

On the exact bit error probability for Viterbi decoding of convolutional codes

Irina E. Bocharova¹, Florian Hug², Rolf Johannesson², and Boris D. Kudryashov¹

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

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

Abstract—Forty years ago, Viterbi published upper bounds on both the first error event (burst error) and bit error probabilities for Viterbi decoding of convolutional codes. These bounds were derived using a signal flow chart technique for convolutional encoders. In 1995, Best et al. published a formula for the exact bit error probability for Viterbi decoding of the rate $R = 1/2$, memory $m = 1$ 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 $m = 2$ 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.

I. INTRODUCTION

In 1971, Viterbi [1] published a now classical upper bound on the bit error probability P_b for Viterbi decoding when convolutional codes are used to communicate over the binary symmetric channel (BSC). This bound was derived from the extended path weight enumerators obtained using a signal flow chart technique for convolutional encoders. Van de Meerberg [2] used a clever observation to tighten Viterbi's bound.

The challenging problem of deriving an expression for the exact bit error probability was first addressed by Morrissey in 1970 [3] for a suboptimum feedback decoding technique. Apparently, 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. [4]. They used a more general approach based on considering a Markov chain of the so-called metric states of the Viterbi decoder [5]. Their new method looked simpler to generalize to larger memory encoders and to other channel models, which was later done in a number of papers. In particular, the extension to the memory $m = 2$ convolutional encoder with generator matrix $G(D) = (1 + D^2 \ 1 + D + D^2)$ was given by Lentmaier et al. [6].

We use a different approach to derive the exact bit error probability for Viterbi decoding of minimal convolutional encoders when used to communicate over the BSC. A matrix

recurrent equation will be derived 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.

In this presentation we consider for notational convenience only rate $R = 1/2$ minimal convolutional feed-forward encoders realized in controller canonical form. The extension to rate $R = 1/c$ is trivial and to rate $R = b/c$ as well as feedback encoders straight-forward. Before proceeding we would like to emphasize that the bit error probability is an encoder property, not a code property.

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 at time t . If its initial value $W_0(\sigma)$ is known then the random process $W_t(\sigma)$ is a function of the random process of the received c -tuples \mathbf{r}_i , $i = 0, 1, \dots, t-1$. Thus, the ensemble $\{\mathbf{r}_i, i = 0, 1, \dots, t-1\}$ 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

$$P_b = \lim_{t \rightarrow \infty} \frac{E[W_t(\sigma = 0)]}{t} \quad (1)$$

assuming that this limit exists.

II. A RECURRENT EQUATION FOR THE INFORMATION WEIGHTS

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, $\sigma_t = u_{t-1}u_{t-2} \dots u_{t-m}$. 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 \mathbf{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,

