New Perspectives on Convolutional Coding/Decoding for 6G

Joseph Jabour, Jeremy Nadal, Stefan Weithoffer, Charbel Abdel Nour, Catherine Douillard

New Perspectives on Convolutional Coding/Decoding for 6G

Joseph Jabour, Jeremy Nadal, Stefan Weithoffer, Charbel Abdel Nour, Catherine Douillard
IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France

***Abstract*—In recent years, significant efforts were invested in (de-)coding of convolutional codes and their concatenations by leveraging alternative representations which aim to lower the decoding complexity. One example is the Local-SOVA algorithm that avoids redundant computations by reformulating the BCJR algorithm towards the notion of providing partial extrinsic information. This latter is now provided along path metric computations in the code trellis leading to a significant reduction in complexity. For high coding rates where most redundant computations are made, the trellis compression technique represents an appealing solution to reduce the complexity of warm-up calculations for BCJR-based turbo decoding. In addition, BP based decoding solutions have come a long way to bridge the gap between initially disappointing results and BCJR performance. In this work, we propose to extend and combine several new representations of the code targeting reduced decoding complexity for short convolutional codes and their concatenations. Based on our new perspective, we give an outlook on convolutional based FEC for 6G.**

***Index Terms*—6G, Convolutional codes, Turbo codes, Spatial coupling**

I. Introduction

Convolutional codes (CCs) are a well-established class of error-correcting codes, known for their flexible code rates and efficient, low-complexity encoding. For instance, CCs are fundamental in 4G control channels [1] and in scenarios with limited receiver hardware resources. Additionally, CCs serve as key components in turbo codes (TCs), and their decoder pipelining capability makes them well suited to high-speed continuous transmissions.

However for finite block lengths and limited number of memory elements, the decoding performance of CCs falls short of the near-capacity performance achieved by advanced coding schemes such as low-density parity-check (LDPC) codes and TCs [2]. Although these advanced codes deliver superior performance, they are far more resource-intensive, with high-throughput pipelined implementations limited to short block lengths and few decoding iterations [3]. Specifically, LDPC codes suffer from a performance degradation when decoding short block lengths [4] with belief propagation algorithms due to unavoidable short cycles in the Tanner graph.

High-throughput communication systems typically operate in a high-coding-rate regime. There, efficient decoding of high-rate CCs remains challenging, as the complexity of the maximum a posteriori (MAP) decoder depends on the structure of the base code rather than that of the punctured code. Simplifying MAP decoding is thus critical for CCs, but also for TCs, which structurally rely on two MAP decoders or more.

This paper introduces new perspectives on two aspects of convolutional coding, detailed in dedicated sections: Section II presents the novel spatial coupling scheme tailored to pipelined MAP decoder architectures, while Section III explores an alternative representation of punctured CCs. Finally, Section IV concludes the paper.

II. A Hardware Implementation Perspective on Spatially coupled CCs

Very high-throughput decoding can be achieved by pipelining every decoding iteration [3], [5]. In such architecture, the extrinsic information of all frames traversing the pipeline are naturally accessible, but the exchange is only performed from one decoder to the next one in the pipeline chain. If this constraint is broken and the exchange between other decoders in the pipeline chain is permitted, then frames can be *coupled* and the encoder becomes a spatially coupled turbo encoder [6]–[8]. Consequently, this structure benefits from spatial coupling (SC) gain while limiting the complexity overhead with respect to the original non-coupled decoder structure. Contrary to the focus of most prior-art on serially concatenated SC CCs, in this section this concept is exploited to build a new pipelined decoder structure for parallel concatenated CCs.

A. Spatial Coupling Procedure

SC is a technique that interconnects multiple codewords to enhance error correction performance. Let the systematic bits of frame n be represented by the $1 \times K$ vector $\boldsymbol{x}_n$, directly encoded to produce a first set of parity bits. A second encoder processes an input vector $\boldsymbol{z}_n$ to generate a second set of parity bits. To leverage SC, the input $\boldsymbol{z}_n$ must include systematic bits drawn from N_{SC} different frames, where N_{SC} represents the number of coupled frames. The first and second encoders are referred to as the uncoupled and coupled encoders, respectively. A two-step process is

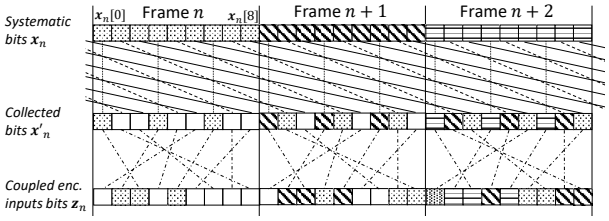

Fig. 1. An example of the spatial coupling procedure for $K = 9$ bits and $N_{\text{SC}} = 3$ coupled frames. Systematic bits are collected into $\boldsymbol{x}'_n$, and subsequently mapped to the coupled encoder inputs $\boldsymbol{z}_n$.

