Soft Decision Decoding of Block Codes

L. D. Baumert and R. J. McEliece
Communications Systems Research Section

The performance of certain block codes on a gaussian channel is evaluated. Two of these codes, the BCH codes of rates 1/2 and 1/3 and length 128, are markedly superior to the constraint length 7 rate 1/2 convolutional code currently used for deep space missions. The algorithm used to derive these results provides a basis for a simple, almost optimum procedure for decoding these codes.

I. Introduction

If one wishes to improve the performance of deep space telemetry beyond the capabilities of short constraint length convolutional codes currently in use, perhaps the most promising approach involves the soft decision decoding of block codes. With this in mind we have employed a general decoding process of Solomon, called "decoding with multipliers" (Ref. 1), to evaluate the performance of certain block codes on a gaussian channel. Two of these codes, the BCH codes of length 128 and rates 1/2 and 1/3, show marked superiority over the constraint length 7 rate 1/2 convolutional code currently in use.

Earlier block code simulations for a gaussian channel are described in Refs. 2-4.

II. Soft Decision Decoding

If a binary quantizer is added to a gaussian channel its capacity is diminished by a factor of $\pi/2(\sim 2$ dB). Thus, optimal code performance on a gaussian channel cannot be achieved by a hard limiting decoding process.

By *soft decision decoding* we mean any decoding process which makes use of the relative magnitudes of the received code symbols. In this article we use a particular soft decision decoding process of Solomon which applies to all linear codes and has the advantage that it does not require the use of a binary decoding algorithm. Thus, it is perfectly general and can be used, for example, to decode the quadratic residue codes.

Suppose a codeword $w = w_1, \ldots, w_n$ from a binary (n,k) linear code is transmitted over a memoryless zero mean gaussian channel. The code symbols are assumed to be ± 1. Thus a received codeword is a sequence of n real numbers, each number representing the integrated value of $w_i + \eta_i$ over one symbol time $-\eta_i$ being the noise. Since the noise is zero mean, the absolute value of these real numbers is a measure of their reliability (the larger the absolute value, the higher the probability that the hard limited version of the real number actually is w_i). In the decoding technique used, the j least probably correct symbols are determined and excluded from consideration. Then k of the remaining $n - j$ symbols are assumed to have proper sign and all codewords (if any) of the code whose signs agree in these k places are constructed. The codewords so

constructed are correlated with the sequence of real numbers from the receiver. Several choices of k of the remaining $n - j$ symbols are made. Among all the codes words thus generated, that one having the maximum correlation value is assumed to have been sent.

III. Simulation Results

The number $Q = (k/n) \cdot d$, where d is the minimum distance of a linear code, is a measure of its asymptotic decoding behavior, since

$$\lim_{\gamma \to \infty} \frac{1}{\gamma} \log P_e = -Q$$

where γ is the bit signal-to-noise ratio and P_e is the bit error probability. Thus, the bigger Q is the better the code should perform, at least for large γ.

As a point of reference our performance graphs show curves for "no coding" and for the maximum likelihood performance of the 7, 1/2 convolutional code being used on NASA's Mariner-class spacecraft (Ref. 5).

A. The (48,24) Quadratic Residue Code

This code is a 5-error-correcting block code with $Q = 6.0$. Two different decodings of this code were simulated. The first of these had $j = 8$, and 130 k-tuples of positions were selected from the remaining $n - j = 40$ positions in such a way that all $\binom{n-j}{4}$ of the 4-tuples were omitted from at least one of these k-tuples. Thus, if there were no more than four hard decision errors among the 40 positions most probably correct, the decoding process would necessarily construct the correct codeword. So, in that instance, the only possible errors would also be made by a maximum likelihood decoder.

The second decoding was done for $j = 16$ with 124 k-tuples such that all 3-tuples from the $n - j = 32$ remaining positions were excluded from at least one of the 124. This performed better, as is shown in Fig. 1. In fact, since about 90% of the errors in this simulation were *identifiably* maximum likelihood errors, it is reasonable to conclude that this decoding is essentially a maximum likelihood decoding of the (48,24) quadratic residue code.

