Decoding of Block Turbo Codes

Mathematical Methods for Cryptography

Dedicated to Celebrate Prof. Tor Helleseth's 70th Birthday

September 4-8, 2017

Kyeongcheol Yang

Pohang University of Science and Technology

Outline

- Product codes
- Block turbo codes (BTCs)
- Soft-input soft-output (SISO) decoding
- Decoding of BTCs Based on the Chase algorithm
- Proposed decoding algorithms for BTCs
- Conclusions

Product Codes

- Product codes were proposed by Elias in 1954 [1].

- Advantages

  ✓ Efficient construction for long codes

$$[n_1, k_1, d_1] \otimes [n_2, k_2, d_2] \;\rightarrow\; [n_1 n_2,\, k_1 k_2,\; d_1 d_2]$$

  ✓ Low-complexity decoding

$$O(n^2) \;\rightarrow\; O(n^{3/2})$$

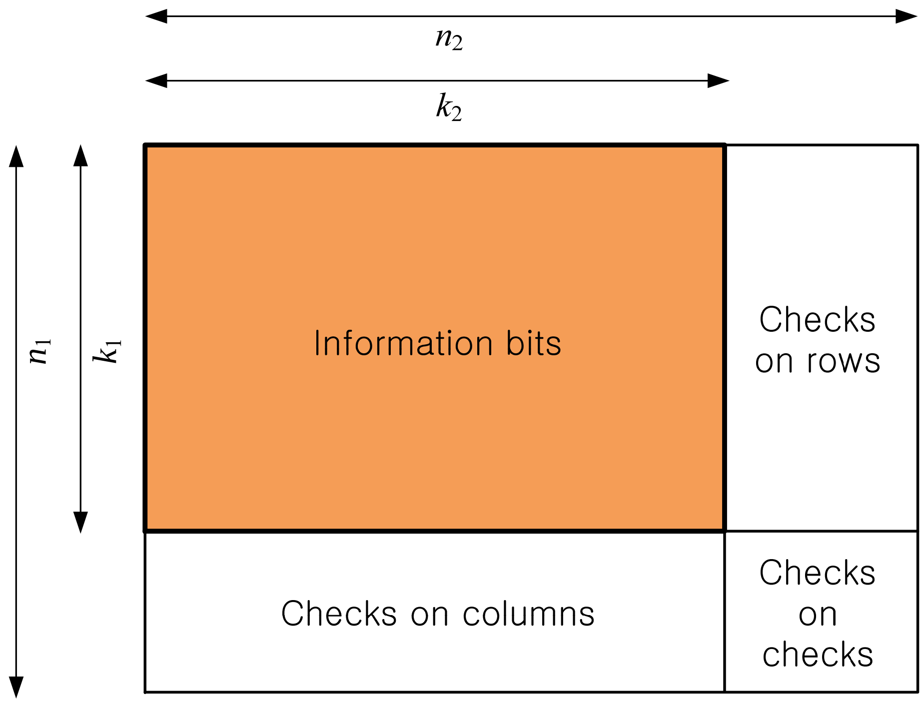
  assuming that codes of length l have decoding complexity $O(l^2)$

  ✓ Robust to burst errors

---

[1] P. Elias, "Error-free coding," *IRE Trans. on Information Theory*, vol. IT-4. pp. 29-37, Sept. 1954.

Product Codes: Construction and Encoding

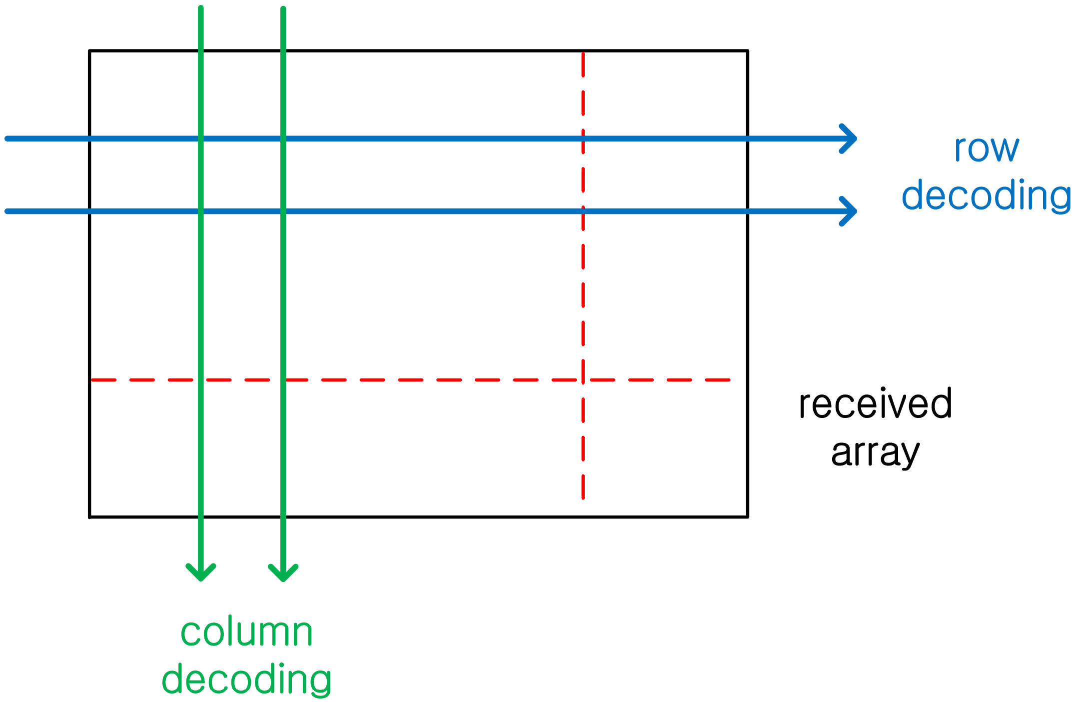


- Parameters
  - length: $n = n_1 n_2$
  - inf. Length: $k = k_1 k_2$
  - min. distance: $d = d_1 d_2$
  - rate: $R = R_1 R_2$

- Encoding
  - ✓ Column encoding by an $[n_1, k_1, d_1]$ code.
  - ✓ Row encoding by an $[n_2, k_2, d_2]$ code.

- The constructed code is an $[n_1 n_2, k_1 k_2, d_1 d_2]$ linear code.

Product Codes: Decoding



- Decoding
  - ✓ Column decoding by an $[n_1, k_1, d_1]$ code.
  - ✓ Row decoding by an $[n_2, k_2, d_2]$ code.
- Hard-decision decoding is conventionally performed only once

Product Codes: Component Codes

- Component codes
  - ✓ Typically, high rate codes are employed.
  - ✓ Hamming codes or extended Hamming codes
  - ✓ BCH codes or extended BCH codes

    …

- Usually, these codes are algebraically decoded.
  - ✓ Berlekamp-Massey algorithm
  - ✓ Euclidean decoding algorithm

    …

- Under algebraic decoding (hard-decision decoding), iterative decoding do not improve the performance of a product code.

