Cyclic Polar Codes

Narayanan Rengaswamy
Texas A&M University
r_narayanan_92@tamu.edu

Henry D. Pfister
Duke University
henry.pfister@duke.edu

2015 IEEE International Symposium on Information Theory

June 16, 2015

Overview

Polar Codes

- Introduced by Arıkan in [Arı09] using the binary 2×2 kernel

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

- Length $N = 2^n$ polar transform matrix is given by

$$G_N = B_N G_2^{\otimes n},$$

where B_N is the length-N bit-reversal permutation matrix.

- Shown to achieve the symmetric capacity of binary input DMCs, asymptotically, under successive cancellation (SC) decoding.

Extensions of Polar Codes

Blocklength $N = \ell^n$, $\ell > 2$; Transformation $G_N = B_N G_\ell^{\otimes n}$

- Korada et al.: Binary G_ℓ for binary DMCs [KŞU10].
- Şaşoğlu et al.: Binary G_ℓ for q-ary DMCs, q prime [ŞTA09].

Extensions of Polar Codes

Blocklength $N = \ell^n$, $\ell > 2$; Transformation $G_N = B_N G_\ell^{\otimes n}$

- Korada et al.: Binary G_ℓ for binary DMCs [KŞU10].
- Şaşoğlu et al.: Binary G_ℓ for q-ary DMCs, q prime [ŞTA09].
- Mori and Tanaka: Non-binary G_ℓ for arbitrary q-ary DMCs, $q = p^m$ [MT10; MT14].
  - For example, using extended RS matrices.

$$G_{RS}(3,3) = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \alpha^0 & \alpha^0 & 0 \\ \alpha^1 & \alpha^0 & 0 \\ \alpha^0 & \alpha^0 & \alpha^0 \end{bmatrix}$$

where $\alpha = 2 \in \mathbb{F}_3$ is primitive.

Motivation for Cyclic Polar Codes

So far, blocklength $N = \ell^n$ and transformation $G_N = B_N G_\ell^{\otimes n}$.

How about mixed-size kernels?

Motivation for Cyclic Polar Codes

System implementation: Many systems use RS or other cyclic codes.

Can we relate polar codes to RS codes?

More fundamentally, can we make polar codes cyclic?

Overview

Galois field Fourier Transform

- Replace polar transform with Galois field Fourier Transform (GFFT).
- Input: $\underline{u} = (u_0, u_1, \ldots, u_{N-1})$, Output: $\underline{v} = (v_0, v_1, \ldots, v_{N-1})$. Then

$$\underline{u} \xleftarrow{GFFT} \underline{v}$$

- If F_N is the GFFT matrix,

$$\underline{u} = F_N \underline{v} \qquad \text{(or)} \qquad u_j = \sum_{i=0}^{N-1} v_i \omega^{ij}$$

for $j = 0, 1, \ldots, N-1$, where ω in $\mathbb{F}_q$ has order N.

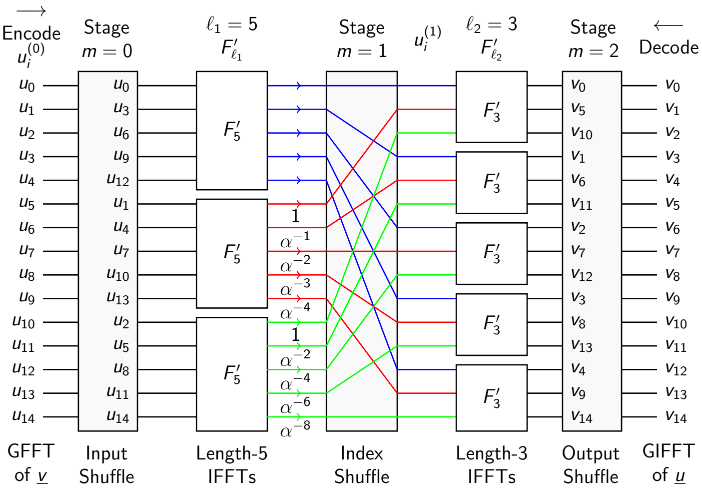
Galois field Fourier Transform

- How is this related to the polar transform?

