} = (W^{-+})^+$
- $W_5 \equiv W^{+--} = (W^{+-})^-$
- $W_6 \equiv W^{+-+} = (W^{+-})^+$
- $W_7 \equiv W^{++-} = (W^{++})^-$
- $W_8 \equiv W^{+++} = (W^{++})^+$

General case

Theorem

For any $N = 2^n$, $n \geq 1$, *and* $i = 1, \ldots, N$, *let* $W_N^{(i)}$ *denote the* i*th bit-channel created by a polar transform of size* N. *Then, there is the following recursive relationship*

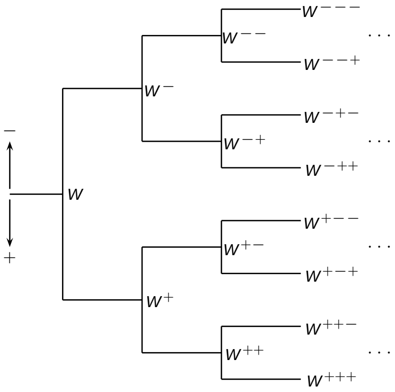
$$W_{2N}^{(2i-1)} = \left(W_N^{(i)}\right)^{-}, \quad \text{and} \quad W_{2N}^{(2i)} = \left(W_N^{(i)}\right)^{+}.$$

In other words, the pair of channels $(W_{2N}^{(2i-1)}, W_{2N}^{(2i)})$ *are obtained from two independent copies of* $W_N^{(i)}$ *by a size-2 polar transform.*

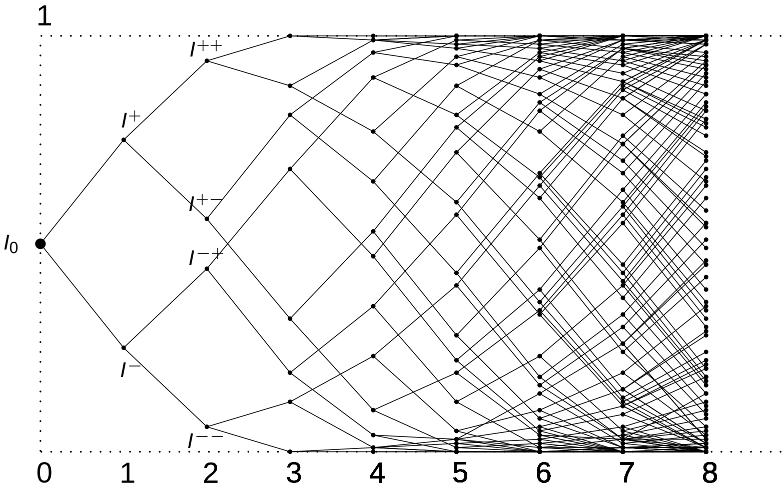
Proof: Section 2.1 of the *Notes*.

Bit-channels as a tree process

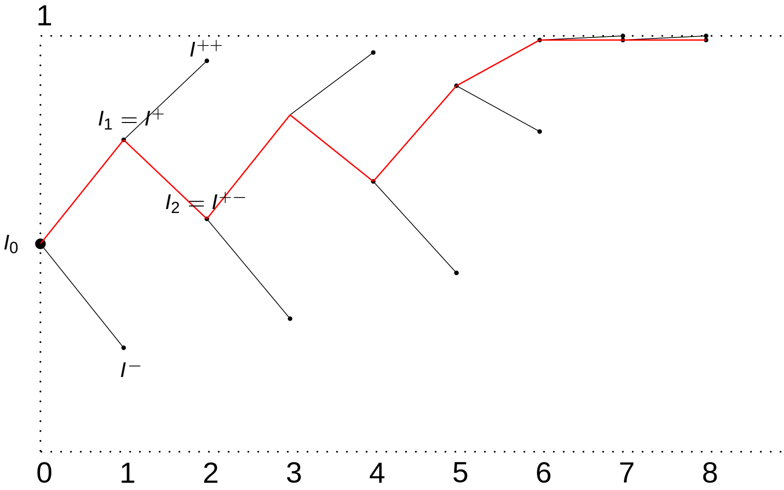
- Represent the bit-channel creation process as a tree
- Consider a random path through the tree driven by fair coin flips and let $W_0, W_1, \ldots$ denote the sequence of bit-channels along the path
- For example, if the coin flips are $+, -, -, \cdots$, then the channels along the path are $W_0 = W$, $W_1 = W^{+}$, $W_2 = W^{+-}$, $W_3 = W^{+--}, \cdots$
- Define a capacity process by $I_n \triangleq I(W_n)$
- Likewise, define a Bhattacharyya process by $Z_n \triangleq Z(W_n)$



Possible trajectories of the capacity process I_n



A sample path of the capacity process I_n



Martingale property of the capacity process I_n

Theorem

The capacity process $\{I_n; n \geq 0\}$ is a martingale w.r.t. the channel process $\{W_n : n \geq 0\}$, i.e.,

$$E[I_{n+1} \mid W_0, \ldots, W_n] = I_n, \quad n \geq 0,$$

and converges to a limit I_∞ a.e. and in L^1, where I_∞ satisfies $E[I_\infty] = I_0$. Furthermore, I_∞ is a 0-1 valued random variable with $P(I_\infty = 1) = I_0$.

