A STUDY OF CODES FOR DEEP SPACE TELEMETRY

By D. R. Lumb
Ames Research Center, NASA
Moffett Field, California 94035

Summary

Results of computer studies are described for several codes which have promise for application to the deep space telemetry channel. Two rate 1/2 trial-and-error convolutional codes and two cyclic block codes have been simulated using majority decision decoding for the binary symmetric channel and "a posteriori probability" (APP) decoding for the gaussian channel. In addition to APP decoding, sequential decoding was examined for a rate 1/2 short-constraint-length convolutional code. The codes were considered from an energy efficiency standpoint for coherent detection of PSK signals. Code performance measures include bit error probabilities, undetected word error probability for six-bit telemetry words, and also detected and undetected word error probabilities when the codes are concatenated with a (7, 6) parity check code.

Introduction

The purpose of introducing coding into the telemetry link of a scientific deep space probe is to permit an increased information rate at some allowable error rate for a fixed amount of effective radiated power. The resultant gain from a coding scheme for a deep space probe must justify the required spacecraft encoder and ground decoding hardware. Thus, it is desirable to consider schemes that either require minimum changes to hardware for an existing program or do not require excessively complex hardware for a new program.

This paper reports the results of computer simulation studies of several codes that may be applicable to deep space telemetry links. Measures of performance and criteria for comparison were directed toward telemetry links which transfer predominately scientific measurements of space phenomena. Although the work was done for potential application to the Pioneer Program, most of the results are of greater generality. However, with this application in mind some system constraints are placed on the classes of coding approaches that may be considered.

System Considerations and Constraints

Code performance depends on the type of channel that is considered. Therefore, a few remarks concerning the characteristics of the S-band deep space telemetry channel will be made, based on experimental tests.* The carrier tracking portion of the Deep Space Station (DSS) radio receivers are basically second-order phase-lock-loops (PLL); therefore, coherent demodulation of the baseband telemetry signal from the phase modulated RF carrier is affected by the carrier tracking PLL signal-to-noise ratio. For a system of fixed phase modulation index operating at intermediate ranges, the phase demodulation of the telemetry signal from the carrier is coherent, since the carrier reference PLL is operating with a strong signal. However, close to the maximum design range the carrier PLL nears "threshold," and hence provides a noisy reference for demodulation of the baseband telemetry signal. The noise characteristics into the data demodulator depend on the PLL parameters, particularly the loop bandwidth. For strong PLL signals, the sampled matched filter noise output for PSK signals has been verified to be independent, additive, and gaussian. In the region near threshold, the overall distribution of the matched filter outputs is still gaussian, but the noisy reference causes considerable intersymbol influence, i.e., the channel exhibits memory. Except in the region of signal-to-noise ratio corresponding to the maximum design range, the classic independent additive gaussian channel is an adequate model for simulations.

It was desired to apply coding techniques to systems that utilize state-of-the-art modulation and demodulation schemes. These may be typified for the approximately gaussian S-band deep space channel by the coherent PSK system as mechanized for the Pioneer program. This system can provide sampled and quantized matched filter outputs for a decoder. For rates as low as 16-bits per second, tests have indicated that with minor equipment modifications, coherent demodulation can be accomplished for $E_S/N_O \geq 0$ dB with a mechanization loss of less than one dB from theoretical. (E_S/N_O is the signal energy per transmitted bit per noise spectral density.) This then sets a bound on the code rates that can be considered. In order to achieve a coding gain of, say, 3 dB for a bit error rate requirement of 10^{-5}, the lowest rate codes that can be considered are about rate 1/2.

Another constraint is placed on the encoder complexity. Since the encoder for the telemetry link is in the spacecraft, it is desirable to keep the encoding function as simple as possible for reasons of weight, power, and reliability. The longer the constraint or block length the more complex the encoder becomes; hence, only relatively short length codes are appropriate.

An additional consideration in selecting coding approaches is the method of decoding. For deep space telemetry, the receiving stations have general purpose computers which are usually used for decommutation of selected scientific and

*Results of test runs with DSS receiving equipment and a Pioneer PSK demodulator detailing the statistics under various signal-to-noise ratios and bit rates, and also giving sequential decoding results for such a channel, will be reported in a NASA publication.

engineering data for real-time monitoring. Provided the additional computation load is not too heavy, this computer may be employed for the decoding function. Hence, only codes requiring relatively small amounts of decoding computation are appropriate.

As a result of the above constraints and considerations, threshold decoding and sequential decoding were studied.

Threshold Decoding

Threshold decoding of both convolutional and cyclic block codes has been described by Massey.[1] Two types of threshold decoding were derived: majority decision decoding, and a posteriori probability (APP) decoding. For the binary symmetric channel, Massey computed the probability of incorrectly decoding the first information bit for majority decision and APP decoding, from which the bit error rate may be estimated. Majority decoding is particularly attractive from a hardware mechanization point of view; however, the theory indicates its performance for the deep space telemetry application is inadequate. APP decoding for the gaussian channel was suggested by Massey to improve performance at the expense of complexity, but the amount of improvement was not known.

The (73, 45) and (15, 7) cyclic block codes, which have been found to be threshold decodable,[2] were the block codes that were majority and APP decoded. The parity bits for these codes are generated by a linear feedback shift register which computes the parity bits by

$$i_n = i_{n-3} + i_{n-9} + i_{n-16} + i_{n-20} + i_{n-21} + i_{n-35} + i_{n-43} + i_{n-45}$$

where $n = 46, 47, \ldots, 73$, $i_1 \ldots i_{45}$ are information bits and $i_{46} \ldots i_{73}$ are parity bits for the (73, 45) code, and

$$i_n = i_{n-1} + i_{n-3} + i_{n-7} \quad n = 8, 9, \ldots 15$$

for the (15, 7) code.

In this paper there will be no attempt to derive the theory of threshold decoding or to explain it in detail since it is adequately covered in reference 1, but rather a few remarks will be made about the basic principles.

For the binary symmetric channel, a set of J orthogonal A equations is evaluated, where each A equation consists of the error term for the bit to be decoded, as well as other error terms, but none of the other error terms occur more than once in the set of A equations. In equation form the rule for error correction is $e_1^i = 1$ if

$$\sum_{n=1}^{J} A_n > \frac{J}{2} \tag{1}$$

where e_1^i is the error term of the decoded bit.

For a channel when all the received bits do not have the same error probability and each probability is known at the receiver, this information can be used to extend majority decoding to the so-called APP decoding. The same code that is majority decision decodable may be APP decoded by the following general rule. Choose $e_1^i = 1$ if

$$\sum_{n=1}^{J} w_n A_n > T \tag{2}$$

where $T = 1/2 \sum_{i=0}^{J} w_i$; w_i are weights which are functions of the probabilities of the received bits; and T is the threshold.

The principle of majority decision and APP decoding of trial-and-error convolutional codes is the same as for cyclic block codes; namely, a set of orthogonal A equations is evaluated for each bit decoded and rule (1) or (2) is used.

The (24, 12) and (44, 22) trial-and-error convolutional codes[1] were simulated by computer using both majority decoding and APP decoding for the gaussian channel. These codes were chosen since they have relatively short constraint length and their performance will be taken as representative of this class of codes.

The encoders for the (24, 12) and (44, 22) codes consist of 12 and 22 stage tapped shift registers, respectively, for the computation of parity. For every information bit into the encoder a parity bit is computed according to the rule

$$p_n = i_n + i_{n-6} + i_{n-7} + i_{n-9} + i_{n-10} + i_{n-11}$$

$n = 1, 2, \ldots$ and $i_{n-k} = 0$ for $n-k \leq 0$

for the (24, 12) code and

$$p_n = i_n + i_{n-11} + i_{n-13} + i_{n-16} + i_{n-17} + i_{n-19} + i_{n-20} + i_{n-21}$$

$n = 1, 2, \ldots$ $i_{n-k} = 0$ for $n-k \leq 0$

for the (44, 22) code.

