DTU Library

Critical Lengths of Error Events in Convolutional Codes

Justesen, Jørn; Andersen, Jakob Dahl

[Link back to DTU Orbit](#)

also (7). The results for P_e as a function of the SNR and TTD as a parameter are given in Fig. 7. From the simulation results it follows that TTD ≥ 20 does not give significant improvements since a saturation effect occurs for TTD ≥ 20. We observe that introducing a delay $d = 6$ already gives a gain of more than 3 dB at an error probability of 10^{-4}. For larger values of d, the expected performance is determined by the free distance of the code, see (4).

Remarks: The described procedure gives a reliable state estimate if sufficient syndrome values equal to zero are received. Instead of estimating encoder states, we can also estimate encoder inputs. These estimates can then be used in decoding algorithms like that of Fano or the M-algorithm. We are presently investigating these applications.

V. CONCLUSIONS

We have shown how to use the inverse G^{-1} as a reliable encoder state-tracker. The estimation error probability strongly depends on the introduced decision delay d and the SNR. The method has a very low complexity and can be used to support decoding algorithms, like the Fano sequential decoding algorithm or the M-algorithm, where the presence or knowledge of the encoder state in the decoding process can be expected to improve performance of the decoding. We showed the relation between the estimation error probability and the column distance function. We also bounded the mean time to decision (MTTD) as an average for the time it takes before we observe the desired TTD $= d + m + m' + i$ zero syndrome outputs.

ACKNOWLEDGMENT

The authors wish to thank Adriaan van Wijngaarden for his support during the stay of the second author in Essen.

REFERENCES

[1] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.
[2] R. Johannesson, "Robustly optimal rate one-half binary convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 464–468, July 1975.
[3] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

Critical Lengths of Error Events in Convolutional Codes

Jørn Justesen and Jakob Dahl Andersen

Abstract—If the calculation of the critical length is based on the expurgated exponent, the length becomes nonzero for low error probabilities. This result applies to typical long codes, but it may also be useful for modeling error events in specific codes.

Index Terms—Burst lengths, convolutional codes, critical length, ensemble analysis.

I. INTRODUCTION

In the analysis of concatenated codes and other systems combining convolutional codes with multiplexing or other stages of coding it is important to know the distribution of error events after decoding of the convolutional code. For a specific code this distribution can be simulated or calculated from the weight generating function of the code and the properties of the channel [1]. However, it is of interest to compare these results to general properties of error patterns by considering the performance of average codes of sufficiently large constraint length.

The analysis of error-correcting codes through the error exponents for randomly chosen codes is a classical technique in information theory [2]. Following this tradition, Forney [3] introduced the concept of the critical length for long convolutional codes as the length of the error events that dominate the lower bound on error probability. This analysis, which also appears in Viterbi and Omura's text [1], indicates that for rates below R_0 the critical length is zero. However, for typical binary codes, the critical length is determined by the length of the minimum-weight codewords, and consequently it is greater than zero. For higher rates, the error events are longer, and the bound on error probability is known to be tight.

In Section II, we discuss the derivation of error exponents for convolutional codes, and make some comments on the relationship between distances and error exponents. In Section III, the correction to the critical length is presented, and in Section IV the critical length is derived as a function of the signal-to-noise ratio. In Section V the exponential decrease in probability for long burst is analyzed and finally, in Section VI, the results of the asymptotic analysis are compared to simulated and calculated results for a specific code with moderate memory.

II. ERROR EXPONENTS FOR BLOCK CODES AND CONVOLUTIONAL CODES

For simplicity we shall discuss only the performance on the binary-symmetric channel (BSC) and the additive white Gaussian noise channel (AWGN). For long codes, the error probability of block codes and convolutional codes is upper-bounded in terms of the error exponents. Usually the exponents are derived for general memoryless channels and later specialized to BSC and other simple cases. This approach may obscure the arguments. However, we shall not give a simplified derivation but rather give a few comments interpreting the
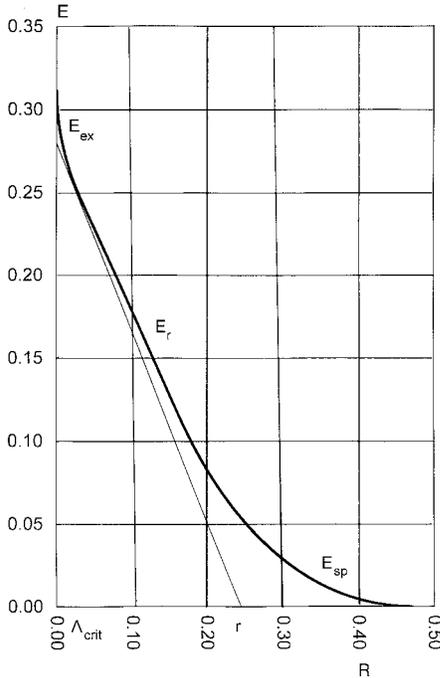
Fig. 1. Error exponents for block codes. BSC with $E_s/N_0 = -1.6$ dB.

