N8211341

JPL PUBLICATION 81-73

Modulation/Demodulation Techniques for Satellite Communications

Part IV: Appendices

Jim K. Omura
Marvin K. Simon

November 1, 1981

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

FOREWORD

This report represents Part IV of a series of reports to be published under the same title with the following subtitles:

Part I:	Background
Part II:	Advanced Techniques - The Linear Channel
Part III:	Advanced Techniques - The Nonlinear Channel

ABSTRACT

In addition to decoding convolutional codes, the Viterbi algorithm is useful in a host of other applications, some of which include: maximum likelihood demodulation of new bandwidth efficient modulations such as minimum-shift-keying (MSK) and continuous phase frequency-shift-keying (CPFSK), demodulation of intersymbol interference and partial response signals, estimation and smoothing, and simultaneous phase synchronization/data detection. Performance bounds for these new and exciting applications of the Viterbi algorithm can be obtained by a generalization of the transfer function approach originally introduced by Viterbi for obtaining bit-error probability bounds on the performance of specific convolutional codes over specific symmetric channels. In Appendix A we examine the use of the Viterbi algorithm in a general context and present the generalized transfer function bounds necessary to carry out the applications mentioned above.

The well-known Chernoff and Bhattacharyya bounds can, under certain condition be made tighter than their commonly quoted standard versions by a factor of one-half. Using a new approach, Appendix B reviews sufficient conditions under which these reductions can occur, at the same time making these conditions less restrictive but also harder to verify.

CONTENTS

Figures

CONTENTS (Contd.)

APPENDIX A

Generalized Transfer Function Bounds

I. Introduction

In this appendix we derive performance bounds for the Viterbi algorithm used in a general estimation/detection context. Special cases include decoding convolutional codes, demodulation of new bandwidth efficient modulations such as MSK and CPFSK, demodulation of intersymbol interference signals, estimation and smoothing, and simultaneous synchronization/data detection.

Our approach is to generalize the transfer function bounds originally used to evaluate the bit error probability of binary convolutional codes with binary-input output-symmetric memoryless channels (Ref. 1). We begin by describing the general estimation/detection problem and the use of the Viterbi algorithm in its solution. Next "super state" diagrams are defined and generalized transfer function bounds are derived. Special forms of these state diagrams and transfer function bounds are then examined.

II. Discrete-Time System Model

We assume the discrete-time system shown in Figure A-1. Here the signal is described as a general finite state system given by the output

$$x_k = f(s_k, u_k) \tag{A.2.1}$$

and state relation

$$s_{k+1} = g(s_k, u_k) \tag{A.2.2}$$

where u_k, x_k, and s_k have finite alphabets denoted $\mathcal{U}$, $\mathcal{X}$, and $\mathcal{S}$ respectively. The sizes of these alphabets are denoted $|\mathcal{U}|$, $|\mathcal{X}|$, and $|\mathcal{S}|$. Note that while $|\mathcal{S}|$ determines the number of states, $|\mathcal{U}|$ determines the number of next state transitions from a given state. The signal inputs $\{u_k\}$ are i.i.d. discrete random variables with probability function

$$q(u),\ u \in \mathcal{U}$$

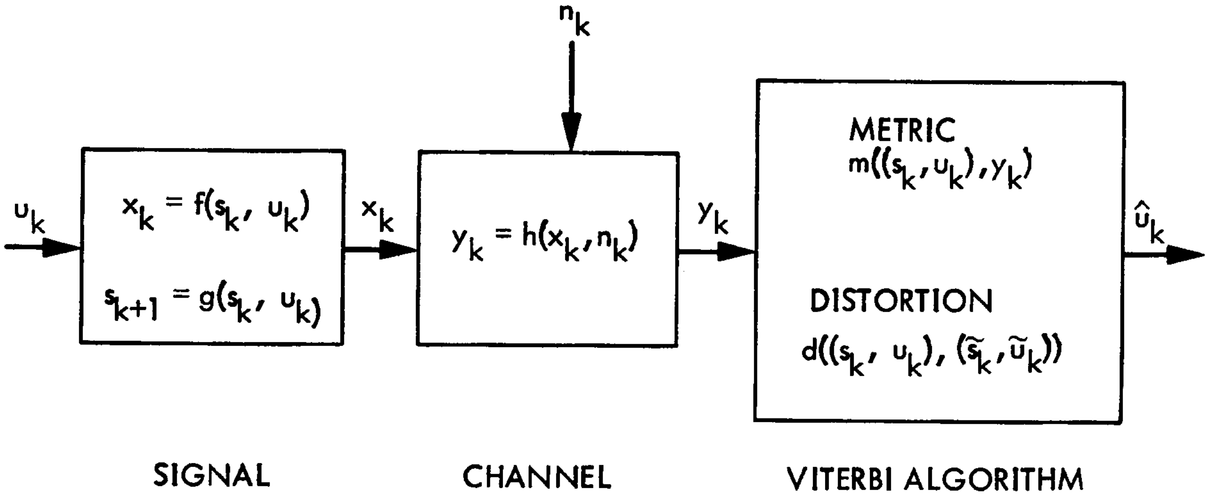


Figure A-1. Discrete-Time System Model

The channel or observation is described by

$$y_k = h(x_k, n_k) \tag{A.2.3}$$

where $\{n_k\}$ are i.i.d. random variables independent of the signal inputs $\{u_k\}$. Here n_k and y_k can be continuous or discrete valued.

The receiver is described by a Viterbi algorithm which uses a metric

$$m((s_k, u_k), y_k) \tag{A.2.4}$$

for the branches of the trellis diagram. This metric may correspond to many possible forms such as:

(a) Maximum Likelihood (ML):

$$m((s_k, u_k), y_k) = \log_e p(y_k | (s_k, u_k))$$

$$= \log_e p(y_k | x_k) \tag{A.2.5a}$$

(b) Maximum A Posteriori (MAP):

$$m((s_k,u_k),y_k) = \log_e [p(y_k|(s_k,u_k))q(u_k)]$$

$$= \log_e p(y_k|x_k) + \log_e q(u_k) \qquad \text{(A.2.5b)}$$

(c) Minimum Mean Square Error (MSE):

$$m((s_k,u_k),y_k) = -(y_k - x_k)^2 \qquad \text{(A.2.5c)}$$

Independent of the metric used by the Viterbi algorithm, we may wish to evaluate the overall performance using a distortion measure $d((\tilde{s}_k,\tilde{u}_k), (s_k,u_k))$. This measure may be any nonnegative function such as:

(a) Error Distortion:

$$d((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) = \begin{cases} 1; & \tilde{u}_k \neq u_k \\ 0 & \tilde{u}_k = u_k \end{cases} \qquad \text{(A.2.6a)}$$

(b) Mean Square Error:

$$d((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) = \alpha(\tilde{s}_k - s_k)^2 + \beta(\tilde{u}_k - u_k)^2$$

$$\text{for any } \alpha,\ \beta \geq 0 \qquad \text{(A.2.6b)}$$

In the usual convolutional coding application of the Viterbi algorithm the ML metric is used and the error distortion measure gives the bit error probability bound. If, on the other hand, we wish to estimate the phase of a signal that is modeled as a Markov chain, the MAP metric might be used in the Viterbi algorithm and the mean square error distortion measure would give the resulting mean square error bound. Although there is a natural relationship between the metric used by the Viterbi algorithm and the distortion measure used for evaluating performance, we do not require any connection between these quantities. Indeed, for a given metric, we shall consider cases where we evaluate performance in terms of two different distortion measures. For example, when a Viterbi algorithm is used to simultaneously estimate phase and demodulate data, we would be interested in both the mean square phase error and the bit error probability.

III. The Viterbi Algorithm

Let us assume that the discrete-time system of Figure A-1 begins at t=0 with initial state s_0 known to the receiver. The receiver then uses the channel output sequence y_0, y_1, y_2, ... to estimate the particular state sequence $\hat{s}_1$, $\hat{s}_2$, $\hat{s}_3$, ... or equivalently the particular signal input sequence $\hat{u}_0$, $\hat{u}_1$, $\hat{u}_2$, ... that maximizes the total metric

$$\sum_{k=0}^{\infty} m((\tilde{s}_k,\tilde{u}_k),y_k)$$

over all possible sequences $\{(\tilde{s}_k,\tilde{u}_k)\}$.

The Viterbi algorithm is an optimum algorithm for any additive metric and a finite state signal model. The key to understanding this algorithm is the trellis diagram description of the signal process. Suppose for example we have

$$|S| = 4$$

$$|U| = 3 \tag{A.3.1}$$

The state diagram for the signal process might then be as shown in Figure A-2 where each of 4 nodes denotes a state and there are 3 next state transitions. If we were to give a time-sequence of the possible state transitions starting with some initial state then we have the corresponding trellis diagram of Figure A-3. The key point here is that all possible sequences $\{(\tilde{s}_k,\tilde{u}_k)\}$ are represented by paths in the trellis diagram.

Suppose now we have a trellis diagram with M states,

$$S = \{\Delta_1,\Delta_2, \ldots\ldots,\Delta_M\} \tag{A.3.2}$$

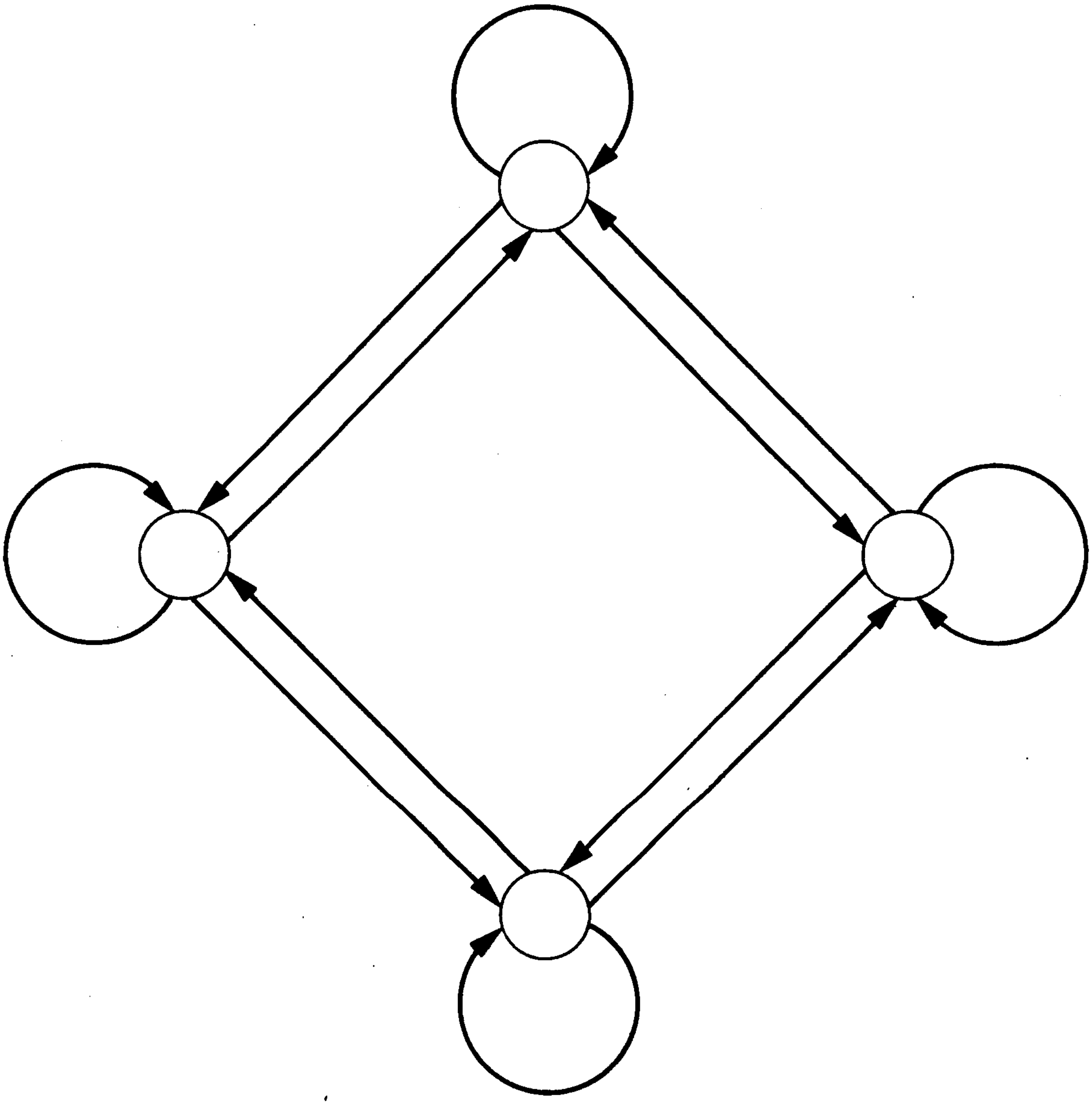


Figure A-2. State Diagram $|U| = 3$, $|S| = 4$

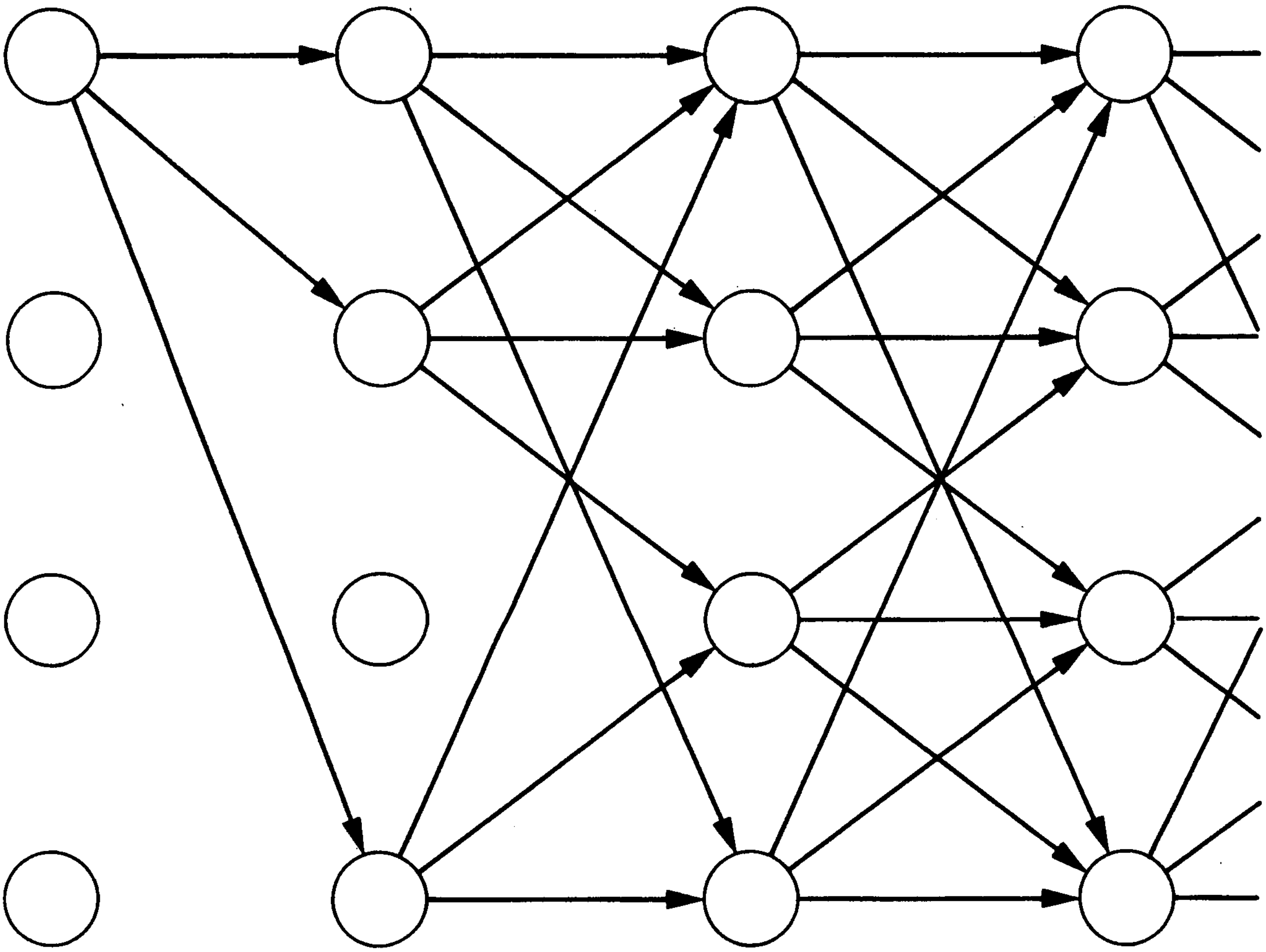


Figure A-3. Trellis Diagram $|U| = 3$, $|S| = 4$

A typical path is sketched in Figure A-4. As the receiver receives the channel output sequence y_0, y_1 ... , it can compute a metric value for each branch or transition from state to state along this path. Thus, in this way the total metric up to time $t = n+1$ is

$$\sum_{k=0}^{n} m((s_k, u_k), y_k)$$

for the particular sequence $\{(s_k, u_k)\}$ which defines the path.

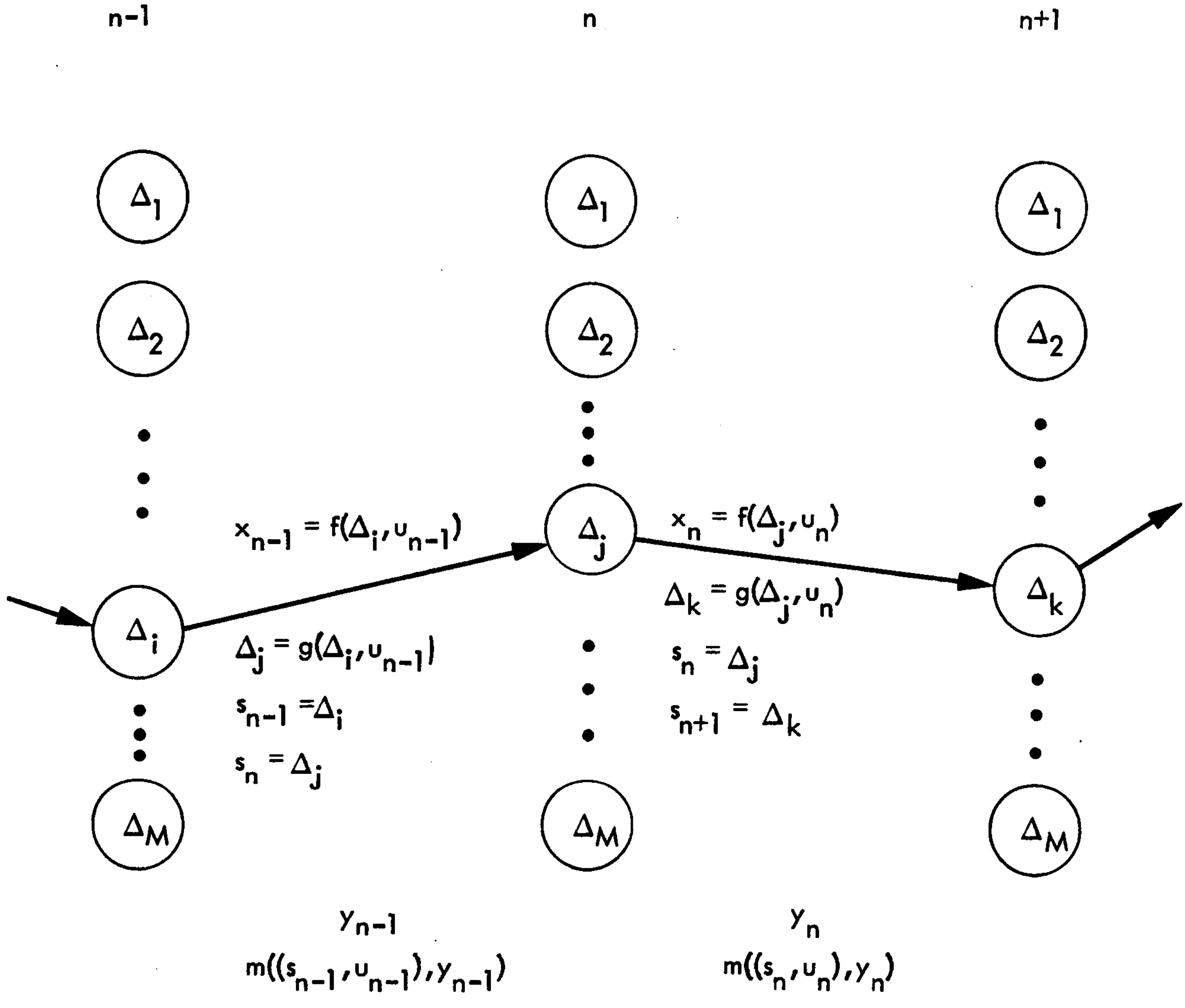


Figure A-4. Typical Path and Metric

The optimum receiver with respect to this additive metric considers all paths in the trellis diagram and as $t\to\infty$ chooses the path which corresponds to the maximum total metric. The key feature of the Viterbi algorithm is the elimination of paths without loss in optimality whenever two or more paths merge to the same state.

