Variable Length Short Constraint-Length Convolutional Codes: A Comparison of Maximum Likelihood and Sequential Decoding

J. W. Layland
Communications Systems Research Section

Maximum likelihood decoding of short constraint-length convolutional codes is one of the likely candidates for implementing high-performance telemetry systems for future deep-space missions. It has, in fact, been considered to be the best choice for video missions, providing better performance at the design point of 5×10^{-3} than other systems of comparable complexity. Recent advances in knowledge of sequential decoding have posed the question as to whether sequential decoding might, in fact, be preferable to maximum likelihood decoding. The answer, developed here in terms of a hypothesized maximum likelihood decoder built technologically similar to the JPL high-speed multi-mission sequential decoder, is that maximum likelihood decoding is preferable to sequential decoding at a 5×10^{-3} bit error rate. The reverse is true at 10^{-3} and below.

Two code families of variable constraint length are also developed which permit easy implementation of encoders for this hypothesized maximum-likelihood decoder.

I. Introduction

Maximum-likelihood decoding of short constraint-length convolutional codes is one of the likely candidates for implementing high performance telemetry systems for future deep-space missions. Proposed designs have variously ranged over constraint-lengths from five to eight. Almost always the proposed decoder is capable of decoding the longest code at the highest anticipated bit rate. A number of future missions will involve more than one planetary encounter with a significantly lower maximum data rate capability at the second encounter than at the first. The question arises in this circumstance whether it might be desirable to consider a coding system for which moderate coding gain is achieved at the highest bit rate, but with significantly better performance at lower data rates. This article proposes characteristics for such a decoder, presents codes which might be used with it, and compares its performance with that of a sequential decoder built using the same technology.

By way of example, consider a possible mission to Jupiter, Uranus, and Neptune at the end of this decade, having a maximum data rate of 10^5 bits/s at Jupiter. The proposed decoder would operate at constraint length $K=5$ for the Jupiter encounter. The usable constraint length would increase to $K = 8$ at Uranus and $K = 10$ at Neptune. At 5×10^{-3} BER, this increase in constraint length is comparable to an increase of 0.85 dB in bit signal-to-noise ratio at the Neptune encounter.

II. The Proposed Decoder

Most decoders for short constraint-length convolutional codes built or proposed to date are of one of two types, either a fully parallel multiple processor device capable of very high-speed operation (Ref. 1), or a single processor device using a somewhat larger constraint length code, having a fast enough semiconductor memory to permit operation at the desired constraint length and data rate. Suppose instead of asking for the fastest technically feasible memory, the designer of the single-processor decoder had asked for the fastest mass-produced complete memory system. Today, the answer to such an inquiry would be a core memory system with approximately a 1-μs cycle time. A decoder implemented with this memory would operate perhaps one fifth as fast as a decoder with high-speed semi-conductor memory. In exchange for this decrease in speed, the designer would have obtained the use of a mature and very reliable memory technology; technology where he could easily obtain memory of the size needed to decode codes of constraint length 12 or longer; technology where the difficulty of testing and maintaining the memory is a very slowly increasing function of the memory size. He could almost ignore memory size as a constraining factor in determining the longest constraint length that his decoder would decode.

Choosing the slower core memory would naturally reduce the upper limit of data rates which could be decoded. It would also reduce the constraint length of the code which could be decoded at the highest data rate designed. However, depending upon mission parameters, the bigger but slower core could increase the usable constraint length, and hence increase the coding gain at the lower data rates occurring later in the mission where the coding gain may be more needed. Let us examine the decoder's behavior. The convolutional encoder at constraint-length k can assume 2^{k-1} possible "states." For each encoder state, the decoder must record a likelihood and a most-likely data path leading to that state. Data for the possible encoder states are accessed two-at-a-time for decoding computations, and data can be arranged so that the pairs of states can be simultaneously accessed from separate memory units. Thus, 2^{k-2} computations are required to decode one bit of a constraint length-k code. Using TTL logic, the decoder computations could be easily performed within the memory cycle in which the needed data is fetched. The hypothetical decoder is thus able to decode one bit of a constraint length k code in 2^{k-2} μs.

