Packet error rates of terminated and tailbiting convolutional codes

Johan Lassing, Tony Ottosson and Erik Ström
Dept. of Signals and Systems, Chalmers University of Technology
S-412 96 Göteborg, Sweden
Johan.Lassing@s2.chalmers.se

Abstract— **When a convolutional code is used to provide forward error correction for packet data transmission, the standard performance measures of convolutional codes, i.e., bit error rate and first-event error rate, become less useful. Instead we are interested in the average probability of block (or packet) error. In this paper a modified transfer function approach is used to obtain a union bound on the block error rate of terminated and tailbiting convolutional codes. The performance of these block codes obtained from termination and tailbiting is compared to the performance of some standard block codes for various information block lengths and also compared to the sphere packing lower bound for optimal block codes. The conclusion is that block codes derived from convolutional codes form a competitive class of short block length block codes.**

Keywords— **convolutional code, block code, block error probability, packet error probability, union bound, terminated convolutional code, tailbiting convolutional code, sphere packing bound**

I. Introduction

We will consider the case where a rate $R = 1/n$, constraint length K (shift register memory elements + 1) convolutional encoder is used to encode a packet $\mathbf{q}$ of r binary digits. The resulting encoded data packet will consist of $N = nr$ bits (disregarding termination effects treated in the next section), due to the redundancy added by the convolutional encoder. For a review on the encoding and decoding of convolutional codes, refer to e.g., [1, ch. 4] [2, ch. 11-13]. The encoded block of data will be transmitted to the receiver, that is allowed to observe the N bits, assumed here to have been disturbed independently in each position (dimension) by a zero-mean Gaussian variable of variance $N_0/2$. The average block (or packet) error probability is the probability $\Pr(\hat{\mathbf{q}} \neq \mathbf{q})$ that the receiver maximum-likelihood (ML) decoder decodes the received block into a message $\hat{\mathbf{q}}$ that was not transmitted. In this paper, a union bound on the block error probability will be derived and evaluated to compare certain short length block codes obtained from convolutional codes to some other common block codes of short block lengths.

II. Using a convolutional code as a block code

The standard application of convolutional codes is in situations where a low bit error probability (as opposed to block error probability) is desired, despite bad channel conditions. Since no requirement is put on the block error probability, the transmitted sequences may be very long, so that the probability that the entire sequence will be decoded correctly goes to zero. The average bit error probability will however still be small for well designed codes and reasonable bit-energy-to-noise ratios. In packet transmission the situation is quite different; a small amount of data bits (in the order of few hundred bits) are to be transmitted and decoded without errors or a repetition will be requested by the receiver.

Several strategies to turn a convolutional code into a block code can be distinguished between among which termination and tailbiting are two practical methods [3]. When zero-tail *termination* is applied the convolutional encoder is initially in a known state, which we assume to be the zero state, and the message sequence is padded at the end by $K-1$ zeros, so that the encoder returns to the zero state after having encoded the message. The price that is paid for this is a fractional rate loss, since $K-1$ known information bits are added to the original message. If the original information block contained r bits, the output codewords will be of length $(r+K-1)n$, so that the effective rate becomes

$$R' = R\left(1 - \frac{K-1}{r+K-1}\right), \tag{1}$$

where the term $\frac{K-1}{r+K-1}$ is the fractional rate loss and goes to zero as $r \to \infty$. The method of *tailbiting* [4] avoids the problem of the fractional rate loss by letting the last $K-1$ information bits define the starting state of the encoder and then clocking the encoder r times, so that exactly rn encoded bits are produced. This means that no rate loss occurs and that the encoding always starts and ends in the same state, but not necessarily the zero state. The price paid is the increased decoder complexity needed to estimate the initial state of the encoder and a poorer weight distribution of the derived block code (see section IV). Both the outlined termination methods result in a number of valid code sequences or codewords. By listing all these codewords we obtain a set $\mathcal{C}$, which is called the codebook. This means that the convolutional code is turned into a block code.

Both zero-tail termination and tailbiting will be considered in this paper. In section III, the union bound on the error probability of a block code is revised and a method of obtaining the necessary weight distributions is suggested in section IV. In the final sections an application to packet data systems is evaluated and results and comments are given.