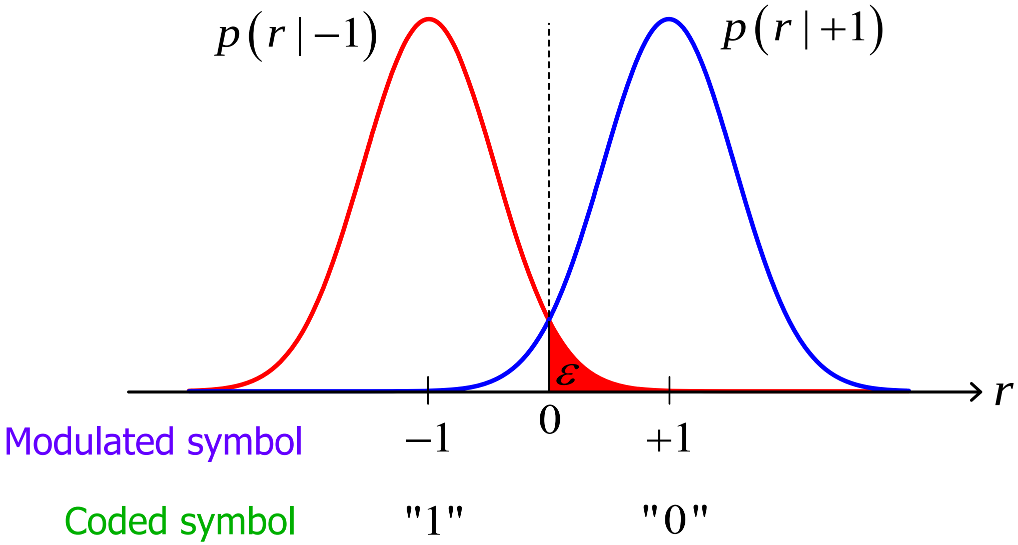
Hard-Decision vs. Soft-Decision Decoding

- Assume that binary phase-shift keying (BPSK) is employed over the additive white Gaussian noise (AWGN) channel.

- The ouput of a matched filter at the receiver is

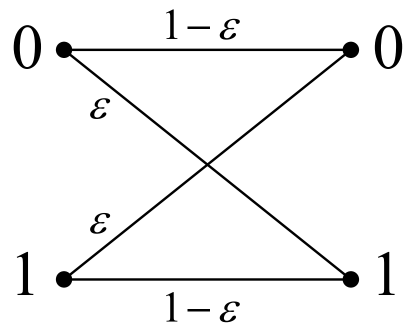
$$r = \pm 1 + z \qquad\qquad z \sim N(0, \sigma^2)$$

- Binary-input AWGN (BI-AWGN) channel



Hard-Decision vs. Soft-Decision Decoding

- Hard-decision:



Binary symmetric channel (BSC)

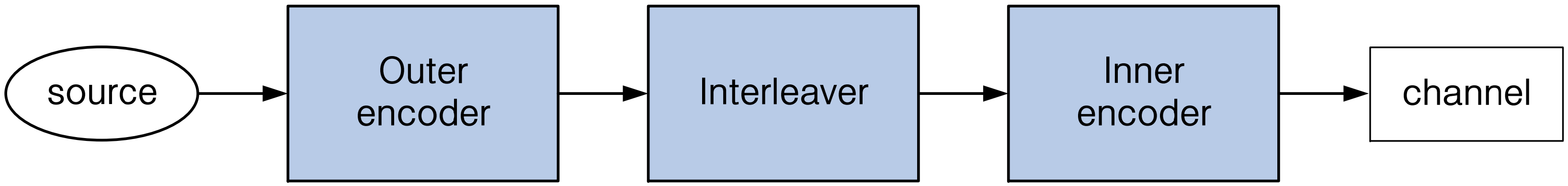
- Soft-decision

$$\text{LLR (log-likelihood ratio)} \quad = \quad \frac{p(r \mid +1)}{p(r \mid -1)} \quad = \quad \frac{2}{\sigma^2} r$$

- The asymptotic coding gain of soft-decision decoding over hard-decision decoding is 3 dB.

Concatenated Codes: A Generalization

- Concatenated codes
  - ✓ Proposed by Forney in 1965 [2]
  - ✓ A generalization of product codes by an interleaver



- As an inner code, soft-decision decodable codes are strongly recommended for better performance.

- Best combination for the AWGN Channel before the turbo era:

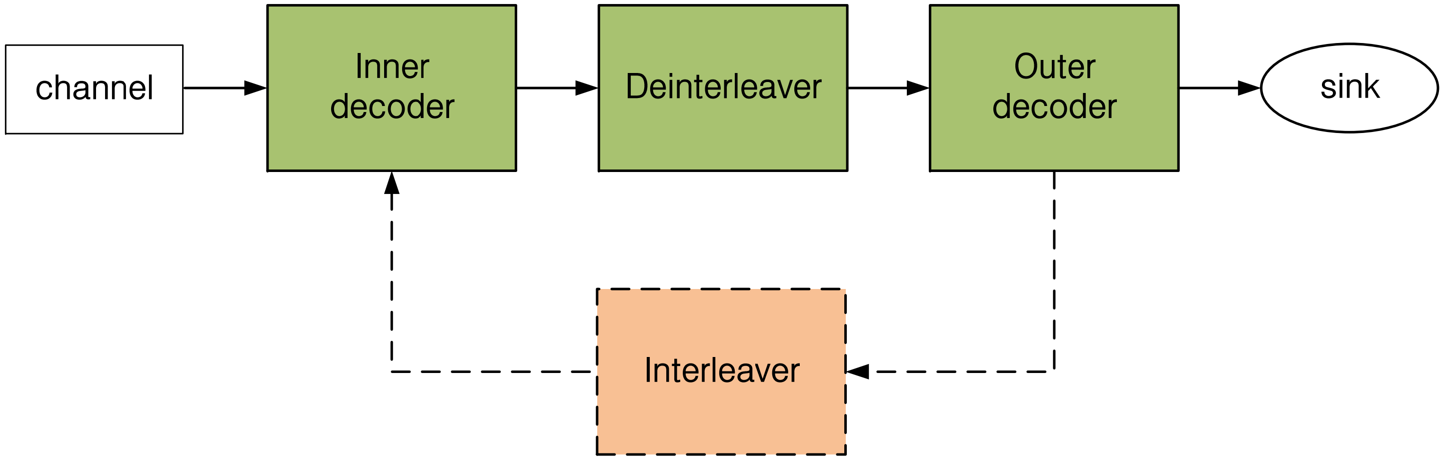
  Reed-Solomon + Convolutional codes (Viterbi algorithm)

---

[2] G. D. Forney, Concatenated Codes, *Ph.D. Dissertation, MIT 1965*.

Concatenated Codes: Decoding

POSTECH CSDL



- Inner and outer codes are decoded only once.

- Iterative decoding (turbo principle)
  - ✓ Inner and outer codes can be iteratively decoded, if they are supported by soft-input soft-output decoders.
  - ✓ Then the overall performance can be significantly decoded.