derivations and the properties of the exponents. For block codes with rate $R = K/N$ we have

$$P(\mathcal{E}) \leq a \cdot 2^{-NE(R)} \tag{1}$$

where a is a constant. For convolutional codes with rate $r = k/n$ and memory M

$$P(\mathcal{E}) \leq a(n) \cdot 2^{-n\operatorname{Me}(r)} \tag{2}$$

where $a(n)$ is a nonexponential function. The error exponents are discussed in [1], [2], and many other books.

The easiest part of the error exponent decreases linearly with the rate of the code and reaches zero at R_0. This follows from the union bound applied to an ensemble of randomly chosen codes.

$$E(R) = E(0) - R = 1 - \log_2(1 + Z) - R = R_0 - R$$
$$\text{BSC:} \quad Z = 2\sqrt{p(1-p)} \tag{3}$$
$$\text{AWGN:} \quad Z = \exp(-E_s/N_0).$$

For high rates the bound may be improved so that nonzero exponents are obtained for rates up to the channel capacity (see Fig. 1).

For the BSC we have

$$E_{sp}(R) = T(p, \delta) - H(\delta)$$
$$= -\delta \log_2 p - (1 - \delta) \log_2(1 - p) - 1 + R \tag{4}$$

since

$$T(p, \delta) = -\delta \log_2 p - (1 - \delta) \log_2(1 - p)$$
$$H(\delta) = 1 - R.$$

For low rates, the error probability is dominated by error events of low weight. On the average, linear codes have a weight distribution that is obtained by scaling the binomial distribution to the right number of codewords. It is known that the straight-line bound is tight for the complete ensemble of linear block codes, but the average is dominated by a small fraction of codes that have very poor minimum weight. The so-called expurgated ensemble of codes is obtained by deleting codes with poor distances. For the simple case of the BSC (or the binary-input AWGN) this simply means that codes with minimum weights below the Gilbert bound are deleted. In the resulting codes there are only a few words at distance $d_{\min}$ from the transmitted word, and the expurgated exponent is found as an approximation to the probability of making a decision in favor of such a word.

$$E_{ex}(R) = -\frac{d_{\min}}{N} \log_2(Z) = -H^{-1}(1 - R) \cdot \log_2(Z) \tag{5}$$

where $d_{\min}/N = H^{-1}(1 - R)$ is the Gilbert bound.

In [3] Forney presented a technique of fundamental importance for the analysis of convolutional codes. After a certain number of information symbols, the code is terminated by making the input zero until the encoder has returned to the zero state. In this way a block code is obtained, and the performance of the convolutional code may be bounded by considering such a sequence of block codes of increasing length and rate. In this way the distances and error exponents for convolutional codes may be derived from known results for block codes. The following construction (named the inverse concatenation construction by Forney) may be applied to a graph of the block code exponents: A tangent to the block code bound at the point corresponding to rate R intersects the rate axis in $(r, 0)$ where r is the rate of the convolutional code. The intersection with the exponent/distance axis gives the performance bound for the convolutional code.

III. THE CRITICAL LENGTH

For a given channel and a convolutional code of rate r, the inverse concatenation construction gives the rate, $\Lambda_{\text{crit}} = (1 - \theta_{\text{crit}})r$, of the critical block code, and thus the number of information symbols in the critical error events. (In the literature the term R_{crit} refers to the rate where the straight-line bound and the high-rate bound for the block-code error exponents meet). The critical length is determined by normalization with respect to M

$$L_{\text{crit}} = \frac{K_{\text{crit}}}{M} = \frac{\Lambda_{\text{crit}}}{r - \Lambda_{\text{crit}}} = \frac{1 - \theta_{\text{crit}}}{\theta_{\text{crit}}} \tag{6}$$

where M is the memory of the convolutional code and K_{crit} the dimension of the critical block code; this definition is a mix of the definitions used in [1] and [3]. The performance of the convolutional code is close to the performance of the critical block code.

