CODING -AN ENGINEERING TOOL FOR THE DIGITAL TELEMETRY LINK

RONALD M. MULLER
Goddard Space Flight Center
NASA
Greenbelt, Maryland

Summary Various digital coding techniques are discussed from an engineer's and user's point of view. Each technique is displayed in a uniform way which measures performance against the best possible (Shannon) channel. Encoding-decoding complexity and other "system" merits and drawbacks of each technique are discussed. The reader is introduced to coding by drawing analogies with an everyday and familiar coded communication channel - spoken English. Bit, word and block error detection and correction techniques are then presented. Lastly, a concatenated block code scheme which combines these techniques is developed.

Introduction Many papers have dealt with the subject of coded telemetry links and such links have been used on occasion to improve the communications channel by some number of decibels. These few missions have tended to be confined to deep space missions where every decibel counts toward an increased range of operations at a given bit rate. The purpose of this paper is to present the telemetry engineer with a concise performance summary picture of a few representative coding techniques. Any additional information needed to apply a particular code to a specific mission may be found in the references.

Spoken English Coding is a channel modulation that facilitates the transmission of information through the channel in spite of noise of various kinds. A communication channel in use everyday is the spoken word. In the U.S.A. the code is usually English. English (and most languages) employ very complex coding structures which are well adapted to the noisy communication channel we normally encounter. The coding is so complex that all that saves the link is the fact that the data rate is low and an extremely complex and sophisticated computer is present at the receiving end to decipher the message. English is not one of the coding schemes being proposed to put into a telemetry link (as this term is usually used), but some reference to features of the language will be made to illustrate a point.

Pulse Code Modulation (PCM) Even though PCM's middle name is "code", most papers on coding (including this one) refer to PCM as "uncoded". The justification for this is common usage and the mathematician's tendency to down-play the trivial example. PCM is often the coding of choice if the choices are between a digital modulation and the analog techniques such as Pulse Amplitude Modulation or Pulse Duration Modulation. Assuming that a PCM or "binary symmetric" channel has been chosen for a given telemetry application, the question is "Should redundancy bits be added in some way to the information bits in order to insure a more reliable transmission to the receiving site?".

Uncoded PCM is analogous to single character Arabic numerals. If the same number is spelled out, you have unconsciously added redundancy and a form of coding. There is a kind of "bandwidth expansion" when spelling out a number since to keep up the information rate, faster writing is required. In most telemetry applications, the desired information rate is a fixed parameter, and any coding redundancy will increase the bandwidth requirements to some extent. All the curves of this paper have included this bandwidth expansion so that the information rate is a constant.

Figure 1 is the bit error rate versus signal-to-noise ratio curve that starts most papers on coding. This curve is basic to all binary telemetry systems, but it has drawbacks when used to compare various coded systems. For instance, it would be desirable to be able to relate a code's performance against a perfectly coded channel. It is also desirable to show the effect of data deletions on the channel throughput. For this purpose, a somewhat different chart is shown in Figure 2. The heavy diagonal line gives the throughput of a perfectly coded communication channel; at all signal-to-noise ratios (SNR's) better than -1.6 db perfect data are detected. At all SNR's worse than -1.6 db, no data are detected. (SNR is defined as signal energy per bit of information per noise power per cycle.) If this perfect channel were operated at a SNR of +8.4 db, it would have no errors, but it is being operated at only 10% of its capacity. Now if an uncoded PCM channel is used at a SNR of 8.4 db, its throughput would again be 10%, but as shown by the negative exponents of 10, it would yield one error every 10,000 bits, or less than perfect performance. The error rates plotted are taken directly from the error rate curve of Figure 1. Figure 2 will be used as a basic chart and the other coding schemes will be added to it so that they may be easily related to both a perfect channel and uncoded PCM. Comparisons will be made at the 10^{-4} bit error rate since for many applications, this is an acceptable error rate. Comparisons at other error rates may be taken from the curves.

Single Bit Parity Probably the simplest coding scheme is to add a single parity bit every so many information bits (usually every word). This bit is used to check for a single bit error in each group of bits. It will also catch triple and any larger odd number of errors. The power of this technique resides in the fact that most errors are single bit errors and so it is able to identify most of the words that are wrong. These words may be given a “low confidence” flag or even discarded from further consideration. When this is done, the “high confidence” bit stream that is left is purged of the most common error, namely, single bit errors; it is left primarily with double errors and infrequently other even number of bit errors. Most communication people pale at the thought of discarding any data, but for the bulk of the telemetry applications, an occasional missing sample is better than including erroneous samples as is the case with uncoded PCM. Figure 3 presents the results of using one parity bit for each eight information bits and discarding data that contains the detectable odd errors. The bit error rates for parity are shown below the line and the effect of throwing away data is shown by the divergence between this curve and the theoretical line. This divergence is barely perceptible until rather poor SNR’s are reached. If an uncoded PCM communications link were operated at +8.4 db SNR and then the same link used one parity bit every eight information bits, the bit error rate in the retained data would go from one in 10,000 bits to one error in 2.5 million bits. Fully 99.8% of the data would be retained and only 0.2% discarded.