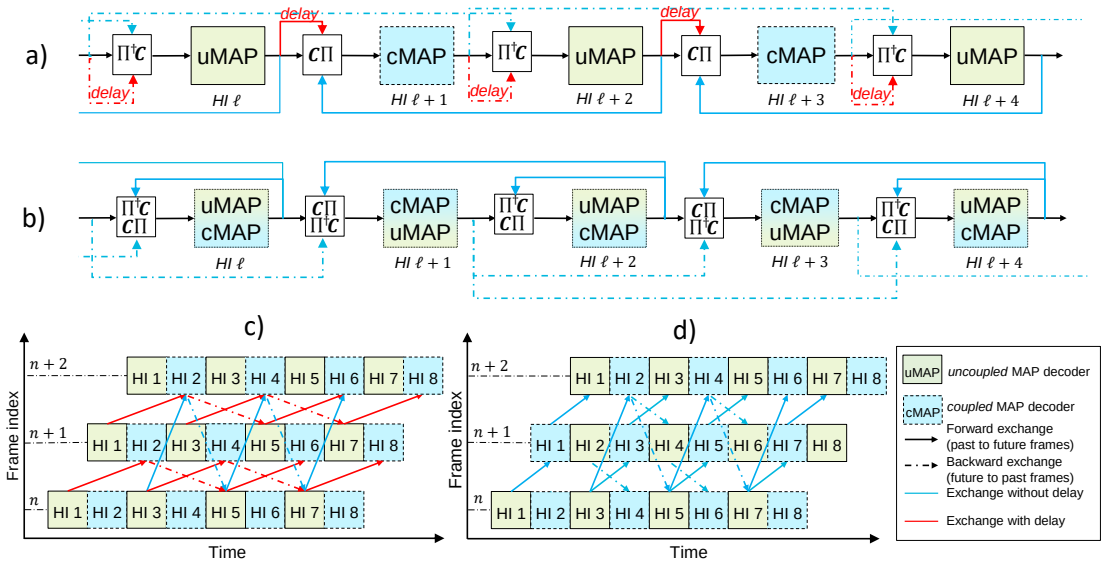

Fig. 2. a) Proposed baseline structure for pipelined SC MAP decoders, b) Improved structure without delay c) Baseline schedule of extrinsic exchanges d) improved schedule without delay.

proposed to couple systematic bits in a predefined order. First, systematic bits from multiple frames are collected into a $1 \times K$ vector $\boldsymbol{x}'_n$, where $\boldsymbol{x}'_n[k] = \boldsymbol{x}_{n-\boldsymbol{c}[k]}[k]$. Here, $\boldsymbol{c}$ is a $1 \times K$ vector[1] that maps each systematic bit $\boldsymbol{x}_n[k]$ transmitted in frame n to one of the encoder inputs associated with frame $n - \boldsymbol{c}[k]$. The collected bits $\boldsymbol{x}'_n$ are interleaved using a permutation matrix $\boldsymbol{\Pi}$, to form $\boldsymbol{z}_n = \boldsymbol{x}'_n\boldsymbol{\Pi}$. The interleaver design is beyond the scope of this work, and well-established families like quadratic permutation polynomial (QPP) [9] and almost regular permutation (ARP) [10], are considered here. To map coupled encoder inputs $\boldsymbol{z}_n$ back to systematic bits $\boldsymbol{x}_n$, the reverse process involves deinterleaving $\boldsymbol{x}'_n = \boldsymbol{z}_n\boldsymbol{\Pi}^\dagger$, where $\dagger$ denotes matrix transpose, followed by collecting $\boldsymbol{x}_n[k] = \boldsymbol{x}'_{n+\boldsymbol{c}[k]}[k]$.

