Searching for high-rate convolutional codes via binary syndrome trellises

Hug, Florian; Bocharova, Irina; Johannesson, Rolf; Kudryashov, Boris

Published in:
[Host publication title missing]

DOI:
10.1109/ISIT.2009.5205916

2009

Citation for published version (APA):
Hug, F., Bocharova, I., Johannesson, R., & Kudryashov, B. (2009). Searching for high-rate convolutional codes via binary syndrome trellises. In *[Host publication title missing]* (pp. 1358-1362) https://doi.org/10.1109/ISIT.2009.5205916

Total number of authors:
4

ISIT 2009, Seoul, Korea, June 28 - July 3, 2009

Searching for High-Rate Convolutional Codes via Binary Syndrome Trellises

Florian Hug[1], Irina E. Bocharova[2], Rolf Johannesson[1], and Boris D. Kudryashov[2]

[1] Dept. of Electrical and Information Technology,
Lund University
P. O. Box 118, SE-22100 Lund, Sweden
Email: {florian, rolf}@eit.lth.se

[2] Dept. of Information Systems
St. Petersburg Univ. of Information Technologies,
Mechanics and Optics
St. Petersburg 197101, Russia
Email: {irina, boris}@eit.lth.se

***Abstract*—Rate $R=(c-1)/c$ convolutional codes of constraint length ν can be represented by conventional syndrome trellises with a *state complexity* of $s=\nu$ or by *binary* syndrome trellises with a state complexity of $s=\nu$ or $s=\nu+1$, which corresponds to at most 2^s states at each trellis level. It is shown that if the parity-check polynomials fulfill certain conditions, there exist binary syndrome trellises with optimum state complexity $s=\nu$.**

The BEAST is modified to handle parity-check matrices and used to generate code tables for optimum free distance rate $R=(c-1)/c$, $c=3,4,5$, convolutional codes for conventional syndrome trellises and binary syndrome trellises with optimum state complexity. These results show that the loss in distance properties due to the optimum state complexity restriction for binary trellises is typically negligible.

I. INTRODUCTION

High-rate convolutional codes are important for many applications due to the combination of a modest rate loss and the existence of efficient maximum-likelihood (ML) decoding algorithms.

If we ignore the so-called start-up phase, every level of the conventional trellis of such a rate $R=b/c$ convolutional code with overall constraint length ν consists of 2^ν states, with 2^b branches arriving at and leaving each state. In [1] the *state complexity* s is called the most widely accepted measure of trellis complexity. The state complexity is given as the maximum value of the logarithm of the number of states at any level of the considered trellis. For a rate $R=b/c$ convolutional code with overall constraint length ν, the state complexity has to be at least greater than or equal to ν, that is,

$$s=\max_i\{\log_2|V_i|\}\geq\nu \tag{1}$$

where $|V_i|$ denotes the number of states at level i in the trellis.

The ML decoding complexity, *e.g.,* of the Viterbi algorithm [2], depends also on the number of branches arriving at and leaving each state, that is, 2^b in the conventional trellis of a rate $R=b/c$ convolutional code. Traditionally this number has been reduced significantly by introducing puncturing which leads to *binary* trellises with only two branches arriving at and leaving each node [3].

Since Paaske [4] reported on early searches for high-rate convolutional codes using the parity-check matrices, a series of papers reporting on various search techniques for good codes has been published [5]–[9]. In this paper we will report the results of a search for high-rate $R=(c-1)/c$, $c=3,4,5$ convolutional codes with overall constraint length ν using the syndrome trellis as proposed in [10]. As an alternative to puncturing, we will consider the binary syndrome trellis representation. In general, this comes at the cost of increasing the state complexity from $s=\nu$ to $s=\nu+1$, that is, the number of states at each level of the trellis doubles [11].

In Section II, we show that a binary syndrome trellis can be realized with state complexity $s=\nu$, that is, at most 2^ν different states, if the parity-check polynomials fulfill certain conditions. The BEAST is modified in Section III to handle parity-check matrices. In Section IV, we present tables for optimum free distance rate $R=(c-1)/c$, $c=3,4,5$, convolutional codes for conventional syndrome trellises as well as for optimum state complexity binary syndrome trellises.

