

# On Maximum-Likelihood Decoding of Time-Varying Trellis Codes

Wenhui Li

Skolkovo Institute of Science and Technology, Moscow, Russia

W.Li@skoltech.ru

Vladimir Sidorenko, Thomas Jerkovits, Gerhard Kramer

Institute for Communications Engineering, Technical University of Munich, Munich, Germany

(vladimir.sidorenko, thomas.jerkovits, gerhard.kramer)@tum.de

**Abstract**—Decoding complexity of convolutional and trellis codes by Viterbi decoder can be reduced by applying suggested merging algorithm to the Forney code trellis. The algorithm can be applied for every trellis section separately, which is convenient for time-varying codes, and it outputs the minimal trellis of the section. In case of convolutional codes, the same minimal trellis of every section can be obtained from the syndrome trellis of proposed split code.

## I. INTRODUCTION

### A. History and Motivation

Convolutional codes were introduced by Elias in 1955 [1]. These codes become popular when in 1967 Viterbi invented maximum-likelihood decoding algorithm [2] and Forney [3] drew a “code trellis” which made understanding of Viterbi algorithm easy and its maximum-likelihood nature obvious. Convolutional codes are linear and their trellis can be obtained from a generator matrix. One can draw a trellis arbitrarily and obtain in this way a wider class of so-called *trellis codes*.

Up to now convolutional codes and trellis codes, see e.g. [4], are widely used in telecommunications, computer security [5] and so on. We would like to decrease decoding complexity for these codes by minimizing code trellises.

To introduce constant and time-varying trellis codes we will give a couple of examples of simple binary convolutional codes. Detailed explanations can be found e.g. in [6].

### B. Example of rate $R = 1/2$ binary convolutional code $C_{1/2}$

Fig. 1 shows encoder of the code based on the shift-register, which consists of  $\nu = 2$  binary ( $q = 2$ ) memory elements. At the input, the decoder has one ( $k = 1$ ) binary information sequence  $u$ ,  $u_i \in \mathbb{F}_q = \mathbb{F}_2$ , and the decoder outputs  $n = 2$  binary code sequences  $v^{(1)}$  and  $v^{(2)}$ . The output can be seen as a sequence of vectors  $(v_i^{(1)}, v_i^{(2)})$  of length  $n = 2$ , which are called blocks of the convolutional code,  $i = 1, 2, \dots$ . The rate of code is  $R = k/n = 1/2$ .

An equivalent description of the code is shown in Fig. 2 by Forney trellis. Every codeword is represented by a path in the trellis that starts from the zero state 00 and goes to

V. Sidorenko is on leave from Institute for Information Transmission Problems, Russian Academy of Sciences. His work is supported by the Russian Government (Contract No 14.W03.31.0019).

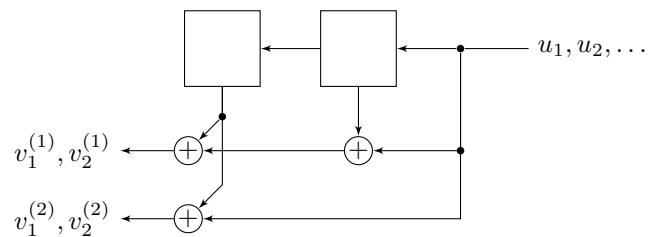


Fig. 1: Encoder of rate  $1/2$  convolutional code  $C_{1/2}$

the right. The branch labels give  $n$ -blocks of the code. The number of states in one level is  $q^\nu = 4$ , which is called the *trellis complexity*. The shown trellis consists of sequentially connected identical trellis sections, hence the code is *constant in time*. If the trellis consist of different sections then the code is called *time-varying*. Such a code can be obtained by changing connections in the encoder for every block. In general, one can arbitrarily draw a trellis and prescribe  $n$ -blocks of  $q$ -ary symbols to the branches. The obtained code is not linear in general and is called *trellis code*. If sections of time-varying trellis code are periodically repeated then the code is called *periodically time-varying* code.