Figure 1 illustrates the two-step SC process with a block of length $K = 9$ bits, $N_{\text{SC}} = 3$ coupled frames, and $\boldsymbol{c}[k] = \text{mod}_3(k)$. In frame n, $K/N_{\text{SC}} = 3$ systematic bits $(\boldsymbol{x}_n[0], \boldsymbol{x}_n[3], \boldsymbol{x}_n[6])$ are collected into $\boldsymbol{x}'_n$ as inputs for the coupled encoder, which outputs parity bits for frame n (see dotted blocks). The bits $(\boldsymbol{x}_n[1], \boldsymbol{x}_n[4], \boldsymbol{x}_n[7])$ are collected in $\boldsymbol{x}'_{n+1}$ for encoding in the next frame, $n+1$, while the remaining bits are collected in $\boldsymbol{x}'_{n+2}$. Once all bits from previous frames have been collected, $\boldsymbol{x}'_n$ is interleaved to produce $\boldsymbol{z}_n$, the input to the coupled encoder.

To simplify implementation, this paper adopts a regular bit collection scheme defined as $\boldsymbol{c}[k] = \text{mod}_{N_{\text{SC}}}(k)$. In this scheme, all systematic bits spaced by N_{SC} positions in $\boldsymbol{x}_n$ are mapped to the same frame at the coupled encoder input. The collection operation is expressed in matrix form as:

$$\boldsymbol{c}'_n = [\boldsymbol{x}_{n-m}]_{m\in[0,N_{\text{SC}}-1]} \times \boldsymbol{C}, \tag{1}$$

[1]The elements of $\boldsymbol{c}[k]$ are restricted to N_{SC} distinct values.

where $\boldsymbol{C}$ is the regular collection matrix, defined by $\boldsymbol{C}\big[i + \text{mod}_{N_{\text{SC}}}(i)K, i\big] = 1\ \forall i \in [0, K-1]$, and 0 for all other non-assigned row or column indices.

B. Proposed Pipelined Decoder Structure

Decoding involves iteratively exchanging and updating extrinsic log-likehood ratios (LLRs) computed by MAP decoders across multiple frames (denoted by the function f_{MAP}[2]). Practical decoders typically implement the max-log-MAP (MLM) or Local-Soft Output Viterbi Algorithms (L-SOVAs) [11]. Let $(\boldsymbol{\gamma}_{x_n}, \boldsymbol{\gamma}_{z_n})$ represent $1 \times K$ vectors corresponding to the decoder's extrinsic LLRs for the uncoupled and coupled bits of frame n, respectively, such that

$$\boldsymbol{\gamma}_{x_n} \leftarrow f_{\text{MAP}}\Big([\boldsymbol{\gamma}_{z_{n+m}} \times \boldsymbol{\Pi}^\dagger]_{m\in[0,N_{\text{SC}}-1]} \times \boldsymbol{C}\Big), \tag{2}$$

$$\boldsymbol{\gamma}_{z_n} \leftarrow f_{\text{MAP}}\Big([\boldsymbol{\gamma}_{x_{n-m}}]_{m\in[0,N_{\text{SC}}-1]} \times \boldsymbol{C}\boldsymbol{\Pi}\Big). \tag{3}$$

The arrow "$\leftarrow$" symbol denotes the update operation. During decoding, the extrinsic LLRs are updated by each MAP decoder and iteratively passed to other decoders according to the above equations. For a frame n, one extrinsic information update constitutes one half-iteration (HI), while a full iteration occurs when the extrinsic LLRs of the coupled and uncoupled MAP decoders are exchanged, corresponding to two HIs. Processing N_{HI} HIs for each frame requires the same number of MAP decoders in the pipeline chain, as shown in Figure 2a. Half of these are coupled MAP (cMAP) decoders, taking uncoupled extrinsic information $\boldsymbol{\gamma}_{x_n}$ as input and producing coupled extrinsic information $\boldsymbol{\gamma}_{z_n}$ as output. The other half are uncoupled MAP (uMAP) decoders, taking $\boldsymbol{\gamma}_{z_n}$ as input and producing $\boldsymbol{\gamma}_{x_n}$ as output. At the output of the decoder,

[2]f_{MAP} implicitly incorporates the received channel LLRs of both systematic and parity bits as inputs. These LLRs remain constant during decoding and are therefore omitted for clarity in the equations and figures.

the extrinsic LLRs for frame n are passed to the decoders processing later frames $n' > n$ (forward exchanges) or earlier frames $n' < n$ (backward exchanges).

C. Improved Extrinsic LLR Exchange Schedule

Collecting and interleaving extrinsic LLRs through MAP decoders may require additional memory elements, depending on the MAP architecture and the SC mapping $\boldsymbol{c}$. In particular, the structure in Figure 2a inherently introduces a delay for certain extrinsic LLRs. This can be observed by analyzing the schedule diagram in Figure 2c, which shows how extrinsic LLRs are exchanged between uMAP and cMAP decoders.