Proof: We have $0 \leq I_n \leq 1$ for all $n \geq 0$; so, the process is bounded and integrable. The process is a martingale since

$$E[I_{n+1} \mid W_0, \ldots, W_n] = \frac{1}{2} I\left[(W_n)^-\right] + \frac{1}{2} I\left[(W_n)^+\right] = I(W_n).$$

Convergence to a limit I_∞ a.e. and in L^1 is a standard result about bounded martingales. The last claim follows from the "guaranteed progress" property.

The Bhattacharyya process Z_n is a supermartingale

Theorem

The Bhattacharyya process $\{Z_n; n \geq 0\}$ *is a supermartingale w.r.t. the channel process* $\{W_n : n \geq 0\}$, *i.e.,*

$$\mathrm{E}[Z_{n+1} \mid W_0, \dots, W_n] \leq Z_n, \quad n \geq 0,$$

and converges to a limit Z_∞ *a.e. and in* L^1, *where* Z_∞ *takes the values 0 and 1 with probabilities* I_0 *and* $1 - I_0$, *respectively.*

Proof: We have $0 \leq Z_n \leq 1$ for all $n \geq 0$; so, the process is bounded and integrable. The process is a supermartingale since

$$\mathrm{E}[Z_{n+1} \mid W_0, \dots, W_n] = \frac{1}{2} Z\left[(W_n)^-\right] + \frac{1}{2} Z\left[(W_n)^+\right] \leq 2Z(W_n) = Z_n.$$

Existence of Z_∞ is by standard convergence results about bounded supermartingales. The distribution of Z_∞ follows from the "guaranteed progress" property.

Polarization theorem

Theorem (Polarization)

The bit-channel capacities $\{I(W_i)\}$ created by the polar transform from independent copies of a binary-input channel W polarize: for any $\delta \in (0,1)$, as the construction size N grows

$$\left[\frac{\text{no. channels with } I(W_i) > 1-\delta}{N}\right] \longrightarrow I(W)$$

and

$$\left[\frac{\text{no. channels with } I(W_i) < \delta}{N}\right] \longrightarrow 1 - I(W).$$

Proof: Immediate corollary to the martingale convergence results.

Remarks

- Polarization alone does not guarantee that polar codes achieve capacity since the union bound
$$P_e \le \sum_{i \in A} Z(W_i)$$
need not converge to zero even if the terms $\{Z(W_i) : i \in A\}$ converge to zero.
- We need to show that there exist a sequence of polar codes $(N, K, \mathbf{G}_N, A_N)$ such that, as N goes to infinity through powers of two, the code rates K/N approach $I(W)$ and the union bounds
$$P_{e,N} \le \sum_{i \in A_N} Z(W_N^{(i)})$$
approach 0.
- For this we need a result on the rate (speed) of convergence of the Bhattacharyya parameters.

Fast polarization

Theorem

Let W be a given binary-input channel. For any fixed rate $R < I(W)$ and constant $\beta < \frac{1}{2}$, there exists a sequence of polar codes with parameters $(N, K, \mathbf{G}_N, A_N)$ such that $|A_N| \geq NR$ and

$$\sum_{i \in A_N} Z(W_N^{(i)}) = o(2^{-N^\beta}). \tag{1}$$

Conversely, if $R > 0$ and $\beta > \frac{1}{2}$, then for any such sequence of polar codes, we have

$$\max\{Z(W_N^{(i)}) : i \in A_N\} = \omega(2^{-N^\beta}). \tag{2}$$

Proof: Chapter 3 of the *Notes*.

Intuitive argument

- A fraction $I(W)$ of the sample paths of the Bhattacharyya process $\{Z_n\}$ go to zero. We choose the data index set A_N along such sample paths.
- We have

$$Z_{n+1} \begin{cases} \leq 2Z_n & \text{with probability 1/2 (a “-” move)} \\ = Z_n^2 & \text{with probability 1/2 (a ”+” move).} \end{cases}$$

- Thus, each sample path $Z_n(\omega)$ that goes to 0 has roughly $n/2$ “+” moves and $n/2$ “-” moves. So, very roughly,

$$Z_n \approx (1/2)^{2^{n/2}} = 2^{-\sqrt{N}}.$$

Polar codes achieve capacity

Corollary

Let W be a given binary-input channel. For any fixed rate $R < I(W)$ and constant $\beta < \frac{1}{2}$, there exists a sequence of polar codes with parameters $(N, K, \mathbf{G}_N, A_N)$ such that $|A_N| \geq NR$ and the corresponding sequence of block error probabilities satisfies

$$P_{e,N} = o(2^{-N^{\beta}}). \tag{3}$$

Conversely, if $R > 0$ and $\beta > \frac{1}{2}$, then for any such sequence of polar codes, the corresponding sequence of block error probabilities satisfies

$$P_{e,N} = \omega(2^{-N^{\beta}}). \tag{4}$$

Roughly stated, P_e goes to zero approximately as $2^{-\sqrt{N}}$ at any rate $R < I(W)$.