Block Turbo Codes

- Turbo codes
  - ✓ Invented by Berrou, Glavieux, and Thitmajshima in 1993 [3]
  - ✓ Parallel concatenated codes
  - ✓ Convolutional codes as component codes
  - ✓ Soft-input soft-output (SISO) decoder for convolutional codes
  - ✓ Iterative decoding
  - ✓ capacity-approaching performance

- Block turbo codes (BTCs)
  - ✓ Introduced by Pyndiah [4],[5]
  - ✓ Product codes: serially concatenated codes
  - ✓ Block codes as component codes
  - ✓ Large minimum Hamming distance
  - ✓ SISO decoder for block codes: a bottleneck for decoding of BTCs.
  - ✓ Iterative decoding

---

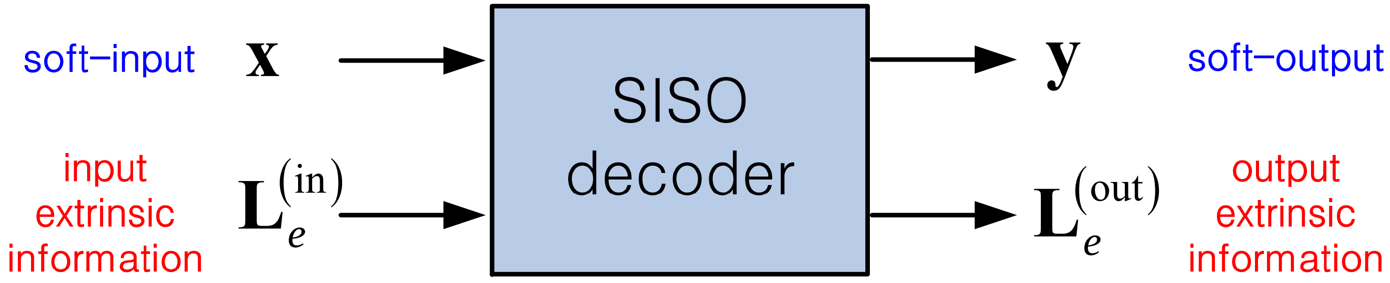
[3] C. Berrou, A. Glavieux, and P. Thitmajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)," *ICC 1993.*

[4] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near optimum decoding of product codes," in *Proc. IEEE GLOBECOM 1994*, vol. 1, pp. 339-343, Nov.-Dec. 1994.

[5] R. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE TCOM*, vol. 46, no. 8, Aug. 1998.

SISO Decoding for Block Codes

- Soft-input soft-output (SISO) decoding



- For convolutional codes, the BCJR Algorithm supports SISO decoding.

- For graph-based codes, SISO decoding can be implemented by message-passing algorithms such as the sum-product algorithms for low-density parity check (LDPC) codes.

- In this talk, we consider block codes which are algebraically constructed.

SISO Decoding for Block Codes

- SISO decoding for block codes can be implemented in two stages:
  - ✓ Soft-decision decoding
  - ✓ Extraction of the extrinsic information

- Soft-decision decoding for block codes
  - ✓ Maximum-likelihood (ML) decoding
  - ✓ Trellis-based decoding
  - ✓ List-based decoding

Maximum-Likelihood (ML) Decoding

- ML decoding is equivalent to minimum distance decoding over the AWGN channel:

$$\boxed{\mathbf{D}=\mathbf{C}^i \quad \text{if} \quad \left\|\mathbf{R}-\varphi\left(\mathbf{C}^i\right)\right\|^2 \le \left\|\mathbf{R}-\varphi\left(\mathbf{C}^j\right)\right\|^2, \quad \forall j \in \left[1,\ 2^k\right],\ j \ne i}$$

impractical for long codes!

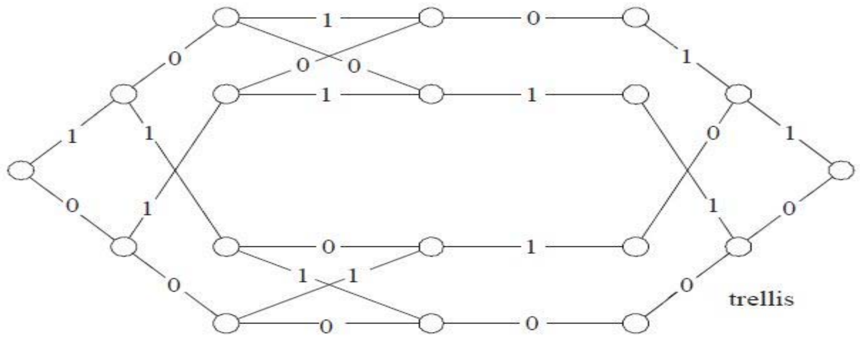
- k : information length of a row or a column code
- $\mathbf{R}=\left(r_1, r_2, \ldots, r_n\right)$: received signal vector
- $\mathbf{D}=\left(d_1, d_2, \ldots, d_n\right) \in C$: optimum decision codeword
- $\mathbf{C}^i=\left(c_1^i, c_2^i, \ldots, c_n^i\right) \in C$: ith codeword of a code C
- $\varphi(\cdot)$: mapping function from $\{0,1\}$ to $\{-1,+1\}$

- ML decoding is optimal in the sense that the block error rate is minimized.
- However, ML decoding is not feasible for high-rate codes.

Trellis-Based Decoding for Block Codes

- Trellis representation of a block code

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



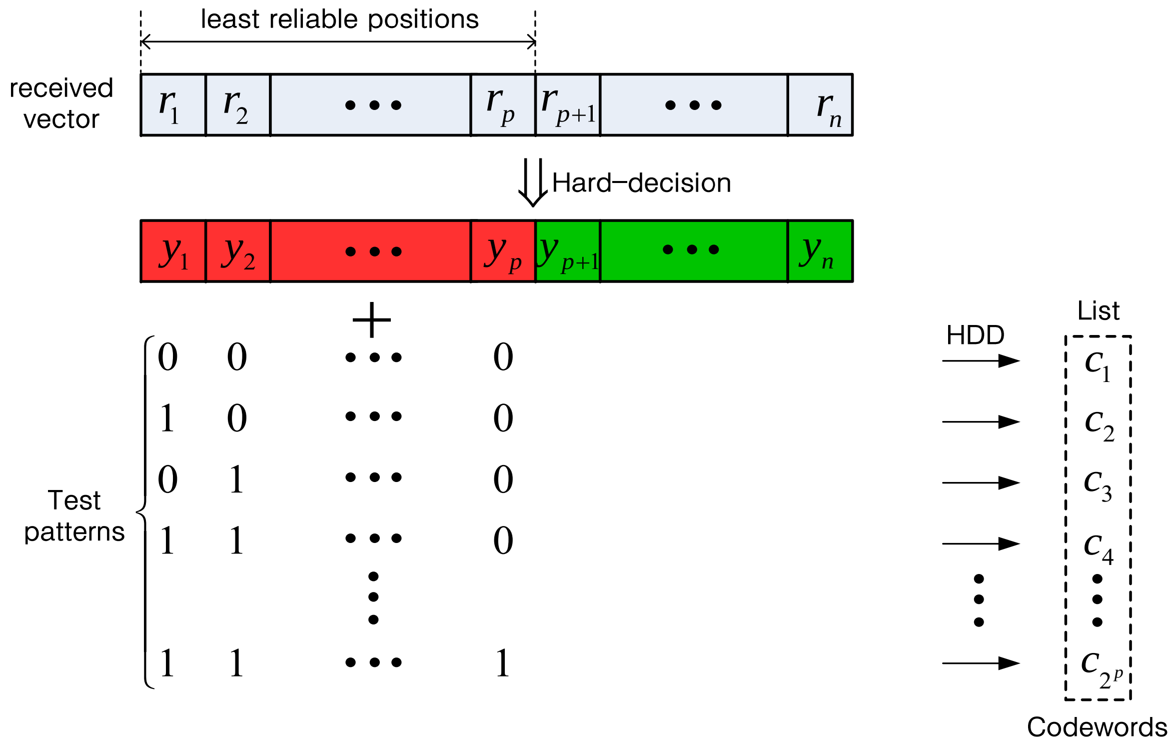
- The Viterbi algorithm or BCJR algorithm is employed.
- Disadvantages
  - ✓ The corresponding trellis is not time-invariant, but time-varying.
  - ✓ The complexity of trellis representation is very high.

