Generalized Staircase Codes with Arbitrary Bit Degree

Frank R. Kschischang
Department of Electrical & Computer Engineering
University of Toronto
frank@ece.utoronto.ca

Simons Institute Workshop on Application-Driven Coding
Berkeley, California

March 6, 2024

Acknowledgements

Joint work with:

Mohannad Shehadeh

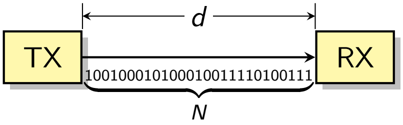
Alvin Y. Sukmadji

Funding:

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada, and in part by Huawei Canada Research Center.

Motivation I: Decoding Window Size

- Distance between transmitter and receiver: d (m)
- Propagation speed: c (m/s)
- Data transmission speed: D (bit/s)
- Bits in transit: $N = \frac{Dd}{c}$ (bit)
- Decoding window size: $\mathcal{O}(N)$



Wireless

$d = 100$ m to 10 km, $D = 1$ Gb/s, $c = 3 \cdot 10^8$ m/s $\Rightarrow N = 0.33$ to 33 kbit.

Mars Link

$d = 225 \cdot 10^6$ km, $D = 2$ Mb/s, $c = 3 \cdot 10^8$ m/s $\Rightarrow N = 1500$ Mbit.

Fiber

$d = 100$ m to 1000 km, $D = 1$ Tb/s, $c = 2 \cdot 10^8$ m/s $\Rightarrow N = 0.5$ to 5000 Mbit.

Motivation II: Power Consumption

- Decoder Power Consumption (W) = Decoder Energy Efficency (J/bit) $\times$ D (bit/s)
- To achieve Decoder Power Consumption of ≈ 1 W, requires

$$\text{Decoder Energy Efficiency} \approx \left(\frac{1 \text{ Watt}}{D}\right).$$

Wireless

$D = 1$ Gb/s $\Rightarrow$ Decoder Energy Efficiency ≈ 1 nJ/bit

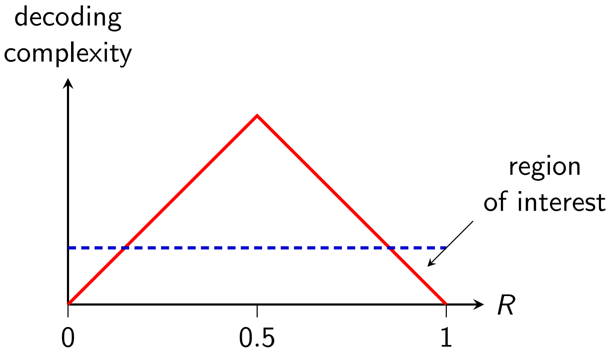
Mars Link

$D = 2$ Mb/s $\Rightarrow$ Decoder Energy Efficiency ≈ 0.5 μJ/bit

Fiber

$D = 1$ Tb/s $\Rightarrow$ Decoder Energy Efficiency ≈ 1 pJ/bit

Motivation II: Power Consumption (cont'd)



For high data rate (fiber-optic communications) applications, we are interested in codes with low decoding complexity:

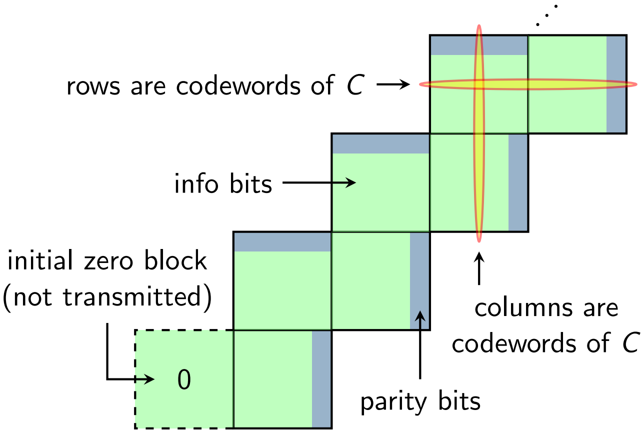
- high code rate: $R \approx 0.9$ to $R \approx 0.99$
- large decoding windows $\approx 10^6$ bits.