When a new frame n is received, it is first processed by the uMAP decoder at HI 1. The extrinsic LLRs generated during this stage can only be forwarded to the cMAP decoder handling frame $n+1$ at HI 2 when the latter is ready to consume them. Meanwhile, these extrinsic LLRs must be temporarily stored while the first uMAP decoder processes the first HI of frame $n+1$. This pattern is repeated for the following HI, which eventually requires delaying half of the extrinsic LLRs, identified by all red lines in Figure 2.

To address this issue, an alternative structure is proposed that switches between the uMAP and cMAP decoders depending on whether the frame index n is even or odd. For example, as shown in Figure 2d, frame n is processed by a uMAP decoder at HI 1. Upon receiving frame $n+1$, the decoder switches to cMAP, allowing immediate consumption of the extrinsic LLRs generated previously. This approach can be generalized to all HIs, where MAP decoders are configured as either uMAP or cMAP within the same processing time intervals. Figure 2b illustrates the structure that supports this improved extrinsic LLR exchange schedule, with significantly reduced memory requirements compared to the structure from Figure 2a. It should be noted, that this structure can be easily obtained from a classical fully pipelined decoder architecture by adding multiplexing at the input of the half iteration decoder stages with minimal overhead.

D. Performance Evaluation

The frame error rate (FER) performance of the proposed SC scheme is evaluated over an additive white Gaussian noise (AWGN) channel using binary phase-shift keying (BPSK) modulation. A TC is considered using 8-state component codes with polynomials $(1, 13/15)$ in octal notation. No puncturing is applied ($R = 1/3$), and tail-biting termination is used. The MAP decoders use the MLM algorithm with an extrinsic scaling factor of 0.75. The evaluation considers several values for the number of pipelined decoders, N_{HI}. When SC is implemented, the coupling length is fixed at $N_{\text{SC}} = 3$, based on prior evaluations to balance complexity and performance. The analysis assumes continuous sequential transmission of an infinite number of frames, each of size K. Thus, the pipeline remains filled, and edge effects are neglected.

To evaluate the performance of the proposed SC scheme, coupled and uncoupled configurations are first compared with $N_{\text{HI}} = 16$. Figure 3 presents the FER performance for both

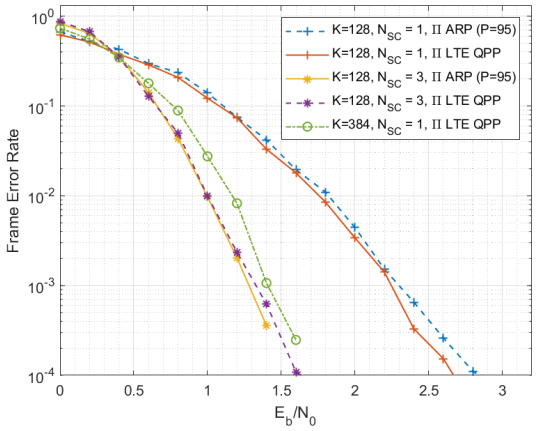

Fig. 3. FER evaluation without and with spatial coupling ($N_{\text{SC}} = 3$) for different decoder configurations and $N_{\text{HI}} = 16$.

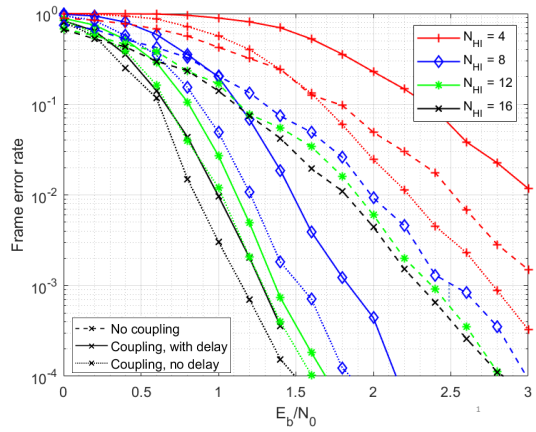

Fig. 4. FER performance for different number of half iterations N_{HI} when considering no SC (dashed lines), SC using base schedule with delay (solid lines) and SC using improved schedule without delay (dotted lines).

cases under different coding schemes, including the QPP interleaver of 4G/LTE [1] for frame sizes $K = 128$ and $K = 384$, as well as an ARP interleaver designed for full iteration overlap [12] at $K = 128$. The ARP interleaver has a period $P = 95$ and the shift vector[3].