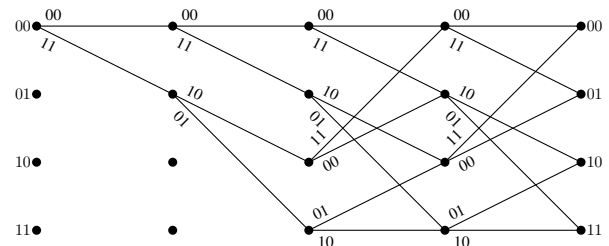


Fig. 2: Forney [3] trellis of the code  $C_{1/2}$

The Viterbi decoder (see e.g. [6]) performs one addition for every  $q$ -ary symbol on trellis and makes one binary selection when two edges merge together. Complexity of Viterbi decoder per trellis section in Fig. 2 is 16 additions and 4 binary selections.

### C. Example of high rate $R = 2/3$ convolutional code $C_{2/3}$

An encoder of rate  $R = 2/3$  convolutional code  $C_{2/3}$  is shown in Fig. 3.

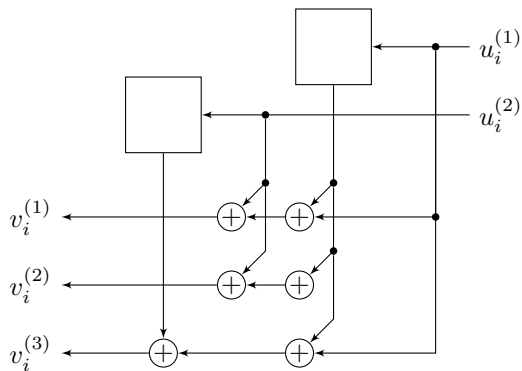


Fig. 3: Encoder of rate  $2/3$  convolutional code  $C_{2/3}$

This encoder has  $\nu = 2$  binary memory elements as well. The Forney trellis section is shown in Fig. 4 and has  $q^\nu = 4$  states  $s_0, s_1, \dots, s_3$ .

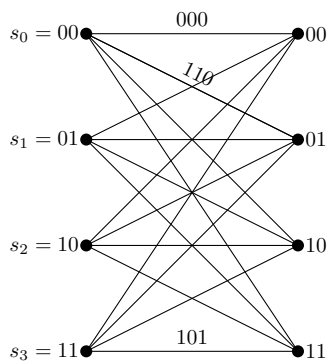


Fig. 4: Forney trellis section  $\mathcal{S}$

Complexity of Viterbi decoder of the Forney trellis section  $\mathcal{S}$  for rate  $2/3$  convolutional code includes:

- $nq^\nu q^k = 48$  additions and
- $q^\nu(q^k - 1) = 12$  binary selections.

Observe that decoding complexity of convolutional code increases with  $k$  by factor  $q^k$ . Can we instead have factor  $q^{\min\{k, n-k\}}$  in complexity like for block codes [7]? In this paper, we give an answer “yes” to this question.

### D. Our contribution

- 1) We show that Viterbi decoding complexity of convolutional and trellis codes can be reduced by applying suggested merging algorithm to the Forney code trellis.
- 2) Most of classical theoretical results [6] are obtained for time-varying convolutional codes, which in general are better than constant codes [8], [9]. Proposed method can be used to obtain the minimal trellis for time-varying codes as well, since it minimizes each trellis section separately.

- 3) In case of convolutional codes, the same minimal trellis of every section can be obtained algebraically from the syndrome trellis of proposed split code.

## II. MERGING ALGORITHM

In Fig. 5a we show a detailed fragment of the trellis section  $\mathcal{S}$  from Fig. 4. *Detailed trellis* means that every edge is labeled by one symbol from  $\mathbb{F}_q$ . We assume that solid edges are labeled by 0 and dashed edges by 1. Let us merge together two shown vertexes since the decoder will come to them from the initial state 00 with edges labeled by the same symbol. The result of merging is shown in Fig. 5b.