To see what this means in terms of code performance, examine the curves shown in Fig. 1 where the SNR needed to achieve a fixed design-goal bit error probability is plotted with respect to data rate, assuming that the decoder is operating at as long a constraint length as possible for each data rate. The performance-intercepts for the various codes were obtained from previously published simulation results (Ref. 2). Performance-intercepts for several other existing or proposed coding systems are included for comparison.

III. Variable Constraint-Length Codes

The codes displayed in Fig. 1 are in all cases the best-known codes at each constraint length. They do not in any sense form a family with common characteristics, so implementation on a spacecraft would require almost a complete encoder for each constraint length. Codes which form a nested family could, however, be implemented with little more logic than a single encoder of the longest constraint length involved.

The rate 1/2 quick-look code devised by Massey (Ref. 3) is a code of this type. It was defined for a constraint length $k = 48$ bits, but can be truncated to $k < 48$ by the addition of coder taps at depth k to both generators and the deletion of all taps beyond k. Very slight modifications to Massey's algorithm are needed to produce a rate 1/3 quick-look code. At $k = 32$, the tap matrix of the resultant code is

$$7630\ 7777\ 0000\ 7707\ 0700\ 7700\ 0077\ 0777 \tag{1}$$

In Eq. (1), 7s represent taps at that depth on all generators, the 6 represents taps on the first and second generators at delay one, and the 3 represents taps on the second and third generator at delay two. This code is quick-look in the sense that it allows two independent estimates of each bit, each one of which requires only one mod-2 operation. The direct inverse for this code has the lowest possible error probability of any nonsystematic rate 1/3 code. The code can be truncated to any depth k, by insertion of a 7 there, and setting all entries beyond k to 0. This code has been simulated in the vicinity of 5×10^{-3} bit error rate at various constraint lengths. The results are plotted in Fig. 2.

Another family of nested codes has been hand constructed with the aid of the hill-climbing code construction software (Ref. 2). We observe, first of all, that the best $k = 4$ code is nested within the best $k = 5$ code. Starting at that $k = 5$ code, a $k = 6$ code was constructed by inserting a 0 into the tap matrix at depth 5, a 7 at depth 6, and asking the hill-climb software to insert taps—hopefully at depth 5. The process then proceeded to $k = 7$, $k = 8$, etc., and involved frequent back tracking, and compromise betweer performance at different constraint lengths. At $k = 11$, the tap matrix of the resultant code is

$$75651234357 \tag{2}$$

The result was a satisfying rather than optimizing one, and could doubtless be improved; but even so, the family of codes performs very close to the non-nested set of "best" codes in the vicinity of 5×10^{-3} error rate. The results of simulating several members of this code family are shown in Fig. 3. In that region, the codes which were compromised to improve others appear to be no worse than 0.1 dB below the best codes (Ref. 2, Fig. 27), while the best members of the nested code family are essentially equal in performance to the overall best code of the same length.

Fig. 4 shows the E_b/N_0 required to achieve a 5×10^{-3} bit error probability as a function of data rate for the three sets of rate 1/3 codes discussed here using the decoder with 1-μs computation time. The quick-look codes perform significantly poorer than the other two types, and could reasonably be dropped from consideration for short constraint-length application. The nested non-quick-look codes appear to be a good choice if a corresponding spacecraft encoder is to be implemented at several constraint lengths.