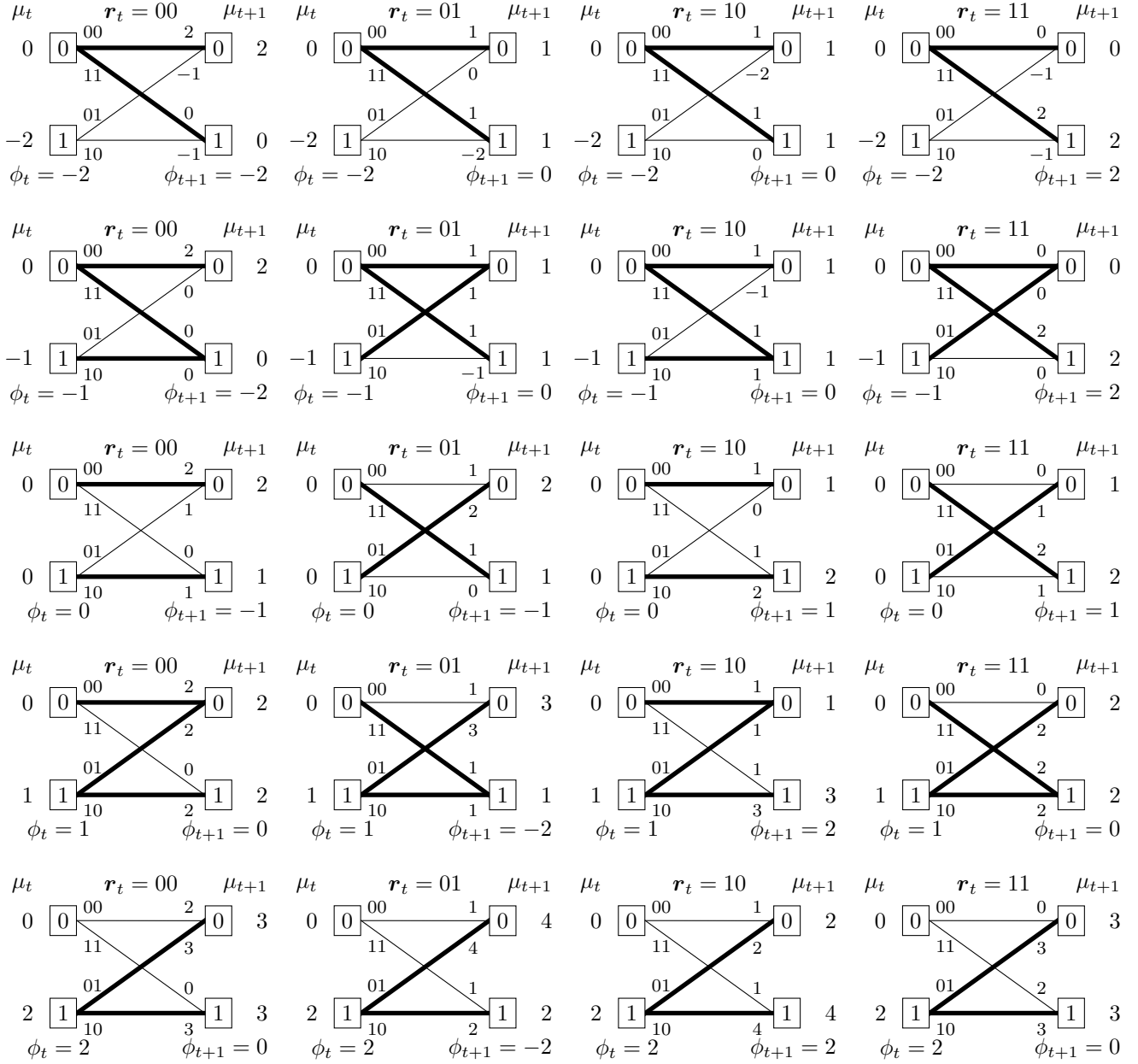


Fig. 1. The 20 different trellis sections for the $G(D) = (1 \ 1 + D)$ generator matrix.

we subtract the value $\mu_t(0)$ from $\mu_t(1), \mu_t(2), \dots, \mu_t(2^m - 1)$ and introduce the cumulative normalized branch metric vector

$$\begin{aligned} \phi_t &= \left(\phi_t(1) \ \phi_t(2) \ \dots \ \phi_t(2^m - 1) \right) \\ &= \left(\mu_t(1) - \mu_t(0) \ \mu_t(2) - \mu_t(0) \ \dots \ \mu_t(2^m - 1) - \mu_t(0) \right) \end{aligned} \quad (2)$$

For a memory $m = 1$ encoder we obtain the scalar

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

while for a memory $m = 2$ encoder we have the vector

$$\phi_t = \left(\phi_t(1) \ \phi_t(2) \ \phi_t(3) \right) \quad (4)$$

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 $\phi_t = -2, -1, 0, 1, 2$ and the four different received tuples $\mathbf{r}_t = 00, 01, 10, 11$. 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 coin-flipping.