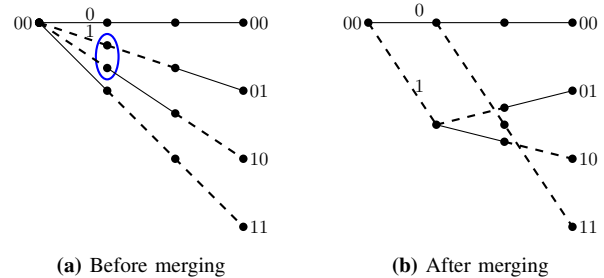


Fig. 5: Merging fragment of the detailed section  $\mathcal{S}$

Such merging from left or right vertexes as shown in Fig. 6 we call twins.

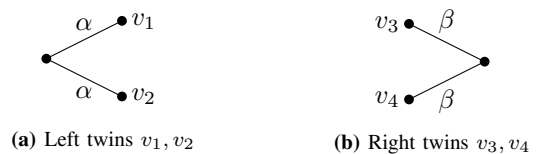


Fig. 6: Twin vertexes

**Algorithm 1** (Merging algorithm). *Merge twin vertexes until no twin vertexes exist.*

If we apply Merging algorithm to the trellis section  $\mathcal{S}$  we will get the merged section shown in Fig. 7.

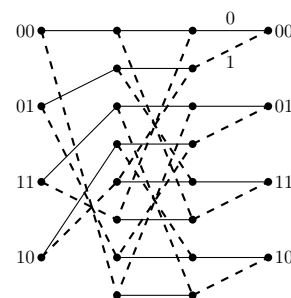


Fig. 7: Section  $\mathcal{S}$  after detalization and merging

For this trellis section the Viterbi algorithm requires:

- 32 additions (edges) and
- 12 selections.

As a result, we decrease the number of additions in comparison with Forney section. We see that Viterbi decoding complexity was decreased despite *trellis complexity* increased from 4 to 8.

Can we apply Merging algorithm to any trellis code? Fig. 8 shows that the answer is “no” in general.

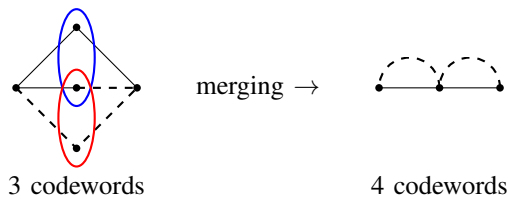


Fig. 8: Illegal merging

**Theorem 1.** *Merging is legal if and only if it does not increase the number of codewords.*

Which codes have mergable trellises? For block codes this question was answered in [10], [11]. We will consider the case of slightly more general codes and will give an answer in the next section.

### III. RECTANGULAR SPLIT CODES

In a trellis section  $\mathcal{S}$ , see Fig. 4, every edge starts from a root state  $s_{root}$ , is labeled by a codeword  $v = (v^{(1)}, v^{(2)}, \dots, v^{(n)})$ , and ends at the state  $s_{goal}$ .

**Definition 1.** *Split code  $\mathcal{S}$  is the set  $\{(s_{root}, v, s_{goal})\}$  of all words in the trellis section.*

This means that every word of a split code includes a starting and ending states in addition to the block  $v$ . Blocks  $v$  can repeat in a split code.

**Example 1.** *For the trellis section  $\mathcal{S}$  in Fig. 4 the split code is*

$$\mathcal{S} = \{(s_0, 0, 0, 0, s_0), (s_0, 1, 1, 0, s_1), \dots, (s_3, 1, 0, 1, s_3)\}$$

or

$$\mathcal{S} = \{ (0, 0, 0, 0, 0, 0), (0, 0, 1, 1, 0, 0, 1), \dots, (1, 1, 1, 0, 1, 1, 1) \}.$$

Hence, we represented a split code as usual block code, which can be shown by a usual trellis like in Fig. 9.

Given  $t$ , we represent a codeword of  $\mathcal{S}$  as  $s = (a, b)$ , where  $a$  is the  $t$ -beginning of  $s$  and  $b$  is the ending.

**Definition 2.** *Split code  $\mathcal{S}$  is rectangular if for every  $t$ : from  $(a, c), (a, d), (b, c) \in \mathcal{S}$  follows  $(b, d) \in \mathcal{S}$ .*

	$c$	$d$
$a$	*	*
$b$	*	?

Rectangular codes were introduced and investigated in [10] and [11] from where we obtain the following corollaries.