Staircase Codes

- Introduced in B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang and J. Lodge, "Staircase Codes: FEC for 100 Gb/s OTN," *J. Lightwave Technology*, vol. 30, Jan. 2012, pp. 110–117.
- Spatially-coupled product codes
- Iterative algebraic decoding in a sliding window
- High-rate (low-complexity) BCH component decoders
- Low error floors with analytic bounds
- Adopted in standards:
  - OIF 400ZR (400 Gb/s, coherent), 2017 (inner Hamming, outer staircase)
  - ITU-T G.709.2 (OTU4 long-reach interface), 2018

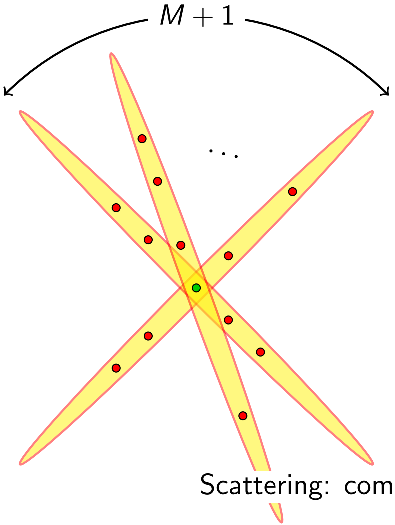
Staircase Codes: Summary

- Each square "staircase block" is of size $S \times S$ bits.
- C is a binary systematic code of length $2S$ (e.g., a t-error-correcting BCH code).
- Iterative decoding occurs within a sliding window: "oldest" block shifted out when "newest block" is filled.
- Every bit is checked by two codes (degree 2)
- Two codes intersect in at most one bit ("scattering" = girth > 4 in Tanner graph with code-constraint vertices)



Increasing Bit Degree while Maintaining Scattering

Minimum weight uncorrectable error pattern contains at least $(M+1)t+1$ errors.



- Error floors below 10^{-15} with $M=1$ usually requires $t \geq 3$.
- Energy cost (J) for t-error-correcting BCH codes scales as $\mathcal{O}(t^2)$, or $\mathcal{O}(t^2/t) = \mathcal{O}(t)$ J per corrected bit.
- This motivates reducing t, compensating with an increased M to maintain low error floor.

Scattering: component codes intersect at most once

Alternative View of Staircase Codes ($M = 1$)

$$
\begin{array}{cc}
0 & B_0 \\
B_0^{\mathsf{T}} & B_1 \\
B_1^{\mathsf{T}} & B_2 \\
B_2^{\mathsf{T}} & B_3 \\
B_3^{\mathsf{T}} & B_4 \\
B_4^{\mathsf{T}} & B_5 \\
B_5^{\mathsf{T}} & B_6 \\
B_6^{\mathsf{T}} & B_7 \\
B_7^{\mathsf{T}} & B_8 \\
B_8^{\mathsf{T}} & B_9 \\
\vdots & \vdots
\end{array}
$$

rows are codewords

not transmitted

transmitted

Generalization: $M > 1$

$$
\begin{array}{ccc}
0 & 0 & B_0 \\
0 & B_0^{\mathsf{T}} & B_1 \\
0 & B_1^{\mathsf{T}} & B_2 \\
B_0^{\pi} & B_2^{\mathsf{T}} & B_3 \\
B_1^{\pi} & B_3^{\mathsf{T}} & B_4 \\
B_2^{\pi} & B_4^{\mathsf{T}} & B_5 \\
B_3^{\pi} & B_5^{\mathsf{T}} & B_6 \\
B_4^{\pi} & B_6^{\mathsf{T}} & B_7 \\
B_5^{\pi} & B_7^{\mathsf{T}} & B_8 \\
B_6^{\pi} & B_8^{\mathsf{T}} & B_9 \\
\vdots & \vdots & \vdots
\end{array}
$$

not transmitted

transmitted

$M = 2$

$$
\begin{array}{cccc}
0 & 0 & 0 & B_0 \\
0 & 0 & B_0^{\mathsf{T}} & B_1 \\
0 & 0 & B_1^{\mathsf{T}} & B_2 \\
0 & 0 & B_2^{\mathsf{T}} & B_3 \\
0 & B_0^{\pi_1} & B_3^{\mathsf{T}} & B_4 \\
0 & B_1^{\pi_1} & B_4^{\mathsf{T}} & B_5 \\
B_0^{\pi_2} & B_2^{\pi_1} & B_5^{\mathsf{T}} & B_6 \\
B_1^{\pi_2} & B_3^{\pi_1} & B_6^{\mathsf{T}} & B_7 \\
B_2^{\pi_2} & B_4^{\pi_1} & B_7^{\mathsf{T}} & B_8 \\
B_3^{\pi_2} & B_5^{\pi_1} & B_8^{\mathsf{T}} & B_9 \\
\vdots & \vdots & \vdots & \vdots
\end{array}
$$

