Improved Error Exponent for Time-Invariant and Periodically Time-Variant Convolutional Codes

Nadav Shulman, *Student Member, IEEE*, and Meir Feder, *Fellow, IEEE*

***Abstract*—An improved upper bound on the error probability (first error event) of *time-invariant* convolutional codes, and the resulting error exponent, is derived in this paper. The improved error bound depends on both the delay of the code K and its width (the number of symbols that enter the delay line in parallel) b. Determining the error exponent of time-invariant convolutional codes is an open problem. While the previously known bounds on the error probability of time-invariant codes led to the block-coding exponent, we obtain a better error exponent (strictly better for $b > 1$). In the limit $b \to \infty$ our error exponent equals the Yudkin–Viterbi exponent derived for *time-variant* convolutional codes. These results are also used to derive an improved error exponent for *periodically time-variant* codes.**

***Index Terms*—Convolutional codes, error exponent, error probability, periodically time-variant codes, time-invariant codes, Yudkin–Viterbi exponent.**

I. Introduction

CONVOLUTIONAL codes, first introduced by Elias [4], are used in numerous communication systems. A linear binary convolutional encoder [5], [17] is a finite-state machine consisting of a $b \cdot K$-bits shift register and n linear output functions. At each time instance a new information vector $\boldsymbol{u}_t = (u_t^1, u_t^2, \cdots, u_t^b)$ of b bits is pushed into the register. Then, n output bits $\boldsymbol{o}_t = (o_t^1, o_t^2, \cdots, o_t^n)$ are calculated by some linear combinations of bits in the register. These linear combinations, denoted $\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_n$, determine the specific convolutional code. The rate of the code R is the ratio between the number of input symbols and the number of output symbols per use, i.e., $R = b/n$. The quantity $\nu = bK = RnK$, the memory size of the code, is often called the constraint length.

In specific terms, let the content of the register at time t be $\boldsymbol{U}_t = (\boldsymbol{u}_t, \boldsymbol{u}_{t-1}, \cdots, \boldsymbol{u}_{t-K+1})$, i.e., $\boldsymbol{U}_t$ is a binary vector of length ν. Each linear combination $\boldsymbol{g}_i$ is represented by a binary row vector of length ν and defines the following $\nu \times n$ binary matrix G:

$$G^T = \begin{pmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_n \end{pmatrix}. \tag{1}$$

Manuscript received January 12, 1999; revised April 20, 1999.

The authors are with the Department of Electrical Engineering–Systems, Tel-Aviv University, Tel-Aviv 69978 Israel (e-mail: {shulman; meir}@eng.tau.ac.il).

Communicated by A. M. Barg, Associate Editor for Coding Theory.
Publisher Item Identifier S 0018-9448(00)00069-9.

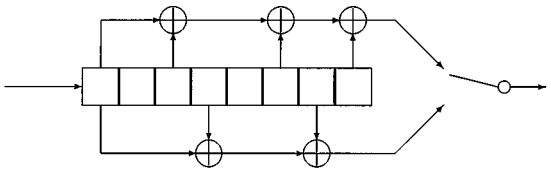

Fig. 1 An example for rate $1/2$ convolutional code

Then, the corresponding n output bits at time t are

$$\boldsymbol{o}_t = \boldsymbol{U}_t G. \tag{2}$$

Fig. 1 shows an example of a rate $1/2$ convolutional code with $b = 1$, $n = 2$, $K = 8$, $\boldsymbol{g}_1 = (10100101)$, and $\boldsymbol{g}_2 = (10010010)$. In principle, the linear combinations $\boldsymbol{g}_i$ can vary with time. General time-varying codes are seldom used, if at all. Mostly, in practice, time-invariant codes where the $\boldsymbol{g}_i$'s are fixed, are used. In the recent years, though, periodically time-variant codes where the $\boldsymbol{g}_i$'s vary periodically with time also become popular.

There are several algorithms to decode a convolutional code [6], [7]. The best decoder in the sense of minimizing the error probability (but not the bit-error rate) is the maximum-likelihood (ML) decoder which is usually implemented by the Viterbi algorithm [16]. The complexity of the Viterbi decoder is proportional to 2^ν [18, p. 374].

In this paper, we analyze the error exponent of time-invariant and periodically time-variant convolutional code. The definition of the error exponent of convolutional codes is somewhat more complicated than the corresponding definition for block codes. In block codes, the error exponent is defined as $\lim_{N\to\infty} -(1/N) \log P_e$ where N is the block length and P_e is the probability of making an error in decoding a codeword of size N. Without assuming any particular structure, the decoding complexity of a block code is proportional to 2^{RN}, the number of codewords. Now, as noted above, the decoding complexity of a convolutional code without assuming any particular structure is proportional to $2^\nu = 2^{RnK}$. Hence, the adopted reasonable definition of the error exponent for convolutional codes is $\lim_{K\to\infty} -(1/nK) \log P_e$. Another issue is the definition of P_e. Since the codewords in a convolutional code can be infinitely long, the common definitions used for P_e are either the probability of the first error event, the probability of error of a finite frame, or the expected fraction of errors. In this paper we define P_e to be the probability of the first error event, whose exponential behavior is essentially the same as the frame error probability.

