



Performance analysis of the maximum likelihood sequence estimator for known channel impulse responses
by Harold Fred Fisher

A thesis submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE in Electrical Engineering
Montana State University
© Copyright by Harold Fred Fisher (1973)

Abstract:

In a pulse amplitude modulation system, intersymbol interference is the primary detriment to reliable high rate digital transmission over narrow bandwidth high signal to noise ratio channels. The maximum likelihood sequence estimator is a receiver structure designed for use under intersymbol interference conditions. The objective of this thesis is to develop and use an analysis technique for evaluating the performance of the maximum likelihood sequence estimator for linear channels with known impulse responses. The performance criterion used is the fractional difference in minimum euclidean weight due to intersymbol interference. Discrete time signal models and flow graph theory are used to develop the analysis technique.

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Montana State University, I agree that the Library shall make it freely available for inspection. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by my major professor, or, in his absence, by the Director of Libraries. It is understood that any copying or publication of this thesis for financial gain shall not be allowed my written permission.

Signature

Harold Fred Fisher

Date

June 18, 1973

PERFORMANCE ANALYSIS OF THE MAXIMUM LIKELIHOOD SEQUENCE
ESTIMATOR FOR KNOWN CHANNEL IMPULSE RESPONSES

by

HAROLD FRED FISHER

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

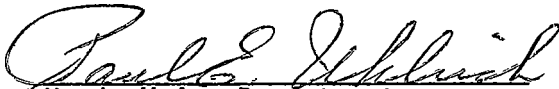
of

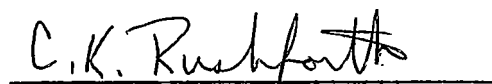
MASTER OF SCIENCE

in

Electrical Engineering

Approved:


Head, Major Department


Chairman, Examining Committee


Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

August, 1973

ACKNOWLEDGMENT

The author wishes to thank his committee C.K.Rushforth, D.A.Rudberg and M.A.Faulkner for their guidance in helping prepare this thesis. Also, the author expresses his appreciation to R.E.Leo his research assistantship advisor for his patiences and understanding showed during the entire time spent preparing the thesis.

TABLE OF CONTENTS

	Page
VITA	ii
ACKNOWLEDGMENT	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
ABSTRACT	viii
 Chapter	
I. INTRODUCTION	1
II. CHANNEL MODELS AND THE OPTIMUM RECEIVER	6
DISCRETE TIME ANALYSIS	6
D-TRANSFORM	8
CHIP D-TRANSFORM	11
A RUNNING EXAMPLE	14
III. PERFORMANCE ANALYSIS	21
ERROR EVENTS	21
EUCLIDEAN WEIGHT	23
FINDING MINIMUM WEIGHT	24

Chapter	Page
IV. ANALYSIS USING FLOW GRAPHS	27
ERROR STATE	28
MINIMUM PATH THROUGH FLOW GRAPH	29
MINIMUM EUCLIDEAN DISTANCE WITH NO INTERSYMBOL INTERFERENCE	32
FRACTIONAL DIFFERENCE IN MINIMUM WEIGHT	33
MINIMUM DISTANCE PATH ALGORITHM	35
V. SEVERAL EXAMPLES	43
VI. CONCLUSION AND SUMMARY	52
APPENDIX	54
REFERENCES	65

LIST OF TABLES

Table	Page
1. Program Output for Examples 1 through 5	49
2. Program Output for Examples 6 through 8	50

LIST OF FIGURES

Figure	Page
1. P.A.M. Channel with White Gaussian Noise and the Maximum Likelihood Sequence Receiver structure	3
2. Discrete Time Model of P.A.M. System	11
3. P.A.M. Channel with White Gaussian Noise, Filter Matched to $h(t)$, and Transversal filter with Discrete Time Response $1/f(D^{-1})$	14
4. Channel Response to (a) $x_0 = 1$, (b) $x_1 = 1$, (c) $x_3 = 1$, (d) $x_4 = 1$ and (e) $x_0 = 1$, $x_1 = 1$, $x_2 = 0$, $x_3 = 1$ and $x_4 = 1$	15
5. Convolutional Encoder Equivalent to the Flat Top Pulse Channel	16
6. Chip Function for $i = 0, 1$ and 2	17
7. Flow Graph of Flat Top Pulse	30
8. Closed Loop Paths Adding Zero Weight	31
9. Flow Graph with Minimum Distance Paths Only	31
10. Flow Graph of Flat Top Pulse Example	38

ABSTRACT

In a pulse amplitude modulation system, intersymbol interference is the primary detriment to reliable high rate digital transmission over narrow bandwidth high signal to noise ratio channels. The maximum likelihood sequence estimator is a receiver structure designed for use under intersymbol interference conditions. The objective of this thesis is to develop and use an analysis technique for evaluating the performance of the maximum likelihood sequence estimator for linear channels with known impulse responses. The performance criterion used is the fractional difference in minimum euclidean weight due to intersymbol interference. Discrete time signal models and flow graph theory are used to develop the analysis technique.

I. INTRODUCTION

In a pulse amplitude modulation system whenever the information rate, measured in units of pulses per second (baud), is larger than the bandwidth of the transmission channel, intersymbol interference may result [1]. In other words, the pulses overlap into adjacent time slots and may cause an erroneous decision at the receiver. This phenomenon is the primary deterrent to reliable, high rate digital transmission over narrow bandwidth, high signal-to-noise ratio channels. Voice-grade telephone channels used for data transmission are typical examples.

The objective of this thesis is to develop and use a technique for analyzing the performance of the maximum likelihood sequence estimator [2] for linear channels whose impulse response is longer than the source symbol separation.

Many receiver structures have been discussed and developed [3,4,5]. Probably the most talked about structure has been the optimum receiver structure--the one that makes symbol decisions based on the entire received sequence [5]. This structure was not pursued, however, since maximum likelihood calculations increased exponentially with sequence length making hardware development too complex and difficult. As an alternative to this overwhelming complexity problem, structures were developed that made simple symbol-by-symbol decisions [4]. The best forms of these receiver structures were derived using certain criterion of optimality, such as minimum probability of error, minimum probability of error with intersymbol interference forced to zero, and minimum mean-square

error. In every case these receiver structures turned out to be a matched filter in cascade with a tapped delay line (transversal filters) [3].

Nonlinear receivers have also been looked at [6]. Several optimum nonlinear structures have been developed, but these turned out to be very complex. This caused the experts to look at suboptimal nonlinear receivers such as decision feedback. These were far too complex and difficult to analyze to justify their use.

The most recent receiver structure devised for channels introducing gaussian noise and intersymbol interference is a maximum likelihood sequence estimator of the entire received sequence [1]. Unlike past formulations of likelihood sequence estimators its complexity does not increase with sequence length, but is proportional to M^L , where M is the size of the input alphabet and L is the length of channel impulse response in units of symbol separation. In fact this receiver can easily be implemented and analyzed.

Figure 1 is a block diagram of a P.A.M. channel with additive white gaussian noise cascaded with the maximum likelihood sequence receiver. The receiver consists of a linear filter called the whitened matched filter, a sampler taking samples once every T seconds (symbol separation), and a nonlinear recursive algorithm called the Viterbi Algorithm. From Figure 1

$$s(t) = \sum_{i=0}^{i=N} x_i h(t-iT) \quad (1)$$

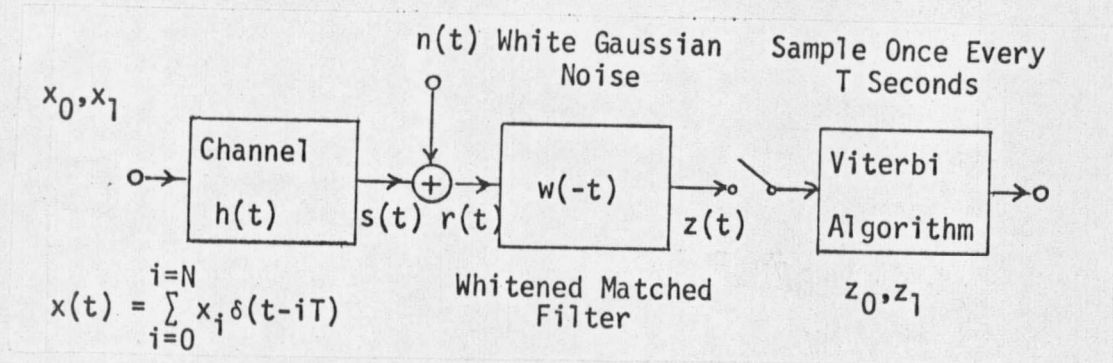


Figure 1. P.A.M. channel with white gaussian noise and the maximum likelihood sequence receiver structure.

where

$s(t)$ = output of channel whose impulse response is $h(t)$

$x(t)$ = input sequence

T = input symbol separation

$x_i = (i + 1)^{\text{th}}$ symbol of the input sequence

The implementation of the whitened matched filter is quite simple. Instead of having a bank of matched filters, one for every input symbol sent, only one filter $[w(-t)]$, with samples taken once every symbol instant is necessary. This leaves the output samples with all the necessary information making them a set of sufficient statistics for the estimation of the transmitted sequence. In addition, the output sequence, $z_0, z_2, \dots, z_{N+L-1}$ is a sequence of statistically independent, identically distributed gaussian random variables. This is necessary to insure the simple and recursive property of the Viterbi algorithm.

The Viterbi algorithm was originally used to decode convolutional

codes. Because of the similarity between a data transmission channel causing intersymbol interference and a convolutional encoder the Viterbi algorithm can be used for maximum likelihood sequence estimation [1]. As mentioned earlier, the algorithm gives likelihood calculation with complexity proportional to M^L . To implement the algorithm only M^{L-1} comparisons and M^L additions per received symbol are required. Also, M^{L-1} registers for remembering the surviving path and M^{L-1} registers for remembering the appropriate metrics are needed [7].

Much of the literature dealing with detection theory classifies a receiver optimum when the probability of error is minimized [2, 3, 8, 9, 10]. For the maximum likelihood sequence estimator, a bound on the minimum probability of error, can be determined by knowing the minimum euclidean distance or weight of the output signal space. Minimum euclidean distance is the distance between the two closest signals or sequences of the output signal space. If the channel causes intersymbol interference, the minimum distance may become smaller. Some channels cause a larger reduction in distance than others, hence, causing a larger increase in probability of error. This decrease directly reduces the reliability of the system. If it is known which channels yield the smallest differences (percent reduction) then communication systems can be devised that provide maximum reliability in the presence of intersymbol interference and additive white gaussian noise.

The performance criteria used is the fractional difference in minimum

euclidean weight. To determine this for each channel, minimum euclidean weight with and without intersymbol interference must be found.

Associated with each channel impulse response there is a set of error states that form nodes of a flow graph. These nodes are connected in such a way that a transition from one node to another determines a single receiver output error event. Further, euclidean weight for each transition can be shown to be the square of a single output error event. This might suggest using the flow graph to solve for minimum euclidean weight by finding the shortest path from some initial node to a final node. Indeed this can be done and is developed and used here in the form of an XDS Extended Fortran IV program as our performance analysis technique.

As a first step toward the flow graph, a more complete analysis of the whitened, matched filter will be given. Also, an equivalent discrete-time model of the P.A.M. channel and maximum likelihood sequence receiver will be presented. This will lead to definitions of input and output error events, state error events, euclidean weight, and then finally to minimum euclidean weight.

II. CHANNEL MODELS AND THE OPTIMUM RECEIVER

Reference to Figure 1 shows the output $r(t)$ of the noisy channel to be a sum of two signals, white gaussian noise $n(t)$ and signal $s(t)$ given by (1). It is well known that detection of signals that are linear combinations of some set of basis signals is accomplished using a bank of filters each matched to a basis signal [8]. Since the signals of (1) are time translations of $h(t)$ (the channel impulse response), only one filter matched to the channel with samples taken once every T seconds is necessary. Letting $a(t)$ represent the output of $h(-t)$ (matched filter), its samples form a sequence of numbers $a(0), a(1), \dots, a(k), \dots, a(N + L - 1)$.

$$a(k) = \int_{-\infty}^{+\infty} r(t)h(t-kT)dt \quad k = 0, 1, \dots, N + L - 1$$

$$= \sum_{i=0}^{i=N} x_i \int_{-\infty}^{+\infty} h(t-iT)h(t-kT)dt + \int_{-\infty}^{+\infty} n(t)h(t-kT)dt \quad (2)$$

Now, let each integral term of (2) be represented as

$$R_{k-i} = \int_{-\infty}^{+\infty} h(t-kT)h(t-iT)dt \quad (3)$$

and

$$n'_k = \int_{-\infty}^{+\infty} n(t)h(t-kT)dt \quad (4)$$

Equation (2) now has a discrete representation given by

$$a(k) = \sum_{i=0}^{i=N} x_i R_{k-i} + n'_k \quad (5)$$

What can be said about the correlation of the noise samples given by (4)? To begin with, the input noise $n(t)$ is white gaussian with autocorrelation function $\sigma^2 \delta(\tau)$ where σ^2 is the value of the power over a unit bandwidth. 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. To get an exact measure of the correlation of the noise samples, their autocorrelation coefficient must be determined. Taking the expected value of the product $n_k' n_i'$, $0 \leq i$ and $k \leq N + L - 1$, gives

$$E(n_k' n_i') = \sigma^2 R_{k-i} \quad (6)$$

Referring to equation (3), 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., $R_{k-i} = R_{i-k}$.