Measures of Performance

For comparing codes, unifying measures of performance are desirable. These measures should reflect the parameters of data accuracy most signigicant to the spacecraft experimenter. For block codes, the block error probability usually is calculated and for convolutional codes with threshold decoding the probability of first error is determined. From these the bit error probability may be estimated. For error correcting coding schemes, once an error is made there is a tendency to decode several bits in error. But this error burst effect differs with the coding technique and may affect the user differently, depending on his minimum unit of information. In most cases a

measurement for a scientific experiment is quantized into several bits. Particle and field type experiments (i.e., plasma, cosmic ray, and magnetometer measurements) generally use at least one telemetry word as a minimum unit of information. For Pioneer this is a 6-bit word with a parity check bit added. With a word as a unit of information and a parity check scheme as a reference, two parameters are of interest from the experimenter's point of view. One is the deletion rate determined by parity detection and the second, and most important, is the undetected word error rate.

Thus, the 6-bit word error rates are tabulated in addition to the usual bit error rates. The encoded data source was also assumed to consist of 6 bits of information plus a parity check bit. For this case the errors in the decoded data stream were tabulated according to the proportion of odd and even numbers of errors in 7-bit word groupings, thereby giving the parity detection and undetected error rates, respectively.

Performance of Threshold Decoding

Methods of Feedback

For threshold decoding, performance is influenced by the type of feedback strategy used. Two types of feedback with APP decoding on the gaussian channel were tested on the (15, 7) and (73, 45) codes, namely, hard decision feedback and full APP feedback.* Hard decision feedback assumes that after a bit is decoded its error probability is zero. This means that the terms used in computing the weighting factors, w_i, for decoding are appropriately modified after each bit is decoded. Full APP feedback implies that after each bit is decoded the bit error probability determined during decoding is fed back to the appropriate terms in the w_i. It should be noted that the bit error probability after each bit is decoded depends strongly on the bit error probability of many other bits. Nevertheless, the full APP decoding method for cyclic block codes is shown in the next section to perform somewhat better than hard decision feedback APP. Only the hard decision feedback was used for the convolutional codes tested. However, recently Dr. G. D. Forney, Jr., has shown that full APP feedback for convolutional codes gives several tenths of a dB advantage over hard decision feedback.

Decoding Simulations

Threshold decoding experiments were performed on an IBM 7094 computer. For APP decoding, the simulated gaussian channel, using matched filter detection of PSK signals, was represented by a sampled and 6-bit quantized output. This output is a gaussian random variable that is proportional to the bit log likelihood ratio, which is the central quantity used in computing the weighing factors, w_i, for APP decoding. Quantization to 6-bits was used since finer quantization was shown to achieve negligible additional gains.

For low values of E_s/N_o, about 0 to 2 dB, a large number of potential error causing events are encountered. However, for higher E_s/N_o, a very long bit stream would have to be examined to obtain a sufficient statistical sample, which would require an excessive amount of computer time. Therefore, code performances for high E_s/N_o were determined by examining only potential error causing situations.

(15, 7) Cyclic Block Code

Figures 1 and 2 show on a bit and on a word error probability basis that APP decoding has a large error reduction capability beyond that of unweighted (majority decision) decoding. This comes from the fact that APP decoding makes efficient use of the received bit log likelihood ratios. On the other hand, while majority decoding always corrects two errors in the (15, 7) code, APP decoding will sometimes make output errors in such cases, which accounts for as much as 20 percent of the errors made for $E_b/N_o = 4.4$ dB (E_b/N_o is the signal energy per information bit transmitted per noise spectral density). This effect gets worse for lower E_b/N_o, which means that eventually, the curves for APP and majority decoding would intersect. At this point, the decoded error rate is so high that neither decoding scheme would be of any use. Figure 1 shows that majority decoding has a very limited gain over no coding of only 1 dB at a bit error probability of 10^{-4}, and the coding gain reduces rapidly as E_b/N_o decreases. APP decoding tends to keep the coding gain constant over the region of interest, with a gain of about 2.6 dB over no coding. Note that there is a small improvement in performance for full APP compared to hard decision feedback decoding.

Since the simple seventh bit parity check (7, 6) code seems to work so well when a small deletion rate is permitted, its effect was investigated when used with the different codes. As figure 2 shows, the performance, which includes the rate loss of 0.67 dB, is slightly worse by 0.2 dB but the deletion rate with coding and the 6-bit plus parity bit data is negligible compared to that of the seventh bit parity check code alone.

It will be noted that with cyclic block codes, such as the (15, 7) code, it is possible to decode the parity bits first, and then the information bits. This would be a useful technique for the full APP decoding, since the word error probability for the second 7-bit word is about 1/2 that of the first word. The APP hard decision feedback decoding has the same word error probability for the two 7-bit words, and it would be wasteful of computer time to decode the parity check bits also.

(73, 45) Cyclic Block Code

The performance curves of this code are shown in figures 1 and 3. Most of the remarks made for the (15, 7) code apply also to the (73, 45) code,

*This feedback strategy was suggested by Dr. G. David Forney, Jr., of Codex Corporation who is performing supplementary coding studies for Ames Research Center under Contract NAS2-3637.

except that its performance is considerably better. There is also a clear advantage compared to the seventh bit parity check code, at least as far as APP decoding is concerned. For E_b/N_o = 4.4 dB (corresponding to a channel bit error rate of P_e = 3 percent), it is clearly established that the full APP decoding is superior to hard decision APP decoding. This was, therefore, the only method used for the 1 percent runs, since these runs must be longer than the 3 percent runs to get statistically significant data.

Figure 4 shows the word error probability normalized to the first word as a function of the word position in the block for various decoding systems. The figure illustrates the effect of the different types of feedback. Data points for majority decision decoding in which the original bit decision is fed back are not shown since they were not measured. But, it is clear that the corresponding curve (fig. 4) should be the horizontal line I, since the word position in the block will not affect any coding decisions. Curve II shows the beneficial effect of hard decision feedback for majority decoding. This effect becomes even more pronounced for APP decoding with hard decision feedback (curve III). But the improvement levels off rapidly. Curve IV shows full APP decoding for identical data. As expected, it starts out like the one for hard decision APP feedback, but the improvement continues all through the block. This clearly shows that it is worthwhile to decode the parity check bits first and then the information bits. It should be noted that figures 2 and 3 give the performance for decoding all of the received bits, both information and parity. The additional improvement from the extra effort of decoding the parity bits first for the (73, 45) code can be estimated from figure 4.

Alarm Reset for Majority Decision Decoding of Convolutional Codes

For threshold decoding of convolutional codes, once a bit is decoded in error the next successive bits have a relatively high probability of being decoded incorrectly. This error propagation effect can be minimized for majority decision decoding by error counting. That is, when the decoder is "correcting" more than the correcting capacity of the code, an "alarm" for feedback resetting of terms in the A equations can be used to minimize this propagation effect. For example, on the (24, 12) code with majority decision decoding, alarm resetting occurs when the decoder output indicates a run of four errors without an intervening sequence of five correct information and parity bit pairs.

(24, 12) Convolutional Code

In spite of its higher error correction capability, the performance of the (24, 12) convolutional code (see figs. 5 and 6) is not much better than that of the (15, 7) block code for majority decision decoding. Figure 7 shows some of the details of the dispersion of errors made by the decoder. The decoder slides into a bit stream, which contains a potential error causing situation within a constraint length of 12 information and 12 parity bits. Figure 7(a) shows that some output error patterns exist that are 35 bits long. Comparison of figures 7(a) and 7(b) shows that the alarm indeed reduces the average length of the error patterns, and also removes the curious peak at a distance of 14 bits. Even though the alarm reduces the average number of errors per potential error causing situation, the actual number of output error packets is increased. For APP decoding the alarm is not useful because the error dispersion is much larger than that for majority decision decoding. However, this is compensated by the fact that the number of output error events per error causing situation is much smaller. Another fact worth noting is that the shape of the dispersion curves seems to be independent of the number of input errors per potential error-causing situation.

(44, 22) Convolutional Code

When comparing figures 6 and 8 one notices that for the convolutional code there is a 0.4 dB loss in performance resulting from concatenating the seventh bit parity check code, while there is only a very small loss for the (73, 45) block code. This is explained by the fact that errors occur more often in bunches in the (44, 22) convolutional code than in the (73, 45) block code. This results in a relatively high ratio of double error words to single error words, thus the seventh bit parity check code is inefficient in detecting decoding errors. As far as error dispersion is concerned, the remarks made for the (24, 12) code hold, except that the average dispersion has also increased with the increased constraint length.

