Syndrome Former Trellis Construction for Punctured Convolutional Codes

Jan Geldmacher and Juergen Goetze
Information Processing Lab
TU Dortmund University, Otto-Hahn-Strasse 4, 44227 Dortmund, Germany.
Correspondence: jan.geldmacher@tu-dortmund.de

***Abstract*—In this paper syndrome former trellis construction for punctured convolutional codes (PCCs) is discussed. The class of $(n_p - 1)/n_p$-rate PCCs derived from $1/n$-rate mother codes is considered. It is shown that a syndrome former trellis of the PCC with identical complexity as the encoder trellis of the mother code can always be constructed. An explicit construction of the syndrome former trellis can be beneficial, when realizing adaptive complexity reduced decoding, or decoding with reduced state transitions, based on the syndrome decoding approach.**

I. Introduction

Punctured convolutional codes (PCCs) [1] are used in many practical applications to obtain high code rates with feasible decoding complexity. They can be derived from $1/n$-rate mother codes by systematically puncturing the outputs of the encoder. They can be decoded with the same effort as the mother code, and thus avoid the increased decoding complexity of true high-rate codes. Conventional Viterbi decoding can be applied to decode PCCs by using the trellis of the encoder $\mathbf{G}(D)$ of the mother code. The puncturing pattern is considered during the branch metric calculation or by a depuncturing stage.

An alternative maximum likelihood (ML) decoding approach is based on the trellis of the syndrome former $\mathbf{H}^T(D)$. This trellis can be used to estimate that particular correction sequence, that has the lowest weight and yields a valid code sequence, when applied to the received sequence [2]. It is also known that $\mathbf{H}^T(D)$ can be realized with the same number of memory elements as the corresponding canonical encoder [3], i.e. the complexities of syndrome former trellis and encoder trellis are identical. Syndrome decoding of PCCs is likewise possible using the trellis associated with the syndrome former $\mathbf{H}^T(D)$ of the mother code.

The syndrome decoder (SD) as proposed by Schalkwijk et. al. [2] has an interesting property: Since it decodes an error-sequence instead of a code sequence, it shows unbalanced state probabilities in the trellis. The probability of the zero-state increases with decreasing number of errors in the received sequence. An equivalent interpretation is, that for error-free parts the decoded ML path will only traverse the zero-state. This property may be used to achieve efficient decoder implementations with less state transitions. A low power decoder implementation based on the Scarce-State-Transition (SST) principle and the syndrome former trellis has for instance been described in [4].

The decoding of error sequences can also be exploited to achieve reduced complexity decoding. In [5] syndrome based decoding is applied to reduce the decoding effort by using a degenerated, less complex trellis. Another approach based on predicting and avoiding the decoding of error-free subsequences has been used in [6], [7]. The prediction is based on analyzing the syndrome $\mathbf{b}(D) = \mathbf{r}(D)\mathbf{H}^T(D)$ of the hard decision $\mathbf{r}(D)$ of the received sequence and searching it for blocks of consecutive zeros. Because $\mathbf{H}^T(D)$ is partially orthogonal to error-free subsequences in $\mathbf{r}(D)$, these subsequences propagate blocks of consecutive zeros into $\mathbf{b}(D)$. This procedure results in an adaptive decoding algorithm, where the decoding effort is adaptively reduced for increasing SNR.

If the SD is applied to decode PCCs based on the mother code, a depuncturing is required, which is realized by inserting zero-reliability bits at the punctured positions in the received soft-bit sequence. The hard-decision $\mathbf{r}(D)$ of this depunctured sequence may consequently contain erroneous bits at the punctured positions, which occur additionally to the actual transmission errors. Thus even for error-free received sequences, the decoder has to correct errors at the punctured positions. These additional errors have no impact on the decoding performance, but they propagate to the syndrome sequence and also decrease the zero-state probability. This renders an reduced complexity decoding concept like [6], [7] or a low power implementation based on scarce state transitions ineffective.

A solution to this is to avoid the depuncturing by implementing the decoder based on the PCC, i.e. use the syndrome former $\tilde{\mathbf{H}}^T(D)$ of the PCC instead of the syndrome former $\mathbf{H}^T(D)$ of the mother code. The purpose of this paper is therefore to show that a trellis of $\tilde{\mathbf{H}}^T(D)$ can be constructed, which has the same complexity as the trellis of $\mathbf{H}^T(D)$. Furthermore a method for constructing this trellis is given. It is shown, that this construction is always possible for $(n_p - 1)/n_p$-rate PCCs given some reasonable preconditions.

The construction of a reduced complexity syndrome former trellis for arbitrary high-rate codes is not new. The authors of [8] propose a decoding algorithm for $(n-1)/n$ rate codes, termed YHM algorithm, based on the trellis of $\mathbf{H}^T(D)$. The basic idea is to form to a trellis module, which consists of multiple sections and doubled number of states, but with only two branches merging into each state. Their results were extended in [9] to arbitrary high-rate codes and shown to be equivalent to the method proposed in [10]. In [11] the authors present a simplification of the YHM method, where they show that the number of states can be reduced and the trellis structure can be simplified in some cases. Compared to the general syndrome trellis complexity reduction procedures presented in [8], [9], [10], [11], this paper is specially focused on the syndrome former trellis of PCCs.

