Comparison of Code Constructions Suitable for High-Throughput Decoding

Sergei Semenov

Lund RC, Sweden

AICT 2022

Security Level:

HUAWEI

Outline:

- Requirements for high throughput codes.
- Choice of candidate code constructions for comparison.
- SC-LDPC codes.
- Staircase codes.
- Comparison code constructions with focus on rate adaptation.
- RS-based Staircase codes.
- Conclusions.

High-T-put Codes: Requirements

- Future Beyond-5G use cases are expected to require wireless speeds in the Terabit/s range.
- Rrequirements for codes:
  - Good enough performance (error correcting capability);
  - Low decoding complexity;
  - Suitable for high level of parallelization;
  - High locality:
    - Allows decoder to use structures that are independent of code length in terms of complexity, storage requirements and latency.

Candidates:

- SC-LDPC:
  - Very good performance (approaching Shannon limit);
  - Decoding complexity: not very low;
  - Allows high level of parallelization (with windowed decoding);
  - Locality: quite high with windowed decoding.

Candidates:

- Generalized Product Codes (GPC):
  - Main representatives:
    - Staircase codes;
    - Braided block codes.
  - Good performance:
    - Depends on decoding algorithm of a component code;
    - With Hard-Decision Decoding (HDD) of a component code inferior to SC-LDPC.
  - Decoding complexity: Could be very low;
  - High level of parallelization (with windowed decoding);
  - Locality: quite high with windowed decoding.

Candidates:

- Polar Codes:
  - Performance:
    - Could be close to SC-LDPCD with list decoding and CRC.
  - Decoding complexity:
    - With high performance, complexity is also close to SC-LDPC;
  - Parallelization:
    - Not easy to achieve;
    - In principle, high level of parallelization is possible but it should be done individually for each particular code.
  - Locality:
    - Same problem as with parallelization: possible in principle but hardly depending on particular code.

Spatial Coupling

- Main idea:
  - codewords $\mathbf{v}_t$ of the block code defined by the parity-check matrix $\mathbf{H}$, instead of being encoded independently, are interconnected (coupled) with their neighbors at times $t-1, t-2, \ldots, t-w$ during the encoding procedure:

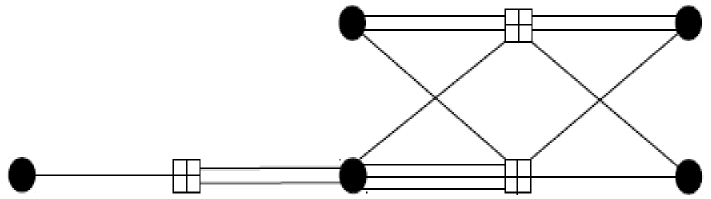
$$\mathbf{v}_t \mathbf{H}_0^T(t) + \mathbf{v}_{t-1}\mathbf{H}_1^T(t) + \cdots + \mathbf{v}_{t-w}\mathbf{H}_w^T(t) = \mathbf{0},$$

$$\mathbf{H}_0(t) + \mathbf{H}_1(t) + \cdots + \mathbf{H}_w(t) = \mathbf{H}, \qquad \forall t.$$

- Both constructions: SC-LDPC and Staircase codes are SC codes.

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - Protograph:

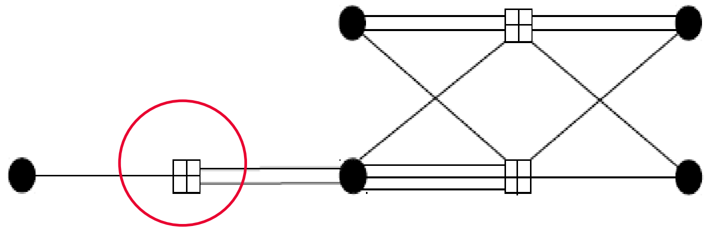


  - Base-graph:

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 2 & 0 \end{bmatrix}$$

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - Protograph:

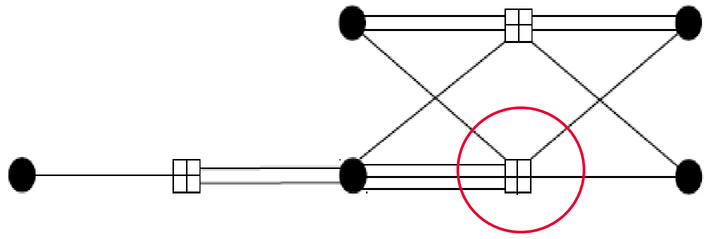


  - Base-graph:

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 2 & 0 \end{bmatrix}$$

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - Protograph:

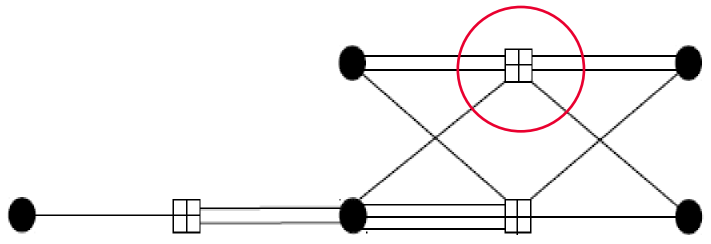


  - Base-graph:

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 2 & 0 \end{bmatrix}$$

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - Protograph:

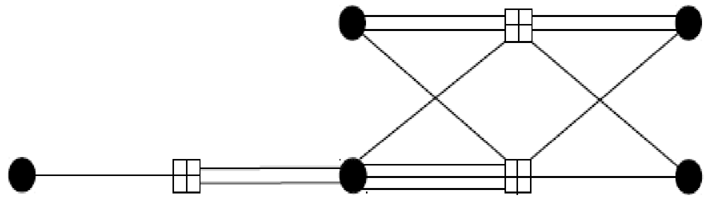


  - Base-graph:

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 2 & 0 \end{bmatrix}$$

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - Protograph:



  - Base-graph:

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 2 & 0 \end{bmatrix}$$

  - Parity-check matrix:
    - Lifting size M = 3.

$$\mathbf{H} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - Split the original BG into a set of BG matrices $\mathbf{B}_i$:

$$\mathbf{B} = \sum_{i=0}^{w} \mathbf{B}_i \,.$$

    - w is the SC memory.
  - BG of the SC-LDPC code:
    - L is the number of transmitted consecutive blocks (initial protographs).

$$\mathbf{B}_L = \begin{bmatrix} \mathbf{B}_0 & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & \\ & & \ddots & & \\ \vdots & \ddots & \mathbf{B}_1 & \mathbf{B}_0 & \\ \mathbf{B}_w & \cdots & & \ddots & \ddots \\ & \ddots & \ddots & \ddots & \mathbf{B}_0 \\ & & \ddots & & \mathbf{B}_1 \\ & & & \ddots & \vdots \\ & & & & \mathbf{B}_w \end{bmatrix}$$

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - w top and w bottom blocks of CNs are of lower degrees
  - Small-degree CNs serve as starting points for the convergence of the iterative decoding process.
  - A “wave” of reliable information propagates towards the middle of the codewords

$$
\mathbf{B}_L = \begin{bmatrix}
\mathbf{B}_0 & & & & \\
\mathbf{B}_1 & \mathbf{B}_0 & & & \\
 & & \ddots & & \\
\vdots & \ddots & \mathbf{B}_1 & \mathbf{B}_0 & \\
\mathbf{B}_w & \cdots & \ddots & \ddots & \\
 & \ddots & \ddots & \ddots & \mathbf{B}_0 \\
 & & \ddots & & \mathbf{B}_1 \\
 & & & \ddots & \vdots \\
 & & & & \mathbf{B}_w
\end{bmatrix}
$$

SC-LDPC codes. Construction.

- Constructing with the help of protographs coupling:
  - w top and w bottom blocks of CNs are of lower degrees
  - Small-degree CNs serve as starting points for the convergence of the iterative decoding process.
  - A “wave” of reliable information propagates towards the middle of the codewords
- As $L \to \infty$, the BP threshold is boosted to that of the optimal Maximum a Posteriori (MAP) decoder.
- Rate $R_L = \frac{Lb_v - (L+w)b_c}{(L+w)b_c}$