Summary of Test Results for Threshold Decoding

To summarize the type of results obtained, assume the following condition. Suppose the 6-bit word error rate was not to exceed 10^{-5}; then, for no coding the E_b/N_o required is about 10.5 dB, while the undetected word error rate for the simple parity check code requires only 7.8 dB. Of course, this includes a 4.8×10^{-3} word deletion rate (due to parity tagging). If one considers this deletion rate negligible, then, for majority decoding, only the (73, 45) code shows a moderate improvement of 0.8 dB over the seventh bit parity check code. For APP decoding all codes show an improvement: 0.2 dB for the (15, 7), 1.0 dB for the (24, 12), 1.5 dB for the (44, 22), and 2.0 dB for the (73, 45) code. For all codes investigated, APP decoding shows about a 1.5 dB gain over majority decision decoding.

Figure 9 helps to visualize the error bunching that occurs after decoding. The figure is essentially a plot of double error 7-bit words vs. single error 7-bit words. The steepest curve is for the (7, 6) parity check code used alone, and corresponds to a binomial distribution of errors per word. The remaining curves show that there are many more double error words after decoding than a binomial distribution of such errors would suggest. The least error clustering occurs in the (73, 45) code, when it is full APP decoded. This is the reason why, except for the last mentioned code, the seventh bit parity check code concatenated on the

other codes generally results in a performance degradation.

Sequential Decoding

Theoretically the limit for sequential decoding is 3 dB from Shannon's limit for coding. This limit for sequential decoding can be achieved with sufficiently low rate codes of "long" constraint length. However, even at rate 1/2 and with short constraint length convolutional codes, sequential decoding can result in a coding gain compared to competitive schemes (in terms of system complexity) such as threshold decoding for the gaussian channel.

For sequential decoding, the constraint length of the code can be made long enough to make the probability of decoding a bit in error negligible. However, for decoding with the computer at the Deep Space Station, decoding speed consideration limits the code constraint length if efficient programming is to be realized. The constraint length is limited to 25 information bits since the encoder shift register replica in the decoder, which is required to hold the 24 most recent information bits, can be represented by one 24-bit register in the SDS 920 computers.

When convolutional encoding and sequential decoding are applied, other factors (in addition to code rate and constraint length) that must be considered are the block length for synchronization and the size of the synchronization sequence per block.

For a particularly "noisy" segment of data to be decoded, the search time required in sequential decoding may be quite long and hence the amount of time and buffer storage may be inadequate to decode a block of data completely. This has been termed the buffer overflow problem. In order for the sequential decoder to recover for decoding, a method of resynchronization is necessary. One way is periodically to send through the encoder a sequence of known data equal to the constraint length of the code. This essentially segments the coded data sequence into independent blocks. An alternative is to reset (between input information bits) the encoder to zeros periodically, thus cutting the code tree. In order to protect the last information bits just before the encoder is reset, a known sequence, which may be less than a constraint length, should be encoded just prior to the encoder resetting.

For the Pioneer application, the present frame size is 224 bits with a 7-bit frame synchronization word adjacent to a 7-bit mode identification word. After initial synchronization, both of these words can be used as known words to form a 14-bit sequence prior to the encoder reset. This is conveniently done every frame and was used as a reference for simulations.

A computer program was written which implemented the Fano sequential decoding algorithm.[3] The coding simulations were performed on an IBM 7094 computer; the channel was assumed to consist of a sampled and 3-bit quantized output of a matched filter for PSK signals.

The (50, 25) code was determined through a two-step process. A code optimization technique reported by Lyne[4] was used to obtain two (42, 21) equivalent codes. The code with the most symmetrical encoder connections when viewed from both ends of the encoder was chosen. (The reason for this criterion was that a reverse decoding technique has been investigated in order to reduce undetected errors.) Due to the excessive amount of computer time required to extend the code to (50, 25) by Lyne's method, the additional encoder connections were determined by sequential decoding simulations with the remaining combinations. The code selection was based on minimizing undetected errors, and the selected code computes parity according to the rule

$$p_n = i_n + i_{n-1} + i_{n-3} + i_{n-5} + i_{n-7} + i_{n-8} + i_{n-11} + i_{n-13} + i_{n-14} + i_{n-15} + i_{n-19} + i_{n-20} + i_{n-21} + i_{n-23} + i_{n-24}$$

where $n = 1, 2, \ldots ; i_{n-k} = 0$ for $n-k \leq 0$.

The data bit stream into the decoder was assumed to originate from two sources (1) a 6-bit word structure and (2) a 6-bit word plus parity check bit.

For data with the (7, 6) code, the Fano sequential decoding algorithm was used to force the parity check bit to be correct, rather than for error detection as done with the other codes simulated. Of course, known words and parity check bits of the (7, 6) code were regarded as rate losses so that E_b/N_o represents only energy per information bit. Results for the basic (50, 25) code with 224-bit blocks, of which 14 bits are known at the end of each block, are shown in figures 5 and 10. The criterion for error detection was to delete the entire frame if the number of node trials in decoding exceeded a preassigned number, which was 12,000 except for the one case noted in figure 10. It was found that 14 bits were not enough known data to make the error probability at the end of the block as low as for the other portion of the block. At $E_b/N_o = 3.5$ dB, about 30 percent of the errors occurred in the two to three words before the known data. For a 6-bit word error probability of 10^{-5}, the basic (50, 25) code concatenated with the (7, 6) code gives a 6.6 dB gain improvement over no coding and a 3.9 dB improvement over the simple parity check code. Also concatenation with the (7, 6) code shows about 0.3 dB advantage over the basic (50, 25) code.

In lieu of increasing the constraint length for improving the performance of the rate 1/2 code, a detection technique, which could be used upon option, was investigated, namely, reverse decoding. Any block of data received can be viewed in the reverse direction as being generated by an encoder with the parity taps turned end for end. This permits the associated information parity bits to be

separated by a total of 48 bits. This provides an interweaving effect which is particularly effective in the real channel where there is an intersymbol influence over several bits.

Reverse decoding was performed on every frame of data that was successfully decoded in the forward direction. The simple detection strategy used was to delete any data words which disagreed after decoding in both directions and to delete all frames which failed on node trials in the reverse direction.

To begin decoding a block of data, the encoder duplicate at the decoder must be "initialized." For decoding in the reverse direction, this consists of loading the last 24 decoded bits, of which only 14 are known. Thus, after completion of decoding in the forward direction, any errors in the last 10 data bits will give incorrect data for decoder initialization. Simulations have shown that with any such errors there was no apparent decoding difficulty and these errors were not detected. Thus, if any significant improvement is to be obtained from the reverse decoding, it must be accompanied with at least 21 known bits per block. With this assumption and the use of reverse decoding, the undetected error rate is shown in figure 10 for E_b/N_o = 2.7 and 3.1 dB. For E_b/N_o = 3.5 dB all errors in the simulation runs were detected. It should also be noted that for data without the (7, 6) code and for $E_b/N_o \geq 2.7$ dB, the reverse decoding permitted detection of all errors committed by decoding in the forward direction in all of the data runs encountered in these simulations.

Conclusions

Of the 4 codes that were APP decoded the gain was greatest with the (73, 45) block code; namely, 4.7 dB over no coding or 2.0 dB over the simple parity check code. Due to the tendency for error clustering from the decoder, independent use of the parity check code is of no value in reducing undetected errors, and in most cases does not make up for the 0.67 dB rate loss. For the block codes full APP shows a small improvement over hard decision feedback APP. The (50, 25) convolutional code with sequential decoding provides a 1.9 dB gain over the (73, 45) code. If reverse decoding is used in conjunction with a 21-bit known sequence per block compared to a 14-bit sequence, a further 0.5 dB improvement is achieved. This places the performance within 1.3 dB of the limit for sequential decoding. Estimates of the computational load on a general purpose computer used to decode the two codes at E_b/N_o = 4.0 dB or more are quite comparable. In addition, the encoder for the (50, 25) code is no more complex than the encoder for the (73, 45) block code.