The normalized cumulative metric Φ_t is a 5-state Markov

chain with transition probability matrix $\Phi = (\phi_{jk})$, where

$$\phi_{jk} = Pr\left(\phi_{t+1} = \phi^{(k)} \mid \phi_t = \phi^{(j)}\right) \quad (5)$$

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

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

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

$$\mathbf{p}_\infty \Phi = \mathbf{p}_\infty \quad (7)$$

and

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

For our $m = 1$ convolutional encoder we obtain

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

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 \mathbf{p}_t . Let \mathbf{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 \mathbf{w}_t as the following vector of $M2^m$ entries:

$$\mathbf{w}_t = \begin{pmatrix} w_t(\phi^{(1)}, \sigma = 0) & \dots & w_t(\phi^{(M)}, \sigma = 0) \\ w_t(\phi^{(1)}, \sigma = 1) & \dots & w_t(\phi^{(M)}, \sigma = 1) \\ \vdots & & \vdots \\ w_t(\phi^{(1)}, \sigma = 2^m - 1) & \dots & w_t(\phi^{(M)}, \sigma = 2^m - 1) \end{pmatrix} \quad (10)$$

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

$$\begin{cases} \mathbf{w}_{t+1} = \mathbf{w}_t A + \mathbf{b}_t \\ \mathbf{b}_{t+1} = \mathbf{b}_t \Pi \end{cases} \quad (11)$$

where A is an $M2^m \times M2^m$ nonnegative matrix 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 vector \mathbf{b}_t of length $M2^m$ describes the increments of the information weights. For simplicity, we choose the initial values

$$\mathbf{w}_0 = \mathbf{0} \quad (12)$$

and

$$\mathbf{b}_0 = \left(\underbrace{\mathbf{0}_{1,M} \ \mathbf{0}_{1,M} \ \dots \ \mathbf{0}_{1,M}}_{2^{m-1}} \ \underbrace{\mathbf{p}_\infty \ \mathbf{p}_\infty \ \dots \ \mathbf{p}_\infty}_{2^{m-1}} \right) \quad (13)$$

where $\mathbf{0}_{1,M}$ denotes the all-zero vector of length M and \mathbf{p}_∞ is the stationary probability distribution of the normalized cumulative metrics Φ_t .

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 $\mathbf{r}_t = 00$. Following the bold branches, we first copy with probability $Pr(\mathbf{r}_t = 00) = q^2$ 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 $\mathbf{r}_t = 00$. 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 (14) 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.

The second equation in (11) determines the dynamics of the information weight increments. We notice that

$$\sum_{i=0}^{2^m-1} A_{ij} = \Phi, \quad j = 0, 1, \dots, 2^m - 1 \quad (15)$$

Moreover, we should only have increments when entering the the states σ_{t+1} whose first digit is a 1. Thus, the first half of the entries of \mathbf{b}_t are 0s for $t = 0, 1, \dots$, and it follows that we can choose Π to be

$$\Pi = \begin{pmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \Phi \end{pmatrix} \quad (16)$$

In the next section we shall solve the recurrent matrix equation (11).

III. SOLVING THE RECURRENT EQUATION

By iterating the recurrent equation (11) and using the initial values (12) and (13) we obtain

$$\mathbf{w}_{t+1} = \mathbf{b}_0 A^t + \mathbf{b}_0 \Pi A^{t-1} + \dots + \mathbf{b}_0 \Pi^t \quad (17)$$

From (17) it follows that

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{\mathbf{w}_t}{t} &= \lim_{t \rightarrow \infty} \frac{\mathbf{w}_{t+1}}{t} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{j=0}^t \mathbf{b}_0 \Pi^j A^{t-j} \\ &= \lim_{t \rightarrow \infty} \frac{2}{t} \sum_{j=0}^{t/2} \mathbf{b}_0 \frac{\Pi^j A^{t-j} + \Pi^{t-j} A^j}{2} - \lim_{t \rightarrow \infty} \frac{\mathbf{b}_0 \Pi^{t/2} A^{t/2}}{t} \\ &= \lim_{t \rightarrow \infty} \mathbf{b}_0 \Pi^{t/2} A^{t/2} = \mathbf{b}_0 \Pi^\infty A^\infty \end{aligned} \quad (18)$$

where Π^∞ and A^∞ denote the limits of the sequences Π^t and A^t when t tends to infinity. We also used that, if a sequence is convergent to a finite limit, then it is Cesàro-summable to the same limit.