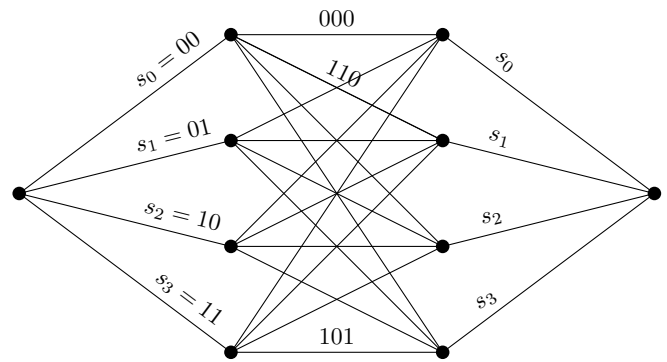


Fig. 9: A trellis for the split code  $\mathcal{S}$

#### A. Properties of split codes

Given a detailed trellis of a split code with the set  $V$  of vertexes, set  $V_t$  of vertexes for level  $t$ , set  $E$  of edges, and the set  $\mathcal{R}$  of root states, the Viterbi algorithm requires:

- $|E|$  additions
- $\rho = |E| - |V| + |\mathcal{R}|$  binary selections (cycle rank of the graph).

A trellis without twin vertexes is called *piproper*.

**Theorem 2.** *A split code has unique biproper trellis if and only if the code is rectangular.*

**Theorem 3.** *The biproper trellis of a rectangular split code:*

- can be obtained from any trellis of the split code using merging algorithm
- minimizes  $|V_t|$  for any  $t$ , i.e., it minimizes trellis complexity and is called the minimal trellis
- minimizes  $|V|$
- minimizes  $|E|$ , the number of additions
- minimizes  $\rho = |E| - |V| + |\mathcal{R}|$ , the number of binary selections

**Theorem 4.** *Every linear split code is rectangular.*

*Proof:* From

$$(a, c), (a, d), (b, c) \in \mathcal{S}$$

follows

$$-(a, c) + (a, d) + (b, c) = (b, d) \in \mathcal{S}.$$

■

**Theorem 5** ([12]). *Split code of a trellis section of a convolutional code is linear.*

Hence for a convolutional code we can apply merging algorithm to obtain a biproper trellis, which is minimal detailed trellis that minimizes decoding complexity. The class of rectangular split codes is much wider than the class of linear split codes. For arbitrary trellis code one can check its rectangularity using Theorem 1, since from Theorem 2 a trellis is mergable iff the code is rectangular. If the code is

rectangular then the following theorem shows that we reached our goal.

**Lemma 1.** *Let the Forney trellis section of  $q$ -ary rectangular trellis code of rate  $R = k/n > 1/2$  has  $q^\nu$  states then the following bounds hold for the minimal detailed trellis section*

$$|V| \leq \left[ \frac{2(q^{n-k+1} - 1)}{q - 1} + (2k - n - 1)q^{n-k} \right] q^\nu \quad (1)$$

$$|E| \leq \left[ \frac{2q(q^{n-k} - 1)}{q - 1} + (2k - n)q^{n-k+1} \right] q^\nu \quad (2)$$

$$\rho \leq [(2k - n)(q - 1)q^{n-k} + q^{n-k} - 1] q^\nu \quad (3)$$

*Proof:* (Sketch). Minimal trellis of a rectangular split code is biproper, hence the number of vertexes  $|V_t|$  and  $|V_{t+1}|$  can differ by the factor  $q$  at maximum. Let us first explain the idea of the proof using an example of the code  $C_{2/3}$ . To maximize  $|V|, |E|, \rho$  the first  $k$  subsections should be of the fork type " $<$ " or fork and merge type " $x$ ". The first subsection in Fig. 7 where the number of vertexes was increased  $q = 2$  times from 4 to 8 is of type " $<$ ". The second subsection in Fig. 7 is of the fork and merge type " $x$ " and it keeps the number of vertexes 8. If it would be of type " $<$ ", then the number of vertexes would increase to 16 and in the last subsection it can not decrease to 4 since the trellis is biproper. Totally we should have  $k = 2$  subsections of diverging types " $<$ " or " $x$ " to have rate  $k/n$ . The last subsection can be of the merging type " $>$ " only.