II. COMPLEXITY OF THE SYNDROME TRELLIS

Consider a rate $R=(c-1)/c$, $c\geq 2$, convolutional code with parity-check matrix

$$H(D)=(h_1(D)\ h_2(D)\ \ldots\ h_c(D)) \tag{2}$$

where $h_i(D)$, $i=1,2,\ldots,c$, denotes the ith parity-check polynomial. Furthermore, let $\text{del}(h_i(D))$ and $\deg(h_i(D))$ denote the delay and the degree of $h_i(D)$, respectively. The overall constraint length ν of this parity-check matrix is given by

$$\nu=\max_i\{\deg(h_i(D))\}\,.$$

For the parity-check polynomials $h_i(D)$, $i=1,\ldots,c$, in (2), we have

$$h_i(D)\ =\ h_i^{(0)}+h_i^{(1)}D+\ldots+h_i^{(\nu)}D^\nu.$$

Then the parity-check matrix $H(D)$ can be represented by the semi-infinite matrix H consisting of $1\times c$ sub-matrices H_l, $l=0,1,\ldots,c$, where

$$H_l\ =\ \left(h_i^{(l)}\right)_{(1\times c)},\quad l=0,1,\ldots,\nu.$$

For example, for $\nu=2$, we have the matrix H given in (3). Following [10], a syndrome trellis can be constructed by connecting the identically full syndrome trellis modules corresponding to the parity-check matrix module $\widetilde{H}$ [11] given by (4).

978-1-4244-4313-0/09/$25.00 ©2009 IEEE

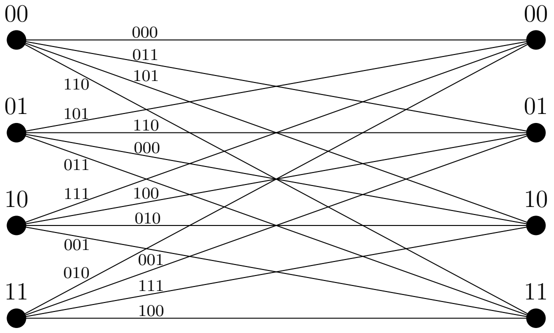

Fig. 1. Syndrome trellis module of the rate $R = 2/3$ convolutional code $H(D) = (1 + D^2 \;\; 1 + D \;\; 1 + D + D^2)$ with 4 states.

In case of a rate $R = (c-1)/c$ convolutional code, every column in the matrix module $\widetilde{H}$ corresponds to one of the c parity-check polynomials $h_i(D)$, $i = 1, 2, \ldots, c$, written in binary notation starting from the top.

$$H = \begin{pmatrix} H_0 & & & \\ H_1 & H_0 & & \\ H_2 & H_1 & H_0 & \\ & H_2 & H_1 & \ddots \\ & & H_2 & \ddots \\ & & & \ddots \end{pmatrix}. \tag{3}$$

$$\widetilde{H} = \begin{pmatrix} H_0 \\ \vdots \\ H_\nu \end{pmatrix}. \tag{4}$$

Every level (after the startup-phase) of the syndrome trellis of a rate $R = (c-1)/c$ convolutional code, when sectionalized to c bits per branch, consists of 2^ν different states with 2^{c-1} branches arriving at and leaving each state. For the rate $R = 2/3$ convolutional code with parity-check matrix $H(D) = (1 + D^2 \;\; 1 + D \;\; 1 + D + D^2)$ the corresponding conventional syndrome trellis module is illustrated in Fig. 1.

A *binary* syndrome trellis is specified using only two branches arriving at and leaving each state. This simplification, however, comes at the cost of $c-1$ additional intermediate layers in each trellis module, where the $c-1$ additional layers may consist of as many as $2^{\nu+1}$ different states. For the same convolutional code as before, the binary syndrome trellis module is depicted in Fig. 2.

Having a closer look at the binary syndrome trellis module, we notice that for this convolutional code it is possible to find an *equivalent* convolutional code [12] whose maximum number of states at each intermediate layer does not exceed 2^ν. Reordering the parity-check polynomials, we obtain the equivalent parity-check matrix $H_{\text{eq}}(D) = (1 + D \;\; 1 + D^2 \;\; 1 + D + D^2)$. Sectionalizing its binary syndrome trellis module, the number of intermediate layers can be reduced by one while the number of states at each intermediate layer decreases to 2^ν, which is illustrated for the given code in Fig. 3.