    Number of states $\approx \min(2^k, 2^{n-k}) \rightarrow \infty$

  - ✓ Trellis-based decoding has high complexity.

[6] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. 24, no. 1, Jan. 1978.

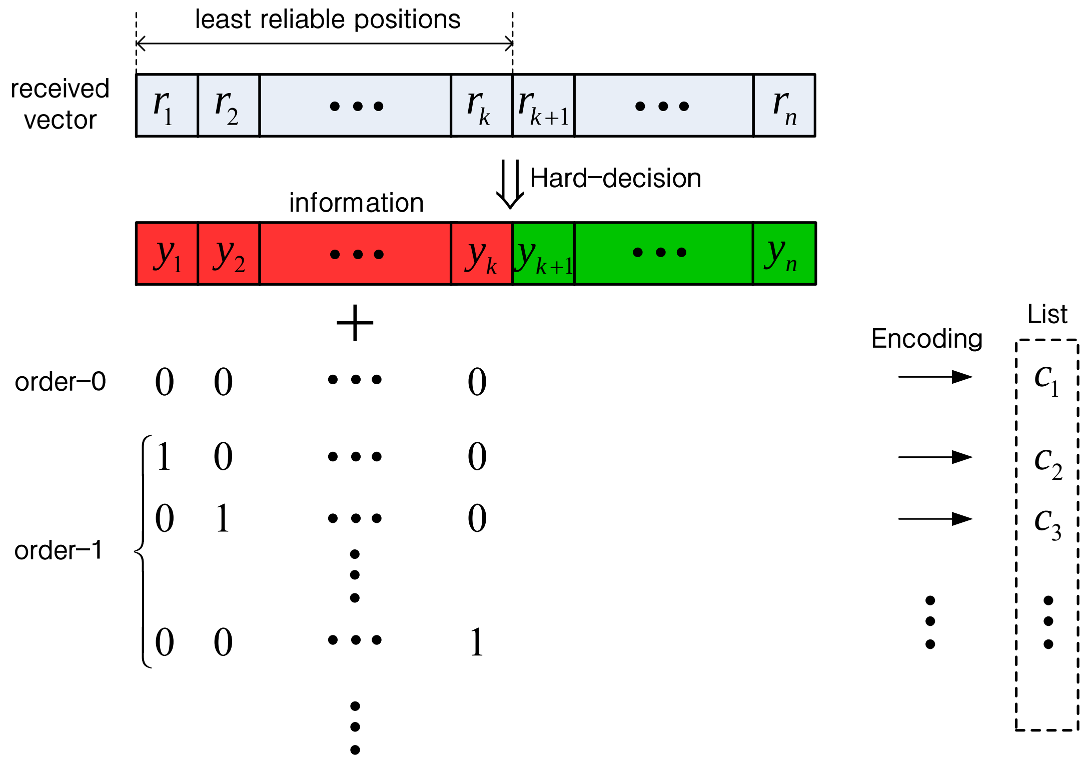
List-Based Decoding: Chase Decoding



- Chase Decoding [7]
  - ✓ Choose some least reliable positions of the received vector
  - ✓ Generate test sequences from the hard-decision vector of the received vector
  - ✓ Decode them by hard-decision decoding
  - ✓ Make a list of candidate codewords
  - ✓ An decision codeword is determined from the list.

[7] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 1, Aug. 1972.

List-Based Decoding: OSD



- Ordered Statistics Decoding (OSD)
  - ✓ Choose some largest reliable positions of the received vector
  - ✓ Generate test information vectors
  - ✓ Encode them into codewords
  - ✓ Make a list of candidate codewords
  - ✓ An decision codeword is determined from the list.

[8] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inform. Theory*, vol. 41, no. 5, pp. 1379-1396, Sep. 1995.

Decoding of Block Turbo Codes

- Each component code of a BTC is decoded in two stages for iterative decoding

- At the first stage, the Chase algorithm is employed.
  - ✓ Choose some least reliable positions of the received vector
  - ✓ Generate test sequences from the hard-decision vector of the received vector
  - ✓ Decode them by hard-decision decoding
  - ✓ Make a list of candidate codewords
  - ✓ An decision codeword is determined from the list.

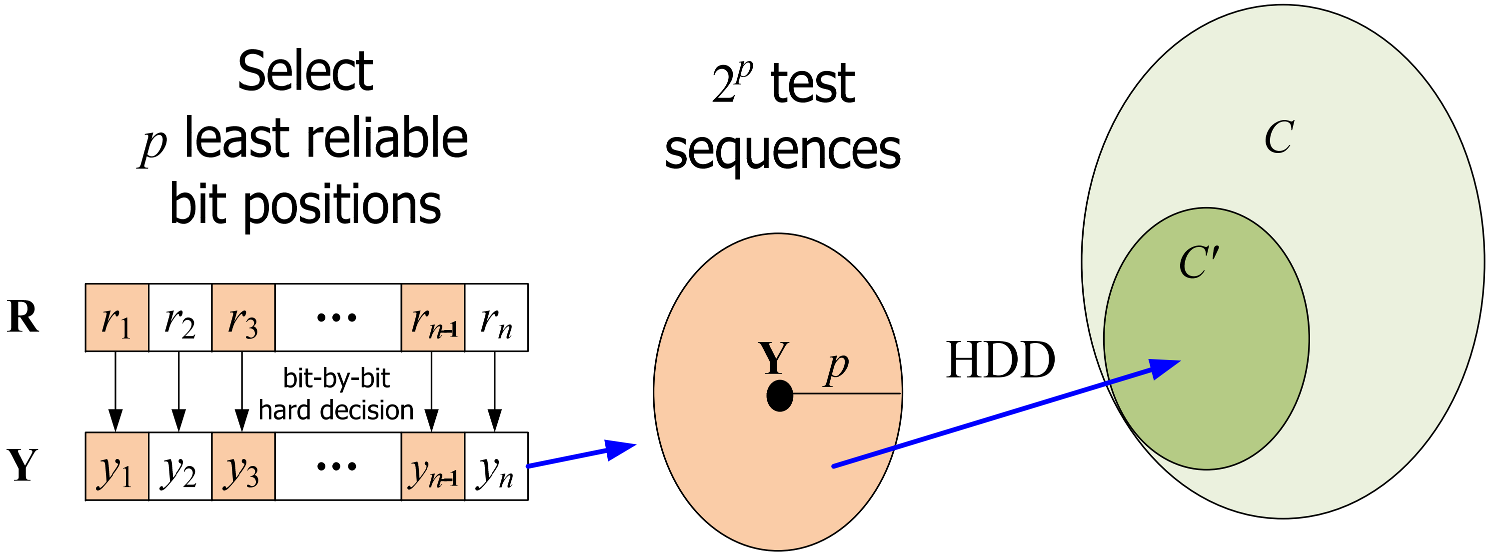
- At the second stage, the extrinsic information is computed for iterative decoding.