In Figure A-5 we show this elimination of paths characteristic of the Viterbi algorithm. Suppose that two paths $\{(s_k, u_k)\}$ and $\{(\tilde{s}_k, \tilde{u}_k)\}$ merge at time $t = n+1$ to state $\tilde{s}_{n+1} = s_{n+1} = \Delta_k$ as shown in Figure A-5. Then the metrics accumulated up to this point are

$$\sum_{k=0}^{n} m((s_k, u_k), y_k)$$

and

$$\sum_{k=0}^{n} m((\tilde{s}_k, \tilde{u}_k), y_k)$$

Note that any remaining segment of the two paths starting at state Δ_k at $t = n+1$ can be the same for either initial sequence. Since we are only interested in finding any maximum metric sequence, without any loss of optimality we can eliminate one of these two initial path sequences from further consideration, namely, the one with the smaller accumulated metric. Thus, for example, if

$$\sum_{k=0}^{n} m((s_k, u_k), y_k) \geq \sum_{k=0}^{n} m((\tilde{s}_k, \tilde{u}_k), y_k) \tag{A.3.3}$$

then we can eliminate the initial path sequence $\{(\tilde{s}_k, \tilde{u}_k)\}_{k=0}^{n}$ from further consideration. When more than two paths merge to one state we can eliminate all but one path from further consideration and keep only the one with the largest accumulated metric.

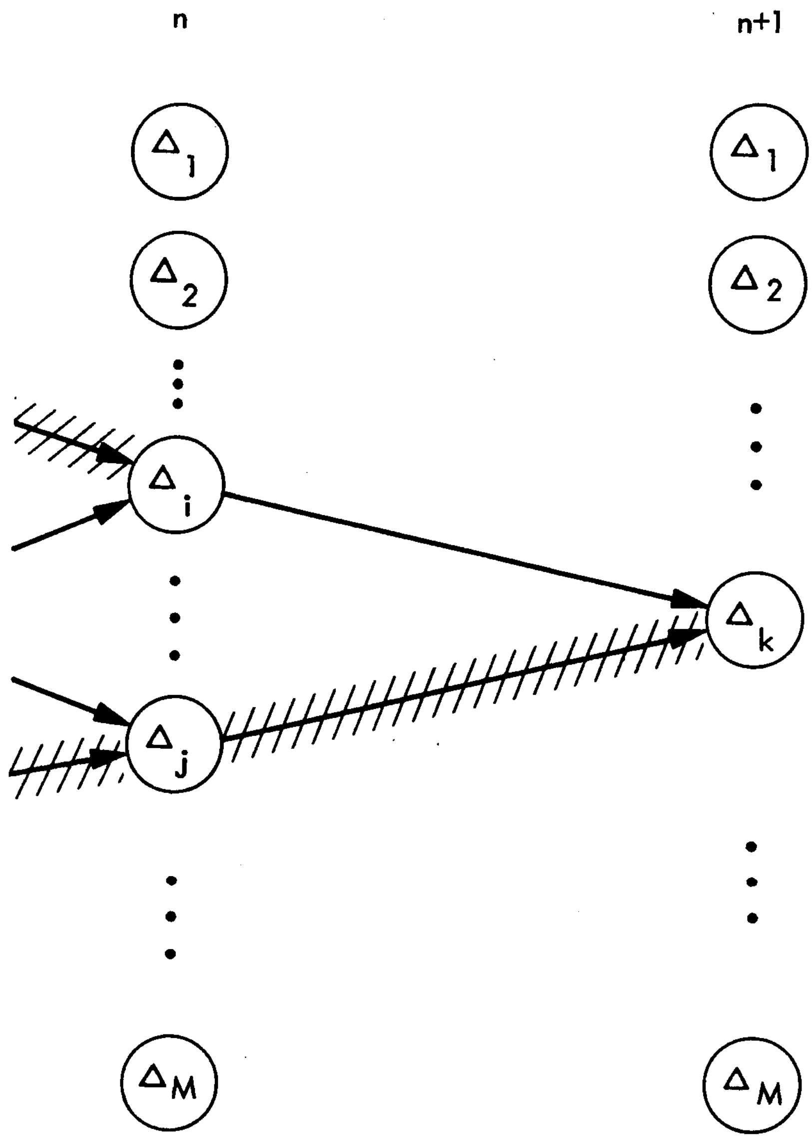


Figure A-5. Elimination of Paths

Since there are only a finite number of states

$$M = |\mathcal{S}|, \qquad (A.3.4)$$

at most M paths are retained by the Viterbi algorithm as the channel output sequence $y_0, y_1, \ldots$ is received. This is in contrast to the number of possible paths $|\mathcal{U}|^n$ up to time $t = n$. By eliminating paths that are not maximum metric each time paths merge in the trellis, the Viterbi algorithm reduces the computation to roughly M rather than an exponential growth with time.

Another important feature of the Viterbi algorithm is that for all metrics of interest, there is negligible loss of optimality associated with making final decisions concerning the maximum metric paths at some fixed lag time as channel outputs are received. This is illustrated in Figure A-6 where we assume that the channel output at time $t = \ell$ is being processed by the Viterbi algorithm so that the M surviving paths are computed up to this time. Typically the M surviving paths, one of which is the true maximum metric path, share common initial parts. By considering a large enough lag time L, then with high probability only one initial part remains for all M paths at this lag time $t = \ell - L$. For convolutional codes the choice (Ref. 1)

$$L \geq 5 \log_2 M \qquad (A.3.5)$$

is large enough to guarantee negligible loss in performance. The Viterbi algorithm is thus practically realized as a fixed lag estimator of the sequence $\{(\hat{s}_k, \hat{u}_k)\}$ that maximizes the total metric

$$\sum_{k=0}^{\infty} m((s_k, u_k), y_k)$$

as it receives from the channel the observations $y_0, y_1, y_2, \ldots$.

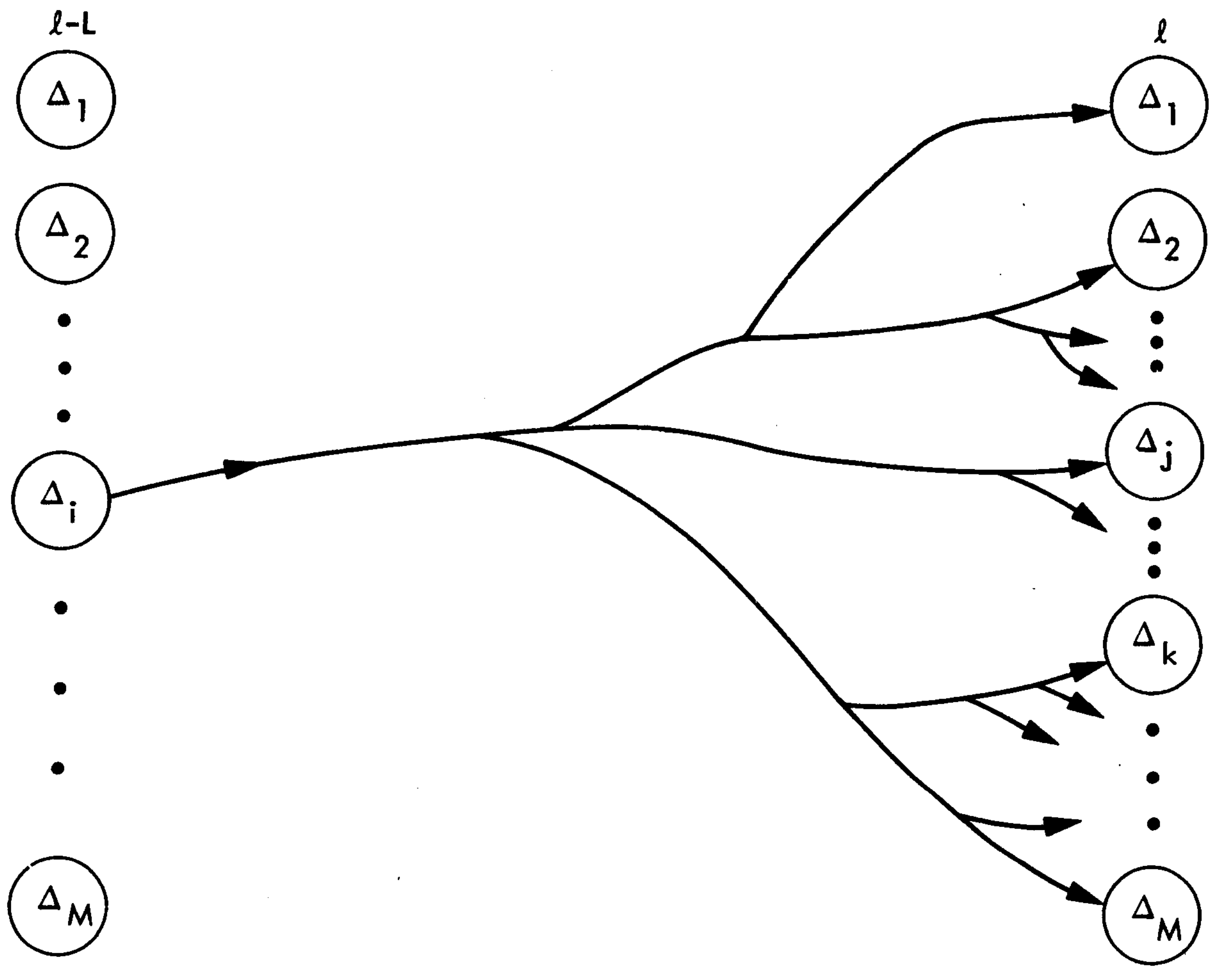


Figure A-6. Fix Lag Decisions

IV. Error Events

The performance of the discrete-time system of Figure A-1 is defined by a distortion measure

$$d((\hat{s}_n,\hat{u}_n),(s_n,u_n))$$

where (s_n, u_n) is the true signal state and input at time $t = n$ while $(\hat{s}_n, \hat{u}_n)$ is the state and input selected by the Viterbi algorithm for the same time. Without loss in generality, we assume this distortion measure is nonnegative and in addition,

$$d((\hat{s}_n,\hat{u}_n),(\hat{s}_n,\hat{u}_n)) = 0 \tag{A.4.1}$$

The condition $(\hat{s}_n, \hat{u}_n) \neq (s_n, u_n)$ can only occur when the Viterbi algorithm eliminates a segment of the true path that includes the state s_n. When this happens we have an <u>error event</u> which is characterized by, say, times i and j where $i \leq n < j$ and

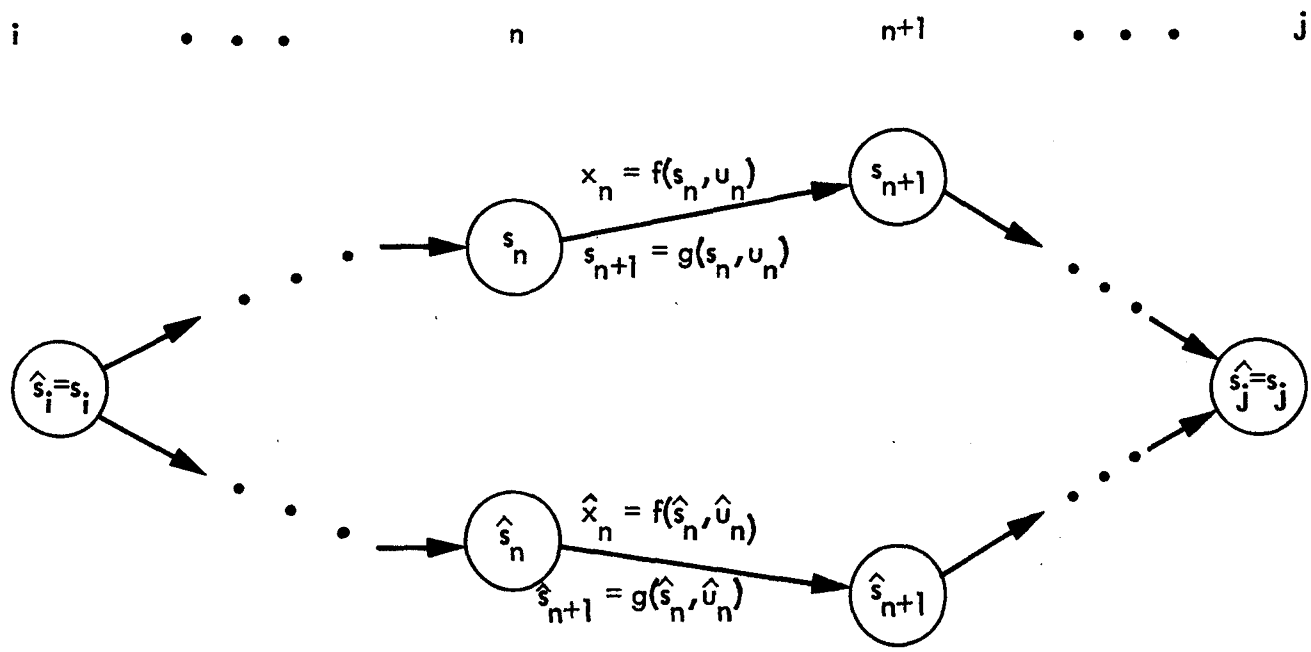
$$\hat{s}_i = s_i,\ \hat{s}_j = s_j$$

$$\hat{s}_k \neq s_k;\ i < k < j$$

$$\sum_{k=1}^{j-1} m((\hat{s}_k,\hat{u}_k),y_k) \geq \sum_{k=1}^{j-1} m((s_k,u_k),y_k) \tag{A.4.2}$$

Figure A-7 illustrates such an error event.

In general for fixed time $t = n$, there are many possible error events that can lead to the condition $(\hat{s}_n, \hat{u}_n) \neq (s_n, u_n)$. The beginning of an error event at time i can be anywhere from $t = 0$ to $t = n$ while the end of an error event can range from $t = n+1$ to $t = \infty$. In the subsequent analysis we shall upper bound our performance by assuming a steady state condition where $t = n>>0$ is assumed so that we allow the initial time of an error event to range from



METRIC: $\sum_{k=i}^{j-1} m((\hat{s}_k, \hat{u}_k), y_k) > \sum_{k=i}^{j-1} m((s_k, u_k), y_k)$

DISTORTION: $d((\hat{s}_k, \hat{u}_k), (s_k, u_k))$

Figure A-7. Error Event

$t = -\infty$ to $t = n$. This will result in an upper bound on performance since in considering error events we include more of these than are necessary.

We now examine the probability of the occurrence of a particular error event. Again let $\{(s_k, u_k)\}$ be the true signal state and input sequence. Suppose $\{(\tilde{s}_k, \tilde{u}_k)\}$ is any other possible state and input sequence where for $i \le n < j$ we have

$$\tilde{s}_i = s_i, \ \tilde{s}_j = s_j$$

$$\tilde{s}_k \neq s_k; \ i < k < j \tag{A.4.3}$$

Since the error event only involves subsequences from i to j, we denote these as

$$\underline{s}[i,j] = (s_i, s_{i+1}, \ldots\ldots, s_j)$$

$$\underline{\tilde{s}}[i,j] = (\tilde{s}_i, \tilde{s}_{i+1}, \ldots\ldots, \tilde{s}_j) \tag{A.4.4}$$

We now bound the probability of this error event denoted by

$$P(\underline{s}[i,j] \rightarrow \underline{\tilde{s}}[i,j]) = \Pr\left\{\sum_{k=i}^{j-1} m((\tilde{s}_k,\tilde{u}_k),y_k) \geq \sum_{k=i}^{j-1} m((s_k,u_k),y_k) \,\middle|\, \underline{s},\underline{\tilde{s}}\right\}$$

$$= \Pr\left\{\sum_{k=i}^{j-1} [m((\tilde{s}_k,\tilde{u}_k),y_k) - m((s_k,u_k),y_k)] \geq 0 \,\middle|\, \underline{s},\underline{\tilde{s}}\right\} \tag{A.4.5}$$

where the probability is over the channel noise sequence $\{n_k\}$; $i \leq k<j$. Using the Chernoff bound (Ref. 2) with parameter $\lambda\geq 0$, and noting that the random variables $\{n_k\}$ are independent, we have

$$P(\underline{s}[i,j] \rightarrow \underline{\tilde{s}}[i,j]) \leq E\left\{\exp\left[\lambda \sum_{k=i}^{j-1} \{m((\tilde{s}_k,\tilde{u}_k),y_k) - m((s_k,u_k),y_k)\}\right] \,\middle|\, \underline{s},\underline{\tilde{s}}\right\}$$

$$= \prod_{k=i}^{j-1} E\left\{\exp\left[\lambda\{m((\tilde{s}_k,\tilde{u}_k),y_k) - m((s_k,u_k),y_k)\}\right] \,\middle|\, \underline{s},\underline{\tilde{s}}\right\}$$

$$= \prod_{k=i}^{j-1} D_\lambda((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) \tag{A.4.6}$$

where

$$D_\lambda((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) = E\left\{\exp\left[\lambda\{m((\tilde{s}_k,\tilde{u}_k),y_k) - m((s_k,u_k),y_k)\}\right] \middle| \underline{s},\underline{\tilde{s}}\right\} \tag{A.4.7}$$

The function $D_\lambda((\tilde{s}_k,\tilde{u}_k),\ (s_k,u_k))$ can, in general, be numerically evaluated and has some well-known special cases. In particular, for the ML metric of (A.2.5a), it can be shown that $\lambda = 1/2$ almost always minimizes the Chernoff bound resulting in the Bhattacharyya bound (Ref. 1)

$$D_{\frac{1}{2}}((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) = \sum_{y_k} \sqrt{p(y_k|x_k)p(y_k|\tilde{x}_k)} \tag{A.4.8}$$

In arriving at (A.4.8), we have made use of the fact that the statistical expectation in (A.4.7) is taken over the conditional probability distribution $p(y_k|x_k)$.

V. Average Distortion

Next we consider the set of all error events beginning at i and ending at j by defining subsequences

$$S(i,j|\underline{s}[i,j]) \triangleq \{\underline{\tilde{s}}(i,j]:\ \tilde{s}_i = s_i,\ \tilde{s}_j = s_j,\ \tilde{s}_k \neq s_k,\ i < k < j\} \tag{A.5.1}$$

Note that for any subsequence

$$\underline{\tilde{s}}[i,j] \in \ (i,j|\underline{s}[i,j])$$

The distortion at time t=n is

$$d((\tilde{s}_n,\tilde{u}_n),(s_n,u_n))$$

Hence the average distortion between the maximum metric sequence $\{(\hat{s}_k,\hat{u}_k)\}$ and the actual sequence $\{(s_k, u_k)\}$ at time t=n is bounded as follows:

$$E\{d((\hat{s}_n,\hat{u}_n),(s_n,u_n))|\underline{s}\} \le \sum_{i\le n}\sum_{j>n}\sum_{S(i,j|\underline{s}[i,j])} d((\tilde{s}_n,\tilde{u}_n),(s_n,u_n)) \Pr(\hat{\underline{s}}[i,j] = \tilde{s}[i,j]|s[i,j])$$

$$\le \sum_{i\le n}\sum_{j>n}\sum_{S(i,j|\underline{s}[i,j])} d((\tilde{s}_n,\tilde{u}_n),(s_n,u_n)) P(\underline{s}[i,j]\rightarrow \underline{\tilde{s}}[i,j]) \qquad (A.5.2)$$

The inequality in (A.5.2) comes about because $\Pr(\hat{\underline{s}}[i,j] = \underline{\tilde{s}}[i,j]|\underline{s}[i,j])$, the probability that $\tilde{s}[i,j]$ has the maximum metric of all error event subsequences, is less than $P(\underline{s}[i,j]\rightarrow\underline{\tilde{s}}[i,j])$ which is the probability that $\underline{\tilde{s}}[i,j]$ has a greater metric than only that of the true subsequence $\underline{s}[i,j]$.

