



Adaptive receiver structures for time-dispersive channels
by Baldev Krishan

A thesis submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY in Electrical Engineering
Montana State University
© Copyright by Baldev Krishan (1974)

Abstract:

Intersymbol interference is one of the main problems encountered in high-speed data transmission over telephone, radio and undersea channels. This interference, caused by the time dispersion and multipath characteristics of the channels, has been a major hindrance to increased data rates at acceptable error probabilities. Various linear and nonlinear receivers have been proposed to combat the intersymbol interference problem. It has been found by Proakis and Magee that a receiver consisting of a Viterbi decoder in combination with a channel estimator provides a receiver structure which is realizable and asymptotically optimum. The object of this research has been the development of faster adjustment algorithms for the channel estimator so that the estimator can be used on more rapidly varying channels. Forney has proposed a suboptimum algorithm for sequence estimation in the presence of intersymbol interference. This suboptimum algorithm is suitable for a particular type of partial response system. Another aspect of this research is to study the sensitivity of Forney's suboptimum algorithm and to develop an adaptive equalization scheme for making Forney's suboptimum algorithm suitable for some other type of partial response system. Simulation results are obtained for this adaptive equalization scheme.

ADAPTIVE RECEIVER STRUCTURES
FOR TIME-DISPERSIVE CHANNELS

by

BALDEV KRISHAN

A thesis submitted to the Graduate Faculty in partial
fulfillment of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Approved:

Paul E. Whitlick
Head, Major Department

C. K. Rushforth
Chairman, Examining Committee

Henry L. Parsons
Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

March, 1974

ACKNOWLEDGMENTS

The author wishes to express his sincere gratitude to his advisor, Dr. C. K. Rushforth, for his active participation and valuable guidance during the course of this research.

The author would also like to thank Dr. D. A. Pierre and Dr. D. N. March for their comments and helpful suggestions.

The author is indebted to Dr. P. E. Uhlrich for the financial support given during the author's stay at Montana State University and to Professor R. C. Seibel for his assistance. The author expresses his appreciation to Mrs. Jean Julian for her excellent technical typing.

Finally, the author would like to thank his parents without whose guidance and support this thesis would not have been possible, and to the author's wife, Tripta, for her patience and encouragement throughout the development of this thesis.

TABLE OF CONTENTS

	<u>Page</u>
VITA	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
ABSTRACT	xi
Chapter	
I. INTRODUCTION	1
1.1 Historical Review	1
1.2 Statement of the Problem	19
II. DEVELOPMENT OF A DISCRETE-TIME CHANNEL MODEL	23
2.1 P.A.M. Communication System	23
2.2 The Matched Filter	25
2.3 The Whitened Matched Filter	28
2.4 Discrete-Time Channel Model	32
III. SEQUENCE ESTIMATION TECHNIQUES	38
3.1 Maximum-Likelihood Sequence Estimation	38
3.2 The Viterbi Algorithm	41

	<u>Page</u>
3.3 Performance of the Maximum-Likelihood Sequence Estimation Receiver	45
3.4 Various Types of Adaptive Maximum-Likelihood Sequence Estimation Receivers and Their Performance . . .	54
3.4.1 An Adaptive Maximum-Likelihood Sequence Estimation Receiver Using a Channel Estimator . .	55
3.4.2 Performance of the Adaptive MLSE Receiver Using a Channel Estimator	63
3.4.3 An Adaptive Maximum-Likelihood Sequence Estimation Receiver Using a Linear Equalizer . . .	69
3.4.4 Performance of an Adaptive MLSE Receiver Using a Linear Equalizer	74
3.4.5 Comparison of These Two Adaptive Receiver Structures	78
IV. DEVELOPMENT OF FASTER CONVERGING CHANNEL ESTIMATION ALGORITHMS	79
4.1 First-Order Algorithm	80
4.1.1 Variable Step Size	80
4.1.2 Performance of First-Order Algorithm and Simulation Results	90
4.2 Second-Order Algorithm	97
4.2.1 Second-Order Algorithm Derivation	97

	<u>Page</u>
4.2.2 Performance of Second-Order Algorithm and Simulation Results	102
V. SENSITIVITY STUDY OF FORNEY'S SUBOPTIMUM ALGORITHM AND EQUALIZATION SCHEME FOR MAKING FORNEY'S SUBOPTIMUM ALGORITHM ADAPTIVE	108
5.1 Forney's Suboptimum Algorithm	108
5.2 Sensitivity Study of Forney's Suboptimum Algorithm	113
5.3 Adaptive Version of Suboptimum Algorithm	118
5.4 Simulation Results Obtained for an Adaptive Equalization Scheme . . .	124
VI. SUMMARY AND SUGGESTED FUTURE RESEARCH . . .	129
6.1 Introduction	129
6.2 Summary and Conclusions	129
6.3 Suggestions for Future Research	132
APPENDICES	
A. DERIVATION OF EQUATIONS (3.22), (4.3), (4.8), AND (4.19)	135
B. DERIVATION OF EQUATIONS (4.47-4.50)	139
C. ANALYSIS FOR RATE OF CONVERGENCE OF A VARIABLE-STEP-SIZE ALGORITHM	143
D. METHOD OF COMPUTER SIMULATION	146
REFERENCES	171

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Comparison of Some Partial Response Systems	18
2. Performance Loss and Corresponding Nonunique Minimum-Distance Impulse Response	53

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1.1 Linear distortion	2
1.2 Linear time-varying channel model	5
1.3 Optimum receiver	8
1.4 Transversal filter	9
1.5 Conventional linear receiver with equalizer	12
1.6 A linear adaptive equalizer	13
2.1 PAM transmission system	24
2.2 Maximum-likelihood sequence estimation receiver	24
2.3 Block diagram of PAM communication system . .	33
2.4 Discrete-time channel model	35
3.1 Discrete-time channel model	39
3.2 Probability of error versus output signal- to-noise ratio for $m=2$ and $m=4$	51
3.3 An adaptive maximum-likelihood receiver . . .	56
3.4 Channel estimator	57
3.5 Samples of channel impulse response	66
3.6(a) $\Pr(e)$ vs. input SNR in dB.	67
3.6(b) $\Pr(e)$ vs. input SNR in dB.	68
3.7 An adaptive receiver with equalizer	70

<u>Figure</u>	<u>Page</u>
3.8 Proposed adaptive receiver	72
3.9 Average bit-error probability vs. output SNR	76
3.10 (a) Samples of channel response, (b) Desired channel-equalizer response, (c) Frequency-domain response	77
4.1 Channel estimator	81
4.2 Samples of channel impulse response	91
4.3(a) Plot of norm of the error vector vs. iterations	93
4.3(b) Plot of norm of the error vector vs. iterations	94
4.4(a) Plot of norm of the error vector vs. iterations	95
4.4(b) Plot of norm of the error vector vs. iterations	96
4.5(a) Plot of norm of the error vector vs. iterations	104
4.5(b) Plot of norm of the error vector vs. iterations	105
4.6(a) Plot of norm of the error vector vs. iterations	106
4.6(b) Plot of norm of the error vector vs. iterations	107
5.1 Inverse linear filter with response $1/(1-D)$	110
5.2 Error-correction algorithm block diagram . .	110

<u>Figure</u>		<u>Page</u>
5.3(a)	Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-D$	114
5.3(b)	Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-0.98D$	115
5.3(c)	Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-D-0.1D^2$	116
5.3(d)	Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-D+0.2D^2$	117
5.4	Block diagram of equalization scheme for suboptimum algorithm	119
5.5	Suboptimum algorithm with equalizer	125
5.6(a)	Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-D+0.5D^2$	126
5.6(b)	Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-0.5D-0.5D^2$	127

ABSTRACT

Intersymbol interference is one of the main problems encountered in high-speed data transmission over telephone, radio and undersea channels. This interference, caused by the time dispersion and multipath characteristics of the channels, has been a major hindrance to increased data rates at acceptable error probabilities. Various linear and nonlinear receivers have been proposed to combat the intersymbol interference problem. It has been found by Proakis and Magee that a receiver consisting of a Viterbi decoder in combination with a channel estimator provides a receiver structure which is realizable and asymptotically optimum. The object of this research has been the development of faster adjustment algorithms for the channel estimator so that the estimator can be used on more rapidly varying channels. Forney has proposed a suboptimum algorithm for sequence estimation in the presence of intersymbol interference. This suboptimum algorithm is suitable for a particular type of partial response system. Another aspect of this research is to study the sensitivity of Forney's suboptimum algorithm and to develop an adaptive equalization scheme for making Forney's suboptimum algorithm suitable for some other type of partial response system. Simulation results are obtained for this adaptive equalization scheme.

Chapter I

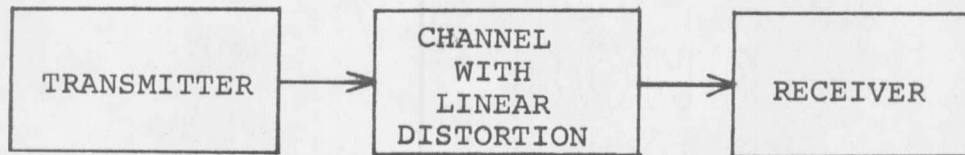
INTRODUCTION

1.1 Historical Review

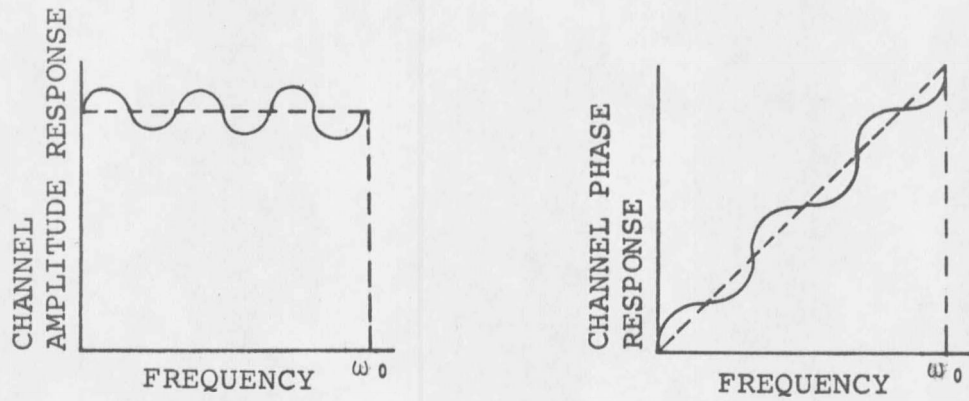
It seems inherent in nature that signals are dispersed--that is, spread out and changed--in transmission. This dispersion manifests itself in two ways: in radio communication it is called multipath, and in wire communication it is called linear distortion. Multipath transmission can be viewed as communication through a group of parallel, distortion-free channels of different lengths. The wire communication channel may have a single obvious transmission path but a non-ideal frequency characteristic (an ideal transmission frequency characteristic is here defined as one having a flat amplitude frequency response and a linear phase frequency response). Figure 1.1 displays frequency response curves for such a non-ideal channel. This linear distortion present in wire communication causes what is called intersymbol interference.

Intersymbol Interference:

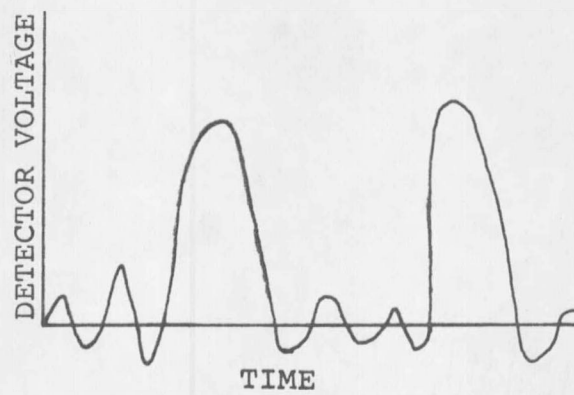
Transmission of data on voice channels, over the switched telephone network, has increased markedly over the past few years. The data is used to amplitude modulate a train of identically shaped pulses. These pulses are



(a)



(b)



(c)

Figure 1.1 Linear distortion.

transmitted over the channel, demodulated by the receiver, and processed. Although the signal-to-noise ratio (SNR) is sufficient to permit higher rates, the non-uniform transmission characteristics of the channel causes what might be termed a distortion barrier prohibiting faster transmission. The distortion of data pulses by the channel results in these pulses being spread out in time so as to overlap other transmitted pulses. The phenomenon of pulse overlap and the resultant difficulty with receiver decisions is termed intersymbol interference. This intersymbol interference is one of the chief degrading factors in present data-transmission system and becomes the determining factor in the design of higher rate system.

Intersymbol interference normally arises in pulse modulation systems. It is the primary impediment to reliable high-rate digital transmission over high signal-to-noise ratio narrow bandwidth channels such as voice-grade telephone circuits. Sometimes a controlled amount of intersymbol interference is introduced deliberately to achieve certain beneficial effects in data transmission. These systems, which utilize a controlled amount of intersymbol interference, are called partial response systems [41-42].

The channel can be represented, as shown in Figure 1.2, by a time-varying impulse response $c(t)$, plus additive white noise. The output of this channel typically differs from its input in two ways. First, the channel may modify the message in a deterministic, although not necessarily known, fashion. Examples of this type of modification are dispersion, frequency offset, and nonlinear action. The channel can also corrupt the transmitted signal statistically. In this category are various types of additive and multiplicative noise such as thermal and impulse noise. Normally, the frequency offset is very small compared to the system bandwidth and is, therefore, neglected. Impulse noise is characterized by long, quiet intervals followed by bursts of much higher amplitudes than would be predicted by a normal or Gaussian distribution law, and, therefore, this noise is usually not considered in the mathematical model of the channel. The model, therefore, accounts for the most deterministic impairment, time dispersion, as well as the most prevalent and mathematically tractable Gaussian noise.

Intersymbol interference represents a kind of deterministic impairment. If the channel characteristics

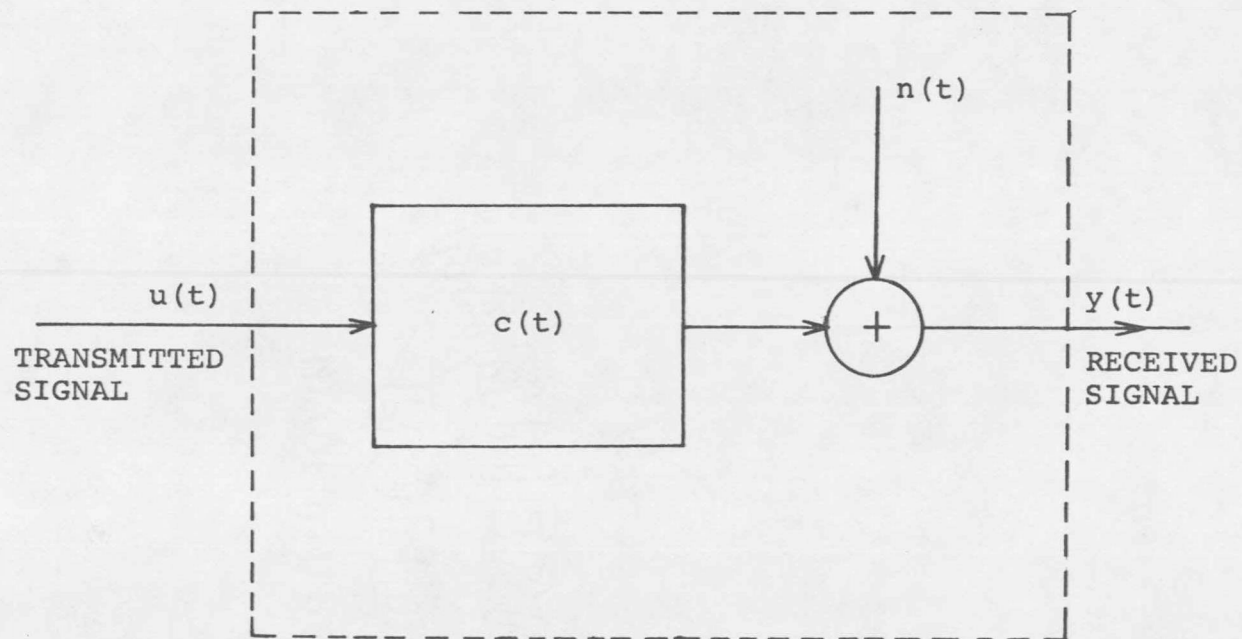


Figure 1.2 Linear time-varying channel model.

are known, it may be possible in theory to remove intersymbol interference. However, the transmission channel is never known exactly; thus, it is impossible to eliminate completely the effects of intersymbol interference in real data transmission systems. There are several methods of combating intersymbol interference. The one which has been quite extensively used in symbol-by-symbol decision systems is that in which the transmitted signals are designed so that all signals except the desired symbol have zero value when the waveform is sampled at the receiver. This method is based on the work by Nyquist [1] and the resulting specification is known as the Nyquist criterion.

In the case of PAM (pulse-amplitude modulation) systems, different criteria have been used to design the optimum receiver in the presence of intersymbol interference. The different criteria used are: (1) Minimization of probability of error caused by intersymbol interference and noise, (2) Minimization of mean-square error resulting from intersymbol interference and noise.

Under either of these criteria, the optimum linear receiver is invariably a matched filter followed by a

transversal filter. The matched filter might be interpreted as treating the noise while the transversal filter corrects intersymbol interference. When the signal-to-noise ratio is low, the principal correction must come from the matched filter; for high signal-to-noise ratios, intersymbol interference must be almost completely suppressed and the principal correction effected by the transversal filter.

The approach used in deriving the optimum receiver has been to minimize the effect of intersymbol interference at the receiver. The decision about the input sequence is again made on a symbol-by-symbol basis. Most of the work in the derivation of optimum linear receiver has been done by Tufts, George, Aaron, and Ericson [2-6]. Ericson [5] has also shown that any reasonable criterion of goodness will lead to an optimum filter that has the form discussed earlier. Berger and Tufts [2,4] considered the joint optimization of transmitter and receiver in pulse amplitude modulation using the criterion of total mean-square error. The optimum linear receiver based on receiver filter optimization is shown in Figure 1.3. The transversal filter is shown in Figure 1.4.

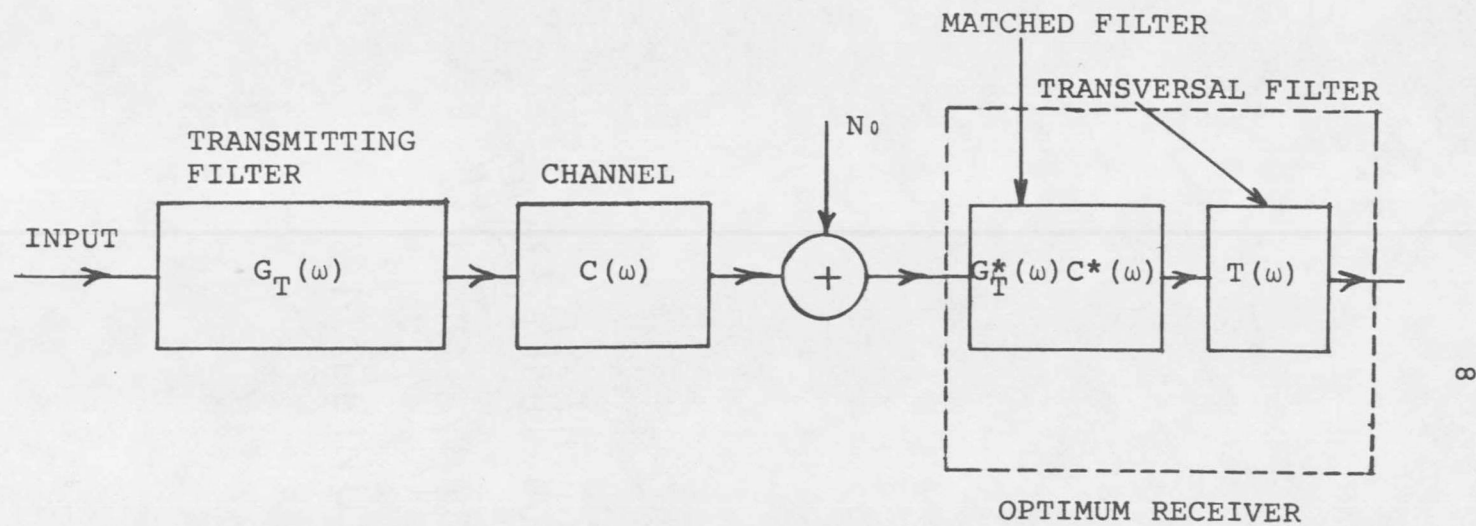


Figure 1.3 Optimum receiver.

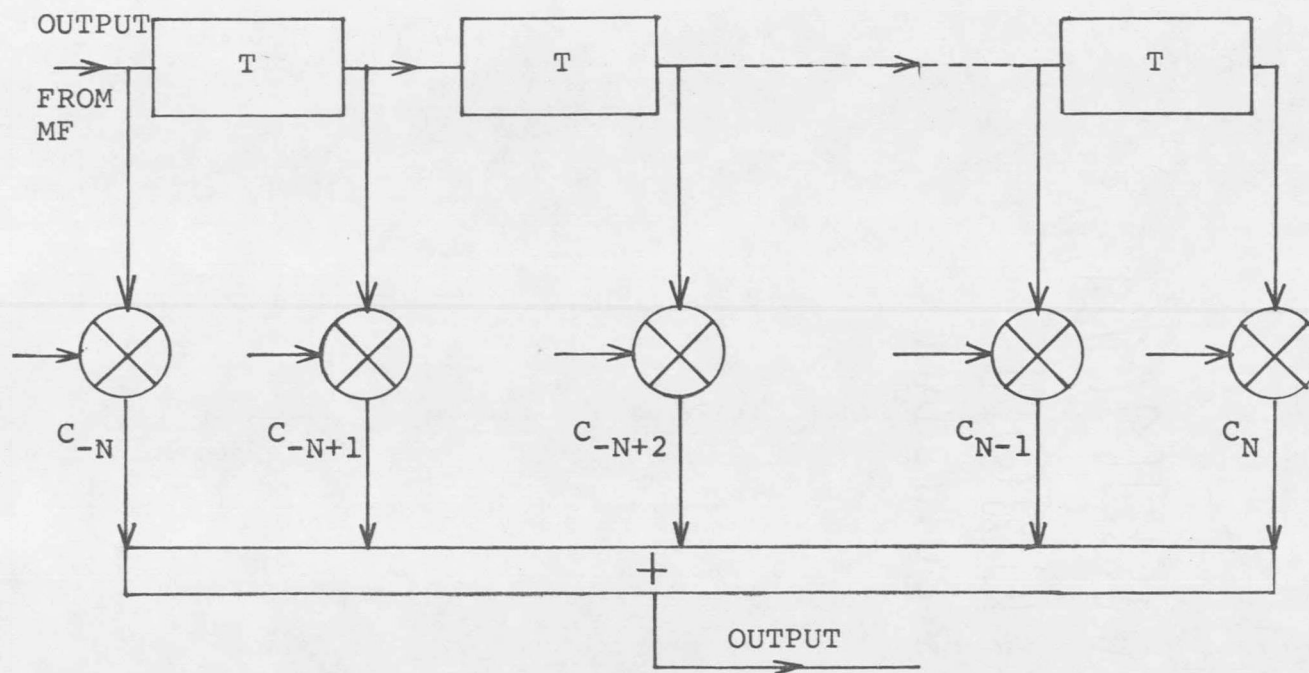


Figure 1.4 Transversal filter.

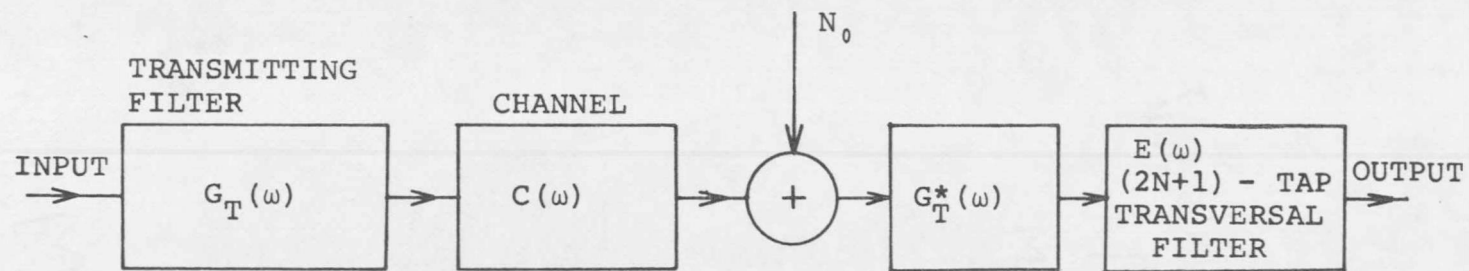
The transversal filter consists of a delay line tapped at T -sec intervals (the Nyquist interval). Each tap along the delay line is connected through an amplifier (and possibly an inverter to enable negative gains) to a summing bus which provides the output. The frequency response for the transversal filter is repetitive with points of symmetry in real and imaginary components located at $\frac{2\pi n}{T}$ radians.

All the optimum receivers that resulted from the work of Tufts, Berger, Aaron, and George [2-6] are linear receivers. They are called linear receivers because the outputs of the matched filter are combined linearly and the estimate of the input alphabet is made from the combined signal obtained at the output of the transversal filter.

In the design of the optimum receiver, it is often assumed that the channel characteristics are known. More commonly, the channel is not known. This implies that the receiver has to be adaptive to the unknown channel. One approach used in making the receiver adaptive is called the adaptive equalization approach. The idea of adaptive equalization was originated by Lucky [15-18]. The approach consists of deriving a receiver

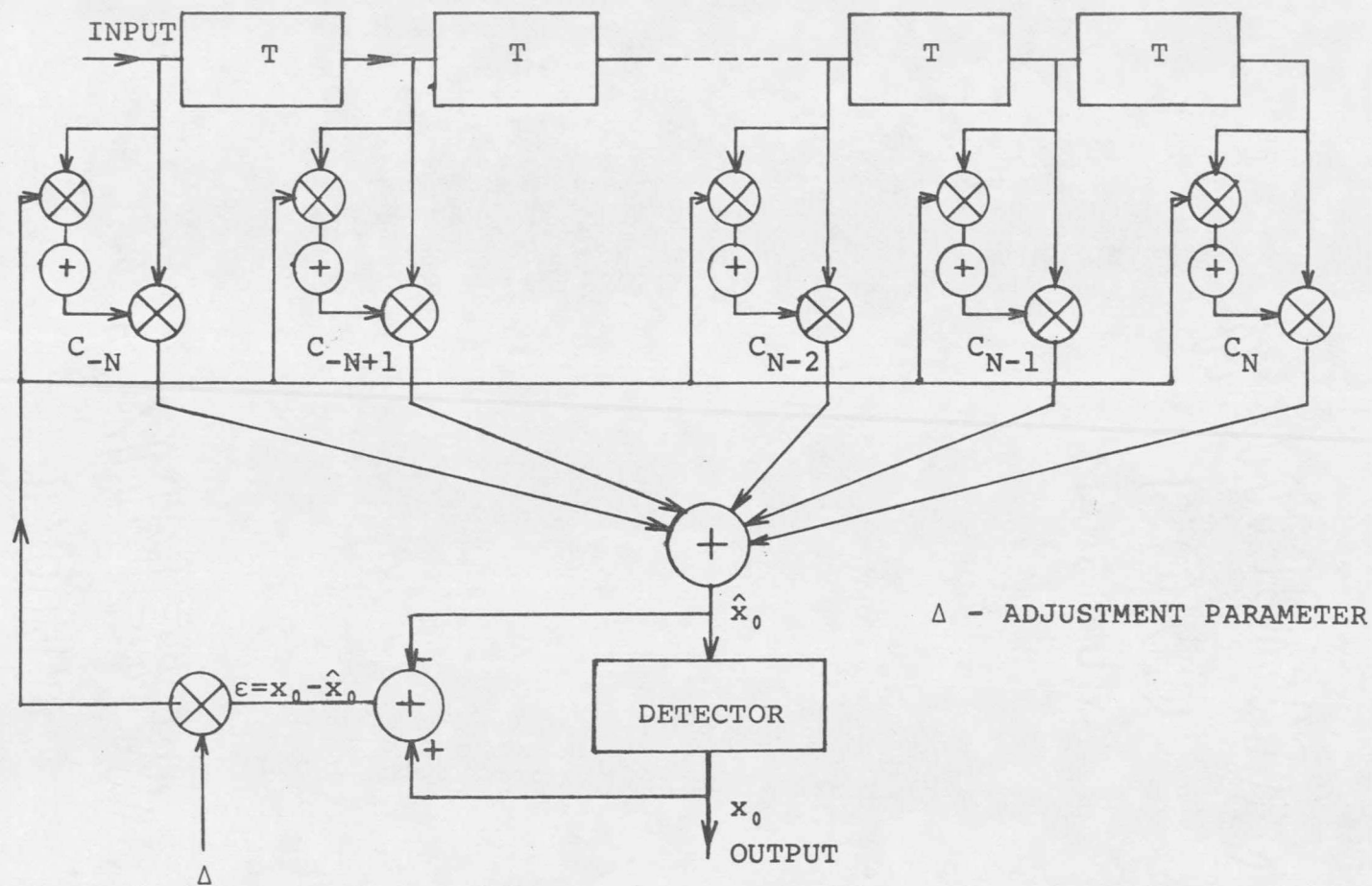
for the time-invariant case and then using a fixed structure with a limited number of parameters, having it adjust to the changing conditions in the channel. This fixed structure with adjustable parameters is called an equalizer. A basic receiver structure has evolved from this approach and is shown in Figure 1.5. The equalizer used in the adaptive receiver is shown in Figure 1.6.

This receiver basically consists of a matched filter followed by a tapped delay line with a weighted sum of the taps being added to form an estimate of the information symbol. This receiver structure is again linear since it provides a linear estimate for the information symbol which is then quantized. A mean-square error criterion is used in the adjustment of the filter shown in Figure 1.6. It is also possible to adjust the parameters using a zero intersymbol interference forcing algorithm. This zero forcing algorithm is the original method developed by Lucky [15] for an adaptive receiver structure. Unfortunately, the zero forcing algorithm is nonconvergent for highly dispersive channels. Gersho [20] has suggested an adaptive equalization scheme using a mean-square error criterion, and it has been found that



12

Figure 1.5 Conventional linear receiver with equalizer.



13

Figure 1.6 A linear adaptive equalizer.

this scheme will converge for any kind of channel. The adjustment parameter (or step size) Δ also affects the convergence of the algorithm. The effect of adjustment parameter Δ has been analyzed by Proakis and Miller [21]. The disadvantage of the linear equalization scheme is that it tends to enhance the noise, particularly at the band edges.

Although the linear receivers discussed earlier are called optimum receivers, they are not optimum in the true sense for the detection of a sequence. The reason for this is that the optimum detection of an N -symbol message requires a joint decision concerning the data sequence $\{x_n\}$, $n = 0, 1, \dots, N-1$, after processing the entire received signal. Since m^N messages are possible, the size of the optimum receiver grows exponentially with the message length N , where m is the size of the input alphabet. In practice, such a receiver is an impossibility. A basic requirement that any practical receiver must meet is that the number of computations required to process a message of N symbols must grow only slowly with N . A signal detection system based on an iterative procedure which meets this requirement (and under certain conditions is optimal), has been proposed by Chang and

Hancock [9]. This nonlinear receiver structure assumes that the symbol x_n causes intersymbol interference only with the $(K-1)$ succeeding symbols $x_{n+1}, \dots, x_{n+K-1}$. The symbols are then detected in block of K symbols. Thus, each block has m^K possible values from which a decision must be made. The advantages of blocking the data into groups of K symbols will not be apparent without discussing the original paper by Chang and Hancock [9] in great detail. In all the computations involved in this procedure, m^K sets of numbers are involved. Furthermore, each current a posteriori probability, of which there are m^K , requires m^K calculations itself. Therefore, in most applications the cost of implementation would be prohibitive.

There have been several other nonlinear receiver structures developed to compensate for intersymbol interference. Austin [8] has analyzed a nonlinear nonadaptive structure which employs decision feedback, i.e., the previous decisions of the receiver are used to cancel the tails of the intersymbol interference. Hilborn [13] has employed the theory of "unsupervised learning" to find a Bayes optimal m -ary communication receiver which operates causally on a received data sequence. This structure is

optimum (but nonlinear) for detection of a digital sequence in the presence of intersymbol interference.

Abend and Fritchman [12] have derived an optimum nonlinear detector under a fixed delay constraint for channels having memory and additive noise. The resulting receiver is recursive and does not grow with the message length. Ungerboeck [14] has provided a general solution to the problem when using binary bipolar signals. All these nonlinear receivers have the disadvantage of being complex in structure. The performance of nonlinear receivers is very difficult to evaluate; resort is often made to simulation. Because of the intimidating complexity of these nonlinear receivers, they are not considered practical solutions to the problem of intersymbol interference.

Recently, Forney [22] has introduced a receiver structure consisting of a linear filter, called a whitened matched filter, a symbol rate sampler and a recursive nonlinear processor, called the Viterbi algorithm. This structure is a maximum-likelihood estimator of the entire sequence. The outputs of the whitened matched filter, sampled once for each input symbol, are shown to form a set of sufficient statistics for estimation of the input

sequence. The Viterbi algorithm is easier to implement than earlier optimum nonlinear processors, and its performance can be easily and accurately estimated. It is shown that performance (by whatever criterion) is effectively as good as could be attained by any receiver structure and in many cases is as good as if intersymbol interference were absent. Forney [22] has also shown that the channel for digital data transmission can be represented as a convolutional coder when the channel impulse response is of finite duration. Since the performance of the Viterbi algorithm can be easily analyzed and the Viterbi algorithm can be implemented in a straight-forward manner, the use of this algorithm has potential for combating the intersymbol interference (ISI). A comparison of the partial-response schemes proposed by Kretzmer [41] on the basis of error probability of the maximum-likelihood sequence estimation (MLSE) is presented in Table 1. The discrete channel model (discussed in Chapter II) is used in evaluating the performance of the three detection methods listed in Table 1. Also note that for the decision-feedback equalizer, degradation in performance due to error propagation is not taken into account. The channel $f(D)$ represents

Table 1. Comparison of Some Partial Response Systems

Channel $f(D)$	Output SNR degradation over ideal binary in dB.		
	Precoded with level detection	Decision-feedback equalization	Maximum likelihood sequence esti.
$1+D$	3.2	3.00	0.5
$0.5+D+0.5D^2$	8.0	7.75	2.2
$1+0.5D-0.5D^2$	8.0	1.6	0.33
$1-D^2$	3.2	3.0	0.5
$-0.5+D-0.5D^2$	8.0	7.75	2.2

the sampled response of the channel in terms of a D-transform.¹

We see that with maximum-likelihood sequence estimation the output SNR (signal-to-noise ratio) degradation relative to ideal binary transmission is much less than for any bit-by-bit detection method. The receiver structure proposed by Forney [22] is discussed in greater detail in Chapter III. Forney [22] has also suggested a simple suboptimum algorithm which works as well as the optimum Viterbi algorithm for a particular type of partial-response system.

1.2 Statement of the Problem

One of the problems encountered in Forney's receiver structure is that this structure is suitable for known channels only. In the real world, the impulse response of the channel is not usually known. Proakis and Magee [31] have developed an adaptive receiver for maximum-likelihood sequence estimation of a digital sequence in the presence

¹The D-transform of a certain function $w(t)$ is given by

$$W(D) = \sum_{n=-\infty}^{n=\infty} w(nT) D^n$$

of intersymbol interference for slowly varying channels. In this adaptive receiver structure, a channel estimator has been developed that, when used in combination with the Viterbi algorithm, forms an adaptive maximum-likelihood sequence estimator. The major problem faced in this adaptive receiver is that the channel estimator is quite slow in convergence. Part of the research described in this thesis has been carried out to develop more rapidly converging algorithms for channel estimation.

As mentioned earlier, Forney [22] has developed a suboptimum algorithm which works as well as the Viterbi algorithm. This suboptimum algorithm has been derived for a particular type of partial-response system. An obvious question arises as to the sensitivity of this algorithm. A second objective of this research is to study the sensitivity of the suboptimum algorithm when the partial system response is slightly different from the required response. It is found that analytic results are hard to obtain, and the sensitivity analysis is carried out by simulation.

Finally a scheme has been developed in this thesis to make the suboptimum algorithm suitable for partial-response system which is different from the one for which

the suboptimum algorithm has been developed. A linear equalizer has been used in conjunction with the channel model and the purpose of the equalizer is to make the overall response (of channel and equalizer) behave like the desired response for the suboptimum algorithm. This scheme is evaluated by simulation results.

The outline of the thesis is as follows:

In Chapter II, a discrete-time channel model is developed. The discussion of this discrete-time channel model is important because this model will be used throughout the thesis. It is shown that the discrete-time channel model is equivalent to the continuous-time channel.

Chapter III deals with the maximum-likelihood receiver for sequence estimation. Basically, this chapter describes the work by Forney [22]. Limitations of Forney's maximum-likelihood sequence estimation receiver are discussed. Several schemes developed for adaptive MLSE receivers are presented in detail. Relative comparison of adaptive MLSE receivers is also made.