Based on practical experience, convolutional codes seem to be better than block codes with block length $N = nK$. How-

ever, theoretically, so far the better error performance bounds were shown only for time-varying codes. In the classical work [16], [18], [20], $\overline{P}_e$, the average error probability over the ensemble of time-variant codes, has been upper-bounded, and the resulting error exponent is given by

$$\lim_{K\to\infty} -\frac{1}{nK}\log \overline{P}_e = E_{\rm YV}(R) \tag{3}$$

where $E_{\rm YV}(R)$ is the Yudkin–Viterbi exponent

$$E_{\rm YV}(R) = \max_{Q} E_0(\rho_0, Q) \triangleq \max_{Q} \log_2 \sum_{y} \left[\sum_{x} Q(x)P(y|x)^{1/(1+\rho_0)}\right]^{1+\rho_0} \tag{4}$$

and $\rho_0 = \min(1, \rho')$, ρ' is the largest solution of $\rho' R = E_0(\rho', Q)$, Q is the channel's symbols input distribution, and $P(y|x)$ is the channel's transition probability. It is easy to see that $E_r(R) \le E_{\rm YV}(R)$, where

$$E_r(R) = \max_{0\le\rho\le 1} [E_0(\rho) - \rho R]$$

is the random coding error exponent for block codes [8]. For the binary-symmetric channel (BSC) considered in this paper the maximizing prior Q is uniform at any rate R, and so the dependency of $E_0(\cdot)$ on Q will be omitted.

As for time-invariant convolutional codes, it was previously proved that they attain the channel capacity, and that the error exponent of a random time-invariant code in BSC is at least as good as the random-coding error exponent of block codes [21], [14], [10]. A similar result for a general discrete memoryless channel (DMC) was shown in [22]. In addition, it was claimed [23], but without a proof, that for large b the error exponent of time-invariant convolutional codes is $E_{\rm YV}(R)$. Thus the determination of the error exponent of time-invariant convolutional codes is still open and considered here.

The main result of this paper is an improved upper bound for the average error probability (first error event) over the ensemble of random time-invariant convolutional codes in BSC. This bound leads to a proof of the claim made in [23] that for large b the time-invariant codes attain the Yudkin–Viterbi exponent. Furthermore, it shows the behavior of the error exponent for general values of the memory K and the width b. The derivation for time-invariant codes is extended, and leads to an improved upper bound for the average error probability over the ensemble of random periodically time-variant codes. The resulting error exponent is given for each period T, and equals the exponent derived for fixed codes, but with b replaced by bT. Clearly, as $T\to\infty$ the exponent approaches $E_{\rm YV}(R)$, but our result specifically shows how the Yudkin–Viterbi exponent is approached.

It should be noted that our analysis can be extended to general DMC's, but it becomes more cumbersome. To make this paper clearer and more concise we concentrate on the BSC channel, and refer to [13] for the more general analysis. In this respect we note that linear convolutional codes, like all linear codes, may have some drawbacks when used over general, nonadditive, DMC's, and may not attain the optimal error exponent.

Another interesting quantity of convolutional codes is the free distance. For time-variant codes a lower bound on the free distance, the "Costello bound," which is better than the corresponding minimal distance of block codes, was shown in [3]. It was later proved that this bound is also attained (see [24]) for time-invariant convolutional codes with $b \gg 1$, and in some specific cases (see [25] and [2]) it is attained for b equals at least 2 (but not necessarily large). This may imply that a similar result can be obtained for the error exponent. Unfortunately, the error exponent we obtain is poorer than the Yudkin–Viterbi exponent for small values of b. As a matter of fact, even after our work, the determination of the *optimal* random coding error exponent for time-invariant and periodically time-variant convolutional codes is still open.

II. Preliminaries: Useful Lemmas

In this section we provide several technical lemmas that will be required for the proof of our main result in Section III. This section may be omitted in first reading, as we shall refer to the necessary results in the proof of the main result.

Lemma 1: Let $\mathcal{C}$ be a random binary block code of length N and M codewords with the property that for any $i \ne j$ and $\boldsymbol{x} \in \{0, 1\}^N$ we have

$$\Pr\left(\boldsymbol{c}(i) \oplus \boldsymbol{c}(j) = \boldsymbol{x}\right) = 2^{-N} \tag{5}$$

where $\oplus$ denotes bit-by-bit exclusive-or. Then the average error probability of a maximum-likelihood (ML) decoder for this code when used in BSC can be bounded by

$$\overline{P}_e \le (M-1)^{\rho} 2^{-NE_0(\rho)} \tag{6}$$

for any $0 \le \rho \le 1$.

This Lemma is well known. For example, a proof of a more general result is given in [13] and [15]. Also, this Lemma can actually follow from classical results in [19] and [8]. The proof of the lemma is based on the fact that if a random vector is added to all codewords, the code's performance does not change. Yet, the resulting new random code, which has more randomness, has the property that the codewords are pairwise-independent, and so its expected error probability is upper-bounded by Gallager's random coding bound [8].

The following technical Lemmas correspond to properties of random binary matrices.

Lemma 2: Let A be an $m \times n$, $n > m$ binary matrix, with a full rank (i.e., its rank is m), and let G be an $n \times k$ uniformly distributed random binary matrix. Then the $m \times k$ random matrix AG is uniformly distributed.