The limit Π^∞ can be written as

$$\Pi^\infty = \begin{pmatrix} \Phi^\infty & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi^\infty & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Phi^\infty \end{pmatrix} \quad (19)$$

where Φ^∞ is a matrix whose rows are identical and equal to the stationary distribution \mathbf{p}_∞ . Then (18) can be written as

$$\lim_{t \rightarrow \infty} \frac{\mathbf{w}_t}{t} = \mathbf{b}_\infty A^\infty \quad (20)$$

where

$$\mathbf{b}_\infty = \underbrace{(\mathbf{0}_{1,M} \ \mathbf{0}_{1,M} \ \dots \ \mathbf{0}_{1,M})}_{2^{m-1}} \underbrace{(\mathbf{p}_\infty \ \mathbf{p}_\infty \ \dots \ \mathbf{p}_\infty)}_{2^{m-1}} \quad (21)$$

We have the following important properties of $A = (a_{ij})$:

- Nonnegativity, that is, $a_{ij} \geq 0$, $0 \leq i, j \leq M(2^m - 1)$.
- For any convolutional encoder, A has a block structure, $A = (A_{ij})$, $i, j = 0, 1, \dots, 2^m - 1$, where the block A_{ij} corresponds to the transitions from $\sigma_t = i$ to $\sigma_{t+1} = j$.
- Summing over the blocks columnwise yields

$$\sum_{i=0}^{2^m-1} A_{ij} = \Phi, \quad j = 0, 1, \dots, 2^m - 1 \quad (22)$$

From (22) follows that

$$\mathbf{e}_L = (\mathbf{p}_\infty \ \mathbf{p}_\infty \ \dots \ \mathbf{p}_\infty) \quad (23)$$

satisfies

$$\mathbf{e}_L A = \mathbf{e}_L \quad (24)$$

and, hence, \mathbf{e}_L is a left eigenvector with eigenvalue $\lambda = 1$.

From the nonnegativity follows (Corollary 8.1.30 [7]) that $\lambda = 1$ is a maximal eigenvalue of A . Let \mathbf{e}_R be the right eigenvector corresponding to the eigenvalue $\lambda = 1$ and let

$$A = \begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} = \begin{matrix} & & & \sigma_{t+1} = 0 & & & & \sigma_{t+1} = 1 & & & & & \phi_{t+1} \\ & & & -2 & -1 & 0 & 1 & 2 & & -2 & -1 & 0 & 1 & 2 & \\ \sigma_t = 0 & & & \left(\begin{array}{cccc|cccc} q^2 & 0 & 2pq & 0 & p^2 & q^2 & 0 & 2pq & 0 & p^2 \\ q^2 & 0 & 3pq/2 & 0 & p^2/2 & q^2/2 & 0 & 3pq/2 & 0 & p^2 \\ 0 & 0 & q^2 & 0 & pq & 0 & pq & 0 & p^2 & 0 \\ 1 & 0 & 0 & q^2/2 & 0 & qp/2 & pq/2 & 0 & p^2/2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right. & & & & \\ \sigma_t = 1 & & & -2 & -1 & 0 & 1 & 2 & & -2 & -1 & 0 & 1 & 2 & \\ & & & \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & pq/2 & 0 & p^2/2 & q^2/2 & 0 & pq/2 & 0 & 0 & 0 \\ 0 & 0 & pq & 0 & p^2 & 0 & 0 & q^2 & 0 & pq & 0 \\ 1 & pq & 0 & q^2/2 + p^2 & 0 & pq/2 & pq/2 & 0 & q^2 + p^2/2 & 0 & pq \\ 2 & pq & 0 & q^2 + p^2 & 0 & pq & pq & 0 & q^2 + p^2 & 0 & pq \end{array} \right) & & & & \end{matrix} \quad (14)$$

