Another look at the exact bit error probability for Viterbi decoding of convolutional codes

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

2011

Another look at the exact bit error probability for Viterbi decoding of convolutional codes

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

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

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

***Abstract*—In 1995, Best et al. published a formula for the exact bit error probability for Viterbi decoding of the rate $R = 1/2$, memory $m = 1$ (2-state) convolutional encoder with generator matrix $G(D) = (1 \;\; 1 + D)$ when used to communicate over the binary symmetric channel. Their method was later extended to the rate $R = 1/2$, memory $m = 2$ (4-state) generator matrix $G(D) = (1 + D^2 \;\; 1 + D + D^2)$ by Lentmaier et al.**

In this paper, we shall use a different approach to derive the exact bit error probability. We derive and solve a general matrix recurrent equation connecting the average information weights at the current and previous steps of the Viterbi decoding. A closed form expression for the exact bit error probability is given. Our general solution yields the expressions for the exact bit error probability obtained by Best et al. ($m = 1$) and Lentmaier et al. ($m = 2$) as special cases. The exact bit error probability for the binary symmetric channel is determined for various 8 and 16 states encoders including both polynomial and rational generator matrices for rates $R = 1/2$ and $R = 2/3$. Finally, the exact bit error probability is calculated for communication over the quantized additive white Gaussian noise channel.

I. Introduction

The challenging problem of deriving an expression for the exact bit error probability for communication over the binary symmetric channel (BSC) was first addressed by Morrissey in 1970 [1] for a suboptimum feedback decoding technique. For the memory $m = 1$ convolutional encoder with generator matrix $G(D) = (1 \;\; 1 + D)$, he got an expression which coincides with the Viterbi decoding bit error probability published in 1995 by Best et al. [2], who used a more general approach based on considering a Markov chain of the so-called metric states of the Viterbi decoder which is due to Burnashev and Cohn [3]. The Best et al. method was extended to the memory $m = 2$ convolutional encoder with generator matrix $G(D) = (1 + D^2 \;\; 1 + D + D^2)$ by Lentmaier et al. [4].

We use a different approach to derive the exact bit error probability for Viterbi decoding of minimal convolutional encoders. A matrix recurrent equation is obtained and solved for the average information weights at the current and previous states that are connected by the branches decided by the Viterbi decoder during the current step [5].

To illustrate our method we use the $R = 1/2$ minimal, memory $m = 1$ (2-state) convolutional feed-forward encoder with generator matrix $G(D) = (1 \;\; 1{+}D)$ realized in controller canonical form to communicate over the BSC. (However, our derivation holds for any memory m.) The extension to rate $R = 1/c$ is trivial while in the presentation we give nontrivial extensions to rate $R = b/c$ as well as feedback encoders. Finally, we consider the quantized additive white Gaussian noise channel. Examples of encoders with 2, 4, 8, and 16 states are given. For 32 states the computational complexity becomes prohibitively large. Before proceeding we would like to emphasize that the bit error probability is an encoder property, not a code property.

II. A recurrent equation for the information weights

Assume that the all-zero sequence is transmitted over the BSC. Let $W_t(\sigma)$ denote the weight of the information sequence corresponding to the code sequence decided by the Viterbi decoder at state σ and time t. If the initial values $W_0(\sigma)$ are known then the random process $W_t(\sigma)$ is a function of the random process of the received c-tuples $\boldsymbol{r}_i$, $i = 0, 1, \dots, t{-}1$. Thus, the ensemble $\{\boldsymbol{r}_i, \, i = 0, 1, \dots, t{-}1\}$ determines the ensemble $\{W_i(\sigma), \, i = 1, 2, \dots, t\}$.

Our goal is to determine the mathematical expectation of the random variable $W_t(\sigma)$ over this ensemble, since for minimal convolutional encoders the bit error probability can be computed as the limit

$$P_\mathrm{b} = \lim_{t \to \infty} \frac{E\left[W_t(\sigma = 0)\right]}{t} \tag{1}$$

assuming that this limit exists.

Since we have chosen realizations in controller canonical form the encoder states can be represented by the m-tuples of the inputs of the shift register, that is, $\boldsymbol{\sigma}_t = u_{t-1}u_{t-2}\dots u_{t-m}$. In the sequel we usually denote these encoder states σ, $\sigma \in \{0, 1, \dots, 2^m - 1\}$. During the decoding step at time $t + 1$ the Viterbi algorithm computes the cumulative Viterbi branch metric vector $\boldsymbol{\mu}_{t+1} = (\mu_{t+1}(0) \;\; \mu_{t+1}(1) \dots \mu_{t+1}(2^m - 1))$ at time $t + 1$ using the vector $\boldsymbol{\mu}_t$ at time t and the received c-tuple $\boldsymbol{r}_t$. In our analysis it is convenient to normalize the metrics such that the cumulative metrics at every all-zero state will be zero, that is, we subtract the value $\mu_t(0)$ from $\mu_t(1), \mu_t(2), \dots, \mu_t(2^m - 1)$