Encoding complexity

Theorem

Encoding complexity for polar coding is $\mathcal{O}(N \log N)$.

Proof: Let C_N be the complexity of computing the polar transform $\mathbf{x} = \mathbf{u}\mathbf{G}_N$. Using the recursion

$$\mathbf{G}_N = \begin{bmatrix} \mathbf{G}_{N/2} & \mathbf{0} \\ \mathbf{G}_{N/2} & \mathbf{G}_{N/2} \end{bmatrix}.$$

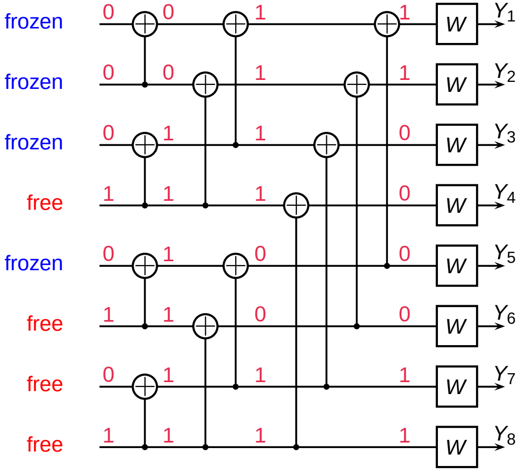
we estimate that

$$C_N \leq 2C_{N/2} + 3N/2, \qquad \text{starting with } C_2 = 3.$$

This yields $C_N \leq (3N/2) \log N$.

Alternative proof

A polar encoder circuit consists of $\log N$ stages with N logic operations per stage.



Decoding complexity

Theorem

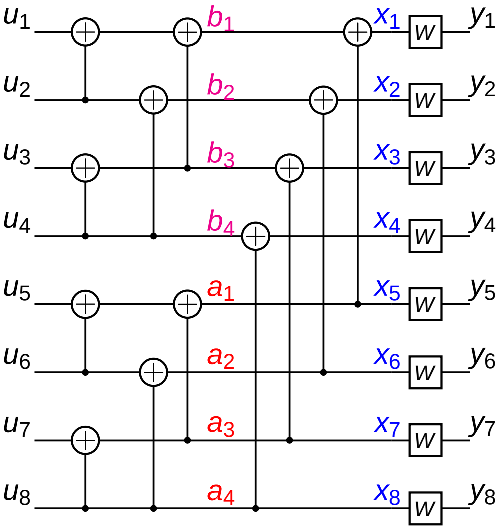
The complexity of successive cancellation decoding for polar codes is $\mathcal{O}(N \log N)$.

Proof: Let C_N be the complexity of decoding a code of length N. Decoding problem of size N for W reduced to two decoding problems of size $N/2$ on channels W^- and W^+, respectively. We estimate that

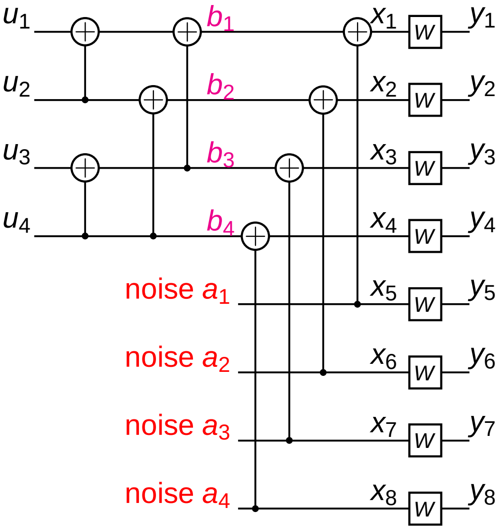
$$C_N = 2C_{N/2} + kN$$

for some constant k. This gives $C_N = \mathcal{O}(N \log N)$. (For details, please see Section 5.2 of *Notes*.)

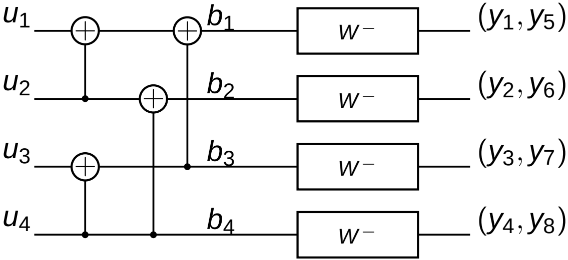
Ilustration of the decoder recursion



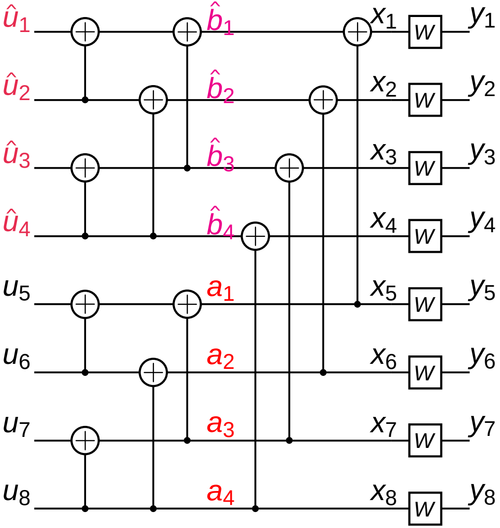
First phase: treat **a** as noise, decode (u_1, u_2, u_3, u_4)



