A Truncation-Depth Rule of Thumb for Convolutional Codes

B. Moision[1]

The commonly used rule of thumb of $5m$ for the truncation depth of a memory m convolutional code is accurate only for rate 1/2 codes and should be replaced by two to three times $m/(1-r)$ for a rate r code.

I. Introduction

An exact maximum-likelihood decoding of a convolutional code via the Viterbi algorithm requires storage of a path history for each state of the code to a depth at which all these paths agree. In practice, the complexity is often reduced by storing only a finite path length corresponding to the prior τ trellis stages, the truncation depth of the decoder. Decisions are forced (regardless of whether the paths agree) after a delay of τ stages. Forcing a decision after a fixed delay yields some performance degradation and, when designing the decoder, one wants to choose a truncation depth sufficiently large as to make this loss negligible, but no larger, so as not to incur unnecessary complexity.

A commonly cited rule of thumb is that a truncation depth of four to five times the memory of the code is acceptably large to limit losses due to finite truncation; see, e.g., [1, p. 258; 2, p. 338; 3, p. 262; 4, p. 485]. However, this rule is accurate only for rate 1/2 codes, a point that is not made in much of the literature. We will show that, for a rate r code, a more appropriate rule of thumb is that the truncation depth be two to three times $m/(1-r)$.

II. Truncation-Depth Bound

Our bound derives from the random coding results of [5]. We first review some notation from [5]. An (M,ν) trellis is a trellis corresponding to a shift register of length ν, where each register contains an M-vector and the input is an M-ary sequence (the corresponding trellis contains M^ν states). An (M,ν,n) trellis code augments an (M,ν) trellis by assigning n channel symbols to each edge. The rate of the code is $r=\log_2(M)/n$ bits/symbol. A random trellis code is an (M,ν,n) trellis in which each channel symbol on each edge is chosen randomly and independently according to some distribution p.

When $M=q^k$, the (M,ν,n) trellis corresponds to a rate $\log_2(q)k/n$ nonsystematic convolutional code over the field $GF(q)$ with k equal constraint lengths $\nu_i=\nu, 1\le i\le k$. The memory of this code is $m=\max_i \nu_i=\nu$.

[1] Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

We assume the code is decoded via the Viterbi algorithm with decisions on edges of the trellis made after a delay of τ trellis stages. A truncation error occurs when an incorrect edge is chosen that would not have been chosen with an infinite truncation depth. Forney [5] proved the following bound on the truncation error rate.

Theorem 1. *The probability of truncation error for an (M, ν, n) random trellis code satisfies*

$$P(\mathcal{E}_t) \leq \exp\big(-n\tau E(r)\big)$$

where $E(r)$ is the block code exponent.

This may be related to the code memory via the error-rate bound for ensembles of random trellis codes [13].

Theorem 2. *For any $\epsilon > 0$, the probability of error per unit time for an (M, ν, n) random trellis code with maximum-likelihood (ML) decoding satisfies*

$$P(\mathcal{E}) \leq K_1 \exp\big(-n\nu\big(e(r) - \epsilon\big)\big)$$

where K_1 is independent of ν, and $e(r)$ is the convolutional code exponent for the input distribution p.

It follows that the probability of error due to finite truncation is asymptotically equal to the ML error probability when [5]

$$\frac{\tau}{\nu} = \frac{e(r)}{E(r)}$$

In the Appendix we show this ratio may be bounded by

$$\frac{e(r)}{E(r)} \geq \frac{1}{1 - r/C} \geq \frac{1}{1 - r}$$

where C is the channel capacity, and the second inequality holds for a binary-input channel ($q = 2$). Moreover, the ratio approaches $1/(1 - r)$ as the fidelity of the channel increases, so that the bound is tight at sufficiently large signal-to-noise ratios (SNRs).

Hence, the loss due to finite truncation is on the order of the error rate when

$$\begin{aligned} \tau &= \frac{\nu e(r)}{E(r)} \\ &\geq \frac{\nu}{1 - r} \end{aligned} \tag{1}$$

Viterbi [6] illustrated a bound for the very noisy channel, that is, a channel where the output is almost independent of the input, e.g., as the SNR approaches zero:

$$\tau \geq \nu \begin{cases} \dfrac{1}{1-2r/C}, & 0 \leq r \leq \dfrac{C}{4} \\ \dfrac{1}{2(1-\sqrt{r/C})^2}, & \dfrac{C}{4} \leq r \leq \dfrac{C}{2} \\ \dfrac{1+\sqrt{r/C}}{1-\sqrt{r/C}}, & \dfrac{C}{2} < r < C \end{cases}$$

This bound is complementary, giving a bound on the truncation depth for a channel with low SNR. As noted, the bound in Eq. (1) is useful at high SNR, corresponding to the region where the minimum distance terms dominate.