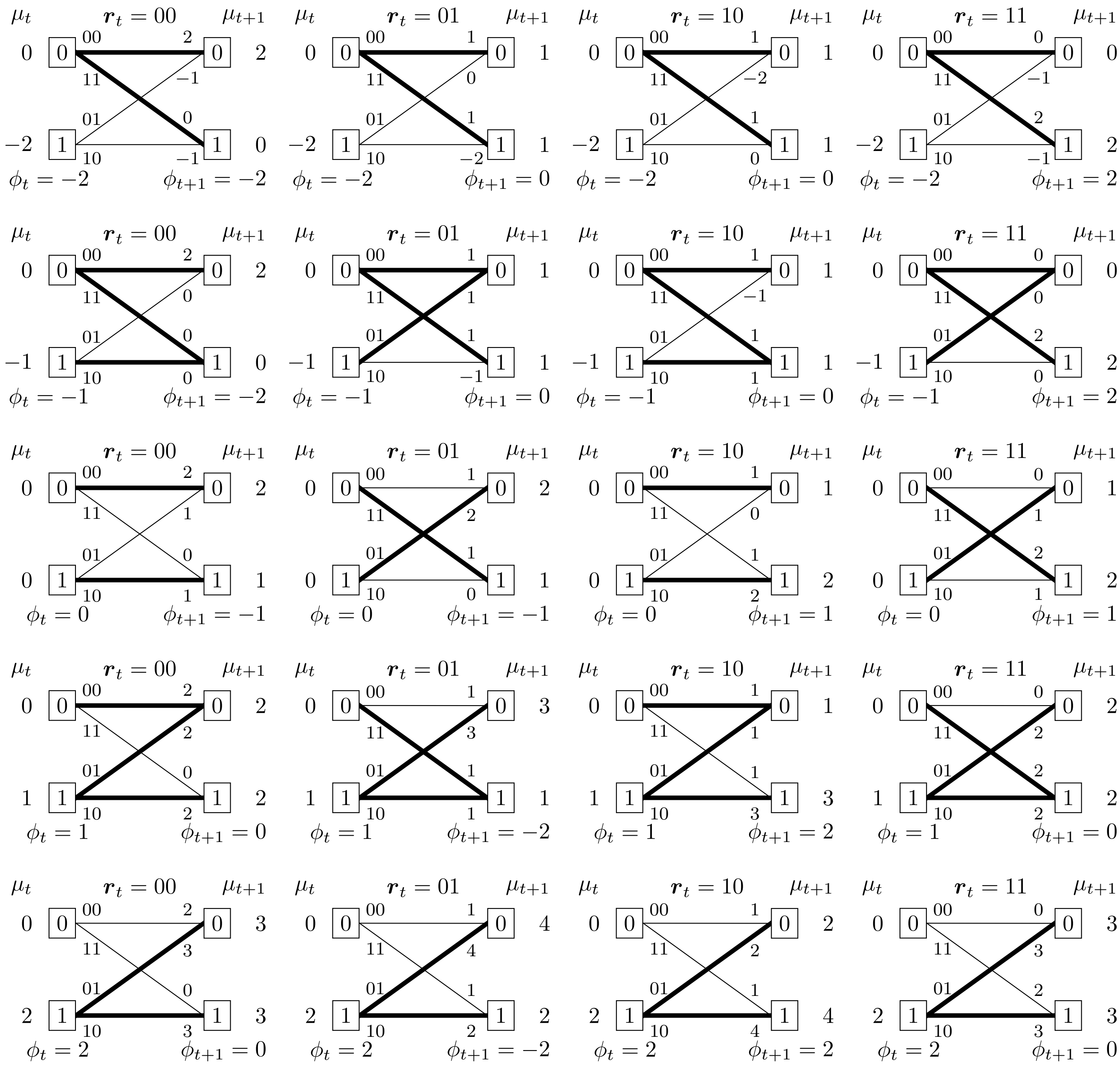

Fig. 1. The 20 different trellis sections for the $G(D) = (1\ \ 1+D)$ generator matrix.

and introduce the normalized cumulative branch metric vector

$$\begin{aligned}\boldsymbol{\phi}_t &= \Big(\phi_t(1)\ \ \phi_t(2)\dots\phi_t(2^m-1)\Big) \qquad (2)\\ &= \Big(\mu_t(1)-\mu_t(0)\ \ \mu_t(2)-\mu_t(0)\dots\mu_t(2^m-1)-\mu_t(0)\Big)\end{aligned}$$