Proof: Since A has full rank and $n > m$, for any vector $\boldsymbol{y}$ of length m, there are 2^{n-m} solution to the equation $A\boldsymbol{x} = \boldsymbol{y}$. Hence, if $\boldsymbol{x}$ is uniformly distributed random vector, we have

$$\Pr(\boldsymbol{y} = \boldsymbol{u}) = \Pr\left(A\boldsymbol{x} = \boldsymbol{u}\right) = 2^{-n}2^{n-m} = 2^{-m}$$

i.e., $\boldsymbol{y}$ is uniformly distributed. Applying the above for each column of G leads to the lemma. □

Lemma 3: Let $\{\boldsymbol{u}_i\}_{i=1}^{\infty}$ be binary independent and identically distributed (i.i.d.) random vectors of length b, with uniform distribution over $\{0, 1\}^b$, and suppose that $\boldsymbol{u}_1 \neq \mathbf{0}$. Then the probability that the first $r-1$ rows in the matrix

$$\begin{pmatrix} \boldsymbol{u}_1 & 0 & \cdots & 0 \\ \boldsymbol{u}_2 & \boldsymbol{u}_1 & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ \boldsymbol{u}_K & \boldsymbol{u}_{K-1} & \cdots & \boldsymbol{u}_1 \\ \vdots & & \ddots & \vdots \\ \boldsymbol{u}_{r-1} & \boldsymbol{u}_{r-2} & \cdots & \boldsymbol{u}_{r-K} \\ \boldsymbol{u}_r & \boldsymbol{u}_{r-1} & \cdots & \boldsymbol{u}_{r-K+1} \end{pmatrix} \tag{7}$$

will be independent but the rth row will be dependent is

$$\begin{aligned} \Pr \leq & (r \text{ dependent}, r-1 \text{ independent}) \\ \leq & \begin{cases} 0, & \text{for } r \leq K \\ 2^{r-1}2^{-bK}, & \text{for } K < r \leq bK+1 \\ 0, & \text{for } bK+1 < r. \end{cases} \end{aligned} \tag{8}$$

Proof: Since $\boldsymbol{u}_1 \neq 0$ the first K rows are linearly independent (with probability 1). The rows' length is bK, hence $bK+1$ rows are linearly dependent (with probability 1).

For the intermediate range, we can upper-bound the desired probability by the probability that the rth row is a linear combination of the $r-1$ previous rows. There are 2^{r-1} different linear combinations and the probability that a specific combination equals the rth row is 2^{-bK}. This can be seen by looking at the last line of the matrix from right to left. The probability that $\boldsymbol{u}_{r-K+1}$ equals a specific combination of the elements above it, is 2^{-b}. Now moving one element to the left, and looking at $\boldsymbol{u}_{r-K+2}$, there is still a probability that 2^{-b} equals a specific combination of the elements above it, because it is independent of the previous equality. We can continue in this manner up to the leftmost element $\boldsymbol{u}_r$. Using the union bound we get the desired bound.

III. Main Result

In this section we bound the average error probability and find an error exponent of a randomly selected time-invariant convolutional code operating in a BSC. The random ensemble of the time-invariant codes, from which the code is chosen, is defined by all possible n linear combinations $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n$. This requires $n \cdot b \cdot K$ random, uniformly distributed, bits. We will provide an upper bound and an error exponent expression for the average probability of a first error event $\overline{P}_e$ and, consequently, for the average frame error probability $\overline{P}_{\text{frame}}$. Recall that the probability of a first error event, at time t, is the *conditional* probability to diverge from the correct path, given that no error occurred until that time[1] while the frame error probability is the error probability of the block code generated by a finite transmission of the convolutional code, with zero padding at the end. Unfortunately, our derivation cannot lead to a bound on the fraction of decoding errors since it cannot assume that in case where the decoder diverges from the correct path it will later return to it. Worse than that, our derivation does not exclude the possibility that from some time point on all bits will be wrongly decoded.

In the sequel we denote the transmitted (infinite) information word by $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots)$, and so the corresponding transmitted codeword (with an initial all zero register content) is given by

$$\boldsymbol{c}(\boldsymbol{U}) = \begin{pmatrix} \boldsymbol{u}_1 & 0 & \cdots & 0 \\ \boldsymbol{u}_2 & \boldsymbol{u}_1 & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ \boldsymbol{u}_K & \boldsymbol{u}_{K-1} & \cdots & \boldsymbol{u}_1 \\ \vdots & & \ddots & \vdots \\ \boldsymbol{u}_t & \boldsymbol{u}_{t-1} & \cdots & \boldsymbol{u}_{t-K+1} \\ \vdots & & & \vdots \end{pmatrix} G = \begin{pmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_2 \\ \vdots \\ \boldsymbol{U}_K \\ \vdots \\ \boldsymbol{U}_t \\ \vdots \end{pmatrix} G \tag{9}$$

where G was defined in (1). Note that $\boldsymbol{c}(\boldsymbol{U})$ is a binary matrix with n columns and an infinite number of rows.

Our proof analyzes a suboptimal decoding procedure, where the information symbol (b bits) $\boldsymbol{u}_t$, at each time point t, is decoded based on limited length of future observed data. This limited length is, at most, $n\tau$, where τ is an arbitrary parameter, discussed later. The decoder also assumes that no error occurred so far, and so it assumes that $(\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_{t-1})$ and hence the current register content (at time $t-1$) is accurately known.

Let

$$\boldsymbol{V} = (\cdots \boldsymbol{v}_t, \boldsymbol{v}_{t+1}, \boldsymbol{v}_{t+2} \cdots)$$

and

$$\boldsymbol{W} = (\cdots \boldsymbol{w}_t, \boldsymbol{w}_{t+1}, \boldsymbol{w}_{t+2} \cdots)$$

be two different information (message) words, where $\boldsymbol{v}_i = \boldsymbol{w}_i = \boldsymbol{u}_i$ for $i < t$, i.e., $\boldsymbol{V}$ and $\boldsymbol{W}$ both fit to the decoded symbols until time $t-1$. If we also have $\boldsymbol{v}_t = \boldsymbol{w}_t$ it is not important which of the two words the decoder has chosen because it tries to decode only $\boldsymbol{u}_t$. In this case, we say that $\boldsymbol{V}$ and $\boldsymbol{W}$ are *friends.* If $\boldsymbol{v}_t \neq \boldsymbol{w}_t$ we say that $\boldsymbol{V}$ and $\boldsymbol{W}$ are *rivals.* On the trellis diagram of the code, rivals at time t are paths that split at that time, while friends stay the same at least until time t.