If the construction is based on the random coding exponent, the critical length equals 0 for all rates below R_0. However, this result is based on the existence of codewords of very low weight in some codes (on the average less than one). For typical codes, the expurgated exponent should be used, and the critical length becomes a positive function which increases with increasing rate. For $r = R_0$ there is still a discontinuity in the critical length due to the straight line portion of the error exponent. The critical length for a fixed channel as function of the rate r is shown in Fig. 2.

For rates $r < R_0$, this derivation of the critical length also relates the error events to the codewords of minimal weight. The codewords of weight d_{free} are minimum-weight codewords in the critical block code, and the critical length is seen to increase with the rate of
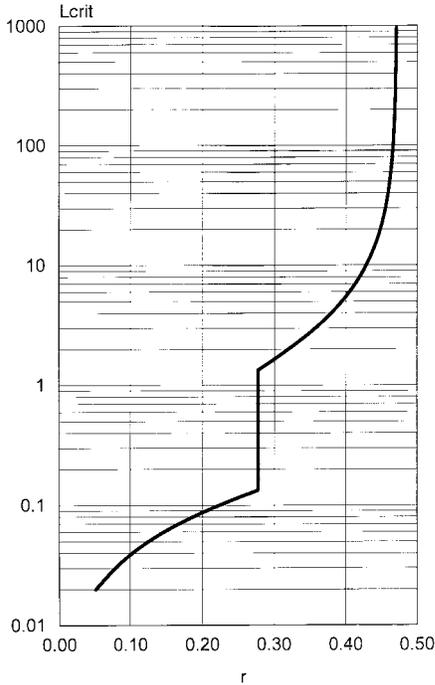
Fig. 2.   Critical length for BSC with $E_s/N_0 = -1.6$ dB.

Fig. 3.   Critical length for fixed rate, $r = 1/4$.

the convolutional code. This property of the distances was clearly not in agreement with the result that the error probability should be dominated by error events of length 0.

IV. PERFORMANCE AS A FUNCTION OF THE ERROR PROBABILITY

While error exponents and related results are usually given as functions of the code rate in the information theory literature, it is customary in communications to fix the code and study the performance as a function of the signal-to-noise ratio (SNR). For the binary-symmetric channel, we may consider the performance of a code with fixed rate as a function of the error probability. For small error probabilities the performance is determined by d_{free}. Considering the sequence of block codes obtained by terminating the convolutional code, the critical block code is the first one with $d_{\min}$ equal to d_{free}, and thus the same code for all small error probabilities. At the error probability which makes R_0 for the channel equal r, there is a discontinuous increase of the critical length. The critical length for a fixed rate $r = 1/4$ and a BSC is shown in Fig. 3.

V. DISTRIBUTION OF THE BURST LENGTH

In simulations of decoding systems it is sometimes assumed that the length of the error events approximately follows a geometric distribution. While the exponential decrease in probability is a very good approximation for bursts of length greater than the critical length, shorter bursts occur with smaller probabilities.

The exponent in the burst length distribution may be found by considering the error probability for the various terminated codes. The following derivation shows that the exponent for the decrease in probability with increasing burst length is approximately the block code error exponent divided by the rate of the code.

The probability of a burst of length K information symbols may be obtained from the block code error exponents for codes with dimension K and length $N = (K + M)/r$ [1], [3].
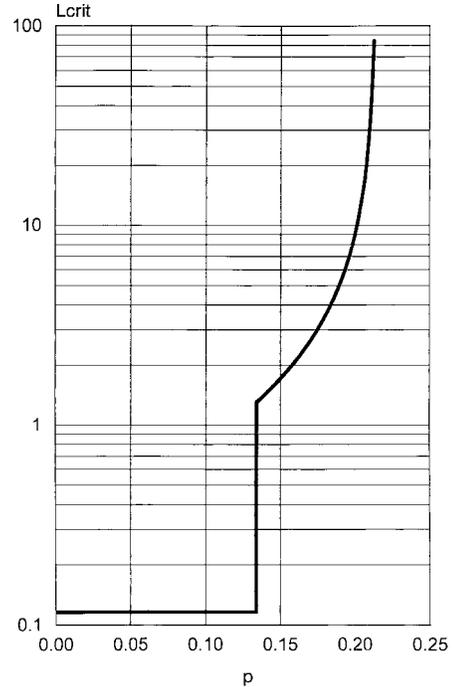
The reduction in error exponent for increasing K is then

$$
\begin{aligned}
\frac{d(-N \cdot E(R))}{dK} &= -E(R)\frac{dN}{dK} - N\frac{d(E(R))}{dK} \\
&= -\frac{E(R)}{R} - N\frac{dE}{dR}\frac{dR}{dK} \\
&= -\frac{E(R)}{R} - \frac{dE}{dR}\frac{M}{K+M}.
\end{aligned}
\tag{7}
$$