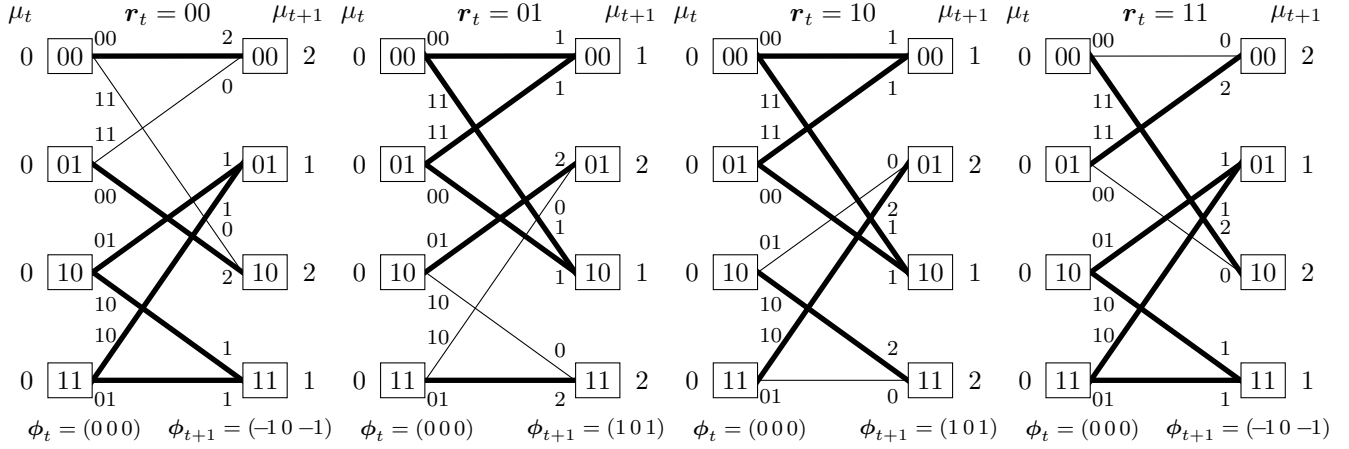


Fig. 2. Four different trellis sections of the in total 124 for the $G(D) = (1 + D^2 1 + D + D^2)$ generator matrix.

e_R be normalized such that $e_L e_R = 1$. If e_L is unique (up to normalization) then it follows (Lemma 8.2.7, statement (i) [7]) that

$$A^\infty = e_R e_L \quad (25)$$

Combining (20), (23), and (25) yields

$$\lim_{t \rightarrow \infty} \frac{\mathbf{w}_t}{t} = \mathbf{b}_\infty e_R (\mathbf{p}_\infty \mathbf{p}_\infty \dots \mathbf{p}_\infty) \quad (26)$$

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

$$\begin{aligned} P_b &= \lim_{t \rightarrow \infty} \frac{E[W_t(\sigma = 0)]}{t} = \lim_{t \rightarrow \infty} \frac{\sum_{i=1}^M \mathbf{w}_t(\phi^{(i)}, \sigma = 0)}{t} \\ &= \lim_{t \rightarrow \infty} \frac{\mathbf{w}_t(\sigma = 0) \mathbf{1}_{1,M}^T}{t} \end{aligned} \quad (27)$$

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 (26), or, equivalently, we multiply this vector by the vector $(\mathbf{1}_{1,M} \mathbf{0}_{1,M} \dots \mathbf{0}_{1,M})^T$. Then we obtain

$$P_b = (\mathbf{0}_{1,M} \mathbf{0}_{1,M} \dots \mathbf{0}_{1,M} \mathbf{p}_\infty \mathbf{p}_\infty \dots \mathbf{p}_\infty) e_R \quad (28)$$

In summary, for rate $R = 1/2$, memory m 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 \mathbf{p}_∞
- Construct the matrix A analogously to the memory $m = 1$ example given above and compute its right eigenvector e_R normalized according to $(\mathbf{p}_\infty \mathbf{p}_\infty \dots \mathbf{p}_\infty) e_R = 1$.
- Compute P_b using (28).

