# Dual Trellis Construction for High-Rate Punctured Convolutional Codes

Vinh Hoang Son Le, Charbel Abdel Nour, Emmanuel Boutillon, Catherine Douillard

# Dual Trellis Construction for High-Rate Punctured Convolutional Codes

Vinh Hoang Son Le*, Charbel Abdel Nour*, Emmanuel Boutillon† and Catherine Douillard*
*IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
Email: {vinh.le, charbel.abdelnour, catherine.douillard}@imt-atlantique.fr
†Lab-STICC, UMR CNRS 6285, Université Bretagne Sud, Lorient, France
Email: emmanuel.boutillon@univ-ubs.fr

***Abstract*—Puncturing a low-rate convolutional code to generate a high-rate code has some drawback in terms of hardware implementation. In fact, a Maximum A-Posteriori (MAP) decoder based on the original trellis will then have a decoding throughput close to the decoding throughput of the mother non-punctured code. A solution to overcome this limitation is to perform MAP decoding on the dual trellis of a high-rate equivalent convolutional code. In the literature, dual trellis construction is only reported for specific punctured codes with rate $k/(k+1)$. In this paper, we propose a multi-step method to construct the equivalent dual code defined by the corresponding dual trellis for any punctured code. First, the equivalent non-systematic generator matrix of the high-rate punctured code is derived. Then, the reciprocal parity-check matrix for the construction of the dual trellis is deduced. As a result, we show that the dual-MAP algorithm applied on the newly constructed dual trellis yields the same performance as the original MAP algorithm while allowing the decoder to achieve a higher throughput. When applied to turbo codes, this method enables highly efficient implementations of high-throughput high-rate turbo decoders.**

***Index Terms*—Convolutional codes, puncturing, high-rate codes, dual trellis, turbo codes**

## I. Introduction

Nowadays, the demand for high data rate communications is steadily increasing. Consequently, in order to limit the occupied bandwidth, transmissions using forward error correction (FEC) schemes with high coding rates have to be considered to keep the number of transmitted redundant data as low as possible.

In this paper, we consider rate-$k/n$ convolutional codes with *constraint length* $\nu$, according to the definition given in [1]. When high coding rates $r > 1/2$ are targeted, two main approaches exist. One can use directly a convolutional generator matrix of size $k \times n$. The code is then referred to as a *true high-rate* convolutional code. The second method, also widely used consists in puncturing a low-rate *mother code* (e.g. with rate $1/2$) [2]. The puncturing method has the advantage of being flexible, since the code rate can be changed without modifying the structure of the encoder-decoder while the true high-rate code yields a better error correction performance in the asymptotic region [3] and is, in theory, more appropriate to high-throughput applications since $k$ bits are decoded at each trellis stage.

For FEC codes requiring iterative decoding, such as turbo codes [4], soft-input soft-output algorithms are needed to decode the convolutional codes. The *maximum a posteriori* (MAP) algorithm or its sub-optimal logarithmic version (Max-Log-MAP) [5] can then be employed to produce the soft output and the extrinsic information. However, for true high-rate convolutional codes, the MAP algorithm is preferably implemented using the *dual trellis* [6], [7] instead of the original trellis. This is particularly useful in order to reduce the decoding complexity, since the dual trellis, i.e., the trellis of the rate-$(n-k)/n$ dual code is used for decoding. In this case, it is expected that a higher throughput and a lower decoding latency can then be achieved. In fact for punctured codes, the decoder implements the MAP algorithm using the trellis of the low-rate mother code, even for high coding rates. The MAP algorithm employed with the dual trellis is referred to as the *dual-MAP* algorithm in this work.

In this paper, we propose a general procedure to construct the dual trellis for any high-rate punctured convolutional code to enable the application of the dual-MAP algorithm for decoding.

The paper is organized as follows. Section II introduces the two common approaches to obtain high-rate convolutional codes: punctured and true high-rate convolutional codes. The respective advantages and drawbacks of these two methods are analyzed in terms of decoding algorithms. Section III describes the proposed procedure to derive the dual trellis from the high-rate punctured convolutional code. Then, Section IV gives a construction example and the corresponding comparison of simulation results of the dual-MAP algorithm with the classical MAP algorithm. Finally, Section V concludes the paper.

## II. High-Rate Punctured Convolutional codes and True High-Rate Convolutional codes

### A. High-Rate Punctured Convolutional Codes

Puncturing [2] is usually employed to obtain high-rate codes from a low-rate code with patterns indicating which bits are transmitted and which bits are discarded (punctured). For instance, starting from an original rate-$1/2$ code, a rate-$4/5$ code can be obtained using the following puncturing pattern

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

where the first row indicates the pattern applied for the systematic bits and the second row for the parity bits of the convolutional code. In this example, no systematic bit is punctured and only one out of four parity bits is transmitted.