Another way to look at the curve is to observe that when using parity, one does not get a bit error rate of 10^{-4} until an SNR of 6.1 db is reached. (At that point the discard rate is three percent.) Because of the throw away rate, this is equivalent to the throughput of a Shannon channel operating at +6.2 db or 16.5 percent of capacity. This 2.2 db difference between parity and uncoded PCM represents a 65% increase in link throughput. In other words, in exchange for discarding three percent of the data transmitted, one will obtain 65% more data in a given time and maintain equal quality in the accepted data!

When compared to uncoded PCM on the basis of bit error, coded techniques do not look as good as when they are compared on a word error basis. Figure 4 illustrates this. For this curve only, the error rate numbers are word error rates based on eight bit words. For uncoded data, the 8.4 db point has a word error rate of 8 in 10,000 and for parity the 6.1 db point has only 4.5 in 10,000. This additional improvement results from the fact that when parity misses an erroneous word, there are always two bit errors in that word (or four or six, etc.) whereas most of the errors in the PCM case are single errors per word. In other words, it takes approximately twice as many bit errors to get the same word error rate when parity is used.

Parity Coding is a simple way to boost data quality and/or quantity that is very easy to implement at both ends of the communication channel. It requires typically less than 12% bandwidth increase for the same information rate. This is much less than the more powerful coding techniques to be discussed below. Another advantage is that the parity error rate gives a direct measure of link quality.

Biorthogonal Coding Both the uncoded PCM and the parity PCM detect the data bit-by-bit. Now consider a system where detection is performed by correlation on a word-by-word basis. This is accomplished by so called “matched filters” at the receiving end. Each filter is set to look for its word and one filter is needed for each word that might be sent. The outputs from the filters are compared to each other and the binary word associated with the filter having the largest output is assumed to have been the true message.

Going back to the spoken language coding for an analogy, matched filters may be developed for many words. (e.g., ones first name, last name, or the name of a company and many, many more.) Thus, at a noisy cocktail party, one’s name may be detected from across the room, even though everything else of that conversation will be gibberish.

Biorthogonal systems, when detected a word at a time, gives improvement because the noise power is integrated over the whole word time, and its effective bandwidth is reduced by a factor equal to the number of bits per word. For an eight bit word, this is a factor of eight (nine db). Unfortunately, with an eight bit biorthogonal system, there are 255 filters that can give an error and only one that is correct. The ‘net result is an approximate 3.9 db channel capacity improvement on a bit error rate basis (10^{-4} error rate). This performance is shown in Figure 5.

A basic disadvantage to this technique is the complexity at the receiving end. Eight bit word systems are about as high a complexity as most designers care to develop and most are satisfied to stay at a more modest six bit word size or less. Figure 6 compares a six bit biorthogonal performance and shows an approximate 3.2 db channel capacity improvement on a bit error basis (10^{-4} error rate).

Picking a code set which has biorthogonal properties is a simple matter. However, one would like a code set that also gives good word synchronization since detection is word-by-word and the system must be in word synchronization to operate. This is analogous to picking a good frame synchronization word in uncoded PCM. As in frame synchronization words, the simplest codes to generate are not the best, but fortunately good codes are plentiful.

As in uncoded PCM, no data are discarded but neither do you have as simple a measure of channel performance. Channel performance can be estimated, however. One technique is to do both a biorthogonal and a bit-by-bit detection of the data. Since the biorthogonal always makes orders of magnitude fewer errors than the bit-by-bit, it can be used as a very good approximation of the original message. This can be re-encoded into the same biorthogonal code bit stream and compared to the bit-by-bit detection data. Note that for six bit biorthogonal words, 32 code bits are sent so that the bit error rate of these code bits is much worse than even a normal uncoded PCM transmission of the same data.

Longer words perform better than short words, but the complexity of the receiving equipment rises exponentially with word length, and the bandwidth rises nearly as rapidly. Figure 7 relates biorthogonal word length to throughput at a constant bit error rate of 10^{-4}. Note that uncoded PCM is the trivial case of both one and two bit word biorthogonal codes.

Convolutional Coded PCM This is a technique which permits the use of very long words without paying the penalty of exponentially increasing the receiving complexity or the bandwidth. The scheme has recently become quite attractive because of new fast decoding algorithms which are quite nicely suited to rapid digital computer solution.