Chapter IV describes more rapidly converging algorithms developed in this thesis for channel estimation. Analytic results are derived, and these analytic results

have been verified by computer simulation. This chapter also compares the results of the algorithms with those obtained by Proakis and Magee [31] using fixed-step-size algorithm.

Chapter V basically contains a description of the suboptimum algorithm developed by Forney [22]. Results obtained by computer simulation regarding sensitivity of the suboptimum algorithm are also given. Finally, an adaptive equalization scheme developed by the author for making this suboptimum algorithm suitable for other channel responses is discussed.

Chapter VI summarizes the results, discusses their significance, and points out possible extensions. The appendices derive some of the equations and discuss the computer simulation.

Chapter II

DEVELOPMENT OF A DISCRETE-TIME CHANNEL MODEL

In this chapter, a discrete-time channel model is developed which is equivalent to the continuous-time model of Figure 1.2, and this discrete model is much easier to use. Moreover, the continuous model is not convenient for most computations.

2.1 P.A.M. Communication System

A pulse amplitude modulation scheme is considered in developing this discrete model. The simplest model of digital communication system subject to intersymbol interference occurs in pulse amplitude modulation (PAM), illustrated in Figure 2.1. The actual transmitted signal $u(t)$ can be written as

$$u(t) = \sum_{k=0}^K x_k g_T(t-kT) \quad (2.1)$$

where $\{x_k\}$ is a sequence of real numbers drawn from a discrete alphabet, $g_T(t)$ is the transmitting filter, and T is the sampling interval. The actual received signal $r(t)$ is characterized by

$$r(t) = \sum_{k=0}^K x_k g_T(t-kT) * c(t) + n(t) \quad (2.2)$$

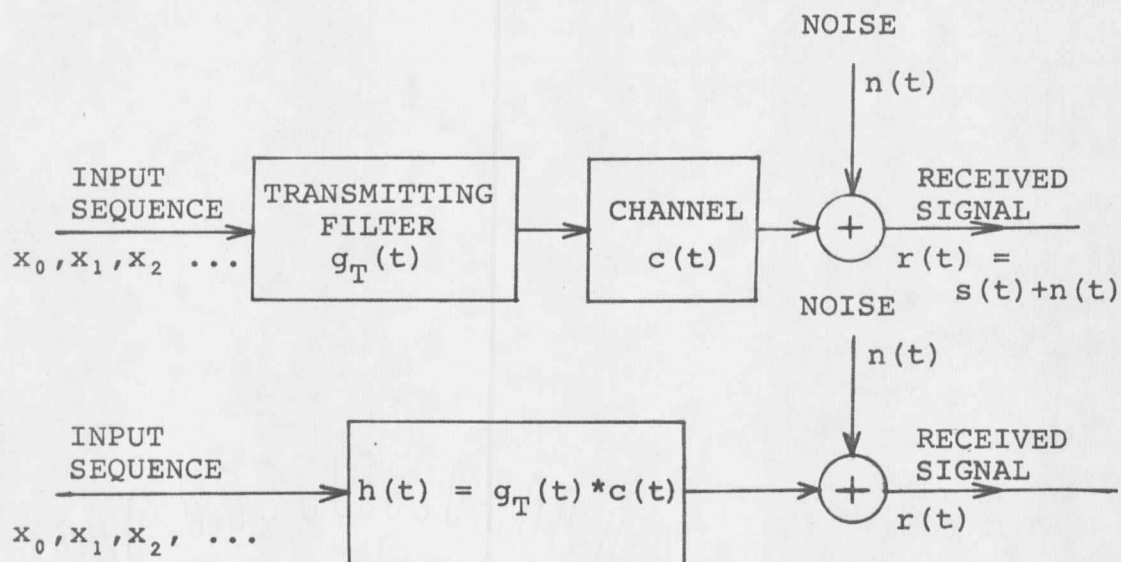


Figure 2.1 PAM transmission system.

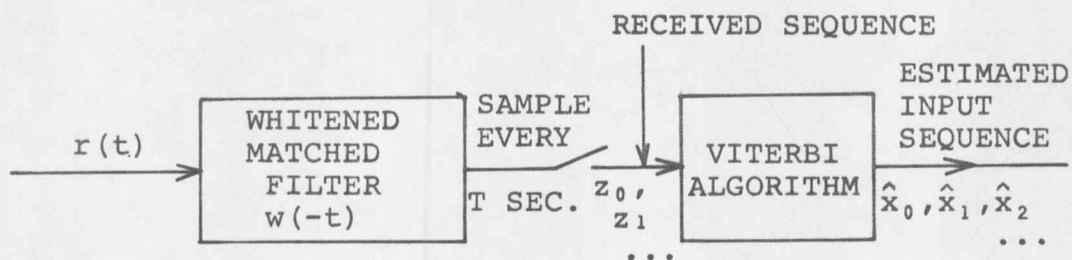


Figure 2.2 Maximum-likelihood sequence estimation receiver.

where the duration of the impulse response $c(t)$ is longer than the time T between symbols and where $*$ denotes convolution. The overall response of the transmitting filter and channel is $h(t)$; $h(t)$ is represented as

$$h(t) = g_T(t) * c(t) \quad (2.3)$$

The noise is assumed white and Gaussian, and the channel impulse response is assumed to be of finite duration. The maximum-likelihood sequence estimation receiver developed by Forney [22] is shown in Figure 2.2. The receiver consists of a linear filter called the whitened matched filter (WMF), a sampler taking samples once every T seconds, and a nonlinear recursive algorithm called the Viterbi algorithm. This receiver structure is optimum for maximum-likelihood estimation of the entire transmitted sequence, provided unbounded delay is allowed at the output, and is effectively optimum in the same sense for reasonable finite delays.

2.2 The Matched Filter

Reference to Figure 2.1 shows that the received signal $r(t)$ is $s(t)$ plus white Gaussian noise $n(t)$. When the intersymbol interference is zero (also called

one-shot communication), the matched filter is the optimum receiver. In the case of serial data transmission, the sampled (every T seconds) outputs of the matched filter have been shown by Forney [22] to provide a set of sufficient statistics for decoding the transmitted sequence. That is, he has shown that the sampled values of the output of the matched filter provide the same information for making a decision as the continuous received waveform. It is true because in the detection of signals that are linear combinations of some finite set of square-integrable¹ basis functions, the output of a bank of matched filters, one matched to each basis function, form a set of sufficient statistics for estimating the transmitting signals. Thus the $K+1$ quantities

$$a_k = \int_{-\infty}^{\infty} r(t) h(t-kT) dt \quad 0 \leq k \leq K \quad (2.4)$$

form a set of sufficient statistics for estimation of the x_k , $0 \leq k \leq K$. But it turns out that a set of sufficient

¹The impulse response $h(t)$ is a square-integrable response if

$$\| h(t) \| \triangleq \int_{-\infty}^{\infty} h^2(t) dt < \infty$$

statistics for estimation of the input sequence is simply the sampled outputs of a filter $h(-t)$ matched to $h(t)$.

$$a_k = \sum_{i=0}^K x_i \int_{-\infty}^{\infty} h(t-iT) h(t-kT) dt + \int_{-\infty}^{\infty} n(t) h(t-kT) dt \quad (2.5)$$

Now, let each integral term of (2.5) be represented as

$$R_{k-i} = \int_{-\infty}^{\infty} h(t-kT) h(t-iT) dt \quad (2.6)$$

$$\text{and} \quad n'_k = \int_{-\infty}^{\infty} n(t) h(t-kT) dt \quad (2.7)$$

Equation (2.5) reduces to

$$a_k = \sum_{i=0}^K x_i R_{k-i} + n'_k \quad (2.8)$$

The input noise $n(t)$ is white Gaussian with autocorrelation function $\sigma^2 \delta(t)$ where σ^2 is the spectral density of the noise $n(t)$. If the duration of $h(t)$ is larger than the input symbol separation, i.e., if $L > 1$, where L is the smallest integer such that $h(t) = 0$ for $t > LT$, then at least adjacent samples are correlated. The channel is thus characterized by a finite impulse response $h(t)$ of length L symbol intervals. To get an exact measure of the correlation of the noise samples, their

autocorrelation must be determined. Taking the expected value of the product $n_k^i n_i^i$ gives

$$E(n_k^i n_i^i) = \sigma^2 R_{k-i} \quad (2.9)$$

Referring to equation (2.6), it is evident that R_{k-i} is nonzero for $k-i < L$ because of the overlap. Also, R_{k-i} is symmetrical, i.e., $R_{k-i} = R_{i-k}$. Therefore, it is obvious from (2.9) that the noise samples n_k^i at the output of the matched filter are correlated whenever $L > 1$. But in order to insure the simple and recursive property of the Viterbi algorithm it is necessary that the output sequence before the final processor should be a sequence of statistically independent, identically distributed Gaussian random variables. Thus, the sample outputs of a matched filter need further processing.

2.3 The Whitened Matched Filter

When the noise is colored, it is a standard procedure in detection theory to have a prewhitening filter to make the noise white; this greatly simplifies optimum receiver derivations as well as performance computations. Forney [22] has basically used this technique to derive a so-called whitened matched filter (WMF). The

properties of this filter are that it provides an equivalent set of statistics while recovering the simplicity of the white noise.

Since the noise samples at the output of a matched filter $h(-t)$ are statistically dependent, some kind of whitening transformation must be performed in order to have the noise samples as statistically independent. From the theory of random variables [52] it is known that a linear transformation on a sequence of Gaussian random variables is also Gaussian. Further, if the random variables are statistically dependent, then there exists a linear transformation that will produce statistically independent random variables. The linear transformation on the output samples of a matched filter $h(-t)$ is accomplished by a transversal filter (tapped delay line). The cascade of a matched filter with a transversal filter characterized by a certain response (to be derived later) is called a whitened matched filter. The required response for the transversal filter is derived next.

Recall from Figure 2.1 that the input to the overall response $h(t)$ (of the transmitting filter and channel) is a sequence of numbers, each separated from its adjacent neighbors by T seconds. Associated with this

sequence is a formal power series in D . This series is called the D -transform of $x(t)$ and is given by

$$x(D) = \sum_{i=0}^K x_i D^i \quad (2.10)$$

where x_i is a symbol chosen at the transmitter from some predetermined alphabet; e.g., in the binary case x_i equals 1 or 0. D^i can be thought of as a delay operator representing iT units of delay. It can be shown easily that the D -transform of the sequence at the output of a matched filter can be represented as

$$a(D) = x(D) R(D) + n'(D) \quad (2.11)$$

where $x(D)$ is the D -transform of the input sequence, $n'(D)$ is the correlated noise D -transform and $R(D)$ is the D -transform of the sequence whose coefficients are given by (2.6). We define the pulse autocorrelation function of $h(t)$ as

$$R(D) = \sum_{k=1-L}^{L-1} R_k D^k \quad (2.12)$$

Since $R(D)$ has $2L-1$ nonzero terms, it has $2L-2$ complex roots; further $R(D) = R(D^{-1})$, the inverse of a root is also a root of $R(D)$, so the roots of $R(D)$ break up into

L-1 pairs. This makes it possible to factor $R(D)$ into a product of two polynomials, $f(D)$ and $f(D^{-1})$. From now onwards, a certain sequence will mean in fact the D-transform of that sequence. With this much background in mind, we are now in position to derive the transversal filter response so that the noise samples at the output of this filter are uncorrelated. The pulse autocorrelation function of $n'(D)$ is

$$R_{n'}(D) = \sigma^2 R(D) = \sigma^2 f(D) f(D^{-1}) \quad (2.13)$$

The white noise sequence must have a pulse autocorrelation function that is constant. We can represent the D-transform $n'(D)$ of the colored noise sequence by

$$n'(D) = n(D) f(D^{-1}) \quad (2.14)$$

where $n(D)$ is our desired white noise sequence. Consequently, we may write (2.11) as

$$a(D) = x(D) f(D) f(D^{-1}) + n(D) f(D^{-1}) \quad (2.15)$$

If we pass the sequence $a(D)$ through a filter with response $1/f(D^{-1})$ (transversal filter with response $1/f(D^{-1})$), we obtain

$$z(D) = \frac{a(D)}{f(D^{-1})} = x(D) f(D) + n(D) \quad (2.16)$$

with pulse autocorrelation for $n(D)$ equal to

$$E[n(D)n(D^{-1})] = \frac{E[n^*(D)n^*(D^{-1})]}{f(D)f(D)} = \frac{\sigma^2 R(D)}{R(D)} = \sigma^2 \quad (2.17)$$

where σ^2 is constant and is the spectral density of the input noise $n(t)$. Thus, noise at the output of the transversal filter is uncorrelated.

2.4 Discrete-Time Channel Model

Equation (2.16) tells us that the cascade of the transmitting filter, a PAM channel, a matched filter, and a transversal filter can be represented as a discrete-time filter with impulse response $f(D)$ plus an additive white Gaussian noise sequence $n(D)$. The block diagram of the PAM communication system including the receiver is shown in Figure 2.3.

We have now seen that by the use of a whitened matched filter we may confine our attention to the discrete-time model, without loss of optimality. Without loss of generality, we assume that $f(D)$ is a polynomial of degree v (equal to $L-1$) with nonzero constant term. Let us define the signal sequence

$$y(D) = x(D)f(D) \quad (2.18)$$

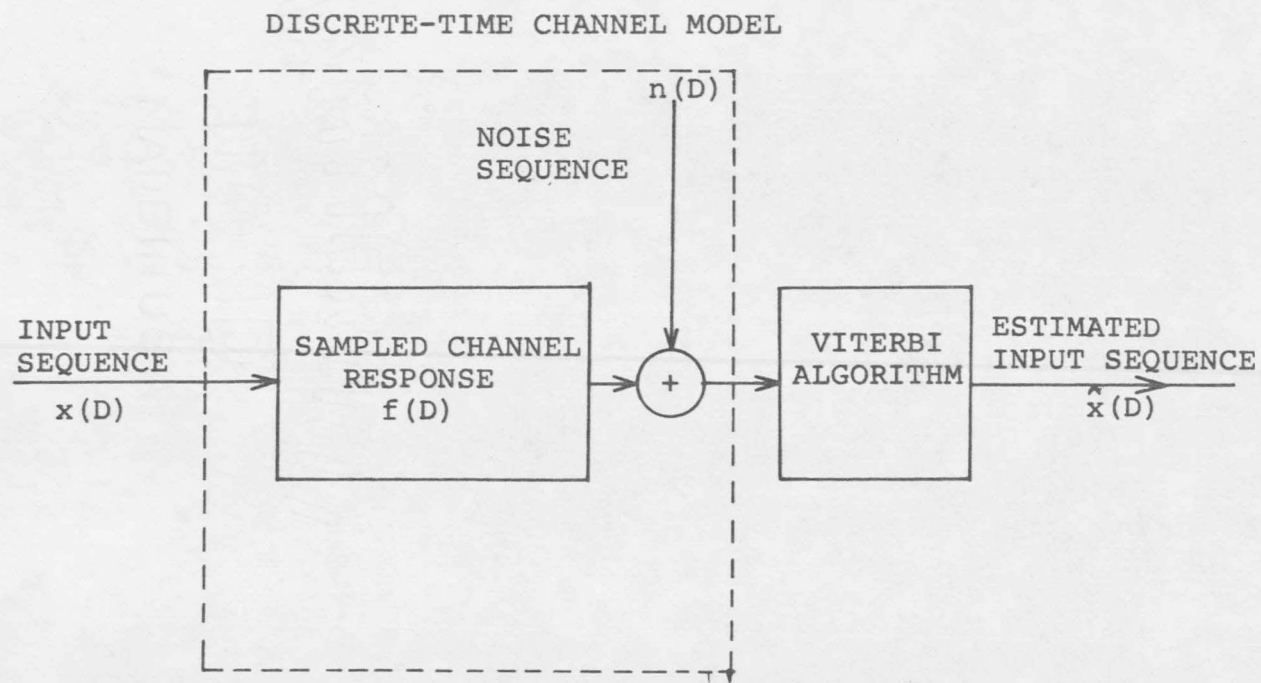
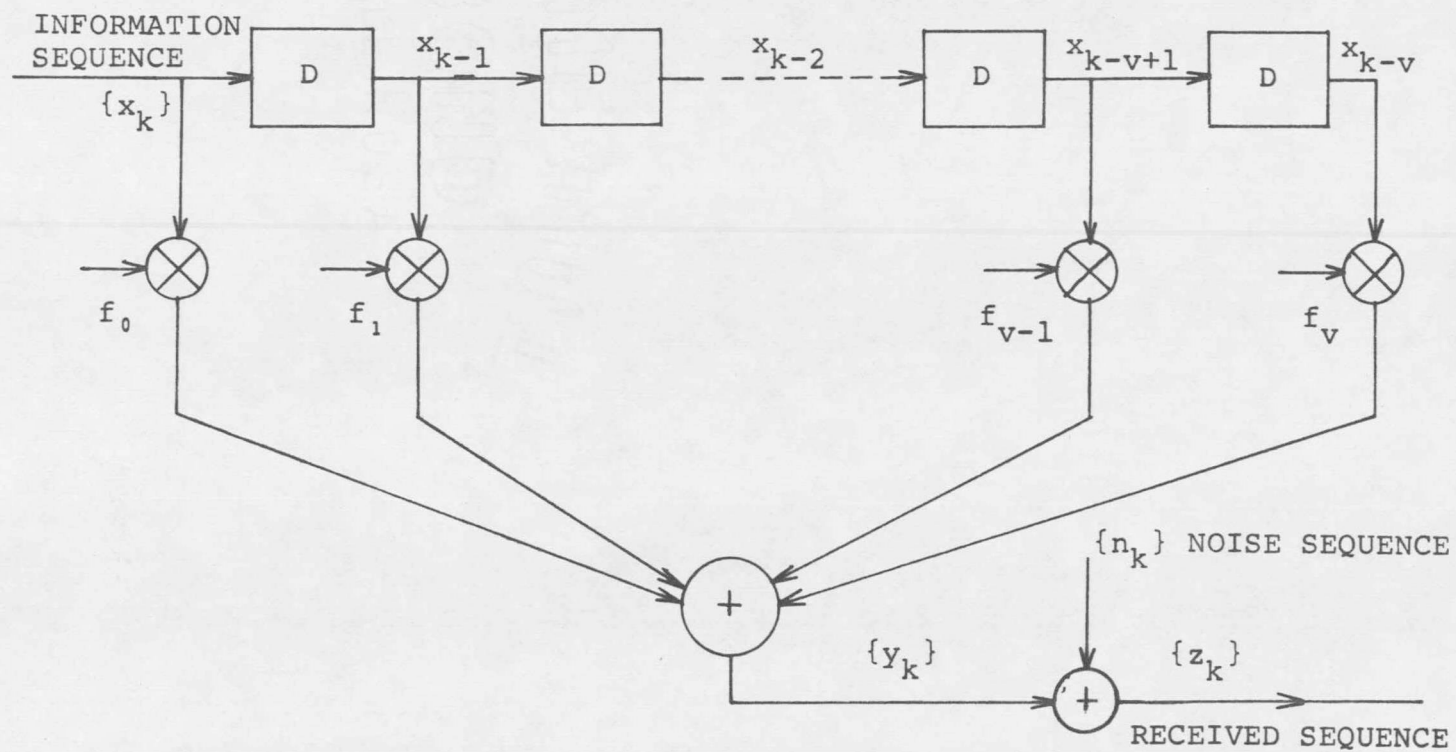


Figure 2.3 Block diagram of PAM communication system.

as the convolution of the input sequence $x(D)$ with the finite impulse response $f(D)$. Thus, the received sequence $z(D)$ is the sum of the signal sequence $y(D)$ and a white Gaussian noise sequence $n(D)$ with autocorrelation function σ^2 . It is obvious that the signal sequence $y(D)$ may be taken to be generated by a finite state machine driven by the input sequence $x(D)$. We may imagine a shift register of v m -state memory elements containing the v most recent inputs, with y_k formed as the weighted sum of the shift register contents and the current input x_k as shown in Figure 2.4. This model of Figure 2.4 will be used mostly in the thesis. Clearly part of the receiver, the whitened matched filter has been absorbed into the channel model.

So far nothing has been said about the impulse response $w(-t)$ of the whitened matched filter. Indeed, do the sampled outputs of the whitened matched filter form a set of sufficient statistics for estimation of the input sequence $x(D)$? Forney [22] has in fact shown that the sequence $z(D)$ is a set of sufficient statistics for estimating the input sequence. We sum Forney's results [22] into the following theorem.



35

Figure 2.4 Discrete-time channel model.

Theorem: Let $h(t)$ be finite with length L and let $R(D) = f(D)f(D^{-1})$. Then the filter whose chip D -transform¹ is $w(D,t) = \frac{h(D,t)}{f(D)}$ has square-integrable impulse response, and the sampled outputs of its time reverse ($w(-t)$)

$$z_k = \int_{-\infty}^{\infty} r(t)w(t-kT) dt \quad (2.19)$$

form a sequence

$$z(D) = x(D)f(D) + n(D)$$

in which $n(D)$ is a white Gaussian noise sequence with variance σ^2 and which is a set of sufficient statistics for estimation of the input sequence $x(D)$.

The discrete-time channel model discussed earlier is derived under the basic assumption that the channel is

¹The chip D -transform of $h(t)$ is a polynomial in D of degree v with coefficients in the set of functions $[0,T)$

$$h(D,t) = \sum_{i=0}^v h_i(t)D^i$$

where $h_i(t)$ is a function that is nonzero only over the interval $[0,T)$; i.e.,

$$h(t) = h_0(t) + h_1(t-T) + \dots + h_v(t-vT)$$

fixed with known impulse response. If the impulse response is now allowed to be slowly varying with time, the same modeling procedure can be applied but the matched filter or the whitened matched filter must vary to match the channel characteristics. Proakis and Magee [31] have assumed that the WMF is always matched to the channel characteristics, and therefore, the model of Figure 2.4 with time-varying tap gains can be used; it is entirely equivalent to the continuous-time model. The design of an adaptive processor to follow the WMF has been developed by Proakis and Magee [31].

In most of the cases, the whitened matched filter is fixed rather than being adaptive; in which case the noise at the output of the matched filter may become correlated. The effect of correlated noise has been analyzed by Kobayashi [29] and Quershi [32]. Their results indicate that even highly correlated noise does not cause significant loss in the performance of the maximum-likelihood sequence estimation receiver using a Viterbi decoder.

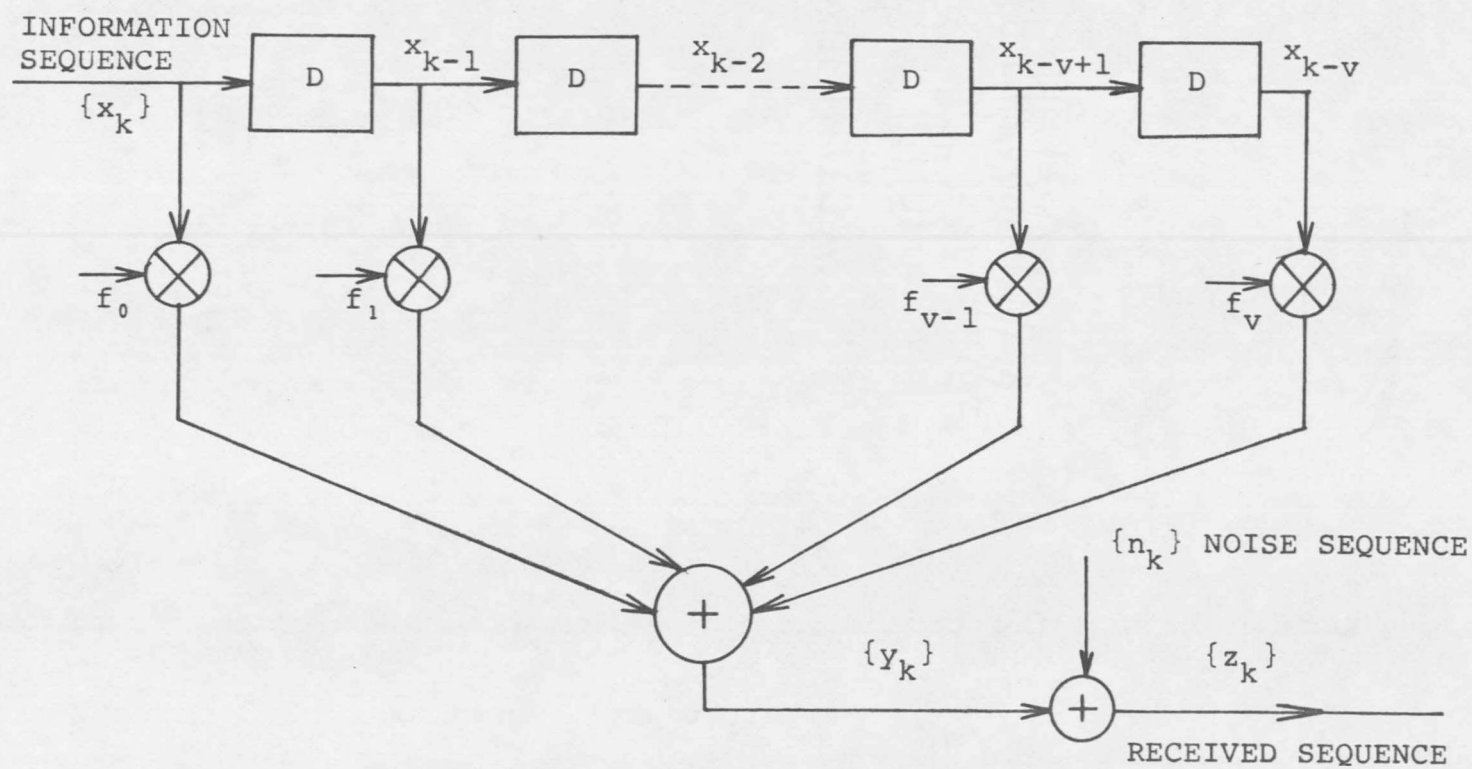
Chapter III

SEQUENCE ESTIMATION TECHNIQUES

3.1 Maximum-Likelihood Sequence Estimation

While lip service has long been paid to the idea that receiver decisions ought to be based on the entire received sequence, the fact that straightforward likelihood calculations grow exponentially with message length has justified a retreat to simple symbol-by-symbol decisions in most theoretical and practical work. As yet, no simplification of these results has been devised which results in a fairly simple receiver without seriously degrading the performance. Having failed to minimize the average number of errors simply, one can investigate minimizing the probability that the entire received sequence is not correct. This type of decision making is asymptotically optimum, which means that at moderate to high signal-to-noise ratios, it is effectively as good as the optimum compound detector [10]. The Viterbi algorithm is a simple efficient algorithm for this task.

The discrete-time channel model developed in Chapter II is shown again here in Figure 3.1. The inputs and outputs are now expressed in D-transform notation, where



39

Figure 3.1 Discrete-time channel model.

D is the delay operator and the unit of delay is the signalling interval T seconds. The output signal sequence

$$y(D) = x(D)f(D) \quad (3.1)$$

is the convolution of the input sequence $x(D)$ with the sampled response $f(D)$ of the discrete-time model, whose pulse autocorrelation function is $R(D) = f(D)f(D^{-1})$. The output sequence $y(D)$ is then corrupted by additive white Gaussian sequence $n(D)$ with autocorrelation function $\sigma^2 \delta_{ij}$ to form the received sequence $z(D)$. The channel consists of a shift register of v memory elements each having m (the size of input alphabet) states. These memory elements contain the v most recent inputs. The machine has m^v possible states given by the v most recent inputs

$$s_k = (x_{k-1}, x_{k-2}, x_{k-3}, \dots, x_{k-v}) \quad (3.2)$$

where by convention $x_k = 0$ for $k < 0$. We define the state sequence $s(D)$ as

$$s(D) = s_0 + s_1 D + s_2 D^2 + \dots \quad (3.3)$$

where each s_k takes on values from an alphabet of m^v states s_j , $1 \leq j \leq m^v$. The maps from input sequence $x(D)$ to state sequence $s(D)$ and thence to signal sequence are

obviously one to one and hence invertible. In fact, two successive states uniquely determine an output

$$y_k = y(s_k, s_{k+1}); \quad (3.4)$$

i.e., given a transition from s_k to s_{k+1} , the corresponding output y_k is determined.

An allowable state sequence $s(D)$ or signal sequence $y(D)$ is defined as one that could result from an allowable input sequence.

Definition: Maximum-likelihood sequence estimation is defined as the choice of that $x(D)$ for which the probability density $p(z(D)/x(D))$ is maximum. Since the maps from $x(D)$ to $s(D)$ and to $y(D)$ are one to one, maximum-likelihood sequence estimation can be defined equivalently as choosing from the allowable $s(D)$ that which maximizes $p(z(D)/s(D))$ or from the allowable $y(D)$ that which maximizes $p(z(D)/y(D))$. It will usually be convenient to consider the estimation of the state sequence of a finite state machine from noisy observations.

3.2 The Viterbi Algorithm

The Viterbi algorithm (VA) was proposed as a method of decoding convolutional codes [24]. Since that time it

has been recognized as an attractive solution to a variety of digital estimation problems, somewhat as the Kalman filter has been adopted to a variety of real-variable estimation problems. This algorithm was also later shown to be a shortest-route algorithm of a type long known in operations research, which in turn can be expressed as a variant of dynamic programming. Forney [22] has also shown that it is the maximum-likelihood decoding rule for the entire sequence in the presence of intersymbol interference.

The application of the Viterbi algorithm to maximum-likelihood sequence estimation of a digital sequence in the presence of intersymbol interference is now described. Since the noise is white and Gaussian, the noise terms n_k are independent, the log likelihood $\ln p(z(D)/s(D))$ reduces to a sum of independent increments.

$$\ln p(z(D)/s(D)) = \sum_k \ln p_n[z_k - y(s_k, s_{k+1})] \quad (3.5)$$

where $p_n(\cdot)$ is the probability density of each noise term n_k . For notational convenience we define the partial sums

$$\Gamma[s(D)]_{k_1}^{k_2} = \sum_{k=k_1}^{k_2-1} \ln p_n[z_k - y(s_k, s_{k+1})] \quad 0 \leq k_1 \leq k_2 \quad (3.6)$$

Suppose for the moment that we knew that the state s_k at time k was s_j . Then for any allowable state sequence $s(D)$ that starts with the initial state s_0 and passes the state s_j at time k , the log likelihood would expand into two independent parts:

$$\Gamma[s(D)]_0^K = \Gamma[s(D)]_0^k + \Gamma[s(D)]_k^K \quad (3.7)$$

If $\hat{s}_j(D)$ is the allowable state sequence from time 0 to k , which has the largest log-likelihood among all allowable state sequences starting with $s_0 = 0$ and ending with $s_k = s_j$, we call $\hat{s}_j(D)$ the survivor at time k corresponding to state s_j . This occurs because if $\hat{s}_j(D)$ is replaced by another sequence in the 0 to k interval, it will have a smaller log-likelihood and, therefore, can not be part of the maximum-likelihood sequence.

In fact, we do not know the state s_k at time k ; but we do know that it must be one of the m^V possible states at time k . Consequently, while we can not make a final decision as to the identity of the initial segment of the maximum-likelihood state sequence at time k , we know that the initial segment must be among the m^V survivors $\hat{s}_j(D)$, $1 \leq j \leq m$, one for each state s_j . Thus, at

each time k , the paths ending at every possible state are compared and the one with the largest likelihood ending in each state is retained as a survivor. Thus, we need only store m^V sequences $\hat{s}_j(D)$ and their log likelihoods $\Gamma[\hat{s}_j(D)]_0^k$, regardless of how large k becomes. To update the memory at time $k+1$, recursion proceeds as follows:

- (1) Compute the partial sum

$$\Gamma[s_j(D)]_0^{k+1} = \Gamma[\hat{s}_j(D)]_0^k + \ln[p_n(z_k - y(\hat{s}_j, s_j'))] \quad (3.8)$$

for each of the m allowable continuations $s_j'(D)$ to time $k+1$ for each of the m^V survivors $s_j(D)$ at time k . This involves $m^{V+1} = m^L$ additions.

- (2) Compare the log likelihood of the m continuations terminating in each of the m^V possible states and select the largest as the survivor.

Using the above procedure, the maximum-likelihood sequence (in the sense of the entire received sequence) is never eliminated but remains along the m^V survivors. The VA can not make a decision until all the surviving sequence at time k have the same initial state sequence up to time $k-\tau$. The decoding delay τ is unbounded but finite with probability 1. In practice, one actually makes a final

decision after some fixed delay δ , with δ chosen large enough so that the degradation due to premature decisions is negligible.

When the noise sequence is white and Gaussian with variance σ^2 ,

$$\ln(p_n(z_k - y_k)) = -\frac{1}{2} \ln(2\pi\sigma^2) - \frac{(z_k - y_k)^2}{2\sigma^2} \quad (3.9)$$

So the metric computation becomes

$$\left[\hat{\Gamma} - \Gamma \right]_{k_1}^{k_1+n+1} = \frac{1}{2\sigma^2} \left[\sum_{k=k_1}^{k_1+n} (z_k - y_k)^2 - (z_k - \hat{y}_k)^2 \right] \quad (3.10)$$

It is obvious from above that the decisions do not depend on the noise variance so that only the possible values for $y(D)$ are needed to compute the metric. If the channel impulse response is available, it is simple to compute the possible values of $y(D)$ since the possible state values are known.

3.3 Performance of the Maximum-Likelihood Sequence Estimation Receiver

Forney's [22] approach for calculating the performance of a maximum-likelihood sequence estimation receiver is summarized here. Before starting the analysis,

the following definitions are important.

Error Event: An error event ξ is said to extend from time k_1 to k_2 if the estimated state sequence $\hat{s}(D)$ is equal to the correct state sequence $s(D)$ at times k_1 and k_2 but nowhere in between ($s_{k_1} = \hat{s}_{k_1}$; $s_{k_2} = \hat{s}_{k_2}$; $s_k \neq \hat{s}_k$, $k_1 < k < k_2$). The length of the error event is defined as $n \triangleq k_2 - k_1 - 1$.

Euclidean Weight ($d^2(\xi)$)
of an Error Event:

The Euclidean weight $d^2(\xi)$ of an error event is defined as the energy in the associated signal error sequence.

Hamming Weight of the
Error Event:

The number of errors in the associated input error sequence is defined as the Hamming weight $W_H(\xi)$ of the event.

We define

$$\epsilon_x(D) \triangleq (x_{k_1} - \hat{x}_{k_1}) + (x_{k_1+1} - \hat{x}_{k_1+1})D + \dots + (x_{k_2-v+1} - \hat{x}_{k_2-v+1})D^{n-v} \quad (3.11)$$

as the input error sequence associated with the error event. $\epsilon_x(D)$ is a polynomial of degree $n-v$ in D and with

nonzero constant term ϵ_{x_0} . Since the channel is linear, the input error sequence maps simply to the output

$$\epsilon_Y(D) = \epsilon_X(D)f(D) \quad (3.12)$$

Thus, $\epsilon_Y(D)$ is a polynomial with nonzero constant term and degree n . Also by definition it follows that

$$\begin{aligned} d^2(\xi) &\triangleq \|\epsilon_Y\|^2 = \sum_{i=0}^n \epsilon_{Yi}^2 \\ &= [\epsilon_Y(D) \epsilon_Y(D^{-1})]_0 \\ &= [\epsilon_X(D)f(D) \epsilon_X(D^{-1})f(D^{-1})]_0 \\ &= [\epsilon_X(D) R(D) \epsilon_X(D^{-1})]_0 \end{aligned} \quad (3.13)$$

where $[]_0$ means the constant term inside the brackets.

Probability of a Particular Error Event:

We now calculate the probability that a particular error event identified by a starting time k_1 and an associated input error sequence $\epsilon_X(D)$ of degree $n-v$ will actually occur. Three subevents must occur:

ξ_1 : At time k_1 we must have $s_{k_1} = \hat{s}_{k_1}$.

ξ_2 : Between k_1 and k_1+n-v the input sequence $x(D)$ must be such that $x(D) + \epsilon_X(D)$ is an allowable sequence $\hat{x}(D)$.

ξ_3 : The noise terms n_k , $k_1 \leq k \leq k_1+n$, must be such that over this segment $\hat{x}(D)$ has greater likelihood than $x(D)$.