This paper is organized as follows: In Section II-A to II-C the construction of the generator matrix $\tilde{\mathbf{G}}(D)$ of PCCs is reviewed. Further on, in section II-D, the impact of the special structure of $\tilde{\mathbf{G}}(D)$ on the structure of $\tilde{\mathbf{H}}^T(D)$ is discussed. Based on the structure of $\tilde{\mathbf{H}}^T(D)$, in Section III the trellis construction method is described. The presentation is clarified using three examples in Section IV and conclusions are finally drawn in Section V.

II. Syndrome former of punctured codes

This section starts with a brief review of the construction of the generator matrix for a punctured convolutional code[12]. After observing the special structure of the punctured generator matrix, the structural properties of the corresponding syndrome former are derived.

A. Original Encoder

The punctured code is derived from a binary $1/n$-rate code with constraint length ν. Let a canonical polynomial generator matrix of the code be the $(1 \times n)$-matrix given by

$$\mathbf{G}(D) = \left[G^{(1,1)}(D) \ \ldots \ G^{(1,n)}(D) \right], \tag{1}$$

where the generators are $G^{(1,i)}(D) = \sum_{k=0}^{\nu} g_k^{(1,i)} D^k$. We assume that $\mathbf{G}(D)$ is antipodal [13], i.e. $g_0^{(1,i)} = g_\nu^{(1,i)} = 1$.

Another representation of (1) can be achieved using the $(1 \times n)$-coefficient matrices $\mathbf{G}_k$:

$$\mathbf{G}(D) = \sum_{k=0}^{\nu} \mathbf{G}_k D^k. \tag{2}$$

Thus, if encoding is described in time domain as $\mathbf{v} = \mathbf{u}\mathbf{G}$, where $\mathbf{v}$ and $\mathbf{u}$ are binary row vectors, then $\mathbf{G}$ is a block toeplitz matrix where the rows contain shifted versions of

$$\mathbf{G} = \left[\mathbf{G}_0 \ldots \mathbf{G}_k \ldots \mathbf{G}_\nu \right]. \tag{3}$$

The generator matrix of the punctured $(n_p-1)/n_p$-rate code can be derived by first blocking $\mathbf{G}(D)$ by the factor (n_p-1) and then applying the puncturing pattern.

B. Blocking

The blocked encoder is obtained by combining (n_p-1) inputs of the original encoder. In D-domain this blocking can be described by replacing each entry of $\mathbf{G}(D)$ with its (n_p-1)-th polyphase component and then interleaving the columns of the resulting $(n_p-1)\times(n_p-1)n$-matrix to the (n_p-1)-th depth [12].

In time domain the same is achieved by first expanding the original matrix $\mathbf{G}$ to form a block toeplitz matrix

$$\begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_\nu & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_\nu & & \\ & & \ddots & & & & \\ & & & \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_\nu \end{bmatrix}. \tag{4}$$

Then the blocked generator matrix is found by extending (4) with a suitable number of all-zero columns on the right and grouping it into $(\hat{m}+1)$ submatrices $\hat{\mathbf{G}}_k$,

$$\hat{\mathbf{G}} = \left[\hat{\mathbf{G}}_0 \ldots \hat{\mathbf{G}}_k \ldots \hat{\mathbf{G}}_{\hat{m}} \right], \tag{5}$$

where the dimension of $\hat{\mathbf{G}}_k$ is $(n_p-1) \times (n_p-1)n$. The generator matrix of the blocked encoder is obtained by

$$\hat{\mathbf{G}}(D) = \sum_{k=0}^{\hat{m}} \hat{\mathbf{G}}_k D^k, \tag{6}$$

where $\hat{m}$ is called the memory of the blocked encoder. For the memory it holds $\hat{m} = \lceil \nu/(n_p-1) \rceil$ and the overall constraint length $\hat{\nu}$ of the blocked matrix is identical to that of the original matrix, $\hat{\nu} = \nu$.

The row degrees $\hat{\nu}_i$ of $\hat{\mathbf{G}}(D)$ are the Forney indices of the blocked code and it holds

$$\hat{\nu}_i = \begin{cases} \lfloor \nu/(n_p-1) \rfloor & \text{for } i = 1 \ldots \tau, \\ \lceil \nu/(n_p-1) \rceil & \text{for } i = \tau+1 \ldots n_p-1, \end{cases} \tag{7}$$

where

$$\tau = \begin{cases} 0 & \text{if } \nu \bmod (n_p-1) = 0, \\ (n_p-1) - \nu \bmod (n_p-1) & \text{else.} \end{cases} \tag{8}$$

In other words, the row degrees of $\hat{\mathbf{G}}(D)$ are ordered such that $\hat{\nu}_i = \lfloor \nu/(n_p-1) \rfloor$ for the first τ rows and $\hat{\nu}_i = \lceil \nu/(n_p-1) \rceil$ for the remaining rows and the blocked code is compact [12]. In the special case $\tau = 0$ all row degrees are equal to $\hat{m}$.