The search for good puncturing patterns has been intensively studied in [8], [9] for feedforward convolutional codes and in [3], [10] for recursive systematic convolutional (RSC) codes, employed in turbo codes. In [10], the authors investigated the joint optimization of puncturing patterns and interleavers for turbo codes.

The advantage of puncturing is that the MAP decoder structure designed for the original code can be reused for the high-rate punctured codes. Therefore, the complexity for decoding high-rate punctured codes is hardly different from the original code. Moreover, the code enjoys the flexibility to change the code rate by applying different puncturing patterns at the encoder, without changing the decoder structure. However, this also implies that, for high coding rates, the decoder inherits the throughput and decoding latency of the mother code. Furthermore, for high-rate schemes, the MAP decoders usually adopt the acquisition (ACQ) technique with long acquisition lengths to maintain the performance thus increasing further the decoding latency [11]. Consequently, for high-rate schemes, other code families such as LDPC codes may provide a better performance in terms of decoding complexity, throughput and latency compared to turbo codes.

### B. True High-Rate Convolutional Codes

Another alternative to achieve a rate-$k/n$ convolutional code is to encode with a $k \times n$ generator matrix (we also call it *encoder* below). In the conventional trellis representation of a convolutional code, $2^k$ branches or transitions go to and from each state. If the code has $\nu$ memory elements, there is a total of $2^{\nu+k}$ state transitions at each trellis stage. Therefore, for high-rate codes (e. g. for $n = k+1$ and $k \geq 4$), the decoding algorithm has to process a large number of branches in each trellis stage, thus significantly increasing the decoding complexity.

The task of decoding a true high-rate convolutional code can be simplified by applying the dual-MAP algorithm. For a rate-$k/n$ convolutional code $\mathcal{C}$ encoded by generator matrix $\mathbf{G}(D)$, there exists an associated rate-$(n-k)/n$ *dual code* $\mathcal{C}^{\perp}$ generated by matrix $\widetilde{\mathbf{H}}(D)$ such that $\mathbf{G}\widetilde{\mathbf{H}}^{\top} = 0$. In other words, any codeword generated by $\mathbf{G}(D)$ should be orthogonal with all codewords generated by $\widetilde{\mathbf{H}}(D)$. The generator matrix $\widetilde{\mathbf{H}}(D)$ is usually referred to as the *reciprocal parity-check matrix* since it is the reciprocal of the parity-check matrix $\mathbf{H}(D)$ of code $\mathcal{C}$. We will refer to the trellis constructed according to $\widetilde{\mathbf{H}}(D)$ as *reciprocal dual trellis* or *dual trellis* for short. In a dual trellis stage, there are $2^{(n-k)}$ branches going to and from a state and there are a total of $2^{(n-k+\nu)}$ branches at each trellis stage. This number is much less than the number of branches processed in the original trellis when high-rate schemes are employed. Hence, the dual-MAP algorithm is preferred to the MAP algorithm in this case.

As far as we know, there is little research on generator matrices for high-rate convolutional codes. The most relevant work that we found is [3] where the authors searched for good rate-$k/(k+1)$ systematic encoders. The following example is drawn from table IV in [3], where a $(5, 4, 3)$ convolutional code is described by the systematic generator matrix $\mathbf{G_{sys}}(D)$:

$$\mathbf{G_{sys}}(D) = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{1+D+D^2+D^3}{1+D+D^3} \\ 0 & 1 & 0 & 0 & \frac{1+D+D^2}{1+D+D^3} \\ 0 & 0 & 1 & 0 & \frac{1+D^2+D^3}{1+D+D^3} \\ 0 & 0 & 0 & 1 & \frac{1+D^3}{1+D+D^3} \end{pmatrix}. \quad (1)$$

In the case of a rate-$k/(k+1)$ code, the parity-check matrix $\mathbf{H}(D)$ can be easily derived since $\mathbf{G}(D)\mathbf{H}^{\top}(D) = 0$ as

$$\mathbf{H}(D) = \big(1 + D + D^2 + D^3,\ \ 1 + D + D^2,\ \ 1 + D^2 + D^3,\ \ 1 + D^3,\ \ 1 + D + D^3\big) \quad (2)$$

and the reciprocal parity-check matrix $\widetilde{\mathbf{H}}(\mathbf{D})$ can be obtained as follows

$$\begin{aligned} \widetilde{\mathbf{H}}(D) &= D^{\nu}\mathbf{H}(D^{-1}) = D^3\mathbf{H}(D^{-1}) \\ &= \big(1 + D + D^2 + D^3,\ \ D + D^2 + D^3,\ \ 1 + D + D^3,\ \ 1 + D^3,\ \ 1 + D^2 + D^3\big) \end{aligned} \quad (3)$$

