Sequential Sequence Estimation for Channels with Intersymbol Interference

by

Fuqin Xiong

A thesis
Presented to the University of Manitoba
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy
in
the Department of Electrical Engineering

Winnipeg, Manitoba

May 1989

ISBN 0-315-54837-1

Canada

SEQUENTIAL SEQUENCE ESTIMATION FOR CHANNELS WITH INTERSYMBOL INTERFERENCE

BY

FUQIN XIONG

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

DOCTOR OF PHILOSOPHY

ABSTRACT

Intersymbol interference (ISI) occurs when a digital sequence is transmitted at a high rate over a bandlimited communication channel. Techniques for combating ISI have been investigated actively for many years. Among them the maximum likelihood sequence estimation using the Viterbi algorithm (MLSE-VA) has superior detection performance. However the computation and storage complexity of the Viterbi algorithm grow exponentially with the length of the channel memory, hence prevent it from real-time implementation for channels with memory of long, possibly infinite, length.

Sequential sequence estimation for channels with finite ISI and infinite ISI is studied in this thesis. The estimator consists of a whitened matched filter followed by a sequential decoder. A key component of a sequential decoder is a proper metric that accounts for the unequal search depth. This metric is derived for the ISI channel. Analysis shows that the error probability of the sequential estimator is asymptotically maximum likelihood and its computational complexity is essentially independent of the channel memory. Thus this technique greatly reduces the computational complexity of MLSE-VA while maintaining a detection performance which is essentially the same as that of MLSE-VA. The important behavior of the sequential algorithm-computational distribution is also analyzed and the important parameter R_{comp}, computational cutoff rate, is related to the channel impulse response.

Extensive computer simulations were conducted for the evaluation of error probability and computational distribution. Comparisons between analytical predictions and simulation results show good agreement.

Extensions to multiple channel systems with both ISI and ICI (inter-channel interference), and to channels with coded input sequences, are also studied in this thesis.

ACKNOWLEDGMENTS

The author wishes to express his sincere thanks to Professor Ed. Shwedyk for his supervision and financial support throughout the Ph.D program and during the work on this thesis.

Grateful acknowledgement is also due the University of Manitoba for the UM Fellowship and Tsinghua University of China for its financial support.

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.

Fuqin Xiong

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.

Fuqin Xiong

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

Digital data communications play an increasingly important role in this era of information explosion. The widespread use of computers and other business machines involves huge amount of data exchange and constitutes a major area of data communications. Digital techniques have also rapidly extended into conventional analogue signal communications. Facing increasing transmission rate and accuracy required by digital communications, man has resorted to not only building more and better communication facilities, but also to making more efficient use of existing communication equipment. Using voice grade telephone lines to transmit high-rate digital symbols and recording data more densely on a magnetic tape are typical examples.

However, when transmitting high-speed data streams over band-limited channels, intersymbol interference (ISI) occurs. The term ISI is used to describe the phenomenon that the received pulse is disturbed by the overlapping in time of the adjacent pulses. In a high signal to noise ratio (SNR) channel

such as a telephone channel, where the SNR is usually above 28 dB [1], the ISI rather than Gaussian noise becomes the predominant degrading factor. Research on combatting ISI dates back to Nyquist. As background, a brief review of techniques for combatting ISI is given below.

1.1 PAM Model and Intersymbol Interference

ISI arises in all pulse-modulation systems, including pulse amplitude modulation (PAM), frequency-shift keying (FSK), phase-shift keying (PSK) and quadrature amplitude modulation (QAM). Its effect, however, can be most easily studied with a baseband PAM system. Fig.1.1 (a) shows a generic passband PAM system. The input data stream is modeled as an infinitely long sequence of impulses spaced at T second intervals and with different weight x_k, where the x_k are drawn from a finite alphabet {0,1,...,m-1}. The impulse response of the transmitting filter determines the shape of the baseband pulses at its output. Usually the transmitting filter is designed so that its output baseband pulses have no intersymbol interference. The baseband pulses are used to modulate a high frequency carrier $cos\omega_0 t$ in order to be transmitted efficiently over a particular physical channel. After modulation the signal spectrum is centred at ω_0. The passband channel allows the frequency components in the vicinity of ω_0 to pass and attenuates the spectrum outside the bandwidth. The channel introduces intersymbol inter-

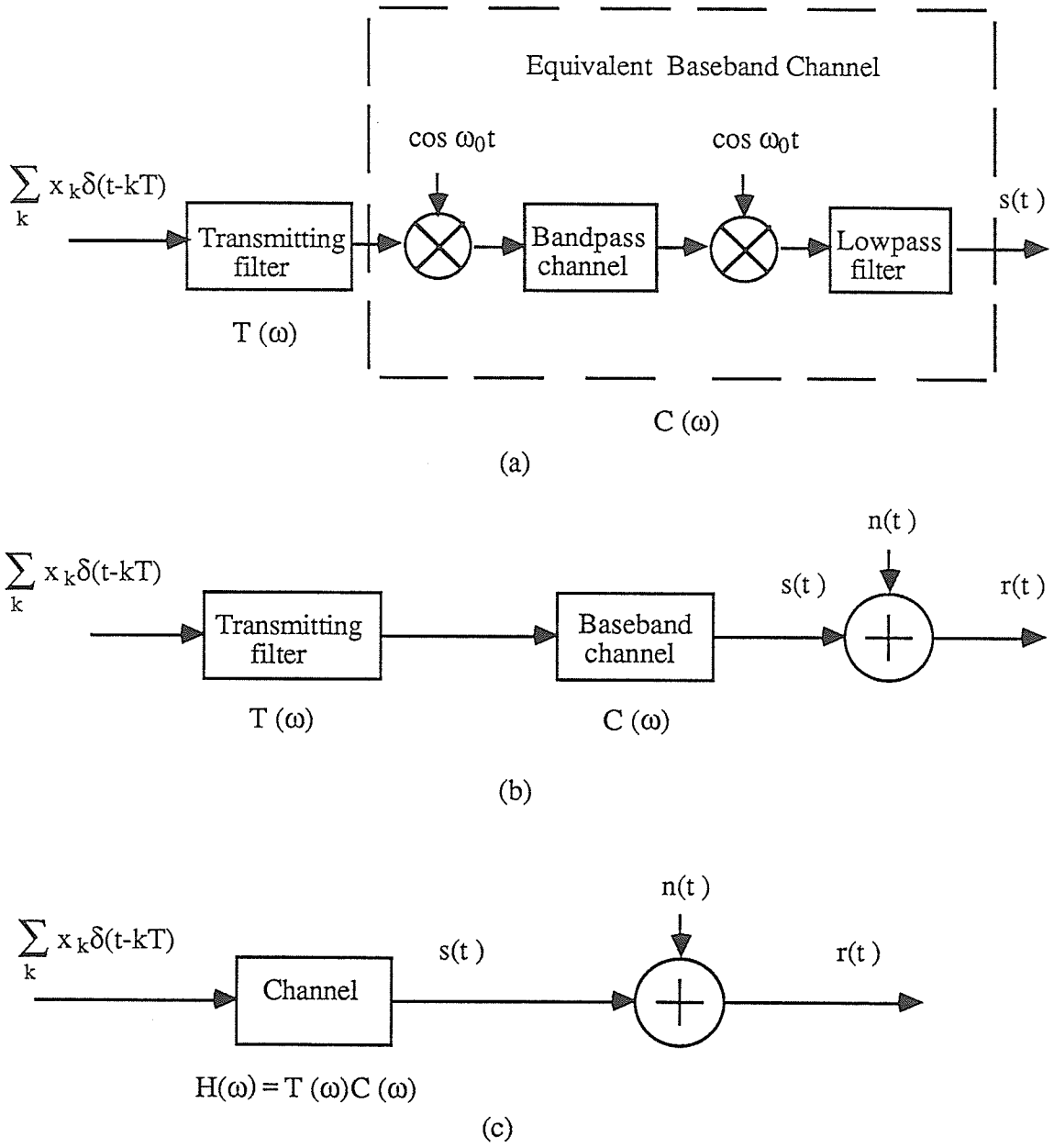

Figure 1.1: PAM system: (a) passband PAM system, (b) equivalent baseband PAM system, (c) Transmitting filter and baseband channel are lumped as 'channel'.

ference whose extent depends upon the channel bandwidth. At the receiver the passband signal is demodulated coherently by multiplying with $cos\omega_0 t$ and lowpass filtering.

For the purpose of ISI analysis, a passband PAM system can be represented by an equivalent baseband system, as shown in Fig. 1.1(b) and further simplified as shown in Fig. 1.1(c). An additive white Gaussian noise $n(t)$ is included to reflect the cumulative effect of the noise sources which enter the system at various points. The transmitting filter and the baseband channel impulse responses are lumped as $h(t) = \mathcal{F}^{-1}[H(\omega)]$, where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. Fig. 1.2 shows an example of $h(t)$, where $h(t)$ has nonzero values at sampling points $t_0 \pm kT$, $k = 1, 2, \ldots$. These are ISI terms. The number of ISI terms (or alternatively called the length of ISI) could be finite or infinite. When it is infinite, however, the number of those ISI terms which have significant value usually is finite for practical channels. But the finite lengths may vary widely. For instance, the length of ISI of a system with duobinary signaling is 2, and the length of ISI of a medium-range (180 to 725 mile) telephone channel of the switched telecommunication network is 20 to 30 transmission periods [2].

The received signal corrupted by ISI and noise is

$$r(t) = \sum_k x_k h(t - kT) + n(t). \tag{1.1}$$

If $r(t)$ is sampled at $t_0 + jT$, where t_0 accounts for the channel delay and

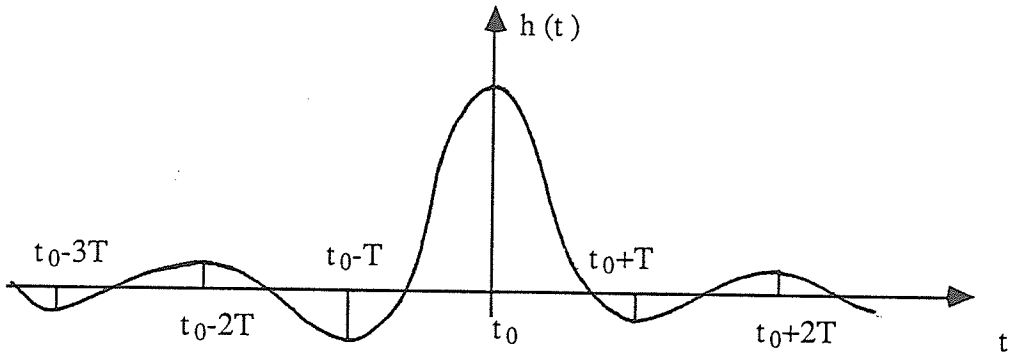

Figure 1.2: An example of h(t) with ISI

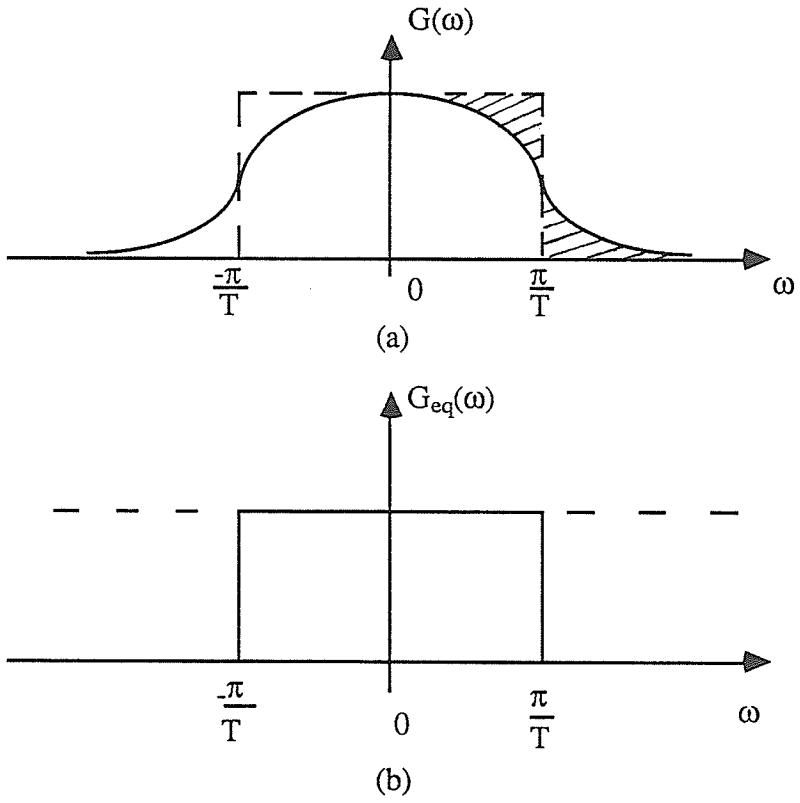

Figure 1.3: Baseband spectrum shaping for zero ISI: (a) A spectrum with odd symmetry, (b) Overlapped spectrum is flat: Nyquist criterion is satisfied.

sampler phase, then

$$r(t_0 + jT) = x_j h(t_0) + \sum_{k \neq j} x_k h(t_0 + jT - kT) + n(t_0 + jT). \qquad (1.2)$$

The first term on the right side is the desired signal produced by the input symbol at $t = jT$ and the second term is the interference from adjacent symbols. This term is nonzero unless $h(t)$ is zero at all the sampling instants except at t_0.

Since the total ISI component of the received signal in a transmission period T is the summation of a number of ISI terms due to adjacent input symbols and the input symbols are random variables, ISI is also a random variable. The probability distribution of the ISI, especially the limiting distribution when $h(t)$ is infinitely long, is important in determining the error probability of the receiver. One might think that the central-limit theorem applies to this summation of large or infinite number of terms and that the distribution is Gaussian. However, it has been shown that the limiting probability density function might not even exist. Even if it does exist it will not be Gaussian if $h(t)$ decays faster than 1/t, since the summation is bounded. For any practical channel impulse response the asymptotic decay must be exponential [3].

1.2 Baseband Spectrum Shaping for Zero ISI

This technique was first introduced by Nyquist ([4],1928) and is described in many texts ([3],Lucky,1968; [5],Schwartz,1980). Denote the overall impulse

response of the cascade of the PAM transmitter,channel and receiver as g(t). To achieve complete suppression of the ISI at any sampling instant, a necessary and sufficient condition is that the frequency spectrum of the system, namely $G(\omega) = \mathcal{F}[g(t)]$, satisfies the following Nyquist criterion,

$$G_{eq}(\omega) = \sum_k G(\omega + \frac{2k\pi}{T}) = constant, |\omega| \leq \frac{\pi}{T}, \tag{1.3}$$

where $G_{eq}(\omega)$ is called the equivalent Nyquist bandwidth characteristic. It is flat within the frequency band defined by the sampling rate. Any $G(\omega)$ which satisfies (1.3) is called a Nyquist characteristic, an example of which is shown in Fig. 1.3. When the bandwidth of $G(\omega)$ is less than π/T it is impossible to satisfy (1.3). The only $G(\omega)$ which satisfies (1.3) and with a bandwidth equal to π/T is the rectangular (ideal) characteristic, which is extremely sensitive to perturbations in rate, timing, or channel characteristics. One class of Nyquist characteristics which is less sensitive to these perturbations is the raised-cosine characteristic. It is in fact just one example of a even larger class of spectra which satisfies (1.3)-the spectra with odd symmetry about $\omega_c = \pi/T$ in amplitude and with zero (or linear) phase shift [5]. However their bandwidths are larger than π/T. Generally speaking, the larger the bandwidth of $G(\omega)$, the less the sensitivity to perturbations.

Schidman ([6],1967) generalized Nyquist's work to the case of multiple waveforms transmitted in a single channel. An optimum linear receiver was found using a minimum mean squared error criterion under a generalized Nyquist criterion which requires that the receiver suppress all ISI and

crosstalk between the different waveforms.

1.3 Partial Response Techniques

In the 60's engineers began to realize that deliberately introducing controlled ISI, then eliminating the effect of the ISI by subtraction can reduce the sensitivity to perturbations and use the bandwidth of $G(\omega)$ more efficiently . This technique is called partial response or correlative coding.

In 1964, Lender [7] suggested the duobinary technique which allows one ISI term, as shown in Fig. 1.4. The $G(\omega)$ is a cosine characteristic and occupies only a π/T bandwidth. At the receiver the ISI of previous input symbol can be eliminated by subtraction if the previous input symbol is detected correctly. However the incorrect detection of an input symbol leads to error propagation. Lender [7] suggested a precoding scheme to avoid this.

The duobinary technique was later generalized to the partial response technique, by Lender ([8],1966) and Kretzmer ([9],1965;[10],1966), which allows any number of ISI terms. Kretzmer tabulated and classified a number of partial response systems [10].

Partial response signaling has more tolerance to perturbations, but due to more signal levels to be distinguished a higher signal to noise ratio is required for equal detection performance compared with ideal (rectangular) binary signaling.

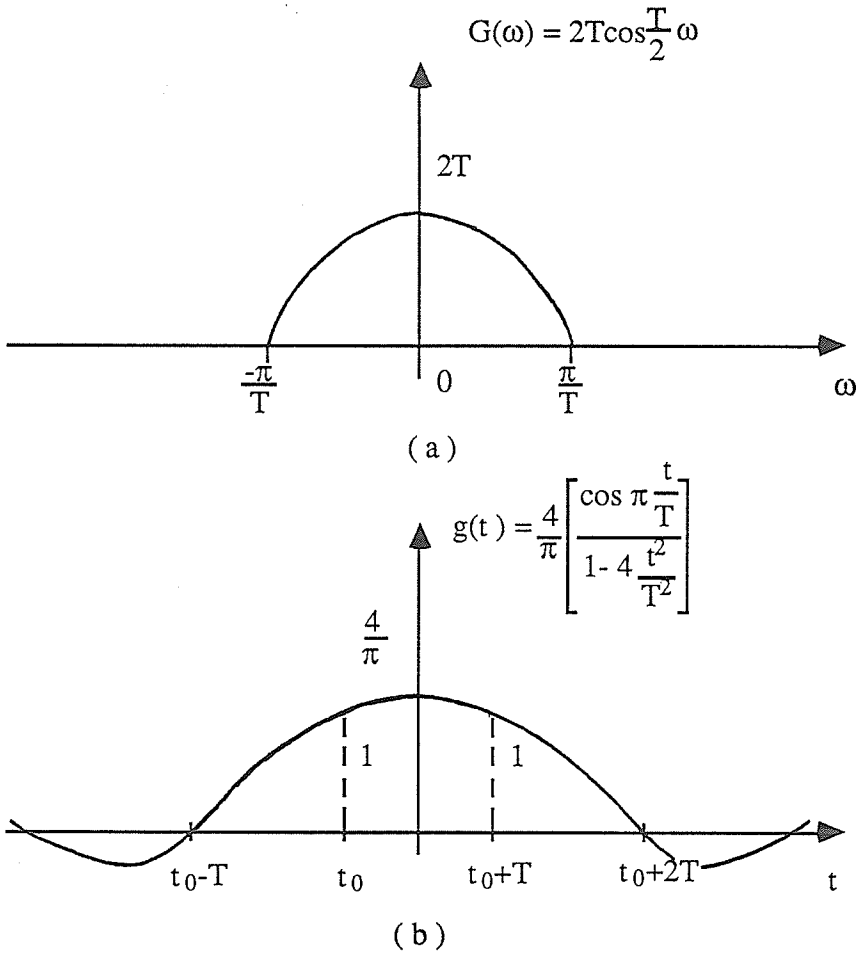

Figure 1.4: Duo-binary signaling: (a) Spectrum of the system, (b) One ISI term exists.

1.4 Linear Optimal Receivers

Other than either forcing the ISI to zero or introducing controlled ISI, many investigators have approached the ISI problem from the point of view of optimum estimation, under various criteria.

Tufts ([11], 1965) jointly optimized the transmitter and receiver for the minimum mean square error criterion. It was shown that the optimum receiver consists of a filter matched to the received pulse followed by a tapped delay line which is alternatively called a linear equalizer or transversal filter. Aaron and Tufts ([12],1966) later showed that this structure also minimizes average error probability which is a more meaningful measure of performance. Yao ([13],1972) also obtained a receiver structure for the minimum-probability-of-error criterion.

All of the above mentioned optimum receivers have the same structure: a matched filter followed by a sampler and a transversal filter. Ericson ([14],1971; [15],1973) later showed that under very general conditions and performance criteria, this structure is always optimum.

Etten ([16],1975) extended the optimum linear receiver for a single channel to multiple channels under both minimum error probability and zero-forcing criterion. Again, the receiver's structure turned out to be a multiple matched filter followed by a sampler and a multiple tapped delay line.

1.5 Linear Equalizers

Although the application of the transversal filter might seem a natural outgrowth of the optimization of the receiver structure, the practical application of transversal filters to data transmission predates the availability of the theoretical results. The transversal filter was invented by Wiener and Lee in 1935 and first described in the literature by Kallman in 1940 [3].

Fig. 1.5 is a linear transversal equalizer for a channel with causal $h(t)$, where $r'(t) = r(t) * h(-t)$ is the output of the matched filter. Tap gain settings $c_i, i = 0, 1, ..., N-1$ are chosen according to various criteria. To achieve zero ISI (zero-forcing criterion) at sampling points, generally an infinite number of the linear equalizer stages is needed, even for a channel with finite ISI. Therefore with a finite transversal equalizer, zero ISI generally is not achievable. Instead tap gain settings are usually chosen according to minimum peak distortion criterion or minimum mean squared distortion criterion [3]. In fact the minimum mean squared distortion equalizers are used in most current high-speed modems since they are more robust and have better convergence properties for adjusting tap gain settings if an adaptive scheme is incorporated.

The delay line taps can be spaced at an interval $\tau < T$ (for instance, $\tau = T/2$) to form a fractional-spaced equalizer [17]. One important property of a fractional-spaced equalizer is insensitivity of its performance to the choice of sampler phase. Since the designer seldom has the exact knowledge of

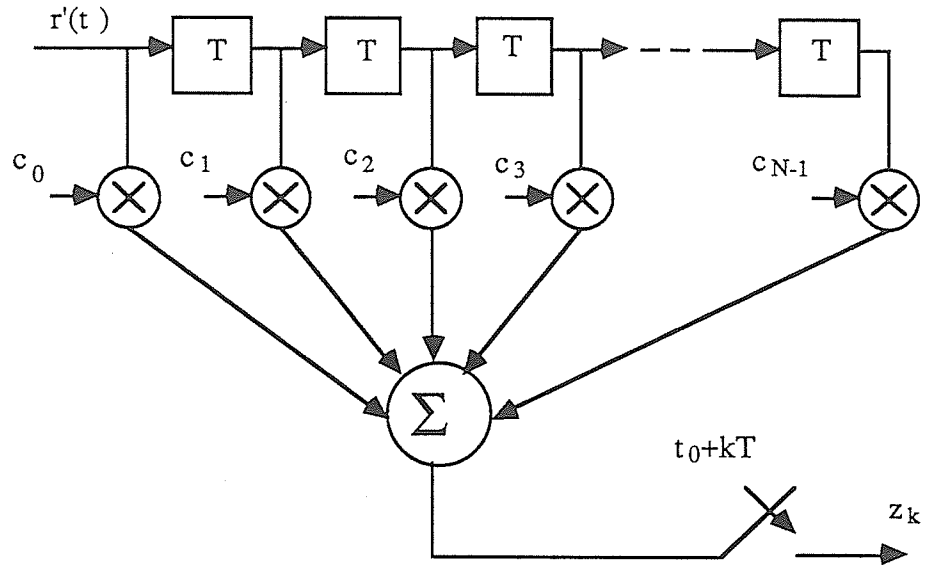

Figure 1.5: Linear transversal equalizer

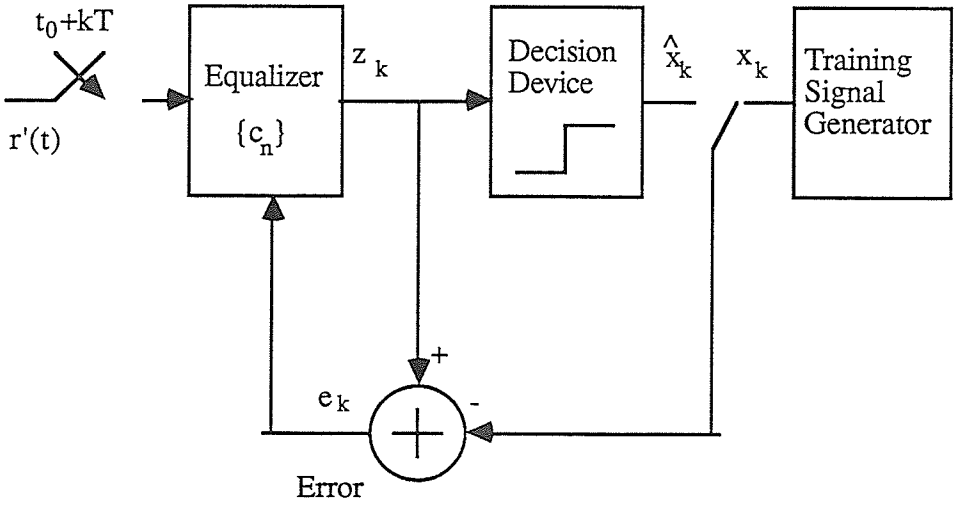

Figure 1.6: Automatic adaptive equalizer

channel characteristics and they may be time variable, an adaptive scheme for adjusting the tap gain settings of the transversal filter is needed. This technique is called automatic or adaptive equalization. Fig. 1.6 shows a block diagram of an adaptive equalizer, where error signal e_k is used to adjust tap gains automatically. Before transmitting a message there is a training period when the transmitter sends a symbol sequence identical to that produced by the training signal generator. Tap gains are set during the training period and adaptively adjusted further in the receiving period.

The pioneering work of Lucky ([18],1965; [19],1966; [3],1968) laid a sound basis for this technique. He suggested several schemes for adjusting the settings of the tapped delay line. The adaptive linear equalizer makes the theoretical results of linear optimum filtering practical for implementation in communication systems. Therefore the adaptive linear equalizer is widely used. A review paper about adaptive equalization is given by Qureshi ([20],1982).

Linear equalizers are simple to implement and analyze. Their major disadvantage is *noise enhancement*, i.e., the noise variance at the output of an equalizer is larger than that at the input. Suppose the input noise variance is σ_i^2. Then the output noise variance of the equalizer in Fig. 1.5 is [3]

$$\sigma_o^2 = \sigma_i^2 \sum_{i=0}^{N-1} c_i^2. \tag{1.4}$$

This implies that the output noise variance has a minimum value only when there is no ISI so that the tap gains are set to be 0, except $c_0 = 1$. When the

channel has severe ISI the equalizer may result in a large noise enhancement.

1.6 Nonlinear Receivers

Nonlinear receivers have also been analyzed extensively and have been shown to provide significant improvement of performance over the simpler linear ones.

Decision feedback equalization (DFE), which is suboptimum, introduced by Austin ([21],1967), is the simplest and earliest nonlinear receiver. Fig. 1.7 shows a DFE. If the past symbols are detected correctly then the ISI effect caused by these symbols can be eliminated by subtracting appropriately weighted values of the past symbols from the output of the linear equalizer, which is the feed-forward part of the DFE. The tap gains of the forward and feedback part may be chosen simultaneously to minimize the mean squared error. An adaptive scheme was later added by Gorge,Bowen and Storey ([22],1971) to improve the DFE.

With the same number of overall taps a decision feedback equalizer does not necessarily have better error probability than the linear equalizer. However, the DFE does have less noise enhancement than the linear equalizer with the same error probability. The disadvantage of a DFE is *error propagation.* That is, if an incorrect decision is fed back, it will traverse the feedback path to cause the next few decisions to be wrong. Fortunately, the error propagation is not catastrophic, it vanishes after several transmission

periods.

Several other nonlinear receivers have been proposed. Abend and Fritchman ([23],1970) developed a receiver for maximum a posteriori probability decisions on a symbol-by-symbol basis. Nonlinear symbol-by-symbol receivers have also been obtained by Gonsalves ([24],1968) using a maximum likelihood criterion and by Ungerboeck ([25],1971) under a maximum a posteriori probability criterion. Chang and Hancock ([26],1966) considering channels with ISI over L transmission periods, developed a receiver that maximizes the joint a posteriori probability of the L consecutive symbols on the basis of the complete message received. The a posteriori probability is computed by a sequential procedure.

These nonlinear receivers have an intimidating complexity in their structure. Also their performance is difficult to analyze.

1.7 Maximum Likelihood Sequence Estimation

All the aforementioned receivers, optimum or suboptimum, linear or nonlinear, make decisions on a symbol-by-symbol basis except Chang and Hancock's detection which bases the decision on L consecutive symbols. The idea that symbol decisions ought to be based on the entire received sequence has long been considered, but the fact that straightforward likelihood calculations grow exponentially with message length has justified a retreat to simple

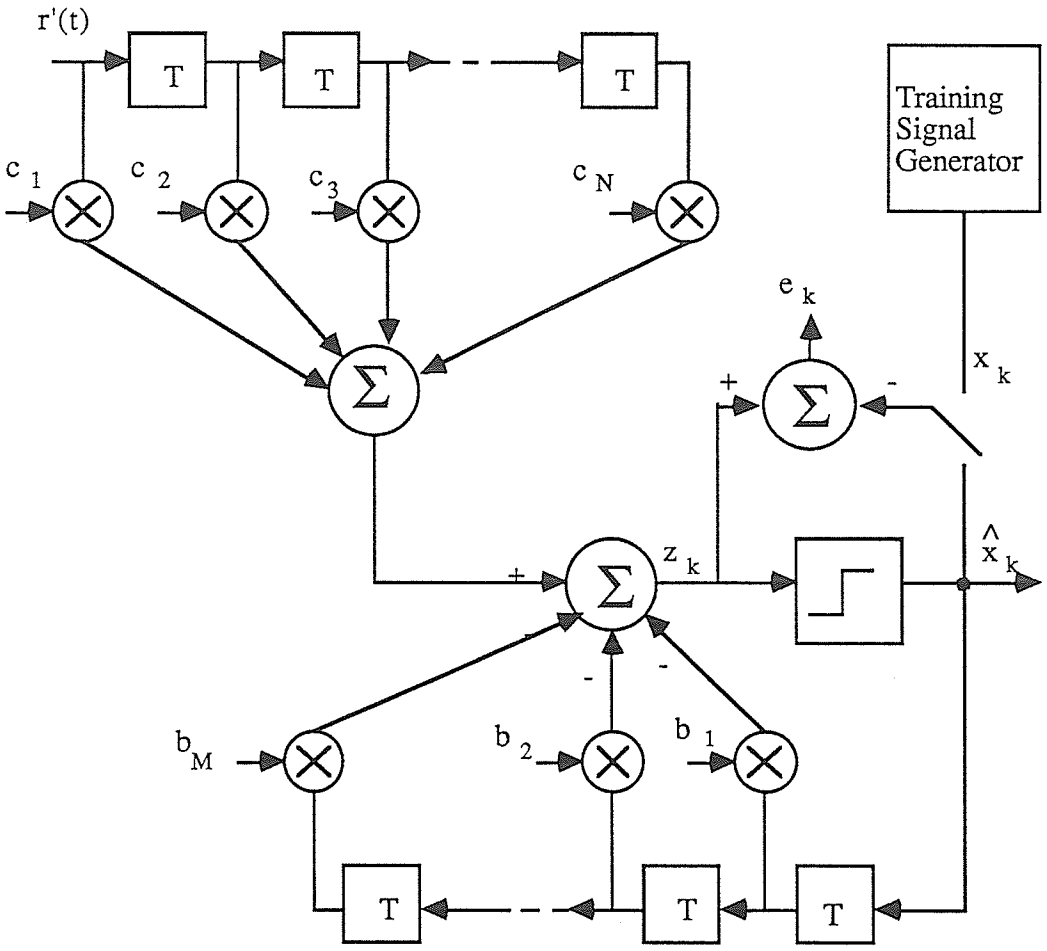

Figure 1.7: Decision feedback equalizer

symbol-by-symbol decisions. In 1967 Viterbi [27] invented a recursive nonlinear processor, called the Viterbi algorithm (VA), for decoding convolutional codes. Its applicability to partial-response systems was noticed independently by Omura ([28],1971) and Kobayashi ([29],1971). Forney ([30],1972) successfully applied VA to more general channels, i.e., channels with finite ISI. He developed a maximum likelihood sequence estimation (MLSE) which consists of a whitened matched filter (WMF) followed by a Viterbi processor. The whitened matched filter has the form of a cascade of a matched filter, a transmission rate sampler and a linear filter. This form is basically consistent with the form of the general optimum linear receiver as discussed above. It was shown [30] that the MLSE has an error performance superior to conventional symbol-by-symbol decision receivers.

Forney's work was extended by Etten ([31],1976) to the multiple channel systems where ISI and interchannel interference (ICI) are present simultaneously. A vector Viterbi algorithm was developed to handle the vector sequence estimation.

An adaptive MSLE receiver was suggested by Magee and Proakis to cope with the problem of slowly time-varying time dispersive channels([32],1973).

Acampora ([33],1976) and Viterbi and Omura ([34],1979) derived a MLSE which consists of a matched filter (without whitening), a transmission rate sampler and a Viterbi processor. Ungerboeck ([35],1974) also found the same structure.

1.8 Reduced State Algorithms

Since the complexity of the Viterbi algorithm grows exponentially with the length of ISI, for the channels with many interfering terms, the implementation of the VA becomes impractical. A great deal of effort has been made to reduce the complexity while achieving nearly the same error probability as MLSE. The common characteristic of these techniques is that they reduce the complexity by reducing the number of trellis states to which the Viterbi algorithm is applied. One may generally refer to these techniques as reduced-state Viterbi algorithms (RSVA). A variety of methods are used in these RSVA's. Earlier researchers reduced the number of trellis states by placing a prefilter prior to a Viterbi estimator. Qureshi and Newhall ([36], 1973) proposed a linear equalizer as the prefilter to reduce the channel memory to a shorter length, then the Viterbi estimator is applied. Falconer and Magee ([37], 1973) and Messerschmitt ([38], 1974) used the same structure but optimized the channel impulse response shape for a given length under several criteria. Lee and Hill ([39], 1975,[40], 1977) shortened the channel memory with a decision feedback equalizer before applying the Viterbi algorithm in an effort to avoid the noise enhancement introduced by the linear equalizer. Other researchers were interested in reducing the complexity by searching only a small number of the most likely paths or states in the trellis [41,42,43,44,45]. The most recent development in this category is the scheme proposed by Eyuboglu and Qureshi ([46],1988). They used the VA with de-

cision feedback to search a reduced-state *subset trellis* which is constructed using Ungerboeck-like set partitioning principles [47]. This technique is particularly useful for QAM modems which have a large size signal set, since a large number of states still exist even if the channel memory length is reduced. When this technique is applied to a binary transmission, it becomes the decision feedback sequence estimation (DFSE) proposed by Duel and Heegard [48], where DFSE was extended to infinite state Markov channel. Polydoros and Kazakos proposed a scheme called modified Viterbi algorithm (MVA) in an effort to solve the problem of sequence estimation for channels with infinite ISI [49]. Seshadri and Anderson used the M-algorithm for ISI channels [50,51]. The M-algorithm searches only M paths in the trellis, where M is a number smaller than the number of paths searched by the VA.

RSVA techniques have the advantages of retaining the structure of MLSE-VA and reducing the computational complexity of MLSE-VA to different extents. However the price paid is the sacrifice of optimality of the MLSE-VA. For any specific RSVA technique the more the computational saving the greater the degradation of detection performance. Therefore compromise has to be made between detection performance and computational complexity.

1.9 Sequential Sequence Estimation

As seen from above review, although the traditional methods of combatting ISI are still used, researchers are actively seeking better ones. The interest

has shifted from symbol-by-symbol detection to sequence estimation. Forney's maximum likelihood sequence estimation seems to provide an ultimate solution to the ISI problem, but its complexity grows exponentially with the length of ISI. Reduced-state algorithms cut down the complexity, but the performance degradation may be substantial if a dramatic reduction of complexity is desired.

Now the following question arises: is it possible to find a receiver which can handle long ISI, even infinite ISI, with much less computational complexity than MLSE-VA, while its performance is essentially the same as that of MLSE-VA? The sequential algorithm (SA) for decoding convolutional codes [52,53,54,55] may be a solution. Historically the sequential algorithm was the first practical decoding technique for convolutional codes [52] and was the only one until the Viterbi algorithm. Due to the regularity of its decoding procedure the VA has been widely used for decoding short-constraint-length convolutional codes. But the sequential algorithm is still attractive for decoding long-constraint-length convolutional codes since the SA's computational complexity is almost independent of the code constraint length and its detection performance is essentially maximum likelihood [56]. The decoding effort of SA, however, is a random variable with a Pareto distribution [57,56]. Although nearly all received signal frames are decoded with very little computation, making the average computational effort much less than that of the VA, there is a small fraction of frames which need more computation,

thus causing buffer overflow (Fano algorithm and stack algorithm) or stack overflow (stack algorithm only). This results in frame erasures of the received sequence. To overcome this Chevillat and Costello proposed the multiple-stack algorithm (MSA)[58]. A backsearch limiting technique may be incorporated into the Fano algorithm to prevent erasures. With these techniques, the great majority of received frames are decoded very quickly according to maximum likelihood criterion, while for those few frames which require excessive searching, a good non-maximum-likelihood estimate is obtained. Thus all received frames are completely decoded within a computational limit.

Since the ISI channel's memory characteristic is equivalent to that of the convolutional encoder, it appears natural to adopt the sequential algorithm to the ISI channel, especially for channels with long memory. So far no full investigation of this problem can be found in literature. Only a conference proceeding abstract [59] described an application of a multiple-path single-stack sequential algorithm to ISI channels with finite memory. From the abstract it is not clear what the receiver structure and metric of the SA are. No mention is made of applying the SA to channels with infinite ISI, to channels with coded input sequence or to multiple channels.

1.10 Objective of this Study

The objective of this study is to fully investigate the applicability of the sequential algorithm to sequence estimation for ISI channels. The structure

of the receiver using the sequential algorithm is to be studied. The error probability and computational complexity are to be analyzed and simulated with computer. To assure a comprehensive applicability, the simulations are planned for different representative channels. The results are to be compared with Viterbi algorithm and selected reduced-state algorithms.

1.11 Outline of the Thesis

In Chapter 2, first the receiver structure to which the sequential algorithm can be applied is derived for the PAM ISI channels. The metric, which is essential to any sequential algorithm, is derived next. Then the results are extended to channels with infinite ISI. The problem of calculation of the metric and the factorization of the autocorrelation function of the channel, which is important to the construction of the receiver, is discussed.

Chapter 3 is devoted to the analysis of error performance and computational complexity of the SA. Both analytical and simulation results are presented.

In Chapter 4, the sequential sequence estimation is extended to multiple channel systems.

Chapter 5 gives the conclusions and the recommendations for future study.

Chapter 2

Sequential Sequence Estimation for ISI Channels

In this chapter the PAM channel model used in this study is introduced first. Then the receiver structure i.e.,the whitened matched filter to which the sequential algorithms are applied is described. Based on this receiver structure the application of the SA's to the output of the whitened matched filter is discussed. Since detailed descriptions for a variety of sequential algorithms can be found in many textbooks and the literature[34,74,52,53,54,55], the emphasis of this study is put on the special problems associated with ISI channel. Particularly, the derivation and the calculation of the metric is discussed in detail. The important extension from the finite ISI case to the infinite ISI case is discussed. Factorization of the channel's autocorrelation function, which is necessary to construct the whitening filter, is also studied in this chapter.

2.1 Channel Model and Whitened Matched Filter

As pointed out in Section 1.1, although ISI arises in all pulse-modulation systems, such as PAM, FSK, PSK and QAM systems, its effect can be studied most easily with a baseband PAM system. The simplified baseband PAM model is shown in Fig.2.1. A sequence of data $\sum_k x_k \delta(t - kT), k = 0, 1, ...K$, where K may be finite or infinite, is applied to the input, where x_k is drawn from a discrete finite alphabet $\{0, 1, ..., m-1\}$. The channel is characterized by a finite impulse response $h(t)$ of length L symbol intervals, i.e. $h(t) = 0$ for $t > LT$ and $t < 0$. The response $h(t)$ is assumed to be square-integrable,

$$\|h\|^2 \triangleq \int_{-\infty}^{\infty} h^2(t)dt < \infty. \tag{2.1}$$

The channel output

$$s(t) = \sum_{k=0}^{K} x_k h(t - kT), \tag{2.2}$$

is corrupted by white Gaussian noise $n(t)$ of zero mean and spectral density σ^2. The received signal is

$$r(t) = s(t) + n(t). \tag{2.3}$$

A whitened matched filter (WMF) is needed for the MLSE as developed by Forney[30], and is also needed for sequential sequence estimation. A brief discussion of the WMF is presented here.

To facilitate the discussion several definitions are used. First a D-transform notation is used, which in fact is the well-known z-transform[81], with z^{-1}

replaced by symbol D. For example, the D-transform for the data sequence $x_0, x_1, x_2, \ldots\ldots$ is

$$x(D) \triangleq x_0 + x_1 D + x_2 D^2 + \ldots\ldots \tag{2.4}$$

All the properties of z-transform can be applied to D-transform provided that z^{-1} is replaced by D. Particularly, the D-transform of the convolution of two sequences is the product of their D-transforms(convolution theorem).

Define the pulse autocorrelation coefficients of $h(t)$ as

$$R_k \triangleq \begin{cases} \int_{-\infty}^{\infty} h(t)h(t-kT)dt, & |k| \leq L-1 \\ 0, & |k| \geq L \end{cases} \tag{2.5}$$

then the D-transform of R_k is,

$$R(D) \triangleq \sum_{k=-\nu}^{\nu} R_k D^k. \tag{2.6}$$

where $\nu = L-1$ is called the *span* of $h(t)$. $R(D)$ is alternatively called the pulse autocorrelation function (in D-transform) or (discrete) spectral function of $h(t)$.

The response $h(t)$ may be regarded as a sequence of L chips $h_i(t), 0 < i < L-1$, where $h_i(t)$ is a function that is nonzero only over the interval [0,T), i.e.

$$h(t) = h_0(t) + h_1(t-T) + \ldots + h_\nu(t-\nu T). \tag{2.7}$$

The chip D-transform of $h(t)$ which is used later is therefore defined as

$$h(D,t) \triangleq \sum_{i=0}^{\nu} h_i(t) D^i. \tag{2.8}$$

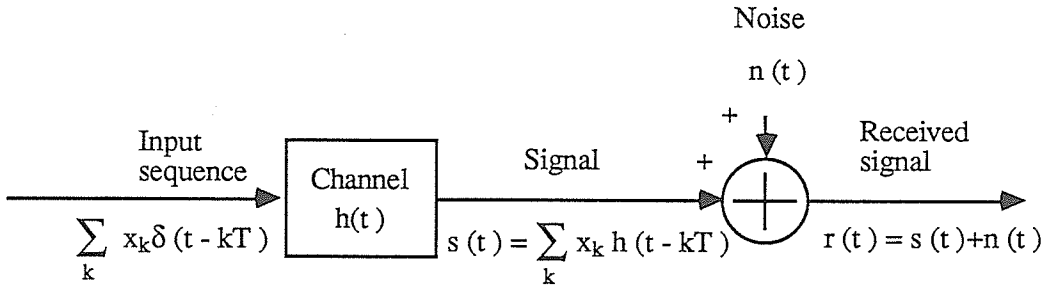

Figure 2.1: PAM channel model

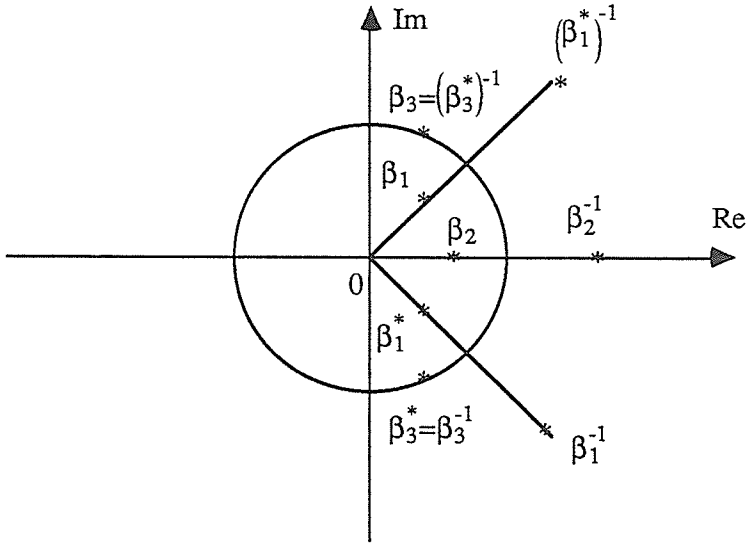

Figure 2.2: Root distribution of a finite R(D)

which is a polynomial in D of degree ν with coefficients in the set of functions over [0,T). With the above as background the whitened matched filter is discussed next.

Since $R(D)$ in eqn.(2.6) is finite with $2\nu + 1$ nonzero terms, it has 2ν complex roots. Further, since $R(D) = R(D^{-1})$, the inverse β^{-1} of any root β is also a root of $R(D)$, so the roots break up into ν pairs. Then if $f'(D)$ is any polynomial of degree ν whose roots consist of one root from each pair of roots of $R(D)$, $R(D)$ has the spectral factorization,

$$R(D) = f'(D)f'(D^{-1}). \tag{2.9}$$

Fig. 2.2 shows a possible root distribution for a finite $R(D)$ on a complex D-plane. Complex roots not on the unit circle are in groups of four. Complex roots on the unit circle are in pairs, but each root must be double[67]. Real roots are in pairs. When there are no unit circle roots these roots can be grouped such that all the roots inside the unit circle form $f'(D)$ and all the roots outside the unit circle form $f'(D^{-1})$ or vice versa.

Eqn.(2.9) can be generalized slightly by letting $f(D) = D^n f'(D)$ for any integer delay n, then

$$R(D) = f(D)f(D^{-1}). \tag{2.10}$$

The signal in (2.2) can also be written as

$$s(t) = \sum_{k=0}^{K} x_k h^{(k)}(t). \tag{2.11}$$

where

$$h^{(k)}(t) \triangleq h(t - kT) \tag{2.12}$$

Eqn.(2.11) shows that $s(t)$ can be expressed as a linear combination of the finite set of square-integrable basis functions $h^{(k)}(t)$. In the detection of these type of signals in white Gaussian noise, the outputs of a bank of matched filters, one matched to each basis function, form a set of sufficient statistics for estimating the coefficients[71]. Thus the $K+1$ quantities

$$a_k \triangleq \int_{-\infty}^{\infty} r(t)h(t - kT)dt, 0 \leq k \leq K \tag{2.13}$$

form a set of sufficient statistics for estimation of the $x_k, 0 \leq k \leq K$, where K is finite or infinite. But these are simply the sampled outputs of a filter $h(-t)$ matched to $h(t)$.

The D-transform of the matched filter output sequence is

$$a(D) \triangleq \sum_{k=0}^{K} a_k D^k. \tag{2.14}$$

Then

$$\begin{aligned} a_k &= \int_{-\infty}^{\infty} r(t)h(t - kT)dt \\ &= \sum_{j=0}^{K} \int_{-\infty}^{\infty} x_j h(t - jT)h(t - kT)dt + \int_{-\infty}^{\infty} n(t)h(t - kT)dt \\ &= \sum_{j=0}^{K} x_j R_{j-k} + n'_k. \end{aligned} \tag{2.15}$$

or

$$a(D) = x(D)R(D) + n'(D). \tag{2.16}$$

Here $n'(D)$ is zero-mean colored Gaussian noise with autocorrelation function $\sigma^2 R(D)$, since

$$\begin{aligned} E\{n'_k n'_j\} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E\{n(t)n(\tau)\}h(t-kT)h(\tau-jT)dtd\tau \\ &= \sigma^2 R_{k-j} \end{aligned} \tag{2.17}$$

where $E\{n(t)n(\tau)\} = \sigma^2\delta(t-\tau)$ because $n(t)$ is white and of strength σ^2.

The colored noise $n'(D)$ can be expressed as

$$n'(D) = n(D)f(D^{-1}), \tag{2.18}$$

since $n'(D)$ then has the autocorrelation function $\sigma^2 f(D^{-1})f(D) = \sigma^2 R(D)$ and zero-mean Gaussian noise is entirely specified by its autocorrelation function. As a result, (2.16) can be written as

$$a(D) = x(D)f(D)f(D^{-1}) + n(D)f(D^{-1}). \tag{2.19}$$

This suggests that by simply dividing out the factor $f(D^{-1})$ one obtains

$$\begin{aligned} z(D) &= \frac{a(D)}{f(D^{-1})} \\ &= x(D)f(D) + n(D) \end{aligned} \tag{2.20}$$

where the noise $n(D)$ is *white*.

Thus the overall receiver can be drawn as Fig.2.3 (a), where to assure causality $h(-t)$ is delayed by LT. This cascade is called *whitened matched filter*.

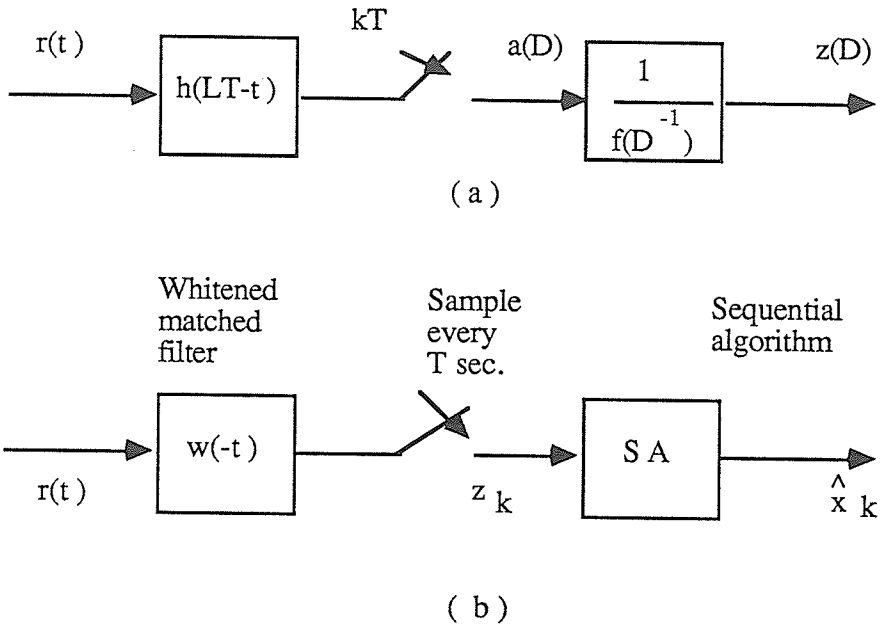

Figure 2.3: (a) Whitened matched filter, (b) Sequential sequence estimator

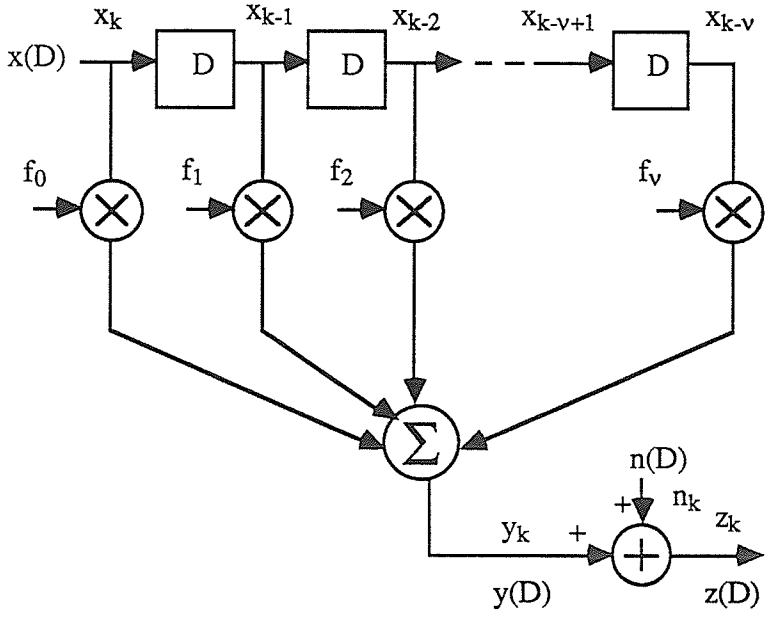

Figure 2.4: Finite state machine model

If causality of the whitening filter is not of concern then *any* spectral factorization of $R(D)$ can be used to form a *whitened matched filter* $w(-t)$ whose time inverse has the chip D-transform

$$w(D,t) = h(D,t)/f(D). \tag{2.21}$$

The sampled outputs z_k of $w(-t)$

$$z_k = \int_{-\infty}^{\infty} r(t)w(t-kT)dt \tag{2.22}$$

form a sequence

$$z(D) = f(D)x(D) + n(D) \tag{2.23}$$

in which $n(D)$ is a white Gaussian noise sequence with spectral density $E\{n_k^2\} = \sigma^2$, and which is a set of sufficient statistics for estimation of the input sequence $x(D)$.

This implies that the relationship between the channel input sequence $\{x_k\}$ and the sampled output sequence $\{z_k\}$ of the whitened matched filter $w(-t)$ can be modelled as a finite state machine (FSM) as shown in Fig.2.4, where

$$y(D) = f(D)x(D). \tag{2.24}$$

But for an *arbitrary* spectral factorization of $R(D)$, the causality of the whitening filter is not guaranteed. To actually realize $w(-t)$, the factorization of $R(D)$ has to be such that the whitening filter $1/f(D^{-1})$ is stable and causal, and preferably is real.

That $f(D)$ and $f(D^{-1})$ are real is guaranteed by the fact that $R(D)$ is an autocorrelation function [67] (Although the proof there is made for rational $R(D)$, finite $R(D)$ is just a special case). If all roots of $R(D)$ are either inside or outside the unit circle, there are two ways to get a causal stable whitening filter.

(1) Have $f(D^{-1})$ contain all the roots outside the unit circle. This is possible since roots of $R(D)$ are in reciprocal pairs. Thus $1/f(D^{-1})$ is stable. Its implementation can be achieved by a feedback structure. Assume

$$f(D^{-1}) = f_0 + f_1 D^{-1} + \ldots + f_\nu D^{-\nu}. \tag{2.25}$$

Then by multiplying both sides of relation $z(D)f(D^{-1}) = a(D)$ by D^ν , one obtains

$$D^\nu a(D) = (f_0 D^\nu + f_1 D^{\nu-1} + \ldots + f_\nu) z(D). \tag{2.26}$$

The inverse D-transform gives

$$z_k = \frac{1}{f_\nu}(a_{k-\nu} - f_0 z_{k-\nu} - f_1 z_{k-\nu+1} - \ldots - f_{\nu-1} z_{k-1}). \tag{2.27}$$

This equation can be implemented as a causal feedback filter as in Fig.2.5.

(2) The other way to implement the whitening filter was suggested by Forney. Factorize $R(D)$ as

$$R(D) = f_c(D) f_c(D^{-1}), \tag{2.28}$$

where $f_c(D)$ contains all the roots of $R(D)$ outside the unit circle. This is called a *canonical* factorization. Then choose the whitening filter as $\frac{1}{D^{-\nu} f_c(D)}$. Thus,

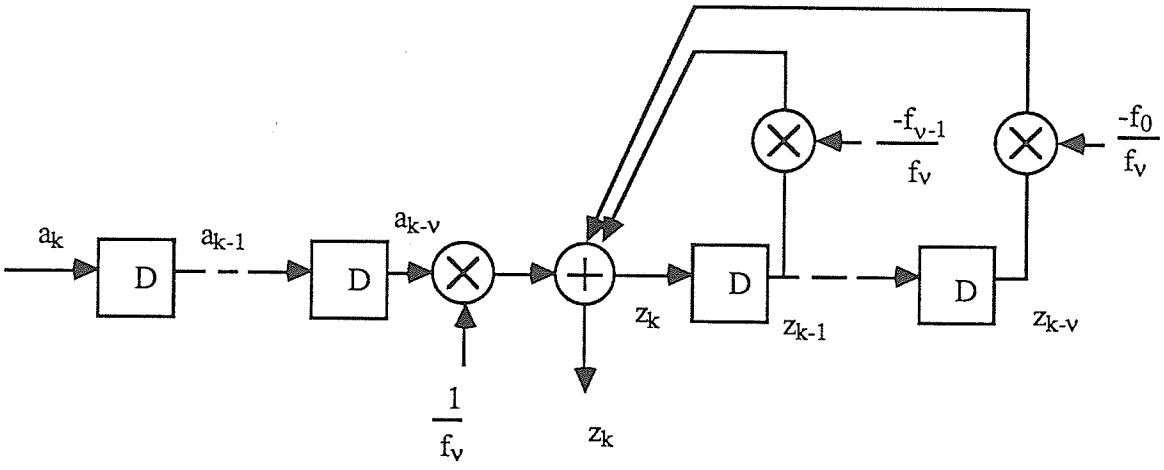

Figure 2.5: Implementation of the whitening filter $\frac{1}{f(D^{-1})}$

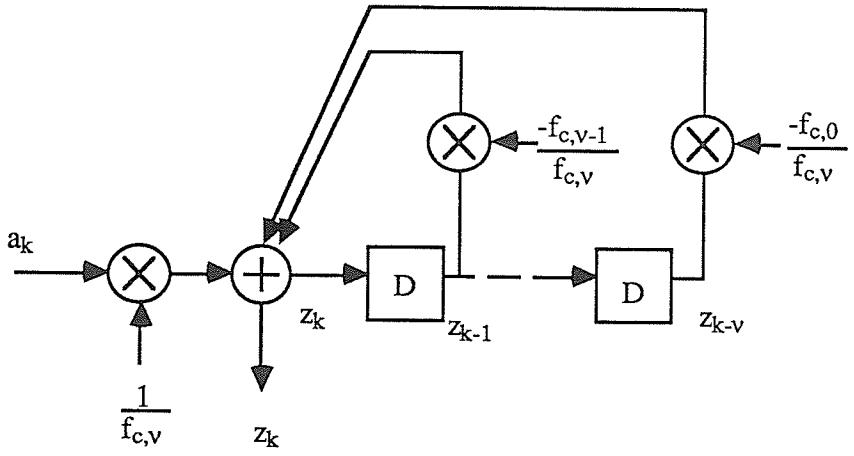

Figure 2.6: Implementation of the whitening filter $\frac{1}{D^{-\nu} f_c(D)}$

$$\frac{a(D)}{D^{-\nu}f_c(D)} = x(D)D^{\nu}f_c(D^{-1}) + n(D), \tag{2.29}$$

where

$$n(D) = \frac{n'(D)}{D^{-\nu}f_c(D)}. \tag{2.30}$$

The autocorrelation function of $n'(D) = D^{-\nu}f_c(D)n(D)$ is $\sigma^2 R(D)$ and still represents the colored Gaussian noise after the matched filter. All the roots of $f_c(D)$ are also roots of $D^{-\nu}f_c(D)$. Since they are outside the unit circle, the whitening filter is stable.

Again implementation of the filter is a feedback filter. Assume

$$f_c(D) = f_{c,0} + f_{c,1}D + \ldots + f_{c,\nu}D^{\nu}. \tag{2.31}$$

Then

$$a(D) = (f_{c,0}D^{-\nu} + f_{c,1}D^{-\nu+1} + \ldots + f_{c,\nu})z(D). \tag{2.32}$$

Take the inverse D-transform, then

$$z_k = \frac{1}{f_{c,\nu}}(a_k - f_{c,0}z_{k-\nu} - f_{c,1}z_{k-\nu+1} - \ldots - f_{c,\nu-1}z_{k-1}). \tag{2.33}$$

This equation can be implemented as a feedback filter as in Fig.2.6, which is causal. Comparing it to Fig.2.5, it can be seen that this canonical implementation is better since a ν period delay is avoided.

However it should be noticed that the output $y(D)$ in this case is different from that of the first case,

$$y(D) = D^{\nu}f_c(D^{-1})x(D). \tag{2.34}$$

Consequently the discrete time model is somewhat different, as shown in Fig.2.7.

To make discussion easier yet without loss of generality, in the rest of this study, the whitening filter is always supposed to be $1/f(D^{-1})$, hence the discrete model is the model shown in Fig.2.4. It is also alternatively called a finite state machine.

In the time domain the outputs of the discrete model are

$$z_k = y_k + n_k, \tag{2.35}$$

where n_k is a zero mean white Gaussian noise with spectral density σ^2, and

$$y_k = \sum_{i=0}^{\nu} f_i x_{k-i}, \tag{2.36}$$

where $f_i, i = 0, 1, ..., \nu$ are the coefficients of $f(D)$. Note that this equation reflects the effect of ISI on the output signal.

In some contexts the channel itself is discrete time and such a model arises directly. For example, in a partial response system the spectral shaping may be achieved by passing the input sequence $x(D)$ through a discrete time filter such as $1 - D$ to give the output $y(D) = (1 - D)x(D)$. Then the finite state machine model in Fig.2.4 is characterized by $f(D) = 1 - D$.

The statistical properties of y_k are of interest. Since $x_{k-i}, i = 0, 1, \ldots, \nu$ are discrete random variables, y_k is a discrete random variable as well. Since y_k is the function of L x's, and each x can be one of m values in $\{0, 1, 2, \ldots, m-1\}$ independently, y_k can be one of at most $Q = m^L$ distinct values. Denote

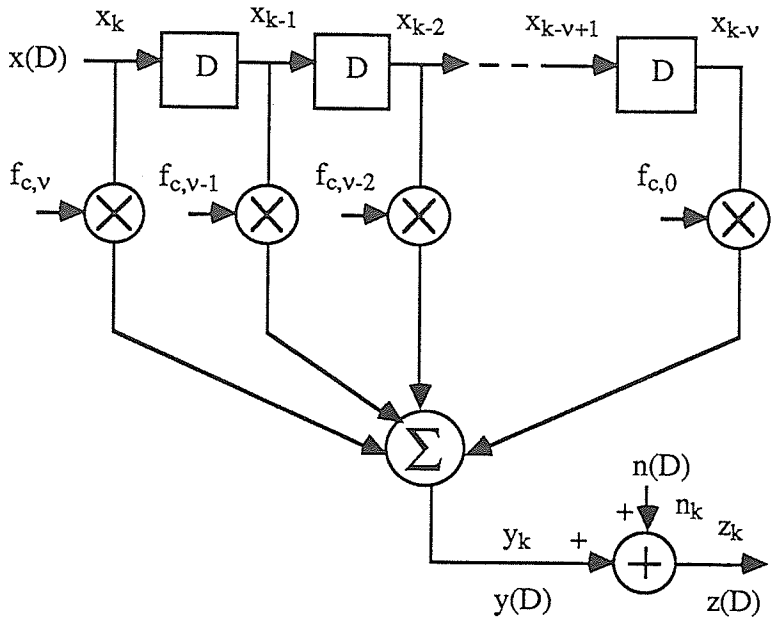

Figure 2.7: Finite state machine model for the canonical implementation of the whitening filter

the set of these values as $\mathcal{Y} \triangleq \{b_1, b_2, \ldots, b_Q\}$, $Q \leq m^L$. If x_k is equally likely to take one of the m elements in the input alphabet, and all the possible values of y_k are distinct, then y_k's are also equally likely, with $Pr(y_k = b_j) = 1/m^L$, $j = 1, 2, \ldots, m^L$. In general, the probability distribution of y_k can be denoted as $q(y_k)$, where $q(y_k) \in \mathbf{q} \triangleq \{q_1, q_2, \ldots, q_Q\}$, $Q \leq m^L$.

2.2 Sequential Estimation and the Metric

Based on the discrete model in Fig.2.4 Forney developed the MLSE by applying the Viterbi algorithm to the sequence $z(D)$. Instead of the Viterbi estimator, a sequential estimator can be used, as shown in Fig. 2.3 (b).

As is well known, sequential decoding is best depicted by a code tree. For the ISI problem here, the input-output relationship of the finite state machine model of Fig.2.4 may also be represented by a tree. Fig.2.8 gives an example for a binary sequence and a channel of memory $\nu = 2$. Once the tree is established, the application of sequential algorithms, such as stack algorithm and Fano algorithm, to the tree is straightforward and well known. Therefore the attention is focused on the problem of finding a suitable metric for the sequential algorithm.

The sequential algorithm is an efficient tree search procedure which attempts to find the maximum likelihood path without examining too many nodes. Since the algorithm explores paths to different depths in the tree, it is necessary to have a metric which would reflect the different path lengths explored. This metric is developed here for the ISI channel. The procedure is essentially that described by Massey [60] and as will be seen the derived metric can be interpreted as the Fano metric.

Consider the set $\{X_1, X_2, ..., X_M\}$ which represents all the possible input sequences that correspond to all paths in the ISI tree that have been searched by a sequential estimator. Denote the input sequence X_l as

$$X_l \triangleq \{x_1, x_2, ..., x_{n_l}\}, \tag{2.37}$$

where n_l is the number of input symbols. The corresponding output sequence of the finite state machine model is

$$Y_l \triangleq \{y_1, y_2, ..., y_{n_l}\}, \tag{2.38}$$

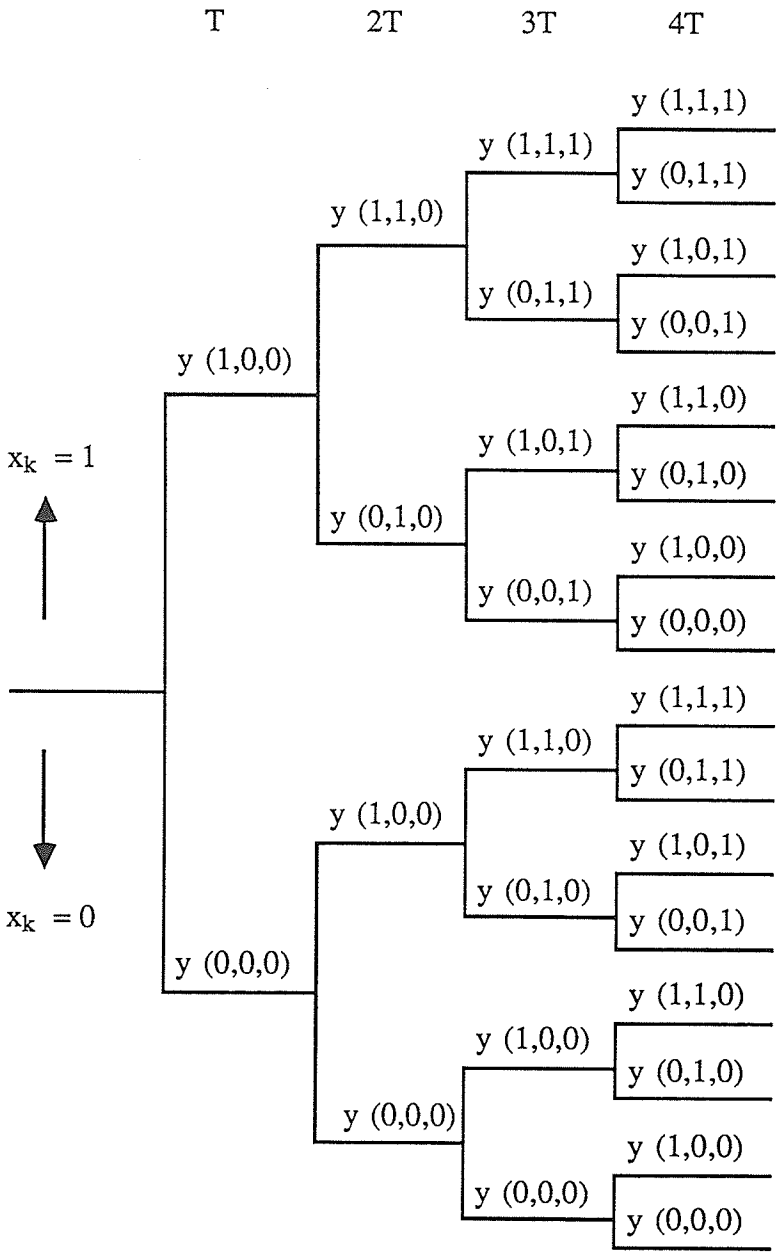

Figure 2.8: ISI tree for a binary input sequence and a channel of memory $\nu = 2$. Note $y = y(x_k, x_{k-1}, x_{k-2})$.

where y_k is the weighted summation of contiguous L symbols of X_l, as shown in (2.36).

Now suppose that $N = \max\{n_1, n_2, ..., n_M\}$ and add a *random tail* to Y_l to form a sequence with a full length N

$$Y = \{y_1, y_2, ..., y_{n_l}, t_1, t_2, ..., t_{N-n_l}\} = \{Y_l, T_l\}, \tag{2.39}$$

where $T_l = \{t_1, t_2, ..., t_{N-n_l}\}$ is the random tail. Further assume that T_l is selected statistically independently of Y_l, and that the symbols in T_l are selected independently according to the probability distribution $\mathbf{q}$ over the finite discrete output set $\mathcal{Y}$. This assumption is made because even though due to the memory effect of the finite state machine model the tail symbols correspond to the *head* Y_l; *without exploring the tree further*, one may as well assume that further symbols are selected randomly. This is a convenient approximation which ignores the correlation between the tail and head. The same approximation was used by Massey to derive the Fano metric for decoding convolutional codes. The simulation results show that the metric works well. As a result,

$$Pr(T_l/Y_l) = Pr(T_l) = \prod_{k=1}^{N-n_l} q(t_k). \tag{2.40}$$

The sufficient statistics for sequence estimation are the sampled outputs of the whitened matched filter, as shown in equation (2.22). Let $Z = \{z_1, z_2, ..., z_N\}$ be the sufficient statistic sequence. Then the conditional prob-

ability density of Z is

$$p(Z/Y) = \prod_{k=1}^{n_l} p_n(z_k - y_k) \prod_{j=1}^{N-n_l} p_n(z_j - t_j), \tag{2.41}$$

where $p_n(\cdot)$ is the noise probability density function, i.e.

$$p_n(x) \triangleq \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{x^2}{2\sigma^2}\right\}. \tag{2.42}$$

Note that the independence of the n_k's is used in (2.41).

The joint probability density of sending message X_l (therefore receiving Y_l), adding the random tail T_l, and receiving Z can be written as

$$\begin{aligned} p(X_l, T_l, Z) &= p(Y_l, T_l, Z), \\ &= Pr(Y_l) Pr(T_l/Y_l) p(Z/Y_l, T_l), \\ &= Pr(X_l) Pr(T_l) p(Z/Y), \\ &= P_l \prod_{j=1}^{N-n_l} q(t_j) \prod_{k=1}^{n_l} p_n(z_k - y_k) \prod_{j=1}^{N-n_l} p_n(z_j - t_j), \end{aligned} \tag{2.43}$$

where $P_l = Pr(X_l)$ is the probability of sending message sequence X_l . Summing over all possible random tails, one obtains

$$p(X_l, Z) = P_l \prod_{k=1}^{n_l} p_n(z_k - y_k) \prod_{k=1}^{N-n_l} p_z(z_k), \tag{2.44}$$

where

$$p_z(z_k) \triangleq \sum_{t_j} p_n(z_k - t_j) q(t_j) \tag{2.45}$$

is the unconditional probability density induced on the sufficient statistics z when the signal t_j is assumed to appear randomly according to $q(t_j)$.

The estimation rule is to choose the path X_l, which maximizes $p(X_l, Z)$ or equivalently maximizes

$$p(X_l, Z) / \prod_{k=1}^{N} p_z(z_k), \tag{2.46}$$

since the denominator is independent of X_l.

Taking the logarithm (arbitrary base) and using (2.44), (2.45) yields

$$L(X_l, Z) = \sum_{k=1}^{n_l} \left[\log \frac{p_n(z_k - y_k)}{p_z(z_k)} + \frac{1}{n_l} \log P_l \right], \tag{2.47}$$

which is the final statistic, called metric, to be maximized by the estimator.

For an equally likely m-ary input sequence,$P_l = 1/m^{n_l}$, and $L(X_l, Z)$ becomes

$$L(X_l, Z) = \sum_{k=1}^{n_l} \left[\log \frac{p_n(z_k - y_k)}{p_z(z_k)} - \log m \right]. \tag{2.48}$$

Eqn.(2.45) also can be expressed as

$$L(X_l, Z) = \sum_{k=1}^{n_l} L(y_k, z_k), \tag{2.49}$$

where

$$L(y_k, z_k) \triangleq \log \frac{p_n(z_k - y_k)}{p_z(z_k)} - \log m \tag{2.50}$$

is called a branch metric, corresponding to a branch in the ISI tree.

It is worthwhile to point out that the metric in (2.50) is actually a special form of the Fano metric. Note that the branch metric $L(y_k, z_k)$ not only depends on the present input x_k, but also depends on the previous ν input symbols (eqn.(2.36)). In the sequential search process, y_k is calculated based

on the tentative estimates of the previous ν consecutive input symbols and the present guess of x_k, i.e.,

$$y_k = f_0 x_k + f_1 \tilde{x}_{k-1} + f_2 \tilde{x}_{k-2} + \ldots + f_\nu \tilde{x}_{k-\nu}, \tag{2.51}$$

where $\tilde{x}_i, i \leq k-1$, is a tentative estimate of x_i, and x_k is assumed to be one of the m alphabet elements. In the binary case, x_k is either 0 or 1.

To determine the metric (2.50) one must determine $p_z(z_k)$. Computation of $p_z(z_k)$ is discussed in Section 2.5.

2.3 Metric Derivation from Another Point of View

In this section it is shown that the finite state machine model in Fig.2.4 is actually a special case of the encoder model used for deriving the ensemble average performance for the sequential decoding of convolutional codes. The encoder model used by Viterbi and Omura [34] is shown in Fig. 2.9. Now redraw Fig.2.4 as Fig.2.10. Apparently the tapped delay line in Fig.2.4 is treated as a special convolutional *encoder* expressed as

$$v(D) = G(D)x(D) \tag{2.52}$$

where

$$G(D) \triangleq \begin{bmatrix} 1 \\ D \\ D^2 \\ \cdot \\ \cdot \\ \cdot \\ D^{\nu} \end{bmatrix} \tag{2.53}$$

is the generator matrix in the D-transform. The encoder has a one-dimensional input sequence $x(D)$ and an L-dimensional output sequence

$$\mathbf{v}(D) \triangleq \begin{bmatrix} v^{(0)}(D) \\ v^{(1)}(D) \\ v^{(2)}(D) \\ \cdot \\ \cdot \\ \cdot \\ v^{(\nu)}(D) \end{bmatrix}. \tag{2.54}$$

The summation part in Fig.2.4 now is considered as a mapping from vector sequence $\mathbf{v}(D)$ to Q-ary sequence $y(D)$, where $Q = m^L$. It is easy to verify that

$$y(D) = \mathbf{f}'\mathbf{v}(D) = (f_0, f_1, ..., f_\nu)\mathbf{v}(D), \tag{2.55}$$

where $'$ denotes transpose. This expression is equivalent to eqn.(2.24). Having the model of Fig.2.10, the problem of applying sequential algorithms to the output $z(D)$ of the finite state machine becomes a problem of applying sequential algorithms to the output of an AWGN channel with Q-ary input $y(D)$, which is the output (through a one-to-one mapping) of a special convolutional code. This interpretation first verifies that the metric of sequential algorithms used for ISI channels should be the Fano metric as derived directly

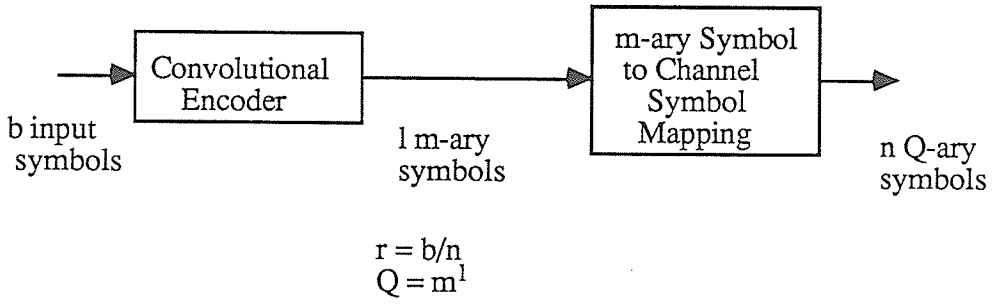

Figure 2.9: Convolutional encoder and Q-ary input channel

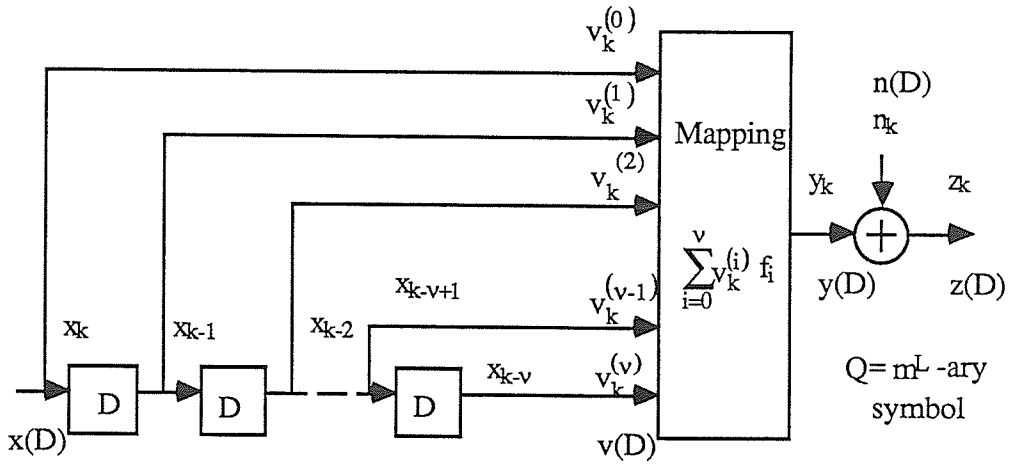

Figure 2.10: The finite state machine model is equivalent to a special convolutional encoder

by Massey's procedure (the previous section). Secondly this interpretation makes the *ensemble performance analysis* for sequential algorithms based on the ensemble of time-varying convolutional codes[34] applicable here. This is discussed in detail in Chapter 4.

2.4 Extension to Channels with Infinite ISI

In the previous sections, the sequential algorithm was developed for channels with finite ISI. Although channels such as partial response channels do have finite ISI, many practical channels, for instance, telephone lines and magnetic recording channels have an infinite memory. Therefore estimation algorithms for infinite impulse response (IIR) channels are of significance. All of the reduced state algorithms except [48,49] mentioned in the introduction cannot handle IIR channels accurately because they assume a finite impulse response. Polydoros and Kazakos[49] suggested a modified Viterbi algorithm(MVA) for channels with rational $f(D)$. Six years later Duel and Heegard[48] developed a delayed decision feedback feedback sequence estimator (DDFSE). To reduce complexity, the Viterbi algorithm only considers the first few terms of ISI, leaving the residual ISI terms to be carried along with the path and recursively updated by feedback. When the branch metric is calculated, the residual ISI terms are taken into account. Although Duel and Heegard do not refer to [49], their scheme is a generalization of the MVA in [49], with the advantage that DDFSE can deal with both finite and infinite

ISI channels. However, the MVA and DDFSE are not maximum likelihood since not all the ISI terms are treated by the original Viterbi algorithm.

In this section, it is shown that sequential algorithms can be applied to IIR channels without any modification provided that the channel has a rational $f(D)$ (what type of channels have a rational $f(D)$ is discussed in Section 2.6). Assume that $h(t)$ has infinite length, and is square-integrable. The received signal $s(t)$ is still the linear combination of a finite set of square-integrable basis functions $h^{(k)}(t)$ (See (2.12) for its definition.). Therefore the sufficient statistics for estimation of the input sequence still may be obtained from a filter matched to $h(t)$. The output sequence of the matched filter is still given by

$$a(D) = x(D)R(D) + n'(D). \tag{2.56}$$

However the matched filter can only be approximately implemented, since theoretically an accurate implementation requires infinite delay.

Suppose the matched filter is implemented with enough delay, so that the loss of optimality is negligible. Then the remaining problem is to find a whitening filter. This is discussed below.

The sampled autocorrelation function of $h(t)$ exists in the region $(-\infty, \infty)$ and is defined as

$$R_k \triangleq \int_{-\infty}^{\infty} h(t)h(t-kT)dt, k = 0, \pm 1, \pm 2, \ldots. \tag{2.57}$$

The D-transform of the R_k's is

$$R(D) \triangleq \sum_{k=-\infty}^{\infty} R_k D^k, \tag{2.58}$$

a polynomial of infinite degree. In general $R(D)$ might not have a closed form expression. However for a rational channel, i.e., $h(t)$ has a rational Laplace transform $H(s)$, $R(D)$ is always rational and factorable[67], therefore

$$R(D) = f(D)f(D^{-1}) \tag{2.59}$$

where

$$f(D) = \frac{N(D)}{D(D)}. \tag{2.60}$$

The denominator $D(D)$ and the numerator $N(D)$ are finite real polynomials of D. The argument in Section 2.2 about finite $R(D)$ applies, i.e., the roots are in reciprocal pairs. Therefore it is possible to group all the roots of the numerator of $R(D)$ outside the unit circle into $N(D^{-1})$. Then the whitening filter

$$\frac{1}{f(D^{-1})} = \frac{D(D^{-1})}{N(D^{-1})} \tag{2.61}$$

has all the poles outside the unit circle, and hence is stable. Further, factorizing this way, $f(D)$ and $f(D^{-1})$ are also real [67].

Now suppose a real stable $f(D^{-1})$ is obtained. Then $f(D)$ and $f(D^{-1})$ can be always written as

$$f(D) = \frac{\sum_{i=0}^{q} u_i D^i}{1 + \sum_{i=1}^{p} v_i D^i}, \tag{2.62}$$

and

$$f(D^{-1}) = \frac{\sum_{i=0}^{q} u_i D^{-i}}{1 + \sum_{i=1}^{p} v_i D^{-i}}, \tag{2.63}$$

where $q \leq p$. Consequently by taking the inverse D-transform of $z(D) = a(D)/f(D^{-1})$, one obtains

$$z_k = \frac{1}{u_q}\left[a_{k-q} + \sum_{i=1}^{p} v_i a_{k-q+i} - \sum_{j=0}^{q-1} u_j z_{k-q+j}\right] \tag{2.64}$$

Note that since $p - q > 0$, hence $k - q + p > k$, the filter is noncausal. Delay must be introduced to $a(D)$ in the filter for causality. Let

$$z(D) = \frac{D^{p-q} a(D)}{f(D^{-1})} \tag{2.65}$$

Then

$$z_k = \frac{1}{u_q}\left[a_{k-p} + \sum_{i=1}^{p} v_i a_{k-p+i} - \sum_{j=0}^{q-1} u_j z_{k-q+j}\right]. \tag{2.66}$$

Thus the whitening filter is implemented as shown in Fig.2.11, which is causal. As a result the output sequence of the whitening matched filter is

$$z(D) = f'(D)x(D) + n(D) \tag{2.67}$$

where

$$f'(D) = D^{p-q} f(D) \tag{2.68}$$

The output signal sequence is $y'(D) = f'(D)x(D)$, which is a delayed version of $y(D) = f(D)x(D)$. The delay is ignored in order to obtain a more concise discrete model, yet without any effect on all the subsequent results. Taking

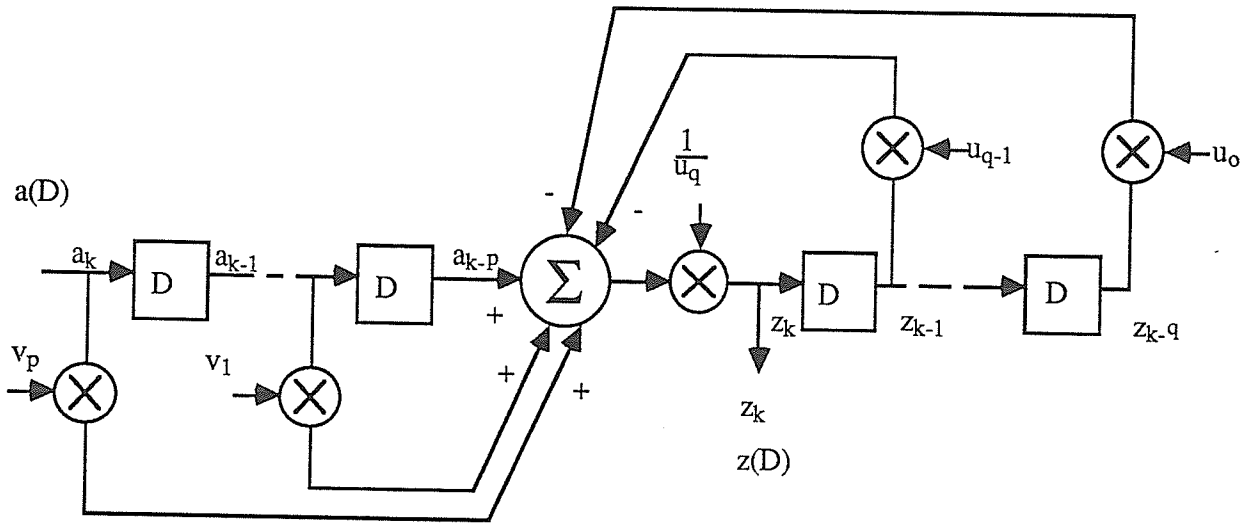

Figure 2.11: Whitening filter for the infinite ISI channel

the inverse D-transform of $y(D)$ one obtains

$$y_k = x_k + \sum_{i=1}^{q} u_i x_{k-i} - \sum_{i=1}^{p} v_i y_{k-i}. \tag{2.69}$$

From this relation the discrete model for rational ISI channel-receiver cascade is shown in Fig.2.12.

From this model it is seen that y_k is a function of the present x_k, previous q inputs x_{k-i} and previous p outputs y_{k-i}. This relationship can also be represented by a tree. Fig.2.13 gives an example for a binary sequence and a channel with

$$f(D) = \frac{1 + u_1 D}{1 + v_1 D + v_2 D^2}. \tag{2.70}$$

If one compares this tree to the tree in Fig.2.8, one observes that the only difference is the output on each branch. Realizing that the data sequence actually is transmitted in *frames*, the infinite ISI affects the channel like finite ISI except that the affected length keeps increasing. It is always equal to the data length, and the ISI component in each transmission interval comes from all the previous transmitted symbols. Therefore for the infinite ISI the metric derivation follows the same steps as in the finite case, and same form of the metric is obtained. That is, the branch metric for an equiprobable m-ary input is still given by (2.50) and $p_z(z_k)$ is still given by (2.45). However y_k is recursively determined by (2.69) instead of (2.36).

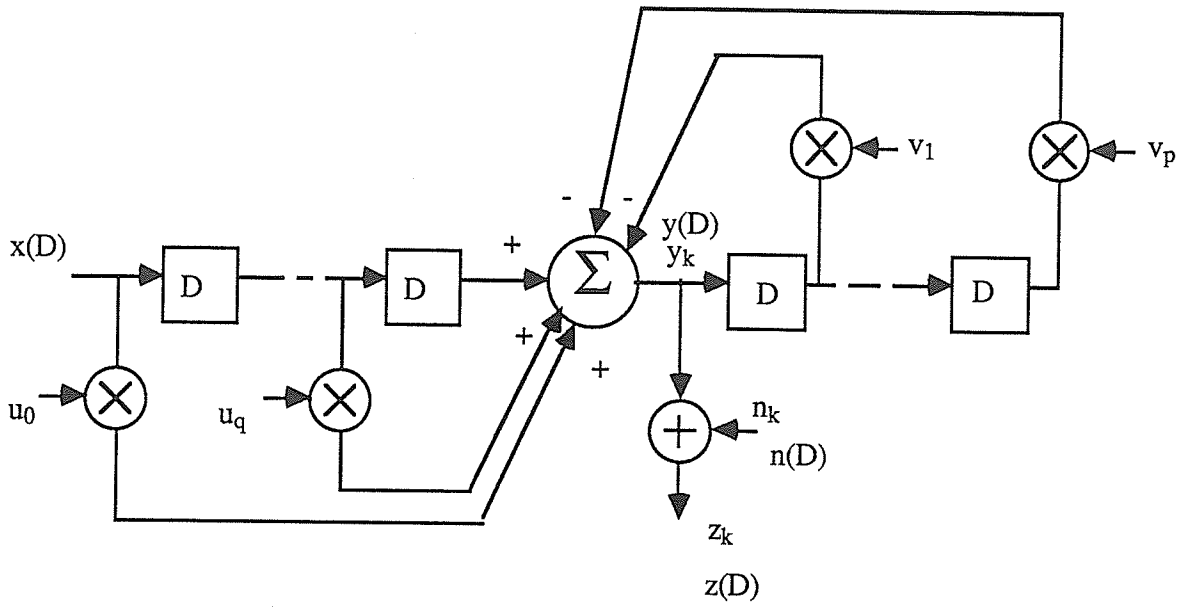

Figure 2.12: Discrete time model for infinite ISI channel

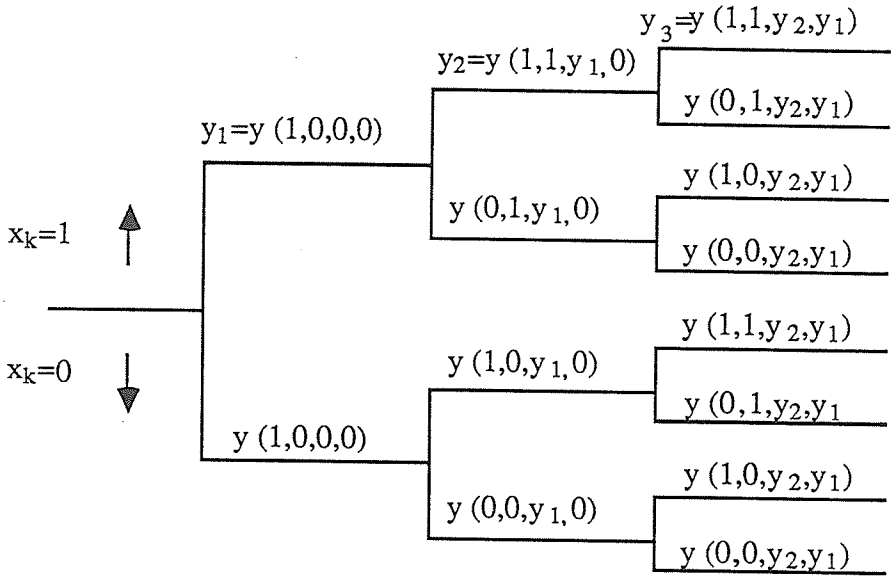

Figure 2.13: Tree diagram for the infinite ISI channel defined by (2.70). Note $y = y(x_k, x_{k-1}, y_{k-1}, y_{k-2})$.

2.5 Calculation of $p_z(z_k)$

To apply the SA to both FIR and IIR channels, one must calculate $p_z(z_k)$ as shown in (2.45). Note that the summation in (2.45) is over all possible values of t_j. In the metric derivation in Section 2.2, though the output symbols in the *random tail* were denoted by t_j to distinguish them from symbols y_j in the *head*, they have an identical probability distribution. Therefore the summation in (2.45) is actually over all possible values of y_j, i.e., the set $\mathcal{Y} \triangleq \{b_1, b_2, \ldots, b_Q\}$, $Q \leq m^L$. Thus, equation (2.45) is rewritten as

$$p_z(z_k) = \sum_{j=1}^{Q} p_n(z_k - b_j) q_j \tag{2.71}$$

2.5.1 $p_z(z_k)$ for FIR Channels

In this case, y_k is a function of sequence $\{x_k, x_{k-1}, \ldots, x_{k-\nu}\}$. For an equiprobable m-ary input, if there are $Q = m^L$ distinct elements $b_j \in \mathcal{Y}$, they are equally likely. Thus

$$p_z(z_k) = \frac{1}{m^L} \sum_{j=1}^{m^L} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(z_k - b_j)^2}{2\sigma^2}\right\}. \tag{2.72}$$

where b_j can be determined quite easily from eqn.(2.36) for each possible sequence $\{x_k, x_{k-1}, \ldots, x_{k-v}\}$. Note that $p_z(z_k)$ is independent of time index k.

If there are less than $Q = m^L$ distinct elements $b_j \in \mathcal{Y}$, this equation is still valid provided that any b_j which corresponds to $n_j > 1$ different input sequences $\{x_k, x_{k-1}, \ldots, x_{k-v}\}$ is used n_j times. In the rest of this thesis, the calculation of $p_z(z_k)$ over a set $\mathcal{Y}$ with nondistinct elements is always interpreted in this way.

For short memory L, $p_z(z_k)$ can be computed in real time by substituting the specific value of z_k into (2.72). When L is large, the case where the application of the sequential estimator is of greatest interest, the computation of $p_z(z_k)$ at each branch is time consuming. In this case one can compute and store $p_z(z_k)$ beforehand and use a table lookup for real time sequential estimation. Since the stored $p_z(z_k)$ can only be discrete, $p_z(z_k)$ obtained from table lookup is an approximation though some kind of interpolation may be used.

2.5.2 $p_z(z_k)$ for IIR Channels

In this case, although the channel has infinite memory, y_k is a function of the input sequence up to time k, i.e., $\{x_k, x_{k-1}, \ldots, x_1\}$, as can be verified by referring to eqn.(2.69). For an equiprobable m-ary input,

$$q(y_k) = \prod_{i=0}^{k-1} Pr(x_{k-i}) = \frac{1}{m^k}. \tag{2.73}$$

Hence from eqn.(2.45)

$$p_z(z_k) = \frac{1}{m^k} \sum_{j=1}^{m^k} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(z_k - b_j)^2}{2\sigma^2}\right\}. \tag{2.74}$$

Note that in contrast to the case of FIR channels, $p_z(z_k)$ is not independent of time index k, because there are $Q = m^k$ elements in $\mathcal{Y}$.

However, since any physical channel has an impulse response that decays, the ISI terms far removed from present input x_k would not have a significant influence on the present output y_k. Therefore eqn. (2.74) can be evaluated by replacing k with a large enough constant L^* such that the accuracy of $p_z(z_k)$ is virtually the same when L^* is increased further. By doing so, the problem of evaluating $p_z(z_k)$ for IIR channels becomes the same as for FIR channels, and $p_z(z_k)$ is independent of time index k.

In both FIR and IIR cases, if L or L^* is unacceptably large, the straightforward evaluation of $p_z(z_k)$ becomes impractical. To deal with this situation, one would rather consider $p_z(z_k)$ as the convolution of the noise pdf $p_n(\cdot)$ and signal pdf $q_y(\cdot)$ (if it exists),

$$p_z(z_k) = \int_{-\infty}^{\infty} p_n(z_k - y) q_y(y) dy. \tag{2.75}$$

Only for a special channel is the limiting pdf $q_y(\cdot)$ analytically tractable[61], namely, the channel with $f(D) = 1 + \frac{1}{2}D + \frac{1}{2^2}D^2 + ...,$. For this channel, $y_k = \sum_{i=0}^{k} f_i x_{k-i}$ has an uniform pdf when $k \to \infty$. However, several numerical estimation methods for $q_y(\cdot)$ are available in the literature[61,62,63].

2.6 Factorization of $R(D)$

As shown in Section 2.1, for finite $h(t)$ and hence finite $R(D)$ factorization is always possible since it simply involves finding the roots of a finite polynomial.

For infinite $h(t)$, however, $R(D)$ is an infinite polynomial and generally it is not clear how to express it, if indeed it is possible to do so, as a ratio of two finite polynomials. This problem is studied in [67]. Here a brief review is given below.

Theorem 2.1: *If $h(t)$ has a Laplace transform $H(s)$ that is rational and stable, then $R(D)$ is rational and can be factored.*

In Chapter 4, this theorem is extended to the multiple channel system. The complete proof for both theorems is given in Appendix B. The proof of Theorem 2.1 was originally given in [67], though not for the most general case.

Another important fact is that *$f(D)$ always has real coefficients if $R(D)$ is an autocorrelation function i.e., $R(e^{j\omega}) \geq 0$ for any ω.*

This fact guarantees that the multiplication coefficients in the feedback

filter model (Fig. 2.12) are all real, therefore the application of SA to the model is feasible. Sometimes $R(D)$ of an IIR channel has no closed form (e.g. $R(D)$ of the Lorentzian magnetic recording channel, which is discussed in the next section.). In this case, truncation may be used to yield a finite $R_T(D)$ which approximates the original $R(D)$. However, caution must be used to ensure that the truncation results in a valid $R_T(D)$, i.e., $R_T(e^{j\omega}) \geq 0$ for any ω. Otherwise a complex $f(D)$ may result. A safer method is to truncate $h(t)$ instead. This always gives a real $f(D)$. Or one can use a rational function to approximate the infinite polynomial $R(D)$ [68].

2.7 Numerical Examples: Channels Chosen for Simulation

Through analysis and simulations, the SA's computational complexities and error probabilities were examined at different signal-to-noise ratios (SNR).

The SNR used here is defined as the signal-to-noise ratio at the output of the finite state machine model or feedback discrete filter model, i.e.,

$$SNR \triangleq \frac{\sigma_y^2}{\sigma^2} = \frac{\sigma_x^2 \|f\|^2}{\sigma^2}, \tag{2.76}$$

where σ_x^2 is the input signal variance $(m^2 - 1)/12$. $\|f\|^2$ is the squared Euclidean norm of $f(D)$ representing the energy in the unit sample response $f(D)$:

$$\|f\|^2 \triangleq \sum_{i=0}^{\nu} f_i^2 = R_0, \tag{2.77}$$

where ν could be finite or infinite.

Three channel models were chosen for analysis and computer simulations.

Channel model 1: partial response channel

A partial response channel with $f(D) = 1 - D$ used by Forney[30] was chosen. This channel is already discrete, hence no whitened matched filter and factorization of $R(D)$ is needed. Despite the fact that there is no need to apply the SA to this channel since MLSE-VA can handle the two-state trellis easily, this channel was chosen to show that because of its good distance distribution, simulated error performance agrees with the prediction very well (see Chapter 3).

Channel model 2: one-pole channel

The impulse response of the one-pole channel is

$$h(t) = \begin{cases} be^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{2.78}$$

where $a > 0, b > 0$. It is chosen because it is a slowly decaying channel when a is small, where the sequential algorithm can exhibit its potential advantage for computational complexity.

The autocorrelation function of $h(t)$ is

$$R_k = \frac{b^2}{2a} A^{|k|}, \tag{2.79}$$

where $A = e^{-aT} < 1$, T is the transmission period. In D-transform

$$R(D) = \frac{b^2(1 - A^2)}{2a} \frac{1}{1 - AD} \frac{1}{1 - AD^{-1}}. \tag{2.80}$$

The factorization of the $R(D)$ is obvious. By choosing a proper constant b to make $b^2(1-A^2)/2a = 1$, one obtains $R(D) = f(D)f(D^{-1})$, with

$$f(D) = \frac{1}{1-AD} = 1 + AD + A^2D^2 + \ldots. \tag{2.81}$$

Thus taking the inverse D-transform of $y(D) = f(D)x(D) = x(D)/(1-AD)$ one has

$$y_k = x_k + Ay_{k-1}. \tag{2.82}$$

Eqn.(2.82) gives the recursive expression for updating the output signal y_k, which is needed in calculating the branch metric (eqn.(2.50)).

For the purpose of comparison, the SA also was applied to the truncated $f(D)$,

$$f(D) = 1 + AD + A^2D^2 + \ldots + A^\nu D^\nu, \tag{2.83}$$

where ν was chosen to have different values in the simulation. Values of f_k for $k = 0, 1, 2, \ldots, 13$ are shown in Fig.2.14, where $T = 1/2a$, $A = e^{-\frac{1}{2}}$.

Channel model 3: magnetic recording channel

The impulse response of a saturation magnetic recording channel may be taken to be [64]

$$h(t) = g(t) - g(t-T), \tag{2.84}$$

where

$$g(t) = \left[1 + \left(\frac{t}{\tau}\right)^2\right]^{-1}, \quad -\infty < t < \infty \tag{2.85}$$

is called the Lorentzian function. τ is a parameter, called the half width of $g(t)$, at which $g(\tau) = g(0)/2$.

The autocorrelation function of $h(t)$ is

$$R(t) = 2\pi\tau \left[\frac{2}{4 + \left(\frac{t}{\tau}\right)^2} - \frac{1}{4 + \left(\frac{t+T}{\tau}\right)^2} - \frac{1}{4 + \left(\frac{t-T}{\tau}\right)^2} \right]. \tag{2.86}$$

Let $t = kT, k = 1, 2, 3, \ldots$, $T = \tau/l$, l a small integer. Then the discrete autocorrelation function R_k for recording period $T = \tau/l$ is

$$R_k = 2\pi lT \left[\frac{2}{4 + \left(\frac{k}{l}\right)^2} - \frac{1}{4 + \left(\frac{k+1}{l}\right)^2} - \frac{1}{4 + \left(\frac{k-1}{l}\right)^2} \right]. \tag{2.87}$$

To factor $R(D)$, a closed rational form of it is needed. Unfortunately, the closed form of $R(D)$ could not be found, if one exists. Therefore truncation either should be applied to $R(D)$ or $h(t)$. Truncation on $h(t)$ is preferred in the sense that the $R(D)$ based on the truncated $h(t)$ is always valid, therefore a real $f(D)$ is always obtainable, whereas the truncated $R(D)$ may not be valid.

The R_k for the truncated $g(t)$, hence truncated $h(t)$, is found to be

$$R_k = 2R_1(k) - R_1(k+1) - R_1(k-1), \tag{2.88}$$

where

$$R_1(k) = \frac{lT}{4 + \left(\frac{k}{l}\right)^2} \left[\frac{2l}{k} \ln \frac{1 + \eta^2}{1 + \left(\eta - \frac{k}{l}\right)^2} + 2\left(\tan^{-1}\eta + \tan^{-1}(\eta - \frac{k}{l}) \right) \right]. \tag{2.89}$$

where $g(t)$ is truncated so that $g(t) = 0$ for $|t| \geq \eta\tau$, η is an integer. In [65], it is shown that when $\eta = 10$, the resulting error on d_{min} is negligible, less than 0.1% of the $g(t)$ truncated at 100τ. d_{min} is the minimum Euclidean

distance between the channel output sequences corresponding to any pair of input sequences. d_{min} dominates the symbol error probability of a MLSE receiver [30]. Since SA is essentially maximum likelihood, choosing $\eta = 10$ is appropriate here. Note that if $\eta \rightarrow \infty$, then eqn.(2.88) becomes eqn.(2.87). f_k's for $\eta = 10$, $l = 1$ are shown in Fig.2.15. They are the actual coefficients used for simulation.

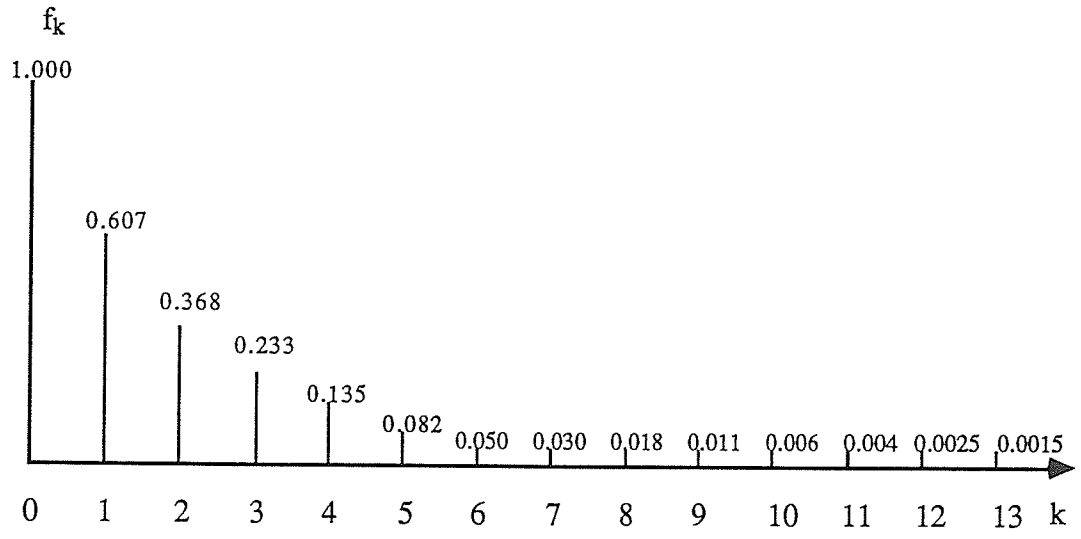

Figure 2.14: f_k of the one-pole channel

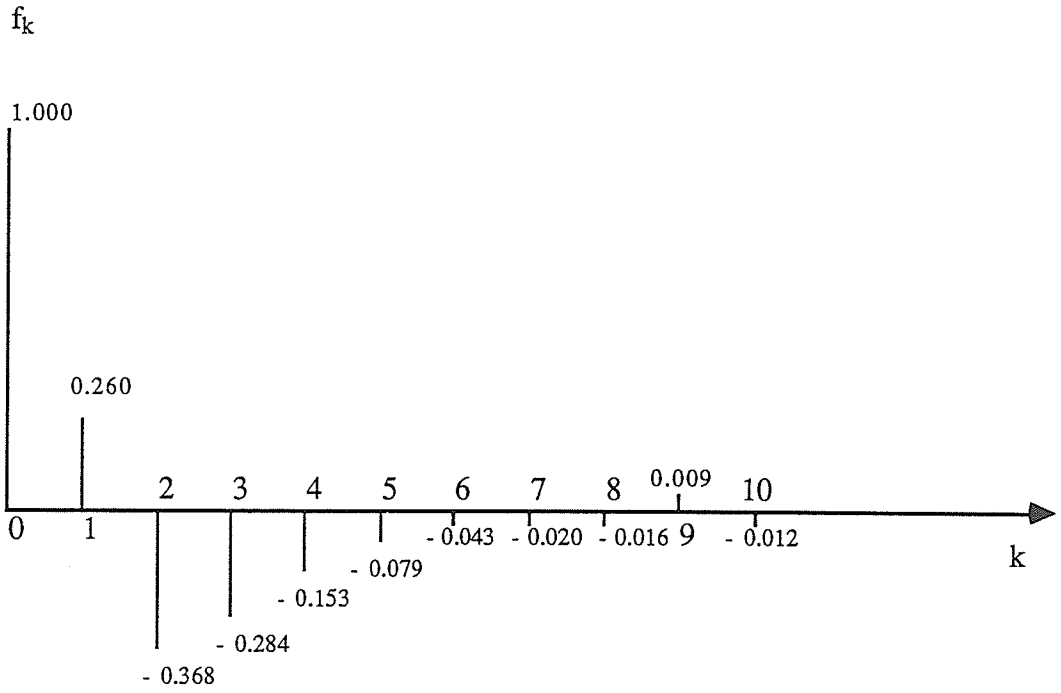

Figure 2.15: f_k of the Lorentzian channel

Chapter 3

Error Performance and Computational Complexity

In this chapter the SA's error performance and computational complexity are discussed first for the FIR channel, then the results are extended to the IIR channel.

As discussed in Section 2.3, sequential sequence estimation for ISI channels can be viewed as a special case of sequential decoding for convolutional codes. Therefore the well known ensemble average performance bounds on symbol error probability and computational complexity [56,34] are applicable. But these bounds do not provide accurate performance evaluation. Bounds on them have been obtained [82] for specific convolutional codes over a BSC. These bounds, however, are not applicable to sequential estimation for ISI channels.

The results of error probability analysis by Forney [30] for the MLSE-VA can be used for SA without degradation in accuracy since the SA is essentially

maximum likelihood. In Section 3.1 bounds on symbol error probability for the MLSE-VA are presented and they are considered valid for the SA too.

The computational complexity evaluation is of particular importance. This is accomplished using the ensemble average results as discussed in Section 3.2.

The specific sequential algorithm chosen in the simulation is the multiple stack algorithm (MSA) which can provide erasurefree decoding and better computational distribution[58]. The features of MSA implementation and simulation details are described in Section 3.3.

Finally in Section 3.4 the simulation results are presented and compared to those of the MLSE-VA and the M-algorithm.

3.1 Symbol Error Probability

Before expressions of symbol error probability are given, several relevant definitions are presented here.

The ν most recent inputs determine a *state* of the finite state machine model in Fig. 2.4. Denote a state s_k at any time k as

$$s_k \triangleq (x_{k-1}, x_{k-2}, \ldots, x_{k-\nu}), \tag{3.1}$$

then the state sequence is defined as

$$s(D) \triangleq \sum_k s_k D^k. \tag{3.2}$$

Let $\hat{x}(D)$ and $\hat{y}(D)$ stand for the estimated input sequence and signal sequence. An *error event* $\mathcal{E}$ is said to extend from time k_1 to k_2 if the estimated state sequence $\hat{s}(D)$ is equal to the correct state sequence $s(D)$ at times k_1 and k_2, but nowhere in between. The length of the error event is $n \triangleq k_2 - k_1 - 1$. Clearly $n \geq \nu$, with no upper bound. The *input error sequence* and *signal error sequence* associated with the error event are

$$\varepsilon_x(D) \triangleq (x_{k_1} - \hat{x}_{k_1}) + (x_{k_1+1} - \hat{x}_{k_1+1})D + \ldots + (x_{k_2-\nu-1} - \hat{x}_{k_2-\nu-1})D^{n-\nu} \quad (3.3)$$

$$\varepsilon_y(D) \triangleq (y_{k_1} - \hat{y}_{k_1}) + (y_{k_1+1} - \hat{y}_{k_1+1})D + \ldots + (y_{k_2-1} - \hat{y}_{k_2-1})D^{n} \quad (3.4)$$

Since $y(D) = x(D)f(D)$ and $\hat{y}(D) = \hat{x}(D)f(D)$, it follows that

$$\varepsilon_y(D) = \varepsilon_x(D)f(D). \quad (3.5)$$

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

$$\begin{aligned} d^2(\mathcal{E}) &\triangleq \|\varepsilon_y\|^2 \\ &= \sum_{i=0}^{n} \varepsilon_{y_i}^2 \\ &= [\varepsilon_x(D)R(D)\varepsilon_x(D^{-1})]_{D=0}. \end{aligned} \quad (3.6)$$

It also can be viewed as the squared Euclidean distance between the correct signal sequence $y(D)$ and an estimated signal sequence $\hat{y}(D)$ in which only one error event $\mathcal{E}$ occurs. The number of symbol errors in the associated input error sequence is defined as the Hamming weight $W_H(\mathcal{E})$ of the event,

i.e.,

$$W_H(\mathcal{E}) \triangleq \textit{ nonzero coefficients in } \varepsilon_x(D). \tag{3.7}$$

The upper bound of the symbol error probability of the MLSE-VA is given by,

$$Pr(e) \leq \sum_{d \in D} Q\left[\frac{d}{2\sigma}\right] \sum_{\mathcal{E} \in E_d} W_H(\mathcal{E}) \left[\prod_{i=0}^{n-\nu} \frac{m - |\varepsilon_{x_i}|}{m}\right]. \tag{3.8}$$

Where D is the set of all possible $d(\mathcal{E})$ and E_d is the subset of error events for which $d(\mathcal{E}) = d$. $Q(\cdot)$ is the error function

$$Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left\{-\frac{y^2}{2}\right\} dy. \tag{3.9}$$

An approximation of eqn.(3.8) is given by

$$Pr(e) \approx K_2 Q\left(\frac{d_{min}}{2\sigma}\right), \tag{3.10}$$

where

$$K_2 \triangleq \sum_{\mathcal{E} \in E_{d_{min}}} \left[W_H(\mathcal{E}) \prod_{i=0}^{n-\nu} \frac{m - |\varepsilon_{x_i}|}{m}\right]. \tag{3.11}$$

This expression is obtained by simply ignoring all the terms in eqn.(3.8) except those involving the minimum distance d_{min} of $d(\mathcal{E})$. This approximation is called an upper estimate. It was shown that the upper bound asymptotically tends to upper estimate when σ goes to zero [66]. When d_{min} is much smaller than the other $d(\mathcal{E})$, the estimate is accurate. An example is the partial response channel $f(D) = 1 - D$ where $d_{min} = \sqrt{2}$ and $d(\mathcal{E}) \geq 2$ if $d(\mathcal{E}) \neq \sqrt{2}$ [30]. However when d_{min} is not much smaller than other $d(\mathcal{E})$, the estimate can be quite poor. As pointed out by Lee and Hill [40], the

simulation results of $Pr(e)$ of the MLSE-VA deviated from that predicated by (3.10) for the truncated one-pole channel ($L = 3$), especially at low SNR.

The distance distributions of the one-pole channel and the Lorentzian channel are listed in Table 3.1. They were obtained by inspection or computer search. As can be seen from Table 3.1, the non-minimum distances for both channels are not ignorable. Therefore one would not be surprised to see fairly large deviations between the upper estimate and the simulation results at low SNR's.

Nevertheless eqn. (3.10) still has reference value for the MLSE-VA *as well as* the SA in terms of error performance prediction.

The lower bound of the symbol error probability of the MLSE-VA is given by

$$Pr(e) \geq K_0 Q\left(\frac{d_{min}}{2\sigma}\right), \tag{3.12}$$

where

$$K_0 \triangleq \sum_{\mathcal{E} \in E_{d_{min}}} \prod_{i=0}^{n-\nu} \frac{m - |\varepsilon_{x_i}|}{m}. \tag{3.13}$$

This is also a lower bound for the SA since SA is not better than MLSE-VA in terms of error performance. Comparing eqn.(3.13) with eqn.(3.11), we can see that if $W_H(\mathcal{E}) = 1$ for all $\mathcal{E} \in E_{d_{min}}$, then $K_0 = K_2$.

So far only the error performance of the estimator for the FIR channel has been considered. For an IIR channel, a finite input error sequence $\varepsilon_x(D)$ would cause an infinite output error sequence since $\varepsilon_y(D) = \varepsilon_x(D)f(D)$, where $f(D)$ is infinite. Except for this difference the entire argument in

One-pole channel (L = 14)	Lorentzian channel (T = τ, η = 10)
$d_1 = 1.116 = d_{min}$	$d_1 = 1.147 = d_{min}$
$d_2 = 1.258$	$d_2 = 1.387$
$d_3 = 1.414$	$d_3 = 1.405$
$d_4 = 1.440$	$d_4 = 1.441$
$d_5 = 1.481$	$d_5 = 1.494$

Table 3.1: Distance distributions of the one-pole channel and the Lorentzian channel

Channels	f(D) = 1-D	0ne-pole	Lorentzian
K_0	2.0	0.5	1.0
K_2	4.0	1.0	1.0
d_{min}	1.414	1.116	1.147
Lower bound	$2Q(\sqrt{SNR})$	$0.5Q(0.887\sqrt{SNR})$	$Q(\sqrt{SNR})$
Upper estimate	$4Q(\sqrt{SNR})$	$Q(0.887\sqrt{SNR})$	$Q(\sqrt{SNR})$

Table 3.2: Pr(e) predictions of the three channels

[30] starting from the concept of error event and ending with eqn.(3.10) and (3.12) are the same for both FIR and IIR channels. Therefore eqns.(3.10) and (3.12) are also valid for the IIR channel where d^2_{min} is still defined as minimum $\|\varepsilon_x(D)f(D)\|^2$ for the infinite $f(D)$. However the degree of $\varepsilon_x(D)$ should not be denoted by $n-\nu$ since n and ν are both infinite this time and $n-\nu$ is undefined. It is better to denote the degree of $\varepsilon_x(D)$ as n_x and rewrite (3.11) and (3.13) as

$$K_2 \triangleq \sum_{\boldsymbol{\varepsilon}\in E_{d_{min}}} \left[W_H(\mathcal{E}) \prod_{i=0}^{n_x} \frac{m-|\varepsilon_{x_i}|}{m} \right] \tag{3.14}$$

and

$$K_0 \triangleq \sum_{\boldsymbol{\varepsilon}\in E_{d_{min}}} \prod_{i=0}^{n_x} \frac{m-|\varepsilon_{x_i}|}{m}. \tag{3.15}$$

These two equations then are valid for both FIR and IIR channels.

It should be pointed out that since the Viterbi algorithm can not be applied to an IIR channel, the validation of above equations for IIR channel has no practical meaning for the VA, whereas it is meaningful for the SA.

Consider now the three channels chosen for simulation. For binary input, d_{min} of the partial response channel $f(D)=1-D$ corresponds to $\varepsilon_x(D)=\pm(1+D+D^2+\ldots+D^{n-1})$, $n\geq 1$, and $d^2_{min}=\|f\|^2=2$. d_{min} of the one-pole channel corresponds to $\varepsilon_x(D)=\pm(1-D)$, and if the channel is truncated at $\nu+1$, then $d^2_{min}=f_0^2+(f_0-f_1)^2+\ldots+(f_{\nu-1}-f_\nu)^2+f_\nu^2$. Let $\nu\rightarrow\infty$, then the limit is the d^2_{min} for the infinite one-pole channel. d_{min} for the Lorentzian channel corresponds to $\varepsilon_x(D)=\pm 1$, and $d^2_{min}=\|f\|^2$.

Parameters K_0, K_2, d_{min}, lower bounds and upper estimates for the three chosen channels for the MLSE-VA and SA were calculated. They are listed in Table 3.2 where the argument of $Q(\cdot)$ has been converted to SNR by referring to eqn.(2.76). Note that the upper estimate for the Lorentzian channel is the same as the lower bound. This again is evidence that the upper estimate is not very accurate in this case.

3.2 Computational Complexity

It is well known that the ensemble average computational distribution of the SA for convolutional decoding is Pareto [34,74,57,56]. It follows immediately from the argument of Section 2.3 that with the receiver structure of Fig. 2.3 the ensemble average computational distribution of the SA for ISI channels is also Pareto. Here the ensemble is the set of all the $1/L$ *time-varying convolutional codes* with constraint length ν, of which the finite state machine model is a member. (See Fig.2.10). If one denotes the number of computations per node as $\bar{C}$, then the average probability distribution of the ensemble satisfies [74]

$$P(\bar{C} \geq N) \approx \Lambda N^{-\rho}, \quad 0 < \rho < \infty \tag{3.16}$$

where Λ is a constant, whose value depends on the version of the SA. ρ is related to the rate $R = r \ln m$ by

$$R = \frac{E_0(\rho)}{\rho}, \; 0 < R < C \tag{3.17}$$

where C is the channel capacity. $E_0(\rho)$ is the Gallager function [72].

$$E_0(\rho) \stackrel{\Delta}{=} \max_{\mathbf{q}} E_0(\rho, \mathbf{q}) \tag{3.18}$$

$$E_0(\rho, \mathbf{q}) \stackrel{\Delta}{=} -\ln \int_{-\infty}^{\infty} \left[\sum_y q(y) p(z/y)^{\frac{1}{1+\rho}} \right]^{1+\rho} dz \tag{3.19}$$

where $y \in \mathcal{Y} = \{b_1, b_2, \ldots, b_Q\}$ is the output signal of the whitened matched filter with a discrete distribution $\mathbf{q}$. Here the conditional density of z is Gaussian,

$$p(z/y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(z-y)^2}{2\sigma^2} \right\}. \tag{3.20}$$

For binary input output-symmetric channels, which include BSC and binary input AWGN channel, the maximizing distribution $\mathbf{q}$ is the uniform distribution [72,34]. Finding the maximizing distribution in general is difficult and must be performed numerically. Fortunately since $E_0(\rho, \mathbf{q})$ is a convex function, it is guaranteed that any steepest ascent algorithm converges to the maximum. An iterative algorithm for computing $\max_{\mathbf{q}} E_0(\rho, \mathbf{q})$ for DMC has been found by Arimoto [83]. It is not clear whether the algorithm can be adapted to discrete-time, continuous-amplitude channels, such as the AWGN channel.

On the other hand, to verify or disprove an intuitive guess for the maximizing distribution, one can use the theorem by Gallager [72]:

Necessary and sufficient condition on the distribution vector $\mathbf{q}$ *which maximizes* $E_0(\rho, \mathbf{q})$, *for* $\rho \geq 0$, *is*

$$\sum_z p(z/y)^{1/(1+\rho)} \alpha(z, \mathbf{q})^{\rho} \geq \sum_z \alpha(z, \mathbf{q})^{1+\rho} \tag{3.21}$$

for all $y \in \mathcal{Y}$, *where*

$$\alpha(z, \mathbf{q}) = \sum_{y} q(y) p(z/y)^{1/(1+\rho)} \tag{3.22}$$

Note that if the output alphabet z is continuous in amplitude, like the case of this study, the only modification for the above theorem is to replace summations over z by integrations([72],p.322). Using this theorem it is easy to verify that uniform distribution is the maximizing distribution for the BSC and the binary input AWGN channel. However for a Q-ary ($Q \geq 2$) input AWGN channel (current case), it is shown that the uniform distribution is not the maximizing distribution (see Appendix A).

An important parameter of sequential decoding is R_{comp}, called *computational cutoff rate.* It is the rate above which the average number of computations per node $E\{\bar{C}\}$ becomes unbounded[74].

$$R_{comp} \triangleq E_0(1) \triangleq \max_{\mathbf{q}} E_0(1, \mathbf{q}). \tag{3.23}$$

Again finding R_{comp} involves maximizing $E_0(\rho, \mathbf{q})$. While it is possible to find R_{comp} numerically, a lower bound R'_{comp} on R_{comp} is easily found by replacing the maximizing $\mathbf{q}$ with the uniform distribution $\mathbf{q}_u$, i.e.,

$$R'_{comp} \triangleq E_0(1, \mathbf{q}_u) < R_{comp}. \tag{3.24}$$

To derive R'_{comp} consider (3.19). Then

$R'_{comp} \triangleq E_0(1, \mathbf{q}_u)$

$$
\begin{aligned}
&= -\ln \int_{-\infty}^{\infty} \left[\sum_{b} \frac{1}{Q} \sqrt{p(z/b)} \right]^2 dz \\
&= 2\ln Q - \ln \int_{\infty}^{\infty} \left[\sum_{i=1}^{Q} p(z/b_i) + \sum_{i,j=1(i<j)}^{Q} 2\sqrt{p(z/b_i)p(z/b_j)} \right] dz \\
&= 2\ln Q - \ln(Q + I) \qquad (3.25)
\end{aligned}
$$

where

$$
I = 2 \sum_{i,j=1(i<j)}^{Q} \int_{-\infty}^{\infty} \sqrt{p(z/b_i)p(z/b_j)} dz. \qquad (3.26)
$$

Substitute (3.20) into (3.26) to obtain

$$
I = 2 \sum_{i,j=1(i<j)}^{Q} \exp\left\{ -\frac{1}{8\sigma^2} (b_i - b_j)^2 \right\}. \qquad (3.27)
$$

Now define the distance between any two signal points b_i and b_j as

$$
d_{ij} = |b_i - b_j| \qquad (3.28)
$$

then

$$
\begin{aligned}
R'_{comp} &= E_0(1, \mathbf{q}_u) \\
&= 2\ln Q - \ln \left[Q + 2 \sum_{i,j=1(i<j)}^{Q} \exp\left\{ -\frac{d_{ij}^2}{8\sigma^2} \right\} \right]. \qquad (3.29)
\end{aligned}
$$

Only uncoded data sequences are considered in this study. Thus the code rate $r = 1$ and the rate in nats is $R = r \ln m = \ln m$. The requirement that $R < R_{comp}$ can be replaced by $R \leq R'_{comp}$ since $R'_{comp} < R_{comp}$, i.e.,

$$
\ln m \leq 2\ln Q - \ln \left[Q + 2 \sum_{i,j=1(i<j)}^{Q} \exp\left(-\frac{d_{ij}^2}{8\sigma^2} \right) \right]. \qquad (3.30)
$$

This condition can be explicitly expressed in terms of the signal to noise ratio defined in (2.76). Denote the SNR which equates (3.30) as SNR'_{comp}. One then obtains the following theorem:

Theorem 3.1: *If the SNR satisfies the following conditions*

$$SNR > SNR'_{comp} \tag{3.31}$$

$$2\ln Q - \ln\left[Q + 2\sum_{i,j=1(i<j)}^{Q} \exp\left\{-\frac{d_{ij}^2 SNR'_{comp}}{2\|f\|^2}\right\}\right] = \ln m, \tag{3.32}$$

then the average computations per node of the sequential sequence estimator for ISI channel is bounded.

It is illustrative to examine SNR'_{comp} associated with each of the signal sets that arise from the three types of channel used for simulation.(These signal sets can be generated by other channels.) Fig.3.1-3.3 show how R'_{comp} changes with SNR, where the input of the channel is binary ($m = 2$). Each SNR'_{comp} is determined by the intersection of the curve R'_{comp} and the horizontal line $R'_{comp} = \ln 2$.

It is seen that R'_{comp} changes considerably with SNR. But it does not change dramatically with L. This is due to the fact that smaller d_{ij}^2's dominate the value of R'_{comp} (see 3.29). An increase in L has no significant effect on small d_{ij}^2's when L is large enough. One might recall that the SA's computational behavior is independent of the constraint length of the codes when the SA is used for decoding convolutional codes. The reason that here R'_{comp} is not completely independent of ISI length L is that the alphabet $\mathcal{Y}$ is

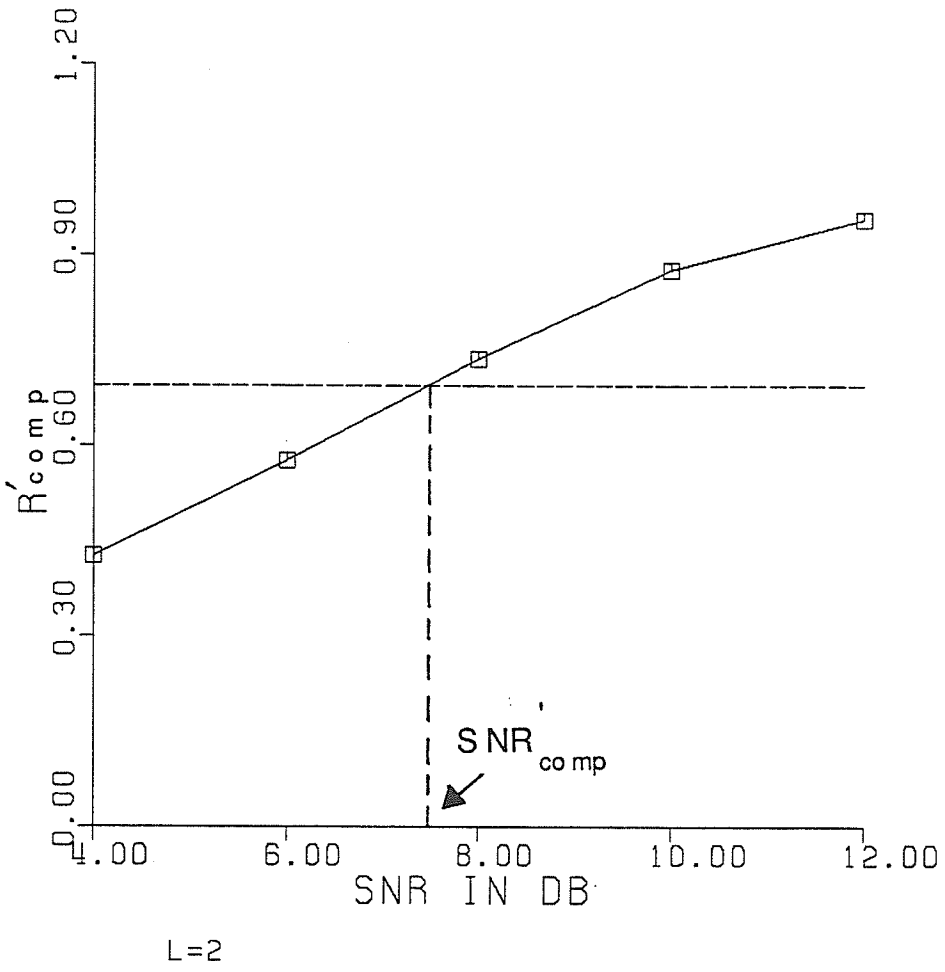

Figure 3.1: SNR'_{comp} of the channel f(D)=1-D

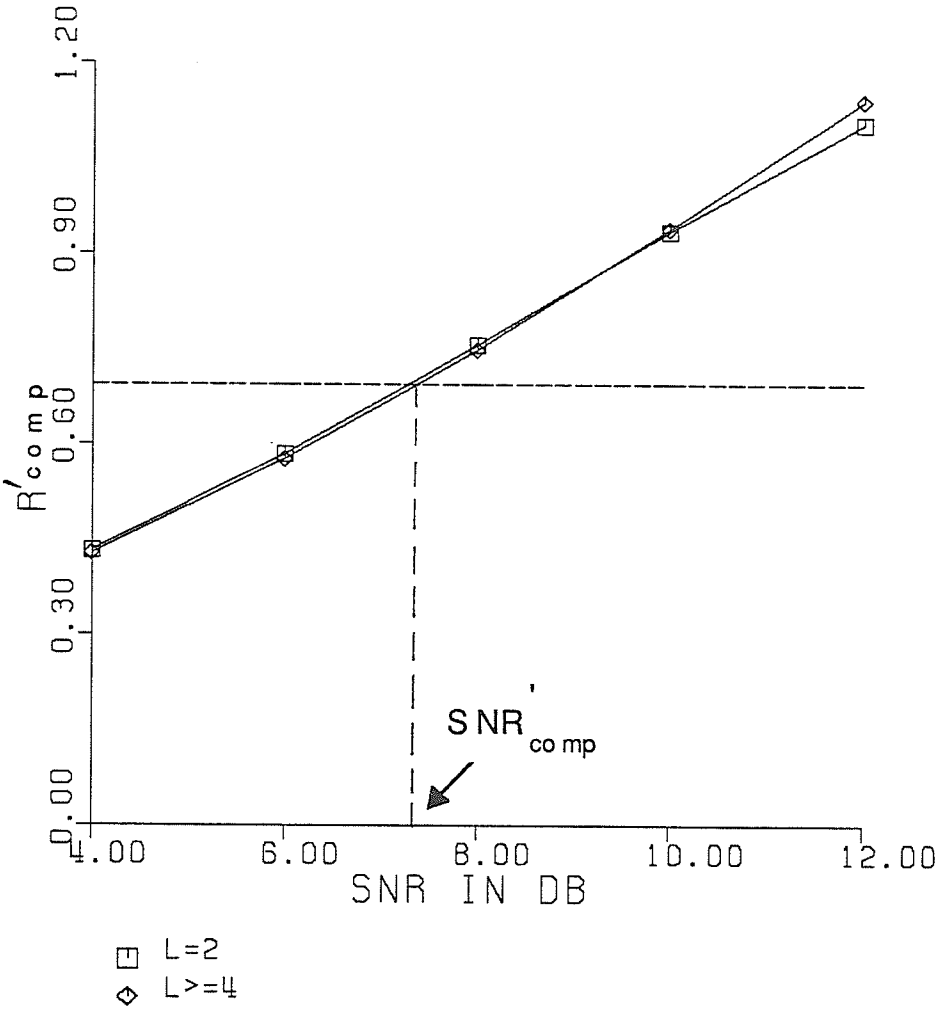

Figure 3.2: SNR'_{comp} of the one-pole channel

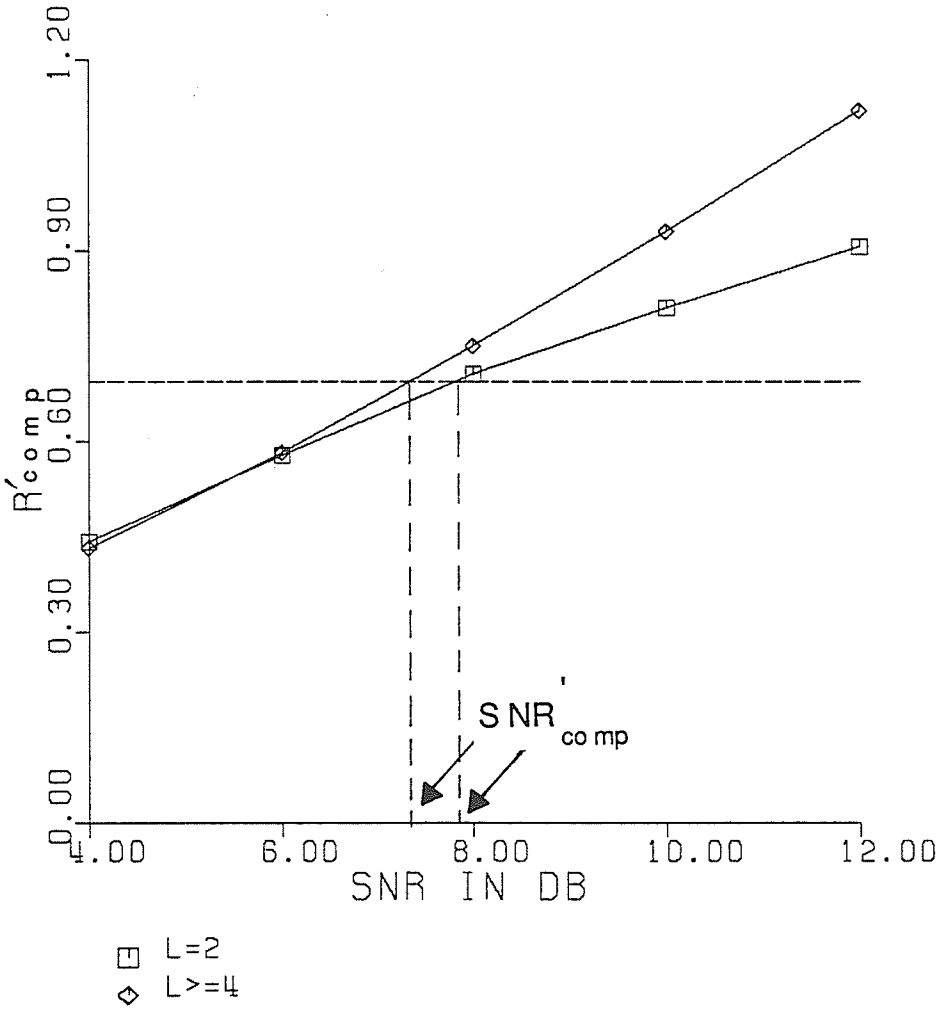

Figure 3.3: SNR'_{comp} of the Lorentzian channel

determined by L. The output alphabet of a convolutional code is independent of its constraint length.

To conclude the discussion on computational complexity the following observations are made:

(1) $SNR'_{comp} > SNR_{comp}$, thus for $E\{\bar{C}\}$ to be bounded, SNR need not be larger than SNR'_{comp}. In other words condition (3.31) is sufficient but not necessary. However since SNR_{comp} can not be found easily, SNR'_{comp} is used as an upper bound on SNR_{comp}.

(2) It should be pointed out that above theorem is only true in the ensemble sense. Thus for a specific ISI channel, although $\mathcal{D}$ can be found and SNR'_{comp} can be calculated from (3.32), it is not guaranteed that if $SNR > SNR'_{comp}$, that $E\{\bar{C}\}$ is bounded. Nonetheless the parameter SNR'_{comp} is still valuable as an approximate indication of the lowest SNR at which the SA will perform satisfactorily.

(3) For an IIR channel the computational behavior of the sequential estimator can be thought as the limiting case of a finite ISI channel. Therefore the computational distribution is still Pareto, and the Theorem 3.1 is still valid for finding the SNR'_{comp}. But now $Q \to \infty$. Therefore when calculating SNR'_{comp} the limit must be taken in eqn.(3.32).

3.3 Multiple Stack Algorithm and Simulation

To verify applicability and performance prediction of the SA and also to evaluate its performance more accurately, computer simulations were done for the three chosen channels. For the purpose of comparison simulations were also done for the Viterbi algorithm and the M-algorithm on the same channels and for the same data sequences. Simulations were based directly on the finite state machine model (for FIR channels) and the feedback discrete model (for IIR channels), without simulating the whitened matched filter of the receiver, since the purpose of the simulations is to examine the sequential algorithm. Simulations were done on an Amdahl V7 main frame computer. The SNR range used in the simulations was 4 to 12 dB, which represents low to moderate signal to noise ratios. The channel input was a binary (0,1) sequence with 512,000 bits, divided into 2000 frames of 256 bits with a tail of L zeros. Although this tail reduced the effective data rate slightly, it enabled reinitialization after each frame and also increased the reliability of the last few bits [55]. For the FIR channel L is the length of the ISI. For the IIR channel L should be a large enough number to assure the feedback discrete model is essentially reinitialized after inputting the tail.

The specific version of the SA used in the simulations was the multiple stack algorithm (MSA). Fig. 3.4 is the flow chart of the MSA. A detailed description of the MSA can be found in [58,70]. By referring to criteria

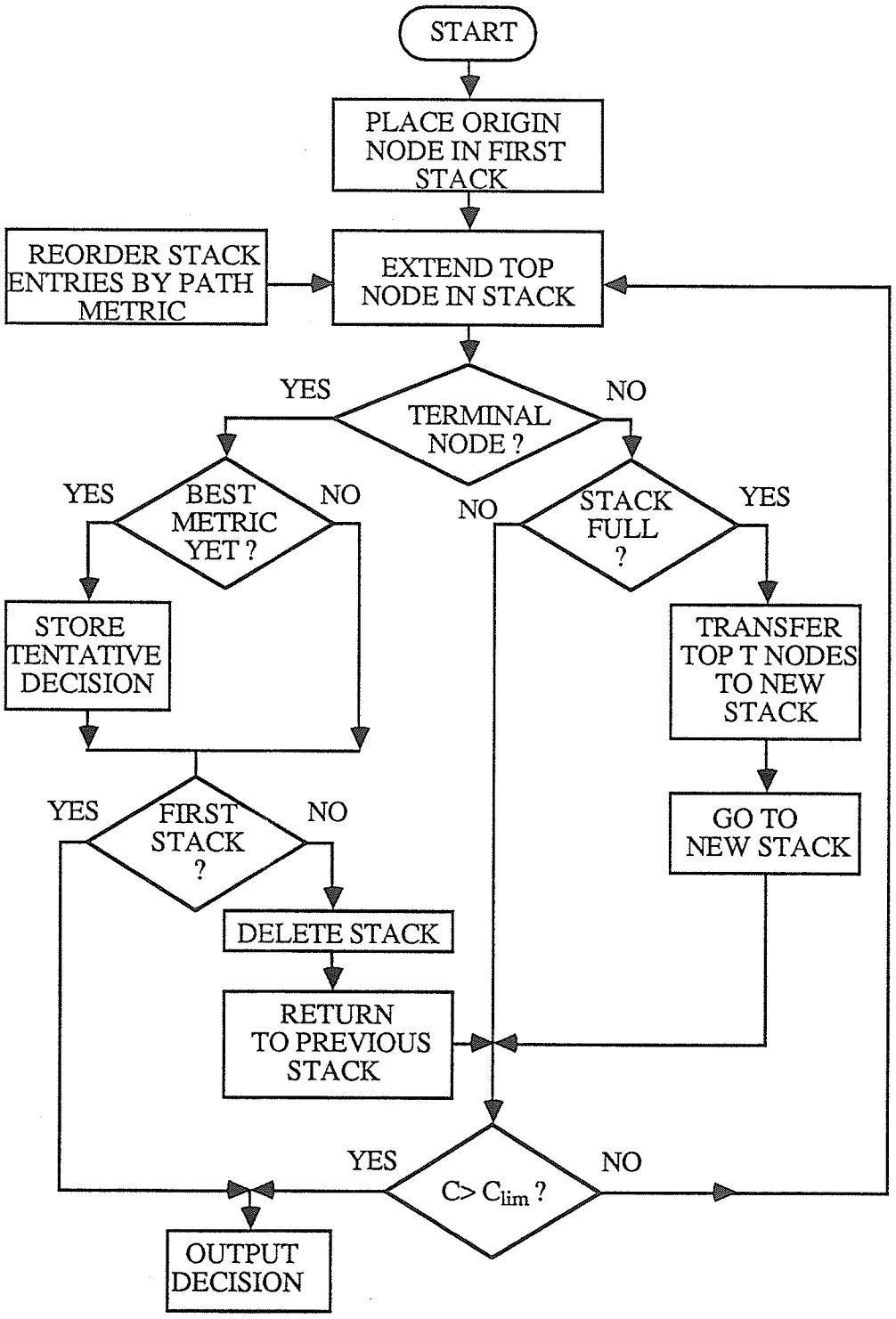

Figure 3.4: Flow chart of the MSA

and empirical parameters reported in [58,70] the parameters chosen for the simulations are listed in Table 3.3.

While straightforward reordering of a stack is conceptually clear, it is not actually done in practice since it is very time-consuming. Therefore the stack-bucket algorithm was suggested [55], in which the contents of the stack are not reordered after each node extension. But the disadvantage of the stack-bucket algorithm is that it is not always the best path that gets extended due to the quantization of the metric values of the paths[74]. In the simulations here, to ensure the best error performance, a search procedure rather than stack-bucket scheme was used to avoid stack reordering. This method finds the *top* entry, i.e., the entry with the largest metric, by comparing the metrics of all the entries in the stack. Once the *top* entry has been located, the entry is extended into two entries. One of them remains in the place of the previous *top* entry with one bit added and the path metric, path length changed. The other is placed in the first available empty entry of the stack if the stack is not full, with only the last bit and metric different from the first one. If the stack is full, according to MSA, the first N_T paths are moved to the next stack. Comparing this method to straightforward reordering a time saving factor of about 100 was observed in the simulations. Although it may take a little longer than the stack-bucket algorithm, it gives the best error performance for SSA or MSA. Moreover, it is also conceptually very simple and easily programmed.

Number of bits of each entry	256
Number of entries of main stack	3000
Number of entries of high order stack	100
Number of high order stacks	15
Number of top entries to be transferred to next stack when current stack is full	3
The computational limit beyond which the algorithm must terminate	5000

Table 3.3: Parameters of the MSA

All the simulation programs, including MSA, VA and M-algorithm and other supporting programs, such as the program for computing $p_z(z_k)$, were written in FORTRAN. The description of the usage of a program is attached at the beginning of the program. They are listed in a separate manual.

3.4 Simulation Results and Comparison

The simulations yielded the following results and observations:

(1) *SA suffers no error performance degradation with respect to MLSE-VA.*

Fig. 3.5 shows $Pr(e)$ simulation results for channel model 1. The simulated $Pr(e)$ of MSA and VA are essentially the same, and both are within

the range delimited by upper estimate and lower bound. This is due to the fact that channel model 1 has a good distance distribution, as pointed out previously.

Performance of the truncated one-pole channel (L=3) in Fig. 3.6 gives further evidence, where simulation results of MSA again agree with the results of MLSE-VA very well. One can see the deviation of the simulation results from the upper estimate. This is due to the poor accuracy of the upper estimate of $Pr(e)$ for this channel.

(2) *SA has the capability to handle long memory channel and even infinite memory channel.*

Simulation results for the infinite ISI one-pole channel are shown in Fig. 3.7. The results for the original one-pole channel ($L = \infty$) and the truncated channel ($L \geq 10$) coincide with each other. Again, there appears to be a degradation from the upper estimate. However as pointed out before this is due to the poor accuracy of the upper estimate. It is asserted with confidence that if VA were simulated (it is too time-consuming with $L \geq 10$ to be simulated), the $Pr(e)$ would be essentially the same as MSA.

However when $L < 10$ there is noticeable degradation of $Pr(e)$ compared to $L \geq 10$, especially at higher SNR. This is due to the fact that ISI terms dominate $Pr(e)$ at higher SNR whereas noise dominates $Pr(e)$ at lower SNR. In Fig. 3.7 results of MSA and VA for L=6 are plotted. Therefore truncating the channel response to a length which is manageable by VA,

say, L=6, would result in performance degradation as well as an increase in computational complexity compared with the SA.

Further for the IIR one-pole channel at each node extension, $y_k = x_k + Ay_{k-1}$ is computed, whereas for the truncated one-pole channel, $y_k = \sum_{i=0}^{L-1} f_i x_{k-i}$, $(L \geq 10)$ must be computed. Although the time consumed in computing y_k is only a small fraction of the entire time used by SA, nevertheless applying SA to original IIR channel does save time and gives a better error performance. The price paid is a small increase in the length of the stack entries that store the y_k's (only y_{k-1} here).

On the other hand, if the channel is a multiple-pole channel, the computations of y_k for the original IIR channel may be more than that for the truncated channel (see eqn.(2.69)). In this case, preference should be given to the latter.

Fig. 3.8 gives the results for the magnetic recording channel. This serves as an example of applying SA to a non-conventional communication channel. Again the error performance of the SA and VA are almost identical.

(3) *Computational complexity of SA is much less than that of VA.*

Table 3.4 gives the average number of computations performed for decoding each symbol by the MSA and VA, where in order to apply the VA, the one-pole channel was truncated to L=10.

Almost no back searches by the MSA were seen for $SNR \geq 10dB$. The reason the average computation is 1.06 instead of 1 is that the tail (14 bits)

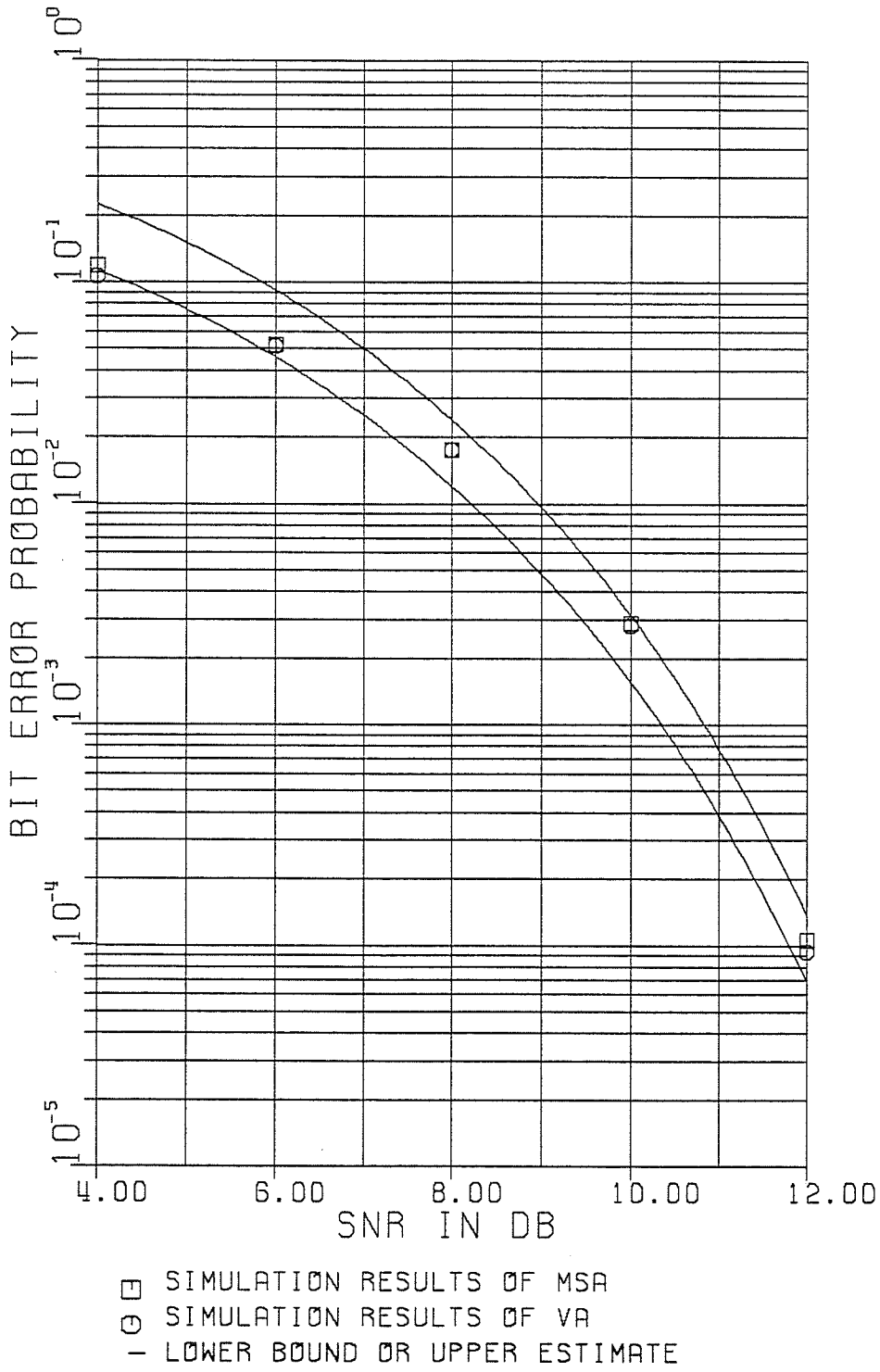

Figure 3.5: Error performance comparison between the MSA and the VA for the channel f(D)=1-D

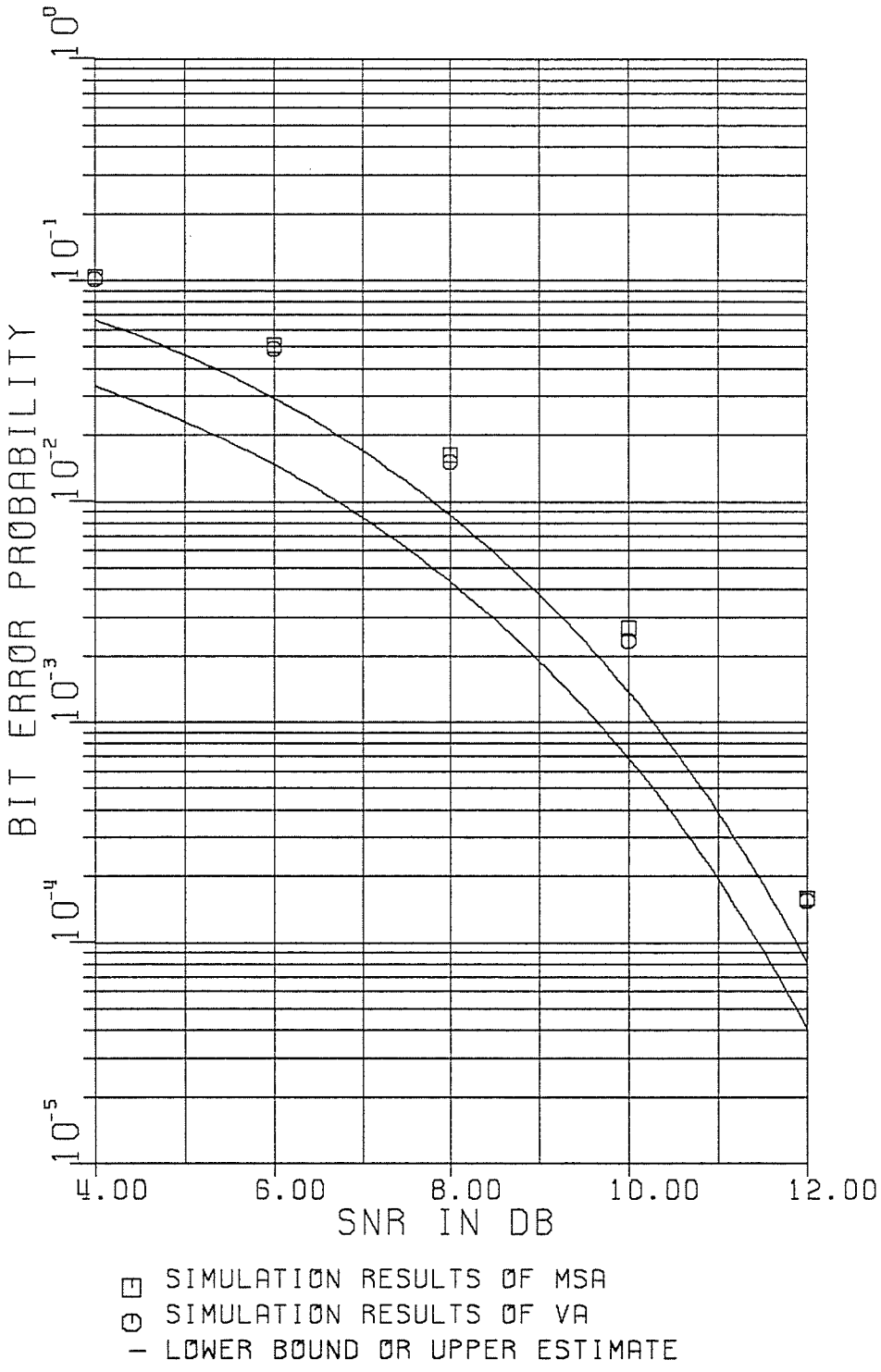

Figure 3.6: Error performance comparison between the MSA and the VA for the truncated one-pole channel

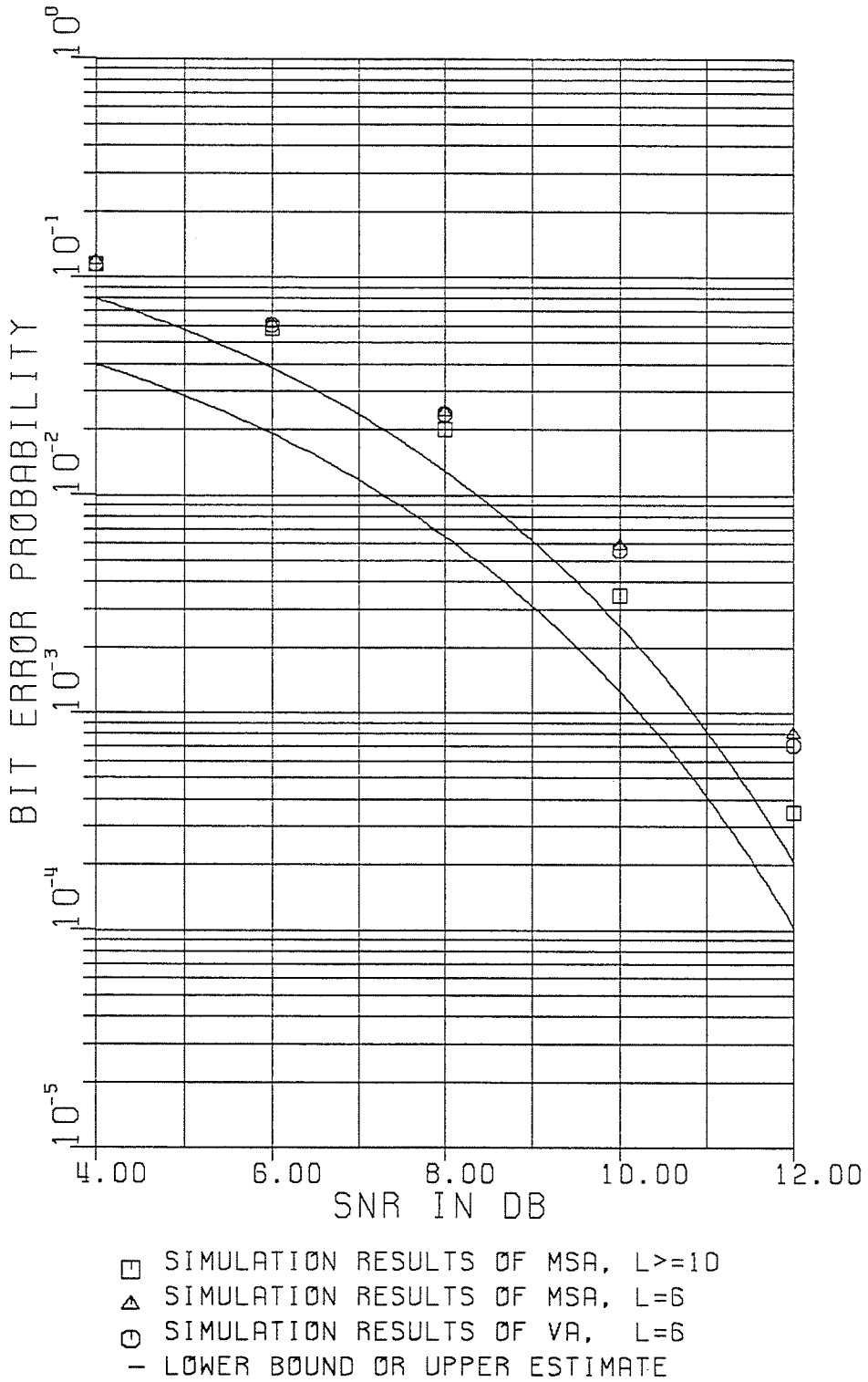

Figure 3.7: Error performance comparison between the MSA and the VA for the one-pole channel for different lengths of truncation

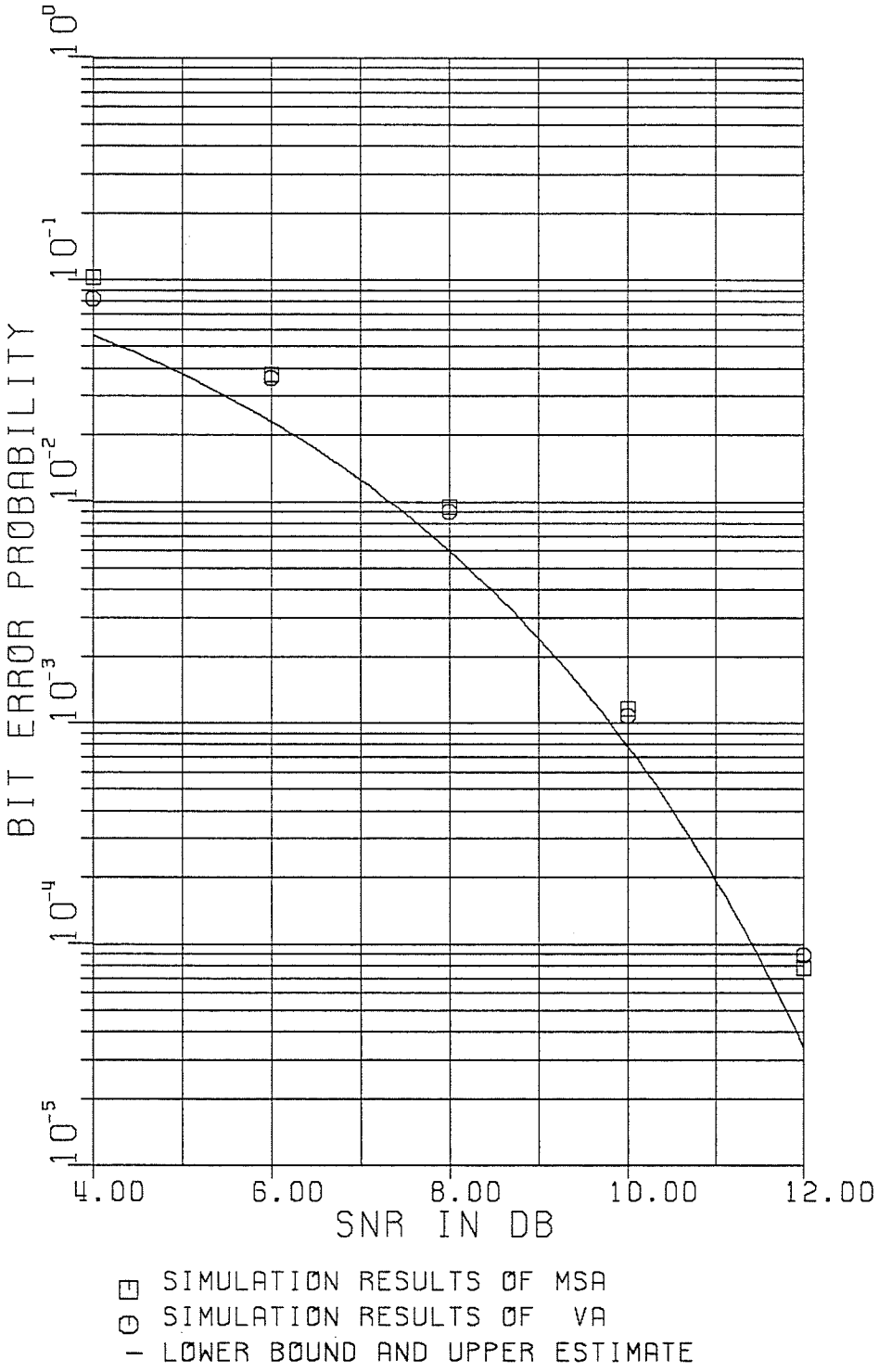

Figure 3.8: Error performance comparison between the MSA and the VA for the Lorentzian channel

SNR	MSA	VA (L=10)
4 dB	14.76	512
6 dB	3.63	512
8 dB	1.20	512
10 dB	1.06	512
12 dB	1.06	512

Table 3.4: Comparison of computations for decoding each input symbol between the MSA and the VA (one-pole channel)

added to each frame increases the average computation slightly.

From the table a large computation saving is observed, ranging from 34 to 487, depending on the signal to noise ratio. However note that the definition of a computation is a node extension. The MSA and VA use different amounts of time for a node extension. The operations needed for each computation in the MSA are more complicated and consume more time. According to [70], where the MSA and VA were implemented with software on a Zilog Z-80 microcomputer, the stack creation/ordering/deletion operation of a node extension of the MSA is about 10 times slower than the add/compare/select operation of a node extension of the VA. If this number is taken as the approximate speed ratio, a time saving factor ranging from 3.4 to 48.7 is obtained for the one-pole channel. Similar time savings are observed in

SNR	Average CPU-time used for one symbol		Ratio
	MSA	VA	
4 dB	0.154 sec.	0.281 sec.	1.82
6 dB	0.041 sec.	0.281 sec.	6.80
8 dB	0.008 sec.	0.281 sec.	35.1
10 dB	0.007 sec.	0.281 sec.	42.6
12 dB	0.007 sec.	0.281 sec.	42.6

Table 3.5: CPU-time comparison between the MSA and the VA

the simulations as shown in Table 3.5, where the CPU time to decode each bit with the MSA and VA are given along with the ratios. It is seen that the time saving ratios range from 1.82 to 42.6. Signal to noise ratios are usually higher than 12 dB in data transmission by voice grade telephone lines and in magnetic recording channels. In these cases, one can expect to have one to two orders of time saving. With the SA one can increase the data rate to the limit where $Pr(e)$ is permissible without worrying about the processing speed. Whereas with a VA one can only increase the data rate to the computational complexity limit.

(4) *Computational distribution*

Fig.3.9 and 3.10 show the simulation results for the computational distribution of the MSA. Since the MSA seldom used the higher order stacks at

$SNR = 6$ dB, the distribution is approximately Pareto. When $SNR = 4$ dB, the MSA used the higher order stacks frequently, the distribution is basically exponential. These results agree with the analysis presented before.

(5) *No erasures in the simulations for* $SNR \geq 4$ *dB*

With the selection of the MSA parameters listed in Table 3.3, no frame erasures were observed in the simulations over the 4-12 dB range for all three channels.

For a $SNR < 4$ dB, however, stack overflows occurred. In this case the MSA decoder took the uncompleted path with the largest metric as the estimate. Surprisingly the $Pr(e)$ did not increase noticeably, i.e., the overflow is *not catastrophic.* Perhaps this is due to the fact that the decoding of the frame was close to the end and the $Pr(e)$ is so high at this SNR that a guess of the undecoded part of the frame did not introduce many more error bits than a decoded one.

Referring to Fig.3.1-Fig.3.3 it is seen that the SNR'_{comp} of each simulated channel is around 7 dB. This fact indicates that working under 7db the SA would *probably* encounter a large computation load. This value of SNR'_{comp} is 3 dB away from 4 dB below which the simulation suffered stack overflows. Again it should be reiterated that the SNR'_{comp} (or SNR_{comp} if it could be found) is the ensemble average, thus it is only a rough indication for a specific ISI channel.

(6) *Comparison with M-algorithm*

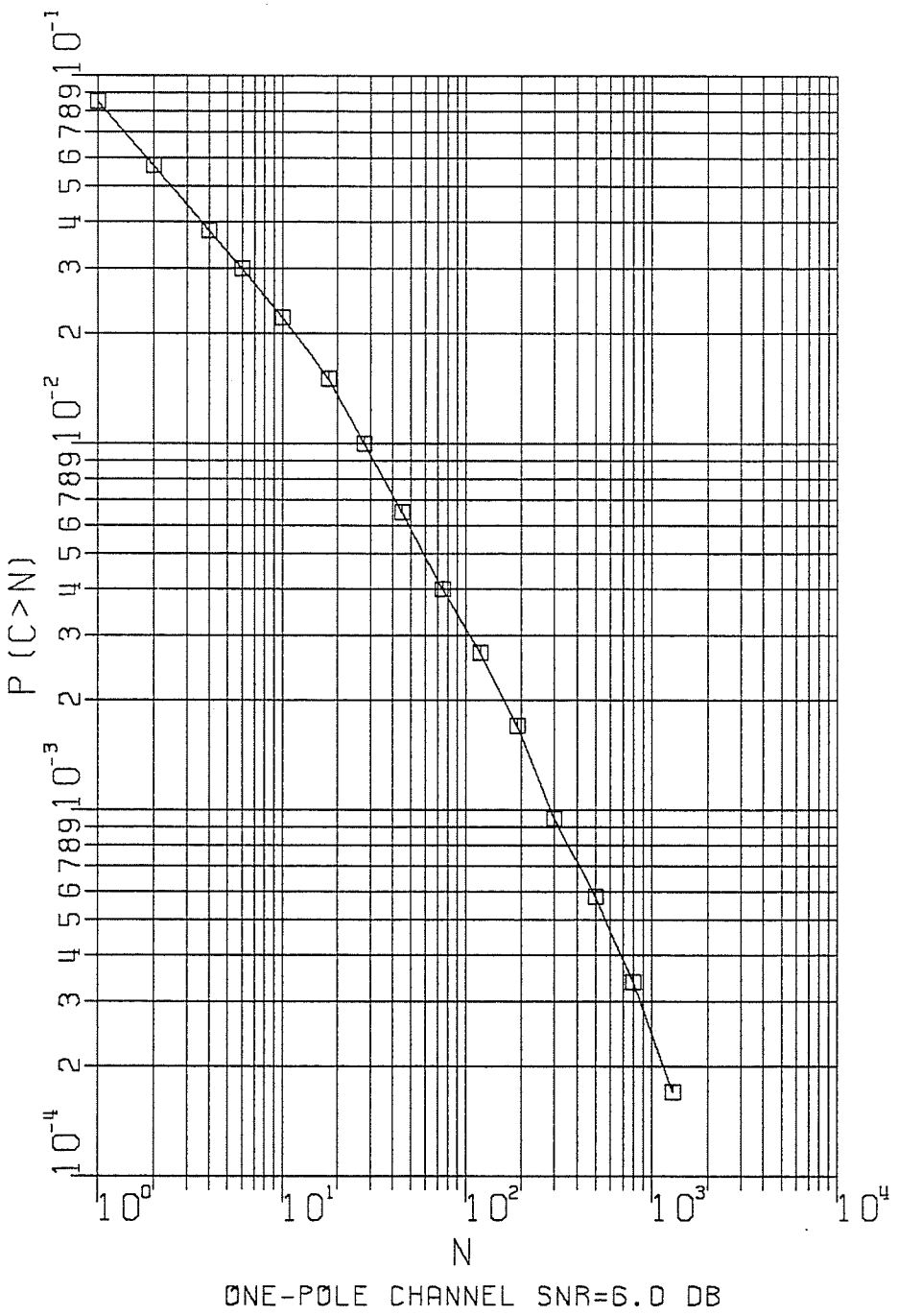

Figure 3.9: Computational distribution of the MSA at SNR=6 dB: Pareto

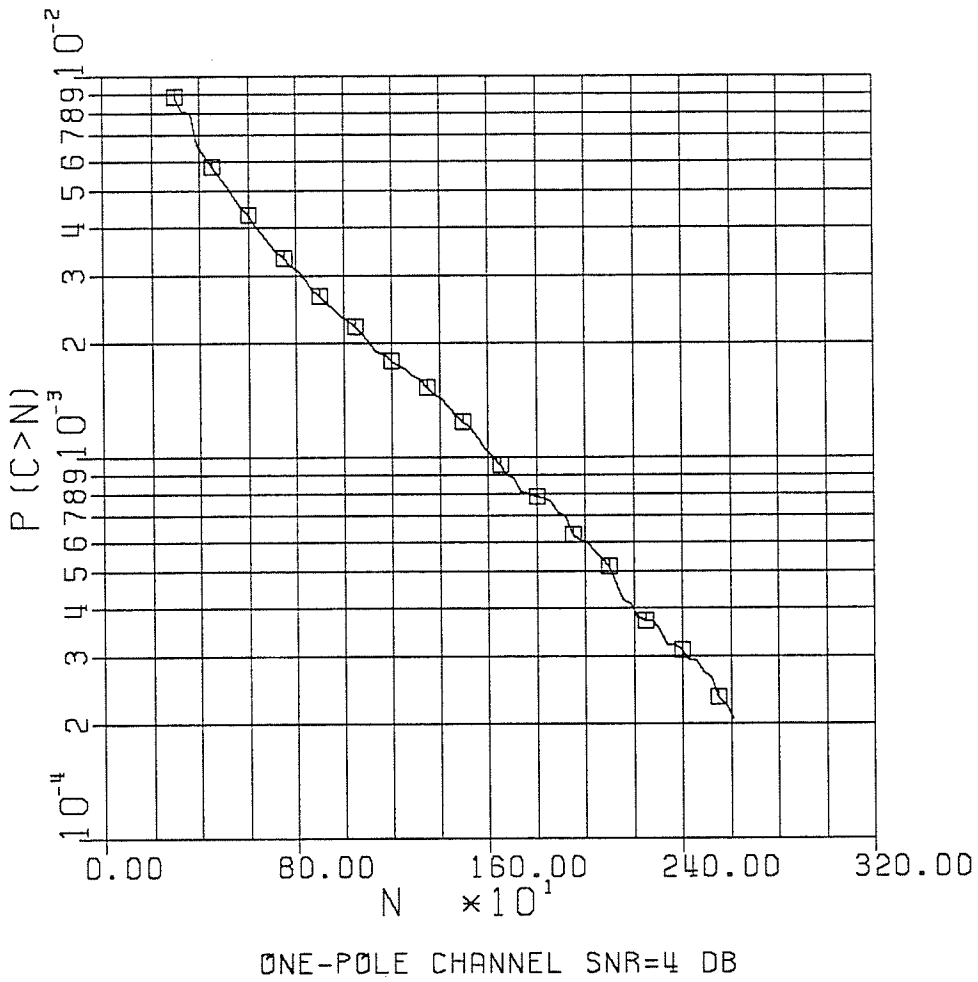

Figure 3.10: Computational distribution of the MSA at SNR=4 dB: Exponential

There are two ways to compare computational complexity of the MSA with the MA. The first way is to compare them for same the $Pr(e)$. This is simple. Since the MSA essentially achieves the same $Pr(e)$ of the VA, in general, the computational complexity of the MA must be very close to that of the VA for same $Pr(e)$ of the MSA. It was seen in the simulations that this is particularly true for higher SNR (8-12 dB). However for lower SNR (4 and 6 dB), the $Pr(e)$ of the MA almost reaches that of the MSA with $M = 32$ instead of 512 for the one-pole channel (L=10), for which the VA must keep 512 survivors all the time. The explanation is that at lower SNR the noise instead of the ISI tail dominates the $Pr(e)$. Therefore the fact that some paths with minimum metric in the ISI trellis are ignored by the M-algorithm does not introduce significant degradation.

The second way is to compare the $Pr(e)$ of the MSA with that of the MA at the same decoding throughput. Multiplying the number in Table 3.4 with the speed factor 10, it is seen that the MA should have M=12 for $SNR = 8, 10, 12$ dB, and $M = 37, 148$ for $SNR = 6, 4$ dB. However it is seen from simulation results that increasing M beyond 32 does not improve $Pr(e)$ noticeably for $SNR = 4, 6$ dB. Thus the comparison is made based on $M = 12$ for $SNR = 8, 10, 12$ dB and $M = 32$ for $SNR = 4, 6$ dB. Since the one-pole channel is infinite in length, the M-algorithm may be designed for the original infinite channel or for a truncated one. Truncation was made in the simulations. Of course the truncation length L must be long enough. But

it is seen from the simulation that L should not be too long. The simulation results are shown in Fig.3.11. Compared to the MSA, if the M-algorithm is designed for $L = 6$, for the same decoding throughput it suffers degradation of 3.0, 1.5 and 1.0 dB at SNR =12,10, and 8 dB respectively and almost no degradation at $SNR < 6$ dB. For $L = 10$, the degradation is larger. The explanation may be as follows. When L increases the channel is more accurately approximated. This leads to some gain in error performance. But the number of states in the trellis becomes larger when L increases. The fixed number of the paths kept by the M-algorithm thus becomes relatively smaller, causing a loss in error performance. These two effects offset each other. Therefore there should be an optimum L for a specific infinite channel. From the simulation it is seen that both the error performance for $L = 5$ and $L = 7$ are worse than that for $L = 6$. Thus the optimum L for the one-pole channel is 6.

(7) *Storage comparison*

To fairly compare complexity the storage requirement of each algorithm should be discussed. The storage requirements of MSA, VA and MA in the simulations are shown in Table 3.6, where the storages of VA and MA could be about 5 times smaller in real-life situation since only about $5L = 50$ bits instead of 256 bits of each survivor need to be kept during decoding. From Table 3.6 it is seen that to achieve the same $Pr(e)$ the SA requires larger storage than the VA at low SNR, but less storage at high SNR, with much

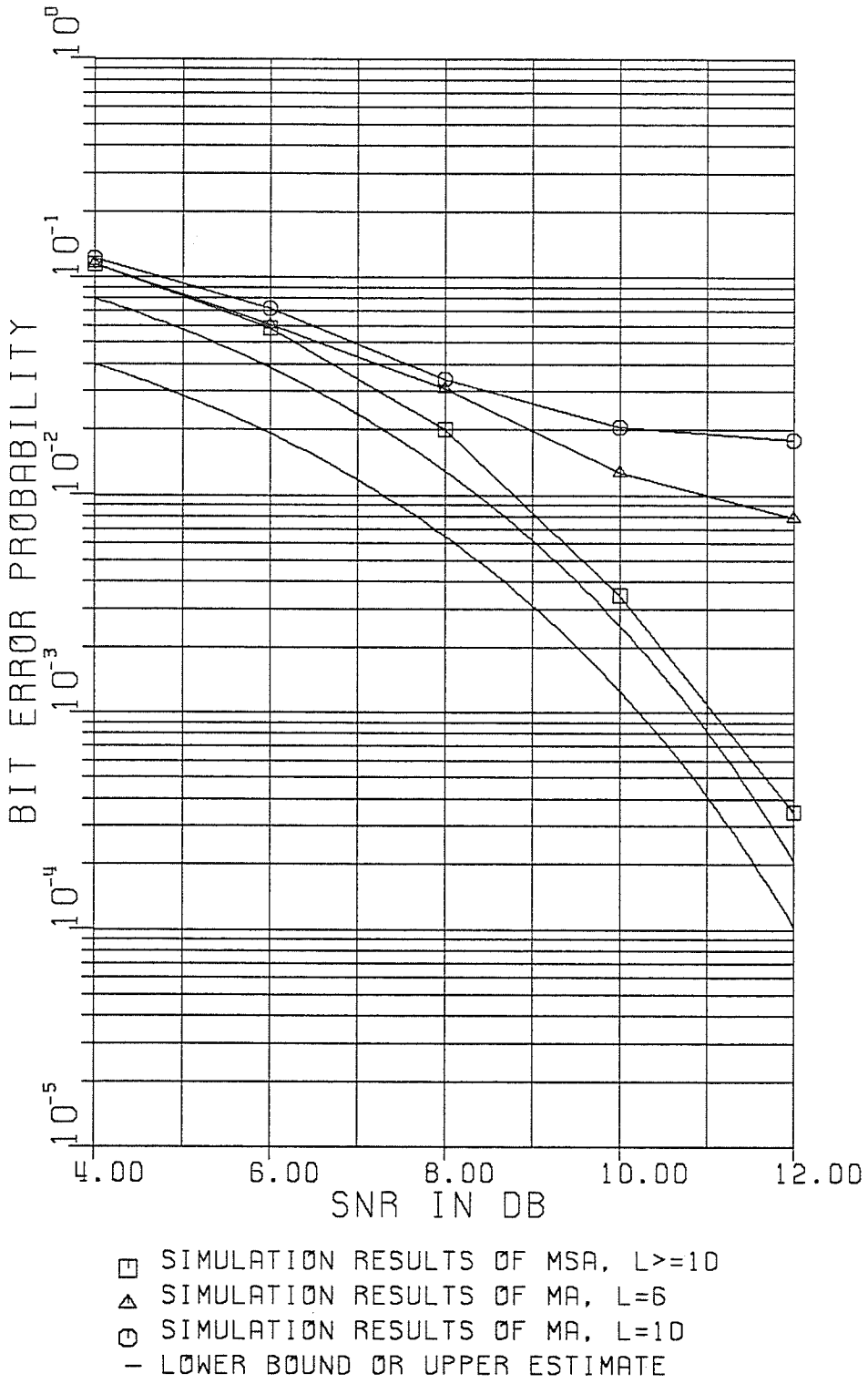

Figure 3.11: Error performance comparison between the MSA and the MA for the one-pole channel

SNR	MSA		VA (k bytes)	MA (k bytes)
	stack entries	k bytes		
4 dB	4500	152	25	1.6 (M=32)
6 dB	1200	41	25	1.6 (M=32)
8 dB	700	24	25	0.6 (M=12)
10 dB	320	11	25	0.6 (M=12)
12 dB	300	10	25	0.6 (M=12)

Table 3.6: Storage comparison between the MSA, VA and the MA

faster decoding speed, of course. Compared to the MA at same decoding speed, the MSA requires much larger storage, but its $Pr(e)$ is significantly better than the MA, as pointed out earlier. Note that storage of MSA can be substantially reduced if the decoder is confined to be used at higher SNR.

A buffer is also required for an SA since the data sequence is sent in frames. The buffer's size depends on C_{lim} and data rate. Suppose for each computation there are μ bits data coming in, then a buffer with size μC_{lim} bits is needed. According to [70], buffer size normally is a small fraction of the stack size.

Chapter 4

Sequential Sequence Estimation for Multiple Channel System with ISI and ICI

In a multiple channel digital transmission system, such as a system with multiwire cable, each channel is used to transmit a data sequence. Besides the intersymbol interference, interchannel interference (ICI) can be a major problem. It has been pointed out by Shnidman[6] that ISI and ICI are essentially identical phenomena. The combined effect of these disturbances can be called multidimensional interference (MDI). Kaye and George worked out an optimal receiver for multiple channel system under the minimum mean-square error criterion[84]. Etten developed an optimum receiver for this system, satisfying both zero-forcing and minimum error probability criterion [85]. Both optimal receivers have the same structure: a multiple matched

filter (MMF) followed by a multiple tapped delay line (MTDL). Later Etten [31] extended Forney's MLSE for a single channel to multiple channel systems. The receiver is a multiple whitened matched filter, which has the same structure as above, except that the whitening filter is a feedback filter. (The whitening filter was erroneously thought to be also an MTDL in [31].) A vector Viterbi algorithm estimator follows to perform MLSE. Based on Etten's receiver model the SA is extended to multiple channel systems in this chapter. The approach used in this chapter is parallel to that used in chapters 2 and 3 for single ISI channels.

4.1 Channel Model and Multiple Whitened Matched Filter

Fig. 4.1 shows the model of an M-dimensional multiple channel system, where $n_j(t)$, $j = 1, 2, ..., M$ are independently identically distributed (i.i.d.) zero mean Gaussian noises with variance σ^2. To each input j a data sequence $\sum_l x_l^{(j)} \delta(t - lT)$ is applied, where $x_l^{(j)}$ are elements of the alphabet $\{0, 1, ..., m-1\}$. Input symbols that are applied at the same instant lT are ordered systematically in the vector form

$$\mathbf{x}_l \triangleq \begin{bmatrix} x_l^{(1)} \\ x_l^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ x_l^{(M)} \end{bmatrix}. \tag{4.1}$$

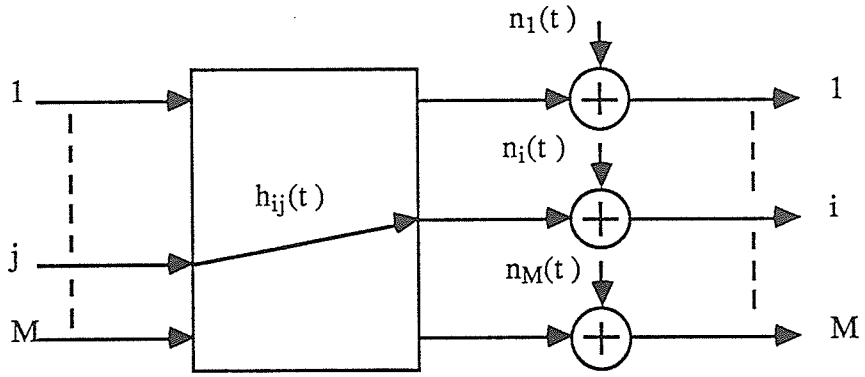

Figure 4.1: Model of a PAM multiple channel system

The D-transform of the input vector sequence is

$$\mathbf{x}(D) \triangleq \sum_{l} \mathbf{x}_l D^l \tag{4.2}$$

The multiple channel system with both ISI and ICI is defined by the impulse response matrix

$$H(t) \triangleq \begin{bmatrix} h_{11}(t) & h_{12}(t) & \dots & h_{1M}(t) \\ h_{21}(t) & h_{22}(t) & \dots & h_{2M}(t) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ h_{M1}(t) & h_{M2}(t) & \dots & h_{MM}(t) \end{bmatrix}. \tag{4.3}$$

All the $h_{ij}(t)$ are assumed square-integrable and of finite length for the time being. The infinite length case is discussed later.

Using an approach parallel with that of Forney[30], Etten developed a multiple whitened matched filter (MWMF)[31], the result is summarized as follows.

Let the chip D-transform of $H(t)$ be $H(D,t)$, which is a matrix with components consisting of the chip D-transforms of the components of $H(t)$. Let the autocorrelation matrix $R(D)$ be factorized as

$$R(D) = F(D^{-1})F'(D). \tag{4.4}$$

A multiple whitened matched filter (MWMF) with chip D-transform

$$W(D,t) = F^{-1}(D^{-1})H'(D^{-1},t) \tag{4.5}$$

in cascade with the channel (Fig.4.2) gives out a vector sequence

$$\begin{aligned} \mathbf{z}(D) &= F'(D)\mathbf{x}(D) + \mathbf{n}(D) \\ &= \mathbf{y}(D) + \mathbf{n}(D) \end{aligned} \tag{4.6}$$

in which $\mathbf{n}(D)$ is a white (in both the discrete time and space dimensions) Gaussian noise vector sequence and $\mathbf{z}(D)$ is a set of sufficient statistics for the estimation of the vector input sequence $\mathbf{x}(D)$.

The autocorrelation matrix is defined as

$$R(D) \triangleq \sum_l R_l D^l. \tag{4.7}$$

Coefficient R_l is an $M \times M$ matrix

$$R_l \triangleq \begin{bmatrix} R_{11}(lT) & R_{12}(lT) & \dots & R_{1M}(lT) \\ R_{21}(lT) & R_{22}(lT) & \dots & R_{2M}(lT) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_{M1}(lT) & R_{M2}(lT) & \dots & R_{MM}(lT) \end{bmatrix} \tag{4.8}$$

and $R_{nj}(lT)$ is determined by the channel,

$$R_{nj}(lT) \triangleq \sum_{i=1}^{M} \int_{-\infty}^{\infty} h_{in}(\tau - lT) h_{ij}(\tau) d\tau. \tag{4.9}$$

There are no problems with implementation of the whitened matched filter since the factorization can be such that both $F^{-1}(D^{-1})$ and $F(D^{-1})$ are stable and nonanticipatory[86,87]. Therefore the MWMF can be implemented as a cascade of an MMF characterized by $H'(D^{-1}, t)$ and an M-dimensional linear filter characterized by $F^{-1}(D^{-1})$. Each element of matrix $F^{-1}(D^{-1})$ is a one-dimensional filter, which can be implemented as a feedback filter as discussed in Section 2.1.

Now from equation (4.5), the entire system of Fig.4.2(a) can be modelled as a multiple finite state machine as in Fig.4.2(b). When F(D) is finite, say, of degree ν, which is the length of ISI of the channel, the signal vector $\mathbf{y}_k$ is determined by the finite input vector sequence $\{\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_{k-\nu}\}$, i.e.,

$$\mathbf{y}_k = y(\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_{k-\nu}). \tag{4.10}$$

The statistical properties of each component of vector $\mathbf{y}_k$ are similar to those of the signal of a single channel whitened matched filter. Since there are a total $Q = m^{ML}$, $L = \nu + 1$ combinations of components of $\mathbf{x}_{k-i}$, $i = 0, 1, \ldots, \nu$, the number of distinct values of each component of vector $\mathbf{y}_k$ is at most $Q = m^{ML}$. When each component of the input vector $\mathbf{x}_k$ is equally likely, and all the possible signal values are distinct, then each component of the signal vector $\mathbf{y}_k$ is also equally likely, with probability $1/Q$.

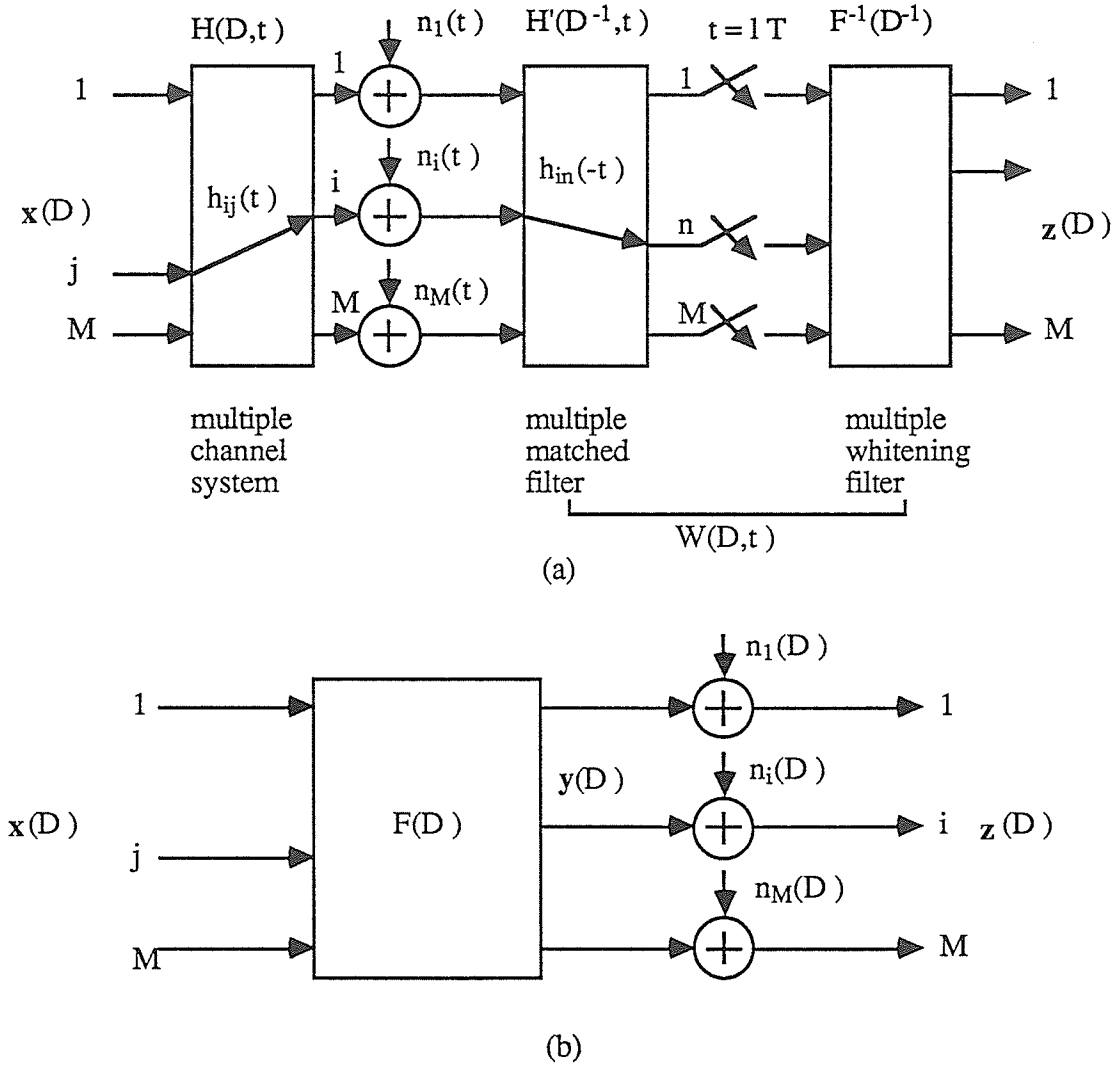

Figure 4.2: (a) Multiple channel system and multiple whitened matched filter (b) finite state machine model

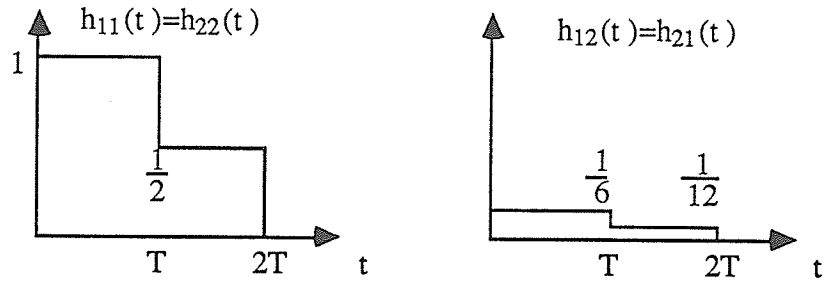

Figure 4.3: An example of multiple channel impulse responses

An example taken from [31], of a multiple channel system with $M = 2$ is shown in Fig.4.3. The $R(D)$ matrix polynomial (or polynomial matrix) is

$$R(D) = R_{-1}D^{-1} + R_0 + R_1 D. \tag{4.11}$$

Letting T=1 one obtains

$$R_0 = \frac{5}{144}\begin{bmatrix} 37 & 12 \\ 12 & 37 \end{bmatrix} \tag{4.12}$$

and

$$R_1 = R_{-1} = \frac{1}{72}\begin{bmatrix} 37 & 12 \\ 12 & 37 \end{bmatrix}. \tag{4.13}$$

The factorization of $R(D)$ is

$$F'(D) = \frac{1}{12}\begin{bmatrix} 6 & 1 \\ 1 & 6 \end{bmatrix}(2 + D) \tag{4.14}$$

The signal vector at the output of the matched filter is

$$\begin{bmatrix} y^{(1)}(D) \\ y^{(2)}(D) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(2+D) & \frac{1}{12}(2+D) \\ \frac{1}{12}(2+D) & \frac{1}{2}(2+D) \end{bmatrix} \begin{bmatrix} x^{(1)}(D) \\ x^{(2)}(D) \end{bmatrix} \tag{4.15}$$

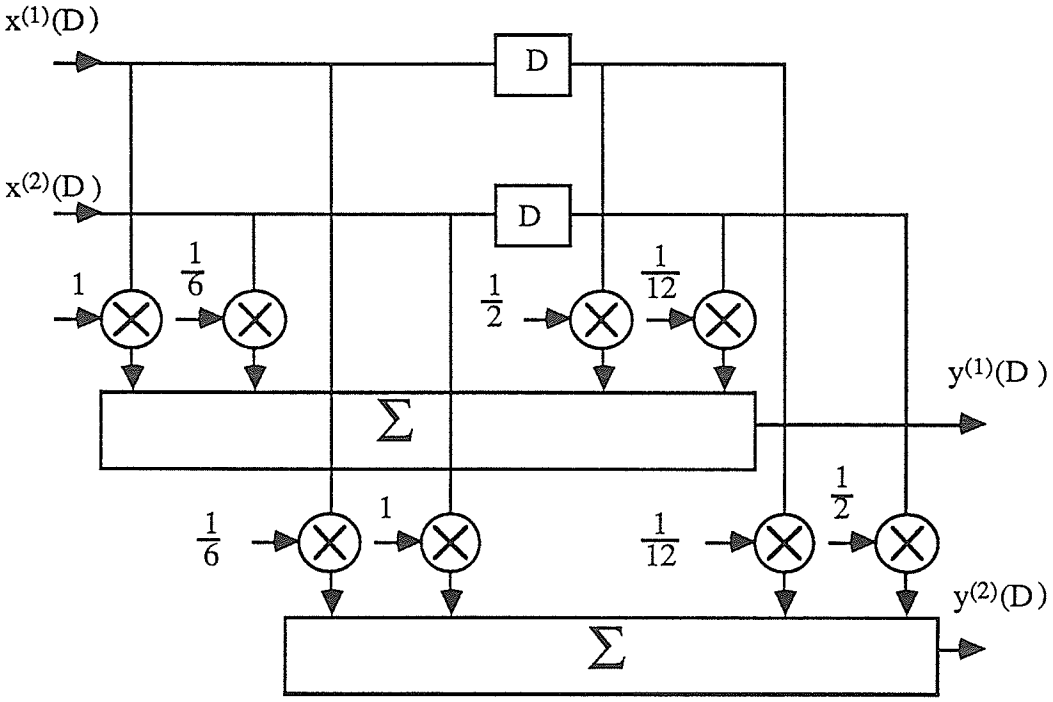

Figure 4.4: Finite state machine model of the example

or in the time domain,

$$\begin{aligned} y_k^{(1)} &= x_k^{(1)} + \frac{1}{2}x_{k-1}^{(1)} + \frac{1}{6}x_k^{(2)} + \frac{1}{12}x_{k-1}^{(2)} \\ y_k^{(2)} &= x_k^{(2)} + \frac{1}{2}x_{k-1}^{(2)} + \frac{1}{6}x_k^{(1)} + \frac{1}{12}x_{k-1}^{(1)} \end{aligned} \tag{4.16}$$

The finite state machine model which reflects the above relation is shown in Fig. 4.4.

The number of states is m^{ML} where m is the input alphabet size, M is the number of channels and L is the length of the ISI in transmission periods covered by the longest $h_{ij}(t)$. The number of states grows exponentially with the product ML. This situation is much worse than the single channel case, where the number of the states is m^L. For instance, with $m = 2, L = 8$, the number of states is $2^8 = 256$ for single channel, which can be handled by Viterbi algorithm without difficulty. But if the system is a two channel system with same length of ISI, then the number of states is $2^{16} = 65536$. The exponential dependence of the number of states on M, the number of channels, makes the sequential algorithm even more attractive as compared to single channel case. In the next section a vector version of the sequential algorithm (VSA) is developed.

4.2 Vector Sequential Algorithm

The input-output relationship of the multiple dimensional finite state machine can be represented by a tree diagram. The tree for Fig. 4.4 is shown

in Fig. 4.5. At each node there are $m^M = 4$ branches, representing the 4 different possible input vectors $\mathbf{x}_l$. Corresponding to these 4 $\mathbf{x}_l$'s. there are 4 different output $\mathbf{y}_l$'s. The sequential algorithm searches through the tree by computing 4 branch metrics at the node and chooses the branch with the largest metric. Any version of the sequential algorithm for decoding, such as stack algorithm, multiple stack algorithm and Fano algorithm, can be applied to the tree, provided a metric is available.

This metric is found by following a procedure similar to that in Section 2.2. If n_l is the length of the l^{th} path $\mathbf{X}_l$ in the tree, $\mathbf{Z}$ is the sufficient statistic vector sequence, then the path metric is

$$L(\mathbf{X}_l, \mathbf{Z}) = \sum_{i=1}^{n_l} L(\mathbf{y}_i, \mathbf{z}_i) \tag{4.17}$$

where

$$L(\mathbf{y}_i, \mathbf{z}_i) = \log \frac{p_{\mathbf{n}}(\mathbf{z}_i - \mathbf{y}_i)}{p_{\mathbf{z}}(\mathbf{z}_i)} + \frac{1}{n_l} \log P_l \tag{4.18}$$

is the branch metric and

$$p_{\mathbf{z}}(\mathbf{z}_i) = \sum_{\mathbf{y}_j} p_{\mathbf{n}}(\mathbf{z}_i - \mathbf{y}_j) Pr(\mathbf{y}_j) \tag{4.19}$$

is the unconditional probability density induced on the sufficient statistic $\mathbf{z}$ and P_l is the probability of sending message vector sequence $\mathbf{X}_l$. Since $\mathbf{n}(D)$ is a white Gaussian noise sequence,

$$p_{\mathbf{n}}(\mathbf{z}_i - \mathbf{y}_j) = \prod_{k=1}^{M} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(z_i^{(k)} - y_j^{(k)})^2}{2\sigma^2} \right\}. \tag{4.20}$$

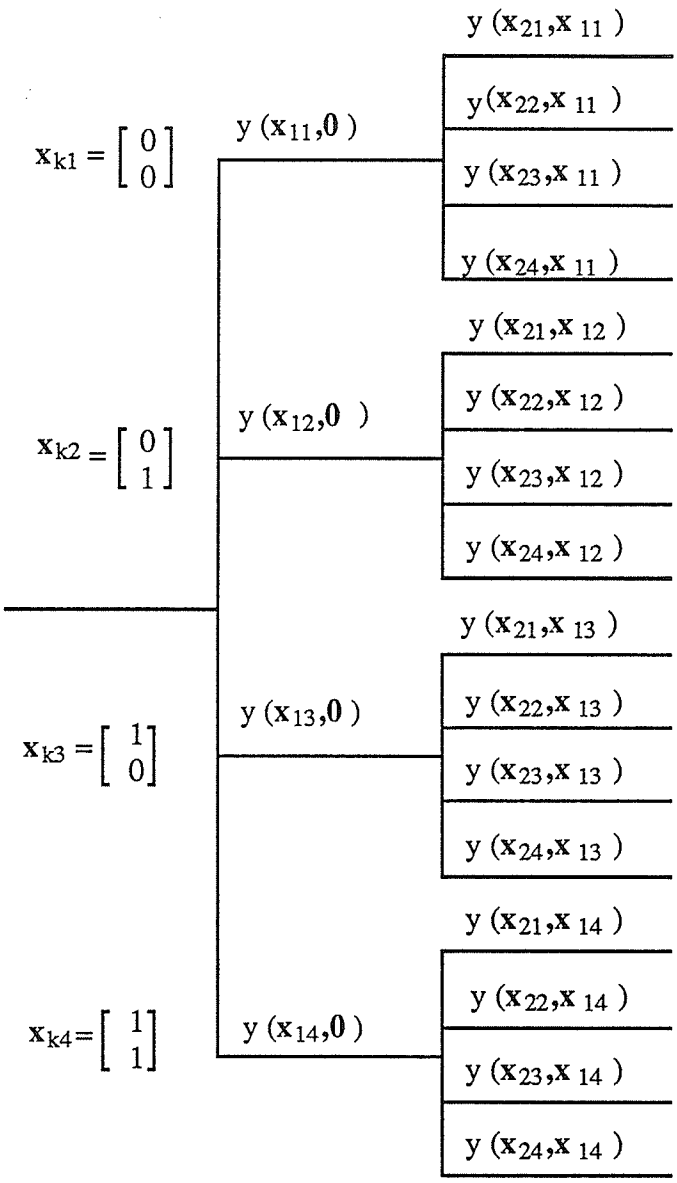

Figure 4.5: Tree diagram for the model in Fig. 4.4

For an equiprobable m-ary input sequence $P_l = 1/m^{Mn_l}$ and $Pr(\mathbf{y}_i) = 1/m^{ML}$. The branch metric becomes

$$L(\mathbf{y}_i, \mathbf{z}_i) = \log \frac{p_{\mathbf{n}}(\mathbf{z}_i - \mathbf{y}_i)}{p_{\mathbf{z}}(\mathbf{z}_i)} - M \log m, \tag{4.21}$$

where

$$p_{\mathbf{z}}(\mathbf{z}_i) = \frac{1}{m^{ML}} \sum_{j=1}^{m^{ML}} p_{\mathbf{n}}(\mathbf{z}_i - \mathbf{y}_j). \tag{4.22}$$

The density $p_{\mathbf{z}}(\mathbf{z}_i)$ can be calculated directly from (4.22) for small m^{ML}. If m^{ML} is unacceptably large for direct calculation, it is preferable to consider $p_{\mathbf{z}}(\mathbf{z}_i)$ as the convolution of the noise pdf $p_{\mathbf{n}}(\cdot)$ and signal pdf $q_{\mathbf{y}}(\cdot)$ (if it exists). The numerical estimation methods for one-dimensional signal pdf in [61,62,63] may be adapted to estimate this multiple-dimensional pdf $q_{\mathbf{y}}(\cdot)$.

Nevertheless $p_{\mathbf{z}}(\mathbf{z}_i)$ has to be calculated or estimated beforehand and stored. For real time decoding $p_{\mathbf{z}}(\mathbf{z}_i)$ is found by table look-up method. The stored $p_{\mathbf{z}}(\mathbf{z}_i)$ can only be discrete, therefore $p_{\mathbf{z}}(\mathbf{z}_i)$ obtained from table look-up is an approximation. Any interpolation method is too time consuming for the multiple vector space of $\mathbf{z}_i$. The simplest method to approximate $p_{\mathbf{z}}(\mathbf{z}_i)$ is to choose the closest point in the space by a minimum Euclidean distance criterion. To make this approximation accurate, the points chosen to be stored should be sufficiently densely distributed.

4.3 Extension to Systems with Infinite MDI

If the multiple channel system has a $H(t) = \{h_{ij}(t)\}$ which is infinite in length, then sequential sequence estimation is still possible, as long as $R(D)$

can be factorized. A theorem similar to Theorem 2.1 is given below:

Theorem 4.1 *If the Laplace transform $H(s)$ of $H(t)$ is rational and stable, then $R(D)$ is rational and can always be factorized.*

The detailed proof is in Appendix B. The outline of the reasoning is simple: $H(s)$ is rational $\rightarrow$ $h_{ij}(t)$ is exponential $\rightarrow$ $R_{nj}(lT)$ is a sum of M exponential functions (eqn.(4.9)) $\rightarrow$ $R(D)$ is rational (see Appendix B). Factorizing rational $R(D)$ is a well solved problem [86,87]. Thus a stable, realizable multiple whitening filter $F^{-1}(D^{-1})$ is obtainable. As in the case of single IIR channel, however, the multiple matched filter needs an infinite delay to be causal. In practice it is implemented approximately by using a finite delay.

The overall system of channel-receiver thus is characterized by $F'(D)$ which can be visualized as a multiple dimensional filter with feedback.

The metric for the vector sequential algorithm is the same as for finite MDI case as shown in (4.18) or (4.21) except that the calculation of $p_{\mathbf{z}}(\mathbf{z}_i)$ involves an infinite number of terms and needs to be approximated by truncation or estimation (see Section 2.5.2).

A two dimensional one-pole channel is presented below to illustrate the application of the vector SA. The impulse response of the channel is

$$H(t) = \begin{bmatrix} \exp(-\frac{2t}{T}) & B\exp(-\frac{2t}{T}) \\ B\exp(-\frac{2t}{T}) & \exp(-\frac{2t}{T}) \end{bmatrix}, \tag{4.23}$$

where $0 < B < 1$ is a constant. Correspondingly

$$R(D) = \frac{1}{(1-AD)(1-AD^{-1})} \begin{bmatrix} 1+B^2 & 2B \\ 2B & 1+B^2 \end{bmatrix}, \tag{4.24}$$

where $A = e^{-2}$ and $T(1-A^2) = 1$ is set. By inspection $R(D)$ is factored to give

$$F(D) = \frac{1}{1-AD} \begin{bmatrix} 1 & B \\ B & 1 \end{bmatrix}. \tag{4.25}$$

The signal vector at the output of the MWMF is

$$\begin{bmatrix} y^{(1)}(D) \\ y^{(2)}(D) \end{bmatrix} = \begin{bmatrix} \frac{1}{1-AD} & \frac{B}{1-AD} \\ \frac{B}{1-AD} & \frac{1}{1-AD} \end{bmatrix} \begin{bmatrix} x^{(1)}(D) \\ x^{(2)}(D) \end{bmatrix}, \tag{4.26}$$

or in the time domain,

$$\begin{aligned} y_k^{(1)} &= x_k^{(1)} + Bx_k^{(2)} + Ay_{k-1}^{(1)} \\ y_k^{(2)} &= x_k^{(2)} + Bx_k^{(1)} + Ay_{k-1}^{(2)}. \end{aligned} \tag{4.27}$$

Note these are recursive equations which can be modeled as a two dimensional feedback filter as shown in Fig.4.6. The vector SA is thus applied to this model, using the metric in (4.21). Each time the branch metric is computed the above equations are used to find $\mathbf{y}_k$ from the current $\mathbf{x}_k$ and the previous output $\mathbf{y}_{k-1}$.

4.4 Performance of the Algorithm

It is difficult to obtain an analytical error probability expression for the vector sequential algorithm. However, recalling the fact that sequential decoding is

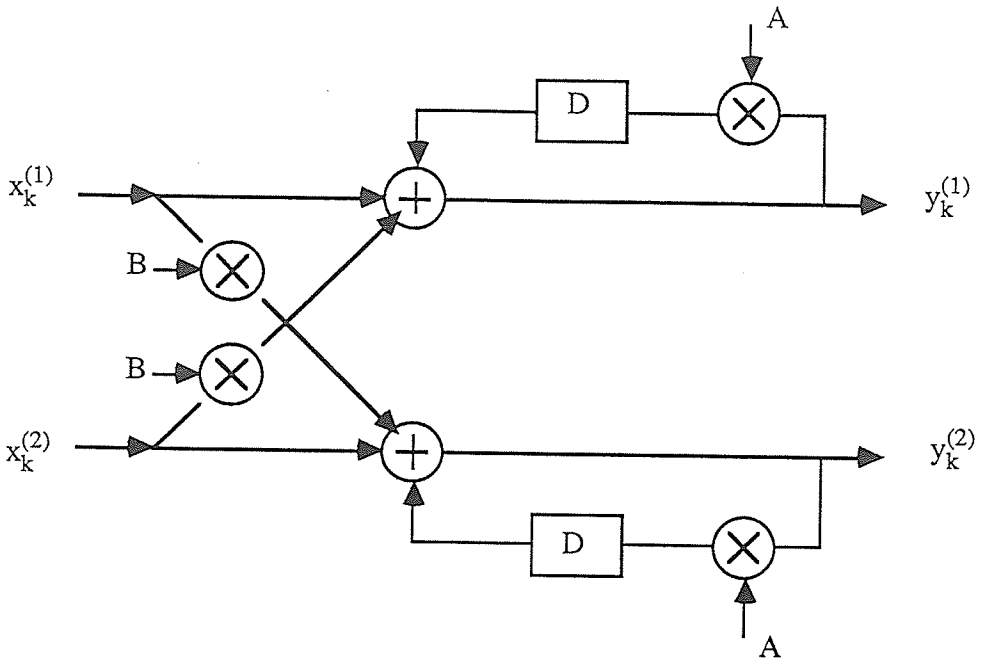

Figure 4.6: Feedback filter model of the 2-dimensional one-pole channel

asymptotically maximum likelihood decoding, one would conjecture that the error performance of the VSA has the same behavior. Therefore the error probability expression for ML estimation developed by Etten [31] can be used for the VSA.

First define the Euclidean norm of any matrix A as

$$\|A\|_2 = (maximum\ \ eigenvalue\ \ of\ \ A^H A)^{\frac{1}{2}}, \tag{4.28}$$

where A^H denotes the Hermitian transpose of A.

Let $\mathbf{x}(D)$ and $\hat{\mathbf{x}}(D)$ denote the transmitted vector sequence and the estimate, then

$$\mathbf{e}(D) \triangleq \hat{\mathbf{x}}(D) - \mathbf{x}(D) \tag{4.29}$$

defines the error vector sequence. Assuming stationarity the starting point of an error event $\mathcal{E}$ can be associated with $t = 0$:

$$\mathcal{E}: \ \mathbf{e}(D) = \mathbf{e}_0 + \mathbf{e}_1 D + \ldots + \mathbf{e}_n D^n \tag{4.30}$$

with $\|\mathbf{e}_i\|_2 \geq d_0$, where d_0 denotes the minimum nonzero value of the Euclidean norm of the vector $\mathbf{e}_i$. This value equals the minimum distance between any two components of a transmitted vector:

$$d_0 = \min_{i \neq j} |x_l^{(i)} - x_l^{(j)}| \tag{4.31}$$

The distance associated with the error event is defined as

$$d^2(\mathcal{E}) \triangleq \|R_0^{-1}\|_2 \sum_{l=0}^{n} \sum_{k=0}^{n} \mathbf{e}'_l R_{l-k} \mathbf{e}_k. \tag{4.32}$$

At moderate or large signal to noise ratios the error probablity is dominated by the error events with minimum distance d_{min}. Let $E_{d_{min}}$ denote the set of error events for which $d(\mathcal{E}) = d_{min}$. Let $W_H(\mathcal{E})$ denote the number of symbol errors associated with error event $\mathcal{E}$. Then the symbol error probability can be estimated by

$$Pr(e) \approx Q\left(\frac{d_{min}}{2\sigma \|R_0^{-1}\|_2^{\frac{1}{2}}}\right) \sum_{\mathcal{E} \in E_{d_{min}}} W_H(\mathcal{E}) \prod_{i=1}^{M} \prod_{l=0}^{n} \frac{m - |e_l^{(i)}|}{m}. \tag{4.33}$$

In [31] Etten argued that under the constraint

$$\|R_0^{-1}\|_2 \sum_{l=-\infty}^{\infty}{}' \|R_l\|_2 \leq 1 \tag{4.34}$$

where $\sum'$ denotes the summation without $\|R_0\|_2$, then $d_{min} \geq d_0$, and d_0 is said to be the distance related to single error events. A single error event

is an error sequence that consists of one error vector, i.e., $\mathbf{e}(D) = \mathbf{e}_0$, and only one component of this vector differs from zero. Thus d_{min} in (4.33) was replaced by d_0 since the single error event has the minimum distance and would dominate the error probability.

However, single error events do not necessarily lead to d_{min}. Furthermore, if they do, $d_{min} \neq d_0$ in general, instead

$$d^2_{min} = \|R_0^{-1}\|_2 \mathbf{e}'_0 R_0 \mathbf{e}_0. \tag{4.35}$$

Therefore replacing d_{min} with d_0 in (4.33) may cause a very large increase in the prediction of $Pr(e)$. Usually d_{min} has to be found by an exhaustive search, which can be done by computer.

As in the single channel case the error prediction equation (4.33) is also applicable to multiple channels with infinite MDI.

For the channel given in Fig.4.3 and the (0,1) alphabet, $d_0 = 1$, d_{min} is found to be 1.216, corresponding to a single error event. $\|R_0^{-1}\|_2 = 1.151$. In this case, the symbol error probability is

$$\begin{aligned} Pr(e) &\approx 2Q\left(\frac{1.133}{2\sigma}\right) \\ &= 2Q(1.133\sqrt{SNR}). \end{aligned} \tag{4.36}$$

where

$$SNR \triangleq \frac{d_0^2}{4\sigma^2} = \frac{1}{4\sigma^2} \tag{4.37}$$

would be the signal to noise ratio if there were no ISI and ICI, i.e., the signal to noise ratio of an isolated single pulse.

In Fig.4.7 the upper estimate of (4.36), the simulation results of the vector VA from Etten and simulation results for the vector MSA are shown. It can be seen that the $Pr(e)$ of the MSA and VA are almost the same, and the simulation results are larger than the prediction. This is not surprising since (4.36) was obtained by ignoring all $Q(\cdot)$ terms with nonminimum distances, though some are quite close to d_{min}. For this example, the first 5 nonminimum distances are 1.33, 1.41, 1.55, 1.60 and 1.64.

For the one-pole multiple channel, if $B = 0.2$ then $d_{min} = 1.275$, corresponding to single error events and adjacent distances are 1.414,1.803,1.904, 2.000,2.208. The $Pr(e)$ prediction is

$$Pr(e) \approx 2Q(1.029\sqrt{SNR}) \tag{4.38}$$

and due to the fact that adjacent distances are quite close to d_{min}, this prediction is smaller than the actual $Pr(e)$.

The prediction and simulation results are shown in Fig.4.8. As expected, the simulation error is larger than predicted. This does not however, imply that the vector SA suffers large degradation in comparison to the vector VA. One can not even apply the VA here due to the infinite MDI. Even simulation can not be done for a truncated $H(t)$ with adequate length (e.g.,L=10 like for single one-pole channel) for the state number is too large to be simulated. (For L=10 M=2, m=2, the number of states is $2^{18} = 262,144$.)

Since the vector sequential algorithm operates in the same way as an ordinary sequential algorithm except that the branch metric is computed

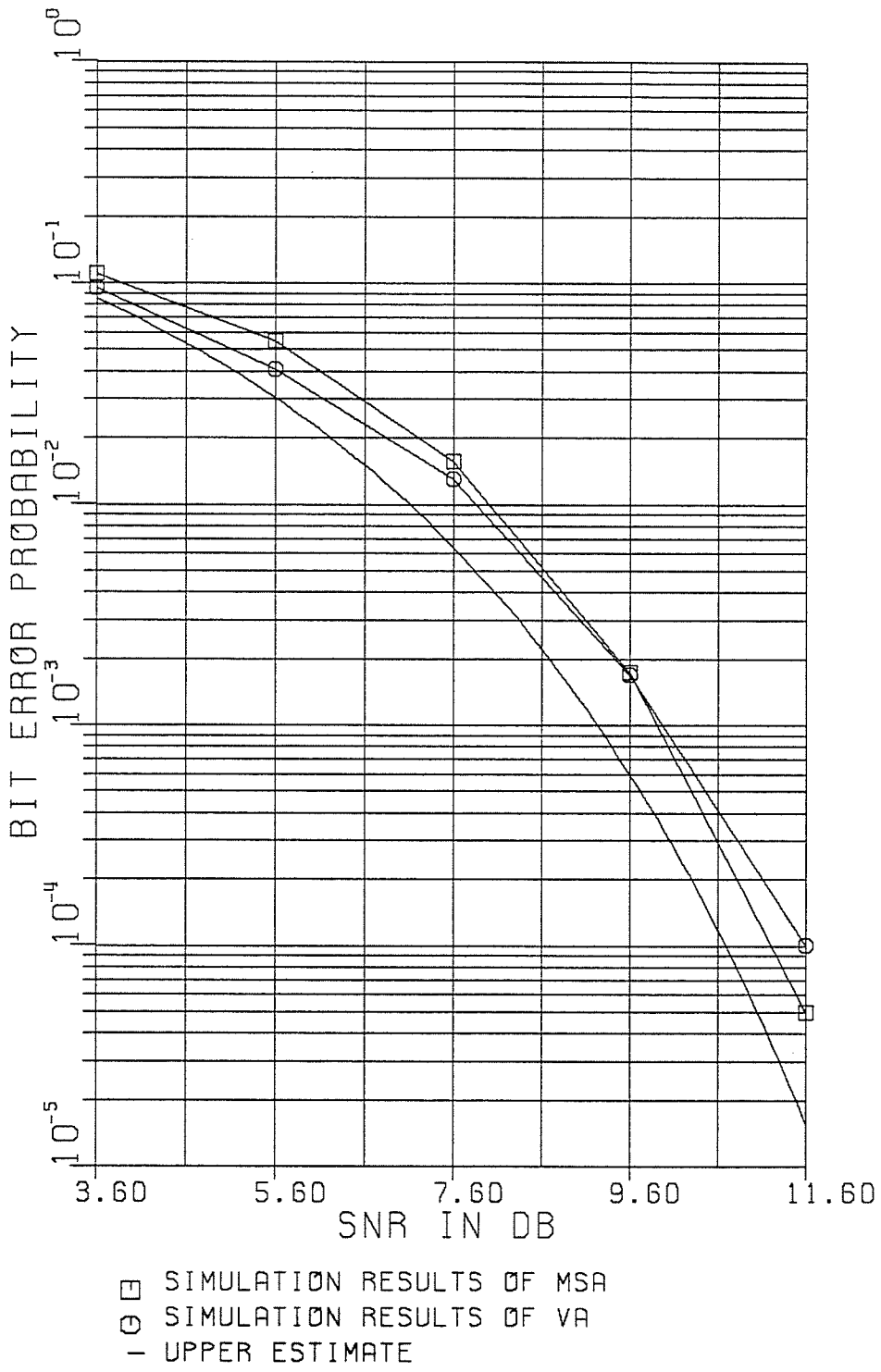

Figure 4.7: Error performance comparison between the vector MSA and the vector VA for the finite 2-dimensional channel in Fig. 4.3

SNR	Vector MSA	Vector VA (L=10)
3.6 dB	17.57	262,144
5.6 dB	13.72	262,144
7.6 dB	4.94	262,144
9.6 dB	1.17	262,144
11.6 dB	1.02	262,144

Table 4.1: Comparison of computations for decoding each 2-D vector (2-dimensional one-pole channel)

from signal vectors, the ensemble argument[34] is still valid for the vector SA provided all the signals are written in vector form. Thus the computational distribution is still Pareto for the vector SA.

Figs.4.9 and 4.10 show the simulated computational distribution of the MSA for the 2-dimensional one-pole channel. When the SNR is high (Fig.4.9), $P(\bar{C} \geq N)$ is basically Pareto since higher order stacks are seldom used. When the SNR is low (Fig.4.10), $P(\bar{C} \geq N)$ becomes more exponential than Pareto since higher order stacks are actually used frequently[58].

Table 4.1 gives the average number of computations per node for the 2-dimensional one-pole channel. Again, as in the case of 1-dimensional one-pole channel, when $SNR \geq 10$ dB the average number of computations per node is almost 1. This means there were almost no back searches at all when decoding, whereas VA needs $2^{18} = 262,144$ computations per node to get the

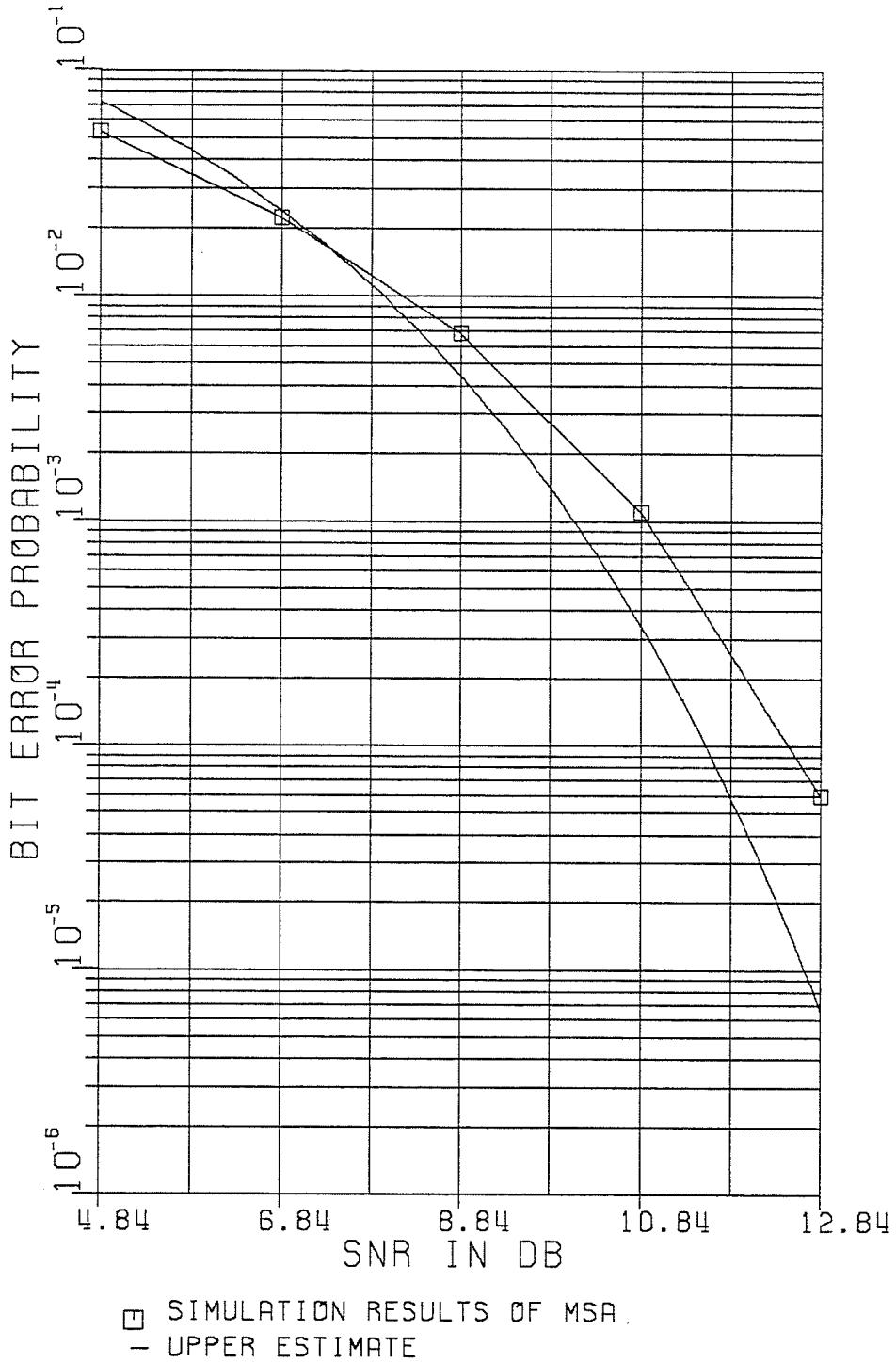

Figure 4.8: Error performance of the vector MSA for the 2-dimensional infinite one-pole channel.

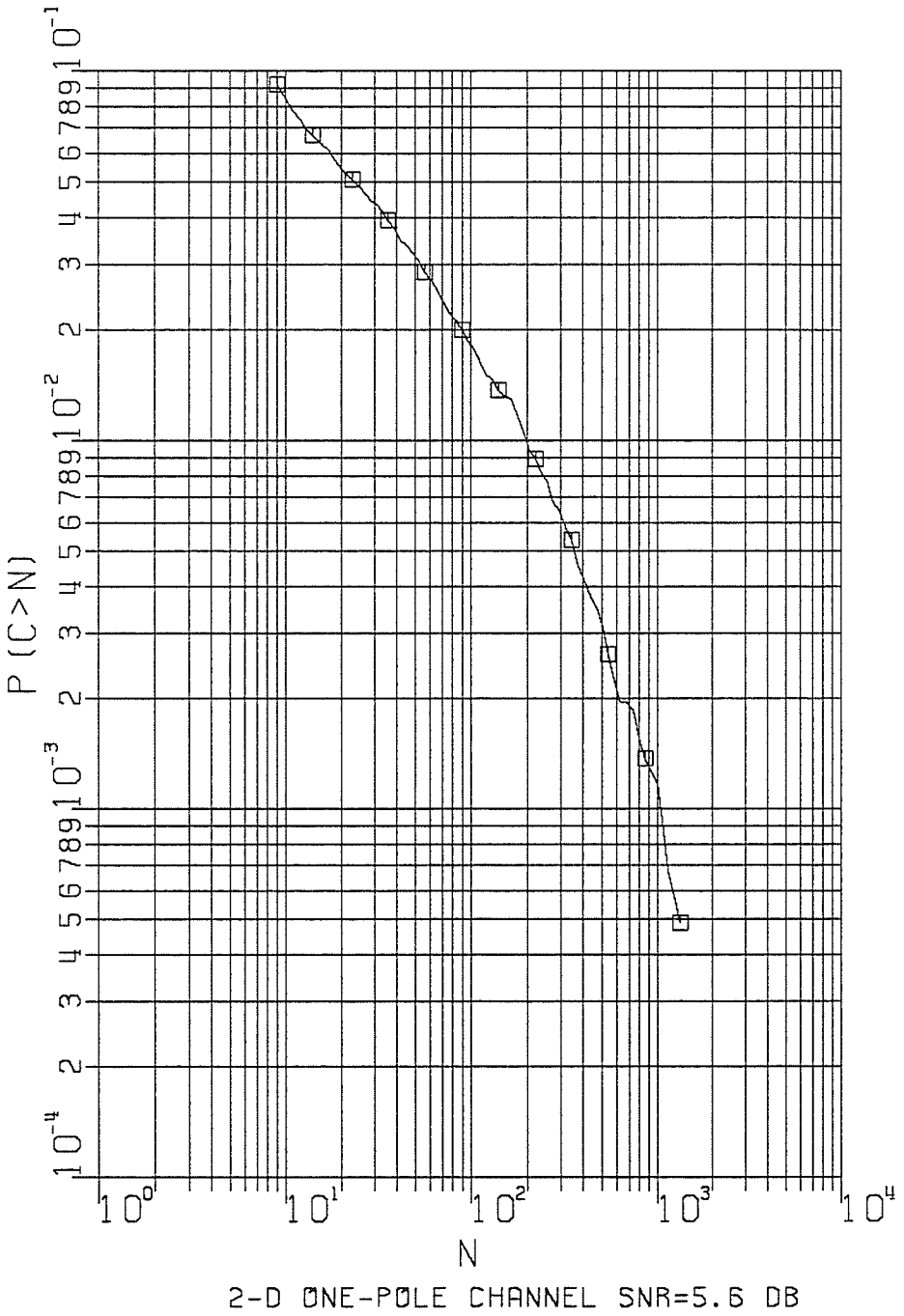

Figure 4.9: Computational distribution of the vector MSA for the 2-dimensional one-pole channel at SNR=5.6 dB: Pareto

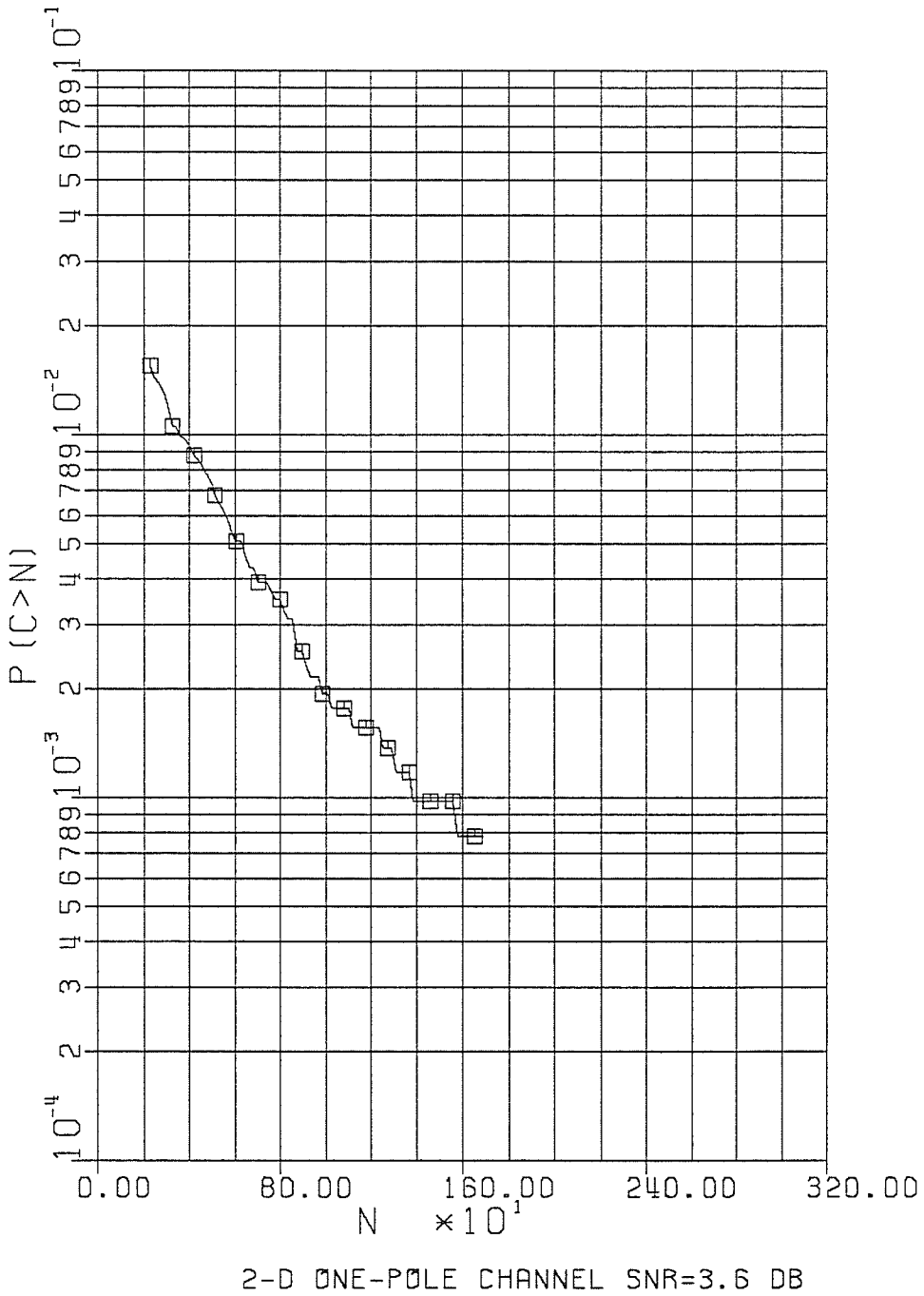

Figure 4.10: Computational distribution of the vector MSA for the 2-dimensional one-pole channel at SNR=3.6 dB: exponential

same error performance (L=10).

No erasures have been observed in the simulation for SNR as low as 4 dB.

Chapter 5

Conclusions and Suggestions for Further Study

5.1 Conclusions

In this thesis the application of sequential sequence estimation to channels with finite or infinite ISI has been investigated. For a PAM channel with either finite or infinite ISI, the output sequence of a whitened matched filter is a set of sufficient statistics for estimation of the input sequence. A near optimum receiver is formed by applying the SA to the output. A multidimensional version of this receiver structure is applicable to multiple channel PAM systems with both ISI and ICI. The metric suitable to the sequential algorithm for the ISI channel is derived, it turns out to be a special form of the Fano metric. Calculation of the metric involves the evaluation of the unconditional probability density $p_z(z_k)$, which can be evaluated by direct calculation for a short memory channel or by estimation for a long or infinite memory channel. To apply the SA to ISI channels, factorization of

the spectrum function $R(D)$ is necessary. This is always possible for any FIR channel and those IIR channels which have a rational, causal and stable transfer function $H(s)$.

The error performance and the computational complexity of the sequential algorithm for ISI channels have been analyzed. Sequential sequence estimation is essentially maximum likelihood. Therefore its error performance prediction is the same as that of MLSE-VA. The finite state machine model can be considered as a special convolutional encoder. Therefore the ensemble average computational distribution of the SA is Pareto. Since the code rate of an uncoded input sequence is fixed, the computational cutoff rate R_{comp} is translated into SNR_{comp}. It is difficult to find SNR_{comp}. Instead an upper bound SNR'_{comp} is found by using an uniform input distribution. SNR'_{comp} serves as a rough indication above which the SA's computational complexity is bounded.

To verify the analytical results simulations were conducted extensively. The simulation results are also compared with the VA and the MA. The comparisons confirm SA's near optimum error performance and considerable computational saving.

However the SA's speed advantage is achieved at the expense of a large stack and/or buffer storage (the Fano algorithm does not need a stack but it is slower). This is tolerable since LSI and VLSI storage devices are available. Another disadvantage of the SA is the erasure of received frames. This can

be essentially overcome by the MSA with enough stack and buffer storage. Finally, the disadvantage that the SA's signal flow graph is not regular makes hardware implementation, particularly in VLSI, more difficult as opposed to the Viterbi algorithm where several architectures have been described [88]. Recently however a VLSI architecture has been described for the sequential algorithm [89].

5.2 Suggestions for Further Study

In this study the SA is primarily compared with the VA to show its near optimum error performance and considerable computational saving. Comparisons between the SA and many reduced state Viterbi algorithms would be of interest. Especially the M-algorithm has been shown to be near optimum with only small number of paths needing to be kept [51], while the simulation results in this study have not shown this. Therefore further comparisons with M-algorithm are of interest.

Another interesting problem is the numerical search for the SNR_{comp}, or equivalently the maximizing input distribution. It is worth trying to adapt Arimoto's search method [83] to the AWGN channel. If it works, the comparison between the SNR_{comp} and the SNR'_{comp} would be interesting.

Since the designer seldom knows the channel impulse response exactly or channel characteristic may change over time, an adaptive scheme for adjusting the parameters of the whitened matched filter and the sequential algo-

rithm is preferable. Theoretically both the whitened matched filter and the sequential decoder must adapt to the channel response. Magee and Proakis proposed an adaptive scheme for the MLSE-VA [32], where only the VA decoder adapts to the channel response and the whitened matched filter is fixed. Their simulation results showed that no significant loss in performance is caused by fixing the whitened matched filter. Further study may investigate if the adaptive scheme is suitable for the sequential algorithm. The possibility and effect of adapting both the whitened matched filter and the sequential algorithm are also worthwhile studying.

Still another interesting problem may be how to design convolutional codes for an ISI channel in order to have a minimum error probability when decoded by a sequential decoder. Calderbank *et al.* designed convolutional codes for a partial response channel with transfer function $(1 - D^N)/2$ [90]. They designed codes that exploit intersymbol interference thus increasing the minimum squared Euclidean distance between outputs corresponding to distinct inputs. Though decoding is accomplished by MLSE-VA in [90], the same improvement in error performance would be achieved if the sequential decoding were used since sequential decoding is essentially maximum likelihood. Furthermore a larger Euclidean distance also reduces the SNR'_{comp}. Therefore designing codes to exploit ISI for a sequential algorithm is the same as designing codes for the MLSE-VA, i.e., exploiting the channel ISI to maximize the minimum Euclidean distance. Further study on this topic may

investigate channels with more complicated transfer function.

Hardware implementation, particularly VLSI implementation of the sequential sequence estimation for ISI channels is an important step toward its practical use. Therefore this is also a research topic of great significance.

Appendix A

Uniform Distribution is not the Maximizing Distribution for $E_0(\rho, \mathbf{q})$ of a Q-ary (Q>2) AWGN channel

Let $y \in \mathcal{Y}$ be the discrete Q-ary input of an AWGN channel. Let $z \in \mathcal{Z}$ be the continuous output of the AWGN channel. Without loss of generality assume the Gaussian noise is of $N(0,1)$ distribution. It is shown in the following that condition (3.21) is not satisfied by a uniform distribution of y unless $Q = 2$ (binary input).

One needs to examine the inequality

$$\sum_z p(z/y)^{1/(1+\rho)} \alpha(z, \mathbf{q})^\rho \geq \sum_z \alpha(z, \mathbf{q})^{1+\rho} \tag{A.1}$$

for all $y \in \mathcal{Y}$, where

$$\alpha(z, \mathbf{q}) = \sum_y q(y) p(z/y)^{1/(1+\rho)} \tag{A.2}$$

Suppose that $q(y) = 1/Q$, then from (A.2)

$$\alpha(z, \mathbf{q}) = \frac{1}{Q} \sum_i \left(\frac{1}{\sqrt{2\pi}} \right)^{\frac{1}{1+\rho}} \exp\left\{ -\frac{(z-y_i)^2}{2(1+\rho)} \right\}. \tag{A.3}$$

Substitute $\alpha(z, \mathbf{q})$ in (A.1) with (A.3) and denote the left side of (A.1) as I_k, then

$$I_k = \int_{-\infty}^{\infty} \frac{1}{Q^\rho \sqrt{2\pi}} \exp\left\{ -\frac{(z-y_k)^2}{2(1+\rho)} \right\} \left[\sum_i \exp\left\{ -\frac{(z-y_i)^2}{2(1+\rho)} \right\} \right]^\rho dy \tag{A.4}$$

and the right side of (A.1) is

$$\textit{Right side} = \frac{1}{Q} \sum_k \int_{-\infty}^{\infty} \frac{1}{Q^\rho \sqrt{2\pi}} \exp\left\{ -\frac{(z-y_k)^2}{2(1+\rho)} \right\} \left[\sum_{y_i} \exp\left\{ -\frac{(z-y_i)^2}{2(1+\rho)} \right\} \right]^\rho dy \tag{A.5}$$

Each term of the summation in above equation is equal to the left side I_k, therefore

$$\textit{Right side} = \frac{1}{Q} \sum_k I_k. \tag{A.6}$$

For a binary input it is easy to verify that the two terms in the above expression are the same, thus (A.1) is satisfied and the uniform distribution is the maximizing distribution.

However for a Q-ary (Q>2) input not all the I_k are the same. Their average (right side) must be larger than those smaller terms, i.e., (A.1) is not always satisfied for every $y_k \in \mathcal{Y}$. Therefore the uniform distribution is not the maximizing distribution.

Appendix B

Proof of Theorem 2.1 and Theorem 4.1

The first part of this appendix is for the case of a single channel (Theorem 2.1). Then it is extended to the case of a multidimensional channel (Theorem 4.1).

B.1 Proof of Theorem 2.1

Suppose $H(s)$ is the Laplace transform of the channel impulse response. It is rational and stable. It has m zeros and n poles (complex or real). Then all the poles are on the left-side s-plane. It can be written as

$$H(s) = \frac{b_m \prod_{l=1}^{m}(s - z_l)}{a_n \prod_{k=1}^{n}(s - p_k)} \tag{B.1}$$

where $m < n$. Since it is stable, $\mathcal{R}e\{p_k\} < 0$, for all k.

(1) First suppose all n poles are distinct, by partial fraction expansion,

$$H(s) = \sum_{k=1}^{n} \frac{c_k}{s - p_k} \tag{B.2}$$

Its Laplace inverse is

$$h(t) = \sum_{k=1}^{n} c_k e^{p_k t} u(t), \tag{B.3}$$

where $u(t)$ is the unit step function. The autocorrelation function is

$$\begin{aligned} R(\tau) &\triangleq \int_{-\infty}^{\infty} h(t)h(t-\tau)dt \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} -\frac{c_i c_j}{p_i + p_j} e^{p_i|\tau|} \\ &= \sum_{i=1}^{n} A_i e^{p_i|\tau|} \end{aligned} \tag{B.4}$$

where

$$A_i \triangleq -\sum_{j=1}^{n} \frac{c_i c_j}{p_i + p_j}. \tag{B.5}$$

Then

$$\begin{aligned} R_k &\triangleq R(kT) \\ &= \sum_{i=1}^{n} A_i e^{p_i|kT|} \end{aligned} \tag{B.6}$$

where $k = 0, \pm 1, \pm 2, \ldots$, $R_k = R_{-k}$. Thus the spectrum function is

$$\begin{aligned} R(D) &\triangleq \sum_{k=-\infty}^{\infty} R_k D^k \\ &= \left[\frac{R(0)}{2} + \sum_{k=1}^{\infty} R_k D^{-k}\right] + \left[\frac{R(0)}{2} + \sum_{k=1}^{\infty} R_k D^k\right] \\ &= g(D^{-1}) + g(D), \end{aligned} \tag{B.7}$$

where

$$g(D) = \sum_{i=1}^{n} \left[\frac{A_i}{1 - e^{p_i T} D} - \frac{A_i}{2}\right], \quad |e^{p_i T} D| < 1. \tag{B.8}$$

Obviously $g(D)$ is rational and can be expressed as

$$g(D) = \frac{A(D)}{B(D)}, \tag{B.9}$$

which is a ratio of two finite polynomials. Hence

$$\begin{aligned} R(D) &= \frac{A(D)}{B(D)} + \frac{A(D^{-1})}{B(D^{-1})} \\ &= \frac{A(D)B(D^{-1}) + B(D)A(D^{-1})}{B(D)B(D^{-1})} \end{aligned} \tag{B.10}$$

where $N(D) = A(D)B(D^{-1}) + B(D)A(D^{-1})$ is a finite polynomial of D and has the property that $N(D) = N(D^{-1})$. Therefore it can be factorized and put in the form $A'(D)A'(D^{-1})$. It follows that

$$\begin{aligned} R(D) &= \frac{A'(D)}{B(D)} \frac{A'(D^{-1})}{B(D^{-1})} \\ &= f(D)f(D^{-1}) \end{aligned} \tag{B.11}$$

where

$$f(D) = \frac{A'(D)}{B(D)}. \tag{B.12}$$

Hence $R(D)$ can be factorized.

(2) Second suppose there is one repeated pole with a multiplicity of $K < n$. Denote it as p_1. Then

$$H(s) = \frac{1}{(s-p_1)^K} \frac{b_m \prod_{l=1}^{m}(s-z_l)}{a_n \prod_{k=2}^{n-K}(s-p_k)} \tag{B.13}$$

By partial fraction

$$H(s) = \frac{c_{11}}{(s-p_1)^K} + \frac{c_{12}}{(s-p_1)^{K-1}} + \ldots + \frac{c_{1K}}{(s-p_1)} + \sum_{k=2}^{n-K} \frac{c_k}{s-p_k} \tag{B.14}$$

By Laplace inversion

$$h(t) = \left[\frac{c_{11}}{(K-1)!}t^{K-1}e^{p_1 t} + \frac{c_{12}}{(K-2)!}t^{K-2}e^{p_1 t} + \ldots + c_{1K}e^{p_1 t} + \sum_{k=2}^{n-K} c_k e^{p_k t}\right] u(t) \tag{B.15}$$

The general form of a term in the autocorrelation function $R(\tau) = h(t) \otimes h(t)$, where $\otimes$ denotes correlation, can be written as (constant factor omitted)

$$\begin{aligned} & \int_{-\infty}^{\infty} t^u e^{p_i t}(t-\tau)^v e^{p_j(t-\tau)} u(t)u(t-\tau)dt \\ = & \int_{-\infty}^{\infty} \sum_{k=0}^{v} \mathrm{C}_v^k t^{v-k}(-\tau)^k e^{-p_i\tau} t^u e^{(p_i+p_j)t} u(t)u(t-\tau)dt \end{aligned} \tag{B.16}$$

where $\max(u) = \max(v) = K-1$ and C_v^k denotes the number of combinations. In the above expression a typical term is

$$e^{-p_j\tau}(-\tau)^k \int_{-\infty}^{\infty} t^l e^{at} u(t)u(t-\tau)dt \tag{B.17}$$

where $l = u+v-k$, $a = p_i + p_j$. When $\tau > 0$, above integral becomes

$$\begin{aligned} & e^{-p_j\tau}(-\tau)^k \int_{\tau}^{\infty} t^l e^{at} dt \\ = & -\frac{(-\tau)^k}{a^{l+1}} e^{(a-p_j)\tau} \\ & \left[(a\tau)^l - l(a\tau)^{l-1} + l(l-1)(a\tau)^{l-2} - \ldots + (-1)^l l!\right] \end{aligned} \tag{B.18}$$

Note that the highest power of τ is $l+k = u+v$, and $\max(u+v) = 2(K-1)$. The exponent is $-p_j + a = p_i$. Therefore $R(\tau)$ is the linear combination of terms of the form

$$\tau^k e^{p_i\tau}, \quad k = 0, 1, 2, \ldots, 2(K-1). \tag{B.19}$$

There are terms with same τ^k but different p_i, and also there are terms with same p_i but different τ^k.

When $\tau < 0$, $R(\tau)$ has the same form except τ should be replaced by $-\tau$ since $R(\tau) = R(-\tau)$ is always true for autocorrelation function. Thus the $R(\tau)$ may be written as

$$R(\tau) = \sum_{k=0}^{2(K-1)} |\tau|^k \left[\sum_{l=1}^{l_k} b_l^{(k)} \exp(p_l^{(k)}|\tau|) \right], \tag{B.20}$$

where l_k denotes the number of the terms with $|\tau|^k$, $p_l^{(k)}$ is a pole and $b_l^{(k)}$ is a constant associated with the term. Due to the symmetry of $R(\tau)$

$$\begin{aligned} R(D) &= \sum_{i=-\infty}^{\infty} R_i D^i \\ &= g(D^{-1}) + g(D). \end{aligned} \tag{B.21}$$

where $R_i = R(iT)$,$R_0 = \sum_{l=1}^{l_0} b_l^{(0)}$ and

$$g(D) = \frac{R_0}{2} + \sum_{i=1}^{\infty} R_i D^i \tag{B.22}$$

Let all $b_l^{(0)}$ be reduced to half, and note that $i \geq 0$ in $g(D)$, then $g(D)$ can be written as

$$\begin{aligned} g(D) &= \sum_{i=0}^{\infty} \left\{ \sum_{k=0}^{2(K-1)} (iT)^k \left[\sum_{l=1}^{l_k} b_l^{(k)} \exp(p_l^{(k)} iT) \right] \right\} D^i \\ &= \sum_{k=0}^{2(K-1)} \sum_{l=1}^{l_k} g_k^{(l)}(D) \end{aligned} \tag{B.23}$$

where

$$g_k^{(l)}(D) = \sum_{i=0}^{\infty} (iT)^k b_l^{(k)} \exp(p_l^{(k)} iT) D^i \tag{B.24}$$

This D-transform is evaluated as follows. To simplify notations, drop all the subscripts and superscripts. First take the z-transform of it and replace z by D^{-1} to get the D-transform. The z-transform of function e^{pTi} is

$$e^{pTi} \overset{\mathcal{Z}}{\Longleftrightarrow} \frac{z}{z - e^{pT}} \tag{B.25}$$

Using the formula

$$i^k x(i) \overset{\mathcal{Z}}{\Longleftrightarrow} \left(-z\frac{d}{dz}\right)^k X(z) \tag{B.26}$$

one obtains

$$b(Ti)^k e^{pTi} \overset{\mathcal{Z}}{\Longleftrightarrow} bT^k e^{pT} k! \frac{z^k}{(z - e^{pT})^{k+1}} \tag{B.27}$$

or in D-transform

$$b(Ti)^k e^{pTi} \overset{\mathcal{D}}{\Longleftrightarrow} bT^k e^{pT} k! \frac{D^{-k}}{(D^{-1} - e^{pT})^{k+1}} \tag{B.28}$$

Thus

$$g_k^{(l)}(D) = bT^k e^{pT} k! \frac{D}{(1 - De^{pT})^{k+1}} \tag{B.29}$$

It is rational. Consequently $R(D)$ is rational and has the form of $R(D) = g(D^{-1}) + g(D)$, which has been proved factorable as $R(D) = f(D)f(D^{-1})$. From above argument it is obvious that the number of repeated poles need not be one. Any number of repeated poles is allowed.

B.2 Proof of Theorem 4.1

Assume the transfer function of the multidimensional channel is

$$H(s) \triangleq \begin{bmatrix} h_{11}(s) & h_{12}(s) & \dots & h_{1M}(s) \\ h_{21}(s) & h_{22}(s) & \dots & h_{2M}(s) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ h_{M1}(s) & h_{M2}(s) & \dots & h_{MM}(s) \end{bmatrix}. \tag{B.30}$$

where $h_{ij}(s)$ is rational and stable. It has n_{ij} complex poles (complex or real, single or repeated), all of them are on the left-side s-plane.

The correlation matrix is defined as

$$\begin{aligned} R(\tau) &\triangleq H'(t) \otimes H(t) \\ &= \int_{-\infty}^{\infty} H'(t)H(t-\tau)dt \end{aligned} \tag{B.31}$$

It is easy to verify that $R(\tau) = R'(-\tau)$. The element of $R(\tau)$ is defined as

$$R_{nj}(\tau) \triangleq \sum_{i=1}^{M} R_{nj}^{(i)}(\tau) \tag{B.32}$$

where

$$R_{nj}^{(i)}(\tau) \triangleq \int_{-\infty}^{\infty} h_{in}(t-\tau)h_{ij}(t)dt. \tag{B.33}$$

$R_{nj}^{(i)}(\tau)$ is the cross-correlation of two one-dimensional impulse responses. Each of them consists of functions of the form $t^u e^{pt}$. According to the argument in the proof of Theorem 2.1, $R_{nj}^{(i)}(D)$ is rational. Consequently $R(D)$ is also rational.

In summary $R(D)$ is rational and satisfies relation $R(D) = R'(D^{-1})$. The factorization of this kind of spectral matrix is a well solved problem[86,87].

Appendix C

Sequential Sequence Estimation for ISI Channel with Coded Input Sequence

In previous chapters only uncoded digital sequences were considered. However, coding is usually incorporated into the communication system. Here the ISI channel with a convolutional coded input sequence is considered.

It has been shown that SA is applicable to both decoding convolutional codes and demodulating ISI channels. This commonality in method is due to their commonality in structure. In the case of finite ISI, both convolutional encoder and ISI channel together with whitened matched filter can be modelled as a finite state machine. The cascade of these two finite state machines is still a finite state machine with longer memory. In the case of infinite ISI, the ISI channel-WMF combination is modelled as a feedback digital filter (Fig.2.12), the cascade of a convolutional encoder with it is still a feedback digital filter with a longer feedforward part. These facts suggest that the SA

should be applicable to convolutional coded transmission over ISI channels (either finite or infinite). This is shown here.

C.1 Channel Model and Receiver

Fig.C.1 shows the cascade of a convolutional encoder and an ISI channel with whitened matched filter. The code is an (n,b,K) code. For every b input symbols of u, there are n output symbols of x, n output symbols of y and n received signals of z accordingly. Parameter K is the memory length of the encoder. These correspondences between groups of symbols are expressed in vector forms in Fig.C.1, where $\mathbf{u}_k$ is a $b \times 1$ vector, $\mathbf{x}_k$, $\mathbf{y}_k$, $\mathbf{n}_k$ and $\mathbf{z}_k$ are $n \times 1$ vectors:

$$\mathbf{u}_k \triangleq (u_k^{(1)}, u_k^{(2)}, \ldots, u_k^{(b)})' \tag{C.1}$$

$$\mathbf{x}_k \triangleq (x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(b)})' \tag{C.2}$$

$$\mathbf{y}_k \triangleq (y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(b)})' \tag{C.3}$$

$$\mathbf{n}_k \triangleq (n_k^{(1)}, n_k^{(2)}, \ldots, n_k^{(b)})' \tag{C.4}$$

$$\mathbf{z}_k \triangleq (z_k^{(1)}, z_k^{(2)}, \ldots, z_k^{(b)})' \tag{C.5}$$

where $'$ denotes the transpose of a vector.

Note that components of above vectors except $\mathbf{u}_k$ actually appear sequentially in the system. They are organized in vector form for the sake of clearness.

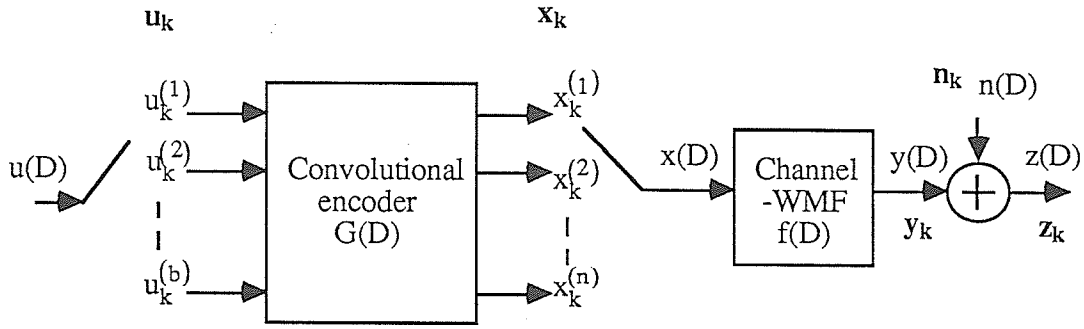

Figure C.1: Cascade of a convolutional encoder and an ISI channel with whitened matched filter

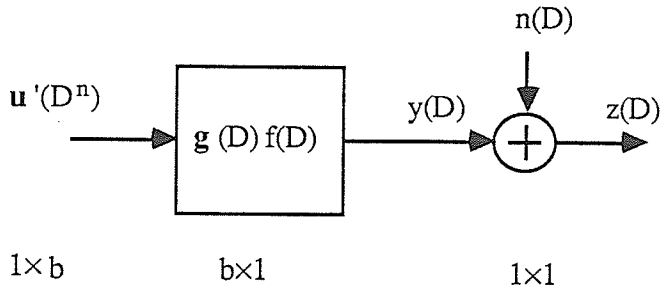

Figure C.2: Finite state machine model of Fig.C.1

To determine the relation between x(D) and u(D), define

$$u^{(i)}(D) \triangleq \sum_k u_k^{(i)} D^k, \quad i = 1, 2, ..., b \tag{C.6}$$

$$x^{(i)}(D) \triangleq \sum_k x_k^{(i)} D^k, \quad i = 1, 2, ..., b \tag{C.7}$$

$$\mathbf{u}(D) \triangleq (u^{(1)}(D), u^{(2)}(D), \ldots, u^{(b)}(D))' \tag{C.8}$$

$$\mathbf{x}(D) \triangleq (x^{(1)}(D), x^{(2)}(D), \ldots, x^{(n)}(D))' \tag{C.9}$$

Then

$$\mathbf{x}(D) = G'(D)\mathbf{u}(D) \tag{C.10}$$

where G(D) is the transfer function matrix of the encoder, defined as

$$G(D) \triangleq \begin{bmatrix} g_1^{(1)}(D) & g_1^{(2)}(D) & \ldots & g_1^{(n)}(D) \\ g_2^{(1)}(D) & g_2^{(2)}(D) & \ldots & g_2^{(n)}(D) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ g_b^{(1)}(D) & g_b^{(2)}(D) & \ldots & g_b^{(n)}(D) \end{bmatrix} \tag{C.11}$$

in which $g_i^{(j)}(D)$ is the generator polynomial relating input $u^{(i)}$ to output $x^{(j)}$, $g_i^{(j)}(D)$ has degree K.

After multiplexing, the code word becomes

$$x(D) = x^{(1)}(D^n) + Dx^{(2)}(D^n) + \ldots + D^{(n-1)}x^{(n)}(D^n) \tag{C.12}$$

Thus from equations (C.10) and (C.12) the explicit relation between $x(D)$ and $u(D)$ is

$$x(D) = \sum_{i=1}^{b} u^{(i)}(D^n) g_i(D) \tag{C.13}$$

where

$$g_i(D) = g_i^{(1)}(D^n) + Dg_i^{(2)}(D^n) + \ldots + D^{(n-1)}g_i^{(n)}(D^n) \tag{C.14}$$

Further denoting $\mathbf{g}(D) = (g_1(D), g_2(D), \ldots, g_b(D))'$, (C.13) can be written as

$$x(D) = \mathbf{g}'(D)\mathbf{u}(D^n) \tag{C.15}$$

Therefore the output sequence y(D) can be related directly to input vector sequence u(D) as

$$y(D) = f(D)x(D) = f(D)\mathbf{g}'(D)\mathbf{u}(D^n) \tag{C.16}$$

or by taking the transpose it becomes

$$y(D) = \mathbf{u}'(D^n)\mathbf{g}(D)f(D) \tag{C.17}$$

The relation of $y(D)$ and $\mathbf{u}(D^n)$ can be shown in Fig.C.2. In terms of input vectors $\mathbf{u}_i$, this system has a memory length of $K + [\nu/n]$, where $[\nu n]$ denotes the smallest integer not less than ν/n. When ν is infinite total memory length is infinite too.

C.2 Sequential Estimation

The relation of Fig.C.2 can be represented by a tree as shown in Fig.C.3 for a (3,2,1) encoder and a channel with $\nu = 2$ ISI terms and a binary input. There are 4 branches at each node, corresponding to 4 possible input vectors,

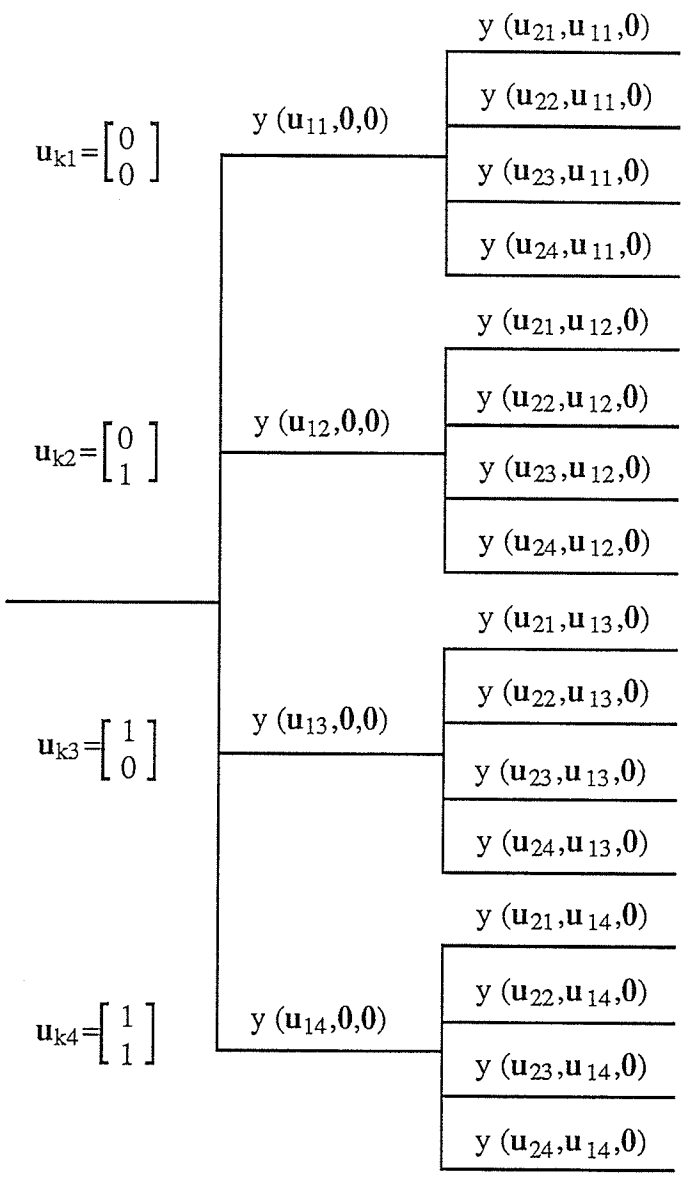

Figure C.3: Tree diagram for a (3,2,1) encoder and a channel with $\nu = 2$ ISI terms and a binary input

hence 4 possible output vectors of y. For the l^{th} path of n_l branches long, the output sequence of y has $n_l n$ symbols, therefore the path metric is

$$L(X_l, Z) = \sum_{i=1}^{n_l n} \left[\log \frac{p_n(z_i - y_i)}{p_z(z_i)} + \frac{1}{n_l n} \log P_l \right]. \tag{C.18}$$

where

$$P_l = Pr(X_l) = Pr(U_l) \tag{C.19}$$

For an equiprobable m-ary input sequence U_l

$$Pr(U_l) = \frac{1}{m^{n_l n R}} \tag{C.20}$$

where R is the code rate, R=b/n. Hence the path metric becomes

$$L(U_l, Z) = \sum_{i=1}^{n_l n} \left[\log \frac{p_n(z_i - y_i)}{p_z(z_i)} - R \log m \right]. \tag{C.21}$$

Note that scalar sequence

$$U_l = \{u_0^{(1)}, u_0^{(2)}, \ldots, u_0^{(b)}, u_1^{(1)}, u_1^{(2)}, \ldots, u_1^{(b)}, \ldots, u_{n_l}^{(1)}, u_{n_l}^{(2)}, \ldots, u_{n_l}^{(b)}\} \tag{C.22}$$

splits into vectors of b symbols: $\mathbf{u}_i$, and sequence Y_l and Z also split into vectors of n symbols: $\mathbf{y}_i$ and $\mathbf{z}_i$, the metric can be written as

$$L(U_l, Z) = \sum_{i=1}^{n_l} L(\mathbf{u}_i, \mathbf{z}_i) \tag{C.23}$$

where

$$L(\mathbf{u}_i, \mathbf{z}_i) = \sum_{j=1}^{n} \left[\log \frac{p_n(z_i^{(j)} - y_i^{(j)})}{p_z(z_i^{(j)})} - R \log m \right]. \tag{C.24}$$

This metric serves as a branch metric in the SA for the tree of Fig.C.3. The SA computes the branch metric at a node for each $\mathbf{u}_i$, $i = 1, 2, \ldots, m^b$,

and chooses the largest. To compute $L(\mathbf{u}_i, \mathbf{z}_i)$, one computes $\mathbf{x}_i$ from $(\mathbf{u}_i, \mathbf{u}_{i-1}, \ldots, \mathbf{u}_{i-K})$ and then computes $\mathbf{y}_i$ from $(\mathbf{x}_i, \mathbf{x}_{i-1}, \ldots, \mathbf{x}_{i-(K+[\nu/n])})$. As a result $\mathbf{y}_i$ can be written as

$$\mathbf{y}_i = y(\mathbf{u}_i, \mathbf{u}_{i-1}, \ldots, \mathbf{u}_{i-(K+[\nu/n])}). \tag{C.25}$$

The computation of $p_z(z_i^{(j)})$ is similar to that described in Chapter 2. Equation (C.25) shows that vector $\mathbf{y}_i$ is a function of $K+[\nu/n]+1$ consecutive input vectors, therefore each component $y_i^{(j)}$ is also a function of the same input vector sequence. Noting that each input vector has b components,$y_i^{(j)}$ is then a function of $L' = b(K + [\nu/n] + 1)$ input symbols. If the input symbols are equiprobable m-ary symbols, then

$$Pr(y_i^{(j)}) = \frac{1}{m^{L'}} \tag{C.26}$$

and

$$p_z(z_k^{(j)}) = \frac{1}{m^{L'}} \sum_{i=1}^{m^{L'}} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(z_k^{(j)} - y_i^{(j)})^2}{2\sigma^2} \right\}. \tag{C.27}$$

The calculation of (C.27), as discussed in Chapter 2, can be done either by direct calculation for small L' or by approximation for long L' even infinite L'.

Bibliography

[1] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, S. U. Qureshi, *Efficient Modulation for Band-Limited Channels,* IEEE Journal on Selected Areas in Communications, Vol. SAC-2, No.5, Sept. 1984.

[2] J. G. Proakis, *Digital Communications.* McGraw-Hill,Inc.,New York,N.Y.,1983.

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

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

[5] M. Schwartz, *Information Transmission, Modulation, and Noise.* McGraw-Hill,Inc.,New York,N.Y.,1980.

[6] D. A. Schnidman, *A Generalized Nyquist Criterion and Optimum Linear Receiver for a Pulse Amplitude Modulation System,* Bell Syst. Tech. J., Vol 46, pp 2163-2177, Nov. 1967.

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

[8] A. Lender, *Correlative Level Coding for Binary-Data Transmission,* IEEE Spectrum, Vol.3, No.2, pp.104-115, Feb. 1966.

[9] E. R. Kretzmer, *Binary Data Communication by Partial Response Transmission,* Conference Record, IEEE Annual Communication Convention, pp.451-455, 1965.

[10] E. R. Kretzmer, *Generalization of a Technique for Binary Data Communication,* IEEE Trans. on Commun. Tech., Vol. COM-14, pp. 67-68, Feb. 1966.

[11] D. W. Tufts, *Nyquist Problem-The Joint Optimization of Transmitter and Receiver in Pulse Amplitude Modulation.* Proc. IEEE, Vol 53, pp. 248-260, Mar. 1965.

[12] M. R. Aaron, D. W. Tufts, *Intersymbol Interference and Error Probability,* IEEE Trans. on Inf. Theory, Vol IT-12, pp. 26-34, Jan. 1966.

[13] K. Yao, *On the Minimum Average Probability of Error Expression for Binary Pulse Communication Systems with Intersymbol Interference,* IEEE Trans. on Inf. Theory, Vol IT-18, pp. 528-531, July 1972.

[14] T. Ericson, *Structure of Optimum Receiving Filters in Data Transmission Systems,* IEEE Trans. on Inf. Theory, Vol IT-17, pp. 352-353, May 1971.

[15] T. Ericson, *Optimum PAM-Filters Are Always Bandlimited,* IEEE Trans. on Inf. Theory, Vol IT-19, pp. 570-573, July 1973.

[16] W. Van Etten, *An Optimum Linear Receiver for Multiple Channel Digital Transmission Systems,* IEEE Trans. on Commun., Vol. COM-23, pp. 828-834, Aug. 1975.

[17] G. Ungerboeck, *Fractional Tap-Spacing Equalizer and Consequences for Clock Recovery in Data Modems,* IEEE Trans. Commun., pp. 856-864, August 1976.

[18] R. Lucky, *Automatic Equalization for Digital Communication,* Bell Syst. Tech. J., Vol 44, pp. 547-588, April 1965.

[19] R. Lucky, *Techniques for Adaptive Equalization of Digital Communication,* Bell Syst. Tech. J., Vol 45, pp. 255-286, Feb. 1966.

[20] S. Qureshi, *Adaptive Equalization,* IEEE Communication Magazine, Vol. 20, pp. 9-16, March 1982. No. 5, Sept. 1984.

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

[22] D. Gorge, R. Bowen, J. Storey, *An Adaptive Decision Feedback Equalizer,* IEEE Trans. on Commun. Tech., Vol. COM-19, pp. 281-292, June 1971.

[23] K. Abend, B. D. Fritchman, *Statistical Detection for Communication Channels with Intersymbol Interference,* Proc. IEEE, Vol 58, pp. 779-785, May 1970.

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

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

[26] 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.

[27] 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.

[28] J. K. Omura, *Optimal Receiver Design for Convolutional Codes and Channels with Memory via Control Theoretical Concepts,* Inform. Sci., Vol 3, pp. 243-266 July 1971.

[29] H. Kobayashi, *Correlative Level Coding and Maximum Likelihood Decoding,* IEEE Trans. on Inf. Theory, Vol IT-17, pp. 586-594, Sept. 1971.

[30] 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.

[31] W. Van Etten, *Maximum Likelihood Receiver for Multiple Channel Transmission Systems,* IEEE Trans. on Commun., Vol. COM-24, pp. 276-283, Feb. 1976.

[32] F. R. Magee, Jr. and J. G. Proakis, *Adaptive Maximum Likelihood Sequence Estimation for Digital Signaling in the Presence of Intersymbol Interference,* IEEE Trans. on Inf. Theory, Vol IT-19, pp. 120-124, Jan. 1973.

[33] A. S. Acampora, *Maximum Likelihood Decoding of Binary Convolutional Codes on Band Limited Satellite Channels,* Conf. Rec., National Telecommunication Conference, 1976.

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

[35] G. Ungerboek, *Adaptive Maximum Likelihood Receiver for Carrier-Modulated Data-Transmission Systems,* IEEE Trans. on Commun., Vol. COM-22, pp. 624-636, May. 1974.

[36] S. Qureshi, E. Newhall, *An Adaptive Receiver for Data Transmission over Time-Dispersive Channels,* IEEE Trans. on Inf. Theory, Vol IT-19, pp. 448-457, July 1973.

[37] D. D. Falconer, F. R. Magee, *Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation,* Bell Syst. Tech. J., Vol 52, pp. 1541-1562, Nov. 1973.

[38] D. Messerschmitt, *Design of a Finite Impulse Response for the Viterbi Algorithm and Decision Feedback Equalizer,* Rec., Inter. Conf. on Commun., ICC-74, June 17-19,1974, Minneapolis, MN.

[39] W. U. Lee, *Decision-Feedback Equalized Maximum Likelihood Sequence Estimation,* Ph.D Dissertation, University of Massachusetts, 1975.

[40] W. U. Lee and F. S. Hill *A maximum-likelihood sequence estimator with decision-feedback equalization,* IEEE Trans. on Commun., Vol. COM-25,p.971-979, Sept.1977

[41] F. L. Vermeulen, *Low Complexity Decoders for Channels with Intersymbol Interference,* Ph.D Dissertation, Stanford University, 1975.

[42] G. J. Foshini, *A Reduced State Variant of Maximum Likelihood Sequence Detection Attaining Optimum Performance for high Signal-to-Noise Ratio,* IEEE Trans. on Inf. Theory, Vol IT-23, pp.605-609, Sept. 1977.

[43] D. Kazakos, *Computational Savings and Implementation of Maximum Likelihood Detectors,* IEEE Trans. on Inf. Theory, Vol IT-24, pp.124-126, Jan. 1978.

[44] A. P. Clark and M. Clayden, *Pseudobinary Viterbi Detector,* IEE Proc. F. Commun., Radar & Signal Processing, pp.208-218, April 1984.

[45] A. P. Clark and S. N. Abdullah, *Near-Maximum-Likelihood Detectors for voiceband Channels,* IEE Proc. F. Commun., Radar & Signal Processing, pp.217-226, June 1987.

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

[47] G. Ungerboeck, *Channel Coding with Multi-Level/Phase Signals,* IEEE Trans. Inform. Theory, Vol. IT-28, pp. 55-67, Jan. 1982

[48] A. Duel and C. Heegard, *Delayed Decision Feedback Sequence Estimation,* 23rd Annual Allertor Conference on Communication., Control, and Computing, pp. 878-887, 1985.

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

[50] J. B. Anderson and C. F. Lin, *M-Algorithm Decoding of Channel Codes,* Proc. 13th Biennial Symp. on Commun., Queen's University, Kingston, Canada, pp. A3.1-A3.4, June 2-4, 1986.

[51] N. Seshadri and J. B. Anderson, *An M-Algorithm Receiver fo Severe Infinite Intersymbol Interference Channels,* Abstracts of Papers, International Symposium on Information Theory, Kobe, Japan, pp. 68, June 1988.

[52] J. M. Wozencraft, *Sequential Decoding for Reliable Communications,* MIT-RLE Tech. Rep. TR325, 1957.

[53] R. M. Fano, *A Heuristic Discussion of Probabilistic Decoding,* IEEE Trans. on Inf. Theory, Vol IT-9, pp. 64-74, April 1963.

[54] K. Zigangirov, *Some Sequential Decoding Procedures,* Probl. Peredachi Inf. 2, pp. 13-25, 1966.

[55] F. Jelinek, *A Fast Sequential Decoding Algorithm Using a Stack,* IBM J. Res. and Dev. 13, pp. 675-685, Nov. 1969.

[56] G. D. Forney, *Convolutional Codes III: Sequential Decoding,* Info. and Control, Vol.25, pp.267-297, July 1974.

[57] J. E. Savage, *Sequential Decoding-The Computation Problem,* Bell Syst. Tech. J.., Vol. 25, pp. 149-175. Jan. 1966.

[58] P. Chevillat, D. J. Costello,Jr., *A Multiple Stack Algorithm for Erasure-free Decoding of Convolutional Codes,* IEEE Trans. on Commun., Vol. COM-25, pp.1460-1470, Dec. 1977.

[59] P. Haccoun and P. Haurie, *Application of multiple-path sequential decoding to the intersymbol interference problem,* Abstracts of Papers, IEEE International Symposium on Information Theory, pp. 110, 1982.

[60] J. L. Massey, *Variable-Length Codes and the Fano Metric,* IEEE Trans. on Inf. Theory, Vol IT-18, pp.196-198, Jan. 1972.

[61] F. S. Hill, Jr., *The Computation of Error Probability for Digital Transmission,* Bell Syst. Tech. J., Vol.-50, pp.2055-2077, July-August, 1971.

[62] F. S. Hill, Jr. and M. A. Blanco, *Random Geometric Series and Intersymbol Interference,* IEEE Trans. on Inform. Theory, Vol. IT-19, pp. 326-335, May 1973.

[63] K. Metzger, *On the Probability Density of Intersymbol Interference,* IEEE Trans. on Commun., Vol. COM-15, pp.369-402, April 1987.

[64] Chao S. Chi, *Characterization and Spectral Equalization for High-Density Disk Recording,* IEEE Trans. on Magnetics, Vol. MAG-15, No.6, Nov. 1979.

[65] J. M. Cioffi and C. M. Melas, *Evaluating the Performance of Maximum Likelihood Sequence Detection in a Magnetic Recording Channel,* IBM Research Report, RJ 5156(53496), May 1986.

[66] G. F. Foschini, *Performance Bound for Maximum Likelihood Reception of Digital Data,* IEEE Trans. on Inform. Theory, Vol. IT-21, pp.47-50, Jan. 1975.

[67] A. Zerik, *Discrete Spectral Factorization in Intersymbol Interference Problems,* M. Sc. Thesis, The Department of Electrical Engineering, University of Manitoba, Dec. 1988.

[68] W. B. Gragg, *The Pade Table and its Relation to Certain Algorithms of Numerical Analysis,* SIAM Rev., Vol. 14, No. 1, pp.1-62, Jan. 1972.

[69] D. Haccoun, M. J. Ferguson, *Generalized Stack Algorithm for Decoding Convolutional Codes,* IEEE Trans. on Inf. Theory, Vol IT-21, pp.638-651, Nov. 1975.

[70] M. Ma, *The Multiple Stack Algorithm Implementated on a Zilog Z-80 Microcomputer,* IEEE Trans. on Commun., Vol. COM-28, pp. 1876-1882, Nov. 1980.

[71] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering.* John Wiley I& Sons Inc., New York, N.Y., 1965.

[72] R. G. Gallager, *Information Theory and Reliable Communication.* John Wiley I& Sons Inc., New York, N.Y., 1968.

[73] R. J. McEliece, *The Theory of Information and Coding.* Addison-Wesley Publishing Company, 1977.

[74] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, Englewood Cliffs, N.J., 1983.

[75] R. E. Blahut, *Theory and Practice of Error Control Codes.* Addison-Wesley, Inc.,Reading, Mass., 1983.

[76] E. A. Lee and D. G. Messerschmitt, *Digital Communication.* Kluwer Academic Publishers, Boston, M.A., 1988.

[77] H. L. Van Trees, *Detection, Estimation, and Modulation Theory.* John Wiley I& Sons Inc., New York, N.Y., 1968.

[78] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill,Inc.,New York,N.Y.,1965.

[79] B. W. Silverman, *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, New York, N.Y., 1986.

[80] B. Noble, *Applied Linear Algebra.* Prentice Hall, Inc., Englewood Cliffs, N.J., 1969.

[81] A. V. Oppenheim, K. W. Schafer, *Digital Signal Processing.* Prentice Hall, Inc., Englewood Cliffs, N.J., 1975.

[82] P. Chevillat and D. J. Costello, Jr., *An Analysis of Sequential Decoding for Specific Time-Invariant Convolutional Codes,* IEEE Trans. on Inform. Theory, Vol. IT-24, pp. 443-451, July 1987.

[83] S. Arimoto, *Computation of Random Coding Exponent Functions,* IEEE Trans. on Inform. Theory. Vol. IT-22, No. 6, pp. 665-671, Nov. 1976.

[84] A. R. Kaye and D. A. George, *Transmission of Multiplexed PAM Signals over Multiple Channel and Diversity Systems,* IEEE Trans. on Commun., Vol. COM-18, pp. 520-526, Oct. 1970.

[85] W. van Etten, *An Optimum Linear Receiver for Multiple Channel Digital Transmission Systems,* IEEE Trans. Commun., Vol COM-23, pp. 828-834, Aug. 1975.

[86] D. N. Prabhakar Murthy, *Factorization of Discrete-Process Spectral Matrices,* IEEE Trans. Inform. Therory, Vol. IT-19, pp. 693-696, Sept. 1973.

[87] P. R. Motyka and J. A. Cadzow, *The Factorization of Discrete-Process Spectral Matrices,* IEEE Trans. Automat. Contr., Vol. AC-12, pp. 698-707, Dec. 1967.

[88] P. G. Gulak, E. Shwedyk, *VLSI Structure for Viterbi Receivers: Part I - General Theory and Applications,* IEEE Journal on Selected Areas in Communications, Vol. SAC-4, No.1, pp. 142-154, Jan. 1986.

[89] C. Y. Chang, K. Yao, *Systolic Array Architecture for the Sequential Stack Decoding Algorithm,* SPIE Conference, San Diego, U.S.A., 1986.

[90] A. R. Calderbank, C. Heegard, and T. Lee, *Binary Convolutional Codes with Application to Magnetic Recording,* IEEE Trans. Inform. Theory, Vol. IT-32, No.6, pp.797-814, Nov. 1986