A. Punctured Codes

Suppose we form a (q^k, ν, n) code by puncturing a (q^{k_1}, ν_1, n_1) mother code, where k_1 divides k and $\nu = \nu_1 k_1/k$; see, e.g., [7]. We refer to the resulting code as the daughter code. The two codes are represented by trellises with the same number of states, with k/k_1 stages of the mother code corresponding to 1 stage of the daughter code. Applying Eq. (1), the required truncation depth is

$$\tau \geq \frac{\nu}{1-r} \text{ stages of daughter code trellis}$$

$$= \frac{\nu_1}{1-r} \text{ stages of mother code trellis}$$

i.e., the truncation depth on the mother code goes as the memory of the mother code scaled by one minus the rate of the punctured code.

That the truncation depth should be increased for punctured codes has been noted, e.g., in [3, Section 6.6.4]. However, this is not emphasized in the literature. For example, [7, Section 4.6] tabulates performance of Consultative Committee for Space Data Systems (CCSDS) standardized rate 2/3, 3/4, 5/6, and 7/8 convolutional codes punctured from a rate 1/2, memory 6 mother code. For the mother code, a truncation depth of 30 has losses relative to an infinite truncation depth of ≈0.1 dB, and at a depth of 60 the losses are negligible. However, in [7, Section 4.6], the truncation depth of 60 is carried through for all daughter codes. This yields losses of 0.5 dB for the rate 7/8 daughter code, where a depth ≈120 should be used to yield negligible losses.

III. Truncation Depths for Particular Codes

How good is the estimate $\tau \geq \nu/(1-r)$ for particular codes from the ensemble? We are interested in the high SNR region, where the minimum distance terms dominate performance. A good indicator of the required truncation depth in this region is the path length at which all paths that diverge from a particular path have accumulated the minimum distance of the code; see, e.g., [8; 9; 3, p. 262]. Onyszchuk [10] showed that the truncation depth required to limit losses to <0.05 dB is slightly larger than this, but it remains a good approximate measure. Without loss of generality, assume the all-zeros sequence is transmitted and let $N(x, \tau)$ denote the number of paths of length τ and weight x that first diverged from the all-zeros state τ stages in the past ($N(x, \tau)$ includes all open, closed, and compound events).

Let d be the free distance of the code. Figure 1 illustrates $N(d-1, \tau), N(d, \tau)$, and $N(d+1, \tau)$ for the rate 1/2 optimum-distance-profile (ODP) codes with $2 \leq m \leq 10$ from [11, Table 8.1]. We see that