III. Union bound on the error probability of block codes

A common way to estimate the performance of a block code is to use the union of the pair-wise error probabilities of all codewords in the block code. For the case of a white, Gaussian disturbance in each dimension, the pair-wise error probability P_2 of two codewords $\mathbf{a}$ and $\mathbf{b}$, with Hamming distance $d_H(\mathbf{a},\mathbf{b}) > 0$, is given as (assuming binary or quaternary phase-shift modulation)

$$P_2 = \Pr(\mathbf{a} \neq \mathbf{b}) = Q\left(\sqrt{2d_H(\mathbf{a},\mathbf{b})\gamma_b/n}\right), \tag{2}$$

where the Q-function is given by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt. \tag{3}$$

The probability P_2 is the probability that the maximum likelihood (ML) receiver decides in favor of codeword $\mathbf{b}$ when codeword $\mathbf{a}$ was transmitted and only those two codewords exist in the communication system codebook. In (2), $\gamma_b = E_b/N_0$ is the bit-energy-to-noise ratio of the bits entering the decoder, where E_b is the bit energy and N_0 is the one-sided power spectral density of the white noise.

The average block-error probability, P_e, of the ML receiver is given by

$$\begin{aligned} P_e &= \sum_{\mathbf{a}\in\mathcal{C}} \Pr(\mathbf{a} \text{ transmitted}) \Pr(\mathbf{a} \text{ not decoded}) \qquad (4)\\ &= \sum_{\mathbf{a}\in\mathcal{C}} \Pr(\mathbf{a} \text{ transmitted}) \sum_{\substack{\mathbf{b}\in\mathcal{C}\\ \mathbf{b}\neq\mathbf{a}}} \Pr\left(\hat{\mathbf{b}} = \mathbf{b} | \mathbf{a} \text{ transmitted}\right), \end{aligned}$$

where $\hat{\mathbf{b}}$ is the decoder decision on the transmitted *encoded* block. Since, there is a one-to-one (i.e., error-free) mapping between the block $\hat{\mathbf{b}}$ and $\hat{\mathbf{q}}$, the above expression is indeed the average block error probability of ML decoder output. It is common and reasonable to assume that each codeword is equally likely to be transmitted, so that $\Pr(\mathbf{a} \text{ transmitted}) = 2^{-r}$. However, the main problem with (4) is the probability terms in the second sum, $\Pr(\hat{\mathbf{b}} = \mathbf{b} | \mathbf{a} \text{ transmitted})$. In most cases of interest, there is no tractable way of obtaining these probabilities, so instead we try to obtain an upper bound on this sum. One way to do this is to consider the union of the pair-wise error probabilities of the transmitted vector $\mathbf{a}$ to all other codewords. This yields,

$$\begin{aligned} P_e &\leq \sum_{\mathbf{a}\in\mathcal{C}} \Pr(\mathbf{a} \text{ transmitted}) \sum_{\substack{\mathbf{b}\in\mathcal{C}\\ \mathbf{b}\neq\mathbf{a}}} Q\left(\sqrt{2d_H(\mathbf{a},\mathbf{b})\gamma_b/n}\right)\\ &= 2^{-r} \sum_{\mathbf{a}\in\mathcal{C}} \sum_{\substack{\mathbf{b}\in\mathcal{C}\\ \mathbf{b}\neq\mathbf{a}}} Q\left(\sqrt{2d_H(\mathbf{a},\mathbf{b})\gamma_b/n}\right). \qquad (5) \end{aligned}$$