Acknowledgement

The author is indebted to Larry B. Hofman for programming the sequential decoding algorithm and to Frank Neuman for his assistance with the threshold decoding simulations.

References

1. Massey, James L., Threshold Decoding. MIT Press, Cambridge, Mass., 1963.

2. Mitchell, Michael E., Performance of Error-Correcting Codes. IRE Trans. on Communications Systems, vol. 10, pp. 72-85, 1962.

3. Wozencraft, John M., and Jacobs, Irwin M., Principles of Communication Engineering. John Wiley and Sons, Inc., N. Y., 1965.

4. Lyne, William H., IV, Some Single Generator Convolutional Codes and a Sequential Decoding Simulation, Rice University Ph.D. Thesis. 1965.

Figure Titles

Figure 1.- Bit error rates of (15, 7) and (73, 45) cyclic block codes.

Figure 2.- 6-bit word error rates of (15, 7) cyclic block code.

Figure 3.- 6-bit word error rate of (73, 45) cyclic block code.

Figure 4.- Normalized 7-bit word error probability as function of the decoding sequence, (73, 45) code.

Figure 5.- Bit error rates for (24, 12), (44, 22), and (50, 25).

Figure 6.- 6-bit word error rate of (24, 12) code.

Figure 7.- Error dispersion for the (24, 12) convolutional code.

(a) (24, 12) code, majority decision decoded without alarm.

(b) (24, 12) code, majority decision decoded with alarm.

Figure 8.- 6-bit word error rate of (44, 22) code.

Figure 9.- Detected versus undetected word errors for several codes and decoding methods.

Figure 10.- 6-bit word error rate for sequential decoding and other selected codes.

