Research Report

No. 82-2

LOW COMPLEXITY DECODERS FOR
BANDWIDTH EFFICIENT
DIGITAL PHASE MODULATIONS

PREPARED FOR

THE DEPARTMENT OF COMMUNICATIONS
UNDER DSS CONTRACT No. OSU81-00246

BY

S.J. SIMMONS
P.H. WITTKE

MARCH 1982

Queen's University at Kingston
Department of Electrical Engineering

LOW COMPLEXITY DECODERS FOR BANDWIDTH EFFICIENT DIGITAL PHASE MODULATIONS

by

S.J. Simmons
P.H. Wittke

Report No. 82-2

Final Report - Part I

Prepared for
The Department of Communications
Under DSS Contract No. OSU81-00246

Department of Electrical Engineering
Queen's University
Kingston, Ontario, Canada

ABSTRACT

Bandwidth efficient digital angle modulations having input symbol memory can be demodulated using maximum likelihood sequence estimation(or Viterbi decoding). Unfortunately, the more bandwidth efficient of these tend to have many states in their Markov process description, and MLSE can be computationally complex. Lower complexity decoding approaches are presented for these modulations.

These decoders use a structured processing order and a general number of survivor signals, S, at every time NT. Processing is performed on the signal sequences using metrics (likelihoods) obtained by a matched filter bank similar to that needed for MLSE. The decoders can achieve asymptotic optimality of error rate while being computationally faster than MLSE for many modulations. In addition, error rate performance can be traded for complexity reduction.

Computational reduction is by a factor of S_v/S where S_v is the number of Viterbi states and S is the number of survivors retained by the new decoders. The lower the modulation index, h, the greater the savings. For example, computational reduction is by a factor of n for indices h=1/n. Simulations are performed for representative modulations: partial response FM with polynominals $(1+D)/2$, $(1+2D+D^2)/4$, and $(1+D+D^2)/3$ over a range of modulation indices less than unity.

Certain modulation indices ($h \neq 1/n$) can lead to a "false lock" condition and error events of substantial lengths (tens to thousands of bits). Expanded decoding rules are developed to deal with this problem and are shown by simulation to greatly reduce the length of long error events. Results suggest that any decoder that does not maintain a survivor for every Markov state at every time NT can encounter these long error events.

Indexing terms: Low complexity decoders, sequence estimation, bandwidth efficient, angle modulations.

CONTENTS

FIGURES

TABLES

NOTATION

Subscripts

i,j	elements of a set
k,n,m	time in digit intervals
t,a	transmitted, actual
min	minimum
$m\ell$	maximum likelihood
V	Viterbi

Superscripts

-	(overbar) vector representation
~	sequence

Symbols

a	input digit
d	distance
f_c	carrier frequency
h	modulation index
p	number of phase states

Symbols

p_i	projection for i'th signal
r	received signal
t	time
A	amplitude
C	correlation
D	distance
L	likelihood
M	input alphabet size
N	number of digit intervals
N_o	single sided noise power density
R	rule parameter
S	number of survivors
T	digit interval
θ	phase
ϕ	phase state
$\mathcal{S}_{ij}^{N}$	subspace defined by signals i, j up to time NT

Mathematical Conventions

$[\cdot]_m^n$	over the time interval $[mT, nT]$
$\|\|\cdot\|\|$	Euclidean magnitude
$[\bar{v}]_{\mathcal{S}}$	projection of vector $\bar{v}$ onto subspace $\mathcal{S}$
$\forall(i,j)$	over all pairs (i,j)
$\bar{a}\cdot\bar{b}$	dot product of vectors $\bar{a}$, $\bar{b}$
D	delay operator
max	maximum
min	minimum
mod x	modulo x

CHAPTER 1

INTRODUCTION

Bandwidth efficient modulation schemes have received considerable attention in recent years. Development of these modulations has come as part of the search for ways of meeting increased data communications demands while minimizing the use of valuable spectral resources. Constant envelope signalling has been of particular interest because of its special immunities to fading, non-linear distortion, and AM-PM conversion.

These constant envelope signals are generated by encoding data symbols and modulating the phase of a constant amplitude carrier. Such a signal may be written in general form (assuming zero initial phase) as

$$y(t) = A \cos\left(2\pi f_c t + \sum_{k=0}^{\infty} a_k g(t-kT)\right) \qquad (1)$$

In this equation, A is the carrier amplitude and f_c its frequency. The data digits a_k are assumed chosen at random from a finite input alphabet $\{a_i\}$, one every T seconds. The input alphabet is of size M with elements $\{\pm1, \pm3, \ldots, \pm(M-1)\}$.

The information is contained in the phase function,

$$\sum_{k=0}^{\infty} a_k g(t-kT).$$

This summation represents the superposition of the phase responses due to the input digits. The modulation index h is defined such that the maximum phase change (relative to the carrier) over any symbol interval is $(M-1)h\pi$ radians. The modulation index therefore depends on the magnitude and shape of g(t).

The function g(t) may be expressed as

$$g(t) = \int_{0}^{t} h(\tau)d\tau \qquad (2)$$

where h(t) is the instantaneous frequency pulse. The duration of h(t) determines the degree of signal dependence on past digits, i.e. the memory in the system.

The spectral occupancy of many of these modulations has been studied, as well as the minimum distance properties of the effective code created by the modulation [1 - 9]. The benchmark constant envelope modulation, against which most comparisons are made, is minimum shift keying (MSK) [1]. The new modulations have shown substantial reductions in bandwidth over MSK, usually with corresponding degradations in coding distance. In addition it is also possible to aim for coding distance gains, usually at the expense of bandwidth [6-10].

In general, the smoother the resultant phase path of the signal for a given modulation index, the lower the signal bandwidth. Here bandwidth refers to the frequency band which contains a specified fraction of the total signal power (e.g. 99%, 99.9%).

These modulations are non-linear in nature, MSK being an exception [11], and standard receiver techniques to deal with memory in linear signalling (equalizers, decision feedback) do not apply. However, the modulations can generally be represented by finite state, first order Markov processes, and maximum likelihood sequence estimation (MLSE or Viterbi detection [12]) has been proposed as a decoding method [13 - 16].

Introduction of longer pulses h(t) has an important beneficial effect; the overall phase path is smoother resulting in lower bandwidth for a given modulation index. Unfortunately, this increased memory greatly increases the number of states in the Markov description. In addition, the number of states depends on the modulation index (see Section 1.3.3), an extra factor contributing to complexity not found in linear modulations. The combined effect of long memory and certain modulation indices (especially low ones) can make Viterbi decoding very complex.

A simple quadrature component receiver has been demonstrated for one particular angle modulation called tamed FM [17]. This receiver incurrs a loss of over 2 dB in effective signal power compared to optimum reception. Such simple receivers do not seem to be generally applicable to the broad class of constant envelope modulations without substantial suboptimality compared to MLSE. Sub-optimum decoding schemes which are computationally faster than the Viterbi algorithm, but which exhibit little degradation in error-rate performance, are therefore desirable.

This study concentrates on the description and development of lower complexity decoders and their application to one particular type of modulation, partial response FM. The basis for adopting these schemes is completely general in nature. This specific application clearly demonstrates the salient features and logical extensions to other modulations can be readily made.