For a memory $m = 1$ (2-state) encoder we obtain the scalar

$$\phi_t = \phi_t(1) \qquad (3)$$

while for a memory $m = 2$ (4-state) encoder we have the vector

$$\boldsymbol{\phi}_t = \Big(\phi_t(1)\ \ \phi_t(2)\ \ \phi_t(3)\Big) \qquad (4)$$

First we consider the rate $R = 1/2$, memory $m = 1$ minimal encoder with generator matrix $G(D) = (1\ \ 1 + D)$. In Fig. 1 we show the 20 different trellis sections corresponding to the $M = 5$ different normalized cumulative metrics $\phi_t \in \{-2, -1, 0, 1, 2\}$ and the four different received tuples $\boldsymbol{r}_t = 00, 01, 10, 11$. The bold branches correspond to the branches decided by the Viterbi decoder at time $t + 1$. When we have two branches entering the same state with the same state metric we have a tie which we, in our analysis, will resolve by fair coin-flipping.

The normalized cumulative metric Φ_t is a 5-state Markov chain with transition probability matrix $\Phi = (\phi_{jk})$, where

$$\phi_{jk} = Pr\left(\phi_{t+1} = \phi^{(k)} \;\middle|\; \phi_t = \phi^{(j)}\right) \tag{5}$$

From the trellis sections in Fig. 1 we obtain the following transition probability matrix

$$\Phi = \begin{array}{c} \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \\ \phi^{(j)} \end{array} \begin{array}{c} \begin{array}{ccccc} -2 & -1 & 0 & 1 & 2 \end{array} \quad \phi^{(k)} \\ \begin{pmatrix} q^2 & 0 & 2pq & 0 & p^2 \\ q^2 & 0 & 2pq & 0 & p^2 \\ 0 & q & 0 & p & 0 \\ pq & 0 & p^2+q^2 & 0 & pq \\ pq & 0 & p^2+q^2 & 0 & pq \end{pmatrix} \\ \end{array} \tag{6}$$

Let $\boldsymbol{p}_t$ denote the probability of the M different metric values of Φ_t, that is, $\phi_t \in \left\{\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(M)}\right\}$. The stationary distribution of the normalized cumulative metrics Φ_t is denoted $\boldsymbol{p}_\infty = \left(p_\infty^{(1)} \; p_\infty^{(2)} \ldots p_\infty^{(M)}\right)$ and is determined as the solution of, for example, the first $M-1$ equations of

$$\boldsymbol{p}_\infty \Phi = \boldsymbol{p}_\infty \tag{7}$$

and

$$\sum_{i=1}^{M} p_\infty^{(i)} = 1 \tag{8}$$

For our $m = 1$ convolutional encoder we obtain

$$\boldsymbol{p}_\infty^T = \frac{1}{1+3p^2-2p^3} \begin{pmatrix} 1-4p+8p^2-7p^3+2p^4 \\ 2p-5p^2+5p^3-2p^4 \\ 2p-3p^2+3p^3 \\ 2p^2-3p^3+2p^4 \\ p^2+p^3-2p^4 \end{pmatrix} \tag{9}$$

Now we return to the information weight $W_t(\sigma)$. From the trellis sections in Fig. 1 it is easily seen how the information weights are transformed during one step of the Viterbi decoding. Transitions from state 0 or state 1 to state 0 decided by the Viterbi decoder without tiebreaking do not cause an increment of the information weights; we simply copy the information weight from the state at the root of the branch to the state at the termini of the branch since such a transmission corresponds to $\hat{u}_t = 0$. Having a transition from state 0 to state 1 decided by the Viterbi decoder without tiebreaking, we obtain the information weight at state 1 and time $t+1$ by incrementing the information weight at state 0 and time t since such a transition corresponds to $\hat{u}_t = 1$. Similarly, coming from state 1 we obtain the information weight at state 1 and time $t+1$ by incrementing the information weight at state 1 and time t. If we have tiebreaking, we use the arithmetic average of the information weights at the two states 0 and 1 at time t in our updating procedure.