IV. EXAMPLES

First we consider the rate $R = 1/2$, memory $m = 1$ 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 \mathbf{p}_∞ is given by (9).

From the trellis sections in Fig. 1 we obtain the matrix A (14). Its normalized right eigenvector is

$$e_R = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{pq}{2} \\ \frac{2}{4pq} \\ \frac{2 - p + 4p^2 - 4p^3}{(2 + 7p - 12p^2 + 13p^3 - 12p^4 + 4p^5)} \\ \frac{2(2 - p + 4p^2 - 4p^3)}{1} \\ 1 \end{pmatrix} \quad (29)$$

Finally, inserting (9) and (29) into (28) yields

$$P_b = \frac{14p^2 - 23p^3 + 16p^4 + 2p^5 - 16p^6 + 8p^7}{(1 + 3p^2 - 2p^3)(2 - p + 4p^2 - 4p^3)} \quad (30)$$

which coincides with the bit error probability formula in [4].

Next we consider the rate $R = 1/2$, memory $m = 2$ convolutional encoder with generator matrix $G(D) = (1 + D^2 \ 1 + D + D^2)$. In Fig. 2 we show the four trellis sections for $\phi_t = (0 \ 0 \ 0)$. The corresponding metric states at time $t + 1$ are $\phi_{t+1} = (-1 \ 0 \ -1)$ and $(1 \ 0 \ 1)$.

Completing the set of trellis sections yields 31 different normalized metric states. Thus, we have 124 different trellis sections. The matrix A is a block matrix that consists of eight different nontrivial blocks corresponding to the eight branches in each trellis section. We obtain the 124×124 matrix

$$A = \begin{pmatrix} A_{00} & \mathbf{0}_{31,31} & A_{02} & \mathbf{0}_{31,31} \\ A_{10} & \mathbf{0}_{31,31} & A_{12} & \mathbf{0}_{31,31} \\ \mathbf{0}_{31,31} & A_{21} & \mathbf{0}_{31,31} & A_{23} \\ \mathbf{0}_{31,31} & A_{31} & \mathbf{0}_{31,31} & A_{33} \end{pmatrix} \quad (31)$$

