INFORMATION TO USERS

This was produced from a copy of a document sent to us for microfilming. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help you understand markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure you of complete continuity.

2. When an image on the film is obliterated with a round black mark it is an indication that the film inspector noticed either blurred copy because of movement during exposure, or duplicate copy. Unless we meant to delete copyrighted materials that should not have been filmed, you will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed the photographer has followed a definite method in "sectioning" the material. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For any illustrations that cannot be reproduced satisfactorily by xerography, photographic prints can be purchased at additional cost and tipped into your xerographic copy. Requests can be made to our Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases we have filmed the best available copy.

University Microfilms International

300 N. ZEEB ROAD, ANN ARBOR, MI 48106
18 BEDFORD ROW, LONDON WC1R 4EJ, ENGLAND

DAVIDOVICI, SORIN

SPREAD SPECTRUM COMMUNICATION SYSTEMS AND DIGITAL ENCODING OF VIDEO SIGNALS

City University of New York PH.D. 1980

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background ✓
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Other _____

University
Microfilms
International

SPREAD SPECTRUM COMMUNICATION SYSTEMS

and

DIGITAL ENCODING OF VIDEO SIGNALS

by

Sorin Davidovici

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1980

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

December 30, 1980
date

Donald L Schilling
Chairman of Examining Committee

December 30, 1980
date

Paul R Kaminski
Executive Officer

Prof. D.L.Schilling (mentor)

Prof. T.Apelewicz

Prof. H.Taub

Prof. G.Eichman

Prof. N.Scheinberg

Dr. J.Garodnick

Supervisory Committee

The City University of New York

Abstract

This dissertation covers two distinct areas: 1)spread spectrum communications and, 2)digital encoding of color video signals using ADM.

In the spread spectrum communications area the the research done addresses itself to the question of optimal PN acquisition performance and to the performance of direct sequence spread spectrum systems in the presence of CW jammers at the carrier frequency.

The PN acquisition research has generated a tight bound to the partial autocorrelation function of a PN sequence. The application of this bound to several PN acquisition methods has made it possible to meaningfully evaluate their performance in terms of the minimum acquisition time required for a given probability of error. In addition, a significant improvement in PN acquisition performance is obtained by applying the concept of recursive estimation. This allows the acquisition process to proceed much faster, as it helps to eliminate most false starting vector choices very rapidly.

The performance of a direct sequence spread spectrum system is evaluated under CW jamming at the carrier frequency. The resulting probability of error is shown to depend on a number of parameters whose effect has not always been well understood. Among others, the results clearly show

the effect of the sequence length, L, as well as the effect of the ratio of the processing gain to the length of the sequence, K/L, on the degree of jammer rejection

The digital encoding of color video signals research attempts to find the limitations of the ADM algorithm in the encoding of NTSC type color video. The approaches taken include composite as well as component coding. The bit rates considered vary from 54MBPS to as low as 8MBPS. All of the above methods and bit rates transmit a good color quality even though there is an increasing general degradation in the picture quality with decreasing bit rates.

The last part of the research concerns itself with the application of ADM to the transmission of packet video. The research has proven the feasibility of such an approach which has a beneficial side effect in the much reduced data set size (2 bits per pixel).

Acknowledgment

I wish to express my gratitude to Prof. Schilling for his interest, encouragement and valuable guidance during the course of this research. The many technical suggestions and his constructive criticism are well appreciated.

I would like to thank Dr. N. Scheinberg for his guidance in building the necessary hardware and Mr. M. Braff and Dr. F. Hemmati for their help with the theoretical aspects of this dissertation.

In addition, I would like to thank to all of my Doctoral Committee members and all my colleagues for their many accomodations extended to me in times of need.

Last but not least, I would like to thank my family for the support they extended to me in good and bad times.

TABLE OF CONTENTS

LIST OF FIGURES

SPREAD SPECTRUM SYSTEMS

DIGITAL ENCODING OF COLOR VIDEO SIGNALS

ACQUISITION OF DIRECT SEQUENCE SPREAD SPECTRUM SYSTEM SIGNALS

Introduction

Spread spectrum systems are designed to permit communications under the difficult condition of very low signal to noise ratios which may be encountered due to high interference (intentional or unintentional) or low signal levels (transmissions with low detectability). The capability of spread spectrum systems to operate at very low signal to noise ratios is achieved by transmitting a signal that is different from all other signals including the interference. Since a signal which lasts T seconds is characterized by 2WT samples, the dimensionality of the signal can be increased by increasing the bandwidth W. Spread spectrum systems take advantage of this and operate over a bandwidth W which is much wider than the bandwidth that would be normally needed to communicate the message signal itself.

Although the bandwidth of the signal can be increased by phase, amplitude or frequency modulation, practical systems favor the discrete phase modulating approach of the PN systems or the discrete frequency hopping approach of FH systems. Figure 1 shows a PN and an FH type system in block diagram.

One of the requirements placed on spread spectrum systems is that the message be decoded only by knowledge of the

specific PN code used to encode it. Even with this knowledge, however, the message will not be decoded unless the local PN code and the transmitted PN code are in synchronism. The acquisition problem is concerned with just this PN code phase synchronization. A number of techniques have evolved which attempt to synchronize PN codes. They trade off acquisition performance versus hardware complexity and, in general, operate in different signal to noise ratio environments. They have one point in common, however, by using the good autocorrelation properties of PN sequences to decide when a state of synchronism has been attained. Since most of the acquisition work done has been based on the properties of PN sequences, it is felt that a brief presentation of these properties prior to the presentation of the results obtained would be helpful.

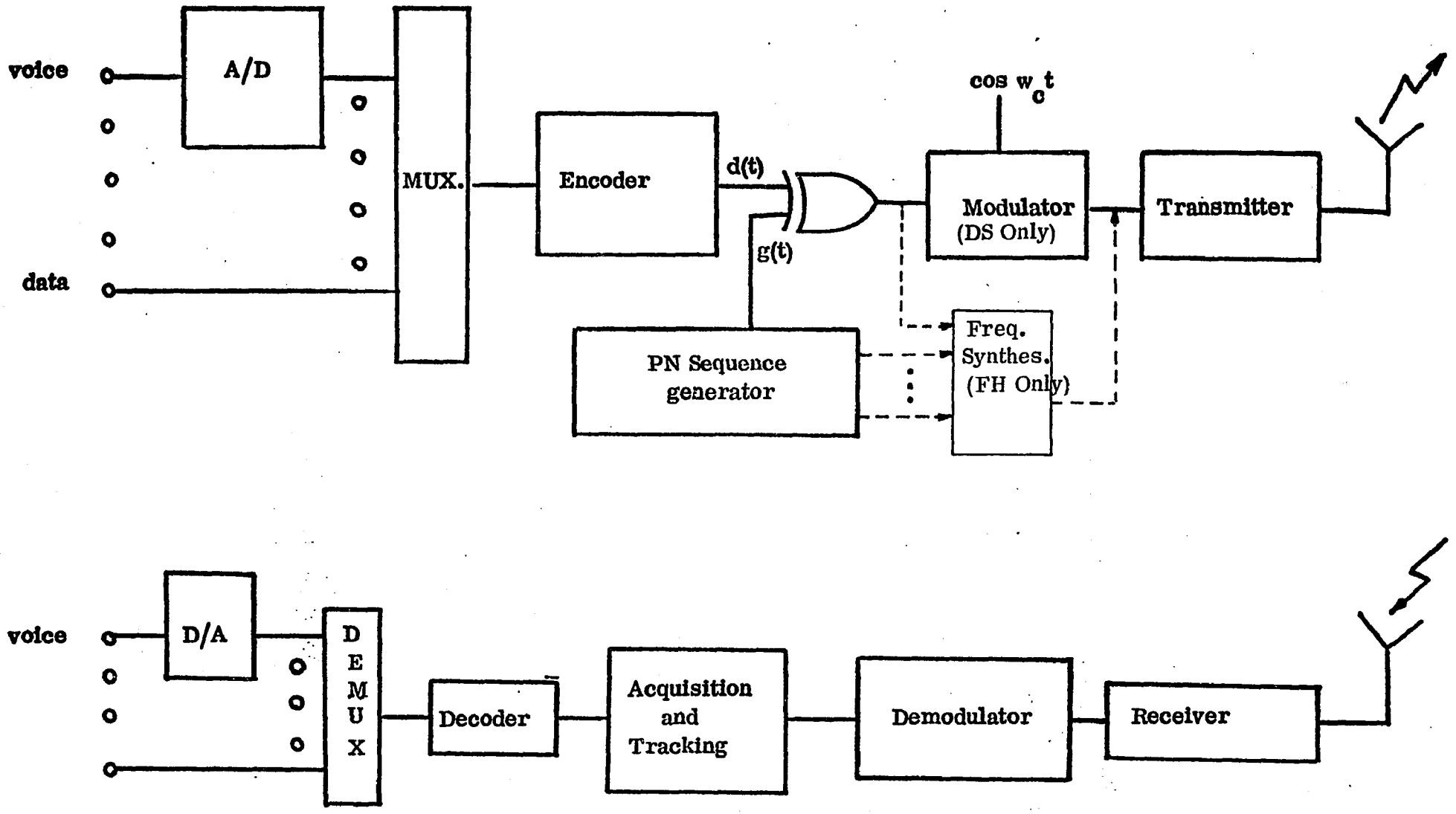


Fig. 1 - Spread Spectrum Communication System

Properties of Pseudo-Noise Sequences

As their name implies, the binary shift register sequences have elements which take one of two values {0,1}. To better understand the properties of different sequences, one should consider the fact that all such sequences are related to a polynomial of the following form:

$$h(x) = h_0 x^n + h_1 x^{n-1} + \ldots + h_{n-1} x + h_n \qquad (1)$$

Since this is a 'binary' polynomial the coefficients take the values {0,1} with the exception of h_0 and h_n which are always 1. It is convenient, whenever one attempts to describe the generating polynomial, h(x), only to give the value of the h_i's in an octal form. For example, let $h = h_0 \ldots h_n$ equal to 100 110 001, where for this example N = 8. The octal value of this polynomial is 461 and this is sufficient to describe it completely. A sequence u is said to be generated by the polynomial h(x) if the following equation holds true:

$$h_0 u_j \oplus h_1 u_{j-1} \oplus \ldots \oplus h_n u_{j-n} = 0 \qquad (2)$$

Equation (2) can be rewritten in a more convenient form by first replacing j by j+n and, second, making use of the fact that $h_0 = 1$. Thus, Eq. (2) becomes

$$u_{j+n} \oplus h_1 u_{j+n-1} \oplus \ldots \oplus h_n u_j = 0 \qquad (3)$$

The operator connecting the terms in Eq. (3) is the addition mod. 2 operator or, as it is also known, the EX-OR operator. Table 1 shows the truth table for mod. 2 addition:

Table 1

Truth Table for Mod. 2 Addition

x	y	x⊕y
0	0	0
0	1	1
1	0	1
1	1	0

This truth table shows that x⊕y could be 0 or 1. The result is not determined by the individual values of x or y but rather by whether x and y are the same or not. If both, x and y, are 0 or 1 the x⊕y will be 0, but when x and y are different then x⊕y will be 1. If x⊕y are 0 only if x=y, then Eq. (3) can be written as:

$$u_{j+n} = h_1 u_{j+n-1} \oplus h_2 u_{j+n-2} \oplus \ldots \oplus h_n u_j \qquad (4)$$

where the subscripts of u indicate integer time delays and the ⊕ operation is performed by an EX-OR gate. This suggests a convenient way to generate such sequences. Fig. 2 shows how by using a shift register with proper feedback we can generate any given sequence.

In Fig. 3 the sequence generated will correspond to the polynomial described by the octal value 461.

It is easy to generate any binary sequence once its generating polynomial h(x) is known, as this specifies the

degree N and the feedback connections. However, it is also possible to find h(x) if a number of elements of the sequence u are known together with the degree N. Assuming one can observe the sequence for 2N-1 different outputs, the following equations can be set up for every new shift u_{j+k}:

$$u_{j+n} \oplus h_1 u_{j+n-1} \oplus h_2 u_{j+n-2} \oplus \ldots \oplus u_j = 0 \qquad (4a)$$

If the observed output is

... 1001110111111 ...

and the degree, N, is known to be seven, then the initial vector is made up of the first seven elements:

1001110 (4b)

which corresponds to $u_{j+7}\ u_{j+6} \ldots u_j$ and the new element, u_{j-1}, is given by:

$$h_7 \oplus h_4 \oplus h_3 \oplus h_2 = 1 \qquad (4c)$$

After the next shift, the new vector becomes

0011101 (4d)

and using Eq. (4a) in a similar fashion one can generate the equation for the next element:

$$h_5 \oplus h_4 \oplus h_3 = 1 \qquad (4e)$$

Proceeding in this manner, N-1 equations can be generated which would specify a unique set of h_i's i = 1, ..., 7 as its solution.

$$
\begin{aligned}
&1)\ h_7 \oplus h_4 \oplus h_3 \oplus h_2 = 1 \\
&2)\ h_5 \oplus h_4 \oplus h_3 \oplus h_1 = 0 \\
&3)\ h_6 \oplus h_5 \oplus h_4 \oplus h_2 = 1 \\
&4)\ h_7 \oplus h_6 \oplus h_5 + h_3 \oplus h_1 = 0 \qquad (4f) \\
&5)\ h_7 \oplus h_6 \oplus h_4 \oplus h_2 = 0 \\
&6)\ h_7 \oplus h_5 \oplus h_3 \oplus h_1 = 0
\end{aligned}
$$

Now, combining 4) and 6) we obtain $h_6 = 0$, while by combining 3) and 5) we obtain $h_5 = 0$ since $h_7 = 1$. Similarly, now we obtain $h_1 = 0$, $h_2 = 0$ and $h_3 = h_4 = 1$, which define the circuit of Fig. 2 .

The linear binary shift register sequences can have any period, L, up to $L = 2^N-1$. It is well known that an N-stage register can have at most 2^N different states. Why, then, are the binary linear sequences limited to at most 2^N-1 different elements? The reason is made rather obvious by looking at Fig. 3 . The next element u_{j+n} is generated by EX-OR logic. The truth table for this operation shows that if the state of the register were to be all 0s, then <u>every</u> new u_j would always be a 0 and the resulting sequence would be all zeros. Because clearly this situation is to be avoided, the maximum period of u is limited to 2^N-1.

If, in fact, the period of u is exactly 2^N-1, then the sequence is called an m, or PN or maximal length sequence and its generating polynomial is called a primitive polynomial. The PN sequences have been studied since the early 1950s and a good amount of information is available on their properties,

some of which are given below.

1) A PN sequence has a period $L = 2^N-1$. This is obviously the case from the preceding argument.

2) There exist 2^N-1 different non-zero sequences, and they are all the different phases of u. In other words, the polynomial h(x) can generate all the different phases of the sequence u. This is clearly the case since the shift register can start with any one of 2^N-1 initial vectors (or initial states) and continue from there on.

3) If one considers two different phases of u, such as $u_{j+\ell}$ and u_j, their modulo 2 addition will generate still another phase of u, u_{j+k}

$$u_{j+k} = u_{j+\ell} \oplus u_j \tag{7}$$

This property, known as the shift and add property, is one of the most important properties of PN sequences. It basically states that element by element mod. 2 addition of two different phases of the sequence u will result in the same sequence u with a different phase. This property is very useful, especially in the acquisition process; it guarantees the good autocorrelation property so useful for this process.

4) The weight of u is equal to 2^{N-1}, where the weight of u is defined as the number of +1s in the sequence u. But if such is the case, and if the period of a PN sequence is 2^N-1, then the number of 0s in the sequence is $2^N-1 - 2^{N-1}-1$. It is relatively easy to prove this property. Looking at an

N-stage shift register (for our example let N = 3) we notice that it can take on any of 2^N different states.

Ex. N = 3

Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

But, as mentioned before, the state 000 is not allowed. Therefore, no matter from which particular output Q the sequence is taken, the number of 1s will be higher by 1 than the number of 0s due precisely to the deletion of the all 0 state. It will be seen that it is this good balance between the ones and zeros which accounts for the good noise-like properties of PN sequences.

5) A PN sequence has a two-valued autocorrelation function:

$$R_{u,u}(\ell) = \begin{cases} L = 2^N - 1 & \text{if } \ell = 0 \\ -1 & \text{if } \ell \neq 0 \end{cases} \tag{8}$$

To prove this property one should consider that in most communication systems it is not desirable to modulate by 1 and 0 but rather by -1 and 1. To obtain the ±1 values from a PN sequence, a transformation is defined as x(1), where

$$x(\alpha) = (-1)^{\alpha} \tag{9}$$

This will replace every +1 by a -1 and every 0 by a +1.

Now:

$$R_{u,u}(\ell) = \sum_{n=0}^{L-1} \chi(u_n)\chi(u_{n+\ell}) \tag{10}$$

Property 3 has shown the EX-OR operation of two phases of the same sequence to result in still a different phase of the same sequence.

$$u_j \oplus u_{j+i} = u_{j+k}.$$

Table 2 shows that

$$\chi(u_n)\chi(u_{n+\ell}) = \chi(u_n \oplus u_{n+\ell}) \tag{11}$$

This implies that multiplying two sequences whose individual elements have been operated on by the $\chi(\cdot)$ operator defined in Eq. (9), element by element and then adding all the results is equivalent to mod. 2 adding the two sequences element by element, operating on every resultant element with the $\chi(\cdot)$ operator and then summing the results.

Table 2

Equivalence of $\chi(x \oplus y)$ and $\chi(x) \times Y(x)$

x	y	x⊕y	$X_{(x)}$	$Y_{(x)}$	$X_{(x \oplus y)}$	$X_{(x)} \times Y_{(x)}$
0	0	0	+1	+1	1	1
0	1	1	+1	-1	-1	-1
1	0	1	-1	+1	-1	-1
1	1	0	-1	-1	1	1

Therefore, using property 3 together with Eq. (11), the summation

of Eq. (10) is reduced to:

$$R_{u,u(\ell)} = \sum_{n=0}^{L-1} \chi(u_n)\ (u_{n+\ell})$$

$$= \sum_{n=0}^{L+1} \chi(u_n \oplus u_{n+\ell}) \qquad (12)$$

$$= \sum_{n=0}^{L-1} (u_{n+k})$$

Property 4 states that the number of 1s in PN sequence u to be 2^{N-1} while the number of 0s in a PN sequence u to be $2^{N-1}-1$. The difference is one more element of value +1 which when transformed by the operator $\chi(\cdot)$ will result in a value of -1. Thus,

$$R_{u,u}(\ell) = \sum_{n=0}^{L-1} \chi(u_n)\chi(u_{n+\ell}) = -1 \text{ if } \ell \neq 0 \qquad (13)$$

If $\ell = 0$ however,

$$R_{u,u}(0) = \sum_{n=0}^{L-1} \chi(u_n)\chi(u_n) = \qquad (14)$$

$$= \sum_{n=0}^{L-1} \chi^2(u_n) = L \quad \text{if } \ell = 0$$

and this proves Eq. (8).

6) All PN sequences have the same run-length statistics.

A "run" is defined as a number of identical symbols which are preceded and proceeded by different symbols. For example, 0 111 0 is a run of three ones while 1000001 is a run of five zeros. All PN sequences have the same number of "runs" of zeros and ones. The relative distribution of these runs is tabulated in Table 3.

Run Length	Symbols	No. of Runs
N	1s	1
N-1	0s	1
N-2	2^0 runs of 1s 2^0 runs of 0s	2^1
N-3	2^1 runs of 1s 2^1 runs of 0s	2^2
.		
.		
.		
N-j	2^{j-2} runs of 1s 2^{j-2} runs of 0s	2^{j-1}
.		
.		
.		
.		
.		
1	2^{N-3} runs of 1s 2^{N-3} runs of 0s	2^{N-2}

Table 3: Run Length Distribution of a PN Sequence ($L = 2^N-1$)

7) The number of maximal-length sequences which can be generated by an M-stage shift register, is:

$$S = \frac{\phi(L)}{N} \leq \frac{L-1}{N} \tag{15}$$

where the equal sign applies if L = prime number and $\phi(L)$ = Euler's number. As an example, take N = 5. Then $L = 2^N-1 = 31$.

Since 31 is a prime number, $S = \frac{30}{5} = 6$. If, however, N = 4, then $L = 2^4-1 = 15$ which is not a prime number. The number of maximal length sequences which can be generated by a 4-stage shift register, is then given by $S < 14/4$ (where the unequal sign applies).

Not all of the generated PN sequences are useful in a communications environment. One half of the generated sequences will be mirror image sequences (i.e. they can be obtained by a "backward" generation of the remaining sequences). As an example, let N = 3 and $L = 2^N-1 = 7$ which is prime. Then $S = \frac{7-1}{3}$ and the two generated sequences are given below.

$$\begin{aligned} &\text{Sequence 1:} \quad \overleftarrow{\ldots 111001011100101110010 \ldots} \\ &\text{Sequence 2:} \quad \ldots 111010011101001110100 \ldots \overrightarrow{} \end{aligned} \tag{16}$$

Listing the sequence on the top in a leftward direction and the sequence on the bottom in a rightward direction one obtains the same identical sequence.

In general, to find a polynomial $h_1(x)$ which generates the reverse PN sequences of polynomial $h_0(x)$, the following procedure can be employed:

If $h_0(x)$ is described by $h_0h_1h_2 \ldots h_n$ the $h_1(x)$ will correspond to $h_nh_{n-1} \ldots h_0$,

or, form a new polynomial $h_1(x)$ where the coefficients h_i are given by the coefficients h_{n-i} of the original polynomial $h_o(x)$.

8) The power spectral density of a PN sequence is discrete.

This property is rather easily derived from the well known relationship between the autocorrelation function and the power spectral density function:

$$G(f) = F[R(\tau)] \tag{17}$$

Since $R(\tau)$ is periodic $G(f)$ is discrete.

$G(f)$ has a $\text{sinc}^2(x)$ shape with the first zero crossings occurring at $\pm f_c$ and with a line spacing of f_c/L Hz where f_c is the clock frequency (or $1/T_c$). The equation describing $G(f)$ is:

$$G(f) = \frac{1}{L^2}\delta(f) + \frac{L+1}{L^2} \sum_{\substack{i=-\infty \\ i\neq 0}}^{\infty} \left(\frac{\sin \frac{\pi f}{f_c}}{\frac{\pi f}{f_c}} \right)^2 \delta(f - i\,\frac{f_c}{L}) \tag{18}$$

From Eq. (18) it appears to be possible experimentally to determine f_c and L if the spectrum could be resolved to $\Delta f = f_c/L$.

9) The crosscorrelation function of two PN sequences is defined as:

$$R_c(j) = \frac{1}{L} \sum_{k=0}^{L-1} g_i(k)\, g_m(k+j) \tag{19}$$

where $g_i(k)$ and $g_m(k+j)$ are two codes of the same length, L. Let θ define the summation of the absolute value of the crosscorrelation function, $R_c(j)$ over all j. Thus

$$\theta = \frac{1}{L} \sum_{j=0}^{L-1} |R_c(j)| \tag{20}$$

The parameter θ is of prime interest, as it describes the "average" orthogonality of the two codes g_i and g_m. It can be shown that as $L \to \infty$

$$\lim_{L>>1} \theta = \left(\frac{2}{\pi L}\right)^{\frac{1}{2}} \approx .8/\sqrt{L} \tag{21}$$

A frequency distribution plot for the different values of $R_c(j)$ would show

$$\lim_{L>>1} \text{VARIANCE } (|R_c(j)|) = .363/L \tag{22}$$

which makes the standard deviation equal to $.6/\sqrt{L}$.

Extensive computer simulations have shown that not all PN sequences are useful for communication applications. What limits their usefulness is the high crosscorrelation values, an indication of which is given by the value of θ as defined above in Eq. (20). Furthermore, the computer simulations have shown θ to vary significantly from best case to worst case in a group of sequences generated by an N-stage shift register. Table 4 shows these results.

It has been observed that, in general, if F is a factor of L then $|R_c(\tau)|$ will be approximately equal to $\frac{1}{F}$ for approximately F values of τ. For example, an 11 stage shift register generates a number of PN sequences of length $L = 2^{11} - 1 = 2047$. The number 23 divides 2047 exactly and,

Table 4

Variation of θ from Best to Worst Pair

N	L	θ (worst pair)	θ (best pair)
5	31	0.35	0.29
6	63	0.36	0.24
7	127	0.32	0.13
8	255	0.37	0.12
9	511	0.22	0.06
10	1023	0.37	0.06
11	2047	0.14	0.03
12	4095	0.34	0.03
13	8191	0.09	0.02

therefore, $|R_c(\tau)|$ can be expected to have roughly 23 positions where it will have a value equal to 1/23.

This section has intended to present some of the reasons for using PN sequences in communication systems, together with some of the properties of PN sequences. Whenever possible, it has been attempted to point out problem areas and limiting factors in the performance of such PN codes. These properties and comments will be used in subsequent sections to devise new bounds on the operational performance of the communication systems employing PN codes.

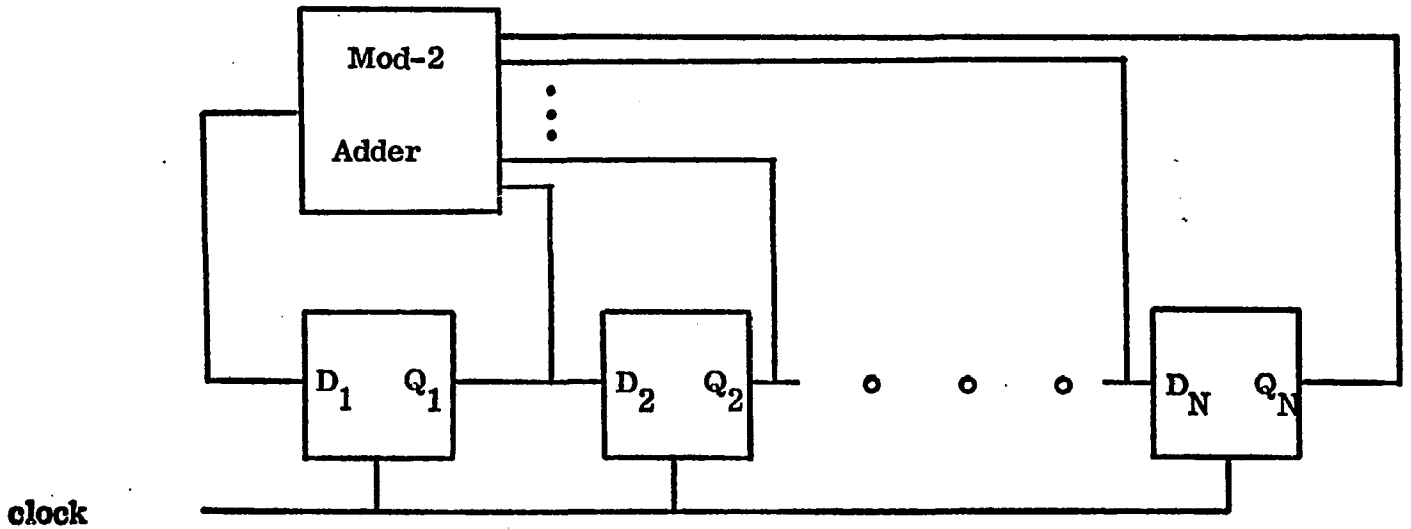


Fig. 2 Shift Register Configuration to Generate PN Sequences

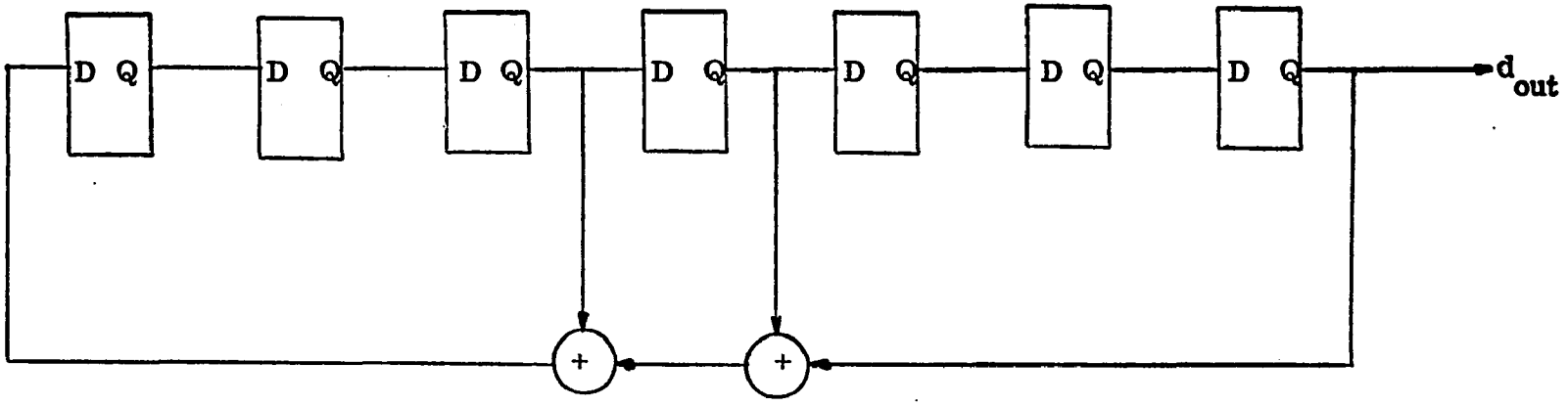


Fig. 3 Generation of PN Sequence Corresponding to h(x)=461

A Worst Case Pseudo-Noise Sequence

Introduction

The previous sections have described in detail the principles of operation of a typical spread spectrum system (DS type). It has been shown that communications may not begin until the acquisition process has ended, where acquisition referred to the phase synchronization of the received and the locally generated PN sequences. In order to acquire the necessary background to the problem to be solved, a relatively concise but rather complete description of the properties of maximal length PN sequence has been given. With this preliminary presentation out of the way, the acquisition problem can be restated as follows: given a received PN sequence, $V_c g(t-jT_c)$, which may or may not be corrupted by additive, white gaussian noise of zero mean and power spectral density $\eta/2$, determine whether the locally generated sequence $g(t-iT_c)$, is synchronized to it (i.e. determine whether i=j). If synchronism does in fact exist, begin communicating. If synchronism does not exist, then change the phase of the local sequence (i.e. change i to some i') and again determine if synchronism has been attained. Repeat the above procedure until synchronism is in fact attained and communications may begin.

In the sections to follow a number of different acquisition systems will be described in some detail. Even though there are major differences between them, it will be clear

that they all must decide at one time or another if, in fact, synchronism has been attained (if i = j). All of the systems described make this decision via a correlation process, where the operations performed are shown in Fig. 4.

As Fig. 4 clearly shows, the received signal is multiplied by the local PN sequence and the result is integrated. At time T the output of the integrator will be:

$$V_o(T) = \int_o^T dt \; V_c g(t-jT_c) g(t-iT_c) + \int_o^T dt \; n_w(t) g(t-iT_c) \quad (1)$$
$$= S_o(T) + n_o(T)$$

The problem faced by the designer is precisely the proper choice of the integrator(correlation) time T. If T is small, the acquisition time may be very short but the probability of an error is large. Figure 5 shows the output of the correlator vs. time, t, for the case where i = j and i ≠ j.

It is clearly shown that even for the special case where noise is not present, a decision should not be reached before $t = T_o$ as the two curves cannot be distinguished. If in fact a decision is reached before $t = T_o$, then the system might believe that i = j when in reality i ≠ j. This kind of error which can, of course, occur at any time $0 < t < \infty$ when noise is present, is called <u>false acquisition</u>. If the system believes that i ≠ j when in fact i = j, then another type of error is made. This error is called <u>false dismissal</u>.

The first problem faced was to determine the minimum time $t = T_o$ such that even in the absence of noise no decision

should be attempted for $t < T_o$. If noise is present, however, then it is possible that the correlation process could last for $t = T_1$ seconds and the system must decide upon examining $V_o(T_1)$ whether the signal is $V_o(T_1) = V_s$ or $V_o(T_1) = V_{ns}$. But this decision cannot be absolutely guaranteed due to the presence of noise. What the system may do is decide if the signal $V_o(T_1)$ is larger than a threshold voltage $V_T(T_1)$ or less than the threshold voltage $V_T(T_1)$ and on the basis of this decision estimate which one of the possible values of $V_o(T_1)$ is most likely.

Obviously an error will be made if the noise term, $n_o(T_1)$, will be larger than ε. But the noise, $n_o(t)$, at time T_1, is a random variable whose statistics are relatively easy to estimate. The designer then may notice that, in general, the two curves tend to get further apart as the time t increases and decide that, if he wants to maintain a low probability of error in estimating the value of $S_o(t)$, he should just integrate for a longer time. The user of such a system, however, may be equally unhappy with a high probability of error, or with an excessive acquisition time, T_a. The designer will then be forced to search for an optimal solution, such that a low probability of error is maintained and the acquisition time T_a is not excessive. He may decide that, in fact, this may not be a very hard problem, as the upper curve is known to be described by a linear equation, $V_{(t)} = V_c t$. The lower curve, however, is very difficult to describe analytically, as it is

a function of the generating polynomial, h(x), of the particular phases i and j and of the integration time, T. It is the equation for this lower curve that we attempt to find, in such a way that it would be universally useful (independent of i, j, and h(x)).

For the special case where $n_o(t) = 0$ the signal $V_o(t)$ is equal to

$$V_o(t) = S_o(t) = V_c \int_o^T dt\ g(t-jT_c)\, g(t-iT_c) \qquad (2)$$

Using the shift and adding property of PN sequences this may be rewritten as:

$$V_o(t) = S_o(t) = V_c \int_o^T dt\ g(t-\ell T_c) \qquad (3)$$

where $\ell \neq i \neq j$. If a PN sequence were to be integrated, the worst phase would be such that it would maximize the time T_o of Fig. 5. But the highest possible T_o occurs if the largest possible run of +1s is integrated. The run length distribution of PN sequences has indicated this to be N successive +1s. Therefore the minimum necessary correlation time should always be larger than NT_c. After at most N successive +1s, however, a -1 will occur which will make the correlator's output decrease for at least a $\Delta t = T_c$. If this curve is to approximate the linear curve, $V_{(t)} = V_c t$ as closely as possible, then the run of -1s will be limited to a length of one and another run of +1s will follow. This run, however, can no longer be made up of N successive +1s, as the run length

distribution clearly limits the number of runs of length N to one. The next longest run is made up of N-2 successive +1s and it should also be followed by a run of length one of -1s. Continuing in this manner until all of the runs are exhausted, the curves of Fig. 6 are obtained.

Since the runs' length distribution property of PN sequences has not been altered, but the particular run arrangement has been selected for a worst case example, the curve $V_{ns}(t)$ obtained in Fig. 6 is a tight, worst case bound to the partial autocorrelation of a PN sequence. This partial autocorrelation is described by:

$$r_p(T=\gamma T_c) \approx V_c T_c \gamma \left(1-\frac{\gamma}{L}\right) \tag{4}$$

where γ represents the number of chips to be correlated and $L = 2^N-1$ if N represents the number of delays in the generating shift register (or the degree of generating polynomial).

Application of the Worst Case Pseudo-Noise Sequence

Bit-by-Bit Detection. Assume that, for this example, the system examines the incoming sequence by integrating bit by bit. This implies that the first bit of the incoming sequence is multiplied by the first bit of the local sequence, the result is kept, the integrator is dumped, and the process starts again. Then for every bit

$$\begin{aligned} V_o(T_c) &= V_o \int_o^{T_c} dt\, g(t-jT_c)\, g(t-iT_c) + \int_o^{T_c} dt\, n_w(t)\, g(t-iT_c) \\ &= S_o(T_c) + n_o(T_c) \end{aligned} \tag{5}$$

Assuming the incoming noise to be white, gaussian with power spectral density $\eta/2$ the output noise $n_o(T_c)$ will be gaussian, zero mean, with variance $\sigma_o^2(T_c) = \frac{\eta}{2}T_c$

If the two sequences are synchronized, then the output of the integrator will always be $V_o(T_c) = V_cT_c + n_o(T_c)$. If, on the other hand, the sequences are not in synchronism, the output of the integrator may be $V_o(T_c) = -V_cT_c + n_o(T_c)$ as well as $V_o(T_c) = V_cT_c + n_o(T_c)$. If we wait for (N+1) chips and then add up the integrator's output for every chip, we are guaranteed that at least once the integrator's output was equal to $-V_cT_c$ and therefore the voltage obtained should be less than $(N+1)V_cT_c$. This can be written as:

$$\sum_{k=1}^{N+1} V_c \int_{(k-1)T_c}^{kT_c} dt\, g(t-jT_c)g(t-iT_c) < (N+1)V_cT_c \qquad (6)$$

since at least for one k the quantity

$$V_c\int_{(k-1)T_c}^{kT_c} dt\, g(t-jT_c)g(t-iT_c) = -V_cT_c$$

This reasoning is correct and it holds well for a no noise environment. If noise is present, however, we have a finite probability of error. Should this error occur during the first N chips, then it would actually aid in our decision. If it occurs in the $(N+1)^{st}$ chip, however, we would appear to be synchronized when, in fact, we are not, and an error would

be made. If no noise is to occur again an examination of at least 2N-1 chips would be necessary to detect the non-synchronism condition. Therefore, the designer should be aware that based on the noise power and the respective probability of error in chip decoding, the number of chips examined might have to be considerably larger than N+1. Figures 7, 8, 9, and 10 plot the worst case number of chips to be examined vs. the number of expected errors in chip decoding for the artificial sequence and typical PN sequences of different lengths. The figures show the bound for the worst case noise placement to be tight.

Block Decoding. It is not a good practice, however, to decode one chip at a time and then sum the results (hard decision). A much better method is to correlate a portion of the received PN sequence with a portion of the local PN sequence and make a decision at time $T = \gamma T_c$ as to their state of synchronism (soft decision). Knowing that the output noise is a gaussian random variable with zero mean and variance $\sigma_o^2(\gamma T_c) = \frac{\eta}{2}\gamma T_c$ a decision can be made if $S_o(t) = V_c \gamma T_c$ or $S_o(t) = V_c T_c \gamma(1-\frac{\gamma}{L})$ (i.e., if the sequences are synchronized or not) with an almost arbitrarily small probability of error by correlating for a long enough time. Indeed, at time $t = T = \gamma T_c$ the distance between the curves is

$$d = 2\varepsilon = V_c \gamma T_c - V_c T_c \gamma(1-\frac{\gamma}{L}) = T_c V_c \frac{\gamma^2}{L} \qquad (7)$$

and the threshold voltage, $V_T(T=\gamma T_c)$ becomes

$$V_T(\gamma T_c) = V_c \gamma T_c - \varepsilon = V_c T_c \gamma \left(1-\frac{\gamma}{2L}\right) \tag{8}$$

The probability of an erroneous decision is given by:

$$P_e = P[n_o(\gamma T_c) > \varepsilon] =$$

$$= \int_{\varepsilon}^{\infty} dn \, \frac{1}{\sqrt{2\pi}\ \sigma_o(\gamma T_c)} \, e^{-\frac{n^2}{2\sigma_o^2(\gamma T_c)}}$$

$$= \frac{1}{2} \operatorname{erfc}\left[\frac{\varepsilon}{\sqrt{2}\sigma_o(\gamma T_c)}\right] \tag{9}$$

$$= \frac{1}{2} \operatorname{erfc}\left[\frac{\gamma}{L}\sqrt{\frac{\gamma E_c}{4\eta}}\right]$$

Eq. (9) shows the argument of the complementary error function to be increasing as $\gamma^{3/2}$ for a given signal to noise ratio E_c/η and this proves that a γ can be found for any desired probability of error, P_e.

The bound has been used in extensive computer simulations which have proved its tightness. Figures 11 and 12 show the result of those simulations, while Fig. 13 shows a plot of γ vs. $\frac{E_c}{\eta}$ for different value of the P_e.

Conclusion

A bound has been developed which upper bounds the partial autocorrelation of a PN sequence. The tightness of the bound has been determined via extensive computer simulations. The

sections following will demonstrate the usefulness of the results as they are applied to the three acquisition schemes considered: a) the shift register acquisition, b) the L-correlator acquisition, and c) the sliding-correlator acquisition.

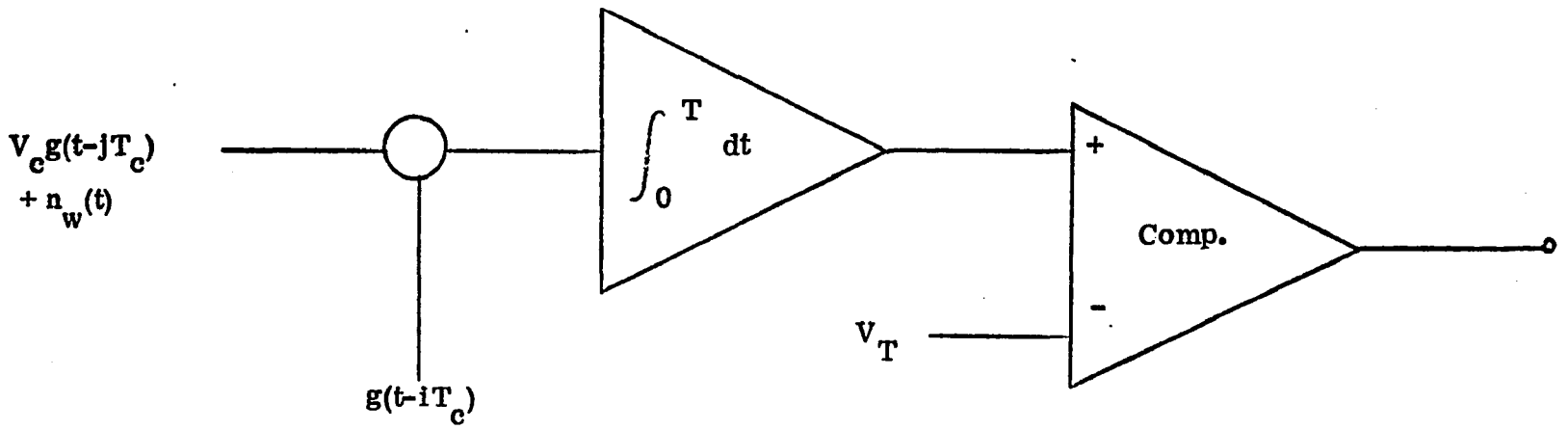


Fig. 4 PN Sequence Correlation

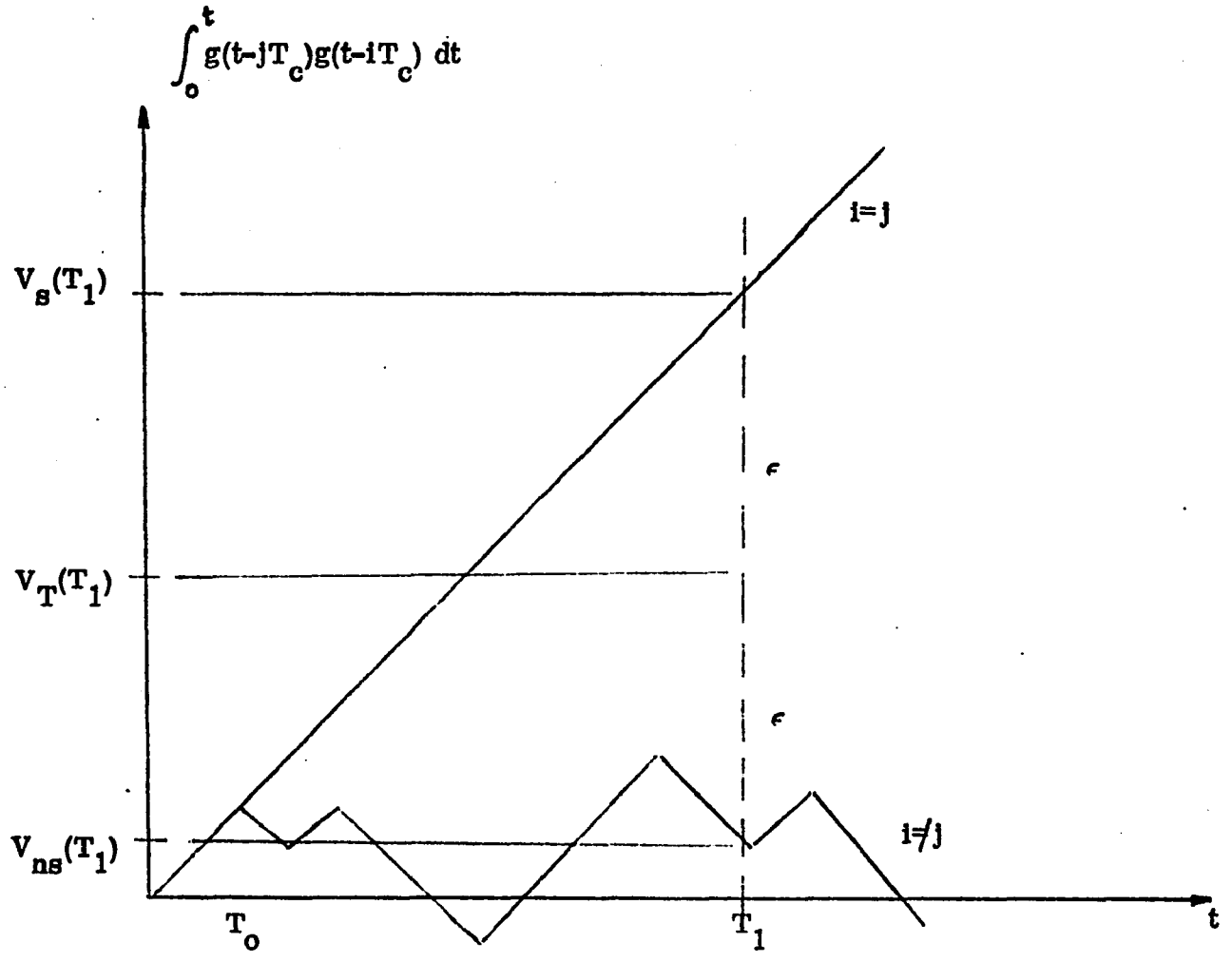


Fig. 5 Typical Autocorrelation of PN Sequences

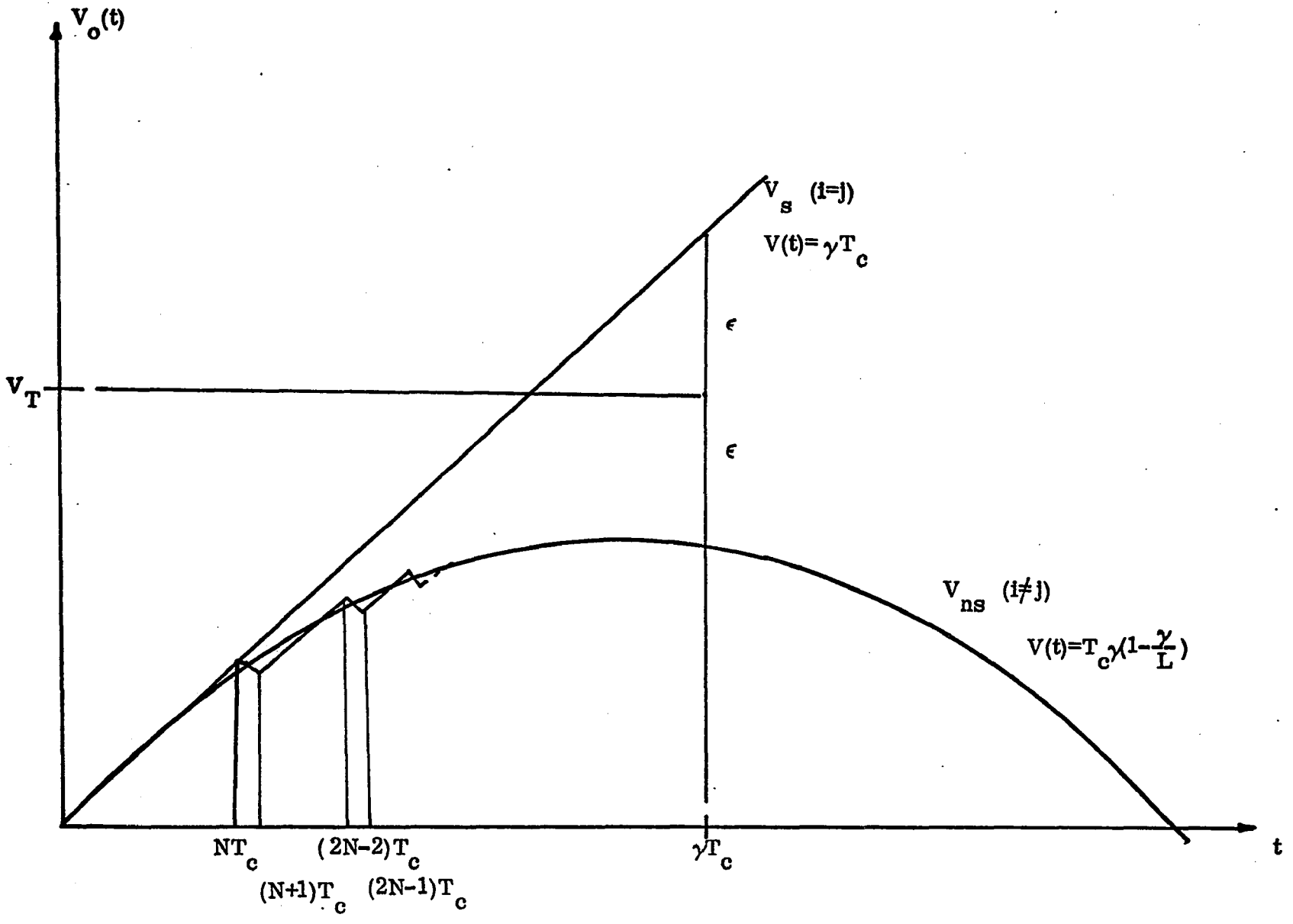


Fig. 6 Worst Case PN Autocorrelation Function

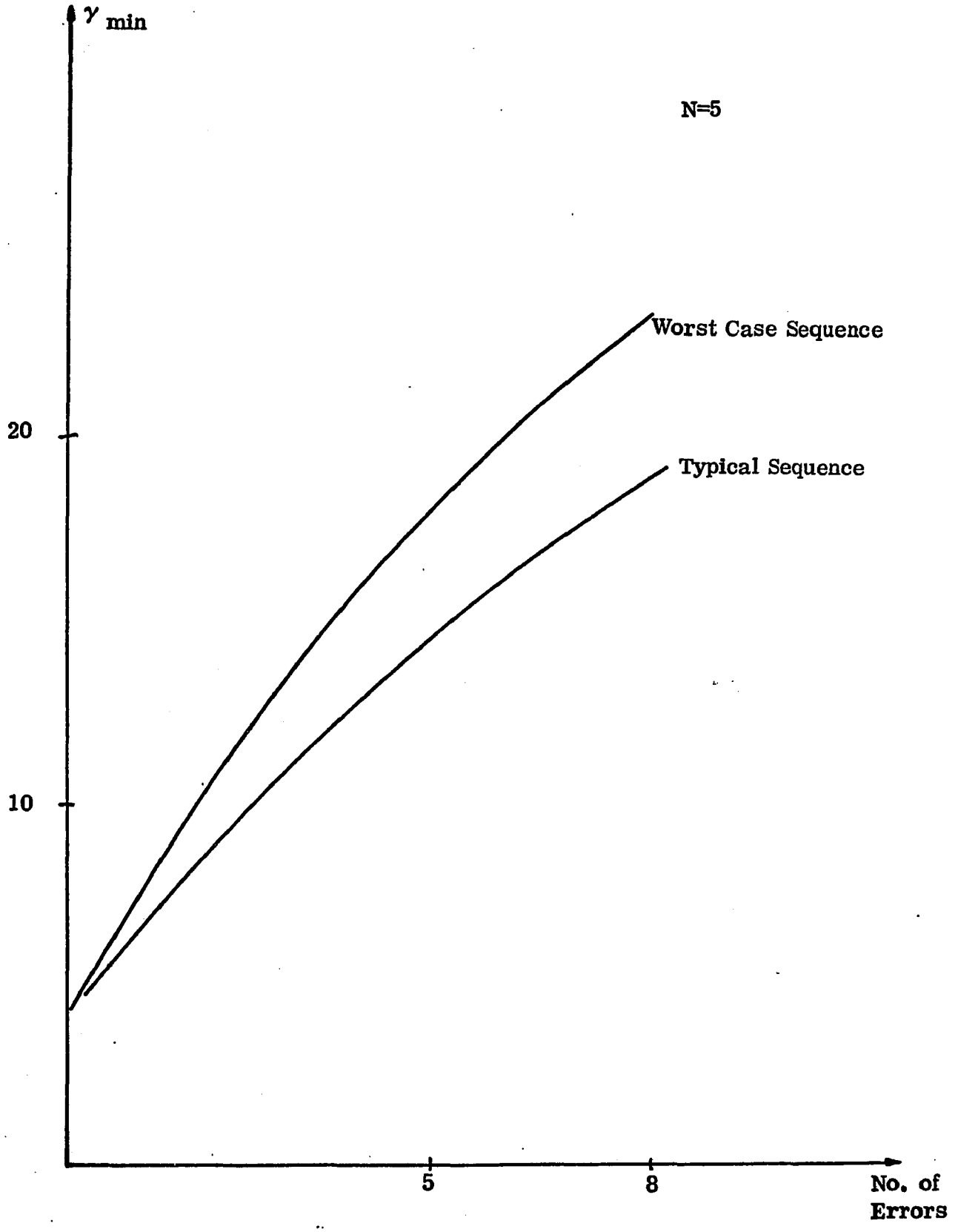


Fig. 7 γ_{min} versus No. of Errors (Hard Decision) for N=5

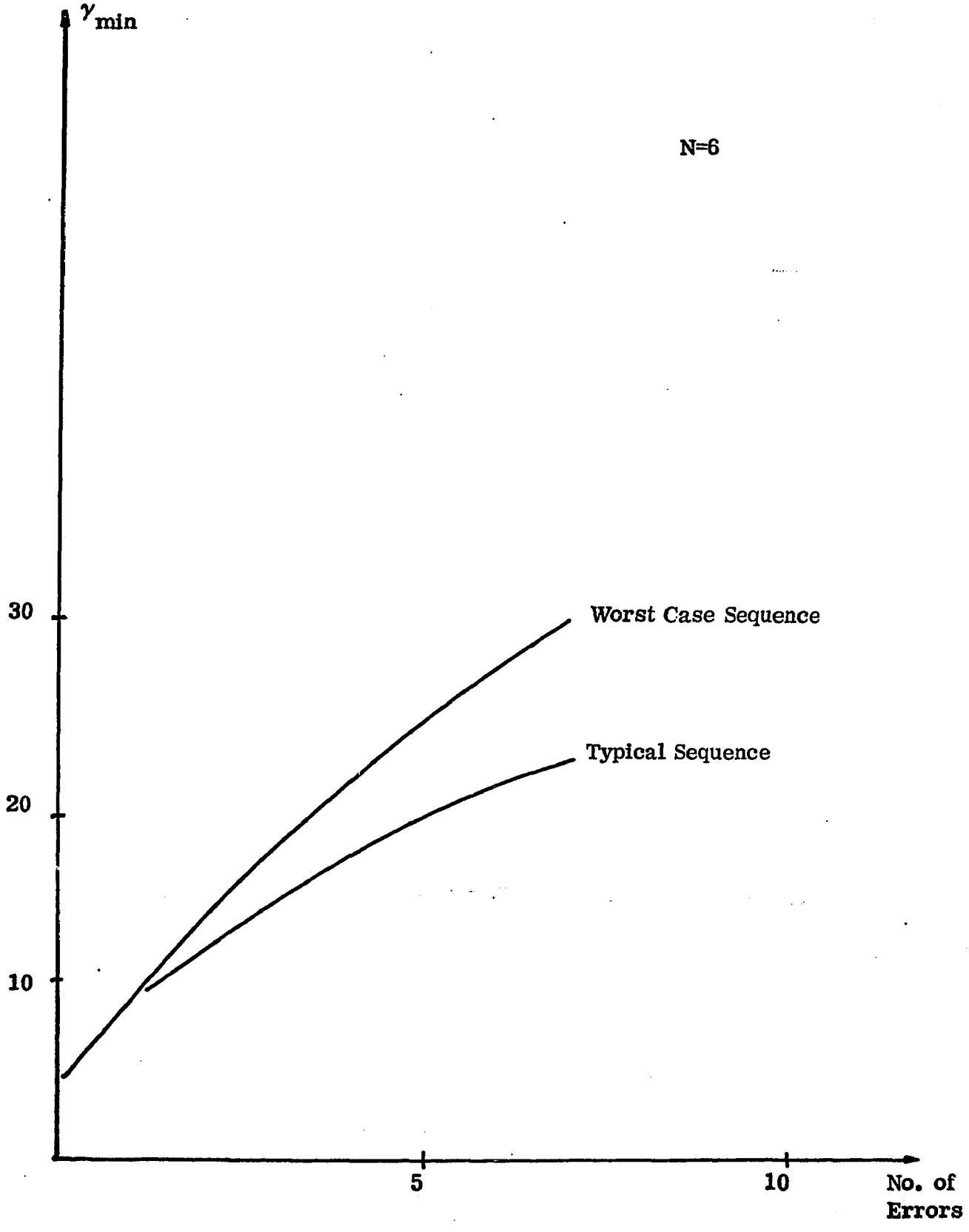


Fig. 8 γ_{min} versus No. of Errors (Hard Decision) for N=6

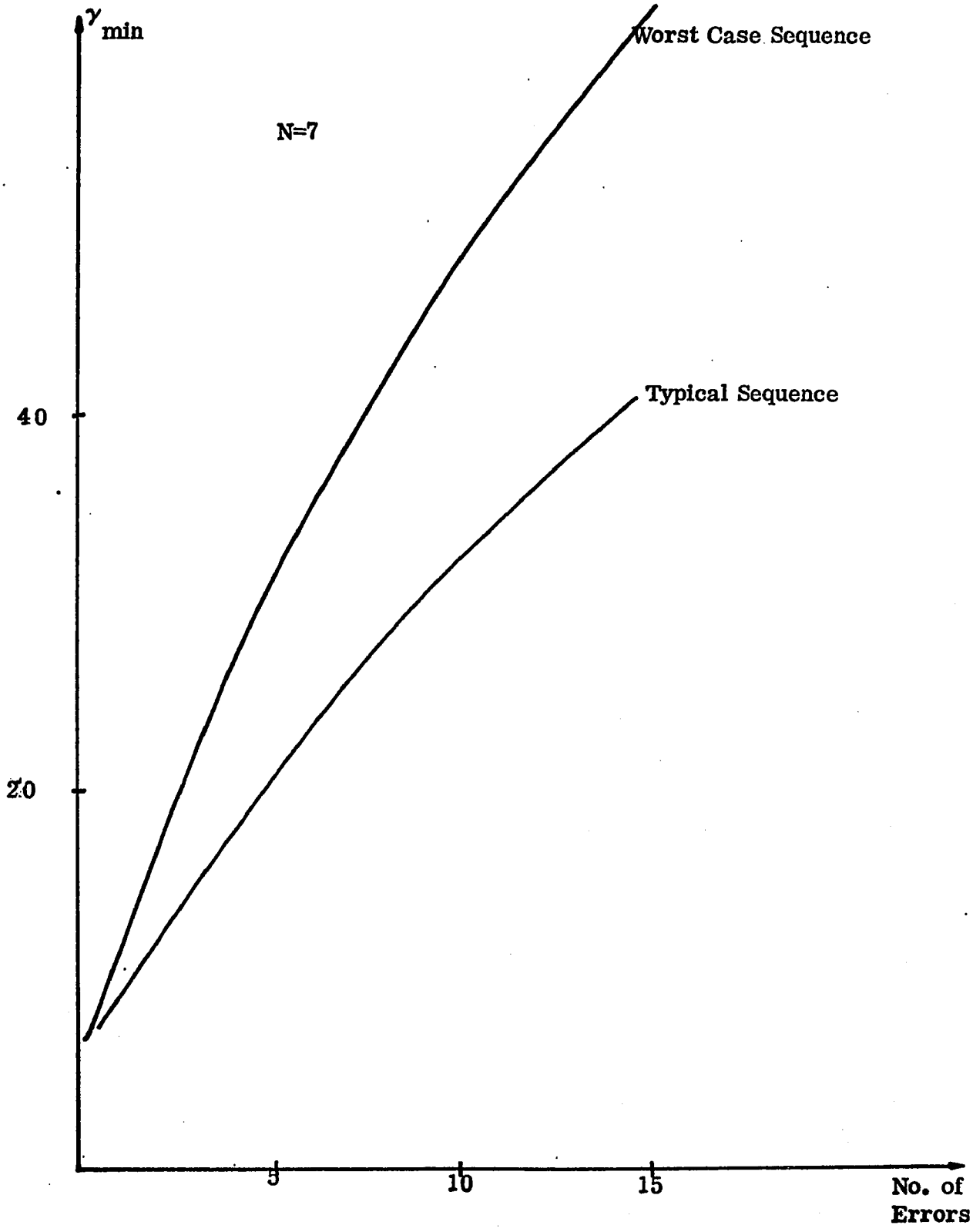


Fig. 9 γ_{min} versus No. of Errors (Hard Decision) for N=7

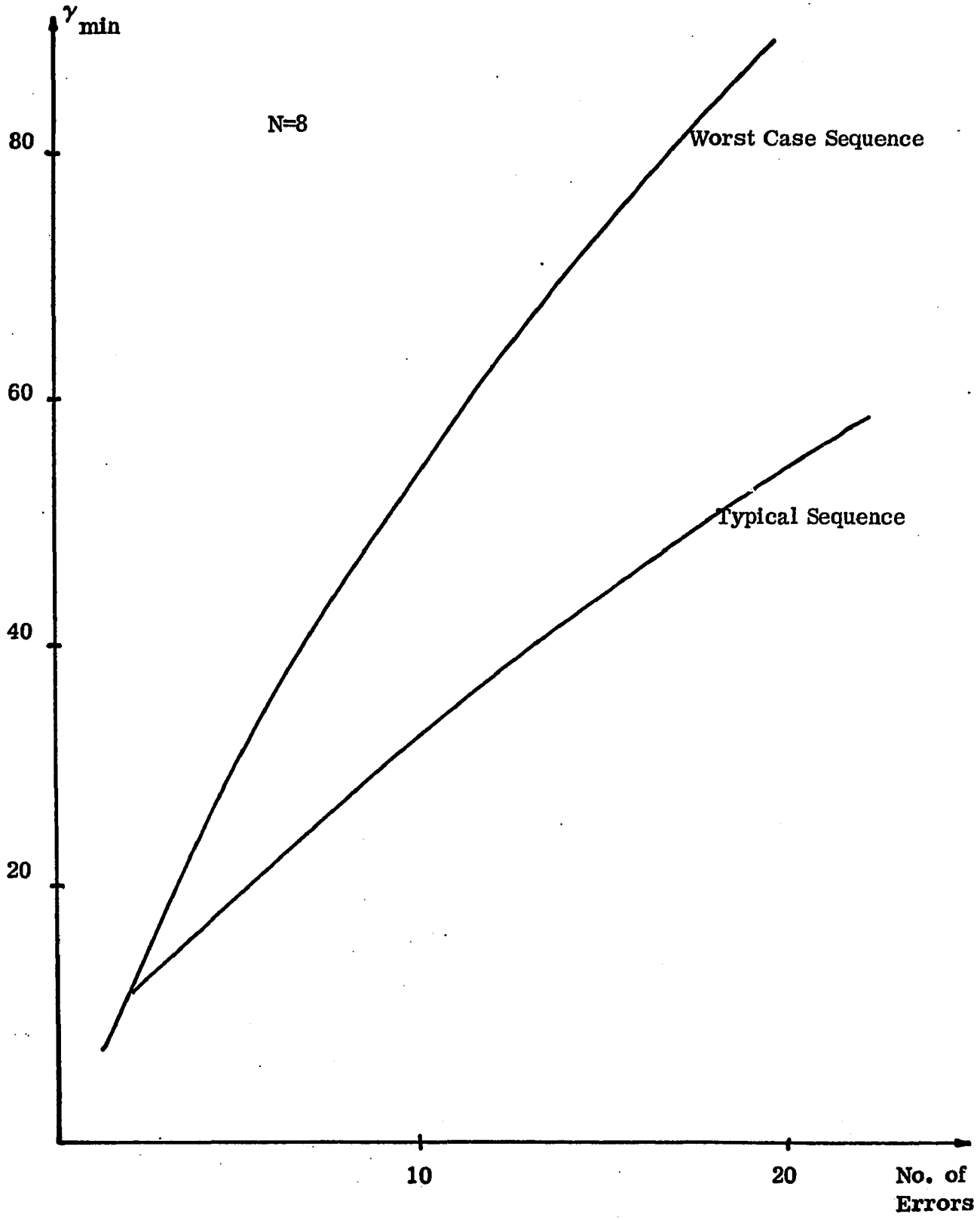


Fig. 10 γ_{min} versus No. of Errors (Hard Decision) for N=8

Fig.11 Probability of Acquisition Error vs. Chip to Noise Power (no jammer)

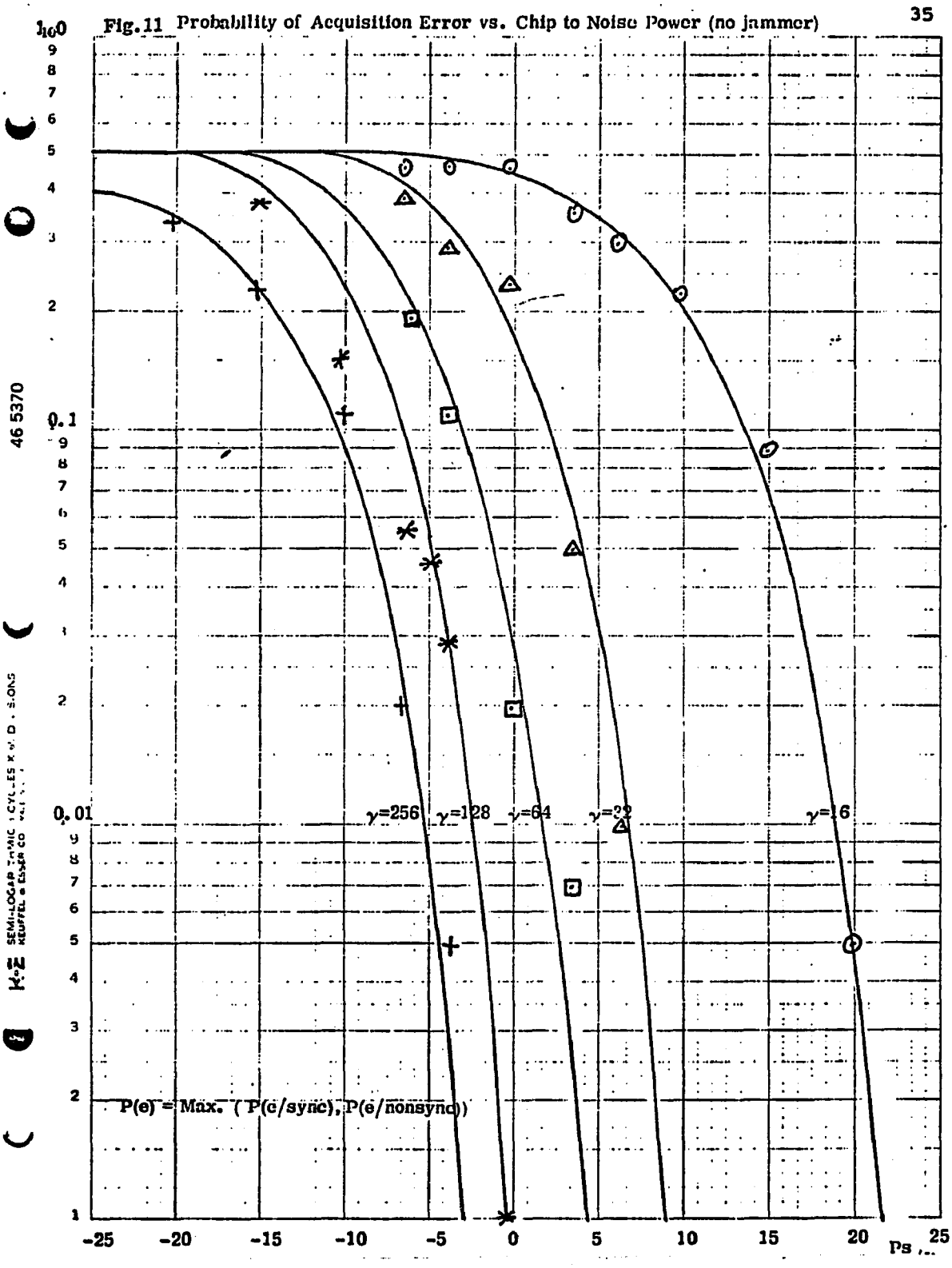
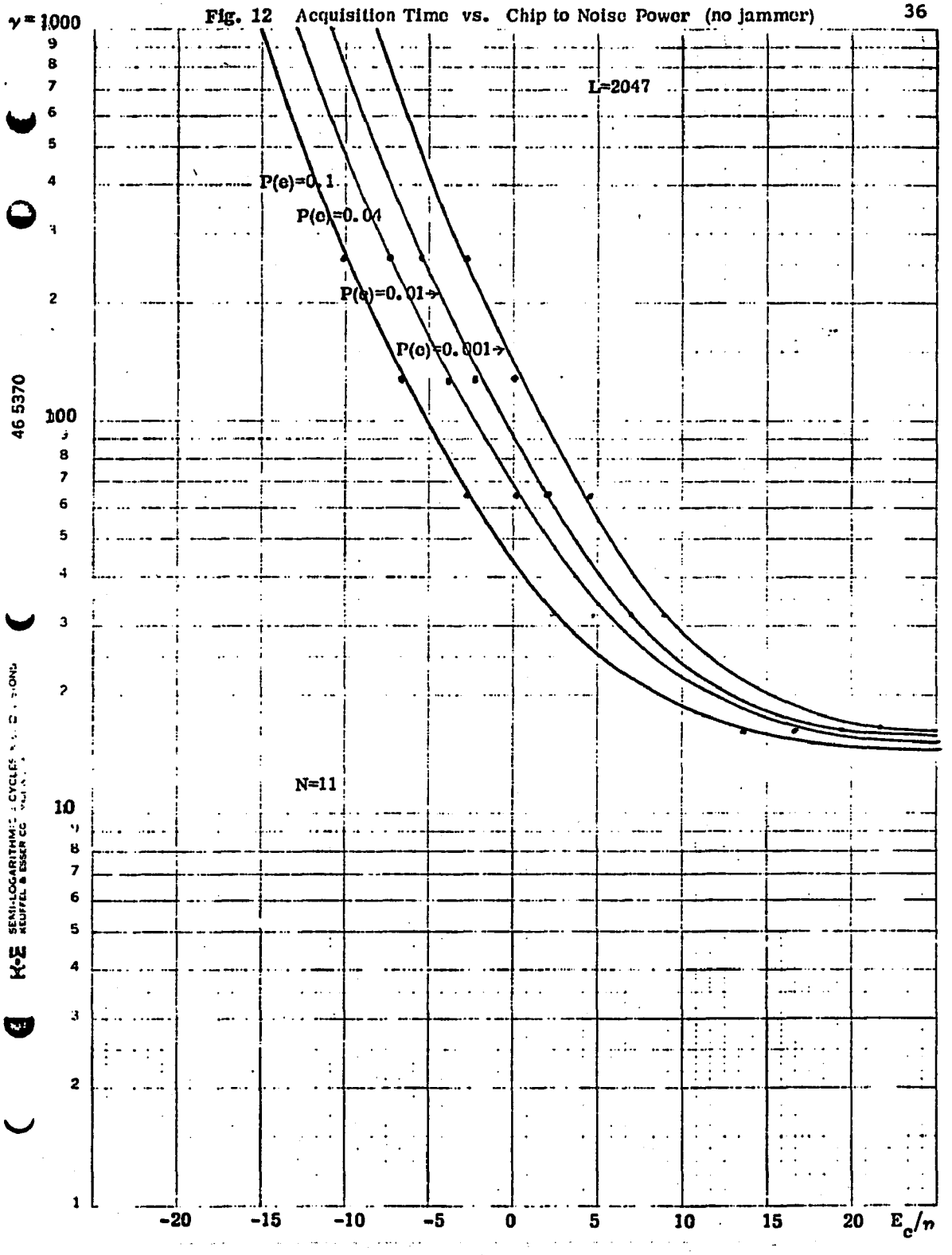


Fig. 12 Acquisition Time vs. Chip to Noise Power (no jammer)



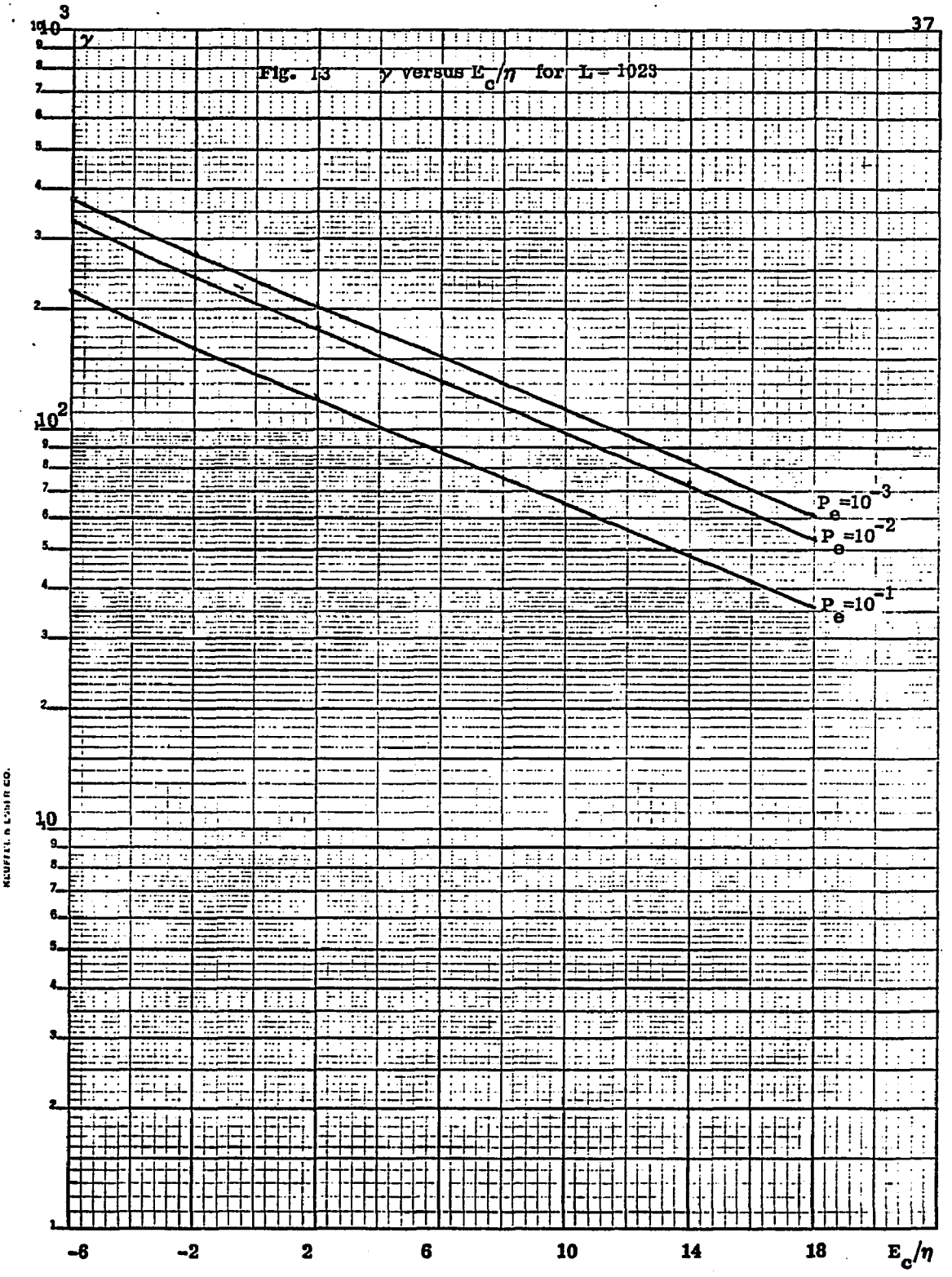


Fig. 13 γ versus E_c/η for $L=1023$

APPLICATIONS OF THE PARTIAL AUTO-CORRELATION BOUND TO EXISTING ACQUISITION SYSTEMS

Shift Register Acquisition

Introduction

A block diagram of a typical shift register acquisition circuit is shown in Fig.14. The circuit can be divided into two parts: 1) The PN generator part and 2) the decision logic part. The PN generator consists of an N stage shift register and its associated feedback logic. The next bit to be shifted into the register, however, can come from one of two sources: a) The first source is directly from the received signal and b) the second source is the feedback logic. The received signal is (after demodulation):

$$S_i(t) = V_c g(t-jT_c) + n_w(t) \qquad (1)$$

where $n_w(t)$ is gaussian, white noise, with power spectral density $\eta/2$. The low pass filter eliminates as much of the noise as possible, and the output of the filter is loaded into the shift register bit by bit. After loading all of the N bits into the shift register, the switch S is changed from position 1 to position 2. The initial N bits which have been loaded from the received signal into the shift register form, what is called a "starting vector." If they have been loaded correctly, then the locally generated sequence will be identical

to the received sequence (i.e., the two sequences will be synchronized in phase). If such is the case, the multiplier preceding the integrator will always receive inputs of the same polarity and will always generate a +V at its output. The integrator's output, then, will always be increasing and at time $T = \gamma T_c$ its output will be above the threshold voltage V_T. This will indicate that the local sequence has been synchronized to the received sequence and data decoding can begin.

It is possible, though, to load at least one of the initial N bits in error. If such is the case, then the local sequence and the generated sequence will be out of phase with each other. In that case, the multiplier will not always multiply the same signal by a replica of itself, but the two signals will only be in agreement for roughly one half the time and in disagreement for roughly the other half of the time (provided the integration time is long). The integrator's output, then, will waver up and down, never significantly departing from a "low" voltage value, and will probably not be able to exceed the threshold voltage V_T at $T = \gamma T_c$. The comparator will sense that at time $T = \gamma T_c$, and the switch will be changed back to position 1, such that a new N bit starting vector can be loaded in and the whole procedure repeated until the two sequences are in synchronism (all the initial N bits will be loaded correctly).

Shift Register Acquisition Performance in the Absence of Noise

The performance of the shift register acquisition circuit depends very heavily on the signal to noise ratio of the received signal. As expected, the minimum acquisition time occurs when there is no noise. In this case the acquisition time can be shown to be:

$$T_a = (N+1)T_c \tag{2}$$

where T_c = chip duration, and N is the length of the shift register.

Shift Register Acquisition Performance in the Presence of Noise

The degree of acceptance of an acquisition system is heavily dependent on its performance throughout the range of signal to noise ratios likely to be encountered in actual operation. It may well be the case that a system is chosen over another system because it performs better under low signal to noise conditions, even though it may have a poorer performance at better signal to noise ratios. It is intuitively obvious that a smaller signal to noise ratio will result in a degradation in the acquisition performance, where the acquisition performance could be measured as the probable acquisition time, T_a, for a given probability of false decisions. A false decision can be defined to be a false dismissal or a false

acquisition with the two probabilities not necessarily being equal. The level set for the two probabilities of false decision will, in turn, affect the acquisition time, T_a.

To see how the acquisition times vary with the given probability of false decisions, let the integration time be LT_c.

Referring back to Fig.14, the signal to the integrator will be:

$$\begin{aligned} V_i(t) &= [V_c g(t-jT_c) + n_w(t)]g(t-kT_c) \\ &= V_c g(t-jT_c)g(t-kT_c) + n_w(t)g(t-kT_c) \end{aligned} \tag{3}$$

and the output of the integrator will be the sum of two signals, $S_o(t)$ and $n_o(t)$.

$$V_o(LT_c) = \int_o^{LT_c} V_c g(t-jT_c)g(t-kT_c)dt + \int_o^{LT_c} u_w(t)g(t-kT_c)dt =$$

$$= S_o(LT_c) + n_o(LT_c) \tag{4}$$

The first term, $S_o(t)$, is the signal term, and it is known that at time $t = LT_c$ it can have one of two exact values:

$$S_o(t) = \int_o^{LT_c} V_c g(t-jT_c)g(t-kT_c) = \begin{cases} V_c LT_c & \text{if } j = k \\ -V_c T_c & \text{if } j \neq k \end{cases}$$

The second term, $n_o(LT_c)$ is gaussian noise with variance $\sigma_o^2 = \frac{n}{2}LT_c$ and zero mean.

The acquisition becomes often a decision problem, as shown in Fig.15.

In the absence of noise the output of the integrator can only take one of two values which are given by Eq. (5). The presence of noise, however, adds the second term of Eq. (4). The decision as to the state of synchronism of the local PN sequence to the received PN sequence reduces to correctly identifying the value of $S_o(t)$. The shaded areas of Fig.15 are proportional to the two errors which are possible to make: The first error is known as the false dismissal and it occurs when the noise makes the output of the integrator fall in region I when the signal $S_o(t) = V_c L T_c$. The second type of error occurs when the noise makes the output of the integrator fall in region II when the signal $S_o(t) = -V_c T_c$ and this type of error is known as false acquisition. Note that the relative probability of occurrence of the two types of error changes with the choice of V_T, the threshold voltage. If the probability of a false dismissal is to equal the probability of a false acquisition, then V_T will be set in the middle and

$$V_T = \frac{1}{2}[V_c L T_c + V_c T_c] = V_c T_c \frac{L+1}{2} = \varepsilon \qquad (6)$$

The probability of an error occurring (where the possible errors have been described above) is given by

$$P_e = P[n_o(t) > \varepsilon] \qquad (7a)$$

where ε is related to the value the threshold voltage V_T is set at by

$$V_T = V_c \gamma T_c - \varepsilon = V_c T_c \gamma \left(1 - \frac{\gamma}{L}\right) + \varepsilon \tag{7b}$$

The output noise, $n_o(LT_c)$, is gaussian, with 0 mean and $\sigma_o^2(LT_c) = \frac{\eta}{2} LT_c$. Therefore:

$$P_e = \int_{\varepsilon}^{\infty} \frac{1}{\sqrt{2\pi}\ \sigma_o(LT_c)}\ e^{-\frac{n^2}{2\sigma_o^2(LT_c)}}\ dn \tag{8}$$

Making the substitution, $m = \frac{n}{\sigma_o \sqrt{2}}$ and $dn = \sqrt{2}\sigma_o(LT_c)\,dm$, Eq. (8) becomes:

$$P_e = \int_{\frac{\varepsilon}{\sqrt{2}\sigma_o(LT_c)}}^{\infty} \frac{1}{\sqrt{\pi}}\ e^{-m^2} = \frac{1}{2}\ \mathrm{erfc}\left[\frac{\varepsilon}{\sqrt{2}\sigma_o(LT_c)}\right] \tag{9}$$

It is more convenient to express the argument of the complementary error function as

$$\frac{\varepsilon}{\sqrt{2}\sigma_o} = \frac{V_c T_c (L+1)}{2\sqrt{2\ \frac{\eta}{2}\ LT_c}} = \left[\frac{V_c^2 T_c^2 (L+1)^2}{4\ \eta\ LT_c}\right]^{1/2} \simeq \sqrt{\frac{E_c L}{4\eta}} \tag{10}$$

To consider a practical example, let $E_c/\eta = -10$ dB and $L = 2047$. The probability of error is then given by

$$P_e = \frac{1}{2}\ \mathrm{erfc}\ \frac{E_c L}{4\eta} = \frac{1}{2}\ \mathrm{erfc}\ \sqrt{50} < 10^{-20} \tag{11}$$

The result of Eq. (11) helps to illustrate the fact that for most cases there is no need to integrate for a whole length, L. If the integration time is carried over only γ chips the noise, $n_o(t)$, will be gaussian with zero mean and variance

$\sigma_o^2(\gamma T_c) = \frac{\eta}{2}\gamma T_c$ and the signal $S_o(t)$ will be:

$$S_o(t) = \int_o^{\gamma T_c} V_c g(t-jT_c)g(t-kT_c)dt = \begin{cases} V_c\gamma T_c & \text{if } j = k \\ V_c T_c r_p(\gamma) & \text{if } j \neq k \end{cases} \quad (12)$$

Eq. (12) clearly shows that the difficulty lies with the partial autocorrelation function $r_p(\gamma)$ which is not easily obtainable for the case where $\tau \neq 0$. The bound found for this function $r_p(\gamma)$, however, is very tight and can be successfully used in this situation. It bounds $r_p(\gamma)$ by:

$$r_p(\gamma) = V_c T_c \gamma(1-\frac{\gamma}{L}) \quad (13)$$

If the decision threshold voltage is set halfway, then ε becomes :

$$\varepsilon = \frac{1}{2}[V_c\gamma T_c - V_c T_c \gamma(1-\frac{\gamma}{L})] = V_c T_c \frac{\gamma^2}{2L} \quad (14)$$

The probability of error of Eq. (9) becomes now

$$P_e = \frac{1}{2}\,\text{erfc}\sqrt{\frac{\varepsilon^2}{2\sigma_o^2}} = \frac{1}{2}\,\text{erfc}\;\frac{\gamma}{L}\sqrt{\frac{\gamma E_c}{4\eta}} \quad (15)$$

Using the same signal to noise ratio, $\frac{E_c}{\eta}$, and the same length L as before:

$$P_e = \frac{1}{2}\,\text{erfc}\,(\frac{\gamma\sqrt{\gamma}}{1950}) \quad (16)$$

If we let P_e equal to 2×10^{-3} then $\gamma = 255$ chips which

represents a substantial savings when compared to $\gamma = 2047$ chips.

Figure 16 shows the relationship between γ and $\frac{E_c}{\eta}$ for different desired P_e's.

So far, we have only considered a single trial. If noise is present, there may not be enough to load the shift register only once to acquire the PN sequence, as any one of the N chips could be loaded erroneously. The quantity of interest then becomes ν_o, where ν_o = the number of trials within which acquisition will take place with a given probability, P_a. Bearing in mind that the probability of acquisition is the same as the probability of loading all N chips correctly into the shift register, let the following notation be used:

p = probability of loading one chip correctly

p^N = probability of loading N chips correctly

$1-p^N$ = probability of making one or more mistakes in loading the chips.

Thus the probability of having a correct loading of all the N chips in the ν^{th} trial is

$$P_c = p^N(1-p^N)^{\nu-1} \tag{20}$$

The probability of loading the register correctly (or acquiring) within ν_o trials is equal to the probability of acquiring in the first trial + probability of acquiring the second trial + ... + probability of acquiring in the ν_o^{th} trial or:

$$P[\text{acq. in } \nu_o \text{ trials}] = P(\text{acq.}/\nu=1) + P(\text{acq}/\nu=2) + \ldots + P(\text{acq}/\nu=\nu_o)$$

$$= \sum_{\nu_i=1}^{\nu_o} P(\text{acq}/\nu=\nu_i) \tag{21}$$

But the probability of acquisition in the ν^{th} trial is given by Eq. (20). Therefore:

$$P[\text{acq. in } \nu_o \text{ trials}] = \sum_{\nu_i=1}^{\nu_o} p^N (1-p^N)^{\nu_i-1} \tag{22}$$

$$= p^N \sum_{\nu_i=1}^{\nu_o} (1-p^N)^{\nu_i-1}$$

Let

$$1 - p^N = \alpha \tag{23}$$

Then

$$\sum_{\nu_1=1}^{\nu_o} (1-p^N)^{\nu_i-1} = \sum_{\nu_i=0}^{\nu_o-1} (1-p^N)^{\nu_i}$$

$$= \sum_{\nu_1=0}^{\nu_o-1} \alpha^{\nu_i} = \frac{1-\alpha^{\nu_o}}{1-\alpha} \tag{24}$$

where α has been defined above. Then, Eq. (22) can be written as:

$$P[\text{acq. in } \nu_o \text{ trials}] = p^N \sum_{\nu_i=1}^{\nu_o} (1-p^N)^{\nu_i-1}$$

$$= p^N \frac{1-(1-p^N)^{\nu_o}}{1-(1-p^N)} = 1-(1-p^N)^{\nu_o} \qquad (25)$$

Fixing the probability of success (acquisition complete) to $P_a = 0.9$, Eq. (25) becomes:

$$1-(1-p^N)^{\nu_o} = 0.9 = P_a \qquad (26)$$

and

$$\nu_o = \frac{\ln(1-P_a)}{\ln(1-p^N)} \qquad (27)$$

As shown previously, it is possible to trade-off a shorter integration time for a larger probability of error which has been defined as a false acquisition or a false dismissal decision. How this trade-off is made depends very heavily on the probability of correctly loading a chip, p. If it is very probable to load N chips correctly, then the integration time can be short. Even if a false dismissal will occur there will probably not be a long wait for the next correct loading. If, on the other hand, it is very improbable to correctly load N chips, then a false dismissal can be very costly, as it could lead to a long wait before another correct loading occurs. It may well be advisable then to be careful against such an eventuality. The probability of loading a chip correctly is given by the probability that the noise is

less than $V = V_c$:

$$p = p[n(t) < V_c] = 1 - P[n(t) > V_c] \tag{28}$$

But the noise power at the output of the low pass filter =

$$\sigma^2 = \frac{\eta}{2} \quad 2f_c = nf_c$$

and Eq. (28) reduces to:

$$p = 1 - \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E_c}{2\eta}} \tag{29}$$

Even if this probability, p is high, N is usually large enough for $p^N << 1$. Then Eq. 20 with $P_a = .9$ becomes

$$\nu_o = \frac{\ln(1-Pa)}{-p^N} = \frac{2.3}{p^N} \tag{30}$$

If on the average each trial has a duration T_T equal to

$$T_T = NT_c + \gamma T_c \tag{31}$$

then, the acquisition time, T_a, necessary to acquire 90% of the time and with a probability of false dismissal or false acquisition as given by γ, will be equal to:

$$T_a = \nu_o(\gamma+N)T_c \tag{32}$$

Figure 17 shows a plot of T_a vs. $\frac{E_c}{\eta}$ for $f_c = 10^6$, $P_e = 10^{-1}, 10^{-2}, 10^{-3}$.

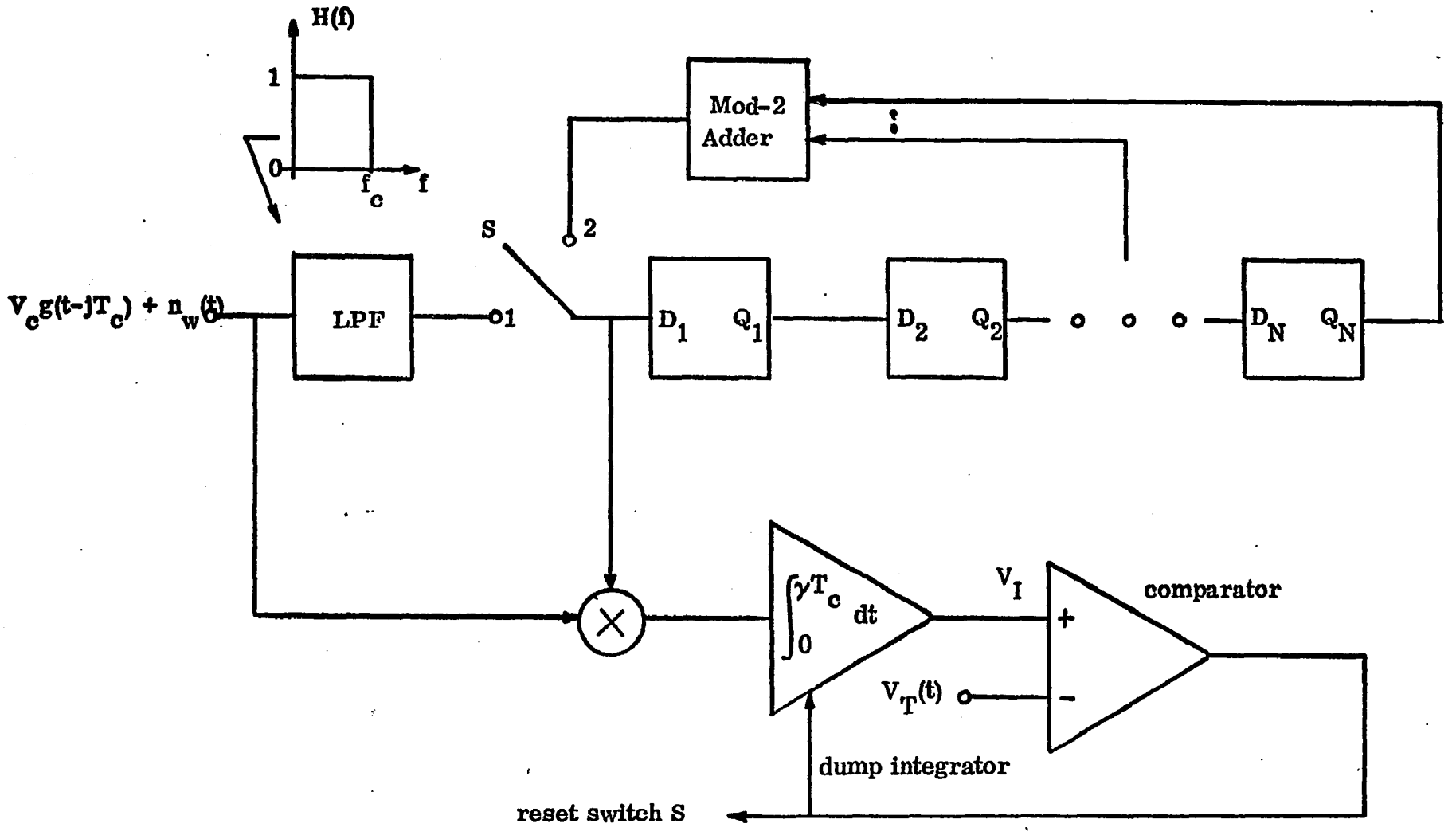


Fig. 14 Shift Register Acquisition Circuit

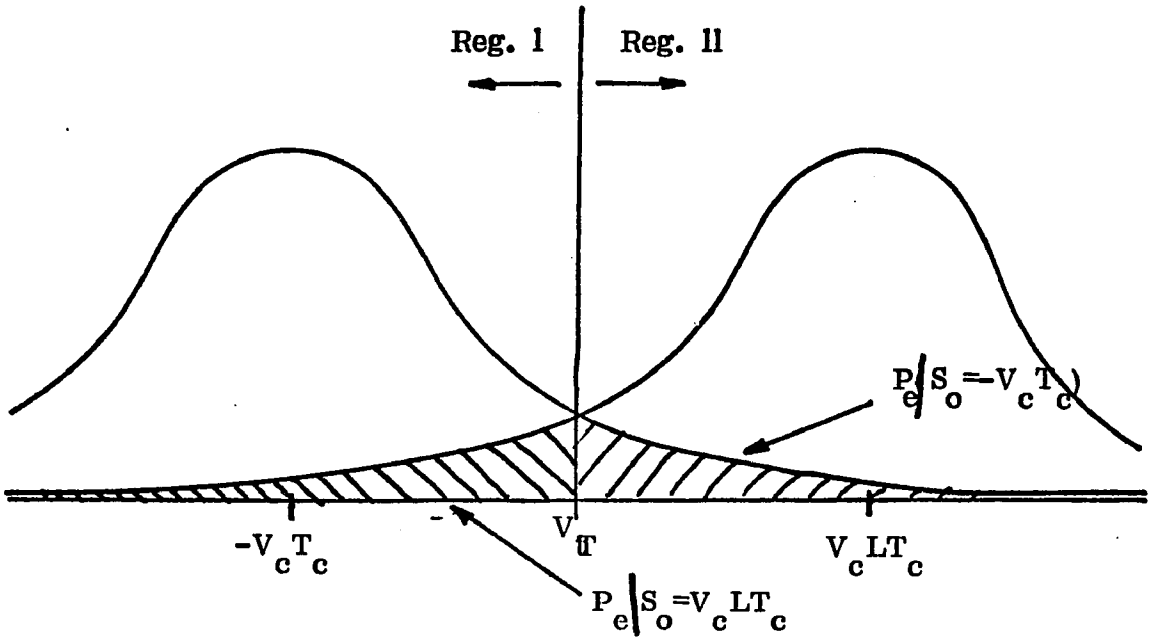
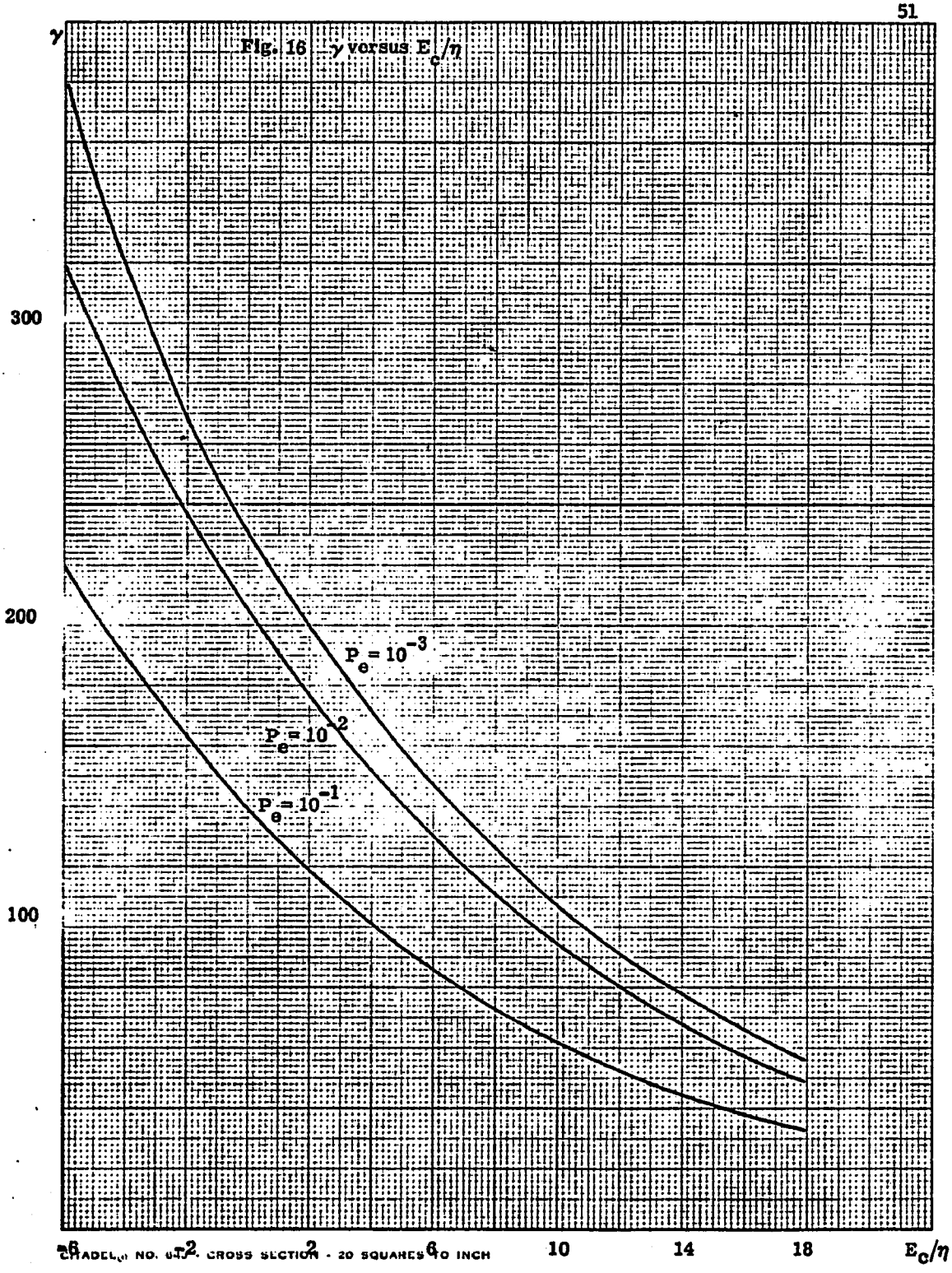


Fig. 15 Acquisition Decision Regions

Fig. 16 γ versus E_c/η



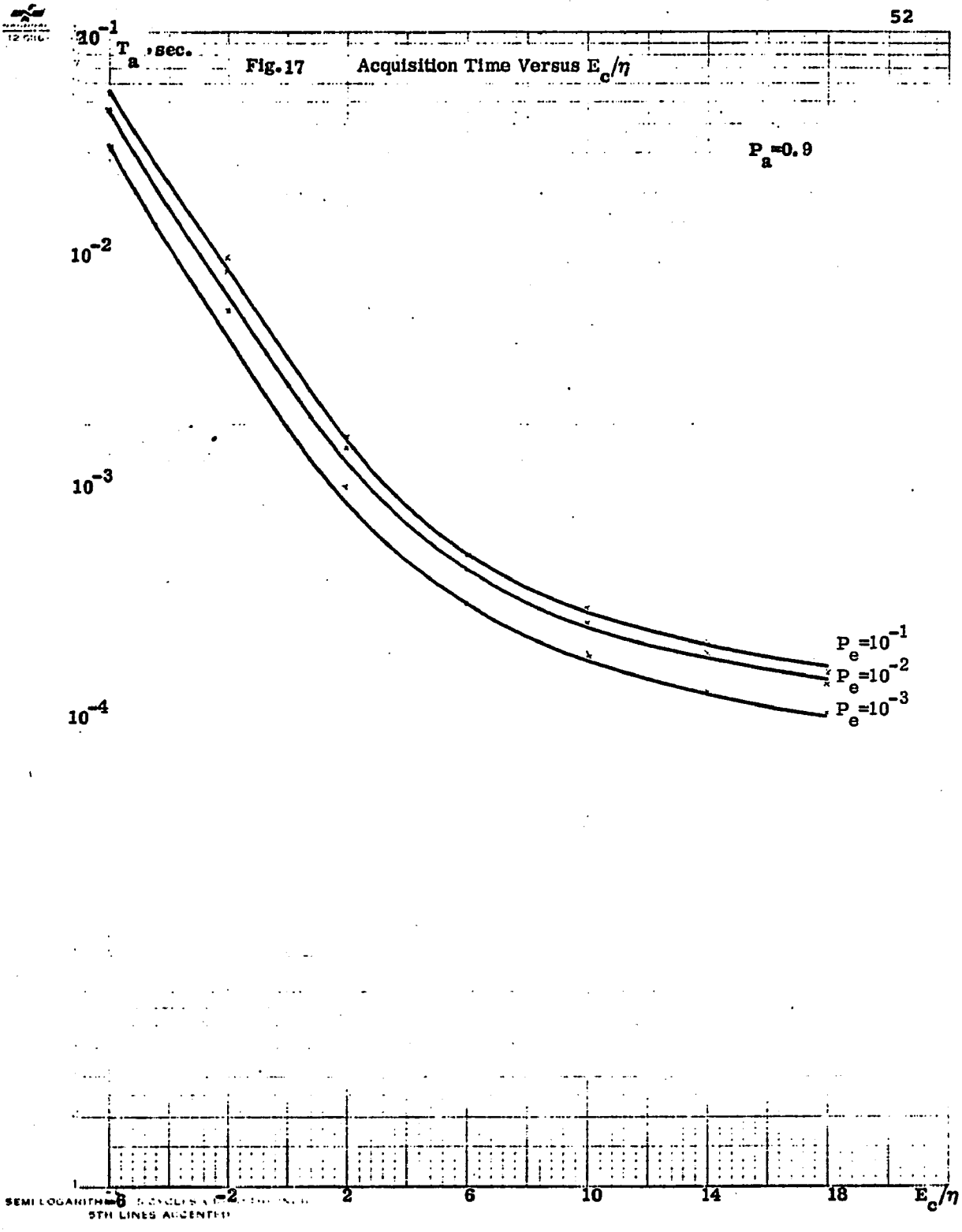


Fig. 17 Acquisition Time Versus E_c/η

L-Correlator Acquisition

Introduction. A block diagram of an L-correlator acquisition circuit is shown in Fig. 18 . As the name implies the circuit consists of L correlators operating in parallel. The received PN sequence, $V_c g(t-jT_c)$, is sent to L different multipliers where it is multiplied with every possible phase shift of itself and the result is then integrated for γ chips. At the end of the integration period the decision logic will select the largest output and so establish the phase of the incoming PN sequence.

L-Correlator Acquisition Performance in the Absence of Noise. It has been proven that the minimal number for a synchronism decision is (N+1) chips. Therefore the smallest possible acquisition will be:

$$T_a = (N+1)T_c \tag{1}$$

L-Correlator Acquisition Performance in the Presence of Noise. In the absence of noise the acquisition time for the L-correlator circuit was limited to $T_a = (N+1)T_c$. When noise is present at the input to the correlators the acquisition time will have to increase to keep the probability of making an error (false dismissal or false acquisition) small. Assume the incoming sequence to be $V_c g(t-jT_c)$ and the integration time $T = \gamma T_c$; then the output of any correlator i can be expressed as:

$$V_i = \int_0^{\gamma T_c} V_c g(t-jT_c) g(t-iT_c) dt \qquad (1)$$

$$+ \int_0^{\gamma T_c} n_w(t) g(t-iT_c) dt = S_{o_i}(t) + n_{o_i}(t)$$

where $i = 1,2, \ldots L-1$.

As it has been previously shown:

$$S_{o_i}(t) = \begin{cases} V_c \gamma T_c & \text{if } i = j \\ V_c T_c r_p(\gamma) & \text{if } i \neq j \end{cases}$$

when the bound to the partial autocorrelation function is:

$$r_p(\gamma) = \gamma\left(1 - \frac{\gamma}{L}\right) \qquad (2)$$

The noise term, $n_{o_i}(t)$, has also been previously shown to be gaussian, with 0 mean and variance $\sigma_o^2(\gamma T_c) = \frac{n}{2}\gamma T_c$.

Now, given that the received sequence is of the form $V_c g(t-jT_c)$ an error is said to be made if any output i, where $i = 1,2,\ldots L-1$ and $i \neq j$ is larger than the j^{th} output. Therefore:

$$P_e = P[V_1 > V_j \text{ OR } V_2 > V_j \text{ OR } \ldots \text{ OR } V_L > V_j] \qquad (3)$$

If we denote $[V_i > V_j]$ by ε_{ij} then

$$P_e = P[\varepsilon_{1j} \cup \varepsilon_{2j} \cdots \cup \varepsilon_{j-1,j} \cup \varepsilon_{j+1,j} \ldots \cup \varepsilon_{L,j}]$$

$$= P[\varepsilon_{1,j}] + P[\varepsilon_{2,j}] + \ldots + P[\varepsilon_{j-1,j}] + P[\varepsilon_{j+1,j}] + \ldots +$$

$$+ P[\varepsilon_{L,j}] - P[\varepsilon_{1,j} \cap \varepsilon_{2,j}] - P[\varepsilon_{1,j} \cap \varepsilon_{3,j}] - \ldots . \qquad (4)$$

Thus:

$$P_e < (L-1)P[V_o > V_j] \tag{5}$$

if we neglect all the intersections.

Equation (5) is called the Union Bound and is valid for all probabilities of error.

Equation 1 describes both, V_o and V_j, with the exception that

$$V_j = S_j + n_j = V_c\gamma T_c + n_{jo}(t) \text{ and}$$

$$V_o = S_o + n_o = V_c T_c \gamma (1-\frac{\gamma}{L}) + n_o(t) \tag{6}$$

where the only difference occurs in the signal terms.

An erroneous decision will be made only if the voltage V_o will be larger then the voltage V_j at time $T = \gamma T_c$ or:

$$\begin{aligned} P_e' &= P[V_o > V_j] \\ &= P[S_o + n_o > S_i + n_j] \\ &= P[S_j - S_o < n_o - n_j] \end{aligned} \tag{7}$$

A new noise term is defined as $n_\Delta(t)$ where:

$$n_\Delta(t) = n_o - n_j \tag{8}$$

The noise n_Δ is gaussian with zero mean and variance equal to:

$$\sigma_\Delta^2 = E[(n_o-n_j)^2] = E[n_o^2] + E[n_j^2] - 2E[n_o n_j] \tag{9}$$

where:

$$E[n_o^2] = E[n_j^2] = \frac{\eta}{2}\gamma T_c \tag{10}$$

Now: n_o and n_j have been defined as:

$$n_o = \int_o^{\gamma T_c} n_w(t)g(t)dt \tag{11a}$$

$$n_i = \int_o^{\gamma T_c} n_w(t)g(t-jT_c)dt \tag{11b}$$

Therefore:

$$E[n_o n_j] = E\left[\int_o^{\gamma T_c} d\lambda \int_o^{\gamma T_c} dt\ n_w(\lambda)n_w(t)g(\lambda)g(t-jT_c)\right]$$

$$= \frac{\eta}{2}\int_o^{\gamma T_c} dt\ g(t)g(t-jT_c) \tag{12}$$

The result of Eq. (12) is easily obtained upon realizing that

$$E[n_w(t)n_w(\lambda)] = \frac{\eta}{2}\delta(t-\lambda) \tag{13}$$

The integral of Eq. (12) is easily bounded and we obtain

$$E[n_o n_j] = \frac{\eta}{2}\gamma\left(1-\frac{\gamma}{L}\right)T_c \tag{14}$$

Using the result of Eq. (10) and Eq. (14), we can write Eq. (9) as

$$\sigma_\Delta^2 = \eta\gamma T_c - \eta\gamma\left(1-\frac{\gamma}{L}\right)T_c$$

$$= \eta\gamma T_c\left[1-\left(1-\frac{\gamma}{L}\right)\right] = \frac{\eta\gamma^2 T_c}{L} \tag{15}$$

Once the distribution function for n_Δ is known, the probability $P_e' = P[V_o > V_j]$ can be expressed as:

$$P_e' = \int_{\varepsilon}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_\Delta} e^{\frac{n^2}{2\sigma_\Delta^2}} dn = \frac{1}{2} \operatorname{erfc} \left[\frac{\varepsilon}{\sqrt{2}\sigma_\Delta} \right] \qquad (16)$$

In Eq. (16) the lower limit, ε, has been found by calculating

$$\varepsilon = S_j - S_o = V_c \gamma T_c - V_c \gamma (1 - \frac{\gamma}{L}) T_c = V_c T_c \frac{\gamma^2}{L} \qquad (17)$$

as required by Eq. (7). If σ_Δ^2 is replaced by the result of Eq. (15) and ε is replaced by the result of Eq. (17), then the $P[V_j < V_o]$ can be written as

$$P_e' = P[V_j < V_o] = \frac{1}{2} \operatorname{erfc} \left[\frac{E_c \gamma^2}{2\eta L} \right]^{1/2} \qquad (18)$$

and $$P_e < (L-1) P[V_j < V_o]$$

$$< (L-1) \frac{1}{2} \operatorname{erfc} \left[\frac{\gamma^2 E_c}{2\eta L} \right]^{1/2} \qquad (19)$$

If we choose $P_e = 10^{-2}$ and $\frac{E_c}{\eta} = -10$ db with a sequence of length $L = 2047$ then

$$\gamma \cong 400$$

which shows a speedy decision time even at negative E_s/η ratios.

Figure 19 shows γ as a function of E_c/η for $P_e = 6 \cdot 10^{-2}$, $8 \cdot 10^{-3}$ and $7 \cdot 10^{-5}$.

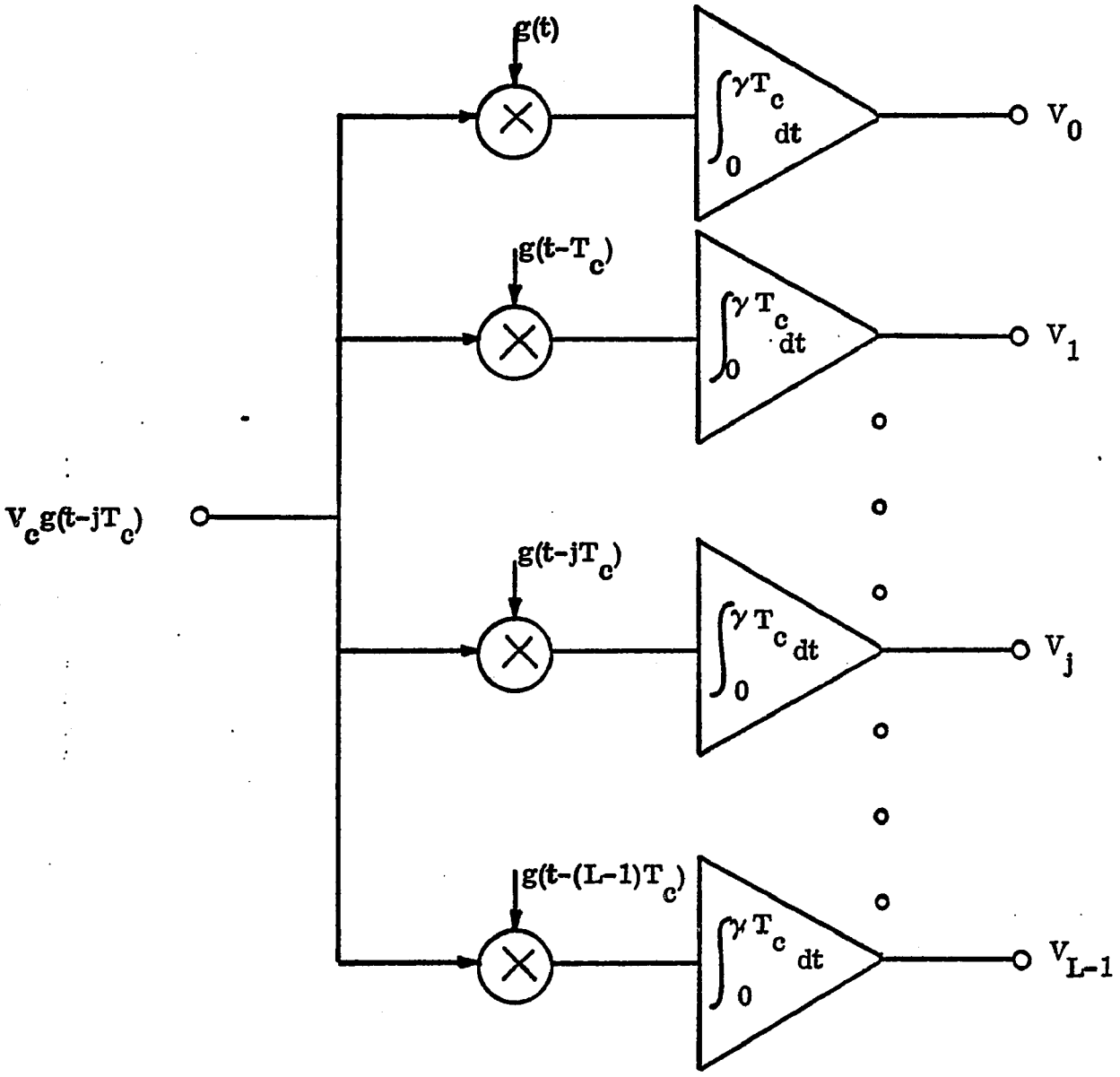


Fig. 18 An L Correlator Acquisition Circuit

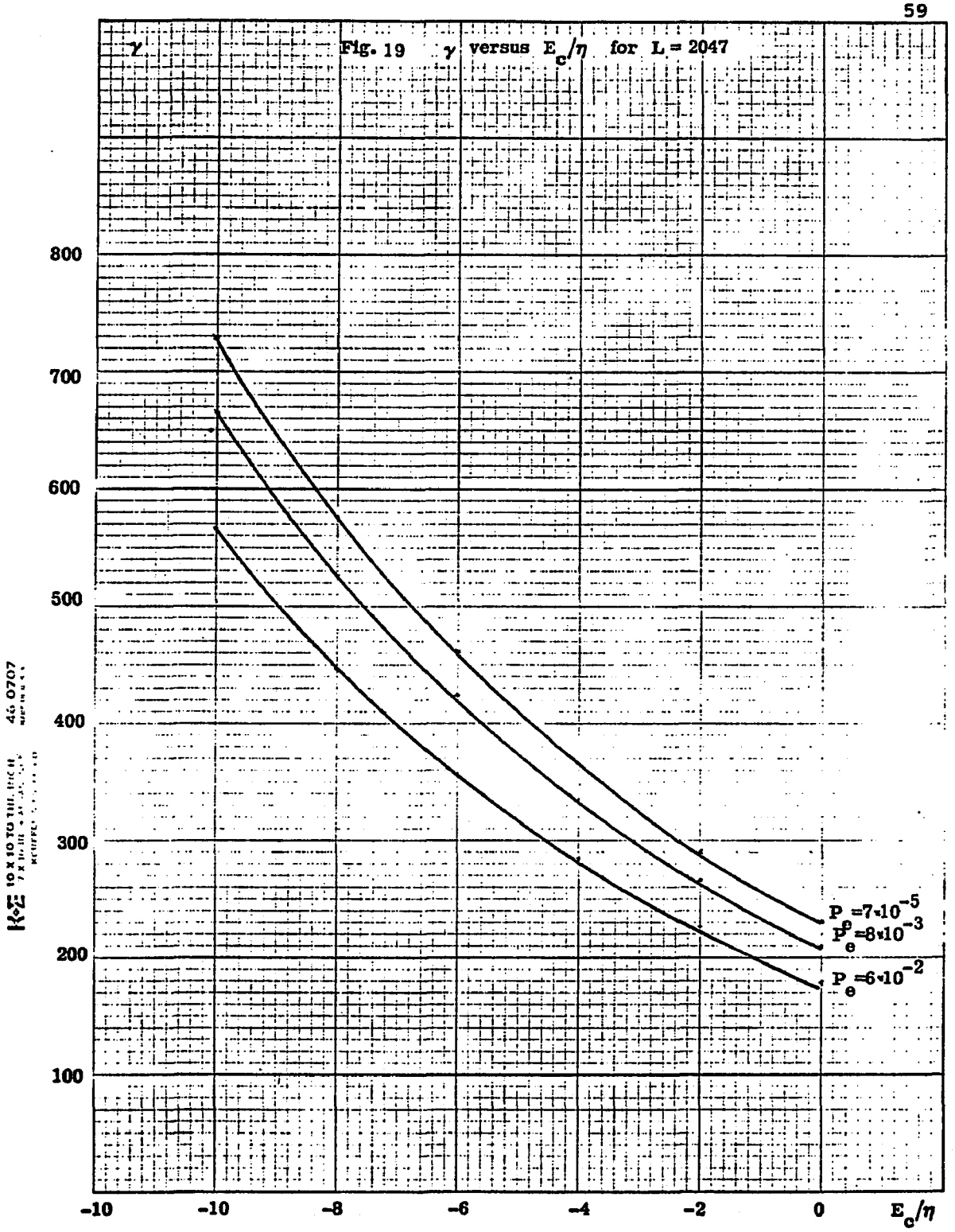


Fig. 19 γ versus E_c/η for L = 2047

Sliding Correlator Acquisition.

Introduction. A block diagram of the sliding correlator acquisition circuit is shown in Fig.20. The received PN sequence, $V_c g(t-jT_c)$ is compared to a locally generated sequence, $g(t- iT_c)$. The two sequences multiply each other and the result is integrated for a time $T = \gamma T_c$. The output of the integrator is compared to a threshold voltage $V_T(t)$ and, if at time $t = T = \gamma T_c$ the threshold voltage V_T is higher in value then the output of the integrator $V_o(t)$, the locally generated PN sequence is delayed by T_c (changes its phase by one chip). The correlation process is restarted and it is repeated until the integrator's output, $V_o(t)$, is higher than $V_T(t)$ at $T = \gamma T_c$ (phase synchronism is attained). The local PN sequence generator starts with any random phase and finds the proper phase by "sliding" through all of the possible phases, hence its name, sliding correlator acquisition.

Sliding Correlator Acquisition Performance in the Absence of Noise. It has been shown that even under no noise conditions the correlator needs $> N$ chips to make a synchronism/no synchronism decision. Therefore, the minimum decision time, $T_a = (N+1)T_c$

The phase of the local PN sequence generator, $g(t-iT_c)$, can be as far away as (L-1) from the phase of the received PN sequence generator. Therefore, the upper limit on the acquisition time, T_a, in the absence of noise is

$$T_a = (L-1)NT_c \qquad (1)$$

An improvement to this upper limit on T_a can be made by using an SW (synch worthiness) type of decision. This would take advantage of the fact that the phase of the local sequence can be changed as soon as a disagreement is noticed between the received PN sequence and the locally generated PN sequence. If a one level SW is used (examine only the first chips), then a significant saving can be achieved. This is a result of the probability of disagreement of the first bits being equal to nearly 0.5. Assuming that the correlation will continue for a time $T = (N+1)T_c$ if the first bits of the received PN sequence and the locally generated PN sequence are identical, then the upper limit of the acquisition time T_a is given by

$$T_a \sim \frac{L-1}{2}(N+1)T_c + \frac{L-1}{2}T_c = \frac{L-1}{2}(N+2)T_c \qquad (2)$$

It is possible to take into consideration not only the first disagreement, but also the second, third, ... , $(N-1)^{st}$ with a corresponding decrease in the upper limit of T_a.

To illustrate this procedure consider the PN sequence ...1110010... formed from a 3-stage sequence generator. Let us assume that initially the sequences are not synchronized and appear as in Fig. 21a. Note that after a single chip comparison, the integral of the product $g(t-jT_c)g(t-kT)$ is equal to $-T_c$ and is less than the negative input to the comparator $(\gamma-1)T_c = 0$. The integrator is therefore reset and the PN

sequence generator is delayed by 1 chip as shown in Fig.21b. Once again, the first chips are dissimilar so we once more delay the generator. This procedure continues, as illustrated in Fig. 21, until, as shown in (g), we observe that after $13T_c$ synchronization is obtained. Note that other initial starting points yield a smaller acquisition time. The average acquisition time can be readily computed from Fig. and shown to be $\bar{T}_a = 9T_c$ for this example.

Note that in any case the acquisition time obtained from this system is greater than for the shift-register or L-correlator acquisition techniques. However, we shall show that the sliding-correlator is robust and performs better than the shift register correlator when the signal-to-noise ratio is very small and when a jammer is present.

Sliding Correlator Acquisition Performance in the Presence of Noise. The sliding correlator's operation in the presence of noise is very similar to the operation of the shift register acquisition circuit's operation, with the exception that now, in order to synchronize, a new starting vector is not loaded any longer but the phase of the local PN sequence generator is changed by T_c. The threshold voltage $V_T(T)$ is again given by

$$V_T = V_c \gamma T_c - \frac{1}{2} V_c T_c \frac{\gamma^2}{L} \qquad (1)$$

The output of the correlator is equal to

$$V_o = \int_o^{\gamma T_c} \left[g(t-jT_c) + n_w(t) \right] g(t-iT_c)\,dt$$

$$= \int_o^{\gamma T_c} g(t-jT_c)\,g(t-iT_c)\,dt + \int_o^{\gamma T_c} n_w(t)\,g(t-iT_c)\,dt$$

$$= S_o(t) + n_o(t) \qquad (2)$$

where, again

$$S_o(t) = \begin{cases} V_c \gamma T_c & \text{if } j = 1 \\ V_c T_c r_p(\gamma) & \text{if } j \neq i \end{cases} \qquad (3)$$

and

$$r_p(\gamma) = \gamma\left(1-\frac{\gamma}{L}\right) \qquad (3)$$

The noise term, $n_o(t)$, is gaussian with 0 mean and variance $\sigma_o^2(\gamma T_c) = \frac{\eta}{2}\gamma T_c$. Since the threshold voltage $V_T(T)$ was set midway between the two signals, the probability of a false dismissal is the same as the probability of a false acquisition and the resulting probability of error, P_e, is given by

$$P_e = \int_\varepsilon^\infty \frac{1}{\sqrt{2\pi}\sigma_o} e^{-\frac{n^2}{2\sigma_o^2}}\,dn = \frac{1}{2}\,\mathrm{erfc}\left[\frac{\varepsilon(T)}{\sqrt{2}\sigma_o}\right]$$

$$= \frac{1}{2}\,\mathrm{erfc}\left[\frac{\gamma}{L}\sqrt{\gamma\frac{E_c}{4\eta}}\right] \qquad (5)$$

The integration time, $T = \gamma T_c$, is significantly larger now than in the absence of noise and therefore the acquisition time T_a is also considerably larger. A simple enhancement, similar to the synch worthiness indicator previously described

can, however, significantly lower the acquisition time, T_a, by detecting very early a large number of wrong phases and going on to a new correlation cycle before the old one is fully completed.

This elimination of improbable phases leads to a significant improvement in performance and can also be applied to other acquisition methods such as the shift register acquisition.

Conclusion

This section has shown how the bound obtained for the partial autocorrelation function can be applied to a number of practical systems. The application of this bound has the effect of minimizing the acquisition time T_a while at the same time guaranteeing a desired level of performance which is specified by the probability of error (false dismissal or false acquisition). Extensive computer simulations have shown the bound to be very tight and therefore the values for the minimum acquisition times which have been found in the previous sections will be very close to the actual optimal values. It is only after considering the tightness of the bound together with the ease of applying it to real life acquisition techniques and considering the numerical results obtained that the significance of the work done becomes apparent. The spread spectrum system designer will now be able to optimize any PN

correlation based acquisition circuit as well as make a choice of the best system to use for a given range of signal to noise ratios.

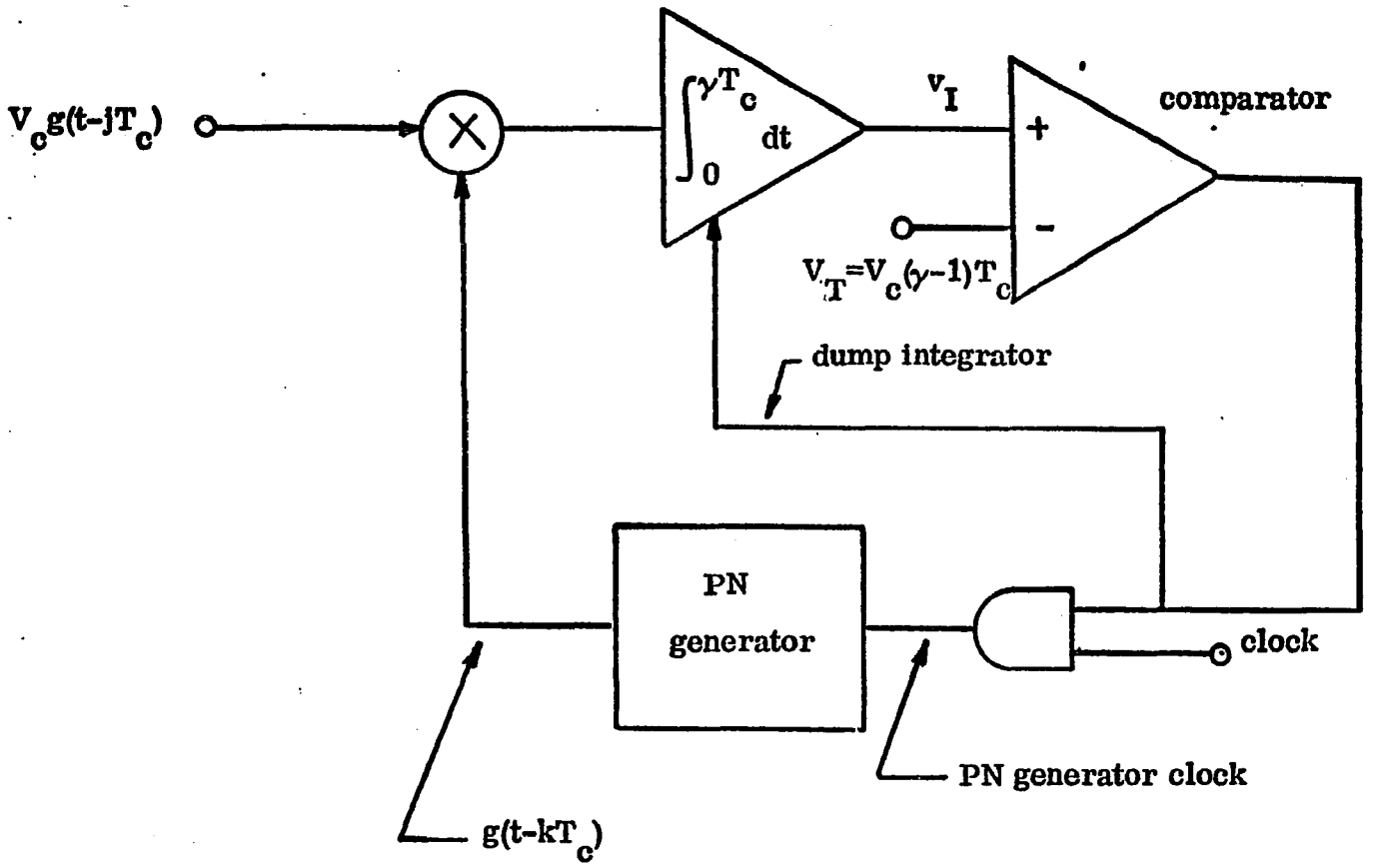


Fig. 20 Sliding Correlator Acquisition

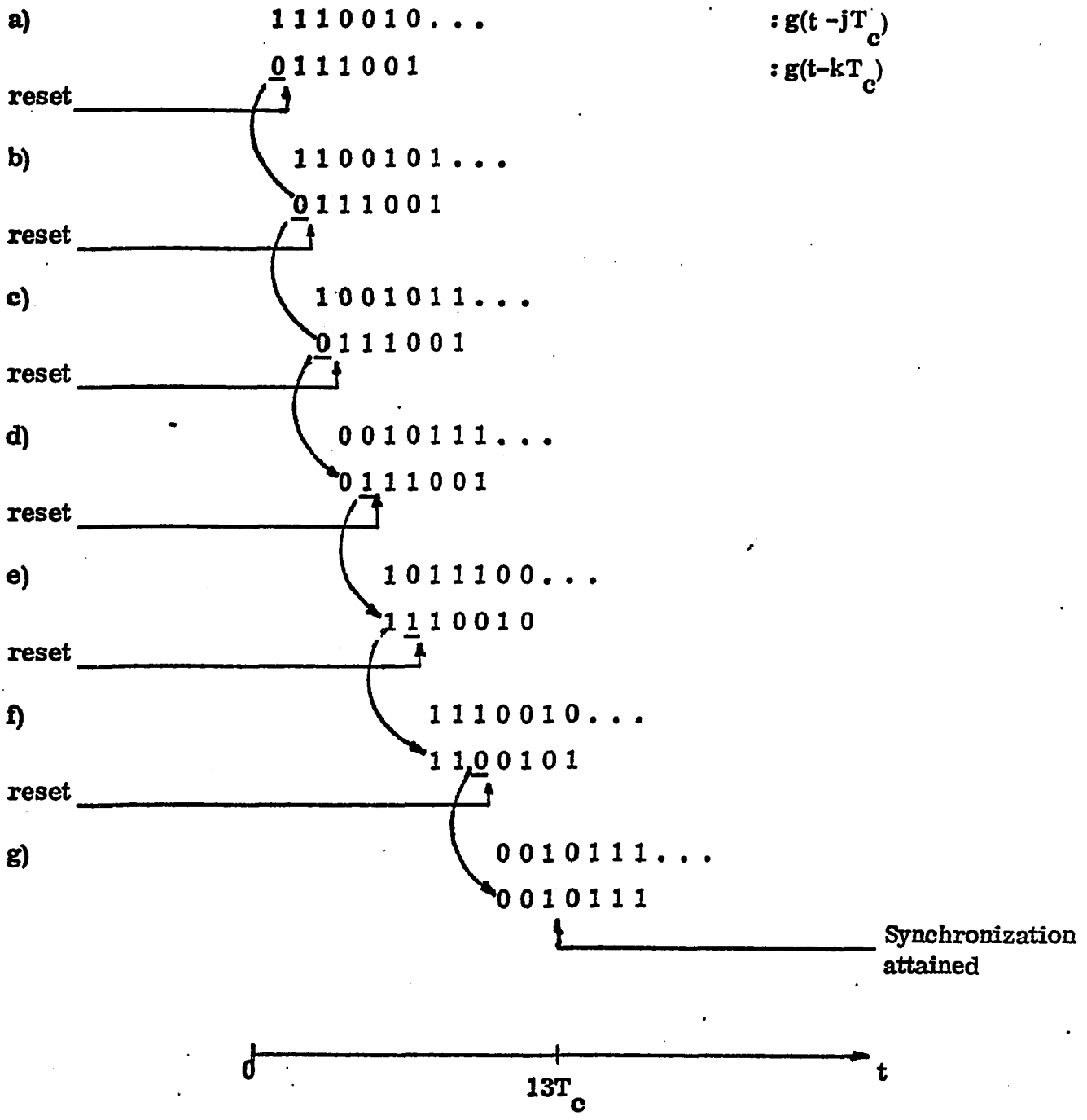


Fig. 21 Example of Acquisition Using Sliding Correlator

Spread Spectrum Signal Acquisition Aided by Recursive Estimation

Introduction

Throughout this work it has been assumed that the acquisition of a spread spectrum signal consists of correctly estimating the phase of a received PN sequence of arbitrary length 2^N-1, where N refers to the number of delay stages in the generating shift register or, equivalently, to the degree of the generating polynomial h(x). The acquisition process has consisted of repeatedly comparing the received PN sequence, $V_c g(t-jT_c)$ with a local generated PN sequence, $g(t-iT_c)$, until the phases iT_c and jT_c were found to be equal. At that point the locally generated PN sequence was said to be synchronized to the received PN sequence and the acquisition process was ended. The synchronism/no synchronism decision was based on an autocorrelation process or, since the correlation did not have to last for a full period, this decision was based on a partial autocorrelation process. The partial autocorrelation function of a PN sequence is not nearly as well behaved as the autocorrelation function of PN sequences, it being a function of the particular starting phases i and j, the generating polynomial h(x), and the length of the correlation γT_c. To circumvent all of these difficulties and be able to use a partial autocorrelation process universally, a bound has been found to the partial autocorrelation function, $r_p(\gamma)$,

where

$$r_p(\gamma) = \gamma(1-\frac{\gamma}{L})T_c \geq \int_0^{T_c\gamma} dt\, g(t-jT_c)g(t-iT_c) \quad \text{if } i \neq j \quad (1)$$

and γ is the number of chips being correlated.

As shown previously, the output of the correlator is, in general,

$$V_o(t) = \int_0^{\gamma T_c} dt[V_c g(t-jT_c) + n_w(t)]g(t-iT_c) \quad (2)$$

which can be said to consist of a signal term $S_o(t)$ and a noise term $n_o(t)$. The noise term has been previously shown to be gaussian with zero mean and variance $\sigma_o^2(\gamma T_c) = \frac{\eta}{2}\gamma T_c$ where $\frac{\eta}{2}$ is the power spectral density of the thermal noise which is added to the received signal. The signal term, $S_o(t)$ is:

$$S_o(t) = \int_0^{\gamma T_c} dt\, V_c g(t-jT_c)g(t-iT_c)$$

$$= \begin{cases} V_c\gamma T_c & \text{if } i = j \\ V_c T_c \gamma(1-\frac{\gamma}{L}) & \text{if } i \neq j. \end{cases} \quad (3)$$

The synchronism/no synchronism decision is now seen only to involve the choice of which signal term, $S_o(t)$ is present at the correlator's output. Due to the presence of noise this decision can only be made with a specified accuracy. If, as shown in Fig. 22, we wait for a time $T = \gamma_1 T_c$ before making a decision, then we can define two voltage regions such that if the correlator's output falls in region I we decide synchronism is attained and if we fall in region II, we decide

the two PN sequences are not synchronized. The probability of an erroneous decision is equal to

$$P_e = P[V_o(\gamma_1 T_c) > V_T = V_c\gamma_1 T_c - \varepsilon / S_o(t) \leq V_c T_c \gamma_1 (1-\frac{\gamma_1}{L})] +$$

$$+ P[V_o(\gamma_1 T_c) < V_T = V_c\gamma_1 T_c - \varepsilon / S_o(t) = V_c T_c \gamma_1]$$

$$= P[n_o(\gamma_1 T_c) > \varepsilon]. \qquad (4)$$

Note that in Fig. 22 and Eq. (4) the assumption has been made that the two errors are equally likely and therefore the decision boundary is located halfway between the two signals or

$$\varepsilon = \frac{1}{2}[V_c\gamma_1 T_c - V_c T_c \gamma_1 (1-\frac{\gamma_1}{L})] = \frac{1}{2} V_c T_c \frac{\gamma_1^2}{L} \qquad (5)$$

The probability of error of Eq. (4) becomes then

$$P_e = P[n_o(\gamma_1 T_c) > \frac{1}{2} V_c T_c \frac{\gamma_1^2}{L}] =$$

$$= \int_{V_c T_c \frac{\gamma_1^2}{2L}}^{\infty} dn \frac{1}{\sqrt{2\pi}\sigma_o(\gamma T_c)} e^{-\frac{n^2}{2\sigma_o^2(\gamma T_c)}} =$$

$$= \frac{1}{2} \operatorname{erfc} \left[\frac{\gamma_1}{L}\sqrt{\frac{\gamma_1 E_c}{4\eta}}\right] \qquad (6)$$

This shows that, in principle, the probability of error P_e can be made very small by waiting γ_1 chips, or given a desired P_e, γ_1 can easily be found as a function of $\frac{E_c}{\eta}$. It is possible, therefore, to correlate for a time $T = \gamma_1 T_c$, then

examine the output voltage of the correlator $V_o(t)$ and make a synchronism/no synchronism decision based on where $V_o(t)$ falls with respect to the threshold voltage $V_T(\gamma_1 T_c)$ where $V_T(\gamma_1 T_c)$ has been defined as:

$$V_T(\gamma_1 T_c) = V_c \gamma_1 T_c - \frac{1}{2} V_c T_c \frac{\gamma_1^2}{L} = V_c T_c \gamma_1 \left[1 - \frac{\gamma_1}{2L}\right] \tag{7}$$

The Early Decision Process. At times it may not be necessary to wait the full γ_1 chips to make a decision. The only requirement is that at any time a decision is to be made, the probability of error, P_e, be kept constant. If we are to attempt making a decision after, let's say γ chips, then the threshold voltages must be set such that

$$P_e = P[n_o(\gamma T_c) > \varepsilon'] = \frac{1}{2} \operatorname{erfc} \left[\frac{\varepsilon'}{\sqrt{2}\sigma_o(\gamma T_c)}\right] \tag{8L}$$

Note that Eq. (8) is identical to Eq. (4) with the exception of the pertinent parameters (i.e. the decision mechanism is identical). But if the probability of error, Pe, is to be kept constant, then:

$$P_e = \int_{\varepsilon}^{\infty} dn \; P_{n_o(\gamma_1 T_c)}(n) = \int_{\varepsilon'}^{\infty} dn \; P_{n_o(\gamma T_c)}(n)$$

or

$$\frac{1}{2} \operatorname{erfc} \left[\frac{\varepsilon}{\sqrt{2}\sigma_o(\gamma_1 T_c)}\right] = \frac{1}{2} \operatorname{erfc} \left[\frac{\varepsilon'}{\sqrt{2}\sigma_o(\gamma T_c)}\right] \tag{9}$$

This implies that the two arguments of the erfc function have to be equal, and:

$$\frac{\varepsilon}{\sqrt{2}\sigma_o(\gamma_1 T_c)} = \frac{\varepsilon'}{\sqrt{2}\sigma_o(\gamma T_c)}$$

or

$$\varepsilon' = \varepsilon \frac{\sigma_o(\gamma T_c)}{\sigma_o(\gamma_1 T_c)} = \frac{V_c T_c}{2L} \gamma_1 \sqrt{\gamma\gamma_1} \tag{10}$$

Eq. (10) finds the parameter ε' such that at any time $T = \gamma T_c$ a decision can be made. It is of interest to note that at any given $T = \gamma T_c$ the "distance" between the two signals $S_o(t)$ is:

$$V_c \gamma T_c - V_c T_c \gamma (1-\frac{\gamma}{L}) = V_c T_c \frac{\gamma^2}{L} \tag{11}$$

and at time $T = \gamma_1 T_c$ this equals precisely 2ε. At any arbitrary time $T = \gamma T_c$ where $\gamma < \gamma_1$ however, the voltage corresponding to $2\varepsilon'$ is given by

$$2\varepsilon' = 2\left[\frac{V_c T_c}{2L} \gamma_1 \sqrt{\gamma_1\gamma}\right] = \frac{V_c T_c}{L} \gamma_1 \sqrt{\gamma_1\gamma} \tag{12}$$

Since $\gamma < \gamma_1$ we can define ℓ such that

$$\ell = \frac{\gamma_1}{\gamma} > 1 \quad \text{or} \quad \gamma_1 = \ell\gamma \tag{13}$$

and Eq. (12) becomes

$$2\varepsilon' = \frac{V_c T_c}{L} \ell\gamma \sqrt{\ell\gamma^2} = V_c T_c \frac{\gamma^2}{L} \ell\sqrt{\ell} \tag{14}$$

Since $\ell > 1$ it follows that

$$2\varepsilon' = V_c T_c \frac{\gamma^2}{L} \ell\sqrt{\ell} > V_c T_c \frac{\gamma^2}{L} \tag{15}$$

Based on Eq. (15), Fig. 22 shows the decision regions at any given $\gamma < \gamma_1$. After examining the decision regions at a time $T = \gamma T_c < \gamma_1 T_c$ the following can be concluded:

If at time $T = \gamma T_c$ the correlator's output voltage equals V_o the following decisions can be made with a given probability of error, P_e: 1) Synchronization has been attained (i=j) if

$$V_o > V_S = V_c T_c \gamma (1-\frac{\gamma}{L}) + \varepsilon'$$

2) Synchronization will not be attained ($i \neq f$) if:

$$V_o < V_n = V_c \gamma T_c - \varepsilon'$$

3) No decision can be made; continue the correlation if:

$$V_n < V_o < V_S$$

where $\varepsilon'(\gamma)$ has been defined in Eq. (10) as:

$$\varepsilon' = \frac{V_c T_c}{2L} \gamma_1 \sqrt{\gamma\gamma_1} \quad \text{and } V_n(\gamma) \text{ and } V_S(\gamma) \text{ become:}$$

$$V_n = V_c \gamma T_c - \varepsilon' = V_c T_c [\gamma - \frac{\gamma_i \sqrt{\gamma\gamma_1}}{2L}] \tag{16a}$$

$$V_s = V_c T_c \gamma (1-\frac{\gamma}{L}) + \varepsilon' =$$

$$= V_c T_c [\gamma(1-\frac{\gamma}{L}) + \frac{\gamma_1 \sqrt{\gamma\gamma_1}}{2L}] \tag{16b}$$

If at a time $T = \gamma T_c$ no decision can be made, that implies

either that we are synchronized and $-\varepsilon' < n < V_s - V_c\gamma T_c$ or that we are not synchronized and $-[V_cT_c\gamma(1-\frac{\gamma}{L}) - V_n] < n < \varepsilon'$.

Figures 23 and 24 show a graphic representation of the probability of no decision as described by the above inequalities.

The noise .. signal $n_o(\gamma T_c)$ is gaussian with zero mean which implies its distribution to be symmetric about the zero axis and therefore the shaded areas of Fig. 23 and Fig. 24 are equal. This can also be seen from Fig. 22 and may be written as:

$$\int_{-\varepsilon'}^{V_s - V_c\gamma T_c} dn \frac{1}{\sqrt{2\pi}\sigma_o(\gamma T_c)} e^{-\frac{n^2}{2\sigma_o^2(\gamma T_c)}} =$$

$$= \int_{-(V_cT_c\gamma(1-\frac{\gamma}{L}) - V_n)}^{\varepsilon'} dn \frac{1}{\sqrt{2\pi}\sigma_o(\gamma T_c)} e^{-\frac{n^2}{2\sigma_o^2(\gamma T_c)}} \qquad (17)$$

The upper and lower limit of the above integrals respectively are a function of γ and for some values of γ they can be further simplified as follows: Taking advantage of the fact that V_s is given by Eq. (16b) to be $V_s = \varepsilon' + V_cT_c\gamma(1-\frac{\gamma}{L})$, then: $V_s - V_c\gamma T_c = \varepsilon' + V_cT_c\gamma(1-\frac{\gamma}{L}) - V_cT_c\gamma = \varepsilon' - V_cT_c\frac{\gamma^2}{L} \approx \varepsilon'$

$$\text{if } V_cT_c\frac{\gamma^2}{L} << \varepsilon' \qquad (18)$$

Again, taking advantage of the fact that V_n is given by Eq. (16a) to be:

$$V_n = V_c\gamma T_c - \varepsilon'$$

then:

$$-[V_cT_c\gamma(1-\tfrac{\gamma}{L}) - V_n] = -[V_cT_c\gamma(1-\tfrac{\gamma}{L}) - (V_c\gamma T_c-\varepsilon')]$$

$$= -[V_cT_c\,\tfrac{\gamma^2}{L}+\varepsilon'] \approx -\varepsilon' \quad \text{if } V_cT_c\,\tfrac{\gamma^2}{L} << \varepsilon' \tag{19}$$

The above approximations can be said to hold if

$$V_cT_c\,\frac{\gamma^2}{L} << \frac{V_cT_c}{2L}\,\gamma_1\sqrt{\gamma\gamma_1} \tag{20}$$

or

$$\gamma^2 << \frac{1}{2}\,\gamma_1\sqrt{\gamma\gamma_1}$$

$$\gamma << .63 \tag{21}$$

for all γ which satisfy Eq. (20) the probability of not making a decision becomes:

$$P_{nd} = P[V_n < n < V_s] < P[-\varepsilon' < n < \varepsilon'] \tag{22}$$

The inequality holds since

$$V_s - V_n = [V_cT_c\gamma(1-\tfrac{\gamma}{L})+\varepsilon'] - [V_c\gamma T_c-\varepsilon']$$

$$= 2\varepsilon' - V_cT_c\,\frac{\gamma^2}{L} \tag{23}$$

and the result is clearly an upper bound.

This inequality can also be clearly seen to hold by examining the graphical representations of Figs. 22, 23 and 24. Due to the fact that the actual probability of no decision, P_{nd}, is less than the value specified by the upper bound, the

results obtained will be slightly pessimistic and the acquisition system's performance will be as guaranteed or better.

Now, at any given time $T = \gamma T_c$ an erroneous synchronism/non-synchronism decision will be made if in fact the noise $n_o(\gamma T_c)$ is larger than ε' given that no synchronism exists, or if the noise $n_o(\gamma T_c)$ is smaller than $-\varepsilon'$ given that synchronism exists. Therefore,

$$P_e = P[n_o(\gamma T_c) > \varepsilon' / \text{no synchronism}] + \\ + P[n_o(\gamma T_c) < -\varepsilon' / \text{synchronism}]. \qquad (24)$$

But the noise term, $n_o(\gamma T_c)$, is gaussian with zero mean, and if we let P_s = probability of synchronism and P_{ns} = probability of no synchronism be independent of the noise probability, then

$$P_e = P[n_o(\gamma T_c) > \varepsilon'] + P[n_o(\gamma T_c) < -\varepsilon'] \\ = 1 - P[-\varepsilon' < n_o < \varepsilon'] = 1 - P_{nd} \qquad (25)$$

Equation (25) shows the simple relation between the probability of error, P_e, which is always specified by the designer, and the upper bound on the probability of no decision, P_{nd}, which is valid only for $\gamma << .63\gamma_1$.

The probability of a decision having been made in a time $T = \gamma_o T_c$ is the probability of a decision having been made at time $T = NT_c$ or $T = (N+1)T_c$ or ... or $T = \gamma_o T_c$. This is equivalent to $1 - [P_{nd}$ at $T=NT_c$ and P_{nd} at $T=(N+1)T_c$ and ...

and P_{nd} at $T = \gamma_o T_c$] or:

$$P_d/T \leq \gamma_o T_c = \sum_{\gamma=1}^{\gamma_o - N} P_d/T = \gamma T_c =$$

$$= 1 - P_{nd}/T = \gamma_o T_c$$

$$= 1 - \prod_{\gamma=1}^{\gamma_o - N} P_{nd}/T = \gamma T_c \qquad (26)$$

where γT_c are the observation times, and P_{nd} stands for probability of no decision. At the first observation time, $T = NT_c$, the noise $n_o(\gamma T_c)$ will be gaussian with zero mean, and it will have a value $n_o(NT_c)$ volts. At the second observation time, $T = (N+1)T_c$, the noise will again be gaussian but its mean will now be $n_o(NT_c)$. The probability density function for the noise signal at time $T = (N+1)T_c$ is

$$P_n(n) = \frac{1}{\sqrt{2\pi}\sigma_o((N+1)T_c)} e^{-\frac{1}{2}\frac{[n-n_o(NT_c)]^2}{2\sigma_o^2((N+1)T_c)}} \qquad (27)$$

Fig. 25 shows a graphical representation of this probability density function. It can clearly be seen how the probability of no decision in this case (the shaded area) is smaller than the equivalent probability as given by a zero mean gaussian noise of the same variance. This in turn implies that if the assumption of independence of the measurements taken at times $T = NT_c, (N+1)T_c, \ldots \gamma T_c$ is made it will serve as an upper bound, further guaranteeing the predicted result. The probability of decision at or before time $T = \gamma_o T_c$, P_o, is then

given by:

$$P_o = P_d/T \leq \gamma_o T_c \geq 1 - \prod_{\gamma=1}^{\gamma_o - N} P_{nd}/T = \gamma T_c$$

$$= 1 - \prod_{\gamma=1}^{\gamma_o - N} (1-P_e)^{\gamma} \tag{28}$$

Eq. (28) is derived from Eq. (26) using the assumption of independence justified above and the result of Eq. (25). Rearranging Eq. (28) and taking logarithms of both sides, we obtain:

$$\ln(1-P_o) = \sum_{\gamma=1}^{\gamma_o - N} \gamma \ln(1-P_e)$$

$$= \frac{(\gamma_o - N)(\gamma_o - N + 1)}{2} \ln(1-P_e) \tag{29}$$

Assuming $\gamma_o - N >> 1$, then

$$\gamma_o \approx 2 \frac{\ln(1-P_o)}{\ln(1-P_e)} + N \text{ for } \gamma_o << .63\gamma_1 \tag{30}$$

Figure 26 plots γ_o versus P_o for several different values of P_e.

Conclusion

It is only through this recursive estimation approach that the full power of the upper bound to the autocorrelation function of a PN sequence is applied to the problem of PN

sequence acquisition. By correctly recognizing the existence of three decision regions prior to $t = .63\gamma_1 T_c$ it is possible now to dismiss or acquire the PN signal much earlier. Again, this technique is independent of any particular acquisition scheme and it can be applied to any system which employs a correlation type decision mechanism and uses PN sequences. It is estimated that at a modest increase in the hardware complexity this approach to the acquisition process could drastically improve the performance of any system, with those systems which require a large number of sequential correlations having most to benefit.

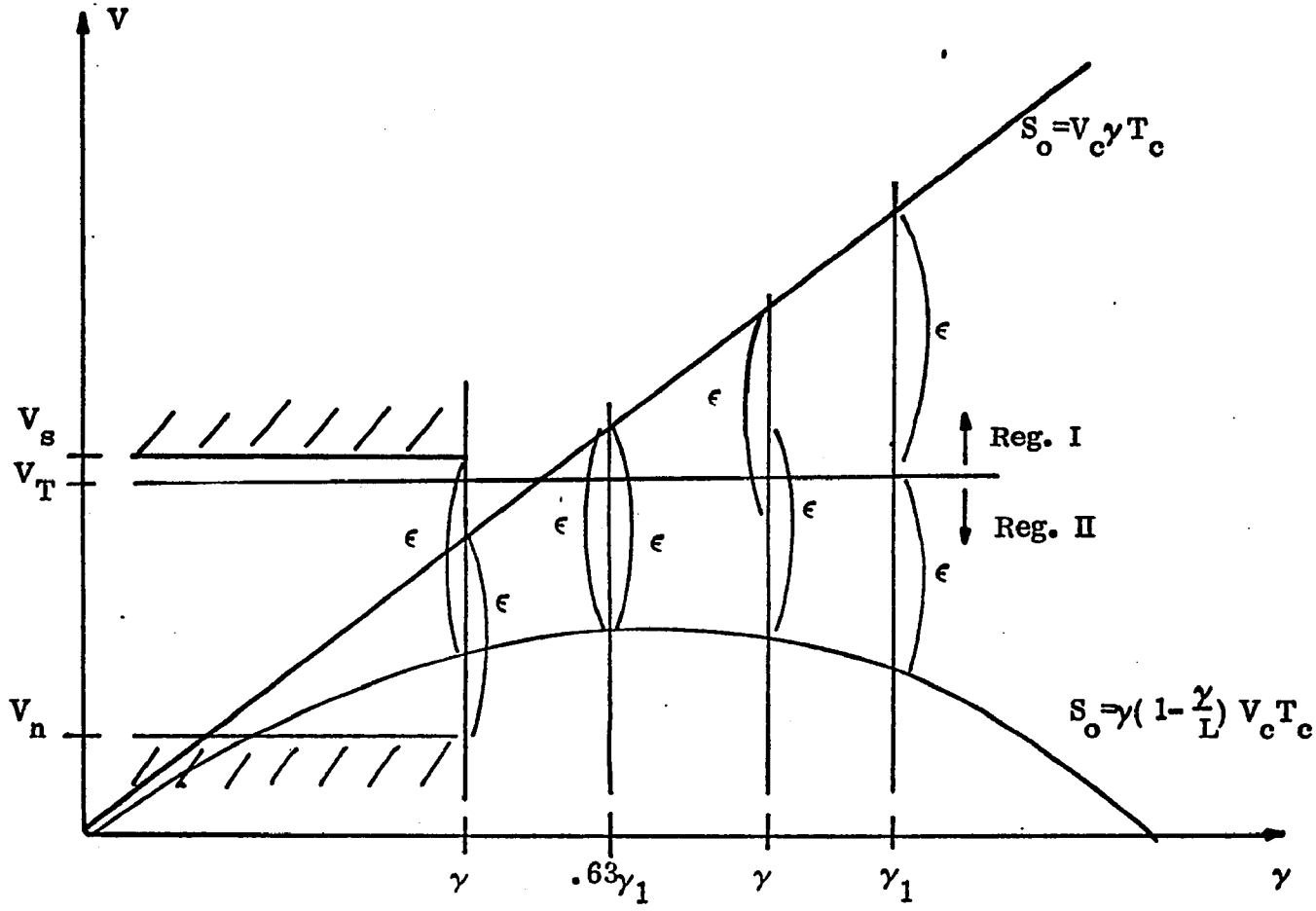


Fig. 22 Synchronism/No Synchronism Decision Regions for PN Sequence Acquisition

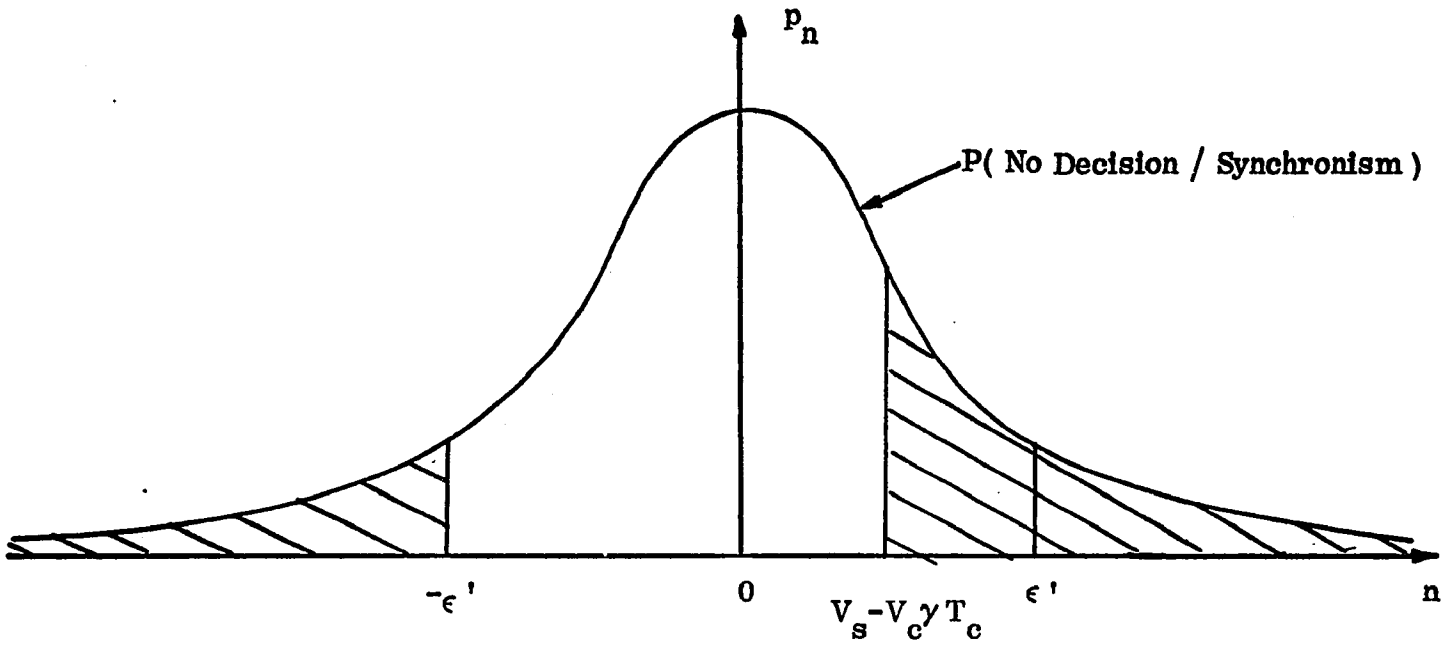


Fig. 23 Probability of No Decision / Synchronism

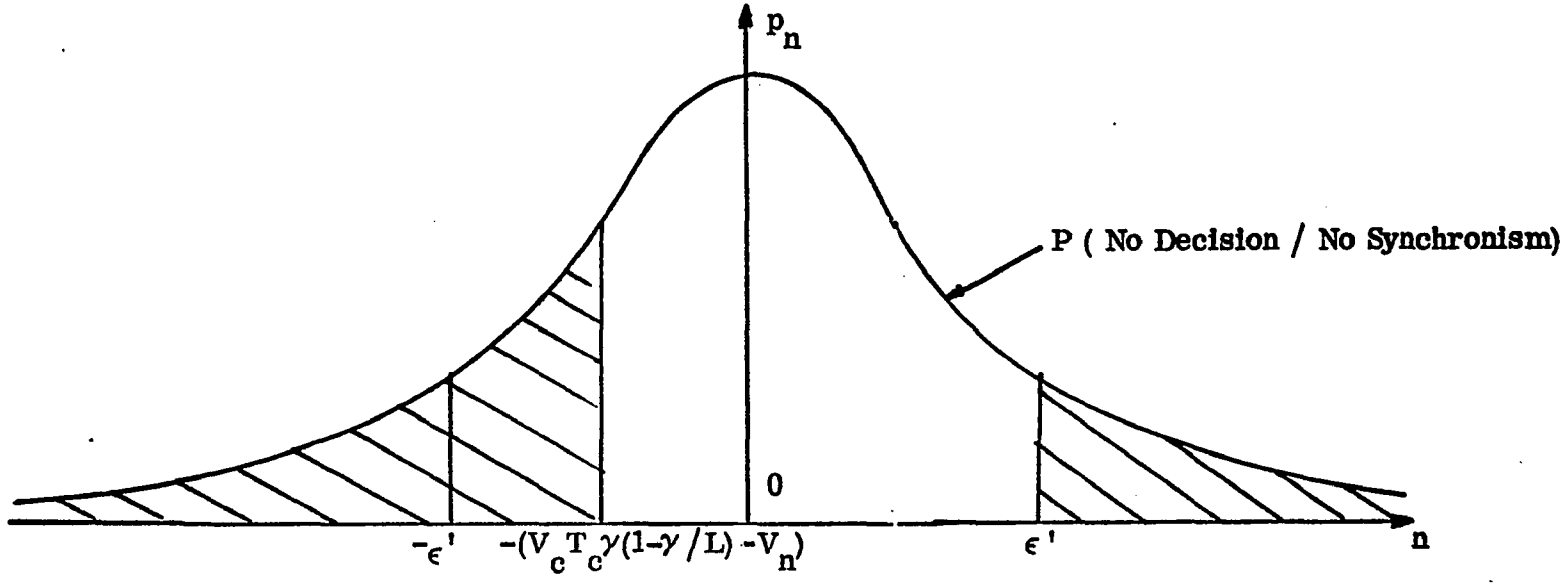


Fig. 24 Probability of No Decision / No Synchronism

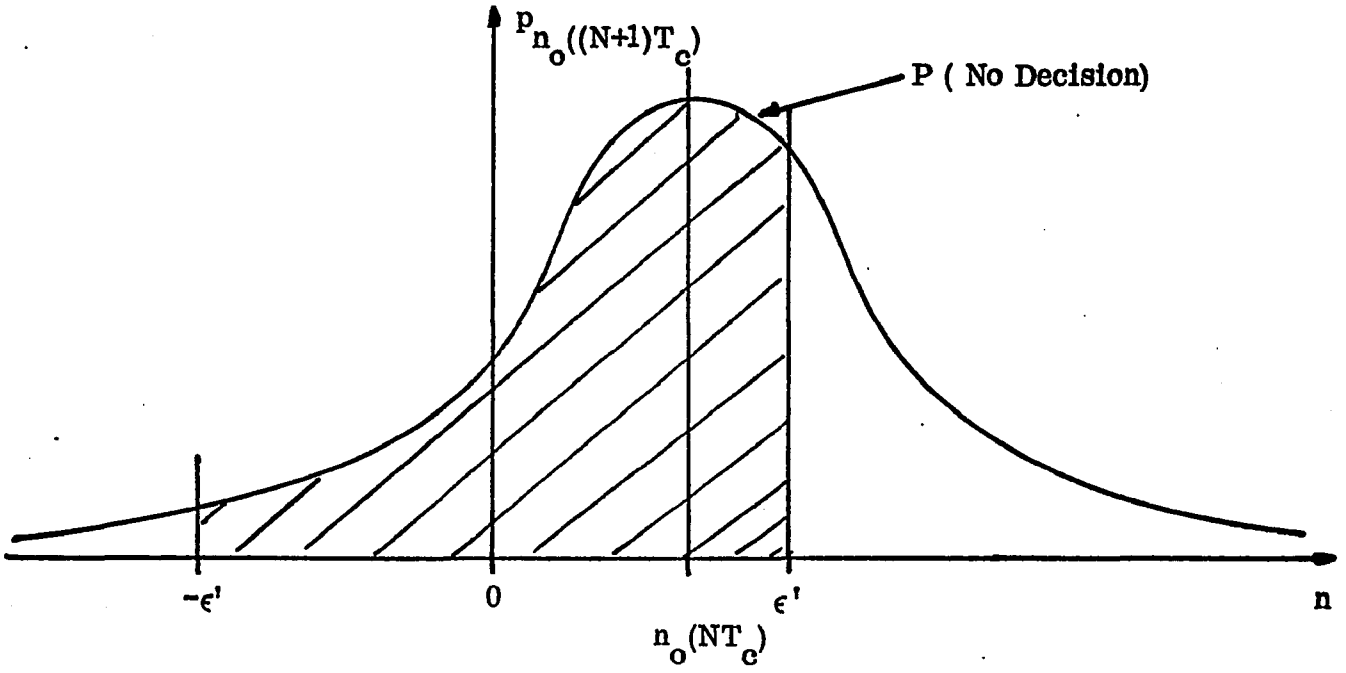


Fig. 25 Probability of No Decision for $n_o((N+1)T_c)$

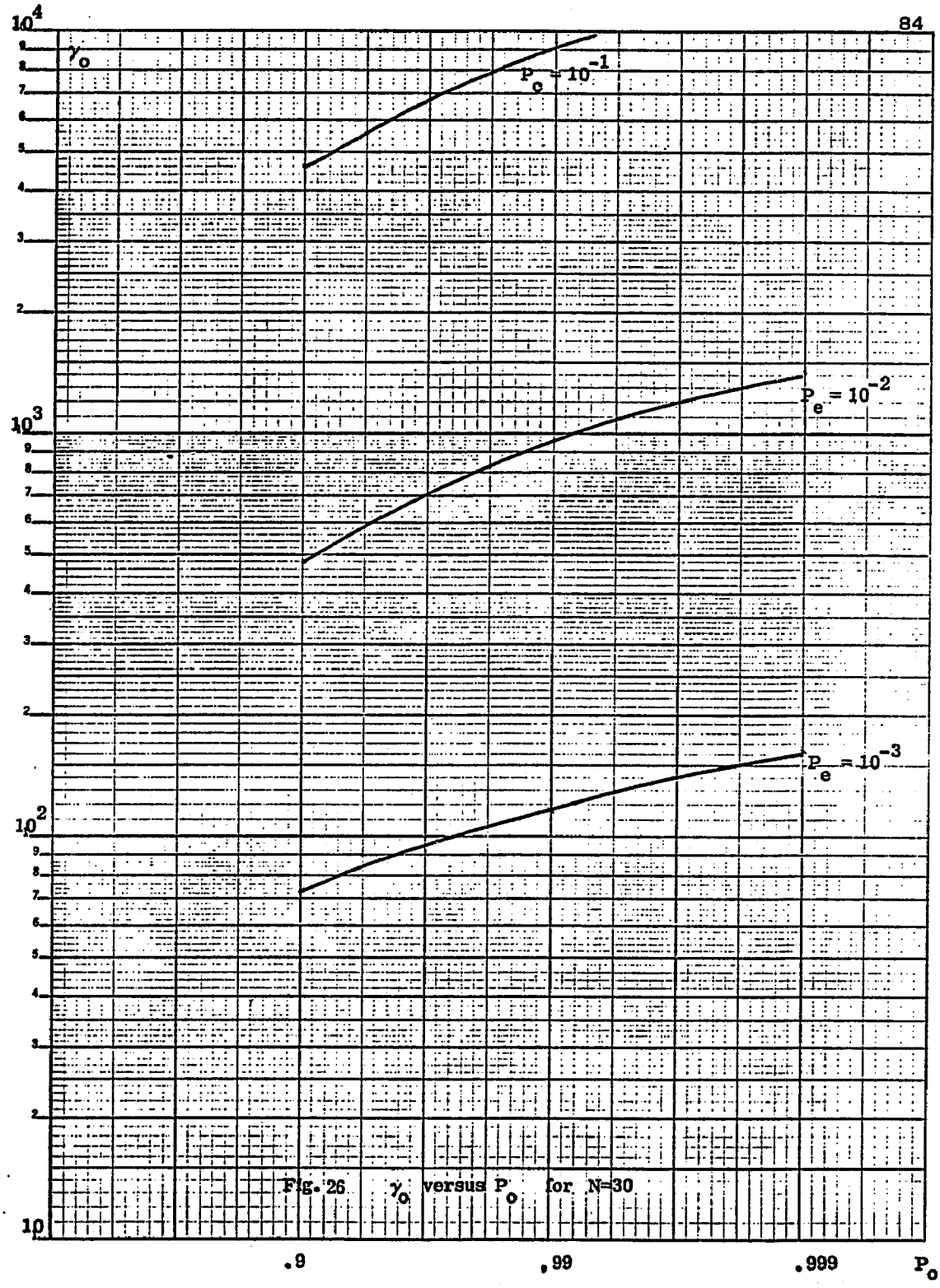


Fig. 26 γ_o versus P_o for N=30

Probability of Error in a Direct Sequence Spread Spectrum System with a CW Jammer at the Carrier Frequency

Direct Sequence Spread Spectrum Systems can operate reliably even in the presence of strong interference, which may be intentional or unintentional. The system operates by transmitting a digital signal of lower power spectral density which occupies a wide bandwidth. It has been shown in the previous sections how the relatively narrow band data is multiplied by a PN type sequence which is user specific and much higher in frequency. It is this operation which encodes the original data into a new signal which has much lower power spectral density and a much wider bandwidth than the original data. The receiver recovers the original data by correlating this wideband signal with the original PN code. The correlation has the beneficial side effect of rejecting the received interference. The interference rejection is very much similar to the original data encoding process, and it converts the interfering signal into a wideband, low power spectral density signal. The advantage is that now very little of the original power of the interference signal will fall into the data bandwidth with the out of band interference being filtered out. This encoding and decoding process is shown in Fig. 27 with the emphasis being placed on the spectral representation of the signals involved.

Based on the relative bandwidth of the PN signal with respect to the data signal which makes the interference at the

receiver have almost a constant value in the band of interest, and assuming the length L of the PN sequence to approach infinity such that the noise-like properties of PN codes are justified, the probability of error of such a system is assumed to be closely approximated by that of an identical system which operates with no interference but in the presence of white, gaussian noise of power spectral density equal to the value of the interference in the band of interest. There is no indication on how adequate this assumption remains when such parameters as E_b/η, P_s/P_j, L and $k(=\frac{f_c}{f_b})$ change their values significantly.

The Effect of CW Jamming at the Carrier Frequency. The data to be transmitted takes the binary value of ±1 with probability of 0.5. The PN sequence used for encoding also consists of binary digits of ±1 value whose distribution is well known. In general, a number of code bits, k, are used to encode each data bit, with the resulting signal modulating a carrier of frequency ω_o radians. The transmitted signal then becomes:

$$S(t) = d(t)\ PN\ (t+\ell T_c)\ \cos\ \omega_o t \qquad (1)$$

The signal received is corrupted by additive white, gaussian noise of zero mean and power spectral density $\eta/2$ and by additive interference. If the interference is considered to be of CW type of the same frequency as the RF carrier and with an arbitrary phase θ, then the received signal is of the form

$$S'(t) = A_s d(t) PN(t+\ell T_c) \cos\ \omega_o t + A_I\ \cos(\omega_o t+\theta) + n_i(t) \qquad (2)$$

At the receiver the signal is synchronously demodulated and decoded by a multiplication with $V_L(t) = 2PN(t+\ell T_c) \cos \omega_o t$. The resulting signal is then integrated for the duration of the data bit at which time a decision is made as to whether d(t) is a +1 or -1. This process is shown in Fig. 28.

Examining any one specific bit d(t) and assuming d(t) to equal +1, the voltage V_o can be expressed in terms of the data, the interference and the noise term.

$$V_o = \int_o^{T_b} [A_s\ PN(t+\ell T_c)\cos\ \omega_o t]\, 2PN(t+\ell T_c)\cos\ \omega_o t\ dt +$$
$$+ \int_o^{T_b} [A_I \cos(\omega_o t+\theta)\, 2PN(t+\ell T_c)\cos\ \omega_o t\ dt +$$
$$+ \int_o^{T_b} [n_i(t)]\, 2PN(t+\ell T_c)\cos\ \omega_o t = S_o + J_o + n_o \qquad (3)$$

The transmitted data bit d(t) will be estimated as a +1 if $V_o > 0$ or as a -1 if $V_o < 0$. An error is said to occur if $S_o + J_o + n_o < 0$ given that d(t) = +1. The probability of such an error is

$$P_e = P[S_o+J_o+n_o<0] = P[n_o > S_o+J_o] \qquad (4)$$

The output noise is also gaussian with zero mean, with variance $\sigma_o^2 = \eta T_b$. The probability of error becomes then

$$P_e = \int_{\alpha=S_o+J_o}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_o}\ e^{-\frac{n^2}{2\sigma_o^2}}\ dn \qquad (5)$$

or, changing the variable of integration n to $m = \frac{n}{\alpha}$ and taking advantage of the symmetry of a zero mean gaussian:

$$P_e = \frac{1}{2} - \int_0^1 \frac{1}{\sqrt{2\pi}\sigma_o} \alpha \, e^{-\frac{m^2\alpha^2}{2\sigma_o^2}} dm \qquad (6)$$

Equation (6) shows the strong dependence of the probability of error on the parameter α. However, even given a signal amplitude A_s, and an interference amplitude A_I and phase θ, the probability of error is not a well defined constant with the exception of the special case where k, the processing gain, equals L. Because the probability of error, P_e, is a function of α which is a random variable, it is itself a random variable. In this section we find the average value of this random variable, showing how it varies with respect to a number of relevant parameters which were known to be significant but their exact significance has not been previously shown. In addition, an indication of the degree of confidence which can be placed on this average value is given by the variance of the distribution of the probability of error.

Expected value of the probability of error. The random variable α has been previously defined as $\alpha = S_o + J_o$ where S_o and J_o are found in Eq. (3).

$$S_o = \int_o^{T_b} [A_s PN(t+\ell T_c)\cos \omega_o t]\, 2PN(t+\ell T_c)\cos \omega_o t \, dt$$

$$= A_s T_b \tag{6a}$$

$$J_o = \int_o^{T_b} [A_I \cos(\omega_o t+\theta)]\, 2PN(t+\ell T_c)\cos \omega_o t \, dt$$

$$= A_I \cos\theta \int_o^{T_b} PN(t+\ell T_c)\, dt$$

$$= A_I T_c \cos\theta \sum_{i=0}^{k-1} g_{i+\ell} \tag{6b}$$

where g_i = i^{th} chip of the PN sequence and k is the processing gain or $k = f_c/f_b$. If Eq. (6b) is examined closely, it becomes apparent that if k is allowed to equal L (with L being the length of the sequence), then J_o is no longer a random variable but takes the value of $J_o\Big|_{k=L} = -A_I T_c \cos\theta$. It is for this very special case that the probability of error is not a random variable. If, however, the value of k is less than L, then a random variable γ can be defined as

$$\gamma = \sum_{i=0}^{k-1} g_{i+\ell} \tag{7}$$

The random variable γ has been shown by Gold to have a hypergeometric distribution with

$$P_\gamma = \binom{n_1}{m}\binom{L-n_1}{k-m} \Big/ \binom{L}{k} \tag{8}$$

This hypergeometric distribution, however, does not lend itself very easily to analytical manipulation. Fortunately, it can be very closely approximated by a gaussian distribution whose mean μ_γ and variance s_γ^2 equal the mean and variance of γ. Therefore:

$$\mu_\gamma = E[\gamma] = E\left[\sum_{i=0}^{k-1} g_{i+\ell}\right] = \frac{1}{L}\sum_{i=0}^{k-1}\sum_{\ell=1}^{L} g_{i+\ell} = -\frac{k}{L} \tag{9a}$$

$$E[\gamma^2] = E\left[\sum_{i=0}^{k-1} g_{i+\ell}\sum_{j=0}^{k-1} g_{j+\ell}\right] = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=0}^{k-1}\sum_{j=0}^{k-1} g_{i+\ell}\, g_{j+\ell} =$$

$$= \frac{1}{L}\sum_{\ell=1}^{L}\left[k + \sum_{\substack{i=0 \\ i\neq j}}^{k-1}\sum_{j=0}^{k-1} g_{i+\ell} g_{j+\ell}\right]$$

$$= k + \sum_{\substack{i=0 \\ i\neq j}}^{k-1}\sum_{j=0}^{k-1}\frac{1}{L}\sum_{\ell=1}^{L} g_{i+\ell}\, g_{j+\ell}$$

Using the shift and add property of PN sequences, the above becomes:

$$E[\gamma^2] = k - \frac{k\,(k-1)}{L} \tag{9b}$$

Knowing the statistics of the r.v.γ we can find the statistics of α where α is a r.v. with mean μ and variance s^2. The mean is given by

$$E[\alpha] = A_s T_b + A_I T_c \cos\theta\, E[\gamma] = A_s T_b - A_I T_c (\cos\theta)\frac{K}{L} \tag{10}$$

The second moment of α is given by:

$$E[\alpha^2] = E[[A_s T_b + A_I T_c(\cos\theta)\gamma][A_s T_b + A_I T_c(\cos\theta)\gamma]]$$

$$= A_s^2 T_b^2 + 2A_s T_b A_I T_c(\cos\theta)E[\gamma] + A_I^2 T_c^2(\cos^2\theta)E[\gamma^2] \quad (11)$$

$$= A_s^2 T_b^2 + 2A_s T_b A_I T_c(\cos\theta)(-\frac{k}{L}) + A_I^2 T_c^2(\cos^2\theta)k(1-\frac{k-1}{L})$$

The variance of α, s^2 is found by definition from

$$s^2 = E[\alpha^2] - E^2[\alpha]$$

$$= A_s^2 T_b^2 + 2A_s T_b A_I T_c(\cos\theta)(-\frac{k}{L}) + A_I^2 T_c^2(\cos^2\theta)k(1-\frac{k-1}{L})$$

$$-[A_s T_b - A_I T_c(\cos\theta)\frac{k}{L}]^2 =$$

$$= A_I^2 T_c^2(\cos^2\theta)[k(1-\frac{k-1}{L}) - \frac{k^2}{L^2}] = A_I^2 T_c^2(\cos^2\theta)[-\frac{k^2}{L}(1+\frac{1}{L})$$

$$+ k(1+\frac{1}{L})] \approx A_I^2 T_c^2(\cos^2\theta)k(1-\frac{k}{L}) \quad (12)$$

The random variable α therefore has a gaussian distribution of the form

$$P_{(\alpha)} = \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(\alpha-\mu)^2}{2s^2}} \quad (13)$$

with μ and s^2 given by Eq. (10) and Eq. (12) respectively.

If the averaging of the probability of error with respect to the random variable α is the desired quantity, where the probability of error is as defined in Eq. (6), then

$$E[P_e] = \int_{-\infty}^{\infty} P_e(\alpha)\, p(\alpha)\, d\alpha$$

$$= \frac{1}{2} - \int_{\alpha=-\infty}^{+\infty} \int_{m=o}^{1} \frac{1}{\sqrt{2\pi}\sigma_o}\, \alpha\, e^{-\frac{m^2\alpha^2}{2\sigma_o^2}} \frac{1}{\sqrt{2\pi}s}\, e^{-\frac{(\alpha-\mu)^2}{2s^2}}\, dm d\alpha \qquad (14)$$

The powers of the exponential functions can be added and by complecting the square we obtain:

$$-\frac{m^2\alpha^2}{2\sigma_o^2} - \frac{(\alpha-\mu)^2}{2s^2} =$$

$$= -\frac{\frac{m^2s^2}{\sigma_o^2} + 1}{2s^2} \left[\alpha^2 - 2\alpha \frac{\mu}{\frac{m^2s^2}{\sigma_o^2} + 1} + \mu^2 \frac{1}{\frac{m^2s^2}{\sigma_o^2} + 1} + \mu^2 \frac{1}{(\frac{m^2s^2}{\sigma_o^2} + 1)^2} - \right.$$

$$\left. - \mu^2 \frac{1}{(\frac{m^2s^2}{\sigma_o^2} + 1)^2} \right] =$$

$$= -\frac{A}{2s^2}\left[(\alpha - \frac{\mu}{A})^2 + \frac{\mu^2}{A} (\frac{A-1}{A}) \right] \qquad (15)$$

where the symbol A represents the quantity:

$$A = \frac{m^2s^2}{\sigma_o^2} + 1 \qquad (16)$$

The average value of the probability of error as expressed in Eq. (14) can now be written as

$$E[P_e] = \frac{1}{2} - \int_{m=o}^{1} \frac{1}{2\pi\sigma_o s}\, e^{-\frac{\mu^2}{2s^2}[\frac{A-1}{A}]}\, dm \int_{\alpha=-\infty}^{\infty} \alpha\, e^{-\frac{A}{2s^2}(\alpha-\frac{\mu}{A})^2} d\alpha$$

$$= \frac{1}{2} - \int_{m=o}^{1} \frac{\mu}{\sqrt{2\pi}\sigma_o A^{3/2}}\, e^{-\frac{\mu^2}{2s^2}(\frac{A-1}{A})}\, dm \qquad (17)$$

It is often convenient to choose as parameters more useful quantities which are easier to use, such as E_b/η, $\frac{P_s}{P_J}$, etc. The quantity $\frac{\mu}{\sigma_o}$ can be rewritten as:

$$\frac{\mu}{\sigma_o} = \frac{A_s T_b - A_I T_c (\cos\theta)\frac{k}{L}}{\sqrt{\eta T_b}}$$

$$= \sqrt{\frac{2E_b}{\eta}} - \sqrt{\frac{P_J/L^2}{P_s} \times \frac{E_b}{\eta} 2\cos^2\theta} \qquad (18)$$

where

$$P_s = \frac{1}{2} A_s^2$$

$$P_J = \frac{1}{2} A_I^2$$

and $$E_b = P_s T_b \qquad (19)$$

The power of the exponential can also be written in a more useful form, such as:

$$\frac{\mu^2}{2s^2}\left[\frac{A-1}{A}\right] = \frac{\mu^2 m^2}{2[m^2 s^2 + \sigma_o^2]} = m^2 \frac{\frac{\mu^2}{\sigma_o^2}}{2\left(m^2 \frac{s^2}{\sigma_o^2} + 1\right)}$$

where

$$\frac{s^2}{\sigma_o^2} = \frac{A_I^2 T_c^2 (\cos^2\theta) k (1-\frac{k}{L})}{\eta T_b}$$

$$= \frac{E_b}{\eta} \frac{P_J/k}{P_s} (1-\frac{k}{L}) 2\cos^2\theta \qquad (20)$$

Finally, the average of the probability of error with respect to the random variable α can be written as:

$$E[P_e] = \frac{1}{2} - \int_{m=0}^{1} \left[\sqrt{\frac{2E_b}{\eta}} - \sqrt{\frac{P_J/L^2}{P_s} \frac{E_b}{\eta} 2\cos^2\theta} \right] \times$$

$$\times \frac{1}{\sqrt{2\pi}\left[1+m^2 \frac{E_b}{\eta} \frac{P_{J/k}}{P_s} (1-\frac{k}{L}) 2\cos^2\theta\right]^{3/2}}$$

$$\times e^{-\frac{1}{2} \left[\frac{m^2 \left(\sqrt{\frac{2E_b}{\eta}} - \frac{P_{J/L^2}}{P_s} \frac{E_b}{\eta} 2\cos^2\theta \right)^2}{1+m^2 \frac{E_b}{\eta} \frac{P_{J/k}}{P_s} (1-\frac{k}{L}) 2\cos^2\theta} \right]} \qquad (21)$$

The average probability of Eq. (21) is not always significant for every data bit. The degree of confidence we can place in it is given by the variance of the probability of error. As heretofore, the variance is defined as:

$$\mathrm{Var}[P_e] = E[P_e^2] - E[P_e^2] \qquad (22)$$

where the expected value of P_e is shown in Eq. (21). The other significant quantity, $E[P_e^2]$, can be expressed as:

$$E[P_e^2] = \int_{\alpha=-\infty}^{\infty} P_e^2(\alpha) p(\alpha) d\alpha$$

$$= \int_{\alpha=-\infty}^{\infty} \left[\int_{x=\alpha}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_o} e^{-\frac{x^2}{2\sigma_o^2}} dx \int_{y=\alpha}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_o} e^{-\frac{y^2}{2\sigma_o^2}} dy \right]$$

$$\frac{1}{\sqrt{2\pi}s} e^{-\frac{(\alpha-\mu)^2}{2s^2}} d\alpha \qquad (23)$$

The integrals of Eq.(23) can not be solved analytically and numerical methods would not necessarily be very useful as they would have to consider a large number of parameters which vary over a wide range. A useful indication of the approximate value of the $E[P_e^2]$ can be obtained if an upper bound of reasonable tightness can be found. It is known that

$$Q'\left(\frac{\alpha}{\sigma_o}\right) = \int_{x=\alpha}^{\infty} \frac{1}{\sqrt{2\ \sigma_o^2}}\ e^{-\frac{x^2}{2\sigma_o^2}}\ dx = \int_{m=\frac{\alpha}{\sigma_o}}^{\infty} \frac{1}{\sqrt{2\pi}}\ e^{-\frac{m^2}{2}} dm \leq$$

$$\leq \begin{cases} < \frac{1}{2}\ e^{-\frac{\alpha^2}{2\sigma_o^2}} & \text{for } \frac{\alpha}{\sigma_o} > 0 \\ \leq 1 & \text{for } \frac{\alpha}{\sigma_o} < 0 \end{cases} \qquad (24)$$

Using the bound of Eq. (24) and integrating Eq. (23) over two distinct ranges of integration for values of α positive and negative, the following equation is obtained:

$$E[P_e^2] = \int_{\alpha=-\infty}^{o} \frac{1}{\sqrt{2\pi s^2}}\ e^{-\frac{(\alpha-\mu)^2}{2s^2}}\ d\alpha\ +$$

$$+\int_{\alpha=o}^{\infty} \frac{1}{4}\ e^{-\frac{\alpha^2}{\sigma_o^2}}\ \frac{1}{\sqrt{2\pi s^2}}\ e^{-\frac{(\alpha-\mu)^2}{2s^2}}\ d\alpha \qquad (25)$$

The first part of Eq. (25) is simply $1 - Q(-\mu/s) = Q(\mu/s)$. Combining the powers of the exponents and complecting the squares in the second part of Eq. (25), the following is

obtained:

$$\int_{\alpha=o}^{\infty} \frac{1}{4\sqrt{2\pi s^2}} e^{-\frac{1}{2}\left[\frac{2\alpha^2 s^2+(\alpha-\mu)^2\sigma_o^2}{s^2\sigma_o^2}\right]} d\alpha$$

$$= \int_{\alpha=o}^{\infty} \frac{1}{4\sqrt{2\pi s^2}} e^{-\frac{1}{2}\left[\frac{(\alpha-\mu H)^2 + \mu^2 H(1-H)}{s^2 H}\right]} d\alpha \qquad (26)$$

where H is defined as

$$H = \frac{\sigma_o^2}{2s^2 + \sigma_o^2} = \frac{1}{1 + 2\frac{s^2}{\sigma_o^2}} \qquad (27)$$

Equation (26) can then be reduced to:

$$\int_{\alpha=o}^{\infty} \frac{1}{4\sqrt{2\pi s^2}} e^{-\frac{1}{2}\frac{2\alpha^2 s^2 + (\alpha-\mu)^2\sigma_o^2}{s^2\sigma_o^2}} d\alpha =$$

$$= \frac{\sqrt{H}}{4} e^{-\frac{1}{2}\left[\frac{\mu^2}{s^2}(1-H)\right]} Q(-\mu\frac{\sqrt{H}}{s}) \qquad (28)$$

Finally, Eq. (25) becomes:

$$E[P_e^2] = 1 - Q[-\frac{\mu}{s}] + \frac{\sqrt{H}}{4} Q(-\mu\frac{\sqrt{H}}{s}) e^{-\frac{1}{2}\left[\frac{\mu^2}{s^2}(1-H)\right]} \qquad (29)$$

Rewriting Eq. (29) in terms of the same parameters as Eq. (21) which defines $E[P_e]$, and again, making use of Eq. (18) and Eq. (20), we obtain:

$$E[P_e^2] = 1 - Q\left[-\frac{P_s}{P_{J/k}}\frac{1}{(\cos\theta)\sqrt{1-k/L}} + \frac{1/L}{\sqrt{1-k/L}}\right] +$$

$$+ \frac{1}{4\sqrt{1+4\frac{E_b}{\eta}\frac{P_{J/k}}{P_s}(1-\frac{k}{L})\cos^2\theta}} \times$$

$$\times Q\left[\frac{1}{4\sqrt{1+4\frac{E_b}{\eta}\frac{P_{J/k}}{P_s}(1-\frac{k}{L})\cos^2\theta}}\left(-\frac{P_s}{P_{J/k}}\frac{1}{(\cos\theta)\sqrt{1-k/L}} + \frac{1/L}{\sqrt{1-k/L}}\right)\right] \times$$

$$\times e^{-\left[\sqrt{2\frac{E_b}{\eta}} - \sqrt{\frac{P_{J/L^2}}{P_s}\frac{E_b}{\eta}2\cos^2\theta}\right]} \frac{1}{1+4\frac{E_b}{\eta}\frac{P_{J/k}}{P_s}(1-k/L)\cos^2\theta} =$$

$$\simeq Q\left[\frac{P_s}{P_{J/k}}\frac{1}{(\cos\theta)\sqrt{1-k/L}}\right] + \frac{1}{4\sqrt{1+4\frac{E_b}{\eta}\frac{P_{J/k}}{P_s}(1-\frac{k}{L})\cos^2\theta}} \times$$

$$\times Q\left[-\frac{1}{4\sqrt{1+4\frac{E_b}{\eta}\frac{P_{J/k}}{P_s}(1-\frac{k}{L})\cos^2\theta}}\left(\frac{P_s}{P_{J/k}}\frac{1}{(\cos\theta)\sqrt{1-k/L}}\right)\right] \times$$

$$\times e^{-\left[\frac{2E_b}{\eta}\right]^{\frac{1}{2}}\left(\frac{1}{1+4\frac{E_b}{\eta}\frac{P_{J/k}}{P_s}(1-k/L)\cos^2\theta}\right)} \qquad (30)$$

Conclusion

The average probability of error values given by the hypergeometric modeling of the subsequence weight distribution is plotted in Fig. 29a through 29i vs. E_b/η for different values of the ratio k/L (processing gain/sequence length). Each average probability of error curve on the plot corresponds to some effective signal to jamming ratio $R = \frac{P_S}{P_J/k}$. Thus, the significance of the effective signal to jamming ratio can be clearly seen.

Figures 30a through 30e plot k/L vs. E_b/η for a constant average probability of error. Again, each curve on the plot corresponds to some value of R, where R has been defined above. These curves clearly show the effect of the k/L ratio on the average probability of error.

These results show the significance of R and k/L on the average probability of error vs. E_b/η, and can serve as a guide to the spread spectrum system designer in chosing the processing gain, k, and the sequence length, L, such that he may obtain the maximum CW jammer protection.

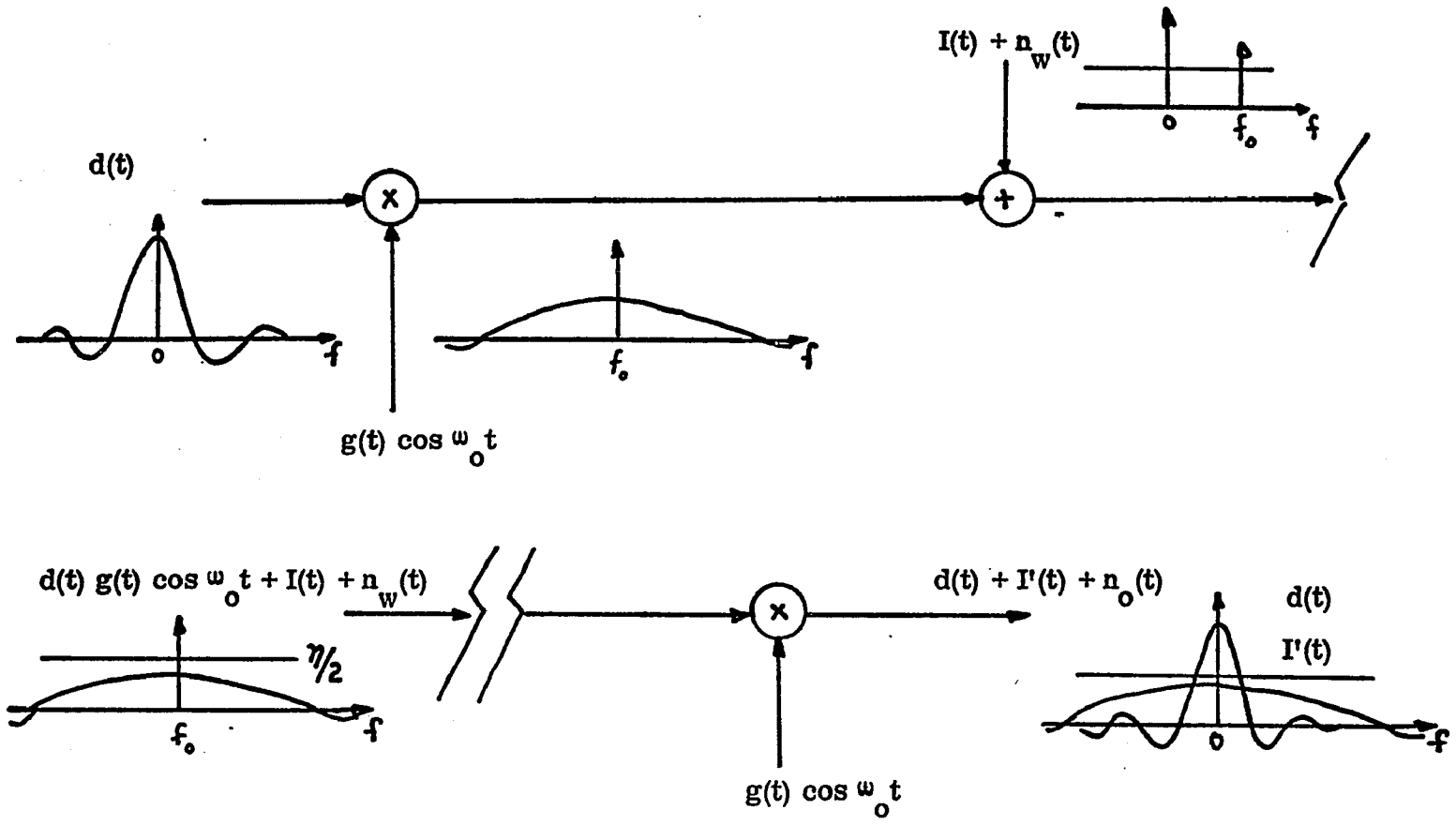


Fig. 27 Spread Spectrum Signals (DS)

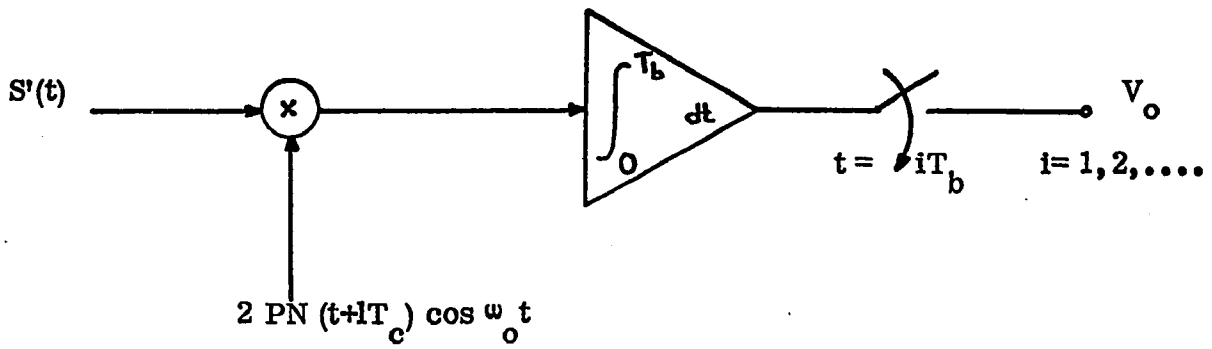


Fig.28 Data Decoding in a Direct Sequence Spread Spectrum System

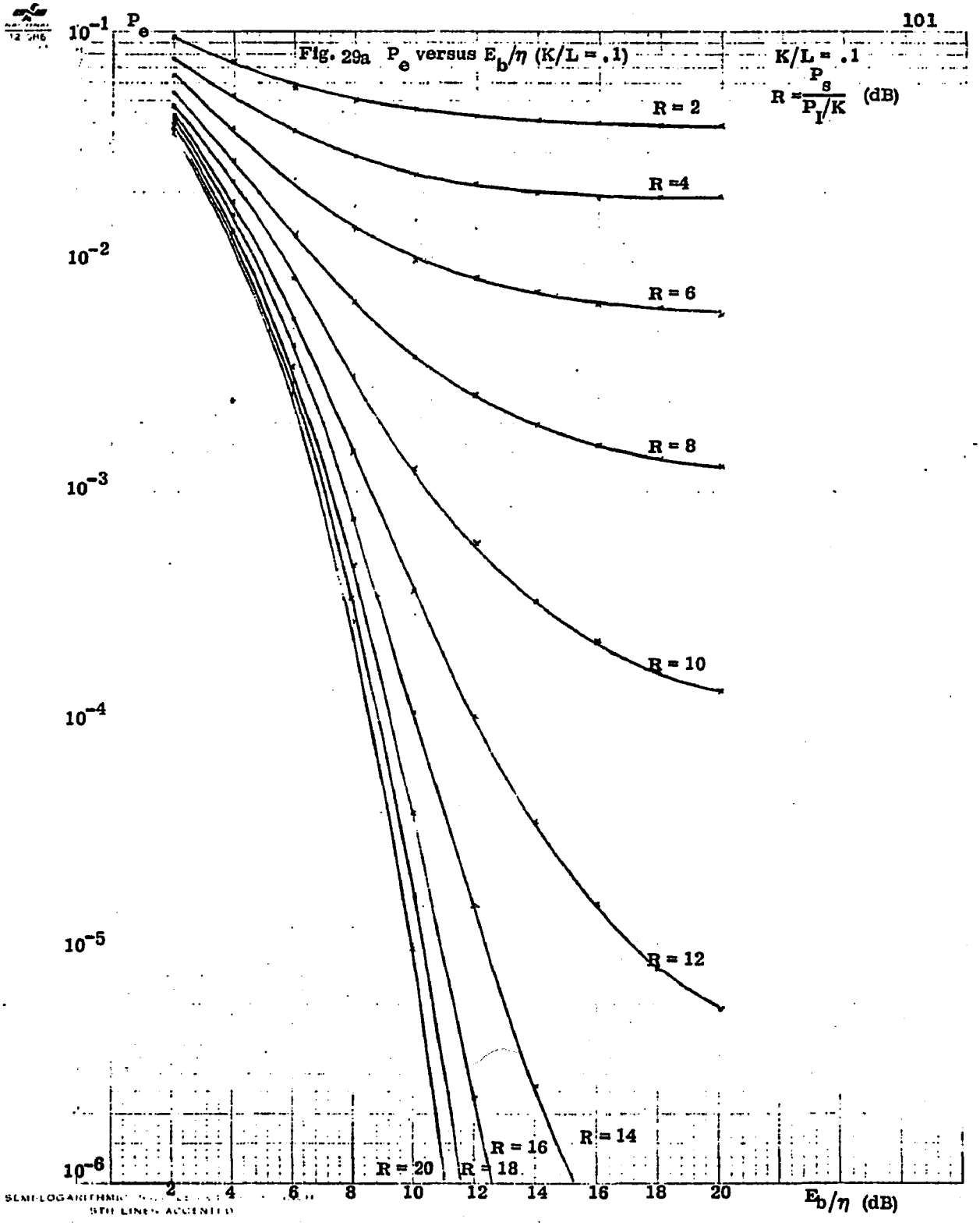


Fig. 29a P_e versus E_b/η (K/L = .1)

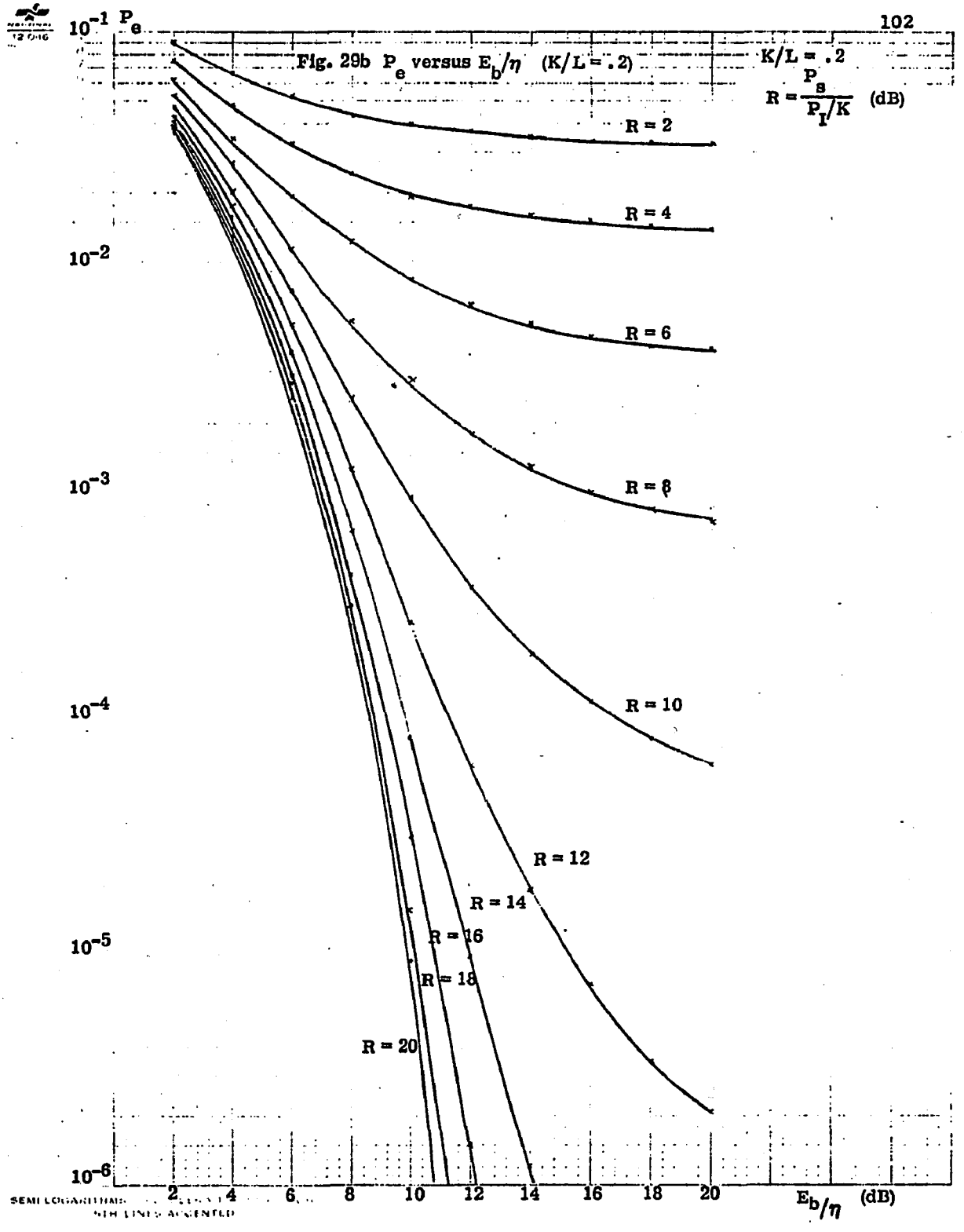


Fig. 29b P_e versus E_b/η (K/L = .2)

SEMI LOGARITHMIC WITH LINES AUGMENTED

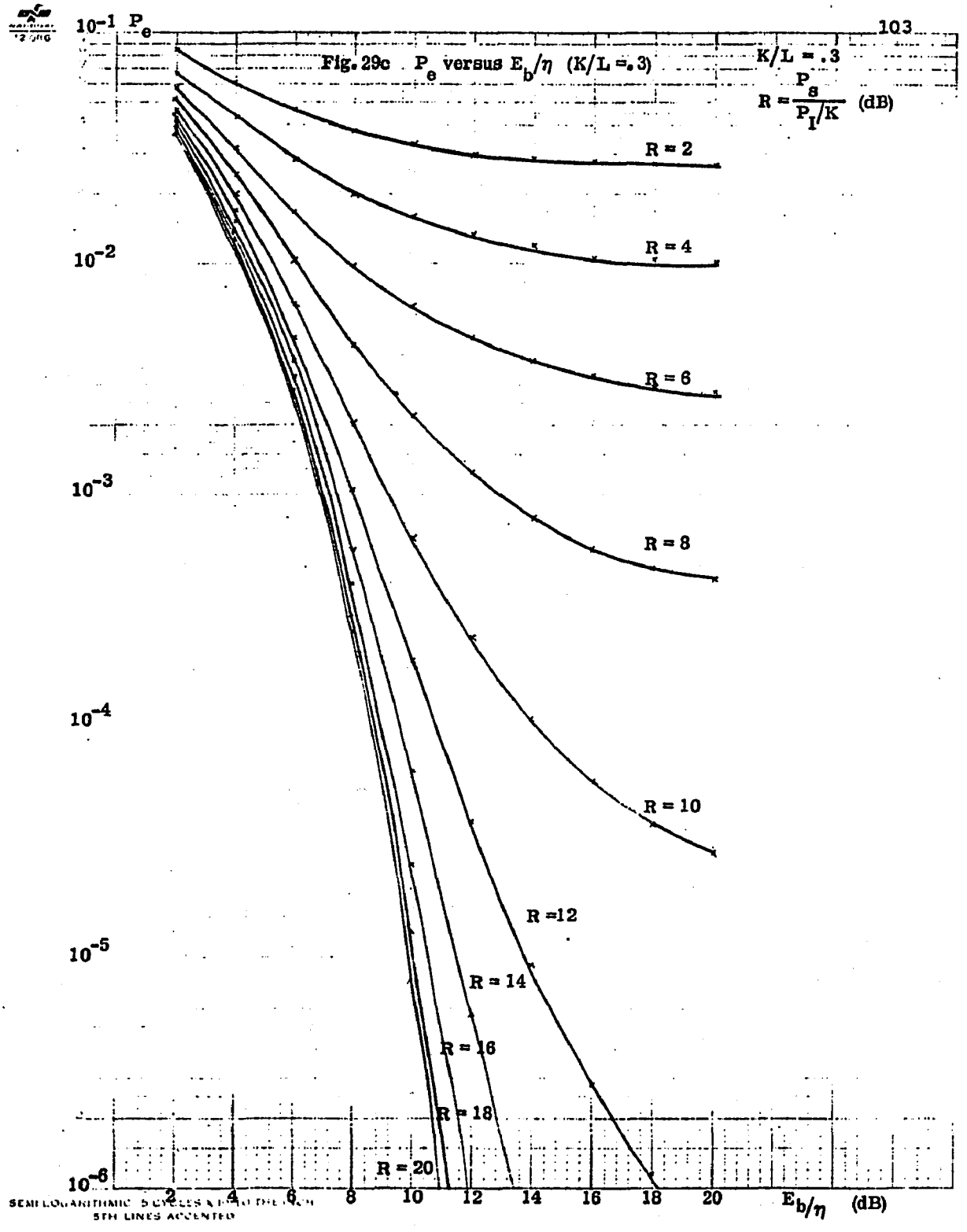


Fig. 29c P_e versus E_b/η (K/L = .3)
K/L = .3
$R = \frac{P_s}{P_I/K}$ (dB)
10^{-1} P_e
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
R = 2
R = 4
R = 6
R = 8
R = 10
R = 12
R = 14
R = 16
R = 18
R = 20
2
4
6
8
10
12
14
16
18
20
$E_{b/\eta}$ (dB)

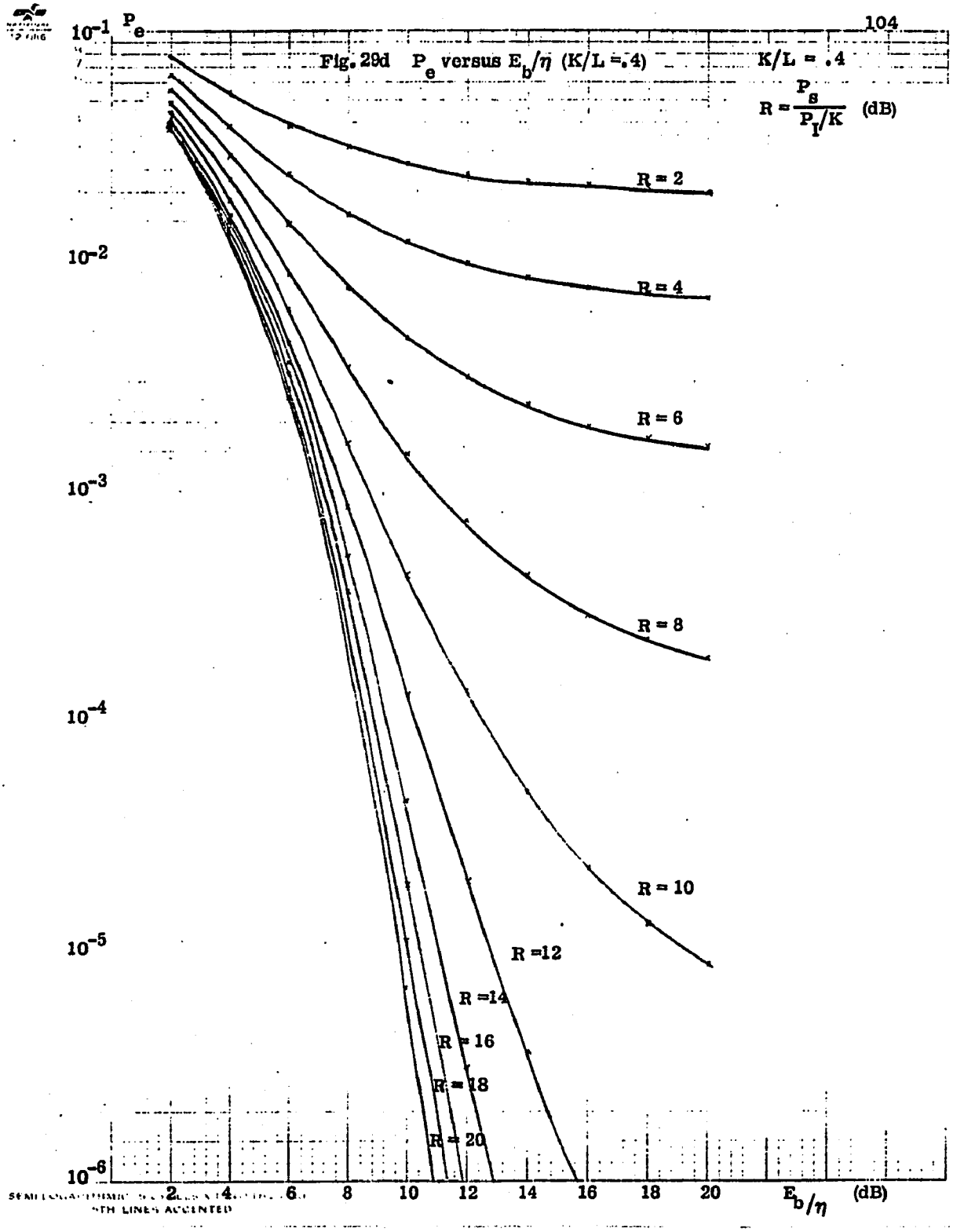


Fig. 29d P_e versus E_b/η (K/L = .4)

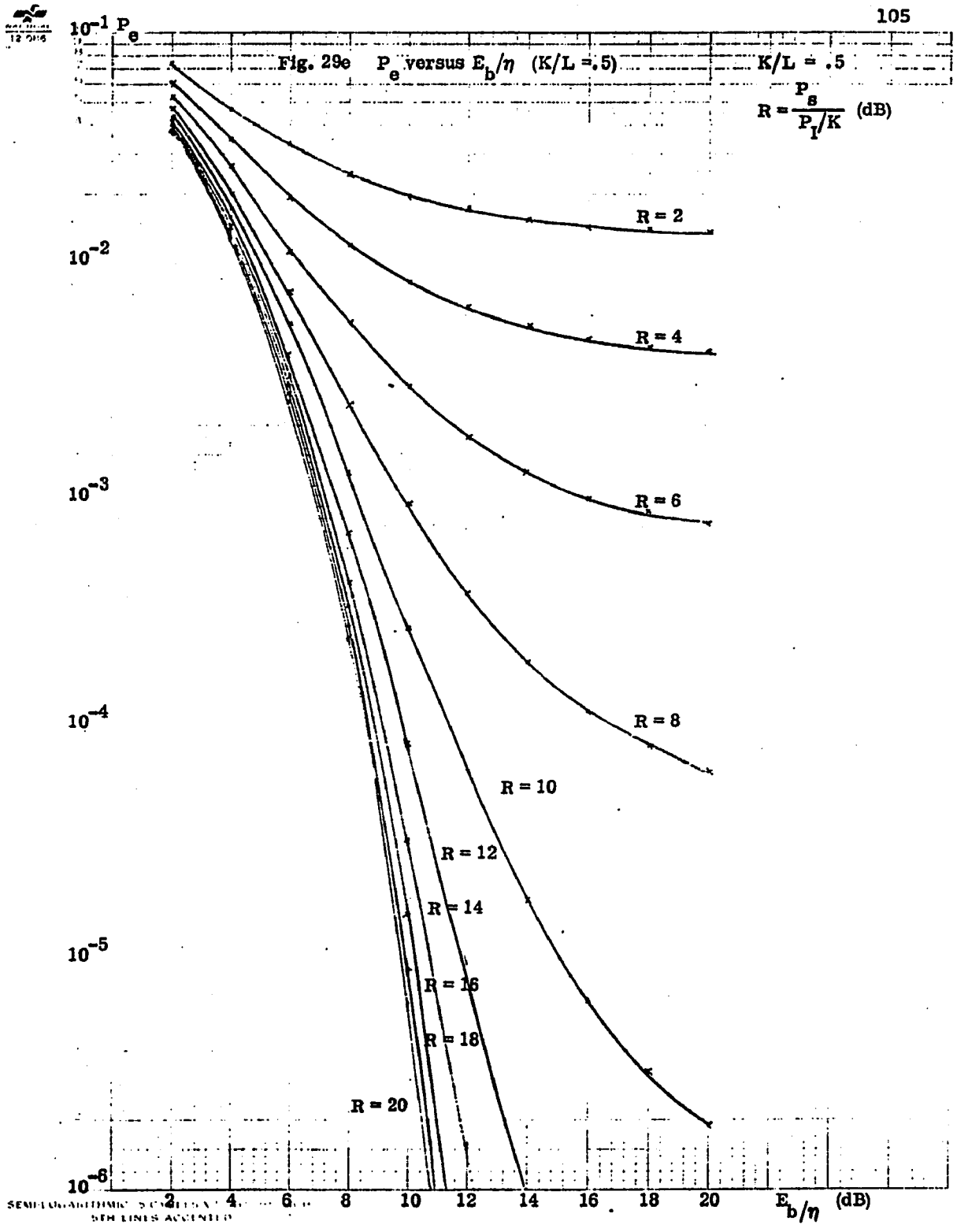
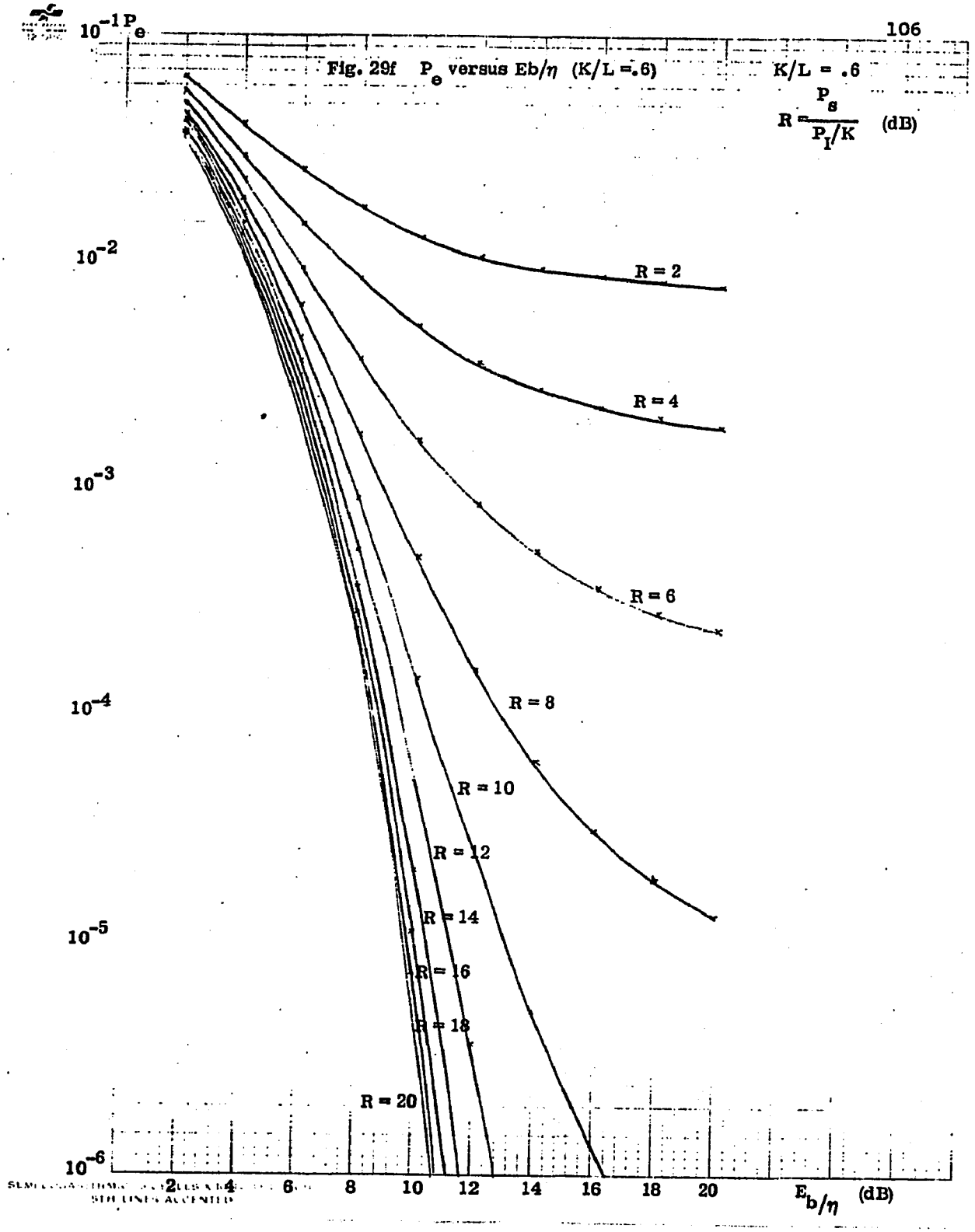

10^{-1} P_e
Fig. 29e P_e versus E_b/η (K/L = .5)
K/L = .5
$R = \frac{P_s}{P_I/K}$ (dB)
R = 2
R = 4
R = 6
R = 8
R = 10
R = 12
R = 14
R = 16
R = 18
R = 20
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
2
4
6
8
10
12
14
16
18
20
E_b/η (dB)

Fig. 29f P_e versus Eb/η (K/L =.6)



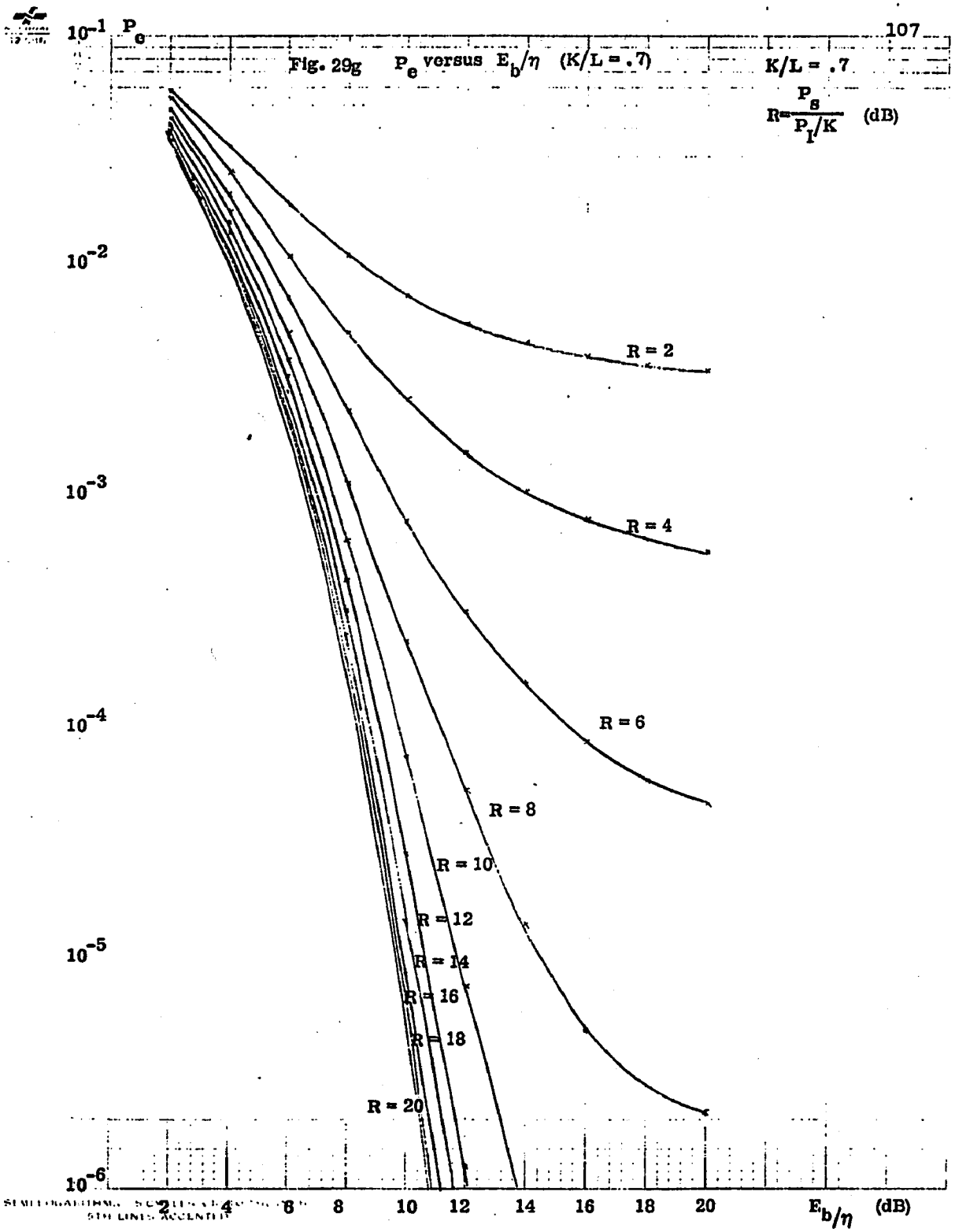


Fig. 29g P_e versus E_b/η (K/L = .7)
K/L = .7
$R=\frac{P_s}{P_I/K}$ (dB)
P_e
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
R = 2
R = 4
R = 6
R = 8
R = 10
R = 12
R = 14
R = 16
R = 18
R = 20
2
4
6
8
10
12
14
16
18
20
$E_{b/\eta}$ (dB)

Fig. 29h P_e versus E_b/η (k/L = .8)

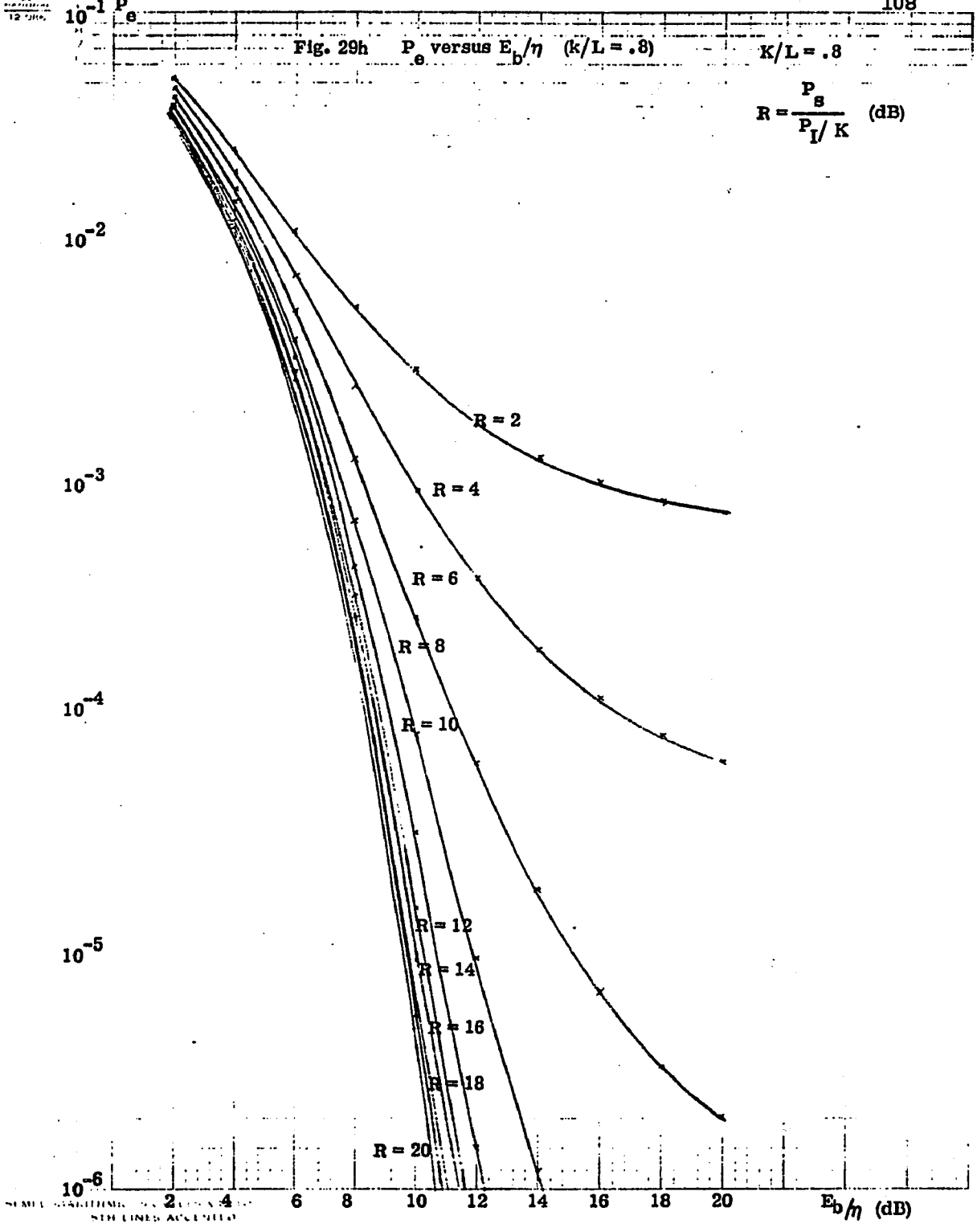
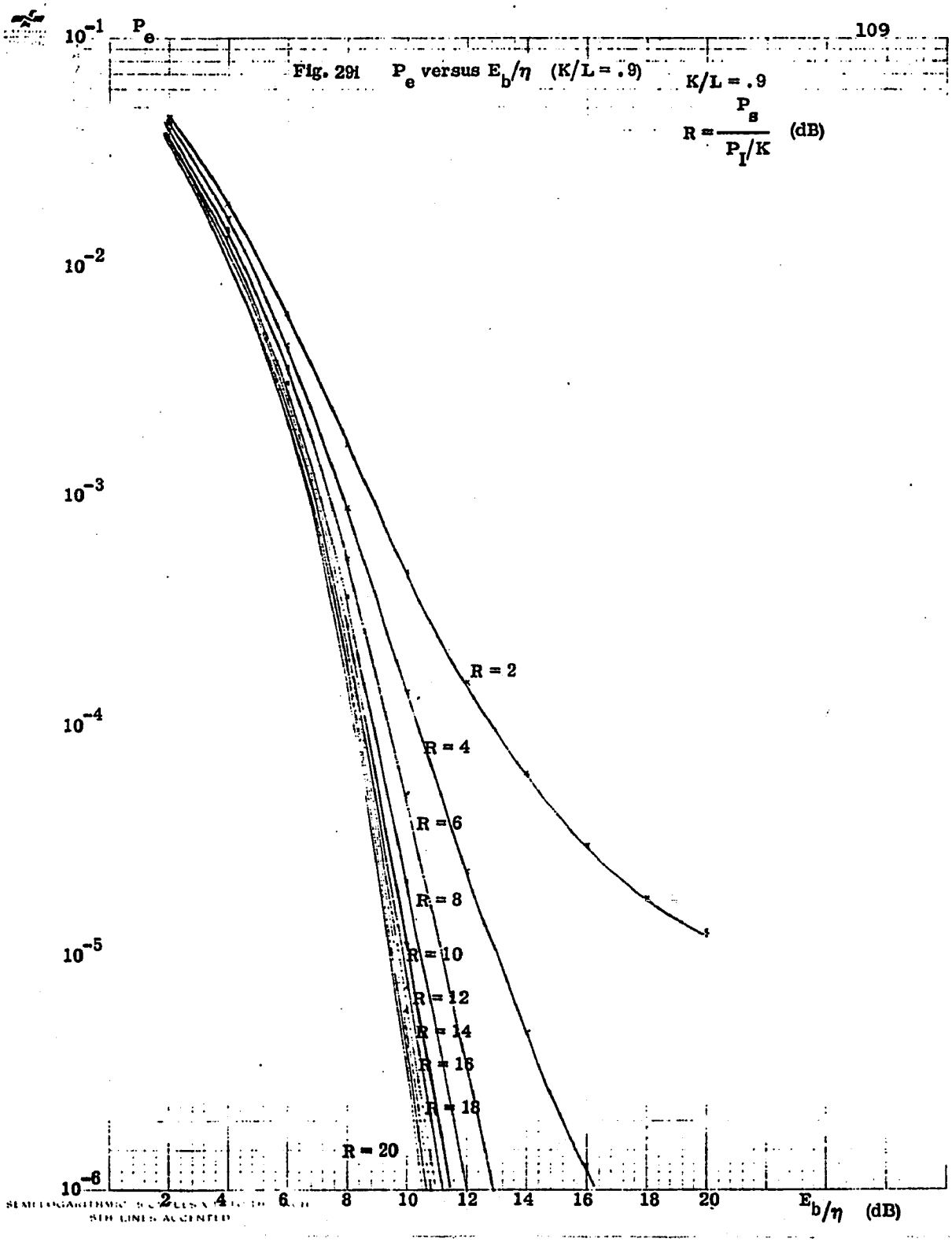


Fig. 29i P_e versus E_b/η (K/L = .9)



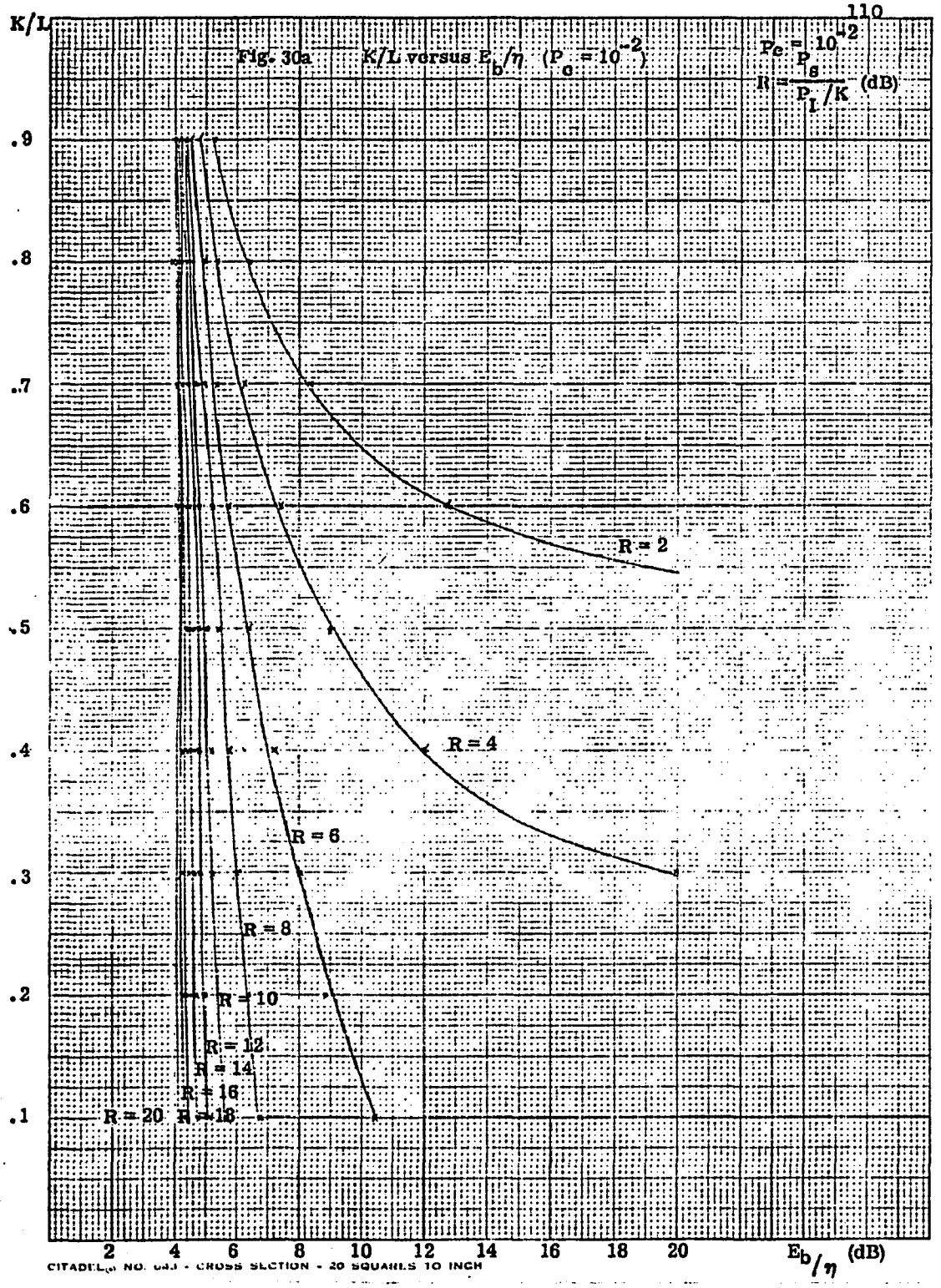
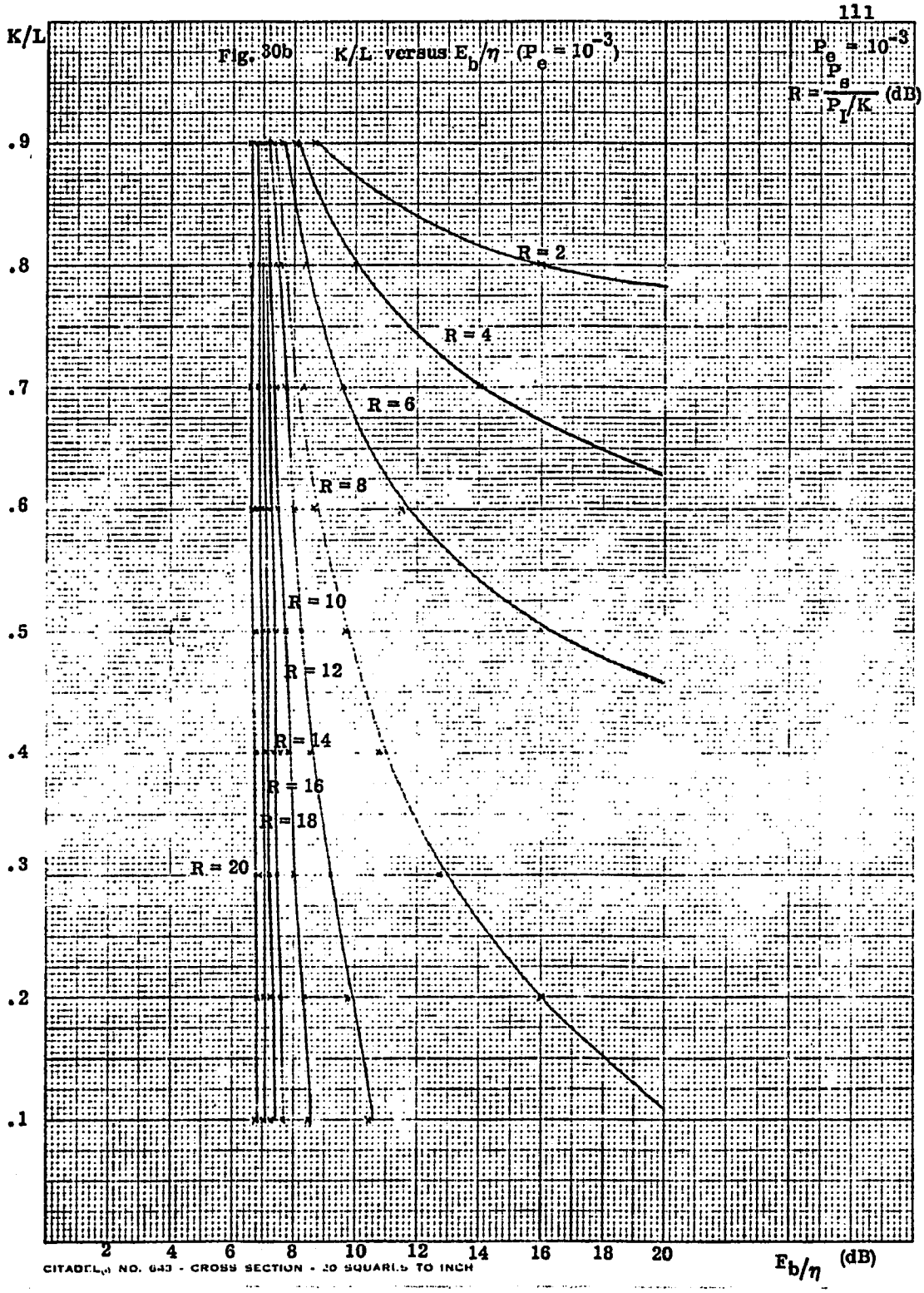
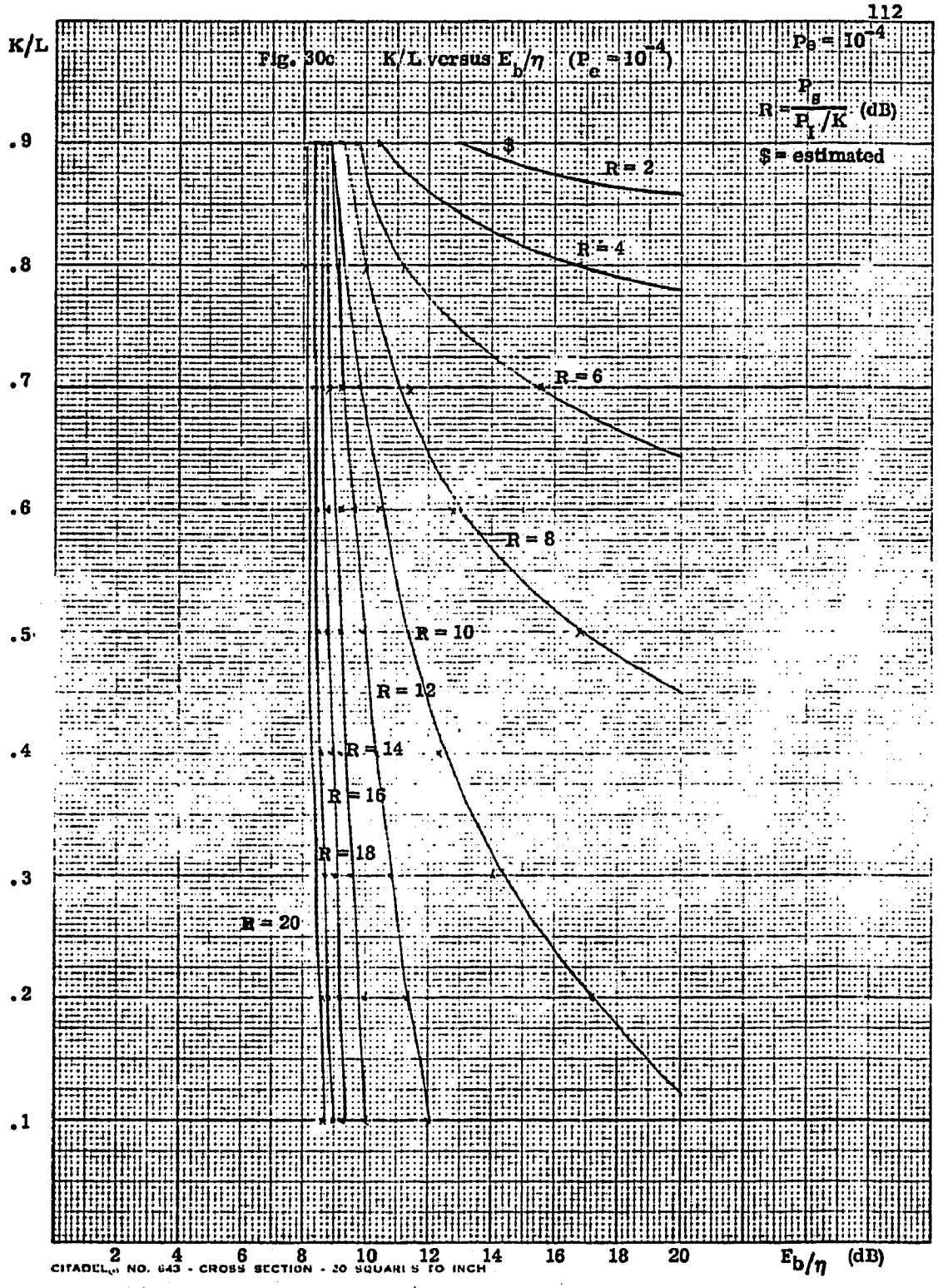


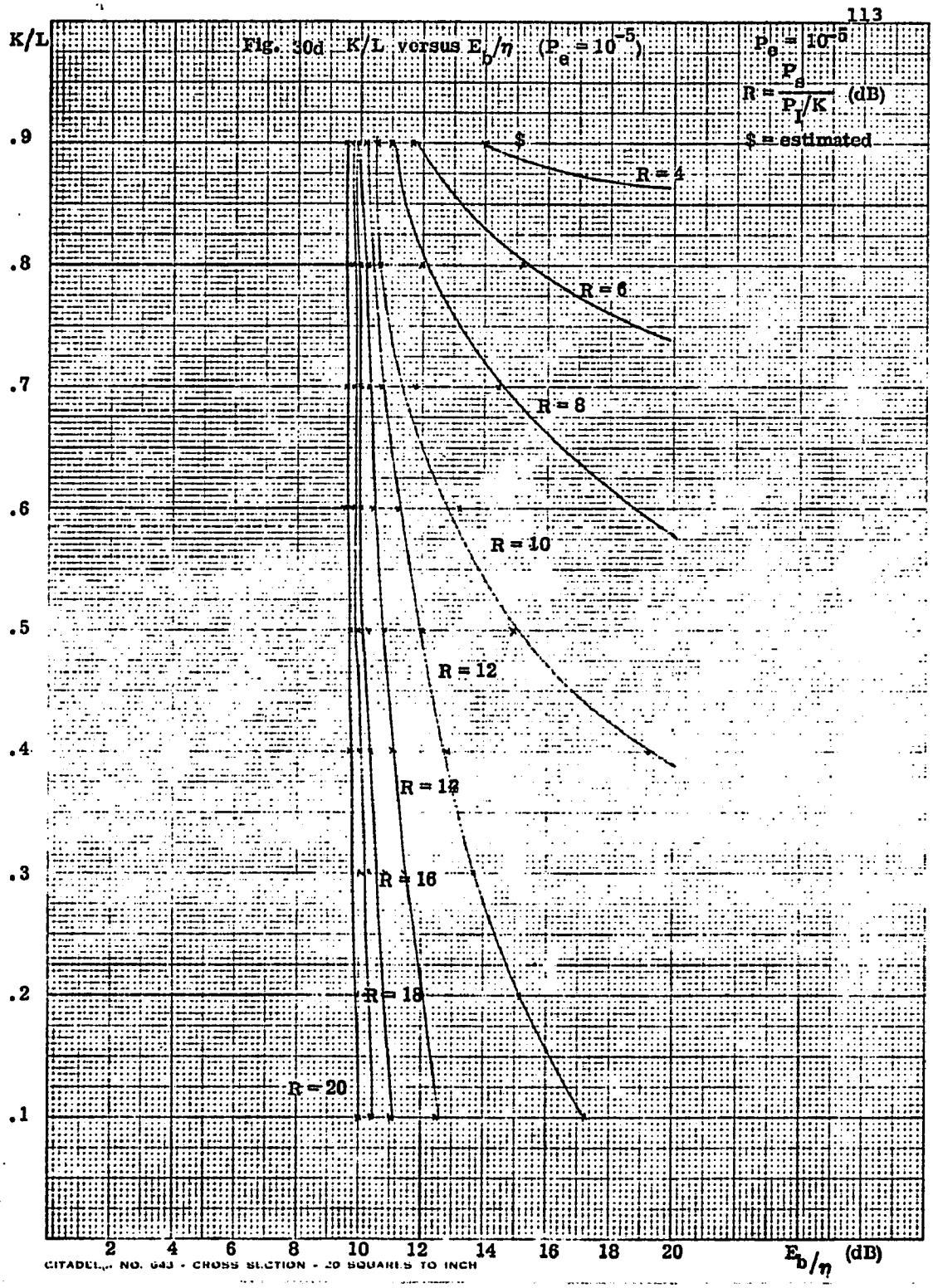
Fig. 30a K/L versus E_b/η ($P_e = 10^{-2}$)

$P_e = P_s$

$R = \frac{10^{42}}{P_I/K}$ (dB)


K/L
Fig. 30b K/L versus E_b/η ($P_e = 10^{-3}$)
$P_e = 10^{-3}$
$R = \frac{P_s}{P_I/K}$ (dB)
.9
.8
.7
.6
.5
.4
.3
.2
.1
R = 2
R = 4
R = 6
R = 8
R = 10
R = 12
R = 14
R = 16
R = 18
R = 20
2
4
6
8
10
12
14
16
18
20
$E_{b/\eta}$ (dB)
CITADEL® NO. 643 - CROSS SECTION - 20 SQUARES TO INCH


K/L
Fig. 30c K/L versus E_b/η ($P_e = 10^{-4}$)
$P_s = 10^{-4}$
$R = \frac{P_s}{P_1/K}$ (dB)
$ = estimated
.9
.8
.7
.6
.5
.4
.3
.2
.1
$
R = 2
R = 4
R = 6
R = 8
R = 10
R = 12
R = 14
R = 16
R = 18
R = 20
2 4 6 8 10 12 14 16 18 20
E_b/η (dB)
CITADEL NO. 643 - CROSS SECTION - 20 SQUARES TO INCH


K/L
Fig. 30d K/L versus E_b/η ($P_e = 10^{-5}$)
$P_e = 10^{-5}$
$R = \frac{P_e}{P_I/K}$ (dB)
$ = estimated
.9
.8
.7
.6
.5
.4
.3
.2
.1
$
R = 4
R = 6
R = 8
R = 10
R = 12
R = 14
R = 16
R = 18
R = 20
2
4
6
8
10
12
14
16
18
20
E_b/η (dB)
CITADEL, NO. 643 - CROSS SECTION - 20 SQUARES TO INCH

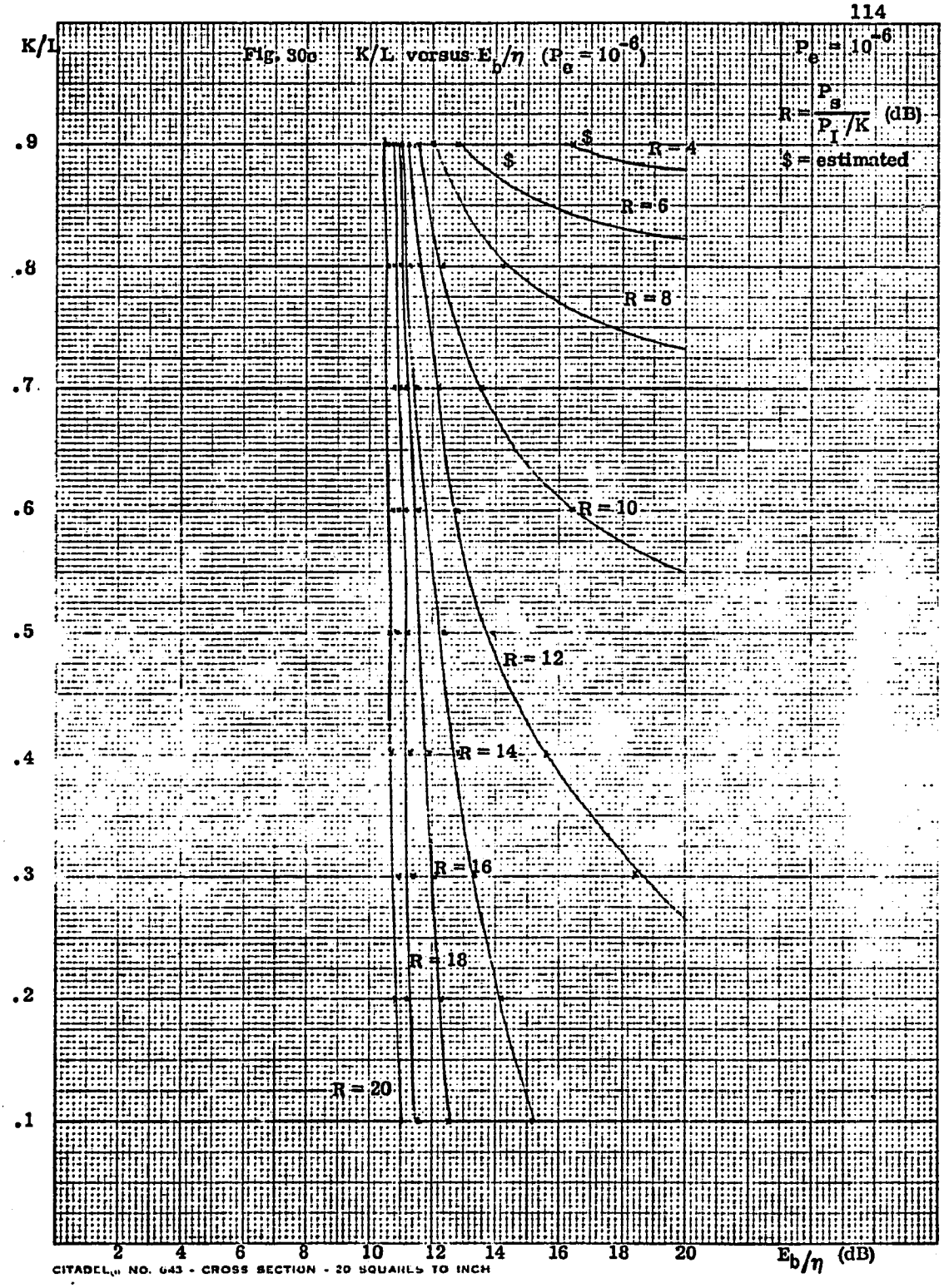


Fig. 30b K/L versus E_b/η ($P_e = 10^{-6}$)

DIGITAL ENCODING OF COLOR VIDEO SIGNALS

Introduction

The Color Video Signal

Out of the early days of video development the industry has inherited the NTSC standard. Devised primarily for black and white video transmission it called for a frame update rate of 30 frames/sec. Each frame is made up of two interlaced fields which are displayed at a rate of 1/60th of a second. The video signal has a bandwidth of 4.5 MHz total. Display synchronization is provided by non-displayed portions of the video signal which accomplish vertical (field) and horizontal (line) synchronization. The introduction of color video has called for a signal structure such that full compatibility with the black and white T.V. sets is maintained. This has resulted in a system where the three primary colors R, G and B are transformed into three other signals, I, Y and Q, with Y being the luminance and I and Q the chrominance. Figure 1 shows the spectrum of a typical NTSC video signal.

In terms of the R, G and B signals,

$$Y = 0.3\ R + 0.59\ G + 0.11\ B \tag{1}$$

$$I = 0.6\ R - 0.28\ G - 0.32\ B \tag{2}$$

and

$$Q = 0.21\ R - 0.52\ G + 0.31\ B \tag{3}$$

The I and Q signals are also transmitted and it is the

detection of the presence of these I and Q signals in the color monitor, in addition to the Y signal, that enables the decoding of the R, G, and B and the displaying of a color picture.

The I and Q signals are quadrature amplitude modulated onto a carrier frequency of 3.57 MHz. Thus, the complete color signal is

$$V(t) = Y(t) + I(t) \cos \omega_o t + Q(t) \sin \omega_o t \qquad (4)$$

In the color monitor the signal is separated into the luminance and chrominance signals and the R, G and B colors are then extracted using a network which, in essence, solved the matrix

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.96 & 1 & 0.62 \\ -0.27 & 1 & -0.65 \\ -1.11 & 1 & 1.7 \end{bmatrix} \cdot \begin{bmatrix} I \\ Y \\ Q \end{bmatrix} \qquad (5)$$

The three color signals R, G and B are bandlimited to about 3.3 MHz and the luminance signal has most of its power in this frequency band. However, the inphase signal I is bandlimited to 1.3 MHz and the quadrature signal Q is limited to 0.5 MHz.

This bandlimiting of the chroma information has the effect of severely distorting the color content of the small picture elements.

However, the psychovisual properties of the eye are such that the color content of small objects is irrelevant and

therefore the effect of the distortion is small.

Other approaches to color video encoding have resulted in the development of the PAL and SECAM systems. All three systems are similar in that they all use a luminance and two chrominance channels. While the NTSC provides for the chrominance information to be transmitted via QAM modulation of a color subcarrier,the PAL system alternates the sign of one of the chroma channels every other line while the SECAM system transmits only one of the chroma channels every line. The net result is that in the PAL system the hue value is maintained much better under the effect of equipment misalignment at the expense of the saturation information, while in the SECAM system the vertical resolution of the chroma information is decreased with the color video signal being made simpler by eliminating the QAM of the color subcarrier.

None of the three systems described above have a decisive technical advantage over the other two, and this is one of the reasons why no one standard was universally adopted.

Digital Encoding of Video Signals

Introduction. The desire to encode video signals digitally and then transmit and receive the digital rather than the analog representation of the video information has generated two principal methods of digital encoding. This section will give a brief presentation of the principles underlying the transform and predictive coding methods.

Transform Encoding. Transform encoders perform two distinct operations: 1) The actual transformation and 2) the quantization of the coefficients. A transform encoer is shown in Fig. 2. The transformation from the time domain to the transform domain involves representing the function f(t) by:

$$f(t) = \sum_i A_i \phi_i$$

where A_i is the i^{th} transform coefficient and the ϕ_i's are the mutually orthagonal basis functions. If the set of functions ϕ_i is properly chosen then only relatively few coefficients A_i will have a substantial magnitude. The quantizer will encode those few coefficients accurately (using enough bits to guarantee a small quantization error) and the rest of the coefficients A_i will be encoded very coarsely (using fewer bits). It is by this unequal importance assigned to the different coefficients A_i that the data reduction is achieved at the expense of negligible picture degradation.

A complete discussion of the various transform encoding techniques, together with a discussion of various fast implementation algorithms and a presentation of their basis functions can be found in reference (23). In general, however, transform encoders are not very widely used in spite of their excellent bit rate reduction capabilities. This is due primarily to enormous computational requirements which are difficult to perform in real time.

Predictive Coding. Predictive encoders are, in general, much simpler to implement and operate than transform encoders. It is for this reason that they are more widely used. The most popular encoders in this category are PCM, DPCM and Delta Modulators. This introduction will give a brief description of each of the above with a slant towards video encoding.

Pulse Code Modulation (PCM). A PCM system is shown in Fig. 3. The input signal $v_i(t)$ is low pass filtered, sampled and quantized. The low pass filter prevents any aliasing errors which may be due to sampling. The quantizer must have a minimum of 64 to 256 levels or 6 to 8 bits per sample. Even though the sampling rate only has to be higher than twice the maximum frequency (or approx. 8 MHz), it is usually set at three times the frequency of the color subcarrier or 10.8 MHz. The PCM bit rate in bits per second is given by:

$$B = M * N * K * F \qquad (1)$$

If M = 512 pixels/line

N = 525 lines/frame

K = 6 to 8 bits/pixel

F = 30 frames/second

then the PCM bit rate required to transmit an NTSC video signal is higher than 48 to 64 MBPS.

PCM systems can transmit any conceivable television image because they do not make use of any pixel correlations. This is not of much interest because a human observer will only

find meaningful those images where there is a high degree of pixel correlation. Extensive simulations have shown the typical information content of a video image to be on the order of 1 bit/pixel. This indicates that an ideal system would operate well at a bit rate 1/6 to 1/8 of the bit rate required for PCM.

The 6 to 8 pixels form a word. In the transmission process noise is added to the transmitted signal. Because of this it is possible to make an error at the receiver. Even though the errors are random (any bit can be in error), the significance of the error varies from bit to bit. An error in the MSB will be over 100 times greater than an error in the LSB (assuming an 8 bit word). The signal to noise ratio of a PCM system is given by:

$$(S/N)_o = \frac{2^{2K}}{1 + 2^{2(K+1)} P_e} \qquad (2)$$

where: K = number of bits/pixel

P_e = probability of error.

Because the human eye does not tolerate this type of error very well, PCM systems are operated in environments where the error rate is 10^{-6} or less. From (2) it is clear that very high S/N ratios can be attained if the P_e is negligible. Because of this high value of the S/N ratio PCM systems are widely used in studio environments.

Delta PCM (DPCM). A DPCM system is shown in Fig. 4. The predictive coder will use the past N samples to generate an estimate of the next pixel by a weighted summation:

$$S_m = \sum_{i=1}^{N} A_i S_{m-1} \tag{3}$$

The estimate is then compared to the incoming pixel and the difference encoded in a number of pits and transmitted. Because now we transmit the difference between the estimate and the pixel rather than the pixel itself, we can use a smaller number of bits per pixel and still obtain good quality video.

The coefficients A_i are chosen in such a manner as to minimize the variance of the error signal. Therefore, the system will perform well only as long as the input signal will maintain the statistics for which the A_i's have been calculated. Because the input signal's statistics can vary widely, some adaptive scheme must be considered if a high quality performance is to be maintained.

DPCM with adaptive predictors. In designing a DPCM system one must either use a predictor with variable parameters such that the parameters would change with the variations in the signal (always generating a stationary differential signal) or one can use a fixed predictor with a variable quantizer to accommodate the resultant nonstationary differential signal.

In a DPCM system with an adaptive linear predictor, the weightings on the adjacent samples used in predicting an incoming sample can change according to variations in the signal value. One way in which these signal variations can be accounted for is to include a delay during which the incoming samples are stored in an input buffer and used to obtain an estimate of the signal covariance matrix. The measured covariance matrix can be used to obtain a set of weightings for the predictor. These values are then used for processing the stored signals. The updated values of the predictor coefficients need to be periodically transmitted to the receiver.

DPCM systems with adaptive quantizers. A DPCM system with a fixed predictor will have a nonstationary differential signal for nonstationary data. Using a fixed quantizer, nonstationary differential signals would cause an abnormal saturation or a frequent utilization of the smallest level in the quantizer. To remedy this situation, the threshold and the reconstruction levels of the quantizer must be made variable to expand and contract according to signal statistics. Adaptation of the quantizer to signal statistics is accomplished using various approaches. One such approach stores k samples of the differential signal to obtain an estimate for the local standard deviation of the signal. Then the stored signal is normalized by the estimated standard deviation and is quantized using a fixed quantizer. Naturally, the scaling coefficient must be transmitted once for every k samples for

receiver synchronization. In a similar approach, called Block-Adaptive DPCM, a block of M samples is stored and is normalized by n possible constants. The total distortion for all M samples using each normalizing constant is calculated at the encoder. The normalizing constant giving the smallest distortion is used to scale the samples in the block prior to their quantization and transmission. The system requires ($\log_2$ n/M binary digits per sample overhead information for receiver synchronization.

Still another approach could utilize a variable set of thresholds and reconstruction levels. This is the self-synchronizing approach used in adaptive delta modulators where the step size increases and decreases depending upon the polarity of sequential output levels. In a DPCM quantizer, the set of threshold and reconstruction levels would contract and expand depending upon the sequential utilization of inner or outer levels of the quantizer. For instance, a variable quantizer can be designed where all reconstruction levels expand by a factor of P (for some optimum value of P) upon two sequential happenings of the outermost level and would contract by a factor of 1/P upon two sequential happenings of the smallest level. This system would have the advantage of being completely adaptive and would not require any overhead information because the receiver would be self-synchronizing.

Delta Modulation (DM). A DM system is shown in Fig. 5. The input signal $v_i(t)$ is low pass filtered and sampled to obtain the $(K+1)^{st}$ sample m_{k+1}. This sample is then compared to the internally generated (K+1) estimate, m_{k+1}, and the sign of the difference is then transmitted. Therefore:

$$e_{k+1} = \text{sgn}(m_{k+1} - m_{k+1}) \tag{4}$$

The estimate is formed by adding the previous estimate m_k to a step size of the proper magnitude and polarity.

$$m_{k+1} = m_k + s_{k+1} \tag{5}$$

The step magnitude can be fixed or adaptive. If the step magnitude is fixed the DM system is said to be linear. If the magnitude of the step size is adaptive, the DM system is said to be adaptive.

Linear DM (LDM). In a linear DM the estimate is being updated by adding or subtracting a fixed voltage, s_0, and the new estimate is given by:

$$m_{k+1} = m_k + s_0 e_k \tag{6}$$

where e_k determines the sign and is the quantity transmitted over the channel. A channel error would change the e_k and now the maximum error would be:

$$m_{k+1} - m'_{k+1} = 2s_0 \tag{7}$$

Because the quantization noise depends on the value of s_0

and in general s_0 is small, the error is not very significant.

Fig. 6 shows a signal which changes very rapidly from a low value to a high value. Because the value of s_0 is small it takes the LDM a number of clock pulses to catch up to the signal. This poor slope tracking characteristic of the LDM is called slope overload and makes LDM units somewhat undesirable. The maximum slope an LDM can track is given by

$$\text{Slope Max} = s_0 * f_s = s_o f_s \text{ volts/sec} \qquad (8)$$

This poor tracking performance could be improved at the expense of the quantizing noise by increasing the magnitude of s_0, but that would result in a greatly increased graininess level. A DM capable of combining the good slope tracking capability due to the large value of s_0 together with the good noise figure given by a small value of s_0 is the adaptive DM.

Adaptive DM (ADM). Adaptive algorithms are based on the detection of a number of e_k's of the same polarity occuring sequentially. This condition indicates that the signal is constantly above or below the estimate and the magnitude of the step, sk is increased thus allowing the DM to track a faster rate of change in the input signal.

When the input e_k pattern is alternating 1's and 0's this indicates the signal to be very close in value to the estimate and the magnitude of the step s_k, is decreased such that the granular noise figure is kept at a very low value.

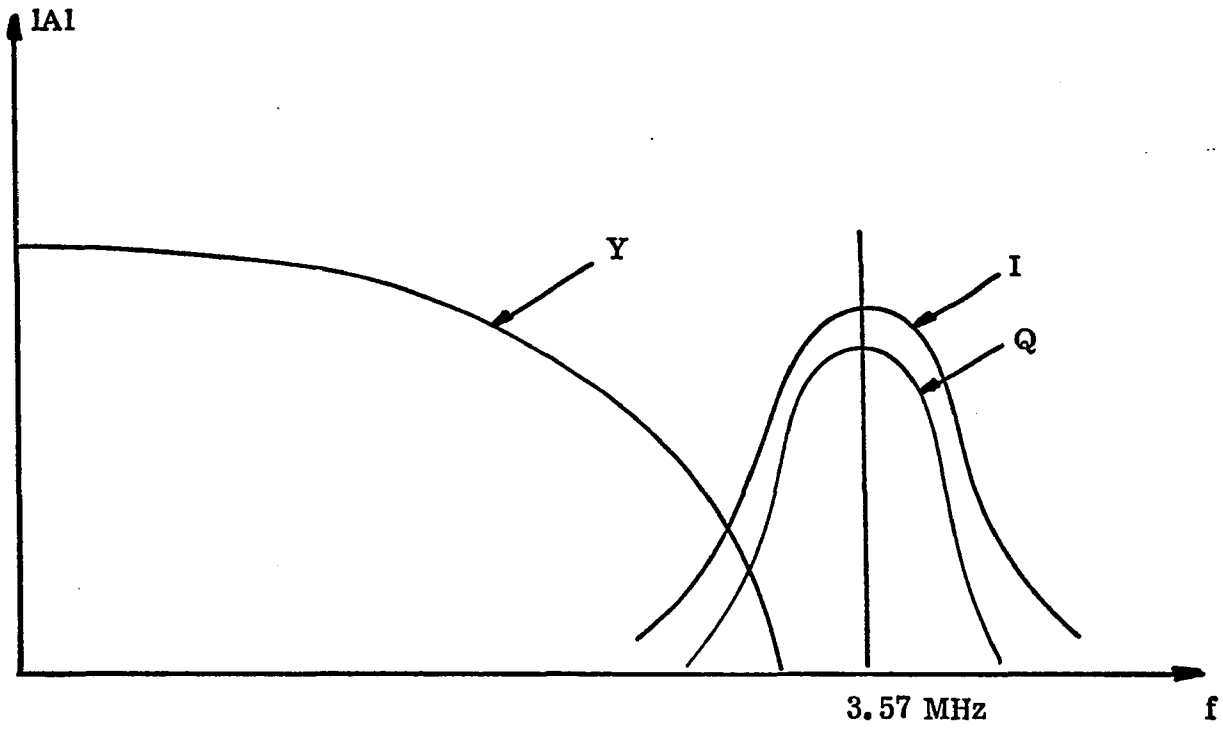


Fig. 1 NTSC Color Video Signal (Spectral Composition)

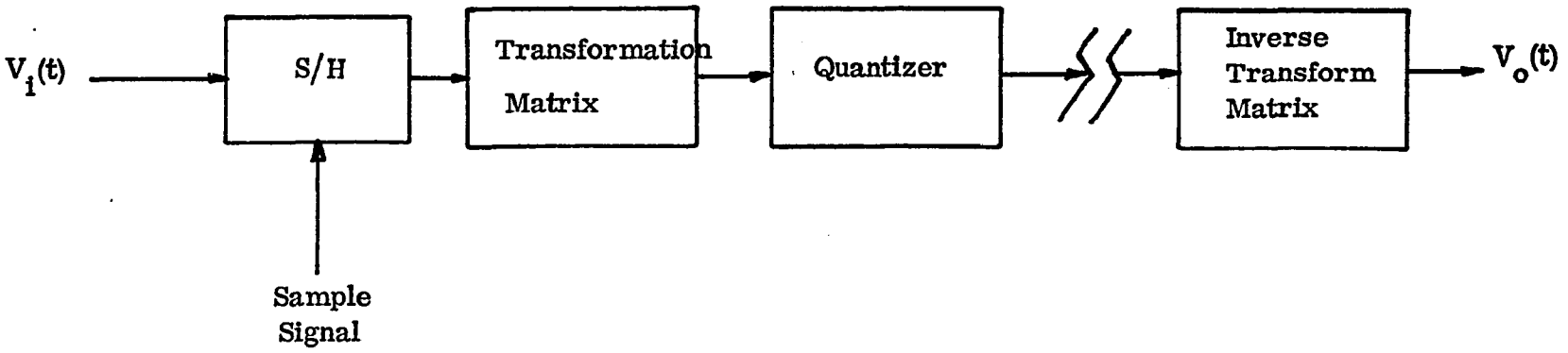


Fig. 2 Transform Encoding System

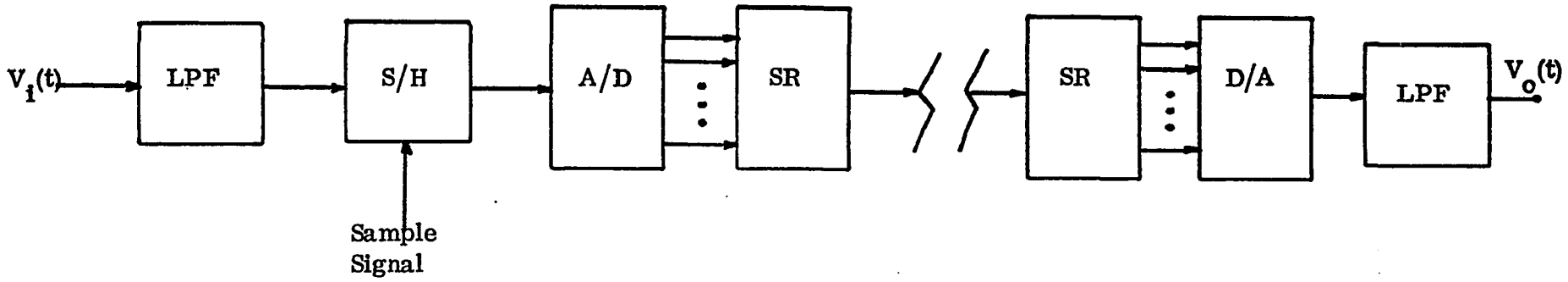


Fig. 3 Pulse Code Modulation System (PCM)

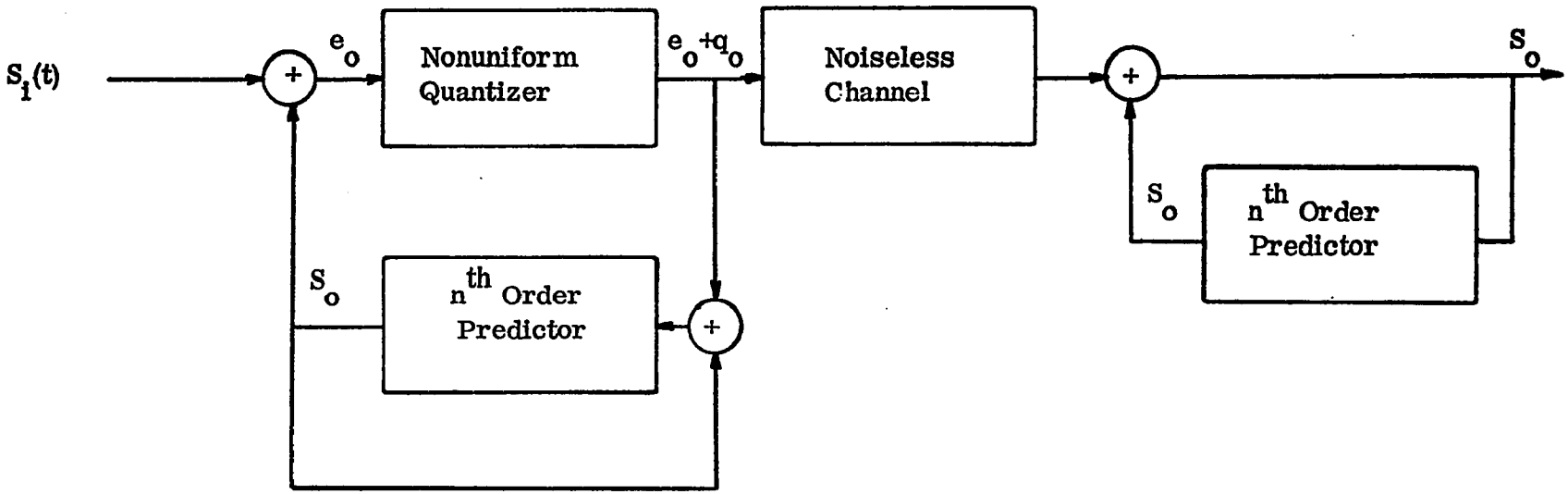


Fig. 4 Delta Pulse Code Modulation System (DPCM)

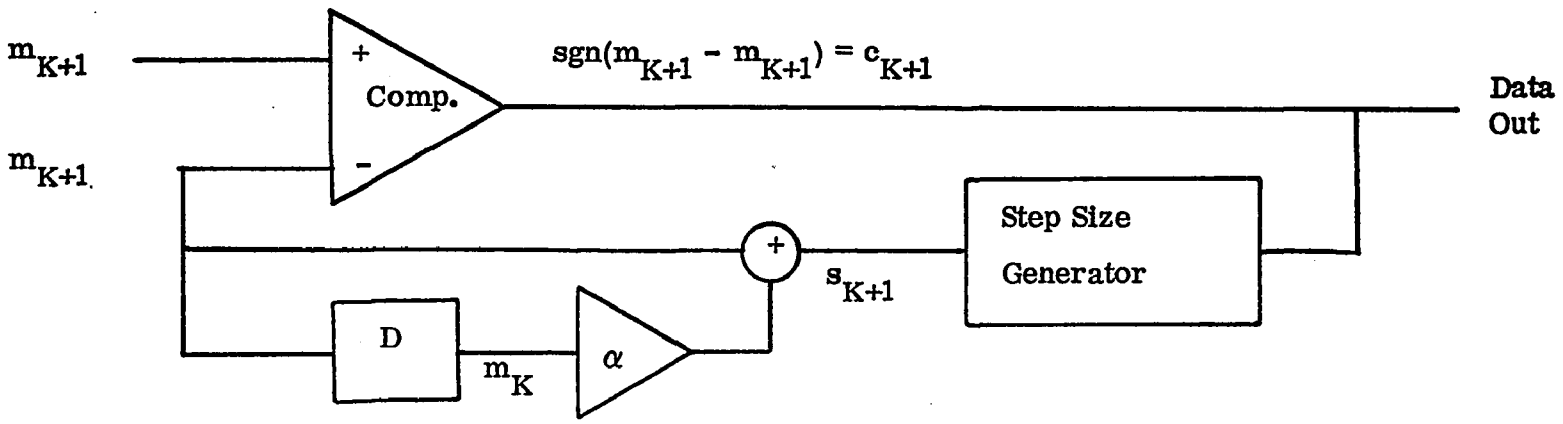


Fig.5 Adaptive Delta Modulation System (ADM)

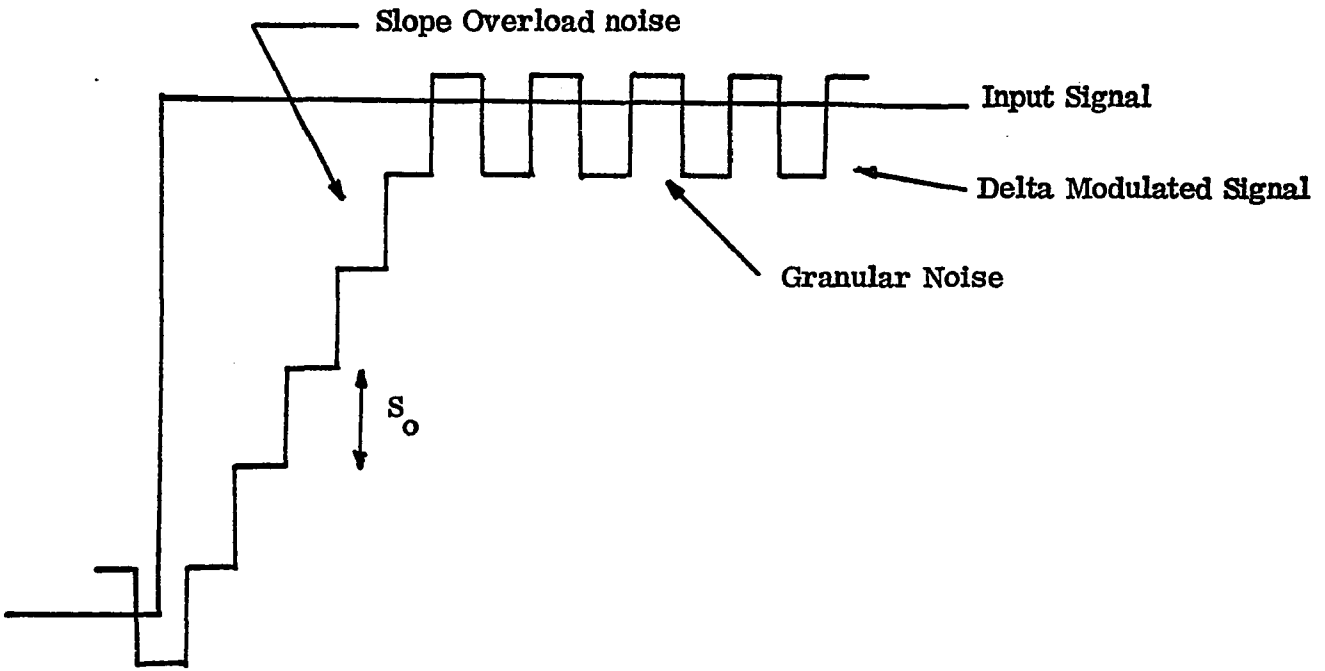


Fig. 6 Slope Overload in Delta Modulation Systems

Applications of ADM to the Encoding of Color Video Signals

NTSC

Our previous experiments have indicated that good video quality can be obtained using ADM encoding of black and white video signals at bit rates of 8 MBPS to 16 MBPS. At a bit rate of 8 MBPS the picture quality was rather poor, suffering from significant edge busyness. As the sampling rate increased, the size of the edge busyness decreased, until at a sampling rate (f_s) of 16 MHz the size of the edge busyness was reduced to approximately pixel size. Even though the edge busyness continued to decrease with an increasing sampling rate, this no longer significantly improved the picture quality. A graph showing subjective picture quality vs. sampling rate is shown in Fig. 7. The encouraging results obtained in the encoding of black and white signals via ADM prompted us to experiment with the encoding of color video signals.

We first attempted to encode the composite signals while compensating for the non-linear low pass filtering effect of the delta modulator. The experimental set up is shown in block diagram in Fig. 8. Our best results were obtained at a bit rate of 24 MBPS (which is identical to the sampling rate) and are shown in Fig. 9. It is clearly seen how the high bit rate manages to transmit a very high quality of luminance information, while somewhat degrading the chrominance information. Our attempts to improve the color quality have shown the poor

performance to be due to improper signal construction. Indeed, while the delta modulator has no difficulty tracking the relatively high amplitude luminance information, it shows a very poor performance when attempting to track the low amplitude, high frequency, quadrature AM modulated chrominance information. This has shown no inherent drawback which prohibits the delta modulator from encoding color video signals, but the quality vs. bit rate of the transmitted signal will be heavily dependent upon choosing the proper format on the signal to be encoded.

RGB

Any color video signals can be represented by its red (R), green (G) and blue (B) components. Therefore, if the R, G and B components are to be sent rather than the composite color signals no information would be lost. This experiment uses more hardware (three sets of delta modulators instead of just one) and also operates at a higher bit rate. It is clear from Fig.10 that the bit rate through the channel is the sum of the bit rates of the R, G and B channels. Based on our black and white experiment results, the similar form of the three color channels and a black and white signal, we expected to obtain good results using 16 MBPS per channel data rate (for a total data rate of 48 MBPS).

The results are shown in Fig.11 for different bit rates. Because the three channels operate independently, the superposition of the three edge busynesses has a cancelling effect

which improves the subjective overall quality of the picture. The bit rate can be reduced to 12 MBPS per channel with only minor degradation. It is at this point that the picture begins to degrade more rapidly until it reaches its lower usable limit of 8 MBPS per channel (24 MBPS total data rate). It is interesting to note that as the picture quality decreases, it is basically the luminance information which is deteriorating rather than the color quality. The RGB experiment has proven the delta modulators to be able to transmit good video quality provided the signal has a proper format. This, in turn, suggests that by restructuring the signal to be transmitted, significant savings in bit rates can be achieved with no loss of picture quality.

IYQ

Introduction. The problem faced consisted of taking three full frequency channels (R, G and B at ~ 4 MHz bandwidth each) and somehow obtaining a different signal set which preserves most of the information while significantly reducing the resultant signal's bandwidth. A similar problem has been solved by the committee which created the NTSC color video standard. By forming the proper linear combination of the red, green and blue components they defined a full bandwidth luminance channel and two (greatly reduced in bandwidth) chrominance channels. The lost information defined the exact color of "small" picture elements. This loss was subjectively insignificant due to the psychovisual properties of the human eye.

The block diagram of the experimental set-up is shown in Fig. 12. This is very similar in hardware complexity to the RGB experiment with the exception of the addition of an RGB to IYQ encoder and an IYQ to RGB decoder. The resulting channel bit rate is again equal to the sum of the individual channel bit rates except that now the I and the Q channels require a much smaller bit rate. Sampling at a rate of two samples per pixel, the I channel will require a bit transmission rate of 6 MBPS while the Q channel will acquire a bit transmission rate of 2 MBPS. Furthermore, at this bit rate the color quality is very good. The Y (luminance) channel responds to delta modulation just like any black and white video signal. By varying the bit rate of the luminance channel the luminance (outline) of the picture degrades due to edge busyness, but the color quality remains very good. The results shown in Fig.13 were obtained by allowing 8 MBPS for the chrominance information and keeping this rate fixed while the rate on the luminance channel is varied from 8 MBPS to 18 MBPS.

This experiment has shown that by proper choice of signals high quality color video information can be sent at bit rates which vary from 26 MBPS for the highest quality, to 16 MBPS for the lowest quality. This total bit rate is lower than the bit rate required for the RGB system which operates at bit rates between 24 MBPS and 48 MBPS while the quality of the transmitted video signal remains approximately the same.

IYQ Encoding in the Presence of Channel Errors. The quality of an IYQ (component) encoded color video signal does not suffer significantly when exposed to channel errors. The I and Q channels are very robust to such errors and are significantly affected only by error rates on the order of 10^{-1}. Figure 14 a,b,c shows a color video signal whose I component is subjected to an error rate of 10^{-1}, 10^{-2} and 10^{-3} respectively, while Fig. 15 a,b,c shows the same signal with the Q channel being subjected to the same error rates. The Y channel is less robust to errors, due principally to the wider bandwidth of this channel as compared to the I and Q channels. The robustness of this channel to errors is the same as expected for a black and white transmission. The effect of channel errors on the black and white ADM encoded video signal has been described in detail by Schilling and Scheinberg (21), for different bit and error rates. Figure 16 a,b,c,d,e shows the effect of channel errors on ADM encoded color video signals at a bit rate of 16 MHz and error rates of 10^{-1}, 10^{-2}, 10^{-3} and 10^{-4} respectively, when these errors occur in the Y channel.

These channel error experiments have shown that a color video signal whose components (IYQ) have been encoded using ADM algorithms, are robust to channel errors. This qualifies the ADM as a robust encoder whether the signal to be transmitted is in black and white or a color video signal.

Hardware Description: RGB to IYQ Converter. The I, Y and Q signals can be obtained from the R, G and B signals by a linear transformation.

$$I = a_1R + a_2G + a_3B$$

$$Y = b_1R + b_2G + b_3B$$

$$Q = c_1R + c_2G + c_3B$$

where the proper values for the a's, b's and c's have been given in the NTSC standard description. Because the transformation is linear, it could have been realized by a resistor weighting matrix. Instead of chosing this approach, an alternate approach was taken which uses active elements and allows buffering of the input signal, thus generating no undesirable side effects such as loading. The first stage buffers the incoming signal via an inverting and noninverting channel obtaining ±R, ±G and ±B. This eases the weighted addition process considerably since it now becomes very simple to add the R, G and B terms multiplied by a negative constant. As Fig. 17 shows, the inverting and non-inverting buffering are done in parallel, which avoids delays between the inverted and non-inverted signal. The second stage acts as an adder stage. By chosing the proper value resistors and passing the proper polarity signals, the I, Y and Q signals are formed at the output of the adders. At this point the converter also adds a fourth signal, the sync, which must be present if the signals are to be compatible with the delta modulators, as the delta modulators use the blanking period to re-initialize all

of the internal registers. The negative level of the sync is determined by the clipping level of the diodes and the height of the sync is determined by the DC level adjust on the summers. The devices can operate with or without a 75Ω load impedance resistors and the output signals will be one volt peak-to-peak under full load. This converter will therefore output a commercial quality IYQ signal with the only difference of the sync addition which serves to make the signals compatible with the delta modulator.

IYQ to RGB Converter. The schematic diagram for the IYQ to RGB converter is shown in Fig. 18. As it could be expected, the operation of the IYQ to RGB converter is very similar to the operation of the RGB to IYQ converter. The two difficulties encountered in the conversion are due to: 1) The presence of a sync pulse and 2) The difference of the slope of the sync information present on the Y channel and that produced on the I and Q channels which operate at a lower check rate, 3) The phase delay between the luminance and chrominance channels, which is due to the different clock rates.

Again, the first stage serves as a buffer and inverter while the second stage does the actual weighting and addition. Now, however, we are confronted by spikes generated by subtraction of two sync signals of different slope. To eliminate these spikes a slightly wider sync pulse is used, which when used in conjunction with the diode clipping stage manages to eliminate all such spikes.

The reconstructed picture will show severe chrominance misalignment. This effect is due to the much lower sampling rates on the chroma channels. To compensate for this a shift register has been built with 36 stages. A four to one multiplexer selects different delay taps at the output. Fig. 19 shows this tapped delay line. The multiplexer's select inputs are controlled by two switches which are set according to the frequency of the sampling clock on the luminance channel. The four settings will give exact alignment for a sampling frequency of 8, 12, 16, and 20 MBPS. When operating the system with the delay line, the chroma and luminance information will coincide and the displayed picture will have a good quality. A second order effect which is visible at times, is the slow color transition of boundaries. This is caused by the slow sampling rate on the chrominance channels and is not of much concern, unless the video image consists of color stripes or other such specialized, stationary inputs.

The IYQ to RGB converter is very similar to commercially available units except for the special delay and sync cancellation features. The amplifiers used on this unit have a wide enough bandwidth to allow full frequency video operation. When showing typical scenes the amount of movement has no effect upon the picture quality.

Line Sequential (LS)

Introduction. The previous experiments have indicated that by using three pairs of delta modulators and little outside

circuitry good quality video can be transmitted provided that the available channel can accomodate a bit rate of at least 16 MHz (the lowest usable quality for IYQ encoding). However, for certain applications a user might be able to tolerate a loss in picture quality, provided the bit rate can be further reduced. A line sequential system attempts to do just that by using an encoding scheme which reduces the bit rate required by the RGB encoding scheme to one third of its value. Figure 20 shows that in order to accomplish this bit reduction we transmit only one of the three color channels for any given line. Assuming that we are encoding the odd field (lines 1, 3,5,7...) we encode and transmit only the red information of line one, and then encode and transmit only the green information of line three and, similarly, we encode and transmit only the blue information of line five. At this point the cycle is completed and the whole color vs. line selection restarts on line seven. In general, the encoder will transmit the red information of line (K-2), the green information of line K, and the blue information of line (K+2). The receiver has available two lines of memory which will store the two previously transmitted colors. To display a video line, the receiver will combine the two stored colors with the presently arriving color, while simultaneously updating its stored information. The results obtained with the line sequential system are shown in Fig.21. The color quality is seen to be very good throughout with the possible exception of the horizontal

edge transitions. Because we encode every other line (i.e., 1, 3, 5, 7, ...) a smearing in the vertical direction occurs which tends to cause some color blending and flicker over small horizontal transition areas. Similarly, this vertical smearing affects small horizontal curvatures and any other small details in the vertical direction which seem to get rather washed out. The effect of this encoding method on slanted lines is to create staircase patterns which even though highly visible do not appear to be very annoying. The color quality remains constant with variations in the bit rate while the horizontal smearing increases at the lower bit rates. Even though a vertical averaging takes place, the amount of movement in the scene has little effect upon the picture quality. This is the result of not operating a field but rather a line sequential system which updates fast enough to adequately represent motion.

This system represents a minimal bit rate color video system. It shows that provided the user can afford a certain amount of picture degradation, it is possible to transmit good color video at low bit rates (larger than or equal to 8 MBPS) and using hardware of medium complexity.

LS Encoding in the Presence of Channel Errors. A line sequential system operates by alternating the color information which is encoded and transmitted on every scan line. At the receiver a video signal is formed and displayed by combining the presently received color channel with the other two

previously received color channels. This transmission of only one color at a time ensures a "softening" of the effect of the errors which now can only affect one color component at a time. This is shown in Fig. 22 a,b,c,d where an LS encoded video signal is transmitted at a rate of 8 MBPS and the bit error rates are 10^{-1}, 10^{-2}, 10^{-3}, and 10^{-4}, respectively.

This experiment proves that the LS encoding of color video signals via ADM is a robust, low bit rate video encoding method which could be advantageously used provided the basic system limitations are not an overiding factor.

LS Hardware Description: Control Unit Hardware. The schematic diagram of the control unit is shown in Fig.23. The clock input is buffered by a 7404 inverter. This inverter drives four other inverters which share the load. This makes the control unit appear as only one TTL load and, at the same time, it helps preserve the shape and rise time of the clock waveform. This is very important in any synchronous system where a number of operations are occurring concurrently and any mistriggering would have serious consequences upon the system's operation. The two 74164 serial to parallel converters form a delay line. The 74193 counter uses its carryout output to load the converter with a 10 ... 0 pattern. Every clock pulse the high output will shift one position. For example, after the first occurrence of the clock, the new output will be 010 ... 0. The Q_o through Q_{15} outputs will therefore specify sixteen distinct phases, which will be used to direct set and

reset a 7474 flip flop such that its output Q will have the proper timing for the memory write enable ($\overline{WE}$) signal.

A 7495 binary counter is wired up to form a modulo three counter. The horizontal drive activates its "count in" input and its output indicates which line (R, G, or B) is to be transmitted. Should any power supply noise mistrigger the counter, the error will not propagate to the next frame because at the beginning of every frame the counter is reset by a new frame pulse ($\overline{NFP}$). The $\overline{NFP}$ signal is obtained by transfering the horizontal drive ($\overline{HD}$) data which is present at the input of a D type flip flop to its output Q, on the rising edge of the vertical drive signal ($\overline{VD}$) which is present at the clock input Q will indicate whether the odd or the even field is being transmitted. To create a new frame pulse the field indicator (Q) triggers a 74123 one shot for a duration of approximately 400 nanoseconds (ns).

The outputs of the 7495 modulo three counter are first decoded and then used to enable three AND gates which will generate the $\overline{WE}$ signal for the proper memory bank. Therefore, one memory will always be updated as the other two memories are being read. The memory addresses are provided by two 74193 counters, the first one being driven by the 16 megahertz clock. To insure the proper phase between the data out and clock signals, a tapped delay line is used which provides five taps with a delay of five nanoseconds per tap.

The 7495 counter, which operates in a modulo three mode,

drives the select inputs of a four to one multiplexer. The multiplexer chooses one of the input channels (R, G, or B) and outputs it. The output of the multiplexer represents the actual channel. The whole control units fits on a 5" x 5" card and contains all of the logic necessary to generate every control signal. This, in turn, allows the memory storage units to be much simpler, which is a definite advantage in the actual operation of the system.

Memory Unit Hardware. A schematic diagram of the memory unit is shown in Fig. 24. Each memory receives all of its control signals and data inputs from the control unit. The memory data out signal does not go back to the control unit but is outputted directly. The memory uses a "memory multiplexing" scheme. The data in is shifted in the 74164 serial to parallel converter until the shift register is full (16 bits). The contents of the shift register are then latched in a number of 74174 latches. The memory unit, which is organized as 64 x 16, will write the data from these latches. When the memory is read, its output is again latched in 74174's and these latches are connected to a parallel to serial converter. The data is read into the converter at a rate of one megahertz and it is shifted out of the converter at a rate of sixteen megahertz. It is readily apparent how this "memory multiplexing" technique allows relatively slow memory units to operate at much faster apparent rates. In this design the memory speed multiplication factor is sixteen. The data out

signal is chosen by a two to one multiplexer from two different data paths. When the memory is being updated, the video signal is formed by the present signal and the two previous colors. Therefore, when the R/W signal is in the write mode, the data out comes from the input serial to parallel converter. The extra delay is necessary for the proper alignment of the three channels. While one memory is being updated, the other two memories are being read. When read, the memory's output is latched, converted to serial form, and shifted out via the multiplexer. The maximum speed of operation for the memory units is limited to ~ 17 MHz by the input shift register. Due to their conceptual simplicity and low chip count, the memory boards have been built with PC boards which made the task of building and debugging the memory units feasible.

Conclusion

A number of color video encoding schemes have been investigated. Even though they all use delta modulators to encode the actual signals, their individual advantages and disadvantages are quite different. This makes an overall comparison extremely difficult. Fortunately, the range of bit rates that the individual systems can operate over are largely non-overlapping. This makes it possible to say that given a requirement for the best possible quality, and given a channel which can accomodate a rate of 24 MBPS to 48 MBPS, an

RBG encoding system would be a proper choice. An IYQ system operates almost as well but requires a substantially lower bit rate. To operate such a system the channel must be able to accommodate a bit rate of 28 MBPS to 16 MBPS. If the channel can only accomodate a bit rate of less than 16 MBPS but higher than 8 MBPS, and if the vertical degradation seems to be unobjectionable, then a line sequential system is the only remaining choice.

We conclude that the above research has accomplished its goal: all of the above methods are able to transmit a video signal with a good color quality. The degradation increases as the available bit rates decrease. In all cases the systems perform better than any PCM system at the same bit rate. Furthermore, when the bit rate is reduced it is not the color but the luminance information which degrades. Given the fact that the ranges of operation are not overlapping, it is possible to specify the best system to be used for any particular application.

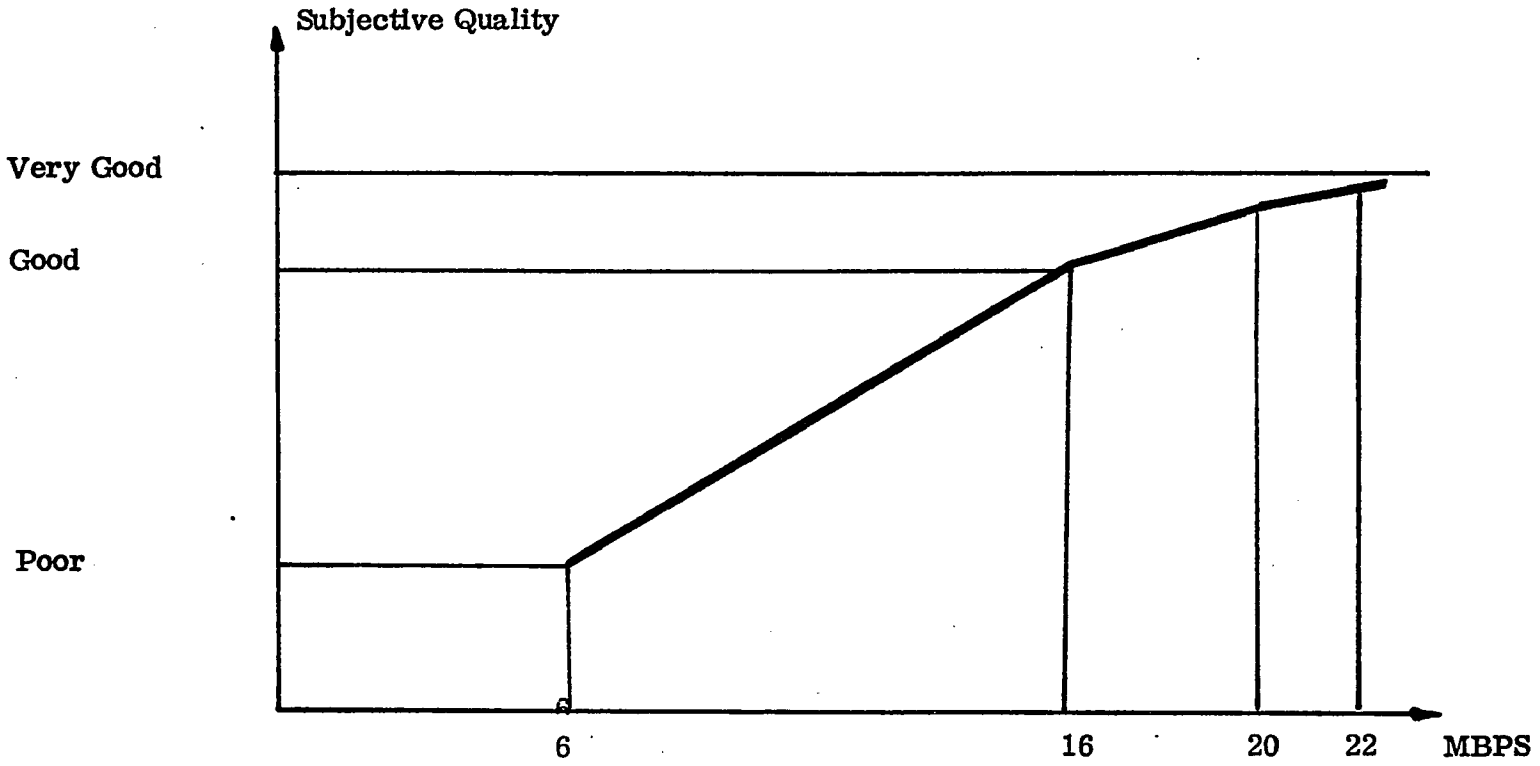


Fig. 7 Subjective Picture Quality versus Bit Rate

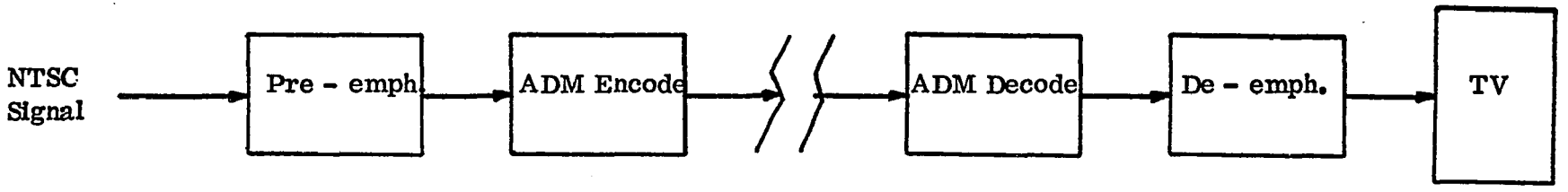


Fig. 8 NTSC Encoding of Color Video Signals

a) 16 MBPS

b) 24 MBPS

Fig. 9 NTSC Encoded Color Video Signals

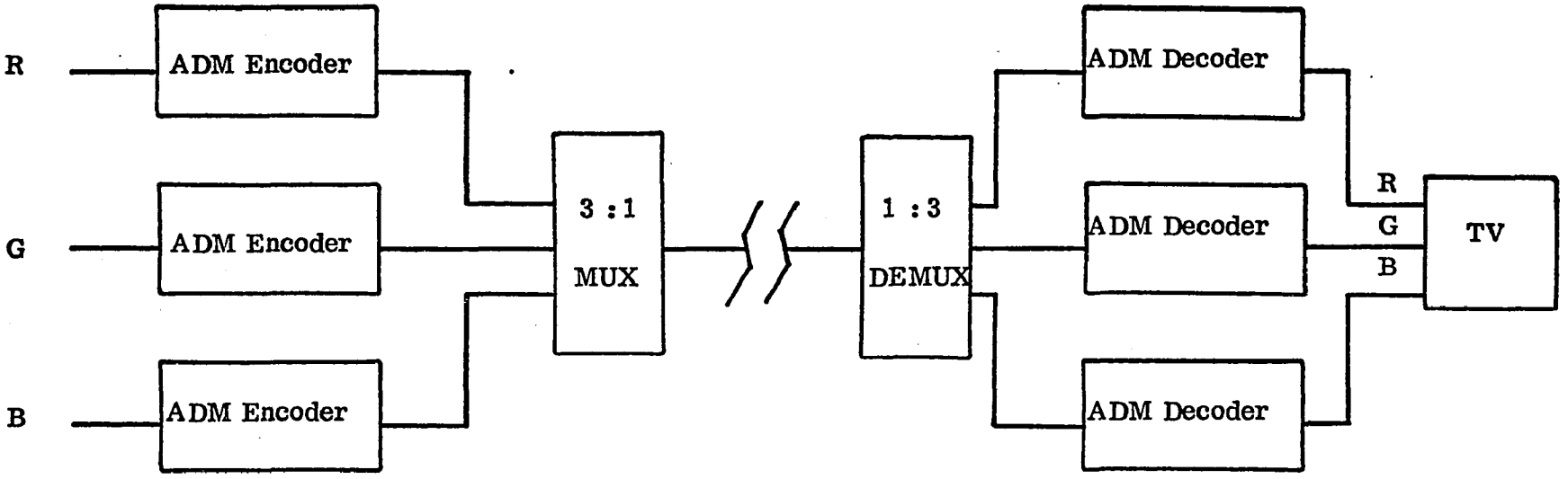


Fig. 10 RGB Encoding of Color Video Signals

a) 16 MBPS

b) 24 MBPS

Fig. 11 RGB Encoded Color Video Signals

c) 36 MBPS

d) 48 MBPS

Fig. 11 RGB Encoded Color Video Signals

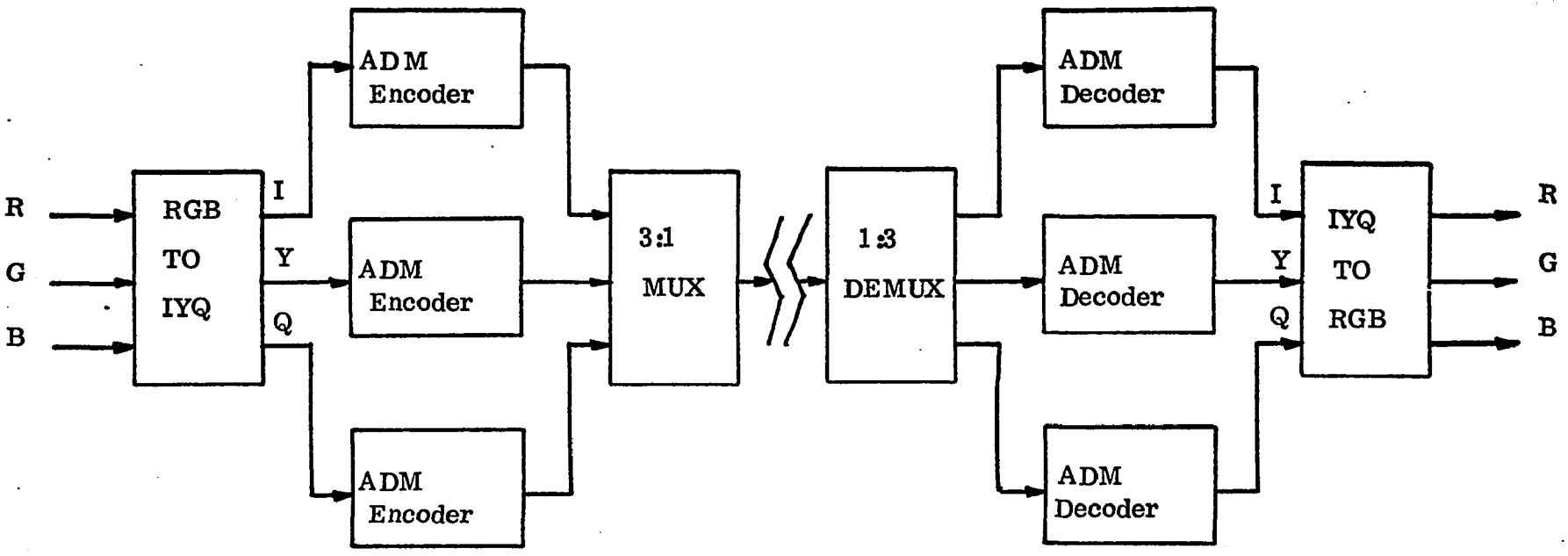


Fig. 12 IYQ Encoding of Color Video Signals

a) 16 MBPS

b) 20 MBPS

Fig. 13 IYQ Encoded Color Video Signals

c) 24 MBPS

Fig. 13 IYQ Encoded Color Video Signals

a) $P_e = 10^{-1}$

b) $P_e = 10^{-2}$

Fig. 14 IYQ Encoded Video Signals with I Channel Errors

c) $P_e = 10^{-3}$

Fig. 14 IYQ Encoded Video Signals with I Channel Errors

a) $P_e = 10^{-1}$

b) $P_e = 10^{-2}$

Fig. 15 IYQ Encoded Video Signals with Q Channel Errors

c) $P_e = 10^{-3}$

Fig. 15 IYQ Encoded Video Signals with Q Channel Errors

a) $P_e = 10^{-1}$

b) $P_e = 10^{-2}$

Fig. 16 IYQ Encoded Video Signals with Y Channel Errors

c) $P_e = 10^{-3}$

d) $P_e = 10^{-4}$

Fig. 16 IYQ Encoded Video Signals with Y Channel Errors

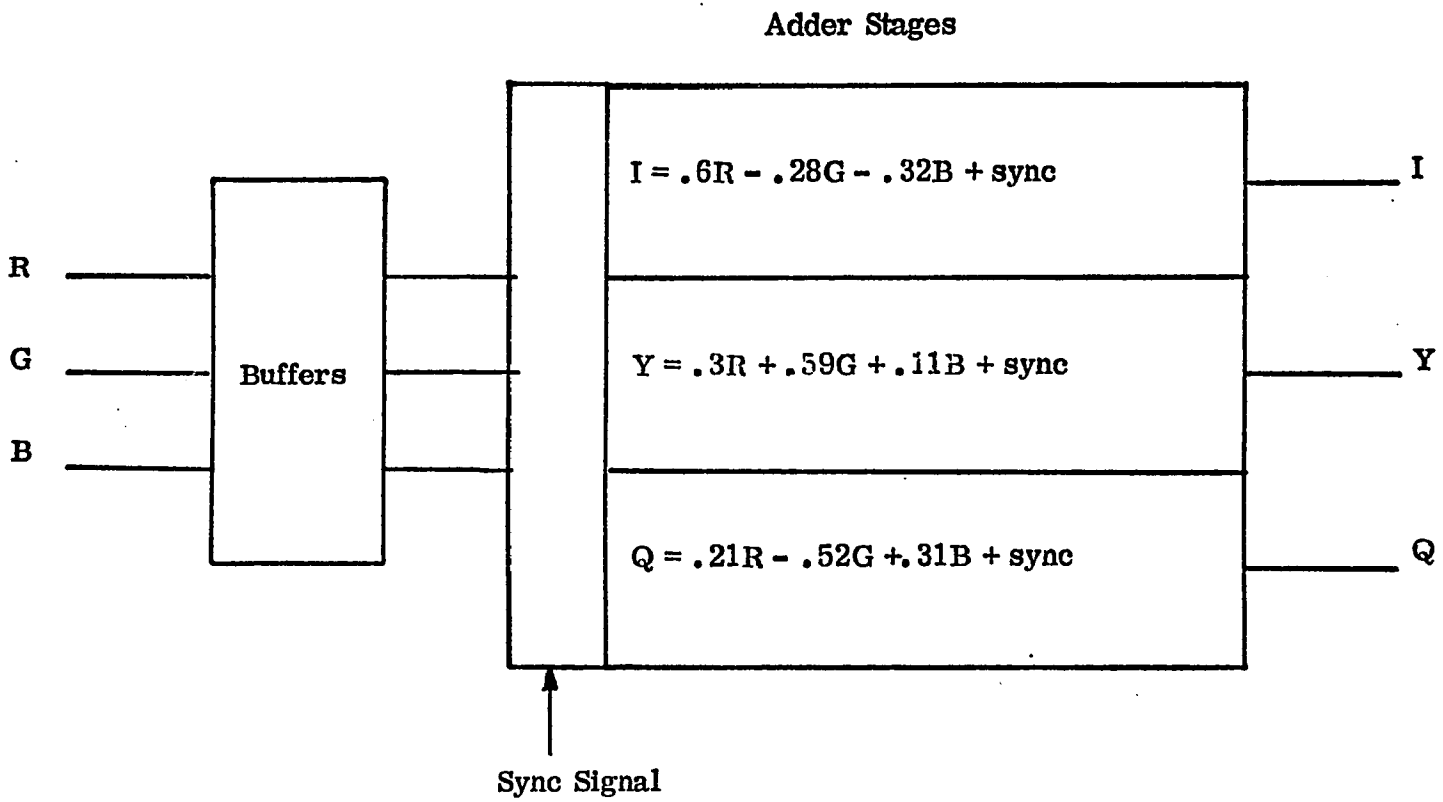


Fig. 17 RGB to IYQ Converter

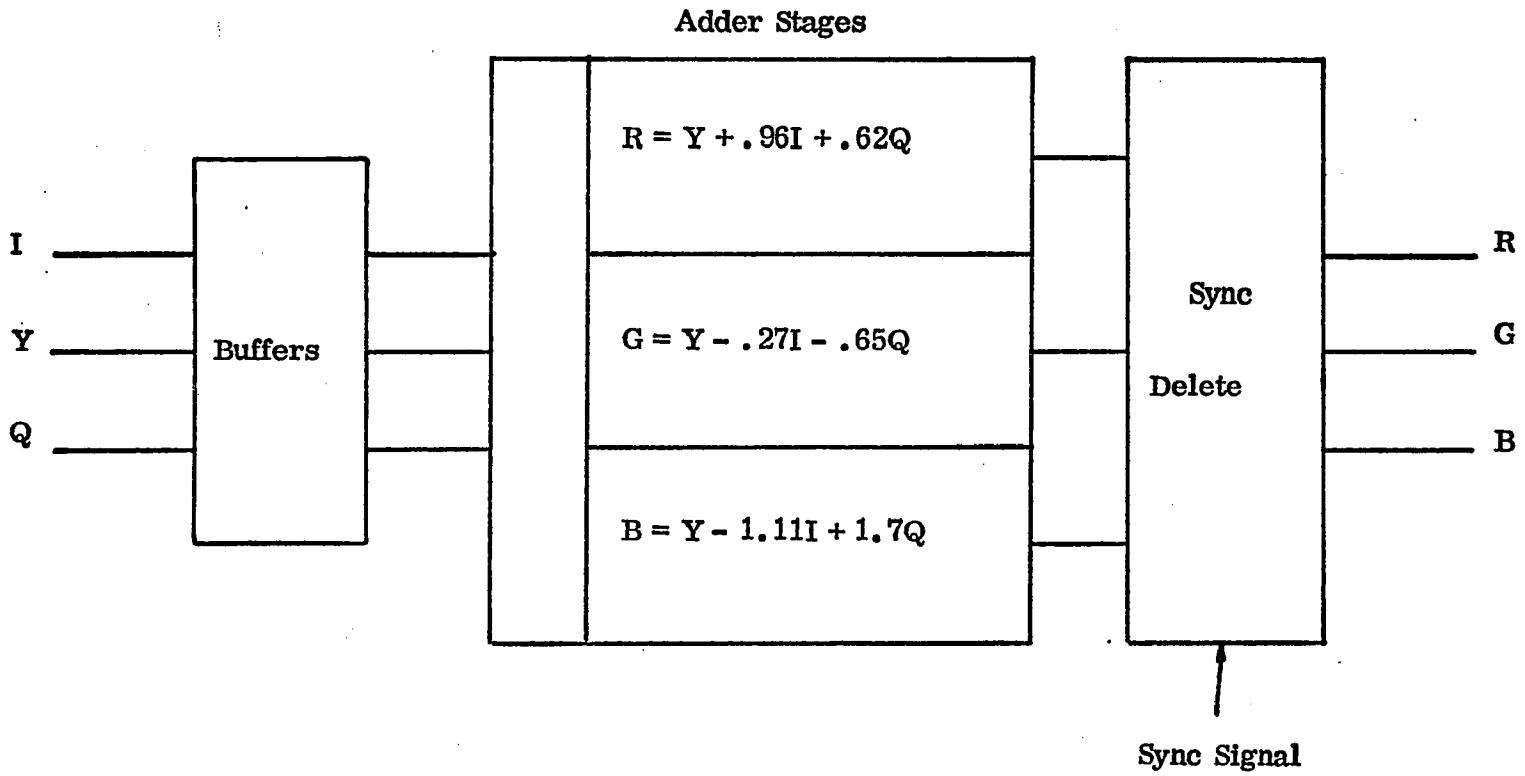


Fig. 18 IYQ to RGB Converter

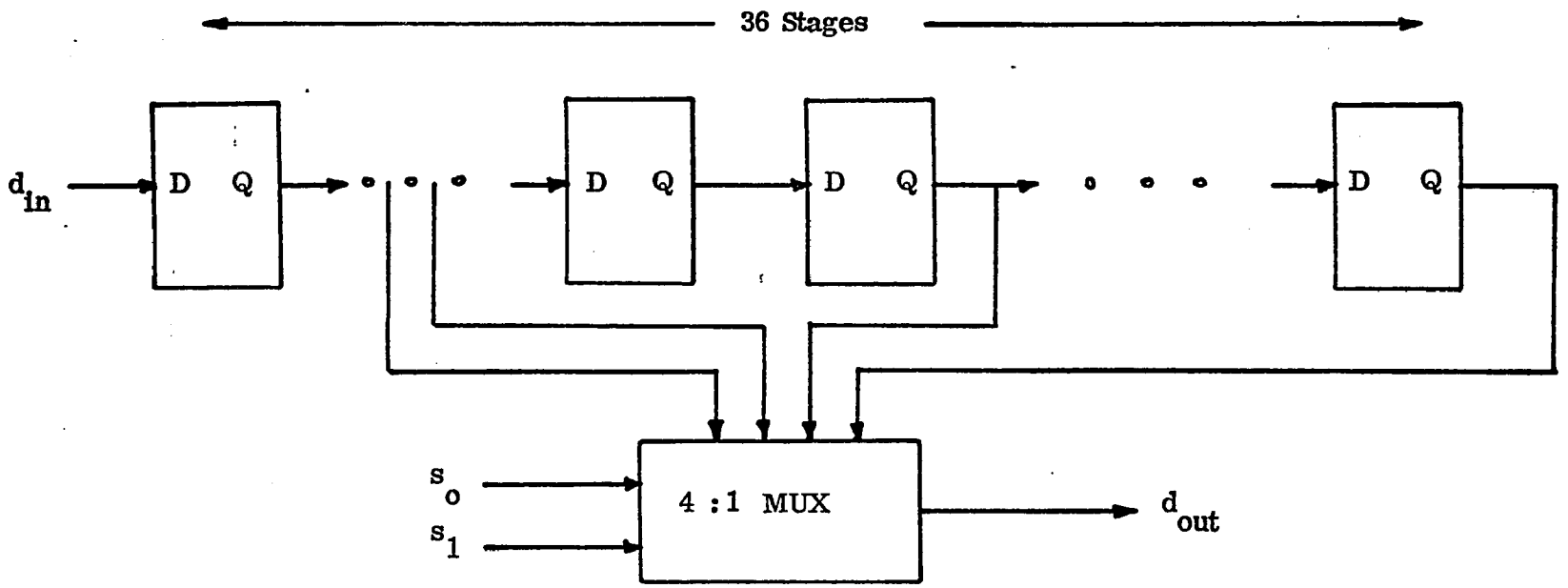


Fig. 19 Tapped Delay Line

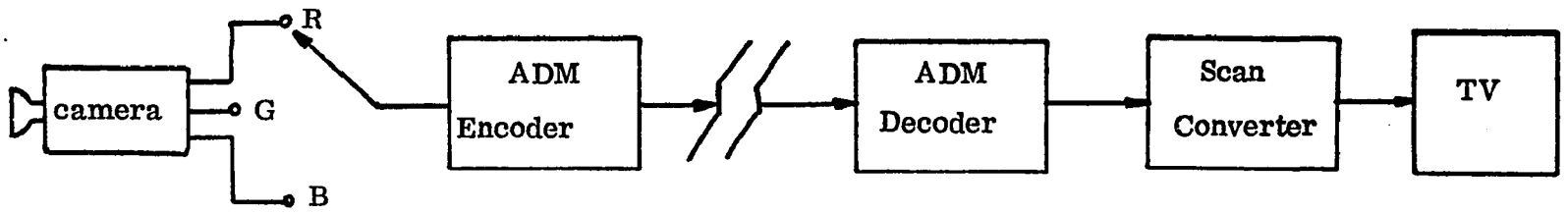


Fig. 20 Line Sequential Encoding of Color Video

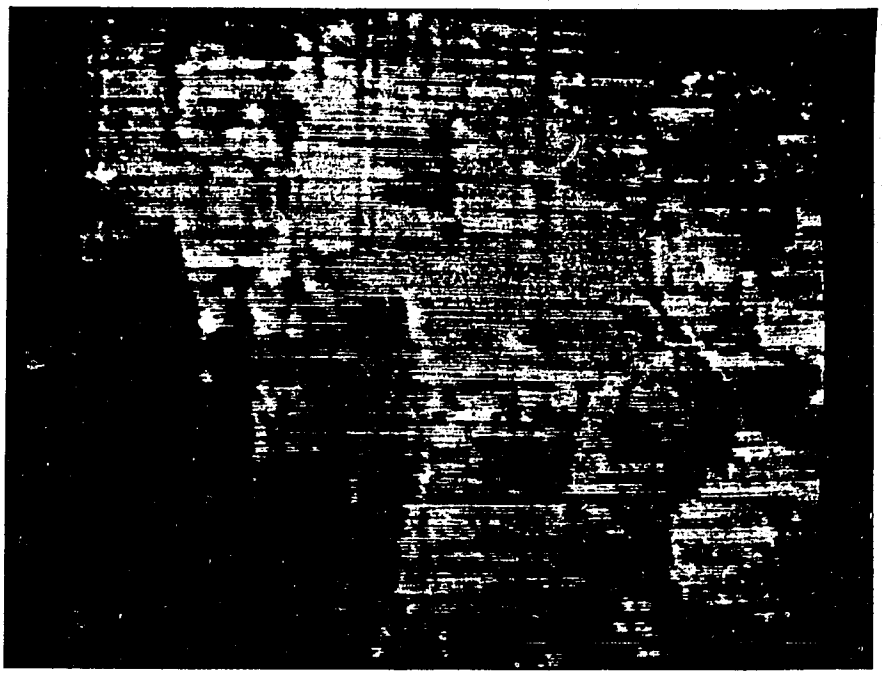
a) 8 MBPS

b) 12 MBPS

Fig. 21 Line Sequential Encoded Color Video Signals

c) 16 MBPS

Fig. 21 Line Sequential Encoded Color Video Signals



a) $P_e = 10^{-1}$

b) $P_e = 10^{-2}$

Fig. 22 Line Sequential Encoded Color Video Signals with Channel Errors

c) $P_e = 10^{-3}$

d) $P_e = 10^{-4}$

Fig. 22 Line Sequential Encoded Color Video Signals with Channel Errors

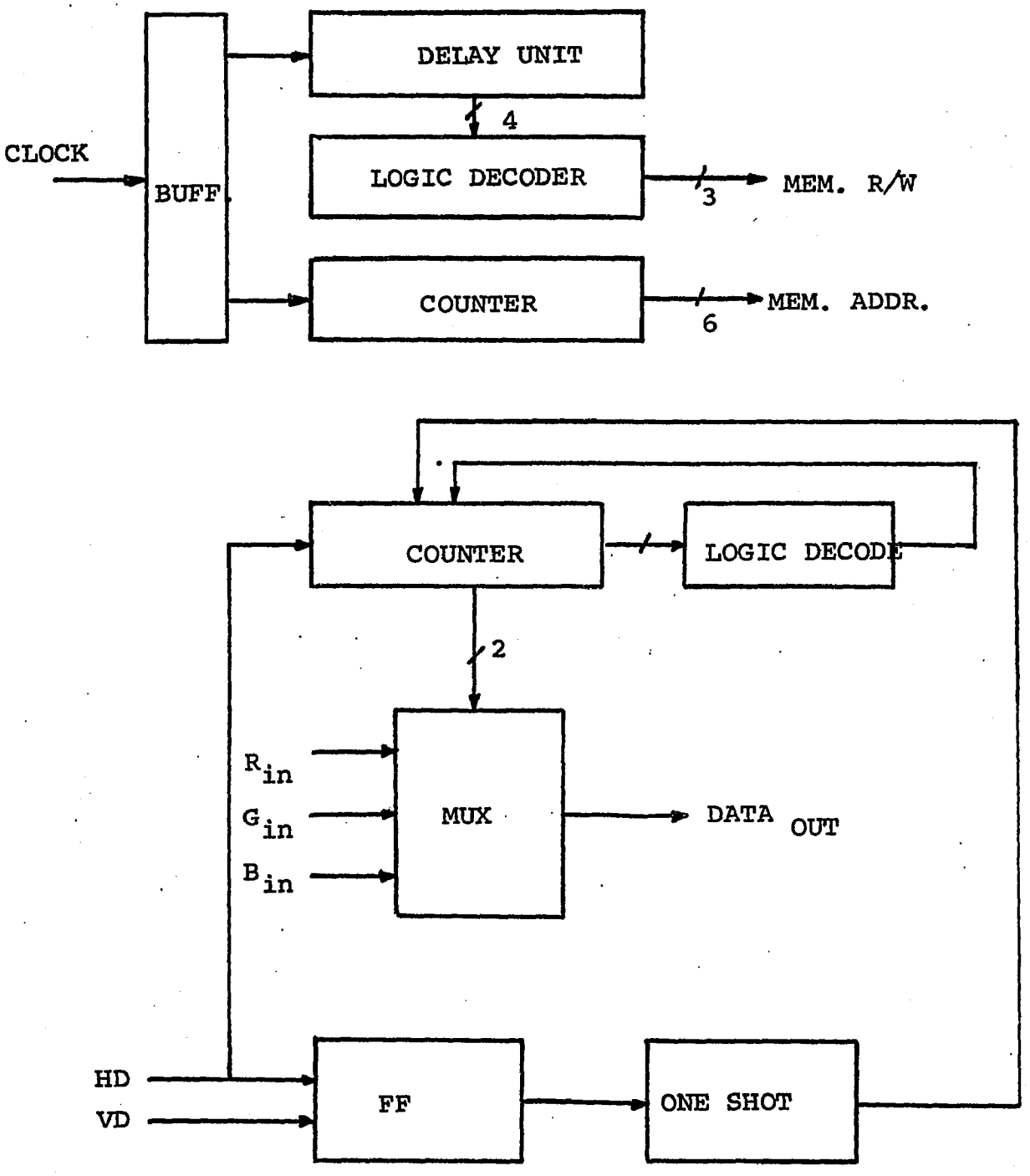


Fig. 23 Line Sequential Control Unit

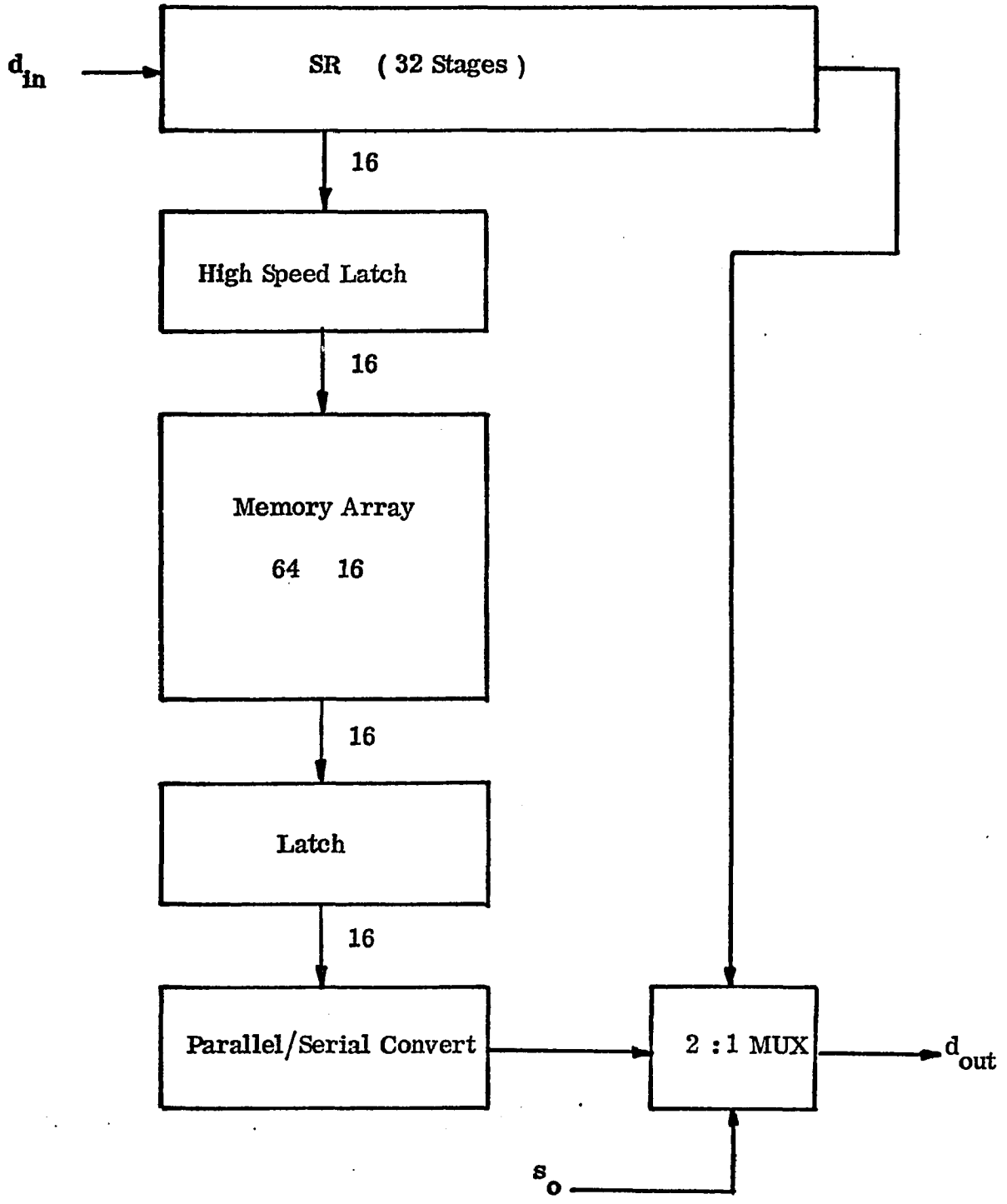


Fig. 24 Line Sequential Memory Unit

APPLICATION OF ADM TO PACKET VIDEO TRANSMISSION

Introduction

The appearance of inexpensive, powerful computers has encouraged their widespread application in almost every conceivable domain. One of their most demanding applications in the communications field is in a traffic supervisory role. Having linked many centers of human activity, the designers next attempt to make the resulting communication process as efficient as possible. As a result, several communication nets came into being which allow direct computer to computer transmissions. For these nets to be effective, new protocols and transmission philosophies had to be created. The need for efficient operation of the net has resulted in computers controlling the local routing and traffic destination in every part of the net. The traffic now consists of messages which are divided into groups of a pre-assigned number of bits, called packets. The local controller adds a destination address and then it proceeds to send the packet via the least busy route.

The largest of these nets is the ARPA net, operated in the U.S. by the Department of Defense.

An investigation was launched to evaluate the kind of quality one could expect from voice and video transmissions over such nets. Reference (22) describes the work done by

Tanaka, Dressler and Chackavarty in the simulation of voice transmission via computer communication nets. The voice was digitally encoded via an ADM scheme which resulted in a much lower bit rate than that corresponding to PCM encoding. Since the sampling rate involved in voice communications is relatively low, there was no need for much support hardware, as much of the processing was done in real time. Video rates prohibit a direct ADM-computer interface, with the speed of the computer being the limiting factor. Therefore, two methods of video acquisition had to be investigated. One would rely on vertical scanning to generate the pixels. Fig.25 shows how a vertical scanner samples all of the n^{th} pixels from every line during a given frame. Every new frame the scanner proceeds to scan the $(n+1)^{st}$ pixels, until the whole frame has been transmitted. By using this scanning method the sampling rate is reduced to less than 16 KHz which is low enough to be handled by a computer directly. The difficulty lies in the video reconstruction process at the receiving end. Even though the receiving computer may have the required memory, it is doubtful that it could sustain the high data rate required for a real time video display. Therefore, the receiver had to have a digital frame storage which had to be controlled by a custom built controller optimized for this application.

Through careful design the same frame storage can be used for both, the transmission and reception of the digital video frame. Due to the high data rates necessary for the transmission

of video images, a decision has been made to use an ADM encoder because this would minimize the bit rate while maintaining an acceptable quality level.

Hardware Description

Fig. 26 shows the controller block diagram. Again, inexpensive, relatively slow memories are being used at an apparent rate which is much higher than their maximum read/write frequency. This is accomplished via the memory multiplexing technique which has been described previously. The data stream is stored in groups of 16 bits which are first converted to a parallel format. The maximum data rate is 16 MBPS which corresponds to roughly 1K bits/line of video. Therefore, each line of video requires at most 64 memory locations of 16 bits/location. The total number of lines in a video frame is 525 and, therefore, the total storage capacity for the frame of memory is 32K x 16. 32K locations require a 15 bit address. Because the individual memory chips have a 4Kx1 internal organization, the low order 12 bits of the address are being used to address each individual chip while the three high order bits are decoded and used to select any one of eight rows of memory chips. This memory organization of 8 x 16 chips is shown in Fig. 27. Synchronism with respect to the camera driving signals is achieved by using the $\overline{NFP}$ and $\overline{HD}$ signals. In addition, the chock used with the ADM encoders is synchronized to the $\overline{HD}$ signal. The read/write command is

synchronized to the $\overline{NFP}$ pulse and guarantees reading or writing an entire frame. Once the memory unit has been completely filled with one video frame it goes into a read state which performs the continuous memory refresh and enables the operator to inspect its contents visually (via an ADM decoder and a TV monitor). Provided the stored frame is satisfactory provisions have been made for a computer interface which has a random access feature, yet it does not interfere with the refresh cycle.

A block diagram of the computer interface is shown in Fig. 28. The computer address is loaded into the upper nine address bits. The remaining six low order bits cannot be jam-loaded from the computer as this would interfere with the refresh cycle. Instead, the low order six address bits are compared to the computer requested address. When a match is made, the data is loaded into the latch. The computer can then load it into its memory and format it for transmission. The maximum waiting time for the data and latch to be loaded (the memory limitation in speed) is given by the maximum time it takes for the address to cycle through $2^6 = 64$ states. At one microsecond per state it takes 64 μs for a complete cycle through and $64 \times 10^{-6} \times 32 \times 10^3 \cong 2$s for the loading of a complete video frame (e_k's). The computer can then format and transmit the video information to any other computer on the net. In turn, it can also receive a video frame and use the same hardware to display it. Upon receiving a frame of e_k's, the host computer

can, via the interface, load the frame of e_k's into its video memory and the operator can view the video image via an ADM decoder on a TV monitor.

Conclusion

An inexpensive digital storage has been described which serves as an interface/buffer between a high speed digital data stream coming from a video ADM and a computer. The same device acts as an interface/buffer in the opposite direction, allowing the computer to load it with a frame of e_k's and then displaying the video frame in real time via an ADM decoder and a TV monitor. The pictures are either black and white or color, provided that a line sequential technique is being used. The maximum bit rate is 16 MBPS. The technique used is independent of the type of memory used, and a controller could be built to operate with any video display such that a full video feature could be added to any terminal provided it has the required amount of memory (32K × 16).

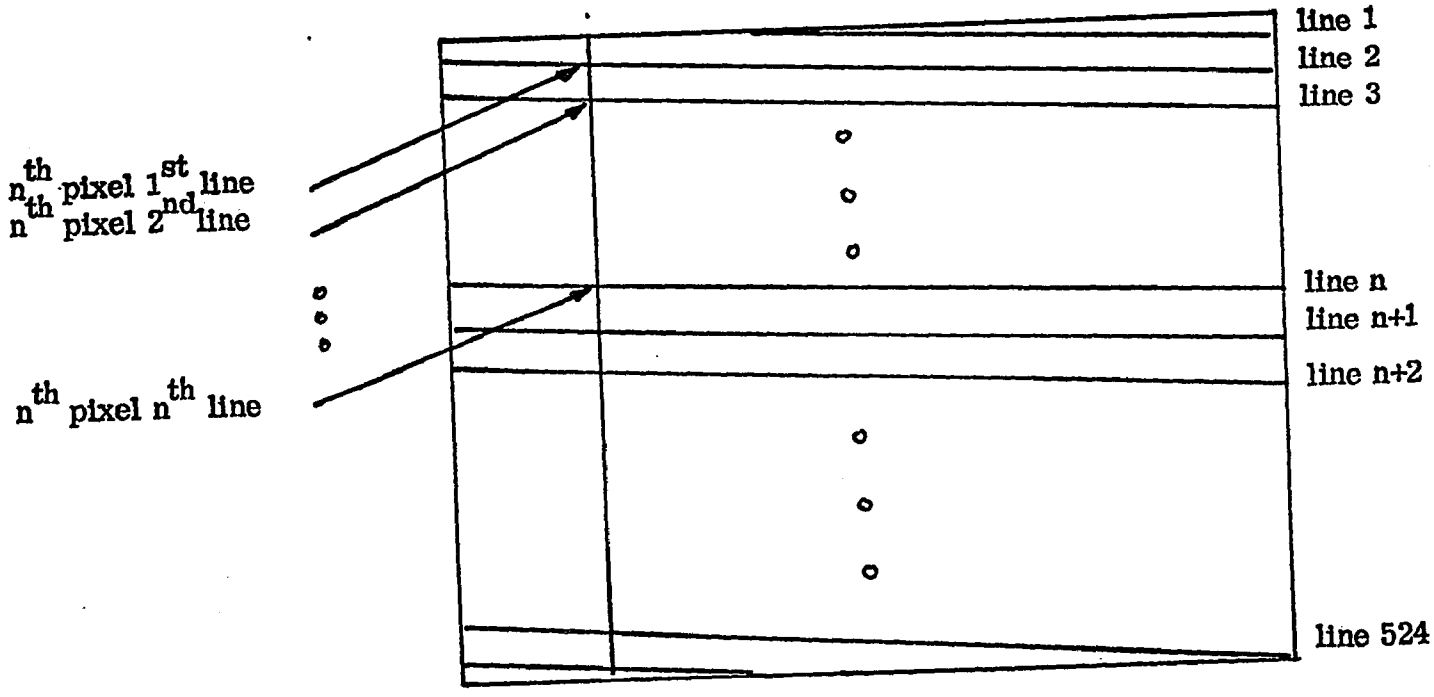


Fig. 25 Vertical Scanning of Video Signals

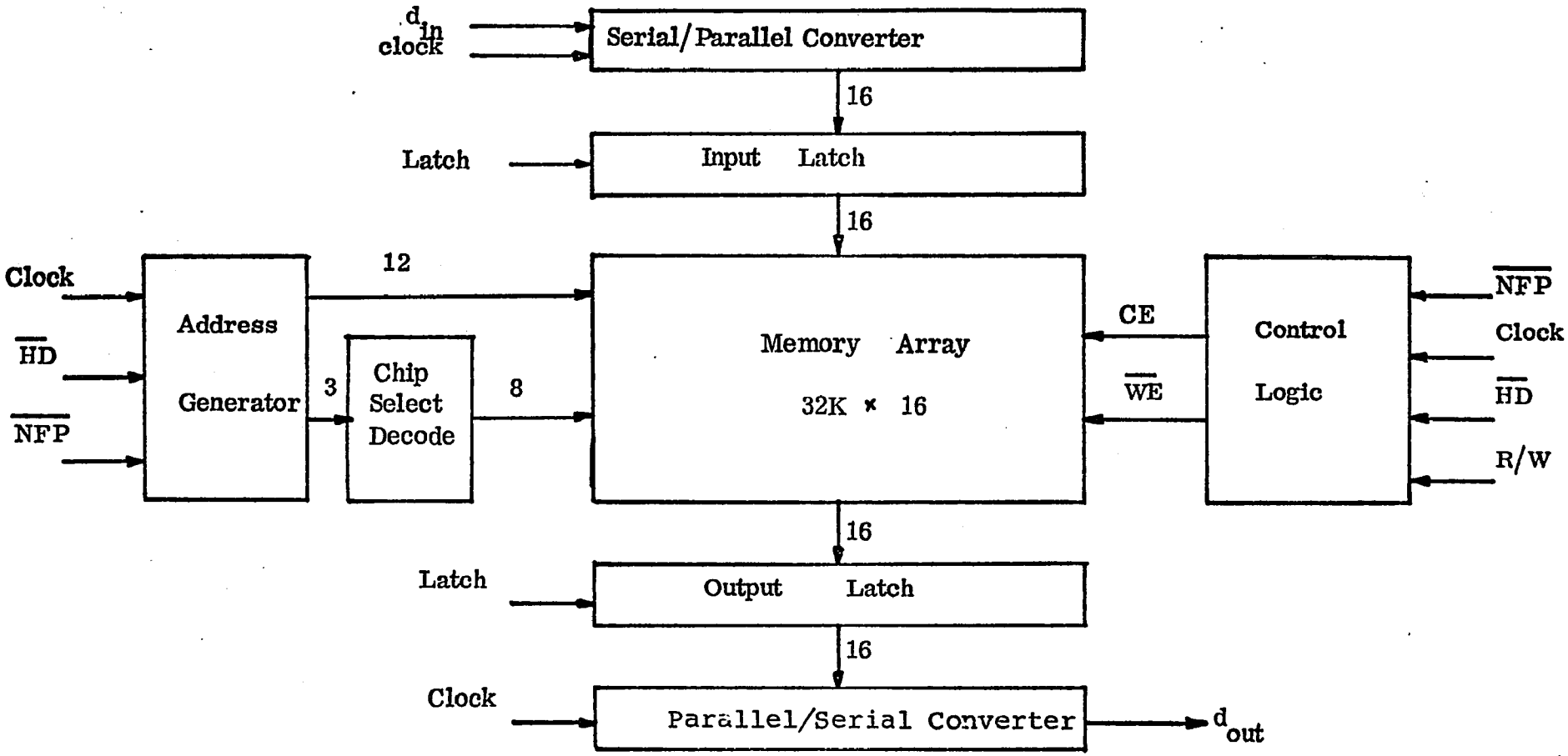


Fig. 26 Slow Scan Controller Block Diagram

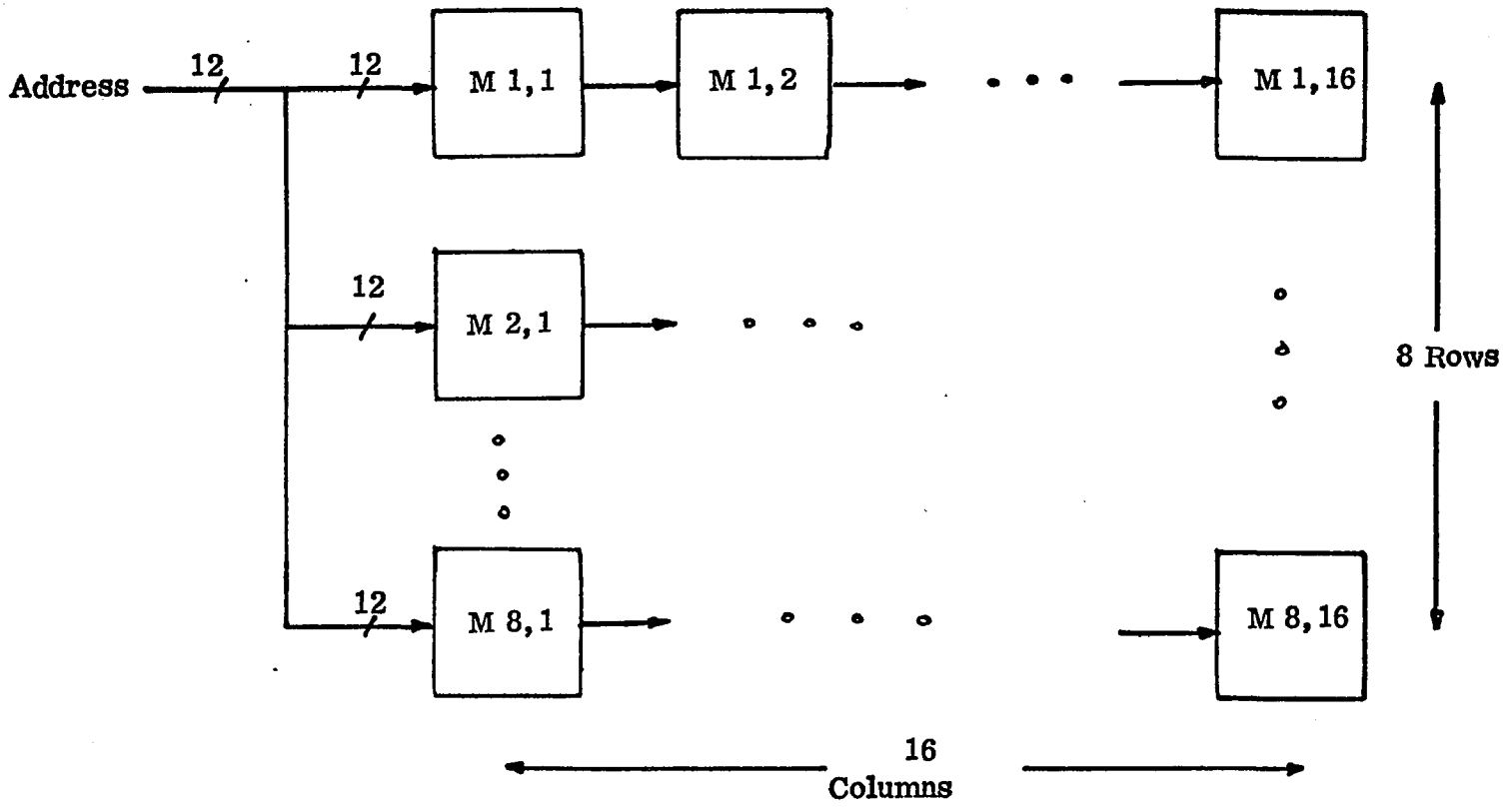


Fig. 27 Memory Array Organization

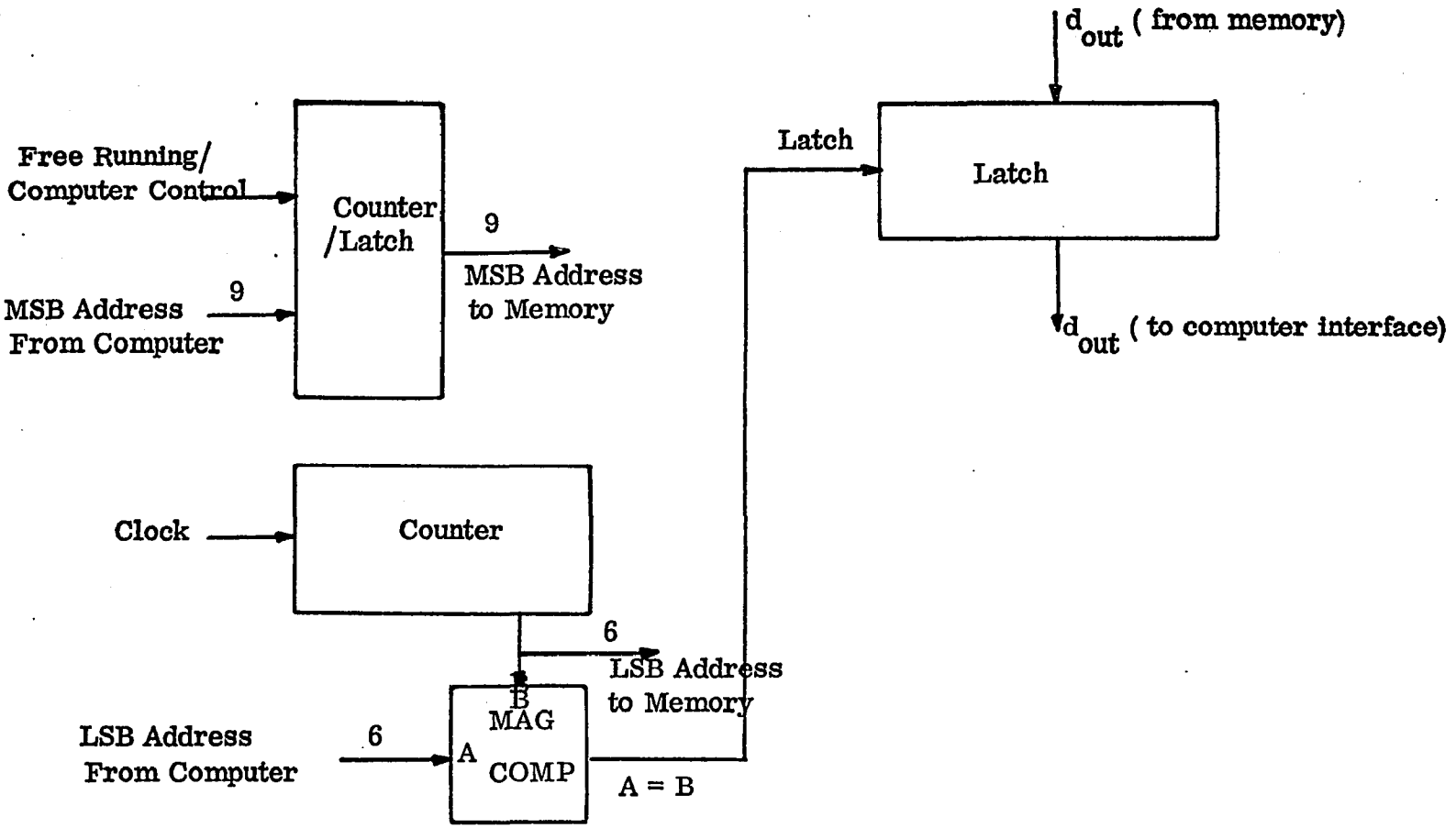


Fig. 28 Computer - Memory Interface

REFERENCES

1. Adaptive Delta Modulation Systems for Video Encoding
T.R.Lei, N.Scheinberg, D.L.Schilling, Transactions on Communications, pp 1302-1314, November 1977

2. Quantization of TV Chrominance Signals Considering the Visibility of Small Color Differences
L.Stenger,IEEE Transactions on Communications, pp 1393-1406, November 1977

3. Redundancy Reduction
Special Issue of the IEEE Proceedings, vol. 55, March 1977

4. Delta Modulation Quantizing Noise
J.B.O'Neal, BSTJ, vol. 45, pp 117-142 January 1966

5. A Variable Step Size Robust Delta Modulator
C.L.Song, J.Garodnick, D.L.Schilling, IEEE Transactions on Communications, pp 1033-1044, December 1971

6. Adaptive Delta Modulation
C.L.Song, PhD Dissertation, CCNY 1971

7. Delta Modulation
Final Report for NASA, Jan.-Dec.1 1972, Grant NGR 33-013-063

8. Delta Modulation
H.R.Schindler, IEEE Spectrum, July 1970

9. Techniques for Correcting Errors in Video ADM
N.Scheinberg, D.L.Schilling, IEEE Transactions on Communications, pp 1064-1070, September 1976

10. Delta Modulation of Video Signals
T.R.Lei, N.Scheinberg, D.L.Schilling,
pp 10.4.1-10.4.4, NTC December 1977

11. US Army Report DAAB07-C-0173-0003

12. Spread Spectrum Communications Notes
AGARD AD-766.914

13. Spread Spectrum Communications
R.C.Dixon, J.Wiley publisher

14. Low Rate Coding for Spread Spectrum
H.Helgert, R.Pickholtz, pp 54.1.1-54.1.7, NTC 1979

15. Optimization of the Processing Gain of an FSK-FH System
R.Pickholtz, D.L.Schilling, L.Milstein,
pp 54.3.1-54.3.6, NTC 1979

16. Partial Period Correlation Properties of PN Sequences
N.Bekir, R.Scholtz, L.Welch, NTC 1978

17. Adaptive Narrow Band Interference Supression
L.Milstein, R.Arsenault, P.K.Das,
pp 15.2.1-15.2.4, NTC 1978

18. The Effect of CW Interferences on Frequency Hopped Waveforms
S.Houston, pp 43.1.1-43.1.4, NTC 1978

19. Why Spread Spectrum
R.C.Dixon, IEEE Communication Magazine, pp July 1975

20. Special Issue on Spread Spectrum Communications
IEEE Transactions on Communications, August 1977

21. Video Encoding Using Adaptive Delta Modulation
D.L.Schilling, N.Scheinberg, J.Garodnick, IEEE Transactions on Communications, pp 1682-1689, November 1978

22. Design of a Packet Voice Transmission System
D.L.Schilling et al., pp 13.1.1-13.1.5, NTC 1979

23. Transform Encoding
Technical Memorandum CL-16-77, COMSAT Labs.