Now we introduce some notations for rate $R = 1/c$, memory m encoders. The values of the random variable $W_t(\sigma)$ are distributed over the cumulative metrics ϕ_t according to the vector $\boldsymbol{p}_t$. Let $\boldsymbol{w}_t$ be the vector of the information weights at time t split both on the 2^m states σ_t and the M metric values ϕ_t; that is, we can write $\boldsymbol{w}_t$ as the following vector of $M2^m$ entries:

$$\begin{aligned} \boldsymbol{w}_t = \Big(\; & w_t(\phi^{(1)}, \sigma = 0) \quad \ldots \quad w_t(\phi^{(M)}, \sigma = 0) \\ & w_t(\phi^{(1)}, \sigma = 1) \quad \ldots \quad w_t(\phi^{(M)}, \sigma = 1) \\ & \qquad \vdots \qquad\qquad\qquad\qquad \vdots \\ & w_t(\phi^{(1)}, \sigma = 2^m-1) \; \ldots \; w_t(\phi^{(M)}, \sigma = 2^m-1) \Big) \end{aligned} \tag{10}$$

The vector $\boldsymbol{w}_t$ describes the dynamics of the information weights when we proceed along the trellis. It satisfies the following recurrent equation

$$\begin{cases} \boldsymbol{w}_{t+1} = \boldsymbol{w}_t A + \boldsymbol{b}_t B \\ \boldsymbol{b}_{t+1} = \boldsymbol{b}_t \Pi \end{cases} \tag{11}$$

where A and B are $M2^m \times M2^m$ nonnegative matrices and Π is an $M2^m \times M2^m$ stochastic matrix. The matrix A is the linear part of the affine transformation of the information weights and it can be determined from the trellis sections in Fig. 1. The matrix B describes the increments of the information weights. The vector $\boldsymbol{b}_t$ of length $M2^m$ is the concatenation of 2^m stochastic vectors $\boldsymbol{p}_t$. Hence, the $M2^m \times M2^m$ matrix Π is given by

$$\Pi = \begin{pmatrix} \Phi & 0 & \ldots & 0 \\ 0 & \Phi & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \Phi \end{pmatrix} \tag{12}$$

For simplicity, we choose the initial value of the vector of information weights to be

$$\boldsymbol{w}_0 = \boldsymbol{0} \tag{13}$$

From (1) follows that we are interested in the asymptotic values. Thus we can exploit the steady-state probabilities $\boldsymbol{p}_\infty$ and use

$$\boldsymbol{b}_\infty = (\boldsymbol{p}_\infty \, \boldsymbol{p}_\infty \cdots \boldsymbol{p}_\infty) \tag{14}$$

as starting value in (11). Since

$$\boldsymbol{b}_\infty = \boldsymbol{b}_\infty \Pi \tag{15}$$

(11) can be simplified to

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t A + \boldsymbol{b}_\infty B \tag{16}$$

The following two examples illustrate how A can be obtained from the trellis sections in Fig. 1.

Consider first a situation without tiebreaking; for example, the trellis section in the upper left corner, where we have $\phi_t = -2$, $\phi_{t+1} = -2$, and $\boldsymbol{r}_t = 00$. Following the bold branches, we first copy with probability $Pr(\boldsymbol{r}_t = 00) = q^2$ the information weight from state $\sigma_t = 0$ to state $\sigma_{t+1} = 0$, and obtain the information weight at $\sigma_{t+1} = 1$ as the information weight at $\sigma_t = 0$ plus 1 (since $\hat{u}_t = 1$ for this branch). We have now determined four of the entries in A, namely, the two entries for $\sigma_t = 0$, $\phi_t = -2$, and $\phi_{t+1} = -2$, which both are q^2, and the two entries for $\sigma_t = 1$, $\phi_t = -2$, and $\phi_{t+1} = -2$, which both are 0. Notice that, when we determine the entry for $\phi_{t+1} = 0$, we have to add the probabilities for the two trellis sections with $\phi_{t+1} = 0$.

Next we include tiebreaking and choose the trellis section with $\phi_t = -1$, $\phi_{t+1} = -2$, and $\boldsymbol{r}_t = 00$. Here we have to resolve ties at $\sigma_{t+1} = 1$. By following the bold branch from $\sigma_t = 0$ to $\sigma_{t+1} = 0$ we conclude that the information weight at state $\sigma_{t+1} = 0$ is a copy of the information weight at state $\sigma_t = 0$. Then we follow the two bold branches to state $\sigma_{t+1} = 1$ where the information weight is the arithmetic average of the information weights at states $\sigma_t = 0$ and $\sigma_t = 1$ plus 1. We have now determined another four entries of A, namely, the entry for $\sigma_t = 0$, $\phi_t = -1$, $\phi_{t+1} = -2$, and $\sigma_{t+1} = 0$ which is q^2, the two entries for $\phi_t = -1$, $\phi_{t+1} = -2$, and $\sigma_{t+1} = 1$ which are both $q^2/2$ (the tie is resolved by coin-flipping), and, finally, the entry for $\sigma_t = 1$, $\phi_t = -2$, $\phi_{t+1} = -2$, and $\sigma_{t+1} = 0$ which is 0 since there is no bold branch between $\sigma_t = 1$ and $\sigma_{t+1} = 0$ in this trellis section.