It was mentioned earlier that the whitened, matched filter has the property that its output samples are statistically independent. This is necessary to insure the simple and recursive nature of the Viterbi algorithm. But, equations (6) and (3) show that the noise samples n_k' are correlated whenever $L > 1$; therefore, the sampled outputs of $h(-t)$ are not statistically independent. Some whitening transformation must be performed. From random variable theory it is known that a linear transformation on a sequence of gaussian random variables is also gaussian. Further, if the sequence is statistically dependent, then there is a linear transformation that will produce a statistically independent sequence.

This is the final process performed by the whitened, matched filter and is accomplished by a transversal filter (tapped delay line). Discussion of the transversal filter is postponed until the discrete time model is introduced.

Recall from Figure 1 that the input to the channel is a sequence of numbers, each separated from its adjacent neighbors by T seconds. Associated with this sequence (or any time sequence of numbers) is a formal power series in D . This series is called the D -transform of $x(t)$ and is given by

$$x(D) = \sum_{i=0}^{i=N} x_i D^i \quad (7)$$

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.

Since the output samples of $h(-t)$ form a sequence with symbol separation T , it too has a D -transform. Equation (5) indicated that each sample $a(k)$ is a function of n_k^i and the first $k + 1$ values for x_i and R_i . This looks as if the D -transform of the sequence $a(0), a(1), \dots$, could be represented in terms of the D -transforms of $x_0, x_1, \dots, x_N$; $n_0^i, n_1^i, \dots, n_N^i$, and the sequence whose coefficients are given by (3). Now equation (5) has terms representing the convolution of the coefficients x_i and R_i . Whenever two polynomials are multiplied, each term of the product is the convolution of terms from each. Hence, the D -transform

of the samples $a(k)$ can be expressed as

$$a(D) = x(D)R(D) + n'(D) \quad (8)$$

where $x(D)$ is the input sequence transform, $n'(D)$ is the correlated noise sample transform and $R(D)$ is the transform of the sequence whose coefficients are given by (3).

Notice, equation (3) is a function of the difference $k-l$. For example,

$$\begin{aligned} R_2 = R_{9-7} &= \int_{-\infty}^{+\infty} h(t-9T)h(t-7T)dt \\ &= R_{3-1} = \int_{-\infty}^{+\infty} h(t-3T)h(t-T)dt \end{aligned}$$

Also, R_{k-i} equals R_{i-k} , making the corresponding sequence symmetrical about the $k-i$ origin. The number of coefficients in the sequence depends on the amount of overlap, or the value of L . All totaled, there are $2(L-1) + 1$ nonzero terms having a D -transform.

$$R(D) = \sum_{i=1-L}^{i=L-1} R_i D^i \quad (9)$$

The D -transform $R(D)$ will be called the autocorrelation function or the pulse autocorrelation function. Since it contains $2L-1$ nonzero terms, it has $2L-2$ complex roots. Since $R(D)$ is symmetrical [$R(D) = R(D^{-1})$] an inverse of a root is also a root. This makes it possible to factor $R(D)$ into a product of polynomials, $f(D)$ and $f(D^{-1})$.

We are now in a position to talk about the transversal filter. This filter, remember, must transform the correlated noise sequence $n'(D)$ into

a white gaussian noise sequence. But the coefficients of (6) are just those of (3) times the noise power per unit bandwidth (σ^2); therefore, the pulse autocorrelation function of $n'(D)$ is

$$R_n'(D) = \sigma^2 R(D) = \sigma^2 f(D)f(D^{-1}) \quad (10)$$

The white noise sequence must have a pulse autocorrelation function that is a constant. Let

$$n'(D) = n(D)f(D^{-1}) \quad (11)$$

where $n(D)$ is our desired white noise sequence. Equation (8) now can be written as

$$a(D) = x(D)f(D)f(D^{-1}) + n(D)f(D^{-1}) \quad (12)$$

Passing the sequence $a(D)$ through a filter with response $f^{-1}(D^{-1})$ (transversal filter) gives

$$z(D) = a(D)/f(D^{-1}) = x(D)f(D) + n(D) \quad (13)$$

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

$$E[n(D)n(D^{-1})] = \frac{E[n'(D)n'(D^{-1})]}{f(D)f(D^{-1})} = \frac{\sigma^2 R(D)}{R(D)} = \sigma^2 \quad (14)$$

where σ^2 is a constant. Equation (13) tells us that the P.A.M. channel and whitened, matched filter can be represented as a discrete time filter with impulse response $f(D)$ plus an additive white gaussian noise sequence (see Figure 2).

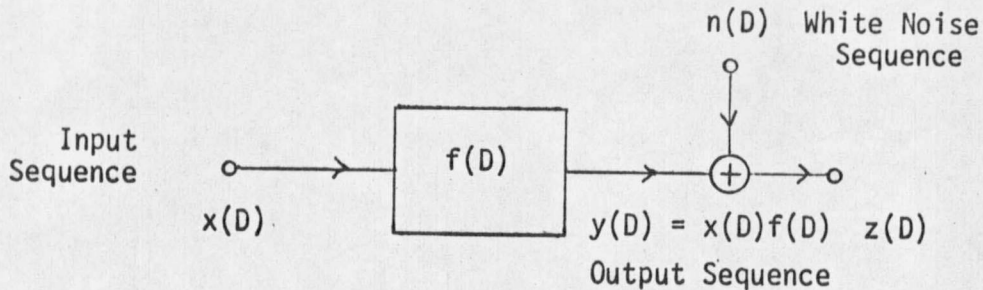


Figure 2. Discrete time model of P.A.M. system.

Nothing at this point has been said about the overall time response, $w(t)$. What kind of waveform is it? Does indeed the set of functions $w(t-kT)$, $k = 0, 1, \dots, N + L - 1$, form an orthonormal basis, insuring the samples z_k , $k = 0, 1, \dots, N + L - 1$ are a set of sufficient statistics for estimating $x(D)$? These questions can be answered by looking at another discrete time transformation, the chip D-transform.

Think of the response $h(t)$ as a sum of L mutually exclusive chips $h_i(t)$, $0 \leq i \leq L - 1$, where $h_i(t) = h(t + iT)$ for $0 \leq t \leq T$ and zero elsewhere,

$$h(t) = \sum_{i=0}^{L-1} h_i(t-iT) \quad (15)$$

Now take the D-transform of the chip sequence, $i = 0, 1, \dots, L - 1$,

$$h(D,t) = \sum_{i=0}^{L-1} h_i(t)D^i \quad (16)$$

The function $h(D,t)$ is called the chip D-transform of $h(t)$.

The function $f(D^{-1})$ is a polynomial in D^{-1} of degree $L - 1$. If the division $1 / f(D^{-1})$ is carried out, what results is possibly an infinite time sequence. Remember, (Figure 1) the input sequence $x_0, x_1, \dots$, had

continuous time representation $x(t) = \sum_{i=0}^{i=N} x_i \delta(t-iT)$. Likewise, the trans-

versal filter sequences can be generally represented in terms of a weighted sum of dirac delta function time translations.

$$g(t) = \sum_{i=0}^{i=\infty} g_i \delta(t-iT) \quad (17)$$

where $g(t)$ is its impulse response.

Since $g(t)$ is a time function, it has a chip D-transform.

Each chip $g_i(t)$ is a weighted delta function, $g_i \delta(t)$. Therefore

$$g(D, t) = g_i \delta(t) D^i = g(D) \delta(t) \quad (18)$$

and also

$$x(D, t) = x(D) \delta(t). \quad (19)$$

Now the whitened, matched filter is represented in terms of a matched filter $h(-t)$ [chip D-transform $h(D^{-1}, t)$] in cascade with a transversal filter $g(D) = f^{-1}(D^{-1})$. What is the chip D-transform of $w(-t)$? Note, that $h(D, t)$ and $g(D, t)$ are functions of both D and T , therefore,

$$\begin{aligned} w(D^{-1}, t) &= h(D^{-1}, t) * g(D, t) = h(D^{-1}, t) * \delta(t) g(D) \\ &= h(D^{-1}, t) g(D) = h(D^{-1}, t) / f(D^{-1}) \end{aligned} \quad (20)$$

A similar development shows that for the output signal $s(t)$ of the channel,

$$s(D,t) = h(D,t)x(D) \quad (21)$$

Equation (9) gives the pulse autocorrelation function of $h(t)$ where the coefficients are given by (3). $R(D)$ has another representation in terms of the chip D-transform $h(D,t)$:

$$R(D) = \int_{-\infty}^{+\infty} h(D,t)h(D^{-1},t)dt \quad (22)$$

How about the orthonormality of $w(t)$ or $w(D,t)$? Does the set $w(t-kT)$, $k = 0; \dots, N + L - 1$, represent an orthonormal basis? Let's look at the pulse autocorrelation function of $w(t)$. From (22) and (20),

$$R_w(D) = \int_0^{\infty} w(D,t)w(D^{-1},t)dt = \int_0^{\infty} \frac{h(D,t)h(D^{-1},t)}{f(D)f(D^{-1})} dt \quad (23)$$

$$= \frac{R(D)}{f(D)f(D^{-1})} = 1 \quad (23)$$

$R_w(D) = 1$ indicating the set $w(t-kT)$, $k = 0, \dots, N + L - 1$, is orthonormal, therefore that the samples z_k , $k = 0, 1, \dots, N + L - 1$, are a set of sufficient statistics for the estimation of $x(D)$.

The whitened, matched filter now consists of a matched filter, $h(-t)$, in cascade with a transversal filter, $f^{-1}(D^{-1})$ (see Figure 3) with a chip D-transform $w(D,t)$ given by (20).

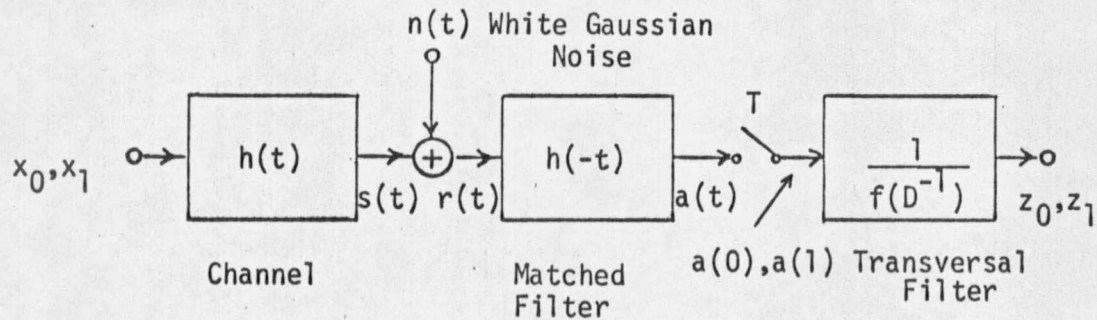


Figure 3. P.A.M. channel with white gaussian noise, filter matched to $h(t)$, and transversal filter with discrete time response $1/f(D^{-1})$.

To gain deeper insight into the communication system illustrated by Figure 3, let's work through a running example. Let's assume the transmission channel has an impulse response given by the flat pulse of Figure 4(a). Also, let's consider the case where the symbols to be transmitted are binary. Specifically, consider an input sequence $x_0 = 1, x_1 = 1, x_2 = 0, x_3 = 1$, and $x_4 = 1$. Therefore, from (15)

$$s(t) = h(t) + h(t-1) + h(t-3) + h(t-4)$$

The channel can be thought of as a three-stage shift register and an adder (Figure 5).

