University of
Waterloo Research Institute

Code Division for Spread Spectrum Multiple Access

Final Report
Prepared for

The Department of Communications under
DSS Contract No. OSU81-00078

by

Ian F. Blake and Jon W. Mark
Department of Electrical Engineering
University of Waterloo

CRC

Code Division for Spread Spectrum Multiple Access

Final Report

Prepared for

The Department of Communications under

DSS Contract No. OSU81-00078

by

Ian F. Blake and Jon W. Mark
Department of Electrical Engineering
University of Waterloo

Scientific Authority

Dr. J.L. Pearce
Communications Research Centre
Ottawa

WRI Project No. 808-01-04.

March 1982

TABLE OF CONTENTS

		page
1.	Introduction.	1
2.	Sequence Acquisition using Bit Estimation Techniques.	2
	2.1 Introduction	2
	2.2 Review of Acquisition Techniques	4
	2.3 State Estimate Bit Error Probabilities using Channel Information	9
	2.4 Comparison of System Performance	20
	2.5 Comments	28
	Appendix	30
3.	Coding Options for the Interference Channel	34
	3.1 General System Considerations and Assumptions	34
	3.2 Coding Options	37
	3.3 Comparison of Coding Complexity	49
	3.4 Coding Recommendations	50
	3.5 Interleaver Recommendations	56
4.	Modulation and Coding for Digital Communication over an Interference Channel	59
	4.1 Digital Communication over an Additive White Gaussian Channel	59
	4.2 Communication over a Partial-Band Jamming Channel	65
	4.3 Comments	68

LIST OF FIGURES

		page
Fig. 2.1	Deviation in Probability of Bit Correctness between the Central Limit Theorem Approximation and the Exact Expression, n = 100 parity checks	15
Fig. 2.2	Probability of Bit Correctness vs Noise Variance for various functions, n = 100	16
Fig. 2.3	Probability of Bit Correctness vs Number of Trinomial Parity Checks for the function $(a_1+n_1)(a_2+n_2)$	17
Fig. 2.4	Output Probability Bit Correctness vs Input Probability Bit Correctness using r Orthogonal Parity Checks	19
Fig. 2.5	Acquisition Time vs Noise Variance for ℓ = 15, k = 5	22
Fig. 2.6	Acquisition Time vs Noise Variance for ℓ = 15, k = 20	23
Fig. 2.7	Acquisition Time vs Noise Variance for ℓ = 30, k = 5	24
Fig. 2.8	Acquisition Time vs Noise Variance for k = 5, n = 200	25
Fig. 2.9	Acquisition Time vs Number of Trinomial Checks for ℓ = 15, k = 5, m = 7 using the function $(a_1+n_1)(a_2+n_2)$	26
Fig. 3.1	A Spread Spectrum Communication Model with Coding	34
Fig. 3.2	Bit Error Probability for different Coding Schemes	39
Fig. 3.3	Comparison of Bit Error Probability for different Coding Schemes	44
Fig. 3.4	Cascaded Convolutional Encoding	46
Fig. 3.5	A Model for Random Interleaving	56
Fig. 4.1	A Typical Reliability Function E(R)	59
Fig. 4.2	Functional Block Diagram of a Digital Communication System	61
Fig. 4.3	Curves of Cutoff Rate for Binary Signalling	66

1. Introduction.

Several aspects of the performance of communication systems operating on an interference or jamming channel are considered in this report.

In the next section a new technique to acquire synchronism with a pn sequence is described and analyzed. The technique is particularly effective for long shift register lengths and low signal-to-noise ratios where other techniques are inoperable. It involves more signal processing than other schemes and requires bit synchronism. This work was done with a student, Mr. Gordon Stuber and will be submitted for publication.

Section 3 considers the various coding options available for the interference channel and their performance. While this work was done for a specific application, it is viewed also as a general background for further work into the problem. Many interesting questions are raised for future investigation.

There have recently appeared a number of conference articles which attempt to analyze the interference channel from an information theoretic/computational cut-off point of view. A brief review of these is given in section 4. It is hoped that future work will extend these approaches and allow them to be used with specific coding systems for evaluation on interference channels. The aim of such work would be to obtain a better understanding of communicating in the presence of intentional interference.

2. Sequence Acquisition using Bit Estimation Techniques.

2.1. Introduction.

Many communication systems use pseudonoise (PN) sequences for either their spectral or acquisition properties. They are especially important for the successful operation of many spread spectrum and multiple access systems where the spectrum spreading and signal discrimination depend crucially on the sequence properties. The performance of these systems depends on the ability of a local generator to synchronize itself with the sequence in the received signal and many techniques have been suggested to achieve this.

Perhaps the simplest technique uses a sliding correlator containing either the full or partial sequence which is correlated with the received signal. Exceedance of some threshold after correlation over some predetermined interval will be an indication of synchronism. If synchronism is not achieved the received sequence is slipped by one bit or chip interval and correlation is again tried. This technique is viewed as essentially passive in that the synchronizer makes no attempt to estimate the state of the sequence generator but waits until the state of the received sequence matches that in its correlator. On average then it will take approximately half the length of the sequence to acquire synchronism. In systems using very long sequences this may be unacceptable.

In these situations techniques which actively estimate the state of the received sequence are likely to perform better. The RASE (rapid acquisition by sequential estimation) method of Ward [1] and the RARASE (recursion-aided RASE) method of Ward and Yiu [2] operate in this manner and give excellent performance at moderate signal-to-noise ratios (SNR's) and relatively short shift register lengths.

The threshold decoding estimate technique of Pearce and Ristenbatt [3] and the majority logic decoding technique of Kilgus [4] use knowledge of the shift register feedback taps to generate parity checks from which a state estimate is obtained. These estimates however are derived from hard quantized data and hence involve a loss of channel information. The technique introduced here also uses parity checks but in a unique manner to take more advantage of the channel information. This method is discussed in section 3. It is more complex and difficult to implement than either the RASE or RARASE methods and its performance is superior for sequences of very long length operating in very low SNR's. For comparison purposes the other techniques will be introduced in the next section. Comparisons between the new technique introduced in this paper and these other methods are made in section 4.

The problems of automatic gain control and synchronization, or their effects on the analysis, are not considered here. These considerations are likely to be similar for any system and should not materially affect the comparisons. In the system which motivated this study, a master clock was available, uncontaminated with noise, and bit synchronization was not a significant problem. In many other systems a coarse synchronization can be maintained. The techniques introduced here depend upon chip interval synchronization being available.

2.2 Review of Acquisition Techniques.

Numerous techniques have been devised for the acquisition of PN sequences under a variety of conditions and assumptions. The model assumed for this paper, to which all other methods will be translated, is as follows. The PN sequence will be assumed to have a feedback generator polynomial h(x) of degree ℓ, the sequence having period $L = 2^{\ell}-1$. The binary (0,1) sequence will be designated by $\{b_i\}$ and the binary (-1, +1) sequence actually transmitted, by $\{a_i\}$, $a_i = 1-2b_i$. The received sequence will be denoted by $y_i = a_i + n_i$ where n_i is a noise variable assumed to be normally distributed with mean 0 and variance σ^2, $n_i \sim N(0,\sigma^2)$, n_i and n_j independent random variables, $i \neq j$. The noise variables are intended to reflect any processing such as low pass and matched filtering prior to sampling.

In the RASE technique ℓ consecutive received samples y_i are hard quantized to form a (-1, +1) sequence from which an estimate of the shift register state is obtained. The probability that any particular bit is correct is given by

$$p = P(y_i > 0 \mid a_i = +1)$$

$$= \int_0^{\infty} \frac{\exp(-(y-1)^2/2\sigma^2)}{\sqrt{2\pi}\,\sigma}\, dy$$

$$= \int_{-1/\sigma}^{\infty} \frac{\exp}{\sqrt{2\pi}}\,(-z^2/2)\, dz = \Phi\,(1/\sigma).$$

The probability the state estimate is correct is then p^{ℓ}. For each such estimate the local register is initiated in that state and correlated with the received sequence for an examination period of N_e chips at the end of

which a decision on the initial state is made. If X is the number of trials to achieve acquisition then

$$P(X=k) = (p^{\ell})(1 - p^{\ell})^{k-1}, \quad k=1,2,...$$

and the average acquisition time (in chips) is then

$$N_a = (\ell + N_e)/p^{\ell}$$

and the probability of acquisition on or before the kth trial is

$$1 - (1-p^{\ell})^{k}.$$

Notice that with the sliding or stepping correlator, on average about half the states would be examined to achieve acquisition yielding

$$N_a \cong 2^{\ell-1} N_e.$$

The recursion aided version of RASE attempts to check the validity of each state estimate before commencing an examination interval, thereby eliminating an examination period for those state estimates which fail. The technique is to use 3-input modulo 2 adders on the state estimate and incoming estimate. If p is the probability a given symbol is correct, the probability the recursion on the three input bits is satisfied is the probability that either no errors or two errors have occurred i.e. $p^3 + 3p(1-p)^2$. If m such adders are used and no bits in the estimate are checked more than once then the probability of attempting to track is

$$P(AT) = (p^3 + 3p(1-p)^2)^m \qquad (2.1)$$

The probability that all $\ell+m$ bits used in the estimation and checking are correct is $p^{\ell+m}$. The conditional probability that the ℓ bit state estimate is correct, given that tracking is attempted (the m checks are satisfied) is P(ST), the probability of a successful track,

$$P(ST) = p^{\ell+m} / (p^3 + 3p(1-p)^2)^m = p^{\ell}/(p^2 + 3(1-p)^2)^m. \qquad (2.2)$$

The ratio of acquisition times between RASE and RARASE is given by

$$\frac{T_{a,RARASE}}{T_{a,RASE}} = (p^2 + 3(1-p)^2)^m$$

if it is assumed that the correlation time is much greater than the phase examination period. A similar but more complicated expression can be derived when the 3-input mod 2 adders are allowed to overlap in the bits they check. When a set of $\ell+m$ bit estimates satisfies the m checks the register is loaded with that estimate and the incoming sequence correlated with the locally generated version for N_e chips. This examination time, and the correlation threshold at the end of it, is set to achieve a given probability of false alarm and false dismissal. For an examination time of N_e chip intervals the average number of chip intervals to achieve synchronization, $N_{a,RARASE}$, is easily derived [2] as follows. The average number of chip intervals that must be processed to give a state estimate that passes the checking adders is

$$\ell+m-1 + \frac{1}{P(AT)}$$

The average number of such "successful" state estimates that must be examined before a correct one is produced is just 1/P(ST) and for each such estimate N_e further bits are processed during the examination interval. Thus the average number of bits examined to achieve acquisition is given by

$$N_{a,RARASE} = \left[\ell+m-1 + \frac{1}{P(AT)}\right] \frac{1}{P(ST)} + \frac{N_e}{P(ST)} . \qquad (2.3)$$

The length of the examination interval, N_e, determines the false alarm and false dismissal probabilities, important parameters for the performance of the system. Expressions for these probabilities will not be considered here.

In the RARASE approach to acquisition, knowledge of the sequence structure is used only to check the validity of the state estimate. A more active method of using this information to obtain an actual estimate of the state using a majority logic, threshold decoding approach is given by Pearce and Ristenbatt [3] and Kilgus [4]. The two approaches are essentially equivalent and the result of Kilgus is briefly described. There are $2^{\ell-1}-1$ parity checks of weight 3 orthogonal on a given position of the sequence. Together with the received bit in the given position a total of $2^{\ell-1}$ estimates result. If q is the probability a hard quantized received bit is in error then the probability a given parity check equation is in error is $Q = 2q(1-q)$. If r parity check equations are used, along with the quantized bit itself, to estimate the bit and r is assumed to be even then the probability the resulting majority logic estimate of the bit is in error is given by

$$q_b = \sum_{k=0}^{\frac{r}{2}-1} \binom{r}{r-k} Q^{r-k} (1-Q)^k + q\binom{r}{\frac{r}{2}} Q^{r/2} (1-Q)^{r/2}$$

and $p_b = 1-q_b$. The bit estimates of the state of the shift register will in general use the same bits in the received sequence and hence the bit estimates are not in general independent. A lower bound on the

probability that the state estimate is correct is $1-nq_b$ (it is readily observed that for q_b small, $(1-q_b)^n \cong 1 - nq_b$). In the next section another technique for using parity checks of weight 3 to obtain bit estimates is introduced and its relationship to that of Kilgus [4] will be clear.

The problem of how to use the sequence structure to obtain bit estimates in an easily implementable way, appears to be difficult. An aposteriori probability approach to the problem using orthogonal parity checks is described by Massey [5] and the performance of a simplified version of it is considered by Tanaka et al [6]. The simplified algorithm is iterative in nature and uses approximations to certain log likelihood ratios. The algorithm performance can only be approximated accurately at high signal-to-noise ratios. For these reasons it was not felt to be suitable for the application of interest here where computational simplicity at low signal-to-noise ratios are important.

The techniques described so far share the common feature that the received sequence must be hard quantized on a chip by chip basis in order to use the parity check information for bit estimation. The technique introduced in the next section uses these parity checks on soft decisions in a novel way to obtain improved bit estimates on the state of the sequence. These improved bit estimates are then used with the ordinary RARASE technique to obtain shorter acquisition times. Thus the central contribution of the paper is to show how to obtain these improved bit estimates and to evaluate their effect on acquisition times.

It is implicit in this discussion that techniques using parity checks are inherently more complicated than the correlator, RASE or RARASE methods and are really only acceptable in situations where the signal must be acquired in relatively low SNR's or for relatively long shift register lengths. In such situations the only alternative is to trade complexity for performance.

2.3. State Estimate Bit Error Probabilities using Channel Information.

The binary (0,1) PN sequence of length $2^{\ell}-1$ is the dual of the Hamming code and is often referred to as the simplex code. The Hamming code of length $(2^{\ell}-1)$ has $(2^{\ell}-1)(2^{\ell}-2)/6$ codewords of weight 3 and if the ith column of the parity check matrix H (generator matrix of the Hamming code) is the representation of α^i with respect to the standard basis in $GF(2^{\ell})$ over GF(2), where α is a primitive element of $GF(2^{\ell})$, then each such parity check of weight 3 corresponds to a "trinomial" equation of the form $1 + \alpha^i + \alpha^j = 0$, in $GF(2^{\ell})$ $1 \le i, j \le 2^{\ell}-2$. The methods of generating these trinomials is an interesting question and some comments are made on the problem in the appendix. Assume that n such trinomials are available. In coding theory terms they represent n orthogonal parity checks on a given bit of the PN sequence. Thus the trinomial $1 + \alpha^i + \alpha^j = 0$ represents the parity check equation

$$b_o + b_i + b_j = 0 \qquad (\text{mod } 2).$$

on the binary (0,1) sequence $\{b_i\}$. On the binary (-1,+1) sequence the equation implies that if $a_o = 1-2b_o = +1$ then a_i and a_j are of like sign, either both +1 or both -1. If $a_o = -1$ then a_i and a_j are of opposite sign. It is this fact which is exploited in this paper to use channel information to estimate a_o assuming we have the n trinomials (parity checks) $1 + \alpha^{i_s} + \alpha^{j_s} = 0$, $s = 1,2,\ldots,n$.

Consider the estimate of a_o, $\hat{a}_o^{(s)}$, formed by

$$\hat{a}_o^{(s)} = (a_{i_s} + n_{i_s})(a_{j_s} + n_{j_s}),\ 1 \le s \le n, n_{i_s}, n_{j_s} \sim N(0,\sigma^2)$$

It is readily seen that $\hat{a}_o^{(s)}$ is a random variable with mean $\mu_e = +1$ if $a_o = +1$ and a mean of $\mu_e = -1$ if $a_o = -1$. In either case the variance of $\hat{a}_o^{(s)}$ is $\sigma_e^2 = 2\sigma^2 + \sigma^4$. The normalized sum of these estimates

$$Z = \frac{1}{n} \sum_{s=1}^{n} \hat{a}_o^{(s)}$$

is, by the Central Limit Theorem ([7]), approximately a normally distributed random variable with mean ± 1 and variance

$$\sigma_n^2 = (2\sigma^2 + \sigma^4)/n.$$

The final (hard) estimate of a_o is then given by the symmetric 2-level quantization of the statistic Z. The probability this bit estimate is correct is given by

$$p_b = \int_o^\infty \frac{1}{\sqrt{2\pi}\,\sigma_n} \exp\left(-\frac{(y-1)^2}{2\sigma_n^2}\right) dy = \Phi(1/\sigma_n)$$

For n large enough the variance of this estimate can be reduced below σ^2 and thus discrimination between the two cases considerably improved over that obtained by using a hard quantization of $a_o + n_o$ for which the probability of a correct bit estimate is $\Phi(1/\sigma)$.