However, based on the codes of table IV in [3], we observe that, for high-rate convolutional codes, trellis termination with tail-biting cannot always be applied. Actually, tail-biting is the preferred trellis termination technique for the component codes of a turbo encoder. When the trellis is terminated by forcing the final state to zero, specific extra processing is necessary for edge bits included to this effect. The introduction of these bits leads to uneven protection between the bits of the information sequence that can entail an undesirable error

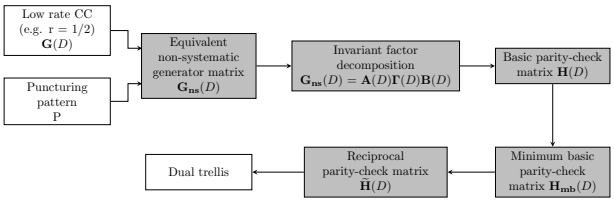


Fig. 1. Procedure for obtaining the reciprocal dual trellis of a high-rate punctured convolutional code.

floor. Additionally, for a rate-$k/n$ code, the number of overhead bits required to perform the zero termination is $n\nu$ bits: for very high-rate codes and short block sizes, the loss in spectral efficiency due to overhead bits may be significant. On the contrary, tail-biting can be applied to most of the low-rate mother codes making the corresponding decoder particularly attractive for highly parallel hardware implementations [12]. The tail-biting property is kept when the code is punctured: this motivated the derivation of the reciprocal dual trellis of high-rate punctured codes.

## III. Constructing Reciprocal Dual Trellis for High-Rate Punctured Convolutional Codes

This section proposes a fully generic procedure to construct the dual trellis for any convolutional code and any puncturing pattern. Different from [3] where only rate-$k/(k+1)$ codes are considered, this method is based on transformations proposed by Forney in [1]. The generic procedure is presented in Fig. 1. The main idea of the procedure is that, for a target rate-$k/n$ code, we find the non-systematic generator matrix $\mathbf{G_{ns}}(D)$ of size $k \times n$ equivalent to the original rate-$k/n$ punctured convolutional code. Then, parity-check matrix $\mathbf{H}(D)$ and its reciprocal $\widetilde{\mathbf{H}}(D)$ can be derived and the dual trellis is constructed directly from matrix $\widetilde{\mathbf{H}}(D)$. The dual-MAP algorithm then decodes the received codeword based on the dual trellis yielding the soft estimate on the punctured codeword. In this section, we reuse several notions and definitions introduced in [1].

Two generator matrices or encoders are said to be *equivalent* if and only if they generate the same codeword space $\mathcal{C}$. From an algebraic perspective, two rate-$k/n$ convolutional generator matrices $\mathbf{G}(D)$ and $\mathbf{G}'(D)$ are equivalent if and only if there exists a $k \times k$ non-singular matrix $\mathbf{T}(D)$ such that

$$\mathbf{G}(D) = \mathbf{T}(D)\mathbf{G}'(D). \tag{4}$$

### A. Equivalent Non-Systematic High-Rate Generator Matrix

The first step towards the reciprocal dual trellis involves converting the punctured convolutional code into a non-systematic encoding form, as described in [13]. This transformation has previously been used to assess the performance of convolutional codes, since all equivalent forms share the same Hamming weight spectrum.

In order to get a rate-$k/n$ punctured convolutional code, we start from the following general form of an encoder generating a rate-$1/N$ code

$$\mathbf{G}(D) = \begin{pmatrix} G_0(D) & G_1(D) & \dots & G_{N-1}(D) \end{pmatrix} \tag{5}$$

where $G_i(D)$, $i = 0, \dots, N-1$ are the generator polynomials. This convolutional code can also be viewed as a rate-$k/kN$ code, for any value of $k$. Then, we derive an equivalent encoder defined by $kN$ polynomials obtained by splitting each polynomial $G_i(D)$, $i = 0, \dots, N-1$ into $k$ sub-polynomials $g_{i,j}(D)$, $j = 0, \dots, k-1$ as follows

$$G_i(D) = \sum_{j=0}^{k-1} D^j g_{i,j}(D^k). \tag{6}$$

The element of the resulting $k \times kN$ generator matrix $\mathbf{G}'(D)$ at row $p$ and column $q$, $G'_{p,q}(D)$, is defined as:

$$\begin{cases} G'_{p,q}(D) = g_{q \bmod n, \lfloor q/N \rfloor - p}(D) & \text{if } p \times N \leq q \\ G'_{p,q}(D) = D \times g_{q \bmod n, \lfloor q/N \rfloor - p + k}(D) & \text{if } p \times N > q \end{cases} \tag{7}$$

The encoder of the rate-$k/n$ code is then obtained by selecting $n$ out of $kN$ columns in $\mathbf{G}'(D)$ according to a selected puncturing pattern.