To come further, we use the fact that our convolutional code is linear, and therefore the derived block code is also linear. A linear block code has the appealing property that the sum (over $GF(2)$) of any two codewords is also a codeword, i.e., $\mathbf{a} + \mathbf{b} = \mathbf{c}$; $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathcal{C}$. In particular, the all-zero codeword must be part of any linear code. Linearity gives us that all 2^r terms in the inner sum over the codewords in (5) are equal, so that for linear block codes (5) simplifies to

$$P_e \leq \sum_{\substack{\mathbf{b}\in\mathcal{C}\\ \mathbf{b}\neq\mathbf{0}}} Q\left(\sqrt{2d_H(\mathbf{0},\mathbf{b})\gamma_b/n}\right), \tag{6}$$

where we have assumed that the all-zero codeword is transmitted. We observe that the only code dependent parameter in this expression is the function $d_H(\mathbf{0},\mathbf{b})$, the Hamming weight of the binary vector $\mathbf{b}$. Therefore, if we define the weight distribution function $w(d)$ as the number of codewords in $\mathcal{C}$ having Hamming weight d, we may further simplify the union bound of (6) to

$$P_e \leq \sum_{d=1}^{d_{max}} w(d) Q\left(\sqrt{2d\gamma_b/n}\right), \tag{7}$$

where d_{max} is the maximum Hamming weight of any codeword in $\mathcal{C}$. Note that the sum starts at $d = 1$ to exclude the transmitted all-zero codeword. For zero-tail terminated convolutional codes, $d_{max} \leq (r + K - 1)n$, while for tail-biting convolutional codes, $d_{max} \leq rn$.

The evaluation of (7) only requires knowledge of the weight distribution $w(d)$ of the code, which is significantly easier to obtain than finding closed forms for the expressions we started with, i.e., (4). The price paid for this simplification is that (7) is an upper bound on the error probability and therefore, at best, it is a good approximation to the true error probability. In general, it is necessary to do computer simulations to establish over what range this approximation is valid.

IV. Finding the weight distribution

The generation of output sequences from a convolutional encoder is conveniently described by the encoder state diagram. In figure 1 the state diagram of an $R = 1/2$, $K = 3$, maximum free distance convolutional code [5] is shown. The state diagram is seen to be a weighted, directed graph,

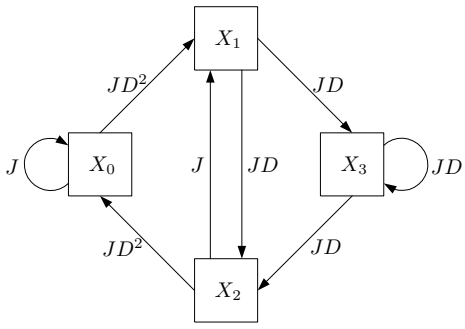

Fig. 1. State diagram for a $R = 1/2$, $K = 3$ convolutional encoder.

where the weights on the graph edges (transitions between states) are called indeterminates and represent some property that will accumulate as the vertices (states) are traversed. In the example diagram of figure 1, J enumerates

how many transitions that are made and D enumerates the Hamming weight of the encoded bits on each transition. The exponents of D are encoder properties and need to be carefully selected to obtain good codes (usually through computer search, see e.g., [5]).

For the present problem we need to be able to find the path enumerators of all paths through the state diagram starting and ending in the same state X_i after L transitions, where $L = r + K - 1$ for terminated convolutional codes and $L = r$ for tailbiting convolutional codes. To find the path enumerators, we modify the standard transfer function method of finding the distance properties of convolutional codes as described in [1, sect 4.3]. In the modified scheme, the enumerators for all paths starting and ending in state X_i are obtained by adding to the state diagram a starting and ending state and connecting them appropriately. This is illustrated for state X_0 in figure 2 for the example encoder. Since the encoding is constrained to start in state X_0, the state δ is introduced. The connections from δ to the other states are the same as for X_0 (refer to figure 2). In the same way, the state X_4 is introduced as a collecting state. By letting the starting state $\delta = 1$, the ending state X_4 will account for all paths through the state diagram starting and ending in state X_0. Note that all possible code sequences starting and ending in state X_0 are counted and not only unmerged sequences, as is normal for evaluating convolutional code performance. To further clarify the procedure, the modified state diagram used to find the enumerators for all paths starting and ending in state X_1 is shown in figure 3. Note that the original state diagram remains unchanged and the only thing that changes are the connections (and the weights) from the added starting and ending states.