IV. Comparison to Sequential Decoding

The performance variation over the (varying constraint-length) set of convolutional codes is analogous to the performance variation of a sequential decoder with decreasing data rate and increasing speed advantage. Recent work[1] has related the performance of an optimally buffered sequential decoder to the decoding unit speed only, and has dispelled the illusion that arbitrary increases in buffer size can produce arbitrary improvement. A hardware sequential decoder exists at JPL which can perform a decoding computation in 1 μs (Ref. 4). This decoder was constructed from approximately the same technology as the decoder for short constraint length codes proposed in *Section II.*

Let us hypothesize, for comparison purposes, a sequential decoding machine using optimum buffer management and capable of performing one computation in 1 μs. The amount of logic necessary to implement this decoder is judged by this writer to be approximately equal to the logic required to implement the decoder of *Section II.* Assuming that this is true, the comparison between the two types of codes is directly meaningful. The speed of either decoder could, doubtless, be pushed upward by factors of 2 to 4 by clever design, at some increase in cost, and, perhaps, flexibility. The 1-μs figure for both represents a straight-forward design of unarguable feasibility. At the very least, comparison with that figure provides a reference point from which more detailed studies could start, if needed.

The proper comparison point corresponds to an error probability for the optimum decoder of P_E, and a deletion probability for the sequential decoder of approximately $2 * P_E$, since on the average, between ¼ and ½ of the bits in a deleted block will be in error. The graphical comparison in Fig. 5 is at 5×10^{-3} and 5×10^{-4} bit error probability. Data for these curves was extracted from Fig. 1, and from Fig. 5 in "Performance of an Optimum Buffer Management Strategy for Sequential Decoding," by J. W. Layland, in this issue. At the higher failure rate, the short constraint-length codes and decoder appear preferable, unless the customer for the data wishes to use the fact that nondeleted blocks from the sequential decoder have an error probability of 10^{-6} or less. At the lower error probability, the sequential decoder performs better, owing, of course, to the greater steepness of the sequential decoding deletion probability versus signal-to-noise ratio (SNR) curves.

Note that the preferences just observed do not change if either of the decoders is sped up small amounts by clever design!

V. Summary

This article has developed a hypothesized convolutional decoder capable of optimum decoding of "short" constraint length codes of perhaps length 12 or more; has developed a workable nested family of codes such that an encoder for all constraint lengths from $K = 4$ to $K = K_m$ is little more complex than an encoder for $K = K_m$ alone; and has compared the performance of this hypothesized

[1]See "Performance of an Optimum Buffer Management Strategy for Sequential Decoding," by J. W. Layland in this issue.

decoder with a sequential decoder built with comparable technology. The presented curves show clearly that if the design failure probability is 10^{-3} or less, the sequential decoder is somewhat superior, whereas with the 5×10^{-3} error rate at which conventional deep-space video missions operate, the softer threshold of the short-constraint length codes means superior performance with the decoder described here.

References

1. Lushbaugh, W. A., "Hardware Version of an Optimal Convolutional Decoder," in *The Deep Space Network Progress Report,* Technical Report 32-1526, Vol. II, pp. 49–55. Jet Propulsion Laboratory, Pasadena, Calif., April 1971.
2. Layland, J. W., "Performance of Short Constraint Length Convolutional Codes and a Heuristic Code-Construction Algorithm," in *The Deep Space Network,* Space Programs Summary 37-64, Vol. II, pp. 41–44. Jet Propulsion Laboratory, Pasadena, Calif., Aug. 31, 1970.
3. Massey, J. L., and Costello, D. J., "Non-Systematic Convolutional Codes for Sequential Decoding in Space Applications," *IEEE Trans. Commun. Tech.,* Vol. COM-19, No. 5, pp. 806–813, October 1971.
4. Lushbaugh, W. A., "Multiple Mission Sequential Decoder," in *The Deep Space Network,* Space Programs Summary 37-58, Vol. II, pp. 33–37. Jet Propulsion Laboratory, Pasadena, Calif., July 1969.