For large K, $R \sim r$ and $M/(K + M) \to 0$. Thus the exponent becomes simply

$$
\log_2 \frac{P(K+1)}{P(K)} = -\frac{E(r)}{r}.
\tag{8}
$$

VI. ERROR EVENTS IN SPECIFIC CODE

We will compare the results of the asymptotic analysis to the error events for a specific convolutional code with rate $r = 1/4$ and moderate memory, i.e, $M = 10$. For low SNR's the burst length distribution can be measured by simulation. For high SNR's the simulations require too much computer time. In this region the burst length distribution can be measured with a calculation based on the union bound. This calculation is only tight at SNR's well above R_0.

For the BSC we have the pairwise error probability for words with weight d

$$
P_d = \begin{cases}
\displaystyle\sum_{k=(d+1)/2}^{d} \binom{d}{k} p^k (1-p)^{d-k}, & d \text{ odd} \\[2ex]
\displaystyle\frac{1}{2}\binom{d}{\frac{d}{2}} p^{\frac{d}{2}} (1-p)^{\frac{d}{2}} + \sum_{k=d/2+1}^{d} \binom{d}{k} p^k (1-p)^{d-k}, & d \text{ even.}
\end{cases}
\tag{9}
$$

The exact weight enumerator polynomial for words of a given length is found by iterative multiplication with a submatrix from the transition matrix $\boldsymbol{W}(i, j)$, $i, j = 0, 1, \cdots, M - 1$ representing the output corresponding to the transition from state i to j, as
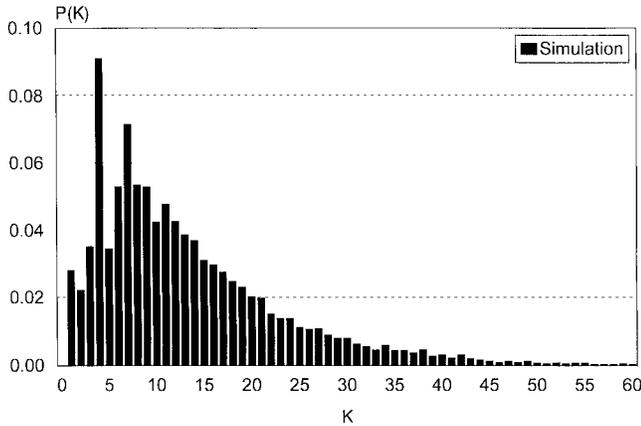
Fig. 4. Simulated burst length distribution for $M = 10$, $r = 1/4$ convolutional code at $E_s/N_0 = -2.6$ dB.
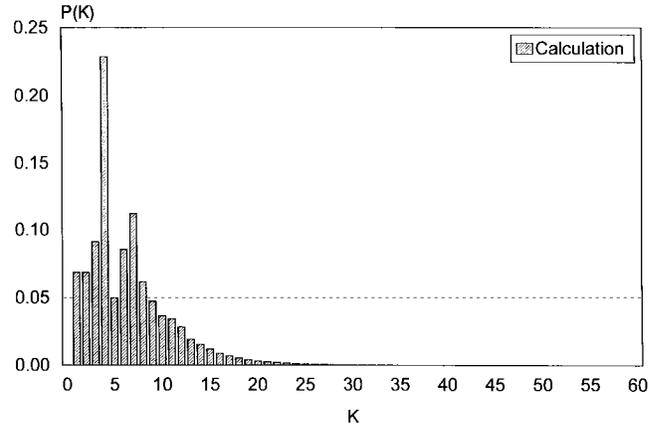

Fig. 6. Calculated burst length distribution for $M = 10$, $r = 1/4$ convolutional code at $E_s/N_0 = -0.6$ dB.
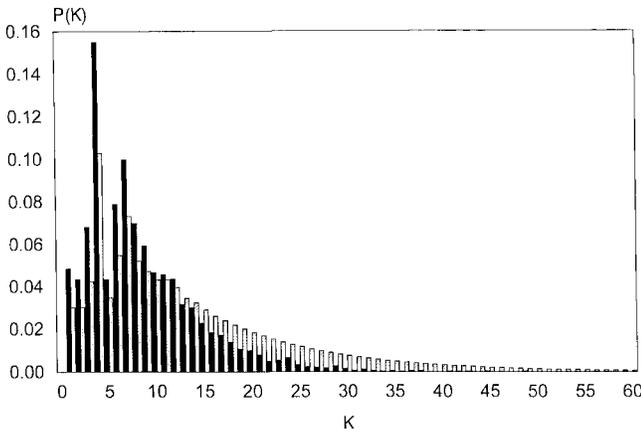