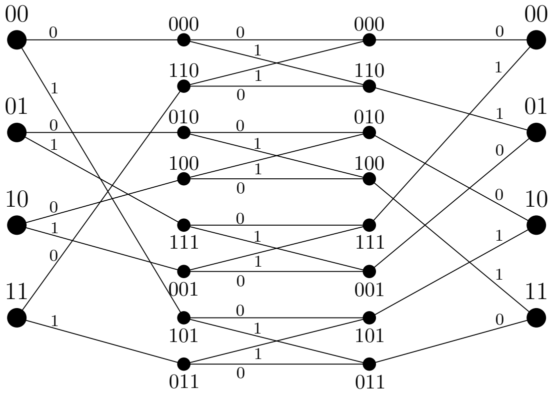

Fig. 2. Binary syndrome trellis module of the rate $R = 2/3$ convolutional code $H(D) = (1 + D^2 \;\; 1 + D \;\; 1 + D + D^2)$ with 4 to 8 states.

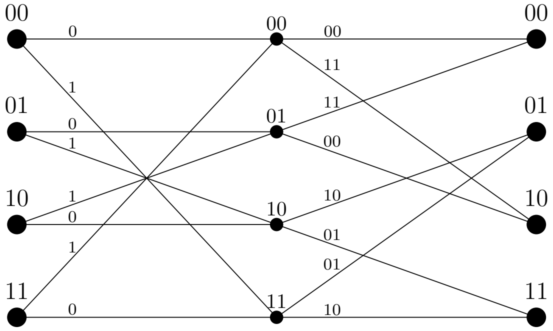

Fig. 3. Sectionalized binary syndrome trellis module of the rate $R = 2/3$ convolutional code $H_{\text{eq}}(D) = (1 + D \;\; 1 + D^2 \;\; 1 + D + D^2)$ with 4 states.

In the following we will give conditions which determine whether it is possible to decrease the number of states at the intermediate layers of the binary syndrome trellis. We introduce the abbreviation $\text{dgl}(p(D))$ for the difference between the degree and the delay of a polynomial $p(D)$, that is,

$$\text{dgl}(p(D)) = \deg(p(D)) - \text{del}(p(D)). \tag{5}$$

Every parity-check polynomial $h_i(D)$, $i = 1, \ldots, c$, belongs to at least one of the sets, $\mathcal{H}_I$, $\mathcal{H}_{II}$, and $\mathcal{H}_{III}$, where

$$\begin{aligned} \mathcal{H}_I &= \{h_i(D), i = 1, 2, \ldots, c \mid \deg(h_i(D)) < \nu\} \\ \mathcal{H}_{II} &= \{h_i(D), i = 1, 2, \ldots, c \mid \text{dgl}(h_i(D)) = \nu\} \\ \mathcal{H}_{III} &= \{h_i(D), i = 1, 2, \ldots, c \mid \text{del}(h_i(D)) > 0\}. \end{aligned} \tag{6}$$

If a parity-check polynomial fulfills the conditions for two sets, it will be assigned to an arbitrary one.

By reordering the parity-check polynomials in the matrix module $\widetilde{H}$ it is possible to obtain an equivalent convolutional code, whose matrix module $\widetilde{H}_{\text{eq}}$ consists of the columns ordered such that the first columns in $\widetilde{H}_{\text{eq}}$ belong to $\mathcal{H}_I$, followed by the columns of $\mathcal{H}_{II}$, and finally by those of $\mathcal{H}_{III}$, as illustrated in Fig. 4.

Next, we let the *span* [13] for each row in (3) denote the interval starting with its first and ending with its last nonzero

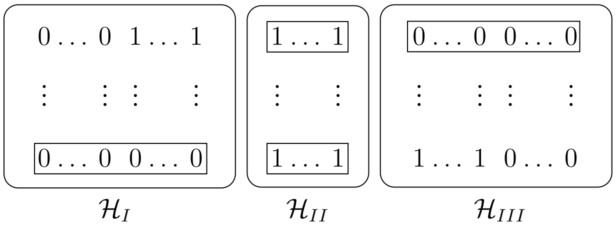

Fig. 4. A conventional matrix module with columns/parity-check polynomials ordered according to $\mathcal{H}_I$, $\mathcal{H}_{II}$, and $\mathcal{H}_{III}$.