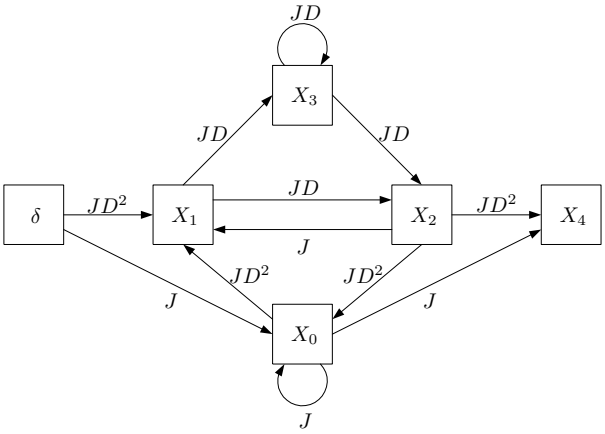

Fig. 2. Modified state diagram for a $R = 1/2$, $K = 3$ convolutional encoder. X_4 enumerates all paths starting and ending in state X_0.

The modified state diagrams in figures 2 and 3 describes a system of equations in 5 $\left(= 2^{K-1} + 1\right)$ unknowns (one for each node, minus the initial state, since $\delta = 1$),

$$\mathbf{M}_i \mathbf{x}_i = \mathbf{r}_i \tag{8}$$

where $\mathbf{M}_i$ is an $\left(2^{K-1} + 1\right) \times \left(2^{K-1} + 1\right)$ matrix and the index $i \in \{0, 1, \ldots, 2^{K-1} - 1\}$ indicates for what starting

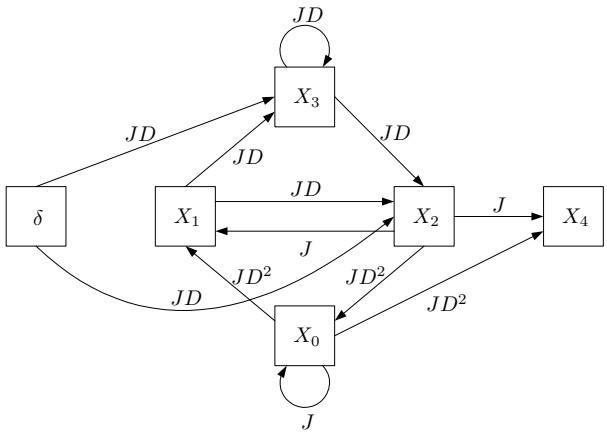

Fig. 3. Modified state diagram for a $R = 1/2$, $K = 3$ convolutional encoder. X_4 enumerates all paths starting and ending in state X_1.

and ending state X_i the solution is valid. The right-hand side vector, $\mathbf{r}_i$, is a vector containing the weights on the transitions from the starting state δ.

It is illustrative to explain how the matrix equation in (8) is obtained. The starting point is the state transition matrix, i.e., the matrix containing the weights of all transitions of the state diagram in figure 1. For this example encoder, the state transition matrix is

$$\mathbf{A} = \begin{bmatrix} J & JD^2 & 0 & 0 \\ 0 & 0 & JD & JD \\ JD^2 & J & 0 & 0 \\ 0 & 0 & JD & JD \end{bmatrix} \tag{9}$$

where matrix element a_{kl} is the path enumerator for the transition from state k to state l (the path enumerator is set to zero if no such transition is available). Note that this indexing convention is slightly different from what is normal for matrices, since the upper left element with this convention is a_{00} and not a_{11} as is more common. Let the columns of the matrix $\mathbf{A}$ be labeled by $\mathbf{A}[j]$ and the rows by $\mathbf{A}'[j]$, where the first column (or row) is given by $j = 0$. Next, if we are interested in all paths starting and ending in a particular state, X_i, we form a new matrix $\mathbf{B}_i$ by appending to $\mathbf{A}$ the ith column of $\mathbf{A}$ and making it square by appending a row of $2^{K-1} + 1$ zeros. This means that this new matrix has the form

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{A} & \mathbf{A}[i] \\ \mathbf{0} & 0 \end{bmatrix} \tag{10}$$

