Miscorrection-free Decoding of Staircase Codes

Christian Häger[1,2] and Henry D. Pfister[2]

[1]Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden

[2]Department of Electrical and Computer Engineering, Duke University, Durham, USA

CHALMERS

FORCE

Abstract

We propose a novel decoding algorithm for staircase codes which can prevent or revert most undetected decoding errors, also known as miscorrections. The algorithm significantly improves performance, while retaining a low-complexity implementation suitable for high-speed optical transport networks (OTNs).

Motivation

- Hard-decision forward error correction (FEC) offers dramatically reduced complexity compared to soft-decision FEC. Applications include regional/metro OTNs [1], optical data center interconnects [2], etc.
- Focus here: Staircase codes [3]—built from short component codes and decoded via iterative bounded-distance decoding (BDD)
- Problem: undetected decoding errors, or miscorrections, may arise during BDD $\implies$ additional errors are introduced (on top of transmission errors) during iterative decoding
- Leads to significant performance degradation in practice [3], [5], [7]
- Notoriously difficult to analyze theoretically [4], [6], [8]

Staircase Codes

Staircase codes: Let $\mathcal{C}$ be a component code of length n and dimension k. A staircase code is defined as the set of all matrix sequences $\mathbf{B}_k \in \{0,1\}^{a\times a}$, $k = 0, 1, 2, \dots$, such that the rows in $[\mathbf{B}_{k-1}^\intercal, \mathbf{B}_k]$ for all $k \geq 1$ form valid codewords of $\mathcal{C}$, where $a = n/2$ and $\mathbf{B}_0$ is the all-zero matrix.

Component codes: We use extended t-error-correcting BCH codes.

Conventional decoding: ℓ iterations of BDD within a sliding window comprising W received blocks (matrices). Component codewords are identified by a pair (i,j), where $i \in \{1,2,\dots,W-1\}$ is the position relative to the current window (see Figure 1) and $j \in \{1,2,\dots,a\}$ enumerates codewords.

```
1 k ← 0
2 while true do
3     for l = 1, 2, ..., ℓ do
4         for i = W − 1, W − 2, ..., 1 do
5             for j = 1, 2, ..., a do
6                 apply BDD to component codeword (i, j)
7     output decision for B_k and shift window
8     k ← k + 1
```

Proposed Anchor-Based Decoding

Miscorrections: Let $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ be a received component codeword, i.e., $\boldsymbol{c} \in \mathcal{C}$, $\boldsymbol{e} \in \{0,1\}^n$. Applying BDD to $\boldsymbol{r}$ can result in

(1) $\boldsymbol{c} \in \mathcal{C}$ if $d_{\mathrm{H}}(\boldsymbol{r},\boldsymbol{c}) = w_{\mathrm{H}}(\boldsymbol{e}) \leq t$,

(2) $\boldsymbol{c}' \in \mathcal{C}$ if $w_{\mathrm{H}}(\boldsymbol{e}) > t$ and $d_{\mathrm{H}}(\boldsymbol{r},\boldsymbol{c}') \leq t$,

(3) FAIL otherwise,

where d_{H}, w_{H} are Hamming distance and weight. Case (2) corresponds to a miscorrection; often ignored for theoretical analysis (referred to as idealized decoding).

- Miscorrections lead to inconsistencies: two component codewords that protect the same bit may disagree on its value
- Idea: make certain codewords anchors and trust their decisions
- Codewords can lose anchor status if they conflict with too many other codewords
- Estimated error locations for codeword (i,j) are denoted by $\mathcal{E}_{i,j} \subset \{1,\dots,n\}$