Proceeding in this manner yields the matrix A (17) for the memory $m = 1$ convolutional encoder with generator matrix $G(D) = (1 \;\; 1 + D)$. This matrix A is specified at the bottom of this page.

Let a_{kl} and b_{kl}, $0 \leq k, l \leq M2^m - 1$, denote the entries of the matrices A and B, respectively. Then, in general, the entries

$$b_{kl} = \beta_{kl} a_{kl} \tag{18}$$

where β_{kl} is the increment of the path information weight corresponding to the transition $(\sigma_t, \phi_t) \to (\sigma_{t+1}, \phi_{t+1})$.

For feed-forward and feedback convolutional encoders realized in controller canonical form the increments are identical for all entries b_{kl} within a given submatrix B_{ij}, $0 \leq i, j \leq 2^m - 1$. Then we use the notation

$$B_{ij} = \beta\left(\sigma_{t+1} = j \mid \sigma_t = i\right) A_{ij} \tag{19}$$

However, for rate $R = b/c$, $b > 1$, convolutional encoders we could have parallel branches. Then the increments β_{kl} for the entries b_{kl} within a given submatrix B_{ij} might assume different values.

For rate $R = 1/c$ convolutional feed-forward encoders realized in controller canonical form we only have increments when entering the states σ_{t+1} whose, when written as an m-tuple, first digit is a 1. Thus it follows that for our 2-state encoder we have

$$B = \begin{pmatrix} \mathbf{0}_{5,5} & A_{01} \\ \mathbf{0}_{5,5} & A_{11} \end{pmatrix} \tag{20}$$

Finally, we notice that every encoder state is reachable with probability 1, thus we have

$$\sum_{i=0}^{2^m-1} A_{ij} = \Phi, \quad j = 0, 1, \ldots, 2^m - 1 \tag{21}$$

In the next section we shall solve the recurrent matrix equation (16).

III. Solving the Recurrent Equation

By iterating the recurrent equation (16) and using the initial value (13) we obtain

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{b}_\infty B A^t + \boldsymbol{b}_\infty \Pi B A^{t-1} + \cdots + \boldsymbol{b}_\infty \Pi^t B \\ &= \boldsymbol{b}_\infty B A^t + \boldsymbol{b}_\infty B A^{t-1} + \cdots + \boldsymbol{b}_\infty B \end{aligned} \tag{22}$$

where the second equality follows from (15). From (22) it follows that

$$\begin{aligned} \lim_{t\to\infty} \frac{\boldsymbol{w}_t}{t} &= \lim_{t\to\infty} \frac{\boldsymbol{w}_{t+1}}{t} = \lim_{t\to\infty} \frac{1}{t} \sum_{j=0}^{t} \boldsymbol{b}_\infty B A^{t-j} \\ &= \lim_{t\to\infty} \frac{2}{t} \sum_{j=0}^{t/2} \boldsymbol{b}_\infty \frac{BA^{t-j} + BA^j}{2} - \lim_{t\to\infty} \frac{\boldsymbol{b}_\infty B A^{t/2}}{t} \\ &= \lim_{t\to\infty} \boldsymbol{b}_\infty B A^{t/2} = \boldsymbol{b}_\infty B A^\infty \end{aligned} \tag{23}$$

where A^∞ denotes the limit of the sequence A^t when t tends to infinity and we used that, if a sequence is convergent to a finite limit, then it is Cesàro-summable to the same limit.

We summarize the following important properties of the matrix $A = (a_{kl})$:

- Nonnegativity, that is, $a_{kl} \geq 0$, $0 \leq k, l \leq M2^m - 1$.

$$A = \begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} = \begin{array}{cc} & \begin{array}{ccccc|ccccc} \multicolumn{5}{c|}{\sigma_{t+1} = 0} & \multicolumn{5}{c}{\sigma_{t+1} = 1} \\ -2 & -1 & 0 & 1 & 2 & -2 & -1 & 0 & 1 & 2 \end{array} \; \phi_{t+1} \\ \begin{array}{rr} & -2 \\ & -1 \\ \sigma_t = 0 & 0 \\ & 1 \\ & 2 \\ & -2 \\ & -1 \\ \sigma_t = 1 & 0 \\ & 1 \\ & 2 \\ & \phi_t \end{array} & \left(\begin{array}{ccccc|ccccc} q^2 & 0 & 2pq & 0 & p^2 & q^2 & 0 & 2pq & 0 & p^2 \\ q^2 & 0 & 3pq/2 & 0 & p^2/2 & q^2/2 & 0 & 3pq/2 & 0 & p^2 \\ 0 & q^2 & 0 & pq & 0 & 0 & pq & 0 & p^2 & 0 \\ 0 & 0 & q^2/2 & 0 & qp/2 & pq/2 & 0 & p^2/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & pq/2 & 0 & p^2/2 & q^2/2 & 0 & pq/2 & 0 & 0 \\ 0 & pq & 0 & p^2 & 0 & 0 & q^2 & 0 & pq & 0 \\ pq & 0 & q^2/2 + p^2 & 0 & pq/2 & pq/2 & 0 & q^2 + p^2/2 & 0 & pq \\ pq & 0 & q^2 + p^2 & 0 & pq & pq & 0 & q^2 + p^2 & 0 & pq \end{array}\right) \end{array} \tag{17}$$