- Encoding-based decoding algorithms such OSD may be employed at the first stage

Decoding of Block Turbo Codes

- Iterative decoding
  - ✓ Suboptimum
  - ✓ Two-stage decoding for each row or column vector of the received array
  - ✓ Decode columns first and then rows in turn
  - ✓ Extrinsic information is fed back

- First stage: Use the Chase algorithm



Decoding of BTCs: First Stage

(1) Obtain the hard-decision vector $\mathbf{Y}$ from the input vector $\mathbf{R}$.

(2) Find the p least reliable bit (LRB) positions in $\mathbf{R}$.

(3) Construct 2^p test patterns $\mathbf{T}^j = \left(t_1^j, t_2^j, \ldots, t_n^j\right),\ j = 1, \ldots, 2^p$
where t_l^j is set to 0 or 1 at the p LRB positions
and zero at the remaining positions.

(4) Construct 2^p test sequences (TSs) $\mathbf{Z}^j = \mathbf{Y} \oplus \mathbf{T}^j$
where $\oplus$ is the component-wise modulo-2 sum
operator.

(5) Apply an algebraic HDD to $\mathbf{Z}^j$.

(6) Compute $\left\|\mathbf{R} - \varphi\left(\mathbf{C}^j\right)\right\|^2, \ \forall j \in \left[1,\ 2^p\right]$.

(7) Select a decision codeword $\mathbf{D} = \left(d_1, d_2, \ldots, d_n\right)$ as

$$\mathbf{D} = \arg\min_{\mathbf{C}^j} \left\|\mathbf{R} - \varphi\left(\mathbf{C}^j\right)\right\|^2.$$

Decoding of BTCs: Second Stage

POSTECH CSDL Communications and Signal Design Lab.

(1) Compute the extrinsic information for the l th bit of the decision codeword as

$$w_l = \begin{cases} \frac{1}{4}\left(\left\|\mathbf{R}-\varphi\left(\mathbf{B}^l\right)\right\|^2 - \left\|\mathbf{R}-\varphi\left(\mathbf{D}\right)\right\|^2\right)\times\varphi\left(d_l\right)-r_l\,, & \text{if } \mathbf{B}^l \text{ exists} \\ \beta\times\varphi\left(d_l\right), & \text{otherwise.} \end{cases}$$

Reliability factor

where $\mathbf{B}^l = \left(b_1, b_2, \ldots, b_n\right) = \arg\min_{\mathbf{C}^j,\, c_l^j \neq d_l} \left\|\mathbf{R}-\varphi\left(\mathbf{C}^j\right)\right\|^2$ is a competing codeword.

(2) Input to the next-iteration decoder is updated as follows:

$$\mathbf{R}_{\text{next}}\left(t\right) = \mathbf{R} + \alpha\left(t\right)\mathbf{W}\left(t-1\right)$$

$t = 1, 2, \cdots, t_{\max}$
Current iteration number

Weighting factor

$\mathbf{W}\left(t-1\right) = \left(w_1, w_2, \cdots, w_n\right)$
Extrinsic information vector from the previous decoder

Decoding of BTCs: Choice of α and β

- Selection of weighting and reliability factors

  - ✓ The optimal weighting factor α and reliability factor β are obtained experimentally through trial and error.

  - ✓ Experimentally, BTCs show good error performance when

$$\alpha(t) = [0.0,\ 0.2,\ 0.3,\ 0.5,\ 0.7,\ 0.9,\ 1.0]$$

$$\beta(t) = [0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0,\ 1.0,\ 1.0]$$

Decoding of BTCs: Issues

POSTECH CSDL
Communications and Signal Design Lab.

- Issues for the conventional decoding algorithm
  - ✓ Decoding complexity
  - ✓ Performance

- Limitations of the conventional decoding algorithm
  - ✓ Employs the Chase algorithm with p fixed, regardless of the SNR or the number of iterations.
  - ✓ The number of hard-decision decoding for each row or column vector is fixed, regardless of the reliability of a given decoder input vector.

Decoding of BTCs: Issues

- Modification of the first stage
  - ✓ Use test pattern elimination:

    Fragiocomo et al. (1999), Hirst et al. (2001),

    Chi et al. (2004), Chen et al. (2009), etc.
  - ✓ Replace the Chase algorithm by OSD

    Fossorier et al. (2002), Fang et al. (2000), etc.

- Modified extraction of the extrinsic information at the second stage
  - ✓ Adaptive scaling:

    Picart and Pyndiah (1999), Martin and Taylor (2000), etc
  - ✓ Amplitude clipping:

    Zhang and Le-Ngoc (2001)

Proposed Algorithm I

- Proposed algorithm I
  - ✓ Check whether the employed HDD outputs a codeword for a given decoder input vector.
  - ✓ Apply one of two estimation rules.

- Based on these two rules, the number of TSs can be made monotonically decreasing with iterations.

- Advantages
  - ✓ can significantly reduce the decoding complexity
  - ✓ with a negligible performance loss,

  compared with the conventional decoding algorithm.

---

[9] J. Son, K. Cheun, and K. Yang, "Low-Complexity Decoding of Block Turbo Codes Based on the Chase Algorithm," IEEE Communications Letters, vol. 21, no. 4, pp. 706-709, Apr. 2017.

Proposed Algorithm I

- Case 1: For a given decoder input vector $\mathbf{Y}$, the employed HDD outputs a codeword $\mathbf{C_Y}$ with

$$d_H\left(\mathbf{Y},\mathbf{C_Y}\right)\le 1.$$

✓ Observation: With high probability,
$\mathbf{C_Y}$ is equal to the transmitted codeword.

✓ Estimation Rule 1:

(1) Estimate $\mathbf{C_Y}$ as the decision codeword $\mathbf{D}$ without applying the Chase algorithm; and

(2) Compute the extrinsic information as

$$w_l=\gamma\times\varphi\left(d_l\right)\qquad l=1,2,\dots,n$$

where γ is a reliability factor larger than β.

Proposed Algorithm I

- Case 2: For a given decoder input vector $\mathbf{Y}$,
  the employed HDD outputs a codeword $\mathbf{C}_{\mathbf{Y}}$ with $d_H\left(\mathbf{Y},\mathbf{C}_{\mathbf{Y}}\right)>1$
  or it does not give any codeword due to a decoding failure.