An unfortunate aspect of the estimate $\hat{a}_o^{(s)} = (a_{i_s} + n_{i_s})(a_{j_s} + n_{j_s})$ is the fact that its variance increases as σ^4 and for large values of σ it requires a large number of parity checks to reduce σ_n below σ and thus improve the probability the bit estimate is correct. It is desirable to

choose a function

$$\hat{a}_o^{(s)} = f(a_{i_s} + n_{i_s}, a_{j_s} + n_{j_s})$$

such that the bit estimate obtained by hard quantizing

$$Z = \frac{1}{n} \sum_{s=1}^{n} \hat{a}_o^{(s)}$$

has a maximum probability of being correct. The authors were unable to find a solution to this problem and several functions were tested. The results for these are reported on briefly below.

Perhaps the simplest such function to implement is to hard quantize each received bit and form the estimate

$$\hat{a}_o^{(s)} = \text{sgn}\,(a_{i_s} + n_{i_s})\ \text{sgn}\,(a_{j_s} + n_{j_s})$$

where $\text{sgn}(\cdot)$ is the signum function. Such a function is related to the work of Pearce and Ristenbatt [3] and Kilgus [4] but is used here in a multiplicative rather than additive manner. Again, the estimate

$$Z = \frac{1}{n} \sum_{s=1}^{n} \hat{a}_o^{(s)}$$

is formed and hard quantized to give the estimate of $\hat{a}_o$. For this estimate it is possible to derive both an exact expression and an approximation to the probability the resulting bit estimate is correct. To develop the approximation it is noted that the mean of $\hat{a}_o^{(s)}$ can be shown to be

$$\mu_e = a_{i_s}\, a_{j_s}\, (2\Phi(1/\sigma)-1)^2$$

and the variance is

$$\sigma_e^2 = 1 - \mu_e^2 .$$

Applying the Central Limit Theorem to the statistic Z gives the approximation to the probability the bit estimate being correct as

$$p_b = \Phi\left(\frac{\mu_e\sqrt{n}}{\sigma_e}\right) .$$

As mentioned, an exact expression for this probability can be determined as follows. The probability that an individual estimate $\hat{a}_o^{(s)}$ is correct is just

$$\begin{aligned} p &= P((1+n_i)>0 \text{ and } (1+n_j)>0) + P((1+n_i)<0 \text{ and } (1+n_j)<0) \\ &= P((1+n)>0)^2 + P((1+n)<0)^2 \\ &= \Phi(1/\sigma)^2 + (1-\Phi(1/\sigma))^2 \end{aligned}$$

assuming that $a_o = +1$. The same expression results if it is assumed $a_o = -1$. If the final decision function is

$$Z = \operatorname{sgn}\left(\sum_{s=1}^{n} \operatorname{sgn}(a_{i_s}+n_{i_s})\ \operatorname{sgn}(a_{j_s}+n_{j_s})\right)$$

then the probability of a final correct estimate for a_o is just the probability that more than one half of the individual estimates are correct:

$$p_b = \sum_{i=\left\lceil\frac{n-1}{2}\right\rceil+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

where in the event of a tie, when m is even, it is assumed the bit estimate is wrong.

It is interesting to compare this exact result with the previous approximation. The results are shown graphically for the range of variances of interest in Figure 2.1.

The two functions considered so far are perhaps the simplest possible. In the search for a function which would provide improved discrimination (higher probability of correct bit estimate for a given noise variance) several other functions were examined. Some of these are given in table 1 along with expressions for their means, variances and probabilities of bit correctness. Curves for the probability of bit correctness for the range of variances of interest are given in Fig. 2.2 from which it is concluded that, for this range, the estimate $(a_{i_s}+n_{i_s})\cdot(a_{j_s}+n_{i_s})$ is superior. It is not at all certain however that functions which offer better bit estimation properties do not exist and it would be an interesting exercise to determine one. Fig. 2.3 gives curves for the probability of correct bit estimation as a function of n, the number of trinomial parity checks, for variances in the range of interest, for this function.

Also shown in Fig. 2.2 is the curve for the probability of correct bit estimate using 100 orthogonal parity checks of weight 3 on hard quantized data using majority logic as in Kilgus [4]. The curve for the probability a single hard quantized bit is correct, as is used for the RARASE technique, is also shown.

In the next section the improved (over one bit hard quantization) bit estimates offered by the functions $(a_{i_s}+n_{i_s})(a_{j_s}+n_{j_s})$ and $\mathrm{sgn}(a_{i_s}+n_{i_s})\mathrm{sgn}(a_{j_s}+n_{j_s})$, which is regarded as a particularly simple

Function	Mean = μ_e	Variance = σ_e^2	P_b, Prob. bit correctness
$(a_1+n_1)(a_2+n_2)$	a_1a_2	$2\sigma^2 + \sigma^4$	$\Phi\left(\frac{\sqrt{n}}{\sigma_e}\right)$
$\text{sgn}(a_1+n_1)\text{sgn}(a_2+n_2)$	$a_1a_2(2\Phi(\frac{1}{\sigma}) - 1)^2$	$1-\mu_e^2$	CLT approx. $\Phi\left(\frac{\mu_e\sqrt{n}}{\sigma_e}\right)$ Exact. $\sum_{i=\left\lceil\frac{n-1}{2}\right\rceil+1}^{n} \binom{n}{i} p^i(1-p)^{n-i}$ $p = \Phi(\frac{1}{\sigma})^2 + (1 - \Phi(\frac{1}{\sigma}))^2$
$\lvert a_1+a_2+n_1+n_2\rvert$	$\mu_{e,1}=2\left(2\Phi\left(\frac{\sqrt{2}}{\sigma}\right)-1\right) + \frac{2\sigma}{\sqrt{\pi}} e^{-1/\sigma^2}$, $a_1a_2=+1$ $\mu_{e,2} = \frac{2\sigma}{\sqrt{\pi}}$, $a_1a_2=-1$	$\sigma^2_{e,1}=(2\sigma^2+4)\left(\Phi\left(\frac{\sqrt{2}}{\sigma}\right) + \Phi\left(-\frac{\sqrt{2}}{\sigma}\right)\right)-\mu^2_{e,1}$ $\sigma^2_{e,2} = 2\sigma^2\ (1 - \frac{2}{\pi})$	no simple expression
$g(a_1+n_1)g(a_2+n_2)$ $g(x) = \begin{cases} x\ , & -1\le x\le 1 \\ 1 & x \ge 1 \\ -1 & x \le -1 \end{cases}$	$a_1a_2\left[(2\Phi(\frac{2}{\sigma})-1) - \frac{\sigma}{\sqrt{2\pi}}\ (1-e^{-2/\sigma^2})\right]^2$	$\left[1 - \sigma\ \frac{\sqrt{2}}{\pi} + \frac{\sigma^2}{2}\ (2\Phi(\frac{2}{\sigma}) - 1)\right]^2-\mu_e^2$	$1 - \Phi\left(\frac{\mu_e\sqrt{n}}{\sigma_e}\right)$
$\text{sgn}(a_1+n_1)\text{sgn}(a_2+n_2)\cdot$ $\lvert a_1+n_1\rvert^{1/3}\lvert a_2+n_2\rvert^{1/3}$	(parameters determined by numerical integration)		

Table 1.

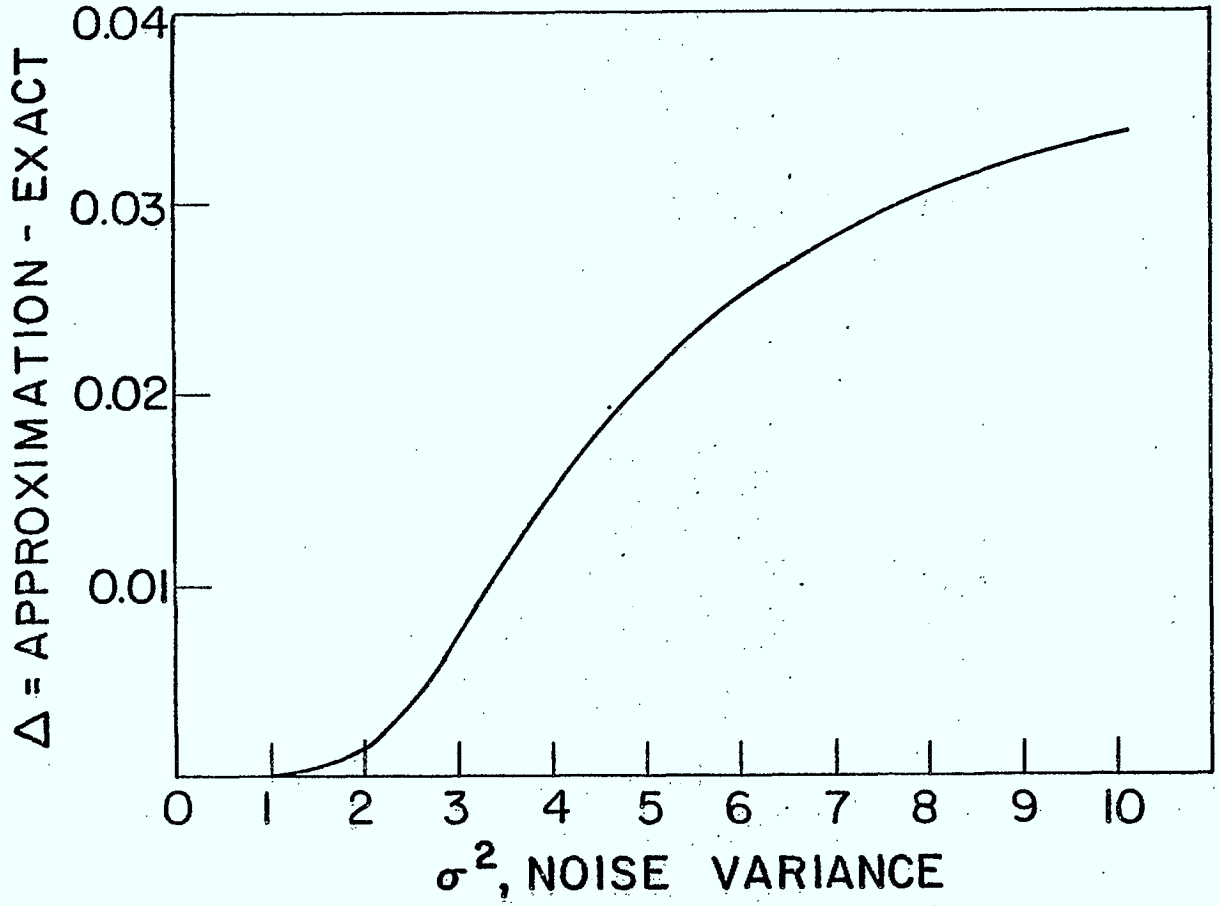


Figure 2.1

Deviation in Probability of Bit Correctness between the Central Limit Theorem Approximation and the Exact Expression n = 100 parity checks.

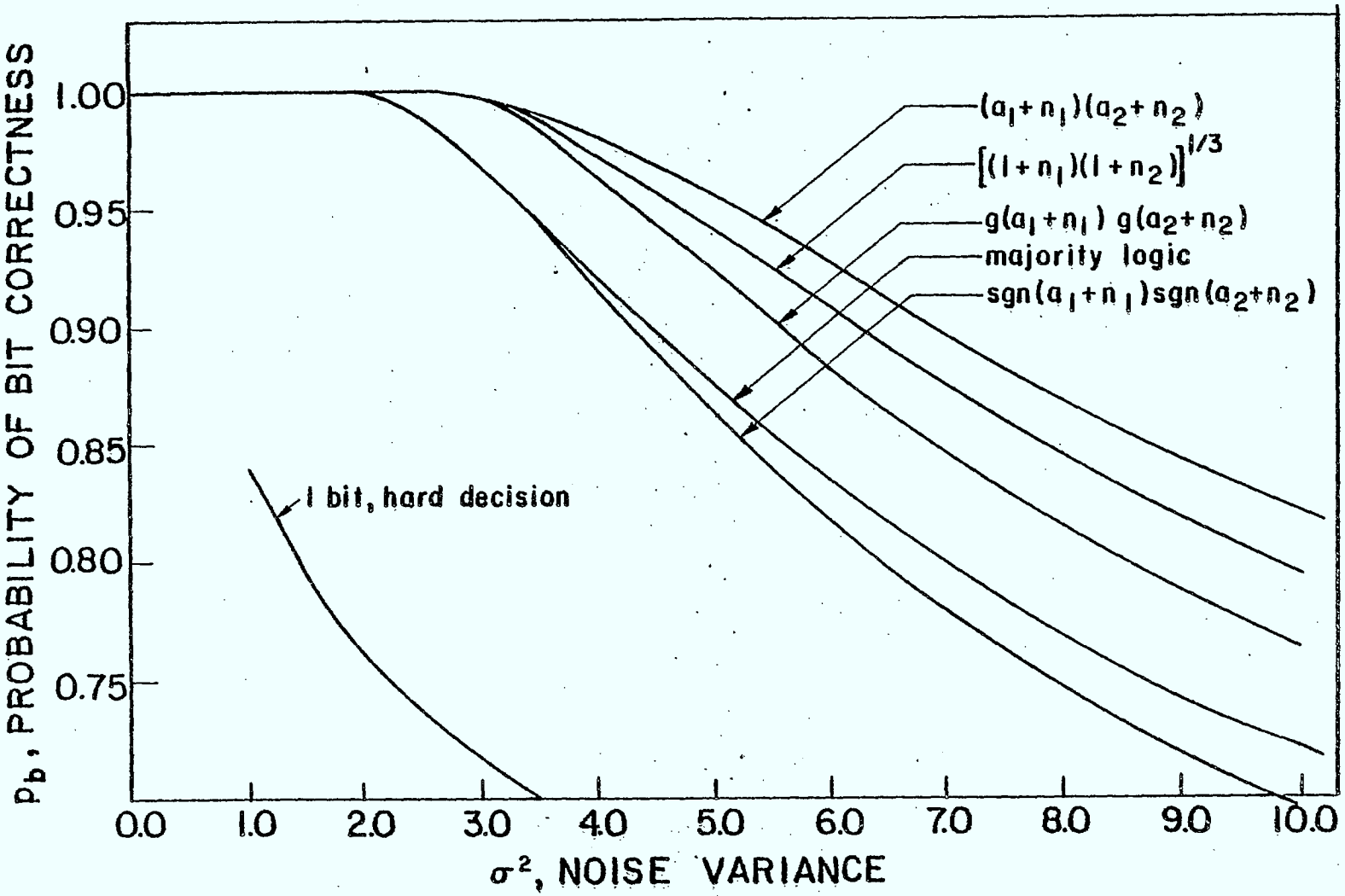


Figure 2.2

Probability of Bit Correctness vs Noise Variance for various functions, n = 100

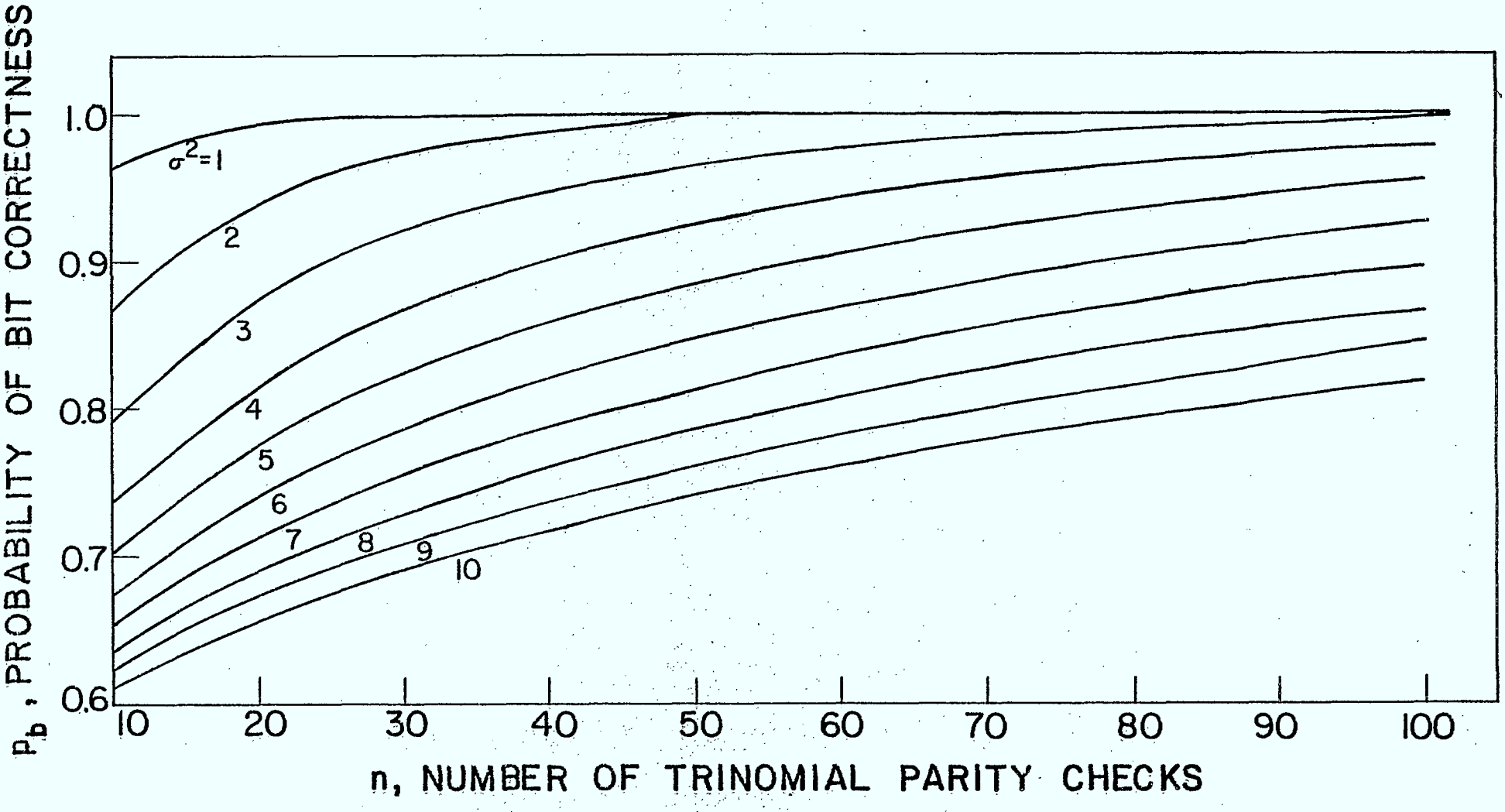


Figure 2.3

Probability of Bit Correctness vs Number of Trinomial Parity Checks for the function $(a_1+n_1)(a_2+n_2)$

function to implement, are used with the RARASE acquisition method and compared in performance with the standard RARASE method. It should be noted again that even the function $(a_{i_s}+n_{i_s})(a_{j_s}+n_{j_s})$ is not optimal in the sense of maximizing the probability of correct bit estimation but is used here simply as a convenient and simple function which offers improved performance over other techniques.

As a final comment of this section it is noted that after obtaining bit estimates of the state, the probability an individual bit in this estimate is correct can be improved by using majority logic techniques on binary data. Suppose for example that r orthogonal parity checks are available on a given bit and that each bit used by these parity checks has a probability p_i of being correct. Thus the estimation techniques described in preceding paragraphs would have to be used on each bit used by these r orthogonal parity checks. These parity checks can then be used in the standard majority logic manner to further improve the probability of bit correctness, p_o. The relationship between p_i and p_o using r orthogonal parity checks is shown in Fig. 2.4. This technique, which uses orthogonal parity checks in two distinct ways, first using them in a soft decision manner to arrive at a hard bit estimate, and then using the second set in a standard majority logic way to enhance the probability of bit correctness, is complicated to implement and is considered no further here. It does however present a real possibility for system improvement.

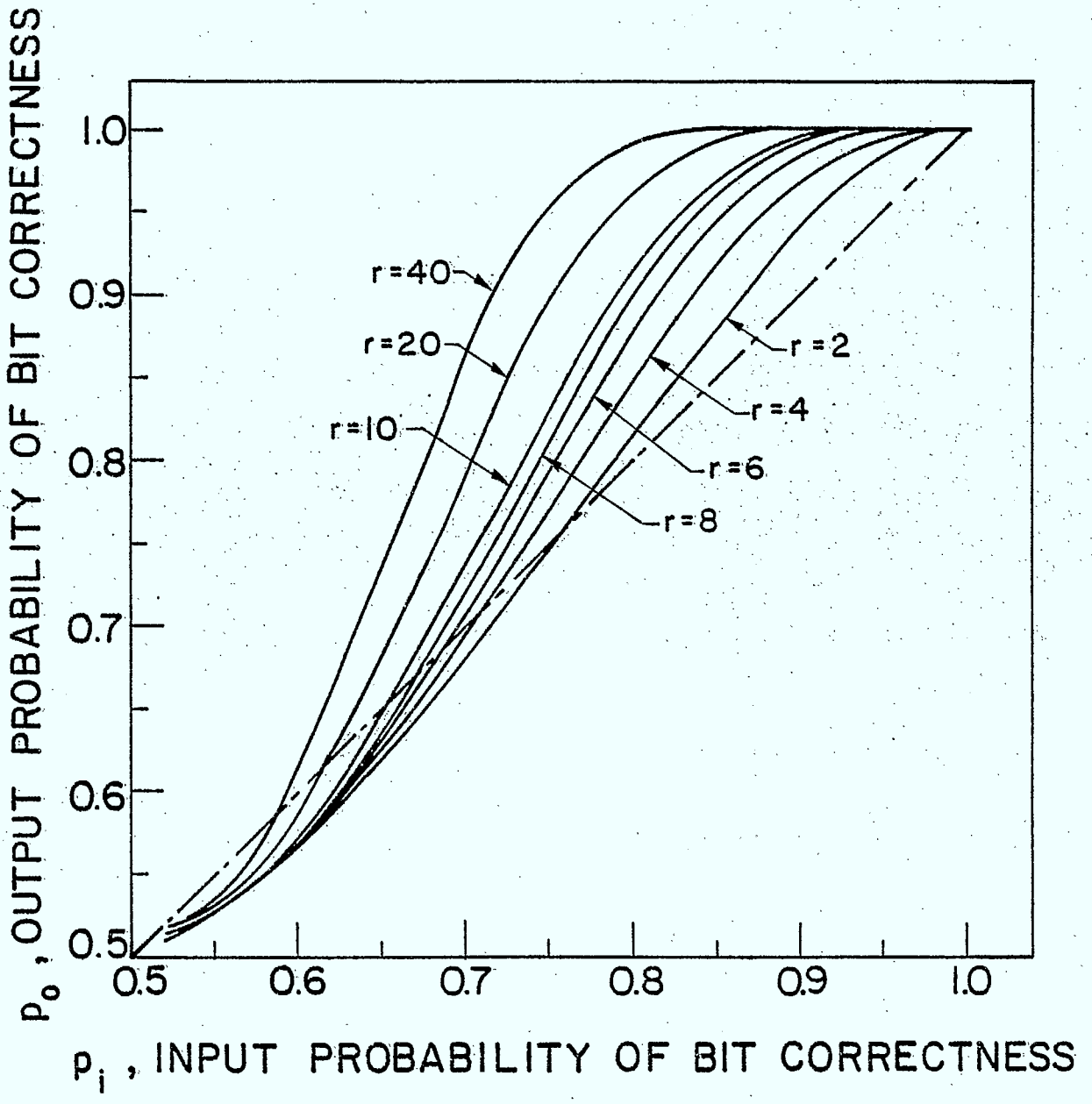


Figure 2.4

Output Probability Bit Correctness vs Input Probability Bit Correctness Using r Orthogonal Parity Checks.

2.4. Comparison of System Performance.

In this section the average number of chip intervals it is required to examine to achieve synchronism is compared for the standard RARASE system and the RARASE system which incorporates the bit estimation techniques introduced in the previous section. Only the bit estimation functions $(a_i+n_i)(a_j+n_j)$ and $\text{sgn}(a_i+n_i)\text{sgn}(a_j+n_j)$ are used and it is noted from figure 2 that this last function gives slightly poorer performance than the majority logic method of Kilgus [4] but is very simple to implement. A precise comparison of the systems is difficult to achieve because of certain assumptions which must be made.

For the RARASE system the average number of chip intervals which must be examined to achieve synchronism, $N_{a,RARASE}$, is given by equation (2.3) where P(AT) and P(ST) are given by equations (2.1) and (2.2) respectively and p is the probability of correct bit estimation using hard quantization. For all systems considered the number of 3 input mod 2 adders checking the phase estimation bits is assumed to be the maximum possible, $\lfloor \ell/2 \rfloor$. N_e, the number of bits examined when the phase passes all of the mod 2 adder checks, is assumed to be $k\ell$ where k is either 5 or 20. It is noted again that N_e and an appropriate threshold set the false alarm and false dismissal probabilities but expressions for these quantities will not be considered here, being comparable for all systems under the conditions stated.

The average number of chip intervals which must be examined to achieve synchronism when the bit estimation techniques are used, $N_{a,est}$, is given by

$$N_{a,est} = \left[\ell+m-1 + \frac{1}{P(AT)}\right] \frac{1}{P(ST)} + \frac{k\ell}{P(ST)} + M(\ell,n) + \ell+m-1 \qquad (2.5)$$

where P(AT) and P(ST) are as in equations (2.1) and (2.2), respectively, with p replaced by p_b. $M(\ell,n)$ is the minimum number of bits that must be considered, from the bit being estimated to have n trinomial parity checks. From the comments in the appendix on this problem it will be assumed that

$$M(\ell,n) = (2n(2^{\ell}-2))^{1/2} \qquad (2.6)$$

From the limited evidence available it would appear that this is a very reasonable approximation when $M(\ell,n)/(2^{\ell}-2) > .1$.

Two comments on the expression in equation (2.6) are in order. First it is noted that each trinomial parity check can actually be used to generate three orthogonal parity checks if it is permissible to use bits in the received sequence on either side of the bit being estimated. This would lead to a considerable saving in the number of bits that must be examined, but also leads to greater complexity and is not considered further. As a second point note that, as a measure of comparison between the two systems, the number of bits that must be "considered" is used. This is a little misleading for the system introduced here since there is considerably more processing involved on the bits used. On the other hand the number of bits actually used in the estimation process is relatively small. Assuming that processing power is relatively cheap, this method would seem to be a fair comparison.

The results are shown in Figs. (2.5) to (2.9) and are largely self explanatory. The curves using the estimating function $(a_{i_s}+n_{i_s})(a_{j_s}+n_{j_s})$

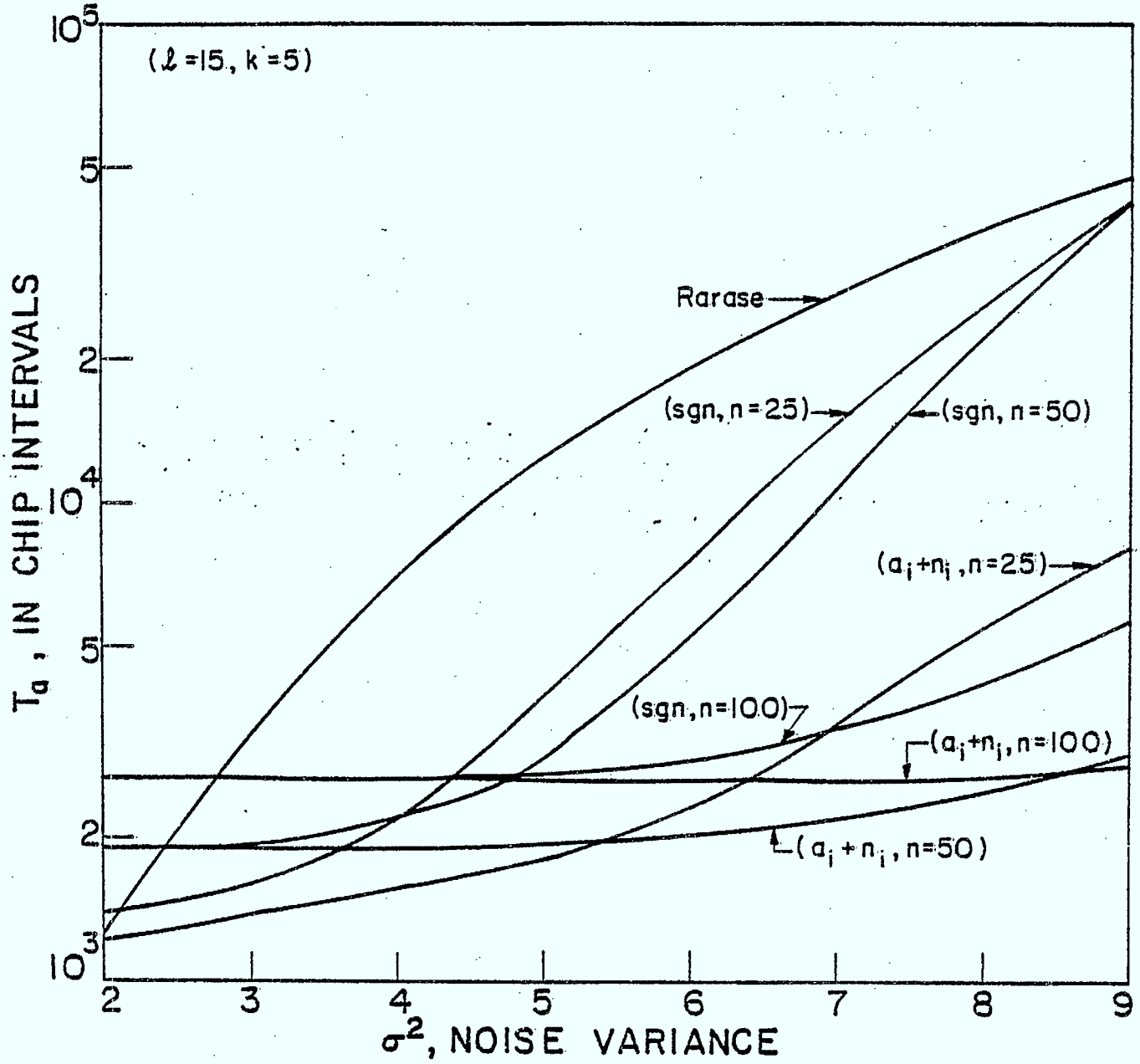


Figure 2.5

Acquisition Time vs Noise Variance for $\ell = 15$, $k = 5$

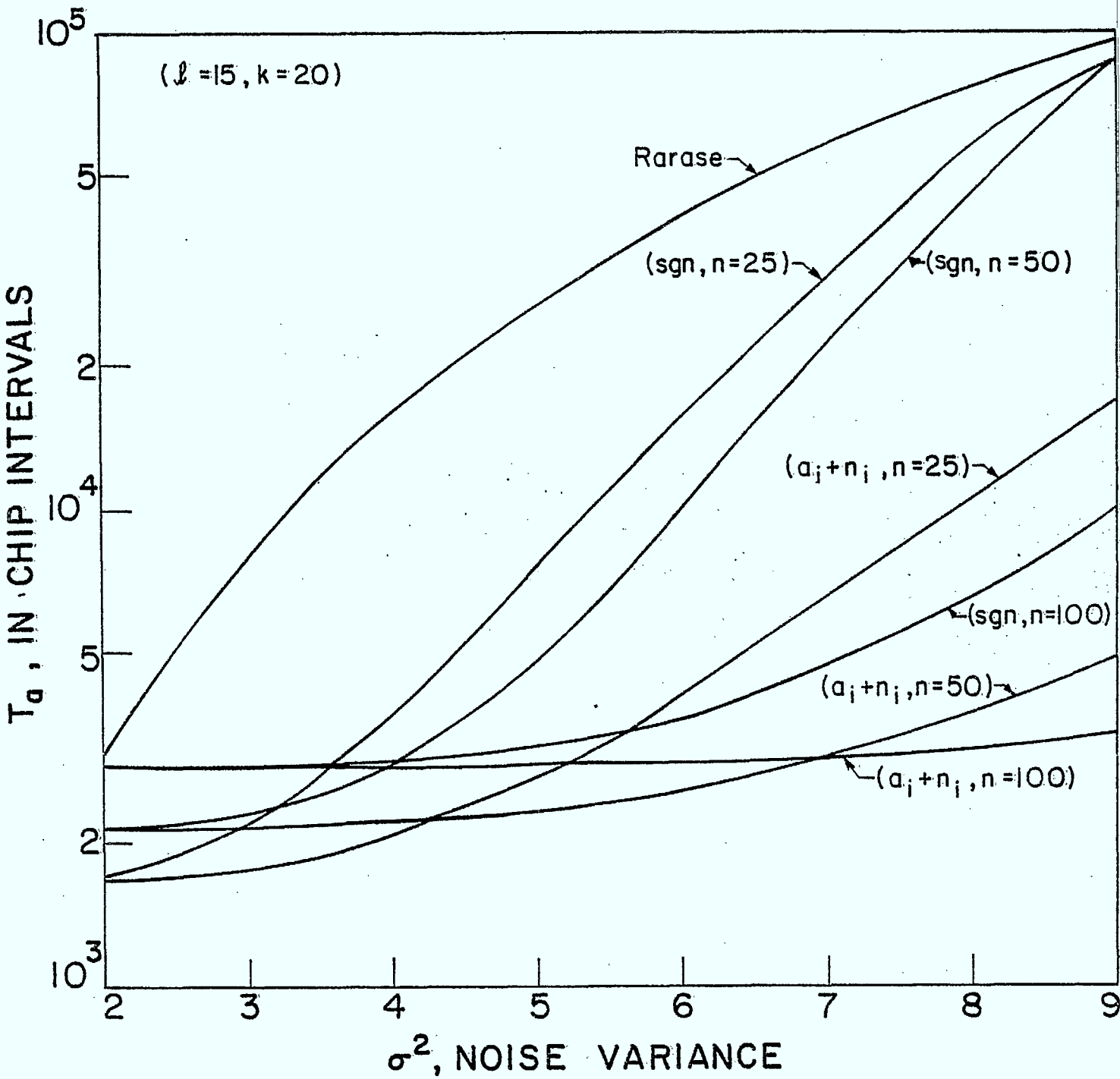


Figure 2.6

Acquisition Time vs Noise Variance for ℓ = 15, k = 20

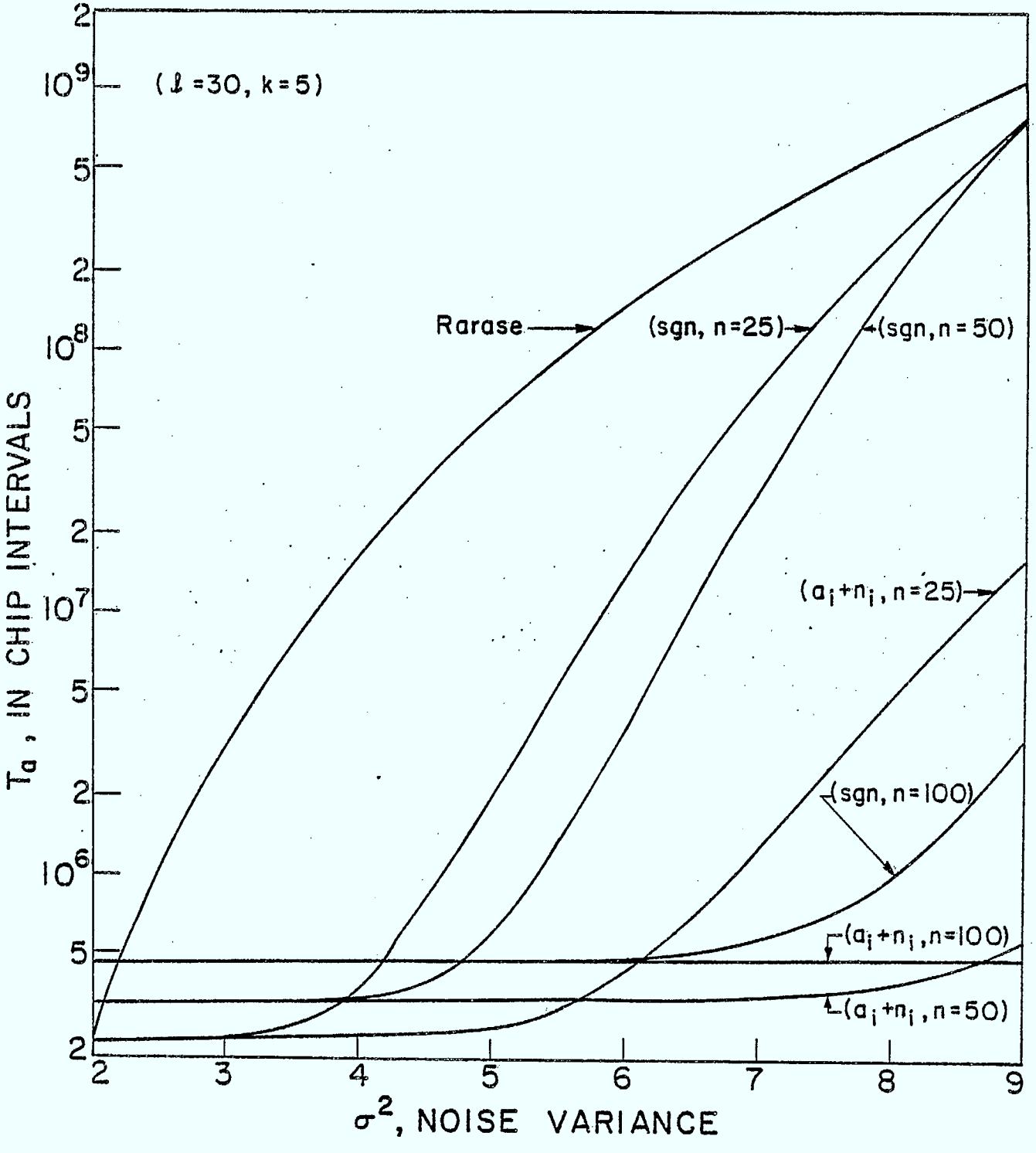


Figure 2.7

Acquisition Time vs Noise Variance for $\ell = 30$, $k = 5$

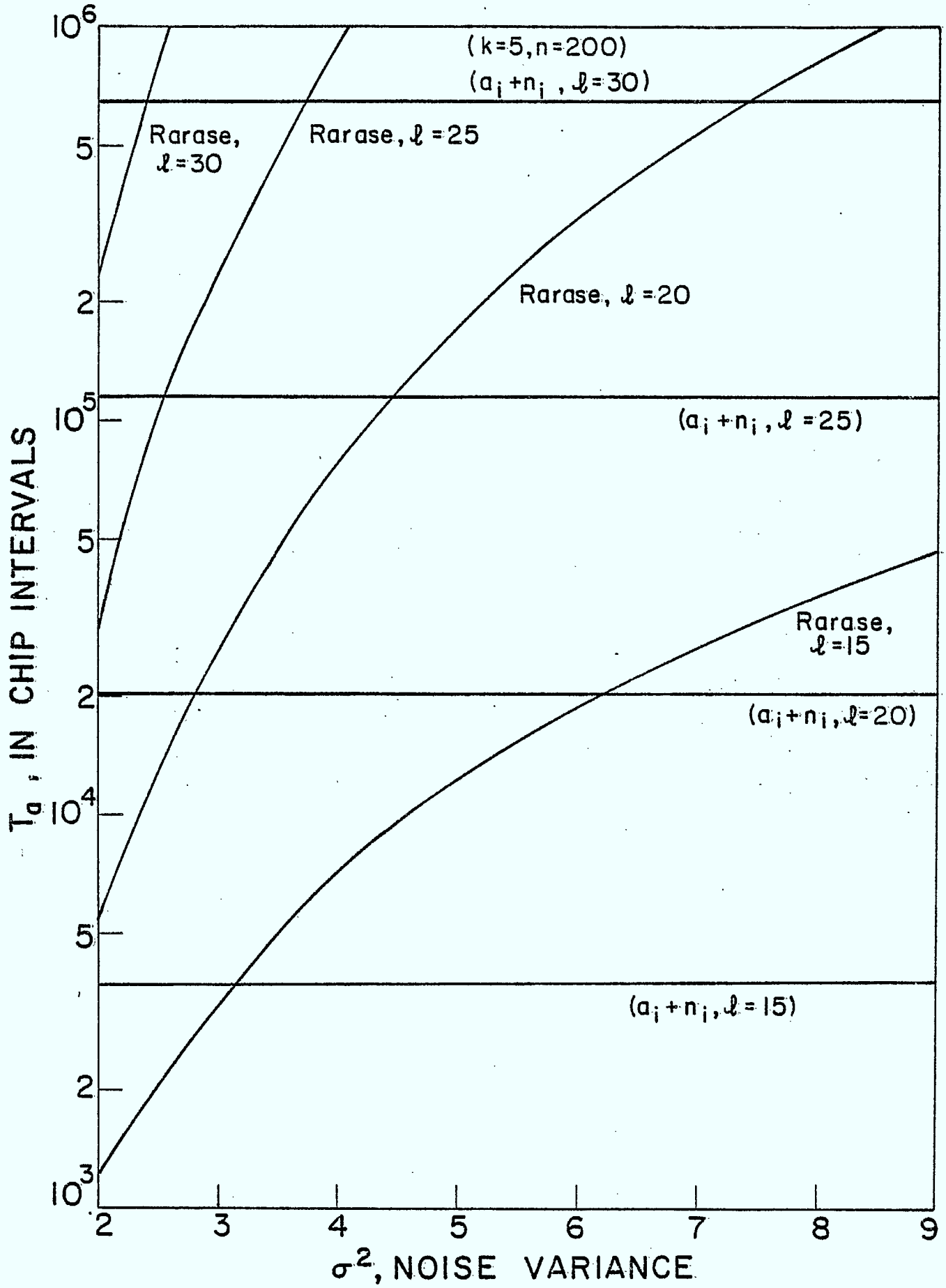


Figure 2.8
Acquisition Time vs Noise Variance for k = 5, n = 200

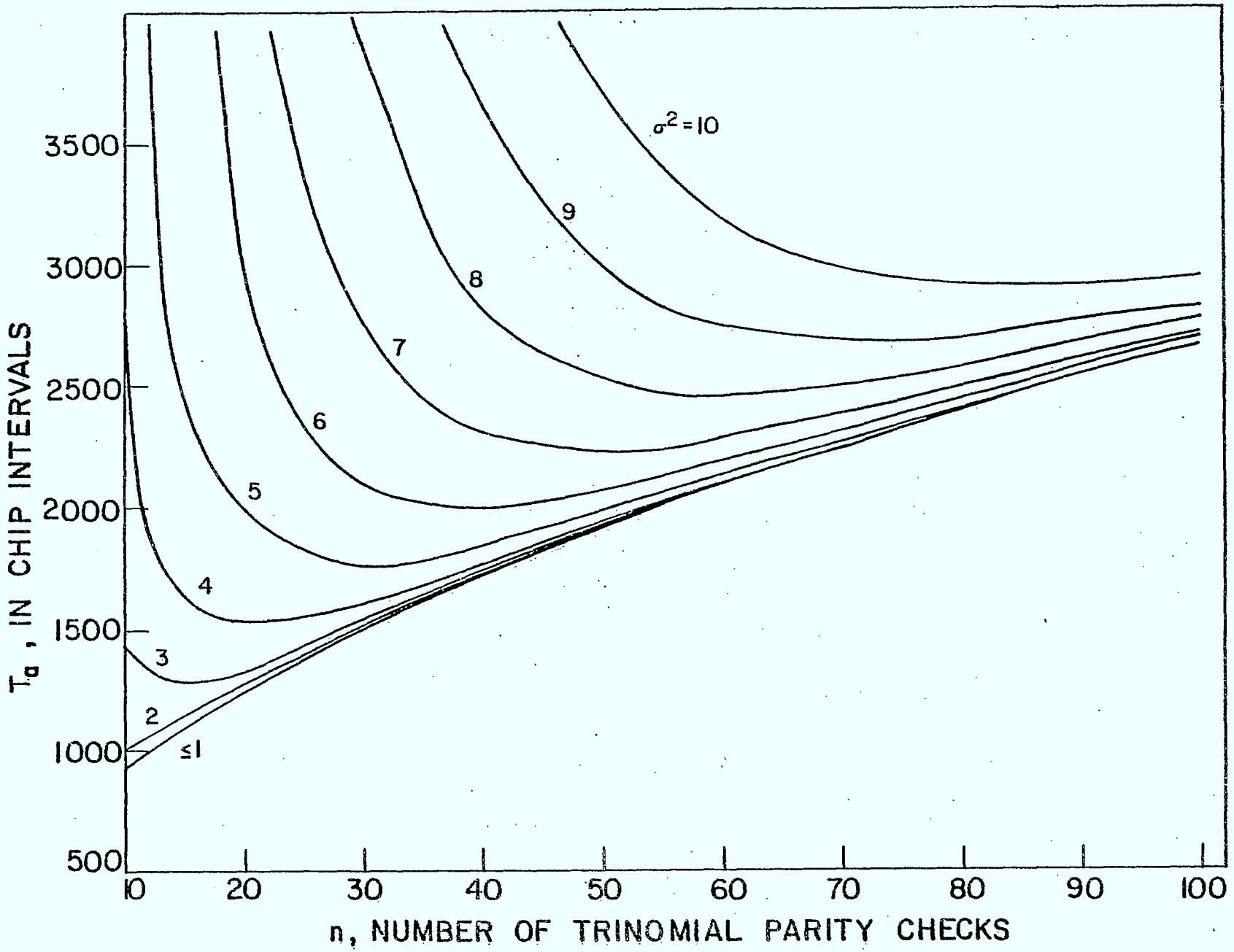


Figure 2.9

Acquisition Time vs Number of Trinomial Checks for $\ell = 15$, $k = 5$, $m = 7$ using the function $(a_1+n_1)(a_2+n_2)$

are simply marked (a_i+n_i) while the curves for the function $\operatorname{sgn}(a_{i_s}+n_{i_s})$ $\operatorname{sgn}(a_{j_s}+n_{j_s})$ are simply marked sgn. The improvement in acquisiton times over a standard RARASE system using either the (a_i+n_i) or signum function over the range of variances indicated, is clear. In Figs. (2.5) - (2.7) it is noticed that there are ranges of variances over which some optimum number of trinomial parity checks should be used. The use of more parity checks makes matters worse by forcing the examination of bits far away from the bit being estimated. This effect is shown more clearly in Fig.(2.9) for the (a_i+n_i) function where, for a given variance, the number of parity checks that should be used is clear. Fig.(2.8) illustrates the effect of increasing the shift register length, using an examination interval of $N_e = 5\ell$ bits and 200 parity checks. In all the figures only the range of variances of 2 to 9 was considered. Improvements over the standard RARASE techniques can also be achieved at lower variances by using fewer parity checks and thus, to show the bit estimation technique off to greater advantage, the number of parity checks used should be optimized for each variance. The apparent advantage of RARASE at the lower variances would then disappear.

2.5. Comments.

A new acquisition technique for PN sequences, effective for very long sequences operating at very low signal to noise ratios, as might occur for example in certain spread spectrum systems, has been discussed. Under these conditions the technique has better performance than previous methods but it is also considerably more complex. A particular problem is the generation of weight 3 parity checks with the properties required. Methods to generate these parity checks are straight forward and in systems for which the sequence is fixed for a relatively long time it is not seen to be a real problem. For some systems, such as perhaps certain antijamming spread spectrum systems, where the sequence generators might regularly change, some computing power would be required at the receiver. This tradeoff between performance and complexity for systems of the type considered here, would seem to be unavoidable.

References.

[1] R.B. Ward, Acquisition of Pseudonoise Signals by Sequential Estimation, IEEE Transactions on Comm. Tech., vol. COM-13, pp. 475-483, 1965.

[2] R.B. Ward and K.P. Yiu, Acquisition of Pseudonoise Signals by Recursion-Aided Sequential Estimation, IEEE Transactions on Communications, vol. COM-25, pp. 784-794, 1977.

[3] H.M. Pearce and M.P. Ristenbatt, The Threshold Decoding Estimator for Synchronization with Binary Linear Recursive Sequences, International Conference on Communications, 1971, pp. 43.26-43.30.

[4] C.C. Kilgus, Pseudonoise Acquisition using Majority Logic Decoding, IEEE Transactions on Communications, vol. COM-21, pp. 772-774, 1973.

[5] J.L. Massey, Threshold Decoding, Cambridge, MA: M.I.T., 1963.

[6] H. Tanaka, K. Furusawa and S. Kaneku, A Novel Approach to Soft Decision Decoding of Threshold Decodable Codes, IEEE Transactions on Information Theory, vol. IT-26, pp. 244-246, 1980.

[7] W. Feller, An Introduction to Probability Theory and its Applications, vol. 11, New York, N.Y.: J. Wiley and Sons. 1966.

[8] J.H. Conway, A Tabulation of Some Information Concerning Finite Fields, in Computers in Mathematical Research, R.F. Churchhouse and J.-C. Herz eds., Amsterdam: North-Holland, 1968.

[9] S.C. Pohlig and M. Hellman, An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance, IEEE Transactions on Information Theory, vol. IT-24, pp. 106-110, 1978.

Appendix.

It is required to generate "trinomial" equations of the form $\alpha^i + \alpha^j + 1 = 0$ in $GF(2^\ell)$ where α is a root of a given primitive polynomial. This problem actually occurs in a variety of contexts and, depending on ℓ, is generally accepted to be a difficult computational problem. For a given value of i, the corresponding value of j is called the Zech logarithm of i ([8]). The problem is also equivalent to the problem of finding logarithms in $GF(2^\ell)$. When the order of the multiplicative subgroup of $GF(2^\ell)$, $2^\ell-1$, is highly composite with the largest prime factor being small, the problem is fairly simple ([9]). When $2^\ell-1$ is prime or has a very large prime factor there is no computionally feasible algorithm known for solving the problem.

In estimating a given bit in the sequence it is desirable to form as many trinomial parity check equations as "close" to the given bit as possible. More precisely it is of interest to determine as many such parity checks $\alpha^{i_s} + \alpha^{j_s} + 1 = 0$ as possible where both i_s and j_s are both less than some given positive integer. Denote by $t = M(\ell,n)$ the largest exponent required to obtain n such parity checks. Thus in the state estimation technique, t bits from the bit being estimated, will have to "be considered". From a computational point of view, for reasonable values of n and ℓ, a brute force method appears to be as efficient as any for finding these parity checks. For example suppose each element $\alpha^i \varepsilon GF(2^\ell)$, $1 \le i \le t$, is expressed as a binary ℓ-tuple with respect to the fixed basis of $GF(2^\ell)$ over GF(2), eg $1,\alpha,\alpha^2,\ldots,\alpha^{\ell-1}$. In this representation of α^i the bit corresponding to the basis element 1 is complemented and this binary ℓ-tuple, the representation of α^j, is compared with others in the list to find the value of j, if $i<j \le t$.

For the analysis of the acquisition scheme of section 4 the determination of $M(\ell,n)$ is of interest. It is very difficult to determine exactly as it depends on the representation of the field used and the particular primitive element used. It is not the intention here to investigate this problem in detail. Instead an approximation to $M(\ell,n)$ is developed and limited experimental evidence given which supports the approximation, as far as it goes.

Consider the set of relationships $\{1+\alpha^i = \alpha^{j_i},\ i = 1,2,\ldots,2^\ell-2=L'\}$ and suppose the integers j_i are assigned at random so that the ordered L'-tuple $(j_1,j_2\ldots,j_{L'})$ is a random permutation of $(1,2,\ldots,L')$. In this model let N be the number of the j_i, $i \leq t$, that are themselves less than or equal to t. Then N is a random variable and the probability that N equals s is given by a hypergeometric distribution, under these assumptions,

$$P(N=s) = \frac{\binom{t}{s}\binom{L'-t}{t-s}}{\binom{L'}{t}}, \qquad L' = 2^\ell-2,\ t = M(\ell,n).$$

The mean of this distribution is t^2/L'. Since the trinomial relationships occur in pairs the approximation to the function $M(\ell,n)$ will be

$$\hat{n} = E(N)/2 = t^2/2L' = M(\ell,n)^2/2L'$$

$$\text{or} \quad M(\ell,n) \cong (2n(2^\ell-2))^{1/2}$$

The same result is achieved by adopting a binomial rather than hypergeometric model.

This approximation is compared with generated results for certain primitive polynomials of degree 10 and one polynomial of degree 15. These are given in table 2 along with the values for the approximation. The

amount of data given is far too limited to draw any conclusions but it would appear that the approximation is reasonable, particularly when the ratio $t/(2^{\ell}-2)$ is greater than about .1. In spite of the restricted amount of evidence, the approximation will be used in the analysis of the acquisition scheme introduced in this paper.

Primitive Polynomial	t = 50	100	150	200	250	300	350	400	450	500	550	600	650	700	750	800	850	900	950	1000
$x^{10}+x^3+1$	3	6	10	19	29	44	65	80	96	118	144	175	208	245	272	315	354	396	442	491
$x^{10}+x^8+x^3+x^2+1$	1	5	10	20	30	43	63	75	98	121	147	174	208	240	275	316	355	395	441	489
$x^{10}+x^6+x^5+x^3+x^2+x+1$	2	4	10	17	28	46	63	81	95	121	147	176	211	241	275	310	353	394	442	489
$x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+1$	0	5	10	20	32	43	60	78	103	124	150	174	206	240	274	313	354	395	440	489
Expected Number $t^2/2(2^\ell-2)$	1.2	4.9	11.0	19.6	30.6	44.0	59.9	78.3	99.1	122.3	148.0	176.1	206.7	239.7	275.2	313.1	353.5	396.3	441.6	489.2

(a) $\ell=10$

Primitive Polynomial	t = 1,000	2,000	3,000
$x^{15}+x+1$	28	71	142
Expected Number $t^2/2(2^\ell-2)$	15.3	61.0	137.3

(b) $\ell=15$

Table 2.

3. Coding Options for Interference Channels.

3.1. General System Considerations and Assumptions.

The interference or jamming channel has received considerable attention in the literature and a variety of models and assumptions arrived at. The appropriateness of these models and assumptions is very much context dependendent. The purpose of this section is to examine these models and to make specific coding recommendations regarding a particular model.

The general model under consideration is shown in figure 3.1

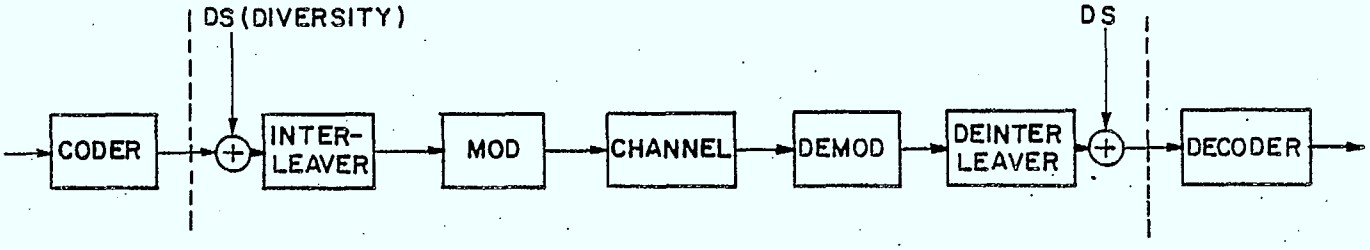


Figure 3.1

The effect of the insertion of the direct sequence (DS) is to simply add time diversity i.e. each coded bit is divided into Z chips which are interleaved to provide, at the receiver, Z independent estimates of the coded bit. It is assumed that the demodulator performs chip-by-chip matched filtering and soft decisions are presented to the input of the decoder. In practice 3-bit quantization would be sufficient, implying that the deinterleaver requires 3 times the storage requirements of the interleaver, depending on how it is implemented. It is assumed the interleaver is ideal implying that it renders the interference channel memoryless which effectively spreads burst type interference into additive white Gaussian noise for Z sufficiently large. For small Z however, the

channel remains bursty and the analysis of the coding schemes presented in succeeding sections may be applicable.

It will be assumed that the input to the receiver DS is of the form $\pm 1 + n$, $n \sim N(0,\sigma^2)$. The effect of the diversity Z will be to reduce the noise variance per symbol, after removal of the DS diversity, to σ^2/Z.

For either the pulse jamming case or the partial band jamming case, it will be assumed that at the output of the deinterleaver (neglecting quantization) the probability a chip is disturbed by noise with variance σ^2/ρ is ρ and with variance 0 is $(1-\rho)$. Thus ρ is either the duty cycle for the pulse jammer or the fraction of the band jammed for the partial band jammer. In this case it is easy to show that after the diversity is removed (equivalent to matched filtering) the symbols are of the form $\pm 1 + n'$, $n' \sim N(0,\sigma'^2)$ $\sigma'^2 = \sigma^2/Z$. Thus the pulse jamming and partial band jamming have the same effect as choosing $\rho = 1$ here, and this will be done.

The inclusion of a quantizer implies some form of automatic gain control should be used. This problem of designing such a controller for the interference channel is seen as a challenging one and is not considered here.

One type of coding system not considered in these notes is the concatenated Reed Solomon/convolutional. The only results available on this system appear to be either from experimentation or simulation of rate 1/2 convolutional codes with Viterbi decoding concatenated with high rate RS codes. The problem lies in characterizing the output error statistics of the Viterbi decoder. While it may in fact be a good system it does not seem reasonable to attempt an analysis of it at this point.

It is considered to be an expensive system and it is doubtful its performance would be significantly better than the other options considered.

3.2. Coding Options.

Various coding options that have appeared in the literature are discussed and analyzed in this section. For the purposes of comparison it will be assumed that the number of chips, after the direct sequence, per data bit is not greater than 288. Thus the code rate and the number of chips per coded symbol combine to give a ratio less than 1/288.

I. Block Orthogonal Signalling.

Code description. K bits are mapped into one of 2^K possible binary ($\pm$ 1) orthogonal sequences. The scheme will be evaluated for K = 4, 6 or 8 with diversities 72, 27 and 9 respectively. On the surface it may appear a complex task to realize, for K = 8, 256 matched filters. Because of the method of generation however, the optimum receiver can be realized with at most 32 matched filters of length 16 used sequentially in two stages or 20 matched filters of lengths 8 and 4 used sequentially in three stages.

Analysis. The analysis here is quite standard except for a computational technique due to Berlekamp [1]. Let g(x) be a normal probability density function with mean zero and variance σ^2, $g(x) \sim N(0,\sigma^2)$, let $f(x) \sim N(0,1)$ and $\Phi(x)$ the cumulative distribution function. Let $P_E(K,Z)$ be the probability of word error using 2^K orthogonal signals with diversity Z. By standard analysis, assuming inputs to the decoder are of the form $\pm 1+n$, $n \sim N(0,\sigma^2)$, then

$$P(\text{correct}) = \int_{-\infty}^{\infty} g(\alpha - 2^{K/2}) \left[\int_{-\infty}^{\alpha} g(\beta)\, d\beta \right]^{2^K - 1} d\alpha.$$

and, after simple transformation:

$$P(\text{correct}) = \int_{-\infty}^{\infty} f(x - \frac{2^{K/2}}{\sigma}) \, \Phi(x)^{2^K-1} \, dx.$$

Integrating by parts gives

$$P_E(K,Z) = (2^K-1) \int_{-\infty}^{\infty} \Phi(x - \frac{2^{K/2}}{\sigma}) \, \Phi^{2^K-2}(x) \, f(x)dx.$$

The probability of bit error is related to this expression:

$$P_b(K,Z) = \frac{2^{K-1}}{2^K-1} P_E(K,Z) = 2^{K-1} \int_{-\infty}^{\infty} \Phi(x - \frac{2^{K/2}}{\sigma}) \, \Phi(x)^{2^K-2} \, f(x)dx.$$

Curves of $P_b(K,Z)$ vs $10\log_{10}(\frac{1}{\sigma^2})$ are given in figure 3.2 for values $(K,Z) = (4,72)$, $(6,27)$ and $(8,9)$ and in each case the overall rate is 1/288.

A convenient upper bound to this expression, apparently good for $P_E \leq 10^{-3}$, is

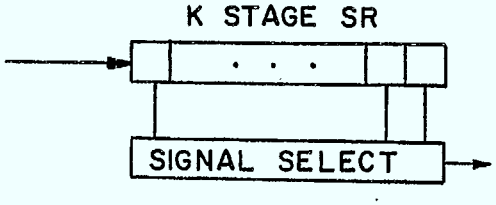
$$P_b(K,Z) \leq 2^{K-1} \, Q \, (\frac{2^{(K-1)/2}\sqrt{Z}}{\sigma}), \quad Q(x) = 1 - \Phi(x).$$

II. Orthogonal Convolutional Codes

Code Description. This method uses a K-stage shift register, bits shifted in 1 bit at a time. After each shift, one of 2^K orthogonal binary (±1) signals is selected for transmission, as in case I. Only 3 cases will be considered:

i) K=8, Z=1 ii) K=7, Z=2 iii) K=6, Z=4.

In each case the overall code rate is 1/256.



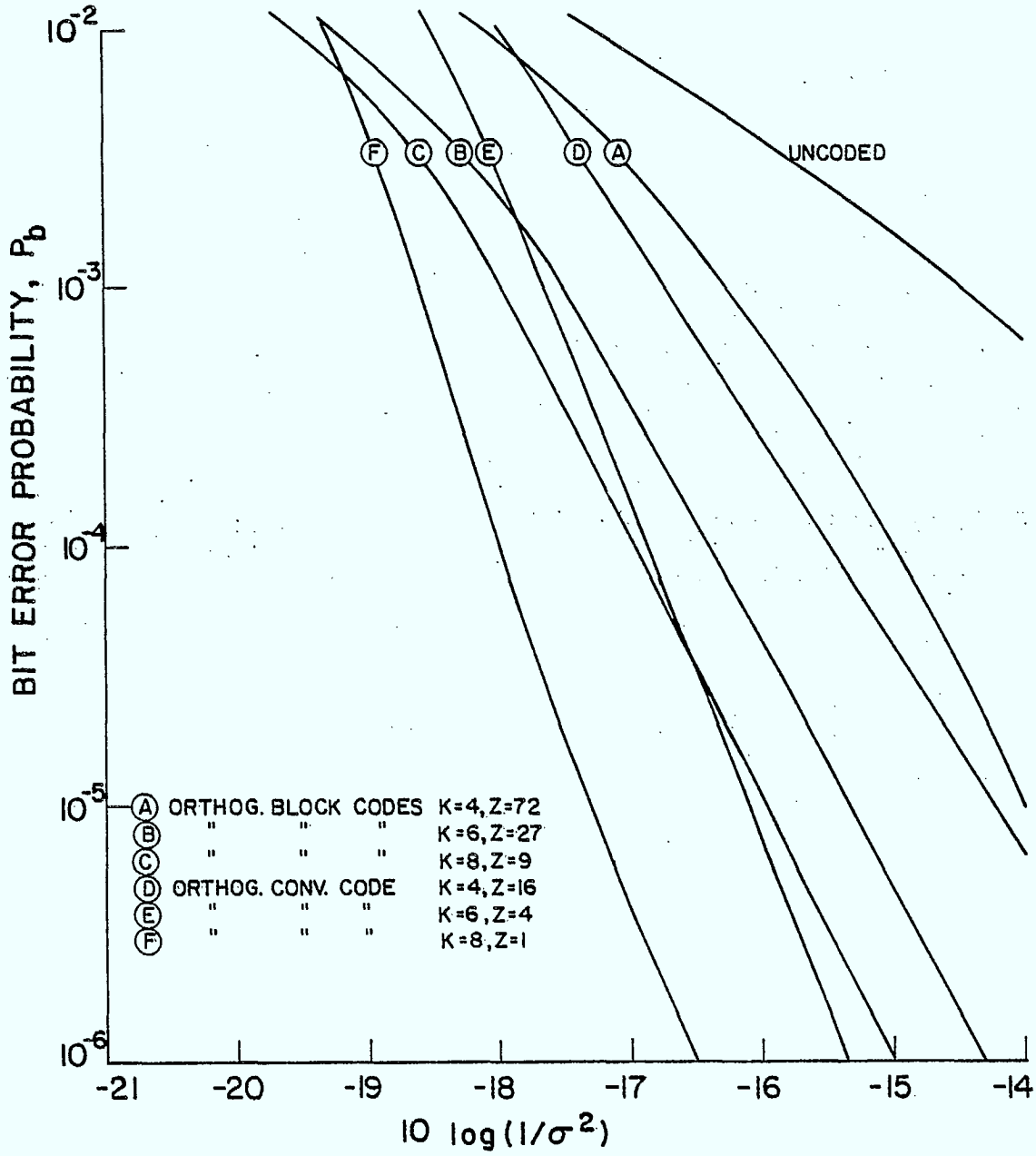


Figure 3.2 Bit Error Probability

Analysis. The analysis of this scheme is straightforward using the techniques of Viterbi [2],[3]. The method is given here for future reference.

Let T(L,N) be the generating function of path lengths and number of input ones:

$$T(L,N) = \frac{NL^K(1-L)}{1-L(1+N)+NL^K}$$

A typical term $t_{\ell n}L^{\ell}N^n$ means there are $t_{\ell n}$ paths of length ℓ caused by n input ones. Consider a path of length ℓ (i.e. ℓ branches away from the all zeros branch) and assume the all zeros branch has $N = 2^K$ (+1)'s on it. Each other branch has precisely $N/2=2^{K-1}$ (-1)'s on it. The probability that such an incorrect path will be chosen over the correct (all zeros) path, is then given by

$$P_\ell = \Pr\left(\sum_{j=1}^{\ell 2^{K-1}} y_j < 0\right) = \Pr\left(\begin{array}{l}\text{path increment along an incorrect path of length } \ell \text{ exceeds that along the correct path}\end{array}\right)$$

$$= Q\left(\sqrt{\frac{\ell 2^{K-1}}{\sigma^2}}\right) \le \exp\left(\frac{-j2^{K-2}}{\sigma^2}\right) \quad Q\left(\sqrt{\frac{d.2^{K-1}}{\sigma^2}}\right) \quad \ell-d=j$$

d = min. dist. of code
= number of branches away from the all zero branch.

Using the arguments of Viterbi [2],[3] the bit error probability is derived as follows:

P_b = bit error probability

$$< \frac{d}{dN} T(L,N)\Big|_{N=1}, \text{ replace } P_\ell \text{ for } L^\ell$$

$$= \frac{L^K(1-L)^2}{(1-2L+L^K)^2}\Bigg|_{\text{subs.}} = \sum_{k=d}^{\infty} C_k P_k$$

$$\leq \sum_{k=d}^{\infty} C_k \, Q\left(\sqrt{\frac{d2^{K-1}}{\sigma^2}}\right) \exp\left(-\frac{(k-d)2^{K-2}}{\sigma^2}\right)$$

$$= Q\left(\sqrt{\frac{d2^{K-1}}{\sigma^2}}\right) \exp\left(\frac{d2^{K-2}}{\sigma^2}\right) \sum_{k=d}^{\infty} C_k \exp\left(-\frac{k2^{K-2}}{\sigma^2}\right).$$

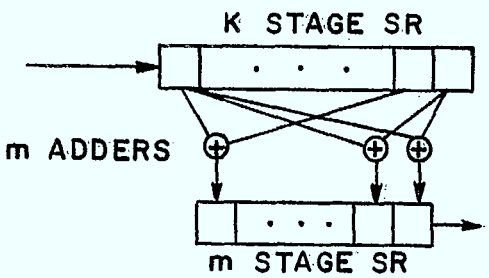
$$= Q\left(\sqrt{\frac{d2^{K-1}}{\sigma^2}}\right) \exp\left(\frac{d2^{K-2}}{\sigma^2}\right) \frac{dT(L,N)}{dN}\Bigg|_{N=1,\ L=\exp\left(-\frac{2^{K-2}}{\sigma^2}\right)}$$

$$P_b \leq Q\left(\sqrt{\frac{d2^{K-1}}{\sigma^2}}\right) \exp\left(\frac{d2^{K-2}}{\sigma^2}\right) \frac{\exp\left(-\frac{K2^{K-2}}{\sigma^2}\right)\left\{1-\exp\left(-\frac{2^{K-2}}{\sigma^2}\right)\right\}^2}{\left[1-2\exp\left(-\frac{2^{K-2}}{\sigma^2}\right)+\exp\left(-\frac{K2^{K-2}}{\sigma^2}\right)\right]^2}$$

This upper bound is plotted in figure 3.2 for the values of K and Z stated i.e. P_b is plotted vs $10\log_{10}(1/\sigma^2)$ with the values σ^2/Z shown.

III. M-ary Orthogonal Signalling.

Code description. This is a variant of the orthogonal convolutional code. Bits are shifted into the K stage shift register, 1 bit at a time. For each K bit in the register, m bits are formed by linear operations on the K bits and these m bits used to select one of 2^m orthogonal signals.



Analysis. The performance of this class of codes depends on the distance structure of the trellis formed by the tap connections. The objective is to maximize the number of levels in the trellis over which

any path is unmerged with the all zeros path, rather than in maximizing d_{free} as is usual. Relatively little work has been done on this problem. Two codes, each with K=7, are listed in Table B-6 in Clark and Cain [4]

For code 1, m=2 and, with the tap connections listed

$$P_b(K,Z) < 7P_7 + 39P_8 + 134P_9 + 808P_{10} \tilde{+} 1000P_{11}$$

where K=7, Z=72 and P_k is as in section II.

For code 2, m=3 and, with the tap connections listed

$$P_b(K,Z) < P_7 + 4P_8 + 8P_9 + 66P_{10} \tilde{+} 600P_{11}$$

where K=7, Z=36.

The performance of this class is expected to be between that of the orthogonal convolutional and pure convolutional. It was not plotted.

IV. Concatenated Reed Solomon-Orthogonal Codes.

Code Description. The outer code is a Reed-Solomon RS(n,k) code and the inner code is an orthogonal code. The outer code is defined over $GF(2^K)$. It forms blocks of K-bits into symbols over $GF(2^K)$ and maps k such symbols into n symbols over $GF(2^K)$. The inner orthogonal code maps each K-bit symbol into one of 2^K orthogonal signals.

Analysis. The overall code rate is $\frac{k}{n} \cdot \frac{K}{2^K} \cdot \frac{1}{Z}$. The word error probability of the block orthogonal code is $P_S = P_E(K,Z)$. The Reed-Solomon decoder will be assumed to be an errors only algorithm. With a symbol error probability of P_S the bit error probability of the output of the RS decoder is given by (Berlekamp [1])

$$P_b = \sum_{j=d}^{n} \frac{j}{2(n+1)} \binom{n}{j} P_s^j (1-P_s)^{n-j}, \qquad P_s = P_E(K,Z).$$

A slightly more pessimistic bound for the bit error probability is given by (Cain and Clark [4])

$$P_b < \frac{2^{K-1}}{2^K-1} \sum_{j=t+1}^{n} \frac{(j+t)}{n} \binom{n}{j} P_s^j (1-P_s)^{n-j} \qquad d=2t+1$$

The Berlekamp expression is evaluated and shown in figure 3.3 for the following sets of parameters:

Inner Code Parameters			Outer Code Parameters			Diversity	Overall Code Rate
K	2^K	Rate	n	k	Rate		
6	64	3/32	32	16	1/2	12	1/256
6	64	3/32	64	32	1/2	12	1/256
6	64	3/32	64	16	1/4	6	1/256
8	256	1/32	100	50	1/2	4	1/256
8	256	1/32	100	25	1/4	2	1/256

V. Pure Convolutional Code.

Code Description. Standard convolutional coder. Unfortunately constructions for very low rate convolutional codes do not appear to be available and only rate 1/2 and rate 1/3, as given in table B-1 of Clark and Cain [4] are considered here.

Analysis. Again using the analysis of Viterbi [3]

$$P_b < Q\left(\sqrt{\frac{d}{\sigma^2}}\right) \exp\left(\frac{d}{2\sigma^2}\right) \left.\frac{dT(D,N)}{dN}\right|_{N=1,\ D=\exp\left(-\frac{1}{2\sigma^2}\right)} = \sum_{k=d}^{\infty} C_k P_k$$

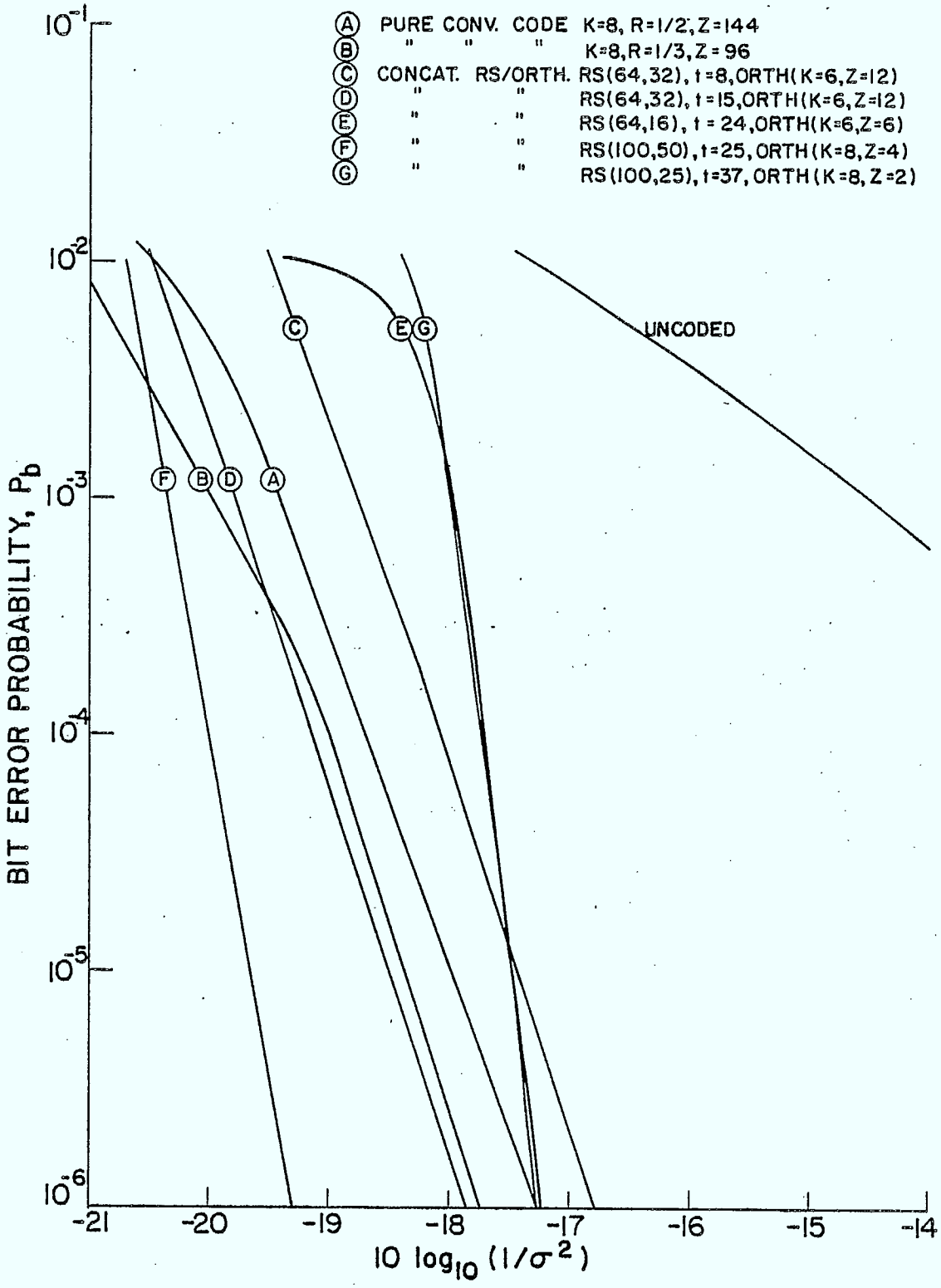


Figure 3.3 Comparison of Bit Error Probability

where $P_k = Q(\sqrt{\frac{k}{\sigma^2}})$ and C_k = total information weight of all paths of weight k merging with the all zeros path. i.e.

$$T(D,N) = \ldots + t_{ij} D^i N^j + \ldots$$

t_{ij} = number of paths at distance i from the all zero path (from divergence to reemergence) caused by j input data 1's.

$$\frac{d}{dN} T(D,N)\Big|_{N=1} = \ldots + jt_{ij} D^i + \ldots$$

and

$C_k = \sum_j jt_{kj}$ = total information weight of all merging paths of weight k.

Code 1: K=8, Rate = 1/3. (Clark & Cain, table B-1, p.402).

$$P_b < P_{16} + 24P_{18} + 113P_{20} \quad , Z=96$$

Code 2: K=8, Rate = 1/2 (Clark & Cain, table B-2, p.402)

$$P_b < 2P_{10} + 22P_{11} + 60P_{12} + 148P_{13} + 340P_{14}, \quad Z=144.$$

(The effective noise variance to be used is σ^2/Z). These upper bounds are shown in figure 3.3.

VI. Cascaded Convolutional Codes.

Code description. It has been mentioned there is a lack of information on the structure of very low rate convolutional codes. Generally the code performance grows with d_{free} which is a function of constraint length. Unfortunately the decoder complexity grows exponentially with constraint length. The possibility of cascading convolutional codes is considered here, as shown in the diagram.

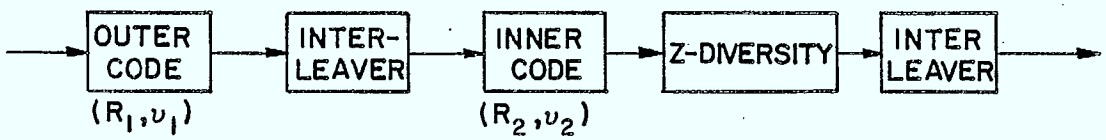


Figure 3.4

The purpose of the inner interleaver is to make the channel appear random to the inner code decoder. The implication here is that adjacent chips on the channel must be separated sufficiently to achieve this. Some comments on interleaving will be given later.

The inner code decoder, if operating effectively, makes errors in bursts when it makes errors. The interleaver between the two coders is there to distribute these errors sufficiently that the outer code decoder can handle them. This requires less complexity than the outer interleaver.

Code analysis. This system can be analyzed by standard methods used in previous sections. The diversity is used to bring the overall rate up to 1/288. The analysis was done and results plotted using convolutional codes of rates 1/2 and 1/3. The problem encountered is that the performance bounds are upper bounds and are very steep with respect to signal to noise ratio. Thus using the upper bound to the probability of bit error of the output of the inner code decoder as the crossover probability of a binary symmetric channel which the outer code decoder sees, because this probability of error is slightly above the actual probability of error (figures of 1/2 to 1 db are often quoted in the literature) and because of the steepness of the outer code performance curve, the overall performance calculated is relatively poor. These results are not given here, being of little interest. Nonetheless there

is little doubt this system could be quite effective for the situation under study.

The values of $10\log_{10}(\frac{1}{\sigma^2})$ required to obtain bit error probabilities of 10^{-3} and 10^{-5} for the various coding schemes are tabulated in table 3.1.

		PROBABILITY OF BIT ERROR.	
		10^{-3}	10^{-5}
ORTHOGONAL BLOCK CODES.	K=4, Z=72	-16.3	-14.0
	K=6, Z=27	-17.5	-15.3
	K=8, Z=9	-18.0	-16.1
ORTHOGONAL CONVOLUTIONAL CODE.	K=4, Z=16	-16.8	-14.3
	K=6, Z=4	-17.7	-16.1
	K=8, Z=1	-18.7	-17.3
CONCATENATED REED SOLOMON/ ORTHOGONAL.	RS=(32,16), K=6, Z=12	-18.7	-17.5
	RS=(64,32), K=6, Z-12	-19.8	-18.5
	RS=(64,16), K=6, Z=6	-18.0	-17.4
	RS=(100,50), K=8, Z=4	-20.4	-19.7
	RS=(100,25), K=8, Z=2	-18.0	-17.5
PURE CONVOLUTIONAL CODE.	K=8, Rate 1/2, Z=144	-19.5	-18.0
	K=8, Rate 1/3, Z=96	-20.0	-18.4
	K=7, Rate 1/3, Z=96	-19.7	
	K=6, Rate 1/3, Z=96	-19.2	
UNCODED.		-14.5	-12.0

Table 3.1. Values of $10\log_{10}(\frac{1}{\sigma^2})$ to obtain probability of bit error of 10^{-3}, 10^{-5}.

3.3. A Comparison of the Coding Complexity.

The simplest system to implement, even with K=8 (256 orthogonal signals) would probably be the block orthogonal. The pure convolutional system would be quite straight forward to implement, using the Viterbi algorithm. The cascaded convolutional coding system would require two such decoders but their constraint lengths might be shorter to obtain the same performance and hence may not be twice as complex as the pure convolutional code. The orthogonal convolutional code requires both an orthogonal code decoder and a Viterbi decoder to implement. Similar comments apply to the M-ary orthogonal convolutional code. The concatenated Reed-Solomon/Orthogonal coding system is perhaps the most complicated, requiring both an orthogonal code decoder and a complex Reed-Solomon decoder. Even if this were made an erasure based decoder it would in all probability require considerable effort to implement effectively.

From the results presented here it appears that the pure convolutional code offers the best compromise between cost-complexity and performance. It should be noted that all systems could be fine-tuned and perhaps better performance obtained. Viterbi decoding is straight forward to implement and for relatively low rates much of it could be implemented in microprocessor software. Other systems, particularly the Reed-Solomon decoder, would require a major effort in hardware design. For the small gains in performance over the pure convolutional code case (with the assumptions made here) the effort is not felt to be cost effective at this time. It is recommended however that some thought be given to the design and construction of a Reed-Solomon decoder as a good investment for future applications.

3.4. Coding Recommendations.

Certain considerations in choosing the rate and constraint length of a convolutional code are discussed here. It will be argued that for the application of interest there is little point in using a code of rate much less than about 1/3 and that the constraint length should be chosen as large as possible consistent with the complexity and time constraints on the decoder.

The bit error probability of a convolutional code on the AWGN channel ($N_o/2$, double sided), is of the form

$$P_b < \sum_{k=d_f}^{\infty} C_k P_k \qquad (3.1)$$

where d_f is the free distance of the code, P_k is given by

$$P_k = Q\left(\sqrt{\frac{2kE_s}{N_o}}\right), \qquad E_s = \text{transmitted energy per coded symbol.}$$

$$Q = \text{complementary error function.}$$

and C_k is the total number of input ones on all paths of weight k merging with the all zeros state at a given point in the trellis. The importance of having a convolutional code with a large d_f is thus clear, since Q(x) is monotonically decreasing with x. The problem of constructing such codes is well investigated and the important point to note here is that, for the same constraint length and code rate, nonsystematic codes have larger free distances than the corresponding systematic code. This is not the case for block codes where the two are equivalent.

The search for good convolutional codes is done largely by computer. For rate R = 1/n codes a tight upper bound on d_f due to Heller (as given in [5],[6]) is

$$d_f \leq \min_{\ell \geq 1} \left[\frac{n2^{\ell-1}}{2^{\ell}-1} (K+\ell-1) \right] \tag{3.2}$$

for constraint length K. In fact, for short constraint lengths ($K \leq 10$) and rates $R = 1/n$, $n \leq 8$, codes have been constructed [6] which meet this bound although only a few, or none, of the parameters C_k are known. In cases where the parameters C_k are not known only the lead term can be used to give an approximation of code performance. To give an indication of how this lead term varies as a function of code rate $R = 1/n$ for fixed constraint length K, we assume a code meeting the Heller bound (3.2) can be constructed for each n. The square of argument of the Q function can be expressed as

$$\frac{2d_f}{N_o}\frac{E_b}{n} = \frac{d_f}{n} \cdot C, \qquad C = \frac{2E_b}{N_o} = \text{constant}.$$

Thus, if other aspects of the codes considered (i.e. the parameters C_k for the various codes) are equal performance is largely determined by d_f/n. For K=8 we have the following:

n	d_f/n
2	5
3	5 1/3
4	5 1/2
5	5 3/5
6	5 2/3
7	5 5/7

Decreasing the rate of the convolutional code much below R = 1/3 or 1/4 is probability not effective for the marginal improvement in performance. It appears that, for this reason, convolutional codes of rates below 1/4 are seldom used on the AWGN, although the price to pay is fairly small - a linear increase in the decoder memory requirements.

It has been stated [7] that increasing the constraint length by 1 will give about a .4 db improvement in performance at the cost of doubling the decoder complexity. This effect was observed in the figures given later. It is also reported that the bound in (3.1) is typically about 1 db pessimistic from the performance observed by simulation or measurements.

Little attention was given to sequential decoding in this work. The error curves for these decoders are steeper than for Viterbi decoders. For a 10^{-5} error probability the performance of a rate 1/2 Viterbi decoder, K=5 to 7 is comparable to a K=41, rate 1/2 hard decision sequential decoder. At lower error rates the sequential decoder is more attractive. (Note that for a sequential decoder complexity increases linearly with K as opposed to exponentially for Viberbi decoders). There is a 2db advantage in using soft decisions over hard decisions and this advantage should be utilized, particularly for the data rates contemplated. This is difficult to do in the implementation of sequential decoders but relatively simple for Viterbi decoders.

A summary of the probability of error performance for various rate codes is given in table 3.2. The rate 1/2 and 1/3 codes come from appendix B in [4]. The rate 1/4 code is from [5] and the rate 1/6 is from [6].

A small study was done on the K=8, R=1/2, 1/3 codes to determine the effect of using only the 1st term for P_e. In each case it amounted to about 1db at $P_e=10^{-5}$, but this figure is given as an indication only. The implication is that the bounds for the rate 1/4 and 1/6 codes, where no information on the path parameters other than d_f are available, are about 1 db too optimistic.

(Rate, Z)	K	$10\log_{10}(1/\sigma^2)$ at P_b 10^{-3}	10^{-5}	
	3	-17.7	-15.7	
	4	-18.05	-16.2	
(1/2, 144)	5	-18.4	-16.6	
	6	-18.65	-16.95	
	7	-19.0	-17.45	
	8	-19.35	-17.8	
	9	-19.55	-18.1	
	3	-1.25	-	
(1/2, 3)	4	-1.60	-	
	5	-1.90	-	
	6	-2.15	-0.2	
	7	-2.45	-0.60	
	8	-2.90	-1.00	
	9	-3.0	-1.3	
	3	-17.55	-15.65	
(1/3, 96)	4	-18.40	-16.40	
	5	-18.85	-16.90	
	6	-19.20	-17.40	
	7	-19.70	-17.85	
	8	-19.95	-18.25	
	3	-0.8	-	
(1/3, 2)	4	-1.6	-	
	5	-2.05	-0.15	
	6	-2.40	-0.6	
	7	-2.9	-1.05	
	8	-3.15	-1.45	
(1/4, 72)	8*	-22.2	-19.4	d_f=22
	8*	-21.3	-18.5	(SYSTEMATIC, d_f=18)
(1/6, 48)	8*	-22.3	-19.5	d_f=34 (SAME PERF. AS K=8, R=1/8, d_f=45 Z=36).
(1/6, 1)	8*	-5.5	-2.7	(d_f=34)

*FIRST TERM ONLY.

Table 3.2. Values of $10\log_{10}(\frac{1}{\sigma^2})$ to obtain probability of bit error of 10^{-3}, 10^{-5}.

From the above comments it is suggested that a rate 1/3 code of constraint length 5,6,7 or 8 is used. The generators for the codes are given in [4], Appendix B. It is anticipated that most of the algorithm will be in software. A constraint length 5 should be considered first to see if it can accommodate the desired rate. If not, some of the functions may have to be transferred to hardware. If it can, an estimate can be made as to the largest constraint length the implementation can handle.

In cases where the system is to support more than one data rate, it may be advantageous to use the same convolutional code and merely change the amount of diversity, rather than using a very low rate code for the lower data rate. Some figures are tabulated in table 3.2 where the overall rates are 1/6 and 1/288 respectively. The figures for the rate 1/6 code, using the lead term of (1) only with coefficient unity, is optimistic and the differences between say a rate 1/2, Z=3 and rate 1/6, Z=1, is not as great as it appears.

3.5. Interleaver Considerations.

The comments of this section are interpretations of section 8.3 in [4], adapted to the problem of interest.

An ordinary periodic block or convolutional interleaver is unsuitable for an AJ situation since periodicities in the interleaver output can be exploited by the jammer. The following pseudo-random interleaver would seem to have the advantages of being simple to implement in software and effective both in distributing burst errors and providing a measure of security against simple jamming strategies.

Suppose, taking into account the code rate, possible jamming strategies and allowable delay at the various data rates, it is decided to use a RAM capable of holding 2^k bits. The coded bits are read into the memory locations in a sequential fashion, using only 2^k-1 locations and omitting the initial location. A set of primitive polynomials of degree k is stored in a ROM. One of these is chosen as the feedback connections for a linear feedback shift register located in a memory address controller. An initial address load obtained from the PN sequence generator used in the DS is also incorporated to avoid a periodic known data bit.

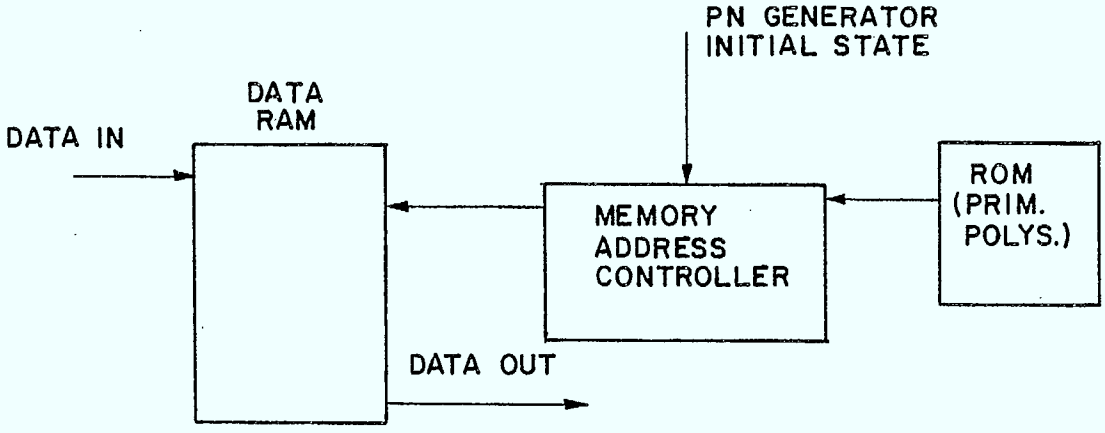


Figure 3.5.

The memory address controller contains a clock which steps the address register (shift register) through its 2^k-1 non-zero states until it returns to the initial state. A new (nonzero) initial state is taken from the PN sequence generator and another primitive polynomial chosen from the ROM, either sequentially or at random, and the process repeats. Dual data RAM's would be used for continuous data flow.

The deinterleaver will require RAM's three times the size of the interleaver RAM's to accommodate soft decision demodulation.

References.

[1] E.R. Berlekamp, The Technology of Error Correcting Codes, Proc. IEEE, V. 68, 1980, pp.564-593.

[2] A.J. Viterbi, Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Trans, Inf. Th., V-13, 1967, 260-269.

[3] A.J. Viterbi, Convolutional Codes and their Performance in Communication Systems, IEEE Trans. Comm. Tech., V-19, 1971, pp. 751-772.

[4] G.C. Clark Jr. and J.B. Cain, Error Correction Coding for Digital Communications, Plenum Press, New York, 1981.

[5] K.J. Larsen, Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3 and 1/4, IEEE Trans. Inf. Th., May, 1973, 371-372.

[6] D.G. Daut, J.W. Modestino and L.D. Wismer, New Short Constraint Length Convolutional Code Constructions for Selected Rational Rates, to appear.

[7] J.A. Heller and I.M. Jacobs, Viterbi Decoding for Satellite and Space Communication, IEEE Trans. Comm. Tech., October 1971, 835-848.

4. Modulation and Coding for Digital Communication over An Interference Channel

4.1 Digital Communication over an Additive White Gaussian Channel.

Forward error correction codes can be used quite effectively against additive white Gaussian noise. A salient feature of white Gaussian noise is that it is memoryless so that the joint probability density function of an n-dimensional Gaussian vector has a product form. For any memoryless discrete channel the minimum attainable error probability is bounded above by an expression of the form [1]

$$P(\varepsilon) \leq 2^{-N\,E(R)} \tag{4.1}$$

in which N is the number of times that the channel is used in the transmission of a code word (or a signal vector), R is the information rate in bits per channel use, and E(R) is a reliability function. Provided E(R)>0, the probability of error, P(ε), can be made as small as we wish by increasing N indefinitely. (N is also the constraint length in sequential decoding so that the decoding complexity increases with N). A typical reliability function E(R) is sketched in Fig. 4.1, where C denotes the channel capacity, R_c is the critical rate, and R_o is the zero-rate intercept of the straight line tangent to the E(R) at $R=R_c$. E(R) is a convex downward function of R

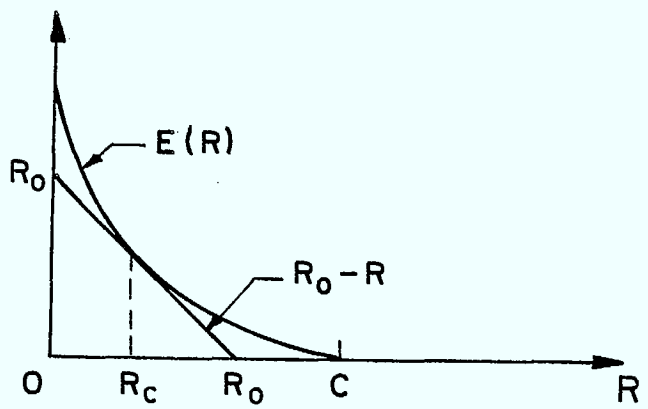


Figure 4.1

with slope -1 at $R=R_c$. Hence

$$E(R) \geq R_o - R \tag{4.2}$$

and the probability of error can be upper bounded by

$$P(\varepsilon) \leq 2^{-N(R_o-R)} \tag{4.3}$$

R_o, which is also R_{comp}, the computational cutoff rate in sequential decoding, is the rate beyond which the average number of computations required per information bit becomes infinitely large.

From the digital encoding and decoding point of view, the channel is comprised of the modulator, transmission link and the demodulator in tandem (the subsystem inside the dotted box of Fig. 4,2) and is characterized by R_o. If $\{\underline{s}_k\}_{k=1}^{M}$ is a set of M orthogonal (or simplex) signals, the channel will consist of M discrete inputs and M continuous outputs $\underline{y} = (y_1, y_2, \ldots, y_M)$. The unquantized R_o is given by

$$R_o = \max_{\{P_k\}} \left\{ -\log_2 \int_{-\infty}^{\infty} \left[\sum_{k=1}^{M} P_k \sqrt{P(\underline{y}|k)} \right]^2 d\underline{y} \right\} \tag{4.4}$$

where $\{P_k\}$, $1 \leq k \leq M$ is the probability distribution of the transmitted signal and $P(\underline{y}|k)$ is the conditional probability of the output $\underline{y}$ given that the k^{th} signal was sent.

For a discrete memoryless channel with M inputs and J outputs,

$$R_o = \max_{\{P_k\}} \left\{ -\log_2 \sum_{j=1}^{J} \left[\sum_{k=1}^{M} P_k \sqrt{P(j|k)} \right]^2 \right\} \tag{4.5}$$

As mentioned above, provided that $E(R) > 0$, $P(\varepsilon)$ can be made as small as we wish by increasing N. It is observed that in the region $R_o<R<C$, E(R) remains positive, but the encoding constraint length needs to be made infinitely large for $P(\varepsilon)$ to approach zero. The region $R_o<R<C$

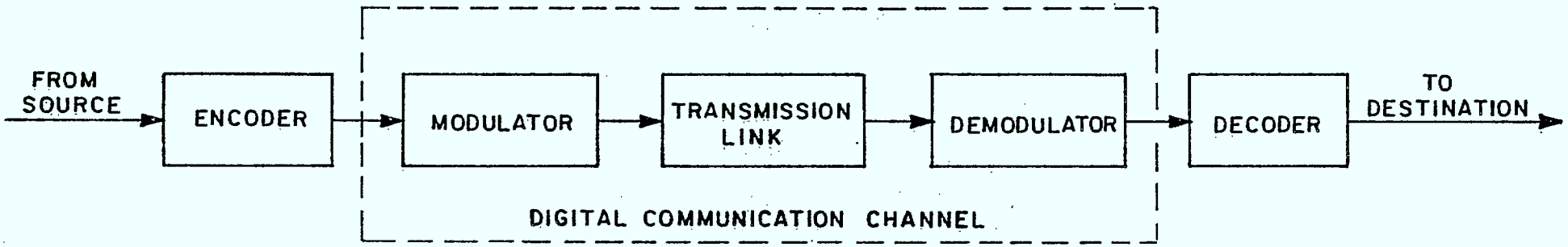


Figure 4.2 Functional Block Diagram of a Digital Communication System

is therefore of no practical significance and, for all intents and purposes, the usuable rate is $0<R\leq R_o$.

Suppose we desire to send one of M messages. We can select M sequences $\{ \underline{s}_k \}$, k=1, 2, ...,M for transmission over the discrete memoryless channel. The information rate R in bits per channel use is then given by

$$R = \frac{1}{N} \log_2 M \tag{4.6}$$

Substituting (4.6) in (4.3), we have

$$P(\varepsilon) \leq M2^{-NR_o} \tag{4.7}$$

which is the union bound.

For a given N, $P(\varepsilon)$ is minimized if E(R) is maximized. From the information transfer point of view it is desired to simultaneously maximize E(R) and R. From Fig. 4.1 it is observed that the best operating point is at $R=R_c$. In the neighbourhood $R \approx R_c$, E(R) can be maximized by maximizing R_o.

The maximization in (4.4) and (4.5) for the Gaussian channel is achieved when $P_k=\frac{1}{M}$, k=1, 2, ...,M and, using the product form of the joint Gaussian pdf, the unquantized R_o simplifies to [2]

$$R_o = \log_2 \frac{M}{1+(M-1)\left[\int_{-\infty}^{\infty} \sqrt{P_{s+n}(y)\ P_n(y)}\ dy\right]}$$

$$= \log_2 M - \log_2 \left[1+(M-1)\left[\int_{-\infty}^{\infty} \sqrt{P_{s+n}(y)\ P_n(y)}\ dy\right]\right] \tag{4.8}$$

where $P_{s+n}(y)$ and $P_n(y)$ are the conditional probabilities of y given the signal present and the signal absent, respectively. Define

$$D = \int_{-\infty}^{\infty} \sqrt{P_{s+n}(y)\ P_n(y)}\ dy \tag{4.9}$$

Then

$$R_o = \log_2 M - \log_2\ [1 + (M-1)D] \tag{4.10}$$

Equ (4.10) is the generalized expression for R_o, derived by Omura and Levitt [10] using a different argument. Since D is a non-negative quantity and $\log_2[1+(M-1)D] \geq 0$, maximizing R_o is equivalent to minimizing D.

For a phase coherent zero mean Gaussian channel with two-sided power spectral density $N_o/2$ watts/Hz,

$$D = \begin{cases} \exp\ (-E_c/2N_o) & \text{for orthogonal signals} \\ \exp\ (-E_c/N_o) & \text{for simplex signals} \end{cases} \tag{4.11}$$

where $E_c = ST$ is the signal energy per T seconds and S is the average signal power. Then

$$R_o = \begin{cases} \log_2 M - \log_2[1+(M-1)\,\mathrm{Exp}(-E_c/2N_o)], & \text{orthogonal signals} \\ \log_2 M - \log_2[1+(M-1)\ \mathrm{Exp}(-E_c/N_o)], & \text{simplex signals} \end{cases} \tag{4.12}$$

To facilitate digital processing at the receiver it is necessary to quantize the received signal to a finite number of levels. Intuitively, R_o is influenced by the effect of quantization and increases with the number of quantizing levels used. Massey [3] has presented numerical results which confirm this intuition.

Let the number of output levels be $J=M^{\ell}$, ℓ an integer. Then $\ell=1$ corresponds to "hard decision" and $\ell>1$ corresponds to varying degrees of "soft decision". From (4.5), we have

$$R_o = \max_{\{P_k\}} \left\{ -\log_2 \sum_{j=1}^{M^\ell} \left[\sum_{k=1}^{M} P_k \sqrt{P(j|k)} \right]^2 \right\}$$

For hard decision, i.e., $\ell=1$, R_o becomes [1]

$$R_o = -2\log_2 \left[\sqrt{p/M} + \sqrt{q\tfrac{M-1}{M}} \right] \tag{4.13}$$

where $q=1-p$ is the probability that any particular hard decision is correct.

With $\log_2 J = \ell \log_2 M$ Wozencraft and Kennedy [1] have considered retaining an ordered list of the ℓ a posteriori most probable signals, resulting in an $\ell \log_2 M$ bit number as the quantizer output and shown that, for a list-of-ℓ channel,

$$R_o = -2 \log_2 \left(\frac{1}{\sqrt{M}} \sum_{j=1}^{\ell} \sqrt{\alpha_j} + \sqrt{\frac{M-1}{M}} \sqrt{1 - \sum_{j=1}^{\ell} \alpha_j} \right)$$

where

$$\alpha_j = \binom{M-1}{j-1} \int_{-\infty}^{\infty} P_{s+n}(x)\,dx \left[\int_{-\infty}^{x} P_n(y)\,dy \right]^{M-j} \left[\int_{x}^{\infty} P_n(y)\,dy \right]^{j-1}$$

is the probability of the correct signal occupying position j in the list.

For binary signalling, the unquantized R_o is, from (4.12), given by

$$R_o = \begin{cases} 1-\log_2 [1+\exp(-E_c/2N_o)], & \text{orthogonol signals} \\ 1-\log_2 [1+\exp(-E_c/N_o)], & \text{antipodal signals} \end{cases} \tag{4.14}$$

The hard-decisioned R_o is, from (4.13), given by

$$R_o = 1-\log_2 \left[1+2\sqrt{P(1-P)} \right] \tag{4.15}$$

where

$$P = \begin{cases} \displaystyle\int_{\sqrt{E_c/N_o}}^{\infty} \frac{1}{2\pi} e^{-x^2/2}\,dx, & \text{orthogonol signals} \\ \displaystyle\int_{\sqrt{2E_c/N_o}}^{\infty} \frac{1}{2\pi} e^{-x^2/2}\,dx, & \text{antipodal signals} \end{cases} \tag{4.16}$$

The unquantized and hard-decisioned R_o's for binary signalling are plotted in Fig. 4.3

4.2 Communication over a Partial-Band Jamming Channel.

Assuming ideal interleaving and deinterleaving, an interference channel, as viewed by the encoder/decoder pair, can be assumed to be memoryless. Then the interference channel problem can be treated as in the additive white Gaussian case. The probability of error is again bounded from above by the union bound:

$$P(\varepsilon) \leq M2^{-NR_o}$$

The cutoff rate R_o is some function of the encoded symbol energy to noise ratio:

$$R_o = f(E_c/N_o)$$

and

$$E_c/N_o = r\, E_b/N_o$$

where r is the code rate in bits/channel signal, E_b is the energy per bit, E_c is the encoded symbol energy, and N_o is the jammer power spectral density. Let S be the average signal, R be the data transmission rate in bits/sec, J be the jammer power and W be the equivalent noise bandwidth, then

$$E_b/N_o = \frac{S/R}{J/W}$$

Define the signal processing gain as the bandwidth expansion factor:

$$PG \triangleq \frac{W}{R}$$

Then

$$E_b/N_o = PG\, \frac{1}{J/S}$$

In fact R_o is a function of the modulation, demodulation and quantization processes as well as E_c/N_o. As shown in the previous section R_o is

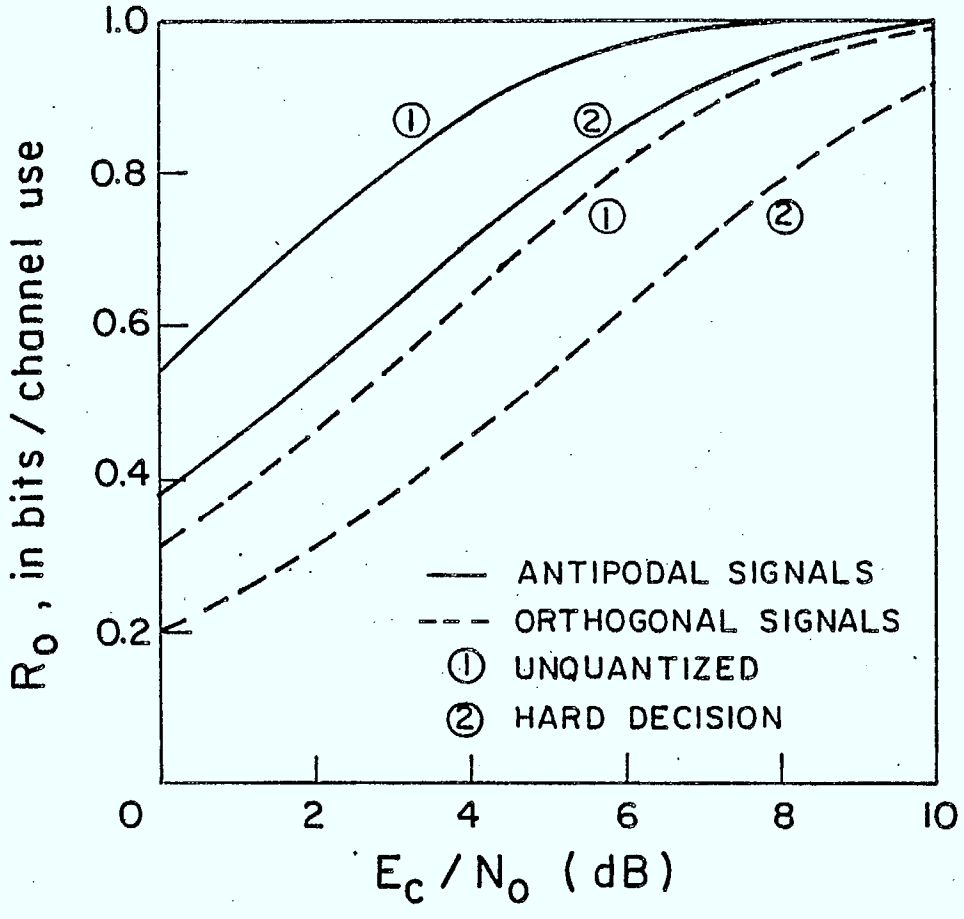


Figure 4.3 Curves of Cutoff Rate for Binary Signalling

expressible as

$$R_o = \log_2 M - \log_2 [1+(M-1)D] \qquad (4.10)$$

where D is a function of modulation, channel characteristic, demodulation and quantization as well as symbol energy-to-noise ratio E_c/N_o. In what follows we focus attention on binary antipodal signalling with optimum matched filter reception under the assumption of ideal interleaving and deinterleaving. Then R_o becomes

$$R_o = 1-\log_2 [1+D] \qquad (4.17)$$

Under a frequency hopping scheme an intelligent jammer will not attempt to jam the whole band. Rather, it will concentrate its jamming power over only a fraction ρ , $0<\rho\leq 1$, of the band. Likewise, in a direct sequence spreading environment the jammer will concentrate his power over a fraction of the signal duration. The fraction ρ is then the duty cycle. These two jamming situations exert similar effects on the transmitted signal. In the discussions to follow we will consider a partial-band jamming situation. Here we model the partial-band jamming by a random variable.

$$z = \begin{cases} 1 & \text{, with probability } \rho \\ 0 & \text{, with probability } (1-\rho) \end{cases}$$

With this partial-band jamming model, the value of D given by (4.11) is appropriately modified to account for knowledge of the jammer state to yield [10]

(i) Unquantized and known jammer state

$$D_1 = \max_{0<\rho\leq 1} \rho\, e^{-\rho(E_c/N_o)}$$

(ii) Hard decision and known jammer state

$$D_2 = \max_{0<\rho\leq 1} 2\rho \sqrt{p\ (1-p)}$$

(iii) Unquantized and unknown jammer

$$D_3 = 1$$

(iv) Hard decision and unknown jammer state

$$D_4 = \max_{0<\rho\leq 1} \sqrt{4\rho p\ (1-\rho p)}$$

where

$$p \triangleq Q\left(\sqrt{\frac{2\rho E_c}{N_o}}\right) = \int_{\sqrt{\frac{2\rho E_c}{N_o}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

Substituting the D_i's in (4.17) we have

$$R_o = 1-\log_2 [1+D_i],\ i=1,2,3,4.$$

It has been shown in [7] that the maximizing ρ is

$$\rho^* = \begin{cases} \dfrac{X_o}{E_c/N_o}, & E_c/N_o > X_o \\ 1, & E_c/N_o \leq X_o \end{cases}$$

where X_o is the solution of the equation

$$Q\ (\sqrt{2X}) - \frac{\sqrt{X}\, e^{-X}}{\sqrt{4\pi}} = 0$$

The value of X_o has been computed in [7] to be X_o=0.709. Thus, an intelligent jammer can optimize his jamming strategy depending on the value of symbol energy-to-jammer noise power spectral density ratio.

4.3 Comments

Optimization of signal transmission over an interference channel is tantamount to a maximization of the cutoff rate R_o or a minimization of the parameter D. We have examined the unquantized and hard decision

performance of communication over an interference channel. In the unquantized case, it has been implicitly assumed that the receiver is a maximum likelihood receiver. With soft decision the assumption of maximum likelihood reception is only suboptimum.

Massey [3] has suggested an approach to optimize the soft decision threshold. It appears that optimum quantization for an interference channel needs further investigation. Also, the selection of R_o is independent of the coder/decoder pair. Intuitively, coupling the code design to the R_o selection can potentially improve the performance of digital communication over an interference channel. The whole area of modelling interference channels and the matching of modulation and coding systems for such channels requires further consideration to improve our understanding. Recent work in this area has indicated considerable improvements may result.

REFERENCES

[1] J.M. Wozencraft and R.S. Kennedy, "Modulation and Demodulation for Probabilistic Coding," IEEE Trans. Infor. Theory, Vol. IT-12, No. 3, pp. 291-297, July 1966.

[2] K.L. Jordan, Jr., "The Performance of Sequential Decoding in Conjunction with Efficient Modulation," IEEE Trans. Commun. Technology, Vol. COM-14, No. 3, pp. 283-297, Juny 1966.

[3] J.L. Massey, "Coding and Modulation in Digital Communications," Zurich Seminar, pp. E2(1)-E2(4), 1974.

[4] S.W. Houston, "Modulation Techniques for Communication, Part I: Time and Noise Jamming Performance of Spread Spectrum M-ary FSK and 2, 4-ary DPSK Waveforms," Proc. National Aerospace Electronics Conf. pp. 51-58, Jan. 1975.

[5] H. Schmidt and P.L. McAdam, "Modulation Techniques for Communication, Part II: Wideband survivable Satellite Communications," Proc. Nat. Aerospace Electronics Conf. pp. 59-65, Jan. 1975.

[6] B.D. Trumpis and P.L. McAdam, "Performance of Convolutional Codes on Burst Noise Channels," Proc. NTC, pp. 36:3-1 to 36:3-14, Dec. 1977.

[7] D.R. Martin and P.L. McAdam, "Convolutional Code Performance with Optimal Jamming," ICC '80, pp. 4.3.1-4.3.7, Seattle, 1980.

[8] R.J. McEliece and W.E. Stark, "An Information Theoretic Study of Communication in the Presence of Jamming," ICC '81, pp. 45.3.1-45.3.5, June 1981.

[9] W.E. Stark and R.J. McEliece, "Capacity and Coding in the Presence of Fading and Jamming," NTC '81, pp. B7.4.1-B7.4.5, New Orleans, Dec. 1981.