Now it is easy to see, that the matrix $\hat{\mathbf{G}}_0$ has a special structure – it is upper block triangular:

$$\hat{\mathbf{G}}_0 = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \ldots \\ & & \ddots & \ddots \\ & & & \mathbf{G}_0 \end{bmatrix} \tag{9}$$

For the matrices $\hat{\mathbf{G}}_{\hat{m}-1}$ and $\hat{\mathbf{G}}_{\hat{m}}$ we can also identify special structures. Let $\hat{\mathbf{G}}_k^{[i,j]}$ denote rows i to j of the matrix $\hat{\mathbf{G}}_k$, then it holds that

$$\hat{\mathbf{G}}_{\hat{m}-1}^{[1,\tau]} = \begin{bmatrix} \ldots & \mathbf{G}_\nu & & \\ & & \ddots & \\ & & \ldots & \mathbf{G}_\nu \end{bmatrix} \tag{10}$$

and

$$\hat{\mathbf{G}}_{\hat{m}}^{[\tau+1,n_p-1]} = \begin{bmatrix} \mathbf{G}_\nu & & & \\ & \ddots & & \mathbf{0} \\ \ldots & \mathbf{G}_\nu & & \end{bmatrix}, \tag{11}$$

where all entries on the right of the $\mathbf{G}_\nu$ are zero and $\hat{\mathbf{G}}_{\hat{m}}^{[\tau+1,n_p-1]}$ contains τn zero columns in the right part. Thus we can construct a lower block triangular matrix $\hat{\mathbf{A}}$,

$$\hat{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{G}}_{\hat{m}}^{[\tau+1,n_p-1]} \\ \\ \hat{\mathbf{G}}_{\hat{m}-1}^{[1,\tau]} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_\nu & & \\ & \ddots & \\ & \dots & \mathbf{G}_\nu \end{bmatrix}. \quad (12)$$

C. Puncturing

The puncturing pattern is specified by an $n \times (n_p - 1)$ matrix $\mathbf{P}$. The entry in the i-th row and j-th column $p^{(i,j)}$ is either 0 or 1, specifying that the corresponding j-th bit of the i-th output of $\mathbf{G}(D)$ is punctured or not, respectively. The number of 1s in $\mathbf{P}$ thus equals n_p. It can be assumed that there is at least one bit transmitted at each time step, so that there is no all-zero column in $\mathbf{P}$. Thus there is one column, denoted by κ, that contains two 1s, all other columns contain only a single 1. Formally, if we define $w(\mathbf{P}^{(j)})$ to be the hamming weight of the j-th column, then it holds that

$$w(\mathbf{P}^{(j)}) = \begin{cases} 2 & \text{for } j = \kappa \\ 1 & \text{for } 1 \leq j \leq (n_p - 1),\ j \neq \kappa. \end{cases} \quad (13)$$

Given $\hat{\mathbf{G}}(D)$ and $\mathbf{P}$, the puncturing can easily be accomplished by pruning each column $l = (j-1)n + i$ in each $\hat{\mathbf{G}}_k$ for which holds $p^{(i,j)} = 0$. This yields the coefficient matrices $\tilde{\mathbf{G}}_k$ of the punctured code and thus the $(n_p - 1) \times n_p$ generator matrix $\tilde{\mathbf{G}}(D)$ of the punctured code is given by

$$\tilde{\mathbf{G}}(D) = \sum_{k=0}^{\tilde{m}} \tilde{\mathbf{G}}_k D^k. \quad (14)$$

From this it follows, that the punctured versions of $\hat{\mathbf{G}}_0$ and $\hat{\mathbf{A}}$ also feature a special structure:

$$\tilde{\mathbf{G}}_0 = \begin{bmatrix} 1 & \times & \cdots & & & & & \\ & \ddots & \ddots & & & & & \\ & & 1 & \times & \cdots & & & \\ & & & 1 & 1 & \times & \cdots & \\ & & & & & 1 & \times & \\ & & & & & & \ddots & \ddots \\ & & & & & & & 1 \end{bmatrix} \quad (15)$$

and

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & & & & & & & \\ \ddots & \ddots & & & & & & \\ & \times & 1 & & & & & \\ & & \times & 1 & 1 & & & \\ & & \cdots & & \times & 1 & & \\ & & & & & \ddots & \ddots & \\ & & & & \cdots & & \times & 1 \end{bmatrix}, \quad (16)$$

i.e. $\tilde{\mathbf{G}}_0$ and $\tilde{\mathbf{A}}$ are upper and lower "quasi" triangular, respectively, each with their κ-th diagonal entry being $[1\ 1]$ and all other diagonal entries 1. Note that we have only one $[1\ 1]$ entry because we are restricted to $(n_p - 1)/n_p$ rates.