B. The (80,40) Quadratic Residue Code

This code is a 7-error-correcting code with $Q = 8.0$. Five decodings were simulated for this code. The best performance occurred for two different decodings. The first of these had $j = 28$ and 130 k-tuples from the remaining 52 positions such that all $\binom{52}{3}$ triples of positions were omitted from k-tuple. The other used $j = 36$ and 165 k-tuples from the remaining 44 positions such that all pairs of positions were omitted from some k-tuple. Figure 2 shows the performance curve for these two decodings as well as our estimate of the maximum likelihood behavior of the code. This estimate is based on the identifiably maximum likelihood errors which occurred during the simulation.

C. The (128,64) BCH Code

This is a 10-error-correcting block code with $Q = 11.0$. Several different decodings were tried here. We tried k-tuples for $j = 40$, 55 and 56 such that 3, 2 and 2 errors would be allowed among the 88, 73 and 72 remaining most reliable positions. The results were not very good. Note that this kind of selection of the k-tuples really divides the received symbols into two classes: $n - j$ symbols among which k are sought with the correct sign and the other j. Within these classes, all symbols are treated the same – just as if they were equally likely to be in error. This is, of course, not a valid assumption. We tried a further breakdown of the $n - j$ symbols in one decoding as follows: $j = 40$, $n - j = 88$ and these 88 symbols were ordered by magnitude to establish their relative error probability. The 16 most likely correct were assumed to be correct; two errors were allowed among the next 40 most likely correct positions and four errors were allowed among the remaining 32. This performed better than the simpler collections of k-tuples mentioned above.

The improved performance in this last decoding suggested that we might try to match the finer gradations of error probability among the symbols a little more closely. We did this by gathering (at 1.5 dB's) the error frequencies of the least likely correct, . . . , most likely correct symbols. Then we constructed k-tuples by a random placing of 1's approximately in accordance with the entropy of these error frequencies. This collection of k-tuples performed much better than the previous collections, and when 1500 of them were used, the identifiably maximum likelihood errors predominated; so much so that it is reasonable to assume that the performance achieved (Fig. 3) is within 0.1 dB of maximum likelihood behavior.

As a by-product of this simulation we also determined that the number of minimum weight (=22) codewords of this code is almost certainly 243,840 (see Ref. 6 for details of this determination).

D. The (128,43) BCH Code

This is a 15-error-correcting code with $Q = 10.75$. It was decoded using 1500 k-tuples selected to fit the observed error statistics at 1.5 dB, as described in more detail above for the (128,64) code. Within the accuracy of this simulation the performance of this code (Fig. 4) is the same as the (128,64) code and seems also to be within 0.1 dB of its maximum likelihood behavior.

Much of the analysis of Ref. 6 applies to this code also, and using it we can conclude that the number of minimum weight (=32) codewords of this code is almost certainly 124,460.

References

1. Baumert, L. D., McEliece, R. J., and Solomon, G., "Decoding with Multipliers," in *The Deep Space Network Progress Report 42-34*, Jet Propulsion Laboratory, Pasadena, Calif., Aug. 1976, pp. 43-46.
2. Chase, D., "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Inform. Th.*, IT-18 (1972), pp. 170-182.
3. Baumert, L. D., and McEliece, R. J., "Performance of Some Block Codes on a Gaussian Channel," Proc. 1975, International Telemetry Conference, Washington, D.C., pp. 189-195.
4. Chase, D., and Goldfein, H. D., "Long Block Codes Can Offer Good Performance," Information Theory International Symposium, Cornell University, October 1977.
5. Webster, L., "Maximum Likelihood Convolutional Decoding (MCD) Performance Due to System Losses," in *The Deep Space Network Progress Report 42-34,* Jet Propulsion Laboratory, Pasadena, Calif., Aug. 1976, pp. 108-118.
6. Baumert, L. D., and Welch, L. R., "Minimum Weight Codewords in the (128,64) BCH Code," in *The Deep Space Network Progress Report 42-42,* Jet Propulsion Laboratory, Pasadena, Calif., Dec. 1977, pp. 92-94.