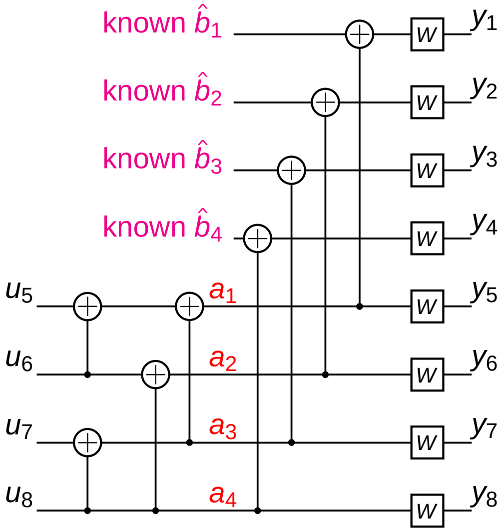
First phase decoding



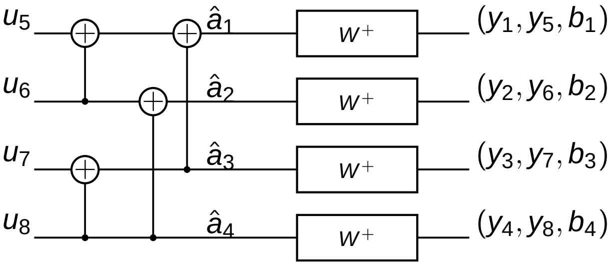
End of first phase decoding



Second phase: Treat $\hat{\mathbf{b}}$ as known, decode (u_5, u_6, u_7, u_8)



Second phase decoding



Construction complexity

The construction problem is to identify which of the N channels synthesized by the polar transform are the best ones for sending data.

Theorem

Given W and a rate $R < I(W)$, a polar code can be constructed in $\mathcal{O}(N poly(log(N)))$ time.

Proof: This is a difficult problem because the ith bit channel W_i synthesized by the polar transform has output symbols of the form $(\mathbf{y}, \mathbf{u}^{i-1})$, which means that the corresponding output alphabet size grows exponentially in N. Fortunately, there are channel approximation techniques that keep the alphabet sizes of the channels under control while providing estimates of the true bit-channel capacities with upper and lower bounds on the estimation error. For details and references, we refer to Section 5.3 of the *Notes*.

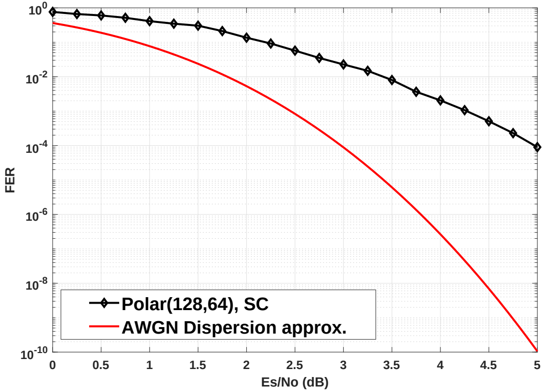
Polar coding summary

Polar codes are a class of linear block codes for binary-input memoryless channels with

- block length $N = 2^n$ for any $n \geq 1$,
- rate $R = K/N$ for any integer $1 \leq K \leq N$
- encoding complexity $N \log N$
- decoding complexity $N \log N$
- construction complexity roughly $\mathcal{O}(N \text{poly}(log(N)))$
- probability of block error goes to zero roughly as $2^{-\sqrt{N}}$ for any fixed rate $R < I(W)$ (no error floor)

Polar code performance

- Simulation results for BPSK signaling on AWGN channel, with polar code optimized for performance at SNR 3dB.



- Performance is not strong enough for polar codes to be a contender for practical applications.

Improving polar code performance

The weak performance of polar codes may be attributed to a number of factors

- Suboptimality of the successive cancellation decoder
- Poor minimum distance properties
- Polarization takes a while to take effect