From $\mathbf{B}_i$ the matrix $\mathbf{M}_i$ is obtained as

$$\mathbf{M}_i = \mathbf{I} - \mathbf{B}_i' \tag{11}$$

where $\mathbf{I}$ is the identity matrix and the prime indicates matrix (or vector) transpose. The right hand side vector $\mathbf{r}_i$ is simply

$$\mathbf{r}_i = [\mathbf{A}'[i] \quad 0]' \tag{12}$$

For zero-tail (zt) terminated convolutional codes we are interested in the solution for $i = 0$, so that for our present

example we obtain

$$\begin{bmatrix} 1-J & 0 & -JD^2 & 0 & 0 \\ -JD^2 & 1 & -J & 0 & 0 \\ 0 & -JD & 1 & -JD & 0 \\ 0 & -JD & 0 & 1-JD & 0 \\ -J & 0 & -JD^2 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} J \\ JD^2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

From the solution of this system of equations, the desired transfer function is obtained from $X_4[i=0] \equiv X_4[0]$, where we have emphasized that the solution was obtained for paths starting and ending in state X_0. In this case,

$$T_{zt}(D,J) = X_4[0] = \frac{J^2 - (J^3+J^4)D + (J^3+J^4)D^5}{1-J-(J-J^3)D-J^3D^5}$$

Since the indeterminate J enumerates the number of transitions, the weight spectrum of a block code obtained from zero-tail termination after L transitions is found from the coefficient of the factor J^L of the serial expansion of $T_{zt}(D,J)$. This coefficient, which is a polynomial in D, is the weight distribution of the block code. For example, a block code obtained from $L=6$ transitions (information block length $r = L-K+1 = 4$) in the modified state diagram of figure (2) has weight distribution polynomial

$$W_6(D) = 1 + 4D^5 + 5D^6 + 4D^7 + D^8 + D^{10}.$$

For tailbiting codes, we are interested in the weights of the *collection* of all paths starting and ending in each of the states X_i after L transitions. There are several possible ways of obtaining this collection, but the straight-forward approach is to realize that, since all paths starting and ending in X_i are accounted for by the last element ($X_4[i]$) of the solution vector $\mathbf{x}_i$, the collection of paths is accounted for by

$$T_{tb}(D,J) = \sum_{i=0}^{2^{K-1}-1} X_4[i] \qquad (13)$$

so that the system of equations in (8) needs to be solved 2^{K-1} times to get $T_{tb}(D,J)$. The weight distribution of the block code formed from the tailbiting convolutional code is now found in exactly the same way as for the zero-tail termination case, as described above. For example, the weight distribution of a block code obtained from tailbiting of the example convolutional code discussed above with $L=4$ transitions (information block length $r = L = 4$) is given in the third column of table II. It is seen from the table that the free distance of the tailbiting code is only 2, whereas it is 5 for the terminated code of the same information block length, but this apparent drawback is counteracted by the fractional rate loss of the terminated code, which reduces the effective bit-energy-to-noise ratio for the terminated code.

Other ways of obtaining the weight distributions discussed above and related weight distributions can be found in [3, 4].

V. Application in packet data systems

As an interesting application of the above suggested methods, let us consider a communication system where data is transmitted in small bursts of less than 100 information bits at a time. We assume that each data packet is encoded by a rate 1/2 encoder (block or convolutional), but that no additional latency is tolerated. This means that the encoder is not allowed to wait for additional packets to arrive and treat these as longer blocks during encoding.

Apart from the terminated and tailbiting convolutional codes, we also assume three different quadratic residue (QR) block codes to be used, all of rate 1/2; the Hamming (8,4,4), the Golay (24,12,8) and the quadratic residue (48,24,12) codes [6, ch. 16]. Standard block code notation is used here; (N, m, d), where N is the coded block length, m is the input information block length and d is the smallest Hamming distance between any two codewords in the code. For convolutional codes, the notation is (n, k, K), where n and k are the number of input bits and coded output bits per cycle, respectively, and K, as before, is the constraint length.