Galois field Fourier Transform

- How is this related to the polar transform?
- Evaluate the GFFT using the Cooley-Tukey FFT [CT65].
  - Factor $N = \prod_{m=1}^{n} \ell_m = \ell_1 \ell_2 \cdots \ell_n$.
  - Implement small GFFTs of length ℓ_m directly.
  - Combine them using appropriate twiddle factors and index-shuffles.
  - Simplest case is $\ell_1 = \ell_2 = \cdots = \ell_n = 2$ for $N = 2^n$; equivalent to standard polar code.
- Ignoring twiddle factors, the Kronecker product of repeated short GFFTs gives a long GFFT.

Galois field Fourier Transform



An example for $N = 15$ over $\mathbb{F}_{16}$ depicting the transform.

α is a primitive element in $\mathbb{F}_{16}$. In this case $\omega = \alpha$.

Cyclic Polar Codes

- Recollect that

$$u_j = \sum_{i=0}^{N-1} v_i \omega^{ij}$$

where $\omega^N = 1$ in $\mathbb{F}_q$.

- In polynomial notation, with $v(x) = \sum_{i=0}^{N-1} v_i x^i$, we have

$$u(x) = \sum_{j=0}^{N-1} u_j x^j = \sum_{j=0}^{N-1} v(\omega^j) x^j$$

- Hence, u_j's are evaluations of $v(x)$.

Cyclic Polar Codes

- Design of the code $\mathcal{C}$ produces the set of information indices $\mathcal{A}$.

Cyclic Polar Codes

- Design of the code $\mathcal{C}$ produces the set of information indices $\mathcal{A}$.
- Given $\mathcal{A}^c$, the indices frozen to **zeros** in $u(x)$, there exists a generator $g(x)$ such that

$$v(x) = u_{\mathcal{A}}(x)g(x) = u_{\mathcal{A}}(x)\prod_{j\in\mathcal{A}^c}(x-\omega^j)$$

where $\omega^N = 1$ in F_q.

Cyclic Polar Codes

- Design of the code $\mathcal{C}$ produces the set of information indices $\mathcal{A}$.
- Given $\mathcal{A}^c$, the indices frozen to **zeros** in $u(x)$, there exists a generator $g(x)$ such that

$$v(x) = u_{\mathcal{A}}(x)g(x) = u_{\mathcal{A}}(x)\prod_{j\in\mathcal{A}^c}(x-\omega^j)$$

where $\omega^N = 1$ in F_q.

- We have a cyclic code!

Cyclic Polar Codes

- Design of the code $\mathcal{C}$ produces the set of information indices $\mathcal{A}$.
- Given $\mathcal{A}^c$, the indices frozen to **zeros** in $u(x)$, there exists a generator $g(x)$ such that

$$v(x) = u_{\mathcal{A}}(x)g(x) = u_{\mathcal{A}}(x)\prod_{j\in\mathcal{A}^c}(x-\omega^j)$$

  where $\omega^N = 1$ in F_q.
- We have a cyclic code!
- Constraint: $N|(q-1)$. Hence, field size must grow with the blocklength.

Cyclic Polar Codes

Is this transformation polarizing?

Cyclic Polar Codes

- Example: $N = 3$ and $N = 5$ over $\mathbb{F}_{16}$.

$$G_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{bmatrix} \quad \text{and} \quad G_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega & \omega^3 \\ 1 & \omega^3 & \omega & \omega^4 & \omega^2 \\ 1 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix}$$

where $\omega^3 = 1$ for G_3, $\omega^5 = 1$ for G_5.

Cyclic Polar Codes

- Example: $N = 3$ and $N = 5$ over $\mathbb{F}_{16}$.

$$G_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{bmatrix} \quad \text{and} \quad G_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega & \omega^3 \\ 1 & \omega^3 & \omega & \omega^4 & \omega^2 \\ 1 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix}$$

  where $\omega^3 = 1$ for G_3, $\omega^5 = 1$ for G_5.

- The transformation G_N, a GFFT matrix, polarizes any q-ary channel because it
  - is invertible and not upper triangular.
  - contains a primitive element [MT14].

Overview

Soft Decoder

- Blocklength $N = 2^n$ over q-ary channel, q prime.