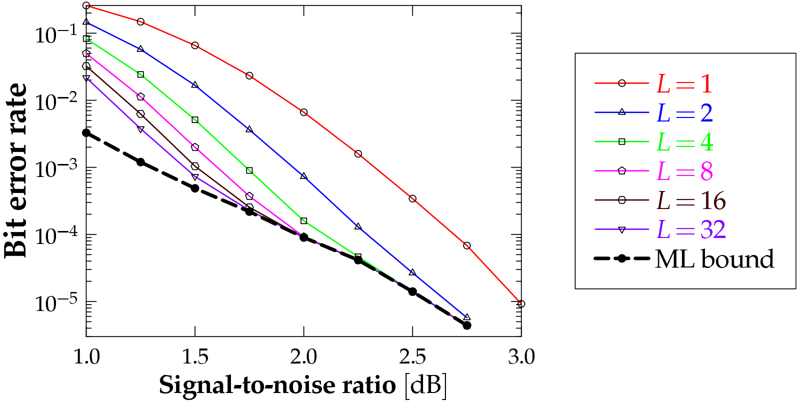
Improving the decoder: List decoding

Tal and Vardy (2011) introduced a list-decoder for polar codes

- Produces L candidate decisions by a breadth-first search technique with pruning
- Picks the most likely decision from the list
- Has complexity $\mathcal{O}(LN \log N)$

List decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



List-of-L performance appears to approach the ML performance!

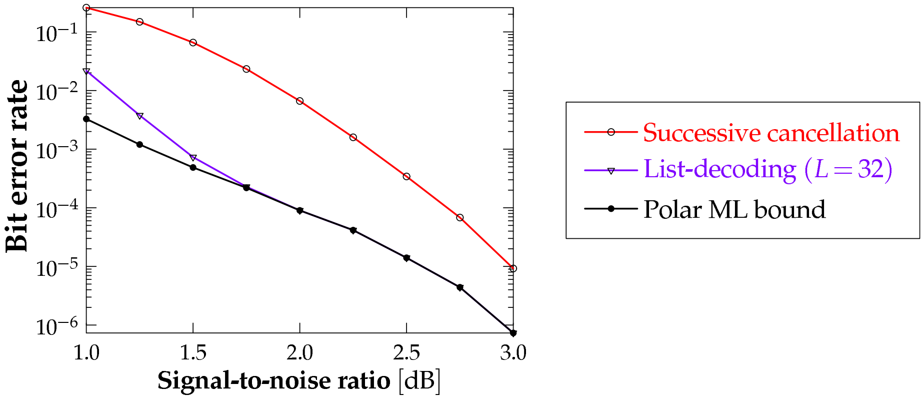
Improving the decoder and the code: List decoder with CRC

Tal-Vardy (2011) also introduced a CRC to reduce errors in identifying the correct codeword in the final list of L candidates

- Same decoder as before but data contains a built-in CRC
- Final list pruned by CRC before selecting the most likely candidate