The starting point for the development of the techniques described here is a basic approach proposed by Vermuellen [18] for reducing receiver complexity with PAM signalling over channels producing intersymbol interference. The basis for complexity reduction lies in maintaining a list of "survivor" signals (and corresponding input sequences) at every time kT reduced in size from that kept in Viterbi decoding. This reduction is effected by a sequential[1] rule that identifies those signal paths with likelihoods small enough to be dropped from further contention, while attaining some specific overall error exponent.

New rules are developed for application to the modulations of interest by a series of modifications to a rule based on the notion of signal space projections. The resulting rules incorporate these modifications along with an ordered processing approach, without which computational savings over MLSE are impossible.

To the author's knowledge, this is the first time such new reduced complexity rules based on signal space considerations have been applied to the kinds of modulations of interest here. The non-linear nature of these modulations leads to unexpected and interesting behaviour in some cases, requiring development of expanded processing decoders.

[1]It should be pointed out that 'sequential' is used here to describe any decoder that operates on <u>sequences</u>. The rules developed here allow no "backtracking" and bear little resemblance to the so called sequential decoder first proposed by Fano [19].

For the purposes of this study, the channel is assumed ideal (distortion-free), the noise Gaussian and white, and the receiver perfectly synchronous. The decoders presented here can achieve optimality in the sense that

$$\lim_{SNR\to\infty} \left(\frac{P(e)}{KP_{OPT}(e)}\right) = 1$$

where SNR is the signal to noise ratio, K a constant, P(e) the probability of decoder symbol error, and $P_{OPT}(e)$ the optimum probability of symbol error for MLSE. The equivalent statement for the probability of event error is proven; the finite length of error events required for the previous assertion is demonstrated by simulation.

The general decoding problem will now be presented from a signal space point of view. This will serve to introduce concepts and notation for later use. This will be followed by a more detailed description of constant envelope modulation and partial response FM. Viterbi decoding will be presented. Chapter 2 will present approaches to reducing receiver complexity and Chapter 3 will show the application of these schemes to partial response FM. Chapter 4 provides a summary of the findings.

1.1 Signal Space Description of The Problem

Consider a general encoder/modulator which transforms information digits into a suitable set of signals for transmission over a channel. In many cases, the modulator output depends not only on the present input digit but also on previous digits. This is the case with convolutional encoders. If such an encoder is modelled as a Markov process, the modulator output is completely determined by the state at the end of the previous interval along with the present input digit. A state transition is produced by each new information digit every T seconds. A unique output signal of duration T is associated with each of the state transitions. Every input digit sequence therefore has a one-to-one correspondence with an output signal sequence.

Suppose one of a set of equally likely signals $\{y_i\}$ corresponding to the information digit sequences $\{\tilde{a}_i\}$ is to be transmitted over an ideal channel corrupted by Gaussian white noise. We are to determine which sequence is a-posteriori most probable by observing the channel output. This is equivalent to finding the corresponding signal with highest likelihood. If $y_i(t)$ was transmitted, and the received signal is r(t), the noise process must have assumed the form $n_i(t) = r(t) - y_i(t)$. The equivalent problem becomes to decide which sample function $n_i(t)$ of the noise process has greatest likelihood. It is well known that this problem can be cast in a vector space format.

Each signal is represented as a weighted sum of orthonormal (orthogonal with unit energy) functions. Each distinct signal is constructed from the same set of functions but with different weightings. Each orthonormal function represents one direction in the signal vector space, and the weighting coefficients represent the coordinates of the vectors describing these signals. Define the Vector $\overline{y}_i$ as the vector of these weighting coefficients for signal $y_i(t)$.

The square of the distance between any pair of vectors $(\overline{y}_i, \overline{y}_j)$ is related to the time signals by

$$d_{ij}^2 = \int_I (y_i(t) - y_j(t))^2 \, dt \qquad (3)$$

This is the integral of the squared difference between the corresponding time signals over the interval (I) of interest.

The relevant portion of the noise process may be described by a noise vector formulated identically to the signal vectors. Given that the noise is white, this noise vector will have independent components. For Gaussian white noise of double-sided power spectral density $N_o/2$ watts/Hz, these components are also of zero mean and equal variance $\sigma^2=N_o/2$.

The received signal vector is simply the transmitted signal vector plus the relevant noise vector. The distance between the received vector $\bar{r}$ and any signal vector $\bar{y}_i$ is then the magnitude that the noise vector must have assumed for $\bar{y}_i$ to have been the signal actually transmitted. A conditional noise vector may then be identified for each signal.

The probability density function for a vector of independent Gaussian random variables of equal variance is exponentally decreasing with the squared vector magnitude. The most likely of the conditional noise vectors is therefore the shortest one. The most likely signal (given that all are a priori equally probable) is then the one closest to the received signal in terms of Euclidian distance. The likelihood of any signal y_i refers to the value of the probability density function evaluated at the length of the noise vector, conditioned on y_i having been transmitted. The logarithm of this likelihood is (less a common constant) proportional to the square of the distance separating received signal $\bar{r}$ and signal $\bar{y}_i$.

The decision regions are bounded by hyperplanes in the n-space which bisect the lines joining the tips of the signal vectors. To decide which of two signals, y_i or y_j, is more probable, the process is equivalent to projecting the vectors $(\bar{r} - \bar{y}_i)$ and $(\bar{r} - \bar{y}_j)$ onto the subspace $\mathcal{S}_{ij} \triangleq (\bar{y}_j - \bar{y}_i)$ and choosing the signal with the shorter projection.

Any signal y_i may be expressed as a sum of sections over distinct time intervals, specifically, choosing these intervals to align with the information digit intervals,

$$[y_i]_0^N = [y_i]_0^1 + [y_i]_1^2 + \cdots + [y_i]_{k-1}^k + \cdots + [y_i]_{N-1}^N$$

where $[y]_m^n$ represents the segment of y over interval $[mT, nT]$. In terms of signal space, because the signal sections are disjoint in time, their vector representations are orthogonal. With this expansion, the overall signal space spanning the long signals can be decomposed into orthogonal subspaces, one per digit interval.

The vectors representing $[r]_0^N$ and any signal $[y_i]_0^N$ may then be expressed as the sum of orthogonal vectors defined for the segments of these signals over each digit interval. The relevant noise vector is also a sum of orthogonal vectors, but because the noise is Gaussian and white, these vectors are independent from one another. The probability density of the overall noise vector is therefore the product of the probability densities for each of these independent noise vectors. We can then identify the likelihood of any signal $y_i(t)$ as the product of the likelihoods of the noise vectors conditioned on the segments of $y_i(t)$ given the received signal r(t) over each interval. The log likelihood will then be the sum of the likelihoods due to each interval. If $[\rho_i]_{k-1}^k$ is the likelihood for signal segment $[y_i]_{k-1}^k$, then

$$[\rho_i]_0^N = \prod_{k=1}^{N} [\rho_i]_{k-1}^k$$

and

$$\ell n\,[\rho_i]_0^N = \sum_{k=1}^{N} \ell n[\rho_i]_{k-1}^k \tag{4}$$

For each interval,

$$[\rho_i]_{k-1}^k = K_1 \exp(-K_2 \cdot ||[\bar{r}-\bar{y}_i]_{k-1}^k||^2) \tag{5}$$

where K_1 and K_2 are constants depending on the noise power, $[\bar{r} - \bar{y}_i]_{k-1}^k$ is the vector from signal vector $[\bar{y}_i]_{k-1}^k$ to $[\bar{r}]_{k-1}^k$, and $||\cdot||$ denotes magnitude or norm which is the Euclidean distance from the vector to the zero element.