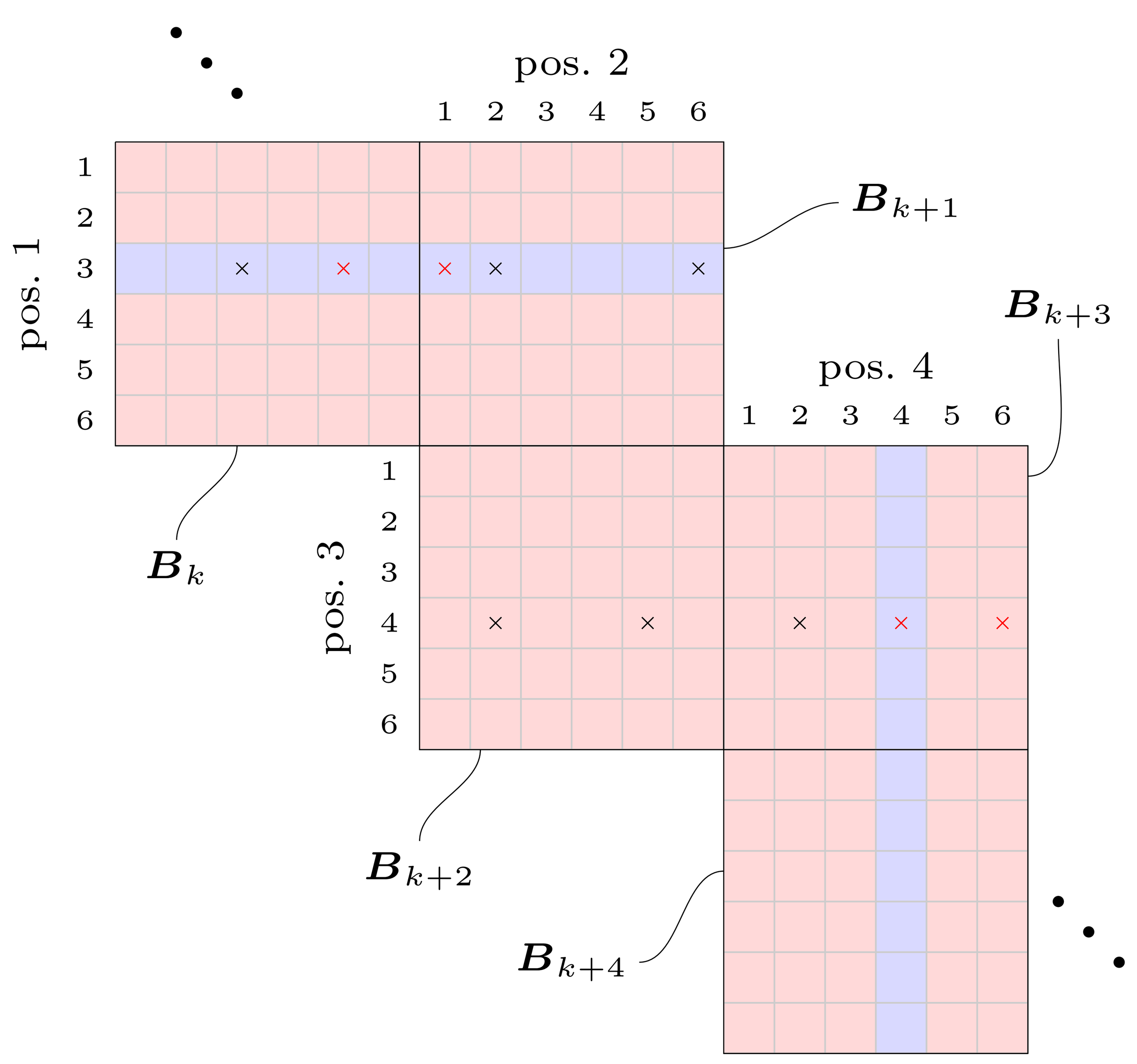


Figure 1: Staircase window of size $W = 5$ with $n = 12$

Algorithm: (replaces line 6 in Alg. above)

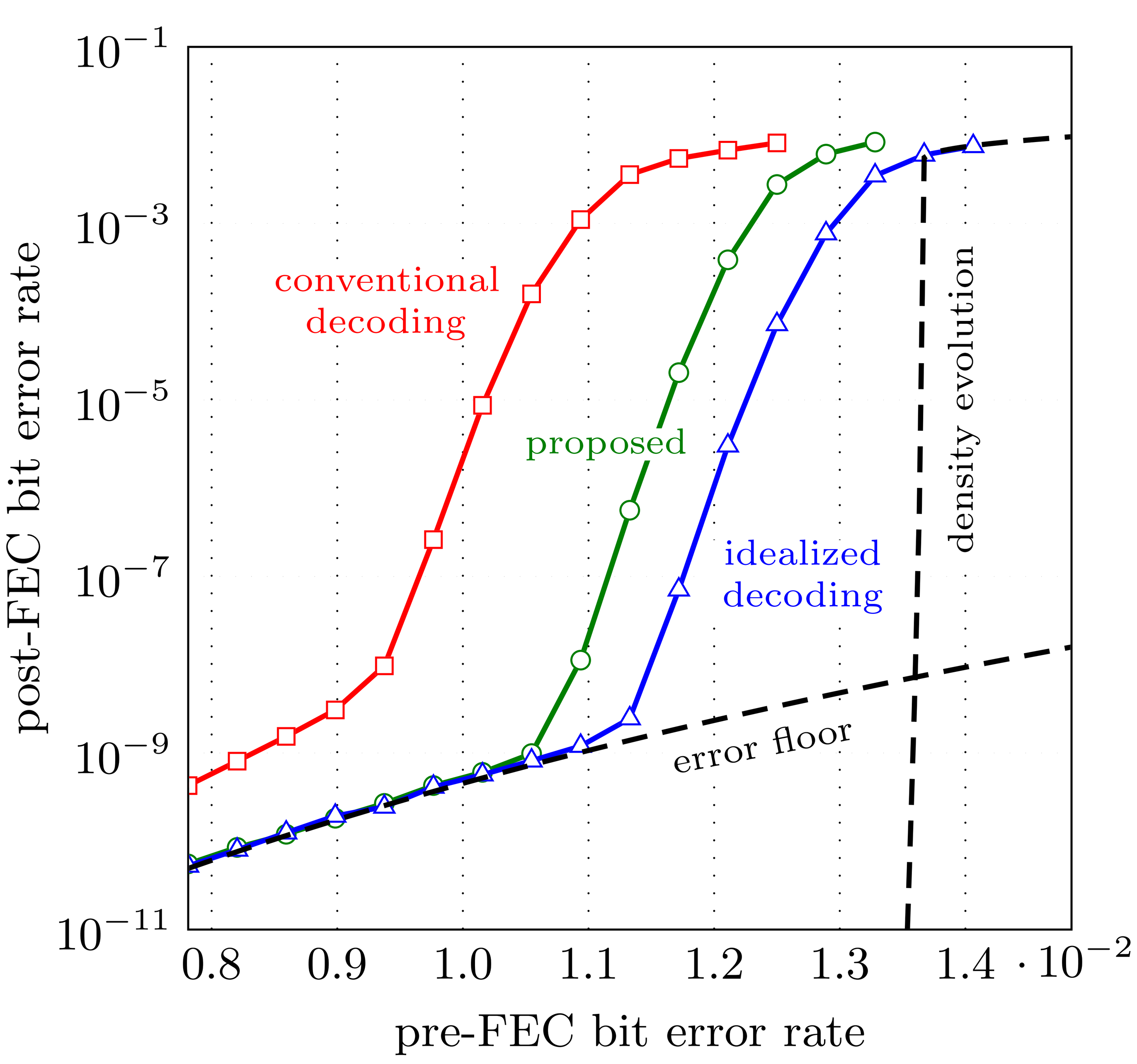
1. If codeword (i,j) is frozen or BDD fails, skip to next codeword, else go to step 2
2. For each error location $e \in \mathcal{E}_{i,j}$, check if bit flip is consistent with anchors. If not, freeze (i,j) and skip to next codeword
3. Flip bits in $\mathcal{E}_{i,j}$ and make (i,j) anchor
4. Backtrack anchors with too many conflicts: revert previously applied bit flips