For example, let us illustrate the decomposition principle for $k = 3$ and $N = 2$. The general form of an encoder generating a rate-$1/2$ code is $\mathbf{G}(D) = (G_0(D), G_1(D))$. Polynomials $G_0(D)$ and $G_1(D)$ are each decomposed into $k = 3$ sub-polynomials: $\{g_{0,j}(D)\}_{j=0,1,2}$ and $\{g_{1,j}(D)\}_{j=0,1,2}$. To lighten notations, polynomials $g_{i,j}(D)$ will be denoted as $g_{i,j}$ in the sequel. The generator matrix $\mathbf{G}'(D)$ of the equivalent rate-$3/6$ code is

$$\begin{pmatrix} g_{0,0} & g_{1,0} & g_{0,1} & g_{1,1} & g_{0,2} & g_{1,2} \\ Dg_{0,2} & Dg_{1,2} & g_{0,0} & g_{1,0} & g_{0,1} & g_{1,1} \\ Dg_{0,1} & Dg_{1,1} & Dg_{0,2} & Dg_{1,2} & g_{0,0} & g_{1,0} \end{pmatrix}. \tag{8}$$

The non-systematic punctured generator matrix $\mathbf{G_{ns}}(D)$ of the rate-$k/n$ code is then obtained by selecting $n$ columns out of $kN$ in $\mathbf{G}'(D)$, according to a selected puncturing pattern. For instance, if columns $1, 2, 3$ and $5$ in $\mathbf{G}'(D)$ are selected, the following matrix generates a rate-$3/4$ code

$$\mathbf{G_{ns}}(D) = \begin{pmatrix} g_{0,0} & g_{1,0} & g_{0,1} & g_{0,2} \\ Dg_{0,2} & Dg_{1,2} & g_{0,0} & g_{0,1} \\ Dg_{0,1} & Dg_{1,1} & Dg_{0,2} & g_{0,0} \end{pmatrix}. \tag{9}$$

### B. Reciprocal Parity-Check Matrix

In this section, we introduce a general procedure to obtain the reciprocal parity-check matrix $\widetilde{\mathbf{H}}(D)$ from the non-systematic generator matrix $\mathbf{G_{ns}}(D)$. A convolutional encoder or encoding matrix $\mathbf{G}(D)$ is called *basic* if it has a right inverse $\mathbf{G}^{-1}(D)$. The right inverse

existence of an encoding matrix ensures that the encoder is a bijective function. Therefore, given any two different inputs, the basic encoder always yields two different codewords.

*1) Invariant-factor decomposition of the generator matrix:* Based on the invariant-factor theorem, as in [1], or on the Smith form, as in [14], the invariant-factor decomposition of the generator matrix $\mathbf{G}(D)$ of a rate-$k/n$ convolutional code gives

$$\mathbf{G}(D) = \mathbf{A}(D)\mathbf{\Gamma}(D)\mathbf{B}(D), \tag{10}$$

where $\mathbf{A}(D)$ is a $k \times k$ polynomial matrix with unit determinant, $\mathbf{B}(D)$ is a $n \times n$ polynomial matrix with unit determinant, thus having an polynomial matrix inverse $\mathbf{B}^{-1}(D)$, and $\mathbf{\Gamma}(D)$ is a $k \times n$ diagonal matrix, whose elements are called the invariant factors of $\mathbf{G}(D)$ and are unique. In fact, matrix $\mathbf{\Gamma}(D)$ is obtained by permuting and linearly combining the rows and the columns of matrix $\mathbf{G}(D)$. Since the permutation or linear combination of rows (or columns) can be represented by the pre- (or post-) multiplication of a square matrix with unit determinant by $\mathbf{G}(D)$, $\mathbf{A}(D)$ is the the result of all row operations and matrix $\mathbf{B}(D)$ is the result of all column operations.

As pointed out in [1], matrix $\mathbf{U}(D)$ consisting of the first $k$ rows of $\mathbf{B}(D)$ is a basic encoding matrix equivalent to $\mathbf{G}(D)$. Since the polynomial matrix $\mathbf{B}(D)$ has a polynomial inverse $\mathbf{B}^{-1}(D)$, the matrix consisting of the last $(n-k)$ columns of $\mathbf{B}^{-1}(D)$ is the transpose of the parity-check matrix, i.e., $\mathbf{H}^{\top}(D)$. This can be explained by the fact that $\mathbf{B}(D)\mathbf{B}^{-1}(D) = \mathbf{I}_n$ where $\mathbf{I}_n$ is the $n \times n$ identity matrix. Then, the inner product between row $i \in \{0, \ldots, k-1\}$ of $\mathbf{B}(D)$ and column $j \in \{n-k, \ldots, n-1\}$ of $\mathbf{B}^{-1}(D)$ is the element at row $i$ and column $j$ of the identity matrix $\mathbf{I}_n$. Since $(\mathbf{I}_n)_{i,j} = 0$ for $j > i$, $\mathbf{U}(D)\mathbf{H}^{\top}(D) = \mathbf{0}$ and consequently, $\mathbf{G}(D)\mathbf{H}^{\top}(D) = \mathbf{0}$. Note that matrix $\mathbf{H}(D)$ is basic since it has a right inverse which is the transpose of the matrix consisting of the last $(n-k)$ rows of $\mathbf{B}(D)$.