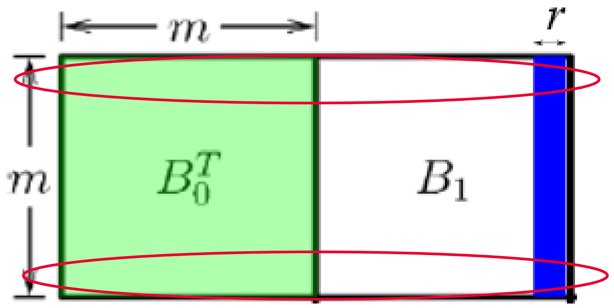
$$\mathbf{B}_L = \begin{bmatrix} \mathbf{B}_0 & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & \\ \vdots & \ddots & \ddots & \\ & & \mathbf{B}_1 \quad \mathbf{B}_0 & \\ \mathbf{B}_w & \vdots & \ddots \quad \ddots & \\ & \ddots & \ddots \quad \ddots & \mathbf{B}_0 \\ & & \ddots & \mathbf{B}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{B}_w \end{bmatrix}$$

Staircase codes. Construction.

- Construction combines ideas from recursive convolutional coding and block coding.
- Code is defined by a sequence $B_0, B_1, B_2, \ldots$ of $(m \times m)$ matrices B_i:
  - Block B_0 is initialized to a known at receiver $(m \times m)$ matrix, e.g., of zero symbols.
  - Choose linear block code in systematic form to serve as a component code.
  - Component code:
    - Length $2m$;
    - r parity symbols, $r < m$;
    - $2m - r$ information symbols.

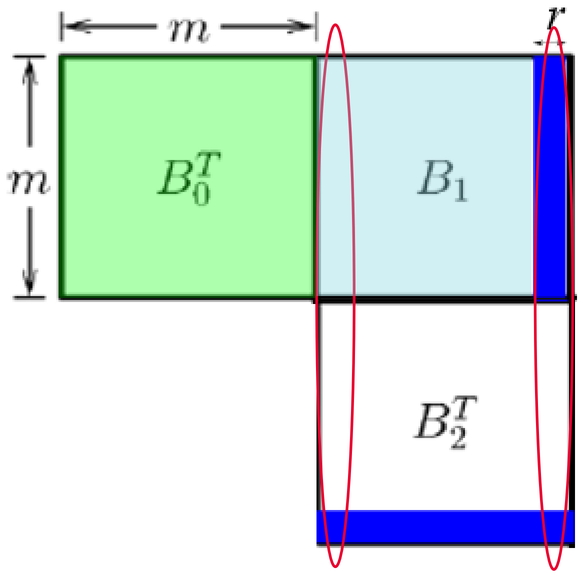
Staircase codes. Construction.

- $2m - r$ information symbols
  - m symbols are fixed already,
  - Choose $m - r$ free inf. symbols
  - Generate r parity symbols for each word of a component code.
- $(m - r)m$ new information symbols per block.
- rm parity symbols per block.



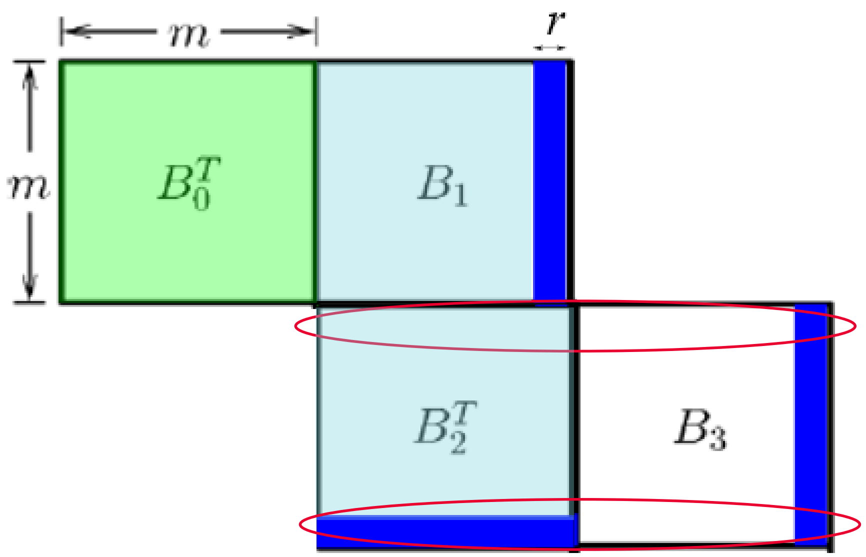
Staircase codes. Construction.