In general we are interested in the above distortion averaged over all true state subsequences $\{\underline{s}[i,j]\}$. For the special case of convolutional codes over symmetric channels, the bound is independent of the particular state subsequence $\underline{s}[i,j]$ and a transfer function bound is easily obtained. In the more general case of interest here, we should average over all possible true signal state sequences. In performing this average, we recognize that any true state subsequence represents a first order Markov chain and thus is characterized by the probability distribution

$$q(\underline{s}[i,j]) = p(s_i)q(u_i)q(u_{i+1}) \cdots q(u_{j-1}) \qquad (A.5.3)$$

where $p(s_i)$ is the steady state probability of state s_i.

Next we define the set of subsequence pairs

$$S(i,j) \triangleq \{(\underline{\tilde{s}}[i,j],s[i,j]):\ \tilde{s}_i = s_i,\ \tilde{s}_j = s_j,\ \tilde{s}_k \ne s_k;\ i < k < j\} \quad (A.5.4)$$

each pair consisting of an error event subsequence and the true state subsequence. Then averaging (A.5.2) over all subsequences $\underline{s}[i,j]$ yields*,

$$E\,\{d((\hat{s}_n,\hat{u}_n),(s_n,u_n))\} = \sum_{\underline{s}} q(\underline{s})\; E\{d((\hat{s}_n,\hat{u}_n),(s_n,u_n))|\underline{s}\}$$

$$\le \sum_{\underline{s}} q(\underline{s}) \sum_{i\le n}\sum_{j>n}\sum_{S(i,j|\underline{s}[i,j])} d((\tilde{s}_n,\tilde{u}_n),(s_n,u_n))\; P(\underline{s}[i,j] \to \underline{\tilde{s}}[i,j])$$

$$= \sum_{i\le n}\sum_{j>n}\sum_{S(i,j)} q(\underline{s}[i,j])\; d((\tilde{s}_n,\tilde{u}_n),(s_n,u_n))\; P(\underline{s}[i,j] \to \underline{\tilde{s}}[i,j]) \qquad \text{(A.5.5)}$$

Using (A.5.3) and the bound (A.4.6) in this expression yields the bound

$$E\,\{d((\tilde{s}_n,\tilde{u}_n),(s_n,u_n))\} \le \sum_{i\le n}\sum_{j>n}\sum_{S(i,j)} d((\tilde{s}_n,\tilde{u}_n),(s_n,u_n))\; p(s_i) \prod_{k=i}^{j-1} q(u_k)\; D_\lambda((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) \qquad \text{(A.5.6)}$$

Since we have steady state conditions, the above bound is the same for all n; that is, it is independent of n. Because of this invariance to time shifts, we can express the bound in (A.5.6) in another more compact form. Suppose we consider two subsequences

$$\underline{\tilde{s}}[i,j],\underline{s}[i,j] \;\varepsilon\; S(i,j)$$

As illustrated in Figure A-8, these two subsequences when shifted by L and denoted

$$\underline{\tilde{s}}[i + L,j + L],\underline{s}[i + L,j + L] \;\varepsilon\; S(i + L,j + L)$$

*For simplicity of notation, we shall where convenient drop the dependence of $\underline{s}$ on i and j.

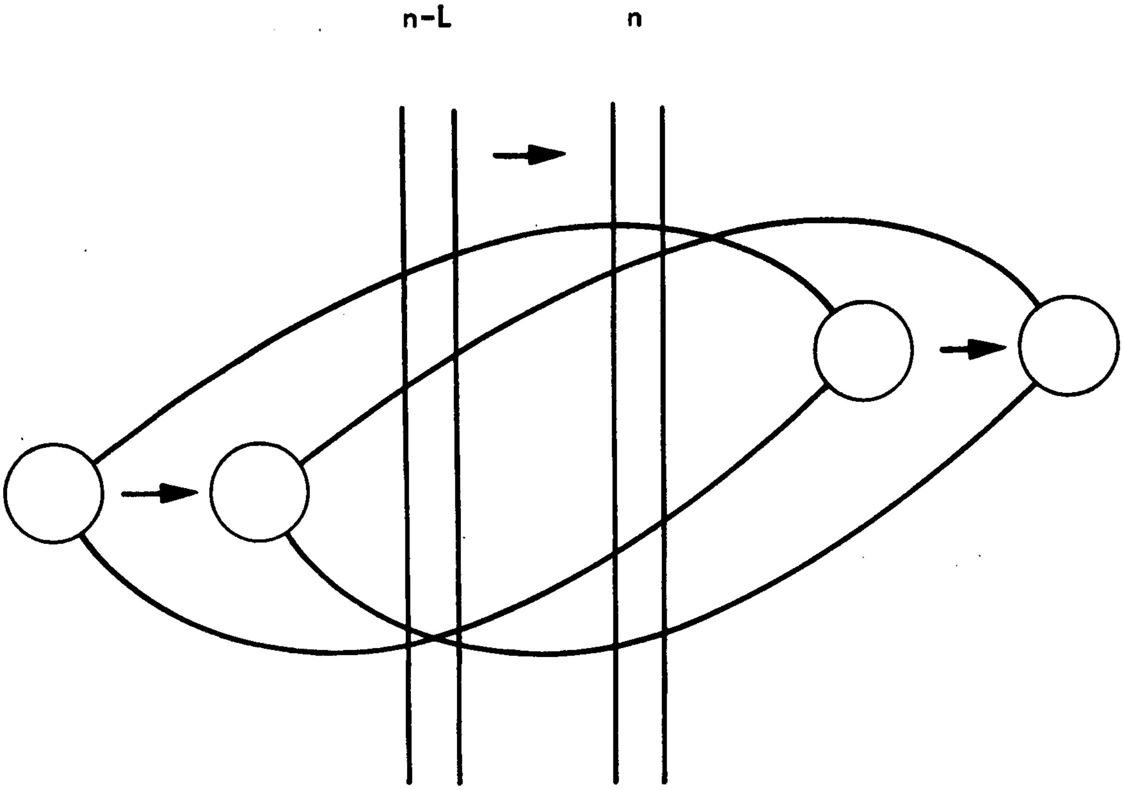


Figure A-8. Shift by L

are also considered in the average distortion provided L satisfies

$$i + L \leq n \leq j + L$$

which are analogous to the conditions on i and j just prior to (A.4.3). Also note that we have the conditions requisite to being stationary:

$$q(\underline{s}[i,j]) = q(\underline{s}[i + L,j + L]) \tag{A.5.7}$$

and

$$P(\underline{s}[i,j] \to \tilde{s}[i,j]) = P(\underline{s}[i + L,j + L] \to \tilde{s}[i + L,j + L]) \tag{A.5.8}$$

We can thus include the distortion due to $\underline{\tilde{s}}[i + L, j + L]$ and $\underline{s}[i + L, j + L]$ at time t=n by considering the distortion

$$d((\tilde{s}_{n-L},\tilde{u}_{n-L}),(s_{n-L},u_{n-L}))$$

due to

$$\underline{\tilde{s}}[i,j],\underline{s}[i,j] \in \mathcal{S}(i,j)$$

This means we can replace all shifts of the set $\mathcal{S}(i,j)$ to $\mathcal{S}(i + L, j + L)$ by including the additional distortion at time $t = n - L$.

Thus, for each term in (A.5.6) corresponding to a given i, j, and n, we can equivalently shift these indices to the left by i, and consider i always fixed at zero and j replaced by j - i and n likewise replaced by n - i. Hence, the double sum in (A.5.6) over the region $i \le n < j$ is equivalent to a double sum in which the first sum runs over fixed $j - i = 1, 2, 3, \ldots$ and the second sum runs over $0 \le n - i < j - i$ or $0 \le n - i \le j - i - 1$. Then, letting $\ell = n - i$ and for simplicity using j to denote j - i, (A.5.6) becomes

$$E\,\{d((\hat{s}_n,\hat{u}_n),(s_n,u_n))\} \le \sum_{j=1}^{\infty} \sum_{\mathcal{S}(0,j)} \left[\sum_{\ell=0}^{j-1} d((\tilde{s}_\ell,\tilde{u}_\ell),(s_\ell,u_\ell))\right] p(s_0)$$

$$\times \prod_{k=0}^{j-1} q(u_k)\; D_\lambda((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) \tag{A.5.9}$$

Note that as stated above, we have also set i = 0 in $\mathcal{S}(i,j)$ of (A.5.6). Also, note that, in this form, the bound shows no dependence on the time n. To emphasize the independence of n the steady state expected distortion is denoted

$$\overline{d} \triangleq E\{d((\hat{s}_n,\hat{u}_n),(s_n,u_n))\} \tag{A.5.10}$$

One final step is required to obtain a transfer function bound. Applying the identity

$$x = \frac{d}{dz} z^{x} \Big|_{z=1} \tag{A.5.11}$$

to the sum on ℓ in (A.5.9) results in

$$\left[\sum_{\ell=0}^{j-i} d((\tilde{s}_\ell,\tilde{u}_\ell),(s_\ell,u_\ell))\right] = \frac{d}{dz} z^{\left[\sum_{\ell=0}^{j-1} d((\tilde{s}_\ell,\tilde{u}_\ell),(s_\ell,u_\ell))\right]}\Bigg|_{z=1}$$

$$= \frac{d}{dz}\left\{\prod_{\ell=0}^{j-1} z^{d((\tilde{s}_\ell,\tilde{u}_\ell),(s_\ell,u_\ell))}\right\}\Bigg|_{z=1} \tag{A.5.12}$$

Finally, substituting (A.5.12) in (A.5.9) and noting that the product on k in (A.5.9) is independent of z, we obtain the desired result, namely,

$$\boxed{\bar{d} \le \frac{d}{dz} T(z)\Big|_{z=1}} \tag{A.5.13}$$

where the <u>transfer function</u> T(z) is given by

$$\boxed{T(z) = \sum_{j=1}^{\infty} \sum_{S(0,j)} p(s_0) \prod_{k=0}^{j-1} z^{d(\tilde{s}_k,\tilde{u}_k),(s_k,u_k))} q(u_k)\; D_\lambda((\tilde{s}_k,\tilde{u}_k),(s_k,u_k))} \tag{A.5.14}$$

VI. Evaluation of the Transfer Function

We now examine the problem of evaluating T(z). First suppose the states are given by

$$S = \{\Delta_1, \Delta_2, \ldots\ldots, \Delta_M\} \tag{A.6.1}$$

and the signal input alphabet by

$$U = \{a_1, a_2, \ldots\ldots, a_m\} \tag{A.6.2}$$

We next define "super states" as elements of $S^2 = S \times S$ where

$$S^2 = \{\delta_1, \delta_2, \ldots\ldots, \delta_{M^2}\} \tag{A.6.3}$$

Also define the "super signal input" alphabet $U^2 = U \times U$, where

$$U^2 = \{\alpha_1, \alpha_2, \ldots\ldots, \alpha_{m^2}\} \tag{A.6.4}$$

Then at time t = k we denote "superstates"

$$S_k = (s_k, \tilde{s}_k) \ \varepsilon \ S^2 \tag{A.6.5}$$

and super inputs

$$U_k = (u_k, \tilde{u}_k) \ \varepsilon \ U^2 \tag{A.6.6}$$

Next, the super states S^2 are split into two disjoint subsets, namely

$$S_A^2 = \{\delta_1, \delta_2, \ldots\ldots, \delta_M\} \tag{A.6.7}$$

which contains the M equal-component super states

$$\delta_\ell = (\Delta_\ell, \Delta_\ell);\ \ell = 1.2,\ \ldots,\ M \tag{A.6.8}$$

and

$$\mathcal{S}_B^2 = \{\delta_{M+1}, \delta_{M+2},\ \ldots.,\ \delta_{M^2}\} \tag{A.6.9}$$

which contains the unequal-component super states.

With this definition, note that in accordance with

$$\begin{aligned} \mathcal{S}(0,j) &= \{(\underline{s}[0,j], \underline{\tilde{s}}[0,j]):\ \tilde{s}_0 = s_0, \tilde{s}_j = s_j, \tilde{s}_k \neq s_k;\ 0 < k < j\} \\ &= \{\underline{S}[0,j]:\ S_0, S_j \in \mathcal{S}_A^{\ 2},\ S_k \in \mathcal{S}_B^{\ 2};\ 0 < k < j\} \end{aligned} \tag{A.6.10}$$

where

$$\underline{S}[0,j] = (S_0, S_1,\ \ldots., S_j) \tag{A.6.11}$$

Next, we use some shorthand notation where the state equations,

$$\begin{aligned} \tilde{s}_{k+1} &= g(\tilde{s}_k, \tilde{u}_k) \\ s_{k+1} &= g(s_k, u_k) \end{aligned} \tag{A.6.12}$$

are expressed as

$$S_{k+1} = G(S_k, U_k)$$

and we define

$$\begin{aligned} p(S_0) &\triangleq p(s_0) \\ d(S_k, U_k) &\triangleq d((\tilde{s}_k, \tilde{u}_k), (s_k, u_k)) \\ q(U_k) &\triangleq q(u_k) \\ D_\lambda(S_k, U_k) &\triangleq D_\lambda((\tilde{s}_k, \tilde{u}_k), (s_k, u_k)) \end{aligned} \tag{A.6.13}$$

Then the transfer function (T(z) of (A.5.14) can be rewritten as

$$T(z) = \sum_{j=1}^{\infty} \sum_{S(0,j)} p(S_0) \prod_{k=0}^{j-1} z^{d(S_k,U_k)} q(U_k) D_\lambda (S_k,U_k) \qquad \text{(A.6.14)}$$

Note, that in the above form, T(z) can be interpreted as a transfer function for the super state diagram of Figure A-9. Here T(z) is the sum of all paths in Figure A-9 each starting with an initial state belonging to S_A^2 and terminating in a final state also belonging to S_A^2 while all intermediate states are those belonging to S_B^2.* The transfer function label of the branch from state δ_i to state δ_j is called a_{ij} where

$$\delta_i \xrightarrow{\qquad a_{ij} \qquad} \delta_j$$

$$a_{ij} = \begin{cases} z^{d(\delta_i,U)} q(U) D_\lambda (\delta_i,U); & \text{if } U \varepsilon \mathcal{U}^2 \text{ exists such that } \delta_j = G(\delta_i,U) \\ 0; & \text{if not} \end{cases} \qquad \text{(A.6.15)}$$

The transfer function can be expressed in matrix form by defining $t_i(z)$; $i = M+1, M+2, \ldots, M^2$ as the transfer function from all initial states to the single intermediate state $\delta_i \varepsilon S_B^2$. Defining the $(M^2-M) \times (M^2-M)$ matrix

*For j = 1, there are no intermediate states.

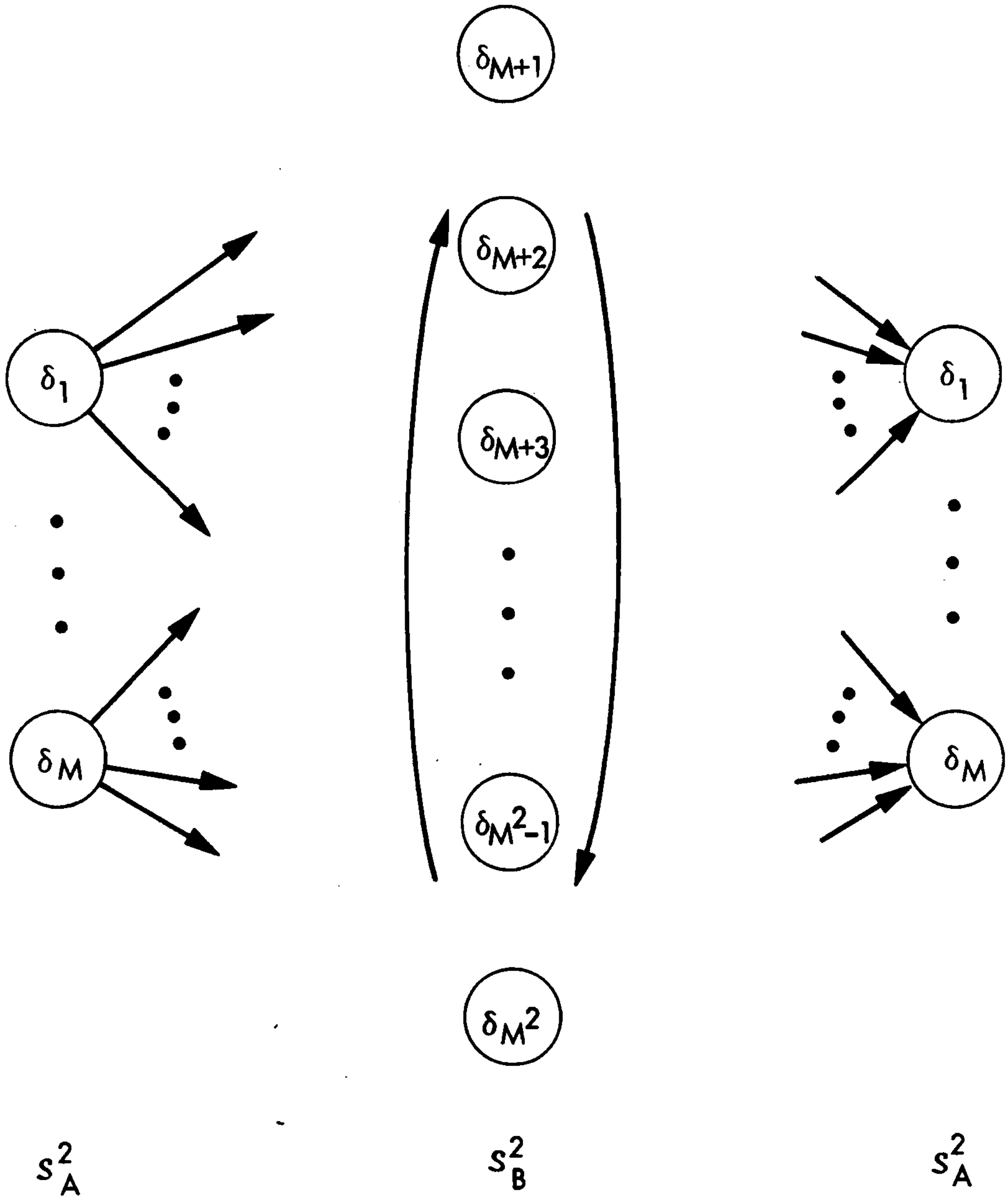


Figure A-9. Super State Diagram

$$
\underline{A} = \begin{bmatrix} a_{M+1,M+1} & a_{M+2,M+1} & \cdots\cdots & a_{M^2,M+1} \\ a_{M+1,M+2} & a_{M+2,M+2} & \cdots\cdots & a_{M^2,M+2} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{M+1,M^2} & a_{M+2,M^2} & \cdots\cdots & a_{M^2,M^2} \end{bmatrix} \tag{A.6.16}
$$

and the vectors

$$
\underline{b}_i = \begin{bmatrix} a_{i,M+1} \\ a_{i,M+2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_{i,M^2} \end{bmatrix}, \quad \underline{c}_i = \begin{bmatrix} a_{M+1,i} \\ a_{M+2,i} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_{M^2,i} \end{bmatrix}; \quad i=1,2,\ \ldots\ldots,M \tag{A.6.17}
$$

then intermediate state transfer function vector

$$\underline{t}(z) = \begin{bmatrix} t_{M+1}(z) \\ t_{M+2}(z) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ t_{M^2}(z) \end{bmatrix} \qquad \text{(A.6.18)}$$

satisfies

$$\underline{t}(z) = \underline{A}\ \underline{t}(z) + \sum_{i=1}^{M} p(\delta_i)\underline{b}_i \qquad \text{(A.6.19)}$$

or

$$\underline{t}(z) = (\underline{I} - \underline{A})^{-1} \sum_{i=1}^{M} p(\delta_i)\underline{b}_i \qquad \text{(A.6.20)}$$

where I is the $(M^2-M) \times (M^2-M)$ identity matrix. The total transfer function is given by

$$\boxed{T(z) = \sum_{j=1}^{M} \underline{c}_j^T \underline{t}(z) = \left(\sum_{j=1}^{M} \underline{c}_j\right)^T (\underline{I} - \underline{A})^{-1} \sum_{i=1}^{M} p(\delta_i)\underline{b}_i} \qquad \text{(A.6.21)}$$

where the superscript T denotes transpose.