An analogous coding structure is used in English. In writing a paper, one may build it up by adding more words to what has already been written. Just as a paper is written starting at the beginning, one normally reads it from that point - one sequentially decodes it. If an error occurs in the text, one guesses at the correct word and reads over the bad spot to see if it makes sense. Some simple misspellings are so obvious that one may not even be conscious of correcting them. Other errors may be so bad that no matter how long one spends guessing, the meaning of the text cannot be deciphered. Textual redundancy both in the word spellings themselves and the context of the other words in combination allows one to decode most messages.

Much the same thing is done by the sequential decoding algorithm. At the transmitting end, redundancy is added to the message in the form of parity bits. Typically one parity bit per information bit is used, but other higher and lower ratios have been demonstrated. A message is sent in frames each of which starts with a known bit pattern. This is detected with the usual frame synchronizer-correlator. Starting at that point, the original message is then reconstructed by trying only the most probable bit combinations in sequence until the whole message makes the most “sense”. Since all possible message combinations are not tried, the decoding costs are much lower than for long biorthogonal codes.

The performance of the Convolutional Encoding - Sequential Decoding scheme is shown in Figure 8. This is shown for a constraint length of 32 bits where constraint length is defined to be the length of the word which is included in each parity calculation. At very poor SNR’s the performance drops off because more and more computer tries are required. If computer time is limited to a factor of two more than that required to go through the process at good SNR, the lowest curve is obtained. If the limit is four, the second lowest curve is the result. If 10 times is available the result is the second highest curve and with an infinite time, the topmost curve is obtained. Note that the data thrown away for unprocessable frames are whole frames. Most decoding systems are set to work for a fixed time on a given frame and if it does not successfully decode in that time, the intermediate results are put on another tape and work starts on the next frame. It returns to the tape during spare computer time and tries for a longer time. If decoding is still not successful, it gives up on that frame.

The time taken to decode a frame is a good measure of the instantaneous liak quality. The long effective word length gives very good performance at low SNR and both the bandwidth and complexity factors are relatively low. Note that there is a peak performance point and that poorer SNR's give rapidly diminishing throughput. Thus one would be better off to slow down the data rate and stay to the right of the peak since less framesutil be discarded and less computer time will be required to achieve the same throughput.

Concatenated Codes - Biorthogonal Plus Parity This code is constructed as a combination of the Parity and Biorthogonal systems and represents another type of block code. To generate the code, one separates data into blocks of some number of words. Another word whose bits are a parity check of the block of words is then generated and inserted in the data stream. The first parity bit is parity for the first bit of each word of the block; the second parity bit is parity for the second bit of each word, etc. (i.e., column parity for each bit position in the word of the block). Each data and parity word is then encoded into biorthogonal words and transmitted. At the receiving end, one reverses the process and decodes the biorthogonal words in matched filters as before and then performs the parity checks. A single parity word will detect all single word errors and most multiple word errors. Figure 9 shows performance for this concatenated scheme using six bit biorthogonal words and blocks of fourteen words (including the parity word). Performance of this technique is comparable to the convolutional scheme in both residual error rate and throw away rate. (Whole blocks are discarded when an error is detected.) It does take wider bandwidth than convolutional coding, but it takes a fixed amount of time to process. This process can be put entirely into hardware if high speed operation is a consideration.

A further variation on the concatenated concept is to put in additional parity words so that you not only detect more of the errors, but also correct most of them. Because of mistakes in correction, this results in poorer error rates, but a lower discard rate. Figures 10 and 11 show the effect of two word parity (column and diagonal parity) added to a block of 12 data words, each again six bits in length. Figure 10 shows that if one chooses not to try to correct the received data, but to merely detect errors, the dual parity does extremely well giving bit error rates better than 10 8 at peak throughput. Figure 11 shows the effect of correcting words; the discard rate is lower, but the error rate goes up. These results show performance comparable to the convolutional technique, but does require slightly more complex equipment on both ends of the communication link.

Conclusion It is shown that various coding techniques may be applied to PCM data to improve intelligibility of received data. At a bit error rate of no greater than one bit in 10,000, improvements over uncoded PCM and discard rate are tabulated below:

	Improvement Over PCM (db)	Data Discard %
Single bit parity - 8 bit words	2.2	3
Biorthogonal 8 bit words	3.9	0
Biorthogonal 6 bit words	3.2	0
Convolutional codes of constraint length 32 bits and bandwidth expansion of 2	5.2	5
Concatenated - 6 bit biorthogonal with one parity word (13 data and 1 parity word)	4.4	14
Concatenated - 6 bit biorthogonal with one word error correcting (12 data and 2 parity words)	4.4	0.4

There is no doubt that further advances are yet to be conceived, but since the best performance shown above is within 4.8 db of Shannon's limit further advances should be directed toward techniques that are easier to use and take less time and equipment to decode.