Soft Decoder

- Blocklength $N = 2^n$ over q-ary channel, q prime.
- Consider a 2×2 butterfly: inputs (a_0, a_1) and outputs (b_0, b_1).

$$b_0 = a_0 + a_1$$
$$b_1 = a_0 + \alpha a_1$$

Soft Decoder

- Blocklength $N = 2^n$ over q-ary channel, q prime.
- Consider a 2×2 butterfly: inputs (a_0, a_1) and outputs (b_0, b_1).

$$
\begin{aligned}
b_0 &= a_0 + a_1 \\
b_1 &= a_0 + \alpha a_1
\end{aligned}
$$

- To estimate (a_0, a_1) from (b_0, b_1) in the polar decoding order, we have

$$\hat{a}_0 = (1 - \alpha)^{-1}(b_1 - \alpha b_0)$$

Soft Decoder

- Blocklength $N = 2^n$ over q-ary channel, q prime.
- Consider a 2×2 butterfly: inputs (a_0, a_1) and outputs (b_0, b_1).

$$\begin{aligned} b_0 &= a_0 + a_1 \\ b_1 &= a_0 + \alpha a_1 \end{aligned}$$

- To estimate (a_0, a_1) from (b_0, b_1) in the polar decoding order, we have

$$\begin{aligned} \hat{a}_0 &= (1-\alpha)^{-1}(b_1 - \alpha b_0) \\ \hat{a}_1 &= b_0 - a_0 \end{aligned}$$

Soft Decoder

- Blocklength $N = 2^n$ over q-ary channel, q prime.
- Consider a 2×2 butterfly: inputs (a_0, a_1) and outputs (b_0, b_1).

$$b_0 = a_0 + a_1$$
$$b_1 = a_0 + \alpha a_1$$

- To estimate (a_0, a_1) from (b_0, b_1) in the polar decoding order, we have

$$\hat{a}_0 = (1 - \alpha)^{-1}(b_1 - \alpha b_0)$$
$$\hat{a}_1 = b_0 - a_0$$
$$\hat{a}_1' = \alpha^{-1}(b_1 - a_0)$$

Soft Decoder

- 2×2 butterfly: Compute optimal soft estimates for (a_0, a_1) from (b_0, b_1) using standard techniques from LDPC codes.

Soft Decoder

- 2×2 butterfly: Compute optimal soft estimates for (a_0, a_1) from (b_0, b_1) using standard techniques from LDPC codes.

- $\ell \times \ell$ butterfly, $\ell > 2$: Hard to implement APP decoder for general length-ℓ code over F_q.

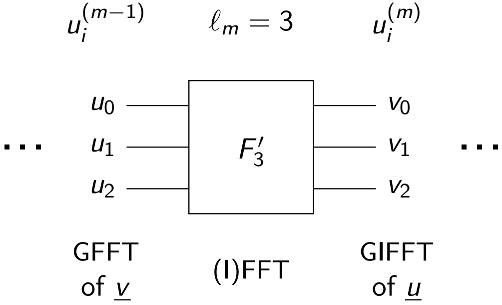
Soft Decoder

- 2×2 butterfly: Compute optimal soft estimates for (a_0, a_1) from (b_0, b_1) using standard techniques from LDPC codes.
- $\ell \times \ell$ butterfly, $\ell > 2$: Hard to implement APP decoder for general length-ℓ code over F_q.
- $\ell \times \ell$ butterfly, $\ell > 2$: Alternatively, use algebraic hard-decision decoding.

Code Design for Algebraic Erasures Decoding

q-ary Erasure Channel (QEC) with $\epsilon = 0.5$.
Use Forney's algebraic decoder.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\}$$

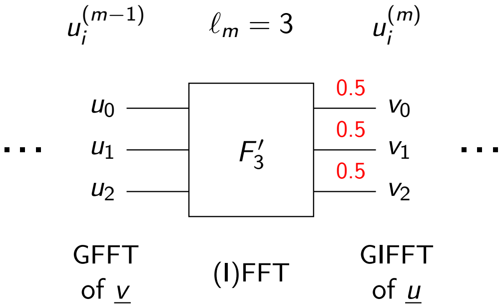


Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Erasures Decoding

q-ary Erasure Channel (QEC) with $\epsilon = 0.5$.
Use Forney's algebraic decoder.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ \epsilon = P\{v_i = ?\} = 0.5$$



Uncover next (unknown) input using outputs and recovered inputs.