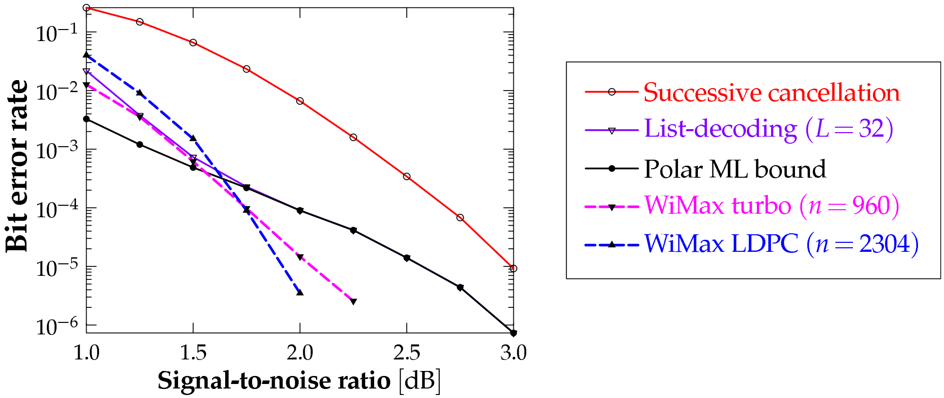
Performance of CRC-aided list decoder

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



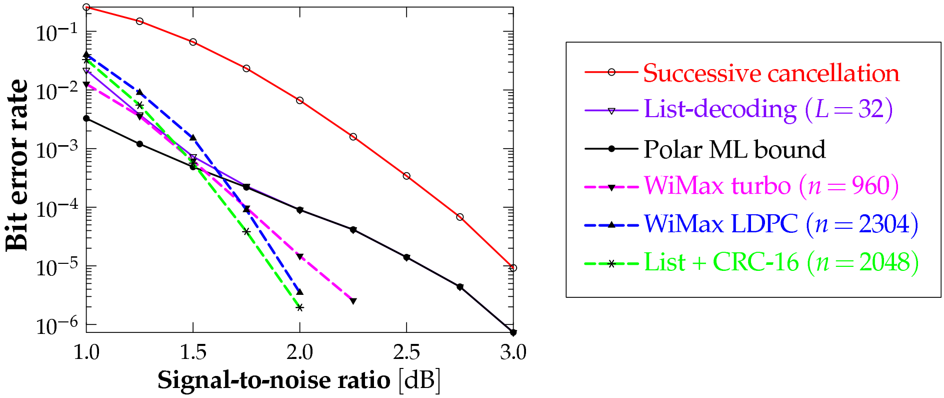
Performance of CRC-aided list decoder

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



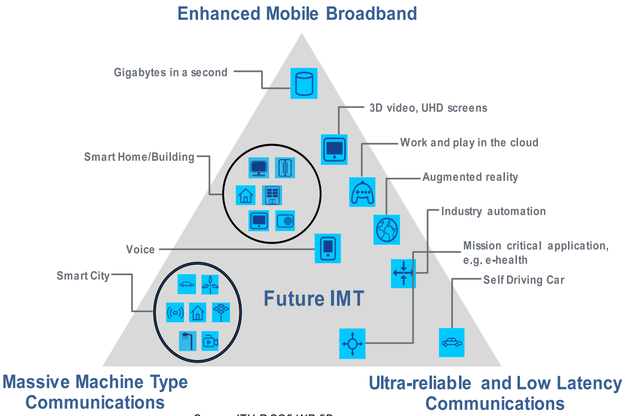
Performance of CRC-aided list decoder

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



Polar codes (+CRC) achieve state-of-the-art performance!

5G Usage Scenarios

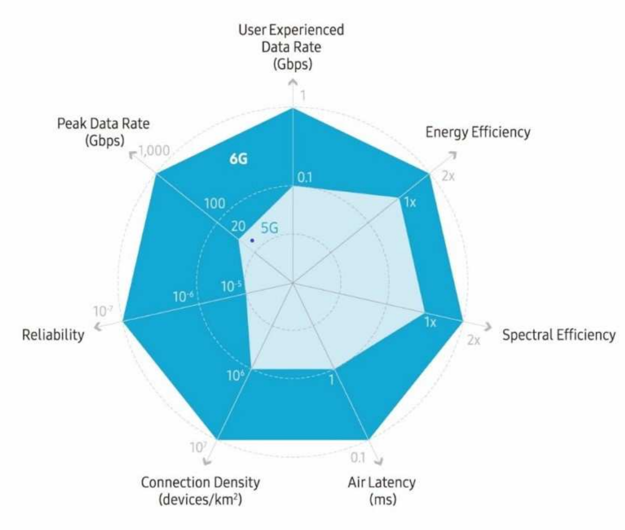


Source: ITU-R SG5 WP-5D

LDPC and polar codes have been adopted as the main channel coding schemes for 5G.

(Ref: “Multiplexing and channel coding, v.15.3.0,” 3GPP, Sophia Antipolis, France, Rep. 38.212, 2018.)

6G vs 5G



Ref: The Next Hyper Connected Experience for All, Samsung Research, July 14, 2020.

Beyond 5G Usage Scenarios

Some challenges:

- eMBB: 1 Tb/s decoder throughput
- uRLLC: 0.1 ms latency and 10^{-7} to 10^{-10} frame error rate
- mMTC: Near optimal performance at short block lengths
- Satisfy implementation metrics with respect to complexity, area and energy efficiency

Beyond 5G uRLLC Usage Scenarios

Application	Performance requirements
Telesurgery, remote driving	E2E RTL $<$ 20 ms, BLER 10^{-9}, Data rate $>$ 10 Gbps
Holographic telesurgery	E2E RTL $<$ 20 ms, BLER 10^{-9}, Data rate $<$ 4.6 Tbps
Digital real estate, eCommerce, eConference, eGames	E2E RTL $<$ 20 ms, BLER 10^{-5} - 10^{-6}, Data rate $>$ 10 Gbps
Factory automation	E2E RTL $<$ 20 ms, BLER 10^{-5} - 10^{-6}
Autonomous driving, Smart highway	E2E 5-10 ms, BLER 10^{-5}
Motion control	Cycle time $<$ 0.5 ms, Payload $<$ 50 bytes
Manufacturing process	E2E $<$ 10 ms, Payload $<$ 50 bytes, BLER 10^{-9}

Ref: Alves, Hirley, et al. “Beyond 5G URLLC Evolution: New service modes and practical considerations.” arXiv preprint arXiv:2106.11825 (2021).

Remarks

- 5G channel codes do not meet the 6G requirements
- Design space will be larger for 6G due to improvements in chip technology and algorithm design
- It is unlikely that a one-size-fits-all type of solution will emerge
- Need unified solutions based on various base designs that can re-use hardware

Polar coding at Tb/s data rates

In a H2020 project called EPIC, we studied Tb/s channel coding under the following targets.

- Throughput 1 Tb/s
- Frame error rate 10^{-6} to 10^{-12}
- VLSI constraints

Technology	7nm
Throughput	1 Tb/s
Clock freq.	$\leq$ 1 GHz
Silicon area	$\leq$ 10 mm^2
Pow. Den.	$\leq$ 0.1 W/mm^2
Area Eff.	$\geq$ 100 $Gb/s/mm^2$
Energy Eff.	$\leq$ 1 pJ/bit