Figure 3 shows that SC achieves approximately a 1 dB performance gain at FER $= 10^{-3}$ for both interleavers. The performance for $K = 128$ with SC is comparable to the uncoupled case with a tripled frame size ($K = 384$) using the LTE QPP interleaver. However, with $N_{\text{SC}} = 3$, SC utilizes MAP decoders configured for $K = 128$, reducing memory requirements by up to $2/3$ compared to uncoupled decoders processing frames with sizes $N_{\text{SC}} - 1 = 2$ times larger. In fully pipelined decoder architectures, the number of MAP decoder instances is equal to the number of HIs, leading to excessive hardware resource usage for high values of N_{HI}. To ensure practical hardware implementation, N_{HI} value must be limited. Figure 4 shows the FER performance for different

[3] $\boldsymbol{S} = [15, 79, 15, 79, 15, 79, 15, 79, 15, 79, 15, 79, 8, 79, 15, 78, 16, 79, 15, 86, 15, 79, 15, 79, 15, 79, 15, 79, 15, 79]$.

N_{HI} values for coupled and uncoupled configurations. Dashed lines correspond to uncoupled coding, while solid lines are obtained with SC using the baseline exchange schedule with delay (Figure 2b). With $N_{\text{HI}} = 4$, delayed extrinsic LLR exchanges cause the coupled decoder to perform worse than the uncoupled one. However, at $N_{\text{HI}} = 8$, spatial coupling achieves a gain of 0.65 dB at FER $= 10^{-3}$, surpassing the uncoupled decoders for $N_{\text{HI}} \leq 16$ by 0.45 dB. Increasing N_{HI} to 12 further amplifies the performance gap, yielding an additional 0.5 dB gain. Dotted lines in Figure 4 correspond to the proposed improved exchange schedule without delay. Here, significant performance gains are observed, with the SC decoder outperforming the uncoupled decoder even at $N_{\text{HI}} = 4$. At $N_{\text{HI}} = 8$, the improved schedule outperforms the baseline by 0.25 dB, resulting in a total gain of 0.75 dB compared to the uncoupled case. These improvements are achieved without additional memory for extrinsic LLR exchanges.

III. New perspective on punctured CCs

Punctured convolutional codes [13] achieve higher coding rates by selectively omitting parity bits during transmission. However, a major drawback of this approach is that the decoding complexity is still dominated by the underlying unpunctured base code. In this section, an alternative representation is proposed for high-rate convolutional codes with a rate $R = m/(m+1)$, designed to reduce the number of decoding stages and, consequently, the latency of the decoder.

A. Alternative Representation of High-Rate CCs

Consider an $R = 1/2$ CC encoding a sequence of K bits. Let $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{s}$ denote $1 \times K$ vectors of systematic bits, parity bits (prior to puncturing), and encoder state bits, respectively. Its generator matrix, $\boldsymbol{T_g}$, is a $K \times K$ Toeplitz matrix over GF(2), generated by a polynomial g. If the encoder's initial state is zero, $\boldsymbol{T_g}$ is upper triangular. Its first row is the binary representation of g using K bits, with the least significant bit in the first column (see Section III-B). The remaining rows, $i \in [\![1, K-1]\!]$, are derived by shifting the first row: $\boldsymbol{T_g}[i,j] = \boldsymbol{T_g}[0, j-i]$ if $j \geq i$, and 0 otherwise. For a tail-biting encoder, $\boldsymbol{T_g}$ is circulant and rows $i \in [\![1, K-1]\!]$ result from a circular shift of the i positions of the first row.

Let $\mathcal{P}$ and $\overline{\mathcal{P}}$ denote the lists of punctured and unpunctured positions, respectively. The number of unpunctured parity bits is $M_{\overline{P}} = K - M_P$, where $M_P = |\mathcal{P}|$ is the cardinality of $\mathcal{P}$. Let $\boldsymbol{y_p} = [y_i]_{i \in \mathcal{P}}$ and $\boldsymbol{y_{\bar{p}}} = [y_i]_{i \in \overline{\mathcal{P}}}$ denote the vectors of punctured and unpunctured parity bits, respectively. These vectors can be concatenated as $[\boldsymbol{y_{\bar{p}}}, \boldsymbol{y_p}]$, where the first M elements are unpunctured bits and the rest are punctured bits. This concatenation is derived from $\boldsymbol{y}$ using a permutation matrix $\boldsymbol{\Pi_{\bar{p}}}$: $[\boldsymbol{y_{\bar{p}}}, \boldsymbol{y_p}] = \boldsymbol{y\Pi_{\bar{p}}}$.