Another direction that will give future systems a greater information throughput is various data pre-processing steps that reduces the data volume and enriches the information content. Coding should then be used primarily to improve data quality since with pre-processing one generally removes redundancies and therefore each bit is proportionally more valuable. In other words, if you reduce the data rate by a factor of 10 or 10 db, use some of that savings in bandwidth to get improved data quality. One way is to use some form of link coding.

References

1. X-562-67-123, "The Use of a Parity Bit in a Telemetry System"; W. Miller, J. Yagelowich; March 1967.

2. Viterbi, A.J.; "On Coded Phase-Coherent Communication"; IRE Transactions on Space Electronics and Telemetry; March 1961.

3. X-562-68-65, “A Simulation of Convolutional Encoding with Sequential Decoding with Applications to IMP-I and AMP-H,J Data Processing; W. Barnes; March 1968.

4. NASA TN-D-4105; “An Efficient Coding System for Deep Space Probes with Specific Application to Pioneer Missions”; D. Lumb and L. Hofman; Aug. 1967.

5. X-562-68-340; “Concatenated Coding for Space Communications”; W. Miller, R. Muller, J. Yagelowichl- August 1968.

6. NASA TR R-292; “Self-Synchronizing Biorthogonal Coded PCM Telemetry System”; W. Miller, et.al.; October 1968.

7. Shannon, C.E.; “The Mathematical Theory of Communication”, Bell Systems Technical Journal, Vol. 27; July-October 1948.