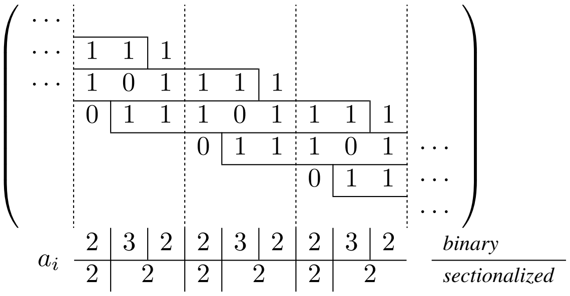

Fig. 5. Number of active rows a_i for the rate $R = 2/3$ convolutional code $H_{\text{eq}} = (1 + D \;\; 1 + D^2 \;\; 1 + D + D^2)$, before (upper row) and after sectionalizing (lower row). The boxes surround the active part of each row.

value. A certain column of a row is considered to be *active* if it lies within the span, but is not the last column of the span [1]. Directly related, the number of states $|V_i|$ at the ith level of the trellis is given by

$$|V_i| = 2^{a_i} \tag{7}$$

where a_i denotes the number of active rows in the ith column, $i = 1, 2, \ldots, c$, of a parity-check matrix. As the trellis is constructed by connecting the identically full syndrome trellis modules, the state complexity s is fully determined by a single matrix module. By combining (1) and (7) we obtain

$$s = \max\{a_i\}\,, \quad i = 1, 2, \ldots, c. \tag{8}$$

Whether a row in (3) is considered to be active at a certain position is obviously determined by the matrix module $\widetilde{H}$. For a rate $R = (c-1)/c$ convolutional code, the last nonzero value of a row is determined by the first row of the matrix module, whereas the first nonzero value of a row is determined by the last row of the matrix module. In other words, in each matrix module, its first row ends being active and its last row starts being active (*cf.* Fig. 5).

Having a closer look at the binary syndrome trellis module, every valid path, that is, every valid partial codeword, corresponds to a linear combination of columns of the parity-check matrix module. Every valid codeword $\boldsymbol{v}$ has to fulfill the *zero-constraint* $\boldsymbol{v}H^T = \boldsymbol{0}$. Consider an arbitrary row j, $j = 1, 2, \ldots$. The span of this row ends within the first row of a certain matrix module $\widetilde{H}$. To fulfill the zero-constraint, that is the linear combination of columns determined by the codeword, the syndrome bit j must be zero already after adding the first jc columns.

Next we take a closer look at the sets $\mathcal{H}_I$, $\mathcal{H}_{II}$, and $\mathcal{H}_{III}$. To store every possible linear combination of parity-check polynomials from $\mathcal{H}_I$ we need at most 2^ν memory elements, while the same holds for every linear combination of parity-check polynomials from $\mathcal{H}_{III}$. For the set $\mathcal{H}_{II}$, however, we will distinguish four different cases:

- $\mathcal{H}_{II}$ is *empty* and thus the span of every row of the matrix H ends with a column of $\mathcal{H}_I$. Having fulfilled the zero-constraint with a linear combination of parity-check polynomials from $\mathcal{H}_I$, we have used at most ν memory elements. Due to the zero-constraint the first partial syndrome bit is zero and will stay so. Continuously adding parity-check polynomials from $\mathcal{H}_{III}$, we update at most the last ν bits of the current partial syndrome. Consequently there is no need for more than ν memory elements at any time.
- $\mathcal{H}_{II}$ contains *one* parity-check polynomial and the span of the first row of the matrix module ends with the single column in $\mathcal{H}_{II}$. If, after the linear combination of parity-check polynomials from $\mathcal{H}_I$, the first bit is one, the parity-check polynomial in $\mathcal{H}_{II}$ will be added to fulfill the zero-constraint. That is, it forces the first bit to get and stay zero and thereby only the last ν bits have to be stored. Proceeding with parity-check polynomials from $\mathcal{H}_{III}$ does not increase the memory requirements as previously explained.
- $\mathcal{H}_{II}$ contains *two* parity-check polynomials and thereby the span ends in the second of the two columns in $\mathcal{H}_{II}$. For rate $R = (c-1)/c$ convolutional codes, one codeword bit within a codeword c-tuple can be determined from the other $c - 1$ codeword bits. By sectionalizing it is possible to combine two parity-check polynomials into a single step, while still preserving the properties of a binary syndrome trellis. If the first bit is already zero after combining the columns from $\mathcal{H}_I$, these two parity-check polynomials are either both added or none of them is added. If the first bit is one after combining the columns from $\mathcal{H}_I$, only one of these two polynomials is added and thereby the zero-constraint is fulfilled. We continue analogously to the previous case with one parity-check polynomial in $\mathcal{H}_{III}$.
- $\mathcal{H}_{II}$ contains *more than two* parity-check polynomials. As it is not possible to combine those parity-check polynomials by sectionalizing without violating the binary syndrome trellis property, we need to have $\nu + 1$ memory elements at least for those layers, and thereby $2^{\nu+1}$ states.