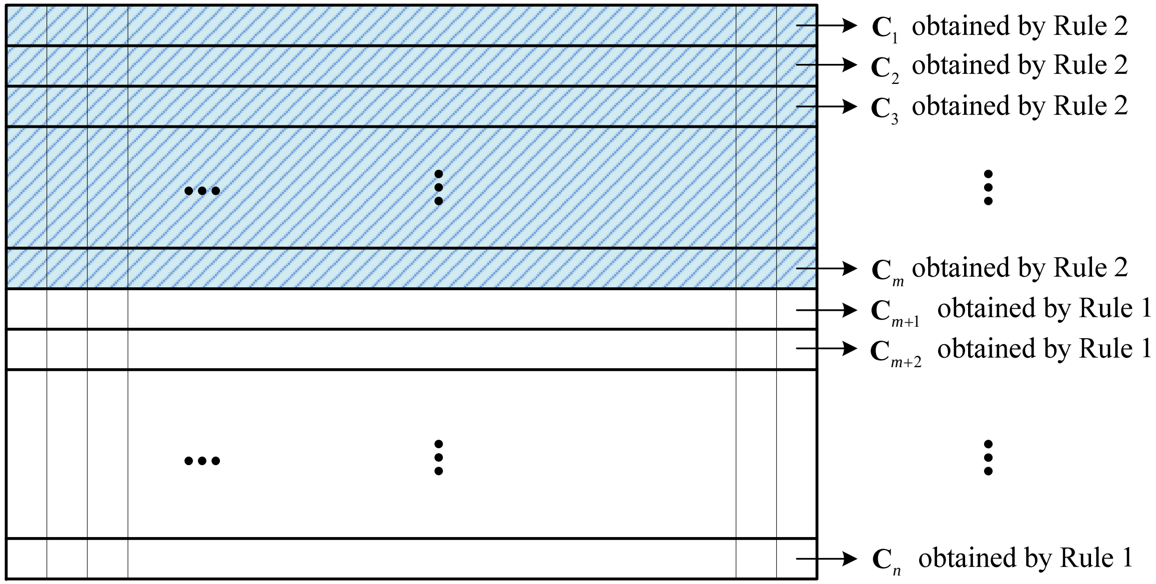
✓ Estimation Rule 2:

(1) Apply the Chase algorithm with parameter p
to get a decision codeword; and

(2) Compute the extrinsic information
by the conventional method

✓ The key to Estimation Rule 2 is to determine how to evolve p with half-iteration.

Proposed Algorithm I



- The partial average distance between the hard-decision vectors and the decision codewords obtained by Rule 2 for the received array at the ith half-iteration is defined by

$$\hat{d}_i = \frac{1}{m}\sum_{j=1}^{m} d_H\left(\mathbf{Y}_j, \mathbf{C}_j\right).$$