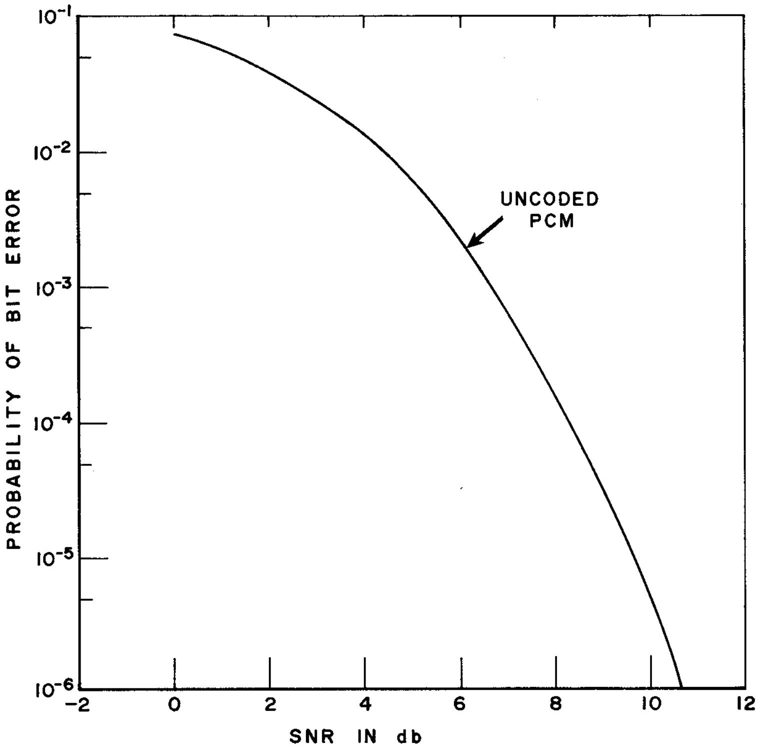

Figure 1 - Uncoded PCM Probability of Bit Error Versus SNR

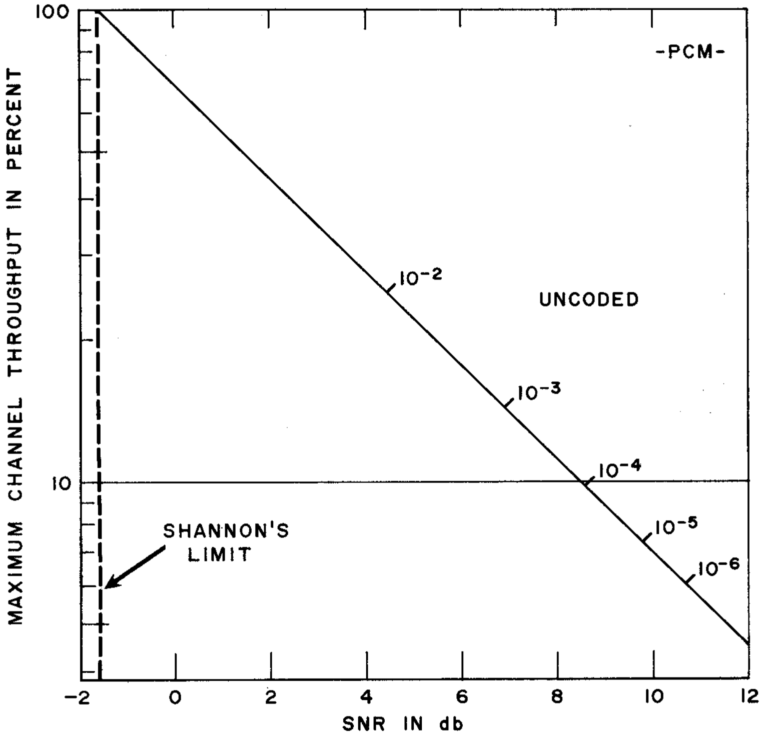

Figure 2 - Uncoded PCM

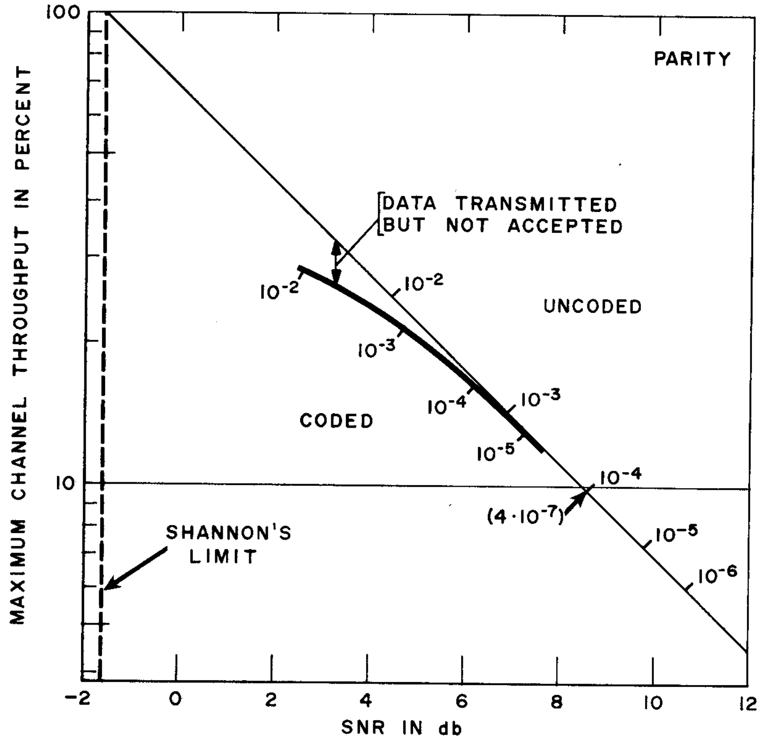

Figure 3 - Parity -- Eight Information Bits Plus One Parity Bit

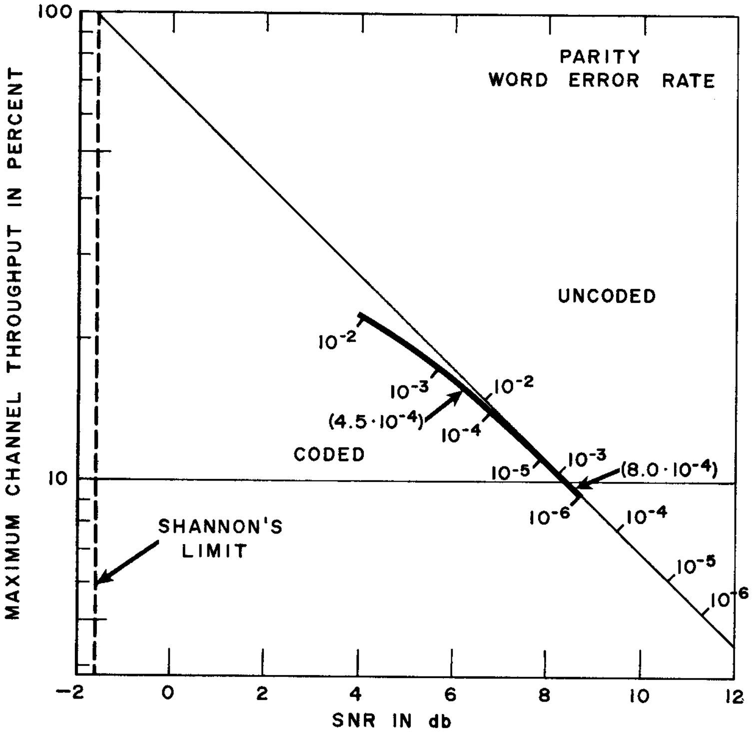

Figure 4 - Parity -- Eight Information Bits Plus One Parity Bit
Probability of Word Error