Given the assumptions from Sec. II-A and II-C, it is known that $\tilde{\mathbf{G}}$ is either catastrophic or canonical [13], depending on $\mathbf{P}$. In context of this paper, we assume that $\mathbf{P}$ has been selected such that $\tilde{\mathbf{G}}$ is canonical. Then for the row degrees $\tilde{\nu}_i$ and the memory $\tilde{m}$ of $\tilde{\mathbf{G}}$ it holds that,

$$\tilde{\nu}_i = \hat{\nu}_i,\ i = 1 \dots (n_p - 1) \text{ and } \tilde{m} = \hat{m}, \quad (17)$$

and consequently that no reduction of overall constraint length takes place,

$$\tilde{\nu} = \hat{\nu} = \nu. \quad (18)$$

From the structures of (15) and (16) it can be noticed that in this case the punctured code is compact, too.

D. Properties of syndrome former

The syndrome former of the punctured code has n_p inputs and 1 output. It is well known [3], that an adjoint obvious realization of the syndrome former can be done with the same number of memory elements as the corresponding canonical generator. Thus considering (18), a canonical (delay-free) syndrome former of the punctured code can be written as an $(n_p \times 1)$-matrix:

$$\tilde{\mathbf{H}}^T(D) = \left[\tilde{H}^{(1,1)}(D) \dots \tilde{H}^{(n_p,1)}(D)\right]^T = \sum_{k=0}^{\nu} \tilde{\mathbf{H}}_k^T D^k, \quad (19)$$

where the polynomials are written as

$$\tilde{H}^{(i,1)}(D) = \tilde{h}_0^{(i,1)} + \ldots + \tilde{h}_\nu^{(i,1)} D^\nu. \quad (20)$$

In the following we will see, how the special structure of $\tilde{\mathbf{G}}(D)$ affects the structure of the syndrome former. By definition the syndrome former is orthogonal to the considered code, which can be written as

$$\tilde{\mathbf{G}}(D)\tilde{\mathbf{H}}^T(D) = \left(\sum_{k=0}^{\tilde{m}} \tilde{\mathbf{G}}_k D^k\right)\left(\sum_{k=0}^{\nu} \tilde{\mathbf{H}}_k^T D^k\right) = \mathbf{0}. \quad (21)$$

The second term can be rewritten as

$$\left(\sum_{k=0}^{\tilde{m}} \tilde{\mathbf{G}}_k D^k\right)\left(\sum_{k=0}^{\nu} \tilde{\mathbf{H}}_k^T D^k\right) = \sum_{k=0}^{\nu+\tilde{m}} \left(\sum_{\substack{i,j \\ i+j=k}} \tilde{\mathbf{G}}_i \tilde{\mathbf{H}}_j^T\right) D^k \quad (22)$$

and it follows that each term must yield

$$\sum_{\substack{i,j \\ i+j=k}} \tilde{\mathbf{G}}_i \tilde{\mathbf{H}}_j^T = \mathbf{0},\ k = 0 \dots \nu + \tilde{m}. \quad (23)$$

From (23) we can get constraints on the coefficient matrices with lowest and highest degree, $\tilde{\mathbf{H}}_0^T$ and $\tilde{\mathbf{H}}_\nu^T$:

$$\tilde{\mathbf{G}}_0 \tilde{\mathbf{H}}_0^T = \mathbf{0} \quad (24)$$
$$\tilde{\mathbf{G}}_{\tilde{m}}^{[\tau+1,n_p-1]} \tilde{\mathbf{H}}_\nu^T = \mathbf{0} \quad (25)$$
$$\tilde{\mathbf{G}}_{\tilde{m}-1}^{[1,\tau]} \tilde{\mathbf{H}}_\nu^T = \mathbf{0} \quad (26)$$

A shorter notation of (25) and (26) can be found using (16):

$$\tilde{\mathbf{A}}\tilde{\mathbf{H}}_\nu^T = \mathbf{0}. \quad (27)$$

As we know that $\tilde{\mathbf{G}}_0$ and $\tilde{\mathbf{A}}$ have upper and lower triangular-like structures, respectively,

- $\tilde{\mathbf{G}}_0\tilde{\mathbf{H}}_0^T = \mathbf{0}$ can be solved in general for $\tilde{h}_0^{(i,1)}$, $i = \kappa \dots n_p$ by backward substitution, going from the n_p-th row up to the κ-th row and
- $\tilde{\mathbf{A}}\tilde{\mathbf{H}}_\nu^T = \mathbf{0}$ can be solved in general for $\tilde{h}_\nu^{(i,1)}$, $i = 1 \dots \kappa + 1$ by forward substitution, going from the first row down to the $(\kappa + 1)$-th row.