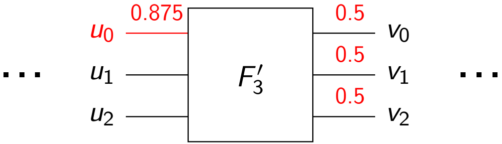
Code Design for Algebraic Erasures Decoding

q-ary Erasure Channel (QEC) with $\epsilon = 0.5$.
Use Forney's algebraic decoder.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ \epsilon = P\{v_i = ?\} = 0.5$$

Theoretical $\quad u_i^{(m-1)} \quad \ell_m = 3 \quad u_i^{(m)}$

$1 - (1 - \epsilon)^3$



GFFT of $\underline{v}$ $\quad$ (I)FFT $\quad$ GIFFT of $\underline{u}$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Erasures Decoding

q-ary Erasure Channel (QEC) with $\epsilon = 0.5$.
Use Forney's algebraic decoder.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ \epsilon = P\{v_i = ?\} = 0.5$$

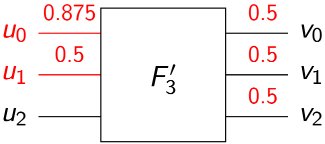
Theoretical

$1 - (1-\epsilon)^3$

$1 - [3(1-\epsilon)^2\epsilon + (1-\epsilon)^3]$

$\cdots$

$u_i^{(m-1)}$ $\quad \ell_m = 3 \quad$ $u_i^{(m)}$



$\cdots$

GFFT of $\underline{v}$ $\quad$ (I)FFT $\quad$ GIFFT of $\underline{u}$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Erasures Decoding

q-ary Erasure Channel (QEC) with $\epsilon = 0.5$.
Use Forney's algebraic decoder.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ \epsilon = P\{v_i = ?\} = 0.5$$

Theoretical

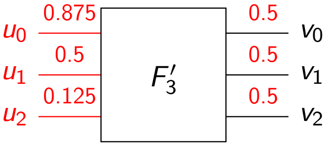
$1 - (1-\epsilon)^3$

$1 - [3(1-\epsilon)^2\epsilon + (1-\epsilon)^3]$

ϵ^3

$\cdots$

$u_i^{(m-1)}$ $\quad \ell_m = 3 \quad$ $u_i^{(m)}$

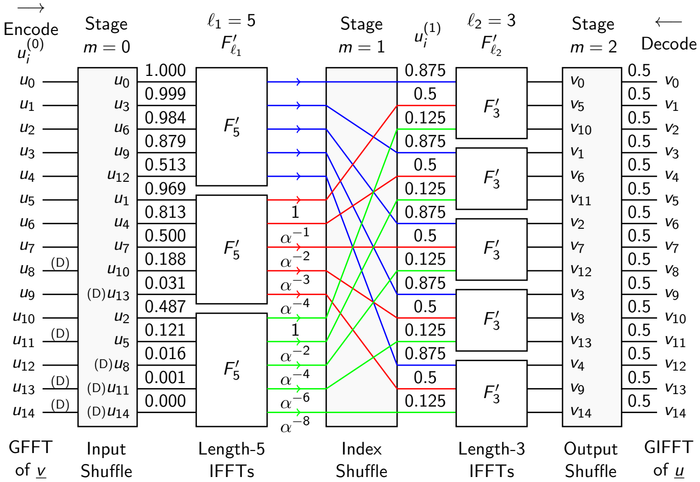


$\cdots$

GFFT of $\underline{v}$ $\quad$ (I)FFT $\quad$ GIFFT of $\underline{u}$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design on QEC

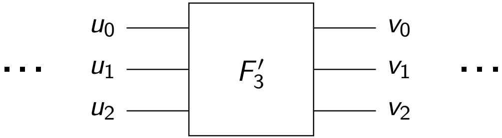


Code Design for Algebraic Errors and Erasures Decoding

q-ary Symmetric Channel with Erasures (QSCE) with $e = 0.5$, $\epsilon = 0$.
Use Berlekamp-Massey (B-M) and Forney algorithms.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\}$$