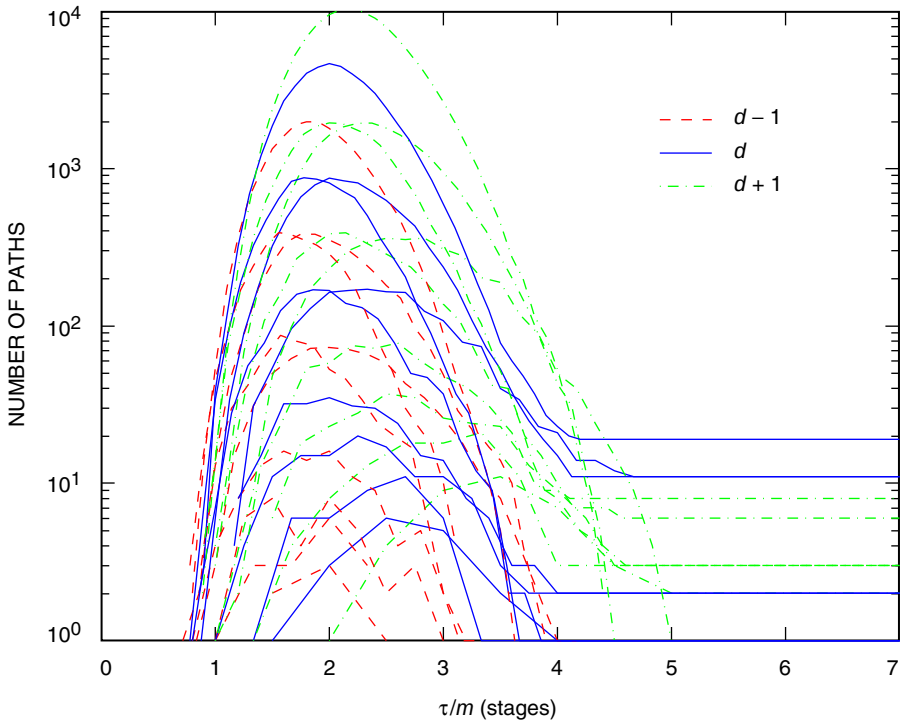

Fig. 1. Number of paths with distance $d-1$, d, and $d+1$ as a function of τ/m for some nonsystematic rate 1/2 codes with $m = 2,3,\ldots,10$.

$\tau \approx 5m$ is sufficient to guarantee all paths have accumulated distance d and that the multiplicity of the minimum distance paths has settled to that for an infinite truncation depth. However, the $\tau \approx 5m$ rule is inaccurate for code rates other than 1/2.

Figure 2 illustrates $N(d,\tau)/N(d,\infty)$ for a large collection of nonsystematic codes with roughly equal constraint lengths ν_i and rates 1/6 to 7/8 as a function of $\tau(1-r)/m$. Included are $(m+1,r) = (15,1/6),(15,1/4)$ codes; the CCSDS standard punctured codes of rate 2/3, 3/4, 5/6, and 7/8; all rate 2/3 ODP codes from [11, Table 8.14]; all rate 1/3 ODP codes with $m \leq 15$ from [11, Table 8.10]; and all rate 1/2 ODP codes with $m \leq 15$ from [11, Table 8.1]. We see that $\tau \approx 2.5m/(1-r)$ is a good predictor of the depth at which all paths have accumulated the minimum distance. The few outlying points correspond to codes with $m = 2$.

Let $\tau_{\min}$ be the first depth at which all distance d events have closed and all open events have accumulated distance greater than d. Figure 3 plots $\tau_{\min}$ versus two estimates of the value, $2m/(1-R)$ and $4m$, for the collection of codes illustrated in Fig. 2 (the estimates agree for rate 1/2 codes). We see that the first estimate is a good predictor of $\tau_{\min}$ for all codes considered.

IV. Conclusions

The commonly used rule of thumb of a truncation depth of five times the memory of a convolutional code is accurate only for rate 1/2 codes. For an arbitrary rate, an accurate rule of thumb is $2.5m/(1-r)$. For a punctured code, the rule also goes as $2.5m/(1-r)$, measured in stages of the mother code, where m is the memory of the mother code and r is the rate of the punctured code.

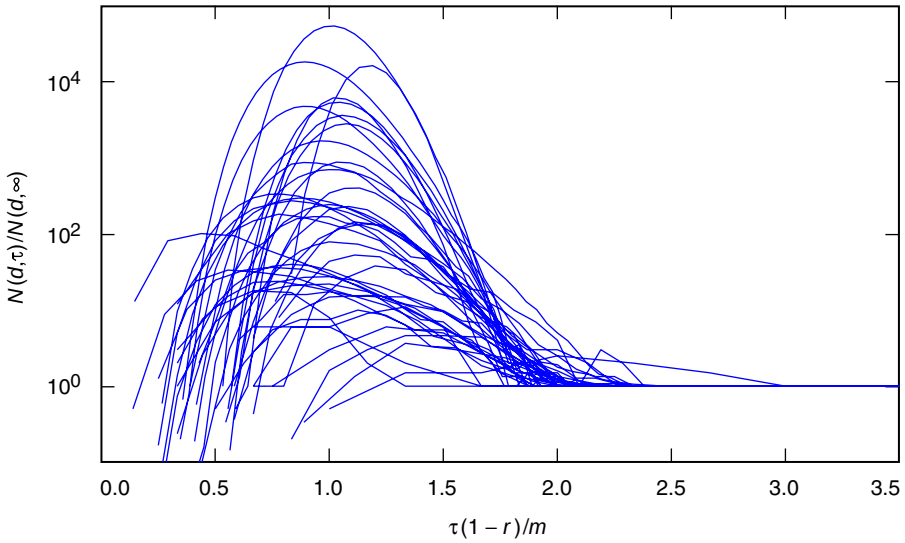

Fig. 2. $N(d,\tau)/N(d,\infty)$ **as a function of** $\tau(1-r)/m$ **for some nonsystematic codes. Code rates 1/6, 1/4, 1/3, 1/2, 2/3, 3/4, 5/6, and 7/8 are represented.**