- $2m - r$ information symbols
  - m symbols are fixed already,
  - Choose $m - r$ free inf. symbols
  - Generate r parity symbols for each word of a component code.
- $(m - r)m$ new information symbols per block.
- rm parity symbols per block.



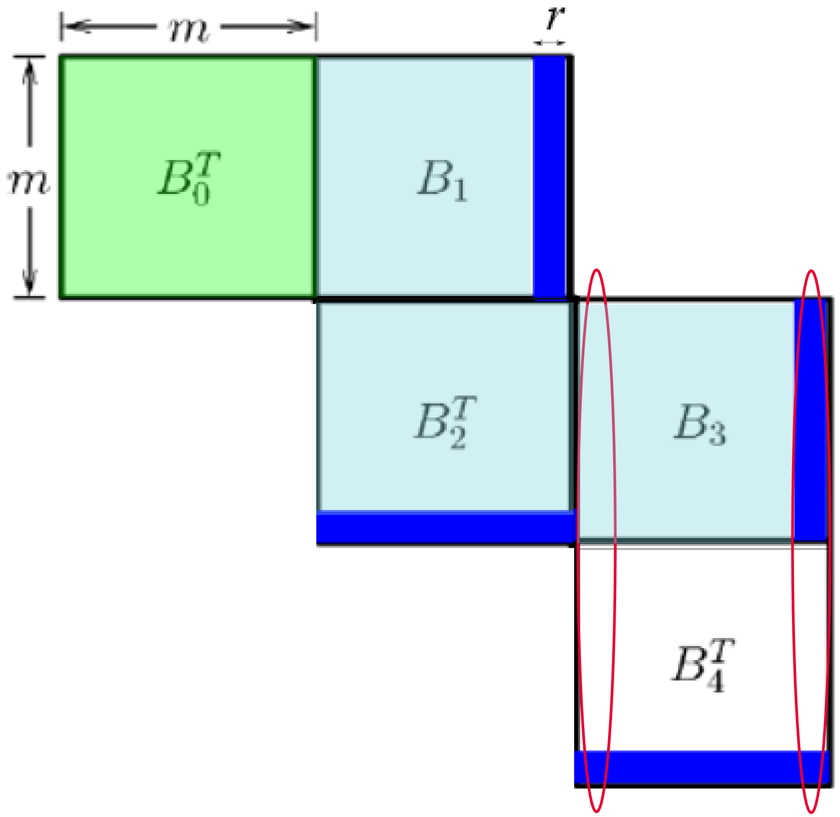
Staircase codes. Construction.

- $2m - r$ information symbols
  - m symbols are fixed already,
  - Choose $m - r$ free inf. symbols
  - Generate r parity symbols for each word of a component code.
- $(m - r)m$ new information symbols per block.
- rm parity symbols per block.



Staircase codes. Construction.

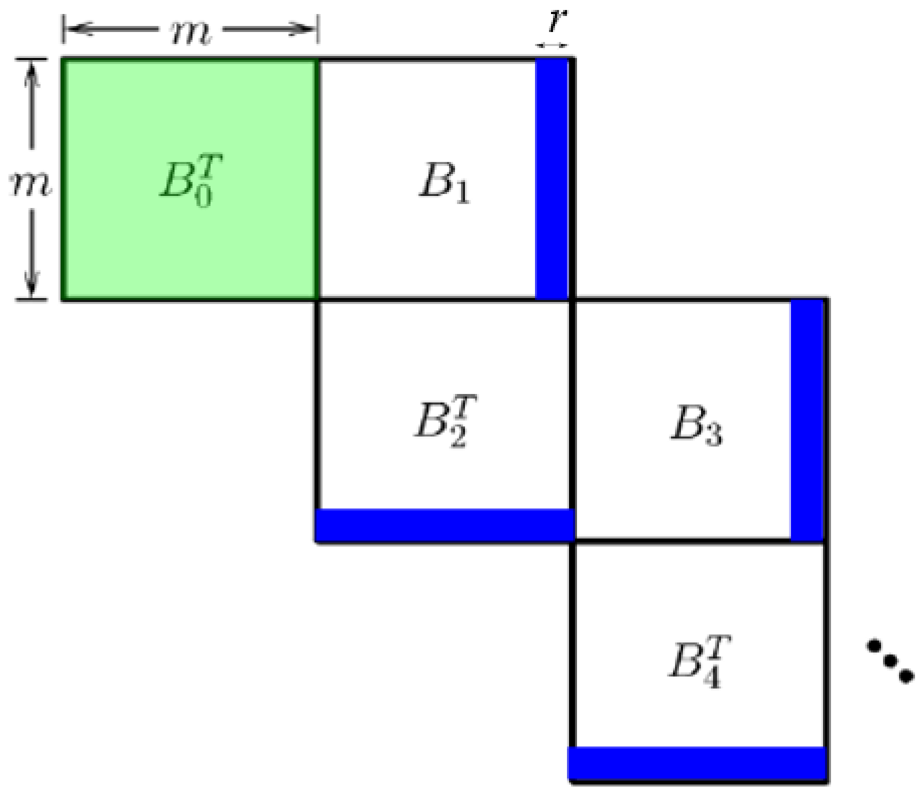
- $2m - r$ information symbols
  - m symbols are fixed already,
  - Choose $m - r$ free inf. symbols
  - Generate r parity symbols for each word of a component code.
- $(m - r)m$ new information symbols per block.
- rm parity symbols per block.
- Code rate $R = 1 - \frac{r}{m}$, or $R = \frac{m-r}{m+\Lambda\frac{r}{l}}$