Assume two $K \times K$ matrices, $\boldsymbol{T_{g_x}}$ and $\boldsymbol{T_{g_y}}$, constructed using a recursive polynomial g_x and a parity polynomial g_y, respectively. The systematic and parity vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ can be expressed as a function of the state vector $\boldsymbol{s}$ as follows:

$$\boldsymbol{sT_{g_x}} = \boldsymbol{x}, \tag{4}$$

$$\boldsymbol{sT_{g_y}} = \boldsymbol{y}. \tag{5}$$

These equations show that $(\boldsymbol{x}, \boldsymbol{y})$ can be generated by encoding $\boldsymbol{s}$ using a non-systematic, non-recursive CC with polynomials (g_x, g_y) [14]. The commutative property of the triangular and circulant Toeplitz matrices allows for $\boldsymbol{T_{g_x}T_{g_y}} = \boldsymbol{T_{g_y}T_{g_x}}$. Multiplying by $\boldsymbol{T_{g_y}}$ and $\boldsymbol{T_{g_x}}$ on both sides of (4) and (5), respectively, gives the following.

$$\boldsymbol{xT_{g_y}} = \boldsymbol{yT_{g_x}}. \tag{6}$$

This equation defines the relationship between the systematic and parity bits after encoding with $\boldsymbol{T_{g_y}}$ and $\boldsymbol{T_{g_x}}$. The ultimate goal is to find a similar expression in which the systematic vector is only expressed as a function of the unpunctured vector through an alternative encoding structure represented by two matrices $\boldsymbol{G_{\bar{p}}}$ and $\boldsymbol{T_{\bar{p}}}$:

$$\boldsymbol{xG_{\bar{p}}} = \boldsymbol{y_{\bar{p}}T_{\bar{p}}}. \tag{7}$$

To derive $\boldsymbol{G_{\bar{p}}}$ and $\boldsymbol{T_{\bar{p}}}$, let $\boldsymbol{T}$, $\boldsymbol{D}$ be two invertible matrices s.t.:

$$\text{(i) } \boldsymbol{T} = \boldsymbol{T_{g_x}D}, \tag{8}$$

$$\text{(ii) } \forall\ (i,j) \in \overline{\mathcal{P}} \times \mathcal{P}, \boldsymbol{T}[i,j] = \boldsymbol{T}[j,i] = 0. \tag{9}$$

Both conditions imply that permuting the rows and columns of $\boldsymbol{T}$ using $\boldsymbol{\Pi_{\bar{p}}}$ results in:

$$\boldsymbol{\Pi_{\bar{p}}^\dagger T \Pi_{\bar{p}}} = \begin{bmatrix} \boldsymbol{T_{\bar{p}}} & \boldsymbol{0}_{M_{\overline{P}} \times M_P} \\ \boldsymbol{0}_{M_P \times M_{\overline{P}}} & \boldsymbol{T_p} \end{bmatrix}, \tag{10}$$

where $\boldsymbol{T_{\bar{p}}} = [\boldsymbol{T}_{i,j}]_{(i,j) \in \overline{\mathcal{P}}^2}$, $\boldsymbol{T_p} = [\boldsymbol{T}_{i,j}]_{(i,j) \in \mathcal{P}^2}$, and $\boldsymbol{0}_{A \times B}$ is a null matrix of size $A \times B$. Since $[\boldsymbol{y_{\bar{p}}}, \boldsymbol{y_p}]\boldsymbol{\Pi_{\bar{p}}^\dagger} = \boldsymbol{y}$, it follows

$$\begin{aligned} \boldsymbol{yT\Pi_{\bar{p}}} &= [\boldsymbol{y_{\bar{p}}}, \boldsymbol{y_p}]\boldsymbol{\Pi_{\bar{p}}^\dagger T \Pi_{\bar{p}}} \\ &= [\boldsymbol{y_{\bar{p}}T_{\bar{p}}}, \boldsymbol{y_pT_p}]. \end{aligned} \tag{11}$$

To preserve the validity of (6), both sides are multiplied by $\boldsymbol{D\Pi_{\bar{p}}}$. Similarly to (9), let $\boldsymbol{G} = \boldsymbol{T_{g_y}D}$, with its permuted form $\boldsymbol{G\Pi_p} = [\boldsymbol{G_{\bar{p}}}, \boldsymbol{G_p}]$. The matrices $\boldsymbol{G_{\bar{p}}}$ and $\boldsymbol{G_p}$ are composed of the first $M_{\overline{P}}$ and last M_P columns of $\boldsymbol{G\Pi_p}$, respectively. Equation (6) is then rewritten as:

$$[\boldsymbol{xG_{\bar{p}}}, \boldsymbol{xG_p}] = [\boldsymbol{y_{\bar{p}}T_{\bar{p}}}, \boldsymbol{y_pT_p}]. \tag{12}$$

Since the punctured parity vector $\boldsymbol{y_p}$ provides no information during decoding, equation (12) reduces to (7).