In general, for  $k > n/2$  it should be  $n - k$  first subsections of type " $<$ " to maximally increase the number of vertexes and edges, then  $2k - n$  subsections of type " $x$ ", and  $n - k$  last subsections of type " $>$ ". The statement of the lemma was obtained by direct computations. ■

**Theorem 6.** *Complexity (the number of additions+selections) of the Viterbi algorithm on a section of  $q$ -ary rectangular trellis code with the minimal trellis has an order of*

$$nq^{\min\{k, n-k\}} q^\nu.$$

*Proof:* For low rate codes,  $k \leq n/2$ , the decoding complexity with detailed Forney trellis consists of  $|E| = nq^k q^\nu$  additions and  $\rho = (q^k - 1)q^\nu$  selections and with minimal trellis it can not be larger. For high rate codes the statement follows from Lemma 1. ■

#### IV. MINIMAL TRELLIS SECTIONS FOR CONVOLUTIONAL CODES

The minimal trellis of a (linear) convolutional code can be obtained using a parity check matrix of the complete code using [13]. If the code is time-varying, then it may be not easy to find a parity check matrix of the complete code. We will show that for any convolutional code one can minimize trellis sections one by one. This is convenient for time-varying codes.

We will show it using an example of the constant convolutional code  $C_{2/3}$ . The same method can be applied for a time-varying convolutional code since we process every trellis section separately.

From the encoder of  $C_{2/3}$  on Fig. 3 we obtain its generator matrix

$$G = \begin{pmatrix} G_0 & G_1 & & \\ & G_0 & G_1 & \\ & & & \dots \end{pmatrix}, \quad (4)$$

where  $G_0 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$  and  $G_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ .

Hence the split code  $\mathcal{S}$  in Fig. 9 for  $i$ -th block,  $i \geq 2$ , of the convolutional code can be generated by the matrix  $G_{\mathcal{S}}$

$$G_{\mathcal{S}} = \begin{pmatrix} I_2 & G_1 & & \\ & G_0 & I_2 & \end{pmatrix} = \left( \begin{array}{cc|ccc|cc} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \right),$$

where  $I_2$  is  $2 \times 2$  identity matrix. The split code  $\mathcal{S}$  is a block binary  $[2\nu + n, \nu + k] = [7, 4]$  code with a generator matrix  $G_{\mathcal{S}}$  in a systematic form. For the information vector  $u = (u_{i-1}^{(1)}, u_{i-1}^{(2)}, u_i^{(1)}, u_i^{(2)})$ , a codeword of the split code  $\mathcal{S}$

$$\begin{aligned} c &= (u_{i-1}^{(1)}, u_{i-1}^{(2)}, u_i^{(1)}, u_i^{(2)}) G_{\mathcal{S}} \\ &= (u_{i-1}^{(1)}, u_{i-1}^{(2)}, v_i^{(1)}, v_i^{(2)}, v_i^{(3)}, u_i^{(1)}, u_i^{(2)}) \\ &= (s_{root}, v, s_{goal}) \end{aligned}$$

consists of the initial (root) state of the encoder

$$s_{root} = (u_{i-1}^{(1)}, u_{i-1}^{(2)}),$$

the  $i$ -th block of the convolutional code

$$v = (v_i^{(1)}, v_i^{(2)}, v_i^{(3)}),$$

and the final (goal) state

$$s_{goal} = (u_i^{(1)}, u_i^{(2)}).$$

Since matrix  $G_{\mathcal{S}}$  is in a systematic form, it is a trivial task to get a parity check matrix of the code  $\mathcal{S}$

$$H_{\mathcal{S}} = \left( \begin{array}{cc|ccc|cc} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{array} \right)$$

using column permutations and the following rule for an  $[n, k]$  linear code: if  $G = (P|I_k)$  is a generator matrix then  $H = (I_{n-k} | -P^T)$  is a parity check matrix for the code.