Staircase codes. Construction.

- Staircase codes are also SC codes
  - Parity-check matrix:

$$\mathbf{H}_{St} = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_1 & \mathbf{H}_0 & 0 & \dots & \dots & 0 & \dots \\ 0 & 0 & \mathbf{H}_2 & \mathbf{H}_1 & \mathbf{H}_0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & \mathbf{H}_2 & \mathbf{H}_1 & \mathbf{H}_0 & \dots \\ \vdots & \vdots & \vdots & \vdots & 0 & 0 & \mathbf{H}_2 & \ddots \end{bmatrix}$$

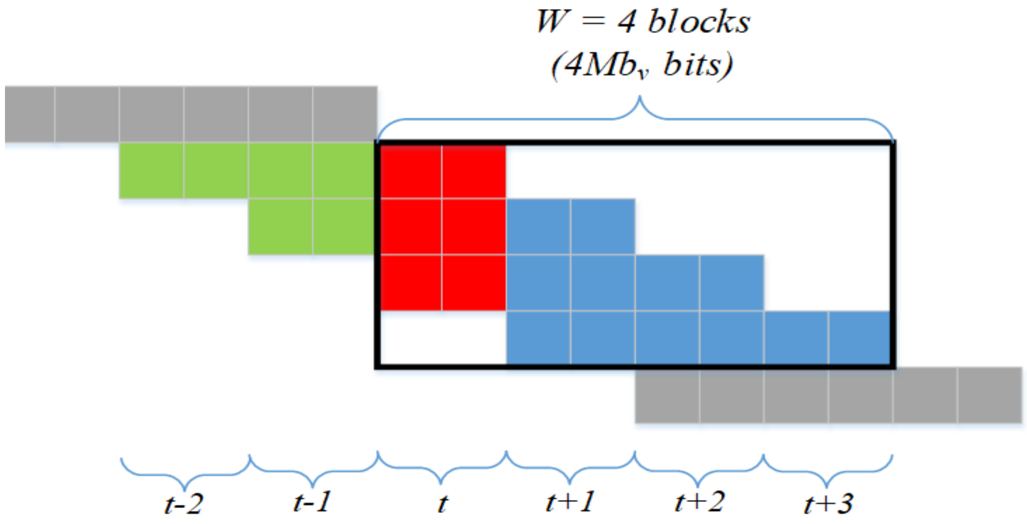


- Similar to SC-LDPC structure.
- Staircase codes are well suited for HDD (very low complexity).
- SDD is also possible to use.

Windowed Decoding.

- SC-LDPC codes:
  - Convolutional structure of BG matrix =>
    - two VN blocks with indices i and j, such that $j \geq i + w + 1$, do not share any parity-check equation
    - VNs from these blocks cannot be connected to the same CN.
  - Window decoder deals with W received blocks such that $W \geq w + 1$.

- Staircase codes:
  - Similar procedure, $W \geq 3$.