Then for (24) the solution can readily be found as

$$\tilde{h}_0^{(i,1)} = 0,\ \kappa + 1 < i \leq n_p, \quad (28)$$

$$\tilde{h}_0^{(i,1)} = 1,\ \kappa \leq i \leq \kappa + 1. \quad (29)$$

While (28) is obvious, we should note that (29) holds because otherwise it would mean $\tilde{h}_0^{(i,1)} = 0$ for all i, which would actually contradict the fact that $\tilde{\mathbf{H}}^T(D)$ is delay-free. For (27) the solution can be derived in the same manner, i.e.

$$\tilde{h}_\nu^{(i,1)} = 0,\ 1 \leq i < \kappa, \quad (30)$$

$$\tilde{h}_\nu^{(i,1)} = 1,\ \kappa \leq i \leq \kappa + 1. \quad (31)$$

Thus from (28)–(31) we finally have the following conclusion about the syndrome former structure of the PCC:

Conclusion: Assume an $(n_p - 1)/n_p$-rate PCC with overall constraint length ν, derived from a $1/n$ rate mother code with constraint length ν and represented by a canonical, antipodal generator matrix, and a puncturing pattern for which holds $w(\mathbf{P}^{(j)}) > 0$. Then for the adjoint obvious realization of the syndrome former, represented by the polynomials $\tilde{H}^{(i,1)}(D)$, $i = 1 \dots n_p$, it holds for

- $1 \leq i < \kappa$ that $\deg(\tilde{H}^{(i,1)}(D)) < \nu$, i.e. the first $(\kappa - 1)$ inputs of the syndrome former are not connected to the highest-order memory element,
- $\kappa \leq i \leq \kappa + 1$ that $\deg(\tilde{H}^{(i,1)}(D)) = \nu$ and $\tilde{H}^{(i,1)}(D)$ is delayfree, i.e. the inputs κ and $\kappa + 1$ of the syndrome former are connected to both the highest-order memory and the output,
- $\kappa + 1 < i \leq n_p$ that $\tilde{H}^{(i,1)}(D)$ has a factor D, i.e. the last $n_p - 1 - \kappa$ inputs are not connected to the ouput.

III. Syndrome former trellis complexity and construction

In this section the conclusion of the previous section is used to derive a trellis representation of $\tilde{\mathbf{H}}^T(D)$ which has the same complexity as the trellis of the mother code $\mathbf{G}(D)$.

We will refer to the repeating part of the trellis, which corresponds to n_p code bits and $(n_p - 1)$ information bits, as the trellis module.

A. Trellis complexity of the encoder

The trellis complexity of the encoder trellis module of an k/n rate code with constraint length ν can be defined as the product of the number of code bits n per edge, the number of edges 2^k entering into each state and the number of states 2^ν, per number of information bits k [14]:

$$n/k\ 2^{\nu+k} \quad (32)$$

For punctured $(n_p - 1)/n_p$-rate codes the complexity is significantly reduced and can be given as [14]

$$n_p/(n_p - 1)\ 2^{\nu+1}. \quad (33)$$

This is achieved, because each trellis module corresponding to $(n_p - 1)$ information bits is divided into $(n_p - 1)$ sections. Each section has 2 edges entering into each of the 2^ν states and $(n_p - 2)$ sections have 1 bit per edge and one section has 2 bit per edge. In practice it is of course not necessary to explicitly construct this trellis module, because the Viterbi decoder can directly consider the puncturing pattern in the metric computation.

B. Trellis complexity of syndrome former and construction

The trellis complexity of the syndrome former of a k/n-rate code with constraint length ν can be defined in the same manner, as the product of the number of error bits n per edge, the number of edges 2^n entering into each state and the number of states 2^ν, per number of information bits k. Additionally, it has to be considered that for Schalkwijk's syndrome decoding algorithm, each module only considers the branches corresponding to one of the 2^{n-k} syndrome symbols. Thus we get the same trellis complexity as for the encoder:

$$n/k\ 2^{\nu+n}/2^{n-k} = n/k\ 2^{\nu+k} \quad (34)$$

Note that the number of bits per edge is the number of output bits for the encoder, while it is the number of inputs for the syndrome former. This is because for the encoder the branch metric depends on the output, while for the syndrome former it depends on the input. Also note that the syndrome former is realized in adjoint obvious form, while the encoder is realized in obvious form [3].

If syndrome decoding is realized based on the trellis of the mother code the same decoding complexity as for Viterbi decoding is achieved (i.e. the trellis complexity is given as in (33)). However it can be advantageous to use the syndrome former of the punctured code $\tilde{\mathbf{H}}(D)$ for decoding, as described in Section I. In this case, the complexity is in general given by (34), i.e. it is increased by a factor $2^{k-1} = 2^{n_p-2}$ compared to (33).

A trellis module of $\tilde{\mathbf{H}}(D)$ with the same complexity as the punctured encoder trellis module (complexity as in (33)) can however be obtained by exploiting the structural properties of $\tilde{\mathbf{H}}(D)$ and by separating one trellis module into multiple sections with each section having 2^ν states. More specifically, the trellis module consists of $n_p - 1$ sections with 2^ν states and 2 branches merging into each state; the branches in $n_p - 2$ sections are labeled with one input (error) bit, while one section has branches labeled with 2 input (error) bits. We note that the idea of shifting single bits into the syndrome former instead of n_p bits generally requires $2^{\nu+1}$ states[8], because it is necessary to keep track of the memory content and of the output. In this case however, 2^ν states are sufficient, because $n_p - 2$ inputs are either not connected to the highest order memory element, or not connected to the output (cf. Section II-D).