We will now summarize these results in a theorem:

Theorem 1: Consider a rate $R = (c-1)/c$ convolutional code, $c \geq 2$, with overall constraint length ν, whose parity-check polynomials are assigned to the sets $\mathcal{H}_I$, $\mathcal{H}_{II}$, and $\mathcal{H}_{III}$ according to their delay, difference of degree and delay, and degree, respectively. Then, if and only if $|\mathcal{H}_{II}| \leq 2$, the binary syndrome trellis (possibly sectionalized) can be realized with 2^ν different states at every layer. This corresponds to a

ν	polynomials			d_{free}	spectrum
1	6	6	4	2	1, 2, 4, 10, 20, 40, 80 [f]
2	7	6	5	3	1, 4, 14, 40, 116, 339, 991 [a]
3	74	64	54	4	1, 5, 24, 71, 238, 862, 2991 [d]
4	62	56	52	5	2, 13, 45, 143, 534, 2014, 7336 [f]
5	61	55	53	6	6, 27, 70, 285, 1103, 4063, 15359 [h]
6	634	514	504	7	17, 53, 133, 569, 2327, 8624, 32412 [b]
7	772	662	576	8	41, 0, 528, 0, 7497, 0, 111071 [h]
8	631	555	477	8	6, 42, 153, 510, 1853, 7338, 28378 [d]
9	7264	6214	4504	9	17, 81, 228, 933, 3469, 13203, 51286 [d]
10	7642	6406	4232	10	69, 0, 925, 0, 13189, 0, 197340 [d]
11	7741	6667	5715	10	10, 80, 260, 864, 3336, 13131, 50279
12	42074	70754	62364	11	32, 144, 477, 1769, 6718, 25717, 98945
13	52536	72166	60302	12	116, 0, 1768, 0, 24984, 0, 370358
14	71341	64657	40773	12	22, 134, 464, 1702, 6477, 24767, 94527

TABLE I
RATE $R = 2/3$ OPTIMUM FREE DISTANCE CONVOLUTIONAL CODES.

ν	polynomials			d_{free}	spectrum
1	6	6	4	2	1, 2, 4, 10, 20, 40, 80 [f]
2	7	6	5	3	1, 4, 14, 40, 116, 339, 991 [a]
3	64	54	50	4	2, 9, 28, 97, 324, 1097, 3721 [h]
4	76	56	44	5	4, 17, 54, 192, 681, 2481, 8962 [e]
5	73	62	57	6	13, 0, 180, 0, 2519, 0, 34748 [e]
6	664	504	470	6	1, 16, 48, 158, 642, 2435, 9174 [g]
7	766	604	546	8	60, 0, 649, 0, 10075, 0, 144847 [c]
8	775	567	442	8	9, 58, 161, 566, 2251, 8668, 33592 [i*]
9	6204	5074	4530	8	1, 24, 100, 321, 1143, 4479, 17194
10	7454	6302	4776	10	92, 0, 1214, 0, 17021, 0, 255676
11	7443	6547	5022	10	16, 105, 320, 1169, 4433, 16980, 65490
12	73454	44644	41370	11	53, 189, 539, 2158, 8581, 32585, 124643
13	63236	40066	55604	12	159, 0, 2223, 0, 32034, 0, 471108
14	76613	67365	57442	12	36, 153, 575, 2047, 7880, 30346, 115732

TABLE II
RATE $R = 2/3$ OPTIMUM FREE DISTANCE CONVOLUTIONAL CODES FULFILLING THEOREM 1.

maximum number of ν active rows at any column, that is, a state complexity of $s = \nu$.