In the specific case of a CC with rate $m/(m+1)$ and a regular puncturing pattern with a fixed interval of $m+1$, the following properties hold: i) there exists a matrix $\boldsymbol{D}$ satisfying the condition in (9), that can be derived from a polynomial g_{D}; ii) $\boldsymbol{T_{\bar{p}}}$ is a Toeplitz matrix generated by a polynomial g_{T}; iii) the set of rows in $\boldsymbol{G_{\bar{p}}}$ starting at index $r \in [\![0, m-1]\!]$ and spaced by $m+1$ form a submatrix $\boldsymbol{G_{\bar{p}}^{(r)}}$. This submatrix is a truncated Toeplitz matrix derived from a polynomial $g_{\text{G}}^{(r)}$, where its initial row and last column may be truncated.

The matrix $\boldsymbol{T_{\bar{p}}}$ acts as the generator matrix for a non-recursive convolutional encoder, denoted E_{T}, associated with the polynomial g_{T}. Similarly, $\boldsymbol{G_{\bar{p}}}$ corresponds to m convolutional encoders, denoted $E_{\text{G}}^{(r)}$, $r \in [\![0, m-1]\!]$, each generated by the polynomial $g_{\text{G}}^{(r)}$. The inputs of E_r may be zero-padded, and its outputs are truncated to align with the

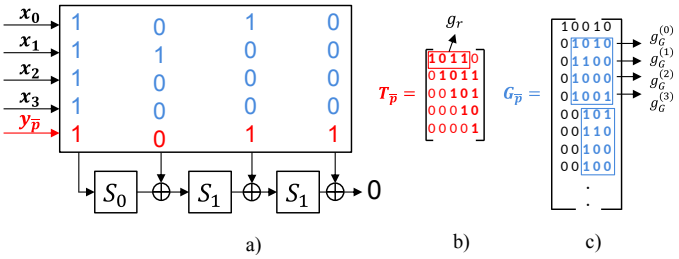

Fig. 5. Proposed alternative representation for a rate-$4/5$ ($m = 4$) CC with $g_x = (15)_8$ and $g_y = (13)_8$.

truncated Toeplitz structure of $\boldsymbol{G}_{\overline{\boldsymbol{P}}}$. When the outputs of all $E_{\mathrm{G}}^{(r)}$ and E_{T} are summed, the result should be zero. Due to the linearity of the codes, these encoders can be represented as a single unified structure depicted in Figure 5a. This alternative encoder structure accepts m systematic bits and one unpunctured parity bit as input, processes them through a linear transformation matrix composed of $m + 1$ rows of polynomial coefficients from $g_{\mathrm{G}}^{(r)}$ and g_{T}, and outputs a result that is added to the outputs of a shift register structure similar to those used in convolutional encoders.

In the end, the proposed representation reduces the number of trellis stages from K to $M_{\overline{P}}$, thereby decreasing the complexity and latency of MAP decoding. Although the computational demand per trellis stage exceeds that of traditional puncture-based decoding, this issue can be effectively mitigated through the use of dual MAP decoding [15].

B. Application Example

Consider a coding rate of $R = 4/5$ and an information block size $K = 18$, with unpunctured indices $\overline{\mathcal{P}} = \{0, 4, 8, 12, 16\}$. Using the polynomials $g_x = (15)_8$ and $g_y = (13)_8$, expressed in octal base, the first rows of matrices $\boldsymbol{T}_{\boldsymbol{g_x}}$ and $\boldsymbol{T}_{\boldsymbol{g_y}}$ are their binary representations, extended to $K = 18$ bits: $[1\ 1\ 0\ 1\ 0\ \ldots\ 0]$ and $[1\ 0\ 1\ 1\ 0\ \ldots\ 0]$. The constraint in (9) is satisfied, for example, using a Toeplitz matrix $\boldsymbol{D}$ generated from the polynomial $(1347)_8$.