$u_i^{(m-1)}$ $\ell_m = 3$ $u_i^{(m)}$



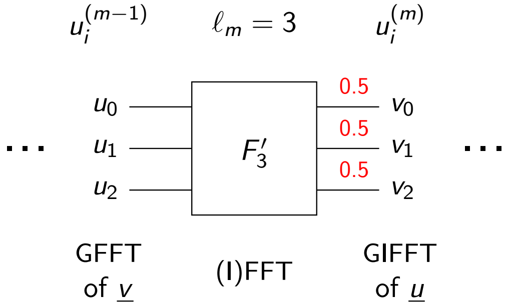
GFFT of $\underline{v}$ (I)FFT GIFFT of $\underline{u}$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Errors and Erasures Decoding

q-ary Symmetric Channel with Erasures (QSCE) with $e = 0.5$, $\epsilon = 0$.
Use Berlekamp-Massey (B-M) and Forney algorithms.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ e = P\{v_i \neq u_i^{(m)}\} = 0.5,\ \epsilon = P\{v_i = ?\} = 0$$



Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Errors and Erasures Decoding

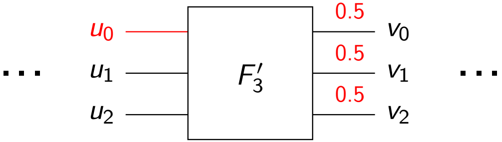
q-ary Symmetric Channel with Erasures (QSCE) with $e = 0.5$, $\epsilon = 0$.
Use Berlekamp-Massey (B-M) and Forney algorithms.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ e = P\{v_i \neq u_i^{(m)}\} = 0.5,\ \epsilon = P\{v_i = ?\} = 0$$

Monte Carlo

$u_i^{(m-1)}$ $\quad \ell_m = 3 \quad$ $u_i^{(m)}$

$e_{u_0} = 0.846,\ \epsilon_{u_0} = 0$



GFFT of $\underline{v}$ $\quad$ (I)FFT $\quad$ GIFFT of $\underline{u}$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Errors and Erasures Decoding

q-ary Symmetric Channel with Erasures (QSCE) with $e = 0.5$, $\epsilon = 0$.
Use Berlekamp-Massey (B-M) and Forney algorithms.

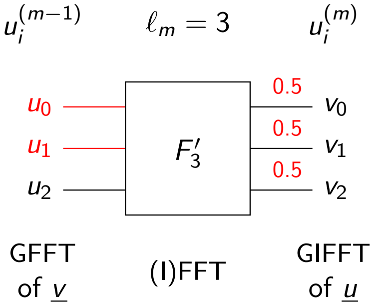
$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ e = P\{v_i \neq u_i^{(m)}\} = 0.5,\ \epsilon = P\{v_i = ?\} = 0$$

Monte Carlo

$e_{u_0} = 0.846,\ \epsilon_{u_0} = 0$

$e_{u_1} = 0.031,\ \epsilon_{u_1} = 0.846$

$\cdots$



$\cdots$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Errors and Erasures Decoding

q-ary Symmetric Channel with Erasures (QSCE) with $e = 0.5$, $\epsilon = 0$.
Use Berlekamp-Massey (B-M) and Forney algorithms.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ e = P\{v_i \neq u_i^{(m)}\} = 0.5,\ \epsilon = P\{v_i = ?\} = 0$$

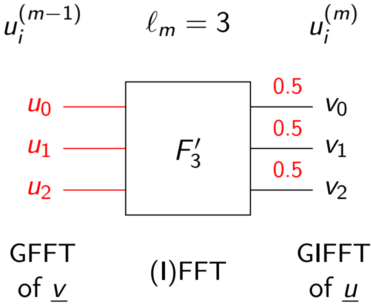
Monte Carlo

$e_{u_0} = 0.846$, $\epsilon_{u_0} = 0$

$e_{u_1} = 0.031$, $\epsilon_{u_1} = 0.846$

$e_{u_2} = 0.055$, $\epsilon_{u_2} = 0.452$

$\cdots$



$\cdots$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Errors and Erasures Decoding