III. THE SYNDROME BEAST

The BEAST—Bidirectional Efficient Algorithm for Searching code Trees—was introduced in [14] and [15]. Based on (binary) trees obtained from generator matrices, it was used both for code search [15] and for decoding of block codes [16].

However, with only minor modifications it is possible to use a (binary) syndrome tree with the BEAST. Consider a rate $R = b/c$ convolutional code and let ξ and $\boldsymbol{s}(\xi)$ denote a node in the syndrome tree and its corresponding partial syndrome, respectively. Every node ξ has a unique parent node ξ^P and 2^b children, referred to as ξ^C. For every valid codeword $\boldsymbol{v}$ of weight ω there exists a path $\xi_{\text{root}} \to \xi_{\text{toor}}$, with $\boldsymbol{s}(\xi_{\text{root}}) = \boldsymbol{s}(\xi_{\text{toor}}) = \boldsymbol{0}$. Hence, for each such path, there exists an intermediate node ξ with $s(\xi) \neq \boldsymbol{0}$, such that

$$w_{\text{F}}(\xi) = f_w + j \quad w_{\text{B}}(\xi) = b_w - j \quad j = 0, 1, \ldots, c-1$$

where w_{F} and w_{B} denote the accumulated branch weights for the sub-paths $\xi_{\text{root}} \to \xi$ and $\xi \to \xi_{\text{toor}}$, respectively, and

$$f_w = \left\lfloor \frac{\omega}{2} \right\rfloor \quad b_w = \left\lceil \frac{\omega}{2} \right\rceil.$$

Based on these observations, BEAST performs the following steps, searching for the number of codewords of weight ω:

1) *Forward search:* Starting at the zero-weight root, extend the forward syndrome code tree to obtain c sets of nodes, indexed by $j = 0, 1, \ldots, c-1$,

$$\mathcal{F}_{+j} = \left\{ \xi \middle| w_{\text{F}}(\xi) = f_w + j, w_{\text{F}}(\xi^P) < f_w, \boldsymbol{s}(\xi) \neq \boldsymbol{0} \right\}$$

2) *Backward search:* Starting at the zero-weight toor, extend the backward syndrome code tree to obtain c sets of nodes, indexed by $j = 0, 1, \ldots, c-1$,

$$\mathcal{B}_{-j} = \left\{ \xi \middle| w_{\text{B}}(\xi) = b_w - j, w_{\text{B}}(\xi^C) > b_w, \boldsymbol{s}(\xi) \neq \boldsymbol{0} \right\}$$

3) *Matching:* For every pair $\{\mathcal{F}_{+j}, \mathcal{B}_{-j}\}$, $j = 0, 1, \ldots, c-1$, count the number of matching node pairs $\{\xi, \xi'\}$ with equal partial syndrome, *i.e.,* $\boldsymbol{s}(\xi) = \boldsymbol{s}(\xi')$, $\xi \in \mathcal{F}_{+j}$ and $\xi' \in \mathcal{B}_{-j}$. Thereby, the number of codewords n_ω of weight ω is determined by

$$n_\omega = \sum_{j=0}^{c-1} \sum_{(\xi,\xi') \in \mathcal{F}_{+j} \times \mathcal{B}_{-j}} \chi(\xi, \xi')$$

where χ is the match-indictor function defined as

$$\chi(\xi, \xi') = \begin{cases} 1, & \text{if } \boldsymbol{s}(\xi) = \boldsymbol{s}(\xi') \\ 0, & \text{otherwise.} \end{cases}$$

Remark: Note that although we can use the binary syndrome tree with the BEAST we always have to complete the processing of a trellis module before storing the nodes in their appropriate sets.

IV. RESULTS

Using the syndrome BEAST, rate $R = (c-1)/c$ convolutional codes with optimum free distance are obtained for various overall constraints lengths ν. In Table I we give the first seven spectral components for optimum free distance, rate $R = 2/3$ convolutional parity-check polynomials with $\nu = 1, 2, \ldots, 13$ in the following *octal notation*: $56 \overset{\text{def}}{=} 101\,110 \overset{\text{def}}{=} 1+D^2+D^3+D^4$. Table III and Table V give similar results with the first six spectral components for rate $R = 3/4$ and rate $R = 4/5$ convolutional parity-check polynomials with $\nu = 1, 2, \ldots, 10$ and $\nu = 1, 2, \ldots, 9$, respectively.