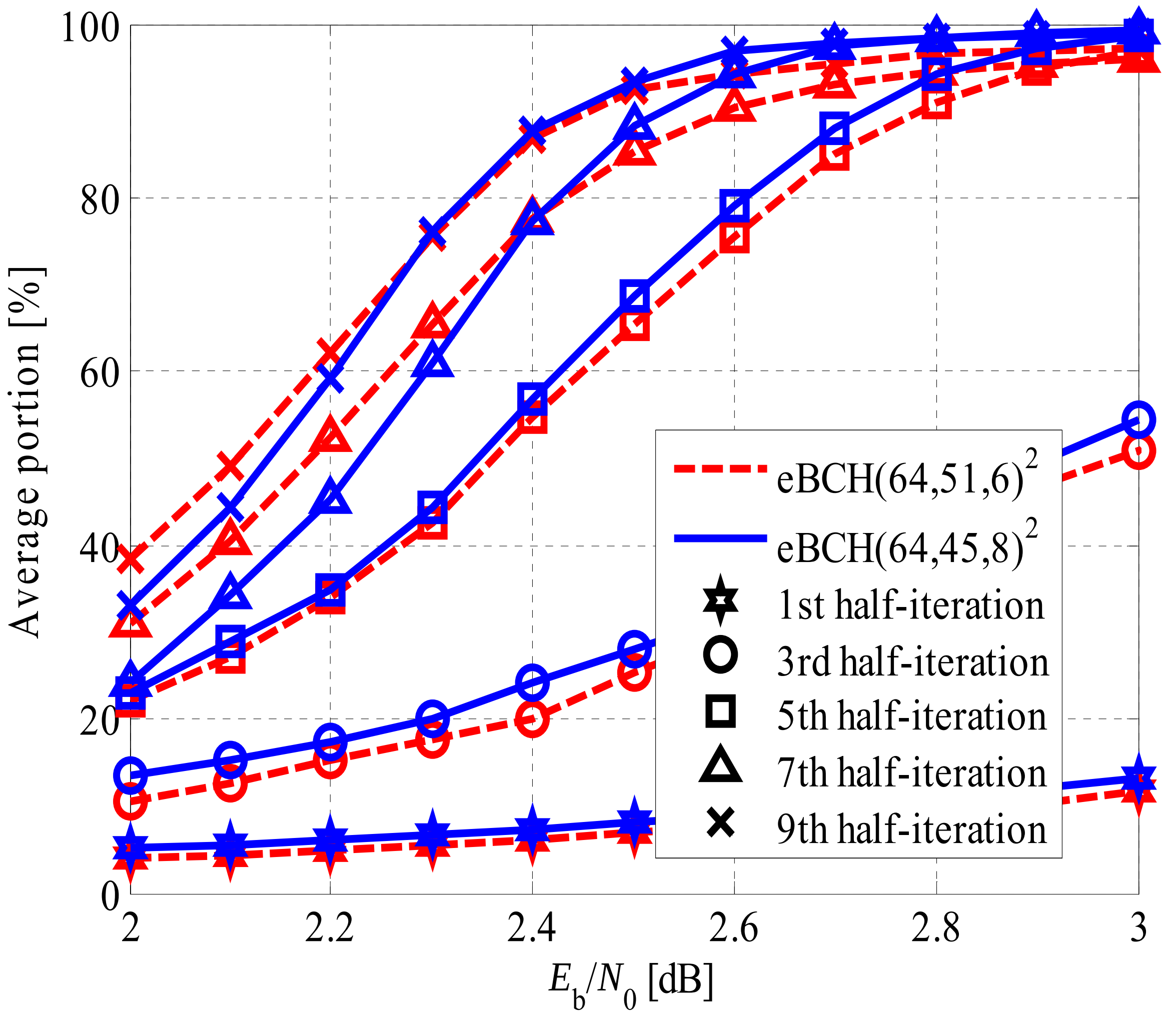
- The parameter p may be evolved as

$$p_{i+1} = \left\lfloor a\hat{d}_i \right\rfloor + b \le p_i \le p.$$

Proposed Algorithm I: Numerical Results

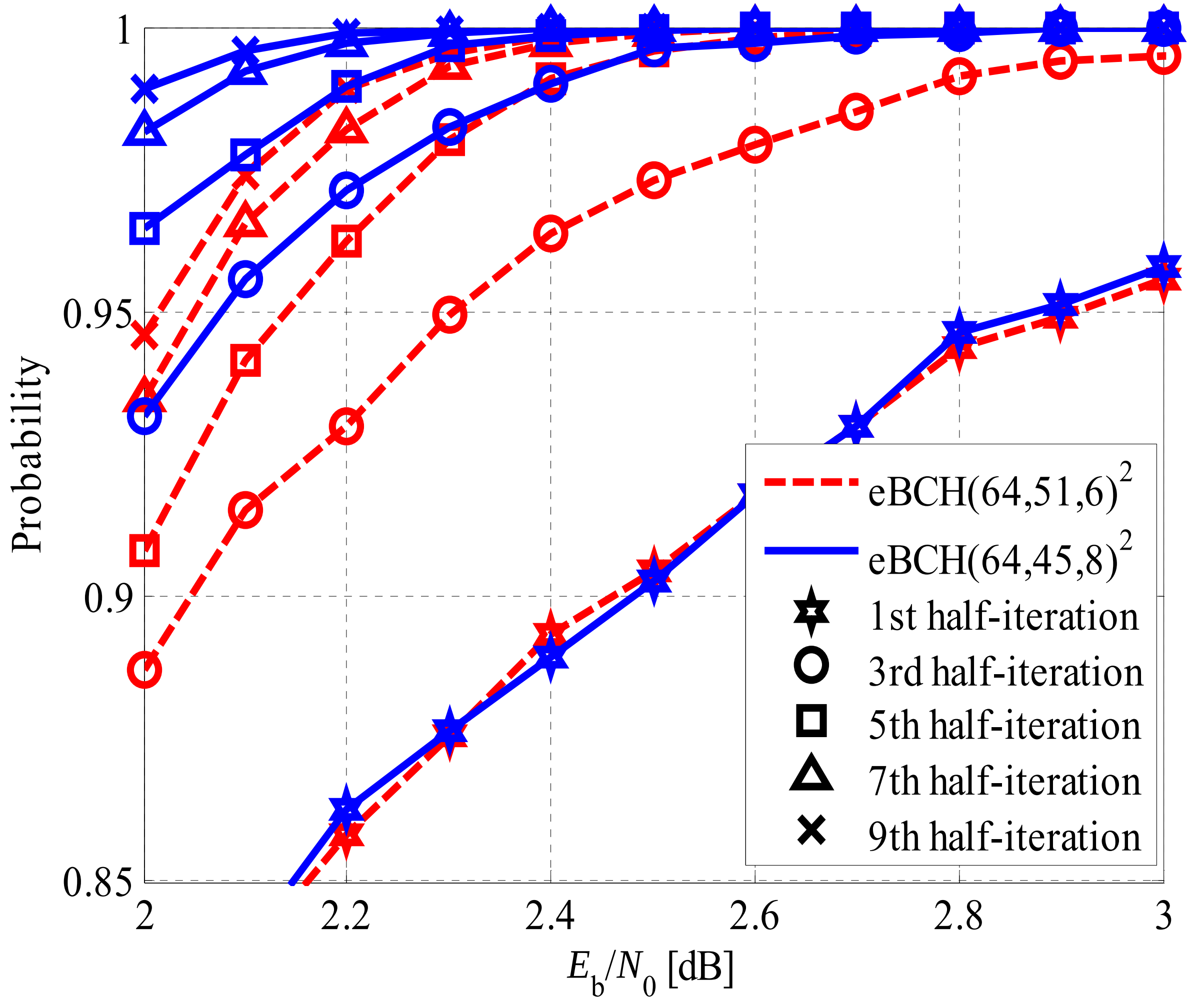
POSTECH CSDL Communications and Signal Design Lab.

- Average portion of row vectors $\mathbf{Y}$ having $d_H(\mathbf{Y},\mathbf{C}_{\mathbf{Y}}) \leq 1$



Proposed Algorithm I: Numerical Results

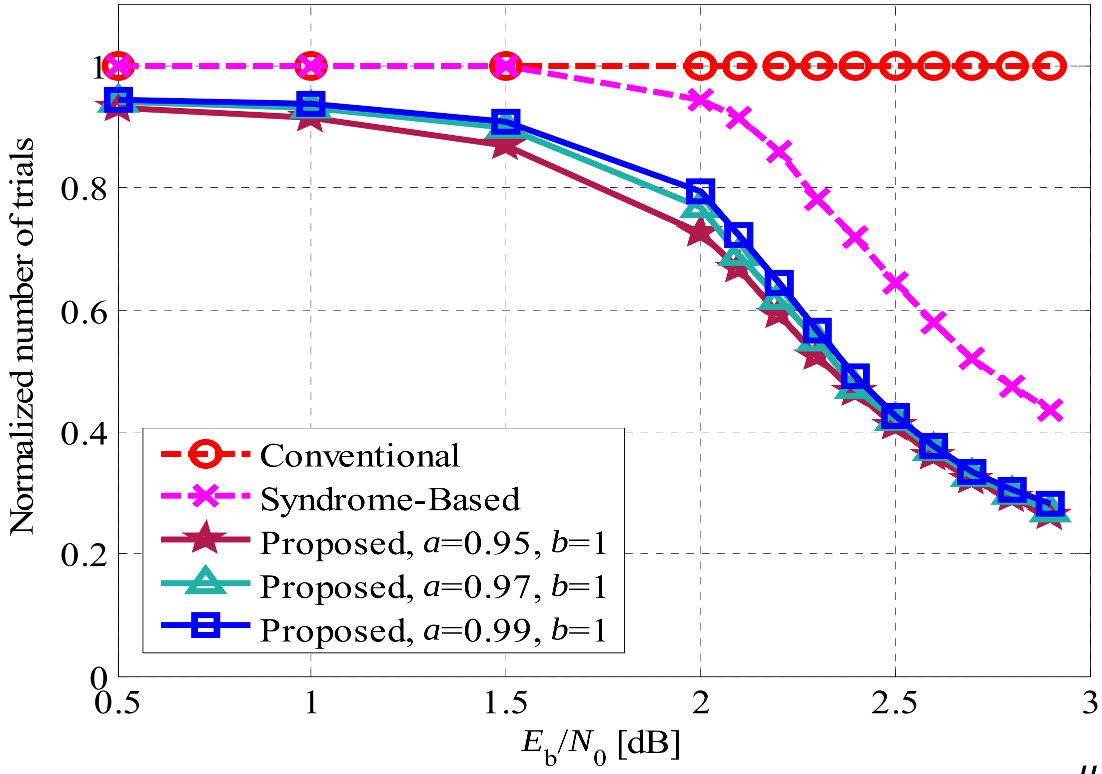
- Probability that $\mathbf{C}_\mathbf{Y}$ is equal to the corresponding transmitted codeword



Proposed Algorithm I: Numerical Results

POSTECH CSDL Communications and Signal Design Lab.