The minimal code trellis of the  $[n, k] = [7, 4]$  split code  $\mathcal{S}$  can be obtained from a parity check matrix  $H_{\mathcal{S}} = (h_1, h_2, \dots, h_7)$  using the Wolf method [7] as follows. The states in the trellis we will numerate by vectors  $a$  of length  $n - k = 3$  over  $\mathbb{F}_2$ . Each codeword  $c = (c_1, c_2, \dots, c_7) \in \mathcal{S}$  starts in the trellis from zero state at level 0, goes to state  $a_1 = c_1 h_1$  at level 1, then goes to state  $a_2 = c_1 h_1 + c_2 h_2 = a_1 + c_2 h_2$  at level 2, and so on. After the last symbol, the codeword arrives to the state  $a_n = c_1 h_1 + \dots + c_n h_n = c H_{\mathcal{S}}^T = 0$ . The obtained "syndrome" trellis is shown in Fig. 10, where each path has form  $(s_{root}, v, s_{goal})$ . This allows us to extract

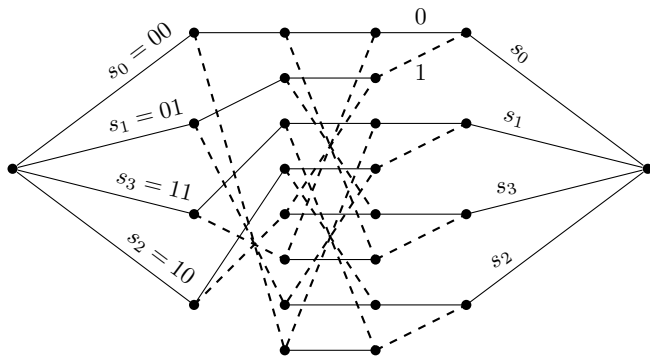


Fig. 10: The minimal syndrome trellis for the split code  $\mathcal{S}$

the minimal trellis for section  $\mathcal{S}$  shown in Fig. 7, hence it coincides with the result of the merging algorithm.

**Theorem 7.** *Obtained syndrome trellis of the section  $\mathcal{S}$  is minimal. Decoding complexity of the Viterbi decoder with this trellis has order of*

$$nq^{\min\{k, n-k\}}q^\nu.$$

*Proof:* The code is linear and hence rectangular. The trellis minimality follows from Theorem 3 since the syndrome trellis is biproper by construction. To estimate complexity one can use the general statements of Lemma 1 or Theorem 6 to estimate the order of complexity. However, in this case it is easy to estimate the order of the complexity as follows. The split code is a linear  $[N, K] = [2\nu + n, \nu + k]$  code with redundancy  $r = N - K = \nu + (n - k)$ . The trellis complexity of the syndrome trellis can not be more than the number  $q^r = q^{n-k}q^\nu$  of  $q$ -ary vectors of length  $r$  and can not be more than the number  $q^K = q^kq^\nu$  of codewords of the split code. From here we get the statement of the theorem. ■

#### V. DECODING WITH PRECOMPUTATION FOR FORNEY TRELLIS

Despite minimization of code trellises has independent interest, to be fare we will compare decoding complexity using a minimal trellis and the following method of precomputation, see also [14].

In the Forney trellis section  $\mathcal{S}$ , every vector  $v$  of length  $n = 3$  is repeated twice and for each branch we spend  $n$  additions to update the metric in the Viterbi decoder. Let us instead precompute the length of each vector in  $\mathbb{F}_q^n$ , then each precomputed length will be used twice and will decrease the number of additions.

For our example with  $q = 2, n = 3, k = 2, \nu = 2$  the direct length precomputation requires  $(n-1)q^n$  additions and  $q^{k+\nu}$  additions to sum the lengths of edges with initial metrics. Totally we have  $(n-1)q^n + q^{k+\nu} = 16 + 16 = 32$  additions like in the case of using the minimal trellis.

The split trellis (tree) of  $F_q^n$  shown in Fig. 11 allows to decrease the number of additions to  $(q^{n+1} - q^2)/(q-1)$  and get totally  $(q^{n+1} - q^2)/(q-1) + q^{k+\nu} = 12 + 16 = 28$  additions.