We have

$$\ell n[\rho_i]_{k-1}^k = -K_2 \cdot ||[\bar{r}-\bar{y}_i]_{k-1}^k||^2 + K_3$$

and

$$\ell n[\rho_i]_0^N = -K_2 \cdot \sum_{k=1}^{N} ||[\bar{r}-\bar{y}_i]_{k-1}^k||^2 + K_3$$

$$= -K_2 \cdot ||[\bar{r}-\bar{y}_i]_0^N||^2 + K_3 \tag{6}$$

where the K_j are constants.

The overall log likelihood of a signal $\bar{y}_i$ is proportional to the distance from received signal $\bar{r}$ to signal $\bar{y}_i$ as expected.

In order to choose the most probable signal, we must observe the received signal for the duration of the transmitted signal. Since these signals correspond to sequences of information digits, they can be of very long duration and there could be very long decoding delays. Also, since the source is providing a new information digit every T seconds, there will be an exponential increase with time in the number of possible signals, making decision after complete observation formidable. It is clear that some sort of sequential decision process is needed. The Viterbi algorithm exploits the highly structured Markov process description of the modulation to effect such a decoding [12].

It is well-known that for practical modulations, such maximum likelihood sequence estimation will work well in terms of symbol error probability. For reasonably high signal-to-noise ratio the maximum likelihood (ML) signal must with high probability be close to the transmitted signal in signal space. For the usual effective encoding schemes, signals that are close to one another differ in only a few digits in their corresponding input sequences. Therefore, for reasonably high SNR, the ML input digit sequence will differ in only a small fraction of places from the transmitted digit sequence.

For longer and longer sequences, it becomes more likely that a signal further away from the transmitted signal will be maximum likelihood. The total number of digit errors should tend to increase with the length of the transmitted sequence. The notion of error events may be used to verify this expectation and to develop a bound on the symbol error probability.

1.2 Viterbi Decoding (MLSE)

While it is true that an infinite delay is involved in finding the most likely of a set of infinite length signals, it is known that with probability one, only a finite delay is required to decide on the most likely sequence of digits transmitted up to time KT.

We know that any signal corresponds to an input digit sequence and a path of state transitions. We can say that signal y_i "passes through" state s_n at time KT if s_n is the state at time KT of the state sequence that uniquely corresponds to signal y_i.

Consider the ensemble of signals $\{[y_j^n]_O^N = [y_j^n]_O^K + [y_{m\ell}^n]_K^N\}$ that pass through state s_n at time KT, and that share the future segment $[y_{m\ell}^n]_K^N$ that has maximum likelihood of all signals emerging from state s_n. Any signal path that can reach state s_n at KT is capable of spawning $[y_{m\ell}^n]_K^N$ regardless of the actual $[y_{m\ell}^n]_K^N$. From the discussion of Section 1.1, the overall log likelihood of any of these signals may be broken into the sum of two log likelihoods, that portion before KT,and that portion after KT;

$$[r_j^n]_O^N = [r_j^n]_O^K + [r_{m\ell}^n]_K^N \tag{7}$$

where

$$[r_i]_m^n = \ell n[\rho_i]_m^n$$

Clearly, the signal in this ensemble that has maximum overall likelihood, $[y^n_{m\ell}]^N_0$, must be the one with initial segment $[y^n_{m\ell}]^K_0$ whose likelihood $[\Gamma^n_{m\ell}]^K_0$ is maximum over all the segments reaching s_n at time KT,

$$[\Gamma^n_{m\ell}]^K_0 = \max_{\forall j} [\Gamma^n_j]^K_0$$

Now if the overall maximum likelihood signal does in fact pass through state s_n at time KT, it must be signal $[y^n_{m\ell}]^N_0$, and therefore have initial segment $[y^n_{m\ell}]^K_0$. The same may be argued for every other possible state at KT. The ML signal must pass through one of the S possible states at every time KT. If we keep only those signals (and corresponding sequences) having maximum likelihood to each state at every time KT, we are assured that one of them is the initial segment of the overall ML signal. The problem is then reduced from working with an exponentially increasing number of signals, to a finite list of S survivor sequences and likelihoods [12].

Each of the S survivor segments will spawn M descendants at time (K+1)T for an M-ary input alphabet. At time (K+1)T, M·S signals will terminate in S states. The algorithm then keeps the segment with largest log likelihood to that time for each state, reducing the list back to S survivors.

Decoding of the ML sequence relies on these S survivors sharing a common history in the not too distant past. First we note that no two of these survivors can share a common state at any time LT and still be distinct up to time LT. Secondly, we can always get from one state to any other within a finite number of (usually few) digit intervals. All survivors will diverge from a common state to reach each of the S possible states.

If the survivor $[y_s^n]_0^K$ for the state s_n is distinct from the ML signal for a long time, it will in general be widely separated from it in signal space (distance is monotonically non-decreasing with separation time). We might expect with high probability to be able to find an alternate signal $[y_a^n]_0^K$ to state s_n that shares a longer common history with the ML signal. Such an alternate signal would be closer to the ML signal and almost certainly have greater likelihood. It should then displace $[y_s^n]_0^K$ as the survivor to state s_n at KT. This argument implies that all survivors to time KT must be reasonably close together and with high probability share a common history a few intervals before KT.

Once this merge has occurred, the common history portion can be unambiguously declared as the initial segment of ML signal. This provides a finite decoding delay for the input digits. There is a finite probability that this merge will not have occurred within the constrained memory length of the receiver. In such an instance, overflow is said to occur. If the receiver memory is chosen long enough, this phenomenon should have negligible effect on the probability of error.

The fact that the log likelihoods of the signals can be broken into the sum of log likelihoods due to each segment also allows these likelihoods to be calculated recursively by the receiver. The relevant portion of these log likelihoods is formed by matched filtering of the received signal. To see this, denote $[\bar{r}]_{k-1}^{k}$ by $\bar{R}$ and $[\bar{y}_i]_{k-1}^{k}$ by $\bar{y}$. Then from Equation 6

$$\ln[\rho_i]_{k-1}^{k} = K_3 - K_2 \cdot ||\bar{R} - \bar{y}||^2$$

$$= K_3 - K_2 \cdot (||\bar{R}||^2 + ||\bar{Y}||^2 - 2\bar{R} \cdot \bar{Y})$$

where K_1 and K_2 are constants common to all signals. The quantity $||\bar{R}||^2$ is common to all signals, and because they are equi-energy, so is $||\bar{Y}||^2$. Since we are interested only in relative likelihoods, all common quantities may be dropped giving the relevant likelihood,