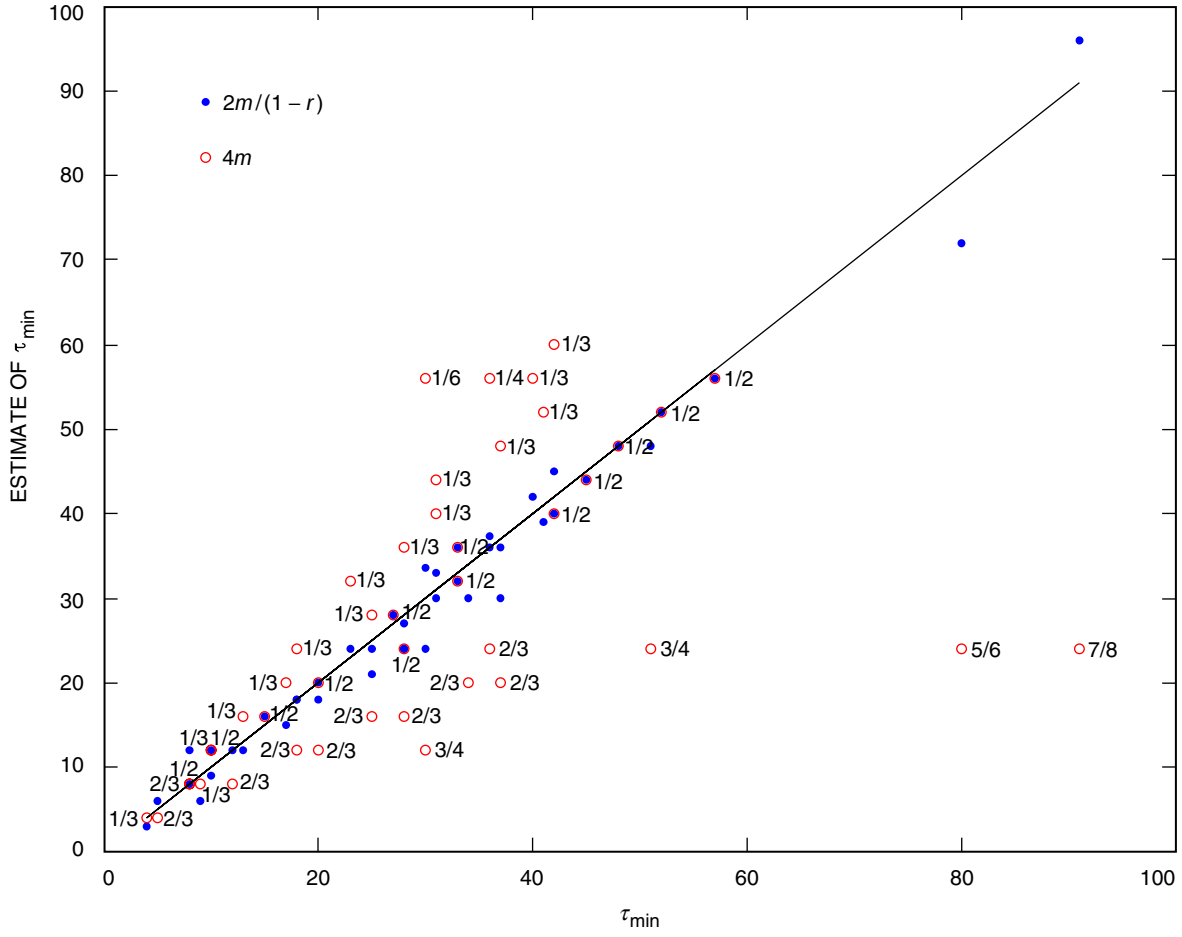

Fig. 3. Approximations $\hat{\tau}_{\min}(d) = 4m$, $\hat{\tau}_{\min}(d) = 2m/(1-r)$ **as a function of** $\tau_{\min}(d)$**. Approximation** $4m$ **points are labeled with the code rate** k/n**.**

References

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

[2] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, New Jersey: Prentice Hall, 1983.