AAA382-1

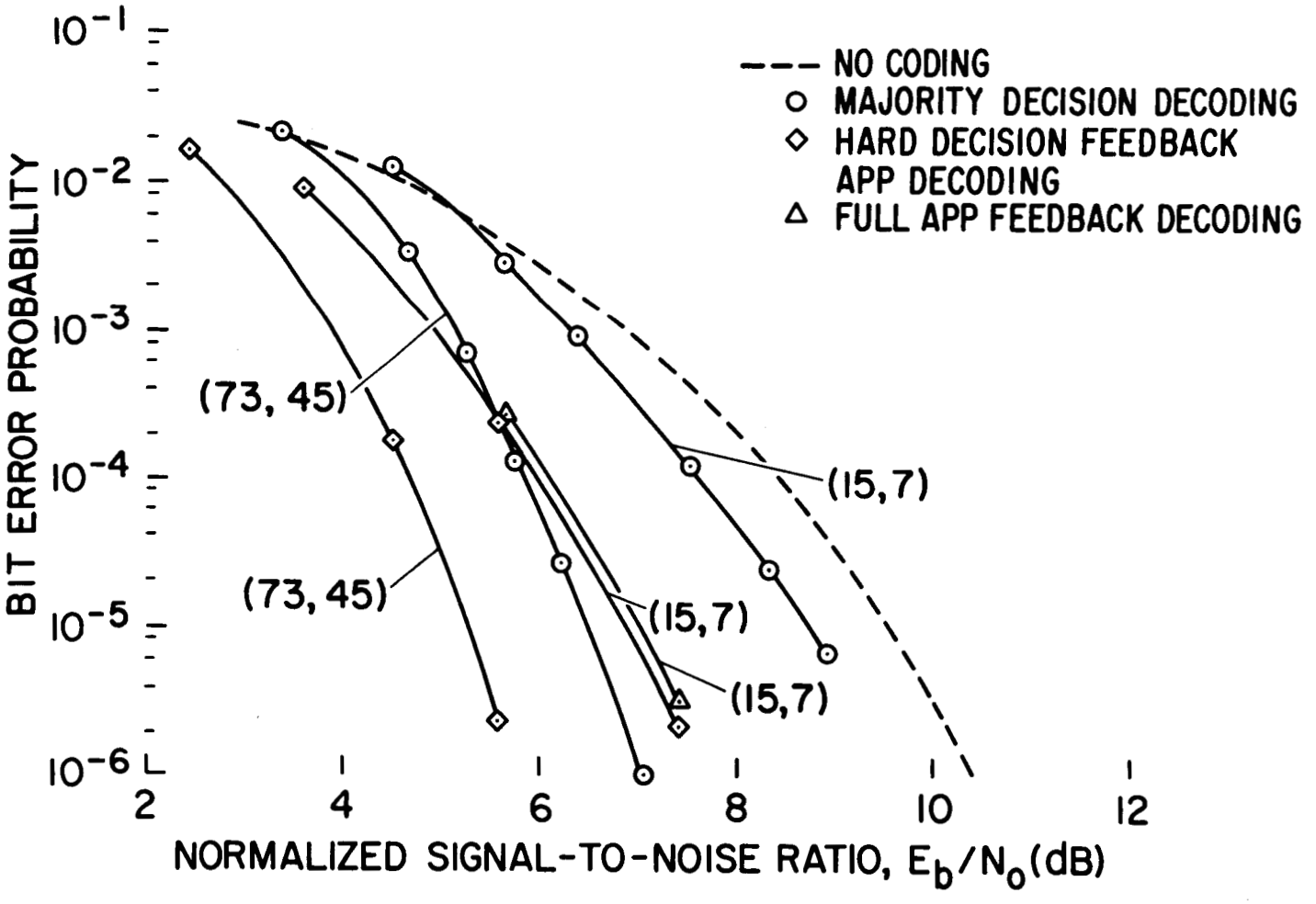


Figure 1

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

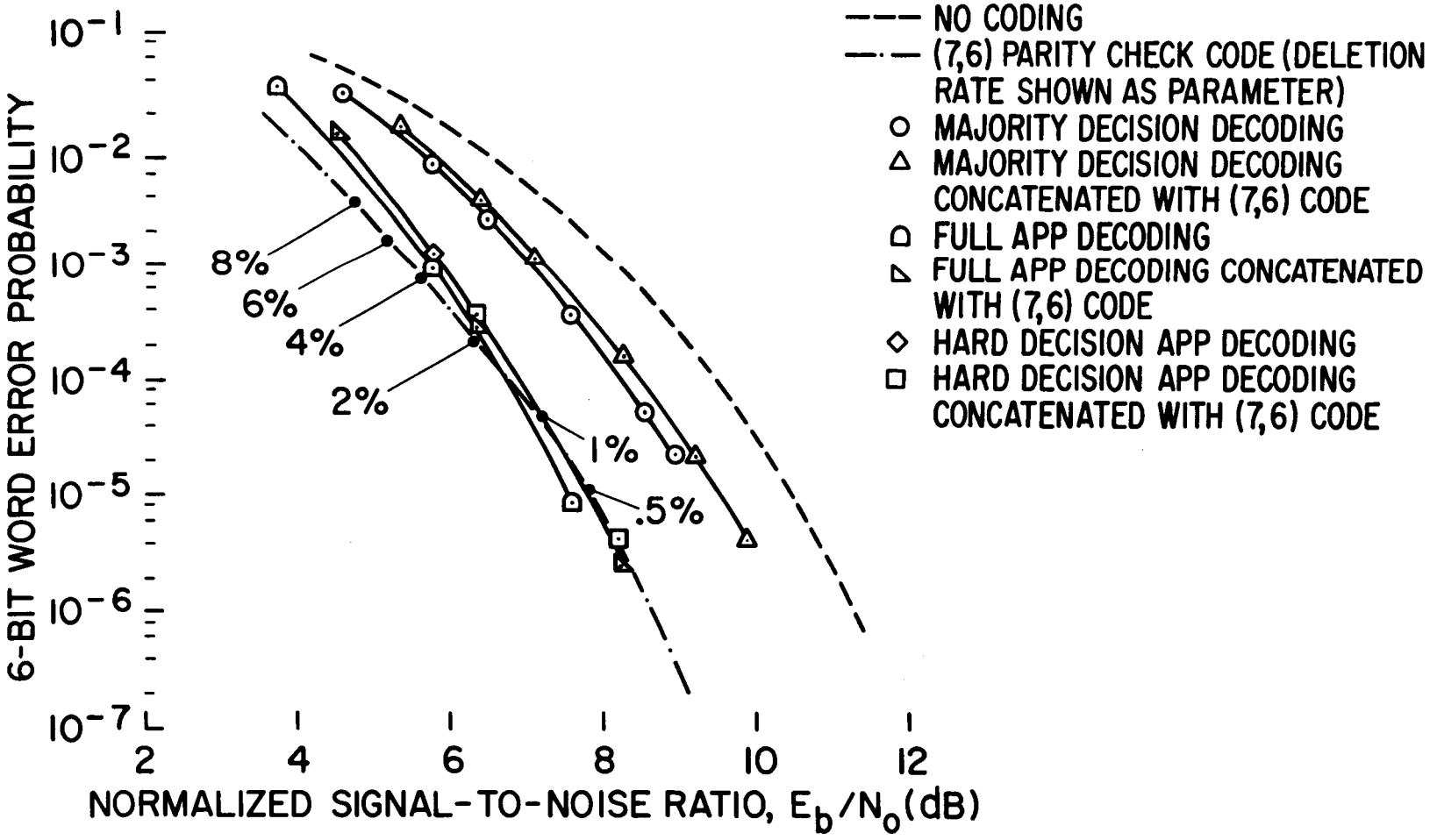


Figure 2

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

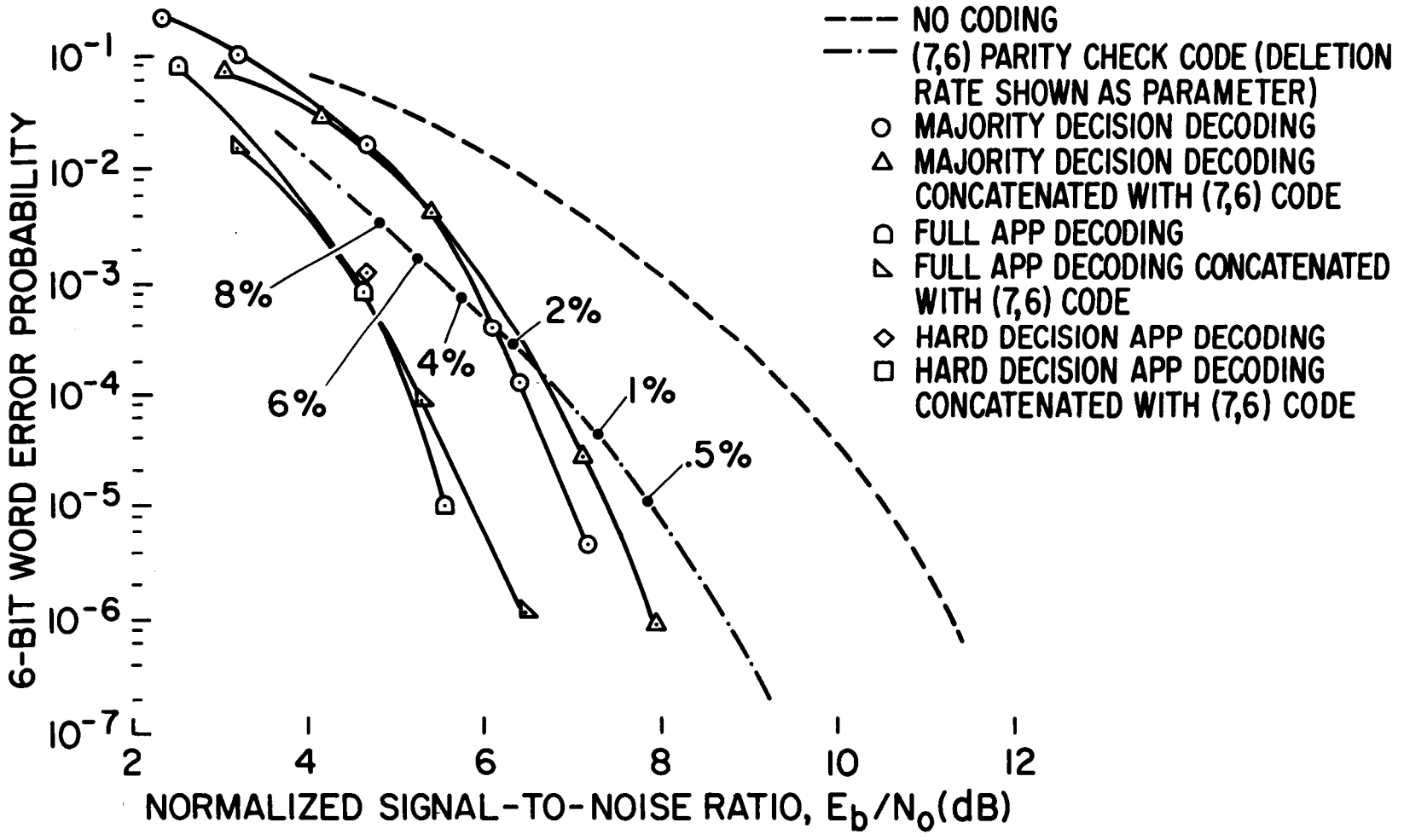


Figure 3

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