The block error rates of the QR codes will be compared to the block error rates of terminated and tailbiting convolutional codes of various constraint lengths using the methods described in section IV. For all codes, maximum-likelihood decoding is assumed, where the complexity (in terms of trellis edge evaluations per decoded bit) of ML decoding of the QR block codes above is roughly comparable to ML decoding of convolutional codes with constraint lengths 3, 7 and 15, respectively [7, 8].

VI. Results and simulations

We start out by giving the detailed calculations for the simplest codes studied here; the Hamming (8,4,4) code and the terminated and tailbiting convolutional codes of constraint length $K=3$. In table I the weight distribution of the Hamming (8,4,4) code [6, p. 597] is given along with the weight distributions for the Golay (24,12,8) [6, p 597] and the quadratic residue (48,24,12) [6, p. 604] codes.

TABLE I
Weight distributions of Hamming (8,4,4), Golay (24,12,8) and quadratic residue (48,24,12) block codes.

d	$w_H(d)$	$w_G(d)$	$w_{QR}(d)$
0	1	1	1
4	14	0	0
8	1	759	0
12	0	2576	17296
16	0	759	535095
20	0	0	3995376
24	0	1	7681680
28	0	0	3995376
32	0	0	535095
36	0	0	17296
48	0	0	1

If we assume that the information packet consists of $r = mp$ bits, where $p \in Z^+$ and $m = 4$ for the Hamming (8,4,4) code, the encoder will output p blocks of $N = 8$ encoded bits each. For the entire encoded packet of $8p$ bits to be received without error, each of the p encoded blocks must be correctly received. Therefore, the probability that the packet is received in error is given by

$$P_w = 1 - (1 - P_e)^p \tag{14}$$

where P_e is the probability that an encoded sub-packet is erroneously decoded, which is given by (7) and evaluated using the weight distribution in the second column of table I.

For a terminated convolutional code, on the other hand, the entire information block of r bits can be encoded and only in the end it needs to be terminated by appending $K - 1$ zeros. The effective rate of a terminated rate 1/2 convolutional code of constraint length K as a function of r is according to (1),

$$R' = \frac{r}{2(r + K - 1)} = \frac{1}{2} - \frac{K - 1}{2(r + K - 1)} \tag{15}$$

so that $R' \to 1/2$ as $r \to \infty$, as expected. As r gets larger, the number of codewords will of course increase, but the free distance of the block code derived from termination will not change. Instead, the multiplicities of the low order terms of $w(d)$ increase, as can be seen from table II. In this table, the weight distributions for $r = 4$, $r = 6$ and $r = 8$ information bits per encoded block is given. For the block codes derived from tailbiting the free distances starts at low values, increasing until the free distance is the same as for the terminated codes. To make fair comparisons between the block codes, the tailbiting convolutional codes and the terminated convolutional codes, the effective bit-energy will be lowered for the terminated convolutional codes according to the rate reduction of (15).

In figure 4 the bit-energy-to-noise ratio required to achieve $P_w = 10^{-4}$ (or $P_e = 10^{-4}$ for convolutional codes) on an additive, white, Gaussian noise (AWGN) channel is plotted as a function of the information block size for the QR codes and the $K = 3$ tailbiting and terminated convolutional codes. The performance of the QR codes is evaluated using (14) and for the convolutional codes (7) is used. Also included for comparison is the sphere packing lower bound (spb) on the performance of the optimal codes for each r [7, 9–11]. This bound is a better alternative for comparison in this case (and many others) than the channel capacity (which gives $E_b/N_0 = 0$ dB for this case), since the channel capacity is assuming infinite block length. The bound shown is valid for the continuous-input AWGN channel, which makes it slightly too optimistic in the comparison made here.

It is obvious from figure 4 that the QR codes all perform very well for their shortest block lengths (within 0.7 dB of an optimal code), but they all start to deviate as the block length is increased. The terminated and tailbiting convolutional codes also perform well at short block lengths, the terminated code being slightly superior for the very shortest block lengths, but as the block lengths increase, the performance curves flattens out. It is interesting to note that the slope of all the curves as the block length increases is very similar.