Searching for convolutional codes fulfilling Theorem 1, we obtain the rate $R = 2/3$ convolutional parity-check polynomials with $\nu = 1, 2, \ldots, 13$ given in Table II, rate $R = 3/4$ convolutional parity-check polynomials with $\nu = 1, 2, \ldots, 10$ given in Table IV, and rate $R = 4/5$ convolutional parity-check polynomials with $\nu = 1, 2, \ldots, 9$ given in Table VI.

Comparing these results, it becomes obvious that by imposing the restrictions in Theorem 1, the performance of convolutional codes is not severely deteriorated. In most cases the same free distance d_{free} can be achieved and only a minor increase in the number of spectral components has to be accepted. On the other hand, decoding such convolutional

ν	polynomials				d_{free}	spectrum
1	6	6	4	4	2	2, 8, 17, 40, 96, 224 ^{f}
2	7	6	5	2	3	6, 23, 80, 284, 1027, 3724 ^{a}
3	74	64	54	44	4	5, 36, 152, 708, 3424, 16312 ^{b}
4	72	62	56	46	4	1, 16, 84, 376, 1912, 9728 ^{f}
5	77	65	61	47	5	7, 45, 223, 1066, 5612, 29012 ^{g}
6	604	564	554	434	6	27, 118, 529, 2978, 15201, 79518 ^{d}
7	702	632	556	422	6	5, 65, 292, 1442, 7618, 39734 ^{d}
8	767	743	551	461	7	25, 184, 714, 4081, 20038, 110599
9	7464	6774	5114	4104	8	131, 0, 3574, 0, 97035, 0
10	7276	6252	5642	4406	8	25, 202, 919, 4552, 24327, 128857

TABLE III
RATE $R = 3/4$ OPTIMUM FREE DISTANCE CONVOLUTIONAL CODES.

ν	polynomials				d_{free}	spectrum
1	6	6	4	4	2	2, 8, 17, 40, 96, 224 ^{f}
2	7	6	5	2	3	6, 23, 80, 284, 1027, 3724 ^{a}
3	70	64	54	20	4	29, 0, 532, 0, 10059, 0 ^{a}
4	74	72	64	62	4	3, 44, 160, 638, 3558, 17210 ^{e}
5	77	62	47	42	5	13, 64, 309, 1584, 8034, 40913
6	750	654	534	410	6	45, 109, 844, 3444, 20880, 100121 ^{i}
7	704	676	566	444	6	9, 88, 401, 1938, 10208, 53915
8	705	641	536	426	7	44, 199, 908, 4941, 25845, 136324 ^{c*}
9	7454	6010	5150	4674	8	207, 0, 4882, 0, 137003, 0
10	7506	6602	2752	2266	8	53, 265, 1275, 6694, 34923, 184669

TABLE IV
RATE $R = 3/4$ OPTIMUM FREE DISTANCE CONVOLUTIONAL CODES FULFILLING THEOREM 1.

ν	polynomials					d_{free}	spectrum
1	6	6	6	4	4	2	4, 12, 39, 148, 492, 1632 ^{f}
2	7	7	6	5	2	2	1, 9, 47, 229, 1095, 5265 ^{f}
3	74	70	64	54	44	3	1, 21, 139, 776, 4583, 27380 ^{f}
4	72	62	66	56	46	4	7, 56, 376, 2236, 14385, 92304 ^{f}
5	71	66	57	45	41	4	1, 36, 220, 1349, 8976, 58757
6	774	704	624	554	514	5	11, 100, 620, 4024, 26557, 177078
7	772	762	612	506	426	6	70, 245, 2504, 11894, 104486, 566209
8	717	667	571	535	441	6	14, 174, 1080, 6936, 46364, 309835
9	7754	6514	6304	5524	4474	6	1, 80, 576, 3374, 22207, 151637

TABLE V
RATE $R = 4/5$ OPTIMUM FREE DISTANCE CONVOLUTIONAL CODES.