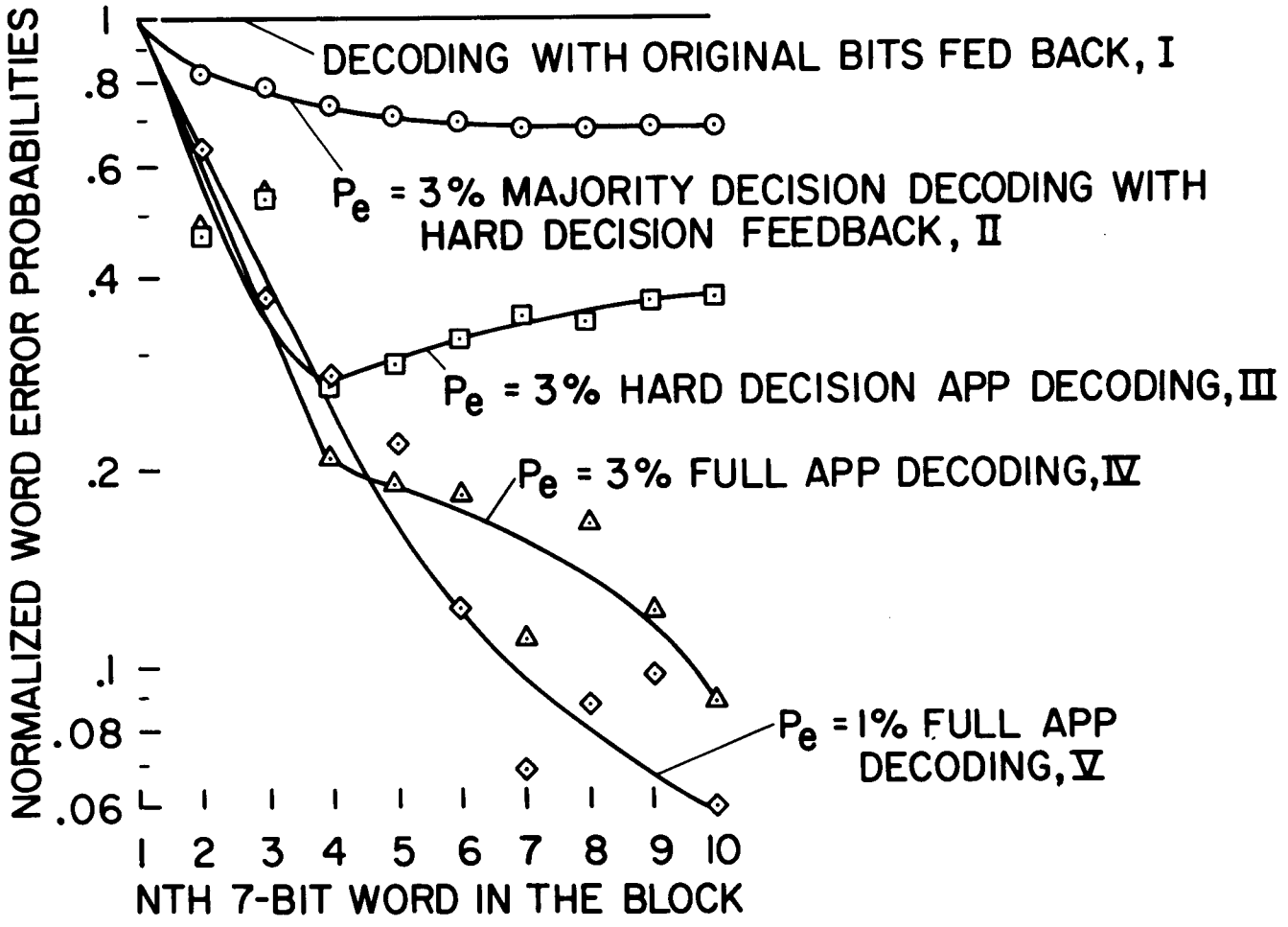


Figure 4

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

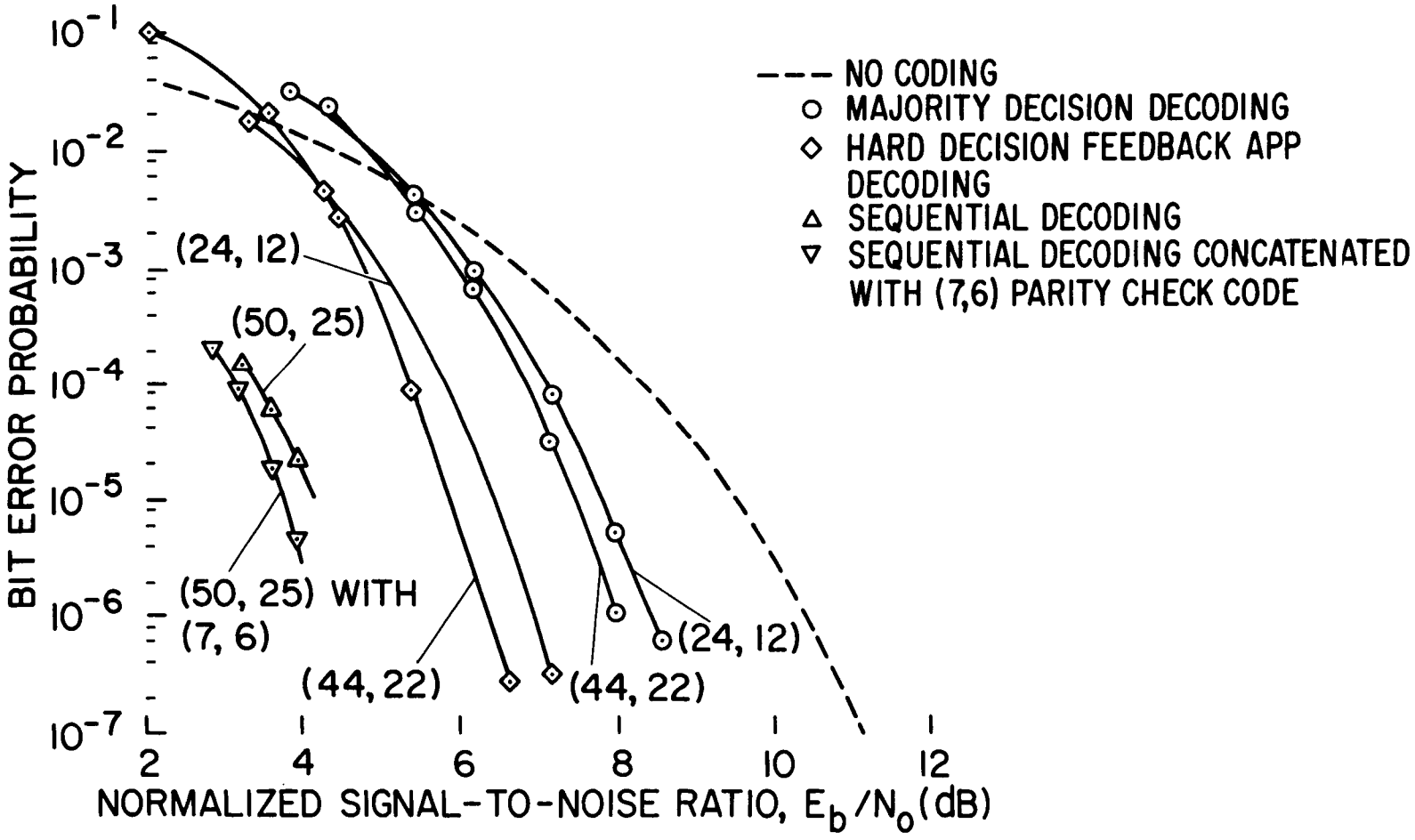


Figure 5

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

AAA382-6

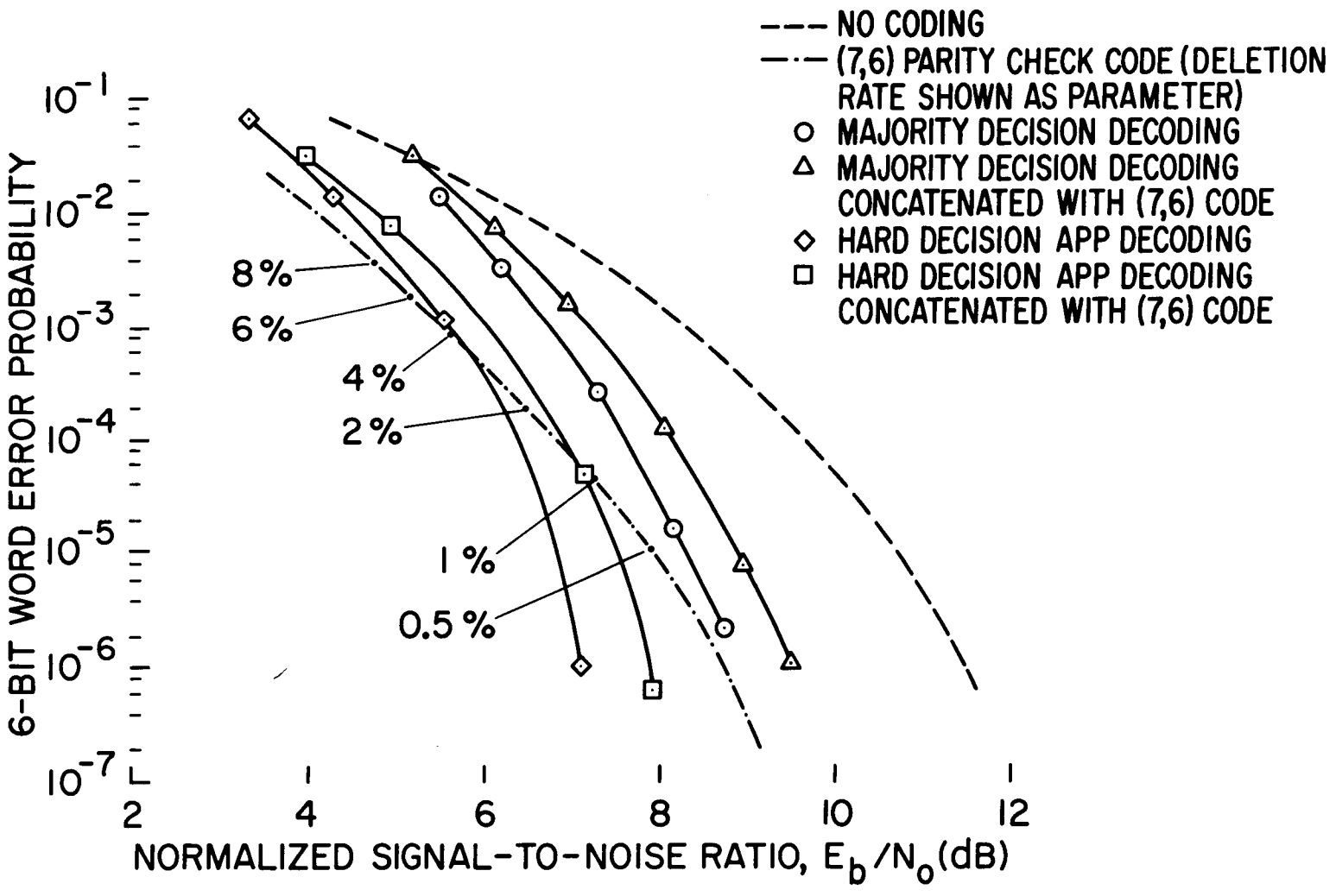
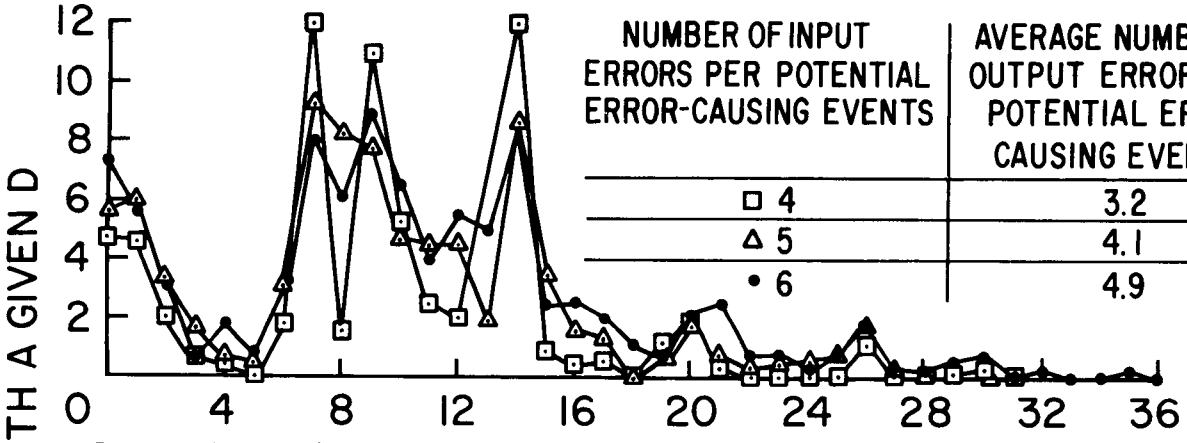