Specifying further the decoder we analyze, it calculates the likelihood of $\boldsymbol{V}$ and $\boldsymbol{W}$ along a *limited* observed future, chosen so that both codewords, $\boldsymbol{c}(\boldsymbol{V})$ and $\boldsymbol{c}(\boldsymbol{W})$, beginning from time t, still satisfy the conditions of Lemma 1, i.e., the exclusive-or (xor) of the two codewords is uniformly distributed. We call this length the *comparison length.* The comparison length, which

[1]We will actually provide a stronger result and bound the error to diverge from the path at time t given that we were on the correct path at time $t-1$, regardless of what happened before that. This error event is called *an error burst that starts at time t* [10].

depends on t and the words $\boldsymbol{V}$ and $\boldsymbol{W}$, turns out to be $n \cdot L_t(\boldsymbol{V}, \boldsymbol{W})$ (i.e., $L_t(\boldsymbol{V}, \boldsymbol{W})$ time steps), where $L_t(\boldsymbol{V}, \boldsymbol{W})$ is the maximum value of ℓ (but not larger than τ) such that the row vectors of the matrix

$$A = \begin{pmatrix} \boldsymbol{V}_t \oplus \boldsymbol{W}_t \\ \boldsymbol{V}_{t+1} \oplus \boldsymbol{W}_{t+1} \\ \vdots \\ \boldsymbol{V}_{t+\ell-1} \oplus \boldsymbol{W}_{t+\ell-1} \end{pmatrix} \tag{10}$$

are linearly independent. To see this, we first note that since the code is linear, $\boldsymbol{c}(\boldsymbol{V}) \oplus \boldsymbol{c}(\boldsymbol{W}) = \boldsymbol{c}(\boldsymbol{V} \oplus \boldsymbol{W})$. The portion of $\boldsymbol{c}(\boldsymbol{V} \oplus \boldsymbol{W})$ between t and $t+\ell-1$ is given by the matrix AG where A is the matrix in (10) and G is the generating (random) matrix of the code. Now, since A has a full rank, Lemma 2 implies that $\boldsymbol{c}(\boldsymbol{V} \oplus \boldsymbol{W})$ is uniformly distributed, as required. Note that the comparison length depends on the xor of the two words, i.e., $L_t(\boldsymbol{V}, \boldsymbol{W}) = L_t(\boldsymbol{V} \oplus \boldsymbol{W})$, and it is 0 if and only if the two words are friends.

Let $S_t(\boldsymbol{V}|\ell)$ be the likelihood of the word $\boldsymbol{V}$ accumulated from time t along a path of length ℓ. Our decoder decides at time t that $\boldsymbol{v}_t$ was sent if it is the symbol at time t of a codeword $\boldsymbol{V}$ that fits all the previously decoded symbols, and in addition

$$S_t(\boldsymbol{V}|L_t(\boldsymbol{V} \oplus \boldsymbol{W})) > S_t(\boldsymbol{W}|L_t(\boldsymbol{V} \oplus \boldsymbol{W}))$$

for all rivals $\boldsymbol{W}$ of $\boldsymbol{V}$ at time t. In other words, $\boldsymbol{V}$ *overcomes* all its rivals at time t. The existence of such a word $\boldsymbol{V}$ is not guaranteed. A situation may occur that $\boldsymbol{V}$ overcomes $\boldsymbol{W}$, $\boldsymbol{W}$ overcomes $\boldsymbol{U}$ but $\boldsymbol{U}$ overcomes $\boldsymbol{V}$. The reason is that different pairs of words may be compared along paths with different lengths. In this situation we declare an error. It is also possible that there exist many words that overcome all their rivals. However, this can only be if these words were not compared between them, i.e., they are friends. Thus the decoder can decode the symbol at time t since it is the same for all friends.

If $\boldsymbol{W}$ and $\boldsymbol{W}'$ are rivals of $\boldsymbol{V}$ such that

$$L_t(\boldsymbol{V}, \boldsymbol{W}) = L_t(\boldsymbol{V}, \boldsymbol{W}')$$

and $\boldsymbol{W}_i = \boldsymbol{W}'_i$ for $t \le i < t + L_t(\boldsymbol{V}, \boldsymbol{W})$ then we say that $\boldsymbol{W}$ and $\boldsymbol{W}'$ are *identical* with respect to $\boldsymbol{V}$ at time t. Clearly, if $\boldsymbol{V}$ *overcomes* $\boldsymbol{W}$, it also overcomes all the identicals of $\boldsymbol{W}$. Effectively, all identical rivals can be considered as a single rival.

Let $M_\ell(\boldsymbol{V}, t)$ be the number of nonidentical rivals of $\boldsymbol{V}$ at time t such that the comparison length is $n \cdot \ell$. Since $L_t(\cdot)$ is a function of the xor between words and the code is linear, then $M_\ell(\boldsymbol{V}, t)$ is independent of $\boldsymbol{V}$. Since a time-shift of any information word is another information word, it is independent of t as well. Thus $M_\ell(\boldsymbol{V}, t) = M_\ell$. Note that for finite transmission of a frame of length N, $M_\ell(\boldsymbol{V}, t)$ depends on t (decreases as t approaches N) due to the constrains at the end (zero padding of the information word). Still, in this case $M_\ell(\boldsymbol{V}, t) \le M_\ell(\boldsymbol{V}, 0) \le M_\ell$.