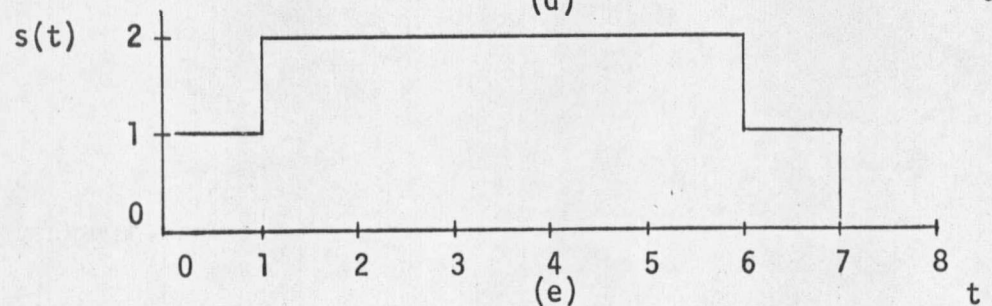
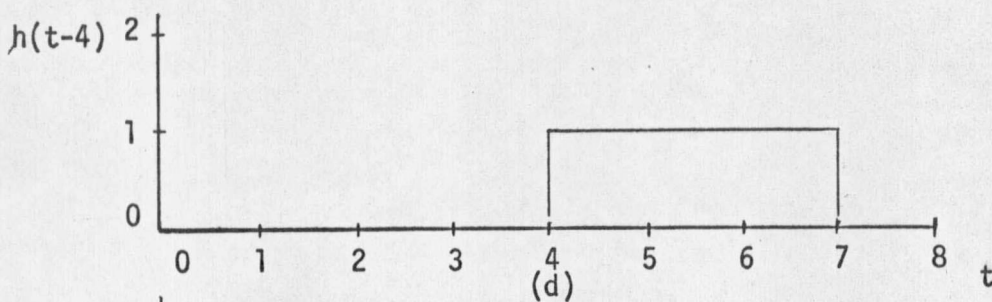
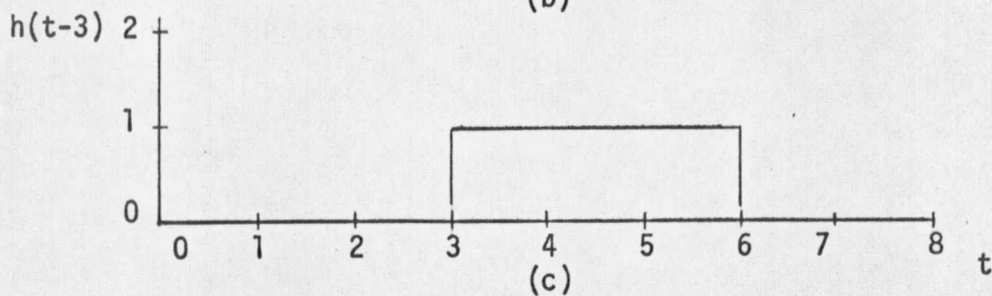
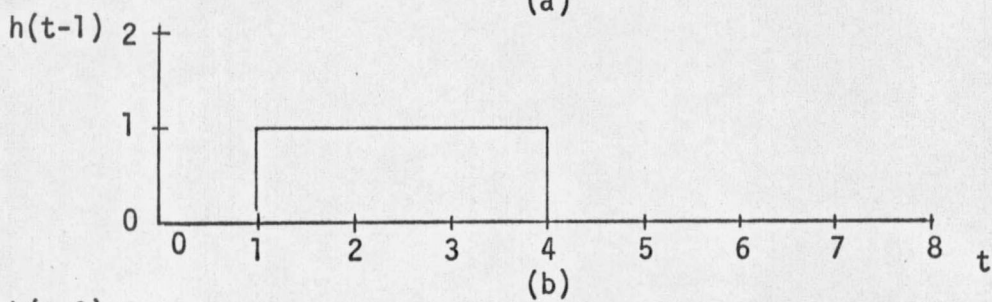
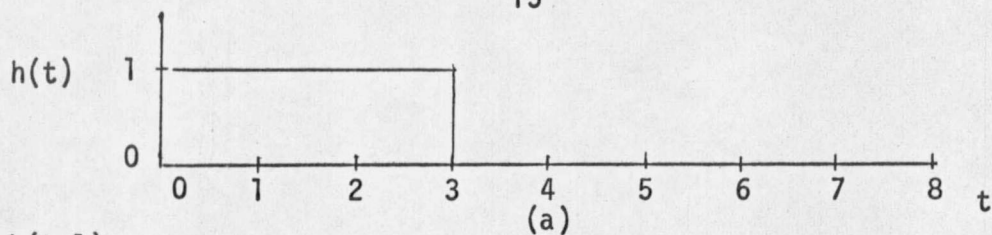


Figure 4. Channel response to (a) $x_0 = 1$, (b) $x_1 = 1$, (c) $x_3 = 1$, (d) $x_4 = 1$ and (e) $x_0 = 1$, $x_1 = 1$, $x_2 = 0$, $x_3 = 1$ and $x_4 = 1$.

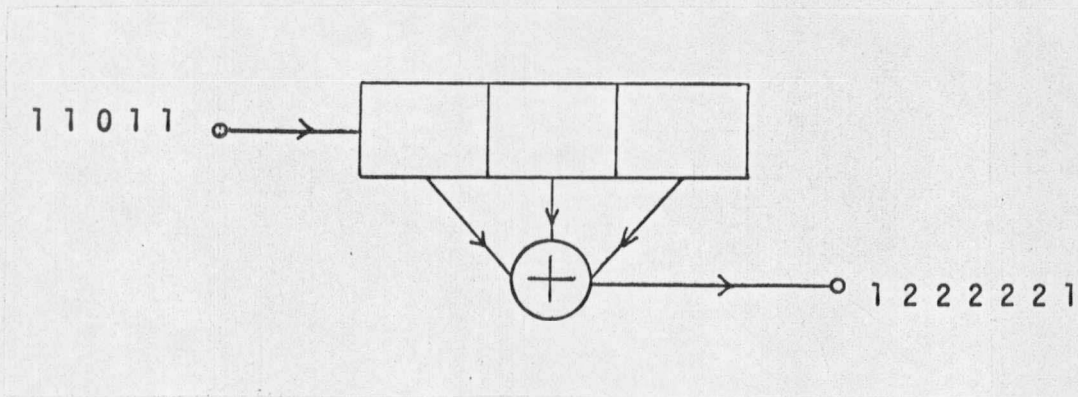


Figure 5. Convolutional encoder equivalent to the flat top pulse channel.

White gaussian noise is added to the signal, $r(t) = s(t) + n(t)$. We now want to estimate the transmitted sequence. From equations (2) and (4)

$$a(k) = s'_k + n'_k$$

where

$$s'_k = \sum_{i=0}^{i=4} x_i \int_{-\infty}^{+\infty} h(t-iT)h(t-kT)dt$$

$$s'_0 = \sum_{i=0}^{i=4} x_i \int_{-\infty}^{+\infty} h(t)h(t)dt = 3 + 2 = 5$$

$$s'_1 = \sum_{i=0}^{i=4} x_i \int_{-\infty}^{+\infty} h(t)h(t-1)dt = 2 + 3 + 1 = 6$$

$$s'_2 = \sum_{i=0}^{i=4} x_i \int_{-\infty}^{+\infty} h(t)h(t-2)dt = 1 + 2 + 2 + 1 = 6$$

$$s'_3 = \sum_{i=0}^{i=4} x_i \int_{-\infty}^{+\infty} h(t)h(t-3)dt = 1 + 3 + 2 = 6$$

$$s'_4 = \sum_{i=0}^{i=4} x_i \int_{-\infty}^{+\infty} s(t)h(t-4)dt = 2 + 3 = 5$$

Now, take the D-transform according to (7),

$$s'(D) = 5 + 6D + 6D^2 + 6D^3 + 5D^4$$

But from (8)

$$s'(D) = x(D)R(D)$$

Let's try finding $R(D)$ by using the above expression $x(D)R(D)$. The input sequence is 1 1 0 1 1, so its D-transform is

$$x(D) = 1 + D + D^3 + D^4$$

We can solve for $R(D)$ using one of two methods, either by (22) or (9).

Let's try both. To use (22) we must solve for the chip D-transform $h(D,t)$. From (15)

$$h(t) = h_0(t) + h_1(t-1) + h_2(t-2)$$

where $h_i(t)$ is given by Figure 6.

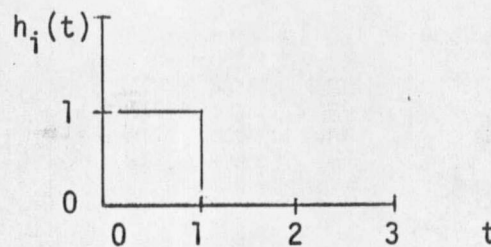


Figure 6. Chip function for $i = 0, 1$, and 2 .

Now, applying the definition of the chip D-transform (16)

$$\begin{aligned} h(D,t) &= h_0(t) + h_1(t)D + h_2(t)D^2 \\ &= h_1(t) (1 + D + D^2) \end{aligned}$$

Let's solve for $R(D)$ using both methods.

1. From (22)

$$\begin{aligned} R(D) &= \int_0^1 h(D,t)h(D^{-1},t)dt \\ &= (1 + D + D^2)(1 + D^{-1} + D^{-2}) \int_0^1 h_1^2(t)dt \\ R(D) &= D^{-2} + 2D^{-1} + 3 + 2D + D^2 \\ &= (1 + D + D^2)(1 + D^{-1} + D^{-2}) = f(D)f(D^{-1}) \end{aligned}$$

2. From (3) and (9)

$$\begin{aligned} R_0 &= R_0 = \int_0^3 h^2(t)dt = 3 \\ R_1 &= R_{-1} = \int_0^2 h(t)h(t+1)dt = 2 \\ R_2 &= R_{-2} = \int_0^1 h(t)h(t+2)dt = 1 \end{aligned}$$

hence

$$R(D) = 1D^{-2} + 2D^{-1} + 3D + 2D + 1D^2$$

Now $s'(D)$ is

$$s'(D) = x(D)R(D) = D^{-2} + 3D^{-1} + 5 + 6D - 6D^2 + 6D^3 + 5D^9 + 3D^5 + D^6$$

Notice the extra terms in $s'(D)$. Multiplication of $R(D)$ and $x(D)$ implies convolution of the time sequences. This assumes sampling started at minus infinity and terminated at plus infinity. But, the input sequence $s'(D)$ started at time $k = 0$ and ended at time $k = 4$, so, only those terms where, $k = 0, \dots, 4$ of $s'(D)$ need to be remembered.

Now, let's look at the waveform

$$w(D,t) = \frac{h(D,t)}{f(D)} = \frac{h_i(t)(1 + D + D^2)}{1 + D + D^2} = h_i(t)$$

This implies that $w(t) = h_i(t)$.

Next, let's find the samples z_k for $k = 0, \dots, 6$.

$$\begin{aligned} z_k &= \int_{-\infty}^{+\infty} r(T)w(T-k)dt \\ &= \int_{-\infty}^{+\infty} [s(t) + n(t)]w(t-k)dt \end{aligned}$$

The signal portions of z_k , $k = 0, \dots, 6$, are

$$z_k^i = \int_{-\infty}^{+\infty} s(t)h_i(t-k)dt$$

$$z_0^i = 1, z_1^i = 2, z_2^i = 2, z_3^i = 2, z_4^i = 2, z_5^i = 2, z_6^i = 1$$

Now, refer to the discrete time model of Figure 2 where

$$\begin{aligned}
 y(D) &= x(D)f(D) \\
 &= (1 + D + D^3 + D^4)(1 + D + D^2) \\
 &= 1 + 2D + 2D^2 + 2D^3 + 2D^4 + 2D^5 + 1D^6
 \end{aligned}$$

Notice, this is the D-transform of the signal portions of z_k , $k = 0, \dots, 6$.

The Viterbi algorithm requires the samples z_k , $k = 0, 1, \dots, 6$, be statistically independent gaussian random variables. Since only the noise portions of the samples are random, let's look at their autocorrelation coefficient.

$$\begin{aligned}
 E(n_k n_j) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \overline{n(t)n(s)w(t-k)w(s-j)} dt ds \\
 &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sigma^2 \delta(t-s) w(t-k) w(s-j) dt ds
 \end{aligned}$$

$$E(n_k n_j) = 0; k \neq j$$

$$= \sigma^2 \int_{-\infty}^{+\infty} h_i^2(t-j) dt = \sigma^2; k = j$$

Hence, the noise sequence is independent (white).

III. PERFORMANCE ANALYSIS

Comparison of the output sequence $y(D)$ and input sequence $x(D)$ suggests that the flat-top pulse with $L = 3T$ creates severe intersymbol interference. How severe is the intersymbol interference. What fractional reduction in the minimum euclidean weight does the flat-top pulse produce? This question leads directly to the definition of an error event E and its euclidean weight $d^2(E)$.