The following construction procedure clarifies the foregoing description: The trellis module is divided into $n_p - 1$ sections, so that there has to be one intial and one final state vector, and $n_p - 2$ intermediate state vectors. The state vetors are denoted by the binary vectors $\boldsymbol{\xi}^{(i)} = [\xi_\nu^{(i)} \ldots \xi_1^{(i)}]$, $i = 0 \ldots (n_p - 1)$, where $\boldsymbol{\xi}^{(0)}$ and $\boldsymbol{\xi}^{(n_p-1)}$ are initial and final state vector, respectively. The input error bits and branch labels are denoted by $e^{(1)} \ldots e^{(n_p)}$. Then we can state the following construction rules:

1) For $1 \leq i < \kappa$ the pairs of successor states $\boldsymbol{\xi}^{(i)}$ of each state $\boldsymbol{\xi}^{(i-1)}$ are found by evaluating $\boldsymbol{\xi}^{(i)} = \boldsymbol{\xi}^{(i-1)} \oplus e^{(i)} \cdot [\tilde{h}_{\nu-1}^{(i,1)} \ldots \tilde{h}_0^{(i,1)}]$.
2) For $i = \kappa$
   a) the output b associated with the trellis module is generated as $b = \xi_1^{(i-1)} \oplus e^{(i)} \oplus e^{(i+1)}$,
   b) and the four successor states $\boldsymbol{\xi}^{(i)}$ of each $\boldsymbol{\xi}^{(i-1)}$ are found by evaluating

$$\begin{aligned} \boldsymbol{\xi}^{(i)} &= [0\ \xi_\nu^{(i-1)} \ldots \xi_2^{(i-1)}] \\ &\oplus e^{(i)} \cdot [\tilde{h}_\nu^{(i,1)} \ldots \tilde{h}_1^{(i,1)}] \\ &\oplus e^{(i+1)} \cdot [\tilde{h}_\nu^{(i+1,1)} \ldots \tilde{h}_1^{(i+1,1)}] \end{aligned}$$

3) For $\kappa < i < n_p$ the pairs of successor states $\boldsymbol{\xi}^{(i)}$ of each $\boldsymbol{\xi}^{(i-1)}$ are found by evaluating $\boldsymbol{\xi}^{(i)} = \boldsymbol{\xi}^{(i-1)} \oplus e^{(i+1)} \cdot [\tilde{h}_\nu^{(i+1,1)} \ldots \tilde{h}_1^{(i+1,1)}]$.

Note that for the first $(\kappa - 1)$ and last $(n_p - \kappa - 1)$ sections, the coefficients $\tilde{h}_\nu^{(i,1)}$ and $\tilde{h}_0^{(i+1,1)}$, respectively, are not used for the trellis construction, because for these sections it holds $\tilde{h}_\nu^{(i,1)} = 0$ and $\tilde{h}_0^{(i+1,1)} = 0$, respectively.

It can be verified, that the resulting trellis complexity is

$$n_p/(n_p - 1)2^{\nu+1}, \tag{35}$$

and therefore identical to the encoder trellis complexity of the mother code.

Regarding a decoder implementation, it can be observed, that only section κ depends on b, while the other $n_p - 2$ sections are independant of the current syndrome symbol. Thus only one of the $n_p - 1$ sections is time varying during the decoding process.

IV. Examples

Three examples are presented to clarify the meaning of the variables in the foregoing description, and illustrate the resulting syndrome former trellis.

Example 1: The first code is a constraint length $\nu = 2$ punctured 2/3-rate code derived from a 1/2-rate mother code. The encoder is $\mathbf{G}(D) = [D^2 + 1,\ D^2 + D + 1]$ and the corresponding coefficient matrices are given as $\mathbf{G} = [\mathbf{G}_0\ \mathbf{G}_1\ \mathbf{G}_2] = [(1\ 1)\ (0\ 1)\ (1\ 1)]$.

With $n_p = 3$ from Section II-B we get

- $\hat{m} = \lceil 2/(3-1) \rceil = 1$,
- $\tau = 0$ from (8),
- $\hat{\nu}_1 = \hat{\nu}_2 = 1$ from (7) and

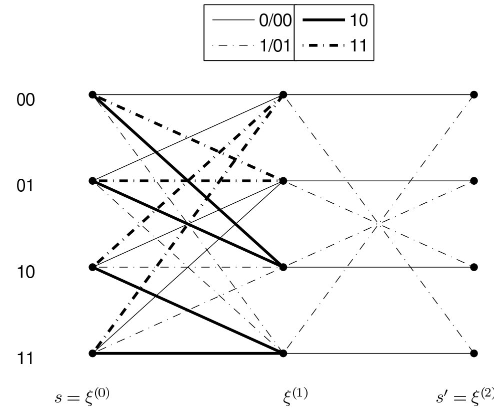

Fig. 1. Syndrome former trellis of punctured $2/3$-rate code.