As will be seen below, to get an upper bound on the average error probability of our decoder we need to specify M_ℓ, or to find an upper bound for it. For this, observe that since the rival can be any sequence (uniformly distributed) except that it must satisfy $\boldsymbol{w}_i = \boldsymbol{v}_i$ for $i < t$ and $\boldsymbol{w}_t \neq \boldsymbol{v}_t$, then the matrix in (10) has the structure of the matrix (7) in Lemma 3. With $r = \ell + 1$, the number of such matrices is $2^{br} = 2^{b(\ell+1)}$. M_ℓ is the number of matrices of that form with the additional property that the first $r-1$ rows are independent, but the rth row depends on the previous rows. A bound on the probability that a matrix of the form (7) will have this property is given by (8). Thus in order to get a bound on M_ℓ we should multiply the bound on the probability by the *total* number of matrices of that form. For $\ell = \tau$ we should consider all matrices of size τ which have a full rank. A simple upper bound for this number is $M_\tau \le 2^{b\tau}$ for $\tau \le Kb$, and 0 otherwise. Combining the above, we hence get

$$M_\ell \le \begin{cases} 0, & \text{if } \ell < K \text{ and } \ell < \tau \\ 2^{b(\ell-K)+b+\ell}, & \text{if } K \le \ell \le Kb \text{ and } \ell < \tau \\ 2^{b\ell}, & \text{if } \ell = \tau \le Kb \\ 0, & \text{if } \ell > Kb \text{ or } \ell > \tau. \end{cases} \tag{11}$$

At this point we are ready to present a general formula for an upper bound on the expected error probability (first error event) associated with our decoder. At each time point, the portion of size $n\ell$ of the true codeword and all its nonidentical rivals with comparison length ℓ essentially form a block code with block size $n\ell$ and with at most $M_\ell + 1$ words. Due to the way the decoder was constructed, we can apply Lemma 1, and get the upper bound $M_\ell^\rho 2^{-n\ell E_0(\rho)}$ on the expected (over the ensemble) error probability of this block code. Now, we have a set of such block codes, associated with all possible comparison lengths. We should find the probability of not making an error in any of these block codes. This probability can be bounded by combining the union bound with Lemma 1, leading to the following upper bound on the average probability of error in decoding at time t, given that no error has occurred until that time[2] (i.e., the first error event probability):

$$\overline{P}_e \le \sum_{\ell=0}^{\tau} M_\ell^\rho 2^{-n\ell E_0(\rho)} \tag{12}$$

where $0 \le \rho \le 1$. Note that in principle we could have used a different ρ for each term in the summation. This upper bound holds for any t and it is independent of the transmitted word $\boldsymbol{V}$, because the bound we have for $M_\ell(\boldsymbol{V}, t)$ is independent of $\boldsymbol{V}$ and t.

Substituting (11) in (12) leads to (13) (at the bottom of the following page).

Setting $\tau = K$ in (13), forces the comparison length to be K, and we get

$$\overline{P}_e \le 2^{\rho bK} 2^{-nKE_0(\rho)} = 2^{-nK(E_0(\rho)-\rho R)} \tag{14}$$

for any $0 \le \rho \le 1$. This is the well known block-coding lower bound on the error exponent of time-invariant convolutional codes [21], [14]. For $K = 1$ the convolutional code is a block code, and hence the error exponent is tight [9].

We next show how (13) leads to a better bound for $b > 1$ and $K > 1$. In principle, we wish to evaluate (13) for an optimal value of τ. A good choice, although not necessarily optimal, is $\tau = b(K-1)$. The motivation for making this choice is that the bound in (11) which is used to upper-bound M_ℓ in each term of (13) is poor (greater than $2^{b\ell}$) for $\ell > b(K-1)$. Thus choosing

[2]Since we search for the decoded codeword only from the group of codewords with the same prefix until $t-1$ as the true word, we must assume no error occurred until now.

$\tau > b(K-1)$ only increases the bound on the error probability. Substituting $\tau = b(K-1)$ in (13) we get

$$\overline{P}_e \le \sum_{\ell=K}^{b(K-1)} 2^{\rho(b(\ell-K)+b+\ell)} 2^{-n\ell E_0(\rho)} = 2^{-\rho n(K-1)R} \sum_{\ell=K}^{b(K-1)} 2^{-n\ell[E_0(\rho)-\rho R(1+1/b)]} \quad (15)$$

for any $0 \le \rho \le 1$. Notice that for this choice of τ, $2^{b\tau} = 2^{b(\ell-K)+b+\ell}$ for $\ell = b(K-1)$.

The sum in (15) can be further upper-bounded by bK times the largest term in the sum. Define $R' = R(1+1/b)$ and let ρ' be the largest solution of the equation $E_0(\rho) = \rho R'$. Utilizing the properties of $E_0(\rho)$ (see [8]), it is easy to see that for $\rho \le \rho'$ the first term in the sum is the largest, and so

$$\overline{P}_e \le 2^{-nK[E_0(\rho)-\rho R(1/b+1/K)-(\log bK)/nK]}. \quad (16)$$

For $\rho \ge \rho'$ the last term is the largest which yields

$$\overline{P}_e \le 2^{-nK[b((K-1)/K)(E_0(\rho)-\rho R)-(\log bK)/nK]}. \quad (17)$$

The bounds above hold for any b and K. Note that under the constraint that $bK = \nu$ is fixed, (16) is optimized by $b = K$ which may indicate the tradeoff between the delay and the width of the code.