- Goal: Obtain the best coding gain per code family subject to these constraints

16 nm Implementation Results

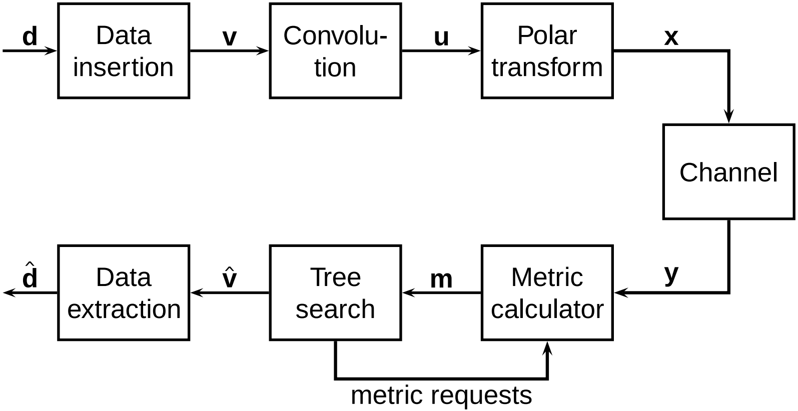
- In EPIC, we implemented a 1 Tb/s polar code using 16nm technology
- Code length 1024, rate 0.83, more than 6 dB coding gain
- 2.5 mm2 area with %70 utilization
- Full timing closure for 1 Tb/s data rate @ 0.8 V with 850 mW power dissipation (Power less than 1 pJ/b)
- It is possible to achieve timing closure @ 0.7 V with 650 mW power dissipation (minor setup, some hold issues to fix)

Architecture	Memory Elements	Cell Area (mm^2)	Pipeline	Power (mW)
1 Core – 1200MHz	395K	0.47	42	682@0.63V
4 Cores – 300MHz	759K	1.00	25	489@0.63V
8 Cores – 150MHz	818K	1.18	12	423@0.63V

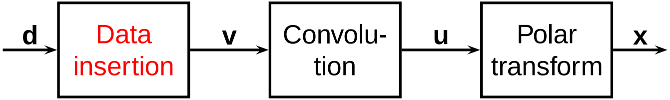
Further details in A. Süral et al, “Terabits-per-Second Throughput for Polar Codes,” in 2019 IEEE PIMRC Workshops, 2019, pp. 1-7.

Polar coding for extreme reliability at short block lengths

Polarization Adjusted Convolutional (PAC) Codes



Data insertion

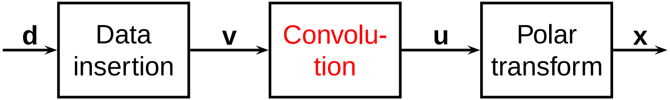


- The data inserter feeds source data **d** to the convolution block in a paced manner so as to avoid information bottlenecks at the decoder.
- The schedule for data insertion is indicated by a set $\mathcal{A}$.
- Example: Suppose the data **d** is of length 4, the code length is $N = 8$, and the schedule is $\mathcal{A} = \{4, 6, 7, 8\}$. Then, the input **v** to the convolution block is as follows

$$\mathbf{v} = (0, 0, 0, d_1, 0, d_2, d_3, d_4)$$

.

Convolution

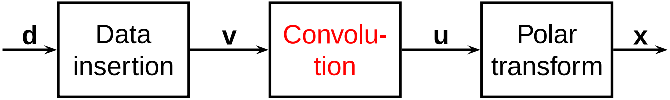


The convolution block receives $\mathbf{v} = (v_1, \ldots, v_N)$ and produces $\mathbf{u} = (u_1, \ldots, u_N)$ so that

$$u_i = \sum_{j=0}^{m} c_j v_{i-j}$$

where $\mathbf{c} = (c_0, \cdots, c_m)$ is an impulse response and $v_{i-j} = 0$ for $j \geq i$.

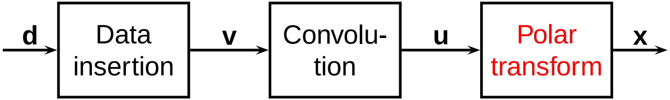
Convolution in matrix form



The convolution block computes $\mathbf{u} = \mathbf{v}\mathbf{T}$ where $\mathbf{T}$ is an upper-triangular Toeplitz matrix

$$\mathbf{T} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_m & 0 & \cdots & 0 \\ 0 & c_0 & c_1 & c_2 & \cdots & c_m & & \vdots \\ 0 & 0 & c_0 & c_1 & \ddots & \cdots & c_m & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & & & \ddots & 0 & c_0 & c_1 & c_2 \\ \vdots & & & & 0 & 0 & c_0 & c_1 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & c_0 \end{bmatrix}$$