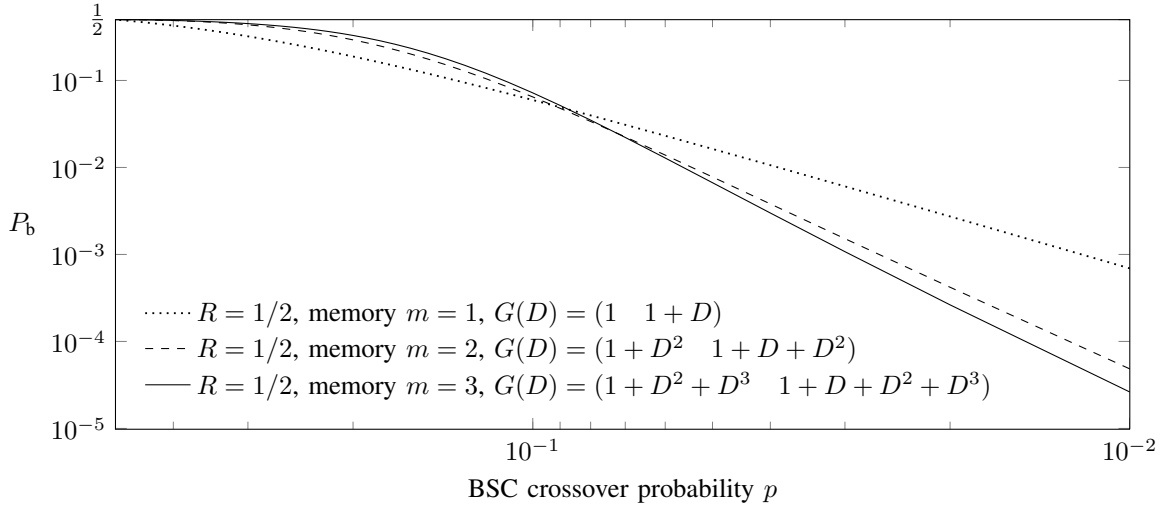


Fig. 3. Exact bit error probability for rate $R = 1/2$, memory $m = 1$ ($G(D) = (1 \ 1 + D)$), memory $m = 2$ ($G(D) = (1 + D^2 \ 1 + D + D^2)$), and memory $m = 3$ ($G(D) = (1 + D^2 + D^3 \ 1 + D + D^2 + D^3)$).

Similarly to (16) and (21) we have

$$\Pi = \begin{pmatrix} \Phi & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Phi & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Phi \end{pmatrix} \quad (32)$$

and

$$\mathbf{b}_\infty = (\mathbf{0}_{1,31} \ \mathbf{0}_{1,31} \ \mathbf{p}_\infty \ \mathbf{p}_\infty) \quad (33)$$

where Φ is the 31×31 transition probability matrix for the 31-state Markov chain for the normalized cumulative metrics Φ_t and \mathbf{p}_∞ is the stationary probability distribution for Φ_t .

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

$$P_b = 44p^3 + \frac{3519}{8}p^4 - \frac{14351}{32}p^5 - \frac{1267079}{64}p^6 - \frac{31646405}{512}p^7 + \frac{978265739}{2048}p^8 + \frac{3931764263}{1024}p^9 - \frac{48978857681}{32768}p^{10} + \dots \quad (34)$$

which coincides with the previously obtained result by Lentmaier et al. [6].

Finally, we consider the rate $R = 1/2$, memory $m = 3$ convolutional encoder with generator matrix $G(D) = (1 + D^2 + D^3 \ 1 + D + D^2 + D^3)$. If we consider all trellis sections we find that the number of normalized metric states is 433. Then we obtain the $433 \cdot 2^3 \times 433 \cdot 2^3$ matrix

$$A = \begin{pmatrix} A_{00} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{04} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} \\ A_{10} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{14} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} \\ \mathbf{0}_{433} & A_{21} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{25} & \mathbf{0}_{433} & \mathbf{0}_{433} \\ \mathbf{0}_{433} & A_{31} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{35} & \mathbf{0}_{433} & \mathbf{0}_{433} \\ \mathbf{0}_{433} & \mathbf{0}_{433} & A_{42} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{46} & \mathbf{0}_{433} \\ \mathbf{0}_{433} & \mathbf{0}_{433} & A_{52} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{56} & \mathbf{0}_{433} \\ \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{63} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{67} \\ \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{73} & \mathbf{0}_{433} & \mathbf{0}_{433} & \mathbf{0}_{433} & A_{77} \end{pmatrix} \quad (35)$$

where $\mathbf{0}_{433}$ is the 433×433 all-zero matrix.

Since this example is essentially more complex we computed the exact bit error probability (following the method in Section III) only numerically. The result is shown in Fig. 3 and compared with the curves for the previously discussed memory $m = 1$ and $m = 2$ encoders.

ACKNOWLEDGMENT

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

REFERENCES

- [1] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 5, pp. 751–772, Oct. 1971.
- [2] L. Van De Meeberg, "A tightened upper bound on the error probability of binary convolutional codes with Viterbi decoding," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 389–391, May 1974.
- [3] 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.
- [4] 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.
- [5] M. V. Burnashev and D. L. Kon, "Symbol error probability for convolutional codes," *Problems on Information Transmission*, vol. 26, no. 4, pp. 289–298, 1990.
- [6] 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.
- [7] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, Feb. 1990.