TABLE II

WEIGHT DISTRIBUTIONS OF BLOCK CODES OBTAINED VIA TERMINATION (ZT) AND TAILBITING (TB) OF A $R = 1/2$, $K = 3$ CONVOLUTIONAL CODE FOR $r = 4$, 6 AND 8 INFORMATION BITS.

d	$w(d)$					
	$r = 4$		$r = 6$		$r = 8$	
	zt	tb	zt	tb	zt	tb
0	1	1	1	1	1	1
1	0	0	0	0	0	0
2	0	2	0	0	0	0
3	0	4	0	2	0	0
4	0	1	0	9	0	2
5	4	4	6	12	8	24
6	5	4	9	13	13	36
7	4	0	12	18	20	40
8	1	0	12	6	28	57
9	0	0	6	0	32	40
10	1	0	3	3	38	20
11	0	0	8	0	40	24
12	0	0	3	0	40	12
13	0	0	0	0	24	0
14	0	0	0	0	5	0
15	0	0	0	0	4	0
16	0	0	0	0	3	0
R'	1/3	1/2	3/8	1/2	2/5	1/2

Regarding the decoding complexity of the codes, the decoding complexity of the Hamming (8,4,4) code and the terminated code is similar, whereas the ML decoding of the tailbiting code requires one pass of the Viterbi algorithm for each state (2^{K-1}), so that it is 4 times more complex for this case. However, at least for block lengths above 50, the gain of the tailbiting code is marginal over the terminated code, and the increased complexity is perhaps not motivated. Note that there are several sub-optimal decoding procedures suggested for tailbiting codes (see e.g., [4]), that would reduce the complexity difference (but also the coding gain), but this is not considered here.

In figure 5 the same comparison is made, but the constraint length of the convolutional code is increased to $K = 5$ and the interesting region is zoomed in on, so that r ranges from 1 to 48. The decoding complexity of this convolutional code is much less than the Golay (24,12,8) code (see section V), but still it is seen that the tailbiting code outperforms the Golay code over almost the entire range of block lengths. For block lengths above 40 both the terminated and the tailbiting codes perform better. Another interesting observation is made in this figure, the heavy variations of the required E_b/N_0 for the tailbiting codes of shorter block lengths. This is a result of the truncation effects that occur when the paths through the state diagram

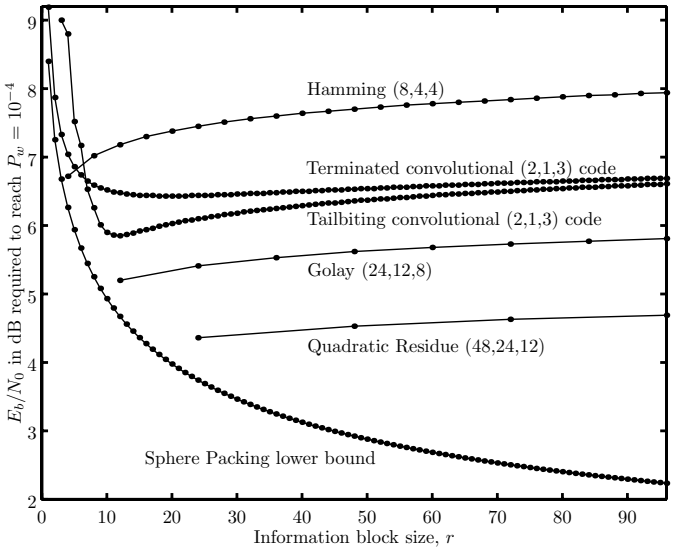

Fig. 4. Plot showing the bit-energy-to-noise ratio required to achieve $P_w = 10^{-4}$ on an AWGN channel for various short $R = 1/2$ block codes. The terminated and tailbiting convolutional codes have constraint length $K = 3$. The information block size ranges from 1 to 96. Included for comparison is the sphere packing lower bound, which is a lower bound on the performance of the optimal codes.