[3] G. C. Clark, Jr., and J. B. Cain, *Error-Correction Coding for Digital Communications*, Applications of Communication Theory, New York: Plenum Press, 1981.

[4] J. G. Proakis, *Digital Communications*, 4 ed., McGraw–Hill Series in Electrical and Computer Engineering, New York: McGraw–Hill, 2001.

[5] G. D. Forney, Jr., "Convolutional Codes II: Maximum Likelihood Decoding," *Information and Control*, vol. 25, pp. 222–226, 1974.

[6] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," *IEEE Transactions on Communications*, vol. COM-19, pp. 751–772, October 1971.

[7] Consultative Committee for Space Data Systems, *Report Concerning Space Data System Standards: TM Synchronization and Channel Coding—Summary of Concept and Rationale*, CCSDS 130.1-g-1 ed., Green Book, June 2006.

[8] J. B. Anderson and K. Balachandran, "Decision Depths of Convolutional Codes," *IEEE Transactions on Communications*, vol. 35, pp. 455–459, March 1989.

[9] F. Hemmati and D. J. Costello, "Truncation Error Probability in Viterbi Decoding," *IEEE Transactions on Communications*, vol. COM-25, pp. 530–532, May 1977.

[10] I. M. Onyszchuk, "Truncation Length for Viterbi Decoding," *IEEE Transactions on Communications*, vol. 39, pp. 1023–1026, July 1991.

[11] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Series on Digital and Mobile Communication, Piscataway, New Jersey: IEEE Press, 1999.

[12] R. G. Gallager, *Information Theory and Reliable Communication*, New York: John Wiley & Sons, 1968.

[13] A. J. Viterbi, "Error Bounds for convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260–269, 1967.

Appendix
Bound on $e(r)/E(r)$

Consider a discrete memoryless channel with input distribution p and channel transition probabilities p_{jk}. Let $C = I(X;Y)$ be the mutual information of the channel (the fixed p capacity in [5]), and

$$E_0(\rho) = -\ln \sum_j \left[\sum_k p_k p_{jk}^{1/(1+\rho)} \right]^{1+\rho}$$

Gallager [12] illustrated that

$$E_0(\rho) \geq 0, \text{ with equality iff } \rho = 0 \tag{A-1}$$

$$C \geq \frac{\partial E_0(\rho)}{\partial \rho} > 0 \tag{A-2}$$

$$\frac{\partial^2 E_0(\rho)}{\partial \rho^2} \leq 0 \tag{A-3}$$

The convolutional code exponent is given by

$$e(r) = \sup\{E_0(\rho) | 0 \leq \rho \leq 1, \rho < \rho_r\}$$

where ρ_r is the parameter that satisfies $r = E_0(\rho_r)/\rho_r$. The block code exponent is given by

$$E(r) = \max_{0 \leq \rho \leq 1} E_0(\rho) - \rho r$$

and the ratio of the block code and convolutional code exponents may be expressed as [5]

$$\frac{E(r)}{e(r)} = \max_s \frac{1-r}{s} \frac{e(s)}{e(r)} \tag{A-4}$$

From Eq. (A-1), $e(r) \geq 0$, hence the maximum in Eq. (A-4) is always achieved for $s \geq r$ (otherwise $E(r) < 0$, a contradiction). Since $e(r) = 0$ for $r > C$, we may also limit $s \leq C$. It is straightforward to show that $e(r)$ is decreasing in r; hence, for any $s \geq r$ we have $e(s) \leq e(r)$ and

$$\frac{E(r)}{e(r)} \leq \max_{r \leq s \leq C} \frac{1-r}{s} \leq \frac{1-r}{C} \tag{A-5}$$

which we may loosely bound for a binary input channel ($C < 1$) as

$$\frac{e(r)}{E(r)} \geq \frac{1}{1-r}$$

In the construction of $e(r)$, when $r \leq E_0(1)$, the constraint $\rho < \rho_r$ is active and $e(r) = E_0(1)$. From Eqs. (A-1) through (A-3), we see that

$$E_0(1) \leq I(X;Y)$$

with equality if $H(X|Y) = 0$ [12]. In this case, $e(r) = C$ for $r \leq C$ and, from the concatenation construction [5], we have equality in Eq. (A-5). Hence, the bound is tight in the limit of very good channel fidelity.