Let's assume that the P.A.M. transmitter sends the sequence

$$x_0 = 1, x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$$

but, the receiver estimates the transmitted sequence as

$$\hat{x}_0 = 1, \hat{x}_1 = 0, \hat{x}_2 = 1, \hat{x}_3 = 1, \hat{x}_4 = 1$$

These sequences have D-transforms

$$x(D) = 1 + D + D^3 + D^4$$

and

$$\hat{x}(D) = 1 + D^2 + D^3 + D^4,$$

respectively. Now, taking the difference between the actual transmitted sequence and the estimated sequence gives the input error sequence

$$x_0 - \hat{x}_0 = 0, x_1 - \hat{x}_1 = +1, x_2 - \hat{x}_2 = -1, x_3 - \hat{x}_3 = 0, x_4 - \hat{x}_4 = 0$$

with D-transform

$$\begin{aligned} E_x(D) &= (x_0 - \hat{x}_0)D^0 + (x_1 - \hat{x}_1)D^1 + (x_2 - \hat{x}_2)D^2 + (x_3 - \hat{x}_3)D^3 + (x_4 - \hat{x}_4)D^4 \\ &= 0 + D - D^2 + 0D^3 + 0D^4 \\ &= D(1 - D) = x(D) - \hat{x}(D) \end{aligned}$$

If the receiver made a wrong estimate of $x(D)$, what about its estimate of $y(D)$ [$\hat{y}(D)$]? From Figure 2

$$y(D) = f(D)x(D) \quad (25)$$

hence

$$\hat{y}(D) = f(D)\hat{x}(D) \quad (26)$$

The difference between $y(D)$ and $\hat{y}(D)$ is

$$\begin{aligned} E_y(D) &= y(D) - \hat{y}(D) \\ &= f(D)[x(D) - \hat{x}(D)] = f(D)E_x(D) \end{aligned} \quad (27)$$

Let's look at $E_y(D)$ for the flat top pulse [$f(D) = 1 + D + D^2$] and input error sequence $D - D^2$:

$$\begin{aligned} E_y(D) &= (1 + D + D^2)(1 - D)D \\ &= D - D^4 = y(D) - \hat{y}(D) \end{aligned}$$

Consider the two sequences whose D -transforms are $y(D)$ and $\hat{y}(D)$. Each is a possible output vector contained in the output vector space. The distance between the two vectors is just the euclidean distance (the square root of the sum of the squares of the difference in each component). The euclidean weight is just the square of this. But, $E_y(D)$ is a polynomial in D whose coefficients are the differences of the components of $y(D)$ and $\hat{y}(D)$, therefore, the euclidean weight is just the sum of the squares of

the components of $E_y(D)$, and given as

$$d^2(E) = \sum_j E_{y_j}^2 \quad (28)$$

Equation (28) can be obtained by another method. Consider the polynomial

$$\begin{aligned} E_y(D)E_y(D^{-1}) &= E_x(D)f(D)f(D^{-1})E_x(D^{-1}) \\ &= E_x(D)R(D)E_x(D^{-1}) \end{aligned} \quad (29)$$

The D^0 coefficient of this polynomial is given by (28), hence

$$d^2(E) = [E_y(D)E_y(D^{-1})]_0 \quad (30)$$

Let's determine the euclidean weight of our running example. Applying (28), we have

$$d^2(E) = 1 + 1 = 2$$

Is this the smallest euclidean weight possible with the flat top pulse?

Let's look at the euclidean weight of the flat top pulse for several other input error sequences and see.

Let's first try single errors

1. A single error

$$E_x(D) = \pm D^k, \quad 0 \leq k \leq N$$

$$E_x(D) = \pm D^k(1 + D + D^2)$$

$$d^2(E) = [(1 + D + D^2)(+D^{-k})(+D^{-k})(1 + D^{-1} + D^{-2})]_0$$

$$= [R(D)]_0 = 3$$

Notice, since $(+D^k)(+D^{-k}) = +1$, $d^2(E)$ is independent of overall delay and sign. From example (1) above, we know single errors do not give smaller euclidean weights. So, let's try some more double error patterns.

2. $E_x(D) = 1 - D$

$$E_y(D) = (1 - D)(1 + D + D^2) = 1 - D^3$$

$$d^2(E) = 1 + 1 = 2$$

3. $E_x(D) = D^2 - D^3$

$$E_y(D) = D^2(1 - D^3)$$

$$d^2(E) = [(1 - D^3)D^2 D^{-2} (1 - D^{-3})]_0 = 2$$

As for the single errors, for double errors $d^2(E)$, is independent of overall delay.

4. $E_x(D) = 1 + D^k \quad 1 \leq k \leq N$

$$E_y(D) = (1 + D^k)(1 + D + D^2)$$

$$= 1 + D + D^2 + D^k + D^{k+1} + D^{k+2}$$

$$k = 1$$

$$d^2(E) = 1 + 2^2 + 2^2 + 1 = 10$$

$$k = 2$$

$$d^2(E) = 1 + 1 + 2^2 + 1 + 1 = 8$$

$$k \geq 3$$

$$d^2(E) = 1 + 1 + 1 + 1 + 1 + 1 = 6$$

$$5. \quad E_x(D) = 1 - D^k \quad 2 \leq k \leq N$$

$$E_x(D) = (1 - D^k)(1 + D + D^2)$$

$$= 1 + D + D^2 - D^k - D^{k+1} - D^{k+2}$$

$$k = 2$$

$$d^2(E) = 1 + 1 + 1 + 1 = 4$$

$$k' \geq 3$$

$$d^2(E) = 1 + 1 + 1 + 1 + 1 + 1 = 6$$

Let's try some triple errors.

$$6. \quad E_x(D) = 1 - D + D^2$$

$$E_y(D) = 1 + D^2 + D^4$$

$$d^2(E) = 1 + 1 + 1 = 3$$

7. $E_X(D) = 1 + D + D^2$

$$d^2(E) = 1 + 4 + 9 + 4 + 1 = 19$$

8. $E_X(D) = 1 - D - D^2$

$$d^2(E) = 7$$

IV. ANALYSIS USING FLOW GRAPHS

From the examples given, we can guess that maybe $d^2(E) = 2$, where $E_x(D) = D^k(1 - D)$, is the minimum euclidean weight for the flat top pulse. But, we need a more reliable method that will always give the minimum euclidean weight for an arbitrary $h(t)$ or $f(D)$. One such method is to find the shortest path through a flow graph representing the euclidean weight distribution of the output error sequences.

$$E_y(D) = E_x(D)f(D)$$

Let's take a look at the i th coefficient of $E_x(D)$, E_{y_i} , $i = 0, 1, \dots, N + L - 1$. Each such coefficient of the product of $E_x(D)$ is given as

$$E_{y_i} = \sum_{k=0}^{k=m} E_{x_{i-k}} f_k \quad (31)$$

where

$$m = \begin{cases} L - 1 & \text{if } L - 1 \leq i \\ i & \text{if } L - 1 > i \end{cases}$$

What we need is a flow graph such that a particular path through the graph will give us the coefficients of a particular output error sequence.

From (31), the state error,

$$E_{s_i} = (E_{x_{i-1}}, E_{x_{i-2}}, \dots, E_{x_{i-L}}) \quad (32)$$

can be defined. Knowing two successive states E_{s_i} and $E_{s_{i+1}}$, we can find the output coefficient E_{y_i} . If we let each node of the graph be a

possible state E_{s_i} , then passing through it amounts to starting at node E_{s_0} and progressing sequentially through to node $E_{s_{N+L-1}}$. Each link between the state E_{s_i} and $E_{s_{i+1}}$ has assigned to it $E_{y_i}^2$, therefore, summing the $E_{y_i}^2$'s for each path will give the euclidean weight of that path.

For a binary input alphabet, which is assumed throughout the entire paper, each element of a state takes one of three possible values

$$E_{x_k} = x_k - \hat{x}_k = \begin{cases} 1 \\ 0 \\ -1 \end{cases} \quad (33)$$

This means there are 3^{L-1} different states of the graph. To insure that all nonzero coefficients of $E_y(D)$ are determined, the final node must be the all zero state. This makes a graph of $3^{L-1} + 1$ nodes, the initial and final nodes being the all zero state.

Since, transition through the graph is done sequentially, links from nodes with state (32) must go only to nodes with state

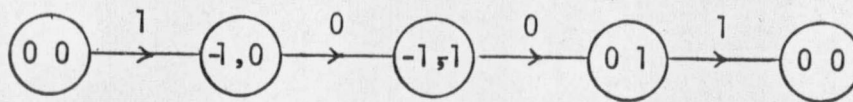
$$E_{s_{i+1}} = (E_{x_i}, E_{x_{i-1}}, \dots, E_{x_{i-L+1}}) \quad (34)$$

But, state (34) contains $L-2$ elements that are also contained in state (32). The only element of (34) that is not contained in (32) is E_{x_i} .

Since it takes on only three values, there are three links leaving each node and three links going to each node. Now since we are interested

only in nonzero $d^2(E)$ values, there is no link from the initial node to the final node; hence, there are only two links leaving the initial node and two links going to the final node.

To see what this flow graph looks like, let's construct it for the flat top pulse with discrete impulse response $f(D) = 1 + D + D^2$. Since $L = 3$, there are 10 nodes composing the graph (see Figure 7). By inspection we see that two shortest paths through the graph are

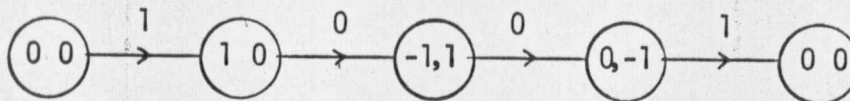


$$d^2_{\min} = 1 + 0 + 0 + 1$$

Input error 0 0, -1 1 0 0

sequence

and



$$d^2_{\min} = 1 + 0 + 0 + 1 = 2$$

Input error 0 0 1, -1 0 0

sequence

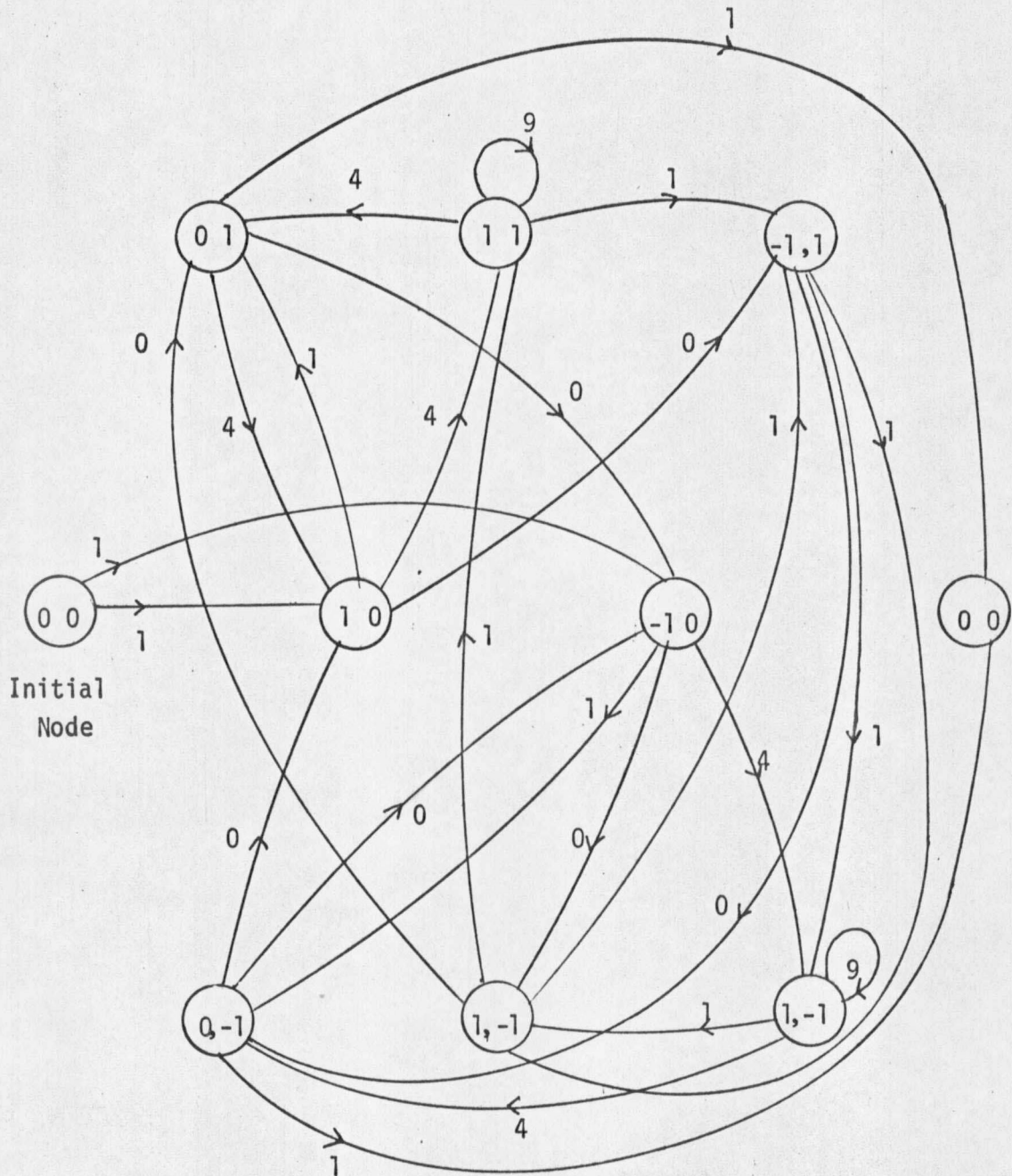


Figure 7. Flow graph of flat top pulse example.

Look at the two loops shown below in Figure 8.

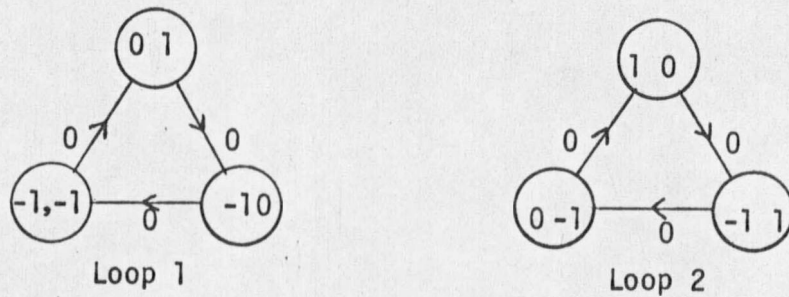


Figure 8. Closed loop paths adding zero weight.

We can travel around these paths any number of times and not add weight. All paths containing these loops have $d^2(E) = 2$, corresponding to minimum distance paths.

The flow graph is redrawn with all except minimum distance routes deleted (Figure 9).

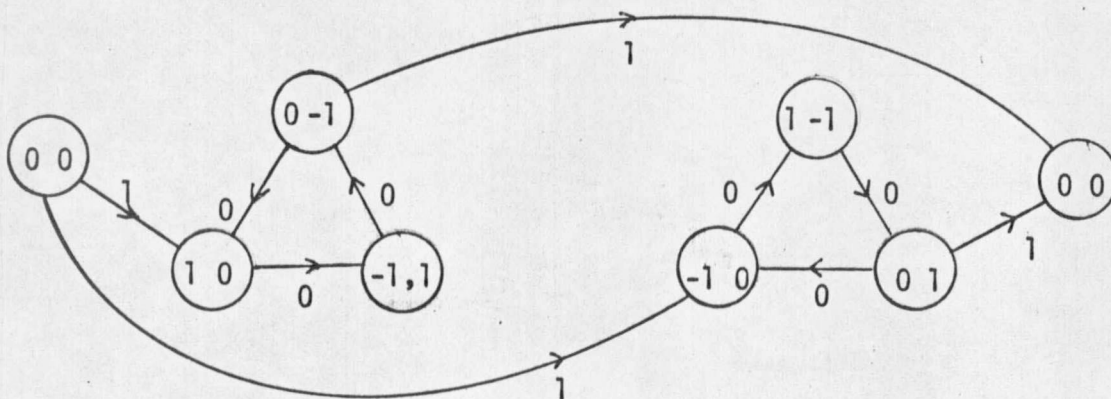


Figure 9. Flow graph with minimum distance paths only.

Using a flow graph, we are able to find the minimum distance for a given discrete impulse response $f(D)$. But, this still does not tell us how severe the intersymbol interference is. We need to find $d^2_{\min}$ when there is no intersymbol interference. Then the fractional difference in minimum euclidean weight can be found. Consider the case where only a single symbol x_k is transmitted. The whitened, matched filter would have to process only one pulse, so there would be no intersymbol interference. Since we are assuming a binary alphabet, there are only two input sequences possible

$$x_1(D) = D^k \quad 0 \leq k \leq N$$

and

$$x_2(D) = 0$$

If the transmitter sends $x_1(D)$, but the receiver estimates $x(D)$ to be $\hat{x}(D) = x_2(D)$ instead, then

$$E_x(D) = x(D) - \hat{x}(D) = x_1(D) - \hat{x}_1(D) = D^k$$

therefore,

$$E_y(D) = D^k f(D)$$

and

$$\begin{aligned} d^2(E) &= [f(D) D^k D^{-k} f(D^{-1})]_0 = [f(D) f(D^{-1})]_0 \\ &= \sum_{i=0}^{L-1} f_i^2 \end{aligned} \quad (35)$$

Equation (35) would not change if instead the transmitter sent a 0 and the receiver made a wrong decision. Since (35) is true for all of the nonzero $E_x(D)$, it is the minimum euclidean weight with no intersymbol interference.

Let's find $d_{\min}^2$ with no intersymbol interference for the flat top pulse, $f(D) = 1 + D + D^2$. Using (35)

$$\begin{aligned} d_{\min}^2 &= [(1 + D + D^2)(1 + D^{-1} + D^{-2})]_0 \\ &= [1 + 2D^{-1} + 3 + 2D + 1]_0 \\ &= 3 \end{aligned}$$

This is the same result obtained with a single input error sequence with intersymbol interference (Example 1).

What is the fractional difference in $d_{\min}^2$ due to intersymbol interference with the flat top pulse? Using (35) and $d_{\min}^2 I = 2$ (intersymbol interference),

$$\begin{aligned} (d_{\min}^2 - d_{\min}^2 I) / d_{\min}^2 &= \frac{3-2}{3} \\ &= .333 \dots \end{aligned}$$

This says that the flat top pulse reduces the minimum euclidean weight by one-third.

Generalizing the above for an arbitrary $f(D)$ gives the fractional difference

$$f_d = (d_{\min}^2 - d_{\min}^2 I) / d_{\min}^2 \quad (36)$$

where

$d_{\min}^2$ = minimum euclidean weight without intersymbol interference.

$d_{\min}^2 I$ = minimum euclidean weight with intersymbol interference.

Expression (36) is the criterion that will be used to compare the performance of our maximum likelihood sequence receiver for several channel impulse responses. Its performance can also be evaluated by comparing the upper bounds on the minimum probability of error. The interested reader is referred to reference [2].

The impulse response given as a running example was for the case $L = 2$. What about when L is greater than 3? Will a minimum distance path through the flow graph be as easy to find? If $L = 4$, that means the number of nodes in the graph is twenty-eight. For $L = 5$, there are 82 nodes. As L becomes larger, the graph grows exponentially as $3^{L-1} + 1$, making the determination of a minimum distance path increasingly difficult. This makes it necessary to use computer programming to find the minimum distance path.

It was mentioned at the beginning of this paper that its objective is to develop a technique for analyzing the performance of the maximum likelihood sequence estimator for known channels. The flow graph analysis

was introduced as a part of this objective. To make the flow graph analysis useful for arbitrary values of L , an XDS Extended Fortran IV computer program has been written that sets up the graph and finds a minimum distance path through it for arbitrary L and $f(D)$ pairs (channel impulse responses).

The minimum distance path finder portion of the program uses an algorithm introduced by W. E. Dijkstra for determining shortest paths through a graph [11, 12]. Let's take a look at this algorithm and use it to find the minimum distance for our running example.

This particular shortest path algorithm finds the shortest route from a given starting node, N_s , to some other given node N_f in the graph. It accomplishes this by finding the shortest path from this node to all other nodes N_i of the graph. The reason for doing this is that any node N_k may be an intermediate node on the shortest path from N_s to N_f . If there is more than one shortest path from N_s to N_i , $i \neq s$ algorithm deletes all but one. The resulting graph, after all deletions is a tree with $N - 1$ (N is the number of nodes in the flow graph) links $L(i,j)$ ($L(i,j)$ is the path from N_i to N_j with no other nodes on that path). Each link $L(i,j)$ is the shortest link between N_i and N_j .

As mentioned above, the algorithm builds a tree which contains shortest paths from the starting node N_s to all other nodes. At the beginning, there are no links of the tree so all links are nontree links. The algorithm then attempts to increase the number of tree links until

there are $N - 1$ in number.

To begin, let L_{sk} be the actual shortest distance from N_s to N_k and let L'_{sk} be the shortest distance from N_s to N_k using tree links and at least one nontree link. If the path from N_s to N_k needs more than one nontree link then $L'_{sk} = \infty$. Now, if $L(i,k)$ is the link between N_i and N_k where N_i is contained in the tree, then it has a distance $0 \leq d_{ik} \leq \infty$. If there is no link between N_i and N_k then $d_{ik} = \infty$. Also, if node N_k is a neighboring node of the existing tree, whose nodes are designated N_i , then $0 \leq d_{ik} \leq \infty$. The algorithm then examines all neighboring nodes of the tree, and for each determines the minimum L'_{sk} for all N_i :

$$L'_{sk} = \min_i (L_{si} + d_{ik}) \quad (37)$$

It then determines the minimum L'_{sk} for all N_k ,

$$L'_{sk} = \min_k L'_{sk} \quad (38)$$

and makes this node N_p a tree node with distance

$$L_{sp} = L'_{sp} = L_{si} + d_{ip} \quad (39)$$

Link $L(i,p)$ now becomes a new tree link with distance d_{ip} .

Each time equations (37) through (39) are executed, a new link is added to the tree. This means that we must recalculate L'_{sk} for all neighboring nodes of the new tree. To do this, we look at all nodes neighboring N_p and compare L'_{sk} determined earlier with $L_{sp} + d_{pi}$. If L'_{sk} is larger than $L_{sp} + d_{pi}$, the new L'_{sk} becomes $L_{sp} + d_{pi}$. If L'_{sk} is

smaller no change is made. This operation is indicated by

$$L'_{sk} = \min (L'_{sk}, L_{sp} + d_{pk}) \quad (40)$$

The above algorithm is summarized using the following steps.

- I. To start, N_s is the only node of the tree so $L'_{sk} = d_{sk}$ and $L_{ss} = 0$.
- II. The minimum of all L'_{sk} is found, $L'_{sp} = \min_k L'_{sk}$, and $L(i,p)$ is made a new tree link, $L_{sp} = L'_{sp}$.
- III. The number of links of the tree is compared with $N - 1$. If it becomes equal to this value the algorithm is terminated.
- IV. Re-examine each L'_{sk} found. Let $L'_{sk} = \min (L'_{sk}, L_{sp} + d_{pk})$.
- V. Return to Step II.

To carry out the above algorithm, each node, N is given a label (L,i) . L is either L'_{sk} or L_{sk} depending on whether N_k is a tree node or a nontree node. The last node on the shortest path from N_s to N_k is designated as i .

Let's use the above algorithm to find $d^2_{\min}$ with the flow graph of the flat top pulse. Its graph is redrawn here as Figure 10. Each node has been given a designation N_i , $i = 1, 2, \dots, 10$.

To begin, the tree contains only one node N_1 and no tree links. The single node N_1 has two neighbors N_4 and N_7 . We give them temporary labels

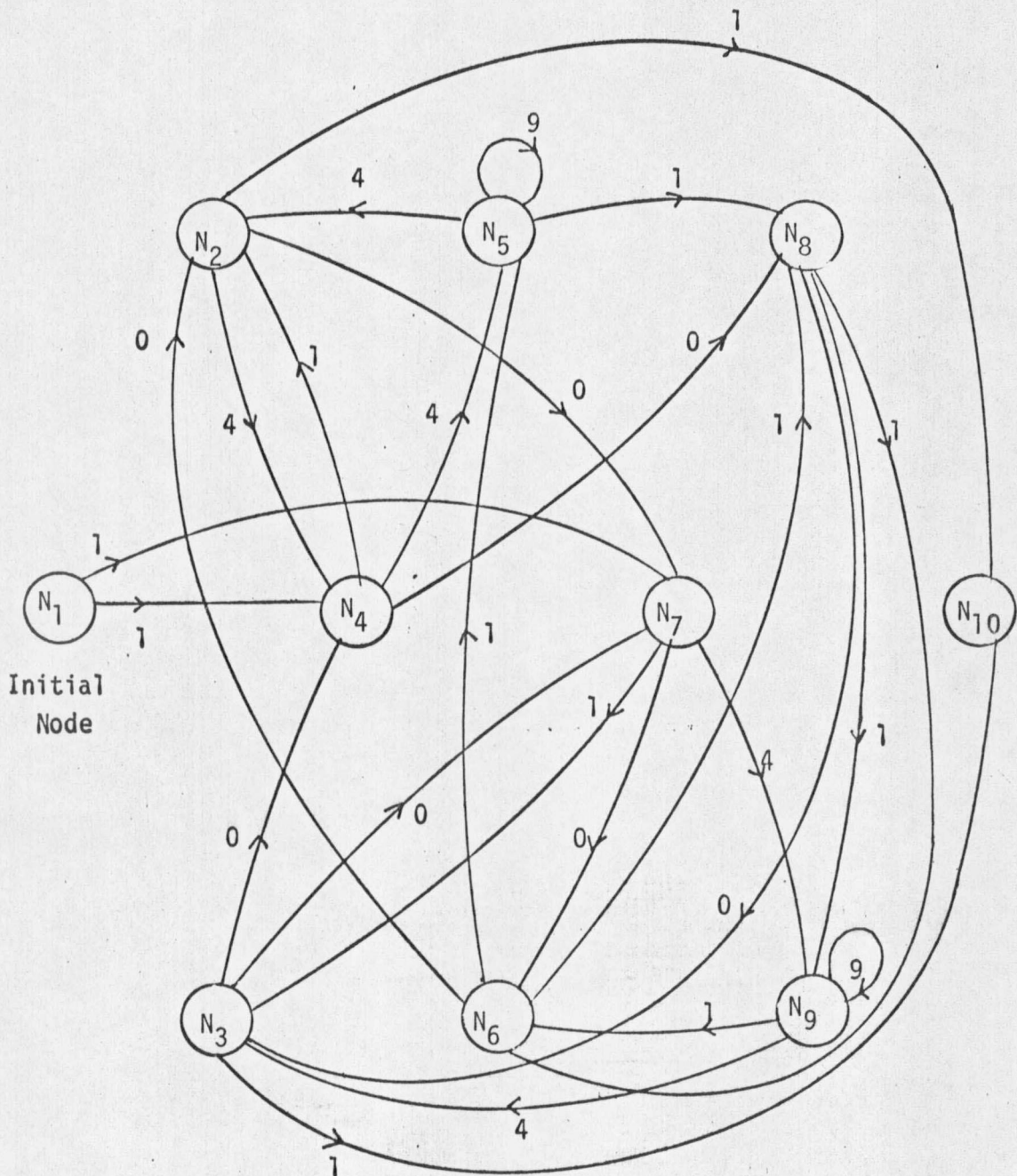


Figure 10. Flow graph of flat top pulse example.

$(L'_{14}, 1) = (d_{14}, 1) = (1, 1)$ and $(L'_{17}, 1) = (1, 1)$. Now, $\min(1, 1) = 1$, but both have distance one; therefore, arbitrarily chose either, say N_7 .

The tree now consists of the link $L(1, 7)$ only.

The neighboring nodes of the tree are N_3, N_5, N_6, N_9 , hence;

$$L'_{13} = \min(L'_{13}, L_{17} + d_{73}) = \min(\infty, 1 + 1) = 2$$

$$L'_{14} = \min(L'_{14}, L_{17} + d_{74}) = \min(1, 1 + \infty) = 1$$

$$L'_{16} = \min(L'_{16}, L_{17} + d_{76}) = \min(\infty, 1 + 0) = 1$$

$$L'_{19} = \min(L'_{19}, L_{17} + d_{79}) = \min(\infty, 1 + 4) = 5$$

Since L'_{14} and L'_{16} are equal, arbitrarily chose one, say L'_{14} . Node N_4 now has the permanent label $(1, 1)$. The tree consists of the two links $L(1, 4)$ and $L(1, 7)$. Continuing, the neighboring nodes are N_2, N_5, N_6, N_3, N_8 , and N_9 , therefore;

$$L'_{12} = \min(L'_{12}, L_{14} + d_{24}) = \min(\infty, 1 + 1) = 2$$

$$L'_{13} = \min(L'_{13}, L_{14} + d_{34}) = \min(2, 1 + \infty) = 2$$

$$L'_{15} = \min(L'_{15}, L_{14} + d_{45}) = \min(\infty, 1 + 4) = 5$$

$$L'_{16} = \min(L'_{16}, L_{14} + d_{46}) = \min(1, 1 + \infty) = 1$$

$$L'_{18} = \min(L'_{18}, L_{14} + d_{48}) = \min(\infty, 1 + 0) = 1$$

$$L'_{19} = \min(L'_{19}, L_{14} + d_{49}) = \min(5, 1 + \infty) = 5$$

Since L'_{18} is one of the smallest it becomes a permanent label $(1, 4)$ and $L(4, 8)$ becomes a tree link. Proceeding, the neighboring nodes are now

N_2, N_3, N_5, N_6 and N_9 with distances

$$L'_{12} = \min (L'_{12}, L_{18} + d_{82}) = \min (2, 1 + \infty) = 2$$

$$L'_{13} = \min (L'_{13}, L_{18} + d_{83}) = \min (2, 1 + 0) = 1$$

$$L'_{15} = \min (L'_{15}, L_{18} + d_{85}) = \min (5, 1 + \infty) = 5$$

$$L'_{16} = \min (L'_{16}, L_{18} + d_{86}) = \min (1, 1 + 1) = 1$$

$$L'_{19} = \min (L'_{19}, L_{18} + d_{89}) = \min (5, 1 + 1) = 2$$

Now choose L'_{16} as a permanent label (1,7). Link $L(6,7)$ becomes a tree link and N_3, N_2, N_5 and N_9 are the neighboring nodes. Pushing on,

$$L'_{12} = \min (L'_{12}, L_{16} + d_{62}) = \min (2, 1 + 0) = 1$$

$$L'_{13} = \min (L'_{13}, L_{16} + d_{63}) = \min (2, 1 + \infty) = 2$$

$$L'_{15} = \min (L'_{15}, L_{16} + d_{65}) = \min (5, 1 + 1) = 2$$

$$L'_{19} = \min (L'_{19}, L_{16} + d_{69}) = \min (2, 1 + \infty) = 2$$

The smallest is L'_{12} and it becomes a permanent label (1,6). There are five links of the tree with four left to go. Continuing,

$$L'_{13} = \min (L'_{13}, L_{12} + d_{23}) = \min (2, 1 + \infty) = 2$$

$$L'_{15} = \min (L'_{15}, L_{12} + d_{25}) = \min (2, 1 + \infty) = 2$$

$$L'_{19} = \min (L'_{19}, L_{12} + d_{29}) = \min (2, 1 + \infty) = 2$$

$$L'_{1,10} = \min (L'_{1,10}, L_{12} + d_{210}) = \min (\infty, 1 + 1) = 2$$

Now, pick at random one of the four as a permanent label, say L_{13}^1 , (2,8).

There are three nodes left, N_5 , N_9 and N_{10} , hence;

$$L_{16}^1 = \min (L_{16}^1, L_{13}^1 + d_{35}) = \min (2, 2 + \infty) = 2$$

$$L_{19}^1 = \min (L_{19}^1, L_{13}^1 + d_{39}) = \min (2, 2 + \infty) = 2$$

$$L_{1,10}^1 = \min (L_{1,10}^1, L_{13}^1 + d_{3,10}) = \min (2, 2 + 1) = 2$$

Once again, randomly choose one, say L_{19}^1 . It has a label (2,8). So far there are seven tree links with two to go. Proceeding as before

$$L_{15}^1 = \min (L_{15}^1, L_{19}^1 + d_{95}) = \min (2, 2 + 4) = 2$$

$$L_{1,10}^1 = \min (L_{1,10}^1, L_{19}^1 + d_{9,10}) = \min (2, 2 + \infty) = 2$$

Let's pick node N_{10} this time. Its permanent label is (2,2). We now have eight tree links so we must proceed on until there are nine.

$$L_{15}^1 = \min (L_{15}^1, L_{1,10}^1 + d_{10,5}) = \min (2, 2 + \infty) = 2$$

Node N_5 becomes a node of the tree with label (2,6). But, there are now nine links ($N - 1$) of the tree, so the algorithm terminates.

Remember, we wanted to find the minimum distance from the starting node N_1 to the final node N_{10} . Node 10 has permanent label (2,2), therefore

$$d_{\min 1}^2 = L_{1,10}^1 = 2$$

Using the algorithm just illustrated with the flat top pulse example, we now have a technique for finding the minimum euclidean weight for an arbitrary L and $f(D)$ pair. This technique along with equations (35) and

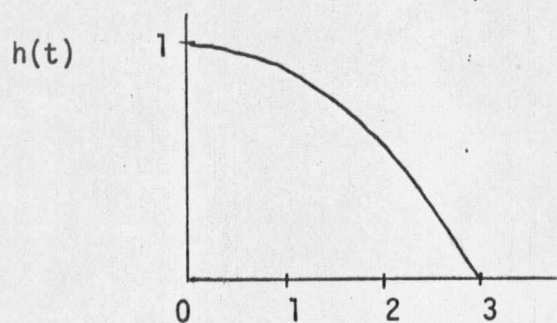
(36) are implemented by an Extended Fortran IV program and used to analyze the performance of the maximum likelihood sequence receiver for several channel impulse response examples.

V. SEVERAL EXAMPLES

To use the program in its entirety, it is necessary only to supply it with values for L and $f(D)$. For example, the flat top pulse for $L = 3$ has a discrete time response $f(D) = 1 + D + D^2$. All the program needs as data is $L = 3$ and the coefficients of $f(D)$, $f_0 = 1$, $f_1 = 1$ and $f_2 = 1$.

Having developed a program that finds the minimum distance through the error state flow graph, let's use it to analyze the performance of our receiver structure for several examples of channel impulse response.

Example Two: Quarter-Wave Cosine Impulse Response



$$h(t) = \begin{cases} \cos \frac{\pi t}{6} & 0 \leq t \leq 3 \\ 0 & \text{elsewhere} \end{cases}$$

$$L = 3$$

$$T = 1$$

1. Pulse Autocorrelation Function

Solving for Coefficients:

$$R_0 = \int_0^3 h^2(t) dt = \int_0^3 \cos^2\left(\frac{\pi t}{6}\right) dt = 1.5$$

$$R_1 = R_{-1} = \int_0^2 \cos\left(\frac{\pi t}{6}\right) \cos \frac{\pi(t+1)}{6} dt = .867$$

$$R_2 = R_{-2} = \int_0^1 \cos\left(\frac{\pi t}{6}\right) \cos\left(\frac{\pi(t+2)}{6}\right) dt = .250$$

$$R(D) = .25D^{-2} + .867D^{-1} + 1.5 + 867D^1 + .25D^{-2}$$

2. Discrete Impulse Response

$$\text{Letting } f(D) = a + bD + cD^2$$

Then:

$$R(D) = acD^{-2} + (ab + bc)D^{-1} + (a^2 + b^2 + c^2) + (ab + bc)D^1 + acD^2$$

Equating Coefficients:

$$R_0 = a^2 + b^2 + c^2 + 1.5$$

$$R_1 = R_{-1} = ab + bc = .867$$

$$R_2 = R_{-2} = ac = .250$$

$$a^8 - a^6 + .1267a^4 - .0625a^2 + .0039 = 0$$

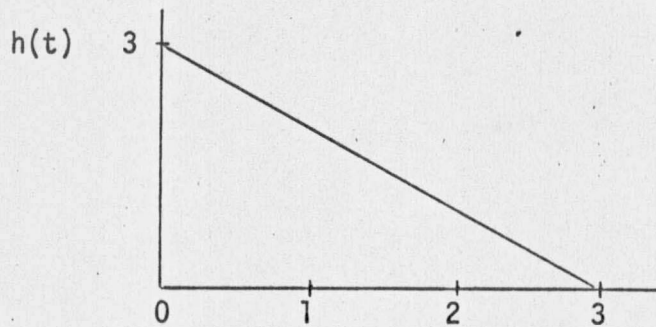
A positive real root is

$$a = .9650$$

Hence

$$f(D) = .9650 + .7083D^1 + .2591D^2$$

Example Three: Straight Line Impulse Response



$$h(t) = \begin{cases} -T/3 + 1 & 0 \leq t \leq 3 \\ 0 & \text{elsewhere} \end{cases}$$

$$L = 3$$

$$T = 1$$

Solving as in Example One:

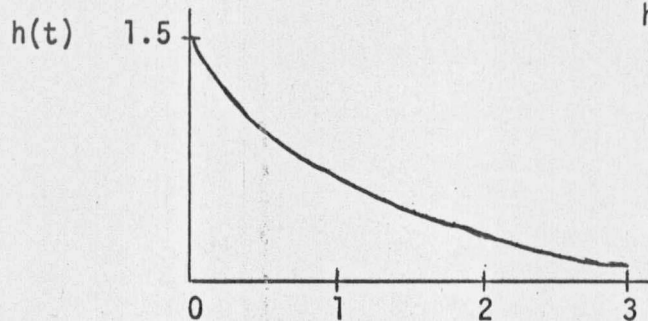
1. Pulse Autocorrelation Function

$$R(D) = 1.333D^{-2} + 4.67D^{-1} + 8.999 + 4.67D + 1.333D^2$$

2. Discrete Impulse Response

$$f(D) = 2.528 + 1.528D + .528D^2$$

Example Four: Decaying Exponential Function



$$h(t) = \begin{cases} 1.5e^{-t} & 0 \leq t \leq 3 \\ 0 & \text{elsewhere} \end{cases}$$

$$L = 3$$

$$T = 1$$

1. Pulse Autocorrelation Function

$$h_0(t) = h(t) \quad 0 \leq t \leq 1$$

$$= 0 \quad \text{elsewhere}$$

$$h(D, t) = (1 + e^{-1}D + e^{-2}D^2)h_0(t)$$

$$R(D) = \int_0^1 (1 + e^{-1}D + e^{-2}D^2)(1 + e^{-1}D^{-1} + e^{-2}D^{-2})h_0^2(t)dt$$

$$= (1 + e^{-1}D^{-1} + e^{-1}D^{-2})(1 + e^{-1}D + e^{-2}D^2)$$

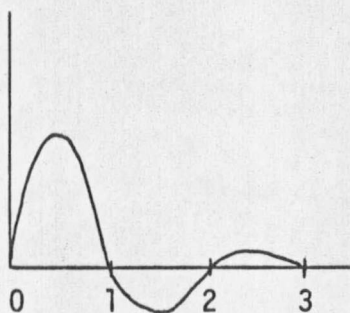
$$= f(-D) \times f(D)$$

2. Discrete Impulse Response

$$f(D) = 1 + e^{-1}D + e^{-2}D^2$$

Example Five: Suppressed Carrier A.M. Modulation

h(t)



$$h(t) = \begin{cases} e^{-t} \sin(\pi t) & 0 \leq t \leq 3 \\ 0 & \text{elsewhere} \end{cases}$$

$$L = 3$$

$$T = 1$$

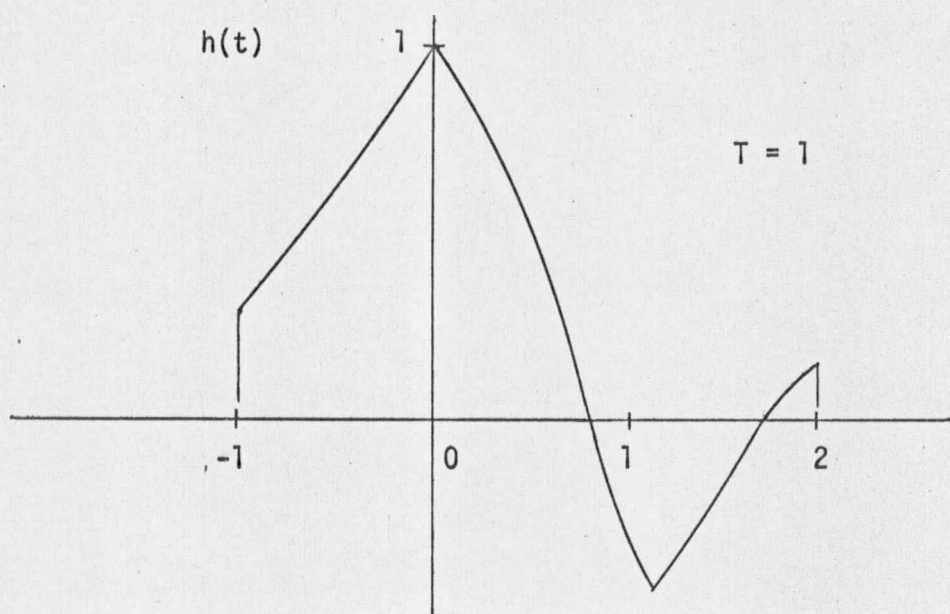
1. Pulse Autocorrelation Function

$$R(D) = .027D^{-2} - .082D^{-1} + .476 - .082D^1 + .027D^2$$

2. Discrete Impulse Response

$$f(D) = .680 - .114D + .040D^2$$

Example Six: Voice-Grade Telephone Channel [13, 14]



1. Pulse Autocorrelation Function

$$R(D) = -.194D^{-2} + .157D^{-1} + 1.518 + .157D - .194D^2$$

2. Discrete Impulse Response

$$f(D) = 1.2125 + .1492D^1 - .1599D^2$$

Examples Seven and Eight: Exponentials for $L > 3$

$$\begin{aligned}
 \text{I.} \quad h(t) &= \begin{cases} \frac{3}{2} e^{-3/4 t} & 0 \leq t \leq 4 \\ 0 & \text{elsewhere} \end{cases} \\
 L &= 4
 \end{aligned}$$

$$1. \quad R(D) = 1.164(1 + e^{-3/4}D + e^{-3/2}D^2 + e^{-9/4}D^3)(1 + e^{-3/4}D^{-1} + e^{-3/2}D^{-2} + e^{-9/4}D^{-3})$$

$$2. \quad f(D) = 1.079 + 509D^1 + .242D^2 + .114D^3$$

$$\begin{aligned}
 \text{II.} \quad h(t) &= \begin{cases} \frac{3}{2} e^{-3 t/5} & 0 \leq t \leq 5 \\ 0 & \text{elsewhere} \end{cases} \\
 L &= 5
 \end{aligned}$$

$$1. \quad R(D) = 1.313(1 + e^{-2/5}D^1 + e^{-6/5}D^2 + e^{-9/5}D^3 + e^{-12/5}D^4)(1 + e^{-2/5}D^{-1} + e^{-6/5}D^{-2} + e^{-9/5}D^{-3} + e^{-12/5}D^{-4})$$

$$2. \quad f(D) = 1.146 + .628D + .345D^2 + .189D^3 + .104D^4$$

Table 1. Program Output for Examples 1 through 5

E	IMPULSE	DISCRETE	MINIMUM EUCLIDEAN WEIGHT				INPUT
X	*	*	*	*	*	*	*
A	RESPONSE	IMPULSE	NO	WITH	FRACTIONAL	ERROR	
M	*	*	INTERSYMBOL	INTERSYMBOL	DIFFERENCE	*	
P	DURATION	RESPONSE	INTERFERENCE	INTERFERENCE	*	SEQUENCE	
L	*	*	*	*	D**2=DI**2	*	
E	L	F(D)	D**2	DI**2	=====	EX(D)	
#	*	*	*	*	D**2	*	

1.	3	*F0= 1.0000*	3.0000	2.0000	.3333	0 1=1 0	
		F1 =1.0000					
		F2 =1.0000					
2.	3	*F0= .9650*	1.5000	1.2660	.1560	0 1=1 0	
		F1 = .7083					
		F2 = .2591					
3.	3	*F0= 2.5275*	8.9998	8.6665	.0370	0 1=1 0	
		F1 =1.5275					
		F2 = .5275					
4.	3	*F0= 1.0000*	1.1537	1.1537	.0000	0=1 0	
		F1 = .3679					
		F2 = .1353					
5.	3	*F0= .6800*	.4770	.4770	.0000	0=1 0	
		F1 =.1140					
		F2 = .0400					

Table 2. Program Output for Examples 6 through 8

E	IMPULSE	DISCRETE	MINIMUM EUCLIDEAN WEIGHT				INPUT
X *	*	*					*
A *	RESPONSE	IMPULSE	NO	WITH	FRACTIONAL	*	ERROR
M *	*	*	INTERSYMBOL	INTERSYMBOL	DIFFERENCE	*	*
P *	DURATION	RESPONSE	INTERFERENCE	INTERFERENCE	*	*	SEQUENCE
L *	*	*	*	*	D**2=DI**2	*	*
E *	L	F(D)	D**2	DI**2	=====	*	EX(D)
# *	*	*	*	*	D**2	*	*

6.	3	*F0= 1.2125*	1.5180	1.5180	.0000		50 * 0=1 0
		F1 = .1492					*
		F2 ==.1599					*
		*					*
7.	4	*F0= 1.0800*	1.4981	1.4981	.0000		* 0 1 0 0
		F1 = .5100					*
		F2 = .2420					*
		F3 = .1140					*
		*					*
8.	5	*F0= 1.1450*	1.8679	1.7021	.0888		* 0 1=1 0
		F1 = .6260					*
		F2 = .3450					*
		F3 = .1870					*
		F4 = .1050					*

Tables 1 and 2 display the program results for the eight channel impulse response examples.

The first six examples are for impulse responses with $L = 3$. Of these six the last three give zero fractional difference in minimum euclidean weight. Notice, all three have exponentially decaying envelopes. Of the six examples we would expect these last three to give the best performance in terms of smallest increase in error rate.

Since those impulse responses having decaying exponential envelopes gave the best results, where $L = 3$, they were evaluated using larger values for L . For $L = 4$ (example seven) the decaying exponential gives zero fractional difference, but for $L = 5$ the minimum euclidean weight is reduced by almost 9 percent due to intersymbol interference.

VI. CONCLUSION AND SUMMARY

A performance criterion and associated analysis techniques have been introduced for evaluating the performance of the maximum likelihood sequence estimator over linear channels with known channel impulse response. Since the minimum euclidean weight or distance of the output signal space gives the receiver structures margin against error when there is additive white gaussian noise on the channel, the fractional difference in the minimum weight gives us a way of comparing the performance of the structure for different channels. When a particular channel gives the smallest fractional difference we know that it gives the smallest increase in error rate. For the linear channels analyzed, this channel would probably produce the most reliable communication system.

In a P.A.M. system, when data is transmitted over narrow bandwidth high signal to noise ratio channel with an information rate (bauds) larger than the channel bandwidth, intersymbol interference may result. The maximum likelihood sequence estimator was designed to help alleviate this interference.

The objective of this thesis is to develop a criterion and associated analysis techniques to evaluate the performance of the maximum likelihood sequence estimator for linear channels with known impulse responses. This receiver structure is optimum in the sense that it minimizes the probability of error. Since bounds on the probability of error can be found by knowing the minimum euclidean weight of the output signal space,

the performance criterion used is the fractional difference in minimum euclidean weight of the output signal space.

Discrete time, D-transform and chip D-transform models of the receiver structure are developed. From these discrete time models, definitions of input, output and state error events are given. These are used to construct a flow graph representing the euclidean weight distribution of the output error events. This flow graph along with a shortest path finding algorithm is implemented by an XDS Extended Fortran IV program and used to find the minimum euclidean weight in the presence of intersymbol interference. The program uses this minimum euclidean weight along with the minimum weight with no intersymbol interference to calculate the fractional difference. Several channel examples are analyzed using the program.

APPENDIX

```

1. C * * * * *
2. C
3. C
4. C THIS PROGRAM CALCULATES THE MINIMUM EUCLIDEAN WEIGHT OF THE OUTPUT
5. C SEQUENCE OF THE MAXIMUM LIKELIHOOD SEQUENCE ESTIMATOR FOR CHANNEL
6. C IMPULSE RESPONSES CAUSING INTERSYMBOL INTERFERENCE. IT ACCOMPLISHES
7. C THIS BY FINDING THE SHORTEST PATH THROUGH A FLOW GRAPH REPRESENTING
8. C THE EUCLIDEAN WEIGHT DISTRIBUTION OF THE OUTPUT ERROR EVENTS. THE
9. C PROGRAM ALSO CALCULATES THE FRACTIONAL DIFFERENCE IN MINIMUM EUCLIDEAN
10. C WEIGHT DUE TO INTERSYMBOL INTERFERENCE.
11. C INPUTS TO THE PROGRAM ARE L THE INPUT
12. C SYMBOL SEPARATION AND THE COEFFICIENTS OF F(D), THE DISCRETE TIME
13. C MODEL IMPULSE RESPONSE. SEVERAL L AND F(D) PAIRS CAN BE INPUTTED AT
14. C ONE TIME. EACH PAIR TAKES TWO CARDS WITH THE FOLLOWING FORMAT:
15. C
16. C
17. C CARD 1 : L , FORMAT(I2)
18. C
19. C CARD 2 : F(J) , FORMAT(LF6.4)
20. C
21. C
22. C * * * * *
23. C
24. C DIMENSION F(0:10),EX(0:41),L1(85),NODE(85)
25. C DIMENSION LINKT(85),LINKP(85),DIST(85,3)
26. C DIMENSION X(10),LASTNODE(85),PTRN(85)
27. C INTEGER EX,X,U,W,PERM,V,TEMP,M,PTRN,U1,W1
28. C REAL L1,L1R
29. C L2=0
30. C CNT=0
31. C

```

```

32.      C          * * *   READ L   * * *
33.      C
34.      500  READ(105,1,END=110) L
35.      1    FORMAT(I2)
36.      IF(L-1) 840,860,860
37.      840  OUTPUT'LOOK BUD,L<1 MAKES NO SENSE IN THIS PROGRAM!!   OK?'
38.      GOTO 110
39.      860  V=L-1
40.      C
41.      C          * * *   READ F   * * *
42.      C
43.      READ(105,2) L,(F(J-1),J=1,L)
44.      2    FORMAT(NF6.4)
45.      C
46.      C          * * *   INITIALIZE VARIABLES   * * *
47.      C
48.      NUM=3**V+1
49.      IF(L.EQ.1) GOTO 850
50.      LE=3** (L-2)
51.      MASK1=0
52.      MASK=3
53.      DO 3 I=1,V-1
54.      MASK=ISC(MASK,2)
55.      MASK1=ICR(MASK,MASK1)
56.      3    CONTINUE
57.      MASK=ISC(MASK,-2*(V-1))
58.      PERM=1
59.      TEMP=N=0
60.      DO 5 I=1,NUM
61.      PTRN(I)=LINKT(I)=LINKP(I)=0
62.      L1(I)=1000000
63.      DO 6 J=1,3
64.      DIST(I,J)=1000000
65.      6    CONTINUE
66.      5    CONTINUE

```

```

67.      L1(1)=LASTNODE(1)=0
68.      IF(L.EQ.L2) GOTO 503
69.      DO 200 A=1,V
70. 200   X(A)=0
71.      DO 4 J=1,NUM
72.      NODE(J)=0
73. 4     CONTINUE
74.  C
75.  C      * * *   CONSTRUCTION OF FLOW GRAPH   * * *
76.  C
77.  C
78.  C      *   BUILDING THE NODES   *
79.  C
80.      J=2
81. 15     I=1
82. 11     IF(X(I)-1)7,8,9
83. 7      X(I)=1
84.      GOTO 10
85. 8      X(I)=2
86.      GOTO 10
87. 9      X(I)=0
88. 10     IF(X(I).NE.0) GOTO 12
89.      I=I+1
90.      IF(I.LE.V) GOTO 11
91. 12     DO 13 J=1,V
92.      X(I)=ISC(X(I),2*(I-1))
93.      NODE(J)=IOR(X(I),NODE(J))
94.      X(I)=ISC(X(I),-2*(I-1))
95. 13     CONTINUE
96.      J=J+1
97.      IF(J.LE.NUM) GOTO 15
98. 503    CONTINUE
99.  C

```

```

100.      C      *      LINKING THE NODES AND SOLVING FOR DIK**2      *
101.      C
102.      I=1
103.      21      K=2
104.      20      NODEI=NODE(I)
105.      NODEI=IAND(NODEI,MASK1)
106.      NODEK=NODE(K)
107.      NODEK=IAND(ISC(NODEK,2),MASK1)
108.      IF(NODEK.NE.NODEI) K=K+1;GOTO20
109.      PTRN(I)=K
110.      17      DO 16 J=0,V-1
111.      MASK=ISC(MASK,2*J)
112.      EX(J)=IAND(MASK,NODE(I))
113.      MASK=ISC(MASK,-2*J)
114.      EX(J)=ISC(EX(J),-2*J)
115.      IF(EX(J).EQ.2) EX(J)=-1
116.      16      CONTINUE
117.      MASK=ISC(MASK,2*(V-1))
118.      EX(V)=IAND(MASK,NODE(K))
119.      MASK=ISC(MASK,-2*(V-1))
120.      EX(V)=ISC(EX(V),-2*(V-1))
121.      IF(EX(V).EQ.2) EX(V)=-1
122.      J=V
123.      EYJ=0
124.      DO 18 II=0,J
125.      EYJ=EYJ+EX(II)*F(J-II)
126.      18      CONTINUE
127.      P=(K+LE-PTRN(I))/LE
128.      DIST(I,P)=EYJ*EYJ
129.      IF(I.EQ.1) NNN=2;GOTO 19
130.      NNN=3
131.      19      IF(P.LT.NNN) K=K+LE;GOTO 17
132.      I=I+1
133.      IF(I.LE.NUM-1) GOTO 21

```

```

134. C
135. C      * * *   MINIMUM PATH FINDER ALGORITHM   * * *
136. C
137. C
138. C      *   INITIAL STEP OF ALGORITHM   *
139. C
140.      M=PTRN(1)
141.      TEMP=M
142.      P=1
143. 22  L1(M)=DIST(1,P)
144.      LASTNODE(M)=1
145.      M=M+LE
146.      P=(M+LE-PTRN(1))/LE
147.      IF(P*LE<2) LINKT(M-LE)=M;GOTO22
148.      LINKT(M-LE)=0
149.      FLAG=0
150.      GOTO 65
151. C
152. C      *   ALL NEIGHBORING NODES ARE LABELED (L,I)   *
153. C
154. 80  M=PTRN(PERM)
155. 40  P=(M+LE-PTRN(PERM))/LE
156.      IF(P*LE<3) GOTO 45
157.      W=TEMP
158. 30  IF(W*EQ*0) GOTO35
159.      IF(W*EQ*M) GOTO37
160.      W=LINKT(W)
161.      GOTO 30
162. 35  Q=PERM
163. 36  IF(Q*EQ*0) LINKT(M)=TEMP;TEMP=M;LASTNODE(M)=PERM; GOTO 37
164.      IF(M*EQ*Q) GOTO 37
165.      Q=LINKP(Q)
166.      GOTO 36
167. 37  M=M+LE
168.      GOTO 40

```

```

169. C
170. C      * LIK' = MIN(LIK',LSP + DPK) *
171. C
172. 45 W=TEMP
173. 50 IF(W.EQ.0) GOTO 65
174. P=W+LE-PTRN(PERM)
175. DT=LE
176. P=P/DT
177. IF(P.EQ.1.) GOTO 51
178. IF(P.EQ.2.) GOTO 51
179. IF(P.EQ.3.) GOTO 51
180. DT= 1000000.
181. GOTO 52
182. 51 DT=DIST(PERM,P)
183. 52 L1R=L1(PERM)+DT
184. IF(L1(W)=L1R)55,55,60
185. 60 L1(W)=L1R
186. LASTNODE(W)=PERM
187. 55 W=LINKT(W)
188. GOTO 50
189. C
190. C      * LSK = LSK' = MIN( LSK' ) *
191. C
192. 65 W=TEMP
193. IF(W.EQ.0) FLAG=1; GOTO 85
194. U1=W
195. W1=W
196. 70 U=LINKT(U1)
197. IF(U.EQ.0) GOTO 75
198. IF(L1(U)=L1(W)) 71,72,72
199. 71 W=U
200. W1=U1

```

```

201.      72      U1=LINKT(U1)
202.          GOTO 70
203.      75      IF(W1.EQ.W) TEMP=LINKT(W)
204.          LINKT(W1)=LINKT(W)
205.          LINKT(W)=0
206.          LINKP(W)=PERM
207.          PERM=W
208.          N=N+1
209.          IF(N.LT.NUM-1) GOTO 80
210.      85      CONTINUE
211.      C
212.      C          * * *   CALCULATION OF EX(D)   * * *
213.      C
214.          W=LASTNODE(NUM)
215.          JJ=0
216.          MASK=ISC(MASK,2*(V-1))
217.      620      NODEI=NODE(W)
218.          NODEI=ISC(IAND(NODEI,MASK),-2*(V-1))
219.          IF(NODEI-1)610,610,600
220.      600      NODEI=-1
221.      610      EX(JJ)=NODEI
222.          W=LASTNODE(W)
223.          IF(W.NE.0) JJ=JJ+1;GOTO 620
224.      C
225.      C          * * *   CALCULATION OF FRACTIONAL DIFFERENCE   * * *
226.      C
227.      850      DMIN=0
228.          DO 800 IH=0,V
229.      800      DMIN=DMIN+F(IH)*F(IH)
230.          IF(L.EQ.1) L1(NUM)=DMIN
231.          DIFF=(DMIN-L1(NUM))/DMIN
232.          CALL WRITESUB(F,L,L1,NUM,DMIN,DIFF,JJ,CNT,EX)

```

```
233.      L2=L
234.      GOTO 500
235.      110  WRITE(108,1000)
236.      1000 FORMAT('1')
237.      STOP
238.      END
```

```

1.      SUBROUTINE WRITESUB(F,L1,NUM,DMIN,DIFF,UJ,CNT,EX)
2.      DIMENSION L1(405),F(0:10),EX(0:50)
3.      INTEGER EX
4.      REAL L1
5.      DATA LINE/'*'/
6.      IF(CNT.EQ.1) GOTO 400
7.      WRITE(108,323)
8.      323  FORMAT('1')
9.      OUTPUT' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
10.     WRITE(108,300)
11.     300  FORMAT(14X,'E',3X,'IMPULSE',4X,'DISCRETE',12X,'MINIMUM EUCLIDEAN W
12.     CEIGHT',13X,'INPUT')
13.     301  WRITE(108,302)
14.     302  FORMAT(14X,'X',1X,'*',10X,'*',10X,'*',44X,'*')
15.     WRITE(108,303)
16.     303  FORMAT(14X,'A',1X,'*',1X,'RESPONSE',1X,'*',1X,'IMPULSE',2X,'*',7X
17.     C,'NO',13X,'WITH',7X,'FRACTIONAL',1X,'*',2X,'ERROR')
18.     WRITE(108,304)
19.     304  FORMAT(14X,'M',1X,'*',10X,'*',10X,'*',2X,'INTERSYMBOL',2X,'*',2X
20.     C,'INTERSYMBOL',2X,'*',1X,'DIFFERENCE',1X,'*')
21.     WRITE(108,305)
22.     305  FORMAT(14X,'P',1X,'*',1X,'DURATION',1X,'*',1X,'RESPONSE',1X,'*',2X
23.     C,'INTERFERENCE',1X,'*',2X,'INTERFERENCE',1X,'*',12X,'*',1X,'SEQUEN
24.     CCE')
25.     WRITE(108,306)
26.     306  FORMAT(14X,'L',1X,'*',10X,'*',10X,'*',15X,'*',15X,'*',1X,'D**2-DI*
27.     C*2',1X,'*')
28.     WRITE(108,307)
29.     307  FORMAT(14X,'E',1X,'*',4X,'L',5X,'*',3X,'F(D)',3X,'*',6X
30.     C,'D**2',5X,'*',5X,'DI**2',5X,'*',1X,'-----',1X,'*',2X,'EX(D)'
31.     C)
32.     401  WRITE(108,404)
33.     404  FORMAT(14X,'#',1X,'*',10X,'*',10X,'*',15X,'*',15X,'*',4X,'D**2',4X
34.     C,'*')

```

```

35.      LP=80
36.      WRITE(108,309) LP,(LINE,J=1,LP)
37.      309  FORMAT(13X,NA1)
38.      CNT=1
39.      N1=6
40.      400  WRITE(108,402)
41.      402  FORMAT(16X,'*',10X,'*',10X,'*',15X,'*',15X,'*',12X,'*')
42.      WRITE(108,403)N1,L,F(0),DMIN,L1(NUM),DIFF,JJ+1,(EX(JJ-J),J=0,JJ)
43.      403  FORMAT(12X,I2,'.',1X,'*',3X,I2,5X,'*FO=',1X,F6.4,'*',4X,F6.4,5X
44.      C,'*',4X,F6.4,5X,'*',2X,F6.4,4X,'*',NI2)
45.      DO 405 J=1,L-1
46.      405  WRITE(108,406)J,F(J)
47.      406  FORMAT(16X,'*',10X,'*F',I,'=',F6.4,'*',15X,'*',15X,'*',12X,'*')
48.      N1=N1+1
49.      RETURN
50.      END

```

REFERENCES

- [1] Mischa Schwartz, Information Transmission, Modulation and Noise. New York: McGraw-Hill, 1970, Ch. 2-3.
- [2] G. David Forney, "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presences of Intersymbol Interference," IEEE Trans. Inform. Theory, Vol IT-18, pp. 363-378, May 1972.
- [3] R. W. Lucky, J. Salz, and E. J. Weldon, Jr., Principles of Data Communication. New York: McGraw-Hill, 1968, Ch 5.
- [4] Donald W. Tufts, "Nyquist's Problem--The Joint Optimization of Transmitter and Receiver in Pulse Amplitude Modulation," Proc. IEEE, Vol. 53, pp. 248-259, Mar. 1965.
- [5] C. W. Helstrom, Statistical Theory of Signal Detection. New York: Pergamon, 1960, Sect. IV.5.
- [6] R. W. Chang and R. C. Hancock, "On Receiver Structures for Channels Having Memory," IEEE Trans. Inform. Theory, Vol. IT-12, pp. 463-468, Oct. 1966.
- [7] Andrew J. Viterbi, "Convolutional Codes and their Performance in Communication Systems," IEEE Trans. Commun. Technol., Vol. COM-19, Oct. 1971, pp. 741-771.
- [8] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering. New York: Wiley, 1965.
- [9] D. J. Sakrison, Communication Theory: Transmission of Waveforms and Digital Information. New York: John Wiley and Sons, Inc., 1968, Ch. 7.
- [10] Jim K. Omura, "Optimal Receiver Design for Convolutional Codes and Channels with Memory Via Control Theoretical Concepts," Information Sciences, Vol. 3, 1971, pp. 243-266.
- [11] T. C. Hu, Integer Programming and Network Flows. Reading, Mass.: Addison-Wesley Publishing Company, 1969, Ch. 10.
- [12] E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs," Num. Math., Vol. 1, 1959, pp. 269-271.
- [13] J. R. Davey, "Modems," Proc. of the IEEE, Vol. 60, No. 11, Nov. 1972, pp. 1289-1290.

- [14] F. P. Duffy and T. W. Thatcher, Jr., "Analog Transmission Performance on the Switched Telecommunications Network," Bell System Technical Journal, Vol. 50, No. 4, April, 1971, pp. 1311-1397.

MONTANA STATE UNIVERSITY LIBRARIES



3 1762 10013707 2

N378	
F533	Fisher, Harold F
cop.2	Performance analysis of the maximum likeli- hood sequence estimator
NAME AND ADDRESS	
Baldwin Kinsman 1205 Congdon St.	
Renew	when due
DEC 2 1972	gm
JAN 11 1973	DS
Baldwin Kinsman 1205 Congdon St.	
N378	
F533	
Cop. 2	



College
Place
Bindery

College Place, Wa.