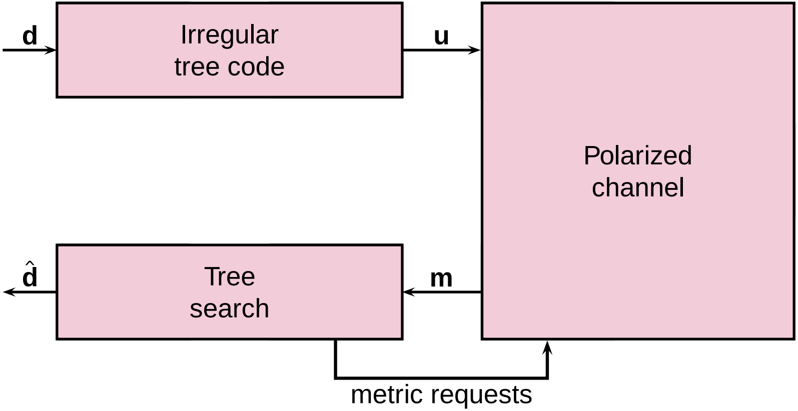
Polar transform



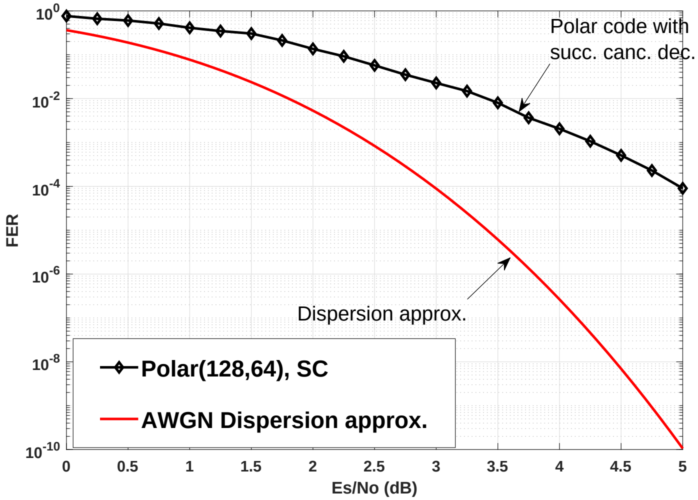
The polar transform block computes $\mathbf{x} = \mathbf{u}\mathbf{P}$ where $\mathbf{P}$ is a Kronecker power

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n}.$$

Decoding

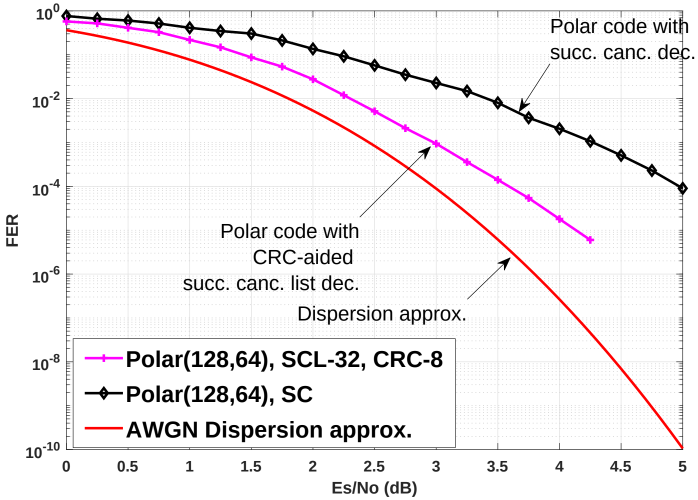

d
Irregular tree code
u
Polarized channel
d̂
Tree search
m
metric requests

Performance comparison



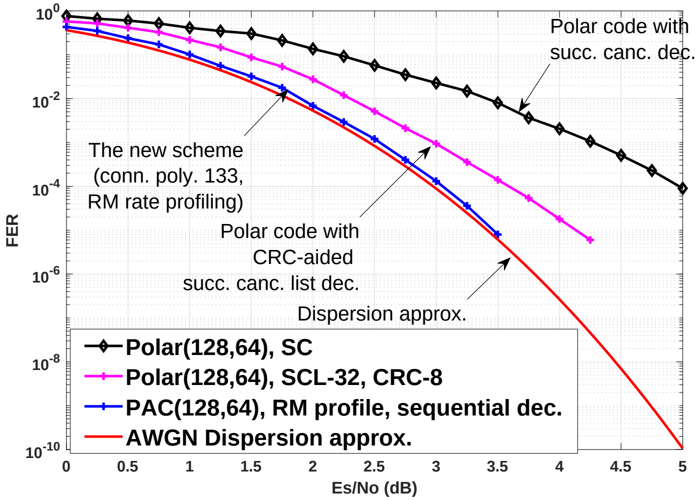
(BPSK-AWGN channel with codes optimized for 3dB SNR)

Performance comparison



(BPSK-AWGN channel with codes optimized for 3dB SNR)

Performance comparison



(BPSK-AWGN channel with codes optimized for 3dB SNR)

How do we explain the performance improvement?

The concatenation of an inner polar transform **P** with a randomly chosen upper-triangular Toeplitz matrix **T** achieves several things at once.

- The inner polar transform boosts the cutoff rate.
- The outer transform creates a tree code that can be decoded by sequential decoding.
- An overall transform $\mathbf{G} = \mathbf{TP}$ is created that looks random and hence achieves the dispersion bound.
- Reference: E. Arıkan, “From sequential decoding to channel polarization and back again,” arXiv:1908.09594 [cs, math], Aug. 2019.

Summary

We presented

- the background ideas that led to polar coding
- channel polarization theory
- methods to improve polar coding performance
- some challenges for polar coding in the near future

Thank you!