- For any convolutional encoder with memory m, A has a block structure, $A = (A_{ij})$, $i, j = 0, 1, \ldots, 2^m - 1$, where the block A_{ij} corresponds to the transitions from $\sigma_t = i$ to $\sigma_{t+1} = j$.
- Summing over the blocks columnwise yields

$$\sum_{i=0}^{2^m-1} A_{ij} = \Phi, \ \ j = 0, 1, \ldots, 2^m - 1 \tag{24}$$

From (24) follows that

$$\boldsymbol{e}_{\mathrm{L}} = (\boldsymbol{p}_\infty \ \ \boldsymbol{p}_\infty \ldots \boldsymbol{p}_\infty) \tag{25}$$

satisfies

$$\boldsymbol{e}_{\mathrm{L}} A = \boldsymbol{e}_{\mathrm{L}} \tag{26}$$

and, hence, $\boldsymbol{e}_{\mathrm{L}}$ is a left eigenvector with eigenvalue $\lambda = 1$.

From the nonnegativity follows (Corollary 8.1.30 [6]) that $\lambda = 1$ is a maximal eigenvalue of A. Let $\boldsymbol{e}_{\mathrm{R}}$ be the right eigenvector corresponding to the eigenvalue $\lambda = 1$ and let $\boldsymbol{e}_{\mathrm{R}}$ be normalized such that $\boldsymbol{e}_{\mathrm{L}}\boldsymbol{e}_{\mathrm{R}} = 1$. If $\boldsymbol{e}_{\mathrm{L}}$ is unique (up to normalization) then it follows (Lemma 8.2.7, statement (i) [6]) that

$$A^\infty = \boldsymbol{e}_{\mathrm{R}}\boldsymbol{e}_{\mathrm{L}} \tag{27}$$

Combining (23), (25), and (27) yields

$$\lim_{t\to\infty} \frac{\boldsymbol{w}_t}{t} = \boldsymbol{b}_\infty B \boldsymbol{e}_{\mathrm{R}} (\boldsymbol{p}_\infty \ \ \boldsymbol{p}_\infty \ldots \boldsymbol{p}_\infty) \tag{28}$$

From (1) it follows that the expression for the exact bit error probability can be written as

$$\begin{aligned} P_{\mathrm{b}} &= \lim_{t\to\infty} \frac{E[W_t(\sigma = 0)]}{t} = \lim_{t\to\infty} \frac{\sum_{i=1}^{M} \boldsymbol{w}_t(\phi^{(i)}, \sigma = 0)}{t} \\ &= \lim_{t\to\infty} \frac{\boldsymbol{w}_t(\sigma = 0)\mathbf{1}_{1,M}^T}{t} \end{aligned} \tag{29}$$

where $\mathbf{1}_{1,M}$ is the all-one row vector of length M. In other words, to get the expression for P_{b} we sum up the first M components of the vector on the right side of (28), or, equivalently, we multiply this vector by the vector $(\mathbf{1}_{1,M} \ \ \mathbf{0}_{1,M} \ldots \mathbf{0}_{1,M})^T$. Then we obtain the following closed-form expression for the exact bit error probability

$$P_{\mathrm{b}} = \boldsymbol{b}_\infty B \boldsymbol{e}_{\mathrm{R}} \tag{30}$$

In summary, for rate $R = b/c$ minimal convolutional encoders we can determine the exact bit error probability P_{b} for Viterbi decoding, when communicating over the BSC, as follows:

- Construct the set of metric states and find the stationary probability distribution $\boldsymbol{p}_\infty$
- Construct the matrices A and B analogously to the memory $m = 1$ example given above and compute its right eigenvector $\boldsymbol{e}_{\mathrm{R}}$ normalized according to $(\boldsymbol{p}_\infty \ \ \boldsymbol{p}_\infty \ldots \boldsymbol{p}_\infty)\boldsymbol{e}_{\mathrm{R}} = 1$.
- Compute P_{b} using (30).

IV. Some examples

First we consider the rate $R = 1/2$, memory $m = 1$ (2-state) convolutional code with generator matrix $G(D) = (1 \ \ 1 + D)$. Its set of metric states is $\{-2, -1, 0, 1, 2\}$ and the stationary probability distribution $\boldsymbol{p}_\infty$ is given by (9).