- Computational complexity of an eBCH(64, 51, 6)^2 code



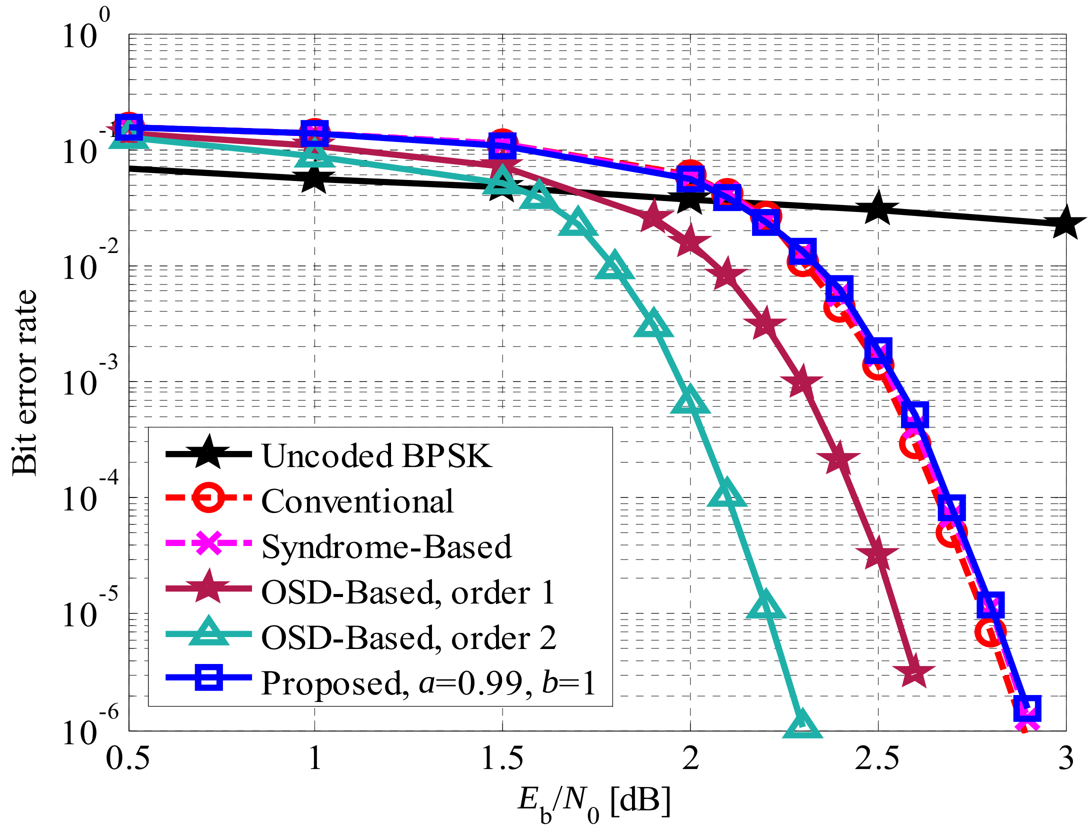
max # iterations: 4

- As the SNR increases, the average number of trials of the employed HDD in the proposed algorithm can be significantly reduced.

Proposed Algorithm I: Numerical Results

POSTECH
CSDL
Communications and Signal Design Lab.

- BER performance of an eBCH(64, 51, 6)^2 code



- The proposed algorithm has only a negligible performance loss, compared with the conventional algorithm.

Proposed Algorithm II

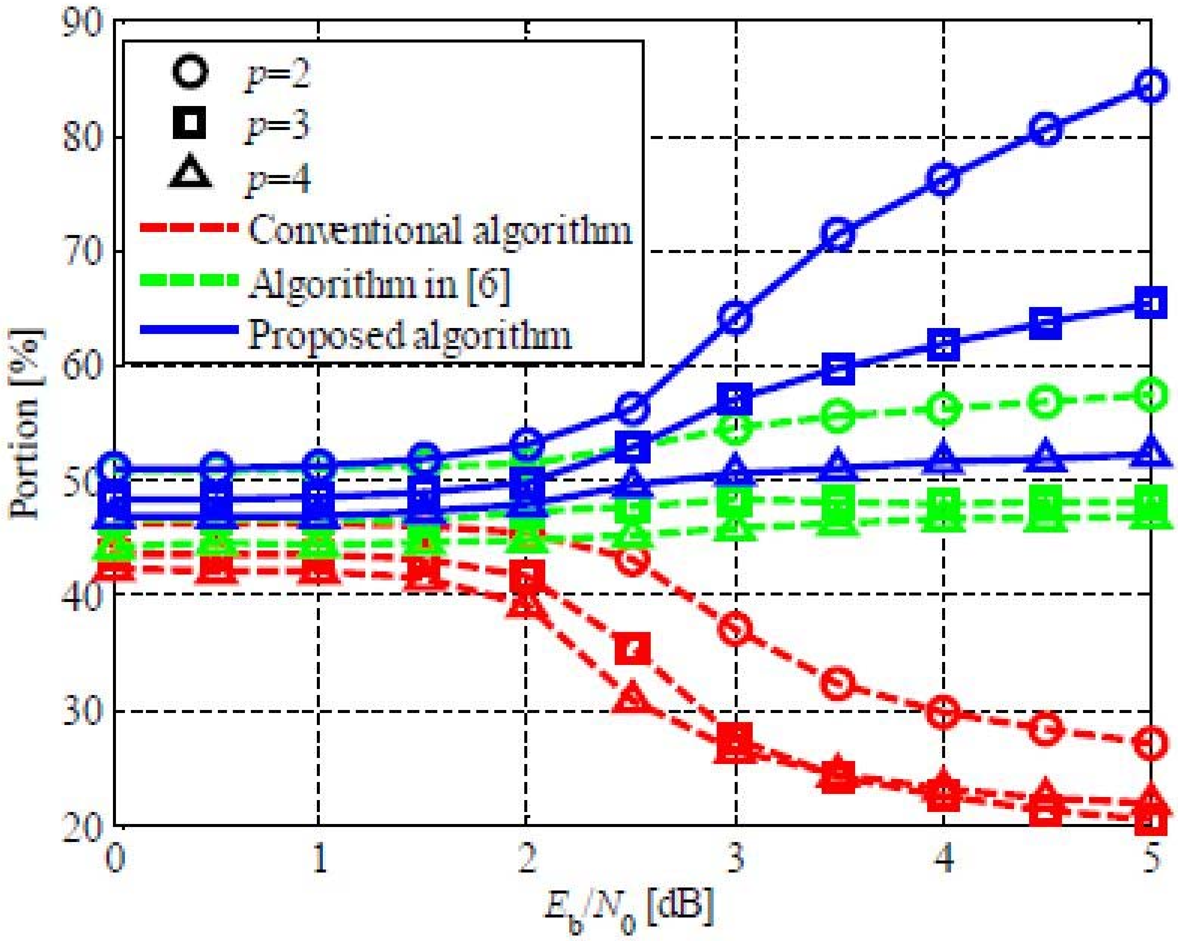
- Proposed algorithm II
  - ✓ imposes two algebraic conditions on the Chase algorithm to avoid a number of unnecessary HDD operations;
  - ✓ simply computes the extrinsic information for the decision codeword.

- Advantages
  - ✓ has much lower computational decoding complexity;
  - ✓ has a little better performance than the conventional decoding algorithm.

[10] J. Son, J. J. Kong, and K. Yang, "Efficient Decoding of Block Turbo Codes," submitted 2017.

Proposed Algorithm II: Numerical Results

- Portion of distinct codewords among the algebraically decoded TSs



- eBCH(64, 51, 6)^2
- 4 iterations

Conclusions

- BTCs under iterative decoding show excellent performance with reasonable complexity.
- We proposed two decoding algorithms for BTCs based on the Chase algorithm.
- They can significantly reduce the decoding complexity with a negligible performance loss or a slightly improved performance, compared with the conventional algorithm for BTCs.
- Low-complexity decoding algorithms for BTCs based on OSD may be further studied.