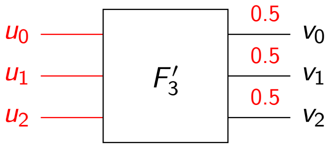
q-ary Symmetric Channel with Erasures (QSCE) with $e = 0.5$, $\epsilon = 0$.
Use Berlekamp-Massey (B-M) and Forney algorithms.

$$u_i \in \mathbb{F}_{16},\ v_i \in \mathbb{F}_{16} \cup \{?\},\ e = P\{v_i \neq u_i^{(m)}\} = 0.5,\ \epsilon = P\{v_i = ?\} = 0$$

Theoretical $\quad u_i^{(m-1)} \quad \ell_m = 3 \quad u_i^{(m)}$

$$1 - (1-e)^3 = 0.875$$
$$1 - (1-e)^3 = 0.875$$
$$1 - [3(1-e)^2 e + (1-e)^3] = 0.5$$

$\cdots$



$\cdots$

GFFT of $\underline{v}$ $\quad$ (I)FFT $\quad$ GIFFT of $\underline{u}$

Uncover next (unknown) input using outputs and recovered inputs.

Code Design for Algebraic Errors and Erasures Decoding

q-ary Symmetric Channel with Erasures (QSCE) with $e = 0.5$, $\epsilon = 0$.
Use Berlekamp-Massey (B-M) and Forney algorithms.

$$u_i \in \mathbb{F}_{256},\ v_i \in \mathbb{F}_{256} \cup \{?\},\ e = P\{v_i \neq u_i^{(m)}\} = 0.5,\ \epsilon = P\{v_i = ?\} = 0$$

Monte Carlo

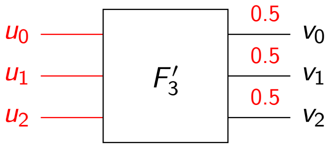
$u_i^{(m-1)}$ $\quad \ell_m = 3 \quad$ $u_i^{(m)}$

$e_{u_0} = 0.875,\ \epsilon_{u_0} = 0$

$e_{u_1} = 0.002,\ \epsilon_{u_1} = 0.875$

$e_{u_2} = 0.003,\ \epsilon_{u_2} = 0.482$

$\cdots$



$\cdots$

GFFT of $\underline{v}$ $\quad$ (I)FFT $\quad$ GIFFT of $\underline{u}$

Uncover next (unknown) input using outputs and recovered inputs.

Decoding Complexity

- For a length-ℓ block, bounded by $C\ell^2$ operations for some $C > 0$.

- Since there are $\prod_{j \neq m} \ell_j = N/\ell_m$ blocks at stage ℓ_m, the decoding complexity is bounded by

$$\sum_{m=1}^{n} \prod_{j \neq m} \ell_j \left(C\ell_m^2 \right) = CN \sum_{m=1}^{n} \ell_m \leq CNn \max_m \ell_m.$$

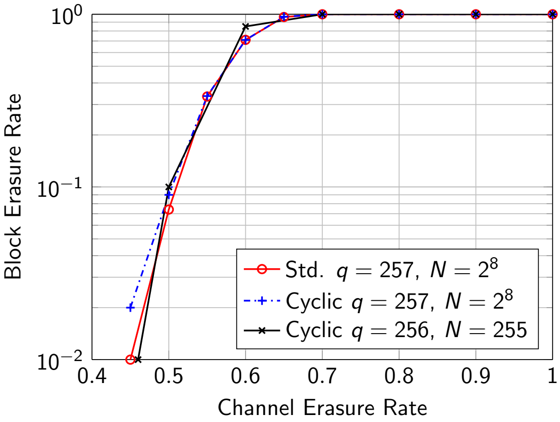
- For comparison, complexity of standard polar codes for $N = 2^n$ is $O(NlogN)$ under SC decoding.

Performance of Cyclic Polar Codes

What is the performance of cyclic polar codes?