Combining(16), (17) and (14) and taking $K \to \infty$ yields the following lower bound on the error exponent $E_c(R)$:

$$E_c(R) = \max\left\{ \max_{0\le\rho\le\min(\rho',1)} \left(E_0(\rho) - \frac{1}{b}\rho R\right), \max_{\rho'\le\rho\le 1} b(E_0(\rho)-\rho R)\right\}. \quad (18)$$

When $\rho' > 1$ we set the right term to be zero. This bound depends on both the rate R and b.

In Fig. 2 the bound above, $E_c(R)$, is plotted for $b = 2, 4$, and 10 assuming that the channel is BSC with transition probability $\varepsilon = 0.1$. For comparison we also plotted the block coding error exponent $E_r(R)$ (which is $E_c(R)$ for $b = 1$) and the Yudkin–Viterbi exponent $E_{\rm YV}(R)$ of time-variant convolutional codes (which is $E_c(R)$ for $b \to \infty$).

More explicit expressions for $E_c(R)$, at least for some region of R, can be obtained by evaluating ρ' and then optimizing with respect to (w.r.t.) ρ. A simple analysis shows the following.

- For $0 \le R \le \frac{b}{b+1} R_0$, i.e., $R' \le R_0 = E_0(1)$ we have $\rho' \ge 1$ and $E_c(R) = R_0 - \frac{1}{b} R$.
- For $\frac{b}{b+1} C \le R < C$ we have $\rho' = 0$ and $E_c(R) = bE_r(R)$.

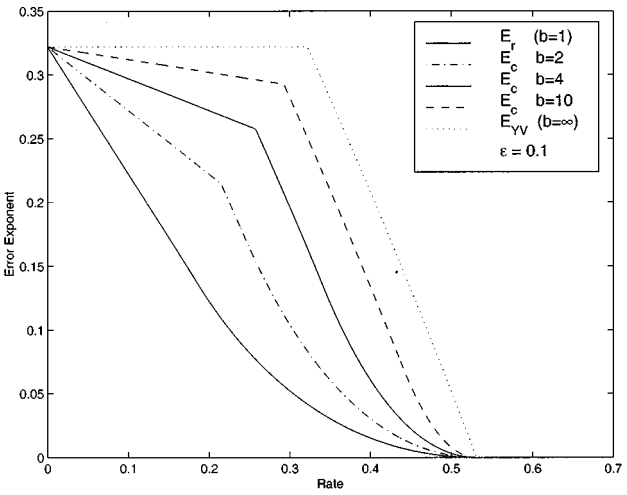

Fig. 2 $E_C(R)$ with $b = 1, 2, 4, 10, \infty$ for a BSC with $\epsilon = 0.1$

- For $\frac{b}{b+1} R_0 < R < \frac{b}{b+1} C$, i.e., $R_0 \le R' \le C$, ρ' varies between 0 and 1.
- At the middle region, if in addition

$$\frac{1}{b} R < \frac{\partial E_0(\rho)}{\partial \rho}\Big|_{\rho=\rho'} < R$$

then we have $E_c(R) = R\rho'$. (In this case the two expressions in (18) are equal.)

We see that for any R and b, $E_c(R) \le bE_r(R)$ and $E_c(R) \le E_{\rm YV}(R)$. On the other hand, for $b = 1$ we have $E_c(R) = E_r(R)$, and for $b > 1$, $E_c(R) > E_r(R)$ for any $R > 0$. For $b \gg 1$ ($R \approx R'$), we have

$$E_c(R) \approx R \min(\rho', 1) \approx E_0(\min(\rho', 1)).$$

Hence $\lim_{b\to\infty} E_c(R) = E_{\rm YV}(R)$, i.e., our bound on the error exponent of time-invariant convolutional codes approaches the Yudkin–Viterbi bound.

So far we have determined the error exponent for a first error event. This result immediately provides the error exponent for the frame error probability $\overline{P}_{\rm frame}$. The frame error probability is associated with the case where the convolutional code operates as a block code, transmitting a finite number of, say, N information symbols. In this mode, after the N information symbols are sent, zeros are pushed into the register until it is cleared. Clearly, the average probability of making an error in decoding the entire vector of N information symbols can be simply bounded by

$$\overline{P}_{\rm frame} \le N \cdot \overline{P}_e \quad (19)$$

$$\overline{P}_e \le \begin{cases} \sum_{\ell=K}^{\tau-1} 2^{\rho(b(\ell-K)+b+\ell)} 2^{-n\ell E_0(\rho)} + 2^{\rho b\tau} 2^{-n\tau E_0(\rho)}, & \text{for } \tau \le Kb \\ \sum_{\ell=K}^{Kb} 2^{\rho(b(\ell-K)+b+\ell)} 2^{-n\ell E_0(\rho)}, & \text{for } \tau > Kb \end{cases} \quad (13)$$

where $\overline{P}_e$ is the average probability of the first error event. Now, the true rate of the code in this mode, is $R_c = Nb/(N+K-1)n$. Since N can be chosen so that while $N \gg K$, it is still subexponential w.r.t. K (e.g., $N = K^2$), the exponential behavior of $\overline{P}_{\text{frame}}$ and $\overline{P}_e$ are the same, yet the effective rate $R_c \to R$ as $K \to \infty$.

Summarizing the above, we proved in effect the following theorem which is the main result of the paper.

Theorem 1: The average error probability (first error event and frame error) over the random ensemble of time-invariant convolutional code, used over a BSC, is upper-bounded by (18). At any choice of width b the error exponent is positive for any $R < C$. For $b > 1$ the error exponent is better than the block-coding error exponent. As $b \to \infty$, the error exponent approaches the Yudkin–Viterbi error exponent of time-varying convolutional codes.