- $\hat{\mathbf{G}} = \left[\hat{\mathbf{G}}_0\ \hat{\mathbf{G}}_1\right] = \left[\begin{pmatrix}1 & 1 & 0 & 1\\ 0 & 0 & 1 & 1\end{pmatrix} \begin{pmatrix}1 & 1 & 0 & 0\\ 0 & 1 & 1 & 1\end{pmatrix}\right]$.

Let the puncturing pattern be given as $\mathbf{P} = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$, then it follows that $\kappa = 1$, and by pruning column $l = 3$ from $\hat{\mathbf{G}}_0$ and $\hat{\mathbf{G}}_1$, the coefficient matrices of the PCC are obtained as

$$\tilde{\mathbf{G}} = \left[\tilde{\mathbf{G}}_0\ \tilde{\mathbf{G}}_1\right] = \left[\begin{pmatrix}1 & 1 & 1\\ 0 & 0 & 1\end{pmatrix} \begin{pmatrix}1 & 1 & 0\\ 0 & 1 & 1\end{pmatrix}\right]$$

and the corresponding polynomial encoder matrix as

$$\tilde{\mathbf{G}}(D) = \begin{bmatrix}1+D & 1+D & 1\\ 0 & D & 1+D\end{bmatrix}.$$

From Section II-C it also follows that

- $\tilde{\mathbf{A}} = \tilde{\mathbf{G}}_1$,
- $\tilde{\nu} = 2$ and $\tilde{m} = 1$ from (18) and
- $\tilde{\nu}_1 = \tilde{\nu}_2 = 1$ from (17).

The syndrome former of the PCC can be computed using the invariant factor decomposition [15] of $\tilde{\mathbf{G}}(D)$ and is given by

$$\tilde{\mathbf{H}}^T(D) = \begin{bmatrix}1\\1\\0\end{bmatrix} + \begin{bmatrix}1\\0\\1\end{bmatrix} D + \begin{bmatrix}1\\1\\1\end{bmatrix} D^2.$$

It can be seen that the foregoing conclusion holds, i.e. that $\tilde{h}_0^{(1,1)} = \tilde{h}_0^{(2,1)} = 1$ and $\tilde{h}_0^{(3,1)} = 0$ and $\tilde{h}_2^{(1,1)} = \tilde{h}_2^{(2,1)} = 1$.

The trellis complexity of the conventional adjoint-obvious realization is given by (34) $n_p/(n_p - 1)\ 2^{\nu + n_p - 1} = 3/2\ 2^4$. The trellis constructed using the proposed method is shown in Fig. 1. Its complexity is given by (35) as $n_p/(n_p - 1)\ 2^{\nu+1} = 3/2\ 2^3$, and thus reduced by a factor 2^1.

Example 2: The puncturing pattern $\mathbf{P} = \begin{bmatrix}0 & 0 & 1 & 0 & 1\\ 1 & 1 & 1 & 1 & 0\end{bmatrix}$ is applied to the 1/2-rate mother code $\mathbf{G}(D) = [D^3 + D^2 + 1,\ D^3 + D^2 + D + 1]$ with $\nu = 3$. It holds $\tau = 2$ and $\kappa = 3$ and the syndrome former of the resulting $5/6$-PCC is given

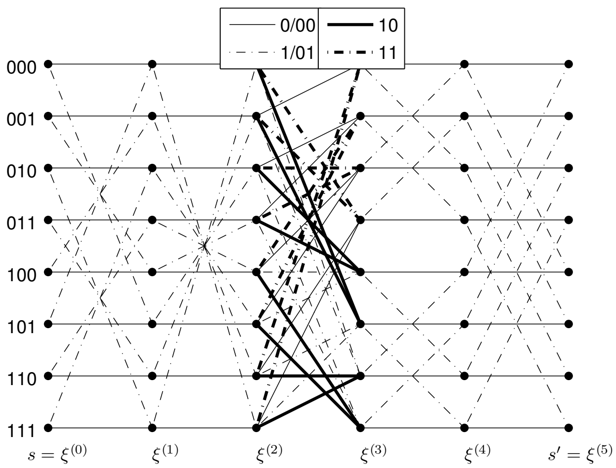

Fig. 2. Syndrome former trellis of punctured $5/6$-rate code.