Examples: (see Figure 1)

Preventing miscorrections: Codeword $(i,j) = (3,4)$ has 3 errors (black crosses). BDD miscorrects with $\mathcal{E}_{3,4} = \{10,12\}$ (red crosses). Assuming that $(4,4)$ is an anchor, codeword $(3,4)$ is frozen and no bits are flipped.

Reverting miscorrections: Let codeword $(1,3)$ be a miscorrected anchor, $\mathcal{E}_{1,3} = \{5,7\}$. Assume that $(i,j) = (2,1)$. The codeword $(2,1)$ has $\mathcal{E}_{2,1} = \{3\}$ and is frozen during step 2. The next codeword $(2,2)$ has $\mathcal{E}_{2,2} = \{3,10\}$. The bit flip at $e = 3$ is inconsistent with $(1,3)$, but this anchor is already in conflict with $(2,1)$. The anchor is backtracked in step 4 and all bits in $\mathcal{E}_{2,2}$ are flipped in step 3.

Results ($n = 256$, $t = 2$, $\ell = 7$, $W = 8$)



Conclusions

- Post-FEC performance of staircase codes significantly improved by adopting a novel anchor-based decoding algorithm
- For BCH component codes with error-correction capability $t = 2$, net coding gain improvements of around $0.4\,\mathrm{dB}$ at bit error rate 10^{-9}
- Error-floor reduction by over an order of magnitute, giving virtually miscorrection-free performance

References

[1] Justesen *et al.*, "Error correcting coding for OTN," *IEEE Commun. Mag.* **59**, 70–75 (2010).

[2] Yu *et al.*, "FPGA demonstration of stretched continuously interleaved BCH code with low error floor for short-range optical transmission," in *Proc. Optical Fiber Communication Conf. (OFC)* (San Diego, 2017)

[3] Smith et al., "Staircase codes: FEC for 100 Gb/s OTN," J. Lightw. Technol. **30**, 110–117 (2012).

[4] Häger *et al.*, "Density evolution for deterministic generalized product codes on the binary erasure channel," *IEEE Trans. Inf. Theory* **63**, 4357–4378 (2017).

[5] Jian et al., "Approaching capacity at high rates with iterative hard-decision decoding," in "Proc. IEEE Int. Symp. Information Theory (ISIT)," (Cambridge, 2012).

[6] Truhachev *et al.*, "Decoding analysis accounting for mis-corrections for spatially-coupled split-component codes," in *Proc. IEEE Int. Symp. Information Theory (ISIT)* (Barcelona, Spain, 2016).

[7] J. Justesen, "Performance of product codes and related structures with iterated decoding," IEEE Trans. Commun. **59**, 407–415 (2011).

[8] Holzbaur *et al.*, "Improved decoding and error floor analysis of staircase codes," *arXiv:1704.01893* (2017).