Fig. 5. Simulated and calculated burst length distributions for $M = 10$, $r = 1/4$ convolutional code at $E_s/N_0 = -1.6$ dB.

described in [4]. The transition matrix W is defined as $W(i, j) = 0$ for the impossible transitions and $W(i, j)$ equal to Z raised to the weight of the corresponding output for the possible transitions. Since the words are measured as paths starting and ending in the zero state with no intermediate returns only a submatrix of $\boldsymbol{W}$ is used, $W'(i, j)$, $i, j = 1, 2, \cdots M - 1$, representing the transitions between the nonzero states. The weight enumerator for words of length L is now found as

$$T_L(Z) = \boldsymbol{a} \cdot (\boldsymbol{W'})^{L+M-2} \cdot \boldsymbol{b} \tag{10}$$

where $\boldsymbol{a}$ represents the initial 1, i.e., $a(j) = W(0, j)$, $j = 1, 2, \cdots, M - 1$, and $\boldsymbol{b}$ represents the final 1, i.e., $b(i) = W(i, 0)$, $i = 1, 2, \cdots M - 1$.

Simulations were made at two signal-to-noise ratios, $E_s/N_0 = -2.6$ dB and $E_s/N_0 = -1.6$ dB. The calculations are presented for $E_s/N_0 = -1.6$ dB, and $E_s/N_0 = -0.6$ dB. At -1.6 dB, which is just above R_0, comparison between the simulated and the calculated distribution shows that the union bound is not very tight at this point. However, at $E_s/N_0 = -0.6$ dB we believe that the bound is tight and the calculated distribution is correct. The complete burst distributions are shown in Figs. 4–6.

We will now make a comparison between the results from the asymptotic analysis and the actual simulated values. At $E_s/N_0 = -2.6$ dB, where $r > R_0$, we found the average burst length to be 13.5 bits. The corresponding critical length is 1.63, giving 16.3

bits for the $M = 10$ code. The reduction in burst probability going from 40 to 50 bits is approximately 0.33. For $K = 40$ we have $N = (K + M)/r = 200$ and $R = 0.20$, for $K = 50$ we have $N = 240$ and $R = 0.21$. We calculate the decrease in the probability as $2^{-240 \times E(0.21) + 200 \times E(0.20)} = 0.32$. For long bursts we have the asymptotic value from (8), $P(K + 1)/P(K) = 0.94$, which for an increase of 10 bits gives $P(K + 10)/P(K) = 0.53$. At $E_s/N_0 = -1.6$ dB, where $r < R_0$, we found the average burst length to be 8.5 bits. The corresponding critical length is 0.12. The reduction in burst probability going from 20 to 30 bits is 0.18, while $2^{-160 \times E(0.19) + 120 \times E(0.17)} = 0.34$. The asymptotic value from (8) is $P(K + 1)/P(K) = 0.87$, and $P(K + 10)/P(K) = 0.24$. For the calculated distribution at $E_s/N_0 = -0.6$ dB we have the average burst length 6.4, while the critical length is still 0.12.

We must conclude that the asymptotic analysis only gives an indication of the actual burst distribution for codes with moderate memory, like the $M = 10$ code. In this case the average burst length will not follow the discontinuous function from Fig. 3, but be continuously increasing. Also the asymptotic value for high SNR's is above the 1.2 found from the analysis. The average burst length for the $M = 10$ code has an asymtotic value of 3, determined by the four minimum-weight words. However, the analysis still provides some usefull information for the comprehension of the distribution of error events after the convolutional code.

REFERENCES

[1] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
[2] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[3] G. D. Forney, "Convolutional codes II: Maximum-likelihood decoding," *Inform. Contr.*, vol. 25, pp. 222–250, 1974.
[4] P. J. Lee, "A very efficient transfer function bounding technique on bit error rate for Viterbi decoded, rate $1/N$ convolutional codes," TDA Progress Rep. 42–79, Jet Propulsion Lab. Pasadena, CA, Nov. 1984, pp. 114–123.