The subevent ξ_1 is possibly dependent on the noise terms involved in ξ_3 and the joint probability is not easily calculable. However, the probability that ξ_1 is not true is of the order of the error probability, so that in the normal operating region $p(\xi_1/\xi_3)$ is closely approximated as well as overbounded by 1. Therefore, the probability of the particular error event ξ is given by

$$\begin{aligned} \Pr(\xi) &= \Pr(\xi_3)\Pr(\xi_2)\Pr(\xi_1/\xi_3) \\ &\leq Q(d(\xi)/2\sigma) \left[\prod_{i=0}^{n-v} \frac{m - |e_{xi}|}{m} \right] \end{aligned} \quad (3.14a)$$

where σ^2 is the variance of the noise, and

$$\Pr(\xi_3) = Q[d(\xi)/2\sigma] \quad (3.14b)$$

$$\Pr(\xi_2) = \prod_{i=0}^{n-v} \frac{m - |e_{xi}|}{m} \quad (3.14c)$$

If E is the set of all possible error events, then the probability $\Pr(E)$ of at least one error event is upper-bounded by

$$\Pr(E) \leq \sum_{\xi \in E} \Pr(\xi) \quad (3.15)$$

The symbol error rate $\text{Pr}(e)$ is obtained by weighting each error event with the number of symbols which it represents. $\text{Pr}(e)$ contains an exponential factor in $d(\xi)$, where $d^2(\xi)$ is the energy associated with the error sequence at the channel output. Because of the presence of the exponential term, a good approximation, at error rates below about 10^{-3} , is to consider only error events associated with the minimum $d(\xi)$ --called $d_{\min}$ --for any nonzero sequence. Therefore, the symbol error rate is approximated by

$$\text{Pr}(e) \approx K_2 Q(d_{\min}/2\sigma) \quad (3.16)$$

where

$$K_2 = \sum_{\xi \in E_{d_{\min}}} [W_H(\xi) \prod_{i=0}^{n-v} \frac{m - |\epsilon_{xi}|}{m}]$$

is the factor which takes account of all error sequence with $d(\xi) = d_{\min}$.

$W_H(\xi)$ = Hamming weight of the error sequence $\epsilon_x(D)$.

$$Q(x) = (2\pi)^{-1/2} \int_x^{\infty} e^{-y^2/2} \cdot dy$$

If the channel is known, the lower bound on performance for any receiver structure can be computed by

$$\Pr(e) \geq K_0 Q(d_{\min}/2\sigma) \quad (3.17)$$

where K_0 is just the probability that the original input sequence $x(D)$ plus the error sequence $\varepsilon_x(D)$ is an allowable input sequence. A slightly tighter version of the lower bound is given by

$$\Pr(e) \geq K_3 Q(d_{\min}/2\sigma) \quad (3.18)$$

where $K_3 = \frac{2(m-1)}{m}$.

Thus, the upper bound is just a constant multiple ($\frac{k_2}{k_0}$ of $\frac{k_2}{k_3}$) times the lower bound. This explicitly demonstrates the asymptotically optimum performance of the Viterbi algorithm. In Figure 3.2, we plot the predicted symbol-error probability as a function of output signal-to noise ratio¹ with $m=2$ and $m=4$ for the three cases:

1) no intersymbol interference; 2) a partial response of the class $f(D)=1 \pm D^n$ with precoding² and maximum-likelihood sequence estimation; and 3) the same partial response with

¹The output signal-to-noise ratio is defined to be [22]. $SNR \triangleq \frac{\sigma_y^2}{\sigma^2} = \frac{\sigma_x^2 \|f\|^2}{\sigma^2}$ where $\sigma_x^2 = \frac{m^2-1}{12}$ and $\|f\|^2 = \sum_{i=0}^v f_i^2$

²Precoding is a technique that has been used in partial response systems to prevent infinite error propagation in recovery of the transmitted sequence [41].

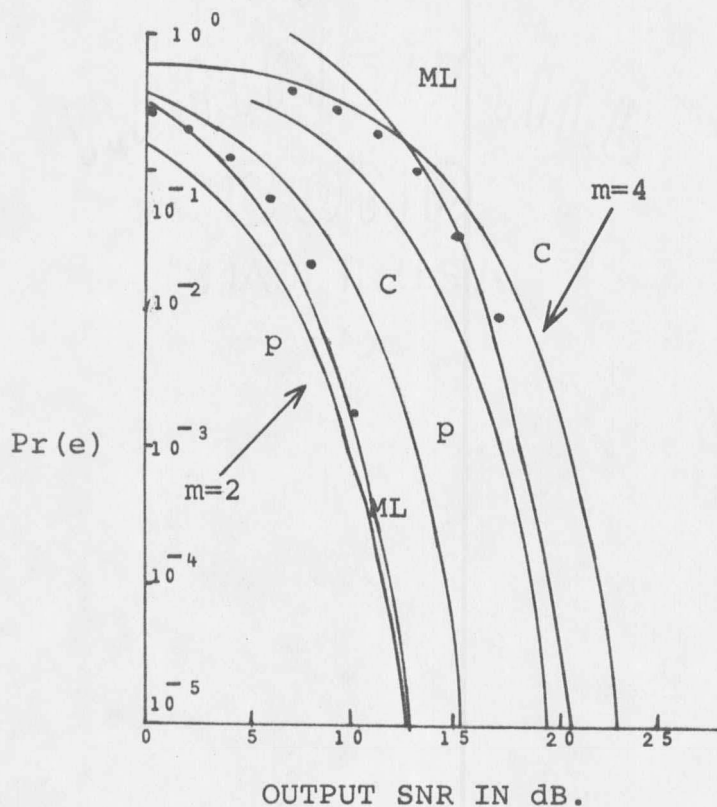


Figure 3.2 Probability of error versus output signal-to-noise ratio for $m=2$ and $m=4$. p-perfect response; ML-partial response $f(D)=1 \pm D^n$ with maximum-likelihood sequence estimation and precoding; C-partial response $f(D)=1-D$ with conventional sample-by-sample decisions and precoding; black dots - results of 10,000 sample simulations of ML (Taken from Forney [22]).

precoding and with symbol-by-symbol decisions. Simulation results (10,000 symbols) for the maximum-likelihood sequence estimator at low signal-to-noise ratios are also given.

Forney [22] has derived these results under the assumption that channel is known exactly. Consequently, for unknown channels these results have to be modified. It would be desirable to know the worst possible performance which can be expected for any channel. Proakis and Magee [31] have derived an expression for an upper bound on the probability of error which depends upon the length of the channel impulse response. This approach basically consists of deriving minimum distance (worst) codes under a fixed energy and impulse response length constraint. Also computations [31] have shown that the error sequence for the minimum distance channel is always of the form 1-D. Proakis and Magee [34] have found minimum distance channels for different lengths and performance loss in dB, for all these channels. The actual nonunique minimum-distance channels for lengths up to 10 are listed in Table 2.

In order to complete the upperbounding for the impulse responses, the constant K_2 in (3.16) must be

Table 2. Performance Loss and Corresponding Nonunique Minimum-Distance Impulse Response

Channel length	Performance loss (dB) $-10 \log_{10} d_{\min}^2$	Minimum-distance channel
1	0	1
2	0	(a,b) where $a^2 + b^2 = 1$
3	2.3	(0.50,0.71,0.50)
4	4.2	(0.38,0.60,0.60,0.38)
5	5.7	(0.29,0.50,0.58,0.50,0.29)
6	7.0	(0.23,0.42,0.52,0.52,0.42, 0.23)
7	8.2	(0.19,0.35,0.46,0.50,0.46, 0.35,0.19)
8	9.2	(0.16,0.30,0.41,0.46,0.46, 0.41,0.30,0.16)
9	10.1	(0.14,0.26,0.36,0.43,0.45, 0.43,0.36,0.26,0.14)
10	10.9	(0.12,0.23,0.32,0.39,0.42, 0.42,0.39,0.32,0.23,0.12)

evaluated. This is a simple matter because the worst possible condition is a channel which can have a continuous string of errors without increasing the minimum error distance. This can be evaluated by actually computing the channels which correspond to this case. These are the so-called maximal distortion channels of Austin [8]. The constant K_2 for these channels and thus for all possible channels is upperbounded by

$$K_2 \leq \sum_{n=1}^{\infty} (2n) \left(\frac{m-1}{m}\right)^n = 2m(m-1) \quad (3.19)$$

3.4 Various Types of Adaptive Maximum-Likelihood Sequence Estimation Receivers and Their Performance

The maximum-likelihood sequence receiver structure was initially formulated under the assumption that the channel response $c(t)$ is known. It is often the case in the real world, the channel response is not known to us. In this case, the receiver has to be adaptive to the unknown channel. In theory the WMF (whitened matched filter) must also adapt to the channel impulse response as well as the Viterbi decoder. If the WMF is fixed rather than being adaptive, the noise at the output of WMF becomes correlated. The effect of correlated noise has

been partially analyzed by Kobayashi [29] and Quershi [32]. These results indicate that even highly correlated noise does not cause significant loss in the performance of the Viterbi decoder. Recently, quite a few adaptive MLSE receiver structures have been proposed, and some of these structures are described briefly now.

3.4.1 An Adaptive Maximum-Likelihood Sequence Estimation Receiver Using a Channel Estimator

This receiver structure has been suggested by Magee and Proakis [31]. Figure 3.3 shows the adaptive receiver to be considered. This receiver structure assumes that a whitened matched filter precedes the sampler and thus only the sequence $\{z_k\}$ of sampled values is considered in the analysis. Since the Viterbi algorithm requires a knowledge of the channel impulse response to function, a channel estimator is provided which operates on the information sequence estimate and received signal sequence to provide a measurement of the channel response. The channel estimator used is shown in Figure 3.4.

The estimator approximates the actual channel with a finite state linear machine, that is, in exactly the same manner as the channel model shown before in

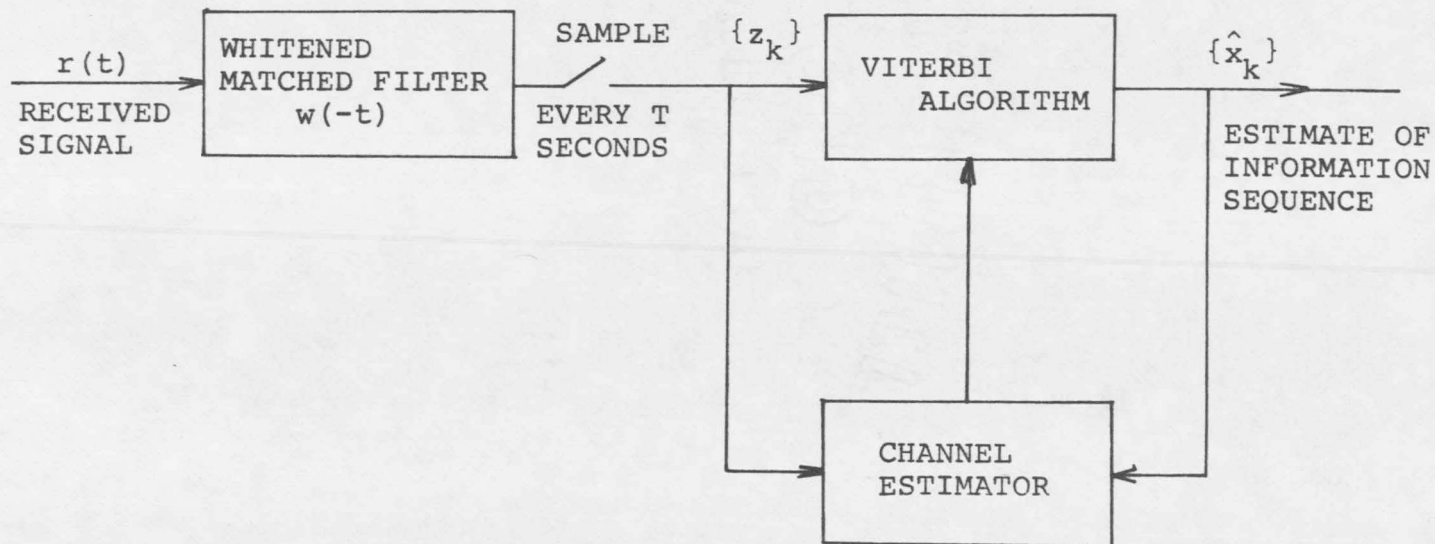


Figure 3.3 An adaptive maximum-likelihood receiver.

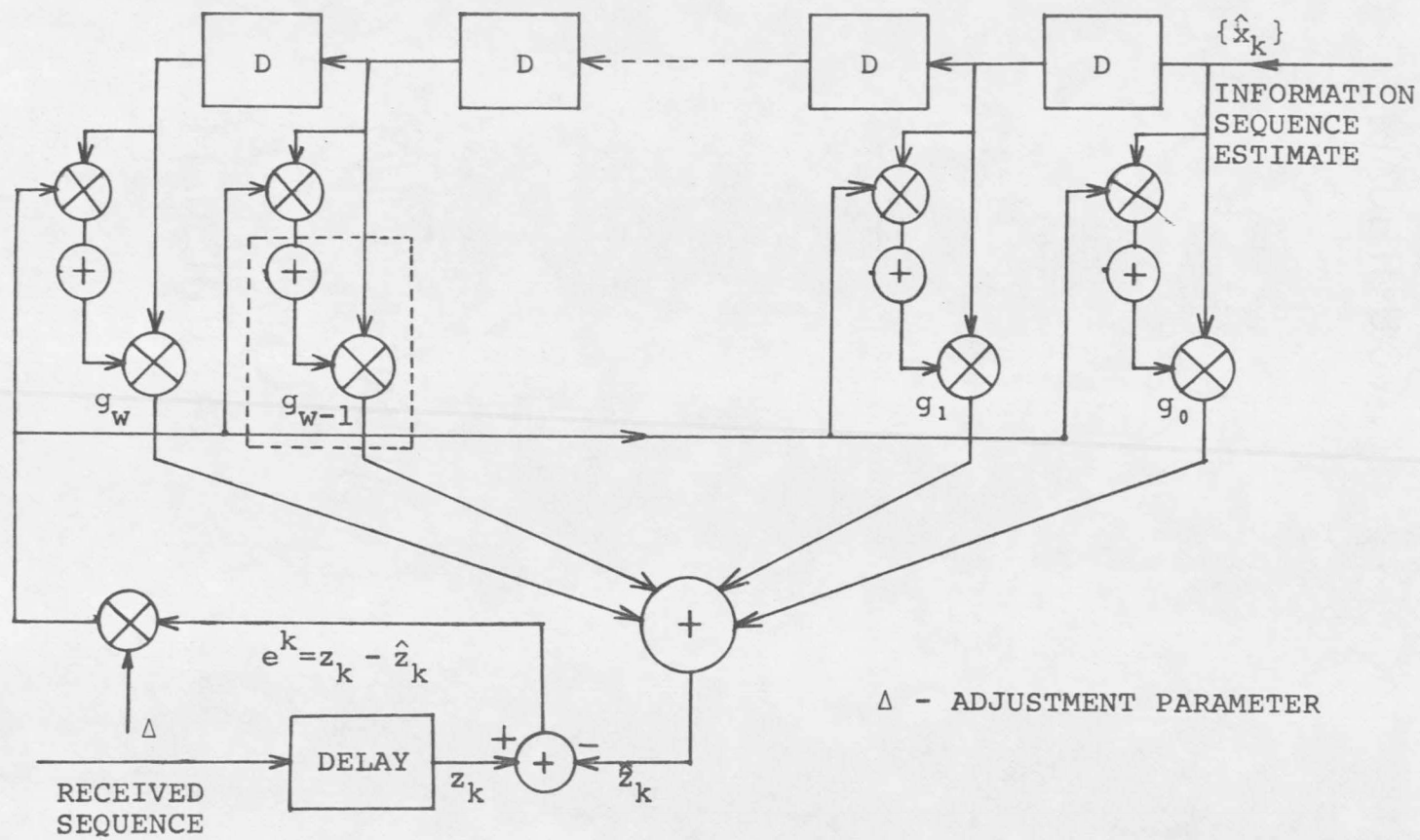


Figure 3.4 Channel estimator.

Figure 3.1. The estimated channel path gains $\{g_i\}$ are adjusted by the steepest descent algorithm to minimize the mean-square error (MSE) between the actual received sequence and a received sequence estimate formed from the output of the finite state machine model. Δ is the adjustment parameter and the rate of convergence and accuracy of the estimate depends upon this parameter. A delay, the same as the VA decision delay, is required to properly time the comparison between the received sequence and the received sequence estimate. The operation of the channel estimator is discussed in detail later.

It will now be shown that when the MSE between z_k and $\hat{z}_k$ is minimized, the resulting values for the tap gains of the channel estimator are the values of the sampled channel impulse response. For mathematical tractability, it is assumed that the estimated information sequence $\{\hat{x}_k\}$ is correct; i.e., $\{\hat{x}_k\}$ is identical to the transmitted sequence $\{x_k\}$. This is a reasonable assumption for a receiver operating at a low probability of error. The MSE between the received sample z_k and the estimated sample $\hat{z}_k$ is

$$\epsilon = E(z_k - \hat{z}_k)^2 \quad (3.20)$$

where

$$z_k = n_k + \sum_{j=0}^v f_j x_{k-j} \quad (3.21)$$

$$\hat{z}_k = \sum_{j=0}^w g_j x_{k-j}$$

It is easily shown (Appendix A) that the tap gains that minimize the MSE in (3.20) must satisfy the set of $w + 1$ linear equations

$$\sum_{k=0}^w g_k \phi_{jk} = B_j \quad j = 0, 1, \dots, w \quad (3.22)$$

where

$$\phi_{jk} = E(x_j x_k) \quad (3.23)$$

$$B_j = \sum_{k=0}^v f_k \phi_{jk}$$

In matrix notation, (3.22) becomes

$$\Phi G = B \quad (3.24)$$

where Φ is a $(w+1) \times (w+1)$ covariance matrix of the information sequence x_k , and B is a column vector of the $(w+1)$ cross-correlations $B_i = E(x_i z_k)$. The solution to

this set of equations is expressed in matrix form as

$$G_{\text{opt}} = \Phi^{-1} B \quad (3.25)$$

where G_{opt} is the optimum set of tap gains in the sense of minimum MSE. The minimum MSE is

$$\begin{aligned} \text{MSE}_{\text{min}} &= E[(z_k - \hat{z}_k) z_k] \\ &= E(z_k \cdot z_k) - B^t G_{\text{opt}} \end{aligned} \quad (3.26)$$

From above equations, it is obvious that as long as the information sequence $\{x_k\}$ is uncorrelated, the optimum tap gains for the channel estimator are exactly equal to the respective values of the channel impulse responses. It is also apparent that when the length w of the channel estimator is greater than or equal to the length v of the channel impulse response, the optimum tap gains are equal to the respective values of the channel impulse response even when the information sequence is correlated.

The obvious way to solve (3.24) is to invert the covariance matrix Φ ; but B is not known, and frequently the statistics of the information sequence may also be unknown. So this is not a practical solution to the

problem of finding G_{opt} . An alternative technique used here is the steepest descent algorithm for finding the minimum of a function. This algorithm also helps in making the receiver self adjusting. The algorithm becomes in this case

$$g_j^{k+1} = g_j^k + \Delta(e)^k \hat{x}_{k-j} \quad j = 0, 1, \dots, w \quad (3.27)$$

where $e^k \hat{x}_{k-j}$ is the product between the error at the k th iteration and the information symbol in the j th tap at the k th iteration. Each iteration corresponds to the entrance of new data into the channel estimator.

The operation of the channel estimator is now explained. Every T seconds a new data point z_k and a new information symbol $\hat{x}_k$ are fed into the channel estimator. An estimate of the received data $\hat{z}_k$ is formed by multiplying the sequence of symbols $\hat{x}_j$ stored in the tapped delay line by the corresponding tap gains g_j and summing the $(w+1)$ products. An error symbol e^k is formed by taking the difference between the actual data sample z_k and the estimate $\hat{z}_k$. This error, scaled by a factor Δ , is then multiplied by the symbol $\hat{x}_j$ at the appropriate tap. The resulting products are added to the corresponding values

of the previous tap gains to form a new set of tap gains. The dotted section of Figure 3.4 does two operations. Firstly, the j th tap gain is incremented by the amount which is $e^k \Delta \cdot x_{k-j}$. Secondly, the incremented tap gain multiplies the stored symbol and the resulting product is then added together with the products obtained from the other tap gains to form an estimate of the received sample. In the receiver structure of Figure 3.3, these new taps are fed to the Viterbi decoder for the next computations. The convergence of this channel estimator depends upon the adjustment parameter Δ . The behavior and choice of the adjustment parameter Δ is discussed in more details in Chapter IV.

The above receiver makes use of the estimated information sequence at the output of the Viterbi decoder in making the adjustments. One may also send a short training sequence to perform the initial adjustment of the tap gains. Magee [31] has shown that this training sequence is not required in binary signalling, but it is expected that multilevel signalling would require a training sequence during start up operation for most dispersive channels.

3.4.2 Performance of the Adaptive MLSE Receiver Using Channel Estimator

The performance of this type of adaptive receiver structure has been derived by Magee [31] in combination with Forney's work [22]. When the information sequence consists of 1's and -1's sent with equal probability, the upper bound on the symbol error probability can be expressed as

$$\Pr(e) \leq K \operatorname{erfc}(d_{\min}/2\sigma_N) \quad (3.28)$$

where K and $d_{\min}$ are constants dependent upon the error structure of the channel and σ_N is dependent on the noise variance and the errors in the channel estimate. The constant K and $d_{\min}$ are determined as described in section 3.3. The error signal at the output of the channel estimator is a result of the noise in the channel, a noise due to misadjustment of the tap gains in the channel adjustment and the error in the estimated information sequence. σ_N has been expressed by Magee [31] to equal

$$\sigma_N^2 = \sigma^2 + \epsilon_\Delta \quad (3.29)$$

where

$$\epsilon_\Delta = \Delta(w+1) \epsilon_{\min} \Phi_{00}/2 \quad (3.30)$$

is the noise variance which takes into account the noise due to the misadjustment in the tap gains and the error in the estimated sequence.

Δ is the adjustment parameter.

w is the length of the estimated channel.

$\epsilon_{\min}$ is the minimum MSE when the channel is known exactly and

$$\epsilon_{\min} = \sigma^2 + \epsilon_p(e) \quad (3.30)$$

The errors inherent in $\hat{x}_k$ form an addition to the error signal caused by the information sequence estimate. The mean-square value of this noise (denoted as $\epsilon_p(e)$) is approximately given by

$$\epsilon_p(e) = d_{\min}^2 K_1 \cdot \text{erfc}\left(\frac{d_{\min}}{\sigma}\right) \quad (3.31)$$

where $d_{\min}^2$ is the energy contained in the minimum weight error event and the remainder of the expression is the probability of that error event described in section 3.3.

It is obvious from (3.29) that there are two independent noise terms affecting the decisions being made: additive noise in the channel and channel estimation noise. These two terms are additive in the way in which they affect the metric computations in the Viterbi decoder. It

is also assumed in deriving (3.28) that total noise is Gaussian. This is approximately true because n_k is a Gaussian random variable and misadjustment noise is approximately Gaussian due to its being generated by channel noise and the information sequence estimate summed over many terms.

The calculations necessary to evaluate (3.28) were made by Magee and Proakis [31] for two channels shown in Figure 3.5. The results obtained by simulation for these two channels are shown in Figure 3.6. The lower bound corresponds to one derived by Forney [22]. The results are plotted with symbol error probability versus input signal-to-noise ratio. Input signal-to-noise ratio is defined to be [31].

$$\text{ISNR} = \frac{\sigma_x^2}{2\sigma^2} \quad \text{and} \quad \sigma_x^2 = \frac{m^2-1}{12} \quad (3.31)$$

It is quite clear from Figure 3.6 that this receiver structure provides a large improvement over the other nonlinear schemes (decision-feedback schemes in particular) and very nearly eliminates the effect of intersymbol interference.

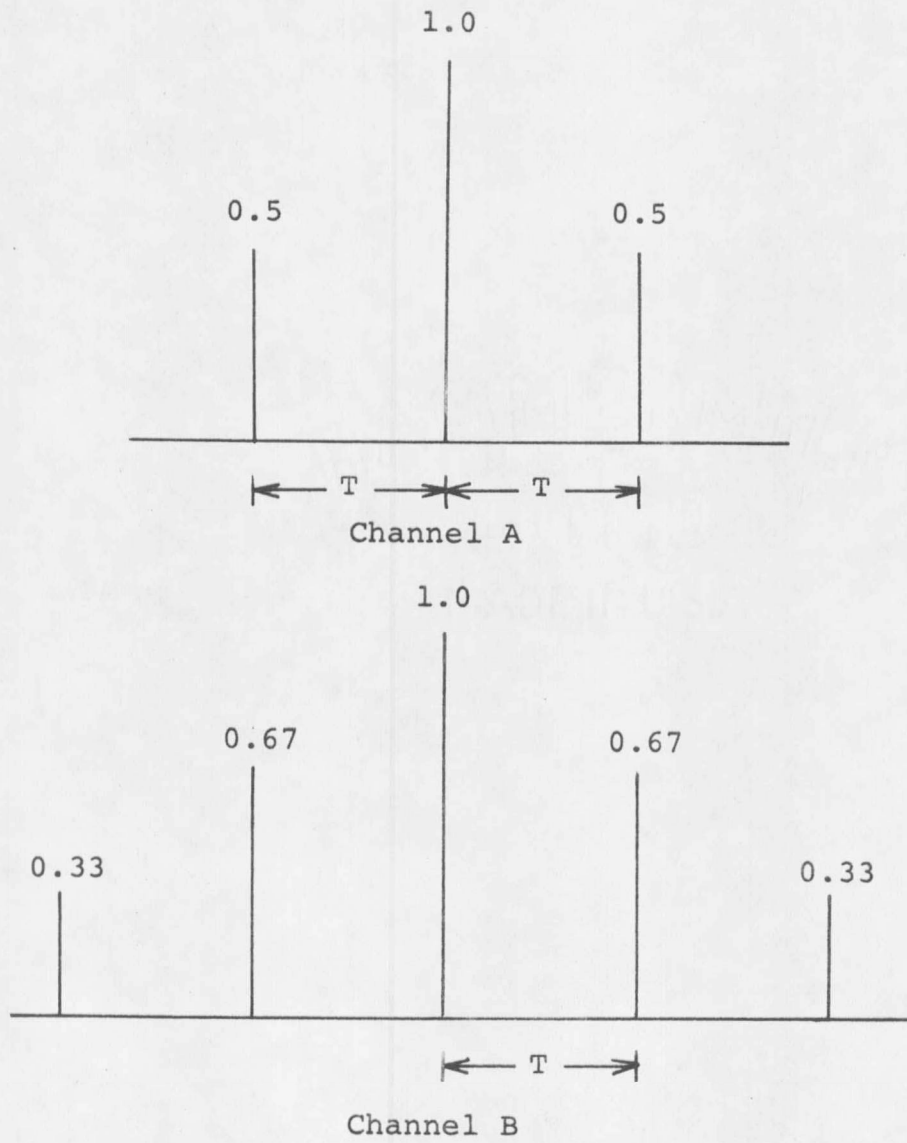


Figure 3.5 Samples of channel impulse response.

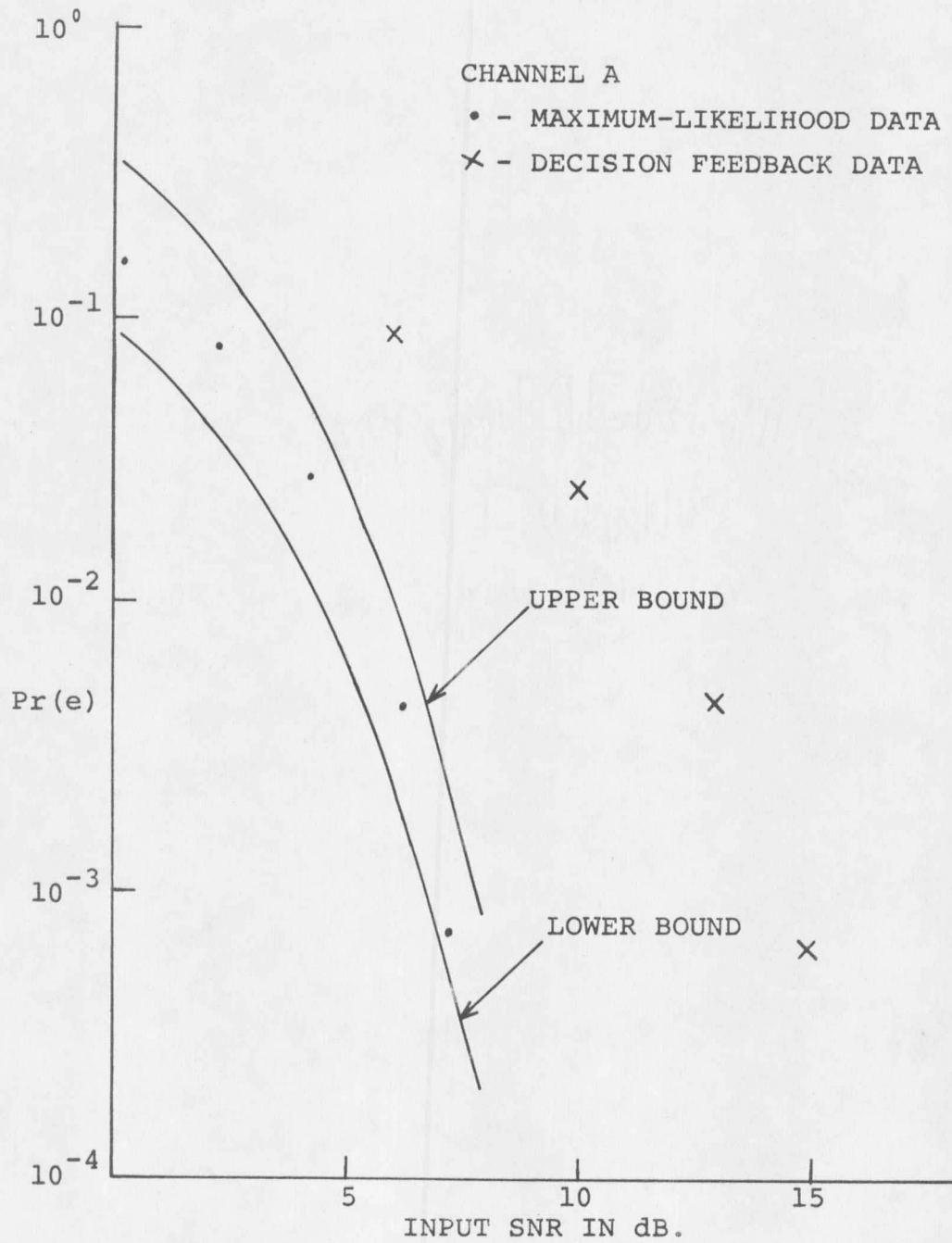


Figure 3.6(a) Pr(e) vs. input SNR in dB. (Taken from [31]).

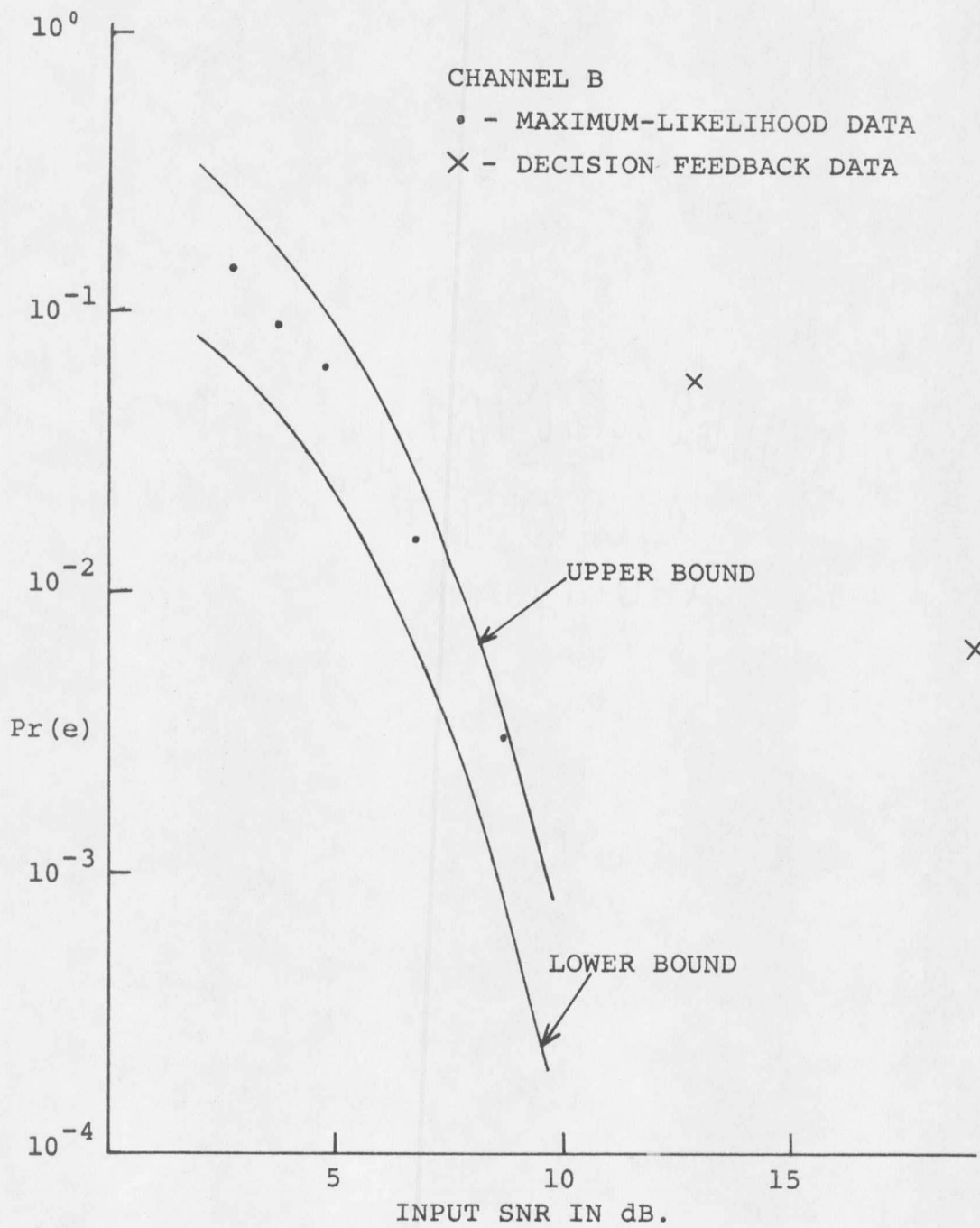
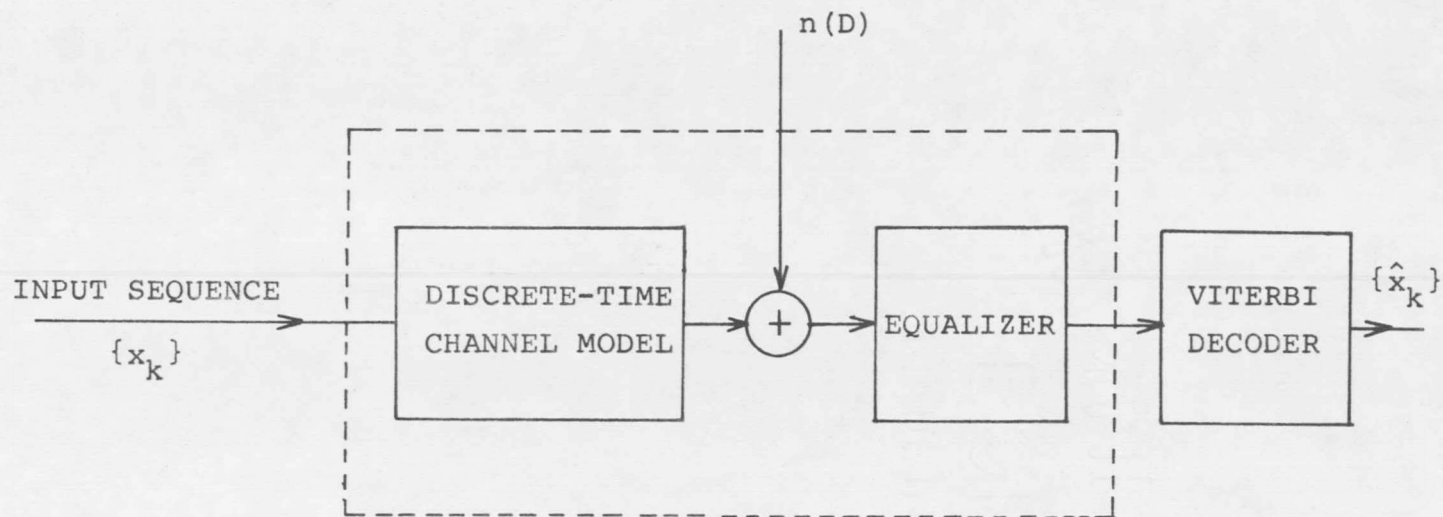


Figure 3.6(b) Pr(e) vs. input SNR in dB. (Taken from [31]).

3.4.3 An Adaptive Maximum-Likelihood Sequence Estimation Receiver Using a Linear Equalizer

This basic receiver structure was mentioned by Forney [22] in his conclusions as a potential research area. But Quershi and Newhall [32] have considered this suboptimum but simple and practical receiver in more detail.