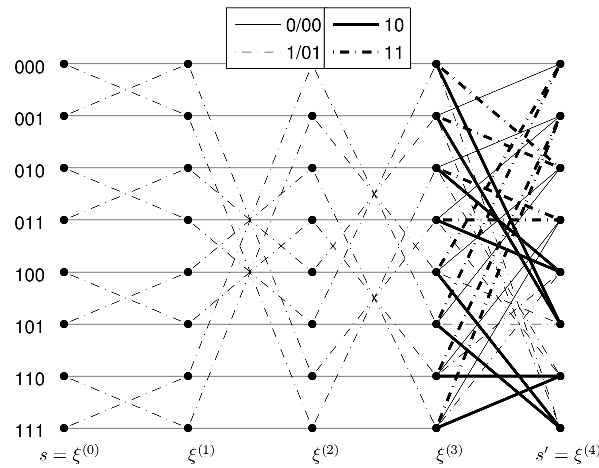

Fig. 3. Syndrome former trellis of punctured $4/5$-rate code.

as

$$\tilde{\mathbf{H}}^T(D) = \begin{bmatrix}1\\1\\1\\1\\0\\0\end{bmatrix} + \begin{bmatrix}0\\1\\0\\1\\0\\0\end{bmatrix} D + \begin{bmatrix}1\\1\\1\\0\\1\\0\end{bmatrix} D^2 + \begin{bmatrix}0\\0\\1\\1\\0\\1\end{bmatrix} D^3.$$

The corresponding trellis module is shown in Fig. 2 and its complexity is given by (35) as $n_p/(n_p-1)\ 2^{\nu+1} = 6/5\ 2^4$.

Example 3: The puncturing pattern $\mathbf{P} = \begin{bmatrix}0 & 0 & 0 & 0\\0 & 1 & 1 & 1\\1 & 0 & 0 & 1\end{bmatrix}$ is applied to the $1/3$-rate mother code $\mathbf{G}(D) = [D^3 + D + 1,\ D^3 + D^2 + 1,\ D^3 + D^2 + D + 1]$ with $\nu = 3$. It holds $\tau = 1$ and $\kappa = 4$ and the syndrome former of the resulting $4/5$-PCC is given as

$$\tilde{\mathbf{H}}^T(D) = \begin{bmatrix}1\\0\\1\\1\\1\end{bmatrix} + \begin{bmatrix}0\\1\\0\\1\\1\end{bmatrix} D + \begin{bmatrix}0\\1\\1\\1\\0\end{bmatrix} D^2 + \begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix} D^3.$$

The corresponding trellis module is shown in Fig. 3 and its complexity is given by (35) as $n_p/(n_p-1)\ 2^{\nu+1} = 5/4\ 2^4$.

V. Conclusions

It has been shown that for the syndrome former of $(n_p - 1)/n_p$-rate PCCs, which are commonly found in applications, a trellis with same complexity as the encoder trellis of the mother code can be constructed. Explicitly constructing the trellis offers advantages for adaptive complexity decoding, and for low state transition implementations based on the syndrome decoding principle. An algorithm for constructing the trellis has been stated. Three examples have been used to clarify the presentation.

References

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

[2] J. Schalkwijk, A. Vinck, and K. Post, "Syndrome decoding of binary-rate n/k-convolutional codes," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 553–562, Sep 1978.

[3] G. Forney Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans on Information Theory*, vol. 16, no. 6, pp. 720–738, Nov. 1970.

[4] L. Lee, D. Tait, P. Farrell, and P. Leung, "Novel scarce-state-transition syndrome-former error-trellis decoding of (n, n-1) convolutional codes," in *IEEE International Symposium on Information Theory*, 1995, p. 219.

[5] M. Ariel and J. Snyders, "Soft syndrome decoding of binary convolutional codes," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 288 –297, feb/mar/apr 1995.

[6] J. Geldmacher, K. Hueske, and J. Goetze, "An adaptive and complexity reduced decoding algorithm for convolutional codes and its application to digital broadcasting systems," in *Proceedings of the International Conference on Ultra Modern Telecommunications (ICUMT2009)*, St. Petersburg, Russia, Oct. 2009.

[7] J. Geldmacher, K. Hueske, S. Bialas, and J. Goetze, "Adaptive low complexity map decoding for turbo equalization," in *Proceedings of 6th International Symposium on Turbo Codes & Iterative Information Processing (ISTC2010)*, Brest, France, 2010.

[8] T. Yamada, H. Harashima, and H. Miyakawa, "A new maximum likelihood decoding of high rate convolutional codes using a trellis," *Electronics and Communications in Japan*, vol. J66-A, no. 7, pp. 611–616, July 1983.

[9] M. Tajima, K. Okino, and T. Miyagoshi, "Minimal code(error)-trellis module construction for rate-k/n convolutional codes: Extension of yamada-harashima-miyakawas construction," *IEICE Transactions on Fundamentals*, vol. E90-A, no. 11, pp. 2629–2634, 2007.

[10] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1663–1666, Sept. 1994.

[11] P. CharnKeitKong, H. Imai, and K. Yamaguchi, "On classes of rate k/(k+1) convolutional codes and their decoding techniques," *IEEE Trans. on Information Theory*, vol. 42, no. 6, pp. 2181–2193, 1996.

[12] R. McEliece, "The algebraic theory of convolutional codes," *Handbook of Coding Theory*, vol. I, pp. 1065–1138, 1998.

[13] K. Hole, "Rate k/(k+1) punctured convolutional encoders," *IEEE Transactions on Information Theory*, vol. 37, no. 3, pp. 653–655, May 1991.

[14] R. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1855–1864, Nov. 1996.

[15] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. IEEE press, 1999.