The matrix $\boldsymbol{T}_{\overline{\boldsymbol{P}}}$, derived via (10), is a Toeplitz matrix generated by the polynomial $(13)_8$, as illustrated in Figure 5b. The corresponding matrix $\boldsymbol{G}_{\overline{\boldsymbol{P}}}$ is shown in Figure 5c, where the sub-matrix highlighted within the blue regions represents the 4 first rows of the linear transform from the alternative representation shown in Figure 5a. Each row of this linear transform matrix corresponds to the binary representation of polynomials $g_{\mathrm{G}}^{(0)} = (5)_8, g_{\mathrm{G}}^{(1)} = (3)_8, g_{\mathrm{G}}^{(2)} = (1)_8, g_{\mathrm{G}}^{(3)} = (11)_8$, and $g_{\mathrm{T}} = (13)_8$, respectively.

IV. Conclusion

In this paper, two novel perspectives on coding and decoding techniques for CCs are introduced: The development of a spatially-coupled decoder architecture, and an alternative representation of punctured CCs. The proposed spatially-coupled CC enables highly parallel decoding architectures, achieving higher throughputs by decoding one frame per clock cycle. Coupling 3 frames with the proposed schedule reduces the memory requirements by up to 66% compared to uncoupled decoders of equivalent frame sizes. Performance evaluations show gains of more than 0.75 dB for 8 half iterations.

Additionally, an alternative representation for punctured CCs is proposed. In this approach, punctured and unpunctured parities are separated to construct two generator matrices, enabling a linear transformation function. This transform block, implemented with a feed-forward convolutional code, ensures output consistency with classical punctured CCs. The proposed representation reduces the number of trellis stages, reducing MAP decoding complexity and latency. The complexity within a trellis stage can be further reduced through the use of dual MAP decoding.

Acknowledgments

This work was partially funded by the French National Research Agency TurboLEAP project (ANR-20-CE25-0007).

References

[1] Third Generation Partnership Project, *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (3GPP TS 36.212 version 18.0.0 Release 18)* , Sep. 2023.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *IEEE Int. Conf. on Commun. (ICC)*, vol. 2, May 1993, pp. 1064–1070 vol.2.

[3] S. Weithoffer, C. Abdel Nour, N. Wehn, C. Douillard, and C. Berrou, "25 Years of Turbo Codes: From Mb/s to beyond 100 Gb/s," in *Int. Symp. on Turbo codes and iter. proc. (ISTC)*, Dec 2018, pp. 1–6.

[4] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, and et al., "Short Block-Length Codes for Ultra-Reliable Low Latency Communications," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 130–137, 2019.

[5] S. Weithoffer, R. Klaimi, C. Abdel Nour, N. Wehn, and C. Douillard, "Fully Pipelined Iteration Unrolled Decoders-The Road to Tb/s Turbo Decoding," in *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing-(ICASSP 2020)*, Barcelona, Spain, May 2020.

[6] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, 2017.

[7] M. Mahdavi, L. Liu, O. Edfors, M. Lentmaier, N. Wehn, and et al., "Towards Fully Pipelined Decoding of Spatially Coupled Serially Concatenated Codes," in *2021 11th Int. Symp. Topics Coding (ISTC)*, 2021, pp. 1–5.

[8] M. Mahdavi, S. Weithoffer, M. Herrmann, L. Liu, O. Edfors, and et al., "Spatially coupled serially concatenated codes: Performance evaluation and vlsi design tradeoffs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 5, pp. 1962–1975, 2022.

[9] J. Sun and O. Y. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. on Inf. Theory*, vol. 51, no. 1, pp. 101–119, Jan. 2005.

[10] C. Berrou, Y. Saouter, C. Douillard, S. Kerouedan, and M. Jezequel, "Designing good permutations for turbo codes: towards a single model," in *IEEE Int. Conf. on Commun. (ICC)*, June 2004, pp. 341–345.

[11] V. H. S. Le, C. Abdel Nour, E. Boutillon, and C. Douillard, "Revisiting the Max-Log-Map algorithm with SOVA update rules: new simplifications for high-radix SISO decoders," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 1991–2004, 2020.

[12] S. Weithoffer, G. Aousaji, J. Nadal, and C. Abdel Nour, "Iteration Overlap for Low-Latency Turbo Decoding," in *Int. Symp. Topics Coding (ISTC)*, 2023, pp. 1–5.

[13] J. Cain, G. Clark, and J. Geist, "Punctured convolutional codes of rate(n-1)/nand simplified maximum likelihood decoding (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 25, no. 1, pp. 97–100, 1979.

[14] G. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inf. Theory*, vol. 16, no. 6, pp. 720–738, 1970.

[15] S. Riedel, "MAP decoding of convolutional codes using reciprocal dual codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1176–1187, 1998.