$$[L_i]_{k-1}^{k} = \bar{R} \cdot \bar{Y} \tag{8}$$

This dot product is expressed in the time domain by the correlation

$$\bar{R} \cdot \bar{Y} = \int_{(k-1)T}^{kT} r(t) \cdot y_i(t)\, dt \tag{9}$$

This is simply the output of a filter matched to the segment of $y_i(t)$ over the interval $[(k-1)T, kT]$. The total relevant likelihood or path metric of signal y_i is the sum of the sectional likelihoods,

$$[L_i]_0^N = \sum_{k=1}^{N} \int_{(k-1)T}^{kT} r(t) \cdot y_i(t) dt$$

$$= \int_0^{NT} r(t) \cdot y_i(t)\, dt \tag{10}$$

or

$$L_i(N+1) = L_i(N) + C_i(N,N+1) \tag{11}$$

where

$$C_i(N,N+1) = \int_{N}^{(N+1)T} r(t)\cdot y_i(t)\, dt$$

This is the recursive formulation of the path metric. The receiver obtains the relevant quantity $C_i(N, N+1)$ as the output of the appropriate matched filter and adds it to the accumulated likelihood (likelihood will be used to mean the relevant portion of the log likelihood, or path metric). Since there are a finite number of states, there are a finite number of state transitions, and therefore a finite number of signal sections of duration T seconds to which filters must be matched.

An error in decoding occurs when the ML sequence deviates from the transmitted sequence. This occurs when a sequence merging with the state of the true sequence at time KT has greater likelihood. This will occur when a component of the noise vector exceeds half the distance d between the transmitted signal and the ML signal. For white noise, the noise vector components have equal variance in all directions. The probability of such an error event is then simply $Q(d/2\sigma)$ where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\alpha^2/2}\, d\alpha$$

and $\sigma^2 = N_o/2$.

For high SNR, the most errors will tend to occur when noise exceeds half the smallest distance between distinct signals, d_{min}, and the signal is decoded into its nearest neighbour. The length of these error events will depend on the particular modulation.

1.3 Constant Envelope Modulation

As already presented, the general form of a constant envelope signal may be written as (Equation 1)

$$y(t) = A \cos\left(2\pi f_c t + \sum_{k=0}^{\infty} a_k g(t-kT)\right)$$

The phase response g(t) determines the ultimate bandwidth and distance structure for a particular input alphabet $\{a_i\}$. The various modulations really only differ in the shape and method of realizing this phase response. Some even allow g(t) to be cyclically altered for different digit intervals [9 - 10].

It is very helpful to think of the signal in terms of its phase relative to the carrier and of the resultant phase path it follows as a function of time. This phase time function,

$$\sum_{k=0}^{\infty} a_k g(t-kT),$$

consists of the superposition of the phase responses due to the input digits, a_k. The phase response function is the integral of the frequency pulse h(t),

$$g(t) = \int_0^t h(\tau)\, d\tau$$

The duration of the basic frequency pulse h(t) determines the memory in the system. If h(t) is everywhere bounded, there will be no discontinuities in g(t), resulting in a continuous phase signal. For many modulations, and the ones of interest here, the area under h(t) is non-zero. In these cases g(t) is of infinite duration and there is a kind of infinite intersymbol interference. Usually h(t) is everywhere positive and limited to a duration LT, giving $g(t)=0$ for $t<0$ and $g(t)=g(LT)$ for $t>LT$. The maximum phase change over any digit interval is equal to $(M-1)\cdot g(LT)$, giving a modulation index $h=g(LT)/\pi$.

It is best to think of the memory in terms of the duration of h(t). When h(t) is of duration T, the phase path during any interval depends only on the phase at the end of the previous interval and the present input. If h(t) is of duration greater than T, the phase path depends on the accumulated past phase, the present digit, and one or more past input digits.

A very useful tool for understanding these phase modulations is the phase tree. The tree is the ensemble of phase trajectories for all possible sequences with a common history at the tree's root node. Since all phases are modulo 2π, this tree actually wraps around a phase cylinder [10] of circumference 2π. For rational values of h, the tree collapses into a trellis with a finite number of phases at every time KT. At every time KT, one of M possible new digits is introduced allowing the phase path to branch in M new directions.

1.3.1 Partial Response FM

Partial response (or correlative encoded) FM is one technique for generating the phase response function g(t). This method produces a correlated digit stream $\{\ldots b_k \ldots\}$ from a sequence of independent digits $\{\ldots a_k \ldots\}$ applied to a tapped delay line. The delay line is characterized by the polynomial $P(D) = (K_0+K_1D+K_2D^2+\ldots+K_mD^m)/C$ where the delay operator D denotes a delay of T seconds and

$$C = \sum_{\ell=0}^{m} |K_\ell|$$

normalizes the polynomial. This correlated digit stream is applied to a constant envelope modulator with basic frequency pulse $h_b(t)$. The structure is shown in Figure 1.

An equivalent modulator exists which maintains the original independent digit stream $\{\ldots a_k \ldots\}$ as input.

The effect of the correlative encoding is to create a new equivalent frequency pulse h(t) [4]. This pulse is given by

$$h(t) = \frac{1}{C} \sum_{i=0}^{m} K_i h_b(t-iT) \qquad (12)$$

The partial response modulations considered in this work have polynomials $(1+D)/2$, $(1+2D+D^2)/4$, and $(1+D+D^2)/3$, with basic frequency pulse $h_b(t)$ rectangular and of duration T.

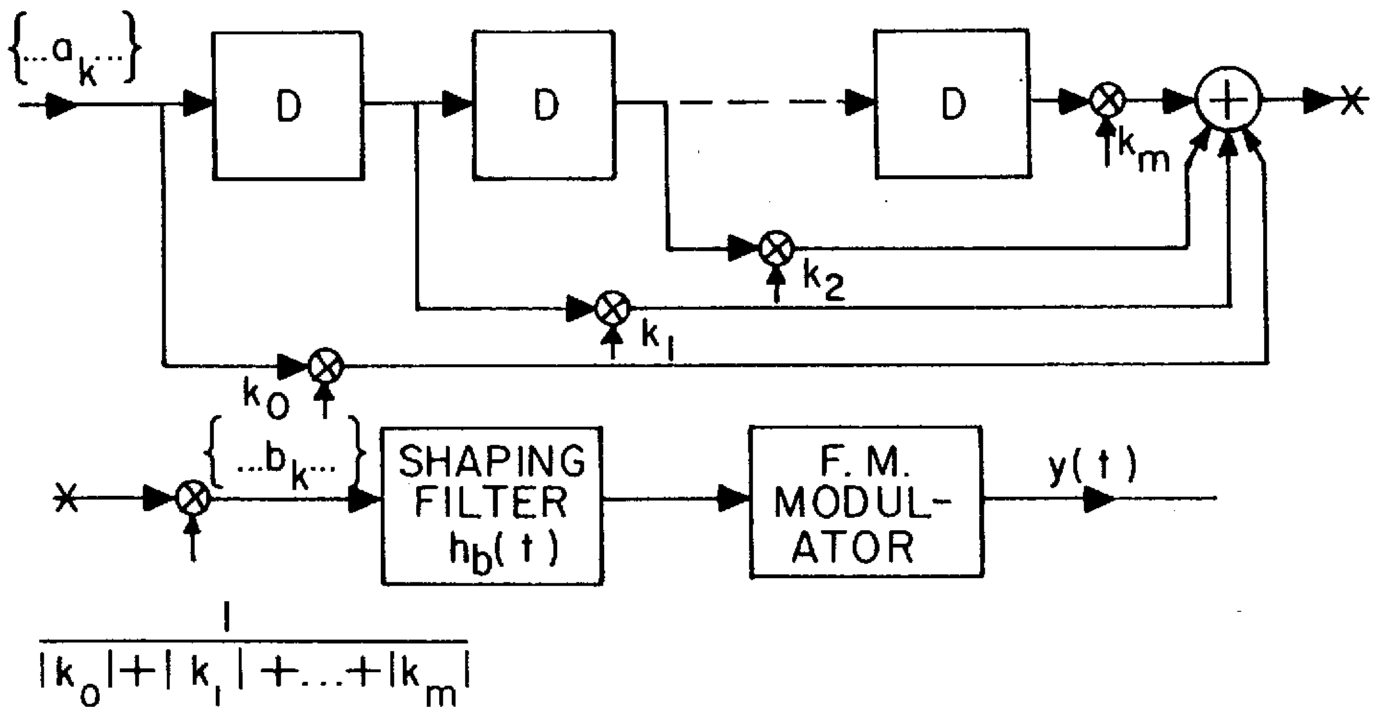

FIGURE 1 Diagram of conceptual partial response FM modulator.

These modulations will be referred to in the format [polynominal, h=mod. index]. The equivalent frequency pulses h(t) and phase responses g(t) for these modulations for arbitrary modulation index h are shown in Figure 2. The resultant phase trees for given bit histories are shown in Figures 3(a), (b), (c). Bit history refers to the time sequence of previous bits.

1.3.2 Distance Structure

The squared distance between any two signals $y_i(t)$, $y_j(t)$, over any interval of interest I is expressed as (Equation 3)

$$d_{ij}^2 = \int_I [y_i(t) - y_j(t)]^2 \, dt$$

If the corresponding phase paths are denoted by $\theta_i(t)$ and $\theta_j(t)$, then from Equation 1

$$d_{ij}^2 = \int_I A^2[\cos(2\pi f_c t+\theta_i(t))-\cos(2\pi f_c t+\theta_j(t))]^2 \, dt$$

$$= \int_I A^2\{\cos^2(2\pi f_c t+\theta_i(t))+\cos^2(2\pi f_c t+\theta_j(t))$$

$$-2\cos(2\pi f_c t+\theta_i(t))\cos(2\pi f_c t+\theta_j(t))\} \, dt$$

We will make assumptions usually made in modulation considerations:

(a) The phase variations due to $\theta_i(t)$, $\theta_j(t)$ are very slow compared to those due to the carrier at f_c, and

(b) $f_c T_I \gg 1$ where T_I is the minimum length of any interval of interest.

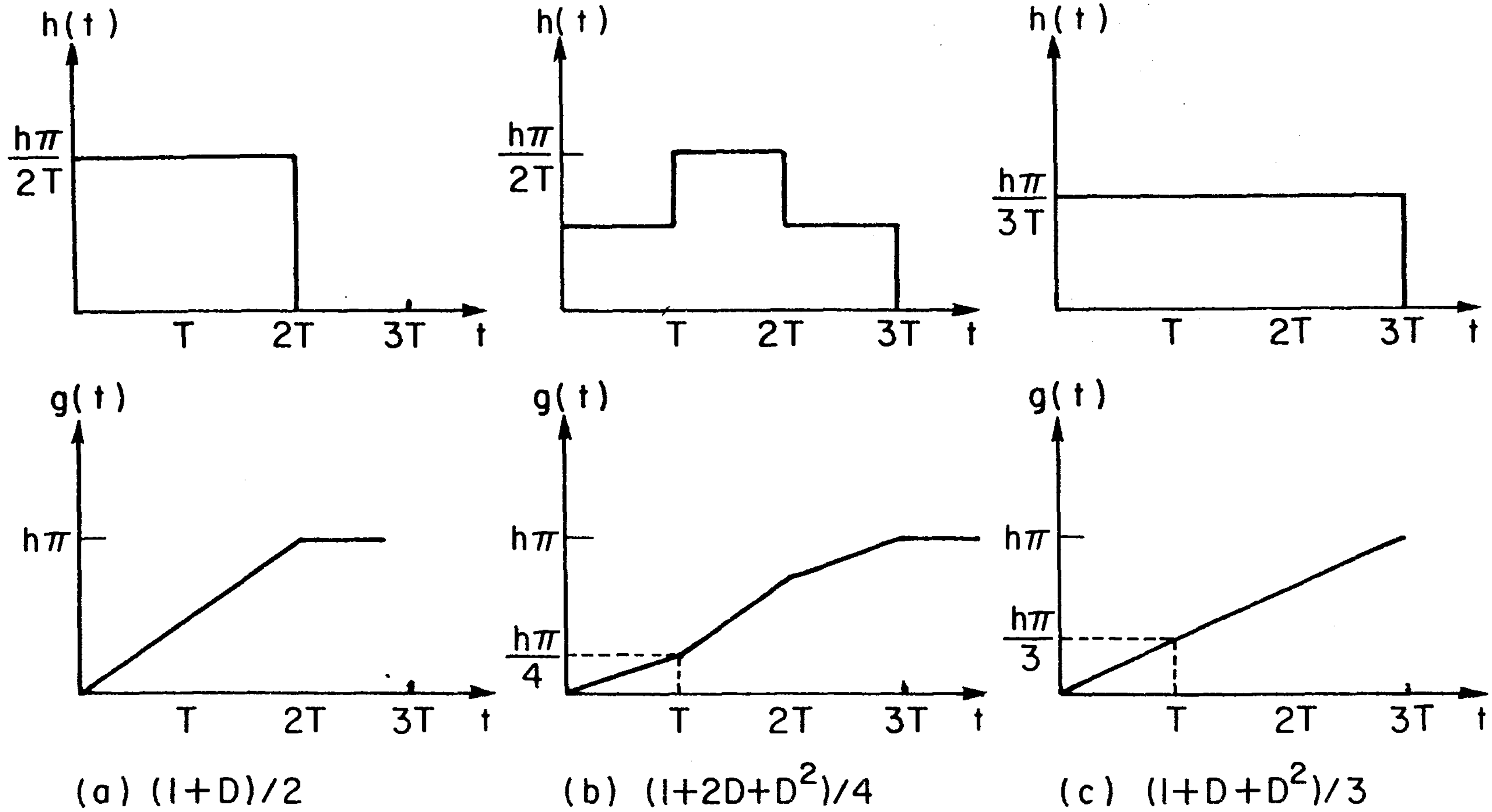

FIGURE 2 Frequency and phase responses for PRS modulations with modulation index h.
Basic frequency pulse $h_b(t)$ is rectangular, with duration T seconds.

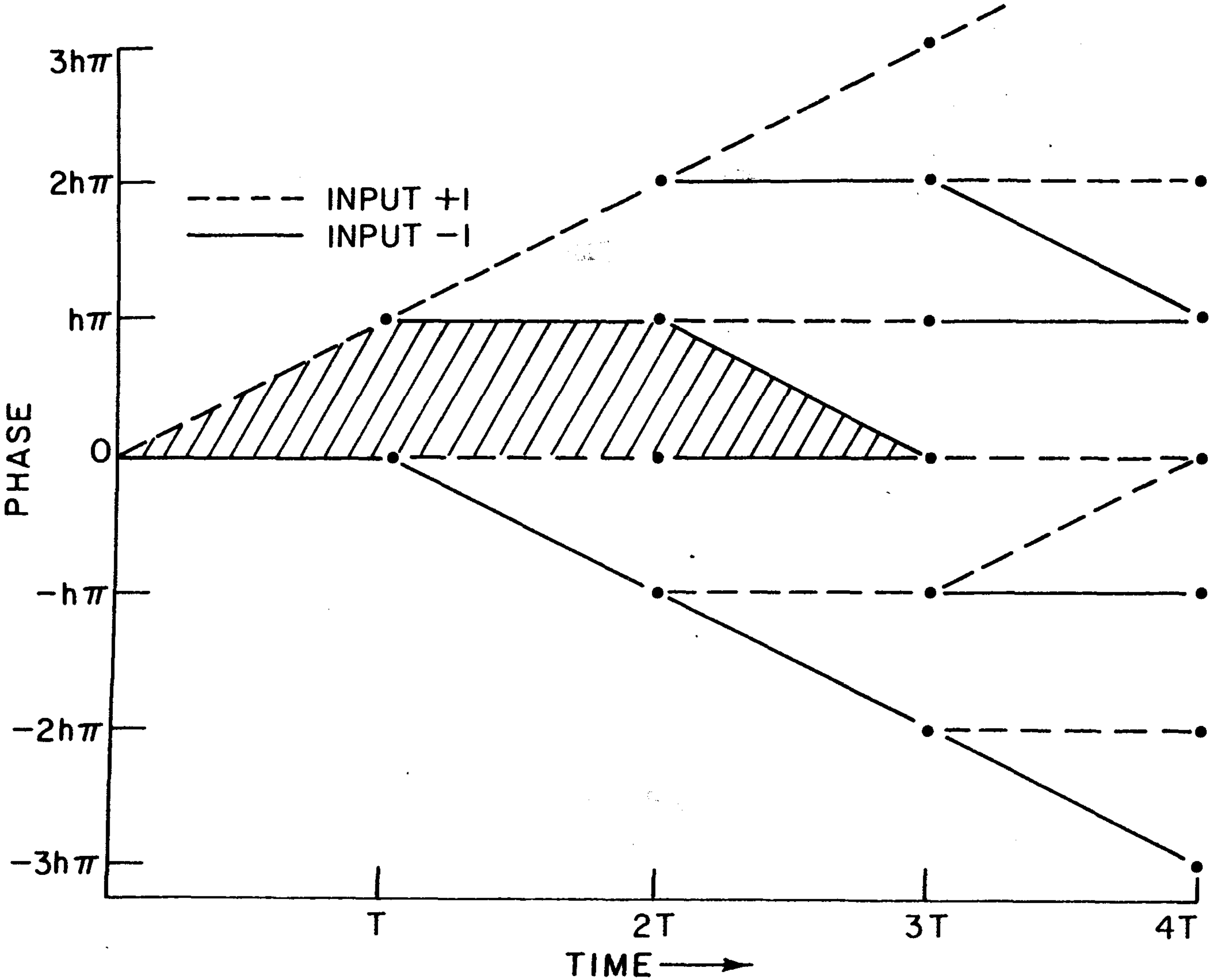

FIGURE 3(a) Phase tree for modulation [(1+D)/2,h], binary input. Bit history is (+1). Minimum distance phase path pair is shaded.

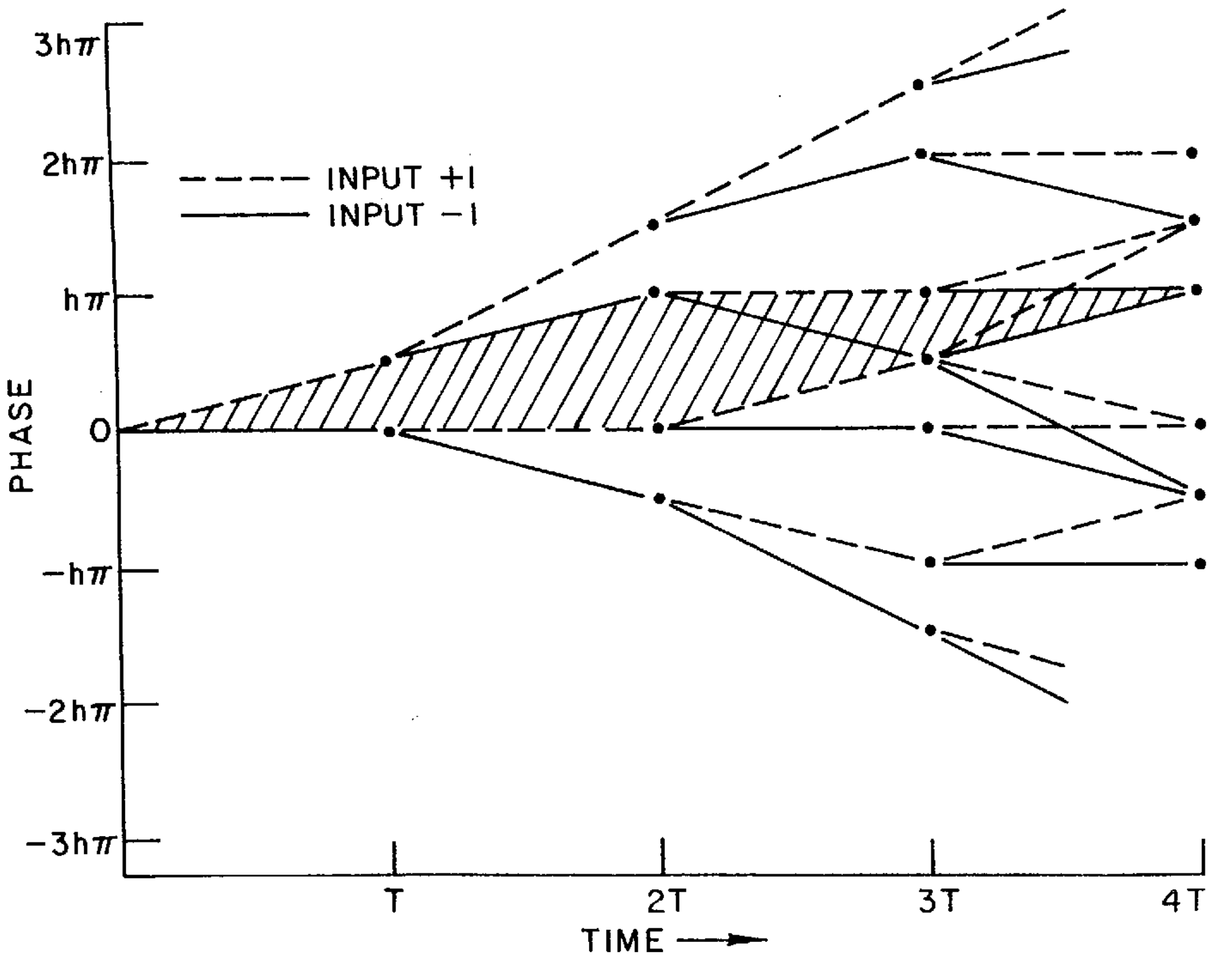

FIGURE 3(b) Phase tree for modulation $[(1+2D+D^2)/4, h]$, binary input.
Bit history is (-1, +1). Minimum distance phase path pair is shaded.

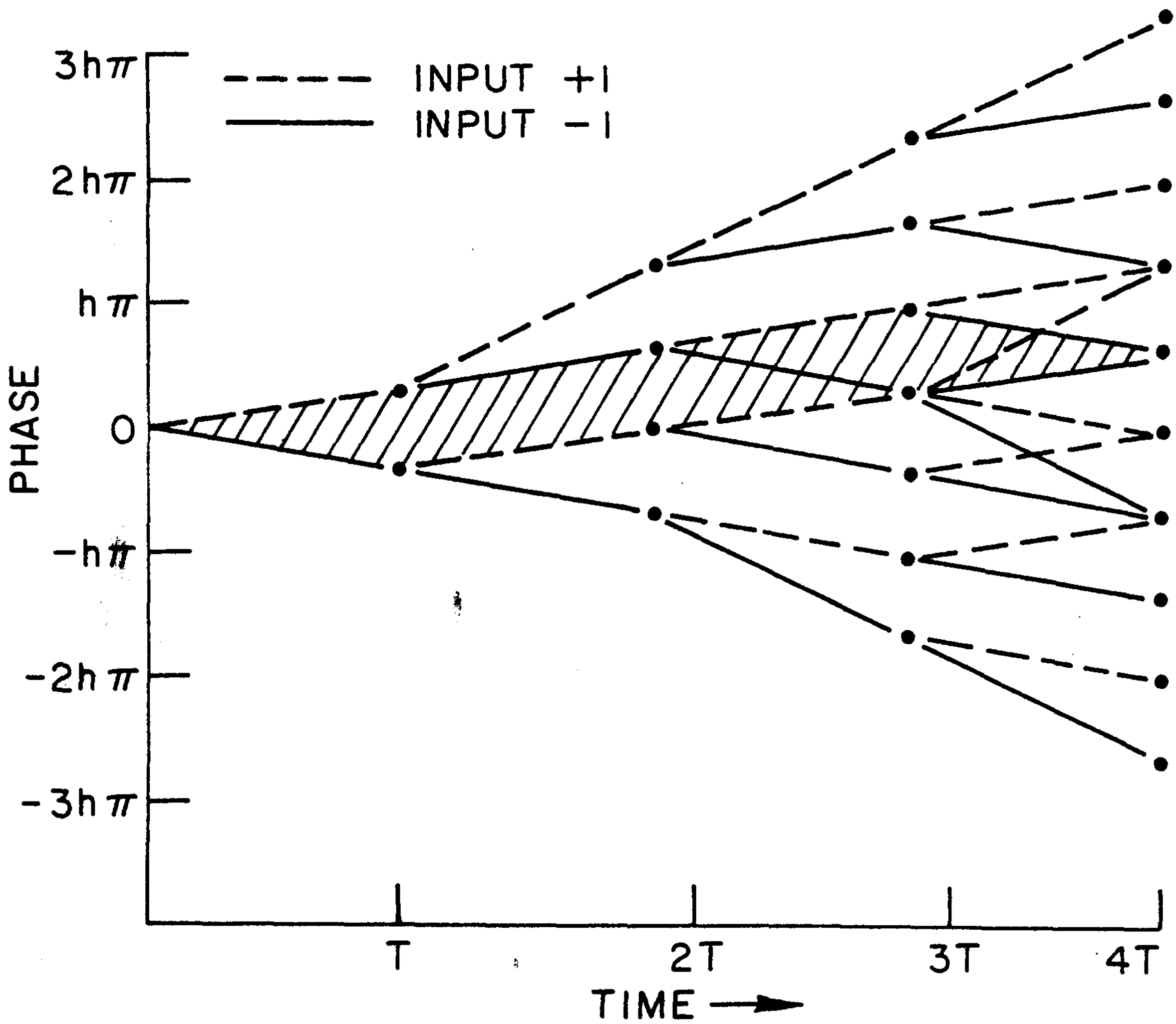

FIGURE 3(c) Phase tree for modulation $[(1+D^2+D^3)/3, h]$, binary input.
Bit history is (-1, +1). Minimum distance phase path pair is shaded.

The above expression then reduces (with double frequency terms disappearing) to

$$d_{ij}^2 = \int_I A^2\left[1-\cos(\theta_i(t) - \theta_j(t))\right]\, dt$$

Under the same assumptions, the signal energy over any interval of length T is simply $E=A^2T/2$. Normalizing and writing $\theta_i(t) - \theta_j(t) = \Delta\theta_{ij}(t)$ gives

$$\frac{d_{ij}^2}{2E} = \frac{1}{T}\int_I \left[1-\cos\Delta\theta_{ij}(t)\right]\, dt \qquad (13)$$

From this expression, the distances between signals may be easily found by inspection of their paths through the phase tree.

The minimum distance between any two distinct signals is called d_{min}:

$$d_{min} = \min_{\forall(i,j)} d_{ij}^{\infty}$$

where

$$d_{ij}^N = \int_0^{NT} \left[y_i(t) - y_j(t)\right]^2\, dt \qquad (14)$$

The minimum distance phase path pair is depicted by shading for the PRS modulations shown in Figures 3 (a), (b), (c). The usual basis against which to compare minimum distances is MSK. For MSK, $d^2 min/2E = 2$.

The ensemble of possible phase paths (or any subset) defines a signal space with distances between signal points given by Equation 13. The non-linear modulations of interest here have a peculiar property. The distance between any signal y_i and the nearest signal whose phase path diverges from and remerges with y_i depends on the actual path of y_i. It is not always possible to find a signal path d_{min} in distance from any given path. In general, if a signal shows maximum phase change over consecutive intervals, it will not have a d_{min} closest neighbour, and if it shows minimum phase change, it will always have a d_{min} neighbour.

1.3.3 State structure

As previously mentioned, it is possible to model constant envelope modulations as first order Markov processes.

The expression for information carrying phase was

$$\theta(t) = \sum_{k=0}^{\infty} a_k g(t-kT)$$

Here it is assumed that $g(t) = 0$ for $t<0$ and $g(t)=g(LT)$ for $t>LT$. Following Aulin et al. [14], the phase during digit interval n can be written as

$$\theta(t) = \sum_{k=0}^{n-L} a_k g(LT) + \sum_{k=n-L+1}^{n} a_k g(t-kT) \qquad (15)$$

$$nT<t<(n+1)T$$

The first term represents the underlying phase due to past inputs, which can be called the phase state,

$$\phi_n = \left[\sum_{k=0}^{n-L} a_k g(LT) \right]_{\text{mod } 2\pi} \tag{16}$$

The second term represents the contribution of inputs actively affecting the shape of the phase path during interval n. A correlative state vector is defined as $A_n \triangleq (a_{n-1}, a_{n-2}, \ldots, a_{n-L+1})$. The phase state, correlative state, and present input a_n completely determine the signal during interval n. A combined state $S_n \triangleq (\phi_n, A_n)$ can be defined for use in the Markov description. The total number of correlative states equals $M^{(L-1)}$. If the total number of underlying phases is equal to p, the total number of states equal $p \cdot M^{(L-1)}$. By definition,

$$\phi_n = \left[g(LT) \sum_{k=0}^{n-L} a_k \right] \text{mod } 2\pi$$

The sum

$$\sum_{k=0}^{n-L} a_k$$

can take on many integer values so that if $g(LT) = 2\pi\ell/p$, for least integers ℓ and p, where ℓ and p are relatively prime, there are a total of p possible distinct phase states, $(0,\ 2\pi/p,\ 4\pi/p,\ \ldots.,\ 2\pi(p-1)/p)$.