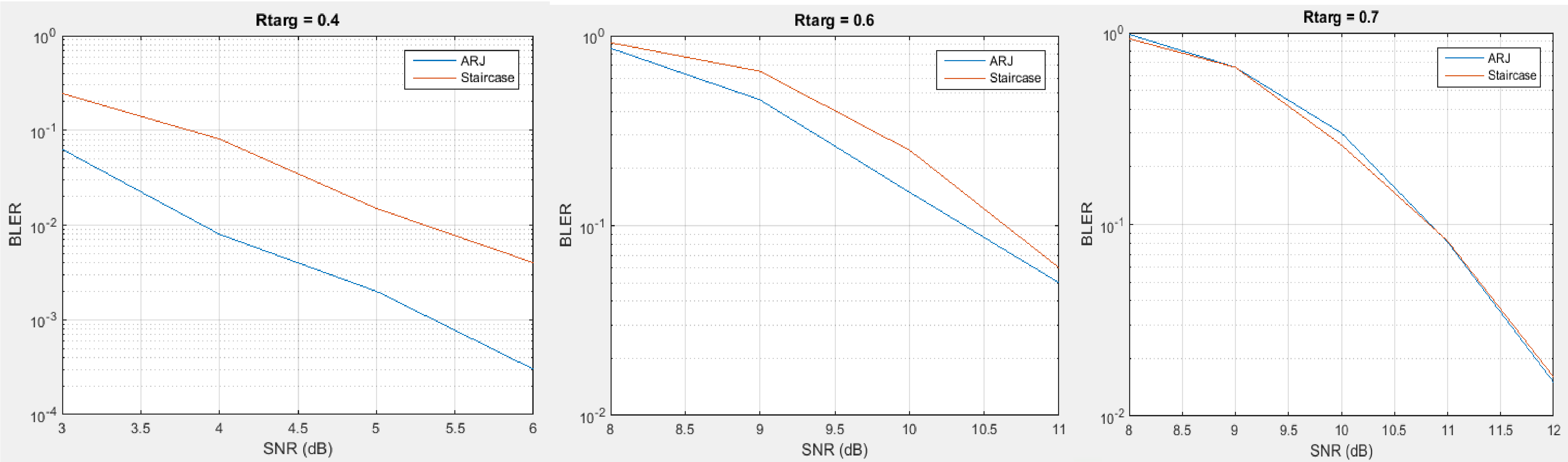
Comparison of SC-LDPC and Staircase codes.

- One more very important criterion:
  - Smooth adaptation of the code to the different code rates.
  - Quite often this criterion is overlooked in comparison of different code constructions.
- Obvious solution of rate adaptation:
  - Construct a mother code of low code rate;
  - Puncture parity-check bits to obtain codes of higher rates.

SDD of both SC-LDPC and Staircase codes.

- SC-LDPC Accumulate Repeat-Jagged (ARJ)-based mother code:
  - Memory $w = 2$, lifting size $M = 8$, Code length $N_{SC-LDPC} = 2200$.
  - Mother code rate $R_{SC-LDPC_{init}} = 0.38$.
- Staircase code:
  - Component code: $(32, 21)$ BCH code;
  - Code length $N_{St} = 2048$;
  - Mother code rate $R_{St_{init}} = 0.31$.

SDD of both SC-LDPC and Staircase codes.



- SC-LDPC code outperforms staircase code at $R_{targ} = 0.4$ by more than 1 dB.
- Decoding complexity of staircase code is 3 – 4 times higher than that of SC-LDPC.
- However, performance of SC-LDPC code deteriorates faster with increasing R_{targ}.

HDD for Staircase codes.

- Staircase codes are especially attractive for HDD:
  - HDD is applied to component code.
  - Main drawback: performance loss to SDD.
  - On the other hand, SDD complexity is often prohibitive to use powerful codes as a component code.
  - Usage of powerful codes with HDD can be an option.
- Benefit of staircase code construction:
  - Performance decreases quite smoothly with code rate increase (obtained by puncturing the parity bits of mother code).

HDD for Staircase codes.