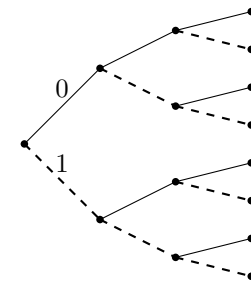


Fig. 11: Split trellis for  $F_2^3$

#### VI. COMPLEXITY OF DIFFERENT DECODING METHODS

Denote the complexity of Viterbi decoder for a trellis section, i.e., the number of additions and binary selections, by  $\kappa$  with index:

- $M$  for merged or minimal trellis,
- $F$  for Forney trellis,
- $P$  - using precomputation.

Then we have:

$$\begin{aligned} \kappa_F &\sim nq^kq^\nu \\ \kappa_P &\sim q^kq^\nu + q^n \\ \kappa_M &\sim nq^{\min\{k, n-k\}}q^\nu \end{aligned}$$

We see that only minimal trellis decoding gives decrease of complexity order for high rate codes. However, for small  $n$  precomputation can be simpler as we can see in the following table, which shows: number of additions  $E$  and selections  $\rho$  for rate  $(n-1)/n$  binary convolutional codes with  $\nu = 2$ .

$n$	3	4	5
$E_M$	32	48	64
$E_P$	28	60	124
$E_F$	48	128	320
$\rho_M$	12	20	28
$\rho_F = \rho_P$	12	28	60

TABLE I: Number of additions  $E$  and selections  $\rho$  for rate  $(n-1)/n$  binary convolutional codes with  $\nu = 2$ .  $M$  - merged,  $F$  - Forney trellis,  $P$  - using precomputations

#### VII. DISCUSSION

The minimal trellis with complexity  $\sim nq^{\min\{k, n-k\}}q^\nu$  can also be obtained using a parity check matrix of complete convolutional code [13].

Advantages of merging (M) and precomputation (P) approaches:

- they do not require linearity or any algebraic structure of the code
- rectangularity can be checked by the proposed test in Theorem 1
- “merging” and “precomputation” approaches can be applied for terminated, truncated, and tail-biting codes
- every trellis section can be processed separately, which is good for time-varying codes.

## CONCLUSIONS

- We proposed a merging algorithm to construct minimal trellis sections for a wide class of time-varying convolutional or trellis codes.
- The complexity of Viterbi decoder with the merged trellis section has order

$$\sim nq^{\min\{k, n-k\}}q^\nu,$$

which depends on the code rate  $k/n$  similar to block codes.

- For small code length decoding with precomputation can be simpler than the one with merged trellis.
- For time-varying or constant convolutional codes we propose an algebraic approach to obtain the minimal trellis section using the syndrome trellis of correspondent split code.

## REFERENCES

- [1] P. Elias, "Coding for noisy channels," *IRE Convention Record*, vol. 4, 01 1955.
- [2] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.
- [3] G. Forney Jr, "Review of random tree codes," *NASA Ames Research Center, Moffett Field, CA, USA, Tech. Rep. NASA CR73176*, 1967.
- [4] Lee-Fang Wei, "Trellis-coded modulation with multidimensional constellations," *IEEE Transactions on Information Theory*, vol. 33, no. 4, pp. 483–501, July 1987.
- [5] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.
- [6] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. Wiley-IEEE Press, 1999.
- [7] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 76–80, Jan 1978.
- [8] M. Mooser, "Some periodic convolutional codes better than any fixed code (corresp.)," *IEEE Transactions on Information Theory*, vol. 29, no. 5, pp. 750–751, Sep. 1983.
- [9] P. J. Lee, "There are many good periodically time-varying convolutional codes," *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 460–463, March 1989.
- [10] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1924–1937, Nov 1995.
- [11] V. Sidorenko, "The Euler Characteristic of the Minimal Code Trellis is Maximum," *Probl. Peredachi Inf.*, vol. 33, no. 1, pp. 87–93, 1997.
- [12] G. D. Forney and M. D. Trott, "The dynamics of group codes: state spaces, trellis diagrams, and canonical encoders," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1491–1513, Sep. 1993.
- [13] 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, Sep. 1994.
- [14] A. Lafourcade and A. Vardy, "Optimal sectionalization of a trellis," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 689–703, May 1996.