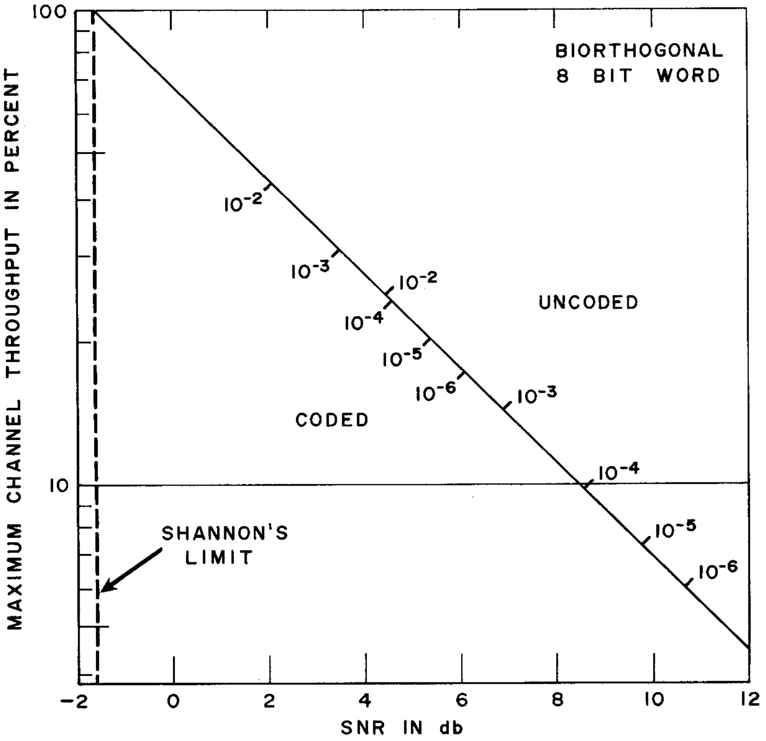

Figure 5 - Biorthogonal -- Eight Bits Per Word

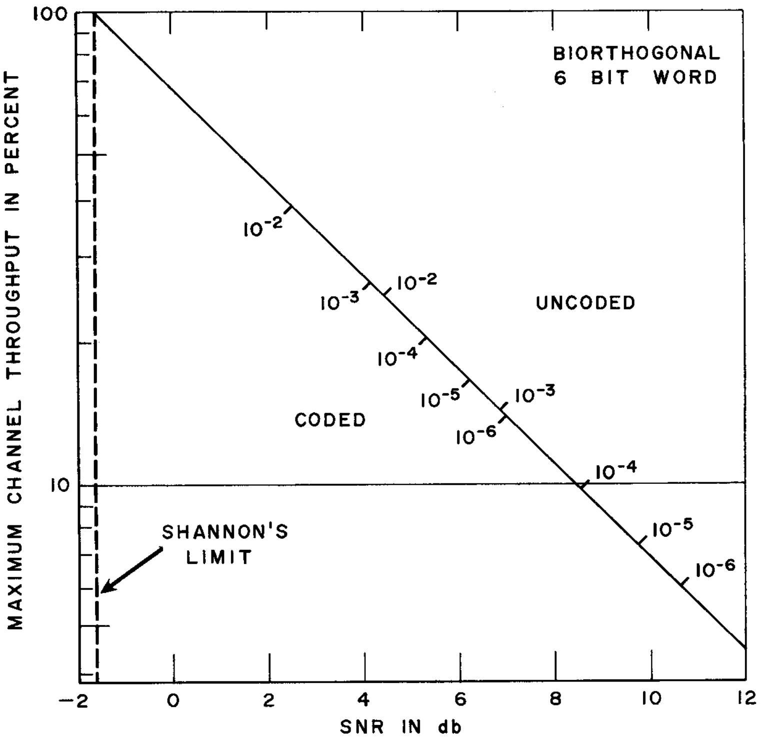

Figure 6 - Biorthogonal -- Six Bits Per Word

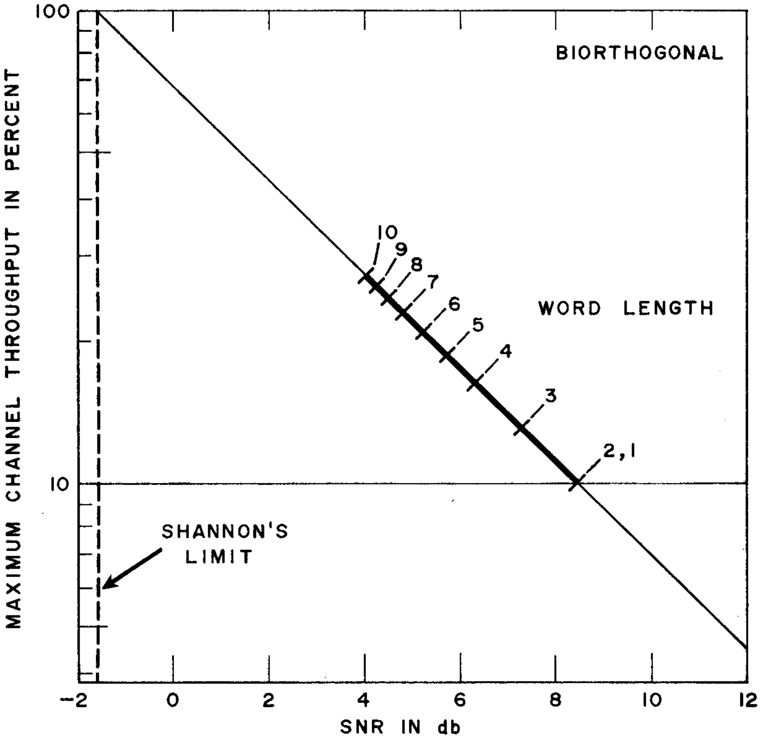

Figure 7 - Biorthogonal Performance for Different Length Words 10^{-4} Bit Error Rate

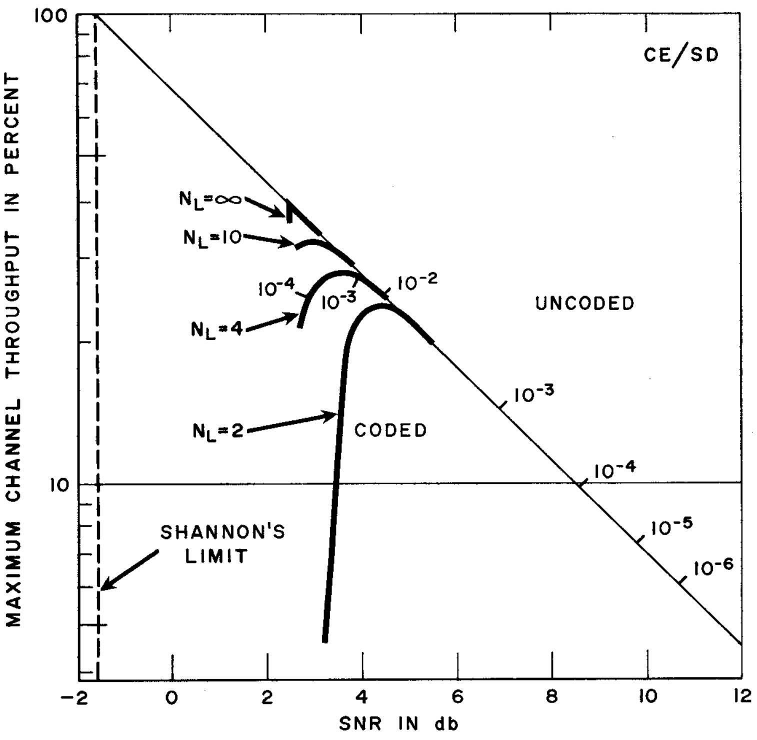

Figure 8 - Convolutional Encoding/Sequential Decoding - One Parity Bit Per Information Bit - 32 Bit Constraint Length

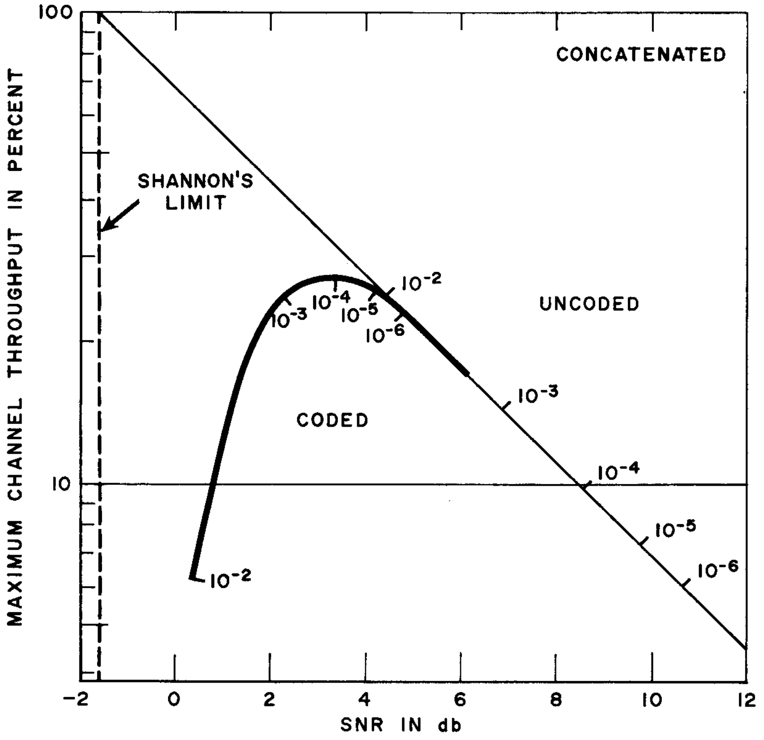

Figure 9 - Concatenated - Six Bit Biorthogonal Words, 14 Words Per Block Including One Parity Word - Error Detect Only

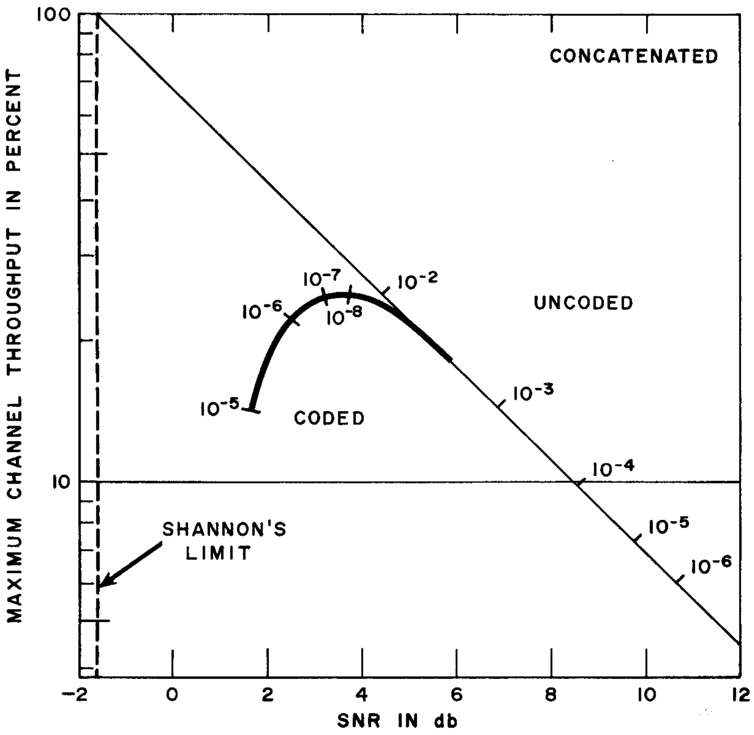

Figure 10 - Concatenated - Six Bit Biorthogonal Words, 14 Words Per Block Including Two Parity Words - Error Detect Only

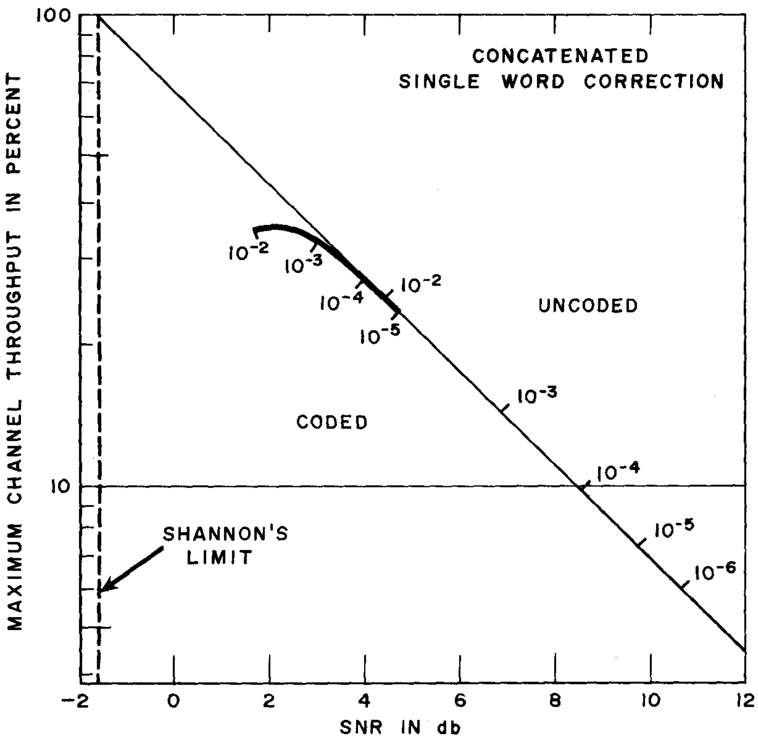

Figure 11 - Concatenated - Six Bit Biorthogonal Words, 14 Words Per Block Including Two Parity Words - Single Word Error Correction