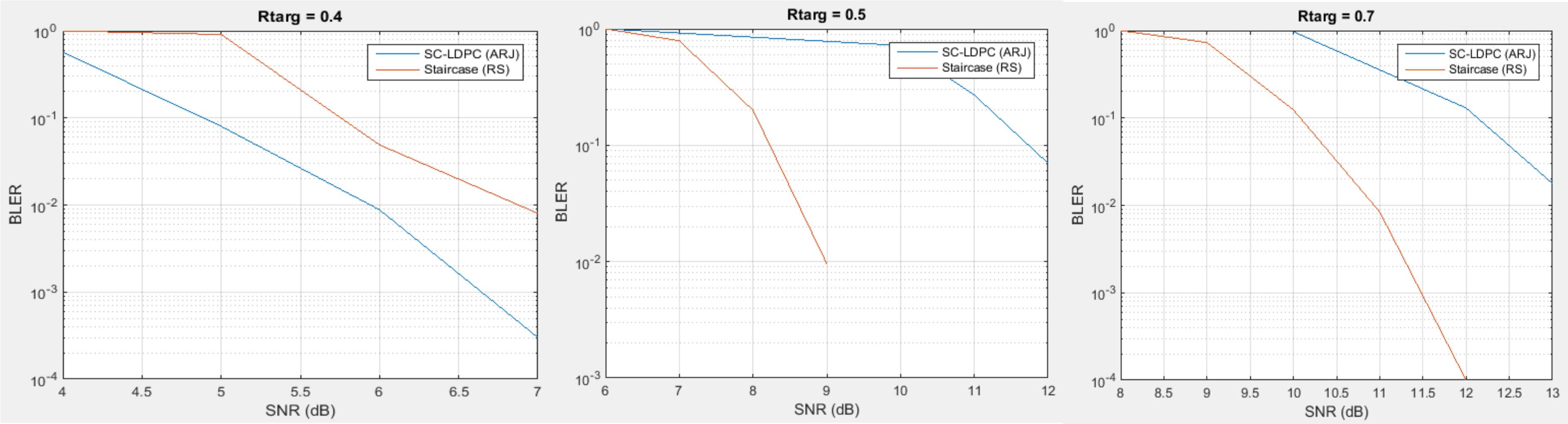
- Consider RS codes as component codes:
  - RS codes are known for good performance.
  - RS codes are Maximum Distance Separable (MDS) codes:
    - Any k (k is number of information symbols) symbols of codeword forms information sequence.
    - All puncturing patterns are equally good.

HDD for Staircase codes.

- Staircase code:
  - Extended $(32, 23)$ RS code over the $GF(2^5)$ capable of correcting 4 errors was used as a component code.
  - Component code length in bits is $32 \cdot 5 = 160$ $(m = 80\ bits,\ r = 45\ bits)$.
  - Code length $N_{St} = 22800$; Mother code rate $R_{St_{init}} = 0.368$.

- SC-LDPC code:
  - Code length $N_{SC-LDPC} = 20400$; Mother code rate $R_{SC-LDPC_{init}} = 0.398$.
  - Decreased decoder complexity: MS rather than SPA, low-resolution message-passing (3 bits).

HDD for Staircase codes.



- SC-LDPC still outperforms Staircase code at close to initial mothercode code rate.
- With even some code rate increase, Staircase code starts to outperform SC-LDPC.
- Decoding complexity of Staircase code is 7 – 8 times less than the complexity of SC-LDPC decoder.