rows are codewords

transmitted

$M = 3$

Notation: intra-block permutations

- Codeword is an infinite sequence $B_0, B_1, B_2, \ldots$ of $S \times S$ matrices satisfying certain constraints
- Assume entries from $\mathbb{F}_2$ for concreteness (but could generally be any alphabet)
- $S \times S$ matrices indexed by $(i, j) \in \{0, 1, \ldots, S-1\} \times \{0, 1, \ldots, S-1\} = [S] \times [S]$
- $M + 1$ permutations π_k indexed by $k \in \{0, 1, \ldots, M\} = [M+1]$:

$$\begin{aligned} \pi_k \colon [S] \times [S] &\longrightarrow [S] \times [S] \\ (i, j) &\longmapsto \pi_k(i, j) \end{aligned}$$

- B^{π_k} is the permuted copy of $B \in \mathbb{F}_2^{S \times S}$ according to π_k

Notation: intra-block permutations (cont'd)

- If

$$B = \begin{bmatrix} b_{(0,0)} & b_{(0,1)} & \cdots & b_{(0,S-1)} \\ b_{(1,0)} & b_{(1,1)} & \cdots & b_{(1,S-1)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(S-1,0)} & b_{(S-1,1)} & \cdots & b_{(S-1,S-1)} \end{bmatrix},$$

- then

$$B^{\pi_k} = \begin{bmatrix} b_{\pi_k(0,0)} & b_{\pi_k(0,1)} & \cdots & b_{\pi_k(0,S-1)} \\ b_{\pi_k(1,0)} & b_{\pi_k(1,1)} & \cdots & b_{\pi_k(1,S-1)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{\pi_k(S-1,0)} & b_{\pi_k(S-1,1)} & \cdots & b_{\pi_k(S-1,S-1)} \end{bmatrix}$$

- Without loss of generality, take $\pi_0 = \mathrm{id}$, so $B^{\pi_0} = B$

Notation: inter-block delays

- Inter-block constraints are causal, characterized by delays
- Consider $M + 1$ distinct non-negative integer block delay values $d_0, d_1, \ldots, d_M$
- Without loss of generality, assume

$$0 = d_0 < d_1 < \cdots < d_M$$

Generalized staircase code construction

- Fix a component code $C \subseteq \mathbb{F}_2^{(M+1)\cdot S}$ of length $(M+1)\cdot S$
- Code is then defined by the constraint that the rows of the matrix

$$\begin{bmatrix} B_{i-d_M}^{\pi_M} & B_{i-d_{M-1}}^{\pi_{M-1}} & \cdots & B_{i-d_2}^{\pi_2} & B_{i-d_1}^{\pi_1} & B_i \end{bmatrix} \in \mathbb{F}_2^{S\times(M+1)\cdot S}$$

  belong to the component code C for all $i \geq d_M$ and that

$$B_{-1} = B_{-2} = \cdots = B_{-d_M} = 0_{S\times S}$$

- Encoding memory is d_M blocks
- If C is linear, systematic with redundancy r, dimension $(M+1)\cdot S - r$

$$R_{\text{unterminated}} = \frac{S\cdot(S-r)}{S\cdot S} = 1 - \frac{r}{S}$$

- Rate of component code C has to be at least $M/(M+1)$

Scattering intra-block permutations

- It is both practically convenient and mathematically sufficient to consider permutations defined by linear-algebraic operations on matrix indices $(i, j) \in [S] \times [S]$
- To do so, must associate the index set $[S] = \{0, 1, \ldots, S-1\}$ with a finite commutative ring $\mathcal{R}$ of cardinality S
- Permutations are then defined by invertible 2×2 matrices:

$$\pi(i, j) = (i, j) \begin{bmatrix} a & b \\ c & d \end{bmatrix} = (ai + cj, bi + dj)$$

where $(ad - bc)^{-1}$ exists in $\mathcal{R}$

Choice of ring $\mathcal{R}$ with $|\mathcal{R}| = S$

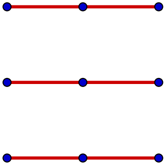
- Intra-block scattering families of permutations are most easily constructed when $\mathcal{R}$ has sufficiently many elements with invertible differences as will be seen shortly
- Finite fields are a good choice since any pair of distinct elements have an invertible difference since all nonzero elements are invertible
- Take $\mathcal{R} = \mathbb{F}_p = \mathbb{Z}_p$ with $S = p$ a prime number, i.e., **perform integer arithmetic on matrix indices modulo a prime-valued S**
- Take $\mathcal{R} = \mathbb{F}_q$ with $S = q$ a prime power and associate $\{0, 1, \ldots, S-1\}$ with $\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{S-2}\}$ where α is a primitive element of $\mathbb{F}_q$
- $\mathcal{R} = \mathbb{Z}_S$ for non-prime S can work if S has a sufficiently large lowest prime factor denoted $\mathrm{lpf}(S)$
- This is because $\{0, 1, \ldots, \mathrm{lpf}(S) - 1\}$ have invertible differences for distinct elements modulo S since $\{\pm 1, \pm 2, \ldots, \pm(\mathrm{lpf}(S) - 1)\}$ are coprime with S