The receiver consists of a linear adaptive equalizer in cascade with a maximum-likelihood sequence estimator. The purpose of the linear adaptive equalizer is to limit the time spread of the channel and mitigate time variation before maximum-likelihood sequence estimation. A block diagram of this type of receiver is shown in Figure 3.7. The adaptive equalizer helps in making the overall response of the dotted portion (see Figure 3.7) equal to the desired response. Since the complexity of the MLSE increases exponentially with L (length of the channel), the adaptive receiver structure proposed by Magee [31] becomes impractical for channels with large L . In such cases this type of receiver becomes a practical alternative. The proposed adaptive receiver is shown in Figure 3.8.



70

Figure 3.7 An adaptive receiver with equalizer.

In the present receiver structure, the matched filter is omitted, thus incurring some information loss. However, for channels that are strictly band limited to $1/2T$ Hz, such as partial response channels, a low pass filter to remove out-of-band noise and a symbol-rate sampler provide a set of sufficient statistics for estimating $\{x_k\}$. The operation of this receiver structure is now explained. The received sequence $\{z_k\}$ is first passed through a linear adaptive equalizer which limits the intersymbol interference introduced by the channel. The equalized sequence $\{q_k\}$ is fed to a Viterbi detector which produces an estimate $\{\hat{x}_k\}$ of the information sequence $\{x_k\}$ with a delay of D symbol periods. This estimate is the recovered data. A sequence $\{\hat{x}_k\}$ of tentative decisions chosen from the surviving path with the maximum log-likelihood is used for equalizer adjustment in the decision directed mode. This sequence, which has suffered a delay of $B < D$ units in the Viterbi decoder, is passed through a fixed non-recursive digital filter, the coefficients of which represent the desired pulse response $\{g_{dk}\}$ of the channel and the equalizer in cascade. The output of this digital filter is then the desired output

to produce the error sequence $\{e_k\}$. The tap gains of the adaptive equalizer are adjusted to minimize the mean-square error $E(e_k^2)$. Since the error sequence is delayed by B symbol periods, B additional storage units are needed in the equalizer delay line to adaptively adjust the B right-most tap gains.

The gist of the proposed approach is that the task of the reception of digital signals in the presence of intersymbol interference is shared by the equalizer and a Viterbi detector. This allows the MLSE receiver to be made of practical size with only a small degradation in performance. In addition, the equalizer performs better and needs fewer taps since it is not required to remove intersymbol interference completely. It is obvious that the choice of the desired channel-equalizer response is an important factor. As mentioned earlier, the complexity of the MLSE is proportional to m^L , practical considerations demand that the number of samples L in the desired impulse response be kept small (say, 2 or 3 for 8- or 4-level systems). Second, a linear equalizer can equalize valleys or nulls, in the transmission characteristics of the channel only at the cost of noise enhancement. Inter-symbol interference caused by such channel characteristics

can be better handled by the nonlinear MLSE. This suggests that the desired impulse response of the channel-equalizer combination should retain the gross channel characteristics. One strategy is to select L contiguous samples, including the reference sample, from the actual channel impulse response. This choice of the desired channel-equalizer pulse response decreases the mean-square error at the input and, consequently, at the output of the equalizer. For unknown channels, the equalizer can initially be used to find a least-mean-square error estimate of the discrete-time channel model.

3.4.4 Performance of an Adaptive MLSE Receiver Using a Linear Equalizer

The performance of this type of adaptive receiver has been evaluated by Quershi and Newhall [32] quite accurately. The noise sequence $\{n_k\}$ at the input to the Viterbi detector consists of 1) white Gaussian noise passing through the equalizer, 2) excess noise due to random fluctuations in the tap gains about their optimal (least mean-square error) values, and 3) any intersymbol interference other than the desired controlled value. The effect of this noise has been taken into account by

Quershi [32], and the upper bound for symbol error probability is

$$\Pr(e) \leq \sum_{\xi \in E} P_u(\xi) [W_H(\xi) \prod_{i=0}^{n-v} \frac{m - |\epsilon_{xi}|}{m}] \quad (3.32)$$

where $W_H(\xi)$ is the Hamming weight defined earlier and $P_u(\xi)$ is defined in [32]. The lower bound for probability of symbol error is taken to be the same as calculated by Forney [22].

The results obtained for binary transmission over a typical telephone channel are illustrated in Figure 3.9. The sampled impulse response of the channel is shown in Figure 3.10(a). The desired channel-equalizer impulse response is shown in Figure 3.10(b). It consists of three main impulses of the channel response. The frequency-domain responses corresponding to the sampled impulse response of Figure 3.10(a) and the desired impulse response of Figure 3.10(b) are illustrated in Figure 3.10(c). The performance of a 33-tap linear adaptive equalizer for the channel of Figure 3.10(a) is evaluated by computer simulation and is also presented in Figure 3.9 for comparison. Figure 3.9 also shows the results obtained by computer simulation.

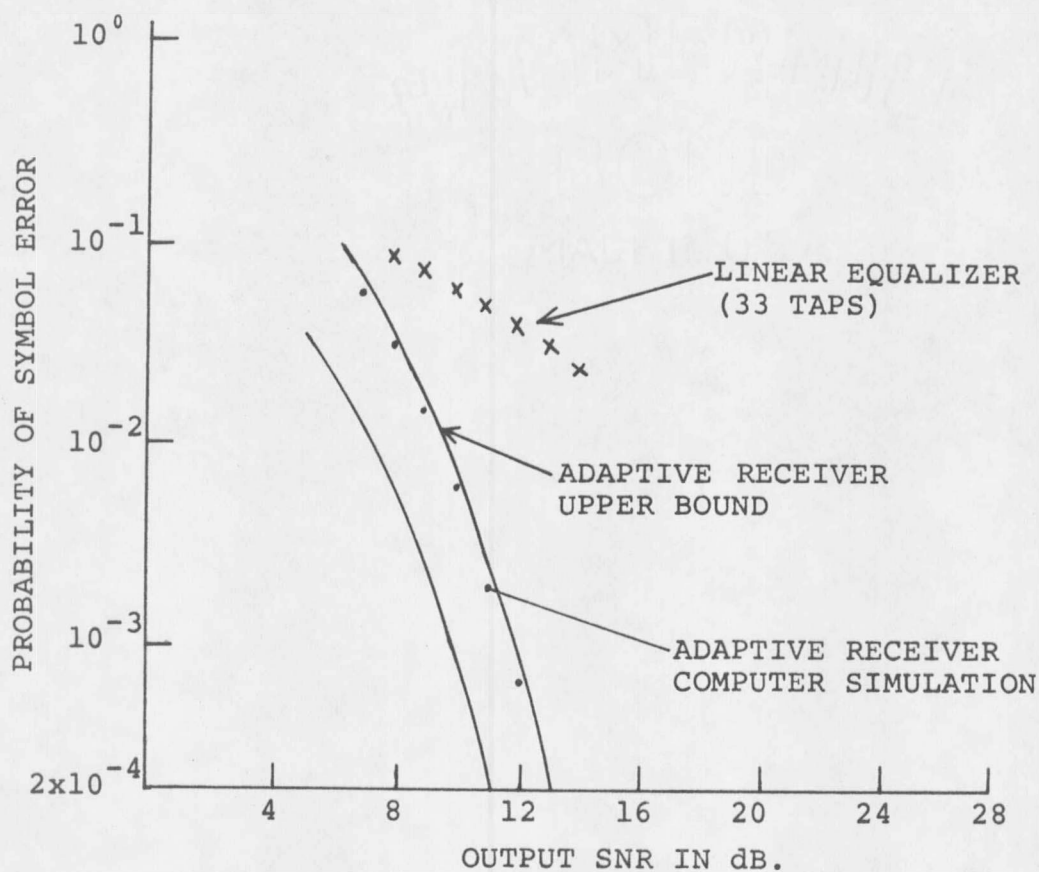


Figure 3.9 Average bit-error probability vs. output SNR
(Taken from Quershi and Newhall [32]).

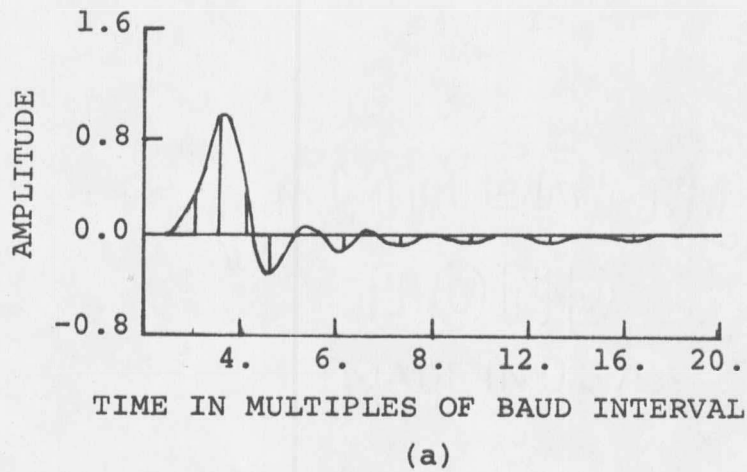
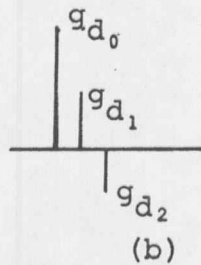
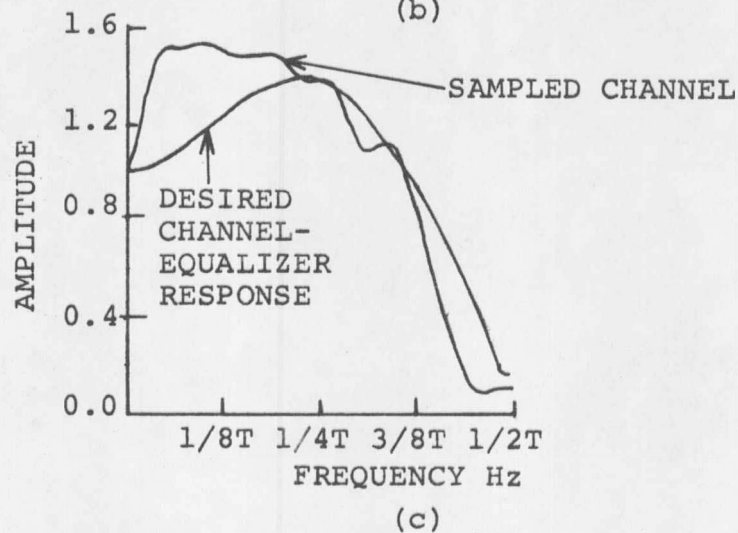


Figure 3.10 (a) Samples of channel response, (b) Desired channel-equalizer response, (c) Frequency-domain response.

3.4.5 Comparison of these Two Adaptive Receiver Structures

If the channel is highly dispersive, the receiver structure proposed by Magee [31] is not suitable because the complexity of the MLSE increases exponentially with L and consequently, this adaptive receiver structure becomes impractical. In such cases a receiver using a linear equalizer is less complex and more practical. For channels with small L , the receiver structure using a channel estimator is nearly as good as optimum MLSE receiver.

Chapter IV

DEVELOPMENT OF FASTER CONVERGING CHANNEL ESTIMATION ALGORITHMS

In Chapter III, some adaptive maximum-likelihood receivers were discussed. One of these receiver structures consists of a channel estimator to be used in conjunction with the Viterbi algorithm. The tap gains of the channel estimator are adjusted by minimizing the mean-square error between the received signal and the estimated signal. The algorithm used in the iterative process is the steepest descent algorithm. The adjustment parameter (or step size) Δ is kept the same in all iterations, i.e., Δ is fixed throughout the time during which the tap gains are adjusted for the channel estimator. This fixed-step-size algorithm often converges rather slowly, many iterations may be needed to achieve reasonable performance. Magee [31] also suggested that more-rapidly-converging algorithms for channel estimation should be developed. In this chapter, two algorithms for channel estimation are developed and are shown to be much faster in convergence than the fixed-step-size algorithm. Both these algorithms make use of a variable step size.

4.1 First-Order Algorithm

The so-called first-order algorithm employs a variable-step-size gradient search procedure for adjusting the tap gains of the channel estimator, and this algorithm guarantees improvement at the end of a fixed number of iterations. Therefore, this algorithm is suitable only in the training mode. In other words, the proposed algorithm makes use of a set of isolated training pulses called the test signal for the iterative adjustment of the parameters of the channel estimator.

4.1.1 Variable Step Size

Some of the equations described in Section 3.4 are reproduced here in order to derive the corresponding relations for a variable step size. The channel estimator is again shown in Figure 4.1. The tap gains of the channel estimator are adjusted so that the MSE between the received sample z_k and the estimated sample $\hat{z}_k$ is minimized.

$$\epsilon(\text{MSE}) = E(z_k - \hat{z}_k)^2 \quad (4.1)$$

where

$$z_k = n_k + \sum_{j=0}^V f_j x_{k-j}$$

$$\hat{z}_k = \sum_{j=0}^W g_j x_{k-j} \quad (4.2)$$

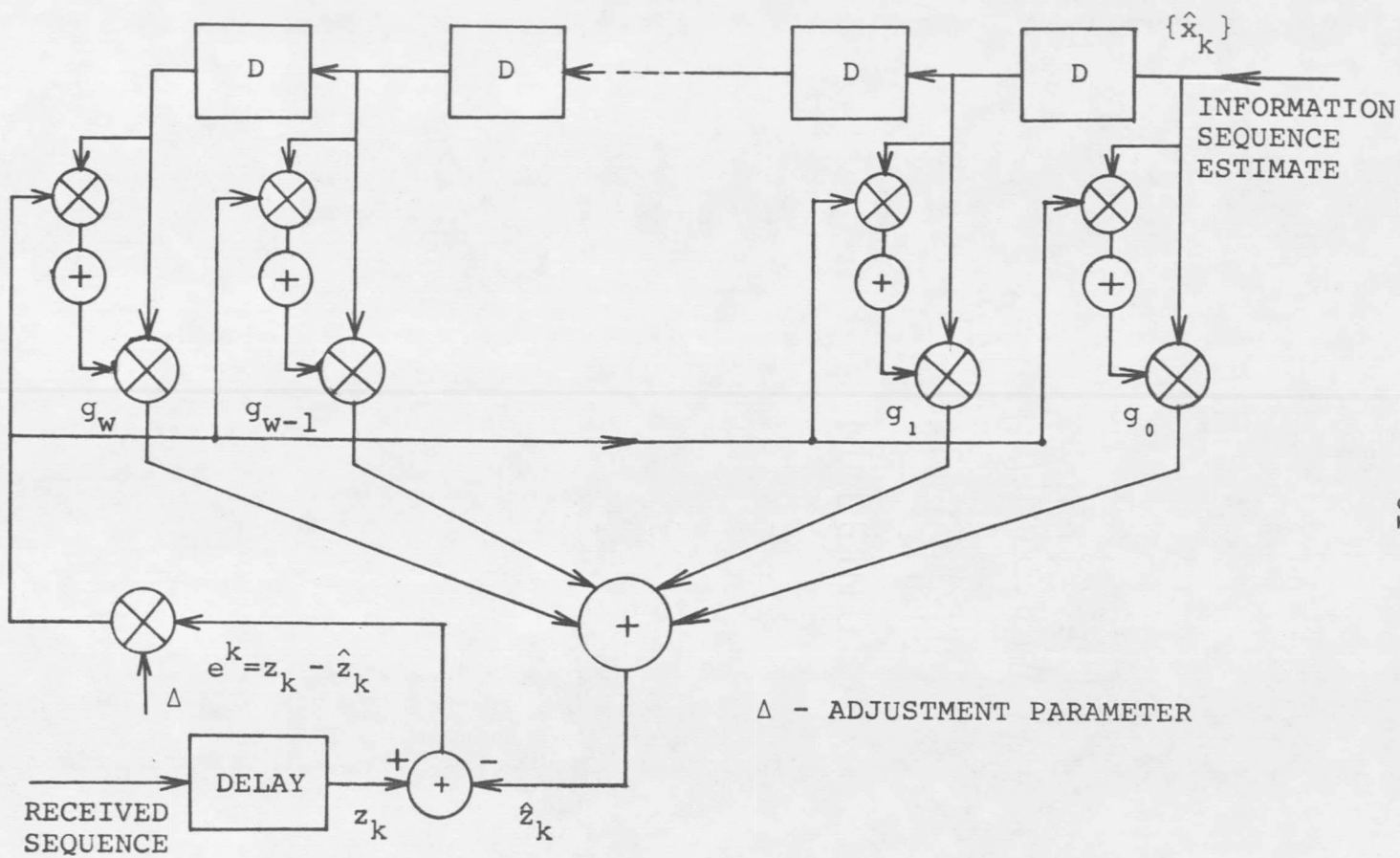


Figure 4.1 Channel estimator.

It is shown in Appendix A that the tap gains that minimize the MSE in (4.1) must satisfy the set of $(w+1)$ linear equations.

$$\sum_{k=0}^w g_k \Phi_{jk} = B_j \quad j = 0, 1, \dots, w \quad (4.3)$$

where

$$\begin{aligned} \Phi_{jk} &= E(x_j x_k) \\ B_j &= \sum_{k=0}^V f_k \Phi_{jk} \end{aligned} \quad (4.4)$$

In matrix notation, (4.3) becomes

$$\Phi G = B \quad (4.5)$$

where Φ is a $(w+1) \times (w+1)$ covariance matrix of the information sequence $\{x_k\}$, and B is a column vector of the $(w+1)$ cross-correlations $B_i = E(x_i z_k)$. The formal solution to this set of equations is expressed in matrix form as

$$G_{\text{opt}} = \Phi^{-1} B \quad (4.6)$$

where G'_{opt} is the row vector for the $(w+1)$ optimal tap gains in the sense of minimum MSE. Since it is not possible (explained earlier in Chapter III) to find the solution of (4.5) directly, gradient techniques are used. A

steepest descent algorithm used by Proakis and Magee [31] has the recursive relation

$$g_j^{k+1} = g_j^k - \frac{\Delta}{2} \left(\frac{\partial \epsilon}{\partial g_j} \right) \quad j = 0, 1, \dots, w \quad (4.7)$$

where $\frac{\partial \epsilon}{\partial g_j}$ is the gradient of the mean-square error (4.1).

It follows from Appendix A that

$$\frac{\partial \epsilon}{\partial g_j} = -2 e^k \hat{x}_{k-j} \quad (4.8)$$

Substituting (4.8) in (4.7), we get

$$g_j^{k+1} = g_j^k + \Delta e^k \hat{x}_{k-j} \quad (4.9)$$

Now in the variable-step-size approach, Δ is replaced by a variable step size, say α_k . Then (4.7) becomes

$$g_j^{k+1} = g_j^k - \frac{\alpha_k}{2} \left(\frac{\partial \epsilon}{\partial g_j} \right) \quad (4.10)$$

Let $\left(\frac{\partial \epsilon}{\partial \mathbf{G}} \right)^k$ represent the gradient row vector at the k th iteration. In vector notation equation (4.10) can be represented as

$$\mathbf{G}^{k+1} = \mathbf{G}^k - \frac{\alpha_k}{2} \left(\frac{\partial \epsilon}{\partial \mathbf{G}} \right)^k \quad (4.11)$$

Definition: The difference between the estimated tap gain and optimal tap gain is defined as the deviation in the estimated tap gain from the optimal tap gain.

$$b_j^k = g_j^k - g_{jopt} \quad j = 0, 1, \dots, w \quad (4.12)$$

Where b_j^k is the deviation in the j th estimated tap gain from the optimal tap gain at the k th iteration. In vector notation equation (4.12) becomes

$$E^k = G^k - G_{opt} \quad (4.13)$$

where E^k , G^k , G_{opt} are all vectors. It follows from equation (4.11) that

$$G^{k+1} - G^k = - \frac{\alpha_k}{2} \left(\frac{\partial \epsilon}{\partial G} \right) \quad (4.14)$$

$$E^{k+1} = G^{k+1} - G_{opt} \quad (4.15)$$

$$E^k = G^k - G_{opt} \quad (4.16)$$

Subtracting (4.16) from (4.15) gives

$$E^{k+1} - E^k = G^{k+1} - G^k \quad (4.17)$$

Substituting (4.17) in (4.14) results into

$$E^{k+1} - E^k = - \frac{\alpha_k}{2} \left(\frac{\partial \epsilon}{\partial G} \right) \quad (4.18)$$

It follows from Appendix A that

$$\frac{\partial \varepsilon}{\partial G} = (-2) (B - \Phi G^k) \quad (4.19)$$

Combining (4.19) and (4.18) yields

$$\begin{aligned} E^{k+1} - E^k &= \left(\frac{-\alpha_k}{2} \right) (-2) (B - \Phi G^k) \\ E^{k+1} &= E^k + \alpha_k (B - \Phi G^k) \end{aligned} \quad (4.20)$$

$$\begin{aligned} &= E^k - \alpha_k \Phi G + \alpha_k B \\ &= E^k - \alpha_k \Phi (G^k - \Phi^{-1} B) \\ &= E^k - \alpha_k \Phi E^k \\ &= (I - \alpha_k \Phi) E^k \end{aligned} \quad (4.21)$$

The solution of (4.21) at the Mth step is

$$E^M = P_M(\Phi) \cdot E^0 \quad (4.22)$$

where

$$P_M(\Phi) = \prod_{k=0}^{M-1} (I - \alpha_k \Phi) \quad (4.23)$$

= polynomial of degree M in Φ .

The best fixed-step-size algorithm suggested by Proakis and Magee [31] is optimum locally in the sense that the

optimum step size to go from one iteration to the next is given by

$$\Delta = 2/\lambda_{\max} \quad (4.24)$$

where $\lambda_{\max}$ is the largest eigenvalue of the matrix Φ . However, we shall consider the error vector only at the Mth iteration and minimize the norm of the tap-gain error vector $\|E^M\|$ by finding the optimum set of step sizes $\{\alpha_i\}_0^{M-1}$ to be used in M iterations. The local behavior of the error norm for the M-1 intermediate iterations will be disregarded. From (4.22) it follows that

$$\|E^M\| = \|P_M(\Phi) \cdot E^0\| \leq \|P_M(\Phi)\| \cdot \|E^0\| \quad (4.25)$$

We shall consider finding the set $\{\alpha_i\}_0^{M-1}$ which minimizes an upper bound to the norm of $P_M(\Phi)$. Thus, the minimum will be independent of any given initial setting.

The norm of $P_M(\Phi)$, as defined by Varga [37], is the largest eigenvalue of $P_M(\Phi)$ in magnitude. Since $P_M(\Phi)$ is a polynomial in Φ , the eigenvalues of $P_M(\Phi)$ are the values of the polynomial $P_M(x)$ when evaluated at the eigenvalues of Φ . Therefore, the problem is reduced to finding the set of step sizes generating the polynomial operator $P_M(x)$ such that

$$\|P_M(\Phi)\| = \max_{\lambda_i(\Phi)} |P_M(\lambda_i)| \quad (4.26)$$

is minimum.

The eigenvalues of Φ are not known in advance since the statistics of the input information are not completely known. However, one can assume that upper and lower bounds to the eigenvalues are known. Let us represent the upper and lower bound as λ_u and λ_ℓ , respectively. Given that only the range of eigenvalues $(\lambda_\ell, \lambda_u)$ is known, the optimization problem can be restated in terms of the upper bound

$$\max_{\lambda_i(\Phi)} |P_M(\lambda_i)| \leq \max_{\lambda_\ell \leq x \leq \lambda_u} |P_M(x)| \quad (4.27)$$

Instead of minimizing the maximum magnitude over the discrete set $\{\lambda_i\}$, we shall minimize for the worst distribution of the eigenvalues. The step sizes $\{\alpha_i\}_0^{M-1}$ will be chosen so as to minimize the peak value of the resulting polynomial over the range $(\lambda_\ell, \lambda_u)$. Among polynomials of degree M , the polynomial of least magnitude in the interval $(-1, 1)$ is the Chebyshev polynomial (Varga [37], p. 137) $T_M(y)$ defined by

$$\begin{aligned} T_M(y) &= \cos (M \cos^{-1} y) & |y| \leq 1, M \geq 0 \\ &= \cosh (M \cosh^{-1} y) & y \geq 1, M \geq 0 \end{aligned} \quad (4.28)$$

The mapping

$$y = \left(\frac{\lambda_u + \lambda_\ell - 2x}{\lambda_u - \lambda_\ell} \right) \quad (4.29)$$

maps the region $(\lambda_u, \lambda_\ell)$ to the $(-1, 1)$ region. Therefore, the solution to our problem is simply given in terms of the Chebyshev polynomial

$$P_M(x) = \text{Constant} \cdot T_M \left(\frac{\lambda_u + \lambda_\ell - 2x}{\lambda_u - \lambda_\ell} \right) \quad (4.30)$$

Using (4.23), we obtain $P_M(0) = I$. Therefore, the optimum polynomial operator is

$$P_M(x) = \frac{T_M \left(\frac{\lambda_u + \lambda_\ell - 2x}{\lambda_u - \lambda_\ell} \right)}{T_M \left(\frac{\lambda_u + \lambda_\ell}{\lambda_u - \lambda_\ell} \right)} \quad (4.31)$$

Equation (4.23) shows that the zeros of the polynomial $P_M(x)$ are at $x = \frac{1}{\alpha_k}$. It follows that the step sizes $\{\alpha_k\}$ are the reciprocals of the zeros of $P_M(x)$. The zeros of the Chebyshev polynomial $T_M(y)$ are given by

$$y_k = \cos \left(\frac{(2k+1)\pi}{2M} \right) \quad (4.32)$$

$$k = 0, 1, \dots, M-1$$

$$= \frac{(\lambda_u + \lambda_\ell - 2x_k)}{\lambda_u - \lambda_\ell} \quad (4.33)$$

Thus, the equation for the optimum set of step sizes is

$$\alpha_k = \frac{1}{x_k} = 2 \left[(\lambda_u + \lambda_\ell) - (\lambda_u - \lambda_\ell) \cos\left(\frac{(2k+1)\Pi}{2M}\right) \right]^{-1}$$

Since Φ is a positive-definite matrix, a lower bound to the eigenvalues of Φ can be taken as zero. Using the relation $1 - \cos\theta = 2 \sin^2 \frac{\theta}{2}$, the relation for α_k becomes

$$\begin{aligned} \alpha_k &= \frac{2}{\lambda_u} \left[\frac{1}{2 \sin^2 \theta/2} \right] \\ &= \frac{1}{\lambda_u \sin^2 \theta/2} \end{aligned} \quad k = 0, 1, \dots, M-1 \quad (4.34)$$

where $\theta = \frac{(2k+1)\Pi}{2M}$

For $k = 0$ or $M = 1$, equation (4.36) yields $\alpha_0 = \frac{2}{\lambda_u}$ which is just the optimum fixed step size (4.24). The upper bound to the largest eigenvalue for the positive definite matrix Φ is given as

$$\text{trace } \Phi = \sum_i \lambda_i > \lambda_{\max} \quad (4.35)$$

Then, a simple upper bound is

$$\lambda_u = \text{trace } \Phi \quad (4.36)$$

4.1.2 Performance of First-Order Algorithm and Simulation Results

As mentioned earlier, the decrease in the norm of the error vector is guaranteed at the Mth step, while after some of the M-1 intermediate steps the norm of the error vector may have increased depending on the order of using step sizes. That is why this algorithm is suitable only in the training mode not in the tracking mode. The tracking mode does not require test signal and thus this mode obviates the necessity for a long training period. The analysis for rate of convergence of this algorithm has been carried out in Appendix C.

Simulation Results

The simulation results were obtained for two channels referred to as channel A and channel B as shown in Figure 4.2. The length of the channel was assumed to be 5. The largest eigenvalue upper bound was calculated from the trace of the covariance matrix Φ . The input symbols were assumed to be equally probable. The fixed step size was taken to be 0.01. The input signal-to-noise ratio was held at 16.99 or 50 dB. Bipolar binary information was used throughout the simulation. The recursive algorithm for updating the tap gains of the channel estimator, given

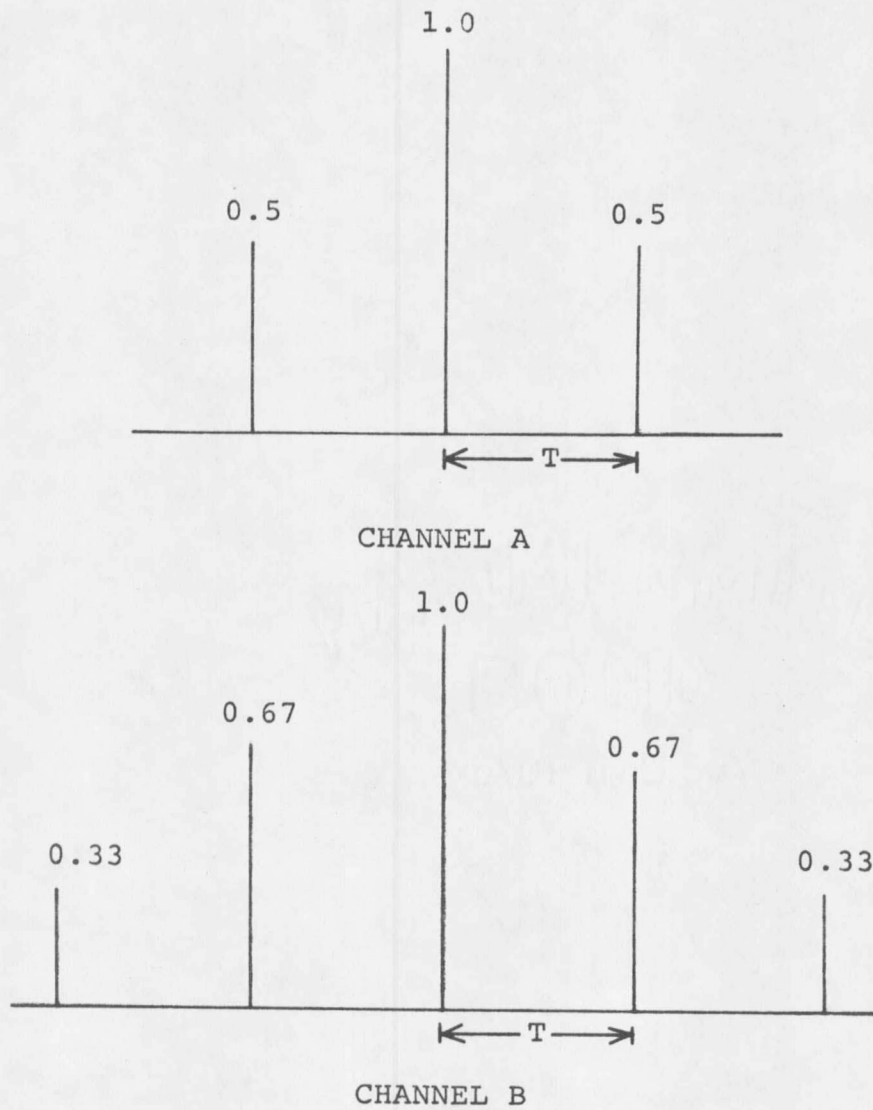


Figure 4.2 Samples of channel impulse response.

in (4.10), employs unbiased noisy estimates of the $(w+1)$ gradients. The noise in these estimates causes random fluctuations in the tap gains about their least-mean-square-error-optimal values and thus leads to an increase in the MSE. The increase in MSE above the minimum achievable MSE due to the estimation of noisy gradients has been termed "excess-mean-square-error" [53].

It is obvious that the amplitude of the random fluctuations of the tap gains increase with an increase in the value of α_k . Thus, a large value of α_k , which causes rapid adaptation results in a large excess MSE, while a small value of α_k , causing slow adaptation, results in a small excess MSE. It is clear from this discussion that the choice of α_k involves a tradeoff between the speed of convergence of the recursive algorithm and the amount of excess MSE that can be tolerated. The author has found that a useful guideline in selecting a value of the parameter α_k for a given channel is to choose α_k so that it is ten percent of its analytic value. The norm of the error vector versus the number of iterations is plotted in Figures 4.3 and 4.4 using first-order algorithm for channels A and B. These plots clearly illustrate the

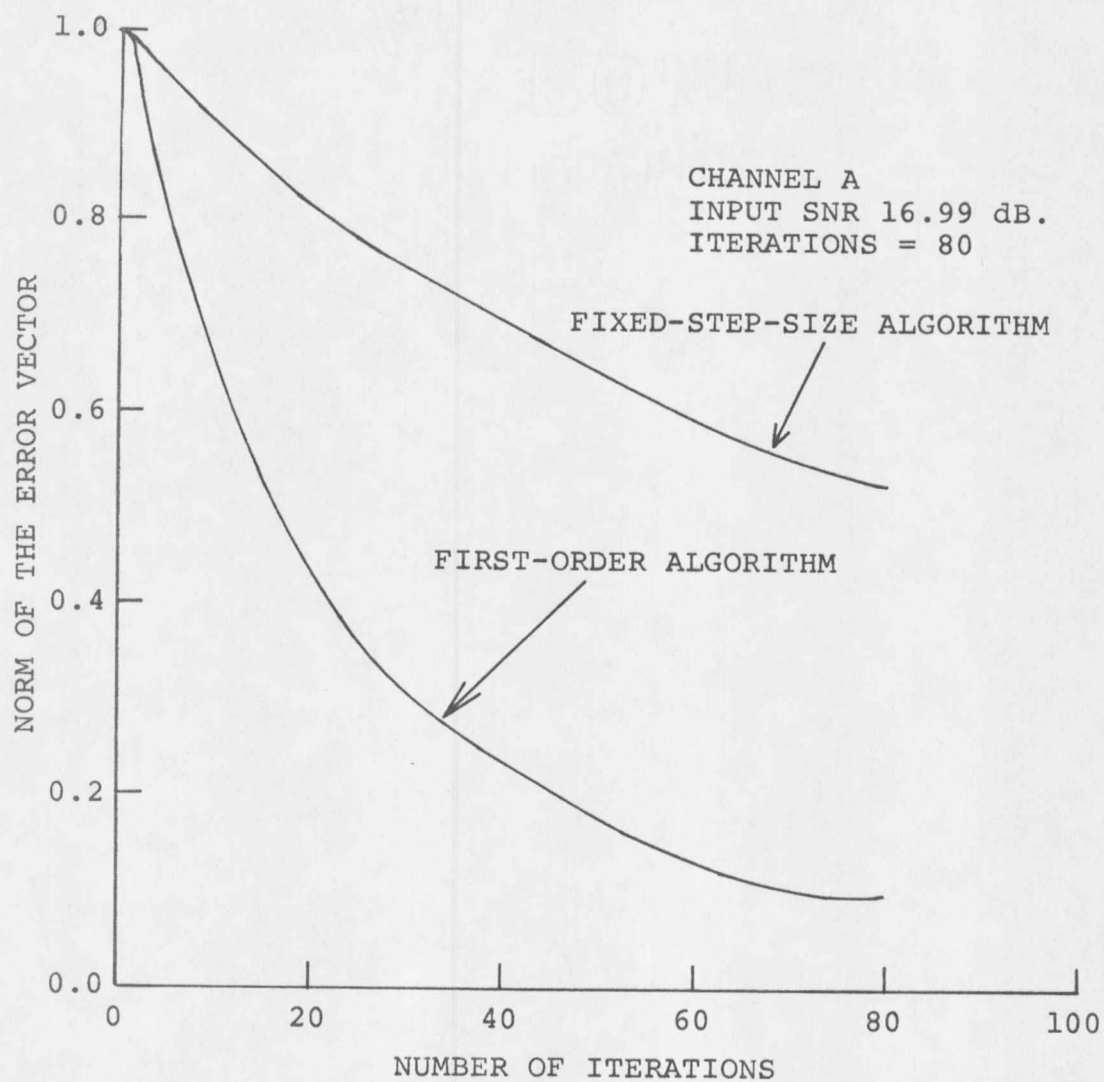


Figure 4.3(a) Plot of norm of the error vector vs. iterations.

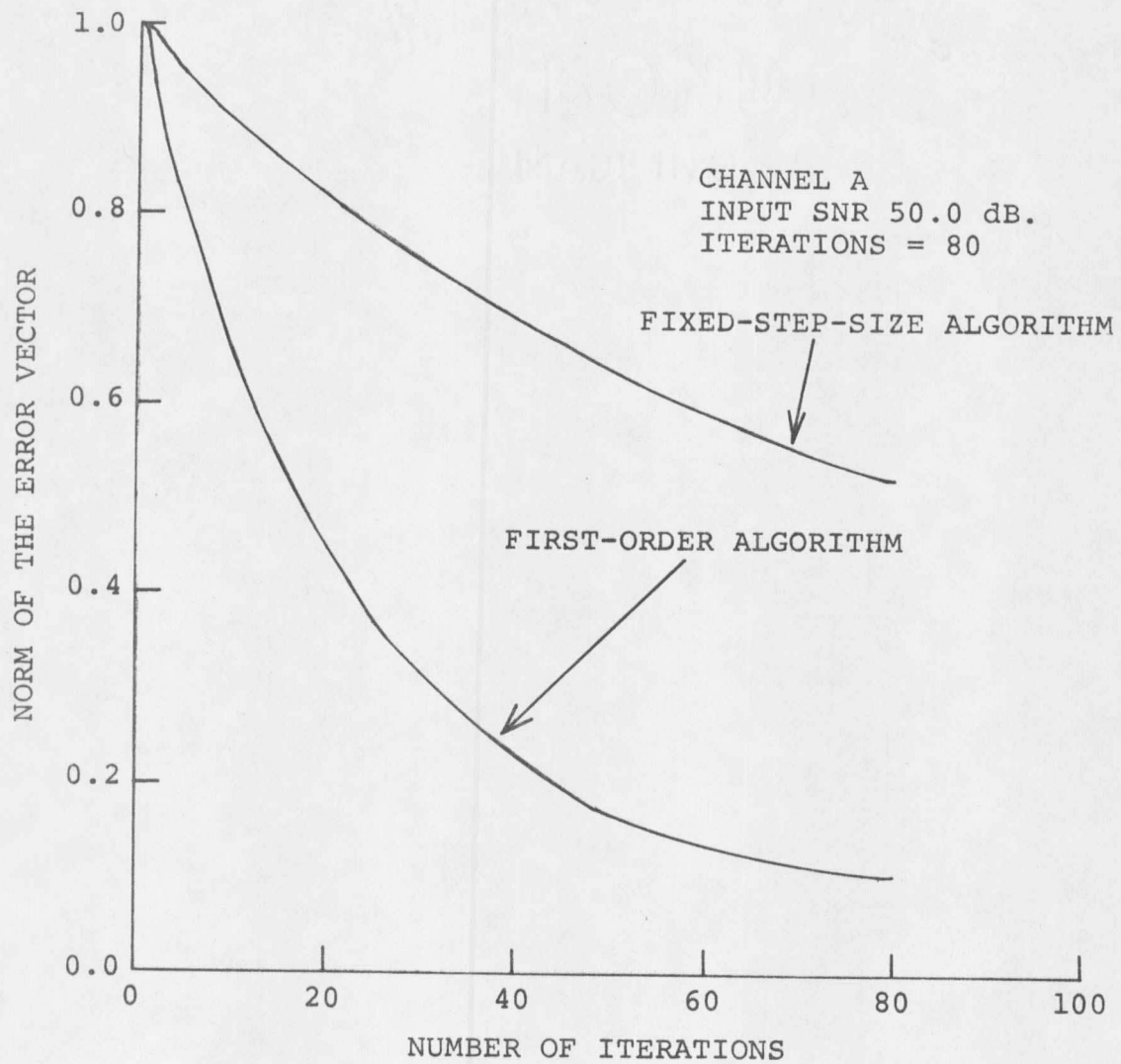


Figure 4.3(b) Plot of norm of the error vector vs. iterations.

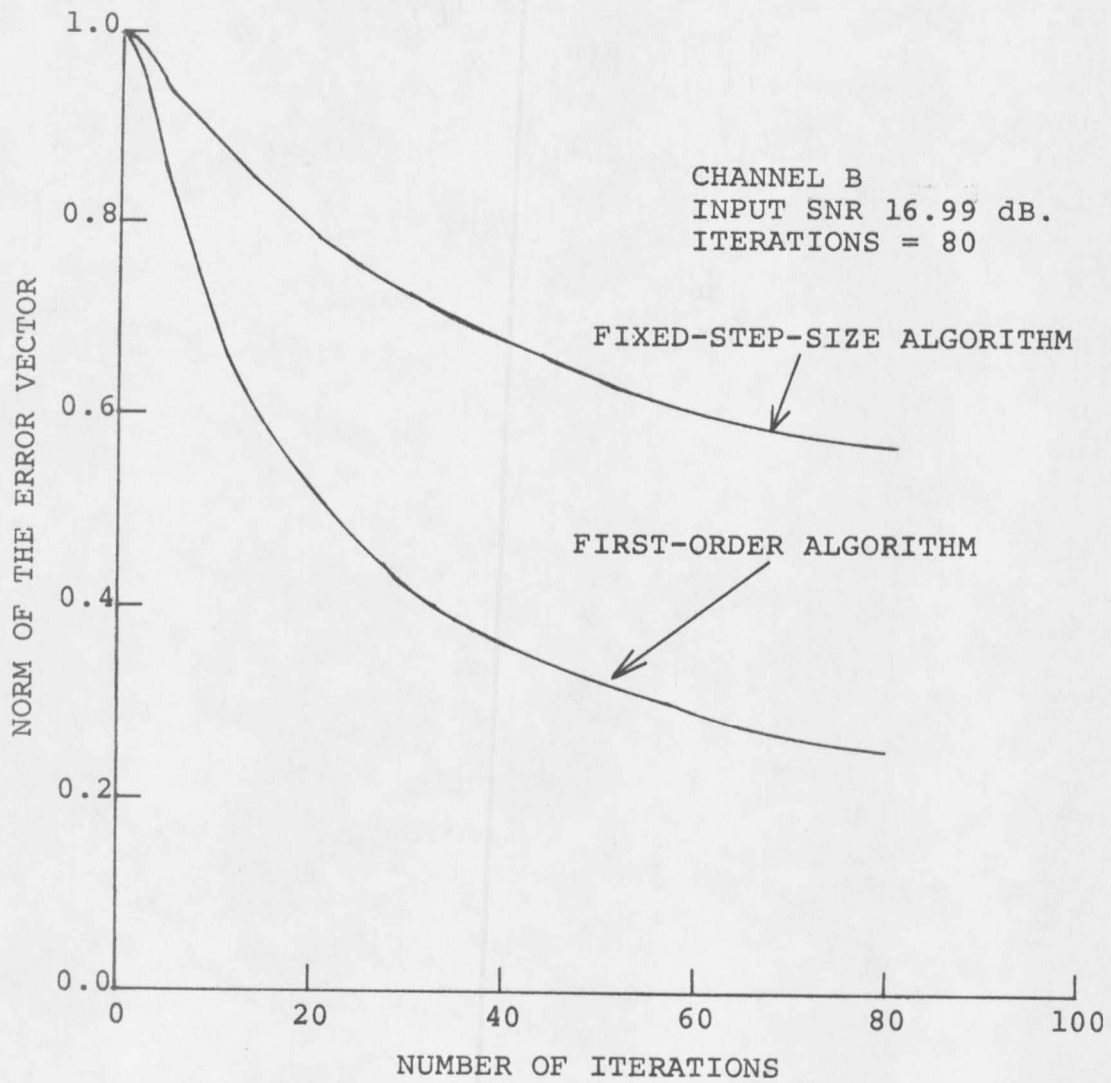


Figure 4.4(a) Plot of norm of the error vector vs. iterations.

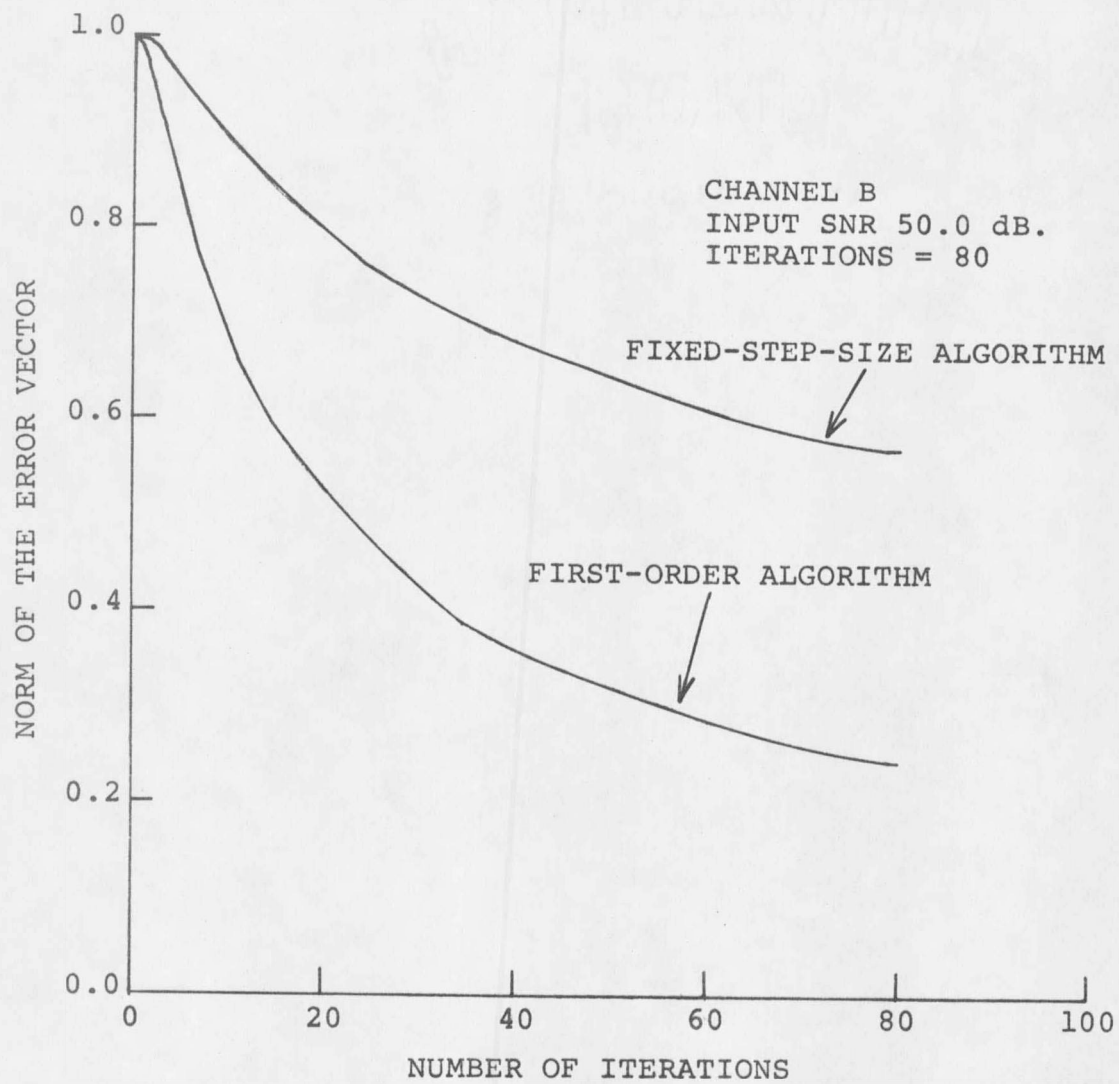


Figure 4.4(b) Plot of norm of the error vector vs. iterations.

advantages of the first-order algorithm relative to the fixed-step-size algorithm.

4.2 Second-Order Algorithm

The first-order algorithm described above is useful only in a training mode because the improvement is guaranteed only at the end of every M iteration, not at iterations in between. Therefore, we consider a second-order search algorithm designed to provide the minimum norm of the tap-gain error at each iteration. Convergence in an average sense is thus always monotonic.

4.2.1 Second-Order Algorithm Derivation

The recursive relation used in the second-order algorithm is of the form

$$g_j^{k+1} = g_j^k - \frac{\alpha_k}{2} \frac{\partial \epsilon}{\partial g_j} + \beta_k (g_j^k - g_j^{k-1}) \quad (4.37)$$

where α_k and β_k are to be found so that the norm of the error vector decreases at each step. We know from our previous results that

$$\Phi G = B$$

and

$$G_{\text{opt}} = \Phi^{-1} B$$

where Φ , G , G_{opt} , and B have the same meaning as described earlier. It is also clear from the derivation of first-order algorithm that

$$\begin{aligned} E^{k+1} &= G^{k+1} - \Phi^{-1} B \\ G^{k+1} &= E^{k+1} + \Phi^{-1} B \end{aligned} \quad (4.38)$$

In matrix notation equation (4.37) becomes

$$G^{k+1} = G^k - \frac{\alpha_k}{2} \left(\frac{\partial \varepsilon}{\partial G} \right) + \beta_k (G^k - G^{k-1}) \quad (4.39)$$

where

$$\frac{\partial \varepsilon}{\partial G} = \left\{ \frac{\partial \varepsilon}{\partial g_j} \right\}$$

Substituting the relation for G^{k+1} from (4.38) into (4.39), we get

$$\begin{aligned} E^{k+1} + \Phi^{-1} B &= G^k - \frac{\alpha_k}{2} \left(\frac{\partial \varepsilon}{\partial G} \right) + \beta_k (G^k - G^{k-1}) \\ E^{k+1} &= G^k - \Phi^{-1} B - \frac{\alpha_k}{2} \left(\frac{\partial \varepsilon}{\partial G} \right) + \beta_k (G^k - G^{k-1}) \end{aligned} \quad (4.40)$$

Now $G^k - \Phi^{-1} B = E^k$ and $G^k - G^{k-1} = E^k - E^{k-1}$ which are used in (4.40) to obtain

$$\begin{aligned} E^{k+1} &= E^k - \frac{\alpha_k}{2} \left(\frac{\partial \varepsilon}{\partial G} \right) + \beta_k (E^k - E^{k-1}) \\ &= (1 + \beta_k) E^k - \frac{\alpha_k}{2} \left(\frac{\partial \varepsilon}{\partial G} \right) - \beta_k E^{k-1} \end{aligned} \quad (4.41)$$

But from Appendix A,

$$\begin{aligned}
 \frac{\partial \varepsilon}{\partial G} &= (-2)(B - \Phi G^k) \text{ and (4.41) becomes} \\
 E^{k+1} &= (1 + \beta_k) E^k - \frac{\alpha_k}{2} (-2)(B - \Phi G^k) - \beta_k E^{k-1} \\
 &= (1 + \beta_k) E^k + \alpha_k \Phi (\Phi^{-1} B - G^k) - \beta_k E^{k-1} \\
 &= (1 + \beta_k) E^k + \alpha_k \Phi (-E^k) - \beta_k E^{k-1} \\
 &= \left[(1 + \beta_k) I - \alpha_k \Phi \right] E^k - \beta_k E^{k-1} \quad (4.42)
 \end{aligned}$$

where I is the $(w+1) \times (w+1)$ identity matrix.

Equation (4.42) can also be written in the form

$$E^{k+1} = P_k(\Phi) E^0 \quad (4.43)$$

where $P_k(\Phi)$ is a polynomial of degree k in the matrix Φ and is readily shown to satisfy the second-order recursion

$$P_{k+1}(\Phi) = \left[(1 + \beta_k) I - \alpha_k \Phi \right] P_k(\Phi) - \beta_k P_{k-1}(\Phi) \quad (4.44)$$

In the first-order algorithm derivation, it is shown that the norm of E^k , $\|E^k\|$, was minimized in a least upper-bound sense, by forcing $P_k(\Phi)$ to be a Chebyshev polynomial of order k . In particular, we showed that

$$P_k(\Phi) = \frac{T_k(\gamma I - \frac{2}{\lambda_u - \lambda_l} \Phi)}{T_k(\gamma)} \quad (4.45)$$

with $T_k(\gamma)$ the first-order Chebyshev polynomial,

$\gamma = \frac{\lambda_u + \lambda_\ell}{\lambda_u - \lambda_\ell}$, and λ_u and λ_ℓ , respectively, the upper and lower bounds to the eigenvalues of Φ . With the first-order algorithm $\|E^k\|$ could be minimized in a least-upper-bound sense at one value of k only. We demonstrate now that the second-order algorithm enables us to continually minimize the norm of the tap gain error. Recall [51] that $T_k(A)$ satisfy the recursive relation

$$T_{k+1}(A) = 2A T_k(A) - T_{k-1}(A) \quad (4.46)$$

Substituting (4.45) into (4.44) and letting

$$A = \gamma I - \frac{2}{\lambda_u - \lambda_\ell} \Phi, \text{ it is apparent (Appendix B) that (4.46)}$$

and (4.44) are identical, provided that

$$\beta_k = \frac{T_{k-1}(\gamma)}{T_{k+1}(\gamma)} \quad k = 1, 2, \dots \quad (4.47)$$

and

$$\beta_k = \frac{4}{(\lambda_u - \lambda_\ell)} \left[\frac{T_k(\gamma)}{T_{k+1}(\gamma)} \right] = \alpha_0 (1 + \beta_k) \quad (4.48)$$

with $\beta_0 = 0$, and $\alpha_0 = \frac{2}{\lambda_u + \lambda_\ell}$

The α_k and β_k parameters can be readily calculated recursively from (4.46) and 4.47). From Appendix B, it also follows that

$$\beta_{k+1} = \frac{(1+\beta_k)}{(4\gamma^2-1-\beta_k)} \quad k = 1, 2, \dots \quad (4.49)$$

with $\beta_1 = (2\gamma^2-1)^{-1}$

$$\begin{aligned} \alpha_0 &= 2/(\lambda_u + \lambda_\ell) \\ \alpha_{k+1} &= \alpha_0 (1+\beta_{k+1}) \quad k = 0, 1, 2, \dots \end{aligned} \quad (4.50)$$

Equations (4.49) and (4.50) thus provide the values of α_k and β_k to be used in the iterative algorithm of (4.37) to guarantee that at any value of k , E is minimum in a least-upper-bound sense.

As mentioned earlier, Φ is in general a positive-definite matrix. So the lower bound λ_ℓ can be assumed as zero. Hence,

$$\gamma = \frac{\lambda_{u+0}}{\lambda_{u-0}} = 1 \quad (4.51)$$

Equations (4.49) and (4.50) are further simplified to the following equations:

$$\beta_{k+1} = \frac{1+\beta_k}{(3-\beta_k)} \quad k = 1, 2, \dots \quad (4.51)$$

$$\beta_0 = 0 \text{ and } \beta_1 = 1$$

$$\alpha_0 = \frac{2}{\lambda_u}$$

$$\alpha_{k+1} = \alpha_0 (1 + \beta_{k+1}) \quad k = 0, 1, 2, \dots \quad (4.52)$$

Equations (4.51) and (4.52) are to be used in the second-order recursive relation (4.37). The superiority of this algorithm over the fixed-step-size algorithm has been shown by simulation as described next.

4.2.2 Performance of Second-Order Algorithm and Simulation Results

It has been shown [36] that with the second-order algorithm, the norm of the tap-gain error decreases at each step; and therefore, this algorithm performs better than the first-order algorithm. Considerable improvement in the rate of convergence over the fixed-step algorithm is possible using the second-order algorithm. In terms of computing machine applications, the second-order iterative method requires an additional vector of storage over the iterative method of first-order, an item which can be

of considerable weight in the solution of large matrix problems.

Simulation Results

Simulation results using second-order algorithm were obtained again for the channels A and B. The recursive (4.37) for the adjustment of the tap gains of the channel estimator makes use of the gradients of the mean-square error. Unbiased estimates of the gradients are used in the recursive relation (4.37) as it is not possible (in general) to find the true-value of the gradients. As explained earlier, this noisy estimate of the gradients caused an increase in an excess MSE. Thus, the α 's obtained analytically in equation (4.52) were not employed directly, but ten percent of actual α 's were used in the recursive relation (4.37) to account for the noisy estimate of the gradients. All other parameters were taken to be the same as for the first-order algorithm. The results obtained are shown in Figures 4.5 and 4.6 for two channels.

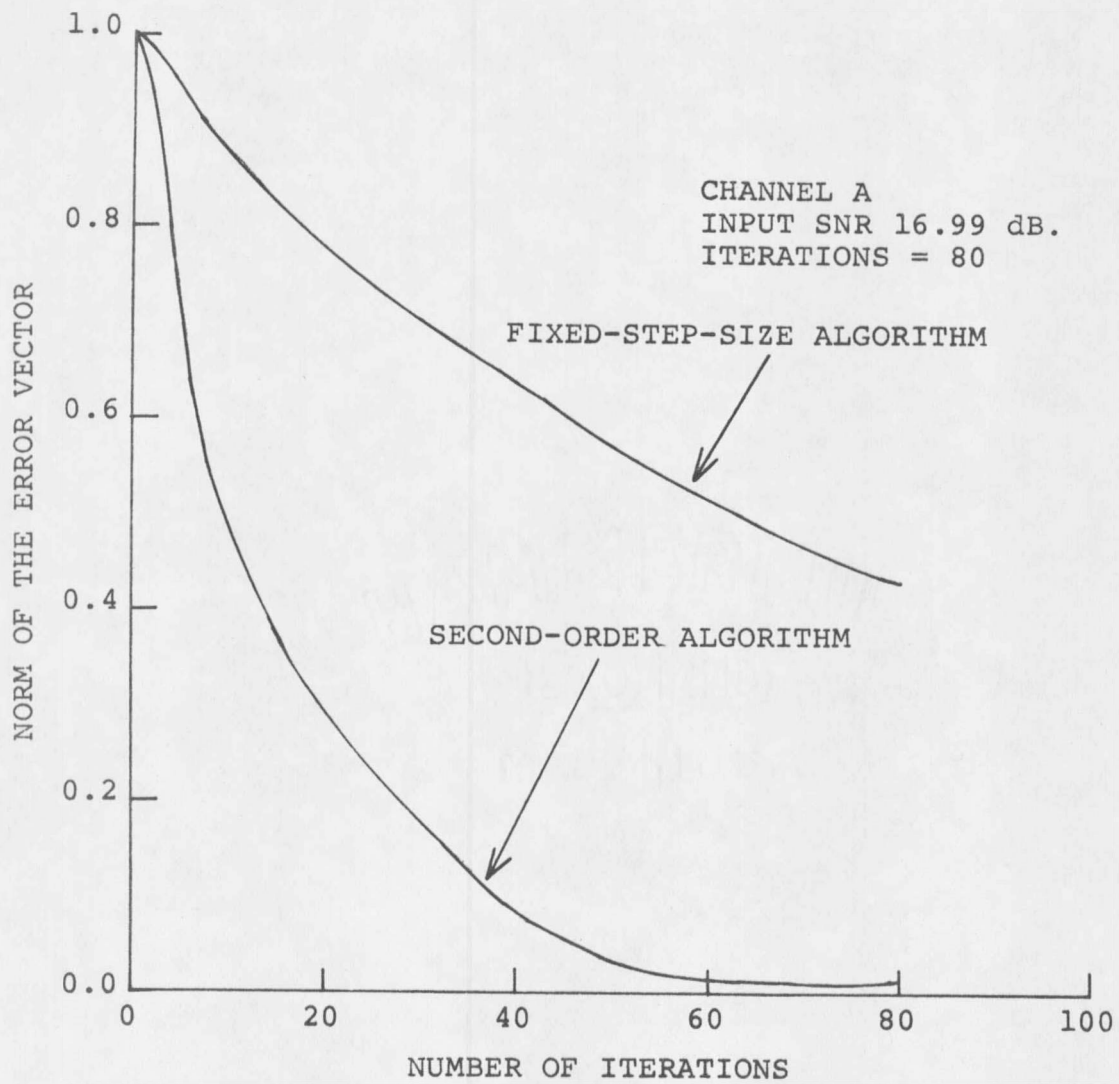


Figure 4.5(a) Plot of norm of the error vector vs. iterations.

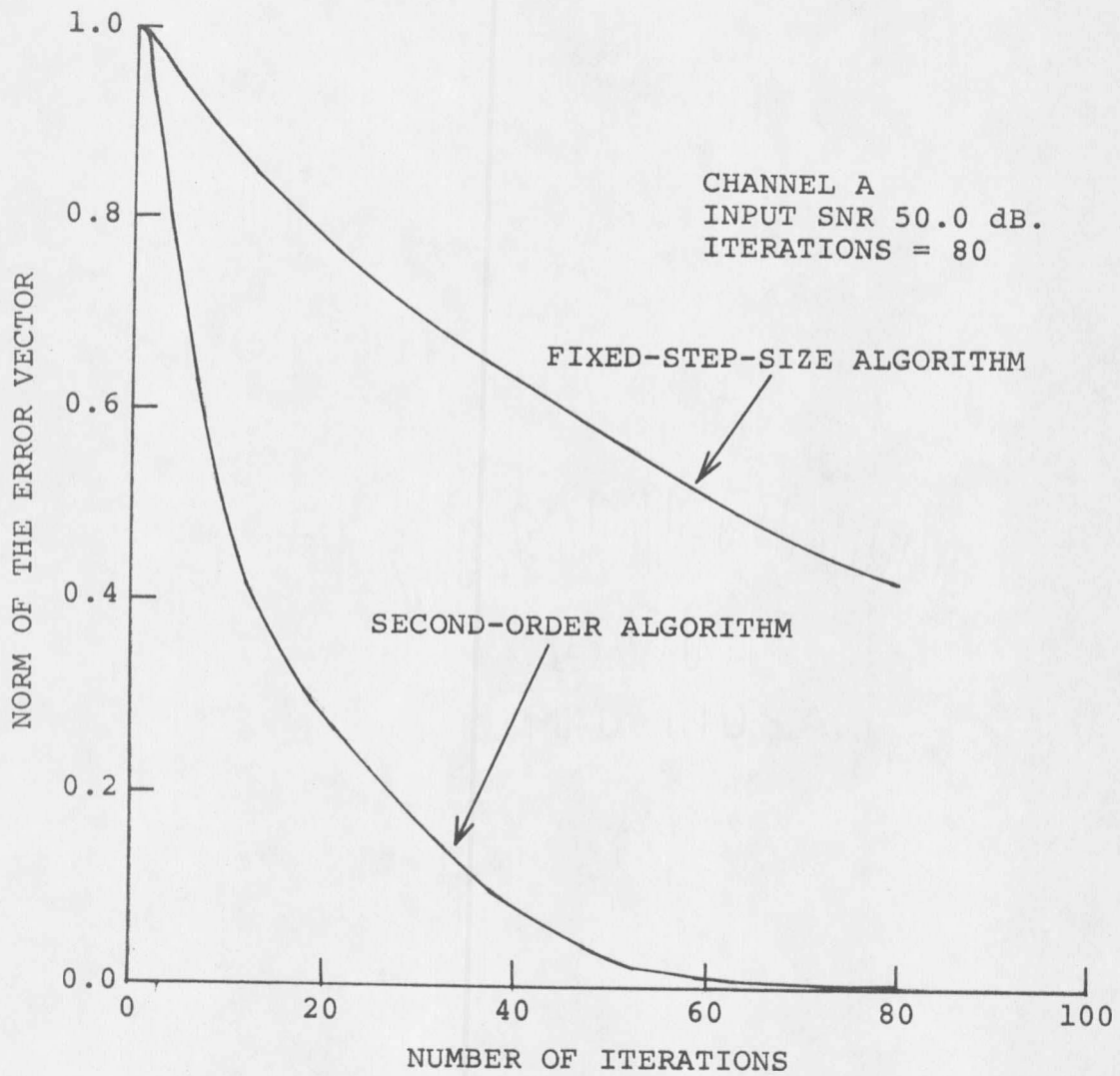


Figure 4.5(b) Plot of norm of the error vector vs. iterations.

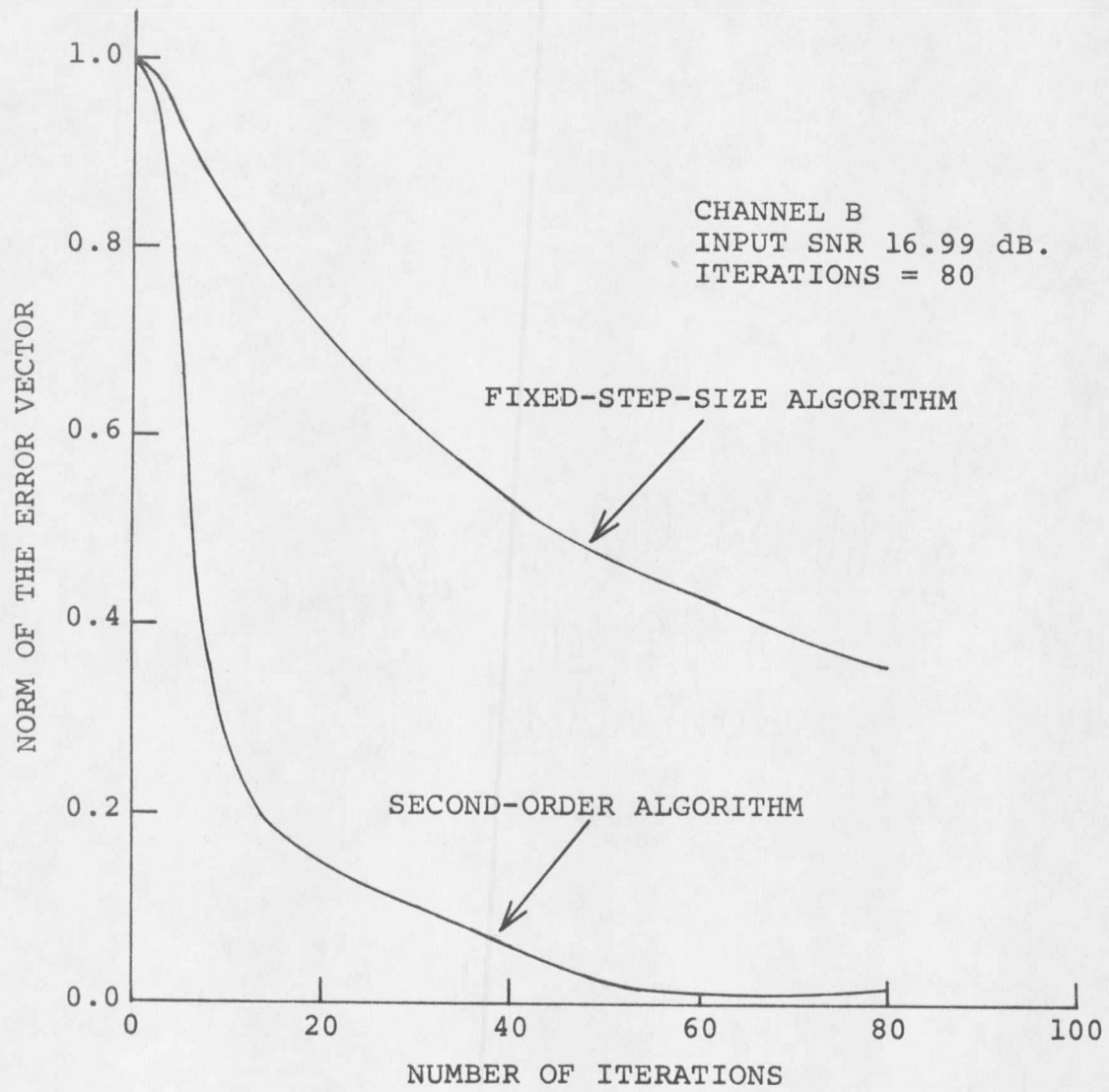


Figure 4.6(a) Plot of norm of the error vector vs. iterations.

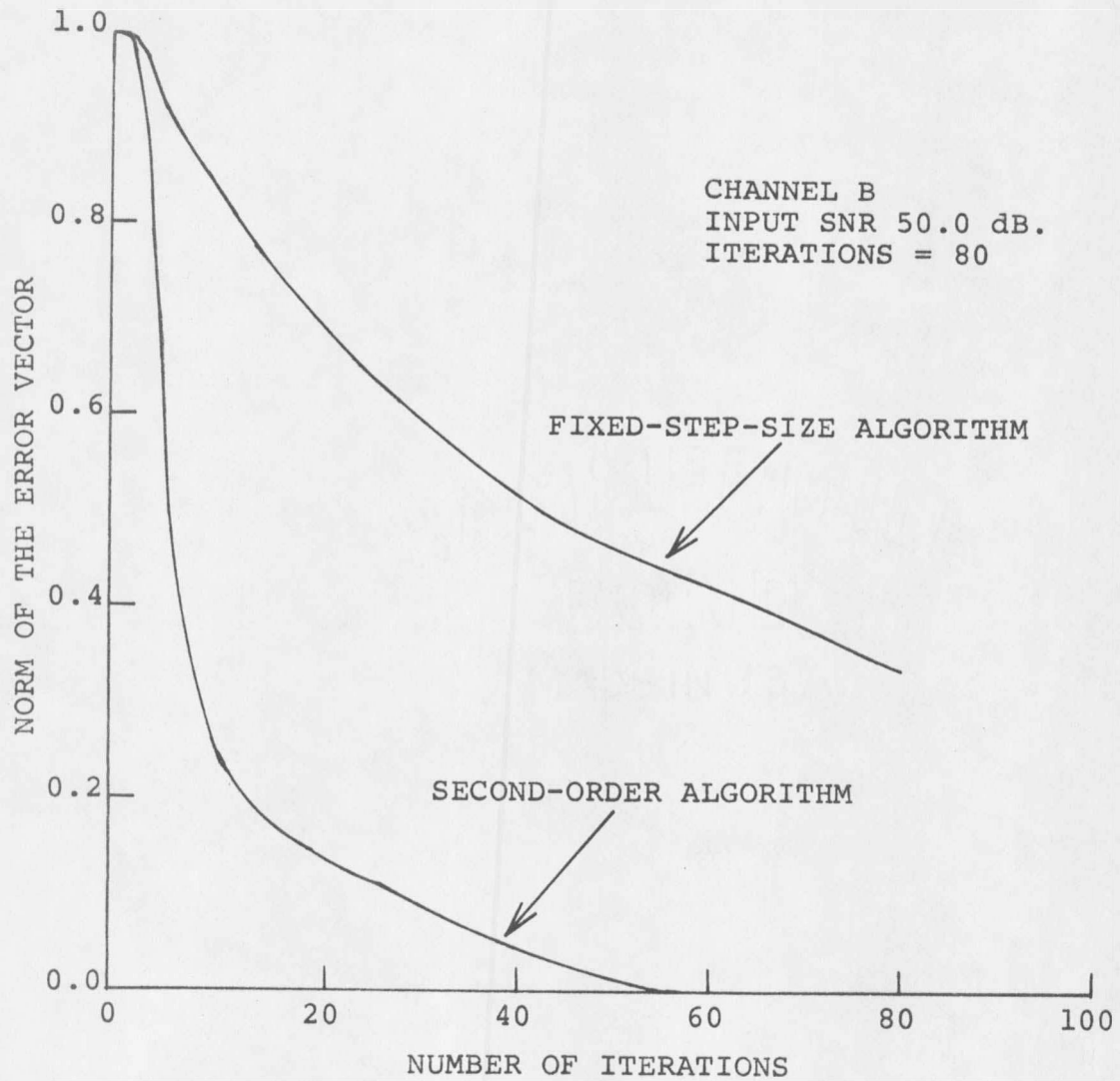


Figure 4.6(b) Plot of norm of the error vector vs. iterations.

Chapter V

SENSITIVITY STUDY OF FORNEY'S SUBOPTIMUM ALGORITHM AND EQUALIZATION SCHEME FOR MAKING FORNEY'S SUBOPTIMUM ALGORITHM ADAPTIVE

5.1 Forney's Suboptimum Algorithm

This suboptimum algorithm is suitable for the class of partial responses $f(D) = 1 \pm D^n$, and we will describe in detail the suboptimum algorithm for a particular type of partial response $f(D) = 1 - D$. This scheme involves making tentative decisions that are then corrected by a single error detecting and correcting decoder. Since

$$f(D) = 1 - D \quad (5.1)$$

and

$$y(D) = x(D)f(D) \quad (5.2)$$

it follows that

$$y_k = x_k - x_{k-1} \quad (5.3)$$

With m -level inputs x_k , the output signals y_k take on $(2m-1)$ different levels. These output signals y_k are corrupted by noise to form z_k which is the observed output. Tentative decisions y_k are made from the noisy outputs as to which output signal level is most probable. Since

there are more $(2m-1)$ -ary sequences than m -ary¹ sequences, certain sequences of tentative decisions can be recognized as impossible in the sense that no allowable input sequence could have caused them. This type of redundancy is used to detect errors in the tentative decisions.

In order to determine if the tentative decision sequence is allowable, the general method is to pass $\hat{y}(D)$ through an inverse filter with impulse response $1/f(D)$ and to see whether an allowable input $\hat{x}(D)$ comes out. For the particular response $f(D) = 1-D$, such a filter is realized by the feedback circuit in which $\hat{x}_k = \hat{y}_k + \hat{x}_{k-1}$. This feedback circuit is shown in Figure 5.1.

Whenever a single error is made, say $\hat{y}_k = y_k + 1$, the output $\hat{x}_k$ at that time will be one unit higher than the corresponding input x_k (see Figure 5.1). Because of the feedback, the error continues to propagate in the circuit and raises all subsequent $\hat{x}_k$ by one unit as well. At the first time $k' \geq k$ for which $x_{k'} = m-1$, we will observe the unallowable level $x_{k'}' = m$ and detect the error. At this time we can reset $x_{k'}'$ to $m-1$ and error propagation ceases.

¹ m -ary sequence is a sequence with alphabet size equal to m .

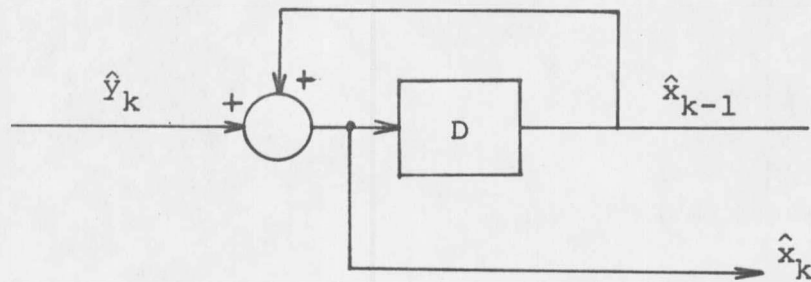


Figure 5.1 Inverse linear filter with response $1/(1-D)$.

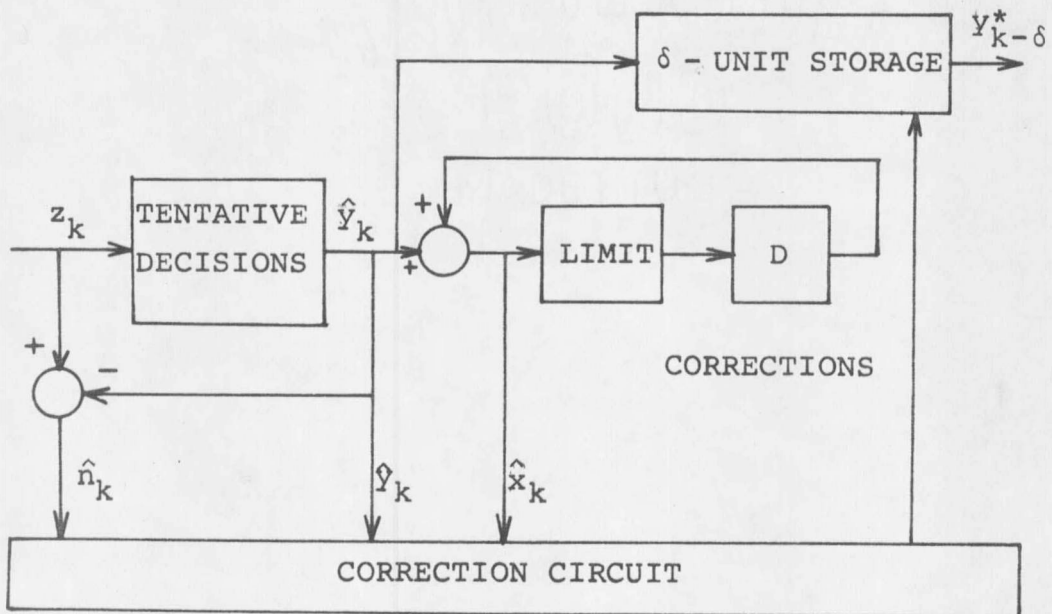


Figure 5.2 Error-correction algorithm block diagram.

Similarly, negative errors are detected when $x_k < 0$.

The error detection circuit yields additional information beyond simply detecting the error. First, it indicates the polarity of the error. Second, observe that whenever $\hat{x}_k = 0$ it is certain that no positive error is circulating in the circuit, since that would imply $x_k < 0$; similarly $\hat{x}_k = m-1$ implies no negative error. When we detect a positive error we can therefore be sure that it occurred in the finite time span since the last time $\hat{x}_k = 0$ and likewise for a negative error.

To localize the error within this time span requires information about the reliability of each tentative decision. Let the apparent error $\hat{n}_k$ be defined as $\hat{n}_k = z_k - \hat{y}_k$; for any reasonable noise distribution the tentative decision most likely to be in error is that for which $\hat{n}_k$ has the largest magnitude with appropriate polarity. We, therefore, correct in that place. Figure 5.2 shows the circuit that generates $\hat{y}_k$, $\hat{n}_k$, and $\hat{x}_k$ and stores tentative decisions for δ time units awaiting correction. To determine how long the storage time δ should be, we note that the probability an error at time k will not have been detected before $k+\tau$ is $(1-1/m)^\tau$, the probability that

τ consecutive input symbols are not equal to $m-1$ or 0 , as the case may be. For $m=4$ and $\delta=20$, $(1 - \frac{1}{m})^\delta = 0.003$; thus, if $\delta=20$, one out of 300 errors will not be detected in time. This is satisfactory if and only if we are looking for not more than about two orders of magnitude of error rate improvement.

This algorithm has been used in a commercially available 9600 bit/s telephone line modem with 4800 inputs/s, $m=2$ or 4 and a partial response for $m=2$ or 4 with $f(D) = 1-D^2$. To implement an equivalent algorithm for $m=2$ or 4 with $f(D) = 1-D^2$ with integrated resistor-transistor logic (RTL) circuits (2 flip/flops per IC) took about 50 integrated circuits, including all of the control logic. With this algorithm, the error rates are in the range of $10^{-3} - 10^{-5}$ which is an order of magnitude or so improvement in error rate with a tendency to less improvement at the higher error rate and more at the lower.

The equivalence of this algorithm to maximum-likelihood sequence estimation under two assumptions has been shown by Forney [22]. These assumptions are:

Assumption 1. No noise of magnitude $|n_k| \geq 1$ occurs.

Assumption 2. After any single tentative decision error $\left[|n_k| \geq \frac{1}{2}\right]$, no second error occurs before the error is detected.

Using these assumptions, Forney [22] has been able to show the equivalence. Moreover, the probability that the above assumptions are not true is very small. Therefore, this suboptimum algorithm works almost as well as the Viterbi algorithm. The advantage of the suboptimum algorithm over the Viterbi algorithm is that it is very simple and much easier to implement.

5.2 Sensitivity Study of Forney's Suboptimum Algorithm

In actual practice $f(D)$ will not be exactly equal to $1-D$. In this section, we consider the effects of slight variations in $f(D) = 1-D$. Since analytic results are hard to obtain, the effects of slight differences in $f(D)$ from $1-D$ have been investigated using simulation. Simulation results were obtained for the following $f(D)$:

$$f(D) = 1-D \quad (5.4)$$

$$f(D) = 1 - 0.98D \quad (5.5)$$

$$f(D) = 1-D - 0.1 D^2 \quad (5.6)$$

$$f(D) = 1-D + 0.2 D^2 \quad (5.7)$$

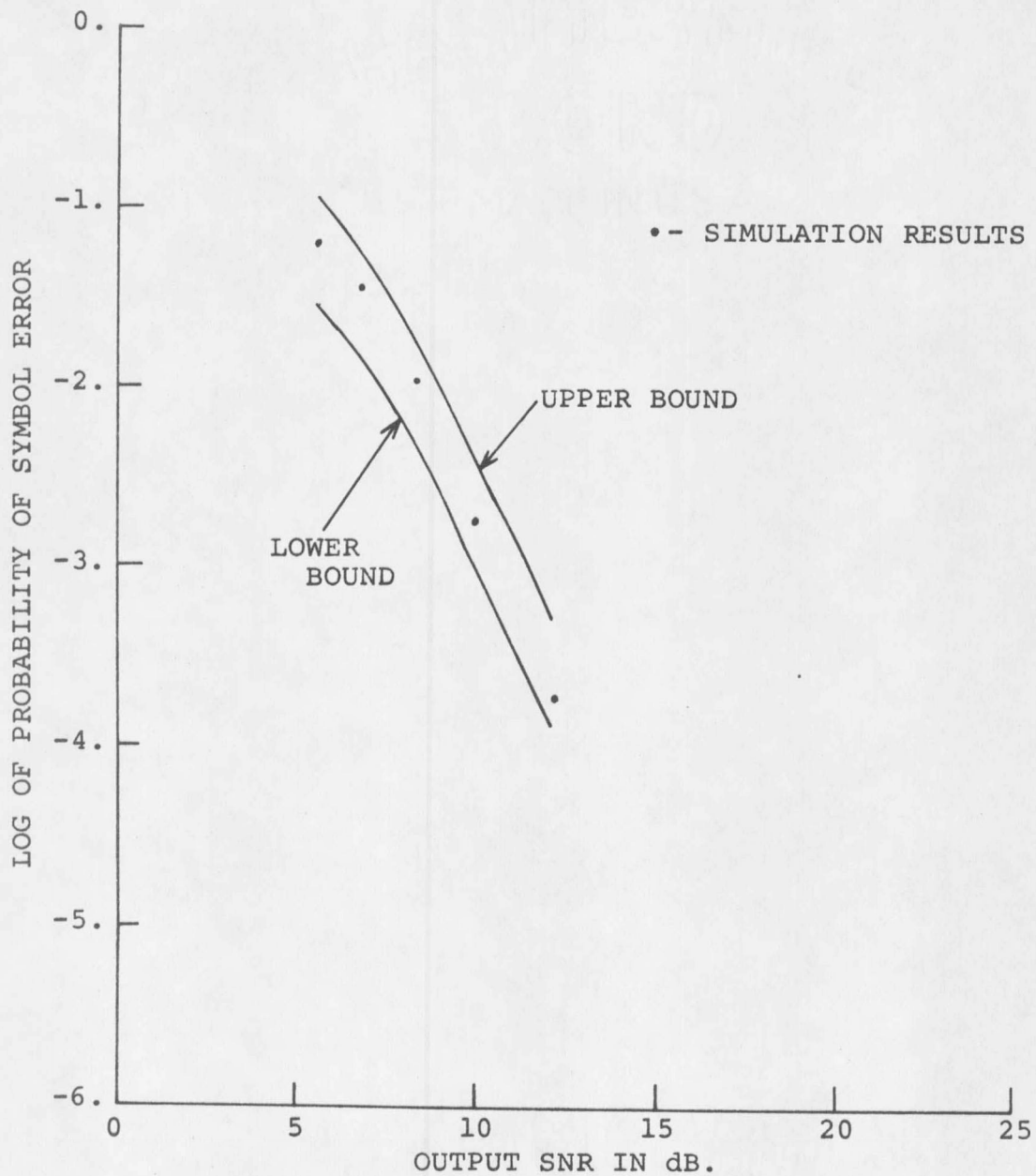


Figure 5.3(a) Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-D$.

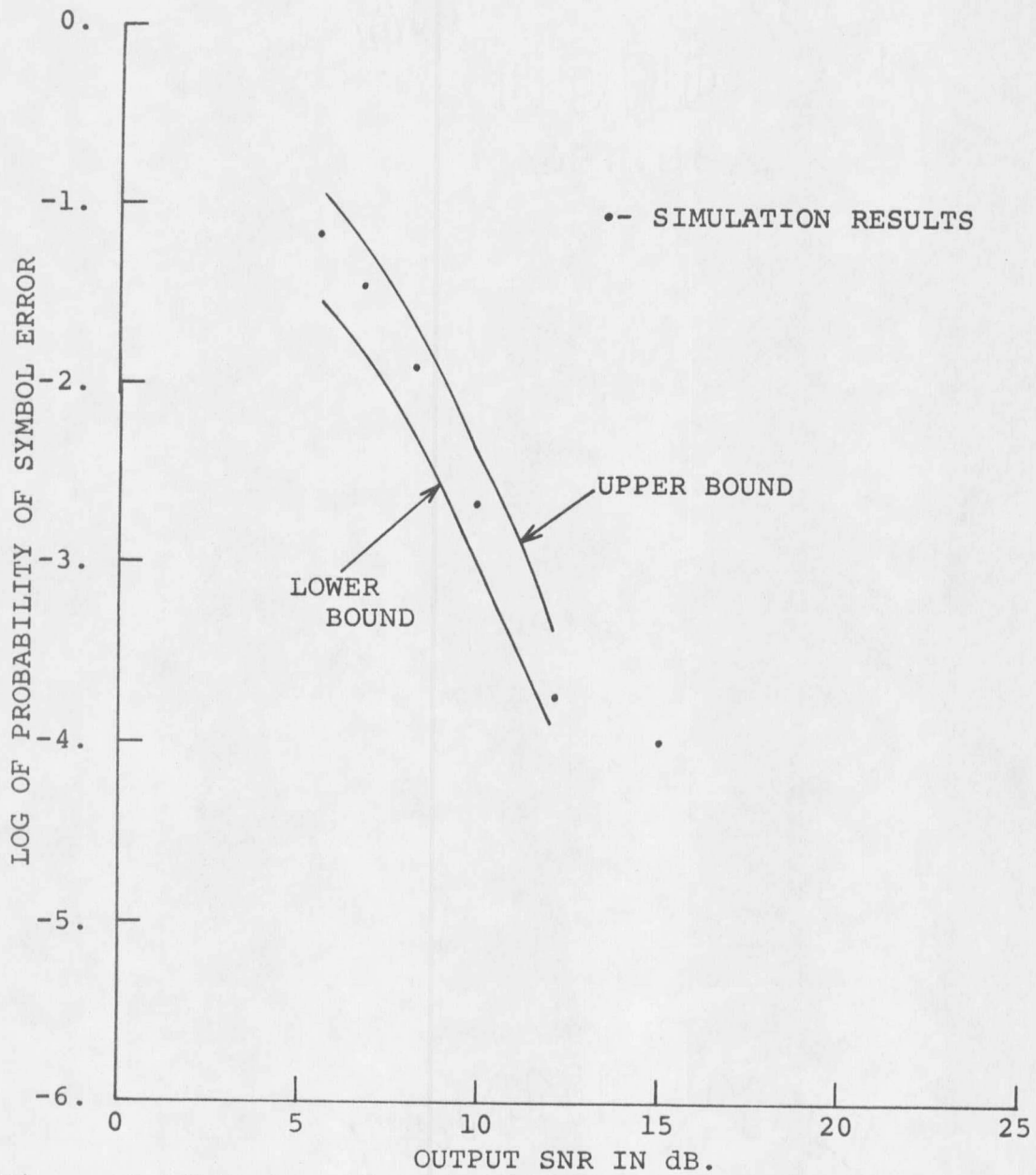


Figure 5.3(b) Log of probability of symbol error vs. output SNR in dB for $f(D) = 1 - 0.98D$.

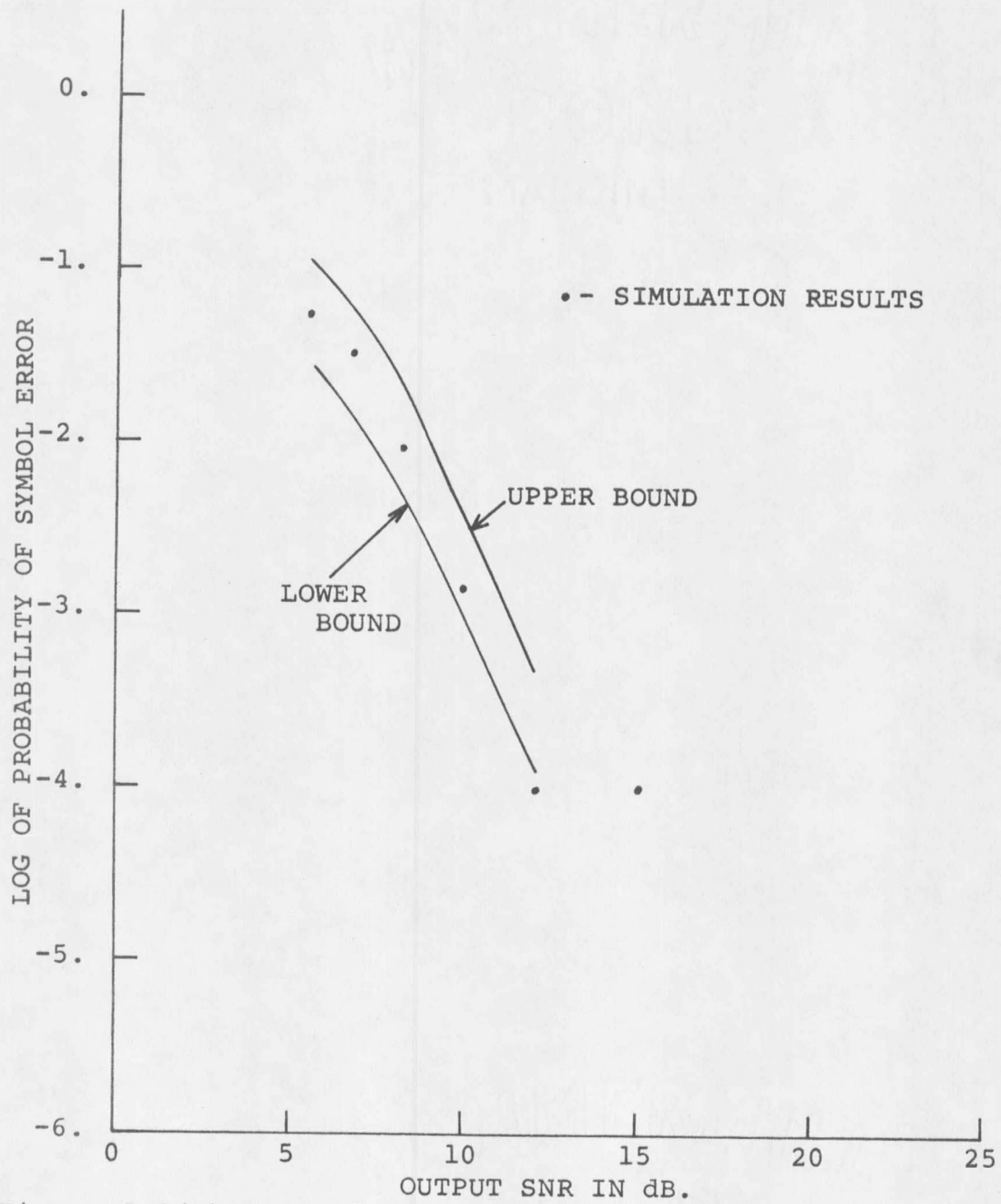


Figure 5.3(c) Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1-D-0.1D^2$.

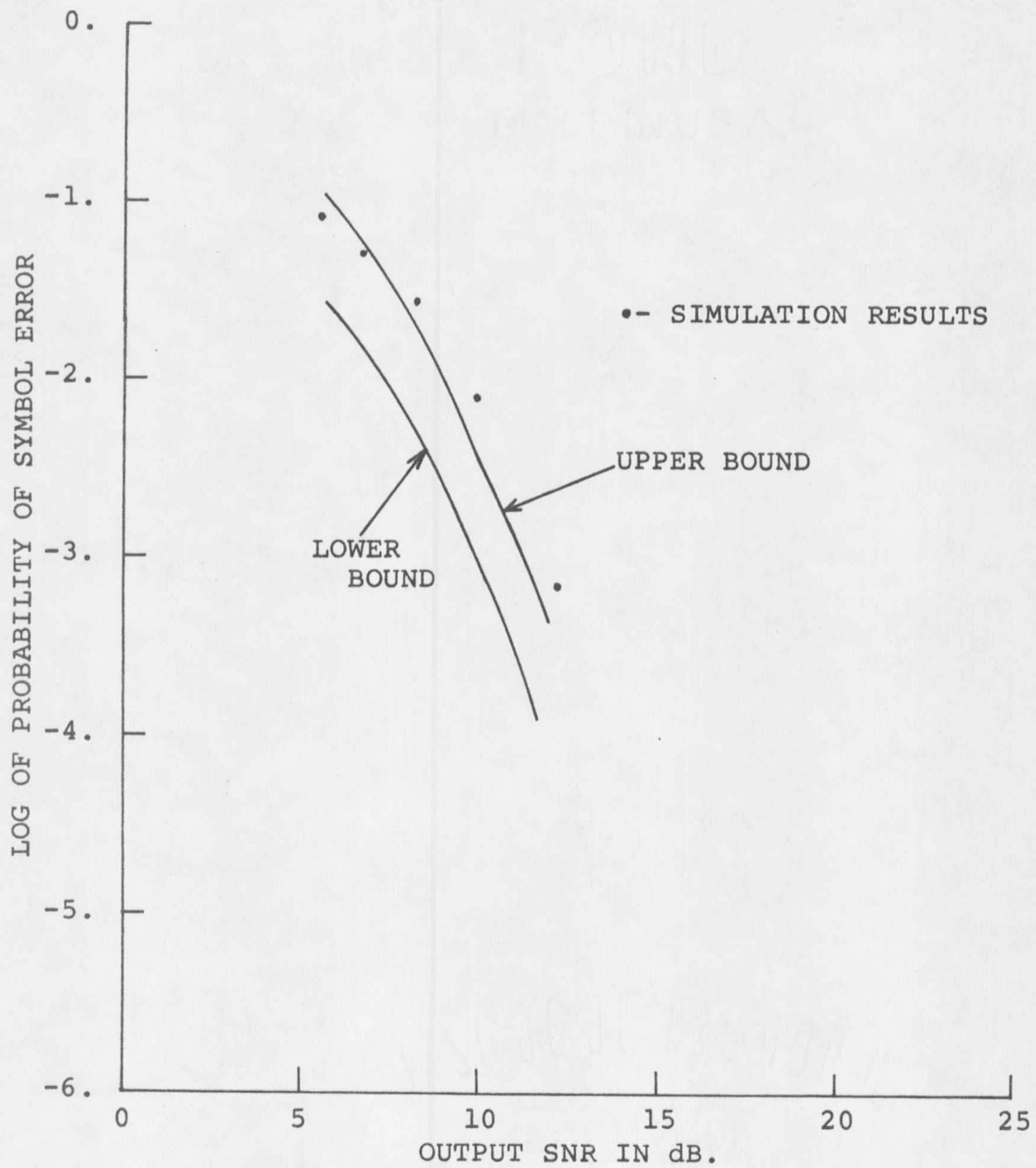


Figure 5.3(d) Log of probability of symbol error vs. output SNR in dB for $f(D) = 1 - D + 0.2D^2$.

Simulation Results

The simulation results were obtained for the above sampled channel responses using the method of computer simulation explained in Appendix D. The lower and upper bounds were taken to be the same as given by Forney [22]. Figure 5.3 shows the results obtained for different $f(D)$ with log of probability of error versus output signal-to-noise ratio. It is obvious from these plots that this suboptimum algorithm is not sensitive to small fluctuations in $f(D)$.

5.3 Adaptive Version of Suboptimum Algorithm

As mentioned earlier, Forney's suboptimum algorithm is suitable for $f(D) = 1 + D^n$. This section will describe a scheme to make the suboptimum algorithm suitable for $f(D)$ which is different from $1 + D^n$. A linear equalizer is used in conjunction with the channel model and the purpose of the equalizer is to make the overall response (of channel and equalizer) behave like the desired response for the suboptimum algorithm.

The test signal is used for adjusting the tap gains of the equalizer, and hence this scheme can be used only in the training mode. The basic scheme is represented by the block diagram in Figure 5.4.

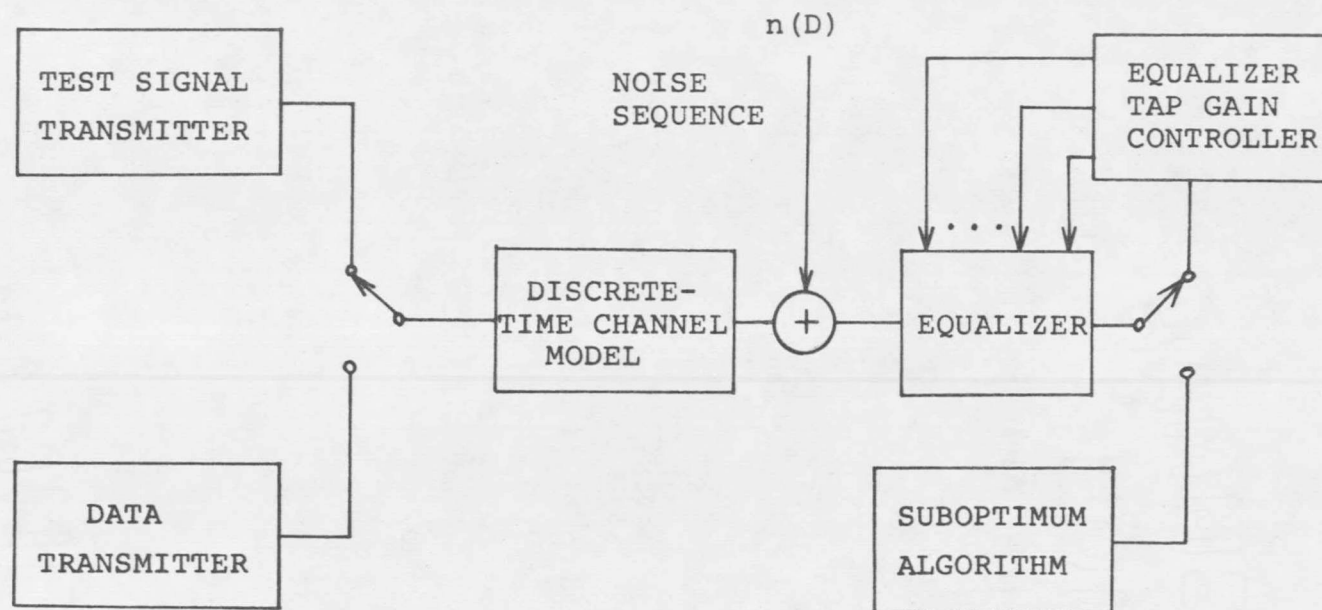


Figure 5.4 Block diagram of equalization scheme for suboptimum algorithm.

Notation:

$x(D)$: Test signal sequence (This scheme will work in the best way if we have advance knowledge of the transmitted signal, i.e., if the transmitted signal is the same as the test signal.).

$f(D)$: Sampled channel response.

$e(D)$: Sampled equalizer response.

$F(D)$: Sampled response obtained by convolving $f(D)$ and $e(D)$, i.e.,

$$F(D) = e(D) f(D) \quad (5.9)$$

v : Span of the channel response.

$$f(D): f(D) = f_0 + f_1 D + f_2 D^2 + \dots + f_v D^v \quad (5.10)$$

$$e(D): e(D) = e_0 + e_1 D + e_2 D^2 + \dots + e_p D^p \quad (5.11)$$

p : The degree of the polynomial $e(D)$.

r : $r = (p+v)$ is the degree of the polynomial $F(D)$.

$$F(D) = (e_0 + e_1 D + \dots + e_p D^p) (f_0 + f_1 D + \dots + f_v D^v)$$

F_j : $F_j = j$ th coefficient of the polynomial $F(D)$.

$$= \sum_{i=0}^p e_i f_{j-i} \quad (5.12)$$

$z(D)$: Output sequence after equalization

$$z(D) = x(D) F(D) + n(D) \quad (5.13)$$

$$n(D) = e(D) n'(D) \quad (5.14)$$

$n'(D)$: Noise sequence before equalization.

$n(D)$: Noise sequence at the output of the equalizer.

$$n_k: \quad n_k = \sum_{j=0}^p e_j n'_{k-j} \quad (5.15)$$

z_k : z_k = Received signal after equalization

$$= \sum_{j=0}^r F_j x_{k-j} + n_k \quad (5.16)$$

Substituting for F_j from (5.12), (5.16) becomes

$$z_k = \sum_{j=0}^r \left(\sum_{i=0}^p e_i f_{j-i} \right) x_{k-j} + \sum_{j=0}^p e_j n'_{k-j} \quad (5.17)$$

z_{kd} : z_{kd} = desired output after equalization

$$= x_k - x_{k-1} \quad (5.18)$$

E_k : $E_k = (z_k - z_{kd})$

$$= \left[\sum_{j=0}^r \left(\sum_{i=0}^p e_i f_{j-i} \right) x_{k-j} + \sum_{j=0}^p e_j n'_{k-j} - x_k + x_{k-1} \right] \quad (5.19)$$

E_R : $E_R = E(E_k^2)$

$$= E \left[\sum_{j=0}^r \left(\sum_{i=0}^p e_i f_{j-i} \right) x_{k-j} + \sum_{j=0}^p e_j n'_{k-j} - x_k + x_{k-1} \right]^2 \quad (5.20)$$

The problem is to minimize the mean-square error E_R by adjusting the taps of the equalizer. Differentiating the

expression (5.20) with respect to e_i and interchanging the order of expectation, we get

$$\frac{\partial E_R}{\partial e_i} = E \left[2(E_k) \left(\sum_{j=0}^r f_{j-i} x_{k-j} + n'_{k-i} \right) \right] \quad i = 0, 1, \dots, p \quad (5.21)$$

For $i = 0$, (5.21) reduces to

$$\frac{\partial E_R}{\partial e_0} = E \left[2(E_k) \left(\sum_{j=0}^r f_j x_{k-j} + n'_k \right) \right]$$

But

$$\sum_{j=0}^r f_j x_{k-j} + n'_k = \sum_{j=0}^v f_j x_{k-j} + n'_k + \sum_{j=v+1}^r f_j x_{k-j} \quad (5.22)$$

and

$$\sum_{j=v+1}^r f_j x_{k-j} = 0 \quad (\text{because } f_j = 0 \text{ for } j > v) \quad (5.23)$$

So (5.22) reduces to

$$\sum_{j=0}^v f_j x_{k-j} + n'_k = z'_k \quad (5.24)$$

where z'_k is the actual received signal obtained before equalization.

$$\frac{\partial E_R}{\partial e_0} = E(2 E_k z'_k) \quad (5.25)$$

Similarly

$$\frac{\partial E_R}{\partial e_1} = E \left[2 E_k \left(\sum_{j=0}^r f_{j-1} x_{k-j} + n'_{k-1} \right) \right] \quad (5.26)$$

$$\begin{aligned}
\sum_{j=0}^r f_{j-1} x_{k-j} + n'_{k-1} &= \sum_{j=1}^r f_{j-1} x_{k-j} + n'_{k-1} \\
&= \sum_{j=1}^{v+1} f_{j-1} x_{k-j} + \sum_{j=v+2}^r f_{j-1} x_{k-j} + n'_{k-1}
\end{aligned} \tag{5.27}$$

Let $j = q+1$,

$$\sum_{q=0}^{r-1} f_q x_{k-q-1} + n'_{k-1} = \sum_{q=0}^v f_q x_{k-q-1} + n'_{k-1}$$

So in general,

$$\frac{\partial E_R}{\partial e_j} = E(2 E_k z'_{k-j}) \tag{5.28}$$

The recursive relation to be used in the adjustment of the taps of the equalizer is

$$e_j^{k+1} = e_j^k - \frac{\Delta}{2} \frac{\partial E_R}{\partial e_j} \quad j = 0, 1, \dots, p \tag{5.29}$$

It is not possible to calculate $E(E_k z'_{k-j})$ because this calculation depends on the channel response which in general is not known. To overcome this difficulty, one can use the unbiased estimate of the gradient as suggested by Widrow [45]. Using the unbiased estimate the recursive relation (5.29) becomes

$$\begin{aligned}
 e_j^{k+1} &= e_j^k - \frac{\Delta}{2}(2)E_k z_{k-j}' \\
 &= e_j^k - \Delta E_k z_{k-j}'
 \end{aligned}
 \qquad j = 0, 1, \dots, p$$

(5.30)

where Δ is the adjustment parameter.

The complete diagram for the adjustment of the tap gains of the equalizer is shown in Figure 5.5.

5.4 Simulation Results Obtained for an Adaptive Equalization Scheme

The simulation results were obtained for two types of channels. The sampled responses for these two channels is given by

$$\begin{aligned}
 f(D) &= 1 - D + 0.5 D^2 \\
 f(D) &= 1 - 0.5 D - 0.5 D^2
 \end{aligned}$$

The equalizer had 11 taps and the main tap was set equal to 1 for start-up operation. 1000 symbols were used in order to adjust the equalizer tap gains before the receiver started making decisions. The simulated probability of error was calculated by actually counting the number of errors in 10,000 symbols. The lower bound for the probability of error was assumed to be the probability of error with no intersymbol interference. The upper

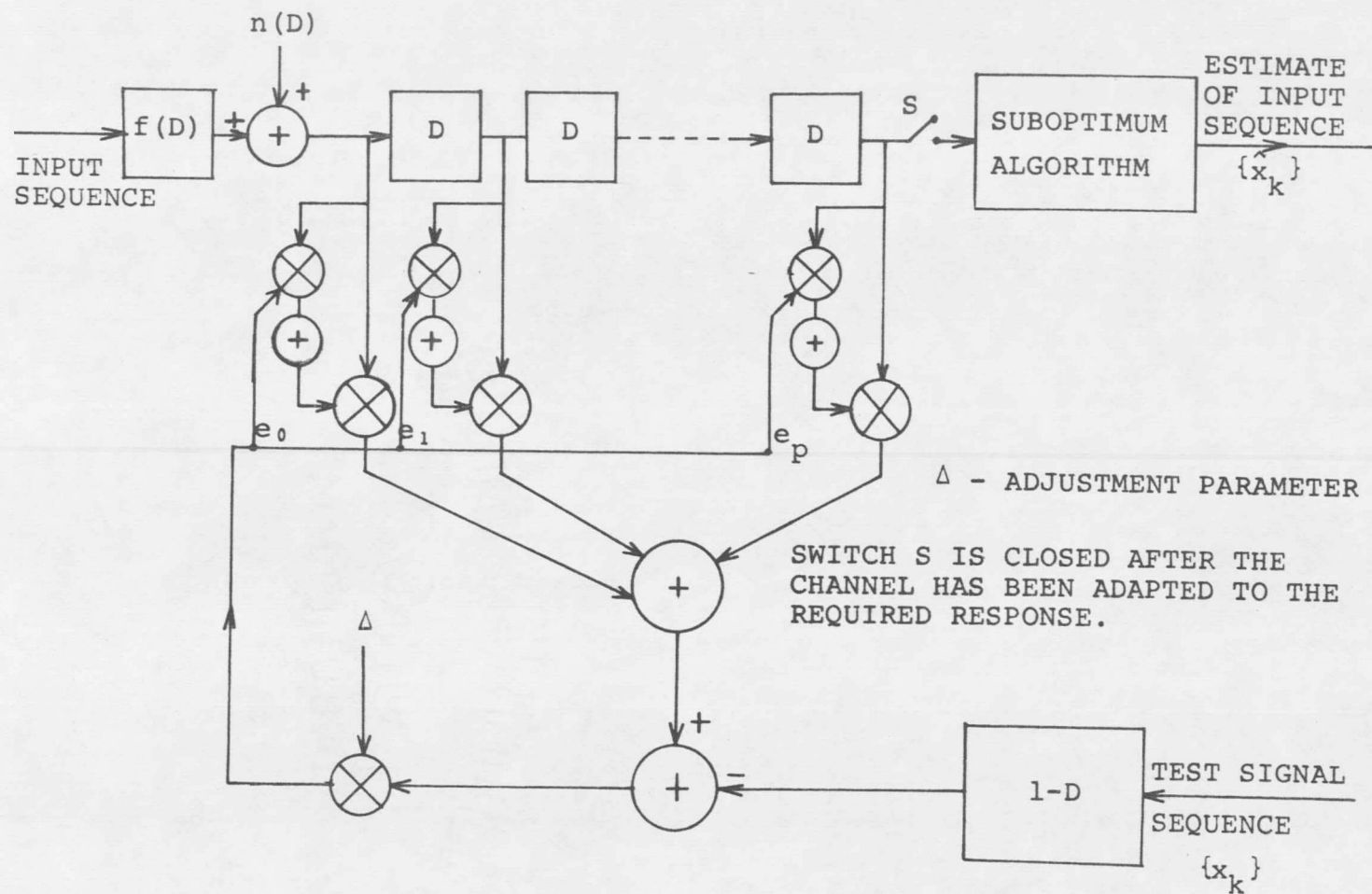


Figure 5.5 Suboptimum algorithm with equalizer.

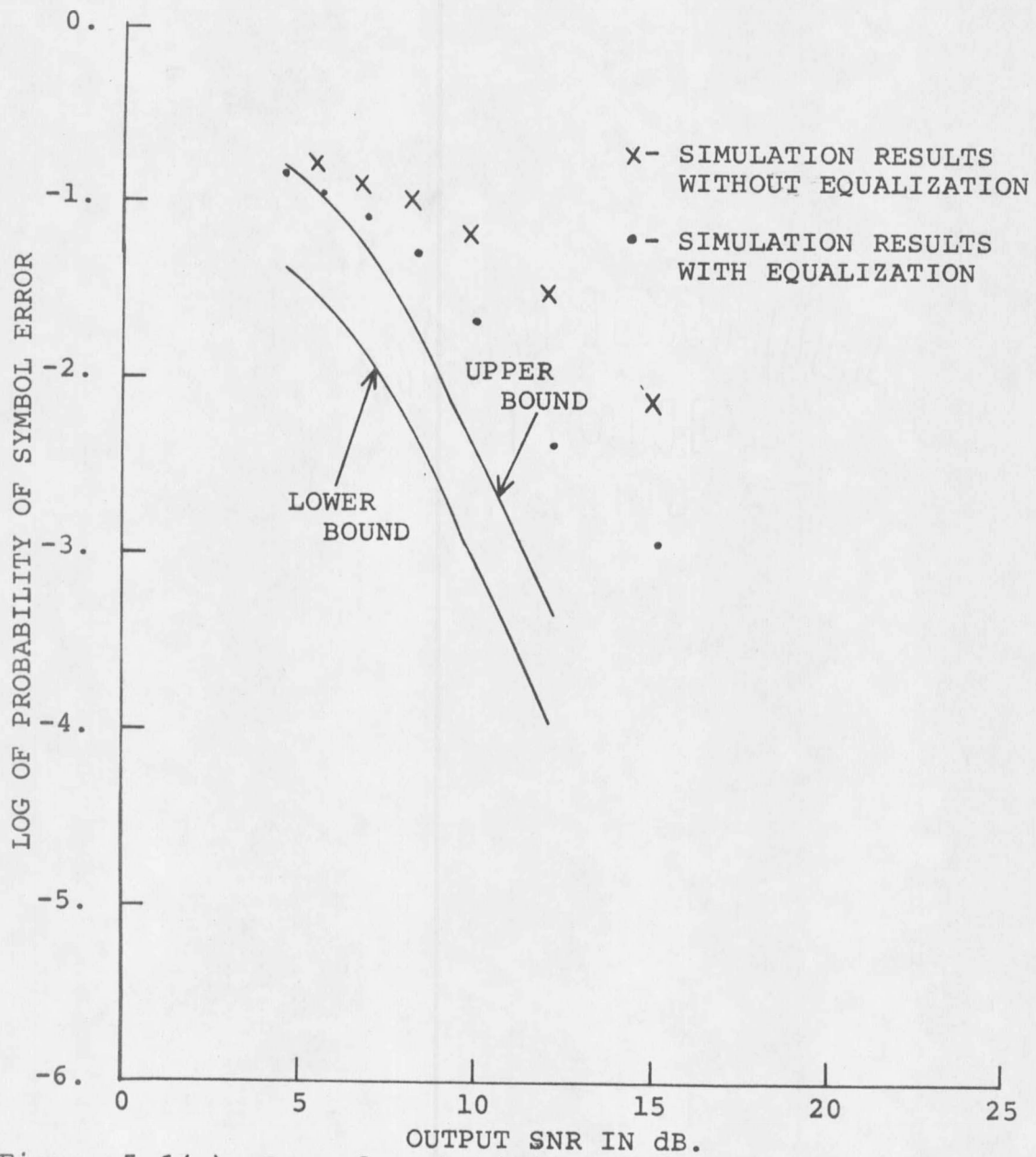


Figure 5.6(a) Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1 - D + 0.5D^2$.

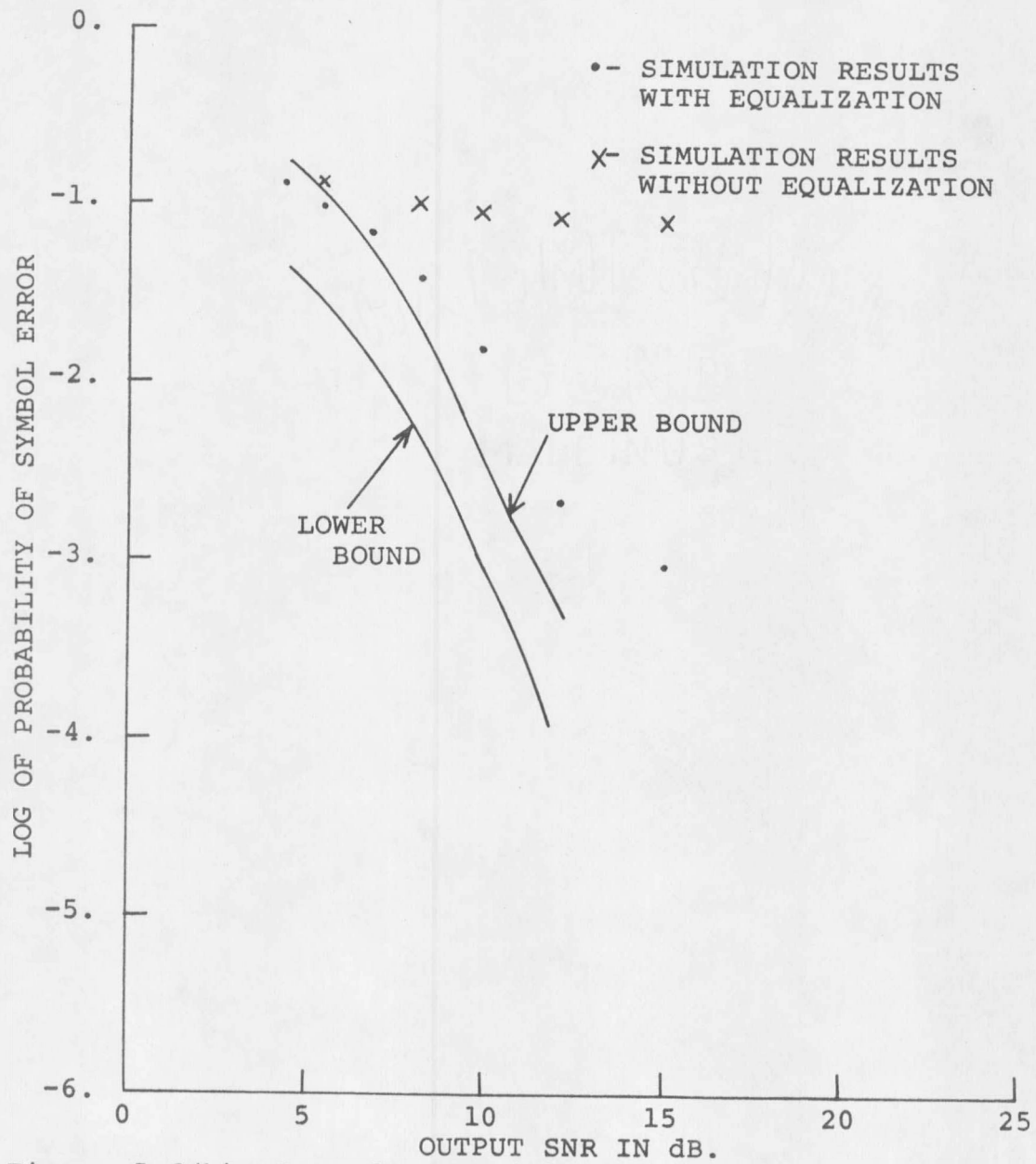


Figure 5.6(b) Log of probability of symbol error vs. output SNR in dB. for $f(D) = 1 - 0.5D - 0.5D^2$.

bound for probability of error was taken to be one derived by Forney [22]. In actual practice, the probability of symbol error will be somewhat higher than given by the above upper bound because the equalizer will not be perfect in having the desired response. Moreover, the equalizer introduces some additional intersymbol interference and it also enhances the noise.

The results were plotted with log of probability of error versus output signal-to-noise ratio. The simulation results are shown by dots in Figures 5.6(a) and 5.6(b). The crosses on the plots show the results without equalization. Thus, it is obvious from Figures 5.6(a) and 5.6(b) that equalization helps a great deal in reducing the probability of error.

Chapter VI

SUMMARY AND SUGGESTED FUTURE RESEARCH

6.1 Introduction

This chapter summarizes the results obtained and describes the conclusions drawn from the research conducted. The significance of the findings of this thesis is discussed, and possible directions for the extension of the work are mentioned.

6.2 Summary and Conclusions

The general problem of intersymbol interference in pulse transmission is of considerable practical interest and has received much attention in this thesis. Most of the attempts to combat the problem generally fell short of optimum performance until recently when the applicability of the Viterbi algorithm was recognized. The superiority of Forney's [22] optimum receiver over the other nonlinear and linear receivers was discussed in this thesis, and it was indicated that Forney's [22] receiver had a great potential for combating the problem of intersymbol interference. The discrete-time model developed by Forney [22] was presented in detail, and it was shown that this discrete model was more suitable than the continuous-time model

Maximum-likelihood techniques for sequence estimation were discussed, and the usefulness of the Viterbi algorithm for the task of sequence estimation was described. Maximum-likelihood sequence estimation using the Viterbi algorithm was later proven to be asymptotically optimum. Some adaptive maximum-likelihood sequence estimation receivers were presented and their relative performance and complexity were discussed.

An adaptive receiver structure proposed by Proakis and Magee [31] had a channel estimator to be used in combination with the Viterbi algorithm. This channel estimator suffered from the disadvantage of being slow in convergence to the optimum channel pulse response. Two algorithms for channel estimation were developed in this thesis and which were shown to converge more rapidly than the fixed-step-size algorithm used by Proakis and Magee [31] in their adaptive receiver structure. Simulation results obtained showed the superiority of these algorithms.

Finally, Forney's [22] suboptimum algorithm for sequence estimation was described; it became apparent from simulation results that small changes in $f(D)$ did not cause a significant loss in the performance of this algorithm. An adaptive version of the suboptimum algorithm

was developed; the results indicated that the equalization scheme did reduce the probability of error significantly.

The following conclusions were reached during the conduct of this research:

(1) Forney's [22] optimum receiver has superceded all the other linear and nonlinear receivers for the detection of digital signals in the presence of intersymbol interference and noise. The work performed on symbol-by-symbol nonlinear detection techniques was nonproductive.

(2) An adaptive MLSE receiver structure using a linear equalizer is found to be more useful for highly dispersive channels than the adaptive receiver suggested by Proakis and Magee [31]. But the adaptive receiver using channel estimator [31] is asymptotically optimum and less complex for channels with lengths of 2 or 3. Also, an adaptive receiver using a linear equalizer is suboptimum, but more readily implemented. The linear equalizer enhances noise near nulls.

(3) Two faster converging algorithms for channel estimation were derived. The second-order algorithm performs better than the first-order algorithm, while the first-order algorithm works much better than the fixed-step-size algorithm.

(4) Forney's suboptimum algorithm performance is not degraded much by the small fluctuations in the channel impulse response. An adaptive equalization scheme used in combination with this suboptimum algorithm helps a great deal in reducing the probability of symbol error for channels whose pulse response is different from $1 + D^n$.

6.3 Suggestions for Future Research

The following areas appear to be fruitful for future work:

(1) It would be desirable to have simplified or approximate forms for the Viterbi algorithm for channels having response other than $1 + D^n$ in order to enhance the economic feasibility of the BA receiver structure.

(2) Some additional work on sequential receiver structures, such as the results of Ungerboeck [14] is in order to see if simplified adaptive realizations of these structures are possible.

(3) The work of Proakis and Magee [31] assumes that an adaptive realization of a whitened matched filter is used to track channel response and that this tracking is exact. They then concentrate on adjusting the parameters of the transversal filter $f(D)$. What happens when these

assumptions are not met? How does this affect the performance of the MLSE receiver?

(4) Forney's [22] work is based on the fact that the channel impulse response is of finite duration. It would be interesting to extend Forney's work to channels with infinite-duration impulse responses.

(5) It should be possible to develop a variable-step-size algorithm for channel estimation using dynamic programming [28] as a tool for selecting the step sizes $\{\alpha_i\}$ by imposing constraints on the average excess mean-squared error and the number of iterations.

APPENDICES

APPENDIX A

DERIVATION OF EQUATIONS (3.22), (4.3), (4.8), AND (4.19)

It follows from (4.1) that MSE is given by

$$\epsilon(\text{MSE}) = E(z_k - \hat{z}_k)^2 \quad (1)$$

where

$$z_k = n_k + \sum_{j=0}^V f_j x_{k-j}$$

and

$$\hat{z}_k = \sum_{j=0}^W g_j x_{k-j}$$

Substituting the relations for z_k and $\hat{z}_k$ into (1) and we get

$$\epsilon(\text{MSE}) = E \left[\left(n_k + \sum_{j=0}^V f_j x_{k-j} - \sum_{j=0}^W g_j x_{k-j} \right)^2 \right] \quad (2)$$

Differentiating (2) with respect to g_i , the relation obtained is

$$\frac{\partial \epsilon}{\partial g_i} = E \left\{ \left(n_k + \sum_{j=0}^V f_j x_{k-j} - \sum_{j=0}^W g_j x_{k-j} \right) (-2 x_{k-i}) \right\} \quad (3)$$

$$\frac{\partial \epsilon}{\partial g_i} = 0 \text{ for minimum mean-square error.}$$

$$E\left\{ \left(n_k + \sum_{j=0}^V f_j x_{k-j} - \sum_{j=0}^W g_j x_{k-j} \right) (-2 x_{k-i}) \right\} = 0$$

$$(-2) E\left\{ n_k x_{k-i} + \left(\sum_{j=0}^V f_j x_{k-j} \right) x_{k-i} - \left(\sum_{j=0}^W g_j x_{k-j} \right) x_{k-i} \right\} = 0$$

$$E\left\{ n_k x_{k-i} + \left(\sum_{j=0}^V f_j x_{k-j} \right) x_{k-i} - \left(\sum_{j=0}^W g_j x_{k-j} \right) x_{k-i} \right\} = 0 \quad (4)$$

Assumptions:

1. The noise sequence and information sequence are not correlated.
2. The information sequence $\{x_k\}$ is uncorrelated.

Using the above assumptions, (4) reduces to

$$\sum_{j=0}^V f_j E(x_{k-j} x_{k-i}) = \sum_{j=0}^W g_j E(x_{k-j} x_{k-i}) \quad (5)$$

$$\text{But } E(x_{k-j} x_{k-i}) = \phi_{ji}$$

Thus, relation (5) reduces to

$$\begin{aligned} \sum_{j=0}^W g_j \phi_{ji} &= \sum_{j=0}^V f_j \phi_{ji} \\ &= B_i \end{aligned} \quad (6)$$

$$i = 0, 1, 2, \dots, w$$

where $B_i = \sum_{j=0}^V f_j \phi_{ji}$

Equation (6) is the same as (4.3) and (3.22).

Equation (3) can also be put in other form as shown below

$$\begin{aligned} \frac{\partial \epsilon}{\partial g_i} &= E\left\{ \left(n_k + \sum_{j=0}^V f_j x_{k-j} - \sum_{j=0}^W g_j x_{k-j} \right) (-2x_{k-i}) \right\} \\ &= E(z_k - \hat{z}_k) (-2x_{k-i}) \\ &= E(-2 e^k x_{k-i}) \end{aligned} \quad (7)$$

Since it is hard to find the expected value of the quantity on the right side of equation (7), an unbiased estimate of the gradient is used. Therefore, (7) becomes

$$\frac{\partial \epsilon}{\partial g_i} = -2 e^k x_{k-i} \quad (8)$$

Furthermore, it is assumed that the receiver is making correct decisions; using this fact in (8), we get

$$\frac{\partial \epsilon}{\partial g_i} = -2 e^k \hat{x}_{k-i} \quad (9)$$

Equation (3) is written here again to derive relation (4.19)

$$\begin{aligned}
 &= E\left\{\left(n_k + \sum_{j=0}^V f_j x_{k-j} - \sum_{j=0}^W g_j x_{k-j}\right)(-2 x_{k-i})\right\} \\
 &= -2E\left\{\left(\sum_{j=0}^V f_j x_{k-j}\right)(x_{k-i}) - \left(\sum_{j=0}^W g_j x_{k-j}\right)(x_{k-i})\right\} \\
 &= -2\left(\sum_{j=0}^V f_j \phi_{ji} - \sum_{j=0}^W g_j \phi_{ji}\right) \\
 &= -2\left(B_i - \sum_{j=0}^W g_j \phi_{ji}\right) \quad (10)
 \end{aligned}$$

In matrix notation (10) can be written as

$$\frac{\partial \epsilon}{\partial G} = -2(B - \Phi G^k) \quad (11)$$

Equation (11) is exactly the same as (4.19), and hence (4.19) is derived.

APPENDIX B

DERIVATION OF EQUATIONS (4.47-4.50)

(4.44) and (4.45) give the following relations:

$$P_{k+1}(\Phi) = \left[(1 + \beta_k) I - \alpha_k \Phi \right] P_k(\Phi) - \beta_k P_{k-1}(\Phi) \quad (1)$$

$$P_k(\) = \frac{T_k(\gamma I - \frac{2}{\lambda_u - \lambda_\ell} \Phi)}{(T_k(\gamma))} \quad (2)$$

$$\gamma = \frac{\lambda_u + \lambda_\ell}{\lambda_u - \lambda_\ell} \quad (3)$$

Substituting (3) into (2) and letting $A = \gamma I - \frac{2}{\lambda_u - \lambda_\ell} \Phi$, we get

$$\begin{aligned} \frac{T_{k+1}(\gamma I - \frac{2}{\lambda_u - \lambda_\ell} \Phi)}{T_{k+1}(\gamma)} &= \left[(1 + \beta_k) I - \alpha_k \Phi \right] \frac{T_k(\gamma I - \frac{2}{\lambda_u - \lambda_\ell} \Phi)}{T_k(\gamma)} - \beta_k \frac{T_{k-1}(\gamma I - \frac{2}{\lambda_u - \lambda_\ell} \Phi)}{T_{k-1}(\gamma)} \\ &= \left[(1 + \beta_k) I - \alpha_k \Phi \right] \frac{T_k(A)}{T_k(\gamma)} - \beta_k \frac{T_{k-1}(A)}{T_{k-1}(\gamma)} \\ T_{k+1}(A) &= \left[(1 + \beta_k) I - \alpha_k \Phi \right] \frac{T_k(A)}{T_k(\gamma)} \cdot T_{k+1}(\gamma) - \beta_k \frac{T_{k-1}(A)}{T_{k-1}(\gamma)} \cdot T_{k+1}(\gamma) \end{aligned} \quad (4)$$

If $T_k(A)$ is to be Chebyshev polynomial,

$$T_{k+1}(A) = 2 A T_k(A) - T_{k-1}(A) \quad (5)$$

Comparing (4) and (5), we get the following relations:

$$\beta_k \cdot \frac{T_{k+1}(\gamma)}{T_{k-1}(\gamma)} = 1$$

$$\beta_k = \frac{T_{k-1}(\gamma)}{T_{k+1}(\gamma)} \quad (6)$$

$$[(1 + \beta_k)I - \alpha_k \Phi] \frac{T_{k+1}(\gamma)}{T_k(\gamma)} = 2 A$$

or

$$(1 + \beta_k)I - \alpha_k \Phi = 2 A \frac{T_k(\gamma)}{T_{k+1}(\gamma)} \quad (7)$$

We know that

$$A = \gamma I - \frac{2}{\lambda_u - \lambda_\ell} \Phi$$

$$A - \gamma I = - \frac{2}{\lambda_u - \lambda_\ell} \Phi$$

$$\Phi = \frac{(\gamma I - A)(\lambda_u - \lambda_\ell)}{2} \quad (8)$$

Substituting (8) into (7),

$$(1 + \beta_k)I + \frac{\alpha_k}{2} A (\lambda_u - \lambda_\ell) - \frac{\alpha_k}{2} (\lambda_u - \lambda_\ell) \gamma I = 2 A \frac{T_k(\gamma)}{T_{k+1}(\gamma)}$$

Comparing terms on both sides of above equation, we get

$$\frac{\alpha_k}{2} (\lambda_u - \lambda_\ell) = \frac{2 T_k(\gamma)}{T_{k+1}(\gamma)}$$

$$\alpha_k = \frac{4}{(\lambda_u - \lambda_\ell)} \cdot \frac{T_k(\gamma)}{T_{k+1}(\gamma)} \quad (9)$$

$$(1 + \beta_k) = \frac{\alpha_k}{2} (\lambda_u - \lambda_\ell) \gamma$$

$$\begin{aligned} \alpha_k &= \frac{2}{\lambda_u - \lambda_\ell} (1 + \beta_k) \left(\frac{\lambda_u - \lambda_\ell}{\lambda_u + \lambda_\ell} \right) \\ &= \frac{2}{\lambda_u + \lambda_\ell} (1 + \beta_k) \\ &= \alpha_0 (1 + \beta_k) \end{aligned} \quad (10)$$

where

$$\alpha_0 = \frac{2}{\lambda_u + \lambda_\ell} \quad (11)$$

Equations (6) and (10) verify (4.47) and (4.48).

In order to verify (4.49), we start with (6)

$$\beta_{k+1} = \frac{T_k(\gamma)}{T_{k+2}(\gamma)}$$

But we know that $T_{k+2}(\gamma) = 2 T_{k+1}(\gamma) - T_k(\gamma)$, and therefore (6) reduces to

$$\begin{aligned}
 \beta_{k+1} &= \frac{T_k(\gamma)}{2\gamma T_{k+1}(\gamma) - T_k(\gamma)} \\
 &= \frac{1}{2\gamma \frac{T_{k+1}(\gamma)}{T_k(\gamma)} - 1}
 \end{aligned} \tag{12}$$

But

$$\begin{aligned}
 \frac{T_k(\gamma)}{T_{k+1}(\gamma)} &= \frac{\alpha_0}{4} (1 + \beta_k) (\lambda_u - \lambda_\ell) \\
 &= \frac{2}{(\lambda_u + \lambda_\ell)} \cdot \frac{(1 + \beta_k) (\lambda_u - \lambda_\ell)}{4} \\
 &= \frac{1}{2\gamma} (1 + \beta_k)
 \end{aligned}$$

Therefore,

$$\frac{T_{k+1}(\gamma)}{T_k(\gamma)} = \frac{2\gamma}{1 + \beta_k} \tag{13}$$

$$\begin{aligned}
 \beta_{k+1} &= \frac{1}{2\gamma \left(\frac{2\gamma}{1 + \beta_k} \right) - 1} \\
 &= \frac{1 + \beta_k}{4\gamma^2 - 1 - \beta_k}
 \end{aligned} \tag{14}$$

$$\beta_1 = \frac{T_0(\gamma)}{T_2(\gamma)} = \frac{1}{2\gamma^2 - 1} \tag{15}$$

APPENDIX C

ANALYSIS FOR RATE OF CONVERGENCE OF A VARIABLE-STEP-SIZE ALGORITHM

It follows from (4.25) that

$$\| E^M \| = \| P_M(\Phi) E^0 \| \quad (1)$$

and

$$\| E_M \| \leq \| P_M(\Phi) \| \cdot \| E^0 \| \quad (2)$$

where

$$\| P_M(\Phi) \| \leq \max_{\lambda_l \leq x \leq \lambda_u} | P_M(x) | \quad (3)$$

$$\leq \max_{\lambda_l \leq x \leq \lambda_u} \left| \frac{T_M\left(\frac{\lambda_u + \lambda_l - 2x}{\lambda_u - \lambda_l}\right)}{T_M\left(\frac{\lambda_u + \lambda_l}{\lambda_u - \lambda_l}\right)} \right| \quad (4)$$

But

$$\left| T_M\left(\frac{\lambda_u + \lambda_l - 2x}{\lambda_u - \lambda_l}\right) \right| \leq 1 \quad \lambda_l \leq x \leq \lambda_u$$

Thus (4) takes the form as

$$\| P_M(\Phi) \| \leq \frac{1}{T_M\left(\frac{\lambda_u + \lambda_l}{\lambda_u - \lambda_l}\right)} \quad (5)$$

Let $\frac{\lambda_u}{\lambda_l} = a$ and (5) reduces to

$$\|P_M(\Phi)\| \leq \frac{1}{T_M\left(\frac{a+1}{a-1}\right)} \quad (6)$$

But

$$T_M\left(\frac{a+1}{a-1}\right) = \cosh Mz = \frac{1}{2}(e^{Mz} + e^{-Mz}) \quad (7)$$

where

$$z = \cosh^{-1}\left(\frac{a+1}{a-1}\right) \quad (8)$$

Solving $\cosh z = \frac{a+1}{a-1}$ for e^z gives

$$e^z = \frac{a^{\frac{1}{2}} - 1}{a^{\frac{1}{2}} + 1} \quad (9)$$

Substituting for e^z in (7), we get

$$T_M\left(\frac{a+1}{a-1}\right) = \frac{1}{2}(\rho^M + \rho^{-M}) \quad (10)$$

where

$$\rho = \frac{a^{\frac{1}{2}} - 1}{a^{\frac{1}{2}} + 1} \quad (11)$$

Thus, $P_M(\Phi)$ can be written in the form

$$\|P_M(\Phi)\| \leq \frac{2}{(\rho^M + \rho^{-M})} \quad (12)$$

Using the above results for $P_M(\Phi)$ in (2), we get

$$\frac{\|E^M\|}{\|E^0\|} \leq \frac{2}{(\rho^M + \rho^{-M})} \quad (13)$$

It is obvious from (13) that with the increase in the number of iterations, the norm of the tap-gain error decreases. The norm of the tap-gain error will approach zero as M becomes larger.

APPENDIX D

METHOD OF COMPUTER SIMULATION

All the computer simulations reported in this thesis were done at the Montana State University computation center using an XDS Sigma 7 computer. The simulations were implemented by writing Fortran statements for the channel model of Figure 2.4, and feeding it a randomly generated information sequence, a set of independent Gaussian random numbers for the noise sequence, and a chosen set of fixed values $\{f_i\}$ for the channel pulse response. The output of the channel model fed directly into the receiver which was simulated by writing Fortran statements for the particular operations involved. The error rates reported were for 10,000 information symbols after the receiver had adapted to the channel.

The information symbols were generated by using the random number generator "RANDOM" and using +1 for the symbol if the random number was greater than 0.5, and 0 if the random number was less than 0.5. The Gaussian noise was generated by using a subroutine "GAUSS" available in one of the files in the Sigma 7 system. This subroutine makes use of a set of random numbers. These numbers were

generated by using the random number generated described above, and by adding 12 of the numbers together to form each Gaussian random number. The Gaussian numbers were generated according to the equation

$$n = R + \sigma(-6 + \sum_{i=1}^{12} X_i)$$

where R is the mean and σ^2 is the variance of noise and X_i are the random numbers.

The listings of the program are given on the next page.

```

C *****
C FIRST=ORDER ALGORITHM
C *****
C
C THIS PROGRAM IS FOR RAPID CHANNEL ESTIMATION
C USING FIRST=ORDER ALGORITHM.
C THIS ALGORITHM MAKES USE OF A VARIABLE STEP SIZE.
C THE FIRST=ORDER ALGORITHM IS OF THE FORM
C  $FV(J) = FV(J) + DELTA * ERROR(K) * X(K=J+1)$ 
C WHERE DELTA IS A VARIABLE STEP SIZE.
C THIS ALGORITHM GUARANTEES IMPROVEMENT AT THE
C END OF A FIXED NUMBER OF ITERATIONS.
C THIS PROGRAM WORKS FOR CHANNEL LENGTH EQUAL TO 5.
C THIS PROGRAM MAKES USE OF SUBROUTINES, GAUSS, RANDOM,
C AND DRAW, AVAILABLE IN THE SYSTEM OF SIGMA 7.
C THIS PROGRAM PLOTS THE NORM OF THE ERROR VECTOR
C VERSUS NUMBER OF ITERATIONS.
C SNR IS THE INPUT SIGNAL=TO=NOISE RATIO.
C TEN PERCENT OF THE ANALYTIC VALUE OF DELTA
C IS USED IN THE ALGORITHM.
C THIS IS DONE IN ORDER TO TAKE INTO ACCOUNT THE
C EFFECT OF NOISE AND NOISY ESTIMATES OF THE GRADIENTS.
C 'LABEL' IS A CURVE IDENTIFIER.
C 'IT' IS AN ARRAY OF 24 CHARACTER HOLLERITH
C WORDS, THE FIRST 12 OF WHICH WILL FORM THE
C FIRST TITLE LINE AND THE LAST 12 THE SECOND
C TITLE LINE. THE ARRAY MUST BE DIMENSIONED
C AT LEAST 24 IN THE CALLING PROGRAM, MUST CONTAIN
C THE USERS NAME, AND MUST LEAVE UNWANTED
C CHARACTERS SET TO BLANK.
C 'LAST' INDICATES TO DRAW PROGRAM THAT
C THIS IS THE LAST CALL TO DRAW TO BE MADE
C IN THE USERS PROGRAM. IN THE VERY LAST
C CALL TO DRAW LAST SHOULD BE SET TO 4HLAST.
C THE SIZE OF THE GRAPH IS 8X8 INCHES.

```

```

C      IP AND IX ARE INTEGER NUMBERS CHOSEN ARBITRARILY.
C      ***** DATA CARDS REQUIRED *****
C      DATA SET
C      FIRST CARD FORMAT (2I3)
C      FIRST FIELD: M, THE DEGREE OF CHEBYCHEV POLYNOMIAL.
C      SECOND FIELD: L, THE NUMBER OF TIMES THE ITERATIVE
C      PROCESS IS TO BE REPEATED.
C      SECOND CARD: SAMPLED CHANNEL RESPONSE
C      TO BE ESTIMATED: FORMAT(5F10.5)
C      THIRD CARD: FOR THE PURPOSE OF IDENTIFICATION
C      OF THE PLOT
C      FOURTH CARD: NAME AND ACCOUNT NUMBER
C      *****
C      DIMENSION ERF(100),ERV(100),IT(24)
C      REAL ITER(100)
C      DIMENSION EOF(100),EOV(100)
C      DIMENSION YF(100),YV(100)
C      INTEGER X(100)
C      DIMENSION Y(100),Z(100),FF(10),FV(10)
C      DIMENSION F(10)
C      DIMENSION EF(100),EV(100)
C      DO 57 KK=1,10
C      READ(105,41) M,L
41      FORMAT(2I3)
C      READ(105,14) (F(I),I=1,5)
14      FORMAT(5F10.5)
C      READ(105,16) (IT(I),I=1,24)
16      FORMAT(12A4/12A4)
C      EIGEN IS THE UPPER BOUND ON THE EIGEN VALUE
C      OF THE COVARIANCE MATRIX OF THE INPUT SEQUENCE.
C      EIGEN=5.0
C      *****
C      S IS THE STANDARD DEVIATION OF THE NOISE.
C      *****
C      IP=99656
C      S=0.001

```

```

C      SNR=(10.)*ALOG10(0.125/(S**2))
C      *****
C      AM IS THE MEAN OF THE RANDOM NOISE.
C      *****

      AM=0.
      IX=65499
      DO 18 I4=1,5
      FF(I4)=0.0
      FV(I4)=0.0
18      CONTINUE
      FV(1)=0.5
      FF(1)=0.5
      DO 55 J4=1,L
      ERF(J4)=0.
      ERV(J4)=0.
55      CONTINUE
      A=F(2)*F(2)+F(3)*F(3)+F(4)*F(4)+F(5)*F(5)
      EOF(1)=(F(1)-FF(1))*2*A
      EOV(1)=(F(1)-FV(1))*2*A
      EOF(1)=SQRT(EOF(1))
      EOV(1)=SQRT(EOV(1))
      ITER(1)=0.
      ERV(1)=EOV(1)
      ERF(1)=EOF(1)
      ITER(1)=0.0
      L1=0
      DO 20 JJ=1,L
      DO 5 JJ=1,M
C      *****
C      NOW THE RANDOM INPUT SEQUENCE IS GENERATED.
C      *****
      CALL RANDOM(IX,IY,YFL)
      IF(YFL=0.5) 2,2,3
2      X(J)=1
      GO TO 4

```

```

3  X(J)=1
4  IX=IY
5  CONTINUE
   DO 32 KA=1,M
32  X(KA)=FLOAT(X(KA))
     Y(1)=X(1)*F(1)
     Y(2)=X(2)*F(1)+X(1)*F(2)
     Y(3)=X(3)*F(1)+X(2)*F(2)+X(1)*F(3)
     Y(4)=X(4)*F(1)+X(3)*F(2)+X(2)*F(3)+X(1)*F(4)
     DO 6 K=5,M
6    Y(K)=X(K)*F(1)+X(K-1)*F(2)+X(K-2)*F(3)+X(K-3)*F(4)+X(K-4)*F(5)
     DO 8 I1=1,M
C    *****
C    V IS THE RANDOM NOISE GENERATED.
C    *****
     CALL GAUSS(IP,S,AM,V)
     Z(I1)=Y(I1)+V
8    CONTINUE
     PI=3.1428
C    *****
C    'STEP' IS THE FIXED STEP SIZE.
C    *****
     STEP=0.01
     YV(1)=X(1)*FV(1)
     YF(1)=X(1)*FF(1)
     EV(1)=Z(1)-YV(1)
     EF(1)=Z(1)-YF(1)
     ARG=3.1428/(4.*FLOAT(M))
     DELTA=1.0/((SIN(ARG))**2)
     DELTA=0.1*DELTA/EIGEN
     FF(1)=FF(1)+STEP*EF(1)*X(1)
     FV(1)=FV(1)+DELTA*EV(1)*X(1)
     L2=L1+2
     RV=(FV(1)-F(1))**2+(FV(2)-F(2))**2+(FV(3)-F(3))**2
     * +(FV(4)-F(4))**2+(FV(5)-F(5))**2
     RF=(FF(1)-F(1))**2+(FF(2)-F(2))**2+(FF(3)-F(3))**2

```

```

* +(FF(4)=F(4))*2+(FF(5)=F(5))*2
ERV(L2)=SQRT(RV)/(EOV(1))
ERF(L2)=SQRT(RF)/(EOF(1))
ITER(L2)=FLOAT(L2=1)
YV(2)=X(2)*FV(1)+X(1)*FV(2)
YF(2)=X(2)*FF(1)+X(1)*FF(2)
EV(2)=Z(2)=YV(2)
EF(2)=Z(2)=YF(2)
ARG=(3.141592653589793)/(4.*FLOAT(M))
DELTA=1.0/((SIN(ARG))*2)
DELTA=0.1*DELTA/EIGEN
DO 11 I5=1,2
FF(I5)=FF(I5)+STEP*EF(2)*X(2=I5+1)
FV(I5)=FV(I5)+DELTA*EV(2)*X(2=I5+1)
11 CONTINUE
L2=L1+3
RV=(FV(1)=F(1))*2+(FV(2)=F(2))*2+(FV(3)=F(3))*2
* +(FV(4)=F(4))*2+(FV(5)=F(5))*2
RF=(FF(1)=F(1))*2+(FF(2)=F(2))*2+(FF(3)=F(3))*2
* +(FF(4)=F(4))*2+(FF(5)=F(5))*2
ERF(L2)=SQRT(RF)/(EOF(1))
ERV(L2)=SQRT(RV)/(EOV(1))
ITER(L2)=FLOAT(L2=1)
YV(3)=X(3)*FV(1)+X(2)*FV(2)+X(1)*FV(3)
YF(3)=X(3)*FF(1)+X(2)*FF(2)+X(1)*FF(3)
EV(3)=Z(3)=YV(3)
EF(3)=Z(3)=YF(3)
ARG=(5.168083440144605)/(4.*FLOAT(M))
DELTA=1.0/((SIN(ARG))*2)
DELTA=0.1*DELTA/EIGEN
DO 12 I6=1,3
FF(I6)=FF(I6)+STEP*EF(3)*X(3=I6+1)
FV(I6)=FV(I6)+DELTA*EV(3)*X(3=I6+1)
12 CONTINUE
L2=L1+4
RV=(FV(1)=F(1))*2+(FV(2)=F(2))*2+(FV(3)=F(3))*2

```

```

* +(FV(4)*F(4))*2+(FV(5)*F(5))*2
RF=(FF(1)*F(1))*2+(FF(2)*F(2))*2+(FF(3)*F(3))*2
* +(FF(4)*F(4))*2+(FF(5)*F(5))*2
ERF(L2)=SQRT(RF)/(EOF(1))
ERV(L2)=SQRT(RV)/(EOV(1))
ITER(L2)=FLOAT(L2-1)
YV(4)=X(4)*FV(1)+X(3)*FV(2)+X(2)*FV(3)+X(1)*FV(4)
YF(4)=X(4)*FF(1)+X(3)*FF(2)+X(2)*FF(3)+X(1)*FF(4)
EV(4)=Z(4)=YV(4)
EF(4)=Z(4)=YF(4)
ARG=(7.*3.1428)/(4.*FLOAT(M))
DELTA=1.0/((SIN(ARG))*2)
DELTA=0.1*DELTA/EIGEN
DO 19 I7=1,4
FF(I7)=FF(I7)+STEP*EF(4)*X(4-I7+1)
FV(I7)=FV(I7)+DELTA*EV(4)*X(4-I7+1)

```

19 CONTINUE

```

L2=L1+5
RV=(FV(1)*F(1))*2+(FV(2)*F(2))*2+(FV(3)*F(3))*2
* +(FV(4)*F(4))*2+(FV(5)*F(5))*2
RF=(FF(1)*F(1))*2+(FF(2)*F(2))*2+(FF(3)*F(3))*2
* +(FF(4)*F(4))*2+(FF(5)*F(5))*2
ERF(L2)=SQRT(RF)/(EOF(1))
ERV(L2)=SQRT(RV)/(EOV(1))
ITER(L2)=FLOAT(L2-1)
DO 9 I2=5,M
YV(I2)=X(I2)*FV(1)+X(I2-1)*FV(2)+X(I2-2)*FV(3)+X(I2-3)*FV(4)
+X(I2-4)*FV(5)
EV(I2)=Z(I2)=YV(I2)
YF(I2)=X(I2)*FF(1)+X(I2-1)*FF(2)+X(I2-2)*FF(3)+
+X(I2-3)*FF(4)+X(I2-4)*FF(5)
EF(I2)=Z(I2)=YF(I2)
ARG=(2.*FLOAT(I2)-1.)*(PI)/(4.*FLOAT(M))
DELTA=1.0/((SIN(ARG))*2)
DELTA=0.1*DELTA/EIGEN
DO 10 I3=1,5

```

```

      FF(I3)=FF(I3)+STEP*EF(I2)*X(I2=I3+1)
      FV(I3)=FV(I3)+DELTA*EV(I2)*X(I2=I3+1)
10  CONTINUE
      L2=L1+I2+1
      ITER(L2)=FLOAT(L2=1)
      RV=(FV(1)=F(1))*2+(FV(2)=F(2))*2+(FV(3)=F(3))*2
      * +(FV(4)=F(4))*2+(FV(5)=F(5))*2
      RF=(FF(1)=F(1))*2+(FF(2)=F(2))*2+(FF(3)=F(3))*2
      * +(FF(4)=F(4))*2+(FF(5)=F(5))*2
      ERF(L2)=SQRT(RF)/(EOF(1))
      ERV(L2)=SQRT(RV)/(EOV(1))
      9  CONTINUE
      L1=L1+M
20  CONTINUE
      ERF(1)=1.0
      ERV(1)=1.0
      N=L*M
      WRITE(108,107) SNR
107  FORMAT(10X,'INPUT SIGNAL TO NOISE RATIO',F10.5,' DECIBELS'//)
      WRITE(108,100) N
100  FORMAT(10X,'NUMBER OF ITERATIONS:',I4//)
      WRITE(108,101)
101  FORMAT(10X,'SAMPLED CHANNEL RESPONSE IS'//)
      WRITE(108,102)(F(I),I=1,5)
102  FORMAT(5F10.3//)
      WRITE(108,103)
103  FORMAT(10X,'ESTIMATED SAMPLED CHANNEL RESPONSE USING FIXED STEP SI
      ZE IS'//)
      WRITE(108,104)(FF(I),I=1,5)
104  FORMAT(5F10.3//)
      WRITE(108,105)
105  FORMAT(10X,'ESTIMATED SAMPLED CHANNEL RESPONSE USING FIRST ORDER A
      LGORITHM IS'//)
      WRITE(108,106)(FV(I),I=1,5)
106  FORMAT(5F10.3//)
      LAST=4H

```

```
LABEL=4HFSTP  
CALL DRAW(ITER,ERF,N,1,0,LABEL,IT,15,,C.200,0,0,8,8,C,LAST)  
LABEL=4HVSTP  
LAST=4H  
IF(KK.EQ.10) LAST=4HLAST  
CALL DRAW(ITER,ERV,N,3,0,LABEL,IT,15,,C.200,0,0,8,8,C,LAST)  
57 CONTINUE  
END
```

```

C *****
C SECOND-ORDER ALGORITHM
C *****
C
C THIS PROGRAM IS FOR RAPID CHANNEL ESTIMATION
C USING SECOND-ORDER ALGORITHM.
C THE SECOND-ORDER ALGORITHM IS OF THE FORM
C  $FS(J)=FS(J)+DELTA*ERROR(K)*X(K=J+1)+BETA*(FS(J)=D(K=1,J))$ 
C WHERE  $D(K=1,J)$  IS THE ESTIMATED SAMPLE AT  $(K=1)$  ITERATION.
C DELTA AND BETA ARE VARYING FOR EACH ITERATION.
C THIS PROGRAM WORKS FOR CHANNEL LENGTH EQUAL TO 5.
C THIS PROGRAM MAKES USE OF SUBROUTINES, GAUSS, RANDOM,
C AND DRAW, AVAILABLE IN THE SYSTEM OF SIGMA 7.
C THIS PROGRAM PLOTS THE NORM OF THE ERROR VECTOR
C VERSUS NUMBER OF ITERATIONS.
C SNR IS THE INPUT SIGNAL-TO-NOISE RATIO.
C TEN PERCENT OF THE ANALYTIC VALUE OF DELTA AND BETA
C ARE USED IN THE ALGORITHM.
C THIS IS DONE IN ORDER TO TAKE INTO ACCOUNT THE
C EFFECT OF NOISE AND NOISY ESTIMATES OF THE GRADIENTS.
C 'LABEL' IS A CURVE IDENTIFIER.
C 'IT' IS AN ARRAY OF 24 CHARACTER HOLLERITH
C WORDS, THE FIRST 12 OF WHICH WILL FORM THE
C FIRST TITLE LINE AND THE LAST 12 THE SECOND
C TITLE LINE. THE ARRAY MUST BE DIMENSIONED
C AT LEAST 24 IN THE CALLING PROGRAM, MUST CONTAIN
C THE USERS NAME, AND MUST LEAVE UNWANTED
C CHARACTERS SET TO BLANK.
C 'LAST' INDICATES TO DRAW PROGRAM THAT
C THIS IS THE LAST CALL TO DRAW TO BE MADE
C IN THE USERS PROGRAM. IN THE VERY LAST
C CALL TO DRAW LAST SHOULD BE SET TO 4HLAST.
C THE SIZE OF THE GRAPH IS 8X8 INCHES.
C IP AND IX ARE INTEGER NUMBERS CHOSEN ARBITRARILY.
C ***** DATA CARDS REQUIRED *****

```

```

C      DATA SET
C      FIRST CARD FORMAT(I4)
C      FIRST CARD FIELD: M, THE NUMBER OF ITERATIONS.
C      SECOND CARD: SAMPLED CHANNEL RESPONSE
C      TO BE ESTIMATED; FORMAT(SF10.5)
C      THIRD CARD: FOR THE PURPOSE OF IDENTIFICATION
C      OF THE PLOT.
C      FOURTH CARD: NAME AND ACCOUNT NUMBER
C      *****
      DIMENSION IT(24)
      DIMENSION D(10,10)
      DIMENSION EOF(100),EOS(100)
      REAL ITER(100)
      DIMENSION FS(10),FF(10),ES(100),EF(100)
      INTEGER X(100)
      DIMENSION Y(100),Z(100),F(10),YF(100),YS(100)
      DO 57 KK=1,10
      READ(105,41) M
41      FORMAT(I4)
      READ(105,24) (F(I),I=1,5)
24      FORMAT(SF10.5)
      READ(105,16) (IT(I),I=1,24)
16      FORMAT(12A4/12A4)
      CON=1.0
      IP=99656
C      *****
C      S IS THE STANDARD DEVIATION OF THE NOISE.
C      *****
      S=0.001
      IX=65499
C      *****
C      AM IS THE MEAN OF THE RANDOM NOISE.
C      *****
      AM=0.0
      DO 18 I4=1,5
      FS(I4)=0.0

```

```

      FF(I4)=0.0
18  CONTINUE
      FS(1)=0.5
      FF(1)=0.5
      A=F(2)*F(2)+F(3)*F(3)+F(4)*F(4)+F(5)*F(5)
      EOF(1)=(FF(1)-F(1))**2+A
      EOS(1)=(FS(1)-F(1))**2+A
      EOF(1)=SQRT(EOF(1))
      EOS(1)=SQRT(EOS(1))
      ITER(1)=0.
C      *****
C      NOW THE RANDOM INPUT SEQUENCE IS GENERATED.
C      *****
      DO 5 J=1,M
      CALL RANDOM(IX,IY,YFL)
      IF(YFL=0.5) 2,2,3
2     X(J)=1
      GO TO 4
3     X(J)=0.1
4     IX=IY
5     CONTINUE
      DO 32 KA=1,M
32    X(KA)=FLOAT(X(KA))
      Y(1)=X(1)*F(1)
      Y(2)=X(2)*F(1)+X(1)*F(2)
      Y(3)=X(3)*F(1)+X(2)*F(2)+X(1)*F(3)
      Y(4)=X(4)*F(1)+X(3)*F(2)+X(2)*F(3)+X(1)*F(4)
      DO 6 K=5,M
6     Y(K)=X(K)*F(1)+X(K-1)*F(2)+X(K-2)*F(3)+X(K-3)*F(4)+X(K-4)*F(5)
C      *****
C      V IS THE RANDOM NOISE GENERATED.
C      *****
      DO 8 I1=1,M
      CALL GAUSS(IP,S,AM,V)
      Z(I1)=Y(I1)+V
8     CONTINUE

```

```

C *****
C 'STEP' IS THE FIXED STEP SIZE.
C *****
STEP=0.01
DO 35 J4=1,10
DO 35 J5=1,10
35 D(J4,J5)=0.0
D(1,1)=FS(1)
YS(1)=X(1)*FS(1)
YF(1)=X(1)*FF(1)
ES(1)=Z(1)=YS(1)
EF(1)=Z(1)=YF(1)
DELTA=0.04
FS(1)=FS(1)+DELTA*ES(1)*X(1)
FF(1)=FF(1)+STEP*EF(1)*X(1)
RE=(FF(1)-F(1))**2+(FF(2)-F(2))**2+(FF(3)-F(3))**2+
$ (FF(4)-F(4))**2+(FF(5)-F(5))**2
EOF(2)=SQRT(RE)/EOF(1)
RS=(FS(1)-F(1))**2+(FS(2)-F(2))**2+(FS(3)-F(3))**2+
$ (FS(4)-F(4))**2+(FS(5)-F(5))**2
EOS(2)=SQRT(RS)/EOS(1)
ITER(2)=1.
D(2,1)=FS(1)
BETA=1.0
YS(2)=X(2)*FS(1)+X(1)*FS(2)
YF(2)=X(2)*FF(1)+X(1)*FF(2)
ES(2)=Z(2)=YS(2)
EF(2)=Z(2)=YF(2)
DELTA=0.04*(1.0+BETA)
DO 12 I2=1,2
FF(I2)=FF(I2)+STEP*EF(2)*X(2=I2+1)
FS(I2)=FS(I2)+DELTA*ES(2)*X(2=I2+1)+CON*BETA*(FS(I2)-D(1,I2))
D(1,I2)=FS(I2)
12 CONTINUE
RE=(FF(1)-F(1))**2+(FF(2)-F(2))**2+(FF(3)-F(3))**2+
$ (FF(4)-F(4))**2+(FF(5)-F(5))**2

```

```

EOF(3)=SQRT(RE)/EOF(1)
RS=(FS(1)-F(1))**2+(FS(2)-F(2))**2+(FS(3)-F(3))**2+
$ (FS(4)-F(4))**2+(FS(5)-F(5))**2
EOS(3)=SQRT(RS)/EOS(1)
ITER(3)=2.
YS(3)=X(3)*FS(1)+X(2)*FS(2)+X(1)*FS(3)
YF(3)=X(3)*FF(1)+X(2)*FF(2)+X(1)*FF(3)
ES(3)=Z(3)=YS(3)
EF(3)=Z(3)=YF(3)
BETA=(1.+BETA)/(3.=BETA)
DELTA=0.04*(1.0+BETA)
DO 14 I=1,3
FS(I)=FS(I)+DELTA*ES(3)*X(3=I+1)+CON*BETA*(FS(I)=D(2,I))
FF(I)=FF(I)+STEP*EF(3)*X(3=I+1)
D(2,I)=FS(I)
14 CONTINUE
RE=(FF(1)-F(1))**2+(FF(2)-F(2))**2+(FF(3)-F(3))**2+
$ (FF(4)-F(4))**2+(FF(5)-F(5))**2
EOF(4)=SQRT(RE)/EOF(1)
RS=(FS(1)-F(1))**2+(FS(2)-F(2))**2+(FS(3)-F(3))**2+
$ (FS(4)-F(4))**2+(FS(5)-F(5))**2
EOS(4)=SQRT(RS)/EOS(1)
ITER(4)=3.
YS(4)=X(4)*FS(1)+X(3)*FS(2)+X(2)*FS(3)+X(1)*FS(4)
YF(4)=X(4)*FF(1)+X(3)*FF(2)+X(2)*FF(3)+X(1)*FF(4)
ES(4)=Z(4)=YS(4)
EF(4)=Z(4)=YF(4)
BETA=(1.+BETA)/(3.=BETA)
DELTA=0.04*(1.0+BETA)
DO 15 J=1,4
FS(J)=FS(J)+DELTA*ES(4)*X(4=J+1)+CON*BETA*(FS(J)=D(1,J))
FF(J)=FF(J)+STEP*EF(4)*X(4=J+1)
D(1,J)=FS(J)
15 CONTINUE
RE=(FF(1)-F(1))**2+(FF(2)-F(2))**2+(FF(3)-F(3))**2+
$ (FF(4)-F(4))**2+(FF(5)-F(5))**2

```

```

EOF(5)=SQRT(RE)/EOF(1)
RS=(FS(1)=F(1))*2+(FS(2)=F(2))*2+(FS(3)=F(3))*2+
$ (FS(4)=F(4))*2+(FS(5)=F(5))*2
EOS(5)=SQRT(RS)/EOS(1)
ITER(5)=4.

```

C THIS PROGRAM USES PAST TAP GAINS

```

DO 9 I3=5,M,2
  YS(I3)=X(I3)*FS(1)+X(I3=1)*FS(2)+X(I3=2)*FS(3)+X(I3=3)*FS(4)
  $+X(I3=4)*FS(5)
  YF(I3)=X(I3)*FF(1)+X(I3=1)*FF(2)+X(I3=2)*FF(3)+X(I3=3)*FF(4)
  $+X(I3=4)*FF(5)
  ES(I3)=Z(I3)=YS(I3)
  EF(I3)=Z(I3)=YF(I3)
  BETA=(1.+BETA)/(3.=BETA)
  DELTA=0.04*(1.0+BETA)
  DO 10 I4=1,5
    FS(I4)=FS(I4)+DELTA*ES(I3)*X(I3=I4+1)+CON*BETA*(FS(I4)=D(2,I4))
    FF(I4)=FF(I4)+STEP*EF(I3)*X(I3=I4+1)
    D(2,I4)=FS(I4)
10  CONTINUE
  RE=(FF(1)=F(1))*2+(FF(2)=F(2))*2+(FF(3)=F(3))*2+
  $ (FF(4)=F(4))*2+(FF(5)=F(5))*2
  JJ=I3+1
  EOF(JJ)=SQRT(RE)/EOF(1)
  RS=(FS(1)=F(1))*2+(FS(2)=F(2))*2+(FS(3)=F(3))*2+
  $ (FS(4)=F(4))*2+(FS(5)=F(5))*2
  EOS(JJ)=SQRT(RS)/EOS(1)
  ITER(JJ)=FLOAT(I3)
  K1=I3+1
  YS(K1)=X(K1)*FS(1)+X(K1=1)*FS(2)+X(K1=2)*FS(3)+
  *X(K1=3)*FS(4)+X(K1=4)*FS(5)
  YF(K1)=X(K1)*FF(1)+X(K1=1)*FF(2)+X(K1=2)*FF(3)+
  *X(K1=3)*FF(4)+X(K1=4)*FF(5)
  ES(K1)=Z(K1)=YS(K1)
  EF(K1)=Z(K1)=YF(K1)
  BETA=(1.+BETA)/(3.=BETA)

```

```

DELTA=0.04*(1.0+BETA)
DO 31 K2=1,5
  FS(K2)=FS(K2)+DELTA*ES(K1)*X(K1=K2+1)+CON*BETA*(FS(K2)=D(1,K2))
  D(1,K2)=FS(K2)
  FF(K2)=FF(K2)+STEP*EF(K1)*X(K1=K2+1)
31 CONTINUE
  JJ=K1+1
  ITER(JJ)=FLOAT(K1)
  RE=(FF(1)=F(1))*2+(FF(2)=F(2))*2+(FF(3)=F(3))*2+
$ (FF(4)=F(4))*2+(FF(5)=F(5))*2
  EOF(JJ)=SQRT(RE)/EOF(1)
  RS=(FS(1)=F(1))*2+(FS(2)=F(2))*2+(FS(3)=F(3))*2+
$ (FS(4)=F(4))*2+(FS(5)=F(5))*2
  EOS(JJ)=SQRT(RS)/EOS(1)
9 CONTINUE
  SNR=(10.)*ALOG10(0.125/(S**2))
  WRITE(108,107) SNR
107 FORMAT(10X,'INPUT SIGNAL TO NOISE RATIO',F10.5,'DECIBELS'//)
  WRITE(108,100) M
100 FORMAT(10X,'NUMBER OF ITERATIONS: ',I4//)
  WRITE(108,101)
101 FORMAT(10X,'SAMPLED CHANNEL RESPONSE IS'//)
  WRITE(108,102)(F(I),I=1,5)
102 FORMAT(5F10.3//)
  WRITE(108,103)
103 FORMAT(10X,'ESTIMATED SAMPLED CHANNEL RESPONSE USING FIXED STEP SI
$ZE IS'//)
  WRITE(108,104)(FF(I),I=1,5)
104 FORMAT(5F10.3//)
  WRITE(108,105)
105 FORMAT(10X,'ESTIMATED SAMPLED CHANNEL RESPONSE USING SECOND ORDER
$ALGORITHM IS'//)
  WRITE(108,106)(FS(I),I=1,5)
106 FORMAT(5F10.3)
  EOF(1)=1.
  EOS(1)=1.

```

```
LABEL=4HFSTP  
LAST=4H  
CALL DRAW(ITER,EOF,M,1,0,LABEL,IT,15,,0.2,0,0,8,8,0, LAST)  
LABEL=4HVSTP  
LAST=4H  
IF(KK.EQ.10) LAST=4HLAST  
CALL DRAW(ITER,EOS,M,3,0,LABEL,IT,15,,0.2,0,0,8,8,0, LAST)  
57 CONTINUE  
END
```

```

C *****
C FORNEY'S SUBOPTIMUM ALGORITHM
C *****
C
C THIS ALGORITHM IS FOR ESTIMATION OF A DIGITAL
C SEQUENCE IN THE PRESENCE OF INTERSYMBOL INTERFERENCE.
C THIS PROGRAM IS FOR  $F(D) \approx 1-D$ .
C THIS PROGRAM HAS BINARY SIGNALLING.
C THIS PROGRAM PLOTS THE LOG OF PROBABILITY OF
C SYMBOL ERROR VERSUS OUTPUT SIGNAL-TO-NOISE-RATIO.
C THIS PROGRAM MAKES USE OF SUBROUTINES, GAUSS,
C RANDOM, AND DRAW, AVAILABLE IN THE SYSTEM OF SIGMA 7.
C A FUNCTION SUBPROGRAM 'DSIMPRH' IS USED TO
C EVALUATE THE ERROR FUNCTION.
C 'DSIMPRH' IS ALSO AVAILABLE IN THE SYSTEM OF SIGMA 7.
C IX AND IP ARE INTEGER NUMBERS CHOSEN ARBITRARILY.
C S IS THE STANDARD DEVIATION OF THE NOISE.
C AM IS THE MEAN OF THE NOISE.
C 'A' IS THE UPPER LIMIT FOR EVALUATING THE ERROR FUNCTION.
C 'AREA2' IS THE VALUE OF THE ERROR FUNCTION WHEN
C THE LOWER LIMIT IS ZERO AND THE UPPER LIMIT IS EQUAL TO A.
C *****
C DEL IS DELAY FOR LOCALIZING THE ERROR.
C *****
C 'LABEL' IS A CURVE IDENTIFIER.
C 'IT' IS AN ARRAY OF 24 CHARACTER HOLLERITH
C WORDS, THE FIRST 12 OF WHICH WILL FORM THE
C FIRST TITLE LINE AND THE LAST 12 THE SECOND
C TITLE LINE. THE ARRAY MUST BE DIMENSIONED
C AT LEAST 24 IN THE CALLING PROGRAM, MUST CONTAIN
C THE USERS NAME, AND MUST LEAVE UNWANTED
C CHARACTERS SET TO BLANK.
C 'LAST' INDICATES TO DRAW PROGRAM THAT
C THIS IS THE LAST CALL TO DRAW TO BE MADE
C IN THE USERS PROGRAM. IN THE VERY LAST
C CALL TO DRAW LAST SHOULD BE SET TO 4HLAST.

```

```

C      THE SIZE OF THE GRAPH IS 8X8 INCHES.
C      DATA SET
C      ***** DATA CARDS REQUIRED *****
C      FIRST CARD FORMAT (I3)
C      FIRST CARD FIELD: DEL
C      SECOND CARD FORMAT(I10,2F10.2)
C      FIRST FIELD: IP
C      SECOND FIELD: S
C      THIRD FIELD: AM
C      THIRD CARD FORMAT(I3,10F5.2)
C      FIRST FIELD: IFA, DEGREE OF F(D)
C      OTHER FIELDS: COEFFICIENTS OF F(D)
C      FOURTH CARD FORMAT(I4)
C      FOURTH CARD FIELD: N, THE NUMBER OF TERMS IN X(D).
C      FIFTH CARD: FOR THE PURPOSE OF IDENTIFICATION OF THE PLOT.
C      SIXTH CARD: NAME AND ACCOUNT NUMBER.
C      *****
      DIMENSION Z(1000)
      DIMENSION F(10)
      INTEGER X(1000),XEST(1000),Y(1000)
      EXTERNAL F1
      REAL*8F1,AREA2,DSIMPRH,EPS,A
      INTEGER DEL
      DIMENSION SN(100),PRE(100),PRL(100),PRC(100),IT(24)
      READ(105,1) DEL
      READ(105,6) IP,S,AM
1      FORMAT(I3)
6      FORMAT(I10,2F10.2)
      READ(105,7) IFA,(F(I),I=1,IFA),F(IFA+1)
7      FORMAT(I3,10F5.2)
      READ(105,33) N
33     FORMAT(I4)
      READ(105,8)(IT(I),I=1,24)
8      FORMAT(12A4/12A4)
      DO 22 K6=1,6
      EPS=1.0D=4

```

```

NERR=0
M=10
DO 47 J1=1,M
  IX=65669
  DO 11 J1=1,N
C   *****
C   NOW THE RANDOM INPUT SEQUENCE IS GENERATED.
C   *****
    CALL RANDOM(IX,IY,YFL)
    IF(YFL=0.5) 14,14,15
14  X(J1)=0
    GO TO 16
15  X(J1)=1
16  IX=IY
11  CONTINUE
    INT=N+1
    X(INT)=0
C   *****
C   V IS THE RANDOM NOISE GENERATED.
C   *****
    CALL GAUSS(IP,S,AM,V)
    Z(1)=F(1)*FLOAT(X(1))+V
    CALL GAUSS(IP,S,AM,V)
    Z(2)=F(1)*FLOAT(X(2))+F(2)*FLOAT(X(1))+V
    DO 4 I=3,INT
      CALL GAUSS(IP,S,AM,V)
4   Z(I)=F(1)*FLOAT(X(I))+F(2)*FLOAT(X(I-1))+F(3)*FLOAT(X(I-2))
      +V
    DO 5 J=1,INT
      IF(Z(J).GE.0.5) Y(J)=1
      IF(Z(J).LE.-0.5) Y(J)=-1
      IF(ABS(Z(J)).LT.0.5) Y(J)=0
5   CONTINUE
    CALL ERDC(Z,Y,INT,DEL)
    DO 27 I1=1,INT
      IF(Y(I1).GT.1) Y(I1)=1

```

```

      IF(Y(I1)*LT=.1) Y(I1)=.1
27  CONTINUE
      J12=INT.1
      XEST(1)=Y(1)
      IF(XEST(1)*GT.1) XEST(1)=1
      IF(XEST(1)*LT.0) XEST(1)=0
      DO 10 K4=2,J12
      XEST(K4)=Y(K4)+XEST(K4-1)
      IF(XEST(K4)*GT.1) XEST(K4)=1
      IF(XEST(K4)*LT.0) XEST(K4)=0
10  CONTINUE
      NER=0
      DO 31 K5=1,J12
      IF((X(K5)-XEST(K5)).NE.0) NER=NER+1
31  CONTINUE
      NERR=NER+NERR
47  CONTINUE
      A=1./(1+.414*S)
      AREA2=DSIMPRH(0,DO,A,F1,EPS,128,5,J)
      PU=2.*(4+AREA2)
      PL=0.5*AREA2
      PRU(K6)=ALOG10(PU)
      PRL(K6)=ALOG10(PL)
      SN(K6)=(10.)*ALOG10(0.5/(S**2))
      AU=FLOAT(NERR)
      PE=AU/(FLOAT(N*M))
      IF(AU.EQ.0.) PRE(K6)=.4. ; GO TO 25
      PRE(K6)=ALOG10(PE)
25  CONTINUE
      S=S+0.05
22  CONTINUE
      LAST=4H
      LABEL =4HUPBD
      CALL DRAW(SN,PRU, 6,1,0,LABEL,IT,5,1,6,0,8,8,0,LAST)
      LABEL=4HSMRL
      LAST=4H

```

```

CALL DRAW(SN,PRE, 6,2,3,LABEL,IT,5,,1,,6,0,8,8,0, LAST)
LABEL=4HLOBD
LAST=4HLAST
CALL DRAW(SN,PRL, 6,3,0,LABEL,IT,5,,1,,6,0,8,8,0, LAST)
END
SUBROUTINE ERDC(Z,Y,INT,J)
INTEGER X(1000),Y(1000)
DIMENSION Z(1000)
X(1)=Y(1)
IF(X(1).LE.0) X(1)=0
J11=INT+1
DO 50 I=2,J11
X(I)=Y(I)+X(I-1)
IF(X(I).GT.1) X(I)=1; CALL CORP(X,Y,Z,I,J)
IF(X(I).LT.0) X(I)=0; CALL CORN(X,Y,Z,I,J)
50 CONTINUE
RETURN
END
SUBROUTINE CORP(X,Y,Z,I,J)
INTEGER X(1000),Y(1000)
DIMENSION Z(1000),GN(1000)
C COMPUTE NOISE TERMS FOR ALL TERMS FOR AFTER X=0
DO 50 K=1,J
KK=I+K+1
IF(KK.EQ.0) GO TO 52
IF(X(KK).EQ.0) I1=KK+1; GO TO 51
GN(KK)=Z(KK)+Y(KK)
50 CONTINUE
52 CONTINUE
XMAX=GN(I)
I2=I
DO 60 K2=2,J
K1=I+K2+1
IF(K1.EQ.0) GO TO 25
IF(GN(K1).LT.XMAX) I2=K1; XMAX=GN(K1)
60 CONTINUE

```

```

25  CONTINUE
    GO TO 20
51  XMAX=GN(I1)
    I2=I1
    DO 55 K3=I1,I
    IF(GN(K3).LT.XMAX) I2=K3; XMAX=GN(K3)
55  CONTINUE
20  Y(I2)=Y(I2)+1
    RETURN
    END
    SUBROUTINE CORN(X,Y,Z,I,J)
    INTEGER X(1000),Y(1000)
    DIMENSION Z(1000),GN(1000)
C   COMPUTE NOISE FOR ALL TERMS AFTER X=1
    DO 50 K=1,J
    KK=I+K+1
    IF(KK.EQ.0) GO TO 52
    IF(X(KK).EQ.1) I1=KK+1; GO TO 51
    GN(KK)=Z(KK)=Y(KK)
50  CONTINUE
52  CONTINUE
C   GN(KK) IS AN APPARENT NOISE TERM
    XMIN=GN(I)
    I2=I
    DO 60 K2=2,J
    K1=I+K2+1
    IF(K1.EQ.0) GO TO 25
    IF(GN(K1).GT.XMIN) I2=K1; XMIN=GN(K1)
60  CONTINUE
25  CONTINUE
    GO TO 20
51  XMIN=GN(I1)
    I2=I1
    DO 55 K3=I1,I
    IF(GN(K3).GT.XMIN) I2=K3; XMIN=GN(K3)
55  CONTINUE

```

```
20 Y(I2)=Y(I2)+1
   RETURN
   END
   FUNCTION F1(X)
   REAL*8 F1,X
   F1=(SQRT(7))/(SQRT(44))*(DEXP((-X*X)/2.))
   RETURN
   END
```

REFERENCES

REFERENCES

- [1] Nyquist, H., "Certain Topics in Telegraph Transmission Theory," Transactions AIEE (Communications and Electronics), Vol. 47, April 1928, pp. 617-644.
- [2] Tufts, D.W., "Nyquist Problem--The Joint Optimization of Transmitter and Receiver in Pulse Amplitude Modulation," Proceedings IEEE, Vol. 53, March 1965, pp. 248-259.
- [3] Berger, T., and Tufts, D.W., "Optimum Pulse Amplitude Modulation. Part 1: Transmitter-Receiver design and Bounds from Information Theory," IEEE Transactions on Information Theory, Vol. IT-13, April 1967, pp. 196-208.
- [4] George, D.A., "Matched Filters for Interferring Signals," IEEE Transactions on Information Theory (Correspondence), Vol. IT-11, January 1965, pp. 153-154.
- [5] Ericson, T., "Structure of Optimum Receiving Filters in Data Transmission Systems," IEEE Transactions on Information Theory (Correspondence), Vol. IT-17, May 1971, pp. 352-353.
- [6] Aaron, M.R., and Tufts, D.W., "Intersymbol Interference and Error Probability," IEEE Transactions on Information Theory, Vol. IT-12, January 1966, pp. 26-34.
- [7] Gerst, I., and Diamond, J., "The Elimination of Intersymbol Interference by Input Signal Shaping," Proceedings IRE, Vol. 49, July 1961, pp. 1195-1203.
- [8] Austin, M.E., "Decision Feedback Equalization for Digital Communication over Dispersive Channels," M.I.T. Research Laboratory of Electronics, Cambridge, Massachusetts, Technical Report, No. 461, August 1967.
- [9] Chang, R.W., and Hancock, J.C., "On Receivers Structures for Channels Having Memory," IEEE Transactions on Information Theory, Vol. IT-12, October 1966, pp. 463-488.

- [10] Abend, K., Harley, T.J. Jr., Fritchman, B.D., and Camacos, C., "On Optimum Receivers for Channels Having Memory," IEEE Transactions on Information Theory (Correspondence), Vol. IT-15, November 1968, pp. 819-820.
- [11] Bowen, R.R., "Bayesian Decision Procedure for Interfering Signals," IEEE Transactions on Information Theory (Correspondence), Vol. IT-15, July 1969, pp. 506-507.
- [12] Abend, K., and Fritchman, B.D., "Statistical Detection for Communication Channels with Intersymbol Interference," Proceedings IEEE, Vol. 58, May 1970, pp. 779-785.
- [13] Hilborn, C.G. Jr., "Applications of Unsupervised Learning to Problems of Digital Communication," Proceedings 9th Symposium on Adaptive Processes, Decision and Control, Austin, Texas, December 1970.
- [14] Ungerboeck, G., "Nonlinear Equalization of Binary Bipolar Signals in Gaussian Noise," IEEE Transactions on Communication Technology, Vol. COM-19, December 1971, pp. 1128-1137.
- [15] Lucky, R.W., "Automatic Equalization for Digital Communication," Bell System Technical Journal, Vol. 44, April 1965, pp. 547-588.
- [16] _____, "Techniques for Adaptive Equalization of Digital Communication Systems," Bell System Technical Journal, Vol. 45, February 1966, pp. 255-286.
- [17] _____, and Rudin, H.R., "Generalized Automatic Equalization for Communication Channels," Proceedings IEEE (Correspondence), Vol. 54, March 1966, pp. 439-440.
- [18] _____, and _____, "An Automatic Equalization for General Purpose Communication Channels," Bell System Technical Journal, Vol. 46, November 1967, pp. 2179-2208.

- [19] Neissen, C.W., "Automatic Channel Equalization Algorithm," Proceedings IEEE (Letters), Vol. 55, May 1967, p. 698.
- [20] Gersho, A., "Adaptive Equalization of Highly Dispersive Channels for Data Transmission," Bell System Technical Journal, Vol. 48, January 1969, pp. 55-70.
- [21] Proakis, J.G., and Miller, J.H., "An Adaptive Receiver for Digital Signalling Through Channels with Intersymbol Interference," IEEE Transactions on Information Theory, Vol. IT-15, July 1969, pp. 484-497.
- [22] Forney, G.D. Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," IEEE Transactions on Information Theory, Vol. IT-18, May 1972, pp. 363-378.
- [23] Viterbi, A.J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, Vol. IT-13, April 1967, pp. 260-269.
- [24] _____, "Convolutional Codes and Their Performance in Communication Systems," IEEE Transactions on Communication Technology, Vol. COM-19, October 1971, pp. 751-772.
- [25] Forney, G.D. Jr., "Lower Bounds on Error Probability in the Presence of Large Intersymbol Interference," IEEE Transactions on Communication Technology (Correspondence), Vol. COM-20, February 1972, pp. 76-77.
- [26] Omura, J.K., "On the Viterbi Decoding Algorithm," IEEE Transactions on Information Theory (Correspondence), Vol. IT-15, January 1969, pp. 177-179.
- [27] Kobayshi, H., "Correlative Level Coding and Maximum-Likelihood Decoding," IEEE Transactions on Information Theory, Vol. IT-17, September 1971, pp. 586-595.

- [28] Bellman, R., Dynamic Programming (Princeton University Press, Princeton, N.J. 1957).
- [29] Kobayshi, H., "Application of Probabilistic Decoding to Digital Magnetic Recording Systems," IBM Journal of Research and Development, Vol. 14, January 1971, pp. 64-74.
- [30] Forney, G.D. Jr., "The Viterbi Algorithm," Proceedings IEEE, Vol. 61, No. 3, March 1973, pp. 268-278.
- [31] Magee, F.R., and Proakis, J.G., "Adaptive Maximum-Likelihood Sequence Estimation for Digital Signalling in the Presence of Intersymbol Interference," IEEE Transactions on Information Theory (Correspondence), Vol. IT-15, January 1973, pp. 120-124.
- [32] Quershi, S.U.H., and Newhall, E.E., "An Adaptive Receiver for Data Transmission over Time-Dispersive Channels," IEEE Transactions on Information Theory, Vol. IT-19, No. 4, July 1973, pp. 448-457.
- [33] Mackechnie, L.K., "Maximum-Likelihood Receivers for Channels Having Memory," Ph.D. dissertation, Dep. Elec. Eng., University of Notre Dame, Notre Dame, Indiana, January 1973.
- [34] Magee, F.R., and Proakis, J.G., "An Estimate of the Upper Bound on Error Probability for Maximum-Likelihood Sequence Estimation on Channels Having a Finite Duration Pulse Response," IEEE Transactions on Information Theory (Correspondence), Vol. IT-15, September 1973, pp. 699-702.
- [35] Schonfeld, T.J., and Schwartz, M., "A Rapidly Converging First-Order Training Algorithm for an Adaptive Equalizer," IEEE Transactions on Information Theory, Vol. IT-17, No. 4, July 1971, pp. 431-439.
- [36] _____, and _____, "Rapidly Converging Second-order Tracking Algorithms for Adaptive Equalization," IEEE Transactions on Information Theory Vol. IT-17, No. 5, September 1971, pp. 572-579.

- [37] Varga, R.S., Matrix Iterative Analysis, Englewood Cliffs, N.J., Prentice-Hall, 1967, p. 135.
- [38] Pierre, D.A., Optimization Theory with Applications, John Wiley & Sons, Inc., New York, 1969.
- [39] Smith, J.W., "Error Control in Duobinary Data Systems by Means of Null Zone Detection," IEEE Transactions on Communication Technology, Vol. COM-16, No. 6, December 1968, pp. 825-830.
- [40] Silverman, R.A., and Balser, M., "Coding for Constant-Data Rate Systems- Part 1," Proceedings IRE, September 1954, pp. 1428-1435.
- [41] Kretzmer, E.R., "Generalization of a Technique for Binary Data Communication," IEEE Transactions on Communication Technology (Concise Paper), Vol. COM-14, February 1966, pp. 67-68.
- [42] Lender, A., "Correlative Digital Communication Techniques," IEEE Transactions on Communication Technology, Vol. COM-17, December 1964, pp. 128-135.
- [43] Ferguson, M.J., "Optimal Reception for Binary Partial Response Channels," Bell System Technical Journal, Vol. 51, February 1972, pp. 493-505.
- [44] Gibson, E.D., "Error Correction with Little or No Digit Redundancy," 1973 IEEE International Symposium on Information Theory, Seattle, Washington, June 11-13.
- [45] Widrow, B., "Adaptive Filters I : Fundamentals,"
Electronics Laboratory Report SEL-66-126,
Stanford, California, December 1966.
- [46] Wozencraft, J.M., and Jacobs, I.M., Principles of Communication Engineering (John Wiley & Sons, Inc., New York, 1965).

- [47] Gray, R.M., "On the Asymptotic Eigen Value Distribution of Toeplitz Matrices," IEEE Transactions on Information Theory, Vol. IT-18, No. 6, November 1972, pp. 725-730.
- [48] Lucky, R.W., Salz, J., and Weldon, E.J. Jr., Principles of Data Communication (McGraw-Hill Book Company, New York, 1968).
- [49] Bennett, W.R., and Davey, J.R., Data Transmission (McGraw-Hill Book Company, New York, 1965).
- [50] Lugannani, R., "Intersymbol Interference and Probability Error in Digital Systems," IEEE Transactions on Information Theory, Vol. IT-18, No. 6, December 1969, pp. 682-688.
- [51] Gerald, C.F., Applied Numerical Analysis (Addison Wesley Publishing Company, Reading, Massachusetts, 1970).
- [52] Van Trees, H.L., Detection, Estimation and Modulation Theory, Part 1 (John Wiley & Sons, Inc., New York, 1968).
- [53] Widrow, B., "Adaptive Filters," Aspects of Network and System, R.E. Kalman and N. DeClaris, Editor, Part 4, Holt, Rinehard and Winston, 1971, pp. 563-567.
- [54] Rudin, H.R., "Automatic Equalization Using Transversal Filters," IEEE Spectrum, January 1967, pp. 53-59.
- [55] Schwartz, M., Information, Transmission, Modulation and Noise (McGraw-Hill Book Company, New York, 1959).

MONTANA STATE UNIVERSITY LIBRARIES

3 1762 10012

3 1762 10010959 2

K897

cop. 2

Krishan, Baldev

Adaptive receiver
structures for time-
dispersive channels

NAME AND ADDRESS

D378
K897
cop. 2