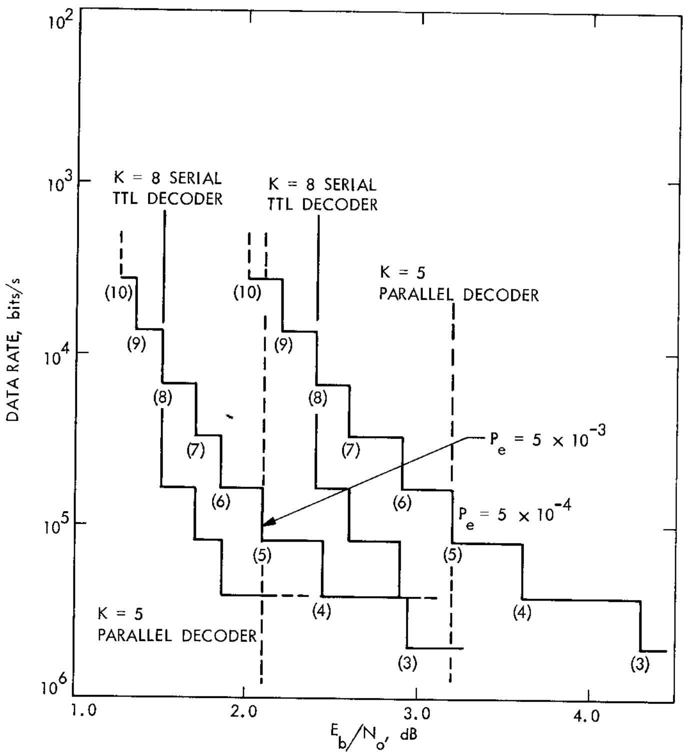

Fig. 1. Signal-to-noise ratio required to achieve desired error probability versus data rate for hypothesized decoder

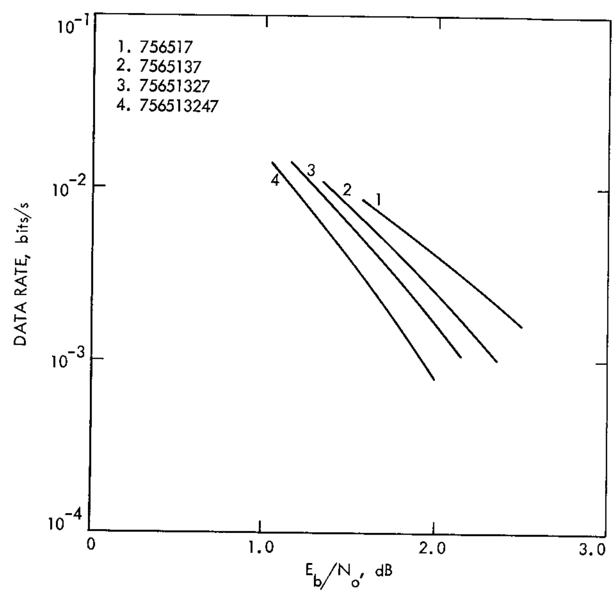

Fig. 3. Performance of a nested family of rate ⅓ convolutional codes

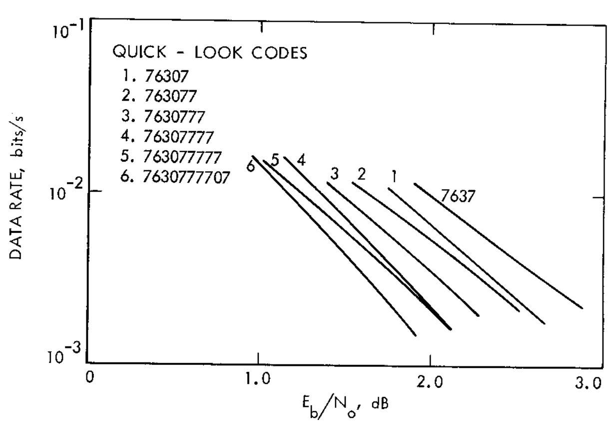

Fig. 2. Performance of a quick-look rate ⅓ convolutional code at various constraint lengths