From the trellis sections in Fig. 1 we obtain the matrix A (17) with normalized right eigenvector

$$\boldsymbol{e}_{\mathrm{R}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{pq}{2} \\ \frac{4pq}{2 - p + 4p^2 - 4p^3} \\ \frac{(2 + 7p - 12p^2 + 13p^3 - 12p^4 + 4p^5)}{2(2 - p + 4p^2 - 4p^3)} \\ 1 \end{pmatrix} \tag{31}$$

Finally, inserting (9) and (31) into (30) yields the exact bit error probability

$$P_{\mathrm{b}} = \frac{14p^2 - 23p^3 + 16p^4 + 2p^5 - 16p^6 + 8p^7}{(1 + 3p^2 - 2p^3)(2 - p + 4p^2 - 4p^3)} \tag{32}$$

which coincides with the bit error probability formula in [2].

Next we consider the rate $R = 1/2$, memory $m = 2$ (4-state) convolutional encoder with generator matrix $G(D) = (1 + D^2 \ \ 1 + D + D^2)$. In Fig. 2 we show the four trellis sections for $\boldsymbol{\phi}_t = (0 \ \ 0 \ \ 0)$. The corresponding metric states at time $t+1$ are $\boldsymbol{\phi}_{t+1} = (-1 \ \ 0 \ \ -1)$ and $(1 \ \ 0 \ \ 1)$. Completing the set of trellis sections yields 31 different normalized metric states.

Following the method of calculating the exact bit error probability in Section III we obtain

$$\begin{aligned} P_{\mathrm{b}} = 44p^3 &+ \frac{3519}{8}p^4 - \frac{14351}{32}p^5 - \frac{1267079}{64}p^6 \\ &- \frac{31646405}{512}p^7 + \frac{978265739}{2048}p^8 \\ &+ \frac{3931764263}{1024}p^9 - \frac{48978857681}{32768}p^{10} + \cdots \end{aligned} \tag{33}$$

which coincides with the previously obtained result by Lentmaier et al. [4].

The rate $R = 1/2$, memory $m = 3$ (8-state) convolutional encoder with generator matrix $G(D) = (1 + D^2 + D^3 \ \ 1 + D + D^2 + D^3)$ has 433 normalized metric states. The rate $R = 1/2$, memory $m = 4$ (16-state) convolutional matrix $G(D) = (1 + D + D^4 \ \ 1 + D^2 + D^3 + D^4)$ has 188686 normalized metric states.

Since these two latter examples are essentially more complex we computed the exact bit error probability (following the method in Section III) only numerically. The results are shown in Fig. 3 and compared with the curves for the previously discussed memory $m = 1$ and $m = 2$ encoders.

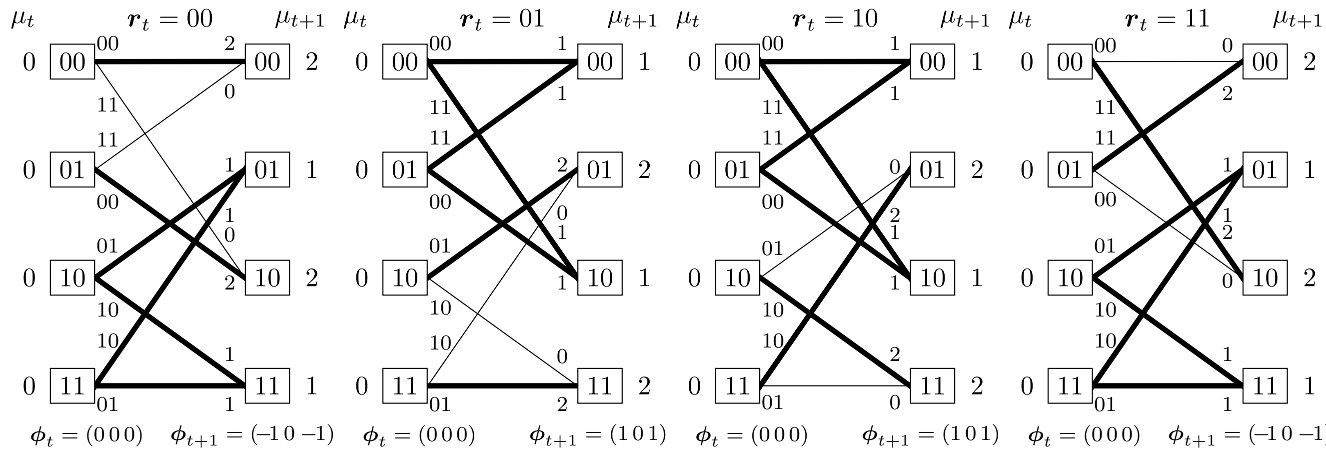

Fig. 2. Four different trellis sections of the in total 124 for the $G(D) = (1 + D^2 \;\; 1 + D + D^2)$ generator matrix.