*2) Minimal basic parity-check matrix:* Let $\mathbf{G}(D)$ be a $k \times n$ encoding matrix. We define the constraint length for the $i$th input as

$$\nu_i = \max_{j=0,\ldots,k-1} \left\{ \deg\left(G_{i,j}(D)\right) \right\}, \tag{11}$$

where $G_{i,j}(D)$ is the polynomial element of $\mathbf{G}(D)$ at row $i$ and column $j$ and $\deg()$ denotes the degree of the polynomial. Then, we define the *overall constraint length* of the encoder as the sum of the constraint lengths for all inputs

$$\nu = \sum_{i=0}^{k} \nu_i \tag{12}$$

According to Theorem 7 in [1], if $\mathbf{G}(D)$ is a basic encoder with overall constraint length $\nu$, then there exists an associated basic parity-check matrix $\mathbf{H}(D)$ with overall constraint length $\nu$.

However, we observed that, in the previous decomposition, the parity-check matrix $\mathbf{H}(D)$ consisting of the last $(n-k)$ columns of $\mathbf{B}^{-1}(D)$ usually has an overall constraint length greater than $\nu$. Therefore, we need to find the minimal basic parity-check matrix $\mathbf{H_{mb}}(D)$ with overall constraint length $\nu$ that is equivalent to $\mathbf{H}(D)$. To this end, the authors in [14] proposed an algorithm for finding the minimal basic form of an encoding matrix, called *Algorithm MB*. The main idea of this algorithm is to lower gradually the constraint length of the parity-check matrix by linearly combining multiple rows. Then, after a number of steps, its overall constraint length becomes equal to $\nu$. Since the operation of linearly combining matrix rows can be represented by the pre-multiplication of a unit determinant matrix by $\mathbf{H}(D)$, the resulting minimal basic parity-check matrix $\mathbf{H_{mb}}(D)$ is equivalent to $\mathbf{H}(D)$.

Assuming that $\mathbf{G}(D)$ is a $k \times n$ encoding matrix, let $[\mathbf{G}(D)]_h$ be a (0,1)-matrix with 1 in the position $(i,j)$ if $\deg\left(G_{i,j}(D)\right) = \nu_i$ and 0 otherwise. The algorithm MB proceeds as follows:

- **Step 1:** If $[\mathbf{G}(D)]_h$ has full rank, then $\mathbf{G}(D)$ is a minimal basic encoding matrix and the algorithm stops; otherwise go to **Step 2**.
- **Step 2:** $[\mathbf{G}(D)]_h$ does not have full rank, meaning that there are at least two linearly dependent rows in $[\mathbf{G}(D)]_h$. Without loss of generality, we let $[\mathbf{r}_{i_1}], [\mathbf{r}_{i_2}], \ldots, [\mathbf{r}_{i_d}]$ denote the set of linearly dependent rows in $[\mathbf{G}(D)]_h$ such that $\nu_{i_d} \geq \nu_{i_j}$, $j = 0, \ldots, d-1$. Then, if we let $\mathbf{r}_{i_1}, \mathbf{r}_{i_2}, \ldots, \mathbf{r}_{i_d}$ be the corresponding set of rows in $\mathbf{G}(D)$, we can lower $\nu_{i_d}$ by adding

  $$D^{\nu_{i_d}-\nu_{i_1}}\mathbf{r}_{i_1} + D^{\nu_{i_d}-\nu_{i_2}}\mathbf{r}_{i_2} + \ldots + D^{\nu_{i_d}-\nu_{i_{d-1}}}\mathbf{r}_{i_{d-1}}$$

  to the $i_d$th row of $\mathbf{G}(D)$. Return to **Step 1**.

*3) Reciprocal parity-check matrix:* After having derived the minimal basic parity-check matrix $\mathbf{H_{mb}}(D)$, the reciprocal parity check matrix $\widetilde{\mathbf{H}}(D)$ can be directly obtained by

$$\widetilde{\mathbf{H}}_i(D) = D^{\nu_i}\mathbf{H_{mb}}_i(D^{-1}),\ 0 \leq i < n-k \tag{13}$$

where $\widetilde{\mathbf{H}}_i(D)$ and $\mathbf{H_{mb}}_i(D)$ are respectively the $i$th row of $\widetilde{\mathbf{H}}(D)$ and $\mathbf{H_{mb}}(D)$.

Matrix $\widetilde{\mathbf{H}}(D)$ is the encoder generating the dual codeword $\widetilde{\mathbf{v}}$ that is orthogonal to any codeword $\mathbf{v}$ generated by the original high-rate punctured convolutional encoder. The dual-MAP algorithm can be run using the dual trellis generated by $\widetilde{\mathbf{H}}(D)$ to yield the soft estimates related to codeword $\mathbf{v}$.