ν	polynomials					d_{free}	spectrum
1	6	6	4	4	4	2	4, 18, 39, 102, 312, 854 ^{f}
2	7	6	6	5	2	2	1, 12, 53, 237, 1079, 4893 ^{c*}
3	74	70	60	54	50	3	5, 36, 200, 1065, 5893, 32633 ^{g}
4	76	74	70	64	54	4	30, 126, 815, 4822, 28896, 173230 ^{i}
5	75	62	51	46	42	4	3, 54, 343, 2025, 13195, 85599
6	774	610	530	464	430	5	20, 144, 896, 5841, 38536, 254172 ^{e}
7	744	714	676	564	466	6	134, 0, 6010, 0, 262004, 0 ^{h}
8	763	701	512	436	251	6	27, 283, 1625, 10305, 69518, 467725
9	1641	1355	643	507	425	6	5, 130, 835, 5072, 33847, 228818

TABLE VI
RATE $R = 4/5$ OPTIMUM FREE DISTANCE CONVOLUTIONAL CODES FULFILLING THEOREM 1.

a) code previously listed in [3]. *c)* code previously listed in [5]. *e)* code previously listed in [7]. *g)* code previously listed in [9]. *h)* code previously listed in [17].
b) code previously listed in [4]. *d)* code previously listed in [6]. *f)* code previously listed in [8]. *i)* code previously listed in [18]. **)* differs in higher spectral components.

codes can be performed with much less complexity, as their binary trellises can be implemented without increasing the state complexity and thereby with a smaller amount of memory elements.

Note that although most of the parity-check polynomials given in Table I-VI and their corresponding generator matrices have been listed in previous publications [3]–[9], [17], [18], their optimum free distance property was mostly unknown.

ACKNOWLEDGMENTS

This research was supported in part by the Swedish Research Council under Grand 621-2007-6281.

REFERENCES

[1] A. Vardy, "Trellis Structure of Codes," in *Handbook of Coding Theory*, R. A. Brualdi, W. C. Huffman, and V. Pless, Eds. Elsevier Science & Technology, Nov. 1998.

[2] G. Forney, "The Viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, no. 3, Mar. 1973, pp. 268–278.

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

[4] E. Paaske, "Short binary convolutional codes with maximal free distance for rates $2/3$ and $3/4$ (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 5, pp. 683–689, Jan. 1974.

[5] P. CharnKeitKong, H. Imai, and K. Yamaguchi, "On classes of rate $k/(k+1)$ convolutional codes and their decoding techniques," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, part 2, pp. 2181–2193, Nov. 1996.

[6] J.-J. Chang, D.-J. Hwang, and M.-C. Lin, "Some extended results on the search for good convolutional codes," *IEEE Trans. Inf. Theory*, vol. 43, no. 5, pp. 1682–1697, Sep. 1997.

[7] H.-H. Tang and M.-C. Lin, "On $(n, n-1)$ convolutional codes with low trellis complexity," *IEEE Trans. Commun.*, vol. 50, no. 1, pp. 37–47, Jan. 2002.

[8] A. 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.

[9] H.-H. Tang, M.-C. Lin, and B. Uchoa-Filho, "Minimal trellis modules and equivalent convolutional codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3738–3746, Aug. 2006.

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

[11] R. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, part 1, pp. 1855–1864, Nov. 1996.

[12] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. Piscataway, NJ: IEEE Press, 1999.

[13] G. Forney, "Coset codes II: Binary lattices and related codes," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, part 1, pp. 1152–1187, Sep. 1988.

[14] I. Bocharova, M. Handlery, R. Johannesson, and B. Kudryashov, "A BEAST for prowling in trees," in *Proc. 39th Annual Allerton Conf. Commun., Control, and Computing*, Monticello, Illinois, USA, Oct. 2001.

[15] ——, "A BEAST for prowling in trees," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1295–1302, Jun. 2004.

[16] I. Bocharova, R. Johannesson, B. Kudryashov, and M. Lončar, "BEAST decoding for block codes," *European Trans. on Telecommunications*, vol. 15, no. 4, pp. 297–305, Jul. 2004.

[17] I. Bocharova and B. Kudryashov, "Rational rate punctured convolutional codes for soft-decision Viterbi decoding," *IEEE Trans. Inf. Theory*, vol. 43, no. 4, pp. 1305–1313, Jul. 1997.

[18] D. Haccoun and G. Begin, "High-rate punctured convolutional codes for Viterbi and sequential decoding," *IEEE Trans. Commun.*, vol. 37, no. 11, pp. 1113–1125, Nov. 1989.