Overview

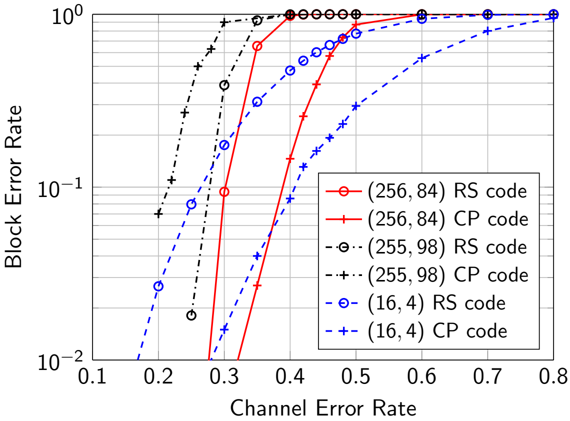
Simulation Results on a q-ary Erasure Channel (QEC)



Comparison of performance of standard polar and cyclic polar codes.

$\delta = 0.1$; $\epsilon = 0.5$ produced rates 0.328 for $N = 256$, 0.384 for $N = 255$.

Simulation Results on a q-ary Symmetric Channel (QSC)



Performance of QEC-designed cyclic polar (CP) codes on the QSC.

$\delta = 0.1$; $\epsilon = 0.5$ produced rates 0.328 for $N = 256$, 0.384 for $N = 255$, 0.25 for $N = 16$.

Overview

Summary

- Cyclic Polar (CP) code construction for arbitrary blocklength N.

Summary

- Cyclic Polar (CP) code construction for arbitrary blocklength N.
- Performance of CP code:
  - QEC: Outperforms standard polar code under deterministic hard-decision SC decoding with much higher rates and larger polarization kernels.

Summary

- Cyclic Polar (CP) code construction for arbitrary blocklength N.
- Performance of CP code:
  - QEC: Outperforms standard polar code under deterministic hard-decision SC decoding with much higher rates and larger polarization kernels.
  - QSC: With the design on QEC, outperforms hard-decision decoding of a similar RS code under soft-decision SC decoding; demonstrated for $N = 2^8$.

Summary

- Cyclic Polar (CP) code construction for arbitrary blocklength N.
- Performance of CP code:
  - QEC: Outperforms standard polar code under deterministic hard-decision SC decoding with much higher rates and larger polarization kernels.
  - QSC: With the design on QEC, outperforms hard-decision decoding of a similar RS code under soft-decision SC decoding; demonstrated for $N = 2^8$.
  - QSC: With the design on QEC, inferior to hard-decision decoding of a similar RS code under algebraic hard-decision SC decoding; demonstrated for $N = 255 = 3 \cdot 5 \cdot 17$.

Future Work

- Theoretical analysis for mixed-size kernels to prove polarization and see if these codes are capacity-achieving on arbitrary q-DMCs.
- Better decoding strategies for finite length cyclic polar codes (essentially better RS decoders for the small blocks).
- Effect of ordering of the factors of blocklength N in the code graph.
- Performance under random and burst errors/erasures.
- Optimizing computational complexity for q-ary alphabets.
- ... and much more!

Thank you!

Questions?

How are the codes cyclic?

- Cyclic codes are (principal) ideals in the ring of polynomials $GF(q)[x]/(x^n - 1)$.
- Find a primitive N^{th} root of unity ω in $GF(q)$.
- The cyclic subgroup generated by ω gives all the N roots of the equation $x^N - 1 = 0$.
- Hence we can factorize as

$$x^N - 1 = \prod_{i=0}^{N-1} (x - \omega^i)$$

- Our generator polynomial $g(x)$ is the product of a subset of these factors and thus divides $x^N - 1$.
- Since $g(x)$ is monic and is the minimal polynomial of degree $N - k$ in $GF(q)[x]/(x^n - 1)$, it is a generator for the ideal.
- Thus the code is cyclic with generator $g(x)$.

Capacity of QSCE

The capacity of q-ary symmetric channel with erasures (QSCE) with parameters α and β representing the probability of symbol erasure and symbol error, respectively, is

$$\boxed{C = (1-\alpha) + (1-\alpha)log_q\left(\frac{1-\alpha-\beta}{1-\alpha}\right) - \beta log_q\left(\frac{1-\alpha-\beta}{\beta}\right) - \beta log_q(q-1)}$$

Hence, the capacities of q-ary erasure channel (QEC) and q-ary symmetric channel (QSC) are

$$\boxed{C_{QEC} = 1-\alpha} \quad \text{and} \quad \boxed{C_{QSC} = 1 - H_q(\beta)}$$

where, $H_q(\cdot)$ is the q-ary entropy function.