To summarize, the procedure to obtain the dual trellis starting from the original high-rate punctured convolutional code can be described as follows:

- Find the equivalent high-rate non-systematic generator matrix $\mathbf{G_{ns}}(D)$;
- Perform the invariant-factor decomposition

$$\mathbf{G_{ns}}(D) = \mathbf{A}(D)\mathbf{\Gamma}(D)\mathbf{B}(D);$$

- Find the inverse matrix $\mathbf{B}^{-1}(D)$, then the parity-check matrix $\mathbf{H}(D)$ is the transpose of the matrix consisting of the last $(n-k)$ columns of $\mathbf{B}^{-1}(D)$;
- Apply Algorithm MB to derive the equivalent minimal basic parity-check matrix $\mathbf{H_{mb}}(D)$ from $\mathbf{H}(D)$;
- Deduce the reciprocal parity-check matrix $\widetilde{\mathbf{H}}(D)$ according to (13) and construct the dual trellis based on $\widetilde{\mathbf{H}}(D)$.

## IV. Example and Numerical Results

### A. An Example of Dual Trellis Construction

In this section, we provide an example of derivation of the dual trellis from a high-rate punctured convolutional code. The considered convolutional code is the constituent RSC code of the LTE turbo code [15], punctured to achieve rate 5/7

$$\mathbf{G_{LTE}}(D) = \begin{pmatrix} 1 & \frac{1+D+D^3}{1+D^2+D^3} \end{pmatrix}; \; P = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

thus yielding a rate-$5/9$ turbo code. In order to derive the dual trellis of the convolution code, we first convert the recursive systematic generator matrix into a non-recursive form

$$\begin{aligned} \mathbf{G}(D) &= (1+D^2+D^3)\begin{pmatrix} 1 & \frac{1+D+D^3}{1+D^2+D^3} \end{pmatrix} \\ &= \begin{pmatrix} 1+D^2+D^3 & 1+D+D^3 \end{pmatrix} \end{aligned}$$

and according to (4), $\mathbf{G_{LTE}}(D)$ and $\mathbf{G}(D)$ are equivalent.

*1) Equivalent non-systematic generator matrix:* First, we find the equivalent non-punctured rate-$5/10$ generator matrix. According to (6), we decompose $G_0(D) = 1+D^2+D^3$ and $G_1(D) = 1+D+D^3$ into 5 sub-polynomials each

$$\begin{array}{lllll} g_{0,0}=1, & g_{0,1}=0, & g_{0,2}=1, & g_{0,3}=1, & g_{0,4}=0, \\ g_{1,0}=1, & g_{1,1}=1, & g_{1,2}=0, & g_{1,3}=1, & g_{1,4}=0. \end{array}$$

Then, the non-punctured rate-$5/10$ generator matrix $\mathbf{G}'(D)$ is

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ D & D & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ D & 0 & D & D & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & D & D & 0 & D & D & 0 & 0 & 1 & 1 \end{pmatrix}$$

The length of puncturing pattern $P$ is 5. When applying it periodically to $\mathbf{G}(D)$, every second, third and fifth parity bits are punctured. Since these parity bits are equivalently generated by the even columns of $\mathbf{G}'(D)$, applying pattern $P$ to $\mathbf{G}'(D)$ amounts to removing columns 4, 6 and 10 from $\mathbf{G}'(D)$. The resulting non-systematic rate-5/7 generator matrix $\mathbf{G_{ns}}(\mathbf{D})$ is then:

$$\mathbf{G_{ns}}(D) = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ D & D & 0 & 1 & 0 & 1 & 1 \\ D & 0 & D & 0 & 1 & 1 & 0 \\ 0 & D & D & D & 0 & 0 & 1 \end{pmatrix}$$

*2) Reciprocal parity-check matrix:* For the sake of simplicity, we do not show all the details of the invariant-factor decomposition here. Readers may refer to Example 2.4 in [14] for a detailed example. The invariant-factor decomposition of $\mathbf{G_{ns}}(D)$ results in $\mathbf{A}(D)\mathbf{\Gamma}(D)\mathbf{B}(D)$ as shown in (15) at the top of the next page.