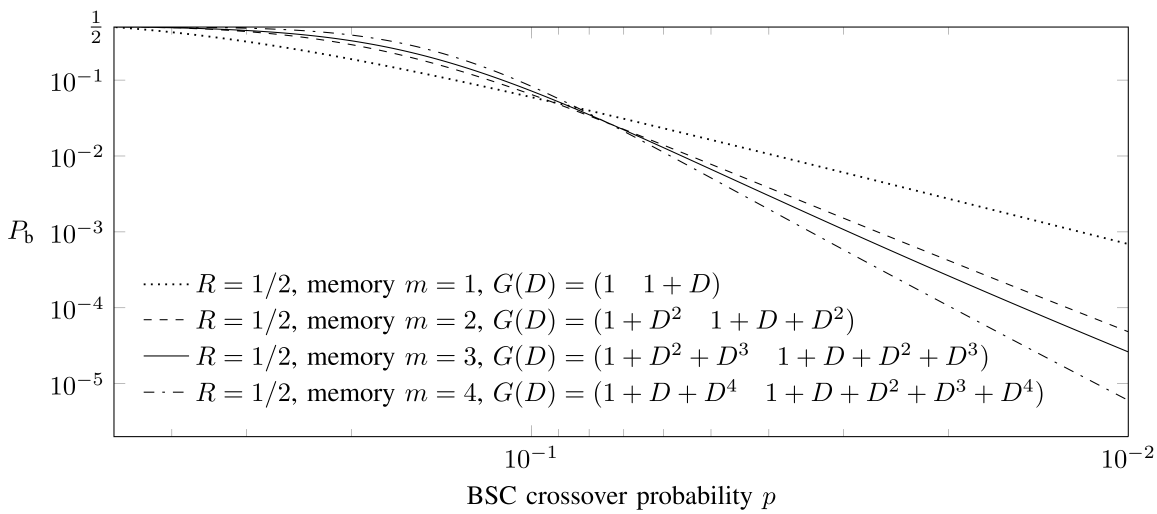

Fig. 3. Exact bit error probability for rate $R = 1/2$, memory $m = 1$ $\big(G(D) = (1 \;\; 1 + D)\big)$, memory $m = 2$ $\big(G(D) = (1 + D^2 \;\; 1 + D + D^2)\big)$, and memory $m = 3$ $\big(G(D) = (1 + D^2 + D^3 \;\; 1 + D + D^2 + D^3)\big)$.

The rate $R = 1/2$, memory $m = 5$ (32-state) convolutional encoder with generator matrix $G(D) = (1 + D + D^2 + D^3 + D^4 + D^5 \;\; 1 + D^3 + D^5)$ has more than 4130000 normalized metric states and we find it not feasible to calculate the exact bit error probability for this encoder.

V. Comments

During the presentation, in addition to the examples in Section IV we shall also consider two rate $R = 1/2$ 4-state systematic convolutional encoders with feedback, rate $R = 2/3$ convolutional feed-forward encoders with up to 16 states, all realized in controller canonical form, as well as a rate $R = 2/3$ 8-state convolutional encoder realized in observer canonical form. Finally, examples of rate $R = 1/2$ convolutional encoders used to communicate over the quantized additive white Gaussian noise channel are given.

Acknowledgment

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

References

[1] T. N. Morrissey, Jr., "Analysis of decoders for convolutional codes by stochastic sequential machine methods," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 4, pp. 460–469, Jul. 1970.

[2] M. R. Best, M. V. Burnashev, Y. Levy, A. Rabinovich, P. C. Fishburn, A. R. Calderbank, and D. J. Costello, Jr., "On a technique to calculate the exact performance of a convolutional code," *IEEE Trans. Inf. Theory*, vol. 41, no. 2, pp. 441–447, Mar. 1995.

[3] M. V. Burnashev and D. L. Cohn, "Symbol error probability for convolutional codes," *Problems on Information Transmission*, vol. 26, no. 4, pp. 289–298, 1990.

[4] M. Lentmaier, D. V. Truhachev, and K. S. Zigangirov, "Analytic expressions for the bit error probabilities of rate-1/2 memory 2 convolutional encoders," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1303–1311, Jun. 2004.

[5] I. Bocharova, F. Hug, R. Johannesson, and B. Kudryashov, "On the exact bit error probability for viterbi decoding of convolutional codes," in *Proc. Information Theory and Applications Workshop (ITA11)*, San Diego, Feb. 06–11, 2011.

[6] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, Feb. 1990.