IV. Periodically Time-Varying Codes

Periodically time-varying convolutional codes have drawn some attention in the recent years [11], [12]. One reason for this interest is the fact that such codes are obtained from the "tail-biting" trellis diagram of good block codes [1]. It is claimed that the performance of such codes is better than time-invariant codes, yet their decoding complexity and design complexity are similar. In this section we explore their error exponent which follows our analysis of time-invariant codes.

In specific terms, periodically time-varying convolutional codes with period T are codes where the matrix G of (1) is periodically time-depended, i.e., the linear functionals $\boldsymbol{g}_i(t)$ depend on the time t, and satisfy $\boldsymbol{g}_i(t) = \boldsymbol{g}_i(t+T)$. The resulting random ensemble of periodic convolutional codes requires $bmKT$ random bits. Each b/n periodically time-variant code, with period T, can be represented as a fixed bT/nT convolutional code (see [12]), but not vice versa, as the class of fixed $b'/n' = bT/nT$ convolutional codes is larger and contains $bmKT^2$ different codes. Also, the decoding complexity of the periodic codes is 2^{bK}, which is independent of T and equals the decoding complexity of b/n fixed codes, but it is smaller than $2^{b'K} = 2^{bTK}$, the decoding complexity of general bT/nT fixed codes.

Clearly, as the period T becomes large, the class of periodically time-varying codes becomes closer to the general class of time-varying codes. Thus as $T \to \infty$, the error exponent of the periodically time-varying codes approaches the Yudkin–Viterbi exponent. Since each b/n periodically time-varying code can be represented as a bT/nT time-invariant code, it is trivially shown that for $b' \to \infty$ there exists a time-invariant code whose error exponent approaches the Yudkin–Viterbi exponent. Note, though, that our results in Theorem 1 above, are different (and stronger) in the following two ways. First, we show that the exponent of the average error probability over the larger ensemble of all bT/nT fixed code approaches the Yudkin–Viterbi exponent as $b' \to \infty$. Second, we provide an error exponent expression for time-invariant codes that holds for any value of b. In this respect, returning to periodically time-variant codes, it is interesting to explore their error exponent for any value of T, and not only asymptotically. As will be shown below, Theorem 1 and its derivation can be used to get such error exponent.

Interestingly, only a few changes are needed in the proof above in order to adapt it to periodic codes. One difference is the definition of the comparison length. For time-invariant codes it is the maximum value of ℓ such that all the row vectors of the matrix (10) are linearly independent. For periodically time-variant codes it is sufficient that the rows

$$i,\ i+T,\ i+2T,\ \cdots,\ i+\lfloor(\ell-i)/T\rfloor$$

for any $1 \le i \le T$, are linearly independent. Thus we should derive a result, similar to Lemma 3, that bounds the probability that in a random matrix the rth row is the first row that is a linear combination of rows $r-T$, $r-2T$, $\cdots$. Following the derivation of Lemma 3, this probability, that r is the first "T-dependent" row, is upper-bounded as

$$\Pr(r \text{ is the first } T\text{-dependent row}) \le \begin{cases} 0, & \text{for } r \le K \\ 2^{\lfloor (r-1)/T \rfloor} 2^{-bK}, & \text{for } K < r \le bTK+1 \\ 0, & \text{for } bTK+1 < r. \end{cases} \quad (20)$$

Now, in order to bound M_ℓ, the number of nonidentical rivals in this case of periodically time-variant codes, we can follow the derivation of (11) but use (20), and the fact that $2^x \ge 2^{\lfloor x \rfloor}$ for $x > 0$, to get

$$M_\ell \le \begin{cases} 0, & \text{if } \ell < K \text{ and } \ell < \tau \\ 2^{b(\ell-K)+b+\ell/T}, & \text{if } K \le \ell \le KbT \text{ and } l < \tau \\ 2^{b\ell}, & \text{if } \ell = \tau \le KbT \\ 0, & \text{if } \ell > KbT \text{ or } \ell > \tau. \end{cases} \quad (21)$$

Substituting this bound on M_ℓ in (12), leads to an upper bound on $\overline{P}_e$ for periodically time-variant codes. Similarly to the above, a good choice for τ, the look-ahead interval, is $\tau = bT(K-1)$. With this choice of τ, and with M_ℓ of (21) we get

$$\overline{P}_e \le 2^{-\rho n(K-1)R} \sum_{\ell=K}^{bT(K-1)} 2^{-n\ell[E_0(\rho)-\rho R(1+1/bT)]}. \quad (22)$$

This upper bound is the analogous expression to (15) for periodically time-varying codes. Looking on both expressions, we see that formally b in (15) is replaced by bT in (22), yet n is the same at both expressions. This actually indicates that we achieve the improvement in the error exponent by going from b to $b' = bT$, yet the complexity of the code remains the same.

Proceeding in the same way that (18) was derived from (15), we get from (22) the following exponent for periodically time-varying codes:

$$E_p(R) = \max\left\{ \max_{0 \le \rho \le \min(\rho', 1)} \left(E_0(\rho) - \frac{1}{bT}\rho R \right), \max_{\rho' \le \rho \le 1} bT(E_0(\rho) - \rho R) \right\} \quad (23)$$

where ρ' is the largest solution of the equation

$$E_0(\rho) = \rho R(1 + 1/bT).$$

As noted above, this is the error exponent attained by bT/nT fixed code, yet the decoding complexity of the periodically time-varying code is only the complexity of a b/n fixed code.