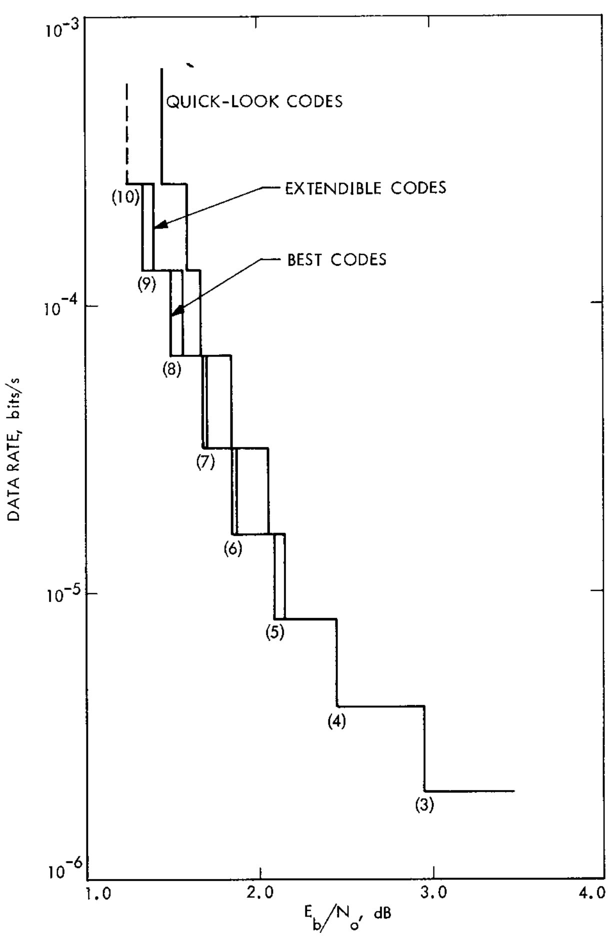

Fig. 4. Signal-to-noise ratio required to achieve 5 × 10^{-3} bit error probability versus data rate for different codes

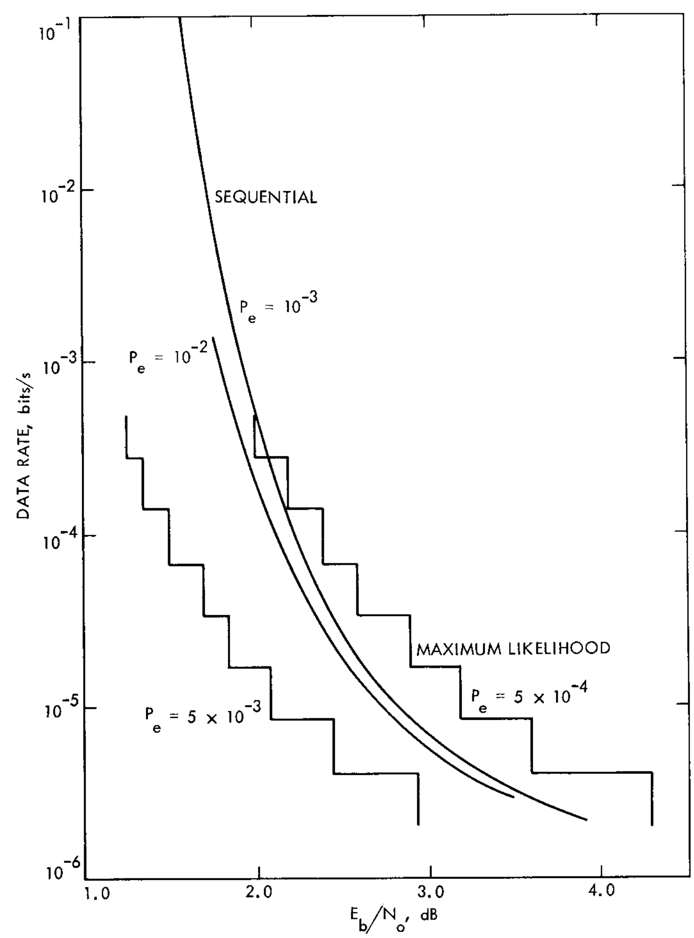

Fig. 5. Comparison of sequential and maximum-likelihood decoding