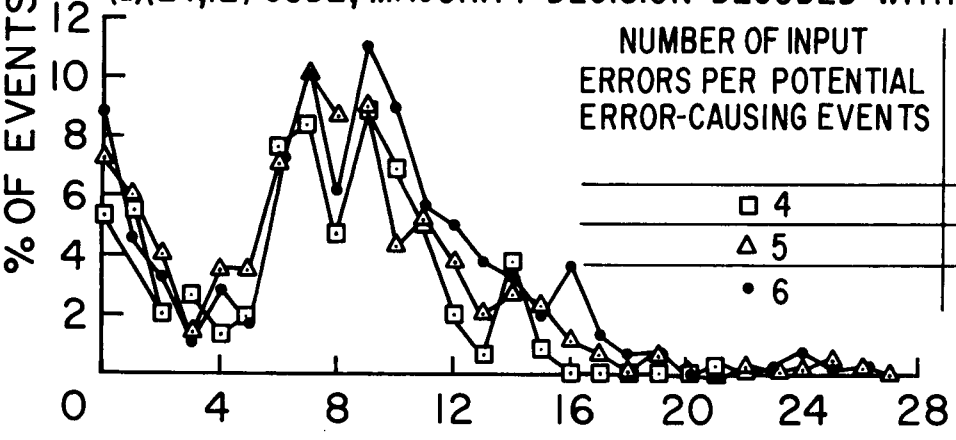
Figure 6

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA



NUMBER OF INPUT ERRORS PER POTENTIAL ERROR-CAUSING EVENTS	AVERAGE NUMBER OF OUTPUT ERRORS PER POTENTIAL ERROR-CAUSING EVENTS	% OF DECODED POTENTIAL ERROR-CAUSING EVENTS WITH NO ERRORS OUT
□ 4	3.2	33.3
△ 5	4.1	16.7
• 6	4.9	5.5

(a)(24,12) CODE, MAJORITY DECISION DECODED WITHOUT ALARM



NUMBER OF INPUT ERRORS PER POTENTIAL ERROR-CAUSING EVENTS	AVERAGE NUMBER OF OUTPUT ERRORS PER POTENTIAL ERROR-CAUSING EVENTS	% OF DECODED POTENTIAL ERROR-CAUSING EVENTS WITH NO ERRORS OUT
□ 4	2.5	33.3
△ 5	3.3	14.7
• 6	4.2	3.1