Definition

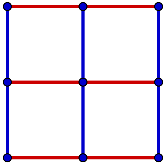
An $(M+1,S)$-net is a set of S^2 elements called "points" along with a collection of $(M+1)$ partitions of these points into subsets of size S called "lines" such that distinct lines intersect in at most one point.

- This is a generalization of the concept of "rows" and "columns" of an $S\times S$ grid which define a $(2,S)$-net
- If the entries of the matrix B are the points and the rows of the $M+1$ matrices B^{π_k} for $k\in\{0,1,\ldots,M\}$ are the $M+1$ partitions, then an $(M+1,S)$-net is equivalent to a special collection of permutations on $[S]\times[S]$
- This collection has the property that for any distinct $k,k'\in\{0,1,\ldots,M\}$, any row of B^{π_k} has at most a single element in common with a row of $B^{\pi_{k'}}$
- $(M+1,S)$-nets are well-studied objects in combinatorics and finite geometry with close connections or equivalences to orthogonal arrays, mutually orthogonal Latin squares, transversal designs, and affine planes
- Standard reference: *Handbook of Combinatorial Designs* by Colbourn and Dinitz

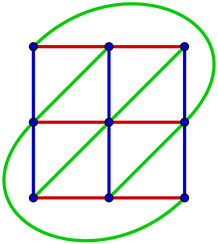
Finite-geometric nets



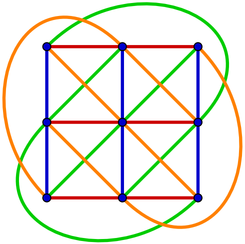
$(1,3)$-net



$(2,3)$-net



$(3,3)$-net



$(4,3)$-net

Theorem (Linear-algebraic intra-block scattering permutations)

A collection of $M+1$ permutations on $\mathcal{R} \times \mathcal{R}$, where $\mathcal{R}$ is a finite commutative ring of cardinality S, defined by a collection of invertible 2×2 matrices define an (M+1,S)-net if and only if, for any pair of distinct matrices $A, \tilde{A}$ in the collection where

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \ \tilde{A} = \begin{bmatrix} \tilde{a} & \tilde{b} \\ \tilde{c} & \tilde{d} \end{bmatrix},$$

we have that $c\tilde{d} - d\tilde{c}$ is invertible in $\mathcal{R}$.

Proof.

By linear algebra. □

Example

If $\mathcal{R} = \mathbb{Z}_p$ with p a prime, the identity permutation $I_{2\times 2}$ together with

$$\begin{bmatrix} 0 & 1 \\ 1 & z \end{bmatrix}$$

for $z \in \{0, 1, \ldots, M-1\}$ define an $(M+1, p)$-net if $M \leq p$. (Theorem condition becomes that $z_1 - z_2$ is invertible for **distinct** $z_1, z_2 \in \{0, 1, \ldots, M-1\}$ which is automatically true in $\mathbb{Z}_p = \mathbb{F}_p$ since $z_1 - z_2 \neq 0$.)

Example

If $\mathcal{R} = \mathbb{Z}_p$ with p a prime, the identity permutation $I_{2\times 2}$ together with the involutions

$$\begin{bmatrix} -z & 1 - z^2 \\ 1 & z \end{bmatrix}$$

for $z \in \{0, 1, \ldots, M-1\}$ define an $(M+1, p)$-net if $M \leq p$.

Example

If $\mathcal{R} = \mathbb{Z}_S$, the identity permutation $I_{2\times 2}$ together with

$$\begin{bmatrix} 0 & 1 \\ 1 & z \end{bmatrix}$$

for $z \in \{0, 1, \ldots, M-1\}$ define an $(M+1, S)$-net if $M \leq \mathrm{lpf}(S)$.

Example

If $\mathcal{R} = \mathbb{F}_q$ with q a prime power, the identity permutation $I_{2\times 2}$ together with

$$\begin{bmatrix} 0 & 1 \\ 1 & z \end{bmatrix}$$

for $z \in \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{M-2}\}$ where α is a primitive element of $\mathbb{F}_q$ form an $(M+1, q)$-net if $M \leq q$.

$M/(M+1)$ overlap problem for $(d_0, d_1, \ldots, d_M) = (0, 1, \ldots, M)$

$$\begin{array}{cc} 0 & B_0 \\ B_0^{\mathsf{T}} & B_1 \\ B_1^{\mathsf{T}} & B_2 \\ B_2^{\mathsf{T}} & B_3 \\ B_3^{\mathsf{T}} & B_4 \\ B_4^{\mathsf{T}} & B_5 \\ B_5^{\mathsf{T}} & B_6 \\ B_6^{\mathsf{T}} & B_7 \\ B_7^{\mathsf{T}} & B_8 \\ B_8^{\mathsf{T}} & B_9 \\ \vdots & \vdots \end{array} \qquad \begin{array}{ccc} 0 & 0 & B_0 \\ 0 & B_0^{\mathsf{T}} & B_1 \\ B_0^{\pi} & B_1^{\mathsf{T}} & B_2 \\ B_1^{\pi} & B_2^{\mathsf{T}} & B_3 \\ B_2^{\pi} & B_3^{\mathsf{T}} & B_4 \\ B_3^{\pi} & B_4^{\mathsf{T}} & B_5 \\ B_4^{\pi} & B_5^{\mathsf{T}} & B_6 \\ B_5^{\pi} & B_6^{\mathsf{T}} & B_7 \\ B_6^{\pi} & B_7^{\mathsf{T}} & B_8 \\ B_7^{\pi} & B_8^{\mathsf{T}} & B_9 \\ \vdots & \vdots & \vdots \end{array}$$

$M = 1$ $\qquad$ $M = 2$

Overlap problem:

- no problem when $M = 1$, but ...
- when $M = 2$, bits in the same row of $[B_{i-2}^{\pi}\ B_{i-1}^{T}\ B_i]$ may appear in the same row of $[B_{i-1}^{\pi}\ B_i^{T}\ B_{i+1}]$, violating the scattering property.
- same issue when $M \geq 2$.

Inter-block scattering delays for $M = 2$

$$
\begin{array}{ccc}
0 & 0 & B_0 \\
0 & B_0^{\mathsf{T}} & B_1 \\
0 & B_1^{\mathsf{T}} & B_2 \\
B_0^{\pi} & B_2^{\mathsf{T}} & B_3 \\
B_1^{\pi} & B_3^{\mathsf{T}} & B_4 \\
B_2^{\pi} & B_4^{\mathsf{T}} & B_5 \\
B_3^{\pi} & B_5^{\mathsf{T}} & B_6 \\
B_4^{\pi} & B_6^{\mathsf{T}} & B_7 \\
B_5^{\pi} & B_7^{\mathsf{T}} & B_8 \\
B_6^{\pi} & B_8^{\mathsf{T}} & B_9 \\
\vdots & \vdots & \vdots
\end{array}
$$

- Take $(d_0, d_1, d_2) = (0, 1, 3)$ instead of $(d_0, d_1, d_2) = (0, 1, 2)$
- No overlap: for all $i \neq j$, $B_i^{\pi_k}$ and $B_j^{\pi_\ell}$ appear side-by-side at most once

Inter-block scattering delays

Definition (Golomb ruler)

The $M+1$ integers $0 = d_0 < d_1 < \cdots < d_M$ referred to as "marks" form a Golomb ruler of order $M+1$ and length d_M if no two distinct pairs of marks are the same distance apart, i.e., have the same difference. An optimal Golomb ruler is the ruler with the shortest length d_M (thus smallest memory) for a given order $M+1$.

- **Any Golomb ruler whether optimal or suboptimal yields inter-block scattering**
- Optimal Golomb rulers are known for all $M+1 \leq 28$ as of today
- Optimal Golomb rulers have at least quadratic length $d_M = \Omega(M^2)$
- Though many more suboptimal or near-optimal constructions are known
- E.g., $d_k = 2^k - 1$ gives a naive construction which is optimal for $M \in \{0, 1, 2\}$ but highly sub-optimal for $M \geq 3$ (exponential length $d_M = 2^M - 1$)
- Good rulers for $M+1 \leq 65000$: `www.cs.toronto.edu/~apostol/golomb/`

Optimal Golomb Rulers

order	length	optimal Golomb ruler (see Wikipedia for sources)
1	0	0
2	1	0 1
3	3	0 1 3
4	6	0 1 4 6
5	11	0 1 4 9 11
6	17	0 1 4 10 12 17
7	25	0 1 4 10 18 23 25
8	34	0 1 4 9 15 22 32 34
9	44	0 1 5 12 25 27 35 41 44
10	55	0 1 6 10 23 26 34 41 53 55
11	72	0 1 4 13 28 33 47 54 64 70 72
12	85	0 2 6 24 29 40 43 55 68 75 76 85
13	106	0 2 5 25 37 43 59 70 85 89 98 99 106
14	127	0 4 6 20 35 52 59 77 78 86 89 99 122 127
15	151	0 4 20 30 57 59 62 76 100 111 123 136 144 145 151

LUT-free decoding of systematic Hamming codes

- Consider a $[2^m, 2^m - (m+1)]$ extended Hamming code whose parity-check matrix has columns given by the binary representations of $2j+1$ for $j \in \{0, 1, \ldots, 2^m - 1\}$

$$H = \begin{bmatrix} 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & \ldots & 1 \\ 0 & 1 & 0 & \ldots & 1 \\ 1 & 1 & 1 & \ldots & 1 \end{bmatrix}$$

- Famously, syndrome decoding of such a code is trivial with the first m bits of the syndrome column interpreted as an integer directly giving the error location $j \in \{0, 1, \ldots, 2^m - 1\}$
- Unfortunately, H must have its columns permuted in order to have a corresponding systematic generator matrix thus ruining this decoding method

Systematic Hamming Codes (cont'd)

- However, if we can find a systematizing permutation which is algebraically-defined so that it's easy to invert, this problem is solved
- Interpret the column index set $\{0, 1, \ldots, 2^m - 1\}$ as the ring of integers modulo 2^m, i.e., $\mathbb{Z}_{2^m}$
- All odd integers are invertible modulo 2^m so any odd $a \in \mathbb{Z}_{2^m}$ and possibly non-odd $b \in \mathbb{Z}_{2^m}$ define an algebraic (affine) permutation τ

$$\begin{aligned} \tau : \mathbb{Z}_{2^m} &\longrightarrow \mathbb{Z}_{2^m} \\ j &\longmapsto \tau(j) = a \cdot j + b \end{aligned}$$

$$\begin{aligned} \tau^{-1} : \mathbb{Z}_{2^m} &\longrightarrow \mathbb{Z}_{2^m} \\ j &\longmapsto \tau^{-1}(j) = a^{-1} \cdot (j - b) \end{aligned}$$

Systematic Hamming Codes (cont'd)

- τ and τ^{-1} are trivially implemented in software/hardware as a simple $(m+1)$-bit add-and-multiply
- By a computer search, we found values of a and b for all $m \in \{3, 4, \ldots, 15, 16\}$ such that the the parity check matrix H permuted according to τ has a systematic generator matrix
- This means that both decoding these Hamming codes as well as generating columns of H can be done as simply as computing τ and τ^{-1}
- Both such steps must be done repeatedly in a higher-order staircase code with high bit degree and many Hamming components
- The alternative would be to have multiple possibly redundant LUTs with 2^m $(m+1)$-bit entries wherever these computations are needed
- Adaptation to a code shortened in the first s positions is accomplished by replacing b with $b + a \cdot s$

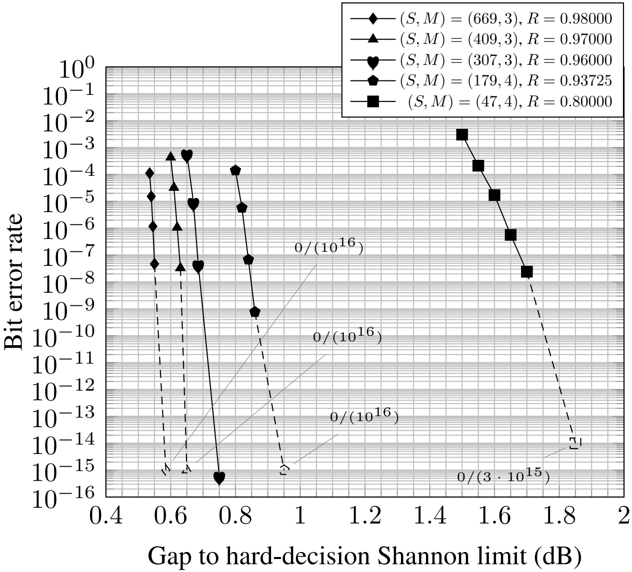
Systematizing affine permutations for extended Hamming codes

m	a	b	a^{-1}
3	1	1	1
4	3	0	11
5	3	0	11
6	3	3	43
7	5	5	77
8	9	11	57
9	19	19	27
10	27	27	531
11	53	53	541
12	89	89	2025
13	163	170	4875
14	301	308	13989
15	553	553	14873
16	1065	1155	55321

Performance results (extended Hamming components)

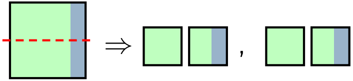
- Highly-optimized C-based software simulators achieve simulated throughputs of several Gbps per core of a modern consumer multi-core CPU allowing for direct verification of sub-10^{-15} error floors
- Relative to conventional staircase codes with $t = 3$, performance is roughly 0.2 to 0.5 dB worse in terms of the gap to the hard-decision Shannon limit as the code rate ranges from 0.98 down to 0.8
- Under concatenation with soft-decoded inner codes, gap can be much smaller: e.g., using proposed codes as drop-in replacement for CFEC rate $239/255$ outer code which is paired with rate $120/128$ soft-decision Hamming code, loss is under 0.2 dB
- $M = 3$ or $M = 4$ seems to suffice for getting sub-10^{-15} error floors
- Memory (decoding window size) is always smaller

Simulation Results



Final Generalization

- For L an integer divisor of S, subdivide $S \times S$ codeword blocks evenly into $(S/L) \times S$ wide sub-blocks.

  e.g., $L = 2$: 

- Define a new code in terms of $(S/L) \times (S/L)$ blocks.
- This preserves rate, but can reduce memory.

Example $(L = 2,\ M = 2)$

Golomb ruler

Golomb ruler

$$
\begin{array}{cccccc}
 & & & & B_0 & B_1 \\
 & & & & B_2 & B_3 \\
 & & & B_0^{\mathsf{T}} & B_4 & B_5 \\
 & & & B_2^{\mathsf{T}} & B_6 & B_7 \\
 & & & B_4^{\mathsf{T}} & B_8 & B_9 \\
 & & B_0^{\pi} & B_6^{\mathsf{T}} & B_{10} & B_{11} \\
 & B_1^{\mathsf{T}} & B_2^{\pi} & B_8^{\mathsf{T}} & B_{12} & B_{13} \\
B_1^{\pi} & B_3^{\mathsf{T}} & B_4^{\pi} & B_{10}^{\mathsf{T}} & B_{14} & B_{15} \\
B_3^{\pi} & B_5^{\mathsf{T}} & B_6^{\pi} & B_{12}^{\mathsf{T}} & B_{16} & B_{17} \\
B_5^{\pi} & B_7^{\mathsf{T}} & B_8^{\pi} & B_{14}^{\mathsf{T}} & B_{18} & B_{19} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

- Now require L Golomb rulers, each with $M + 1$ marks, having disjoint distance spectra
- Intra-block permutations $\{\pi_0, \ldots, \pi_M\}$ can be "recycled" in different phases

Difference Triangle Sets

Definition

An (L, M)-DTS is a set of L Golomb rulers of order $M+1$ whose respective sets of distances are disjoint. Equivalently, it is a set of L rulers of order $M+1$ given by

$$d_0^{(\ell)} < d_1^{(\ell)} < \cdots < d_M^{(\ell)}$$

for $\ell \in [L]$ such that all positive differences

$$d_{k_2}^{(\ell)} - d_{k_1}^{(\ell)}$$

for $k_1, k_2 \in [M+1]$, $k_2 > k_1$, and $\ell \in [L]$ are distinct.

- See, e.g, Chapter 19 of Colbourn and Dinitz (eds), *Handbook of Combinatorial Designs*, 2007.

L-uniform Ruler

Definition

An L*-uniform* ruler of order $M'+1$ is a set of integers $d_0 < d_1 < \cdots < d_{M'}$ for which

$$\left|\{d_k : k \in [M'+1], d_k \equiv \ell \bmod L\}\right| = \frac{M'+1}{L}$$

for each $\ell \in [L]$.