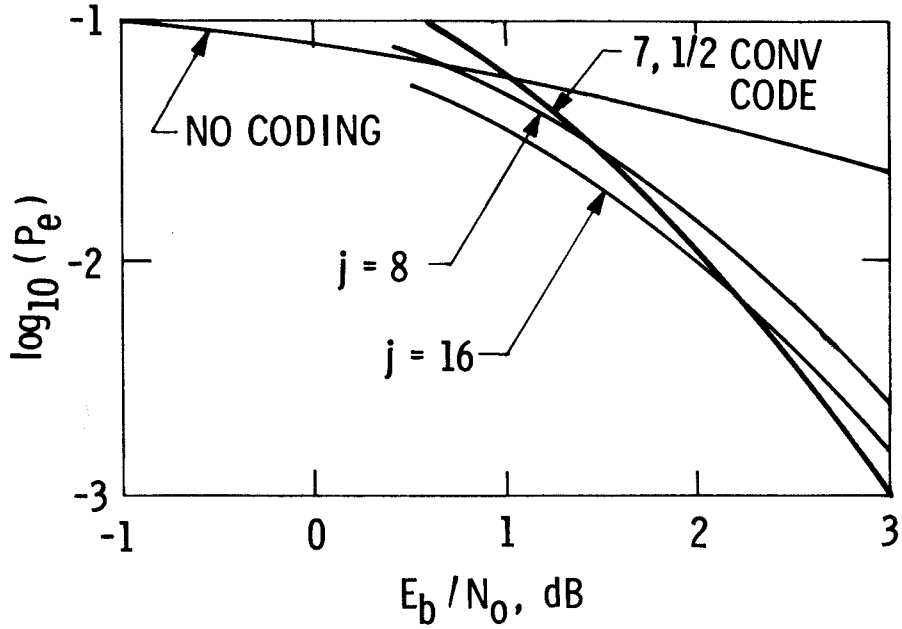

Fig. 1. (48,24) QR, $Q = 6.0$

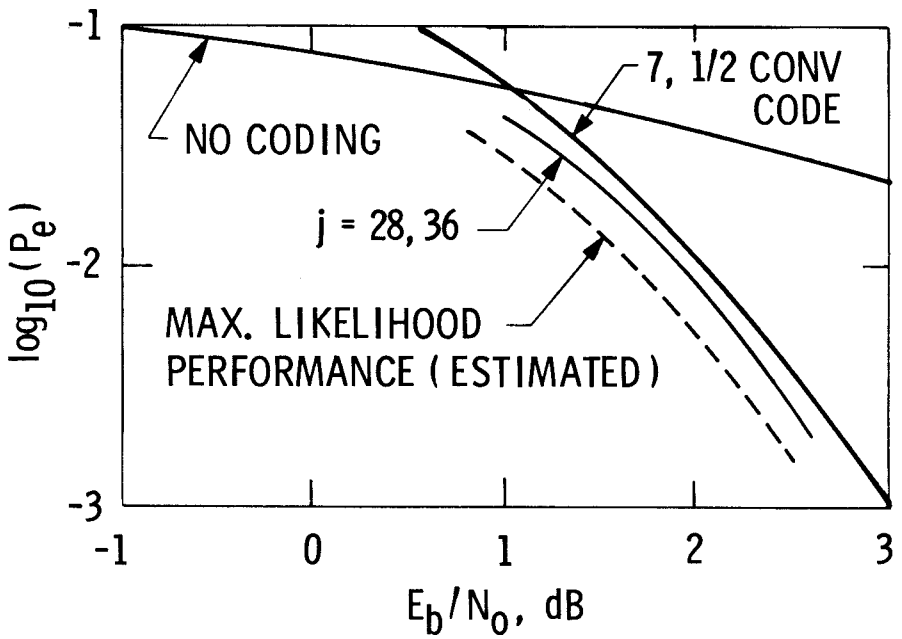

Fig. 2. (80,40) QR, $Q = 8.0$