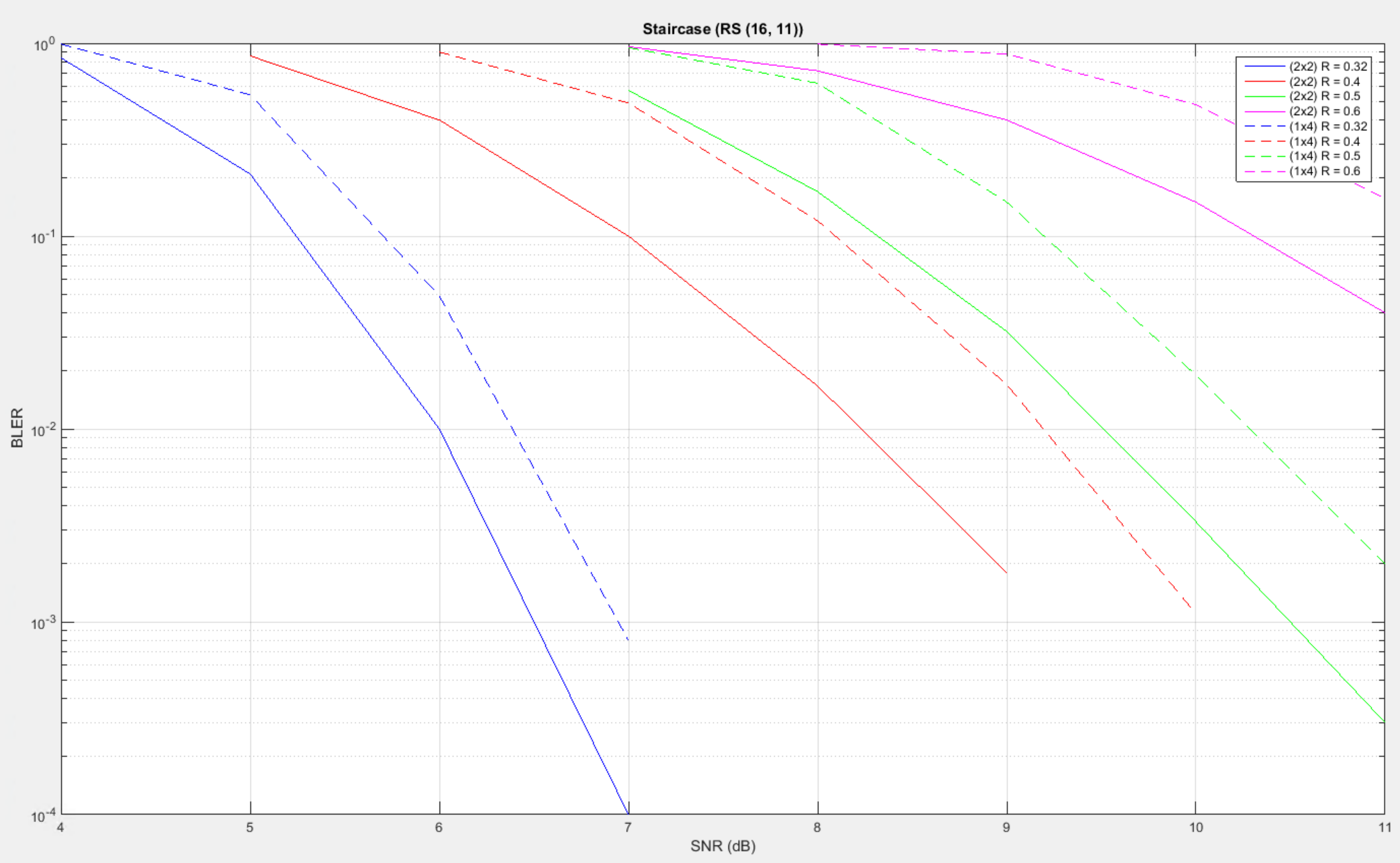
RS-based Staircase codes with HDD.

- Pros:
  - Very low decoding complexity;
  - High level of parallelization;
  - High locality;
  - Smooth performance degradation with code rate increase.
- Cons:
  - Performance is inferior to SC-LDPC at low code rate;
  - Low flexibility with the choice of the code length:
    - should be a multiple of m^2.

Optimization of code length choice.

- Galois field $GF(2^p)$:
  - If p is not a prime:
    - Different representation of a field element:
      - $(1 \times p)$ bits: $m = p2^{p-1}$
        - $p2^{p-1}$ RS codes should be decoded for each block.
      - $(q_1 \times q_2)$ bits: $m = q_2 2^{p-1}$
        - $\frac{q_2 2^{p-1}}{q_1}$ RS codes should be decoded for each block.
- Example $GF(2^4)$:
  - $L = 4, \Lambda = 1$.
    - (1×4): $m = 32$, (32 RS codes per block), $N_{St} = 4736$;
    - (2×2): $m = 16$, (8 RS codes per block), $N_{St} = 1184$.

Optimization of code length choice.



Conclusions.

- SDD for both SC-LDPC and Staircase codes:
  - SC-LDPC codes provide better performance and lower complexity than the staircase codes.
  - However, performance of SC-LDPC codes deteriorates very fast with code rate increase.
- RS-based Staircase codes under HDD:
  - Significant complexity decrease SC-LDPC;
  - Affordable performance loss;
  - At high code rates outperform SC-LDPC with MS.
  - Binary message passing => significant decrease in amount of data exchanged. Important for reaching high throughput.

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI