

~~77-79829~~
NASA CR-121293
N71-34190

POLYTECHNIC INSTITUTE OF BROOKLYN
DEPARTMENT OF ELECTRICAL ENGINEERING

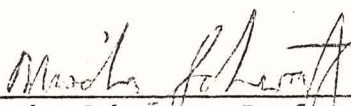
Final Status Report

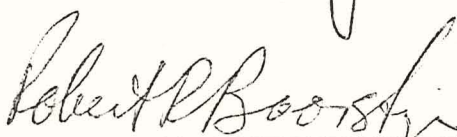
NASA Grant NGR 33-006-040, Supplement 2

A STUDY
of
DIGITAL TECHNIQUES
for
SIGNAL PROCESSING

August 1, 1970 to January 31, 1971

Principal Investigators:


Mischa Schwartz, Professor


Robert R. Boorstyn, Associate Professor

Introduction

We report here on activities in the following five areas carried out during the last six months of this grant period:

1. Adaptive maximum-likelihood receiver for digital data transmission.
2. Dynamic programming applied to adaptive equalizer design.
3. Automatic equalization in the frequency domain.
4. Recursive adaptive equalizers.
5. Finite memory communication systems.

The work reported on is in all cases, except that of area 3, an extension of previously reported work. The work of area 3, fast Fourier transform techniques applied to the equalization problem, was intensively carried out during the past six months because of its dramatic possibilities: the results obtained indicate that both computational complexity and convergence time are reduced considerably using the frequency approach, particularly for the large-parameter size equalizers required for those channels introducing large amounts of intersymbol interference.

The work on finite memory systems reported on here is, it is felt, also quite significant since it demonstrates the possibility of using small memory devices to attain requisite low probabilities of error in the detection of binary signals.

Both of these areas, as well as the others reported on, illustrate the basic philosophy underlying the activities carried out under this grant--the advent of high-speed computational techniques has made it imperative to incorporate digital processing as a fundamental element in communication system design. We have stressed throughout the need to develop algorithmic approaches to communication signal processing, geared

specifically to software or hard-wired computer implementation. On this basis signal processing techniques must now be compared not only on their probability of error or mean-squared error performance characteristics, but on computational speed and complexity as well.

The activities on adaptive array processing and rapidly-converging adaptive equalizers reported on in the last semi-annual report, as well as in prior reports under this grant, also exemplify this philosophy: the development of computer-communication processing systems.

It will be noted that a large portion of the work on this grant, in the past year particularly, has focused specifically on the adaptive equalizer problem. There are two main reasons for this. First is the technological significance of the equalizer: its use on telephone as well as radio transmission facilities allows higher-speed data transmission systems to be utilized. (It is particularly appropriate for transmission channels over which the data signals may undergo unknown amounts of distortion, or those whose characteristics may change randomly during transmission). Second, as borne out by the investigations under this grant, is the fact that the equalizer serves as a good vehicle for trying out new computational approaches to signal processing problems. Some of the search techniques and nonlinear programming concepts used in this equalizer work are currently being applied as well to the design of digital filters.

Papers, based on this grant, written or delivered during the reporting period

Mischa Schwartz

"Computer Processing in Communications", invited tutorial lecture,
International Telemetering Conference, Los Angeles, Oct. 1970.

"Telephony and Telecommunications", invited major article, Encyclo-
pedia Britannica, December 1970.

"Electrical Communications", article, McGraw-Hill Year-
book on Science and Technology, 1971, pp. 150-153.

"Rapidly-Converging First-Order Training Algorithm for an Adaptive
Equalizer", to be published IEEE Trans. on Information Theory,
July 1971 (with T. Schoenfeld).

"Rapidly-Converging Second-Order Algorithms for Adaptive Equaliza-
tion", to be published, IEEE Trans. on Information Theory, Sept.
1971 (with T. Schoenfeld).

"Application of a Stochastic Constrained Optimization Technique to
Detector Array Processing", submitted to J. of the Acoustical
Society (with L. Winkler).

Also invited seminar talks at the following universities:

McMaster University, Hamilton, Ont., Sept. 1970

UCLA, Oct. 1970

U. of Rochester, Nov. 1970

College of the City of N. Y., Feb. 1971

New York University, April 1971

Louisiana State University, May 1971

Professor Schwartz has also been invited to present a series of in-
tensive seminars June 28 - July 16, 1971, at the Technische Universitat,
Braunschweig, Germany, and the Technische Hochschule, Munich, Germany.

Robert R. Boorstyn

"A Technique for Adaptive Quantization", Proceedings of UMR-Mervin J. Kelly Communications Conference, Oct. 1970, Univ. of Missouri-Rolla (with Raymond W. Stroh).

"Recursive Equalizers", presented at Seoul International Conference on Electrical & Electronics Engineering, Seoul, Korea, Sept. 1970 (with M. Yang).

"Digital Filtering for Radar Signal Processing Applications", submitted to IEEE Transactions on Audio and Electroacoustics (with J. Echard).

"Bounds in Finite-Memory Detectors", CCNY Symposium, March 1971.

An Adaptive Maximum-likelihood Receiver for Digital Data Transmission over Channels with Intersymbol Interference

The past few years have seen a flurry of work on adaptive equalizers for minimizing intersymbol interference, as well as noise, introduced in the transmission of digital data over channels with unknown characteristics. In past reports we have discussed our own work on rapidly-converging adaptive equalizers. In the sections following we discuss work just concluded on a dynamic programming approach to the design of non-recursive type adaptive equalizers, and work on automatic equalization using fast Fourier transform techniques. In this section we report on another approach to minimizing the effects of intersymbol interference and noise--an adaptive maximum-likelihood receiver. Initial work in this area was reported in the previous semi annual report. In the paragraphs following we first summarize this past work, then discuss work carried out in this last reporting period on the maximum likelihood receiver.

Summary of Past Work

The maximum-likelihood (M.L.) receiver is known to be optimum in the sense of minimum probability of error. The receiver processes the received digital data in a sequential manner and makes decisions in a block of data of length L , where L is the number of time periods over which the intersymbol interference extends. The receiver uses a decision directed scheme to estimate the channel parameters and is therefore capable of being adaptive. Binary bipolar data sequences $\{a_k\}$ are assumed transmitted over the channel and white gaussian noise added just prior to reception. The M.L. receiver computes the ratio

$$\lambda_K = \ln \frac{f(\dots g_{K-1}, g_K | y_K = i)}{f(\dots g_{K-1}, g_K | y_K = j)} \quad i, j = 1, 2, \dots$$

Here $g_K = a_K x_0 + a_{K-1} x_1 + \dots + a_{K-L} x_L + n_K$ is the received signal, the

x 's represent the sampled signal response at the receiver,

$$y_K = (a_K, a_{K-1}, \dots, a_{K-L})$$

is an L -tuple block of the transmitted sequence, and the $f(\cdot)$'s represent gaussian conditional density functions. The receiver decides that a sequence $y_K = i$ was transmitted if $\lambda_K > 0$ for all $j \neq i$. A recursive vector equation may be obtained to compute λ_K . We define

$$\underline{P}_{g/y}(K) = \begin{bmatrix} f(\dots g_{K-1}, g_K | y_K = 1) \\ \dots \\ f(\dots g_{K-1}, g_K | y_K = 2^L) \end{bmatrix}$$

and $A(K) = [a_{ij}(K)]$, with $a_{ij} \equiv P(g_K | y_K = i, y_{K-1} = j) P(y_K = i | y_K = j)$

The first term in $a_{ij}(K)$ involves the received information, while the second term is the transitional probability for the y_K 's which form a Markov chain.

We then have the recursive relation

$$\underline{P}_{g/y}(K) = A(K) \underline{P}_{g/y}(K-1).$$

The elements of matrix $A(K)$ are known in terms of the noise distribution and the transitional probabilities in the Markov chain.

Work During Reporting Period

1) The recursive equation above was simulated on a digital computer for a known channel, i.e. fixed x_0, x_1, x_2, x_3 , for the case $L = 3$. It was found that the system does decode correctly for various values of $SNR = \frac{x_0^2}{\sigma^2}$, and various ratios of intersymbol interference terms to signal term, within the bounds of error probability discussed below.

2) Under the decision directed scheme that the decoded symbol $\hat{a}_K = a_K$, the transmitted symbol, the decoded symbols were used to provide a linear

estimate of the channel parameters $x_0, x_1, \dots$. One such estimate is given by

$$\hat{x}_K = \frac{1}{(N-K)S} \sum_{j=K+1}^N g_j \hat{a}_{j-K} \quad \text{where} \quad S = E[a_j^2] = 1 \quad \text{for } a_j = \pm 1.$$

(We have assumed here that $E[a_i a_K] = S \delta_{iK}$). This type of scheme

gives an unbiased estimate of x 's. Its variance is quite large even for large N , however, due to the term x_0^2 in the variance of $x_1, x_2, \dots$.

A smaller variance estimate may be obtained by estimating x_0 as above, subtracting this from g_K to get $g_K - a_K \hat{x}_0$ and then estimating x_1 by using the linear estimator

$$x_1 = \frac{1}{(N-1)} \sum_{K=2}^N (g_K - a_K \hat{x}_0) a_{K-1}$$

Similarly, we may then use these estimates $\hat{x}_0$ and $\hat{x}_1$ to provide an estimate of x_2 and so on.

The recursive equation for the M.L. receiver was then simulated on the digital computer for an unknown channel. An initial training period was set up and a continuous estimate of the channel parameters obtained as the system decoded the received data. The scheme was found qualitatively to work over a wide range of variations in the estimates of channel parameters.

An attempt was also made to study the effect of variations in the x_K 's on the elements A_{ij} of the matrix $A(K)$ with a view to calculating the effect of channel parameter estimation on the probability of error.

3) a. A lower bound on the error probability was obtained by calculating the probability of error P_e for the case of no intersymbol interference. In this case

$$\Omega_K = \frac{f(\dots g_{K-1}, g_K | a_K = 1)}{f(\dots g_{K-1}, g_K | a_K = -1)} = \frac{f(g_K | a_K = 1)}{f(g_K | a_K = -1)}$$

and

$$(P_e)_{\text{lower}} = \frac{1}{2} P(\Omega_K > 1 | a_K = -1) + \frac{1}{2} P(\Omega_K < 1 | a_K = 1)$$

b. For the upper bound the intersymbol interference was completely ignored, the decision rule in this case being binary: $g_K > 0 \rightarrow a_K = 1$
 $< 0 \rightarrow a_K = -1$

$$(P_e)_{\text{upper}} = \frac{1}{16} \sum_{i=1}^8 \left\{ P(\eta_K < -x_0 - d_i) + P(\eta_K > x_0 - d_i) \right\}$$

where $d = a_{K-1}x_1 + a_{K-2}x_2 + a_{K-3}x_3$ is a random variable which takes on the value d_i $i = 1, 2, \dots, 8$, depending on the values of a_{K-1} , a_{K-2} , and a_{K-3} .

At present work is being done to obtain tighter bounds. The approach is as follows:

$$\text{Consider } \Omega_K = \frac{f(\dots g_{K-1}, g_K | a_K = 1)}{f(\dots g_{K-1}, g_K | a_K = -1)}$$

For the case when the intersymbol interference exists only in the next adjacent time period the following recursive relation for the likelihood ratio may be obtained.

$$\Omega_K = \frac{a_1 \Omega_{K-1} + b_1}{a_2 \Omega_{K-1} + b_2} \quad \text{where } \begin{aligned} a_1 &= f(g_K | a_K = 1, a_{K-1} = 1) \\ b_1 &= f(g_K | a_K = 1, a_{K-1} = -1) \\ a_2 &= f(g_K | a_K = -1, a_{K-1} = 1) \\ b_2 &= f(g_K | a_K = -1, a_{K-1} = -1) \end{aligned}$$

The boundary between the two hypotheses is given by setting $\Omega_K = 1$.

This gives a relationship between Ω_{K-1} and g_K defining the region where

an error can occur. This relationship is given by

$$\ln \Omega_{K-1} = \lambda_{K-1} = -2p_K d + \ln \sinh(sd + p_K) - \ln \sinh(sd - p_K)$$

where

$$p_K = \frac{g_K x_0}{\sigma^2}$$

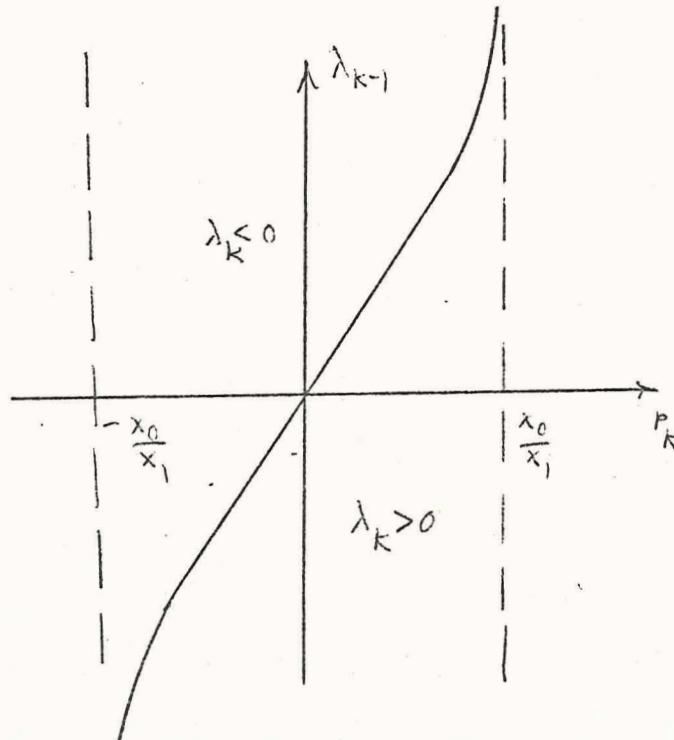
$$d = x_1/x_0$$

$$s = x_0^2/x_1^2$$

In graphical form the boundary is as shown below.

Here the a's and b's represent the various conditional density functions of the received signal g_K , under the four possible combinations of the two successive data symbols a_K and a_{K-1} .

Since $\Omega_K \geq 1$ represents the two possible decisions on symbol a_K , the decision regions may be represented by the accompanying figure plotting $\lambda_{K-1} \equiv \ln \Omega_{K-1}$, vs. $p_K \equiv \frac{x_0}{\sigma^2} g_K$. The probability of error is obtained by integrating over the $\lambda_K > 0$ region for $a_K = -1$, and the $\lambda_K < 0$ region for $a_K = 1$. This is in general a difficult task. By suitably approximating these regions we hope to obtain reasonably tight upper and lower bounds.



Dynamic Programming Applied to Adaptive Equalizer Design:

In the previous report (Semi-Annual Status Report, July 31, 1970), we discussed work done on the application of dynamic programming techniques to the adaptive equalization problem. Specifically, we discussed variable step size gradient search training algorithms for non-recursive digital filters with the step sizes chosen via a dynamic programming approach. Two possibilities were considered: 1. a straight dynamic programming technique in which the step sizes were found to be given by an explicit expression involving the measured statistics of input signal, noise, and output error; 2. a sub-optimum quantized step size equalizer in which one of a number of predetermined step sizes is chosen.

The work on these two approaches to equalization has now been concluded. Approximate bounds on the convergence rate and variance of the error as a function of iteration have been derived, for both types of schemes, and simulation results carried out to check these. The dynamic programming equalizer has been found to converge somewhat more quickly than the fixed step size scheme with no significant deterioration in the variance.

The convergence bounds of the quantized step size equalizers were calculated for 2, 3, and 5 step size equalizers. As expected, the 2 step system gave some improvement in convergence over the fixed step size scheme, while the 5 step system came fairly close in performance to the continuous step size dynamic programming scheme.

The theoretical analysis of these schemes is rather difficult because of the variable nature of the step sizes: in the continuous step size case they are random variables which depend upon the channel noise,

intersymbol interference, and the state of the equalizer. In the iterative equation governing convergence the (random) step size α_i on the i^{th} iteration appears multiplying the instantaneous and hence random sample covariance matrix Q_i of incoming signal samples and noise. It is necessary to evaluate $E[\alpha_i Q_i]$, an almost hopeless task. To carry out the analysis it was necessary to replace α_i by the estimated average step size for that iteration. The resultant convergence analysis was thus only approximate providing a loose bond on the convergence rate. Reliance had to be placed on the simulation results to verify the assumptions made.

As noted above the step size to be used in the dynamic programming equalizer turns out to depend on the measured output error. For the quantized step size equalizer a fixed number of step sizes, determined beforehand, are stored in the equalizer. The expected range of output error is broken up into a set of regions, each one corresponding to one of the step sizes. In operation, then, the measured error on any iteration falling in any region causes the step size corresponding to that region to be read out. The quantized step sizes to be used were determined by minimizing the average mean squared difference between the true step sizes and the quantized equivalents. The true step sizes were found, by computer simulation, to be very nearly Rayleigh distributed and this information was used in the derivation of the quantized values. The output error thresholds, denoting the boundaries between the various error regions, were found by noting that the step size was proportional to the output error squared. The step size regions corresponding to the required

quantized step sizes could thus be converted directly to output error thresholds.

The practical usefulness of the dynamic programming equalizers remains of course to be seen: there is definite improvement in the speed of convergence but at the cost of increased system complexity. Specifically, for a typical poorly-conditioned telephone line, (true condition number of 17.8^{*}, with estimated value of 164), and SNR of 30db, the dynamic programming equalizer reduced the mean-squared output error, on the average, to 20% of its initial value in 2-3 iterations, while the fixed step size scheme required 5 iterations to get to the same point. (The irreducible mean squared error in this case was 4% of the initial value). The 5 step scheme performed almost as well as the dynamic programming equalizer. The dynamic programming equalizer took 4-5 iterations to reduce the error to 10% of its initial value, while the fixed step size scheme was down to 12% after 11 iterations.

A doctoral thesis covering this project in detail has been completed and the preparation of a paper for publication is underway.

*Condition number is defined as the ratio of maximum to minimum eigenvalues of the sample covariance matrix of the signal samples at the input to the equalizer. See previous reports for further discussion.

Automatic Equalization in the Frequency Domain

Prior work on adaptive equalizers reported on under this grant has concentrated on equalization in the time domain. (See the sections in this report as well on Dynamic Programming applied to Adaptive Equalizer Design and Recursive Equalizers). We summarize in this section work carried out during the last reporting period on the study and analysis of techniques which use discrete Fourier transforms (DFT) in conjunction with fast Fourier transform (FFT) algorithms, for automatic equalization of synchronous data transmission. Details of the work will appear in a doctoral dissertation currently being completed and several papers are planned covering aspects of the work.

We have shown in this work the problem of minimizing mean-square intersymbol interference can be analyzed in the discrete frequency domain as an optimization problem with constraints. Various solutions to this problem were studied including Rosen's gradient projection method, Lagrange multipliers, and direct substitution. We have proven that the rate of convergence toward the optimum parameter setting is faster for the gradient projection scheme, for channels usually discussed in the literature, than for the corresponding time domain technique. We then developed an alternate gradient projection method which provides savings proportional to $N/\log_2 N$ in the number of required computations, where N is the number of discrete frequency parameters. Finally, we have devised a scheme for finding an approximate solution in one iteration using gradient projection. (This solution was found to be almost exact for the particular cases we simulated, even in the presence of noise!).

The speed of convergence for all the schemes is dependent on the overall channel characteristics and each method has advantages in certain special

situations. We have shown that all the aforementioned methods converge in the mean in the presence of noise. A variance bound indicates that the variance about this setting is finite and can be made as small as desired by reducing the gradient step size and hence the speed of convergence.

Another objective in addition to frequency domain equalization was to use the DFT and FFT algorithms to make time domain equalization computationally more efficient. We have shown that the number of computations needed to set a time domain equalizer can be made proportional to $M \log_2 M$ instead of M^2 , where M is the number of adjustable parameters, by use of FFT algorithms. Savings always ensue for sufficiently large M and grow rapidly thereafter. The breakeven point is approximately $M = 16$. We have also derived tight and easily obtainable bounds on the eigenvalues of the time domain iteration matrix in terms of DFT coefficients. This allows us to increase the speed of convergence.

Finally we have shown that for both time domain and frequency domain equalization we can process the data more efficiently in the "transmission mode" (i.e., after the equalizer is set and we transmit information) using FFT's. For large values of M the savings are proportional to $\log_2 M/M$. It can therefore be assumed that the automatic equalizer would include an FFT processor and therefore no extra hardware and/or software will be necessary to perform FFT's in the "training mode".

Recursive adaptive equalizers

Most adaptive equalizers utilize non-recursive (transversal) digital filters as processors. That is, the output at time $t = T$, $y_n = y(nT)$, is related to the input $x_n = x(nT)$ by

$$y_n = \sum_{k=1}^L c_k x_{n-k} \quad (1)$$

where the filter coefficients $\{c_k\}$ can be found using various iterative search techniques based on standard performance criteria such as the mean-squared error. This transversal filter is an effective device for combatting intersymbol interference in a communication channel. However, if the channel exhibits an extremely long time dispersion, the transversal filter equalizer will in general, require a large number of taps (coefficients). Equivalently, it has also been shown that devices whose frequency responses have sharp transitions between passband and stopband can be more economically represented by recursive filters than by non-recursive filters. Hence it is appropriate to try to use a recursive filter for equalization of a highly dispersive channel.

The input-output relationship of a recursive filter is given by

$$y'_n = \sum_{k=0}^N a_k x_{n-k} - \sum_{k=1}^M b_k y'_{n-k} \quad (2)$$

where the prime denotes the output of the recursive filter. In general, fewer coefficients (i. e., a simple structure) are needed for such a filter as contrasted with the non-recursive type. The possible use of such structures as adaptive equalizers is tempting because of the possibility of improved computational simplicity. The analysis is complicated by the complex non-linear (non-quadratic actually) relationship between the error and the two sets of coefficients a_k and b_k . We first outline a time-domain technique for designing

(non-optimum) recursive filters and then extend this method to the design of recursive equalizers.

Assume that the desired non-recursive structure has been found. That is, the $\{c_k\}$ are determined so as to perform, say optimally, a certain task. This is usually easy. We now try to find a recursive filter to approximate the performance of the above non-recursive filter. We want to find coefficients $\{a_k\}$ and $\{b_k\}$ so that y_n and y'_n are as close as possible. (If $x_n = \delta_n$, $\delta_n = 0$, $n \neq 0$, $\delta_0 = 1$, then $y_n = h_n$, and $y'_n = h'_n$ where h_n and h'_n are the impulse responses of the two filters). As indicated above solution of this problem as stated leads to complex non-linear relationships between the error and coefficients. We circumvent this by introducing a new psuedo-error which is a measure of the degree to which equation (2) is not satisfied when y'_n is replaced by y_n . (Note that y_n is known since we assume the desired $\{c_k\}$ have already been found). This "psuedo-error" is given by

$$\hat{e}_n = \sum_{k=0}^N a_k x_{n-k} - (y_n + \sum_{k=1}^M b_k y_{n-k}). \quad (3)$$

Using the minimum squared error $\hat{\epsilon} = \sum_{n=0}^{\infty} \hat{e}_n^2$ as a performance

criterion it is easily found that $\hat{\epsilon}$ is a convex (quadratic) function of the coefficients $\{a_k\}$ and $\{b_k\}$. Thus there is a unique minimum and the appropriate coefficients can be easily found by any of many algorithms. The coefficients that minimize the psuedo-error are given by

$$\underline{c} = Q^{-1} \underline{g} \quad (4)$$

where $\underline{c}$ and $\underline{g}$ are $N + M + 1$ - vectors given by

$$\underline{c}^T = (a_0, a_1, \dots, a_N, -b_1, -b_2, \dots, -b_M)$$

$$\underline{g}^T = (g_0, g_1, \dots, g_N, g_{N+1}, g_{N+2}, \dots, g_{N+M})$$

where T indicates transpose and

$$g_i = \sum_{n=0}^{\infty} x_{n-i} y_n, \quad i = 0, \dots, N \text{ and}$$

$$g_{N+i} = \sum_{n=0}^{\infty} y_{n-i} y_n, \quad i = 1, \dots, M.$$

Q is an $(N + M + 1) \times (N + M + 1)$ with elements

$$Q_{ij} = \begin{cases} \sum_{n=0}^{\infty} x_{n-i} x_{n-j}, & i, j = 0, 1, \dots, N \\ \sum_{n=0}^{\infty} x_{n-i} y_{n-(j-N)}, & i = 0, 1, \dots, N, \quad j = N+1, \dots, N+M \\ \sum_{n=0}^{\infty} y_{n-(i-N)} x_{n-j}, & i = N+1, \dots, N+M, \quad j = 0, 1, \dots, N \\ \sum_{n=0}^{\infty} y_{n-(i-N)} y_{n-(j-N)}, & i, j = N+1, \dots, N+M. \end{cases}$$

Thus if the inverse of Q exists we have found a set of coefficients for the recursive filter that minimizes the psuedo-error. However Q is in general not non-singular. Furthermore, even if Q is non-singular, thus providing a set of coefficients, it is possible that the resultant filter will be unstable, that is some zeros of $B(Z^{-1}) = 1 + b_1 Z^{-1} + \dots + b_M Z^{-M}$ in the Z domain will lie outside the unit circle. Finally minimizing the psuedo-error does not guarantee that the actual error will also be small. (This is, as we shall see, tied in with the problem of stability). We have been able to show that if we restrict ourselves to auto-regressive recursive filters then Q is non-singular and the resultant filter is stable.

For an autoregressive filter

$$y'_n = a_0 x_n - \sum_{k=1}^M b_k y'_{n-k}. \quad (5)$$

That is $a_k = 0$, $k = 1, \dots, N$, or $N = 0$. The matrix Q becomes an $M+1 \times M+1$ matrix with elements

$$Q_{ij} = \sum_{n=0}^{\infty} y_{n-i} y_{n-j}, \quad i, j = 0, 1, \dots, M$$

which is a Gramian matrix. When equations (3) and (5) are rearranged slightly and $x_n = \delta_n$, so that $y_n = h_n$ and $y'_n = h'_n$ are impulse responses it is possible to show that Q is non-singular and the resultant auto-regressive recursive filter is stable. Two examples illustrate the results.

Example 1. Let $c_k = (-0.5)^k$, $k = 0, 1, \dots, L$. That is,

$$H(Z^{-1}) \triangleq \sum_{n=0}^{\infty} h_n Z^{-n} = \frac{Y(Z^{-1})}{X(Z^{-1})} = C(Z^{-1}) = 1 + 0.5Z^{-1} - (0.5)^2 Z^{-2} + \dots + (-0.5)^L Z^{-L} = Y(Z^{-1})$$

if $x_n = \delta_n$. Using the above procedure we obtain $a_0 = 1$, $b_1 = 0.5$, and

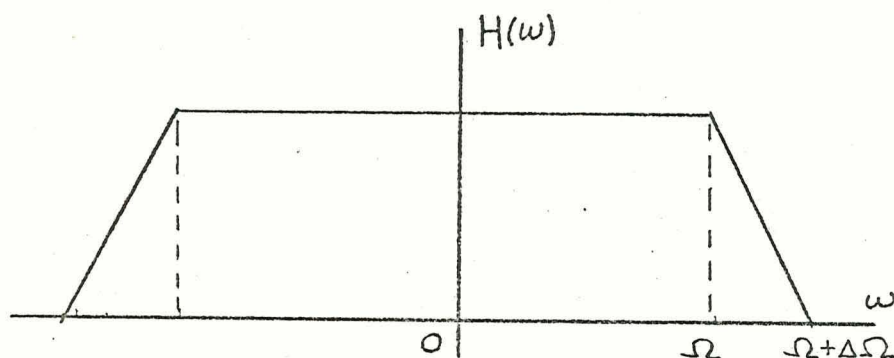
$$b_2 = b_3 = \dots = b_M = 0. \quad \text{Thus } H'(Z^{-1}) = \frac{Y'(Z^{-1})}{X(Z^{-1})} = Y'(Z^{-1}) = \frac{1}{1 + 0.5Z^{-1}}.$$

Therefore $h_k = h'_k$, $k = 0, 1, \dots, L$, $h'_k = (-0.5)^k$, $k > L$. The error is negligible.

Example 2. Let

$$c_k = \frac{\sin\left(\frac{0.01}{2} \frac{\pi}{7} k\right)}{\frac{0.01}{2} \frac{\pi}{7} k} \cdot \frac{\sin\left(\frac{2.01}{2} \frac{\pi}{7} k\right)}{\frac{2.01}{2} \frac{\pi}{7} k}, \quad k = 0, 1, \dots, 200.$$

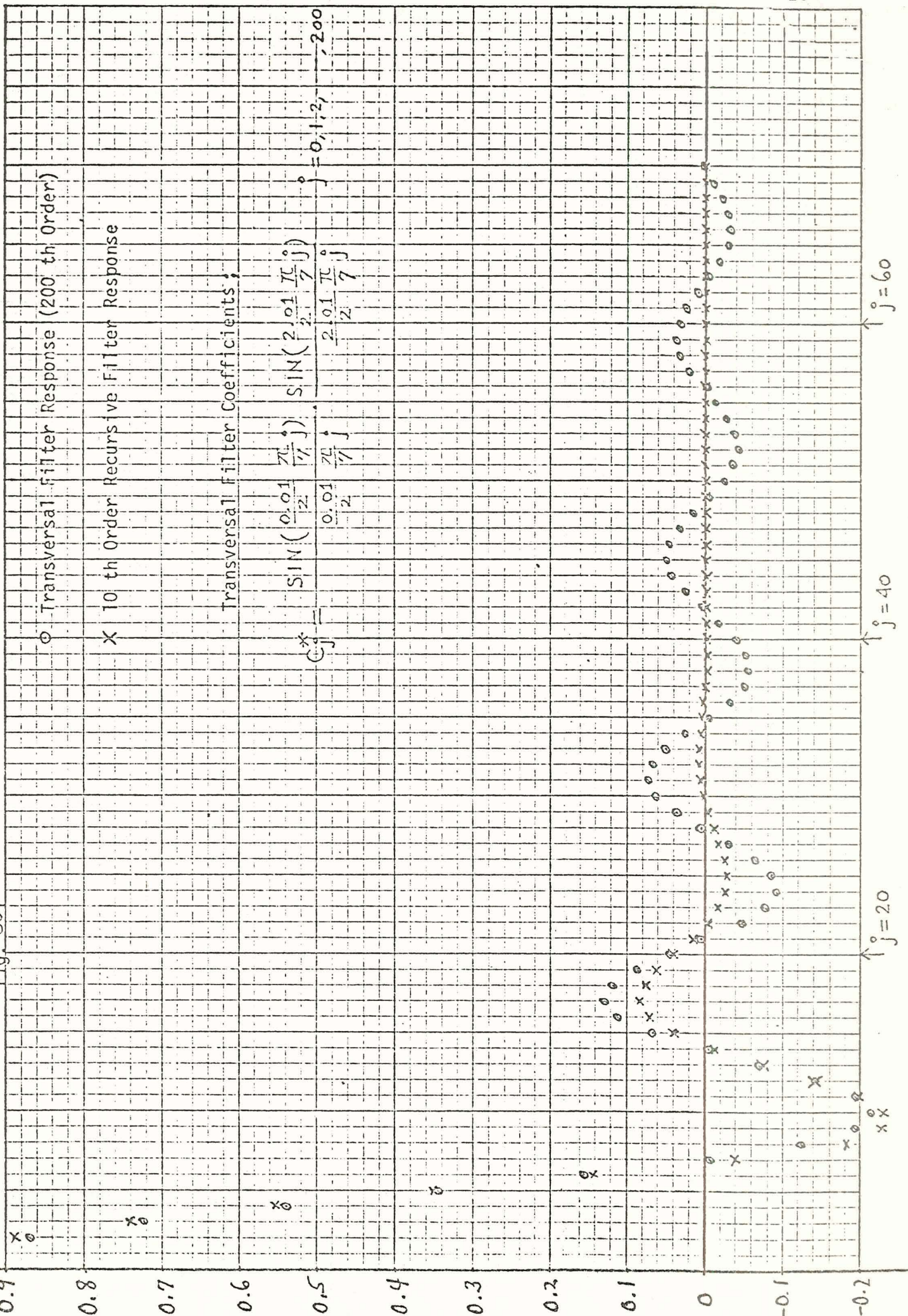
This could be thought of as truncated impulse response of a low-pass filter which has the following amplitude characteristic.



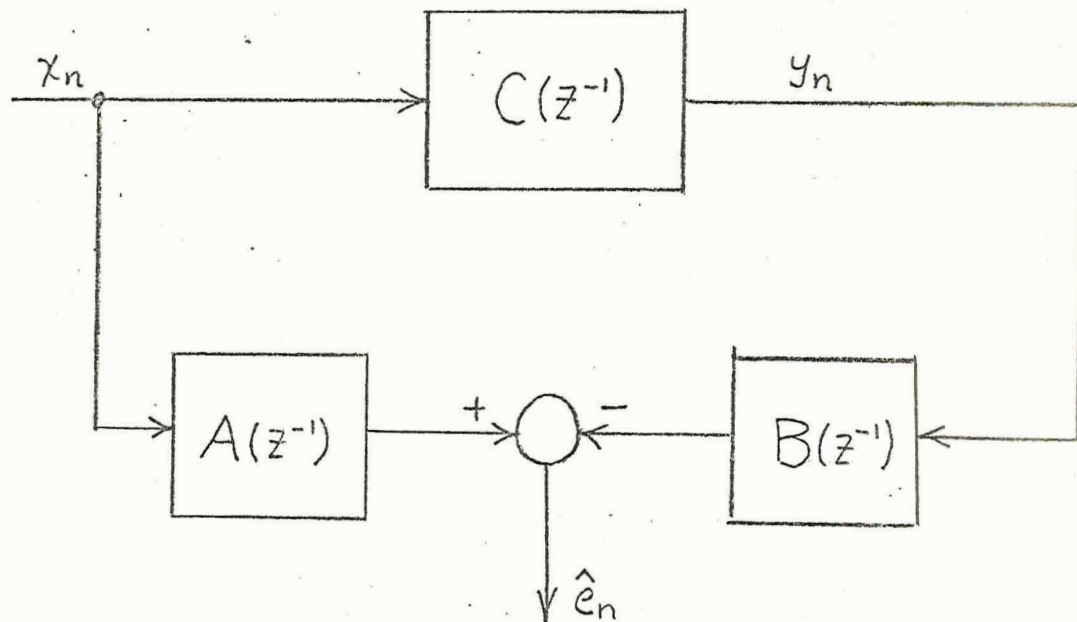
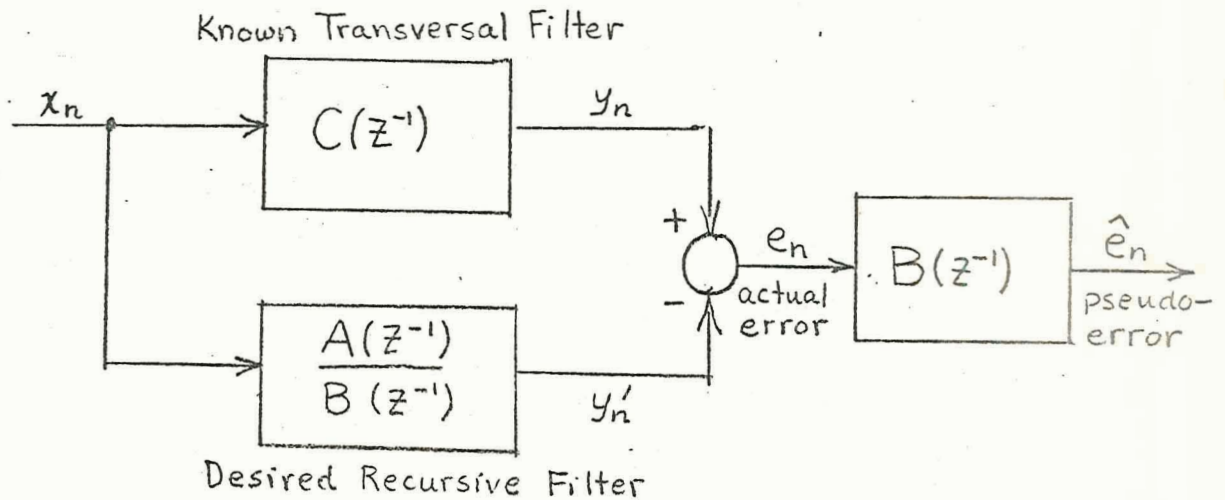
Since $\Delta\Omega/\Omega = 0.01$ the transition between passband and stopband is very sharp. A comparison between the transversal filter with the above coefficients and an auto-regressive recursive filter with $M = 9$ found using the above technique is shown in the figure below

KE 10 X 10 TO 1 1/4 INCH 4G 0620
7 X 10 INCHES
MADE IN U.S.A.
KEUFFEL & ESSER CO.

Fig. 001

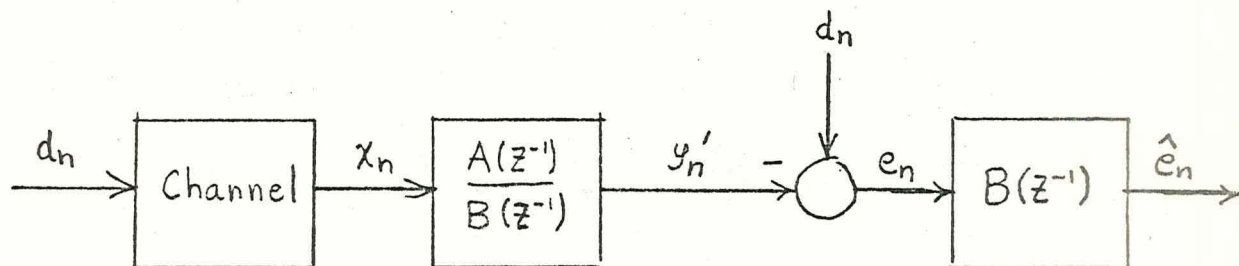


The relationship between the psuedo-error $\hat{e}_n$ and the actual error $e_n = y_n - y'_n$ is easily found by considering the two equivalent block diagrams shown below.



In the above $A(Z^{-1}) = a_0 + a_1 Z^{-1} + \dots + a_N Z^{-N}$ and $B(Z^{-1}) = 1 + b_1 Z^{-1} + \dots + b_M Z^{-M}$. Thus $\hat{E}(Z^{-1}) = B(Z^{-1})E(Z^{-1})$.

The extension of this method to the design of equalizers is straightforward. Consider the block diagram below



$\{d_n\}$ is the sequence being transmitted in the tracking or training mode. In the training mode $\{d_n\}$ is assumed known at the receiver. In the tracking mode it is usually derived from y'_n by simple quantization. This approximation is good if the equalizer is working well. The purpose of the recursive equalizer $\frac{A(z^{-1})}{B(z^{-1})}$ is to produce y'_n which is close to d_n . It is only necessary to replace Y_n by d_n in the above development. All comments still apply.

This investigation is continuing with respect to the singularity of Q , instability of resultant recursive filters, relationship between pseudo-error and actual error, choice of $\{d_n\}$ in training mode to get good performance, performance in tracking mode, design of better algorithms, comparison of recursive and non-recursive filters, etc.

Bounds on Finite Memory Detectors

This report contains the major results obtained in an investigation of time invariant finite memory detectors for the simple binary hypothesis testing problem where the number of observations is constrained to be finite.

Let $\{x_n\} = (x_1, x_2, \dots, x_n)$ be a sequence of n independent, identically distributed random observations. Let $f_0(x)$ and $f_1(x)$ be the probability density functions for each x_i , $i = 1, \dots, n$ under hypothesis H_0 and H_1 respectively. The problem is that at time n one must decide whether $\{x_n\}$ is distributed according to hypothesis H_0 or H_1 .

Let $d_n \in \{H_0, H_1\}$ be the decision made at time n . It is well known that if d_n is allowed to depend on $(x_1, x_2, \dots, x_n)$ then the standard likelihood ratio test yields a probability of error tending exponentially to zero as the number of observations, n , tends to infinity. However, to store and process the observations requires infinite memory.

In order to restrict the memory to a finite size consider time invariant detectors that operate according to the following algorithm

$$\begin{aligned} T_j &= f(T_{j-1}, x_j) \\ d_j &= d(T_j) \end{aligned} \tag{1}$$

where x_j is the j^{th} observation, T_j is the state of the memory at time j , d_j is the decision at time j , and f is a function independent of j and the data.

The algorithm is said to have finite memory of size m if T is m valued (i. e., $T_j \in \{1, 2, \dots, m\}$ for $j = 1, 2, \dots, n$). Note the state of the memory at time j depends only on the state of the memory at time $j - 1$ and the current observation, x_j .

Hellman and Cover⁽¹⁾ have investigated detectors whose operation

(1) Hellman, M. E., and Cover, T. M., Learning with Finite Memory, PhD Thesis, Stanford University.

is described by equation (1) for the case where the observation sequence $\{x_n\}$ is infinite in extent. They were able to find a lower bound P^* on the asymptotic probability of error, $P_e = \lim_{n \rightarrow \infty} P_r\{d_n = H_j | H_i\}, i \neq j$. In general no detector was able to achieve P^* , but it is possible to design a detector whose probability of error, P_e , approaches P^* , as close as desired, i.e., $P_e < P^* + \epsilon$ for $\epsilon > 0$. The class of detectors for which $P_e < P^* + \epsilon$ is called ϵ -optimal. This class of detectors in general requires artificial randomization in the function f and transitions between states of memory occur only when an observation maximizes or minimizes the likelihood ratio, $f_0(x)/f_1(x)$. The decision function d_n assigns each state of the memory to a particular hypothesis.

As a particular example of an ϵ -optimal machine consider the following hypothesis testing problem where the sequence of observation is infinite in extent. Assume under hypothesis H_0 the observations x_i are normally distributed with mean $-\mu$ and variance σ^2 , and under hypothesis H_1 the observations x_i are normally distributed with mean $+\mu$ and variance σ^2 , and that $P_r\{H_0\} = P_r\{H_1\}$. The ϵ -optimal detector consists of two states of memory whose transitions between states are as follows. If the detector is in state 1 (2) and the current observation $x_j \geq D$ ($\leq -D$) move to state 2 (1). Remain in the present state otherwise. The decision function assigns state 1 to hypothesis H_0 and state 2 to hypothesis H_1 . The machine starts in state 1 or 2 with a probability of $1/2$ since both hypotheses are equally probable. The detector thus described is a threshold detector with a null zone between $-D$ and D . The lower bound P^* on the asymptotic probability of error is zero. This bound can be approached as closely as desired by allowing the magnitude of the threshold settings D to tend towards infinity.

However, if the sequence of observations is constrained to be finite, the probability of error at time n , $P_e(n) = P_r\{d_n = H_j | H_i\}, i \neq j$, of the above

detector approaches $1/2$. This follows from the fact if n is finite, the probability of at least one observation exceeding a threshold, $\pm D$, approaches zero, if $D \rightarrow \infty$. Therefore, the probability that the machine changes state approaches zero, and the probability of error, $P_e(n)$, is just equal to the probability of starting in the wrong state initially. Thus the detector which is ϵ -optimal for infinite observation sequences performs poorly for finite observation sequences.

At the opposite extreme, if the thresholds $\pm D$ are set at zero, then the final state of the above two state detector will depend only on the last observation. Thus nothing is gained by making multiple measurements.

This particular problem stimulated the research into the following two classes of time-invariant finite memory detectors where the observation sequence is constrained to be finite. For both classes of detectors to be discussed it will be assumed that $P_r\{H_0\} = P_r\{H_1\}$ and $f_0(x) = f_1(-x)$ where $f_0(x)$ and $f_1(x)$ are the probability density functions of the independent observations under hypotheses H_0 and H_1 respectively.

Class I. Finite Memory Linear Detectors.

A finite memory linear detector can best be described by the following figure.

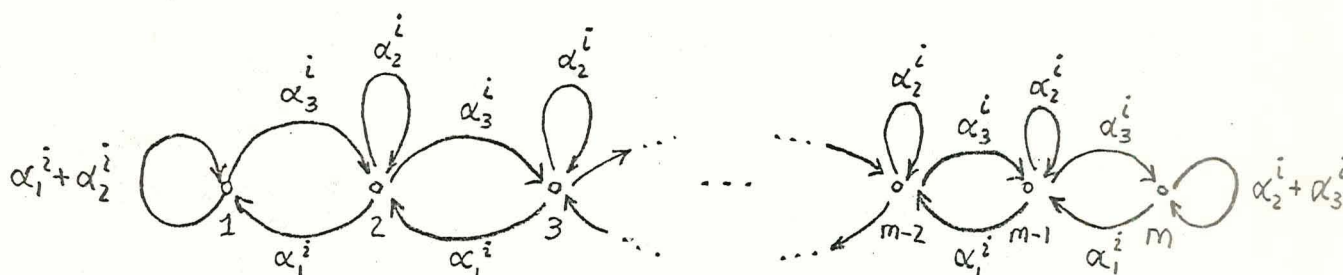


Fig. 1. Finite Memory Linear Detector

Where

$$\alpha_1^i = P_r \{x_j \geq D | H_i\}$$

$$\alpha_3^i = P_r \{x_j \leq -D | H_i\}$$

$$\alpha_2^i = P_r \{(-D < x_j < D | H_i\} = 1 - (\alpha_1^i + \alpha_3^i)$$

where $i = 0, 1$ and $D \geq 0$.

From Fig. 1 it is seen that memory linear detector has the following properties:

1. The number of states are finite, $m < \infty$.
2. A transition to a lower state from the next higher state occurs at time j , with probability α_1^i , where $\alpha_1^i = P_r \{x_j \leq -D | H_i\}$, unless the machine is in the lowest state.
3. A transition to a state from the next lower state occurs at time j , with probability $\alpha_3^i = P_r \{x_j \geq D | H_i\}$ unless the machine is in the highest

state.

4. No other transitions between states are allowed.

Due to the assumed symmetry of the probability density functions $f_0(x)$ and $f_1(x)$ under each hypothesis,

$$\alpha_1^0 = \alpha_3^1$$

$$\text{and } \alpha_3^0 = \alpha_1^1$$

Assume without loss of generality that $\alpha_1^0 > \alpha_1^1$, which implies $\alpha_3^1 > \alpha_3^0$.

Therefore under hypothesis H_0 the detector tends to move to a lower state and under hypothesis H_1 the opposite is true. If m , the number of states is even, the decision function d_n assigns at time n states 1 thru $m/2$ to hypothesis H_0 and states $\frac{m}{2} + 1$ thru m to hypothesis H_1 . The detector starts initially in states $\frac{m}{2}$ or $\frac{m}{2} + 1$ with a probability of $1/2$. If m , the number of states of memory is odd, then the initial starting state is $\frac{m}{2} + 1$, the center state, and the decision is arbitrary if this state is the final state. All other states are assigned to the same hypothesis as in the case of m even.

The operation of the finite memory linear detector thus described is given by equation (1). The independence of the observations, x_j , $j = 1, \dots, n$, set up a Markov process on the state space $S = \{1, \dots, m\}$. Thus the finite memory linear detector can be analyzed using the techniques developed for Markov processes.

For the case of a two state memory, the resultant probability of error is found to be

$$P_{Le}(2, n) = \frac{1}{2} \left\{ 1 - (\alpha_3^1 + \alpha_1^1) \left(\frac{1 - (\alpha_2^1)^n}{1 - \alpha_2^1} \right) \right\} \quad (2)$$

where the superscript 1 signifies all probabilities are to be calculated assuming hypothesis H_1 is true. If under hypothesis H_1 the observations are assumed normally distributed with mean μ and variance σ^2 , then

$$\alpha_1^1 = Q\left(\frac{D + \mu}{\sigma}\right)$$

$$\alpha_3^1 = Q\left(\frac{D - \mu}{\sigma}\right)$$

where $Q(x) = \frac{1}{2\pi} \int_x^\infty e^{-t^2/2} dt$. Equation (2) was programmed on a computer and a search made for the optimum threshold settings $\pm D$ for the cases of $\gamma = \mu^2 / \sigma^2 = 1$ and 4. A plot of probability of error vs. threshold settings is shown in Fig. 1 s 2a and 2b.

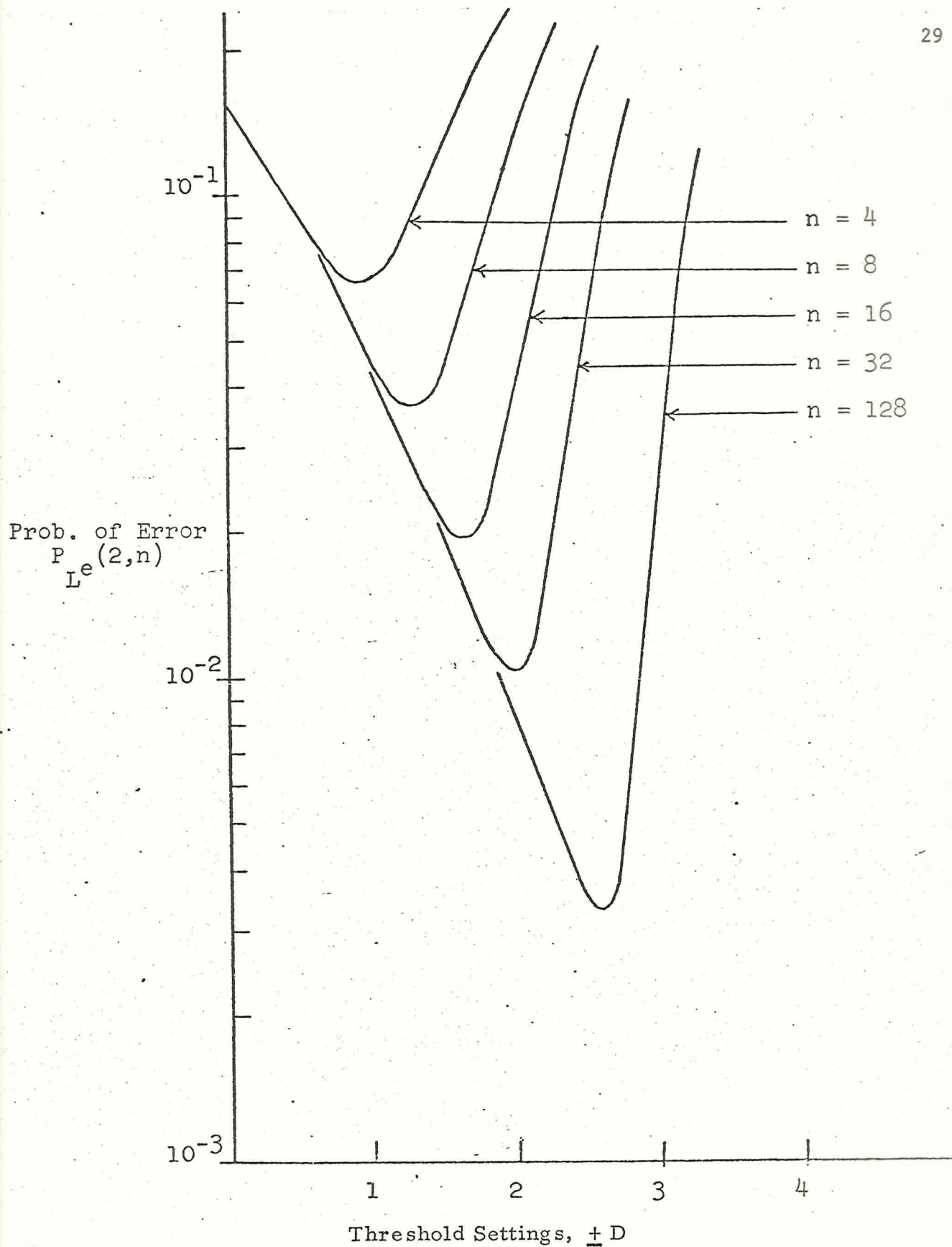


Figure 2a. Probability of Error vs. Threshold for a Two State Linear Detectors for Gaussian Observations, with $\gamma = \mu^2/\sigma^2 = 1$.

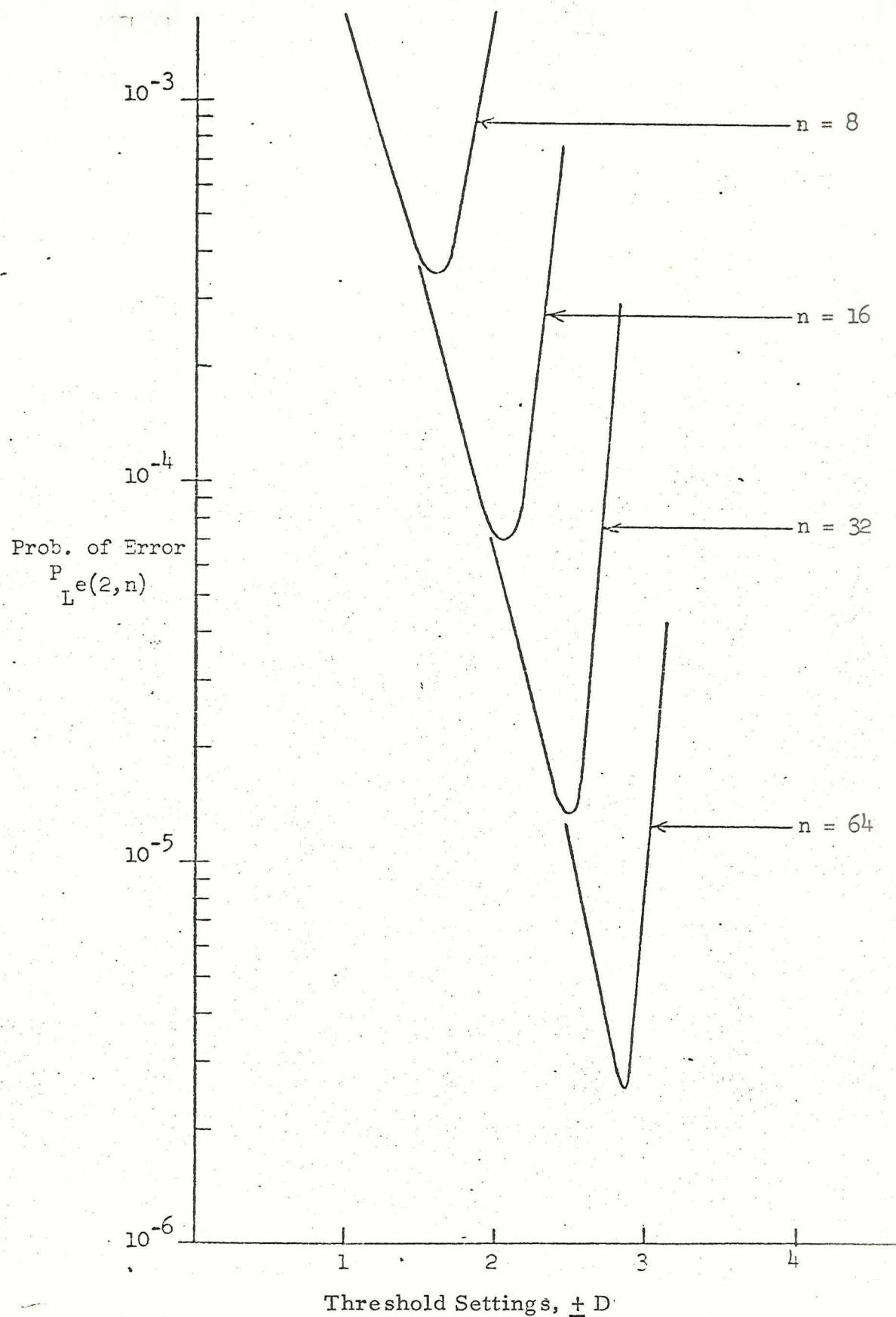


Figure 2b. Probability of Error vs. Threshold for a Two State Linear Detector for Gaussian Observations with $\gamma = \mu^2 / \sigma^2 = 4$.

Fig. 2b shows that for the case of $\gamma = 4$, probabilities of error on the order of 10^{-5} to 10^{-6} are obtainable with only two states of memory if there are between 32 and 64 observations.

The three and four state linear detectors were analyzed also for the case of gaussian distributed data. Using a four state linear detector a probability of error of 10^{-5} is achievable at optimum threshold setting for 16 observations and $\gamma = 2.25$.

Although their simplicity and reasonable error performance makes the finite memory linear detector practical, unfortunately the extreme difficulty in analyzing higher order machines appears to prevent one from evaluating their performance. The next class of finite memory detectors is structurely more complicated than the finite memory linear detectors, but will yield performance bounds for the linear detector and most importantly they can be analyzed for any memory size.

Class II. Finite Memory Majority-rule Detectors.

In order to understand the description of a finite memory majority rule detector the following definitions are necessary.

Definition: An observation x_j will be called a zero if $x_j \leq -D$ where $D \geq 0$.

Definition: An observation x_j will be called a one if $x_j \geq D$ where $D \geq 0$.

Collectively all observations x_j such that x_j is zero or one will be called non-null observations. All other observations are referred to as null observations.

Using the above definitions, under hypothesis H_i , $i = 0, 1$, the probability of a zero observation is

$$\alpha_1^i = P_r\{0|H_i\} = P_r\{x_j \leq -D|H_i\}$$

and the probability of a one observation is

$$\alpha_3^i = P_r\{1|H_i\} = P_r\{x_j \geq D|H_i\}$$

Also the probability of a null observation is

$$\alpha_2^i = P_r\{\text{null}|H_i\} = 1 - (\alpha_1^i + \alpha_3^i)$$

As in the case of the linear detector assume

$$\alpha_1^0 > \alpha_1^1$$

which implies $\alpha_3^1 > \alpha_3^0$

since $\alpha_1^0 = \alpha_3^1$ and $\alpha_3^0 = \alpha_1^1$.

Since $\alpha_1^0 > \alpha_1^1$ and $\alpha_3^1 > \alpha_3^0$ a zero observation has a higher probability of occurrence under hypothesis H_0 and a one observation has a higher probability of occurrence under hypothesis H_1 .

Definition: A 2^{2k-1} State Majority Rule Detector, $k \geq 1$, is a detector whose decision function d_n assumes hypothesis H_0 is true if at least k of the last $2k - 1$ non-null observation are zeros, and assumes hypothesis H_1 is true

2. Transition occurs from a state (a) to a state whose first $2k - 2$ bits are the same as the last $2k - 2$ bits of state (a).
3. The initial starting states are the two states that contain a $2k - 1$ pattern of alternating zeros and ones, and the probability of starting in either state is $1/2$.
4. For the above two examples, if a vertical line is drawn thru the center of the detector, the decision function d_n assigns states to the left of this line to hypothesis H_0 and states to the right to hypothesis H_1 .
5. Finally, notice for $k = 1$, the $2^{2k-1} = 2$ state majority rule detector is identical to the 2 state linear detector.

The probability that a 2^{2k-1} state majority rule detector will decide in favor of the wrong hypothesis after an n bit sequence of observations can be calculated using combinatorial techniques. The resultant expression for the probability of error is,

$$P_{Me}(2^{2k-1}, n) = \left\{ \frac{Z! \alpha_1^{\frac{Z}{2}+1} \alpha_3^{\frac{Z}{2}}}{(\frac{Z}{2}!)^2 (1-\alpha_2)^{Z+1}} + \frac{1}{(1-\alpha_2)^Z} \sum_{j=\frac{Z+2}{2}}^Z \binom{Z}{j} \alpha_1^j \alpha_3^{Z-j} \right\} \left\{ 1 - \sum_{r=0}^Z \binom{n}{r} \alpha_2^{n-r} (1-\alpha_2)^r \right\} \\ + \sum_{i=0}^Z \left\{ \binom{n}{i} \alpha_2^{n-i} \sum_{j=x}^i \binom{i}{j} \gamma(j) \alpha_1^j \alpha_3^{i-j} \right\} \quad (3)$$

where $x = \begin{cases} \frac{i}{2} & \text{for } i \text{ even,} \\ \frac{i+1}{2} & \text{for } i \text{ odd} \end{cases}$ $\gamma(j) = \begin{cases} \frac{1}{2} & \text{for } j = \frac{i}{2} \\ 1 & \text{for } j \neq \frac{i}{2} \end{cases}$, $Z = 2k - 2$

and α_1 , α_2 , and α_3 are to be calculated assuming hypothesis H_1 is true.

The importance of equation (3) follows from the following two theorems:

Theorem I. Let $\{x_n\} = (x_1, x_2, \dots, x_n)$ be a finite sequence of independent, identically distributed random observations. Consider the simple hypothesis testing problem. Under hypothesis H_i , $i = 0, 1$ let

$$\alpha_1^i = P_r \{x_j \leq -D | H_i\}$$

$$\text{and } \alpha_3^i = P_r \{x_j \geq D | H_i\} \quad , \quad j = 1, \dots, n \text{ and } D \geq 0.$$

If $P_r\{H_0\} = P_r\{H_1\}$, $\alpha_1^0 = \alpha_3^1$, $\alpha_1^1 = \alpha_3^0$ and $\alpha_1^1 < \alpha_3^1$ for all D , then the identical threshold setting, $\pm D$,

$$P_{Le}(2k, n) \leq P_{Me}(2^{2k-1}, n) \quad , \quad k \geq 1$$

where $P_{Le}(2k, n)$ is the probability of error of a $2k$ state linear detector and $P_{Me}(2^{2k-1}, n)$ is the probability of error of a 2^{2k-1} state majority rule detector.

Theorem II. Let $\{x_n\} = (x_1, x_2, \dots, x_n)$ be a finite sequence of independent, identically distributed random observations. Consider the simple hypothesis testing problem. Under hypothesis $H_i, i = 0, 1$, let

$$\alpha_1^i = P_r \{x_j \leq -D | H_i\}$$

$$\text{and } \alpha_3^i = P_r \{x_j \geq D | H_i\} \quad , \quad j = 1, \dots, n \text{ and } D \geq 0.$$

If $P_r\{H_0\} = P_r\{H_1\}$, $\alpha_1^0 = \alpha_3^1$, $\alpha_1^1 = \alpha_3^0$, and $\alpha_1^1 < \alpha_3^1$ for all D then for identical threshold settings, $\pm D$, a sufficient condition for

$$P_{Me}(2^{2(k+1)-1}, n) \leq P_{Le}(2^k, n) \quad , \quad k \geq 1 \text{ is that}$$

$$\alpha_1^0 \geq M(k) \alpha_3^0$$

$$\text{or } \alpha_3^1 \geq M(k) \alpha_1^1$$

$$\text{where } M(k) = \begin{cases} 1 & \text{for } k = 1 \\ \binom{2k+1}{k+1} - (2k+1) & \text{for } 2 \leq k \leq 3 \\ \binom{2k+1}{k+1} - 2k & \text{for } k \geq 4 \end{cases}$$

where $P_{Le}(2k, n)$ is the probability of error of a $2k$ state linear detector and $P_{Me}(2^{2(k+1)-1}, n)$ is the probability of error of a $2^{2(k+1)-1}$ state majority rule

detector.

Theorem I establishes an upper bound on the probability of error for a $2k$ state linear detector. Hence using equation (3) one can find a sufficient number of states and the threshold settings for a linear detector such that the probability of error is below some desired value.

Theorem II establishes a sufficient condition for a majority rule detector to be a lower bound on the performance of a linear detector. Unfortunately, the condition that $\alpha_1^0 \geq M(k) \alpha_3^0$ can be severe at times.

Equation (3) was programmed on a computer and the probability of error for a 2^{2k-1} state majority rule detector versus threshold settings is plotted in Fig. 4, for $k = 2, \dots, 6$, $n = 32$, and Gaussian statistics with $\gamma = 1$.

Fig. 5 shows the variation of probability of error at optimum threshold settings versus signal separation for a 2^9 state majority rule detector, with Gaussian statistics.

Finally, fig. 6 shows the probability of error at optimum threshold settings versus the number of observations for several small memory linear detectors. Again all statistics are Gaussian with $\gamma = 1$. Also plotted in this figure is the optimum probability of error obtained by using a standard likelihood ratio test.

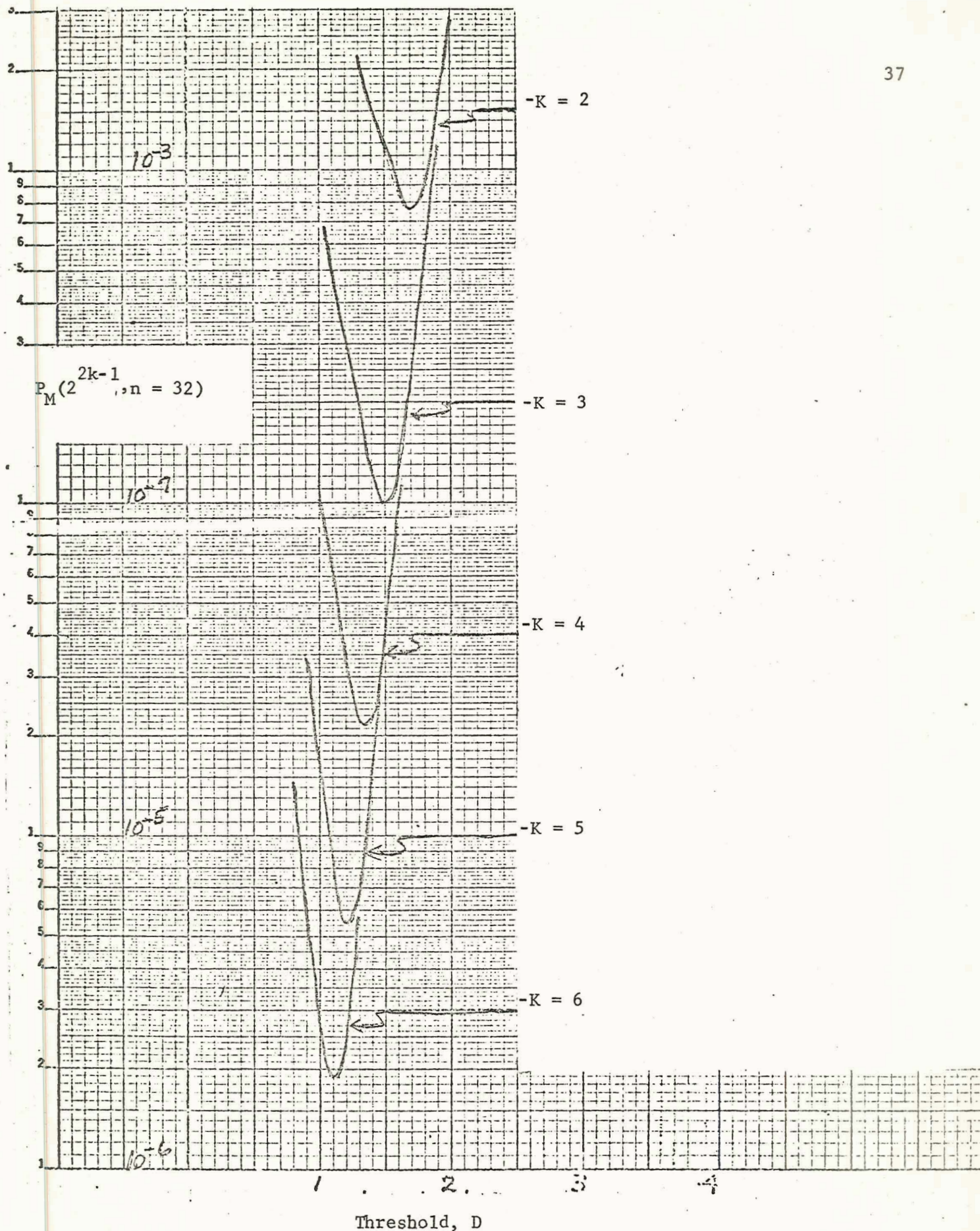


Fig. 6. Probability of Error vs. Threshold for a Majority Rule Detector and Gaussian Observations, with $\gamma=1$ and $n=32$.

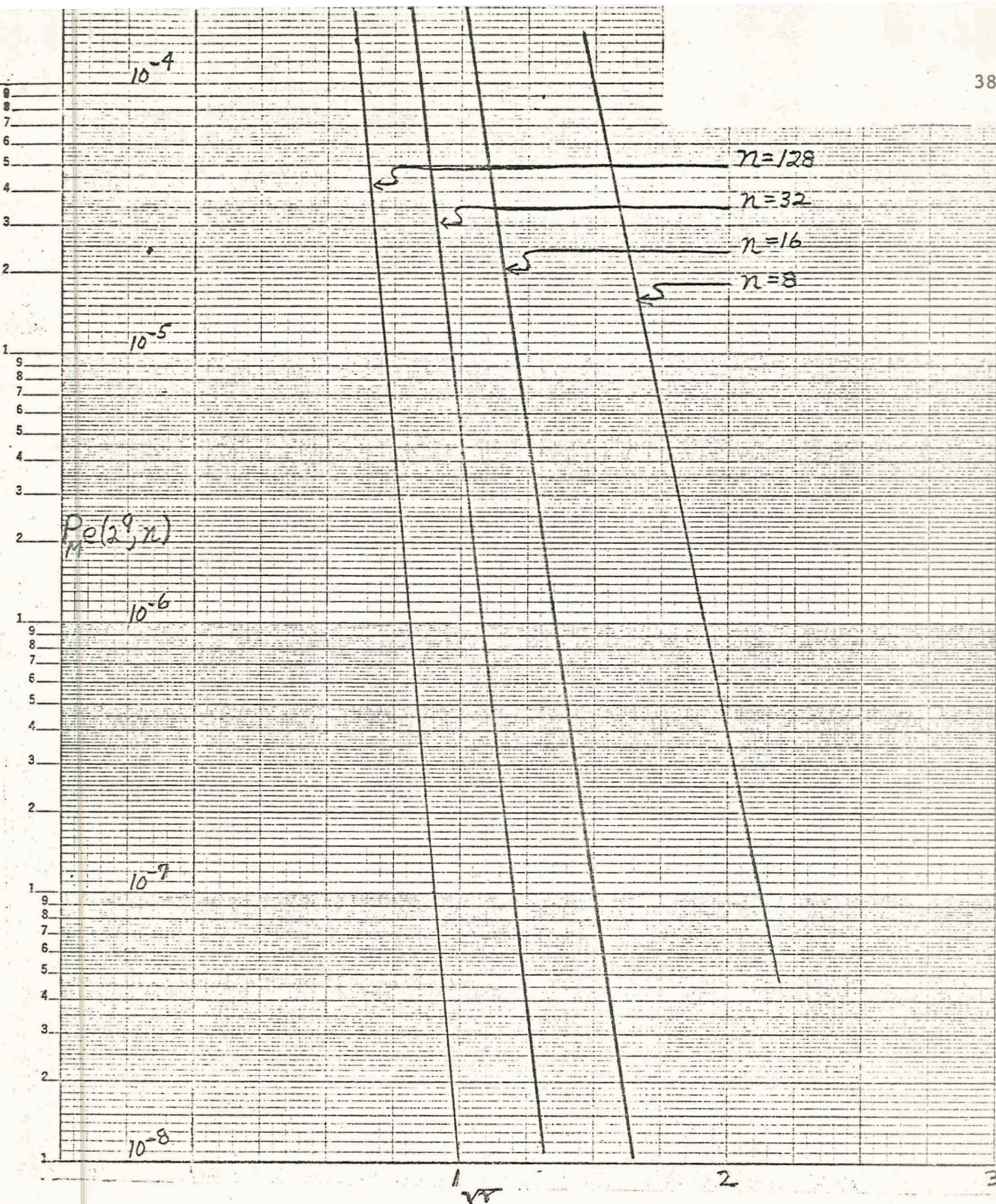


Fig. 5. Probability of Error for a 2^9 State Majority Rule Detector vs. $\sqrt{N}$ for Gaussian Observations.

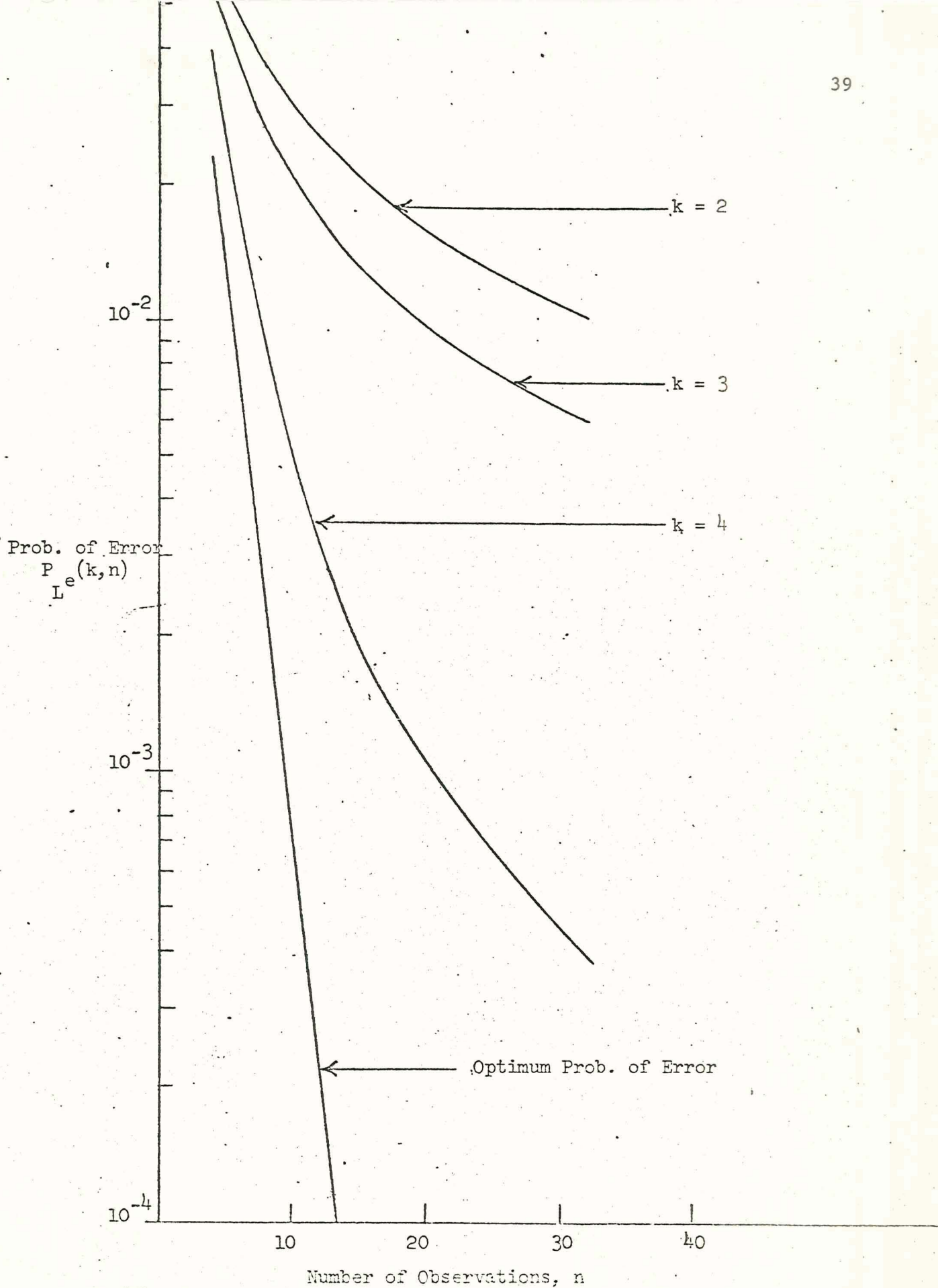


Fig. 6. Probability of Error at Optimum Threshold Setting vs. Number of Observations for a k State Linear Detector and Gaussian Observations, with $\gamma = 1$.