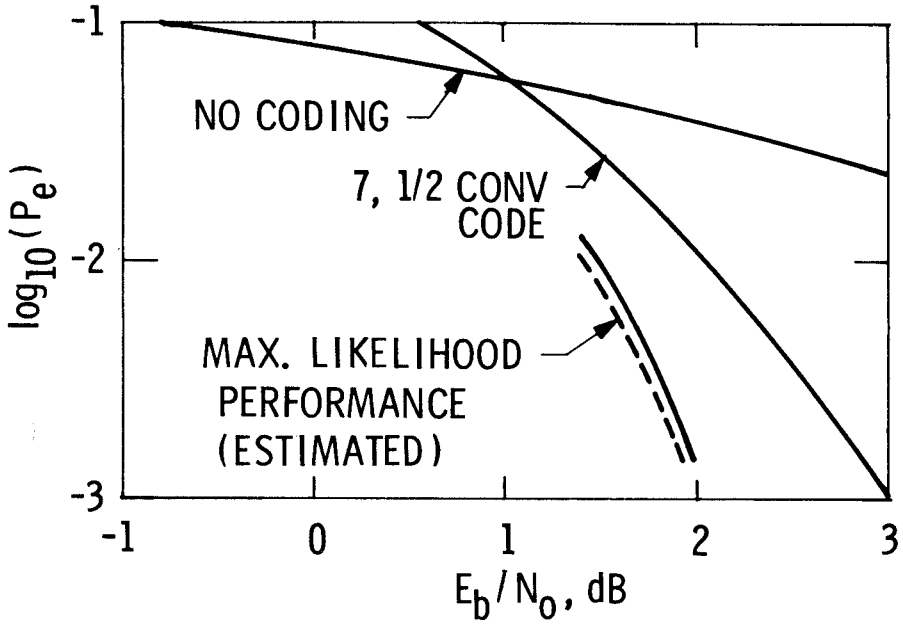

Fig. 3. (128,64) BCH, $Q = 11.0$

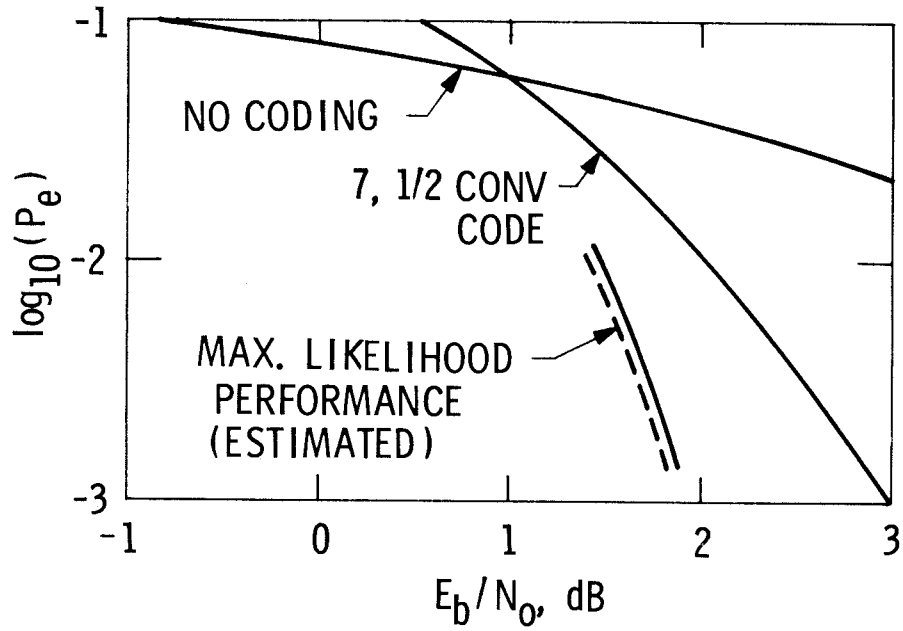

Fig. 4. (128,43) BCH, $Q = 10.75$