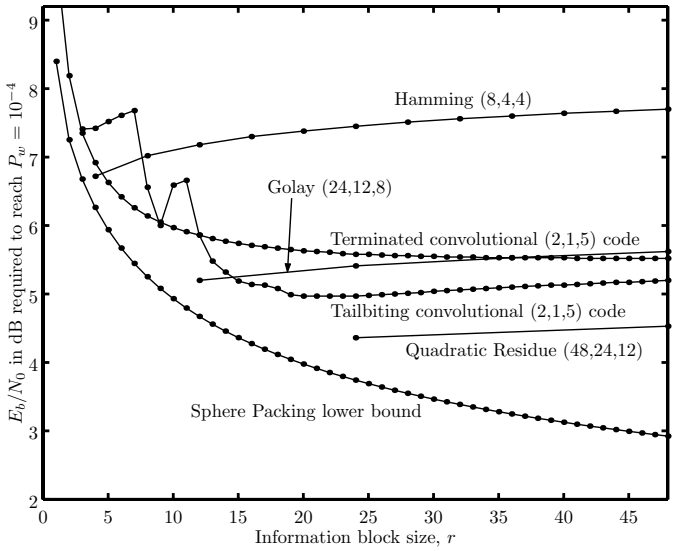

Fig. 5. Plot showing the bit-energy-to-noise ratio required to achieve $P_w = 10^{-4}$ on an AWGN channel for various short $R = 1/2$ block codes. The terminated and tailbiting convolutional codes have constraint length $K = 5$. The information block size ranges from 1 to 48. Included for comparison is the sphere packing lower bound, which is a lower bound on the performance of the optimal codes.

are required to start and end in the same state.

VII. Conclusions

A method to calculate the union bound on the average block-error probability of a block code obtained from a zero-tail terminated or tailbiting convolutional code is given. The only code dependent quantity needed to evaluate the performance is the weight distribution of the derived block code, and a method to obtain the weight distribution of both terminated and tailbiting convolutional codes is given. The proposed method involves a minor modification to the well known procedure to obtain the transfer function of a convolutional code given in, e.g., [1]. The union bound was applied to terminated and tailbiting rate $1/2$ convolutional codes of constraint lengths $K = 3$ and $K = 5$ and the performance was compared to that of three common rate $1/2$ quadratic residue codes. For comparable complexity, it was seen that the block codes derived from convolutional codes outperformed the QR codes and that the tailbiting codes are superior to the terminated codes, except for the very shortest block lengths. However, when the information block lengths exceed 50 bits, the energy gain becomes marginal. It is concluded that block codes derived from convolutional codes forms an attractive class of block codes that perform very well compared to the QR block codes of similar decoding complexity studied in this paper.

References

[1] A. Viterbi and J. K. Omura, *Principles of digital communication and coding*, McGraw-Hill, New-York, 1979.

[2] S. Wicker, *Error control systems for digital communication and storage*, Prentice-Hall, Upper Saddle River, NJ, 1995.

[3] J.K. Wolf and A.J. Viterbi, "On the weight distribution of linear block codes formed from convolutional codes," *IEEE Transactions on Communications*, vol. COM-44, no. 9, pp. 1049–1051, Sept. 1996.

[4] H. H. Ma and J. K. Wolf, "On tail biting convolutional codes," *IEEE Transactions on Communications*, vol. COM-34, no. 2, pp. 104–111, Feb. 1986.

[5] P. Frenger, P. Orten, and T. Ottosson, "Convolutional codes with optimum distance spectrum," *IEEE Communications Letters*, vol. 3, no. 11, pp. 317–319, Nov. 1999.

[6] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, North-Holland, Amsterdam, 1977.

[7] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of block size," *JPL TDA Progress Report 42-133*, May 1998.

[8] A. Kiely, S. Dolinar, R. McEliece, L. Ekroot, and W. Lin, "Trellis decoding complexity of linear block codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1687–1697, Nov. 1996.

[9] C.E. Shannon, "Probability of error for optimal codes in a gaussian channel," *Bell System Technical Journal*, vol. 38, pp. 611–656, May 1959.

[10] D. Slepian, "Bounds on communication," *Bell System Technical Journal*, vol. 42, pp. 681–707, May 1963.

[11] S.J. MacMullan and O.M. Collins, "A comparison of known codes, random codes, and the best codes," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 3009–3022, Nov. 1998.