We next consider taking the derivative of T(z) where we denote

$$\underline{c}'_j = \frac{d}{dz}\underline{c}_j$$

$$\underline{b}'_i = \frac{d}{dz}\underline{b}_i \tag{A.6.22}$$

and

$$\underline{A}' = \frac{d}{dz}\underline{A} \tag{A.6.23}$$

The understanding here is that the derivative is taken term by term in each vector and matrix. Also using the identify

$$\underline{I} = (\underline{I} - \underline{A})^{-1}(\underline{I} - \underline{A}) \tag{A.6.24}$$

we have

$$0 = \frac{d}{dz}\{(\underline{I} - \underline{A})^{-1}(\underline{I} - \underline{A})\}$$

$$= (\underline{I} - \underline{A})^{-1}(-\frac{d}{dz}\underline{A}) + (\underline{I} - \underline{A})\left(\frac{d}{dz}(\underline{I} - \underline{A})^{-1}\right) \tag{A.6.25}$$

or

$$\frac{d}{dz}(\underline{I} - \underline{A})^{-1} = (\underline{I} - \underline{A})^{-1}\underline{A}'(\underline{I} - \underline{A})^{-1} \tag{A.6.26}$$

Thus, using (A.6.26)

$$\frac{dT(z)}{dz} = \left(\sum_{j=1}^{M} \underline{c}'_j\right)^T (\underline{I} - \underline{A})^{-1} \left(\sum_{i=1}^{M} p(\delta_i)\underline{b}_i\right)$$

$$+ \left(\sum_{j=1}^{M} c_j\right)^T (\underline{I} - \underline{A})^{-1} \left(\sum_{i=1}^{M} p(\delta_i)\underline{b}'_i\right)$$

$$+ \left(\sum_{j=1}^{M} \underline{c}_j\right)^T (\underline{I} - \underline{A})^{-1} \underline{A}' (\underline{I} - \underline{A})^{-1} \left(\sum_{i=1}^{M} p(\delta_i)\underline{b}_i\right) \quad \text{(A.6.27)}$$

which enables us to evaluate the bound on $\overline{d}$ given in (A.5.13). This evaluation is limited only by the ability to evaluate

$$(\underline{I} - \underline{A})^{-1} = \underline{I} + \underline{A} + \underline{A}^2 + \underline{A}^3 + \ldots.. \quad \text{(A.6.28)}$$

The complexity of computing $\underline{A}^k$ is determined by the number of nonzero elements in $\underline{A}$. Roughly 2^{15} nonzero elements can be handled by a large general purpose computer.

Finally, we note that in most cases of interest the bound given above can be reduced by a factor of one half. That is, (A.5.13) can be improved to

$$\overline{d} \leq \frac{1}{2} \left.\frac{dT(z)}{dz}\right|_{z=1} \quad \text{(A.6.29)}$$

General sufficient conditions for this factor of one half are presented in Appendix B.

VII. Special Cases and Examples

There are special cases where symmetry conditions may allow us to reduce the number of "super states" which are necessary for the evaluation of transfer functions.

A. Sequence Independence

Recall that the average distortion given the transmitted sequence $\underline{s}$ is bounded by (A.5.2). In some cases this bound is independent of the actual transmitted signal state sequence $\underline{s}$. For such cases, we may pick a convenient sequence $\underline{s}^\circ$ such as one whose elements are all identical, e.g.,

$$s_k^\circ = \Delta_1 \qquad \text{for all } k \tag{A.7.1}$$

assuming this is an allowed sequence. Then, for any sequence $\underline{s}$ we evaluate the bound on average distortion using $\underline{s}^\circ$ as the assumed sequence. Thus, under this assumption, (A.5.2) becomes

$$E\,\{d((\hat{s}_n,\hat{u}_n),(s_n,u_n))\,|\,\underline{s}\}$$

$$\leq \sum_{i\leq n}\sum_{j\geq n}\sum_{\mathcal{S}(i,j|\underline{s}^\circ[i,j])} d((\tilde{s}_n,\tilde{u}_n),(\Delta_1,u_1))\; P(\underline{s}^\circ[i,j]\rightarrow\underline{\tilde{s}}[i,j]) \tag{A.7.2}$$

where

$$u_k = u_1 \qquad \text{for all } k \tag{A.7.3}$$

is assumed to yield the sequence $\underline{s}^\circ$.

Next using the bound of (A.4.6), namely,

$$P(\underline{s}^\circ[i,j]\rightarrow\underline{\tilde{s}}[i,j]) \leq \prod_{k=i}^{j=1} D_\lambda((\tilde{s}_k,\tilde{u}_k),(\Delta_1,u_1)), \tag{A.7.4}$$

equality (A.5.3) with all probabilities equal to unity, and the shift invariance property, we have from (A.5.9) that

$$E\,\{d((\hat{s}_n,\hat{u}_n),(s_n,u_n))\,|\,\underline{s}\}$$

$$\leq \sum_{j=1}^{\infty} \sum_{S(0,j|\underline{s}^\circ[0,j])} \left[\sum_{\ell=0}^{j-1} d((\tilde{s}_\ell,\tilde{u}_\ell),(\Delta_1,u_1))\right] \prod_{k=0}^{j-1} D_\lambda((\tilde{s}_k,\tilde{u}_k),(\Delta_1,u_1))$$

$$= \frac{d}{dz} T_0(z)\Big|_{z=1} \tag{A.7.5}$$

where

$$T_0(z) = \sum_{j=1}^{\infty} \sum_{S(0,j|\underline{s}^\circ[0,j])} \prod_{k=0}^{j-1} z^{d((\tilde{s}_k,\tilde{u}_k),(\Delta_1,u_1))} D_\lambda((\tilde{s}_k,\tilde{u}_k),(\Delta_1,u_1)) \tag{A.7.6}$$

To evaluate the bound of (A.7.5) we need to find the transfer function $T_0(z)$. Here, we define state transitions from state a_i to state a_j as (see (A.6.15)

$$a_{ij} = \begin{cases} z^{d((\Delta_i,u),(\Delta_1,u_1))}\, D_\lambda((\Delta_i,u),(\Delta_1,u_1)); & \text{if } u\in\mathcal{U} \text{ exists such that } \Delta_j = g(\Delta_i,u) \\ 0; & \text{if not} \end{cases} \tag{A.7.7}$$

Then define the (M-1) x (M-1) matrix

$$\underline{A} = \begin{bmatrix} a_{2,2} & a_{3,2} & \cdot \;\cdot \;\cdot \;\cdot \;\cdot & a_{M,2} \\ a_{2,3} & a_{3,3} & \cdot \;\cdot \;\cdot \;\cdot \;\cdot & a_{M,3} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{2,M} & a_{3,M} & \cdot \;\cdot \;\cdot \;\cdot \;\cdot & a_{M,M} \end{bmatrix} \qquad (A.7.8)$$

and the M-1 dimensional vectors

$$\underline{b} = \begin{bmatrix} a_{1,2} \\ a_{1,3} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_{1,M} \end{bmatrix} \qquad (A.7.9)$$

and

$$\underline{c} = \begin{bmatrix} a_{2,1} \\ a_{3,1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_{M,1} \end{bmatrix} \qquad \text{(A.7.10)}$$

By analogy with (A.6.21) the transfer function $T_0(z)$ is then given by

$$T_0(z) = \underline{c}^T(\underline{I} - \underline{A})^{-1}b \qquad \text{(A.7.11)}$$

and its derivative becomes

$$\frac{dT_0(z)}{dz} = (\underline{c}')^T(\underline{I} - \underline{A})^{-1}\underline{b} + \underline{c}^T(\underline{I} - \underline{A})^{-1}\underline{b}'$$

$$+ \underline{c}^T(\underline{I} - \underline{A})^{-1}\underline{A}'(\underline{I} - \underline{A})^{-1}\underline{b} \qquad \text{(A.7.12)}$$

where the primes again denote differentiation with respect to z. The final bound has the form given by (A.5.13) with T(z) replaced by $T_0(z)$. Note that here the evaluation of the bound involves only the M states defined by the original signal model whereas in the most general case of the previous section we considered M^2 "super states."

The most common class of examples where the bound in (A.5.2) is independent of the actual signal sequence is that corresponding to convolutional codes transmitted over symmetric channels (Ref. 1). For example, consider the binary convolutional code shown in Figure A-10a. This is a rate $r = \frac{2}{3}$, constraint length K = 2 code with input alphabet

$$\mathcal{U} = \{(00),(01),(10),(11)\} \tag{A.7.13}$$

where the first bit of each pair enters the top unit delay and the second bit enters the bottom unit delay. The output alphabet is

$$\mathcal{X} = \{(000),(001),(010),(011),(100),(101),(110),(111)\} \tag{A.7.14}$$

and the state is given by $s_n = u_{n-1}$ for all n, so that

$$\mathcal{S} = \mathcal{U} = \{\Delta_1,\Delta_2,\Delta_3,\Delta_4\} \tag{A.7.15}$$

Next suppose we have a symmetric channel such as that created by a BPSK modulated signal with additive white Gaussian noise and soft decision decoding (Ref. 1). Here the channel has input alphabet $\mathcal{I} = \{0,1\}$, output alphabet $\mathcal{W} = (-\infty,\infty)$, and the channel conditional probability density function $p(w|i)$ for each $i\varepsilon\mathcal{I}$, $w\varepsilon\mathcal{W}$ given by

$$p(w|i = 0) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(w - \sqrt{\frac{2E_s}{N_0}}\right)^2\right\}$$

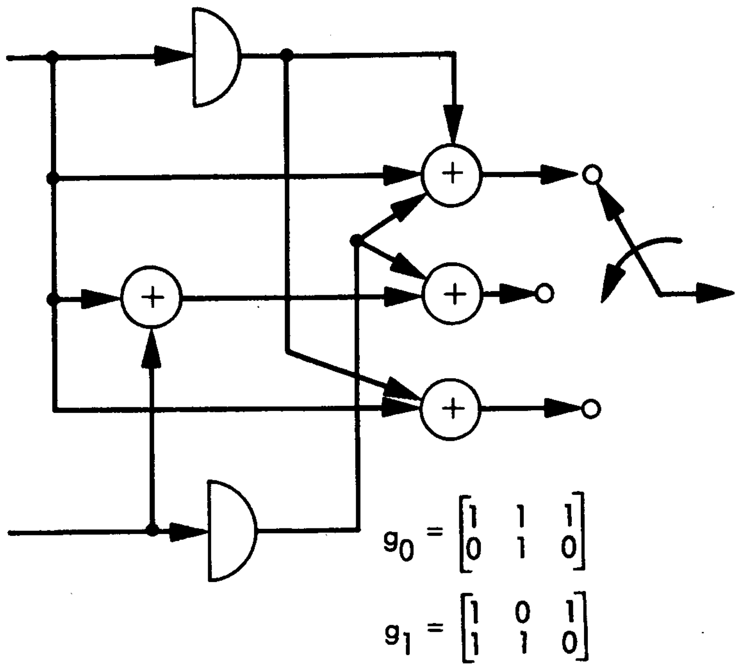
$$p(w|i = 1) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(w + \sqrt{\frac{2E_s}{N_0}}\right)^2\right\} \tag{A.7.16}$$

where E_s/N_0 is the BPSK pulse energy-to-noise ratio. In our example the convolutional code output consists of three binary symbols so that

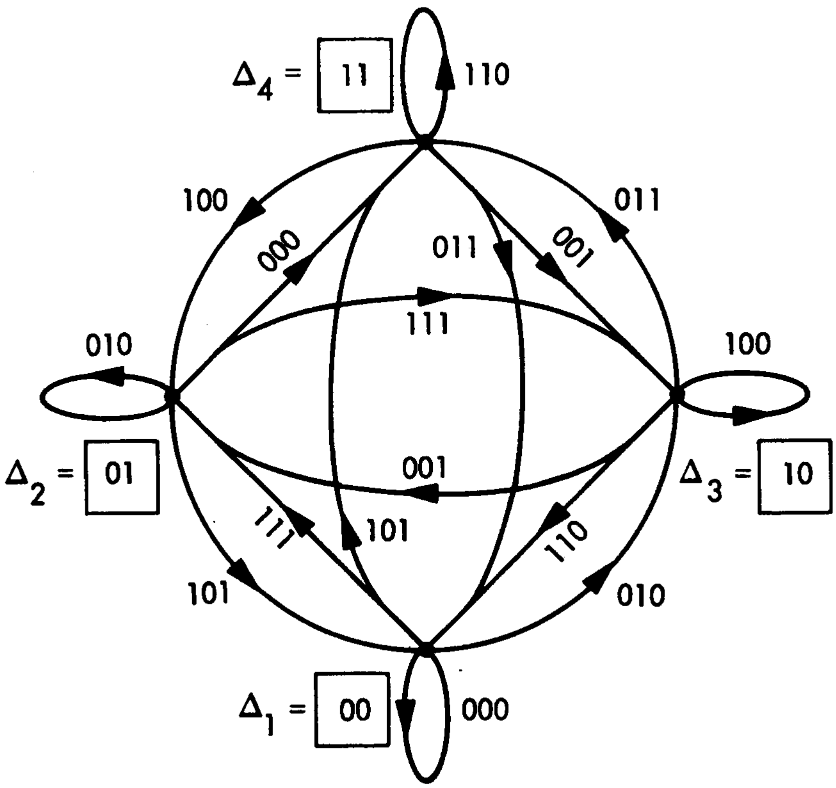
$$\mathcal{X} = \mathcal{I}^3 \tag{A.7.17}$$

and

$$\mathcal{Y} = \mathcal{W}^3 \tag{A.7.18}$$



a) Encoder



b) State Diagram - Outputs

Figure A-10. K = 2, r = $\frac{2}{3}$ Convolutional Code

Let us assume a maximum likelihood metric as in (A.2.5a) where now

$$p(y_k|x_k) = \prod_{n=1}^{3} p(w_{kn}|i_{kn}) \tag{A.7.19}$$

with $y_k \varepsilon \mathcal{Y}$ and $x_k \varepsilon X$. Letting $\bar{x}_k$ denote the binary component vector x_k with components coverted to ±1 by the rule

$$\begin{aligned} 0 &\rightarrow 1 \\ 1 &\rightarrow -1 \end{aligned} \tag{A.7.20}$$

then

$$p(y_k|\bar{x}_k) = \prod_{n=1}^{3} p(w_{kn}|\bar{i}_{kn}) \tag{A.7.21}$$

where $\bar{I}$ is the ±1 representation of i according to the rule in (A.7.20) and from (A.7.16)

$$p(w_{kn}|\bar{i}_{kn}) = \frac{1}{\sqrt{2\pi}} \exp\left\{- \frac{1}{2}\left(w_{kn} - i_{kn}\sqrt{\frac{2E_s}{N_0}}\right)^2\right\} \tag{A.7.22}$$

Substituting (A.7.22) into (A.7.21) and taking the natural logarithm of the result in accordance with (A.2.5a) gives

$$m((s_k,u_k),y_k) = \log_e (2\pi^{-3/2}) - \frac{1}{2} \sum_{n=1}^{3} w_{kn}{}^2 - \frac{3E_s}{N_0}$$

$$+ \sqrt{\frac{2E_s}{N_0}} \sum_{n=1}^{3} w_{kn}\bar{i}_{kn} \tag{A.7.23}$$

Since the first three terms of (A.7.23) are independent of x_k, we can equivalently consider the metric

$$m((s_k,u_k),y_k) = \sum_{n=1}^{3} w_{kn}\bar{i}_{kn} \triangleq (y_k,\bar{x}_k) \qquad \text{(A.7.24)}$$

where $(\cdot,\cdot)$ denotes the usual inner product of real vectors of dimension three.

In this case, the Bhattacharyya bound of (A.4.8) becomes*

$$D_{\frac{1}{2}}((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) = \prod_{n=1}^{3} \int \sqrt{p(w_{kn}|\bar{i}_{kn})\, p(w_{kn}|\tilde{\bar{i}}_{kn})}\; dw_{kn} \qquad \text{(A.7.25)}$$

Substituting (A.7.22) into (A.7.24), we get

$$\begin{aligned} D_{\frac{1}{2}}((s_k,u_k),(s_k,u_k)) &= \prod_{n=1}^{3} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left[w_{kn}^{\ 2} - 2\sqrt{\frac{2E_s}{N_0}}\, w_{kn}\left(\frac{\bar{i}_{kn}+\tilde{\bar{i}}_{kn}}{2}\right) + \frac{2E_s}{N_0}\right]\right\} dw_{kn} \\ &= \prod_{n=1}^{3} \exp\left(-\frac{E_s}{N_0}\right) \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\left[w_{kn}^{\ 2} - 2\sqrt{\frac{2E_s}{N_0}}\, w_{kn}\left(\frac{\bar{i}_{kn}+\tilde{\bar{i}}_{kn}}{2}\right)\right]\right\} dw_{kn} \\ &= \prod_{n=1}^{3} \exp\left\{-\frac{E_s}{N_0}\left[1-\left(\frac{\bar{i}_{kn}+\tilde{\bar{i}}_{kn}}{2}\right)^2\right]\right\} \\ &= \exp\left\{-\frac{E_s}{N_0}\sum_{n=1}^{3}\left[1-\left(\frac{\bar{i}_{kn}+\tilde{\bar{i}}_{kn}}{2}\right)^2\right]\right\} \\ &= \exp\left\{-\frac{E_s}{N_0}\sum_{n=1}^{3}\left[1-\frac{\bar{i}_{kn}\tilde{\bar{i}}_{kn}}{2}\right]\right\} \end{aligned}$$

(A.7.26)

---

*Note, the components of y_k are now continuous random variables and thus the sum over y_k in (A.4.8) is replaced by integrations over each component.

Letting $d_H(x_k,\bar{x}_k)$ denote the Hamming distance between x_k and $\bar{x}_k$ or equivalently the number of components of x_k and $\bar{x}_k$ which disagree, then

$$d_H(x_k,\tilde{x}_k) = \sum_{n=1}^{3}\left[1 - \frac{\bar{i}_{kn}\tilde{\bar{i}}_{kn}}{2}\right] \tag{A.7.27}$$

Substuting (A.7.26) into (A.7.25) gives the desired result, namely,

$$\boxed{D_{\frac{1}{2}}\left((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)\right) = \exp\left\{-\frac{E_s}{N_0}\,d_H(x_k,\tilde{x}_k)\right\}} \tag{A.7.28}$$

where, furthermore, $E_s = 2E_b/3$ with E_b the energy per data bit. Alternately, letting

$$D = \exp\left(-\frac{E_s}{N_0}\right) = \exp\left[-\frac{2}{3}\left(\frac{E_b}{N_0}\right)\right] \tag{A.7.29}$$

we can rewrite (A.7.28) as

$$D_{\frac{1}{2}}\left((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)\right) = D^{d_H(x_k,\tilde{x}_k)} \tag{A.7.30}$$

For a coded system we are typically interested in the average bit error probability. Suppose we consider the distortion measure $d((\tilde{s}_k,\tilde{u}_k),\ (s_k,u_k))$ that depends only on $\tilde{u}_k$ and u_k according to the following table:

Table A-1

$d((\tilde{s}_k,\tilde{u}_k),(s_k,u_k))$		$\tilde{u}_k$ 00	01	10	11
u_k	00	0	β	α	$\alpha+\beta$
	01	β	0	$\alpha+\beta$	α
	10	α	$\alpha+\beta$	0	β
	11	$\alpha+\beta$	α	β	0

for any $\alpha \geq 0$, $\beta \geq 0$. By choosing $\alpha = 1$, $\beta = 0$, the entries in the above table would be one whenever the first bit in $\tilde{u}_k$ and u_k disagree, and zero whenever they agree. Thus, the average distortion would give the average bit error probability for input bits entering the convolutional encoder at the upper unit delay in Fig. A-10a. Conversely, $\alpha = 0$, $\beta = 1$ results in table entries which are one whenever the second bit in $\tilde{u}_k$ and u_k disagree and zero whenever they agree. Thus, the average distortion would now give the average bit error probability of the input bits entering the lower unit delay of the encoder. Finally, $\alpha = \beta = \frac{1}{2}$ yields the total average bit error probability.

Figure A-11 illustrates the modified state diagram with the initial and final states given by state Δ_1 and intermediate states Δ_2, Δ_3, and Δ_4. The branch labels between states are determined by substituting (A.7.30) and the entries of Table A-1 into (A.7.7). By observation of Fig. A-11, we can directly obtain the transition matrix among nonzero states which is given by

$$\underline{A} = \begin{bmatrix} Dz^{\beta} & D^3z^{\beta} & Dz^{\beta} \\ Dz^{\alpha} & Dz^{\alpha} & Dz^{\alpha} \\ D^2z^{\alpha+\beta} & z^{\alpha+\beta} & D^2z^{\alpha+\beta} \end{bmatrix} \qquad (A.7.31)$$

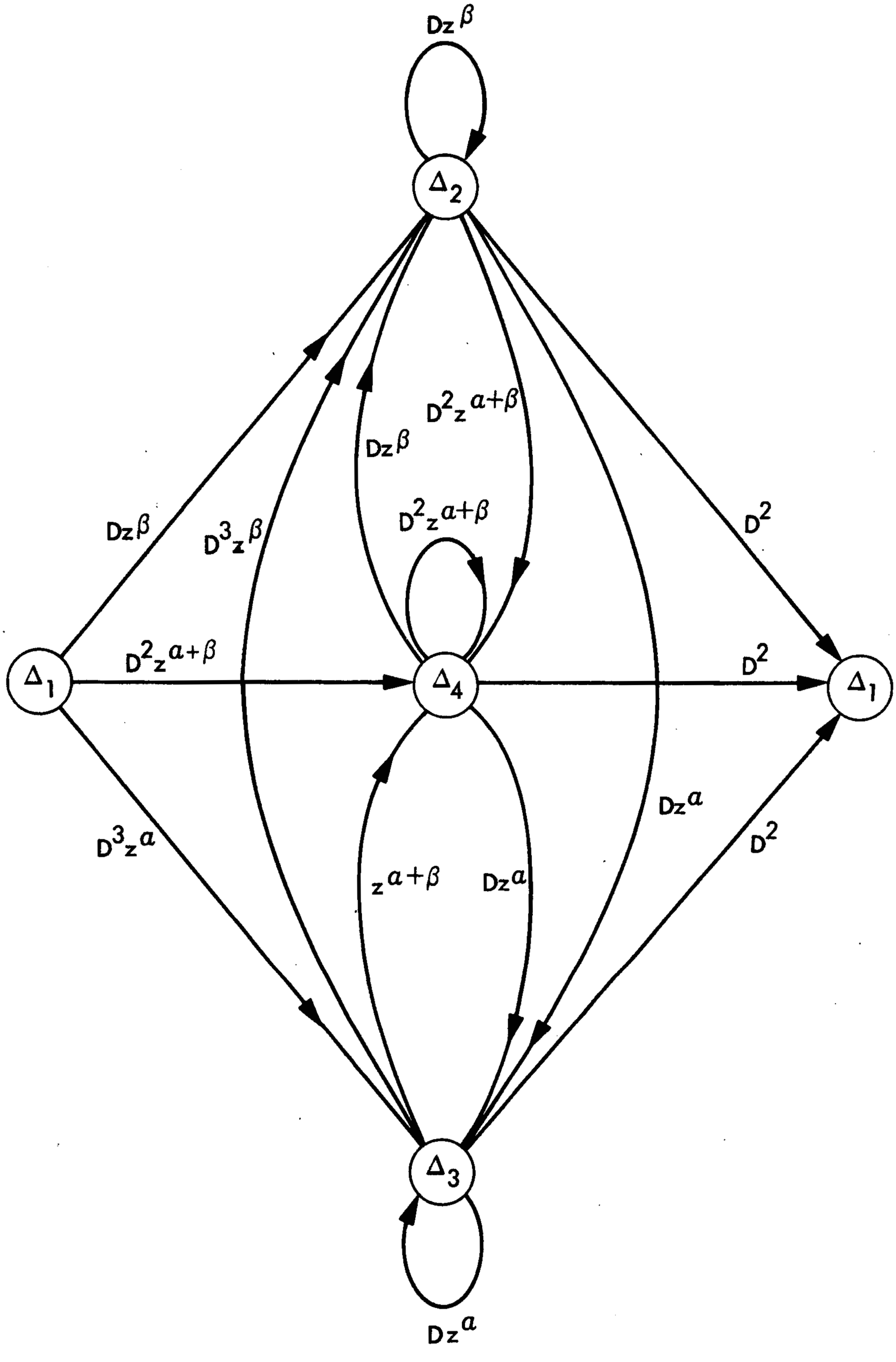


Figure A-11. Transfer Function State Diagram for K = 3, r = $\frac{2}{3}$ Code and vectors

$$\underline{b} = \begin{bmatrix} Dz^{\beta} \\ D^3 z^{\alpha} \\ D^2 z^{\alpha+\beta} \end{bmatrix}, \quad \underline{c} = \begin{bmatrix} D^2 \\ D^2 \\ D^2 \end{bmatrix} \tag{A.7.32}$$

Here the average distortion is given by*

$$\bar{d} \le \frac{1}{2} \left. \frac{dT_0(z)}{dz} \right|_{z=1}$$

$$= \frac{1}{2} \left[\underline{c}^T(\underline{I} - \underline{A})^{-1}\underline{b}' + \underline{c}^T(\underline{I} - \underline{A})^{-1}\underline{A}'(\underline{I} - \underline{A})^{-1}\underline{b} \Big|_{z=1} \right] \tag{A.7.33}$$

where for z = 1

$$\underline{b}' = \begin{bmatrix} \beta D \\ \alpha D^3 \\ (\alpha+\beta)D^2 \end{bmatrix} \tag{A.7.34}$$

and

$$\underline{A}' = \begin{bmatrix} \beta D & \beta D^3 & \beta D \\ \alpha D & \alpha D & \alpha D \\ (\alpha+\beta)D^2 & \alpha+\beta & (\alpha+\beta)D^2 \end{bmatrix} \tag{A.7.35}$$

*The factor of 1/2 is used here as discussed in Appendix B. Also note that for this case $\underline{c}' = \underline{0}$ which eliminates the first term of (A.7.12).

In Figure A-12 we show this bound on $\bar{d}$, for the two cases $\alpha = 1$, $\beta = 0$ and $\alpha = 0$, $\beta = 1$ corresponding, respectively, to the bit error probabilities of the two data bit sequences entering the upper and lower unit delays of the convolutional encoder. Note that for $E_b/N_0 = 7$ there is a factor of 10 difference in the bit error probabilities of the two data bit sequences entering the encoder.

B. Difference Sequences

In some examples the conditional average distortion bound given in (A.5.2) may depend only on differences, e.g. $\tilde{s}_n - s_n$ and $\tilde{u}_n - u_n$ for all n. This allows us to define "difference states" rather than general "super states" in evaluating the transfer function bounds on the average distortion. Typically the number of difference states is much smaller than the number of "super states."

Uncoded amplitude modulated signals transmitted over a linear channel with intersymbol interference and additive white Gaussian noise is a common example where only differences are important. For example, with uncoded BPSK modulation, we typically have the equal probable data bits* $u_n \in \mathcal{U} = \{-1, 1\}$ which after intersymbol interference results in an equivalent discrete-time signal

$$x_k = \sum_{i=0}^{\nu} h_i u_{k-i} \tag{A.7.36}$$

where ν is the assumed finite memory of the intersymbol interference and $h_0, h_1, \ldots, h_\nu$ are expressed in terms of the BPSK pulse rate and channel filter causing the intersymbol interference. The state is defined as the vector

*Here it is convenient to use {-1, 1} rather than {0, 1}. For simplicity, however, we shall omit the overbar notation on x_k.

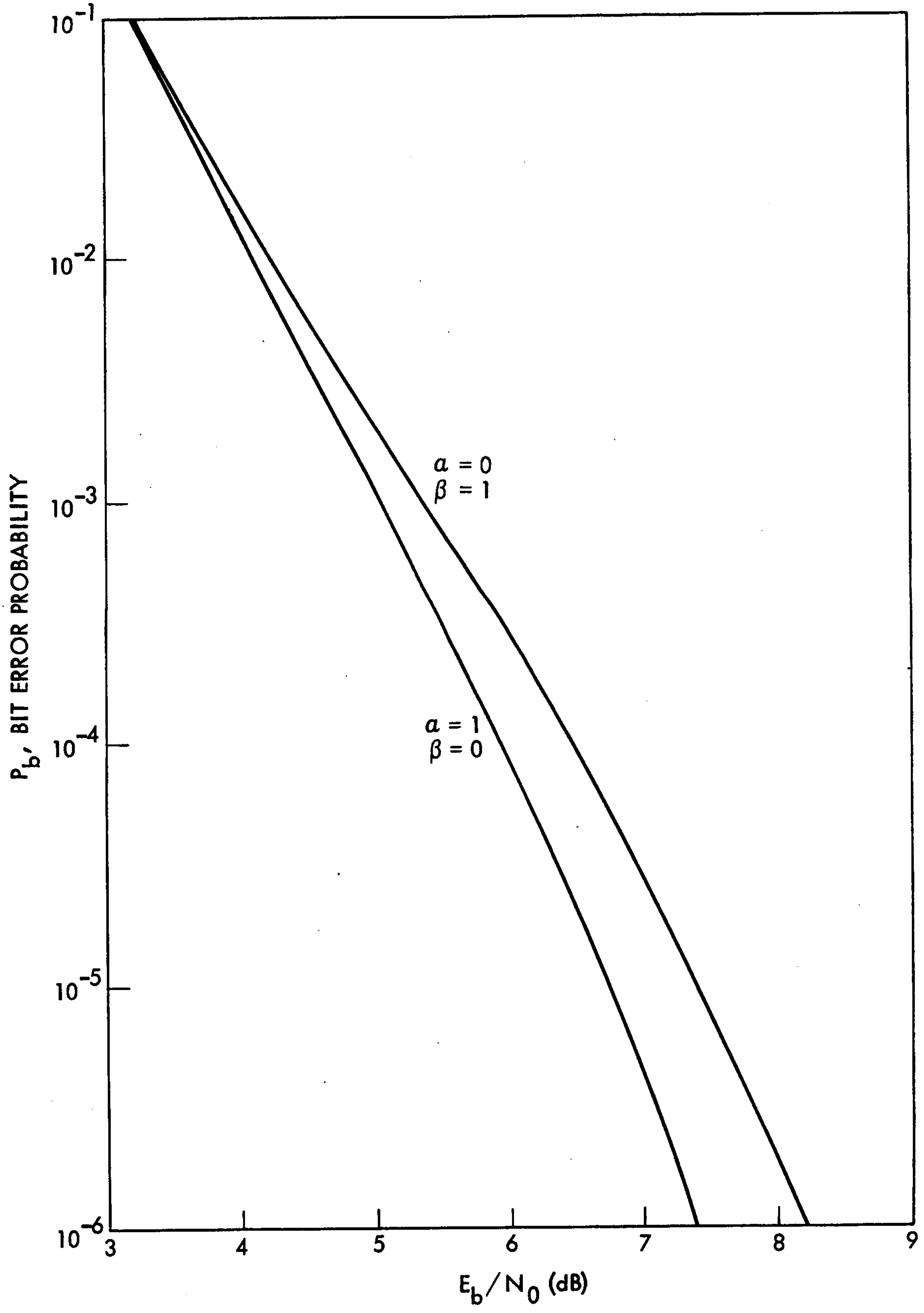


Figure A-12. Coded Bit Error Probabilities for K = 3, $r = \frac{2}{3}$ Code

$$s_k = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u_{k-\nu} \end{bmatrix} \tag{A.7.37}$$

and the filter vector is given by

$$h = \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ h_\nu \end{bmatrix} \tag{A.7.38}$$

Then, the signal has the form

$$x_k = (h, s_k) + h_0 u_k \tag{A.7.39}$$

where $(\cdot,\cdot)$ is again used to denote the inner product of vectors. The state s_{k+1} is obtained from shifting state s_k and adding component u_k, i.e., replacing k by k + 1 in (A.7.37).

The channel output is given by

$$y_k = x_k + n_k \tag{A.7.40}$$

where $\{n_k\}$ is an i.i.d. sequence of zero mean Gaussian random variables which are normalized to have unit variance. We use the natural maximum likelihood metric of (A.7.24) which results in the Chernoff bound becoming the Bhattacharyya bound

$$D_{\frac{1}{2}}((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) = \int \sqrt{p(y_k|x_k)p(y_k|\tilde{x}_k)}\ dy_k$$

$$= \int_{-\infty}^{\infty} \sqrt{\frac{1}{2\pi} \exp\left[-\frac{1}{2}(y_k - x_k)^2\right] \exp\left[-\frac{1}{2}(y_k - \tilde{x}_k)^2\right]}\ dy_k$$

$$= \exp\left[-\frac{1}{8}(x_k - \tilde{x}_k)^2\right]$$

$$= \exp\left\{-\frac{1}{8}[(h,s_k - \tilde{s}_k) + h_0(u_k - \tilde{u}_k)]^2\right\} \tag{A.7.41}$$

We are typically interested in bit error probabilities so we use the error distortion measure of (A.2.6a) which can be rewritten in the form

$$d((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)) = \frac{1}{4}(u_k - \tilde{u}_k)^2 = \begin{cases} 0; & u_k = \tilde{u}_k \\ 1; & u_k \neq \tilde{u}_k \end{cases} \tag{A.7.42}$$

Note that here both the metric and the Chernoff bound depend only on the differences $u_k - \tilde{u}_k$ and $s_k - \tilde{s}_k$. We now examine the transfer function T(z) given by (A.6.14), which upon substitution of (A.7.41) and (A.7.42) becomes

$$T(z) = \sum_{j=1}^{\infty} \sum_{S(0,j)} p(s_0) \prod_{k=0}^{j=1} z^{\frac{1}{4}(u_k-\tilde{u}_k)^2} q(u_k) \exp\left\{-\frac{1}{8}[(h, s_k - \tilde{s}_k) + h_0(u_k - \tilde{u}_k)]^2\right\} \tag{A.7.43}$$

To evaluate this, we now take advantage of the fact that only differences occur by defining

$$\varepsilon_k = \frac{1}{2}(u_k - \tilde{u}_k) \tag{A.7.44}$$

which takes on values {-1, 0, 1} and the difference state

$$\delta_k = \frac{1}{2}(s_k - \tilde{s}_k) \tag{A.7.45}$$

Then, the difference state δ_{k+1} would be obtained by shifting δ_k and adding component ε_k. Here there are 2^ν possible values of the state s_k while there are 3^ν possible values of the difference state δ_k. Recall that the "super states," consisting of pairs $(s_k, \tilde{s}_k)$, would have 4^ν possible values.

With the difference formulation and the fact that equally probable bits means

$$q(u_k) = \frac{1}{2} \tag{A.7.46}$$

we have from (A.7.43) that

$$T(z) = \sum_{j=1}^{\infty} \sum_{S(0,j)} p(s_0) \prod_{k=0}^{j=1} z^{\varepsilon_k^2}\left(\frac{1}{2}\right) \exp\left\{-\frac{1}{2}[(h, \delta_k) + h_0\varepsilon_k]^2\right\} \tag{A.7.47}$$

Recall that the sum over $S(0,j)$ consists of all pairs s[0,j] and $\tilde{s}[0,j]$ such that

$$\tilde{s}_0 = s_0,\ \tilde{s}_j = s_j \tag{A.7.48}$$

and

$$\tilde{s}_k \neq s_k;\ k = 1,2,\ \ldots.,\ j-1 \tag{A.7.49}$$

or, equivalently, a difference sequence

$$\delta[0,j] = (\delta_0,\delta_1,\ \ldots.,\ \delta_j) \tag{A.7.50}$$

where

$$\delta_0 = \delta_j = \underline{0} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{A.7.51}$$

and

$$\delta_k \neq \underline{0};\ k = 1,2,\ \ldots.,\ j-1 \tag{A.7.52}$$

Note that there are 2^{ν} choices of initial conditions $\tilde{s}_0 = s_0$ and thus $p(s_0) = 1/2^{\nu}$ whereas there is only one choice of initial condition for δ_0. Also note that the error sequence $\{\varepsilon_k\}$ does not uniquely specify the pair sequence $(u_k, \tilde{u}_k)$ since

$$\varepsilon_k = 1 \qquad \text{when } u_k = 1,\ \tilde{u}_k = -1$$

$$\varepsilon_k = -1 \qquad \text{when } u_k = -1,\ \tilde{u}_k = 1$$

$$\varepsilon_k = 0 \qquad \text{when} \begin{cases} u_k = -1,\ \tilde{u}_k = -1 \\ \qquad \text{or} \\ u_k = 1,\ \tilde{u}_k = 1 \end{cases} \tag{A.7.53}$$

Thus, if we replace the sum over $\mathcal{S}(0,j)$ by the sum over all difference state sequences

$$\mathcal{D}(0,j) = \left\{ \delta[0,j]:\ \delta_0 = \underline{0}, \delta_j = \underline{0}, \delta_k \neq \underline{0}; k = 1,2, \ldots, j-1 \right\} \tag{A.7.54}$$

then we must also replace $p(s_0) = 1/2^{\nu}$ by one and $q(u_k) = 1/2$ by

$$c(\varepsilon_k) = \begin{cases} \frac{1}{2};\ \varepsilon_k = 1 \\ \frac{1}{2};\ \varepsilon_k = -1 \\ 1;\ \varepsilon_k = 0 \end{cases} \tag{A.7.55}$$

or

$$c(\varepsilon_k) = \left(\frac{1}{2}\right)^{\varepsilon_k^2} \tag{A.7.56}$$

Note that (A.7.55) takes into account the fact that u_k can be +1 or -1 when $\varepsilon_k = 0$.

Thus the transfer function of (A.7.47) takes on the new form

$$T(z) = \sum_{j=1}^{\infty} \sum_{\mathcal{D}(0,j)} \prod_{k=0}^{j-1} \left(\frac{z}{2}\right)^{\varepsilon_k^2} \exp\left\{-\frac{1}{2}[(h,\delta_k) + h_0\varepsilon_k]^2\right\} \tag{A.7.57}$$

To evaluate the transfer function T(z), let the set of difference states be

$$\mathcal{D} = \{d_0, d_1, d_2, \ldots, d_{L-1}\} \tag{A.7.58}$$

where $L = 3^{\nu}$. Next define (see (A.6.15))

$$a_{ij} = \begin{cases} \left(\frac{z}{2}\right)^{\varepsilon^2} \exp\left\{-\frac{1}{2}[(h,d_i) + h_0\varepsilon]^2\right\}; & \text{if } d_j \text{ can be reached from } d_i \text{ with some G} \\ 0; \text{ if not} & \end{cases} \tag{A.7.59}$$

and

$$\underline{A} = \begin{bmatrix} a_{11} & a_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{L-1,1} \\ a_{12} & a_{22} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{L-1,2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1,L-1} & a_{2,L-1} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{L-1,L-1} \end{bmatrix}$$

$$\underline{b} = \begin{bmatrix} a_{01} \\ a_{02} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_{0,L-1} \end{bmatrix} ; \quad \underline{c} = \begin{bmatrix} a_{10} \\ a_{20} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ a_{L-1,0} \end{bmatrix} \qquad \text{(A.7.60)}$$

Then (see (A.7.11))

$$T(z) = \underline{c}^T(\underline{I} - \underline{A})^{-1}\underline{b} \qquad \text{(A.7.61)}$$

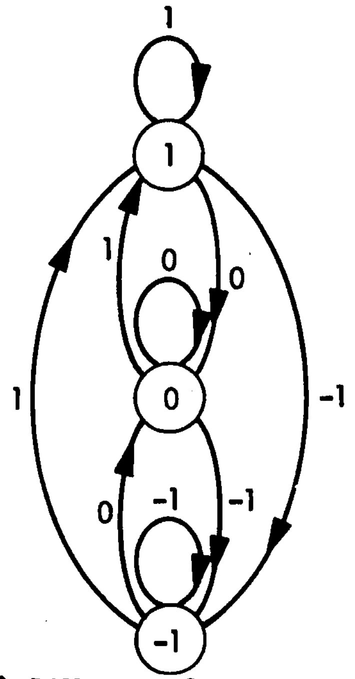
Consider the example where $\nu = 1$ so that we only have h_0 and h_1 and the difference states, $\delta_k = \varepsilon_{k-1}$, are

$$d_0 = 0$$

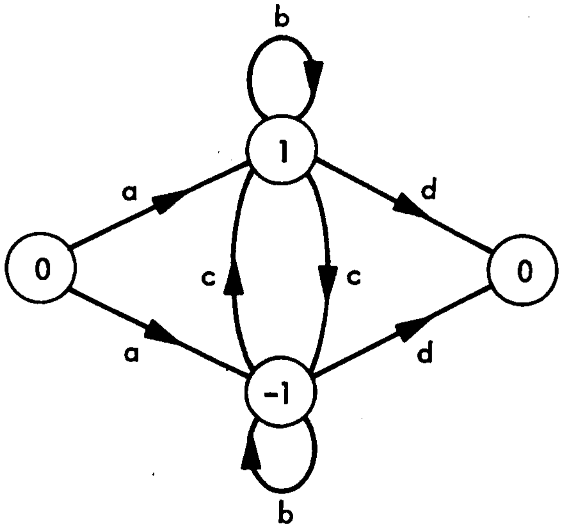
$$d_1 = 1$$

$$d_2 = -1 \qquad \text{(A.7.62)}$$

Figure A-13a shows the difference state diagram with ε as branch values while Figure A-13b shows the transfer function difference state diagram with a_{ij} as branch values. Thus



a) Difference State Diagram



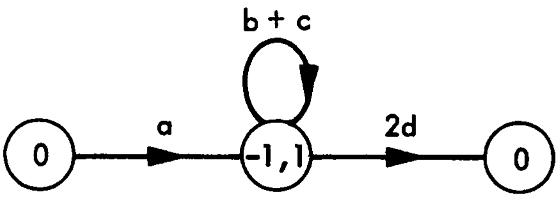
$$a = \frac{z}{2} e^{-\frac{1}{2} h_0^2}$$

$$b = \frac{z}{2} e^{-\frac{1}{2} (h_0 + h_1)^2}$$

$$c = \frac{z}{2} e^{-\frac{1}{2} (h_0 - h_1)^2}$$

$$d = e^{-\frac{1}{2} h_1^2}$$

b) Transfer Function Difference State Diagram



c) Reduced Transfer Function Difference State Diagram

Figure A-13. Example with $\nu = 1$

$$\underline{A} = \begin{bmatrix} b & c \\ c & b \end{bmatrix}$$

$$\underline{b} = \begin{bmatrix} a \\ a \end{bmatrix}; \ \underline{c} = \begin{bmatrix} d \\ d \end{bmatrix} \tag{A.7.63}$$

Substituting (A.7.63) into (A.7.61), we then have

$$T(z) = \frac{2ad}{1 - (b + c)} = \frac{z \exp\left[-\frac{1}{2}\left(h_0^2 + h_1^2\right)\right]}{1 - z \exp\left[-\frac{1}{2}\left(h_0^2 + h_1^2\right)\right] \cosh\ (h_0 h_1)} \tag{A.7.64}$$

and the bit error probability bound

$$\boxed{\begin{aligned} P_b &\leq \frac{1}{2} \left.\frac{dT(z)}{dz}\right|_{z=1} \\ &= \frac{1}{2} \frac{\exp\left[-\frac{1}{2}\left(h_0^2 + h_1^2\right)\right]}{\left[1 - \exp\left[-\frac{1}{2}\left(h_0^2 + h_1^2\right)\right] \cosh\ (h_0 h_1)\right]^2} \end{aligned}} \tag{A.7.65}$$

We can compare this result with the no intersymbol interference case. This corresponds to conventional BPSK with bit error probability*

$$P_b^* = Q\left(\sqrt{h_0^2 + h_1^2}\right) \leq \frac{1}{2} \exp\left[-\frac{1}{2}\left(h_0^2 + h_1^2\right)\right] \tag{A.7.66}$$

$*Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\ (-y^2/2)dy$ is the usual error probability integral. Also we have normalized both cases to have the same energy.

where this bound on $Q(x)$ is within 0.5 dB for $P_b^* \le 10^{-2}$. For the special case

$$h_0 = \sqrt{\frac{2E_b}{N_0}}$$

$$h_1 = \frac{h_0}{\sqrt{2}} \tag{A.7.67}$$

Figure A-14 illustrates the bound on P_b given by (A.7.65) and the bound on P_b^* given by (A.7.66). For large values of E_b/N_0 the difference is asymptotically equal to zero.

Another possible comparison is with a conventional single sample data detector which makes no use of the energy in the intersymbol interference to improve performance. Here, the average bit error probability is simply given by

$$\begin{aligned} P_b^{**} &= \tfrac{1}{2}Q(h_0 + h_1) + \tfrac{1}{2}Q(h_0 - h_1) \\ &\le \frac{1}{4} \exp\left[-\frac{1}{2}(h_0 + h_1)^2\right] + \frac{1}{4} \exp\left[-\frac{1}{2}(h_0 - h_1)^2\right] \\ &= \frac{1}{2} \exp\left[-\frac{1}{2}\left(h_0^2 + h_1^2\right)\right] \cosh\,(h_0 h_1) \end{aligned} \tag{A.7.68}$$

This result is also illustrated in Figure A-14. Notice how the Viterbi algorithm has been successful in combating intersymbol interference.

Several other examples of intersymbol interference channels and their analysis are given in Ref. 1. There, continuous-time signals are reduced to equivalent discrete-time signals and the corresponding transfer function bounds as in (A.7.57) are derived. In these examples one can see further that for a difference state sequence as in (A.7.50) with corresponding error sequence $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{j-1}$ there is an equivalent difference state sequence

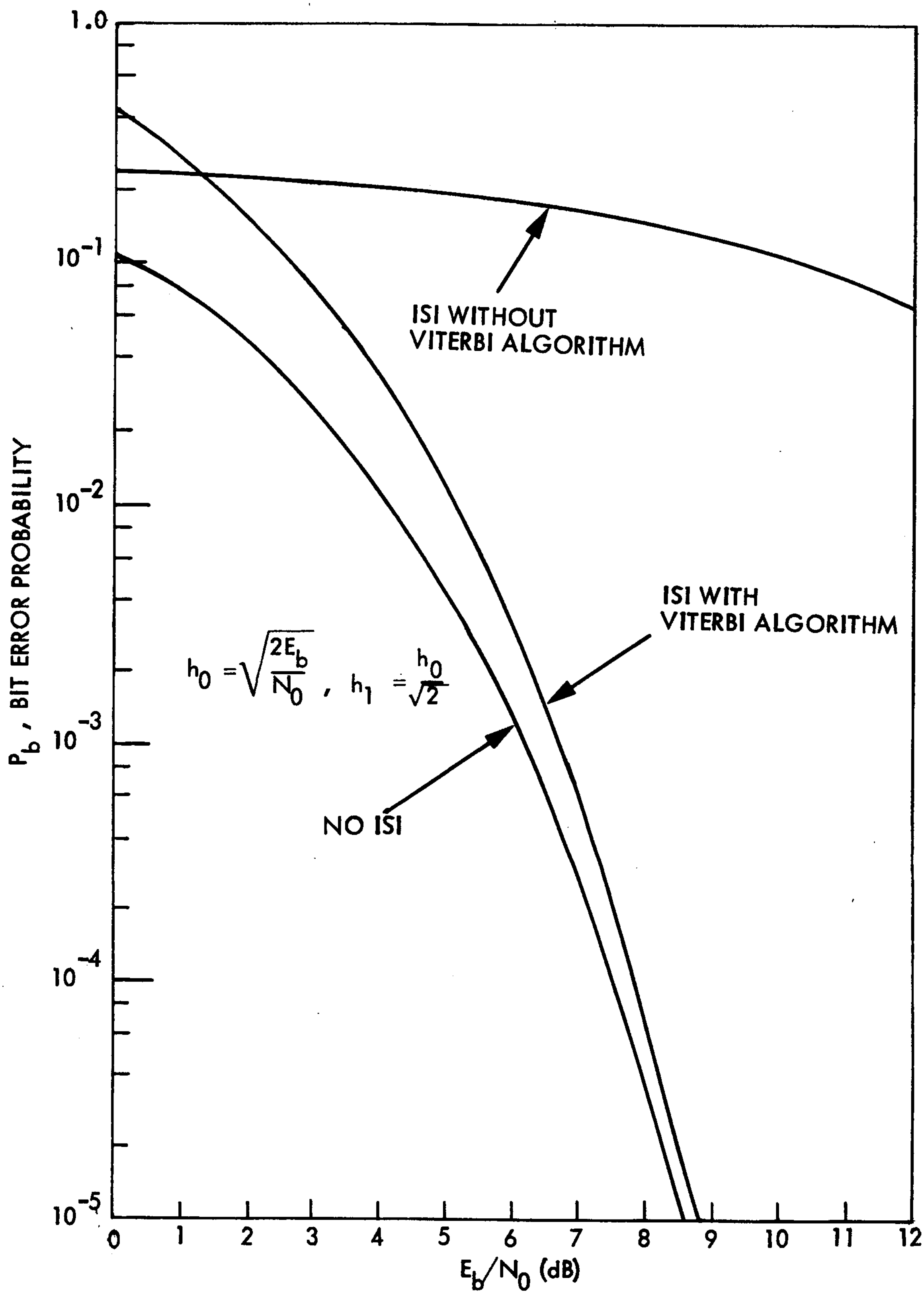


Figure A-14. Intersymbol Interference Example

$$-\delta[0,j] = (-\delta_0, -\delta_1, \ldots, -\delta_j) \tag{A.7.69}$$

with corresponding error sequence $-\varepsilon_0, -\varepsilon_1, \ldots, -\varepsilon_{j-1}$. Both of these have identical transfer function values

$$\prod_{k=0}^{j-1} \left(\frac{z}{2}\right)^{\varepsilon_k^2} \exp\left\{-\frac{1}{2}\left[(h,\delta_k) + h_0\varepsilon_k\right]^2\right\}$$

This means that all nonzero difference states can be merged with their opposite sign state resulting in a reduced state diagram consisting of $3^\nu/2$ nonzero states (see Figure A-13c for our example).

Next we shall consider an example where the number of states necessary to compute the transfer function bound is actually less than the number of signal states $|S|$.

C. Absolute Difference Sequences

We examine here another problem where "absolute difference states" are used in the transfer function bound. In particular, we consider a phase estimation problem where we quantize the phase space $(0, 2\pi)$ into M values

$$S = \{\Delta_1, \Delta_2, \ldots, \Delta_M\} \tag{A.7.70}$$

where

$$\Delta_k = k\Delta;\ k = 1,2, \ldots, M$$

$$\Delta = \frac{2\pi}{M} \tag{A.7.71}$$

The phase sequence is $s_0, s_1, s_2, \ldots$ which we model as

$$s_{k+1} = s_k \oplus u_k;\ k = 0,1,2, \ldots \tag{A.7.72}$$

where the initial phase random variable s_0 has the probability

$$p(s_0) = \frac{1}{M}; \text{ all } s_0 \varepsilon S \tag{A.7.73}$$

and $\{u_k\}$ are i.i.d. random variables with common probability.

$$q(u) = \begin{cases} \frac{1}{3}; & u = -\Delta, 0, \Delta \\ 0; & \text{otherwise} \end{cases} \tag{A.7.74}$$

Here the symbol $\oplus$ denotes modulo 2π addition. Thus, at any point in the sequence, the phase may either remain the same or take on one of its two adjacent values all with equal probability of occurrence.

The actual signal is assumed to be the sine and cosine of the phase, i.e.,

$$x_k = \begin{bmatrix} \cos\ (s_k \oplus u_k) \\ \sin\ (s_k \oplus u_k) \end{bmatrix} = \begin{bmatrix} \cos\ s_{k+1} \\ \sin\ s_{k+1} \end{bmatrix} \tag{A.7.75}$$

Figure A-15 shows the state diagram for this signal for $M = 8$ and $\Delta = \frac{\pi}{4}$. The branch values are the signal inputs $u \varepsilon \mathcal{U}$.

Suppose the channel adds zero mean independent Gaussian random variables to each component resulting in the channel output vector

$$y_k = a\ x_k + N_k \tag{A.7.76}$$

where

$$a = \sqrt{\frac{2E_s}{N_0}} \tag{A.7.77}$$

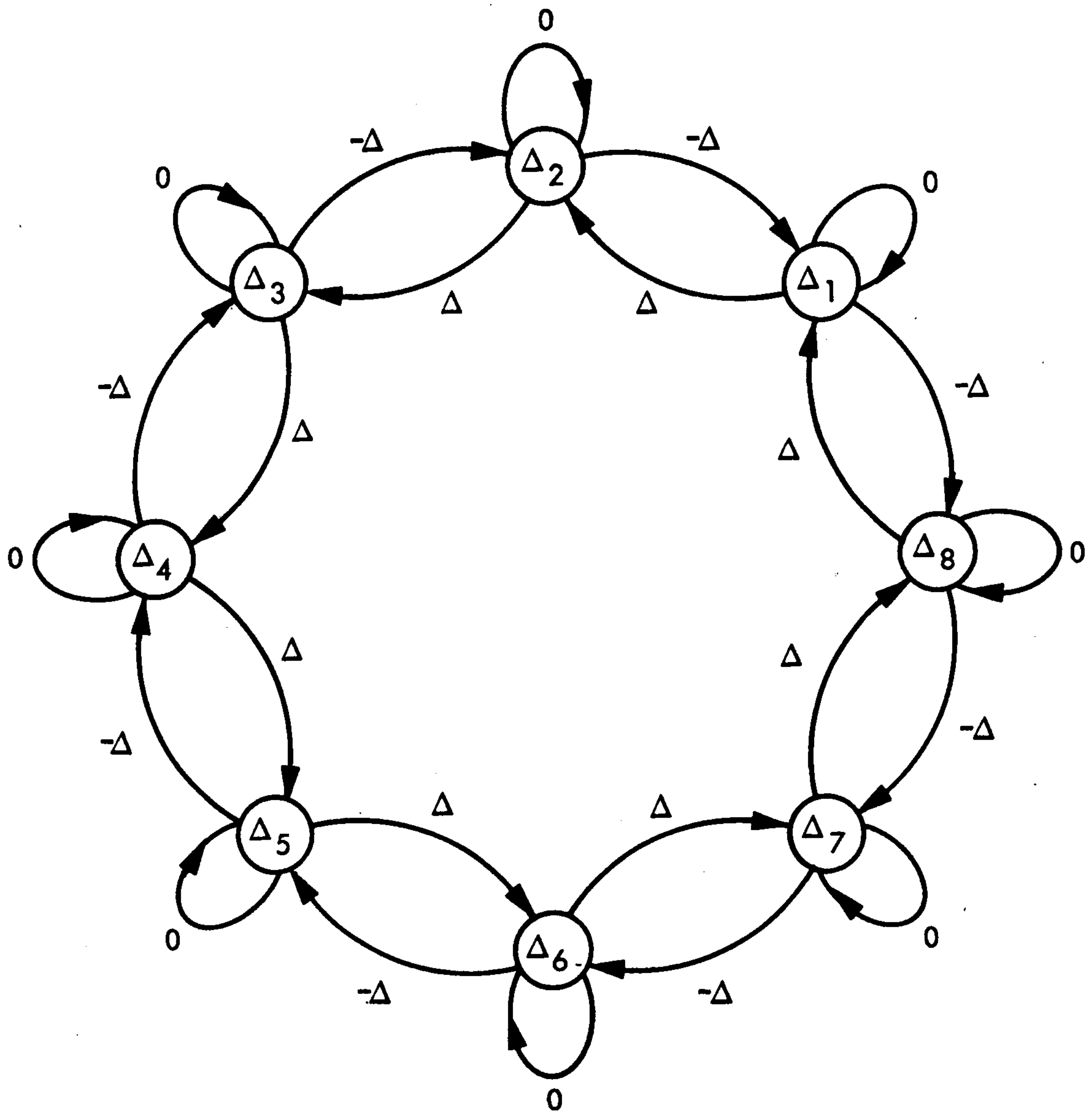


Figure A-15. Phase Model For M = 8

and

$$N_k = \begin{bmatrix} n_k \\ \\ \hat{n}_k \end{bmatrix} \tag{A.7.78}$$

with n_k and $\hat{n}_k$ having unit variance. Also assume that the receiver uses the maximum likelihood metric analogous to (A.7.24) namely

$$m\Big((\tilde{s}_k,\tilde{u}_k),y_k\Big) = (y_k,\tilde{x}_k) \tag{A.7.79}$$

and the squared error distortion measure,

$$d\Big((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)\Big) = \min \Big\{(s_{k+1} \ominus \tilde{s}_{k+1})^2, (\tilde{s}_{k+1} \ominus s_{k+1})^2\Big\} \tag{A.7.80}$$

where $\ominus$ denotes the difference modulo 2π. Note that this distortion depends only on the absolute difference between s_{k+1} and $\tilde{s}_{k+1}$ (modulo 2π) which has values in

$$\mathcal{D} = \{\Delta_0,\Delta_1, \ldots, \Delta_{M/2}\};\ \Delta_0 = 0,\ M \text{ even} \tag{A.7.81}$$

Furthermore let δ_n be defined as this absolute difference, namely,

$$\delta_{k+1} = \sqrt{d\Big((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)\Big)} \tag{A.7.82}$$

which as stated above has values only in $\mathcal{D}$.

Substituting (A.7.75) into (A.7.76), then the metric of (A.7.79) is evaluated as

$$(y_k,\tilde{x}_k) = a \cos (s_{k+1} - \tilde{s}_{k+1}) + \tilde{n}_k \tag{A.7.83}$$

where $\tilde{n}_k \triangleq (N_k, \tilde{x}_k) = n_k \cos \tilde{s}_{k+1} + \hat{n}_k \sin \tilde{s}_{k+1}$ is a zero mean unit variance Gaussian random variable. Note that

$$
\begin{aligned}
\cos (s_{k+1} - \tilde{s}_{k+1}) &= \cos (s_{k+1} \ominus \tilde{s}_{k+1}) \\
&= \cos (\tilde{s}_{k+1} \ominus s_{k+1}) \\
&= \cos \left[\sqrt{d\Big((\tilde{s}_k, \tilde{u}_k), (s_k, u_k)\Big)}\right] \\
&= \cos \delta_{k+1} \qquad \text{(A.7.84)}
\end{aligned}
$$

Hence, both the distortion measure and the metric depend only on the absolute differences $\{\delta_k\}$.

Since we use a maximum likelihood metric the Chernoff bound results in the Bhattacharyya bound

$$
\begin{aligned}
D_{\frac{1}{2}}\Big((\tilde{s}_k, \tilde{u}_k), (s_k, u_k)\Big) &= \int \sqrt{p(y|x_k) p(y|\tilde{x}_k)}\, dy \\
&= \exp \left\{- \frac{1}{8} || a x_k - a \tilde{x}_k ||^2 \right\} \\
&= \exp \left\{- \frac{a^2}{8} \left[(\cos s_{k+1} - \cos s_k)^2 + (\sin s_{k+1} - \sin s_k)^2\right]\right\} \\
&= \exp \left\{- \frac{a^2}{4} \left[1 - \cos (s_{k+1} - \tilde{s}_{k+1})\right]\right\} \\
&= \exp \left\{- \frac{a^2}{4} [1 - \cos \delta_{k+1}]\right\} \qquad \text{(A.7.85)}
\end{aligned}
$$

Thus in addition to the metric, the Chernoff bound also depends only on the absolute differences $\{\delta_k\}$. As in the previous example, define an error term for each k by*

$$\varepsilon_k = u_k - \tilde{u}_k \ \varepsilon \ \{-2\Delta,-\Delta,0,\Delta,2\Delta\} \tag{A.7.86}$$

Then, an absolute difference process can be given by

$$\delta_{k+1} = \begin{cases} |\delta_k + \varepsilon_k| ; & \delta_k \neq \pi - \Delta,\ \pi,\ \text{all } \varepsilon_k \\ |\delta_k + \varepsilon_k| ; & \delta_k = \pi - \Delta,\ \varepsilon_k \neq 2\Delta \\ \pi - \Delta \quad ; & \delta_k = \pi - \Delta,\ \varepsilon_k = 2\Delta \\ \pi - |\varepsilon_k| \ ; & \delta_k = \pi,\ \text{all } \varepsilon_k \end{cases} \tag{A.7.87}$$

We now consider a transfer function bound for this problem. The general transfer function bound T(z) given by (A.6.14) which when using (A.7.82) and (A.7.85), i.e.,

$$z^{d\left((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)\right)} D_\lambda\left((\tilde{s}_k,\tilde{u}_k),(s_k,u_k)\right) = z^{\delta^2_{k+1}} \exp\left\{-\frac{a^2}{4}[1 - \cos\delta_{k+1}]\right\} \tag{A.7.88}$$

has the form

$$T(z) = \sum_{j=1}^{\infty} \sum_{S(0,j)} p(s_0) \prod_{k=0}^{j-1} z^{\delta^2_{k+1}} q(u_k) \exp\left\{-\frac{a^2}{4}[1 - \cos\delta_{k+1}]\right\} \tag{A.7.89}$$

*Note (A.7.86) is analogous to (A.7.44) except for a factor of two.

Recall $S(0,j)$ is the set of sequences $\tilde{s}[0,j]$ and $s[0,j]$ that diverge at the initial node and remerge j branches later in the trellis diagram. This corresponds to a particular absolute difference state sequence as in (A.7.50) where now

$$\delta_0 = \delta_j = 0; \ \delta_k \neq 0, \ k = 1,2, \ \ldots, \ j-1 \tag{A.7.90}$$

We replace the sum over $S(0,j)$ by the sum over all absolute different state sequences as in (A.7.54). Next, we replace $p(s_0)$ by one and $q(u_k)$ by a function $c(\varepsilon_k)$ whose definition we shall now examine by considering the values of u_k associated with each value of ε_k as follows:

$$\varepsilon_k = 0 \quad \text{when} \quad \begin{cases} u_k = 0 \ , \ \tilde{u}_k = 0 \\ u_k = \Delta \ , \ \tilde{u}_k = \Delta \\ u_k = -\Delta, \ \tilde{u}_k = -\Delta \end{cases}$$

$$\varepsilon_k = \Delta \quad \text{when} \quad \begin{cases} u_k = 0 \ , \ \tilde{u}_k = -\Delta \\ u_k = \Delta \ , \ \tilde{u}_k = 0 \end{cases}$$

$$\varepsilon_k = -\Delta \quad \text{when} \quad \begin{cases} u_k = -\Delta, \ \tilde{u}_k = 0 \\ u_k = 0 \ , \ \tilde{u}_k = \Delta \end{cases}$$

$$\varepsilon_k = 2\Delta \quad \text{when} \quad u_k = \Delta \ , \ \tilde{u}_k = -\Delta$$

$$\varepsilon_k = -2\Delta \ \text{when} \quad u_k = -\Delta, \ \tilde{u}_k = \Delta \tag{A.7.91}$$

From this we have

$$c(\varepsilon_k) = \begin{cases} 1; \ \varepsilon_k = 0 \\ \frac{2}{3}; \ \varepsilon_k = \pm\Delta \\ \frac{1}{3}; \ \varepsilon_k = \pm 2\Delta \end{cases} \tag{A.7.92}$$

where we count the number of distinct values of u_k for each ε_k and multiply by $q(u_k) = \frac{1}{3}$ as in (A.7.74).

The final transfer function has the form

$$\boxed{T(z) = \sum_{j=1}^{\infty} \sum_{\mathcal{D}(0,j)} \prod_{k=0}^{j-1} z^{\delta_{k+1}^2} c(\varepsilon_k) \exp\left\{-\frac{a^2}{4}[1 - \cos \delta_{k+1}]\right\}} \tag{A.7.93}$$

Note that (A.7.93) can be evaluated using only M/2 states which is even less than the number of states required for the Viterbi algorithm.

Now we define the branch transfer functions

$$a_{ij} = \begin{cases} z^{\Delta_j^2} c_{ij} \exp\left\{-\frac{a^2}{4}[1 - \cos \Delta_j]\right\}; & \text{if state } \Delta_j \text{ can be reached from state } \Delta_i \\ 0; \text{ if not} & \end{cases} \tag{A.7.94}$$

where c_{ij} is the sum of the numbers $c(\varepsilon)$ corresponding to all error inputs ε that can cause* a transition from state Δ_i to state Δ_j. Then defining

*Note that in (A.7.87), we see that $\varepsilon = \Delta$ or $\varepsilon = -\Delta$ can cause a transition from $\delta_k = 0$ to $\delta_{k+1} = \Delta_1 = \Delta$.

$$
\underline{A} = \begin{bmatrix} a_{11} & a_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{M/2,1} \\ a_{12} & a_{22} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{M/2,2} \\ \vdots & \vdots & & & & & & \vdots \\ a_{1,M/2} & a_{2,M/2} & \cdot & \cdot & \cdot & \cdot & \cdot & a_{M/2,M/2} \end{bmatrix} \qquad (A.7.95)
$$

$$
\underline{b} = \begin{bmatrix} a_{01} \\ a_{02} \\ \vdots \\ a_{0,M/2} \end{bmatrix} ; \; \underline{c} = \begin{bmatrix} a_{10} \\ a_{20} \\ \vdots \\ a_{M/2,0} \end{bmatrix} \qquad (A.7.96)
$$

we may use the simple transfer function of (A.7.61).

For the M = 8 example of Figure A-15 we have

$$
\mathcal{D} = \{0, \Delta_1, \Delta_2, \Delta_3, \Delta_4\} \qquad (A.7.97)
$$

and the state diagram for the absolute difference process given by (A.7.87) is illustrated in Figure A-16. Here the branch values are the input errors ε_k of

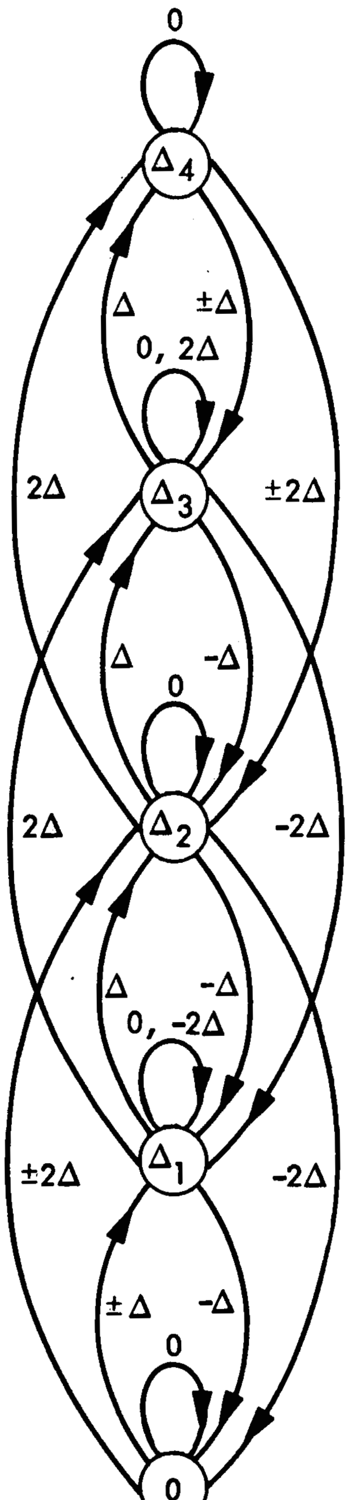


Figure A-16. Absolute Difference State Diagram

(A.7.86). The corresponding transfer function absolute difference state diagram is shown in Figure A-17. Here we have the branch transfer functions given by

$$a_{01} = \frac{4}{3}\alpha(\Delta_1), \quad a_{02} = \frac{2}{3}\alpha(\Delta_2)$$

$$a_{43} = \frac{4}{3}\alpha(\Delta_3), \quad a_{42} = \frac{2}{3}\alpha(\Delta_2)$$

$$a_{11} = \frac{4}{3}\alpha(\Delta_1), \quad a_{22} = \alpha(\Delta_2)$$

$$a_{33} = \frac{4}{3}\alpha(\Delta_3), \quad a_{44} = \alpha(\Delta_4)$$

$$a_{i,i+1} = \frac{2}{3}\alpha(\Delta_{i+1}); \; i = 1,2,3$$

$$a_{i,i-1} = \frac{2}{3}\alpha(\Delta_{i-1}); \; i = 1,2,3,4$$

$$a_{i,i+2} = \frac{1}{3}\alpha(\Delta_{i+2}); \; i = 1,2$$

$$a_{i,i-2} = \frac{1}{3}\alpha(\Delta_{i-2}); \; i = 2,3,4 \tag{A.7.98}$$

where

$$\alpha(\Delta) = z^{\Delta^2} \exp\left\{-\frac{a^2}{4}[1 - \cos\Delta]\right\} \tag{A.7.99}$$

In arriving at (A.7.98), we have made use of the fact that when two values of ε_k can cause the same transition between states, then the branch functions corresponding to the different values of ε_k can be added together to form a single branch function. Thus, for this example, we have from (A.7.44) and (A.7.96) that

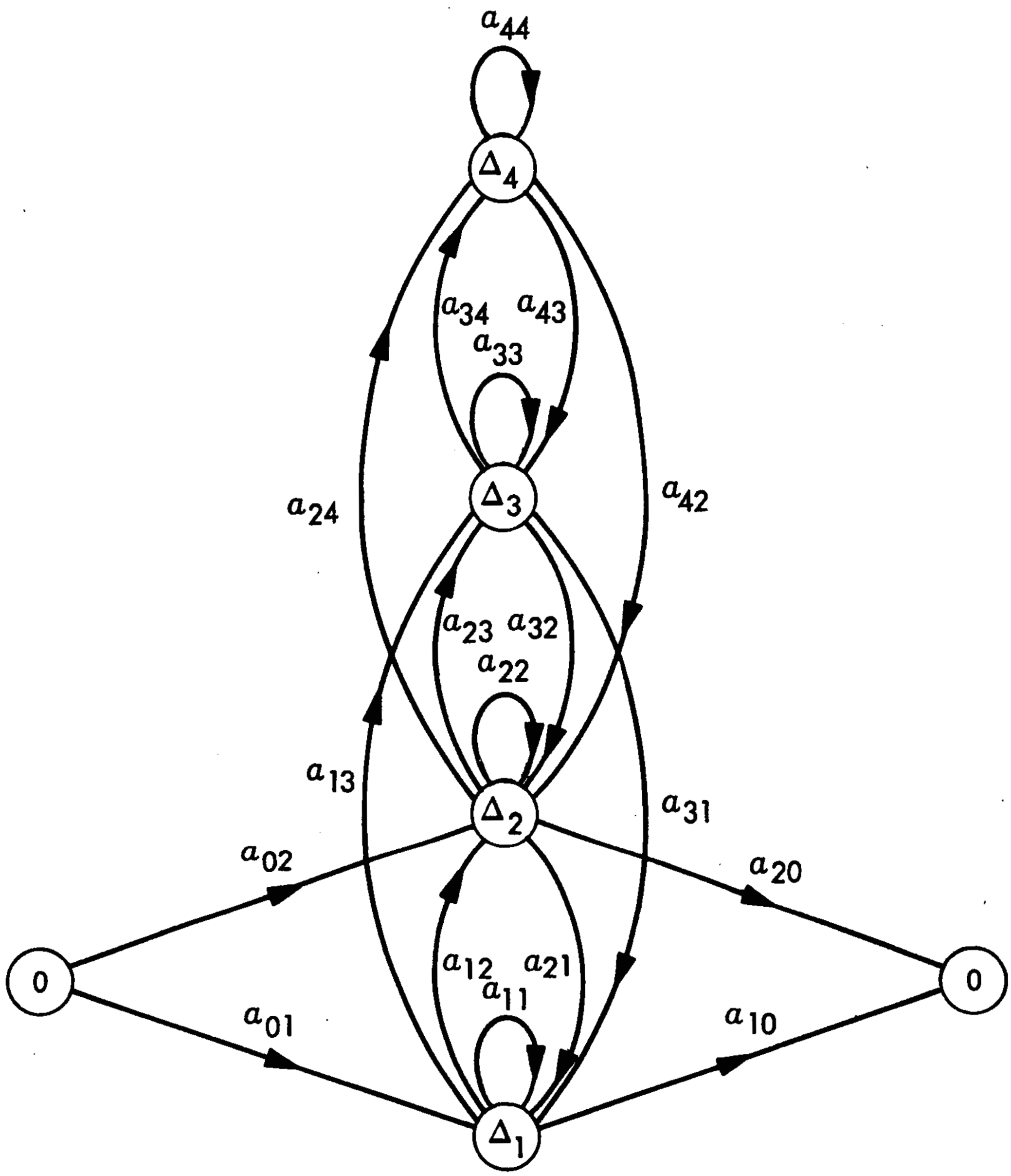


Figure A-17. Absolute Difference Transfer Function State Diagram

$$\underline{A} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & 0 \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ 0 & a_{24} & a_{34} & a_{44} \end{bmatrix} \tag{A.7.100}$$

and

$$\underline{b} = \begin{bmatrix} a_{01} \\ a_{02} \\ 0 \\ 0 \end{bmatrix} ; \underline{c} = \begin{bmatrix} a_{10} \\ a_{20} \\ 0 \\ 0 \end{bmatrix} \tag{A.7.101}$$

Figure A-18 illustrates the mean square phase error bound, as computed from (A.5.13) together with (A.7.61), (A.7.100) and (A.7.101) using the maximum likelihood Viterbi algorithm which in this application is basically a smoothing algorithm. This is shown for $M = 8$ as a function of the signal energy-to-noise ratio E_s/N_0. For large E_s/N_0 the remaining error is due to quantization of the 2π interval into M quantized values.

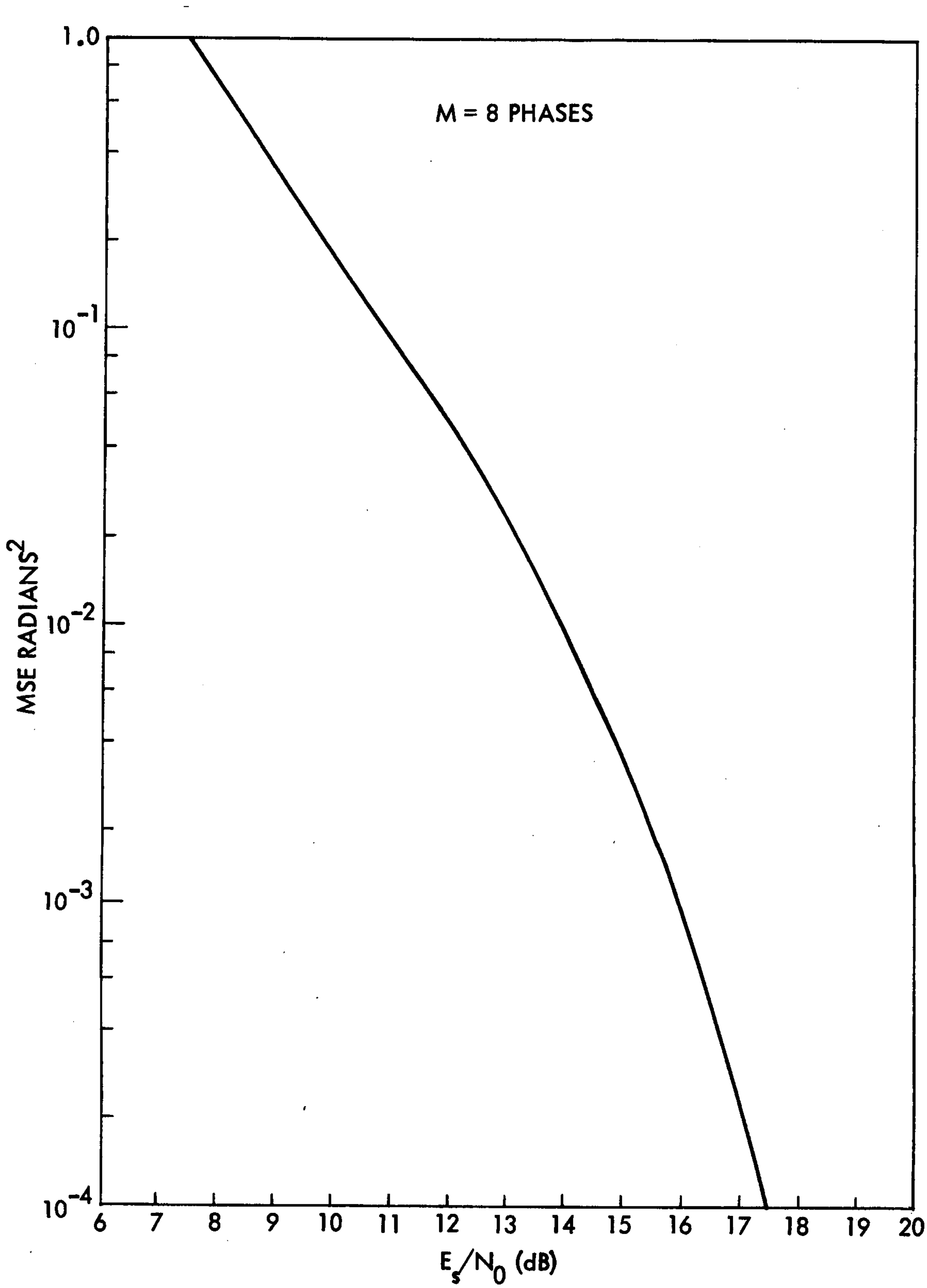


Figure A-18. Phase Estimation

APPENDIX B

A Factor of One Half in Error Probability Bounds

I. Introduction

In many complex communication systems, error probabilities are often difficult to evaluate, and thus, easily computed bounds are highly desirable. Two such bounds are the Chernoff bound and the Bhattacharyya bound (Ref. 1).

For any error probability bound, one desires that it be as tight as possible. Jacobs (Ref. 2) gave sufficient conditions for reducing the standard Chernoff bound by a factor of one half. In this appendix, we rederive this result and give less restrictive but harder to verify sufficient conditions. We also present some related results of Hellman and Raviv (Ref. 3) which show that all Bhattacharyya bounds can be reduced by a factor of one half.

II. Decision Function and Error Probability Models

Let Z be a continuous random variable that can have one of two probability densities:

$$H_1 : f_1(z), \quad -\infty < z < \infty$$

$$H_2 : f_2(z), \quad -\infty < z < \infty \tag{B.2.1}$$

where the a priori probabilities for these two hypotheses are denoted by

$$\pi_1 = \Pr\{H_1\} \text{ and } \pi_2 = \Pr\{H_2\} = 1 - \pi_1 \tag{B.2.2}$$

We assume an arbitrary deterministic decision rule characterized by the following binary-valued decision function: Given an observed value z of the random variable Z, then if

$$\phi(z) = 1, \text{ decide } H_1 \tag{B.2.3a}$$

and if

$$\phi(z) = 0, \text{ decide } H_2 \tag{B.2.3b}$$

In terms of this decision function, we have conditional error probabilities

$$P_{E_1} = \Pr\{\text{decide } H_2 | H_1\}$$

$$= \int_{-\infty}^{\infty} [1 - \phi(z)] f_1(z) dz \tag{B.2.4}$$

and

$$P_{E_2} = \Pr\{\text{decide } H_1 | H_2\}$$

$$= \int_{-\infty}^{\infty} \phi(z) f_2(z) dz \tag{B.2.5}$$

The average error probability is

$$P_E = \pi_1 P_{E_1} + \pi_2 P_{E_2}$$

$$= \int_{-\infty}^{\infty} \{\pi_1 f_1(z)[1 - \phi(z)] + \pi_2 f_2(z)\phi(z)\} dz \tag{B.2.6}$$

In the following, we examine Bhattacharyya and Chernoff bounds for various decision rules.

III. Maximum A Posteriori (MAP) Decision Rule

The decision rule that minimizes P_E is the MAP rule,

$$\phi(z) = \begin{cases} 1, & \pi_1 f_1(z) \geq \pi_2 f_2(z) \\ 0, & \pi_1 f_1(z) < \pi_2 f_2(z) \end{cases} \tag{B.3.1}$$

which satisfies the inequalities

$$\phi(z) \leq \left[\frac{\pi_1 f_1(z)}{\pi_2 f_2(z)}\right]^{\alpha} \tag{B.3.2}$$

$$1 - \phi(z) \leq \left[\frac{\pi_2 f_2(z)}{\pi_1 f_1(z)}\right]^{\beta} \tag{B.3.3}$$

for any $\alpha \geq 0$, $\beta \geq 0$. These inequalities are typically used in (B.2.4) and (B.2.5) to derive the bounds

$$P_{E_1} \leq \int_{-\infty}^{\infty} \left[\frac{\pi_2 f_2(z)}{\pi_1 f_1(z)}\right]^{\beta} f_1(z)dz$$

$$= \left(\frac{\pi_2}{\pi_1}\right)^{\beta} \int_{-\infty}^{\infty} f_1(z)^{1-\beta} f_2(z)^{\beta} dz \tag{B.3.4}$$

and

$$P_{E_2} \leq \int_{-\infty}^{\infty} \left[\frac{\pi_1 f_1(z)}{\pi_2 f_2(z)}\right]^{\alpha} f_2(z)dz$$

$$= \left(\frac{\pi_1}{\pi_2}\right)^{\alpha} \int_{-\infty}^{\infty} f_1(z)^{\alpha} f_2(z)^{1-\alpha} dz \tag{B.3.5}$$

Next define the function

$$B(\alpha) = \int_{-\infty}^{\infty} f_1(z)^{\alpha} f_2(z)^{1-\alpha} dz \tag{B.3.6}$$

Then from (B.2.6), the average error probability has the upper bound

$$P_E \le \pi_1^{1-\beta}\pi_2^{\beta}B(1-\beta) + \pi_1^{\alpha}\pi_2^{1-\alpha}B(\alpha) \tag{B.3.7}$$

for any $\alpha \ge 0$, $\beta \ge 0$. In general we would choose α and β to minimize the bounds of (B.3.4) and (B.3.5). The special case where

$$\alpha = \beta = \frac{1}{2} \tag{B.3.8}$$

results in the Bhattacharyya bound

$$\begin{aligned} P_E &\le 2\sqrt{\pi_1\pi_2}\; B(\tfrac{1}{2}) \\ &\le B(\tfrac{1}{2}) \\ &= \int_{-\infty}^{\infty} \sqrt{f_1(z)f_2(z)}\,dz \end{aligned} \tag{B.3.9}$$

since

$$\sqrt{\pi_1\pi_2} \le \frac{1}{2} \tag{B.3.10}$$

In most cases of interest, such as when*

$$f_1(z) = f_2(-z) \text{ for all } z \tag{B.3.11}$$

we have $\alpha = \frac{1}{2}$ minimizing the function $B(\alpha)$.

*When $f_1(z)$ and $f_2(z)$ are conditional probabilities of a communication channel model, this is usually the case.

Let us now reexamine the general form for the average error probability using the MAP decision rule. Note from (B.2.6) and (B.3.1) that

$$P_E = \int_{-\infty}^{\infty} \{\pi_1 f_1(z)[1 - \phi(z)] + \pi_2 f_2(z)\phi(z)\}dz$$

$$= \int_{-\infty}^{\infty} \min\{\pi_1 f_1(z), \pi_2 f_2(z)\}dz \qquad \text{(B.3.12)}$$

Following Hellman and Raviv (Ref. 3) we note that for any $a \geq 0$, $b \geq 0$ and $0 \leq \alpha \leq 1$ we have

$$\min\{a,b\} \leq a^{\alpha} b^{1-\alpha} \qquad \text{(B.3.13)}$$

This yields the upper bound on the average error probability

$$P_E \leq \int_{-\infty}^{\infty} [\pi_1 f_1(z)]^{\alpha} [\pi_2 f_2(z)]^{1-\alpha} dz$$

$$= \pi_1^{\alpha} \pi_2^{1-\alpha} B(\alpha) \qquad \text{(B.3.14)}$$

Since the minimizing choice of α is in the unit interval [0,1] then this bound is always a factor of one-half smaller than the bound given in (B.3.7). In particular for the Bhattacharyya bound where $\alpha = \frac{1}{2}$, this bound, due to Hellman and Raviv, is always a factor of one half smaller, i.e.,

$$P_E \leq \frac{1}{2} \int_{-\infty}^{\infty} \sqrt{f_1(z) f_2(z)}\, dz \qquad \text{(B.3.15)}$$

Thus, the commonly used Bhattacharyya bound, particularly in its application to deriving transfer function bit error probability bounds for convolutional codes, can be tightened by a factor of one-half.

IV. Maximum Likelihood (ML) Decision Rule

The ML decision rule, namely,

$$\phi(z) = \begin{cases} 1, & f_1(z) \geq f_2(z) \\ 0, & f_1(z) < f_2(z) \end{cases} \tag{B.4.1}$$

tends to keep both conditional probabilities closer in value but only minimizes P_E when $\pi_1 = \pi_2 = \frac{1}{2}$, i.e., the equal a priori probability case. In general, we have inequalities

$$\phi(z) \leq \left[\frac{f_1(z)}{f_2(z)}\right]^{\alpha} \tag{B.4.2}$$

and

$$[1 - \phi(z)] \leq \left[\frac{f_2(z)}{f_1(z)}\right]^{\beta} \tag{B.4.3}$$

resulting in conditional error bounds

$$P_{E_1} \leq B(1-\beta) \tag{B.4.4}$$

and

$$P_{E_2} \leq B(\alpha) \tag{B.4.5}$$

The average error probability is simply bounded by

$$P_E \leq \pi_1 B(1 - \beta) + \pi_2 B(\alpha) \tag{B.4.6}$$

While the choice $\alpha = \beta = \frac{1}{2}$ which often minimizes this bound yields the usual Bhattacharyya bound

$$P_E \leq B(\tfrac{1}{2}) \tag{B.4.7}$$

since $\pi_1 + \pi_2 = 1$.

Again using the inequality (B.3.13), we can show a tighter bound as follows:

$$P_E = \int_{-\infty}^{\infty} \{\pi_1 f_1(z)[1 - \phi(z)] + \pi_2 f_2(z)\phi(z)\}dz$$

$$\leq \max\{\pi_1, \pi_2\} \int_{-\infty}^{\infty} \{f_1(z)[1 - \phi(z)] + f_2(z)\phi(z)\}dz$$

$$= \max\{\pi_1, \pi_2\} \int_{-\infty}^{\infty} \min\{f_1(z), f_2(z)\}dz$$

$$\leq \max\{\pi_1, \pi_2\} \int_{-\infty}^{\infty} f_1(z)^{\alpha} f_2(z)^{1-\alpha} dz$$

$$= \max\{\pi_1, \pi_2\} B(\alpha) \tag{B.4.8}$$

for $0 \leq \alpha \leq 1$. For the case where

$$\pi_1 = \pi_2 = \frac{1}{2}$$

and $\alpha = \frac{1}{2}$ we again reduce the bound of (B.4.7) by a factor of one half. Most cases of interest have equal a priori probabilities.

V. Maximum Metric and Chernoff Bounds

We now assume that Z is some sort of metric used to make the decision such that for the particular value Z = z, we have the rule:

$$\text{If } z \geq 0, \text{ choose } H_1$$

$$\text{If } z < 0, \text{ choose } H_2 \tag{B.5.1}$$

The decision function is then

$$\phi(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \tag{B.5.2}$$

and conditional errors are

$$P_{E_1} = \int_{-\infty}^{\infty} [1 - \phi(z)] f_1(z) dz$$

$$= \int_{-\infty}^{0} f_1(z) dz \tag{B.5.3}$$

and

$$P_{E_2} = \int_{-\infty}^{\infty} \phi(z) f_2(z) dz$$

$$= \int_{0}^{\infty} f_2(z) dz \tag{B.5.4}$$

For $\alpha \geq 0$ and $\beta \geq 0$ we have the standard Chernoff bounds

$$P_{E_1} \leq \int_{-\infty}^{\infty} e^{-\alpha z} f_1(z) dz \triangleq C_1(\alpha) \tag{B.5.5}$$

and

$$P_{E_2} \leq \int_{-\infty}^{\infty} e^{\beta z} f_2(z) dz \triangleq C_2(\beta) \tag{B.5.6}$$

Thus, the average error probability has the upper bound

$$\boxed{P_E \leq \pi_1 C_1(\alpha) + \pi_2 C_2(\beta)} \tag{B.5.7}$$

Note that in general if P_{E_1} and P_{E_2} are less than 0.5 then the Chernoff bounds are minimized by nonnegative parameters α and β. Hence the Chernoff bounds apply for all real values of α and β.

Jacobs (Ref. 2) considered the conditions

$$f_1(-z) \geq f_1(z) \qquad \text{all } z \leq 0 \tag{B.5.8a}$$

and

$$f_2(-z) \geq f_2(z) \qquad \text{all } z \geq 0 \tag{B.5.8b}$$

Then, using the inequality

$$\frac{e^{\omega} + e^{-\omega}}{2} = \cosh \omega$$

$$\geq 1 \text{ for all } \omega \tag{B.5.9}$$

and appropriate changes of variables of integration, he showed the following inequalities:

$$C_1(\alpha) = \int_{-\infty}^{\infty} e^{-\alpha z} f_1(z)dz$$

$$= \int_{-\infty}^{0} e^{-\alpha z} f_1(z)dz + \int_{0}^{\infty} e^{-\alpha z} f_1(z)dz$$

$$= \int_{-\infty}^{0} e^{-\alpha z} f_1(z)dz + \int_{-\infty}^{0} e^{\alpha z} f_1(-z)dz$$

$$\geq \int_{-\infty}^{0} e^{-\alpha z} f_1(z)dz + \int_{-\infty}^{0} e^{\alpha z} f_1(z)dz$$

$$= 2\int_{-\infty}^{0} \cosh \alpha z \; f_1(z)dz$$

$$\geq 2\int_{-\infty}^{0} f_1(z)dz \tag{B.5.10}$$

or

$$P_{E_1} \leq \frac{1}{2} C_1(\alpha) \qquad \text{(B.5.11)}$$

Similarly, it can be shown that

$$P_{E_2} \leq \frac{1}{2} C_2(\beta) \qquad \text{(B.5.12)}$$

Thus the often satisfied condition given by Jacobs in (B.5.8) results in a factor of one half in the usual Chernoff bounds.

Less restrictive but more difficult to prove conditions are that

$$\int_0^{\infty} e^{-\alpha^* z} f_1(z)dz \geq \int_{-\infty}^{0} e^{\alpha^* z} f_1(z)dz \qquad \text{(B.5.13a)}$$

and

$$\int_{-\infty}^{0} e^{\beta^* z} f_2(z)dz \geq \int_0^{\infty} e^{-\beta^* z} f_2(z)dz \qquad \text{(B.5.13b)}$$

where α^* minimizes $C_1(\alpha)$ of (B.5.5) and β^* minimizes $C_2(\beta^*)$ of (B.5.6). Note that, for the special case of $\alpha^* = 0$, we have

$$\int_0^{\infty} f_1(z)dz = 1 - P_{E_1} \geq \int_{-\infty}^{0} f_1(z)dz = P_{E_1} \qquad \text{(B.5.14)}$$

which is always satisfied when

$$P_{E_1} < \frac{1}{2}. \qquad \text{(B.5.15)}$$

Similarly for $\beta^* = 0$, we would have

$$\int_{-\infty}^{0} f_2(z)dz = 1 - P_{E_2} \geq \int_{0}^{\infty} f_2(z)dz = P_{E_2} \tag{B.5.16}$$

which is always satisfied when

$$P_{E_2} < \frac{1}{2} \tag{B.5.17}$$

Indeed conditions (B.5.13) are also true for some nonnegative range of α^* and β^* values. We assume it is true for the minimizing choices of the Chernoff bound parameters. Note that conditions (B.5.13) are less restrictive than those of (B.5.8) since the latter imply the former but not vice versa.

Now consider the inequalities

$$\begin{aligned}
C_1(\alpha) &\geq C_1(\alpha^*) \\
&= \int_{-\infty}^{\infty} e^{-\alpha^* z} f_1(z)dz \\
&= \int_{-\infty}^{0} e^{-\alpha^* z} f_1(z)dz + \int_{0}^{\infty} e^{-\alpha^* z} f_1(z)dz \\
&\geq \int_{-\infty}^{0} e^{-\alpha^* z} f_1(z)dz + \int_{-\infty}^{0} e^{\alpha^* z} f_1(z)dz \\
&= 2\int_{-\infty}^{0} \cosh \alpha^* z \; f_1(z)dz
\end{aligned}$$

$$\geq 2\int_{-\infty}^{0} f_1(z)dz$$

$$= 2\ P_{E_1} \tag{B.5.18}$$

or

$$P_{E_1} \leq \frac{1}{2}\ C_1(\alpha). \tag{B.5.19}$$

Similarly

$$P_{E_2} \leq \frac{1}{2}\ C_2(\beta). \tag{B.5.20}$$

Thus, since (B.5.19) and (B.5.20) are identical, respectively, to (B.5.11) and (B.5.12), we have shown that the less restrictive conditions of (B.5.13) result in a factor of one-half in the usual Chernoff bounds.

Next for the special case where

$$\pi_1 = \pi_2 = \frac{1}{2} \tag{B.5.21}$$

and

$$\alpha^* = \beta^* \tag{B.5.22}$$

sufficient conditions can alternately be given by

$$\int_0^{\infty} e^{-\alpha^* z} f_1(z)dz \geq \int_0^{\infty} e^{-\alpha^* z} f_2(z)dz \tag{B.5.23a}$$

and

$$\int_{-\infty}^{0} e^{\alpha^* z} f_2(z)dz \geq \int_{-\infty}^{0} e^{\alpha^* z} f_1(z)dz \tag{B.5.23b}$$

Note that these conditions are always satisfied if our decision rule is a maximum likelihood decision rule where

$$f_2(z) \leq f_1(z) \text{ for all } z \geq 0 \tag{B.5.24a}$$

and

$$f_2(z) > f_1(z) \text{ for all } z < 0. \tag{B.5.24b}$$

Assuming conditions (B.5.23) we have

$$C_1(\alpha) + C_2(\beta) \geq C_1(\alpha^*) + C_2(\alpha^*)$$

$$= \int_{-\infty}^{\infty} e^{-\alpha^* z} f_1(z)dz$$

$$+ \int_{-\infty}^{\infty} e^{\alpha^* z} f_2(z)dz$$

$$= \int_{-\infty}^{0} e^{-\alpha^* z} f_1(z)dz + \int_{0}^{\infty} e^{-\alpha^* z} f_1(z)dz$$

$$+ \int_{-\infty}^{0} e^{\alpha^* z} f_2(z)dz + \int_{0}^{\infty} e^{\alpha^* z} f_2(z)dz$$

$$\geq \int_{-\infty}^{0} e^{-\alpha^* z} f_1(z)dz + \int_{0}^{\infty} e^{-\alpha^* z} f_2(z)dz$$

$$+ \int_{-\infty}^{0} e^{\alpha^* z} f_1(z)dz + \int_{0}^{\infty} e^{\alpha^* z} f_2(z)dz$$

$$= 2\int_{-\infty}^{0} \cosh \alpha^* z \; f_1(z)dz$$

$$+ 2\int_{0}^{\infty} \cosh \alpha^* z \; f_2(z)dz$$

$$\geq 2\, P_{E_1} + 2\, P_{E_2} \qquad \text{(B.5.25)}$$

or

$$P_E = \frac{1}{2} \cdot P_{E_1} + \frac{1}{2} P_{E_2}$$

$$\leq \frac{1}{4}\, C_1(\alpha) + \frac{1}{4}\, C_2(\beta) \qquad \text{(B.5.26)}$$

which is again a factor of one half less than the original Chernoff bound on the average error probability (B.5.7) for $\pi_1 = \pi_2 = \frac{1}{2}$.

For the special case where Z happens to be a maximum likelihood metric, i.e.,

$$\frac{f_1(z)}{f_2(z)} = e^{z} \text{ or } z = \ln\left[\frac{f_1(z)}{f_2(z)}\right] \qquad \text{(B.5.27)}$$

then the conditions (B.5.23) hold whereupon

$$C_1(\alpha) = \int_{-\infty}^{\infty} e^{-\alpha z} f_1(z)dz$$

$$= \int_{-\infty}^{\infty} \left[\frac{f_2(z)}{f_1(z)}\right]^{\alpha} f_1(z)dz$$

$$= \int_{-\infty}^{\infty} f_1(z)^{1-\alpha} f_2(z)^{\alpha} dz$$

$$= B(1 - \alpha) \tag{B.5.28}$$

and

$$C_2(\beta) = \int_{-\infty}^{\infty} e^{\beta z} f_2(z)dz$$

$$= \int_{-\infty}^{\infty} \left[\frac{f_1(z)}{f_2(z)}\right]^{\beta} f_2(z)dz$$

$$= \int_{-\infty}^{\infty} f_1(z)^{\beta} f_2(z)^{1-\beta} dz$$

$$= B(\beta) \tag{B.5.29}$$

where $B(\cdot)$ is defined in (B.3.6). Recall that $B(\frac{1}{2})$ is the Bhattacharyya bound. Thus, (B.5.28) and (B.5.29) together with (B.2.6), (B.5.19), and (B.5.20) again show a reduction by a factor of one half in the bound of (B.4.6).

VI. Applications

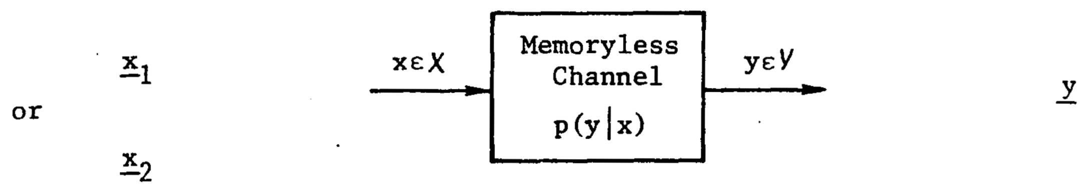
In most applications of interest, we consider two sequences of length N,

$$\underline{x}_1, \underline{x}_2 \varepsilon \mathcal{X}^N$$

that can be transmitted over a memoryless channel with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$ and conditional probability

$$P(y|x); \ x \varepsilon \mathcal{X}, y \varepsilon \mathcal{Y}$$

This is shown in the following figure.



$$p_N(\underline{y}|\underline{x}_i) = \prod_{n=1}^{N} p(y_{in}|x_{in}); \ i = 1,2$$

Figure B-1. A Simple Example - One of Two Sequences Transmitted over a Memoryless Channel

The receiver obtains a sequence

$$\underline{y} \ \varepsilon \ \mathcal{Y}^N \tag{B.6.1}$$

from the channel and must decide between the two hypotheses

$$H_1: \ \underline{x}_1 \text{ is sent}$$

$$H_2: \ \underline{x}_2 \text{ is sent} \tag{B.6.2}$$

which have a priori probabilities given by (B.2.2). The receiver will typically use a metric

$$m(y,x); \ x\varepsilon\mathcal{X}, y\varepsilon\mathcal{Y}$$

and the corresponding decision rule where, if and only if

$$\sum_{n=1}^{N} m(y_n,x_{1n}) \geq \sum_{n=1}^{N} m(y_n,x_{2n}) \ , \tag{B.6.3}$$

do we choose H_1. By defining the random variable

$$Z = \sum_{n=1}^{N} [m(y_n,x_{1n}) - m(y_n,x_{2n})] \tag{B.6.4}$$

we have the basic problem considered in previous sections.

For M sequences of length N denoted $\underline{x}_1$, $\underline{x}_2$, ..., $\underline{x}_M$, we have the decision rule: Given $\underline{y}\varepsilon\mathcal{Y}^N$ choose the sequence $\underline{x}_{\hat{m}}$ that has the largest total metric

$$\sum_{n=1}^{N} m(y_n,x_{\hat{m}n})$$

for $\underline{x}_{\hat{m}} = (x_{\hat{m}1},x_{\hat{m}2}, \ldots,x_{\hat{m}N}) \ \varepsilon \ \mathcal{X}^N$. (B.6.5)

The union bound for each conditional error probability is,

$$P_{E_m} = \Pr\{\text{error}|\underline{x}_m\} \le \sum_{\hat{m}\ne m} \Pr\{\text{decide } \underline{x}_{\hat{m}}|\underline{x}_m\};\ m = 1,2, \ldots, M. \tag{B.6.6}$$

Here we have

$$\Pr\{\text{deciding } \underline{x}_{\hat{m}}|\underline{x}_m\} \le P(\underline{x}_m \to \underline{x}_{\hat{m}}) \tag{B.6.7}$$

where $P(\underline{x}_m \to \underline{x}_{\hat{m}})$ is the probability of deciding $\underline{x}_{\hat{m}}$ when $\underline{x}_m$ is sent assuming $\underline{x}_m$ and $\underline{x}_{\hat{m}}$ are the only two possible sequences. That is,

$$P(\underline{x}_m \to \underline{x}_{\hat{m}}) = \Pr\left\{\sum_{n=1}^{N} [m(y_n, x_{\hat{m}n}) - m(y_n, x_{mn})] \ge 0 \middle| \underline{x}_m\right\} \tag{B.6.8}$$

which is the two hypothesis error probability. Thus, in this case, we can apply our two hypothesis results discussed earlier.

REFERENCES

1. Viterbi, A. J. and Omura, J. K., Principles of Digital Communication and Coding, McGraw-Hill Book Co., 1979.

2. Jacobs, I. M., "Probability-of-Error Bounds for Binary Transmission on the Slowly Fading Rician Channel," IEEE Transactions on Information Theory, Vol. IT-12, pp. 431-441, October 1966.

3. Hellman, M. E. and Raviv, J., "Probability of Error, Equivocation, and the Chernoff Bound," IEEE Transactions on Information Theory, Vol. IT-16, pp. 368-372, July 1970.