- In other words, elements of $\{0, \ldots, L-1\}$ appear equally often as residues of $\{d_0, \ldots, d_{M-1}\}$ mod L.
- Necessarily, $M'+1 = L(M+1)$ for some positive integer $M+1$ and we can construct any such ruler from L *base* rulers of order $M+1$ given by

$$d_0^{(\ell)} < d_1^{(\ell)} < \cdots < d_M^{(\ell)}$$

for each $\ell \in [L]$ as

$$\{d_k : k \in [M'+1]\} = \{Ld_k^{(\ell)} + \ell : k \in [M+1], \ell \in [L]\}.$$

Higher-order Staircase Codes

Ingredient 1 (A difference triangle set)

An (L, M)-DTS given by

$$0 = d_0^{(\ell)} < d_1^{(\ell)} < \cdots < d_M^{(\ell)}$$

for $\ell \in [L]$ with corresponding L-uniform ruler of order $L(M+1)$

$$d_0 < d_1 < \cdots < d_{L(M+1)-1}$$

given accordingly as

$$\{d_k : k \in [L(M+1)]\} = \{Ld_k^{(\ell)} + \ell : k \in [M+1], \ell \in [L]\}$$

Higher-order Staircase Codes

Ingredient 2 (geometric net)

An $(M+1, S/L)$-net with corresponding $M+1$ permutations of $[S/L] \times [S/L]$ given by π_k for $k \in [M+1]$ where π_0 is the identity permutation, and a resulting collection of $L(M+1)$ permutations of $[S/L] \times [S/L]$ given by $\pi'_{k'} = \pi_k$ for every $k \in [M+1]$ and $k' \in [L(M+1)]$ such that $d_{k'} \in \{Ld_k^{(\ell)} + \ell : \ell \in [L]\}$

Ingredient 3 (component code)

A component code C of length $(M+1)S$ and dimension $(M+1)S - r$

Higher-order Staircase Codes

Definition

A *higher-order staircase code* of rate $1 - r/S$ is defined by the constraint on the bi-infinite sequence of $(S/L) \times (S/L)$ matrices $\dots, B_{-2}, B_{-1}, B_0, B_1, B_2, \dots$ that the rows of

$$\left(B_{n-d_{L(M+1)-1}}^{\pi'_{L(M+1)-1}} \middle| \cdots \middle| B_{n-d_1}^{\pi'_1} \middle| B_{n-d_0}^{\pi'_0}\right)$$

belong to C for all $n \in L\mathbb{Z}$.

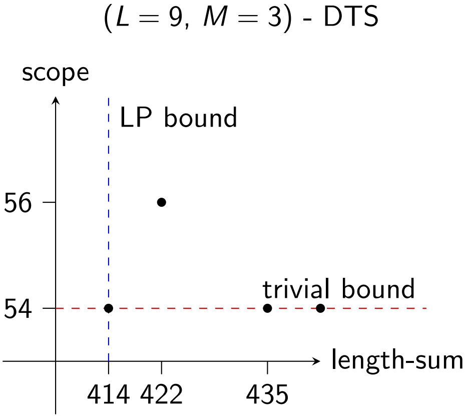
Recovery of Well-Known Codes

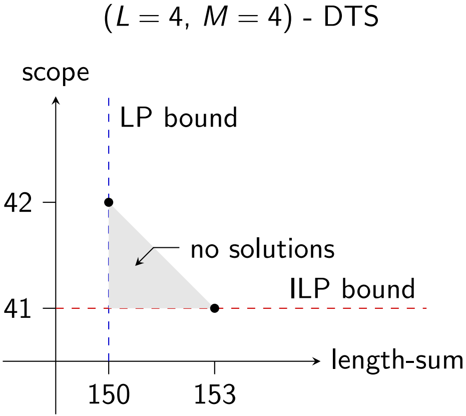
- When $L = M = 1$, the classical staircase codes of Smith, et al. (2012) are recovered.
- When $L > 1$ and $M = 1$, the tiled diagonal zipper codes of Sukmadji, et al. (2022) are recovered.
- When $S = L$ and $r = 1$, a recursive (rather than feedforward) version of the self-orthogonal convolutional codes of Robinson and Bernstein (1967), et al. are recovered

On Difference Triangle Sets

- Usually characterized by **scope**: the maximum length of the constituent rulers.
- The scope is a proxy for the *decoding memory* requirement.
- The **length-sum** — an unstudied DTS parameter — is the sum of the lengths of the constituent rulers.
- The length-sum is a proxy for the *encoding memory* requirement.
- Minimum scope DTSs and minimum length-sum DTSs do not necessarily coincide.
- For $M = 1$, a minimum scope $(L, 1)$-DTS is given by $\{0, 1\}$, $\{0, 2\}$, . . ., $\{0, L\}$.
- For $M = 2$, minimum scope $(L, 2)$-DTSs have an explicit construction (Skolem, 1957)

Scope versus Length-Sum of DTSs (Examples)





To Do

- search for DTSs that are optimal with respect to the scope and length-sum tradeoff
- extend simulations to larger L and larger t ($t = 2$ in particular)
- evaluate performance/complexity tradeoffs
- evaluate performance in concatenation with suitable soft-decision inner code