V. Summary and Conclusions

In this paper we found an improved upper bound for the expected error probability (first error event) over a random choice of b/n time-invariant convolutional code. This bound is strictly better, for $b > 1$, than the corresponding bound for block codes. As $b \to \infty$ (yet $b/n = R$) the error exponent associated with this bound approaches the Yudkin–Viterbi exponent. While the fact that the Yudkin–Viterbi exponent is achieved by some fixed code for large enough b is not too surprising, our results provide a bound that holds for any constraint length and for any value of b, the number of symbols that enter the code in parallel.

The result on time-invariant codes was extended and we also provided an improved upper bound for the expected error probability over a random choice of b/n periodically time-varying codes, that hold for any period T. The expression for the bound is the same expression obtained for fixed codes, but with bT replacing b. Thus for the same decoding complexity $2^{\nu} = 2^{bK}$ of the fixed and the periodically time-variant b/n codes with delay K, the resulting bound on the error probability of the periodically time-variant code is better. This may confirm the practical evidence noticed recently regarding the better performance of periodically time-variant codes.

The results obtained in this paper do not claim to provide the best random-coding error exponent of fixed and periodically time-varying convolutional codes. Actually, finding the optimal error exponent is still an open problem. Nevertheless, the exponent obtained in this paper cannot be improved significantly, unless a different approach is taken. The reason is that by using as a basic block the random coding exponent for block codes, one cannot achieve a better bound than $2^{-HE_r(R)}$ where H indicates the ensemble randomness, i.e., its entropy [13]. In our case, the ensemble randomness, or the number of random bits that are needed to specify a (periodically time-variant) code is $H = bTnK$. Hence, one cannot expect to achieve a better exponent than that of results then a block code of length $bTnK$. This is exactly what we got for near-capacity rates (rates that use $\rho \geq \rho'$). In this paper we analyzed the average performance over the random ensemble of convolutional codes. As in block codes, the best possible convolutional code may have (at least in some rates) a better exponent. Thus a more refined analysis, that uses, e.g., expurgating techniques, may lead to an improved exponent, as was done in [18] for time-varying codes.

Finally, our results should also be extended to provide the error exponent for the nit error rate (BER) of time-invariant and periodically time-variant convolutional codes, and not only the first error event.

Acknowledgment

The authors wish to thank K. Zigangirov for carefully reading the manuscript and for his many useful remarks. They also acknowledge D. Forney for his suggestion to apply the results obtained for fixed codes to periodically time-variant codes.

References

[1] A. R. Calderbank, G. D. Forney, and A. Vardy, "Minimal tail-biting trellises: The Golay code and more," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1435–1455, July 1999.

[2] V. V. Chepyzhov, B. J. M. Smeets, and K. Sh. Zigangirov, "The free distance of fixed convolutional rate 2/4 codes meets the Costello bound," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 1360–1366, July 1992.

[3] D. J. Costello, "Free distance bounds for convolutional codes," *IEEE Trans. Inform. Theory*, vol. 20, pp. 356–365, May 1974.

[4] P. Ellias, "Coding for noisy channels," in *IRE Conv. Rec.*, 1955, pp. 37–46.

[5] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure (Also, correction)," *IEEE Trans. Inform. Theory*, vol. IT-16 (IT-17), pp. 720 –738 (p. 300), Nov. (May) 1970 (1971).

[6] ——, "Convolutional codes II: Maximum-likelihood decoding," *Inform. Contr.*, vol. 25, pp. 222–266, 1974.

[7] ——, "Convolutional codes III: Sequential decoding," *Inform. Contr.*, vol. 25, pp. 267–297, 1974.

[8] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY: Wiley, 1968.

[9] ——, "The random coding bound is tight for the average code," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 244–246, Mar. 1973.

[10] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of Convolutional Coding*. Boston, MA: Kluwer, 1999.

[11] P. J. Lee, "There are many good periodically time-varying convolutional codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 460–463, Mar. 1989.

[12] M. Mooser, "Some periodic convolutional codes are better than any fixed code," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 750–751, Sept. 1983.

[13] N. Shulman, "Coding theorems for structured code families," Masters thesis, Dept. Elec. Eng.-Syst., Tel-Aviv Univ., Tel-Aviv, Israel, 1995.

[14] N. Shulman and M. Feder, "A simple proof that time-invariant convolutional codes attain capacity," in *IEEE Int. Symp. Information Theory*, Whistler, B.C., Canada, Sept. 1995.

[15] ——, "Random coding techniques for nonrandom codes," *IEEE Trans. Inform. Theory*, pp. 2101–2104, Sept. 1999.

[16] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[17] ——, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 751–772, Oct. 1971.

[18] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York, NY: McGraw-Hill, 1979.

[19] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York, NY: Wiley, 1965.

[20] H. L. Yudkin, "A framework for low-complexity communication over channels with feedback," Sc.D. dissertation, Dept. Elec. Eng. Comput. Sci., MIT, Cambridge, MA, 1965.

[21] K. Sh. Zigangirov, "On the error probability of sequential decoding on the BSC," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 199–202, Jan. 1972.

[22] ——, *Procedures of Sequential Decoding*. Moscow, Russia: Svjaz, 1974.

[23] ——, "Time-invariant convolutional codes: Reliability function," in *Proc. 2nd Joint Soviet-Swedish Workshop Information Theory*, Gränna, Sweden, Apr. 1985.

[24] ——, "New asymptotic lower bounds on the free distance for time-constant convolution codes," *Prob. Inform. Transm.*, pp. 104–111, 1986.

[25] K. Sh. Zigangirov and V. V. Chepyzhov, "On the existence of time-invariant convolutional codes with transmission rate $2/c$, $c \geq 4$, which meets the Costello bound" (in Russian), *Probl. Inform. Transm. (Probl. Pered. Inform.)*, pp. 16–29, 1991.