(b)(24,12) CODE, MAJORITY DECISION DECODED WITH ALARM

Figure 7

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

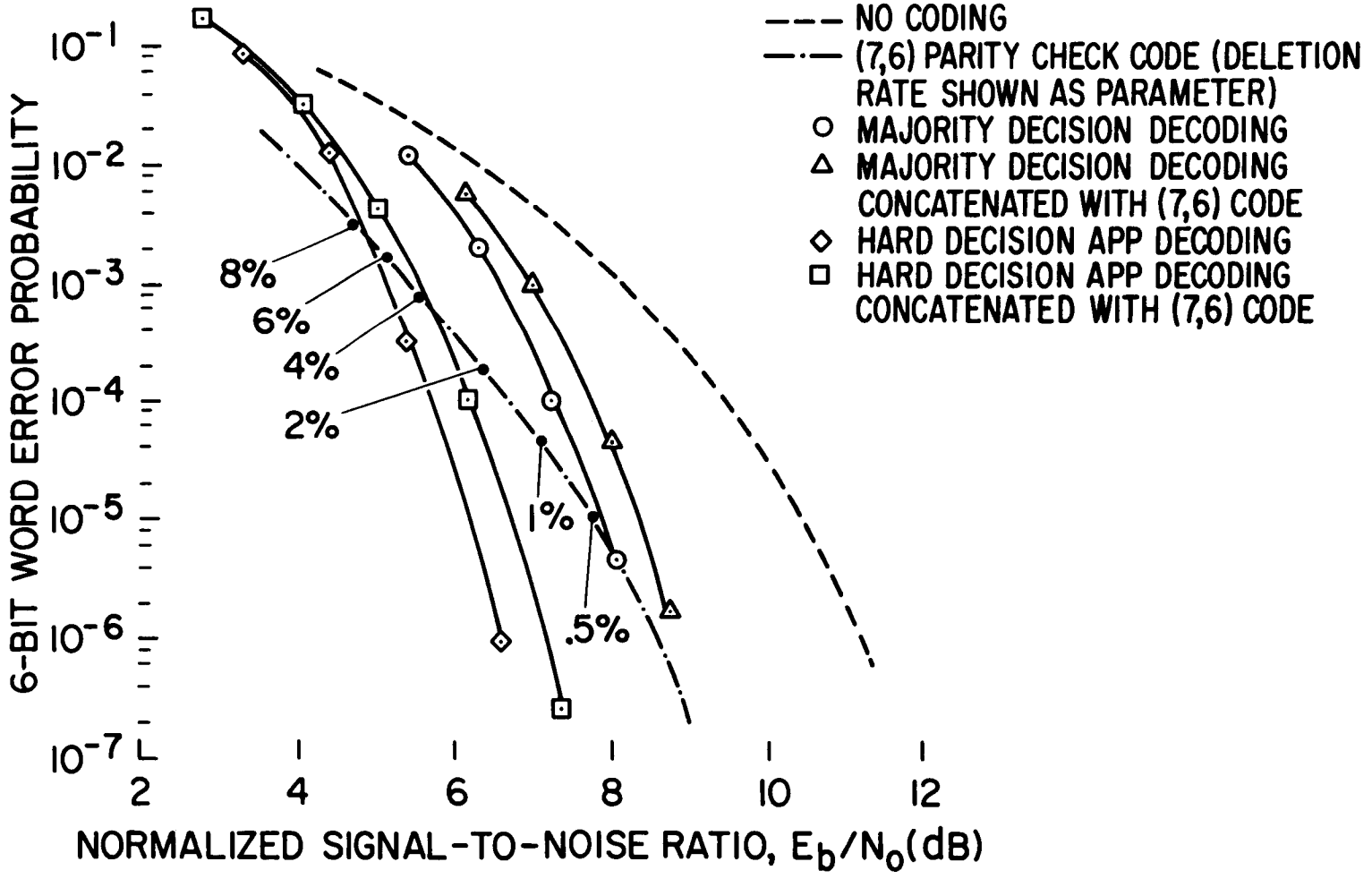


Figure 8

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

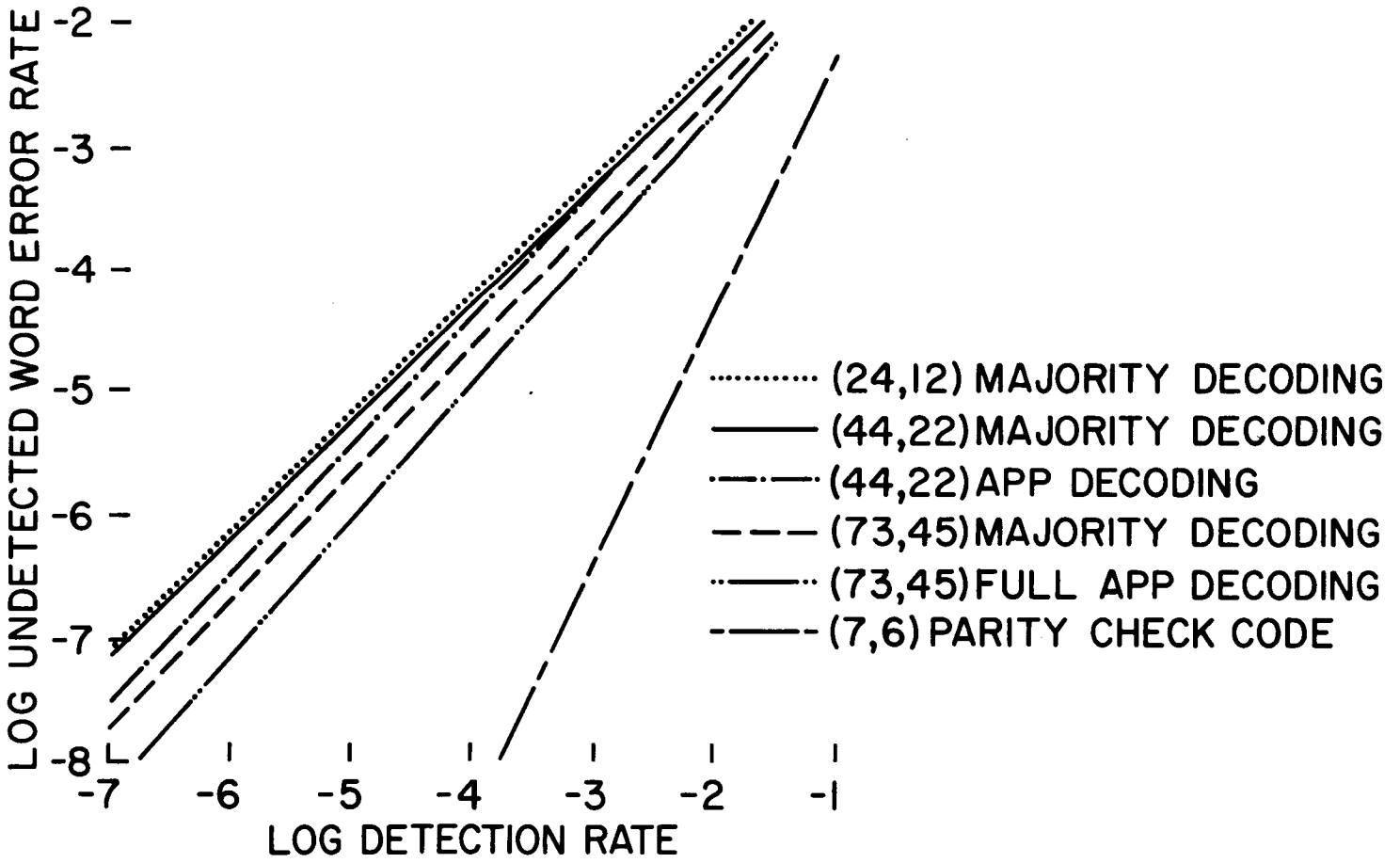


Figure 9

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

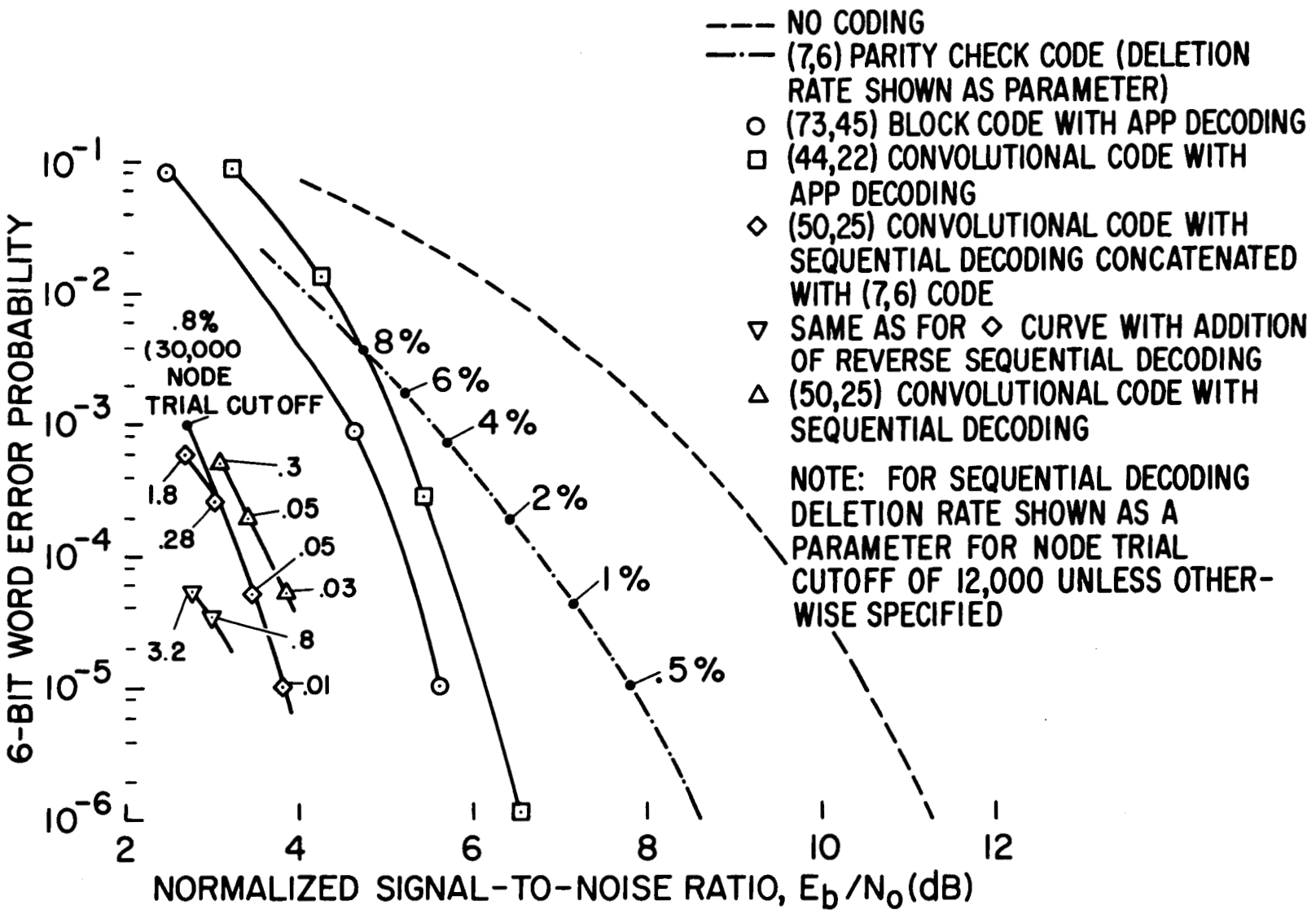


Figure 10

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA