DEMODULATION FOR INTERSYMBOL INTERFERENCE CHANNELS IN THE PRESENCE OF COLORED GAUSSIAN NOISE

by

R. M. A. P. Rajatheva

A thesis

Presented to the University of Manitoba

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in

the Department of Electrical Engineering

Winnipeg, Manitoba

May 1991

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-76935-1

Canada

DEMODULATION FOR INTERSYMBOL INTERFERENCE CHANNELS IN THE PRESENCE OF COLORED GAUSSIAN NOISE

BY

R.M.A.P. RAJATHEVA

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

© 1991

Permission has been granted to the LIBRARY OF THE UNIVERSITY OF MANITOBA to lend or sell copies of this thesis. to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film, and UNIVERSITY MICROFILMS to publish an abstract of this thesis.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

ABSTRACT

Maximum likelihood sequence estimation (MLSE) in additive white Gaussian noise with finite intersymbol interference has been thoroughly investigated by several authors. Generally the Viterbi algorithm is applied for the estimation of data. Extension to the case of infinite intersymbol interference has been developed recently using the Sequential algorithm.

Application of these algorithms for an intersymbol interference channel with additive colored Gaussian noise is presented in this thesis. A maximum likelihood metric for the Viterbi algorithm is derived using a finite time whitening approach and is referred to as a finite time metric. The receiver structure in this case consists of L matched filters where the channel impulse response is of finite duration LT seconds. A receiver structure for the Sequential algorithm is also obtained considering an infinite time whitening interval. Only one matched filter is required at the receiver.

Simulations have been carried out in an attempt to acquire some knowledge as to how these metrics perform under different noise conditions and with different channels. The results show that the Viterbi algorithm with a finite time metric and the Sequential algorithm give a better error performance in comparison to the Viterbi algorithm applied with an infinite time metric particularly, when the noise is more colored. The computational complexity of the Sequential algorithm is comparably less than that required by the Viterbi algorithm at moderate signal to noise ratios.

ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to Professor E. Shwedyk for his supervision and continual guidance throughout the M. Sc. program and during the work on this thesis. Grateful acknowledgement is also due the Association of Universities and Colleges of Canada for the Commonwealth Scholarship.

I hereby declare that I am the sole author of this thesis. I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

R. M. A. P. Rajatheva

I further authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in whole or in part, at the request of other institutions or individuals for the purpose of scholarly research.

R. M. A. P. Rajatheva

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is required in many communication systems to transmit digital data at high speed over channels with limited bandwidth. Bandlimited channels usually produce intersymbol interference (ISI) where the transmitted pulse overlaps with other transmission pulses. The number of pulses that overlap with a given transmitted pulse is known as the memory of the channel or the length of ISI. The simplest communication system to exhibit ISI is a pulse amplitude modulation (PAM) communication system. The input to the channel is a real number sequence drawn from a finite alphabet which, if it passes through a linear channel whose impulse response is longer than one transmission time interval, shall result in ISI. The presence of ISI degrades the receiver performance.

Various techniques have been developed for combating ISI. These techniques date back to Nyquist [12] who introduced baseband spectrum shaping for completely eliminating ISI. Lender [9] introduced the duobinary technique which allowed one ISI term. This was later generalized to partial response techniques by Lender [10] and Kretzmer [7] where any number of ISI terms are allowed. Tomlinson [18] suggested a precoding technique to eliminate ISI. The input sequence is coded according to the inverse of the discrete channel response. Other methods for eliminating or controlling of ISI are linear equalization [11] and decision feed back equalization (DFE) [1]. Both reduce ISI by subtracting out the actual ISI. The decision at the receiver is taken on symbol by symbol basis for all the methods described above. Chang and Hancock [2] presented a method which bases the decision on L consecutive symbols.

As an alternative to forcing ISI to zero or introducing controlled ISI the problem can be treated using decision theoretic estimation methods [6,19]. Forney [5] showed that the Viterbi algorithm (VA), which was originally developed by Viterbi [22] for the decoding of convolutional codes, can be applied to ISI channels for maximum likelihood sequence estimation (MLSE). The detection problem is modelled as a graph search which in this case can be implemented through a trellis. The receiver consists of a whitened matched filter and a symbol rate sampler. The advantages of the VA are that there are a fixed number of computations per decoded symbol and that its structure is regular. However, the computational complexity of the VA grows exponentially with the channel memory, therefore making it difficult to apply when the memory is large.

To decrease memory requirement other algorithms known as reduced state algorithms have been developed. The common characteristic of these algorithms is that they reduce the computational complexity by reducing the number of sequences to which they are applied. They can be classified as variations of the VA.

The VA with decision feed back to search a reduced state sub trellis has been introduced by Eyuboglu and Qureshi [4]. Duel and Heegard [3] applied this technique to binary transmission and called it decision feed back sequence estimation. With some changes, Polyduros and Kazakos [13] showed that the VA can be applied to infinite ISI channels with rational spectrum. It is known as a modified Viterbi algorithm (MVA). Sheshadri and Anderson [15] used the M-algorithm which keeps only best M paths for the trellis search. A similar algorithm to the M algorithm is the T algorithm proposed by Simmons [17] which searches paths with metrics less than an adaptive threshold. All of the above algorithms obtain reduced complexity while retaining the VA structure with a justifiable loss of optimality in estimation. Recently, the Sequential algorithm (SA), another well known algorithm for the decoding of convolutional codes, has been extended to the ISI case by Xiong [24]. The average number of computations for the SA is variable and is independent of the channel memory. Therefore the SA can be easily applied for cases where channel memory is very large or even infinite. A review of the literature regarding ISI can be found in [25].

Most of the research has been concerned with the estimation of the data sequence in the presence of additive white Gaussian noise (AWGN). Comparably little work has been done when data is corrupted with colored noise [8,14,16,20]. The straightforward procedure in this case is to use a whitening filter to whiten the colored noise. Because theoretically the resulting ISI can be infinite this approach is suitable only if the SA is used. The VA may be applied with the ISI terms truncated as discussed by Ungerboeck [20]. To be applied without truncation a whitening filter which does not cause infinite ISI has to be found. Then the maximum likelihood metric for the VA can be derived.

In this thesis MLSE in colored Gaussian noise is studied. Chapter 2 provides the general background about ISI and MLSE and a description of the VA and SA as they are used in the thesis. Chapter 3 describes the theoretical procedures to develop maximum likelihood metrics for the VA and SA. Computer simulations and results are presented in chapter 4. The VA using the maximum likelihood metric and a sub optimal metric are compared with the SA. Two kinds of noise models, one with a white noise component and the other without, are considered for simulations. Chapter 5 presents conclusions.

Chapter 2

Background

2.1 Introduction

This chapter provides a description of the application of the Viterbi algorithm to ISI in additive white Gaussian noise as developed by Forney [5]. Then the procedure to apply the sequential algorithm is explained for the infinite ISI case. Finally the methods available when colored noise is present without ISI are discussed.

2.2 ISI with additive white Gaussian noise (AWGN)

Consider the PAM communication system shown in Figure 2.1 where the x_k's are the input alphabet which initially pass through an impulse modulator before transmission. The channel impulse response is h(t). The length of ISI present in s(t) is determined by the length of h(t).

The received signal is

$$r(t)=s(t)+n(t) \tag{2.1}$$

$$=\sum_{k} x_k \; h(t-kT)+n(t) \tag{2.2}$$

If sampled at lT $$r(lT)=\sum_{k} x_k \; h(lT-kT) +n(lT) \tag{2.3}$$

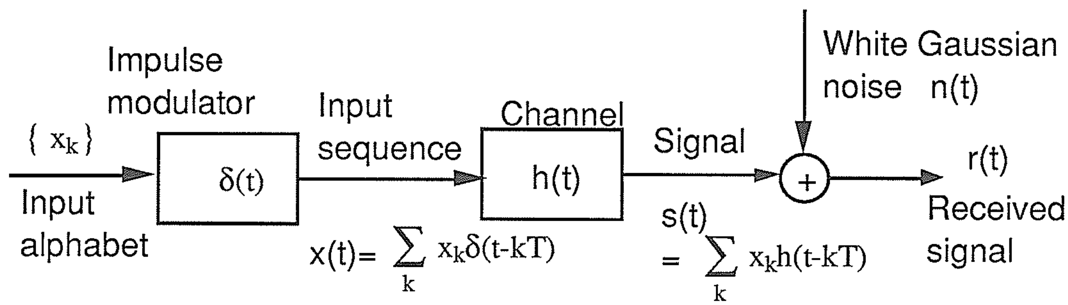


Figure 2.1

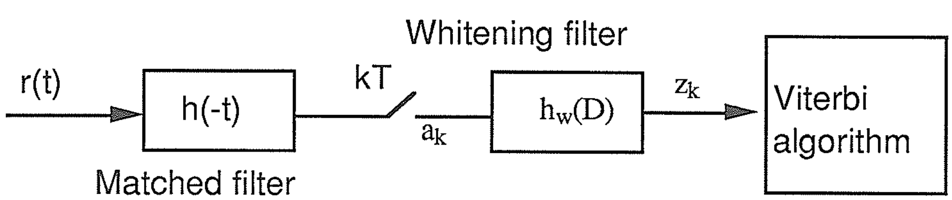


Figure 2.2

$$=h_o x_l + \sum_{k \neq l} x_k h_{l-k} + n_l \tag{2.4}$$

The term $\sum_{k \neq l} x_k h_{l-k}$ represents the intersymbol interference.

For maximum likelihood sequence estimation(MLSE) in additive white Gaussian noise(AWGN) the receiver structure, shown in Figure 2.2, can be used [5]. The sampled output of the matched filter forms a set of sufficient statistics.

$$a_k = \int_{-\infty}^{\infty} r(t)h(t-kT)dt \tag{2.5}$$

$$= \int_{-\infty}^{\infty} \sum_{k'} x_{k'} h(t-k'T)\, h(t-kt)dt + \int_{-\infty}^{\infty} n(t)h(t-kT)dt \tag{2.6}$$

$$= \sum_{k'} x_{k'} R_{k'-k} + n'_k \tag{2.7}$$

where $R_{k'-k}$ represents the sampled autocorrelation function of the channel impulse response and $n_{k'}$ represents the noise sample at the output of the matched filter.

The above difference equation may be expressed using the delay operator -D(similar to z^{-1} in Z-Transform theory) as

$$a(D) = \sum_k a_k D^k = x(D)R(D) + n'(D) \tag{2.8}$$

-R(D) is the D-transform of the discrete autocorrelation function of h(t).

The statistics of the noise samples at the output of the matched filter are given by

$$E\{n'_k n'_m\} = E\left\{ \int_{-\infty}^{\infty} n(t)h(t-kT)dt \int_{-\infty}^{\infty} n(\tau)h(\tau-mT)d\tau \right\} \tag{2.9}$$

$$= \int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} h(t-kT)h(\tau-mT)E\{n(t)n(\tau)\}d\tau \tag{2.10}$$

Since n(t) is white Gaussian noise

$$E\{n(t)n(\tau)\}=\sigma^2\delta(t-\tau) \tag{2.11}$$

where σ^2 is the spectral strength of the noise in watts/Hz.

$$E\{n'_k\, n'_m\}=\sigma^2\int_{-\infty}^{\infty} h(t-kT)\, h(t-mT)dt \tag{2.12}$$

$$= \sigma^2\, R_{k-m} \tag{2.13}$$

Therefore $n'(D)$ is colored Gaussian noise with autocorrelation function $\sigma^2 R(D)$.

It can be shown that R(D) has the spectral factorization $R(D)=f(D)f(D^{-1})$ whether h(t) has a finite length or infinite length [26]. Since

$$R_{n'}(D)=\sigma^2\, R(D) \tag{2.14}$$

the colored noise n'(D) can be expressed as

$$n'(D)=n(D)f(D^{-1}) \tag{2.15}$$

Thus
$$a(D)=x(D)f(D)f(D^{-1}) + n(D)f(D^{-1}) \tag{2.16}$$

$$z(D)=\frac{a(D)}{f(D^{-1})} \tag{2.17}$$

$$= x(D)f(D) + n(D) \tag{2.18}$$

$$= y(D) + n(D) \tag{2.19}$$

where, provided it is stable, $1/f(D^{-1})$ represents the whitening filter .

2.3 Finite ISI

The discrete model described above can be realized as a finite state machine when the ISI is finite. Since h(t) is of finite length LT, R(D) can be expressed as

$$R(D)= \sum_{k=-v}^{v} R_k\, D^k=f(D)f(D^{-1});\ v=L-1 \tag{2.20}$$

v=length of inteference

$$f(D) = \sum_{i=0}^{v} f_i D^i \quad (2.21)$$

In the time domain $$y_k = \sum_{i=0}^{v} f_i x_{k-i} \quad (2.22)$$

Therefore the model shown in Figure 2.3 can be used to obtain y_k . The channel symbols y_k are a function of the current input x_k and v past inputs. ie, $y_k = g(x_k, s_{k-1})$ where s_k is a state uniquely determined by the previous v inputs prior to time kT. The one to one mapping between input sequence **x** and channel symbol sequence **y** is described by a trellis of width $s=m^v$ states,where m is the input alphabet size. There are m branches out of each state, one per possible input symbol, and each branch has a corresponding channel symbol y_k. A four state trellis is shown in Figure 2.4 as an example.

The maximum likelihood metric for the sequences **x** and **y** can be denoted by

$$\Gamma(\mathbf{x},\mathbf{y}) = \log_e p(y|x) = \log_e \prod_k p(y_k|x_k) \quad (2.23)$$

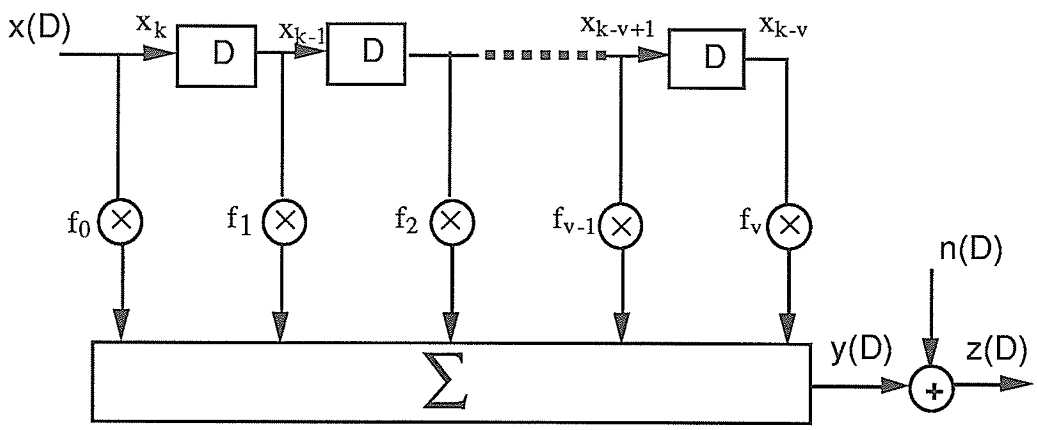
$$= \sum_k \log_e p(y_k|x_k) \quad (2.24)$$

$$= \sum_k \lambda(y_k, x_k) \quad (2.25)$$

where $\lambda(y_k, x_k)$ is the branch metric.

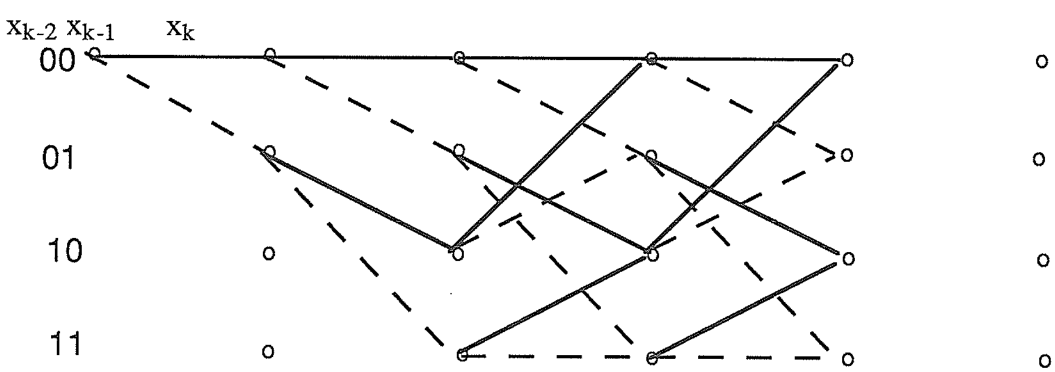
where the probability density function p(y/x) factors into a product of terms since the noise samples are statistically independent.

The number of states m^v grow exponentially with the length of ISI. Hence the Viterbi algorithm cannot be applied to infinite ISI. The whitening filter used in the approach described earlier can be avoided using the procedure described by Viterbi and Omura [23].



Finite State Model

Figure 2.3



4- state trellis

Figure 2.4

Here one starts from the maximum likelihood metric for a signal in AWGN

$$\lambda = \int_{-\infty}^{\infty} r(t)s(t)dt - 1/2 \int_{-\infty}^{\infty} s^2(t)dt \tag{2.26}$$

which can be simplified to [23, pp. 272-284]

$$\lambda = \sum_k \Big[(a_k - \sum_{i=1}^{v} x_{k-i} R_i) x_k \Big] \tag{2.27}$$

with the notation used earlier. Therefore in this case it is sufficient to obtain the output of the matched filter. A similar trellis search can be done in this situation as well. What changes is the branch metric of the trellis.

2.4 Sequential algorithm and infinite ISI

The difference between finite ISI and infinite ISI is apparent from the form of f(D). For infinite ISI f(D) is of the form

$$f(D) = \frac{U(D)}{V(D)} = \frac{\sum_{i=0}^{n} u_i D^i}{1 + \sum_{i=1}^{m} v_i D^i} \tag{2.28}$$

$$y(D) = x(D)f(D) \tag{2.29}$$

In the time domain

$$y_k = u_0 x_k + \sum_{i=1}^{n} u_i x_{k-i} - \sum_{i=1}^{m} v_i y_{k-i} \tag{2.30}$$

Based on the difference equation for y_k the system can be modelled as a feedback filter and a tree can be developed as illustrated by Figures 2.5 and 2.6 respectively for f(D). As shown by Xiong [24] the sequential algorithm can be applied in this case. The algorithm searches a tree to determine the maximum likelihood path among the explored paths of different lengths.