Then, the inverse matrix $\mathbf{B}^{-1}(D)$ is derived from $\mathbf{B}(D)$ by Gaussian elimination. It gives $\mathbf{B}^{-1}(D)$ equal to

$$\begin{pmatrix} 1 & 0 & 0 & 1+D & 0 & 1+D+D^2 & 1+D^2+D^3 \\ 0 & 0 & 0 & D & 0 & 1+D^2 & D^3 \\ 0 & 1 & 1 & 1+D & 0 & D+D^2 & 1+D+D^2+D^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1+D \\ 0 & 0 & 0 & 0 & 1 & D & 1+D+D^2 \\ 0 & 0 & 1 & D & 1 & D+D^2 & D^2+D^3 \end{pmatrix}$$

and parity-check matrix $\mathbf{H}(D)$ is obtained from the last 2 columns of $\mathbf{B}^{-1}(D)$ as shown in (16). We can observe that the overall constraint length of $\mathbf{H}(D)$ in (16) is 5 while the overall constraint length of $\mathbf{G_{ns}}(D)$ is only 3. Therefore, one should apply algorithm MB to matrix $\mathbf{H}(D)$. We first derive the following matrix

$$[\mathbf{H}(D)]_h = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (14)$$

Row 1 and row 2 of $[\mathbf{H}(D)]_h$ are linearly dependent. Therefore, applying **Step 2** of algorithm MB, row 2 is changed according to

$$[\text{Row 2}] \Longleftarrow [\text{Row 2}] + D \times [\text{Row 1}]$$

resulting in the minimal basic parity-check matrix shown in (17), whose overall constraint length is 3.

Finally, applying (13) yields the reciprocal parity-check matrix $\widetilde{\mathbf{H}}(D)$ given in (18).

### B. Simulation Results and Discussion

In this section, we present simulation results that compare the error correction performance of turbo codes implementing the conventional MAP algorithm and the dual-MAP algorithm based on the dual trellis, for three coding rates higher than $1/2$. The original low-rate mother code is the rate-$1/2$ constituent RSC code of the LTE turbo code. Data are transmitted in AWGN channel, using BPSK modulation. The information frame length is $K = 400$ bits. The puncturing patterns of the parity bits (Table I) and the internal Almost Regular Permutation

$$\mathbf{G_{ns}}(D) = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ D & 0 & 1 & 0 & 0 \\ D & D & D & 1 & 0 \\ 0 & D & 1{+}D & 0 & 1 \end{pmatrix}}_{\mathbf{A}(D)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}}_{\mathbf{\Gamma}(D)} \underbrace{\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1{+}D & D & 1{+}D & 1 \\ 0 & D & 0 & D^2 & 1{+}D^2 & 1{+}D^2 & 0 \\ 0 & D & 0 & 1{+}D{+}D^2 & D^2 & 1{+}D^2 & 0 \\ 0 & 1 & 0 & 0 & D & D & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{B}(D)} \tag{15}$$

$$\mathbf{H}(D) = \begin{pmatrix} 1{+}D{+}D^2 & 1{+}D^2 & D{+}D^2 & 0 & 0 & D & D{+}D^2 \\ 1{+}D^2{+}D^3 & D^3 & 1{+}D{+}D^2{+}D^3 & 1 & 1{+}D & 1{+}D{+}D^2 & D^2{+}D^3 \end{pmatrix} \tag{16}$$

$$\mathbf{H_{mb}}(D) = \begin{pmatrix} 1{+}D{+}D^2 & 1{+}D^2 & D{+}D^2 & 0 & 0 & D & D{+}D^2 \\ 1{+}D & D & 1{+}D & 1 & 1{+}D & 1{+}D & 0 \end{pmatrix} \tag{17}$$

$$\widetilde{\mathbf{H}}(D) = \begin{pmatrix} 1{+}D{+}D^2 & 1{+}D^2 & 1{+}D & 0 & 0 & D & 1{+}D \\ 1{+}D & 1 & 1{+}D & D & 1{+}D & 1{+}D & 0 \end{pmatrix} \tag{18}$$

TABLE I
PARITY PUNCTURING PATTERN FOR $K = 400$

| Turbo rate | Parity puncturing pattern |
|---|---|
| 8/11 | 1100000000000010 |
| 4/5 | 0100000000000010 |
| 8/9 | 0100000000000000 |

TABLE II
ARP INTERLEAVER PARAMETERS FOR $K = 400$

| $Q$ | $P$ | $\big(S(0), \ldots, S(Q-1)\big)$ |
|---|---|---|
| 16 | 383 | (8, 80, 311, 394, 58, 55, 250, 298, 56, 197, 280, 40, 229, 40, 136, 192) |

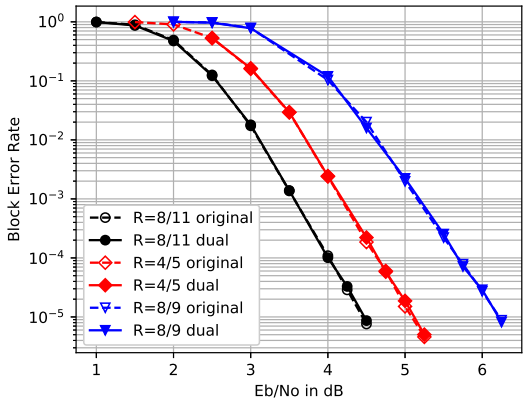


Fig. 2. Performance comparison between MAP and dual-MAP algorithms for various high-rate schemes for $K = 400$. AWGN channel and BPSK modulation.

(ARP) interleaver have been jointly optimized according to the method described in [10]. The interleaver is defined by

$$\pi(i) = \Big(Pi + S\big(i \bmod Q\big)\Big) \bmod K. \tag{19}$$

and the corresponding parameters are given in Table II.

The simulations were carried out with floating point representation of data and the number of iterations is set to 8. Fig. 2 shows that both decoding approaches yield similar error correction performance for the turbo code.

On another note, the authors of [16] carry out a comparison, in terms of throughput and circuit area, for LTE SISO decoders using the dual-MAP and the radix-4 Max-Log-MAP algorithms. They show that, for a medium coding rate such as $r = 2/3$, both algorithms yield the same throughput but the circuit area of the dual-MAP decoder is more than twice compared to the radix-4 Max-Log-MAP decoder (see Fig. 8 in [16]). However, for higher coding rates such as $r = 4/5$ or $r = 8/9$, the throughput of the Max-Log-MAP decoder remains unchanged while the circuit area increases with the coding rate, due to the need of longer decoding windows and/or long acquisition sequences for state metric initialization to avoid any correction performance loss. On the contrary, the throughput of the dual-MAP decoder is doubled from $r = 2/3$ to $r = 4/5$ and from $r = 4/5$ to $r = 8/9$, while its circuit area only increases from 140 kgates at $r = 2/3$ to 180 kgates at $r = 8/9$. This is due to the fact that, for a code rate $k/n$, the dual trellis decoder is able to process simultaneously $k$ bits while the conventional radix-4 Max-Log MAP decoder can only process 2 bits at each trellis stage. This is the main advantage of decoding using the dual trellis instead of the original trellis for high-rate convolutional codes.

## V. Conclusion

In this paper, we propose a solution to combine the advantages of punctured and true high-rate convolutional codes. We applied the dual-MAP decoding algorithm, usually used for true high-rate codes, to high-rate punctured convolutional codes. To this end, we described a generic procedure to construct the dual trellis for a punctured convolutional code. This approach was validated with simulations in AWGN channel where the dual-MAP decoder based on the derived dual trellis was shown to achieve the same performance as the MAP algorithm employed on the original trellis.

As shown in [16], using the dual trellis leads to an increase in the circuit area compared to the classical MAP decoder since a large number of extrinsic information has to be processed at the same time. However, this approach offers the advantage of high throughput decoding for high-rate coding schemes and the ratio of throughput to chip area is largely increased. Nonetheless, for future works, we aim to find a sub-optimal decoding algorithm for the dual-MAP which offers several trade-offs of performance/complexity.

## Acknowledgement

This work was partially funded by the EPIC project of the EU's Horizon 2020 research and innovation programme under grant agreement No. 760150.

## References

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

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

[3] A. Graell i Amat, G. Montorsi, and S. Benedetto, "Design and decoding of optimal high-rate convolutional codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 867–881, May 2004.

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

[5] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, June 1995, pp. 1009–1013 vol.2.

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

[7] S. Srinivasan and S. S. Pietrobon, "Decoding of high rate convolutional codes using the dual trellis," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 273–295, Jan 2010.

[8] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, April 1988.

[9] G. Bégin, D. Haccoun, and C. Paquin, "Further results on high-rate punctured convolutional codes for Viterbi and sequential decoding," *IEEE Trans. Commun.*, vol. 38, no. 11, pp. 1922–1928, Nov 1990.

[10] R. Garzón-Bohórquez, C. Abdel Nour, and C. Douillard, "Protograph-based interleavers for punctured turbo codes," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 1833–1844, May 2018.

[12] E. Boutillon, C. Douillard, and G. Montorsi, "Iterative decoding of concatenated convolutional codes: Implementation issues," *Proc. IEEE*, vol. 95, no. 6, pp. 1201–1227, June 2007.

[13] H. Sasano and S. Moriya, "A construction of high rate punctured convolutional codes," in *Int. Symp. Inf. Theory Applications*, Oct 2012, pp. 662–666.

[14] R. Johannesson and K. S. Zigangirov, *Fundamentals of convolutional coding*. Piscataway, NJ: IEEE Press, 1999.

[15] *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding*, ETSI TS 136 212 V14.8.0, Jan 2019.

[16] C. Lin, C. Wong, and H. Chang, "A 40 nm 535 Mbps multiple code-rate turbo decoder chip using reciprocal dual trellis," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2662–2670, Nov 2013.

[11] E. Boutillon, J. Sánchez-Rojas, and C. Marchand, "Simplified compression of redundancy free trellis sections in turbo decoder," *IEEE Commun. Lett.*, vol. 18, no. 6, pp. 941–944, June 2014.