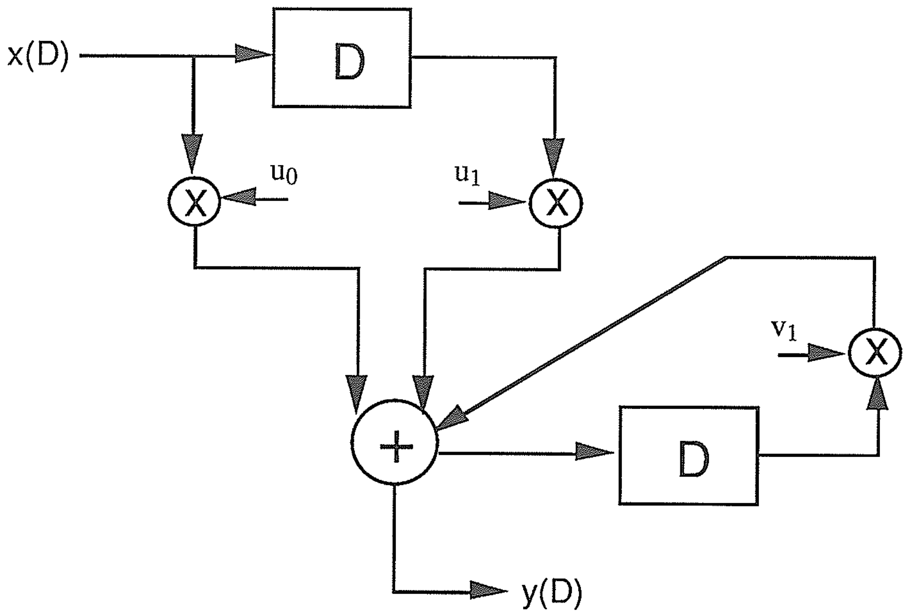


Figure 2.5: Feed Back Filter ; $f(D) = \frac{u_0 + u_1 D}{1 + v_1 D}$

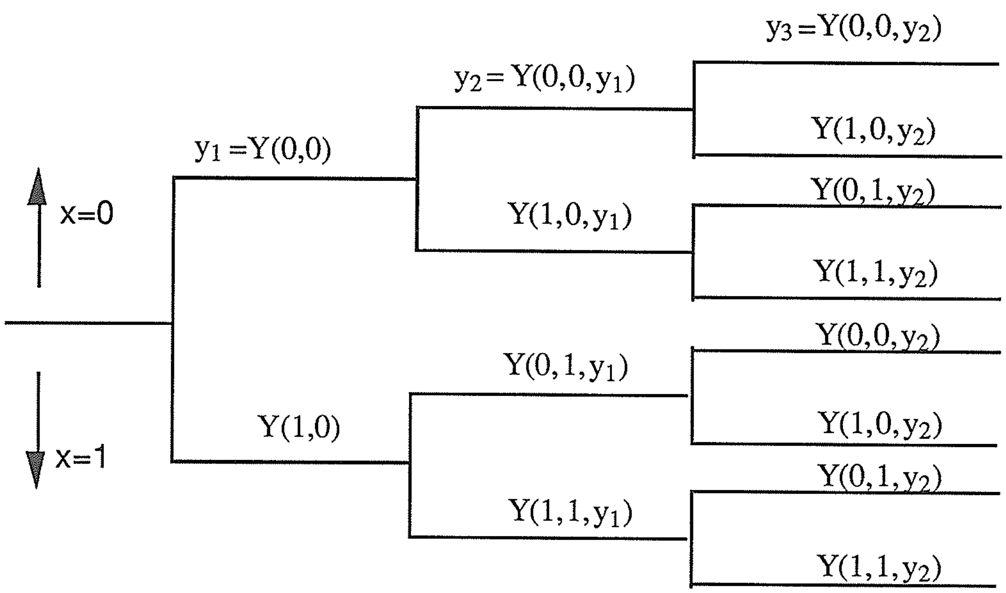


Figure 2.6: Tree $\quad y_k = Y(x_k, x_{k-1}, y_{k-1})$

The metric for the sequential algorithm for the ISI channel with an equally likely m-ary input sequence, is given below.

$$L(X_1, z) = \sum_{k=1}^{n_1} [\log \frac{p_n(z_k - y_k)}{p_z(z_k)} - \log m] \tag{2.31}$$

$$= \sum_k L(y_k, z_k) \tag{2.32}$$

where

$$L(y_k, z_k) = \log \frac{p_n(z_k - y_k)}{p_z(z_k)} - \log m \tag{2.33}$$

is the branch metric.

Here X_1 is the input sequence, n_1 is the number of input symbols, z_k is the received sequence and $p_n(.)$ is the noise probability density function, i.e.

$$p_n(x) = \frac{1}{\sqrt{2\Pi}\sigma} \exp\left\{\frac{-x^2}{2\sigma^2}\right\} \tag{2.34}$$

In the finite case

$$p_z(z_k) = \frac{1}{m^L} \sum_{j=1}^{m^L} \frac{1}{\sqrt{2\Pi}\sigma} \exp\left\{\frac{-(z_k - b_j)^2}{2\sigma^2}\right\} \tag{2.35}$$

$$b_j \in y_k \qquad y_k = f(x_k, \ldots\ldots, x_{k-v}) \tag{2.36}$$

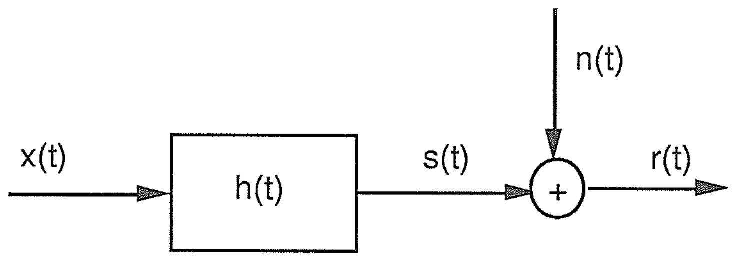
For the infinite case, although the channel has infinite memory, y_k is a function of the input sequence up to time k. ie,

$$y_k = f(x_1, \ldots\ldots, x_k) \tag{2.37}$$

as is easily seen from equation (2.29). Thus

$$p_z(z_k) = \frac{1}{m^k} \sum_{j=1}^{m^k} \frac{1}{\sqrt{2\Pi}\sigma} \exp\left\{\frac{-(z_k - b_j)^2}{2\sigma^2}\right\} \tag{2.38}$$

Unlike the finite ISI case, here $p_z(z_k)$ is dependent on the time index k. However, since the impulse response of physical channels decays with time one can safely truncate it at a length L^*T. Beyond this length the terms should not have a significant effect on ISI and also on $p_z(z_k)$.



$$E\{n(t),n(\tau)\} = K_n(t-\tau)\ ;\ S_n(\omega) = \int_{-\infty}^{\infty} K_n(t-\tau)e^{-j\omega(t-\tau)}\, d(t-\tau)$$

Figure 2.7

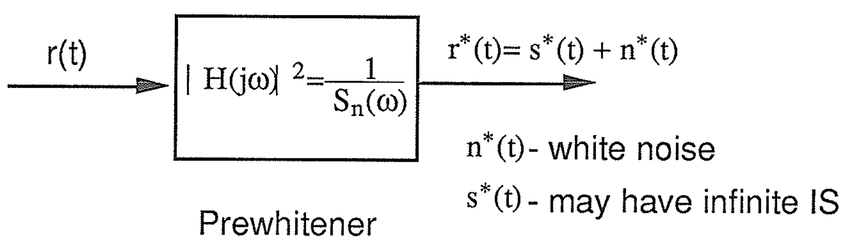


Figure 2.8

2.5 Colored noise

The communication system model, shown in Figure 2.7 is same as previously discussed except that the noise is colored. As mentioned if this problem can be transformed into a white Gaussian problem, then the methods discussed earlier can be applied. A straightforward approach is to use a simple prewhitener to whiten the noise. (See Figure 2.8). Since, in general, the autocorrelation function extends to infinity the impulse response of the prewhitener can also last an infinite time leading to infinite ISI. One may use the sequential algorithm or the Viterbi algorithm if the ISI is truncated. Ungerboeck [20] used the Viterbi algorithm where he obtained finite ISI by truncation.

An alternative approach would be to develop a prewhitener whose impulse response is finite. Without ISI this is a well studied problem and is described in many texts. Van Trees [21] suggested three different interpretations of the same method for obtaining a finite impulse response prewhitener. These are shown in the block diagrams of Figures 2.9a, 2.9b and 2.9c. Here $h_w(t,u)$ is the whitening filter over the finite time interval $[T_i , T_f]$, s(t) is the signal transmitted in $[T_i , T_f]$ and $K_n(t,u)$ is the covariance of the colored noise. $Q_n(v,x)$ and g(z) are the solutions of the following integral equations.

$$\delta(z-v)=\int_{T_i}^{T_f} K_n(x,z)Q_n(v,x)dx \quad , T_i \leq z,v \leq T_f \tag{2.39}$$

$$g(z)=\int_{T_i}^{T_f} Q_n(z,v)s(v)dv \quad , T_i \leq v \leq T_f \tag{2.40}$$

$$s(t)=\int_{T_i}^{T_f} K_n(t,u)g(u)du \quad , T_i \leq u \leq T_f \tag{2.41}$$

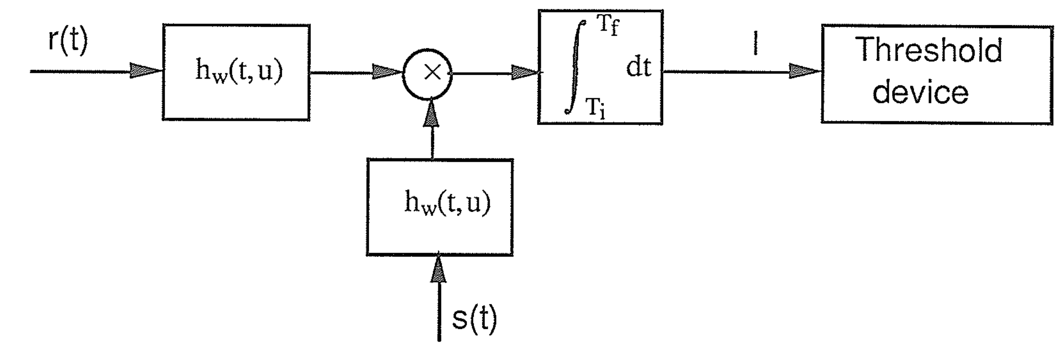


Figure 2.9a

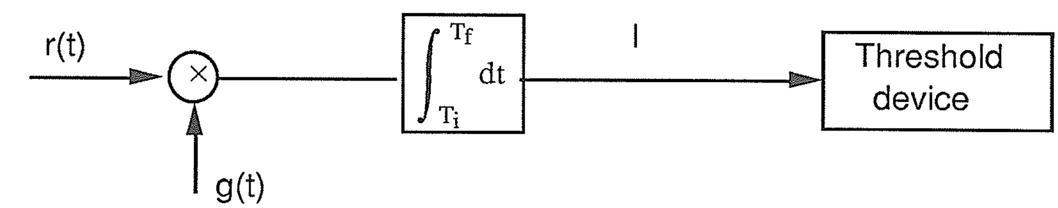


Figure 2.9b

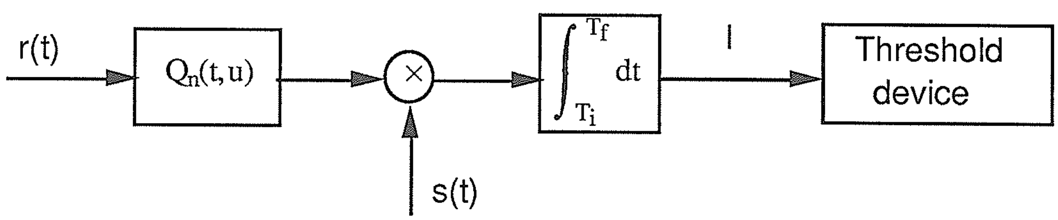


Figure 2.9c

Chapter 3

Theory

3.1 Introduction

In chapter 2, various options available to deal with the colored noise problem were discussed. The procedures are different depending on whether the Viterbi algorithm or the sequential algorithm are to be used. In this chapter a method is presented to obtain the maximum likelihood metric for the application of the Viterbi algorithm. Also the receiver structure to obtain sufficient statistics for the sequential algorithm is derived. The metric in the case of the sequential algorithm is the same as that developed by Xiong [24].

3.2 Development of the maximum likelihood metric for ISI in colored Gaussian noise

The approach here is to whiten the colored noise. To apply the Viterbi algorithm the intersymbol interference should be finite.Therefore a whitening filter with a finite impulse response has to be used. A PAM communication system with additive colored Gaussian noise introduced previously is used as shown in Figure 3.1.

The receiver is shown in Figure 3.2, where the receiver structure of Figure 2.9b,

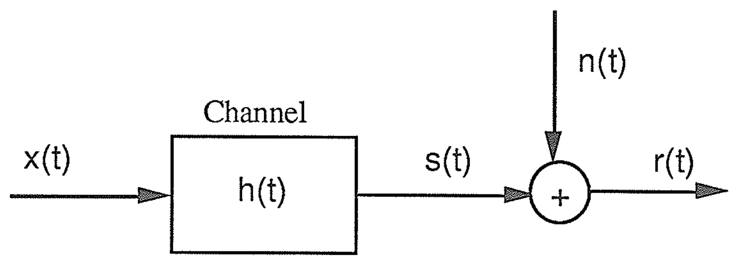


Figure 3.1

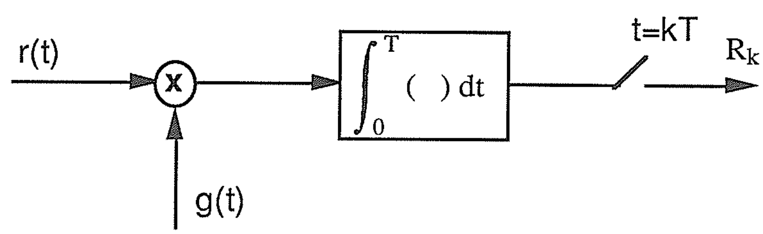


Figure 3.2

is used to whiten the noise. The whitening is achieved through g(t), and is equivalent to passing both r(t) and s(t) through a whitening filter. The output samples provide the sufficient statistics for the maximum likelihood sequence estimation (MLSE).

Given the finite time whitening of the colored noise of Figure 3.2 the metric to be evaluated is

$$\lambda_T = \int_0^T r(t)g(t)dt - 1/2 \int_0^T s(t)g(t)dt \quad , 0 \le t \le T \tag{3.1}$$

where g(t) satisfies the integral equation

$$s(t) = \int_0^T g(u)K_n(t,u)du \quad , 0 \le t \le T \tag{3.2}$$

Since the time interval is [0,T] , the only part of s(t) that needs to be considered in the equations is that within the time interval [0,T]. It is equal to

$$s_T(t) = \sum_{k=-v}^{0} a_k h(t-kT) \tag{3.3}$$

$s_T(t)$ is affected by the present input and previous v inputs, ie, the length of ISI.The metric, λ_T, also is calculated over this interval.

Since at the output of the receiver the noise samples are statistically independent, one can express the complete metric as

$$\lambda = \sum_k \lambda_{T_k} \tag{3.4}$$

where λ_{T_k} is the branch metric for the k^{th} interval.

To illustrate the procedure of whitening, a model for the noise spectrum has to be assumed. When the spectrum does not have a white noise component, it can be expressed as

$$S_n(\omega) = \frac{N(\omega^2)}{D(\omega^2)} \quad , \quad \text{degree } N(\omega^2) \le \text{degree } D(\omega^2) + 1 \tag{3.5}$$

Since there is no white noise component, the noise autocorrelation function $K_n(t,u)$ does not contain singularities.

The general solution to the integral equation (3.2) in this case is given by

$$g(t)=g_\infty(t) + \sum_i a_i g_i(t) + \sum_k \left[b_k\delta^{(k)}(t) + c_k\delta^{(k)}(t-T)\right] \quad ,0 \le t \le T . \qquad (3.6)$$

$g_\infty(t)$ is the infinite time solution and $g_i(t)$'s are the homogeneous solutions for the corresponding differential equation. The coefficients b_k's and c_k's associated with the impulse function, δ, and its derivatives, $\delta^{(\kappa)}$, are determined by the end conditions.

When a white component is present in the noise spectrum it can be given by

$$S_n(\omega) = \frac{N_0}{2} + S_c(\omega) = \frac{N(\omega^2)}{D(\omega^2)} \qquad (3.7)$$

where $S_c(\omega)$ is the colored component. Both $N(\omega^2)$ and $D(\omega^2)$ are of the same degree due to the white component .

The autocorrelation function therefore, has the form

$$K_n(t,u) = \frac{N_0}{2}\delta(t-u) + K_c(t,u) \qquad (3.8)$$

The integral equation (3.2) now becomes

$$s(t) = \frac{N_0}{2} g(t) + \int_0^T K_c(t,u)g(u)du \qquad (3.9)$$

with a solution of the form

$$g(t) = g_\infty(t) + \sum_i a_i g_i(t) \qquad (3.10)$$

The presence of the white component results in a solution for g(t) that does not contain singular functions.

3.3 Examples

Derivation of the metrics for the noise models given above are presented using the following examples. First finite time whitening is considered.

(i)Noise does not contain a white component

$$S_n(\omega) = \frac{2k\sigma_n^2}{\omega^2 + k^2} \Rightarrow K_n(t,u) = \sigma_n^2 \exp(-k|t-u|) \qquad (3.11)$$

The results are not dependent on the channel and all that is assumed about the channel impulse response, h(t), is that it is of finite duration, LT seconds.

The integral equation corresponding to (3.11) is

$$s_T(t) = \sigma_n^2 \int_0^T \exp(-k|t-u|)g(u)du \qquad (3.12)$$

Converting this into a differential equation, the infinite solution, $g_\infty(t)$, is obtained from the differential equation

$$s''_T(t) + k^2 s_T(t) = 2k\sigma_n^2 g_\infty(t) \quad , 0 \le t \le T \qquad (3.13)$$

and is

$$g_\infty(t) = \frac{1}{2k\sigma_n^2}\left\{- s''_T(t) + k^2 s_T(t)\right\} \qquad (3.14)$$

To get the complete solution for g(t) , (3.13) is substituted back into the integral equation, where δ functions are included in the solution to satisfy the end conditions. Therefore,

$$g(t) = g_\infty(t) + b_1\delta(t) + c_1\delta(t - T) \qquad (3.15)$$

$$b_1 = \frac{ks_T(0) - s'_T(0)}{k\sigma_n^2} \qquad c_1 = \frac{ks_T(T) + s'_T(T)}{k\sigma_n^2} \qquad (3.16)$$

$$s_T(t) = \sum_{l=-v}^{0} a_l\, h(t - lT) \ , \ 0 \le t \le T \qquad (3.17)$$

Given g(t) and $s_T(t)$, the branch metric from equation (3.1) is

$$\lambda_T = \sum_{l=-v}^{0} a_l \left[R_l - \sum_{m=-v}^{0} c_{lm}\, a_m \right] +$$
$$b_1[r(0) - 0.5\, s_T(0)] + c_1[r(T) - 0.5\, s_T(T)] \qquad (3.18)$$

$$\text{where } R_l = \int_0^T \left[k^2 h(t-lT) - h^{(2)}(t - lT)\right] r(t)\, dt \qquad (3.19)$$

$$c_{lm} = 0.5 \int_0^T \left[k^2 h(t-mT) - h^{(2)}(t - mT)\right] h(t - lT)\, dt \qquad (3.20)$$

For the simulation a raised cosine channel of impulse response

$$h(t) = \frac{1}{2LT}\left\{1 - \mathrm{Cos}(\frac{2\Pi}{LT})t\right\} \quad , 0 \le t \le LT \qquad (3.21)$$

is used.

(ii) Colored noise with a white noise component

$$S_n(\omega) = \frac{\sigma_n^2 (\omega^2 + \gamma^2)}{(\omega^2 + k^2)} = \sigma_n^2 + \frac{2k\sigma_c^2}{(\omega^2 + k^2)} \qquad (3.22)$$

Because of the white noise component the results are dependent on the channel impulse response. The response that is used here for the channel is that of the simple low pass single pole filter ;

$$h(t) = a\,e^{-bt} \quad ; \quad 0 \le h(t) \le LT \qquad (3.23)$$

$$s_T(t) = \sum_{l=-v}^{0} a_l\, h(t - lT) \;,\; 0 \le t \le T \qquad (3.24)$$

The integral equation to be solved for g(t) is

$$s_T(t) = \sigma_n^2\, g(t) + \sigma_c^2 \int_0^T e^{-k|t-u|}\, g(u)du \qquad (3.25)$$

From (3.24) the complete solution for g(t) is found to be

$$g(t) = \left[\sum_{l=-v}^{0} a_l e^{blT}\right] (A_1 e^{-bt} + A_2 e^{-\gamma t} + A_3 e^{\gamma t}) \qquad (3.26)$$

The branch metric, from equation (3.1) , then is

$$\lambda_T = \left[\sum_{l=-v}^{0} a_l e^{blT}\right] (R_T - c \sum_{m=-v}^{0} a_m e^{bmT}) \qquad (3.27)$$

where $$R_T = \int_0^T r(t)\,(A_1 e^{-bt} + A_2 e^{-\gamma t} + A_3 e^{\gamma t})dt \qquad (3.28)$$

and c is a known constant.

In the above two examples finite time whitening has been discussed. For comparison consider whitening over an infinite time interval. The metric can be

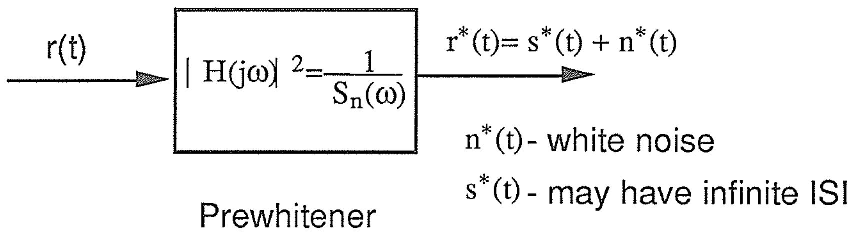


Figure 3.3

obtained simply by finding $g_\infty(t)$ instead of $g(t)$ and evaluating the following equation,

$$\lambda = \int_{-\infty}^{\infty} r(t) g_\infty(t) dt - 1/2 \int_{-\infty}^{\infty} s(t) g_\infty(t) dt \tag{3.29}$$

$$G_\infty(\omega) = \frac{S(\omega)}{S_n(\omega)} \tag{3.30}$$

where $G_\infty(\omega)$ is the Fourier transform of $g_\infty(t)$.

$$s(t) = \sum a_k h(t - kT) \quad , \quad -\infty < t < \infty \tag{3.31}$$

Or, one may use the prewhitener in frequency domain as shown in Figure 3.3.

The metric for this method is

$$\lambda = \int_{-\infty}^{\infty} r^*(t) s^*(t) dt - 1/2 \int_{-\infty}^{\infty} [s^*(t)]^2 dt \tag{3.32}$$

where $\quad s^*(t) = \sum_{l=-N}^{N-1} a_l f(t - lT) = s(t) * h_{pw}(t) \; ; \; f(t) = h(t) * h_{pw}(t) \quad (3.33)$

The metric for the above examples for infinite time whitening can easily be obtained from equations (3.31) and (3.32). Using these, the metric simplifies to

$$\lambda = \sum_{l} \left(R_l - \sum_{j=1}^{v1} a_{l-j} f_j \right) a_l \quad , \quad R_l = \int_{-\infty}^{\infty} r^*(t) f(t - lT) dt \tag{3.34}$$

$$f_j = \int_{-\infty}^{\infty} f(t) f(t - jT) dt \tag{3.35}$$

where v1 is the length of interference due to f(t).

For the first example of single pole noise

$$f(t) = c_1.h(t) + c_2.h'(t) \tag{3.36}$$

where c_1 and c_2 are constants.

In general if the noise spectrum is all pole then f(t) consists only of derivatives of h(t). Therefore in this case the number of interference terms is equal to that obtained for the finite solution.

For the second example $f(t) = c_3.h(t) + c_4.e^{-\gamma t}$ (3.37)

where c_3 and c_4 are constants.

Because of the term $e^{-\gamma t}$, f(t) lasts for an infinite time. This means that in this case if the infinite time solution is to be used infinite intersymbol interference results. Therefore in this case the tail of the interference terms has to be truncated to v1 which is generally larger than the interference terms due to channel.

3.4 Comparison between finite and infinite time metrics

For the finite time solution initially the branch metric is evaluated which is directly used by the Viterbi algorithm. Here g(t) is always dependent on the present input and the previous v inputs which constitute the ISI. Thus the general form of g(t) is

$$g(t) = \sum_{l=-v}^{0} a_l f_l(t) \quad , 0 \leq t \leq T \tag{3.38}$$

Therefore

$$\lambda_T = \sum_{l=-v}^{0} \int_0^T r(t) f_l(t)dt - 1/2 \int_0^T s(t)g(t)dt \tag{3.39}$$

$$\lambda_T = \sum_{l=-v}^{0} a_l R_l - 1/2 \int_0^T s(t)g(t)dt \tag{3.40}$$

λ_T is the branch metric which is affected only by the v interfering terms.Thus the amount of ISI is independent of the whitening process. The term $\sum_{l=-v}^{0} a_l R_l$ indicates that L observed variables are necessary to calculate λ_T. Thus L matched filters are required at the receiver.

For the infinite time whitening as seen from equation (3.32)

$$f(t)= h(t)*h_{pw}(t)$$

If $S_n(\omega)$ contains zeros they would appear in $h_{pw}(t)$ as exponential terms $\sum_i p_i e^{-a_i t}$ which span an infinite time. This immediately implies that unless these terms are truncated, infinite ISI results .

The complete metric, from equation (3.33)

$$\lambda = \sum_l \left(R_l - \sum_{j=1}^{v1} a_{l-j} f_j \right) a_l \quad , \; R_l = \int_{-\infty}^{\infty} r^*(t)\, f(\, t\text{-}lT \,)\, dt$$

Hence only one observed variable is needed to calculate the metric requiring only one matched filter as opposed to L required for the implementation of the finite time metric.

3.5 Application of the Sequential algorithm

To apply the sequential algorithm the approach developed by Xiong is used here. For this approach a sufficient condition is that the noise samples at the output of the receiver must be statistically independent. A prewhitener is used up front to whiten the colored noise as in the previous section dealing with the infinite time interval whitening. This, as shown in section 3.3, could lead to infinite intersymbol interference. Since the output of the prewhitener can be considered as a signal with intersymbol interference in additive white Gaussian noise, a whitened matched filter is followed next as shown by Forney [5], representing the standard procedure for this kind of system. The communication system is shown in Figure 3.4 in block diagram form.

$$s^*(t) = s(t) * h_{pw}(t) = x(t) * h(t) * h_{pw}(t) \tag{3.41}$$

$$= x(t) * h_R(t) = \sum_k x_k h_R(t\text{-}kT) \tag{3.42}$$

Now as in the usual case sufficient statistics can be obtained after passing $r^*(t)$ through

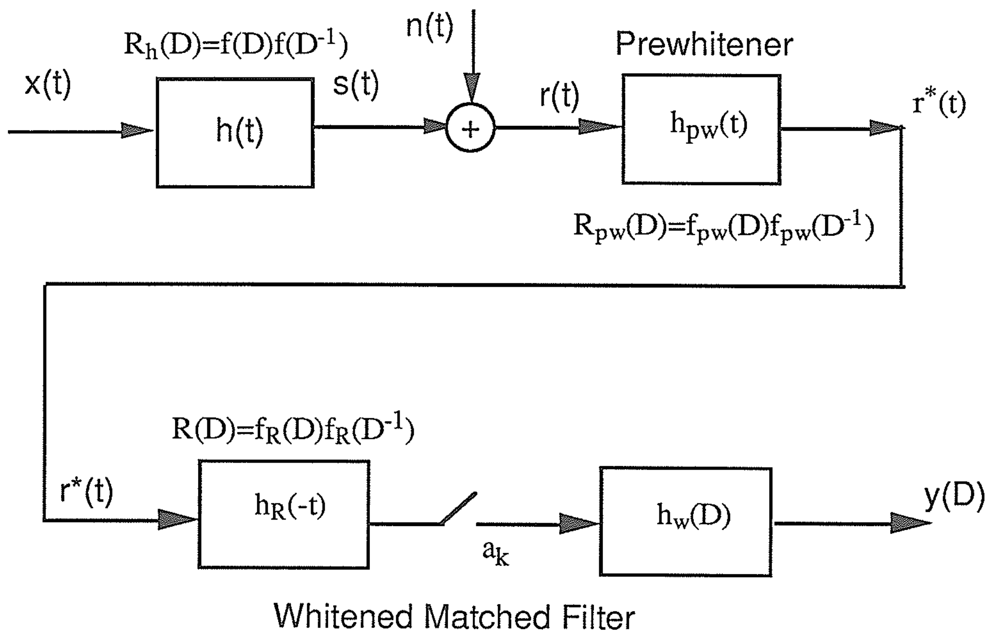

$R_h(D)=f(D)f(D^{-1})$
n(t)
Prewhitener
x(t)
h(t)
s(t)
+
r(t)
$h_{pw}(t)$
$r^*(t)$
$R_{pw}(D)=f_{pw}(D)f_{pw}(D^{-1})$
$R(D)=f_R(D)f_R(D^{-1})$
$r^*(t)$
$h_R(-t)$
a_k
$h_w(D)$
y(D)
Whitened Matched Filter

Figure 3.4

the matched filter $h_R(-t)$. At the output of $h_R(-t)$ the noise samples have to be whitened again.

$$a_j = \int_{-\infty}^{\infty} \sum_k x_k\, h_R(t - kT)\, h_R(t - jT)\, dt + \int_{-\infty}^{\infty} w(t)\, h_R(t - jT)\, dt \qquad (3.43)$$

$$= \sum_i x_{j-i}\, h_{R_i} + n_j \qquad (3.44)$$

In D transform notation

$$a(D) = x(D)R(D) + n(D) \qquad (3.45)$$

Using spectral factorization

$$R(D) = f_R(D)\, f_R(D^{-1}) \qquad (3.46)$$

where R(D) is the discrete autocorrelation function of $h_R(t)$.

Following Forney's approach

$$a(D) = x(D)f_R(D)\, f_R(D^{-1}) + w(D)f_R(D^{-1}) \qquad (3.47)$$

Since statistically independent samples are required, a(D) has to be passed through a whitening filter in the discrete domain, ie, $1/f_R(D^{-1})$.

$$y(D) = \frac{a(D)}{f_R D^{-1})} = x(D)f_R(D) + w(D) \qquad (3.48)$$

$$h_w(D) = \frac{1}{f_R(D^{-1})} \qquad (3.49)$$

where $h_w(D)$ is the whitening filter.

Since $h_R(t) = h(t) * h_{pw}(t)$, R(D) can be expressed as

$$R(D) = R_h(D) * R_{pw}(D) \qquad (3.50)$$

Also $\quad R_h(D) = f(D)f(D^{-1})$ and $R_{pw}(D) = f_{pw}(D)f_{pw}(D^{-1}) \qquad (3.51)$

Therefore one can express $f_R(D) = f(D)f_{pw}(D) \qquad (3.52)$

Thus $\quad y(D) = x(D)f(D)f_{pw}(D) + w(D) \qquad (3.53)$

y_j can be found from this and hence the sequential algorithm can be applied following the standard procedure. In general the form of $f_{pw}(D)$ is $\frac{V(D)}{U(D)}$ which usually gives rise to infinite ISI.

$$y(D) = x(D)f(D)\frac{V(D)}{U(D)} + w(D) \tag{3.54}$$

Chapter 4

Simulation Results

4.1 Introduction

Simulations have been carried out using both the Viterbi algorithm and the Sequential algorithm for different channels, namely the truncated single pole channel, the Butterworth channel and the raised cosine channel. In the case of the Viterbi algorithm, the finite time metric and infinite time metric were investigated. The Sequential algorithm was applied using the approach described in 3.4.

4.2 Simulation : General

All the simulations were run on the IBM AMDHAL V7 mainframe computer at the University of Manitoba. The programming language used was FORTRAN 77. The sequential algorithm used in the simulations is a modified version of the one developed by Dr. F.Xiong, and a listing of the program is given in appendix C. Also included is a listing of a program for the Viterbi algorithm in appendix B.

To generate the white noise samples a random number generator subroutine was used. By passing the white noise samples through a shaping filter in the discrete time domain colored noise samples were generated. The discrete shaping filter must be such that the output noise samples have the corresponding discrete autocorrelation function of the colored noise considered. This can usually be done by first finding the shaping filter in the continuous time domain and then obtaining the discrete spectral factorization of it's autocorrelation function. This procedure is described in detail in appendix A along with an example.

4.3 The Raised Cosine Channel

The first channel studied was the raised cosine channel ; a channel commonly found in communication systems. The noise model is single pole, zero mean colored Gaussian noise. As shown in 3.2 the intersymbol interference in this case is finite. Different bandwidths of the noise with respect to the signal were considered for two different lengths of interference ; the impulse response being three symbol intervals long and five symbol intervals long.

For an intersymbol interference channel in additive white Gaussian noise the signal to noise ratio is defined at the output of the whitened matched filter. It cannot be defined at the receiver input because the noise power is infinite there. However the power of colored noise is finite, and therefore the signal to noise ratio is, for colored noise, defined at the receiver input.

The simulation results for the finite time metric and the infinite time metric with the Viterbi algorithm are presented in Figures 4.1-4.5. From the results it is seen that the finite time metric performs better when the bandwidth of the colored noise is less than that of the signal, or when the noise looks more colored.

4.4 The Truncated Single Pole Channel

This channel was used to gain more insight into the colored noise problem. The impulse response of the channel was truncated at three symbol intervals. The noise in this case was single pole colored noise with a white component,

Raised Cosine Channel

Noise BW = 0.5Signal BW

2 interference terms

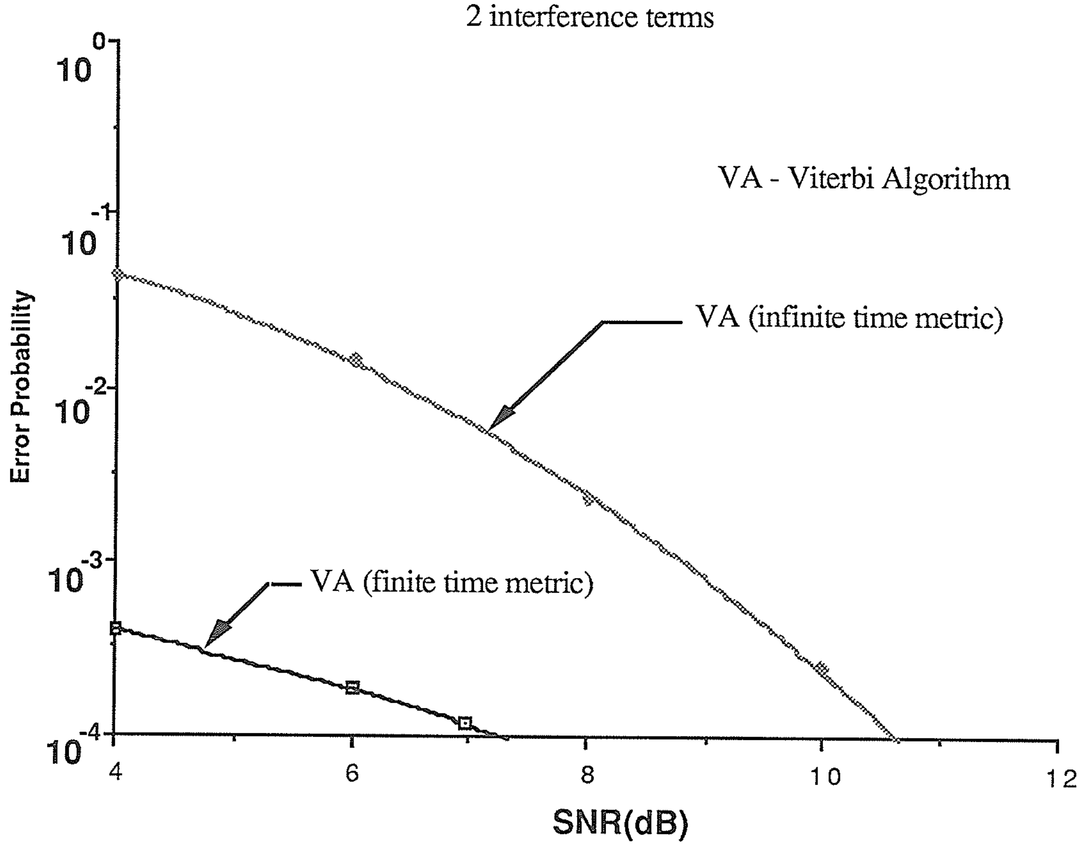


Figure 4.1

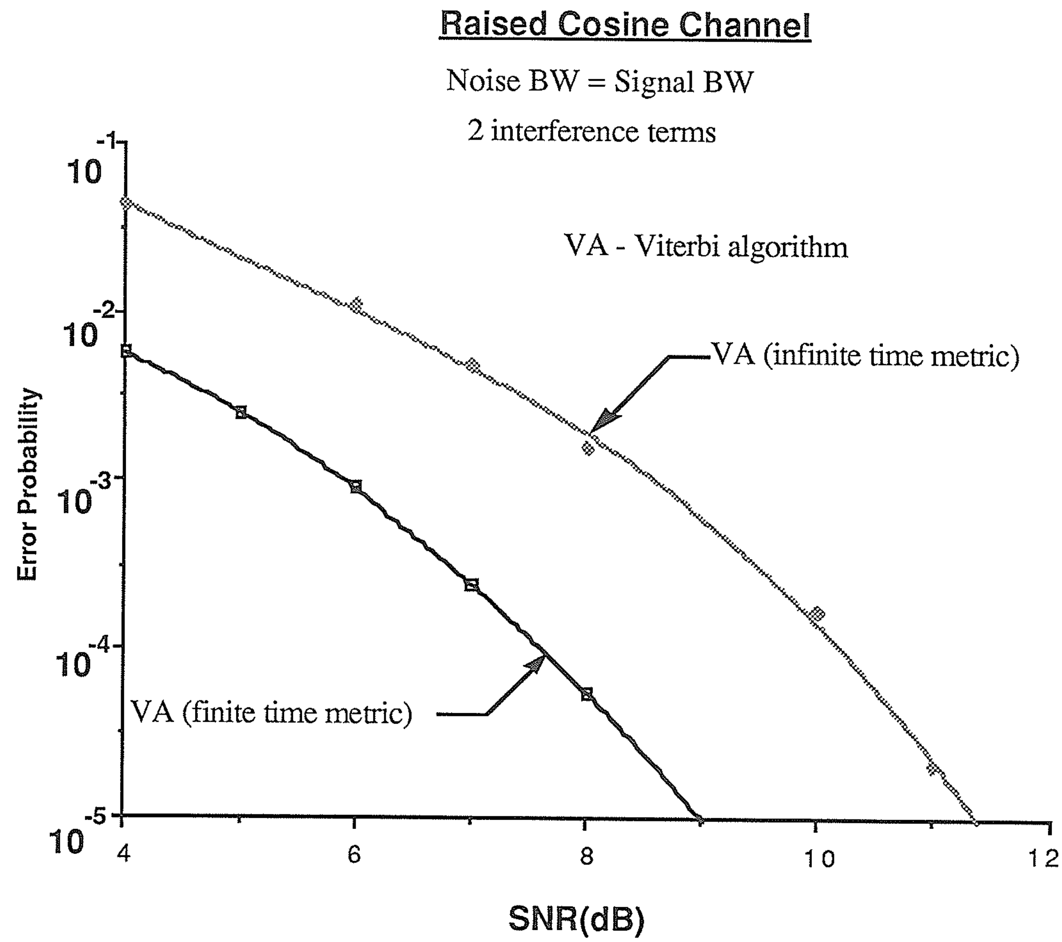
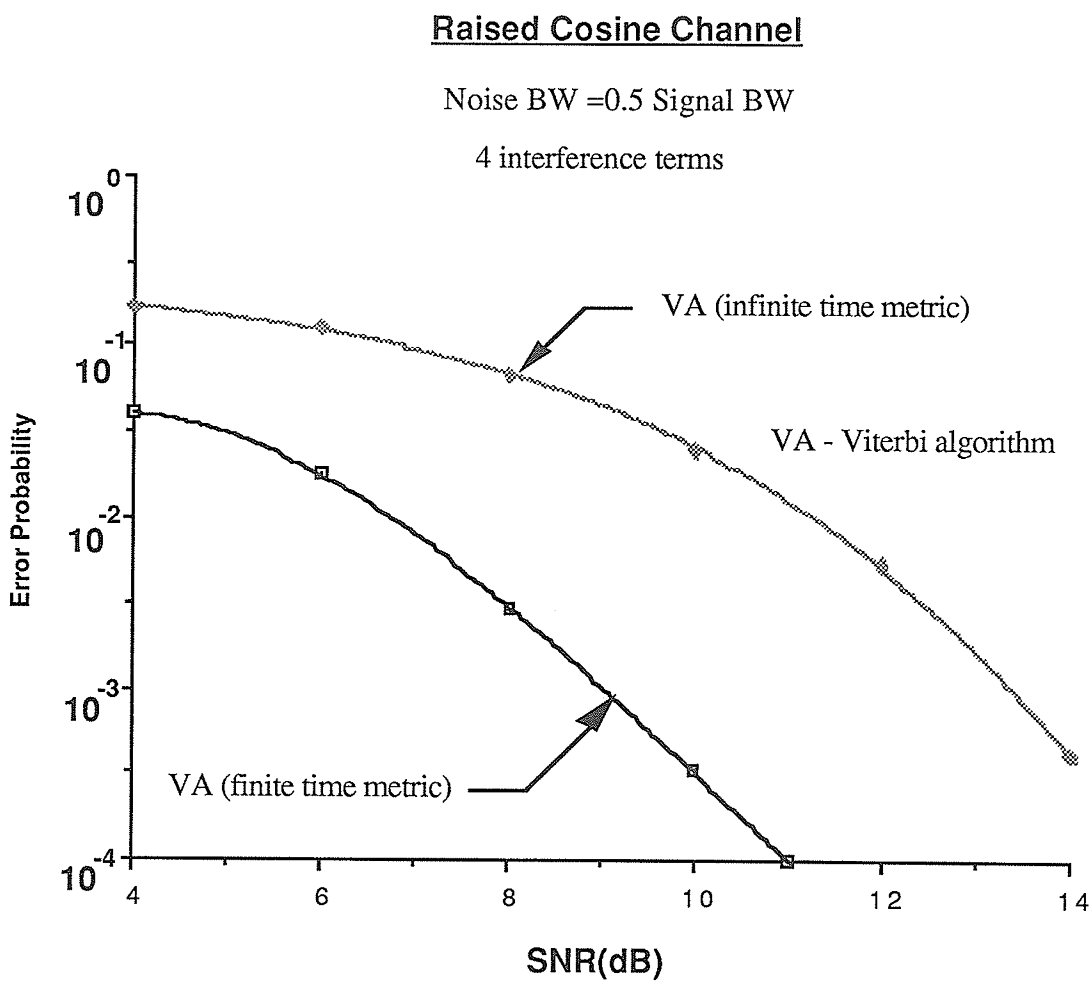

Raised Cosine Channel
Noise BW = Signal BW
2 interference terms
VA - Viterbi algorithm
VA (infinite time metric)
VA (finite time metric)
Error Probability
10^-1
10^-2
10^-3
10^-4
10^-5
4
6
8
10
12
SNR(dB)

Figure 4.2



Raised Cosine Channel
Noise BW =0.5 Signal BW
4 interference terms
VA (infinite time metric)
VA - Viterbi algorithm
VA (finite time metric)
Error Probability
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
4
6
8
10
12
14
SNR(dB)

Figure 4.3

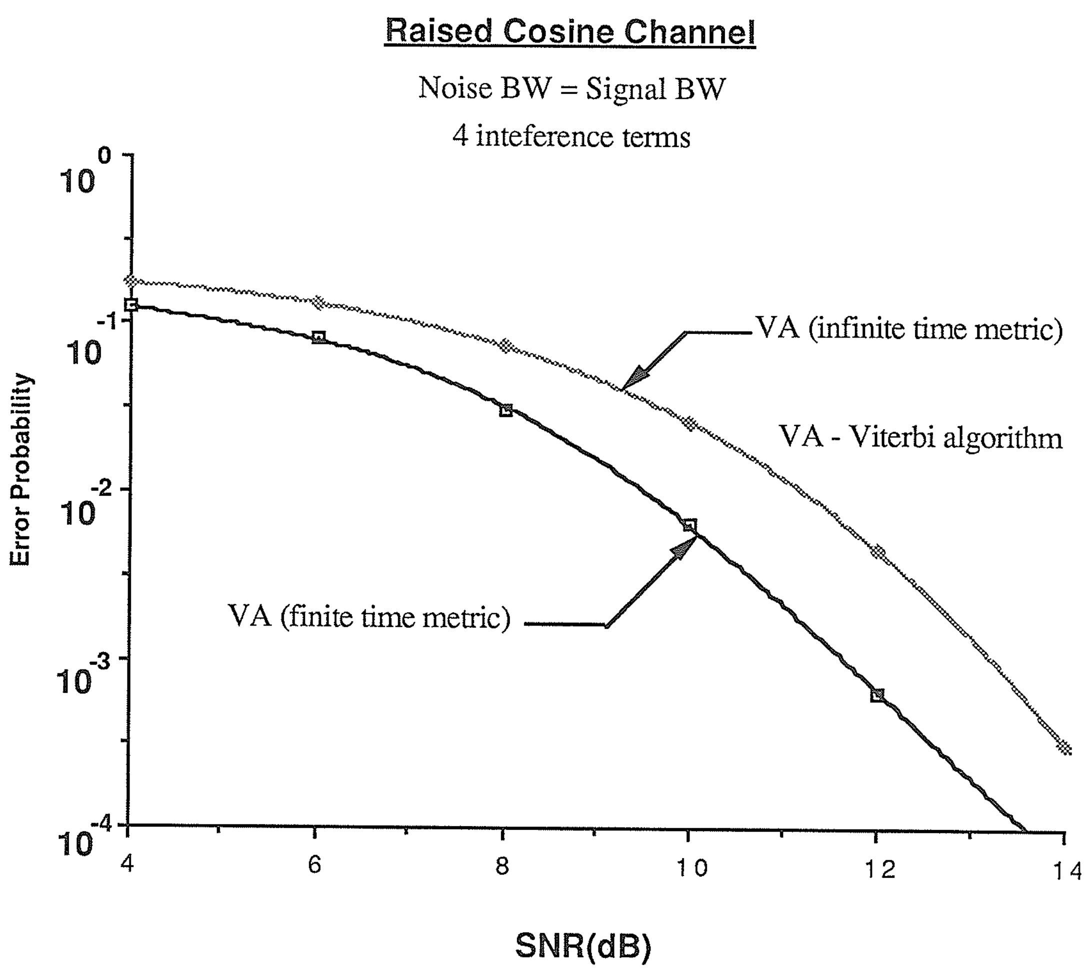
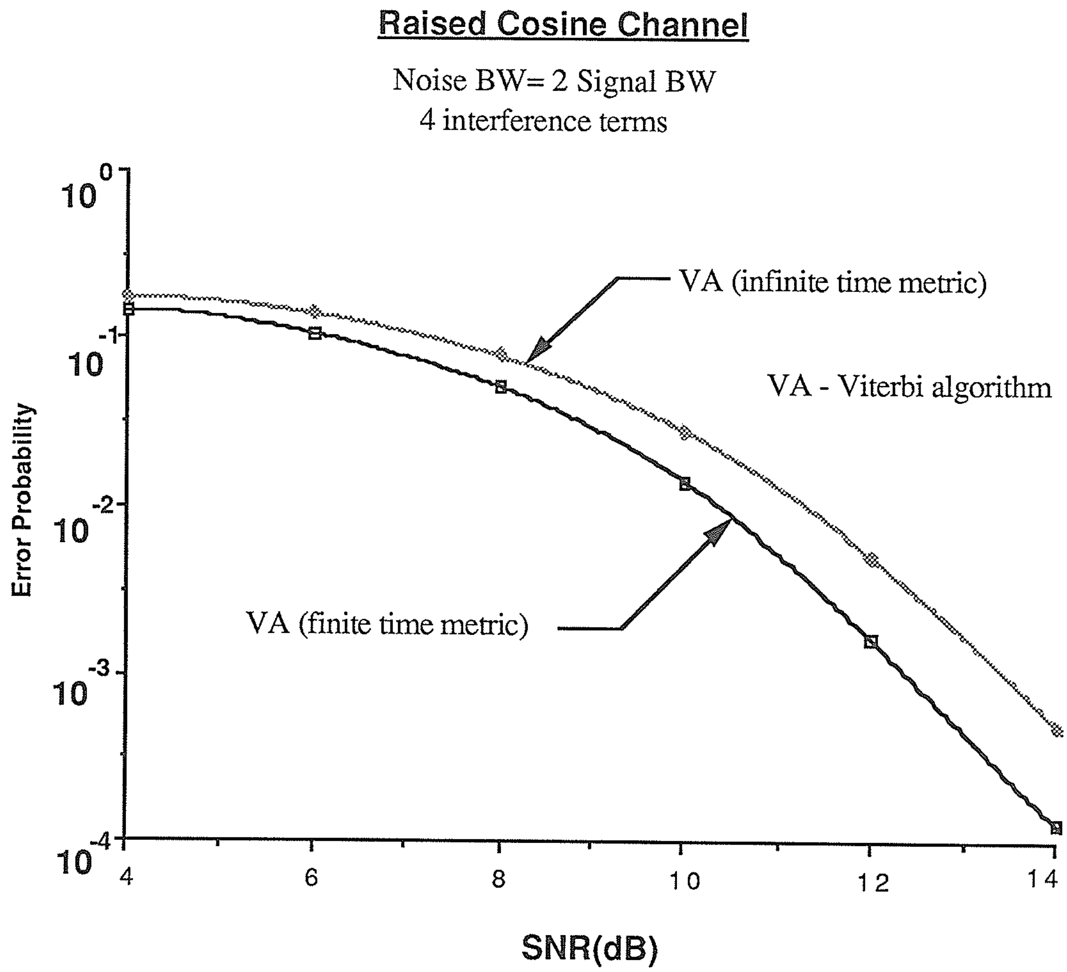

Raised Cosine Channel
Noise BW = Signal BW
4 inteference terms
Error Probability
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
4
6
8
10
12
14
SNR(dB)
VA (infinite time metric)
VA - Viterbi algorithm
VA (finite time metric)

Figure 4.4



Raised Cosine Channel
Noise BW= 2 Signal BW
4 interference terms
VA (infinite time metric)
VA - Viterbi algorithm
VA (finite time metric)
Error Probability
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
4
6
8
10
12
14
SNR(dB)

Figure 4.5

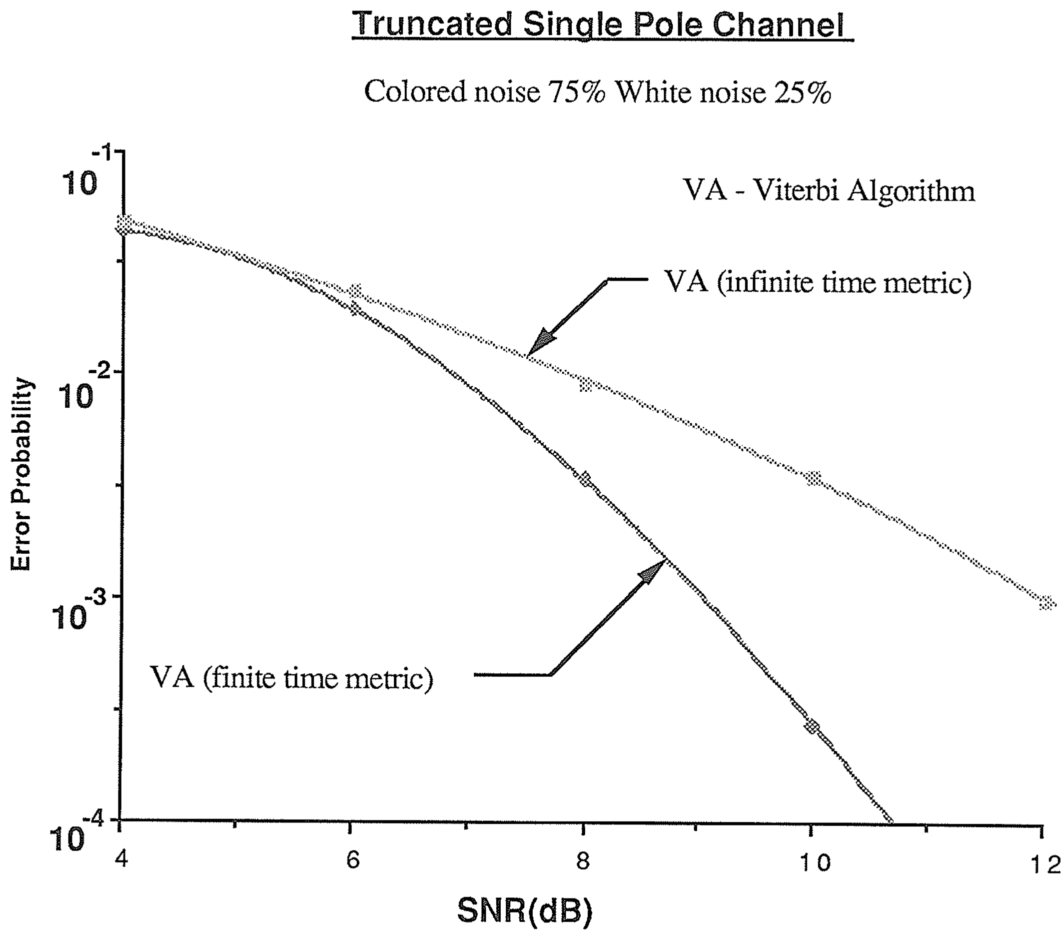
and therefore the noise spectrum has a zero and a pole. Three different mixtures of the colored and white components of the noise spectrum were considered, with the bandwidth of the colored component being the same as that of the signal.

Both the finite time and infinite time metrics for the Viterbi algorithm were applied to this channel with the signal to noise ratio is defined at the receiver input here as well. For the infinite time case, infinite intersymbol interference results due to the zero in the noise spectrum. Thus the interference terms were truncated to five terms. The truncation was based on the decay of the coefficients, where the coefficients smaller than 0.1% of the leading coefficient, were neglected. For all three noise mixtures the length of truncation was five terms. The number of the states of the trellis for the finite time metric is four compared to thirty two required for the truncated infinite time metric.

The Sequential algorithm was also applied for each of the noise mixtures in an effort to compare the performances of the three approaches. The signal to noise ratio (SNR) was defined at the output of the whitened matched filter as it is the standard way to define the SNR. Simulation results for the algorithms are shown in Figures 4.6-4.9. The number of computations and the CPU time taken per decoded symbol by the two algorithms in each noise mixture are given in Tables 4.1a and 4.1b.

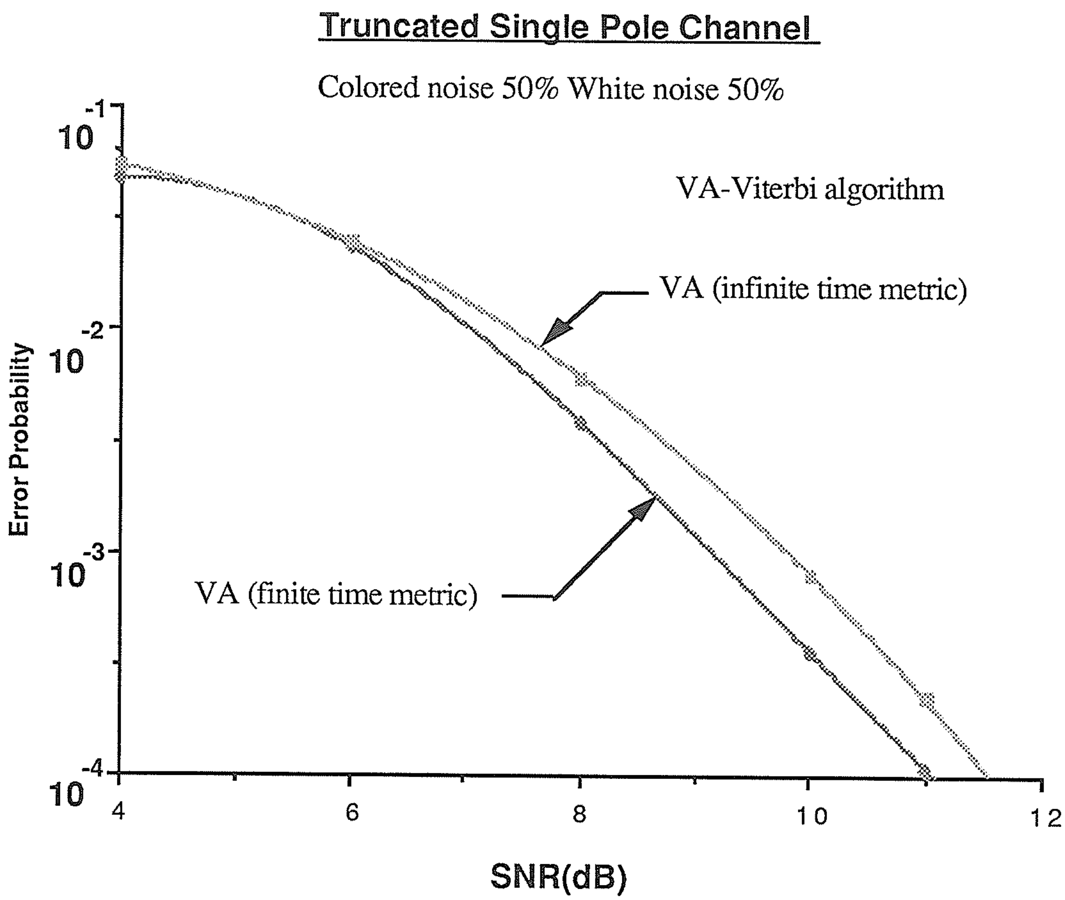
4.5 The Butterworth Channel

Finally the Butterworth channel, another commonly encountered channel in communication systems was studied. Since this channel produces infinite intersymbol interference only the Sequential algorithm was applied. For the simulation a two pole Butterworth channel was selected. The noise in this case was Butterworth with a white component ; i.e. having a two pole two zero spectrum. As in the single pole case three different combinations of the colored and white components were considered. The



Truncated Single Pole Channel
Colored noise 75% White noise 25%
VA - Viterbi Algorithm
VA (infinite time metric)
VA (finite time metric)
Error Probability
10^{-1}
10^{-2}
10^{-3}
10^{-4}
4
6
8
10
12
SNR(dB)

Figure 4.6



Truncated Single Pole Channel
Colored noise 50% White noise 50%
VA-Viterbi algorithm
VA (infinite time metric)
VA (finite time metric)
Error Probability
10^{-1}
10^{-2}
10^{-3}
10^{-4}
4
6
8
10
12
SNR(dB)

Figure 4.7

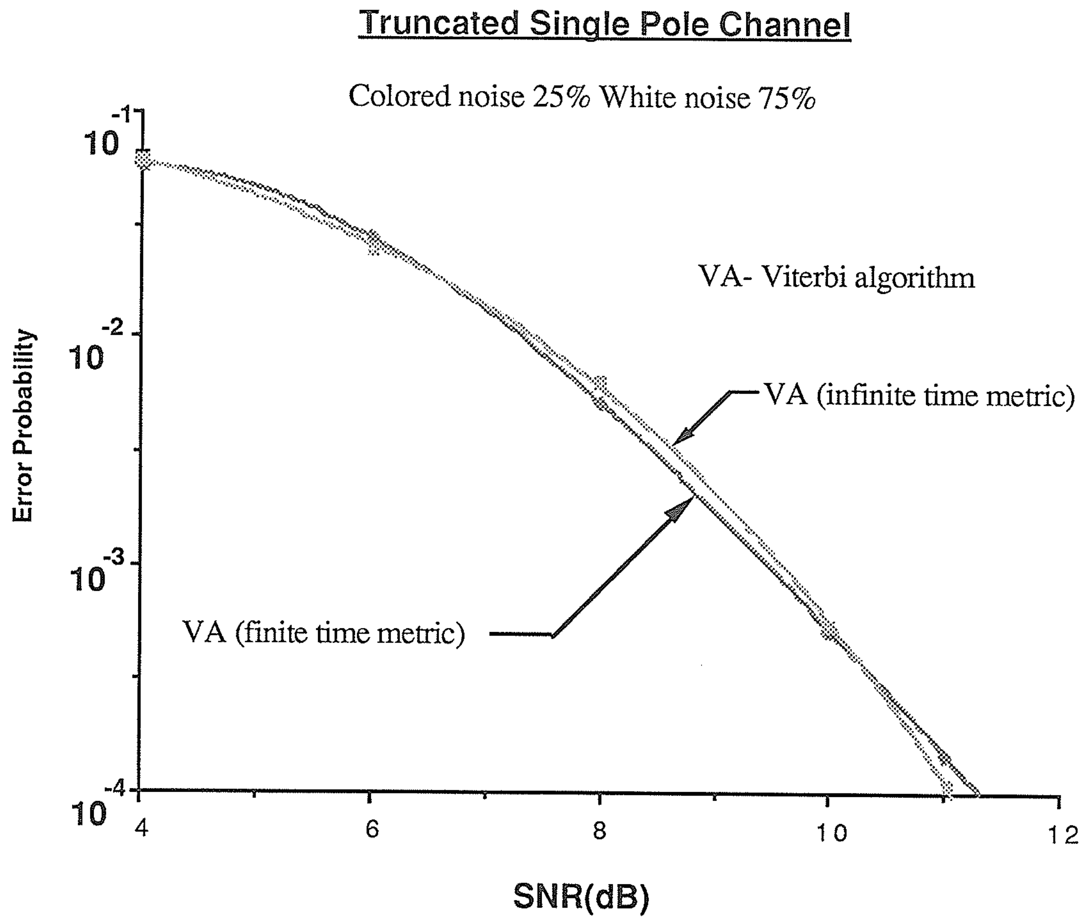
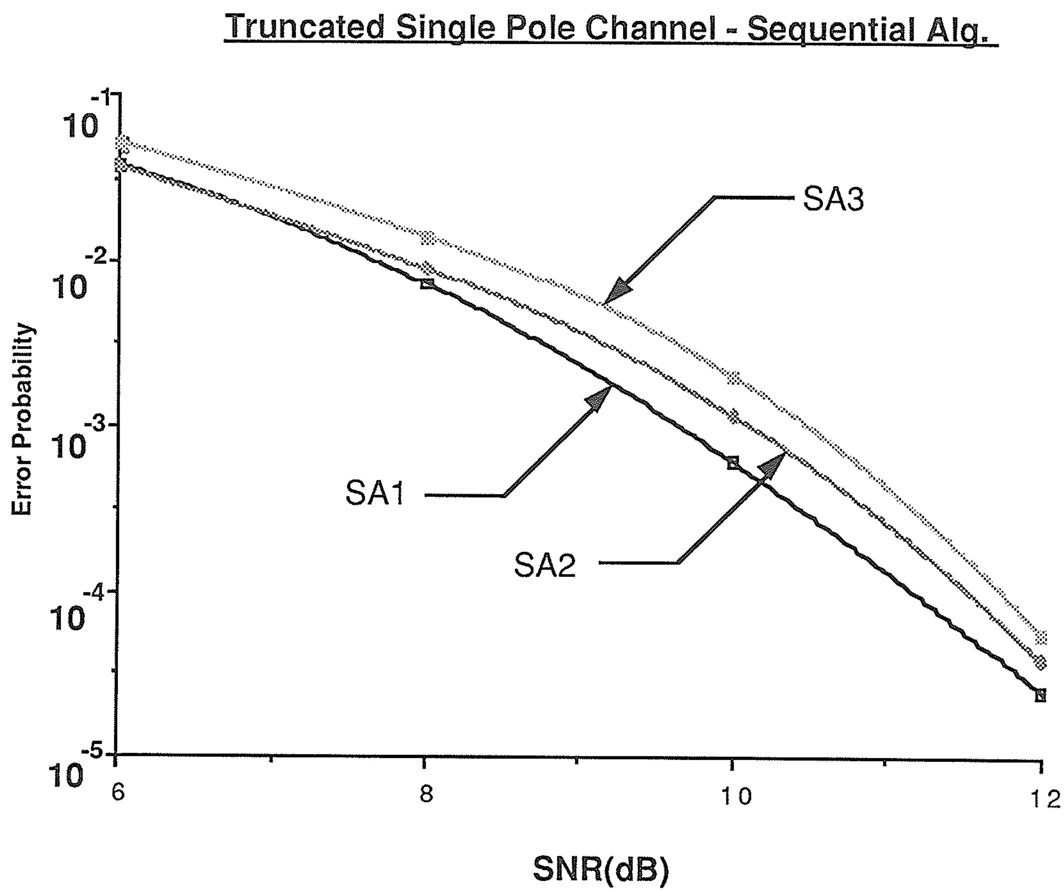

Truncated Single Pole Channel
Colored noise 25% White noise 75%
VA- Viterbi algorithm
VA (infinite time metric)
VA (finite time metric)
Error Probability
10^{-1}
10^{-2}
10^{-3}
10^{-4}
4
6
8
10
12
SNR(dB)

Figure 4.8



SA1 -Col. noise 75% White 25%

SA2-Col. noise 50% White 50%

SA3-Col. noise 25% White 75%

Figure 4.9

Average no. of Computations / decoded symbol

SNR(dB)	Sequential 25% W. N.	Sequential 50% W. N.	Sequential 75% W. N.	VA1 L=3	VA2 L=6
6	9.82	9.71	8.9	4	32
8	4.62	4.8	5.17	4	32
10	3.34	3.46	3.59	4	32
12	2.72	2.8	2.92	4	32

W.N.-White Noise

Single Pole Truncated Channel

VA1- Viterbi finite time

VA1- Viterbi infinite time

Table 4.1a

Average CPU time / decoded symbol (seconds)

SNR(dB)	Sequential 25% W. N.	Sequential 50% W. N.	Sequential 75% W. N.	VA1 L=3	VA2 L=6
6	0.107	0.107	0.105	0.002	0.016
8	0.042	0.045	0.05	0.002	0.016
10	0.026	0.027	0.03	0.002	0.016
12	0.02	0.021	0.023	0.002	0.016

W.N.-White Noise

Single Pole Truncated Channel

VA1- Viterbi finite time

VA1- Viterbi infinite time

Table 4.1b

signal to noise ratio was defined at the output of the whitened matched filter.

Figure 4.10 gives the results for the channel in each case. The number of computations and the CPU time per decoded symbol are also given as shown in Tables 4.2a and 4.2b. The computational complexity of the Sequential algorithm is compared with that of the Viterbi algorithm, which would be required had a simulation been done obtaining a reasonable length of truncation for the interference terms using the procedure described in 4.4. The Sequential algorithm seems to be following the general trend of giving a better error performance when the noise mixture is more colored.

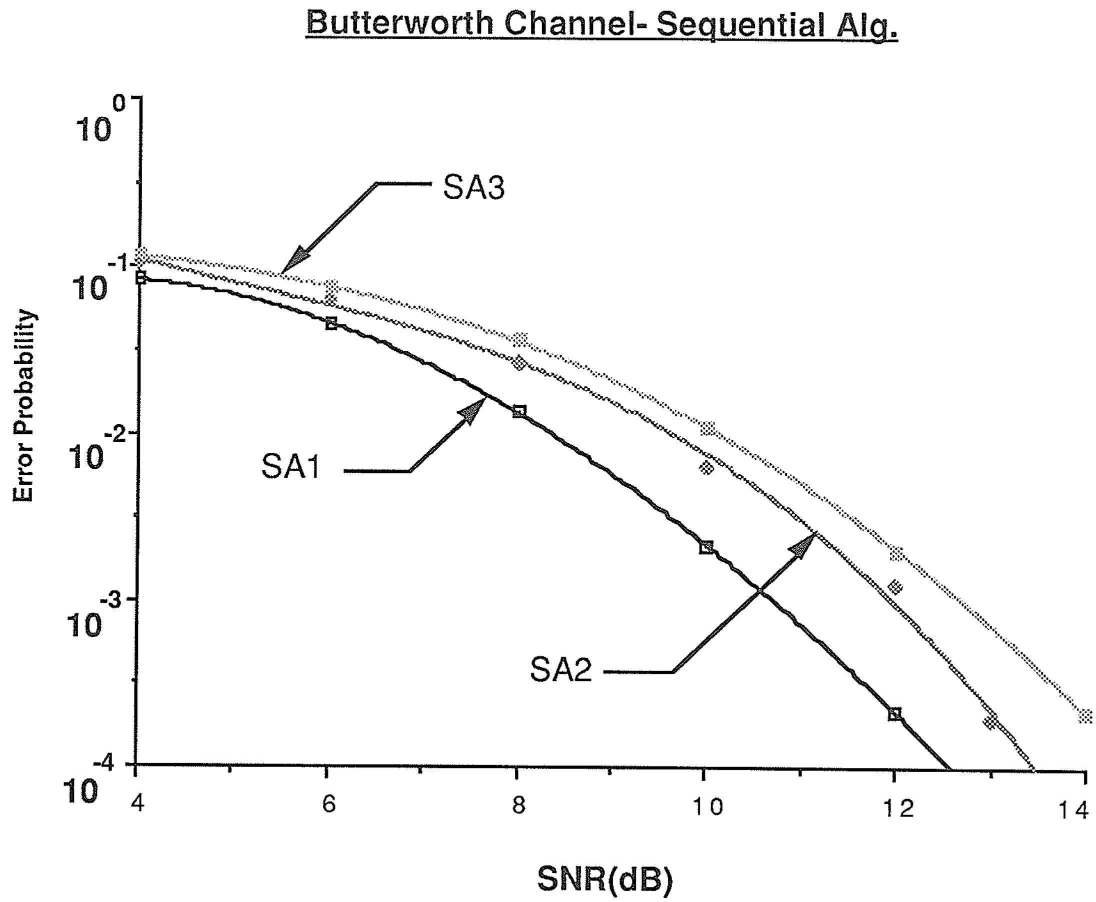
4.6 Summary and Discussion

4.61Summary

(a) The Viterbi Algorithm

(i) The Raised Cosine Channel Results : Single Pole Noise

For the raised cosine channel the finite time metric performs better when the noise bandwidth decreases relative to signal bandwidth. The computational complexity of both metrics with the Viterbi algorithm is the same for this case. Generally, in the case of all pole colored noise model the prewhitener consists of differentiators, which do not produce more interference terms. Thus the amount of interference present is determined by the channel. The solutions for the finite time metric and the infinite time metric are different in this situation because of the samplers present in the finite solution due to the singular functions added to satisfy the end conditions of the integral equation. Therefore the Viterbi algorithm behaves differently for each metric.



SA1 -Col. noise 75% White 25%

SA2-Col. noise 50% White 50%

SA3-Col. noise 25% White 75%

Channel Transfer Function : $|H(j\omega)|^2 = \dfrac{1}{1 + (\frac{\omega}{\omega_c})^4}$

Figure 4.10

No. of Computations / decoded symbol

SNR(dB)	Sequential 25% W. N.	Sequential 50% W. N.	Sequential 75% W. N.	Viterbi L=10
4	17.35	17.35	17.39	512
6	5.4	5.64	6.14	512
8	1.77	1.9	2.1	512
10	1.42	1.47	1.58	512
12	1.31	1.37	1.44	512

W.N.-White Noise

Butterworth Channel

Table 4.2a

Average CPU time / decoded symbol (seconds)

SNR(dB)	Sequential 25% W. N.	Sequential 50% W. N.	Sequential 75% W. N.	Viterbi L=10
4	0.184	0.185	0.189	0.303
6	0.055	0.056	0.064	0.303
8	0.013	0.014	0.016	0.303
10	0.011	0.011	0.012	0.303
12	0.01	0.01	0.011	0.303

W.N.-White Noise

Butterworth Channel

Table 4.2b

(ii) The Truncated Single Pole Channel Results : Pole Zero Noise

In the case of the truncated single pole channel where the noise has a pole zero spectrum the error performance of the finite and infinite metrics with the Viterbi algorithm are quite close though the finite time metric is the better one.

(iii) Discussion of Metrics

Generally, the simulations indicate that the finite time metric gives a better error performance than the infinite time metric. The main advantage of the finite time metric is that no truncation is required if the channel intersymbol interference is finite. Therefore the number of computations and the CPU time is mainly determined by the channel.

For the infinite time metric the effective intersymbol interference is determined by the convolution of the channel impulse response and the prewhitener impulse response. When there is a zero in the noise spectrum giving rise to an exponential term in the impulse response for the prewhitener, the effective impulse response increases giving rise to a large number of interference terms. Since these should be truncated to apply the Viterbi algorithm, the length of truncation determines the computational complexity for the infinite time metric.

(b) The Sequential Algorithm

The metric for the Sequential algorithm can be considered as an infinite metric without truncation. First consider the single pole channel. The error performance of the Sequential algorithm is better when the noise is more colored. It is difficult to compare the error performances of the Sequential and Viterbi algorithms since the signal to noise ratios have been defined at different points. When the computational complexities are compared, one can easily see that as expected, if the signal to noise ratio is sufficiently high the number of computations required for the Sequential algorithm is much less than that required by the Viterbi algorithm .

For the Butterworth channel a Viterbi simulation was not carried out. The computational requirements are compared based on a truncated model. As in the previous case the computational complexity of the Sequential algorithm is much less justifying its use in the colored noise problem.

4.62 Discussion

In general, from the simulation results one can conclude that the algorithms especially the Viterbi algorithm with the finite time metric and the Sequential algorithm perform better when the noise is more colored, either compared to the signal or to the noise itself when it contains a white component.

This result may be explained in the following way. When the noise samples are correlated the receiver can extract more information about the transmitted sequence rather than when the noise samples are statistically independent. The metric for the colored noise represents this information contained in the correlation of the received samples. Therefore, the metric gives a better performance when the noise is more colored. For the extreme case in the context of the colored noise, the noise spectrum becomes a delta function in the frequency domain or alternatively noise can be treated as an unknown constant in the time domain. Thus one can obtain virtually an error free channel since the noise amplitude does not change with time.

Chapter 5

Conclusions

The problem of the maximum likelihood sequence estimation (MLSE) of data in the presence of the additive colored Gaussian noise has been studied in this thesis. Application of both the Viterbi and the Sequential algorithms have been discussed. The approach is based on the whitening of the colored noise using a prewhitener at the receiver. In general , prewhitening results in infinite intersymbol interference.

Thus, to apply the Viterbi algorithm one should obtain a whitening filter with a finite impulse response. This is achieved by solving an integral equation derived using the noise autocorrelation function as shown in [21]. The time interval selected for whitening is one symbol interval over which the branch metric is evaluated in the algorithm. Basically two types of noise models have been considered ; one which contains a white component and the other does not. For the latter case singular functions appear in the solution for the corresponding integral equation to satisfy the end conditions. The receiver in this finite time metric case needs L matched filters at the receiver where the duration of the impulse response of the channel is LT seconds. The amount of intersymbol interference for the calculation of the metric is determined by the channel, independent of whitening for this procedure.

If one considers an infinite time interval for whitening, exponential terms are present in the prewhitener impulse response when the noise spectrum contains zeros. The total amount of interference terms that need to be considered increases due to this reason. Therefore, these terms have to be truncated in order to apply the Viterbi algorithm. The Sequential algorithm, on the other hand, does not require any truncation as it can be applied to the infinite intersymbol interference case. The procedure for the Sequential algorithm has been developed based on the Xiong' s work [24]. The receiver requires only one matched filter with the metric being the same as that derived by Xiong.

Simulation results have been obtained for both the Viterbi and the Sequential algorithms considering several channels and noise types. The error performance and the computational complexity have been analyzed. The results are compared to those obtained using an infinite time metric with the Viterbi algorithm. From the simulation, one can see that the finite time metric and the Sequential algorithm give a better error performance, when the noise looks more colored. Also at relatively high signal to noise ratios, the computational complexity of the Sequential algorithm is comparatively less than that for the Viterbi algorithm.

Performance of the Sequential algorithm needs further investigation to determine whether it in fact gives a better error performance when the noise is more colored and also to gain more knowledge about the algorithm since it has only been recently applied to the intersymbol interference problem. Establishment of the equivalence of the Viterbi and Sequential receivers is required to properly compare the error performances of the two algorithms. Also the application of the reduced state algorithms, such as the M algorithm and the T algorithm is a possible extension to the work presented here.

REFERENCES

[1] M.Austin, *Decision Feedback Equalization for Digital Communication over Dispersive Channels*, M.I.T.Research Lab. on Electronics, Tech. Report 461, Aug.1967.

[2] R.W.Chang, J.C. Hancock, *On Receiver Structure for Channels Having Memory*, IEEE Trans. on Inf. Theory, Vol IT-12, pp.463-468, Oct. 1966.

[3] A. Duel and C. Heegard, *Delayed Decision Feedback Sequence Estimation*, 23rd Annual Allerton Conf. on Comm., Control and Computing, pp. 878-887, 1985.

[4] M. V. Eyuboglu and S. U. Qureshi, *Reduced State Sequence Estimation with Set Partitioning and Decision Feedback*, IEEE Trans. on Comm., Vol 36, No. 1, pp.13-20, Jan. 1988.

[5] G.D. Forney, Jr., *Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference*, IEEE Trans. on Inf. Theory , Vol IT-18, pp. 363-378, May 1972.

[6] R. A. Gonsalves, *Maximum Lilkelihood Receiver for Digital Data Transmission*, IEEE Trans. on Comm. Tech., Vol COM- 16, pp.392-398, June 1968.

[7] E.R. Kretzmer, *Binary Data Communication by Partial Response Transmission*, Conference record, IEEE Annual Comm. Convention, pp451-455, 1965.

[8] J. H. Lee, R. H. French, *Error behaviors of binary DPSK system due to Intersymbol Interference and Correlated Noise*, Conference record of IEEE 1978 National Telecom. Conf., Pt II, Birmingham, AL, USA, Dec. 1978, pp. 27.2.1-27.2.5.

[9] A. Lender, *Correlative Digital Communication Techniques*, IEEE Trans. on Comm. Tech, Vol. COM-12, pp.128-135, Dec.1964.

[10] A.Lender, *Correlative Level Coding for Binary - Data Transmission*, IEEE Spectrum, Vol.3, No.2, pp104-115, Feb.1966.

[11] R. W. Lucky, J. Salz, and E. J. Weldon, Jr., *Principles of Data Communication.*, McGraw-Hill, Inc., New York, N. Y., 1968.

[12] H. Nyquist, *Certain Topics in Telegraph Transmission Theory*,Trans. AIEE (Comm. and Electronics), Vol 47,pp 617-644, April 1928.

[13] A. Polydoros and D. Kazakos, *Maximum Likelihood Sequence Estimation in the Presence of Infinite Intersymbol Interference*, Proc. ICC'79 , Boston, MA, June 1979.

[14] S.Prasad, S.Pathak, *Optimum data receivers for low SNR data signals in non-Gaussian noise and intersymbol interference:receiver structures and their performance analysis*, IEE Proceedings, Vol.35, No.5, October 1988, pp.471 - 480.

[15] N. Seshadri, J. B. Anderson, An *M-Algorithm Receiver for Severe Infinite Intersymbol Interference Channels*, Abstracts of papers, International Symposium on Information theory, Kobe, Japan, pp.68, June 1988.

[16] S. Shitz, *Maximum likelihood detection of discrete time PAM in Non-Gaussian Noise*, Convention of Electrical & Electronic Engineers in Israel Proceedings, March 1985, p1.1.1/1-6.

[17] S. J. Simmons, *Breadth - First Trellis Decoding with Adaptive Effort*, IEEE Trans. on Comm., Vol. 38, No. 1, pp. 3-12, June 1990.

[18] M. Tomlinson, *New automatic equalizer employing modulo arithmetic*, Electron. Lett.,Vol. 7, pp. 138 - 139, 1971.

[19] G. Ungerboeck, *Nonlinear Equalization of Binary Signal in Gaussian Noise*, IEEE Trans. on Comm. Tech., Vol. COM-19, pp. 1128-1137, Dec. 1971.

[20] G.Ungerboeck, *Adaptive Maximum-Likelihood Receiver for Carrier-Modulated Data- Transmission Systems*, IEEE Trans. on Comm., Vol. 22, No. 5, May 1974 , pp. 624 - 636 .

[21] H.L.Van Trees, *Detection, Estimation and Modulation Theory: Part* I.J.Wiley and Sons, Inc., 1968.

[22] A. J. Viterbi, *Error bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*, IEEE Trans. on Inf. Theory, Vol IT-13, pp. 260-269, April 1967.

[23] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill Inc., New York , N. Y., 1979.

[24] F.Xiong, A.Zerik, E.Shwedyk, *Sequential sequence estimation for channels with intersymbol interference of finite or infinite length*, IEEE Trans. on Comm., Vol.38, No. 6, June 1990, pp. 795 - 804.

[25] F. Xiong, *Sequential Sequence Estimation for Channels with Intersymbol Interference*, Ph. D. Thesis, Dept. of Elec. and Comp. Eng., University of Manitoba, 1989.

[26] A. Zerik, *Discrete Spectral Factorization in Intersymbol Interference Problems*, M. Sc. Thesis, Dept. of Elec. and Comp. Eng., University of Manitoba, 1988.

APPENDIX A

Generation of Colored Noise :

The procedure can be given by following steps.

1. Find the impulse response of the shaping filter for noise.
2. Obtain the discrete spectral factorization of the autocorrelation function of the shaping filter.
3. Obtain the difference equation to generate the noise samples.

Step 3 is described for a general channel in [26].

Consider the example given below.

Ex : Power density spectrum of noise

$$S_n(\omega) = \frac{2k\sigma_n^2}{(\omega^2 + k^2)} \Leftrightarrow K_n(\tau) = \sigma_n^2 e^{-k|\tau|} \tag{1}$$

where $K_n(\tau)$ is the autocorrelation function

A stable filter to generate this spectrum using white noise of strength σ_n^2 is

$$H(j\omega) = \frac{\sqrt{2k}}{j\omega + k} \tag{2}$$

The impulse response is

$$h(t) = \sqrt{2k}\, e^{-kt}u(t) \tag{3}$$

As a verification, the autocorrelation function of h(t) can be found. The autocorrelation function is defined as

$$R(\tau) = \int_{-\infty}^{\infty} h(t)\, h(t+\tau)\, dt \tag{4}$$

$$= 2k \int_{0}^{\infty} e^{-kt} e^{-k(t+\tau)}\, dt \quad , \tau > 0 \tag{5}$$

$$= 2k\, e^{-k\tau} \int_0^{\infty} e^{-2kt}\, dt \tag{6}$$

i.e., $$R(\tau) = e^{-k\tau} \tag{7}$$

From symmetry $$R(\tau) = R(-\tau) = e^{-k|\tau|} \tag{8}$$

Now using the D- Transform theory

$$R(D) = \sum_{m=-\infty}^{\infty} R_m D^m \tag{9}$$

$$= \frac{1}{2} + \sum_{m=1}^{\infty} \left[e^{-kT}\right]^m D^{-m} + \frac{1}{2} + \sum_{m=1}^{\infty} \left[e^{-kT}\right]^m D^m \tag{10}$$

$$= g(D^{-1}) + g(D) \tag{11}$$

$$g(D) = \frac{1}{2} + \sum_{m=1}^{\infty} \left[e^{-kT}\right]^m D^m - 1 \tag{12}$$

$$= \sum_{m=1}^{\infty} \left[e^{-kT}\right]^m D^m - \frac{1}{2} \tag{13}$$

$$= \frac{1}{1 - e^{-kT} D} - \frac{1}{2} \tag{14}$$

Therefore,

$$R(D) = \frac{1}{1 - e^{-kT} D} - \frac{1}{2} + \frac{1}{1 - e^{-kT} D^{-1}} - \frac{1}{2} \tag{15}$$

$$= \frac{1 - e^{-2kT}}{\left(1 - e^{-kT} D^{-1}\right)\left(1 - e^{-kT} D\right)} = f(D) f(D^{-1}) \tag{16}$$

This is the spectral factorization. Using f(D) as the discrete transfer function, the difference equation to generate noise samples can easily be obtained.

$$f(D) = \frac{N(D)}{W(D)} = \frac{\sqrt{1 - e^{-2kT}}}{\left(1 - e^{-kT} D\right)} \tag{17}$$

where N(D) and W(D) are the colored and white noise in discrete domain.

Hence the difference equation ;

$$n(m) = \sqrt{1 - e^{-2kT}}\ w(m) + e^{-kT}\, n(m-1) \tag{18}$$

To check the result , the autocorrelation function can be found using equation (18)

$$E\{n(m)\, n(m-j)\} = \sqrt{1 - e^{-2kT}}\, E\{w(m)\, n(m-j)\} + e^{-kT}\, E\{n(m-1)\, n(m-j)\} \tag{19}$$

Using the stationarity of noise,

$$R_j = E\{n(m)\, n(m-j)\} \quad , \qquad R_{j-1} = E\{n(m-1)\, n(m-j)\} \tag{20}$$

$$R_j = 0 \ + \ e^{-kT} R_{j-1} \tag{21}$$

$$R_j = \ e^{-kT} R_{j-1} \tag{22}$$

$$E\{n^2(m)\} = R0 = \{1 - e^{-2kT}\}\, E\{w^2(m)\} +$$

$$2\sqrt{1 - e^{-2kT}}\ e^{-kT}\, E\{w(m)n(m-1)\} + e^{-2kT}\, E\{n^2(m-1)\} \tag{23}$$

$$R_0 = \{1 - e^{-2kT}\}\, \sigma_n^2 \ + \ 0 \ + \ e^{-2kT} R_0 \tag{24}$$

$$R_0 = \sigma_n^2 \tag{25}$$

Thus the difference equation, in this case, generates noise samples according to the statistics desired.

APPENDIX B

Viterbi algorithm program listing

```
C
C
C FILENAME :COLORED,SIMULATION OF VITERBI ALGORITHM
C FOR DETECTION OF A BINARY SIGNAL SEQUENCE IN
C COLORED NOISE
C INFINITE METRIC : TRUNCATED SINGLE POLE IMPULSE RESPONSE
C
  INTEGER NR,X(2060),XF(300),IA(2),V,MLP(300),V1,
*  ER,TER,TTER,TAIL,JFIN
  REAL Z(2060),N1(2060),N2(2060),N3(2060),NF1(300),NF2(300),
*  K1,L1,I1,NF3(300),F(0:10),FM(0:10),G(0:10),H(0:10),
*  P(2),WK(2),R1(300),R2(300),R3(300),R(300),CN(0:2060),
*  HM(0:10),FF(0:10),ND1(3),ND2(3),ND3(6)
  DIMENSION DSEEDA(12)
  DOUBLE PRECISION DSEED,DSEEDA
C
  PARAMETER(L=3,SNR=12.,MKM=1,MDM=1,MAM=8,K1=1.0,T=1.,GM=2.,
*          A=2.074,B=0.5,PI=3.1416,MG=6,WC=0.6667)
  DATA DSEEDA/0.1594574556D+10,0.1208465533D+10,0.1200323639D+10,
*          0.7875096350D+09,0.1993410604D+10,0.1836865622D+10,
*          0.1173313865D+10,0.1735559417D+10,0.1314304831D+10,
*          0.2525285650D+09,0.5706309100D+08,0.5196523400D+09/
  DATA ND1/54.6,7.389,1.0/
  DATA ND2/1.649,1.0,0.606/
  DATA ND3/22038.8,2982.25,403.57,54.6,7.39,1.0/
  CALL $TRTM(TIME)
C
C CALCULATION OF SOME CONSTANTS
C
  V=L-1
  V1=MG-1
  JFIN=MKM*MDM*MAM
```

```
      L1=FLOAT(L)
      AA=(K1**2-GM**2)/(2.*GM*(B-GM))
      CC=(K1**2-B**2)/(GM**2-B**2)
      A1=A*AA*EXP((GM-B)*L)
      A2=A*CC
      A3=A*AA*(1.0-EXP((GM-B)*L))
      DO 6 I=0,V
      F(I)=A*EXP(B*I)*(EXP(-(GM+B)*I)-EXP(-(GM+B)*L))/(GM+B)
      G(I)=A*EXP(B*I)*(EXP(-2.*B*I)-EXP(-2.*B*L))/(2.*B)
      IF(I.EQ.0) GOTO 6
      FM(I)=A*EXP(-B*I)*(1.0-EXP(-(GM+B)*(L-I)))/(GM+B)
      HM(I)=FM(I)
6     CONTINUE
      DO 16 I=0,V1
      IF(I.GE.(MG-L)) GOTO  17
      H(I)=A*EXP(B*I)*(EXP(-(GM+B)*I)-EXP(-(GM+B)*(L+I)))/(GM+B)
      GOTO  16
17    H(I)=A*EXP(B*I)*(EXP(-(GM+B)*I)-EXP(-(GM+B)*MG))/(GM+B)
16    CONTINUE
      FF(0)= (A1*F(0)+A2*G(0)+A3*H(0))/2.
      DO 31 I=1,V
      FF(I)=0.5*A1*(F(I)+FM(I))+A2*G(I)+0.5*A3*(H(I)+HM(I))
31    CONTINUE
      DO 32 I=L,V1
      FF(I)=0.5*A3*H(I)
32    CONTINUE
      SGM=((A**2)/(2.*B))*(1.0-EXP(-2.*B*L))
      SGN=SGM/((1.0+(2./WC*K1))*(10.0**(0.1*SNR)))
C
C GENERATION OF SIGNAL & NOISE SEQUENCES
C
      P(1)=0.5
      P(2)=0.5
      NR=256
      NDMP=2
      IA(1)=-1
      TTER =0
      DGC=0
      DDGC=0
      DO 999 MK=1,MKM
      DSEED = DSEEDA(MK)
      TER =0
      DO 140 MD=1,MDM
```

```
      DGC=DGC+1
      NR1=2060
      NR2=2060
      CALL GGDA(DSEED,NR1,NDMP,P,IA,WK,X)
      DO 1 K=1,NR1
      IF(X(K).EQ.2) THEN
      X(K)=1
      ELSE
      X(K)=-1
      ENDIF
 1    CONTINUE
      CALL GGNML(DSEED,NR2,Z)
      DO 2 I=1,NR2
 2    Z(I)=SQRT(SGN)*Z(I)
C
C FORMING COLORED NOISE COMPONENT
C
      LF1=1
      LL1=8*NR
      CN(0)=0.0
      DO 42 I=LF1,LL1
      CN(I)=-1.489*Z(I)+EXP(-K1)*CN(I-1)
      IF(I .EQ.1) GOTO 42
      CN(I)=CN(I)+0.245*Z(I-1)
 42   CONTINUE
      DO 51 I=LF1,LL1
      N1(I)=0.0
      N2(I)=0.0
      N3(I)=0.0
      DO 52 J=1,3
      IF((I-J+1).LE.0)  GOTO 52
      N1(I)=N1(I)+CN(I-J+1)*ND1(J)
 52   CONTINUE
      N1(I)=SQRT((0.4495E-02)*(0.0183))*N1(I)
      DO 53 J=1,3
      IF((I-J+1).LE.0)  GOTO 53
      N2(I)=N2(I)+CN(I-J+1)*ND2(J)
 53   CONTINUE
      DO 54 J=1,6
      IF((I-J+1).LE.0)  GOTO 54
      N3(I)=N3(I)+CN(I-J+1)*ND3(J)
 54   CONTINUE
      N3(I)=SQRT((0.1114E-04)*(4.53E-05))*N3(I)
```

```
 51   CONTINUE
C
C
      DO 141 M=1,MAM
      DO 41 I=1,NR+V1
      NF1(I)=N1(NR*(M-1)+I)
      NF2(I)=N2(NR*(M-1)+I)
      NF3(I)=N3(NR*(M-1)+I)
      XF(I)=X(NR*(M-1)+I)
 41   CONTINUE
      DO 441 K=NR+1,NR+V1
441   XF(K)=0
C
C FORMING MATCHED FILTER O/P
C
      DO 7 LL=1,NR+V1
      R1(LL)=0.0
      R2(LL)=0.0
      R3(LL)=0.0
      DO 81 J=0,V
      R1(LL)=R1(LL)+XF(LL+J)*F(J)
 81   CONTINUE
      DO 82 J=1,V
      IF((LL-J).LE.0) GOTO 83
      R1(LL)=R1(LL)+XF(LL-J)*FM(J)
 82   CONTINUE
 83   R1(LL)=R1(LL)+NF1(LL)
      DO 84 J=0,V
      R2(LL)=R2(LL)+XF(LL+J)*G(J)
 84   CONTINUE
      DO 85 J=1,V
      IF((LL-J).LE.0) GOTO 86
      R2(LL)=R2(LL)+XF(LL-J)*G(J)
 85   CONTINUE
 86   R2(LL)=A*R2(LL)+NF2(LL)
      DO 87 J=0,V1
      R3(LL)=R3(LL)+XF(LL+J)*H(J)
 87   CONTINUE
      DO 88 J=1,V
      IF((LL-J).LE.0) GOTO 89
      R3(LL)=R3(LL)+XF(LL-J)*HM(J)
 88   CONTINUE
 89   R3(LL)=R3(LL)+NF3(LL)
```

```
      R(LL)=A1*R1(LL)+A2*R2(LL)/A+A3*R3(LL)
7     CONTINUE
C
C   ESTIMATION OF THE SEQUENCE
C
      CALL VA(R,FF,V1,NR,MLP)
      ER=0
      MC5=NR
      DO 514 I=1,MC5
      ER=ER+IABS(MLP(I)-X(NR*(M-1)+I))
      WRITE(6,*)'MLP(',I,')=',MLP(I),'X(',NR*(M-1)+I,')=',X(NR*(M-1)+I)
514   CONTINUE
      TER=TER+ER
141   CONTINUE
140   CONTINUE
      TTER=TTER+TER
999   CONTINUE
      PRE=FLOAT(TTER)/(2*NR*JFIN)
      WRITE(6,*)'OVERALL ERROR',TTER/2,'PR(E)=',PRE
      WRITE(6,*)'ALOG10(PR(E))=',ALOG10(PRE)
      WRITE(6,*)'TOTAL NO.OF BITS TXED',NR*JFIN,'K1=',K1
      WRITE(6,*)'SGN=',SGN,'SNR=',SNR,'SGM=',SGM
      CALL $TPTM(TIME)
      WRITE(6,727) TIME
727   FORMAT('','CPU TIME:',F10.3,'','SECONDS')
      STOP
      END
C
C
      SUBROUTINE VA(R,FF,V1,NR,MLP)
      REAL M(0:65),MT(0:65),R(300),MM(300,0:65),S(0:65),
     *     QQ,M0,M1,MAX,FF(0:10)
      INTEGER TAIL,P(0:65,300),MLP(300),PT(0:65,300),PP(0:65,30),
     *        PTT(0:65,300),RC,NR,V1,MLJ
C
C  INITIALIZATION
C
      DO 70 I=0,2**V1-1
      DO 70 J=1,V1
      K=I/(2**(J-1))
      PT(I,J)=MOD(K,2)
      IF(PT(I,J) .EQ.0) THEN
      PT(I,J)= -1
```

```
      ENDIF
 70   CONTINUE
      DO 75 I=0,2**V1-1
      DO 75 J=1,V1
      P(I,V1-J+1)=PT(I,J)
 75   CONTINUE
      DO 80 J=0,2**V1-1
      M(J)=0.0
      DO 80 K=1,V1
      QQ=0.0
      DO 85 I=1,K
      QQ=QQ+FF(I-1)*P(J,K-I+1)
 85   CONTINUE
      M(J)=M(J)+(R(K)-QQ)*P(J,K)
 80   CONTINUE
      DO 90 J=0,2**V1-1
      S(J)=0.0
      DO 95 I=1,V1
      S(J)=S(J)+FF(I)*P(J,V1-I+1)
 95   CONTINUE
 90   CONTINUE

C
C VITERBI ALGORITHM
C
      LF1=V1+1
      LN1=NR
      DO 100 K=LF1,LN1
      DO 951  J=0,2**V1-1
      IF (MOD(J,2) .EQ. 0) THEN
      J0=J/2
      J1=(J+2**V1)/2
      P(J,K)=-1
      ELSE
      J0=(J-1)/2
      J1=(J-1+2**V1)/2
      P(J,K)=1
      ENDIF
      M0=M(J0)+ (R(K)-S(J0)-FF(0)*P(J,K))*P(J,K)
      M1=M(J1)+ (R(K)-S(J1)-FF(0)*P(J,K))*P(J,K)
      IF (M0 .GT. M1) THEN
      MT(J)=M0
      DO 110 I=1,K-1
```

```
110 PTT(J,I)=P(J0,I)
    ELSE
    MT(J)=M1
    DO 120 I=1,K-1
120 PTT(J,I)=P(J1,I)
    ENDIF
951 CONTINUE
    DO 130 J=0,2**V1-1
    M(J)=MT(J)
    MM(K,J)=M(J)
    DO 135 I=1,K-1
    P(J,I)=PTT(J,I)
135 CONTINUE
130 CONTINUE
100 CONTINUE
C
C  DECISION MAKING: ML PATH
C
    NR1=NR
    CALL MAXM(MM,V1,NR1,MAX,MLJ)
    DO 781 I=1,NR
781 MLP(I) =P(MLJ,I)
    RETURN
    END
C
C  FIND THE MAXIMUM METRIC VALUE
C
    SUBROUTINE MAXM(MM,V1,K,MAX,MLJ)
    REAL MM(300,0:65),MAX
    INTEGER V1,K,MLJ
    MAX=MM(K,0)
    MLJ=0
    DO 940 J=0,2**V1-1
    IF(MAX .GT. MM(K,J)) GOTO 940
    MAX=MM(K,J)
    MLJ=J
940 CONTINUE
    RETURN
    END
```

APPENDIX C

Sequential algorithm program listing

```
C  FILENAME:SIMULATION OF SEQUENTIAL DETECTION OF BINARY
C  SIGNAL SEQUENCE IN THE PRESENCE OF INFINITE ISI AND
C  COLORED GAUSSIAN NOISE
C  BUTTERWORTH CHANNEL
C
   INTEGER NR,NDMP,X(2048),XF(280),XEF(256),IA(2),V,ER,TER,CL,TCL
   REAL P(2),P0(200),WK(2),Z(2160),ZF(280),Y(280),
  * F(2),G(10),AA(2),AAA(2)
   DIMENSION DSEEDA(25)
   DOUBLE PRECISION DSEED,DSEEDA
C
   DATA DSEEDA/0.1594574556D+10, 0.1208465533D+10, 0.1200323639D+10,
  *         0.7875096350D+09, 0.1993410604D+10, 0.1836865622D+10,
  *         0.1173313865D+10, 0.1735559417D+10, 0.1314304831D+10,
  *         0.2525285650D+09, 0.5706309100D+08, 0.5196523400D+09,
  *         0.2827359430D+09, 0.1943029728D+10, 0.4351107100D+08,
  *         0.1910901950D+10, 0.1893766405D+10, 0.4715504610D+09,
  *         0.2134076531D+10, 0.1814803971D+10, 0.1970428848D+10,
  *         0.4805104160D+09, 0.2093775370D+10, 0.3952421830D+09,
  *         0.2092423249D+10/
   DATA F/0.1385,0.5355/
   DATA AAA/0.3474,-0.1222/
   CALL $TRTM(TIME)
C
C   ** READING IN P0(Z)
C
   READ(8,150) L,SNR,SGN
150 FORMAT(3X,I2,5X,F6.3,5X,F8.3)
   READ(8,200) (P0(I),I=1,200)
   WRITE(6,200) (P0(I),I=1,200)
200 FORMAT(1X,5E15.6)
C
```

```
C  ** CALCULATING OF SOME CONSTANTS
C
   LD=256
   LD1=LD+1
   LF=LD+L
   V=L-1
   L1=2
   V1=L1-1
   DO 113 I=1,2
   AA(I)=AAA(I)
113 CONTINUE
   DO 117 I=1,L1
117 G(I)=F(I)
   WRITE(6,250)L,SNR,SGN,LD
250 FORMAT(1X,'L=',I2/1X,'SNR=',F10.6/
  * 1X,'SGN=',F10.6/1X,'LD=',I3/)
C
C  ** GENERATION OF SIGNAL AND NOISE SEQUENCES
C
   DSEED=DSEEDA(1)
   WRITE(6,115) DSEED
   P(1)=0.5
   P(2)=0.5
   NR=2048
   NDMP=2
   IA(1)=-1
   TER=0
   NF=NR/LD
   TCL=0
   MDM=25
   DO 140 MD=1,MDM
   CALL GGDA(DSEED,NR,NDMP,P,IA,WK,X)
   DO 1 K=1,NR
1   X(K)=X(K)-1
   CALL GGNML(DSEED,2160,Z)
   WRITE(6,115) DSEED
115 FORMAT(1X/1X,'DSEED=',D20.12)
   DO 13 I=1,2160
13 Z(I)=SQRT(SGN)*Z(I)
C
```

```
C
C FORMING { Z } SEQUENCE(THE OUTPUT OF THE WHITENING MATCHED FILTER)
C
C (1) FORMING INFORMATION AND NOISE FRAMES
C
      MF=8
      DO 3 M=1,MF
      WRITE(6,300) M
300   FORMAT(1X//1X,'M=',I2/)
      DO 4 I=1,LD
      XF(I)=X(LD*(M-1)+I)
4     ZF(I)=Z(LD*(M-1)+I)
      DO 5 I=LD1,LF
      XF(I)=0
5     ZF(I)=Z(LD*(M-1)+I)
C
C FORMING OUTPUT OF THE WHITENED MATCHED FILTER
C
      DO 301 K=1,LF
      Y(K)=0.0
      DO 302 I=1,L1
      IF((K-I+1).LE.0) GOTO 302
      Y(K)=Y(K)+F(I)*XF(K-I+1)
302   CONTINUE
      DO 303 I=1,2
      IF((K-I) .LE.0) GOTO 303
      Y(K)=Y(K)+AAA(I)*Y(K-I)
303   CONTINUE
301   CONTINUE
      DO 10 K=1,LF
10    ZF(K)=ZF(K)+Y(K)
      DO 312 K=1,LD
312   XEF(K)=0
C
C
C ESTIMATION OF THE SEQUENCE
C
      CALL MSA(ZF,XEF,G,AA,L,LD,LF,SGN,P0,CL)
      K1=(M-1)*LD+1
      K2=M*LD
```

```
      WRITE(6,130) (X(I),I=K1,K2)
130   FORMAT(/1X,'TRANSMMITED SEQUENCE:'/(1X,100I1))
      WRITE(6,110) XEF
110   FORMAT(/1X,'ESTIMATED   SEQUENCE:'/(1X,100I1))
C
C CALCULATE THE ERROR
C
      ER=0
      DO 14 I=1,LD
14    ER=ER+IABS(XEF(I)-X(K1-1+I))
      WRITE(6,125) ER
125   FORMAT(1X/1X,'ERRO=',I3)
      TER=TER+ER
      TCL=TCL+CL
3     CONTINUE
140   CONTINUE
      WRITE(6,120) TER,TCL
120   FORMAT(1X/1X,'TOTAL ERROR=',I6,5X,'TOTAL COMPUTATIONS=',I6)
      WRITE(6,*) 'PRE=',FLOAT(TER)/(LD*MF*MDM)
      WRITE(6,*) 'LOG(PRE)=',ALOG10(FLOAT(TER)/(LD*MF*MDM))
      CALL $TPTM(TIME)
      WRITE(6,727) TIME
727   FORMAT('','CPU TIME:',F10.3,'','SECONDS')
      STOP
      END

C
C
      SUBROUTINE MSA(ZF,XEF,G,AA,L,LD,LF,SGN,P0,CL)
      INTEGER*2 S(3000,284),S1(100,284),S2(100,284),S3(100,284),
     * S4(100,284),
     * S5(100,284),S6(100,284),S7(100,284),S8(100,284),S9(100,284),
     * S10(100,284),S11(100,284),S12(100,284),S13(100,284),S14(100,284),
     * S15(100,284)
      REAL ZF(LF),G(10),AA(2),P0(200),SS(3000,5),AM,M0,M1,ME,
     * SS1(100,5),SS2(100,5),SS3(100,5),SS4(100,5),SS5(100,5),
     * SS7(100,5),SS8(100,5),SS9(100,5),SS10(100,5),SS11(100,5),
     * SS13(100,5),SS14(100,5),SS15(100,5),SS12(100,5),SS6(100,5)
      INTEGER*2 LPP(5000),A(282)
      INTEGER   CL,T,V,SH(16)
```

```
      INTEGER XEF(LD)
C
C DEFINING METRIC FUNCTION
C
      ME(Z,Y,SGN,P0Z)=(-1.442695*(0.9189385+0.5*ALOG(SGN)+
     * ((Z-Y)**2)/(2*SGN)+ALOG(P0Z))-1.0)
C
C INITIALIZATION
C
      AM=-10000.0
      DO 4 K=1,16
 4    SH(K)=K-1
      DO 7444 J=1,5
      DO 7442 I=1,3000
 7442 SS(I,J)=0.0
      DO 7444 I=1,100
      SS1(I,J)=0.0
      SS2(I,J)=0.0
      SS3(I,J)=0.0
      SS4(I,J)=0.0
      SS5(I,J)=0.0
      SS6(I,J)=0.0
      SS7(I,J)=0.0
      SS8(I,J)=0.0
      SS9(I,J)=0.0
      SS10(I,J)=0.0
      SS11(I,J)=0.0
      SS12(I,J)=0.0
      SS13(I,J)=0.0
      SS14(I,J)=0.0
 7444 SS15(I,J)=0.0
      DO 7 J=1,284
      DO 2 I=1,3000
 2    S(I,J)=0
      DO 7 I=1,100
      S1(I,J)=0
      S2(I,J)=0
      S3(I,J)=0
      S4(I,J)=0
      S5(I,J)=0
```

```
      S6(I,J)=0
      S7(I,J)=0
      S8(I,J)=0
      S9(I,J)=0
      S10(I,J)=0
      S11(I,J)=0
      S12(I,J)=0
      S13(I,J)=0
      S14(I,J)=0
7     S15(I,J)=0
      Y0=0.
      Y1=G(1)
      S(1,281)=1
      S(2,281)=1
      ZK=10*(ZF(1)+4.1)
      KZ1=INT(ZK)
      KZ2=KZ1+1
      P0Z=(P0(KZ2)-P0(KZ1))*(ZK-KZ1)+P0(KZ1)
      M0=ME(ZF(1),Y0,SGN,P0Z)
      M1=ME(ZF(1),Y1,SGN,P0Z)
      IF(M0-M1) 1,1,3
1     S(1,1)=1
      SS(1,1)=M1
      SS(1,2)=Y1
      S(2,1)=0
      SS(2,1)=M0
      SS(2,2)=Y0
      GOTO 30
3     S(1,1)=0
      SS(1,1)=M0
      SS(1,2)=Y0
      S(2,1)=1
      SS(2,1)=M1
      SS(2,2)=Y1
C
C   METRICS OF SUCCESSORS OF TOP PATH
C
30    NS=2
      LPP(1)=1
      V=L-1
```

```
      M=100
      T=3
      CL=1
10    ASSIGN 40 TO KS
      CALL HOS(ZF,S,SS,G,AA,LPP,T,3000,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(1))
      IF(NC) 115,115,118
C
118   DO 100 I=1,T
      DO 6100 J=1,5
6100  SS1(I,J)=SS(I,J)
      DO 100 J=1,284
100   S1(I,J)=S(I,J)
      NS=T
11    ASSIGN 41 TO KS
      CALL HOS(ZF,S1,SS1,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(2))
      IF(NC) 115,115,218
C
218   DO 200 I=1,T
      DO 6200 J=1,5
6200  SS2(I,J)=SS1(I,J)
      DO 200 J=1,284
200   S2(I,J)=S1(I,J)
      NS=T
12    ASSIGN 42 TO KS
      CALL HOS(ZF,S2,SS2,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(3))
      IF(NC) 115,115,318
C
318   DO 300 I=1,T
      DO 6300 J=1,5
6300  SS3(I,J)=SS2(I,J)
      DO 300 J=1,284
300   S3(I,J)=S2(I,J)
      NS=T
13    ASSIGN 43 TO KS
      CALL HOS(ZF,S3,SS3,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(4))
      IF(NC) 115,115,418
C
418   DO 400 I=1,T
      DO 6400 J=1,5
6400  SS4(I,J)=SS3(I,J)
```

```
      DO 400 J=1,284
 400  S4(I,J)=S3(I,J)
      NS=T
 14   ASSIGN 44 TO KS
      CALL HOS(ZF,S4,SS4,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(5))
      IF(NC) 115,115,518
C
 518  DO 500 I=1,T
      DO 6500 J=1,5
6500  SS5(I,J)=SS4(I,J)
      DO 500 J=1,284
 500  S5(I,J)=S4(I,J)
      NS=T
 15   ASSIGN 45 TO KS
      CALL HOS(ZF,S5,SS5,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(6))
      IF(NC) 115,115,618
C
 618  DO 600 I=1,T
      DO 6600 J=1,5
6600  SS6(I,J)=SS5(I,J)
      DO 600 J=1,284
 600  S6(I,J)=S5(I,J)
      NS=T
 16   ASSIGN 46 TO KS
      CALL HOS(ZF,S6,SS6,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(7))
      IF(NC) 115,115,718
C
 718  DO 700 I=1,T
      DO 6700 J=1,5
6700  SS7(I,J)=SS6(I,J)
      DO 700 J=1,284
 700  S7(I,J)=S6(I,J)
      NS=T
 17   ASSIGN 47 TO KS
      CALL HOS(ZF,S7,SS7,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(8))
      IF(NC) 115,115,818
C
 818  DO 800 I=1,T
      DO 6800 J=1,5
6800  SS8(I,J)=SS7(I,J)
```

```
      DO 800 J=1,284
800   S8(I,J)=S7(I,J)
      NS=T
18    ASSIGN 48 TO KS
      CALL HOS(ZF,S8,SS8,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(9))
      IF(NC) 115,115,918
C
918   DO 900 I=1,T
      DO 6900 J=1,5
6900  SS9(I,J)=SS8(I,J)
      DO 900 J=1,284
900   S9(I,J)=S8(I,J)
      NS=T
19    ASSIGN 49 TO KS
      CALL HOS(ZF,S9,SS9,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(10))
      IF(NC) 115,115,1018
C
1018  DO 1000 I=1,T
      DO 7000 J=1,5
7000  SS10(I,J)=SS9(I,J)
      DO 1000 J=1,284
1000  S10(I,J)=S9(I,J)
      NS=T
20    ASSIGN 50 TO KS
      CALL HOS(ZF,S10,SS10,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(11))
      IF(NC) 115,115,1118
C
1118  DO 1100 I=1,T
      DO 7100 J=1,5
7100  SS11(I,J)=SS10(I,J)
      DO 1100 J=1,284
1100  S11(I,J)=S10(I,J)
      NS=T
21    ASSIGN 51 TO KS
      CALL HOS(ZF,S11,SS11,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(12))
      IF(NC) 115,115,1218
C
1218  DO 1200 I=1,T
      DO 7200 J=1,5
7200  SS12(I,J)=SS11(I,J)
```

```
      DO 1200 J=1,284
1200  S12(I,J)=S11(I,J)
      NS=T
22    ASSIGN 52 TO KS
      CALL HOS(ZF,S12,SS12,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(13))
      IF(NC) 115,115,1318
C
1318  DO 1300 I=1,T
      DO 7300 J=1,5
7300  SS13(I,J)=SS12(I,J)
      DO 1300 J=1,284
1300  S13(I,J)=S12(I,J)
      NS=T
23    ASSIGN 53 TO KS
      CALL HOS(ZF,S13,SS13,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(14))
      IF(NC) 115,115,1418
C
1418  DO 1400 I=1,T
      DO 7400 J=1,5
7400  SS14(I,J)=SS13(I,J)
      DO 1400 J=1,284
1400  S14(I,J)=S13(I,J)
      NS=T
24    ASSIGN 54 TO KS
      CALL HOS(ZF,S14,SS14,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(15))
      IF(NC) 115,115,1518
C
1518  DO 1500 I=1,T
      DO 7500 J=1,5
7500  SS15(I,J)=SS14(I,J)
      DO 1500 J=1,284
1500  S15(I,J)=S14(I,J)
      NS=T
25    ASSIGN 55 TO KS
      CALL HOS(ZF,S15,SS15,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH(16))
      IF(NC) 115,115,1618
C
C     HAS COME OUT FROM A STACK. EXAME WHICH STACK IT HAS COME OUT
C     FROM AND GO TO FINAL DECISION IF THE STACK IS THE FIRST STACK,
C     OR MAKE A TENTATIVE DECISION AND RETURN TO PREVIOUS STACK TO
```

```
C   CONTINUE SEARCHING.
C
C
1618 IF(AM.GT.-10000.0) GOTO 116
     WRITE(6,9) AM
9    FORMAT(1X/1X,'STACKS OVERFLOW'/1X,'AM=',F20.8)
     DO 92 J=1,282
92   A(J)=S15(1,J)
     AM=SS15(1,1)
     GOTO 116
C
115  IF(CL.EQ.5000) GOTO 116
     GOTO KS, (40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55)
40   IF(AM.GT.SS(1,1)) GOTO 116
     DO 401 J=1,282
401  A(J)=S(1,J)
     AM=SS(1,1)
     GOTO 116
C
41   IF(AM.GT.SS1(1,1)) GOTO 412
     DO 411 J=1,282
411  A(J)=S1(1,J)
     AM=SS1(1,1)
412  DO 413 I=1,100
     DO 4413 J=1,5
4413 SS1(I,J) =0.0
     DO 413 J=1,284
413  S1(I,J)=0.0
     DO 414 J=1,284
     DO 415 I=1,2997
415  S(I,J)=S(I+T,J)
     DO 414 I=2998,3000
414  S(I,J)=0.0
     DO 4414 J=1,5
     DO 4415 I=1,2997
4415 SS(I,J)=SS(I+T,J)
     DO 4414 I=2998,3000
4414 SS(I,J)=0.0
     NS=2997
     GOTO 10
```

```
C
42   IF(AM.GT.SS2(1,1)) GOTO 422
     DO 421 J=1,282
421  A(J)=S2(1,J)
     AM=SS2(1,1)
422  DO 423 I=1,100
     DO 4423 J=1,5
4423 SS2(I,J) =0.0
     DO 423 J=1,284
423  S2(I,J)=0.0
     DO 424 J=1,284
     DO 425 I=1,97
425  S1(I,J)=S1(I+T,J)
     DO 424 I=98,100
424  S1(I,J)=0.0
     DO 4424 J=1,5
     DO 4425 I=1,97
4425 SS1(I,J)=SS1(I+T,J)
     DO 4424 I=98,100
4424 SS1(I,J)=0.0
     NS=97
     GOTO 11
C
43   IF(AM.GT.SS3(1,1)) GOTO 432
     DO 431 J=1,282
431  A(J)=S3(1,J)
     AM=SS3(1,1)
432  DO 433 I=1,100
     DO 4433 J=1,5
4433 SS3(I,J)=0.0
     DO 433 J=1,284
433  S3(I,J)=0.0
     DO 434 J=1,284
     DO 435 I=1,97
435  S2(I,J)=S2(I+T,J)
     DO 434 I=98,100
434  S2(I,J)=0.0
     DO 4434 J=1,5
     DO 4435 I=1,97
4435 SS2(I,J)=SS2(I+T,J)
```

```
      DO 4434 I=98,100
4434  SS2(I,J)=0.0
      NS=97
      GOTO 12
C
44    IF(AM.GT.SS4(1,1)) GOTO 442
      DO 441 J=1,282
441   A(J)=S4(1,J)
      AM=SS4(1,1)
442   DO 443 I=1,100
      DO 4443 J=1,5
4443  SS4(I,J)=0.0
      DO 443 J=1,284
443   S4(I,J)=0.0
      DO 444 J=1,284
      DO 445 I=1,97
445   S3(I,J)=S3(I+T,J)
      DO 444 I=98,100
444   S3(I,J)=0.0
      DO 4444 J=1,5
      DO 4445 I=1,97
4445  SS3(I,J)=SS3(I+T,J)
      DO 4444 I=98,100
4444  SS3(I,J)=0.0
      NS=97
      GOTO 13
C
45    IF(AM.GT.SS5(1,1)) GOTO 452
      DO 451 J=1,282
451   A(J)=S5(1,J)
      AM=SS5(1,1)
452   DO 453 I=1,100
      DO 4453 J=1,5
4453  SS5(I,J)=0.0
      DO 453 J=1,284
453   S5(I,J)=0.0
      DO 454 J=1,284
      DO 455 I=1,97
455   S4(I,J)=S4(I+T,J)
      DO 454 I=98,100
```

```
454   S4(I,J)=0.0
      DO 4454 J=1,5
      DO 4455 I=1,97
4455  SS4(I,J)=SS4(I+T,J)
      DO 4454 I=98,100
4454  SS4(I,J)=0.0
      NS=97
      GOTO 14
C
46    IF(AM.GT.SS6(1,1)) GOTO 462
      DO 461 J=1,282
461   A(J)=S6(1,J)
      AM=SS6(1,1)
462   DO 463 I=1,100
      DO 4463 J=1,5
4463  SS6(I,J)=0.0
      DO 463 J=1,284
463   S6(I,J)=0.0
      DO 464 J=1,284
      DO 465 I=1,97
465   S5(I,J)=S5(I+T,J)
      DO 464 I=98,100
464   S5(I,J)=0.0
      DO 4464 J=1,5
      DO 4465 I=1,97
4465  SS5(I,J)=SS5(I+T,J)
      DO 4464 I=98,100
4464  SS5(I,J)=0.0
      NS=97
      GOTO 15
C
47    IF(AM.GT.SS7(1,1)) GOTO 472
      DO 471 J=1,282
471   A(J)=S7(1,J)
      AM=SS7(1,1)
472   DO 473 I=1,100
      DO 4473 J=1,5
4473  SS7(I,J)=0.0
      DO 473 J=1,284
473   S7(I,J)=0.0
```

```
      DO 474 J=1,284
      DO 475 I=1,97
475   S6(I,J)=S6(I+T,J)
      DO 474 I=98,100
474   S6(I,J)=0.0
      DO 4474 J=1,5
      DO 4475 I=1,97
4475  SS6(I,J)=SS6(I+T,J)
      DO 4474 I=98,100
4474  SS6(I,J)=0.0
      NS=97
      GOTO 16
C
48    IF(AM.GT.SS8(1,1)) GOTO 482
      DO 481 J=1,282
481   A(J)=S8(1,J)
      AM=SS8(1,1)
482   DO 483 I=1,100
      DO 4483 J=1,5
4483  SS8(I,J)=0.0
      DO 483 J=1,284
483   S8(I,J)=0.0
      DO 484 J=1,284
      DO 485 I=1,97
485   S7(I,J)=S7(I+T,J)
      DO 484 I=98,100
484   S7(I,J)=0.0
      DO 4484 J=1,5
      DO 4485 I=1,97
4485  SS7(I,J)=SS7(I+T,J)
      DO 4484 I=98,100
4484  SS7(I,J)=0.0
      NS=97
      GOTO 17
C
49    IF(AM.GT.SS9(1,1)) GOTO 492
      DO 491 J=1,282
491   A(J)=S9(1,J)
      AM=SS9(1,1)
492   DO 493 I=1,100
```

```
      DO 4493 J=1,5
4493 SS9(I,J)=0.0
      DO 493  J=1,284
493   S9(I,J)=0.0
      DO 494  J=1,284
      DO 495 I=1,97
495   S8(I,J)=S8(I+T,J)
      DO 494 I=98,100
494   S8(I,J)=0.0
      DO 4494 J=1,5
      DO 4495 I=1,97
4495 SS8(I,J)=SS8(I+T,J)
      DO 4494 I=98,100
4494 SS8(I,J)=0.0
      NS=97
      GOTO 18
C
50    IF(AM.GT.SS10(1,1)) GOTO 502
      DO 501 J=1,282
501   A(J)=S10(1,J)
      AM=SS10(1,1)
502   DO 503 I=1,100
      DO 4503 J=1,5
4503 SS10(I,J)=0.0
      DO 503  J=1,284
503   S10(I,J)=0.0
      DO 504  J=1,284
      DO 505 I=1,97
505   S9(I,J)=S9(I+T,J)
      DO 504 I=98,100
504   S9(I,J)=0.0
      DO 4504 J=1,5
      DO 4505 I=1,97
4505 SS9(I,J)=SS9(I+T,J)
      DO 4504 I=98,100
4504 SS9(I,J)=0.0
      NS=97
      GOTO 19
C
51    IF(AM.GT.SS11(1,1)) GOTO 512
```

```
      DO 511 J=1,282
511   A(J)=S11(1,J)
      AM=SS11(1,1)
512   DO 513 I=1,100
      DO 4513 J=1,5
4513  SS11(I,J)=0.0
      DO 513  J=1,284
513   S11(I,J)=0.0
      DO 514  J=1,284
      DO 515 I=1,97
515   S10(I,J)=S10(I+T,J)
      DO 514 I=98,100
514   S10(I,J)=0.0
      DO 4514 J=1,5
      DO 4515 I=1,97
4515  SS10(I,J)=SS10(I+T,J)
      DO 4514 I=98,100
4514  SS10(I,J)=0.0
      NS=97
      GOTO 20
C
52    IF(AM.GT.SS12(1,1)) GOTO 522
      DO 521 J=1,282
521   A(J)=S12(1,J)
      AM=SS12(1,1)
522   DO 523 I=1,100
      DO 4523 J=1,5
4523  SS12(I,J)=0.0
      DO 523  J=1,284
523   S12(I,J)=0.0
      DO 524  J=1,284
      DO 525 I=1,97
525   S11(I,J)=S11(I+T,J)
      DO 524 I=98,100
524   S11(I,J)=0.0
      DO 4524 J=1,5
      DO 4525 I=1,97
4525  SS11(I,J)=SS11(I+T,J)
      DO 4524 I=98,100
4524  SS11(I,J)=0.0
```

```
      NS=97
      GOTO 21
C
53    IF(AM.GT.SS13(1,1)) GOTO 532
      DO 531 J=1,282
531   A(J)=S13(1,J)
      AM=SS13(1,1)
532   DO 533 I=1,100
      DO 4533 J=1,5
4533  SS13(I,J)=0.0
      DO 533  J=1,284
533   S13(I,J)=0.0
      DO 534  J=1,284
      DO 535 I=1,97
535   S12(I,J)=S12(I+T,J)
      DO 534 I=98,100
534   S12(I,J)=0.0
      DO 4534 J=1,5
      DO 4535 I=1,97
4535  S12(I,J)=SS12(I+T,J)
      DO 4534 I=98,100
4534  SS12(I,J)=0.0
      NS=97
      GOTO 22
C
54    IF(AM.GT.SS14(1,1)) GOTO 542
      DO 541 J=1,282
541   A(J)=S14(1,J)
      AM=SS14(1,1)
542   DO 543 I=1,100
      DO 5543 J=1,5
5543  SS14(I,J)=0.0
      DO 543  J=1,284
543   S1 4(I,J)=0.0
      DO 544  J=1,284
      DO 545 I=1,97
545   S13(I,J)=S13(I+T,J)
      DO 544 I=98,100
544   S13(I,J)=0.0
      DO 4544 J=1,5
```

```
      DO 4545 I=1,97
4545 SS13(I,J)=SS13(I+T,J)
      DO 4544 I=98,100
4544 SS13(I,J)=0.0
      NS=97
      GOTO 23
  C
55   IF(AM.GT.SS15(1,1)) GOTO 552
      DO 551 J=1,282
551   A(J)=S15(1,J)
      AM=SS15(1,1)
552   DO 553 I=1,100
      DO 5553 J=1,5
5553 SS15(I,J)=0.0
      DO 553  J=1,284
553   S15(I,J)=0.0
      DO 554  J=1,284
      DO 555 I=1,97
555   S14(I,J)=S14(I+T,J)
      DO 554 I=98,100
554   S14(I,J)=0.0
      DO 4554 J=1,5
      DO 4555 I=1,97
4555 SS14(I,J)=SS14(I+T,J)
      DO 4554 I=98,100
4554 SS14(I,J)=0.0
      NS=97
      GOTO 24
116   DO 170 J=1,LD
170   XEF(J)=A(J)
      WRITE(6,999) CL,A(281),AM
999   FORMAT(1X,'CL=',I4,1X,'LP=',I4,1X,'METRIC=',F20.8)
      RETURN
      END
 C
 C
   SUBROUTINE HOS(ZF,S,SS,G,AA,LPP,T,M,L,LD,LF,V,SGN,P0,CL,NS,NC,SH)
   DIMENSION ZF(LF),P0(200)
   INTEGER T,CL,V,CL0,NS1,XEF(256),SH
   INTEGER*2 S(M,284),LPP(5000),S1271
```

```
      REAL G(10),AA(2),SS(M,5),ME,M0,M1,SGN
      ME(Z,Y,SGN,P0Z)=(-1.442695*(0.9189385+0.5*ALOG(SGN)+
     * ((Z-Y)**2)/(2*SGN)+ALOG(P0Z))-1.0)
C
      L1=2
      V1=L1-1
      S1281=S(1,281)
      CL0=CL
      IF(S(1,281).EQ.LF) GOTO 115
      NS1=NS+1
      DO 110 K=NS1,3000
      LP=S(1,281)+1
      Y0=AA(1)*SS(1,2)+AA(2)*SS(1,3)
      IF(LP .GT.V1) GOTO 125
      DO 120 J=2,LP
      LJ=LP-J+1
120   Y0=Y0+G(J)*S(1,LJ)
      Y1=Y0+G(1)
      GOTO 140
125   DO 130 J=2,L1
      LJ=LP-J+1
130   Y0=Y0+G(J)*S(1,LJ)
      Y1=Y0+G(1)
140   ZK=10*(ZF(LP)+4.1)
      KZ1=INT(ZK)
      KZ2=KZ1+1
      P0Z=(P0(KZ2)- P0(KZ1))*(ZK- KZ1)+P0(KZ1)
      M0=ME(ZF(LP),Y0,SGN,P0Z)+SS(1,1)
      M1=ME(ZF(LP),Y1,SGN,P0Z)+SS(1,1)
C
C PLACING TWO NEWEST PATHES INTO STACK, ONE REPLACES THE ENTRY 1,
C THE OTHER ENTERS ENTRY NS+1, REODERING IS TO BE DONE LATTER
C
C  ADDING THE NEWEST ESTIMATED BIT AND CHANGING THE 2 PARAMETERS
C  (LENGTH AND METRIC OF THE PATH) OF ENTRY 1 AND NS+1 OF THE STACK
C
      S(1,LP)=0
      S(1,281)=LP
      SS(1,1)=M0
      SS(1,3)=SS(1,2)
```

```
      SS(1,2)=Y0
      IF(LP.GT.256) GOTO  145
      NS=NS+1
      DO 160 J=1,LP-1
160   S(NS,J)=S(1,J)
      S(NS,281)=LP
      S(NS,LP)=1
      SS(NS,1)=M1
      SS(NS,3)=SS(1,2)
      SS(NS,2)=Y1
C
C  REORDERING THE STACK BY COMPARING THEIR METRICS AND PLACING THE
C  LARGEST T ENTRIES IN THE TOP OF THE STACK
C
145   CALL REODR(S,T,M,NS,SS)
C
      CL=CL+1
      LPP(CL)=S(1,281)
      IF(LPP(CL).EQ.LF.OR.CL.EQ.5000) GOTO 115
      IF(NS.EQ.M) GOTO 112
110   CONTINUE
112   NC=1
      GOTO 113
115   NC=0
      LPP(CL)=S(1,281)
113   IF((CL-CL0).GT.20) CL0=CL-19
      RETURN
      END
C
C
C
      SUBROUTINE REODR(S,T,M,NS,SS)
      INTEGER T,T1
      INTEGER*2 S(M,284),C(284)
      REAL SS(M,5),CC(10),AM
      T1=T+1
      IF(T1.GT.NS) T1=NS
      DO 10 K=1,T1
      AM=-10000.0
      DO 2 I=K,NS
```

```
      IF(SS(I,1).GT.AM) AM=SS(I,1)
 2    CONTINUE
      DO 3 I=K,NS
      IF(SS(I,1).EQ.AM) NM=I
 3    CONTINUE
      IF(NM.EQ.K) GOTO 10
      DO 5 J=1,5
      CC(J)=SS(K,J)
      SS(K,J)=SS(NM,J)
 5    SS(NM,J)=CC(J)
      DO 4 J=1,284
      C(J)=S(K,J)
      S(K,J)=S(NM,J)
 4    S(NM,J)=C(J)
 10   CONTINUE
      RETURN
      END
```