This does not necessarily mean there are p possible phase states at any given time nT. We notice that

$$\phi_{n+2} - \phi_n = g(LT)\cdot(a_{n+2-L} + a_{n+1-L})$$

Since a_j can assume values $\{\pm 1, \pm 3, \ldots, \pm(M-1)\}$, then

$$(\phi_{n+2} - \phi_n)\,\varepsilon\,\{-2(M-1)g(LT), \ldots, -2g(LT), \ldots, 2(M-1)g(LT)\} \qquad (17)$$

We now wish to express g(LT) in the form $g(LT) = \pi n/k$ for least integers n, k. From Equation 17, given the initial phase state at time zero, there can be only k phase states possible at the "even" times (n=2, 4, 6,...) and k possible states at the "odd" times (n=1, 3, 5, ...). But $g(LT)=2\pi\ell/p$ implies p total phase states. If p is even therefore, there will be only p/2 of the states possible at the even times, and the other p/2 states at the odd times. On the other hand, if p is odd, there will be all p states possible at <u>every</u> time nT. To relate the number of phase states to modulation index h, recall that $g(LT) = h\pi$.

For example, consider $h = \frac{1}{2}$, $g(LT) = \pi/2$. There are a total of four possible phase states, $(0, \pm\pi/2, \pi)$ but the possible phase states alternate as below

	$\pi/2$	0	$\pi/2$	0	
•••	$-\pi/2$	π	$-\pi/2$	π	•••
	nT	(n+1)T	(n+2)T	(n+3)T	

This means that for $h = n/k$, for least integers n, k, there are $k \cdot M^{(L-1)}$ states per interval in the Markov description.

1.3.4 Viterbi Receiver

Implementation of MLSE using the Viterbi algorithm was discussed earlier. The relevant likelihoods of the signal in contention are calculated recursively using matched filter outputs. The receiver must be able to provide the correlation of the received signal with every possible duration-T signal segment. The correlation over interval $[nT, (n+1)T]$ required for signal y_i is

$$\int_{nT}^{(n+1)T} r(t) \cdot y_i(t)\, dt.$$

If

$$\theta_i(t) = \sum_{k=0}^{\infty} a_{ik} g(t-kT)$$

is the phase of y_i, then from Equation 1

$$\int_{nT}^{(n+1)T} r(t) \cdot y_i(t) dt = \int_{nT}^{(n+1)T} r(t) \cdot [A\cos(2\pi f_c t + \theta_i(t))] dt$$

$$= A \int_{nT}^{(n+1)T} r(t)\cos(2\pi f_c t)\cos\theta_i(t) dt - A \int_{nT}^{(n+1)T} r(t)\sin(2\pi f_c t)\sin\theta_i(t)\, dt$$

$$= A \int_{nT}^{(n+1)T} r_c(t)\cos\theta_i(t)\, dt - A \int_{nT}^{(n+1)T} r_s(t)\, \sin\theta_i(t)\, dt \qquad (18)$$

where $r_c(t) = r(t)\cos 2\pi f_c t$ and $r_s(t)=r(t)\sin 2\pi f_c t$. These are obtained by multiplying the received signal by $\cos 2\pi f_c t$ and $\sin 2\pi f_c t$ to form quadrature channels. It is then necessary to provide a baseband matched filter bank which provides the correlation with the cosine and sine of all possible phase paths $\theta_i(t)$ over each interval. Again, following Aulin et al. [14], during interval n,

$$\begin{aligned}\cos\theta_i(t) &= \cos\left[\phi_{in} + \sum_{k=n-L+1}^{n} a_{ik}g(t-kT)\right] \\ &= \cos\phi_{in}\cos\left(\sum_{k=n-L+1}^{n} a_{ik}g(t-kT)\right) \\ &\quad -\sin\phi_{in}\sin\left(\sum_{k=n-L+1}^{n} a_{ik}g(t-kT)\right)\end{aligned}$$

$$nt \leq t \leq (n+1)T \qquad (19)$$

Similarly,

$$\begin{aligned}\sin\theta_i(t) &= \sin\phi_{in}\cos\left(\sum_{k=n-L+1}^{n} a_{ik}g(t-kT)\right) \\ &\quad + \cos\phi_{in}\sin\left(\sum_{k=n-L+1}^{n} a_{ik}g(t-kT)\right)\end{aligned}$$

There are p possible values of ϕ_{in}, and $\sin\phi_{in}$ and $\cos\phi_{in}$ are interpreted as scaling multipliers. There are M^L possible phase paths

$$\sum_{k=n-L+1}^{n} a_{ik} g(t-kT),$$

but not all are necessarily distinct. There are therefore at most $2 \cdot M^L$ matched filters required per quadrature channel with impulse responses [14]:

$$h_{cj}(t) = \begin{cases} \cos\left(\sum_{\ell=-L+1}^{0} a_{j\ell} g[(1-\ell)(T-t)] \right) & 0<t<T \\ 0 \text{ elsewhere} & \end{cases}$$

$$h_{sj}(t) = \begin{cases} \sin\left(\sum_{\ell=-L+1}^{0} a_{j\ell} g[(1-\ell)(T-t)] \right) & 0<t<T \\ 0 \text{ elsewhere} & \end{cases}$$

As Aulin et al. [14] note, this number can be reduced by a factor of two by noticing that every digit sequence has one with opposite sign. Therefore at most $2 \cdot M^L$ baseband matched filters are needed in all.

The matched filter outputs are sampled at every time nT to provide the correlation for interval [(n-1)T,nT]. A block diagram of an MLSE receiver is shown in Figure 4. The Viterbi algorithm works with S_v states per interval and performs $M \cdot S_v$ additions and $(M-1)S_v$ binary comparisons per digit interval.

The number of states, S_v, and the number of matched filters for the partial response modulations of interest are given in Table 1 (page 70) along with other relevant quantities.

Schonhoff et al. [13] simulated Viterbi detection for CPFSK. In their analysis they assume passband matched filters which result in a fewer number of required filters. Aulin [15] provides an analysis of symbol error probability bounds for Viterbi detection of continuous phase modulated signals. The major point is that at high SNR, as expected, the error probabiltiy is dominated by a term of the form $KQ(d_{min}/2\sigma)$ due to minimum distance error events. Aulin notes that a minimum distance error event is not necessarily possible with all transmitted sequences. At high SNR, the error <u>event</u> probability may then be less than $Q(d_{min}/2\sigma)$.

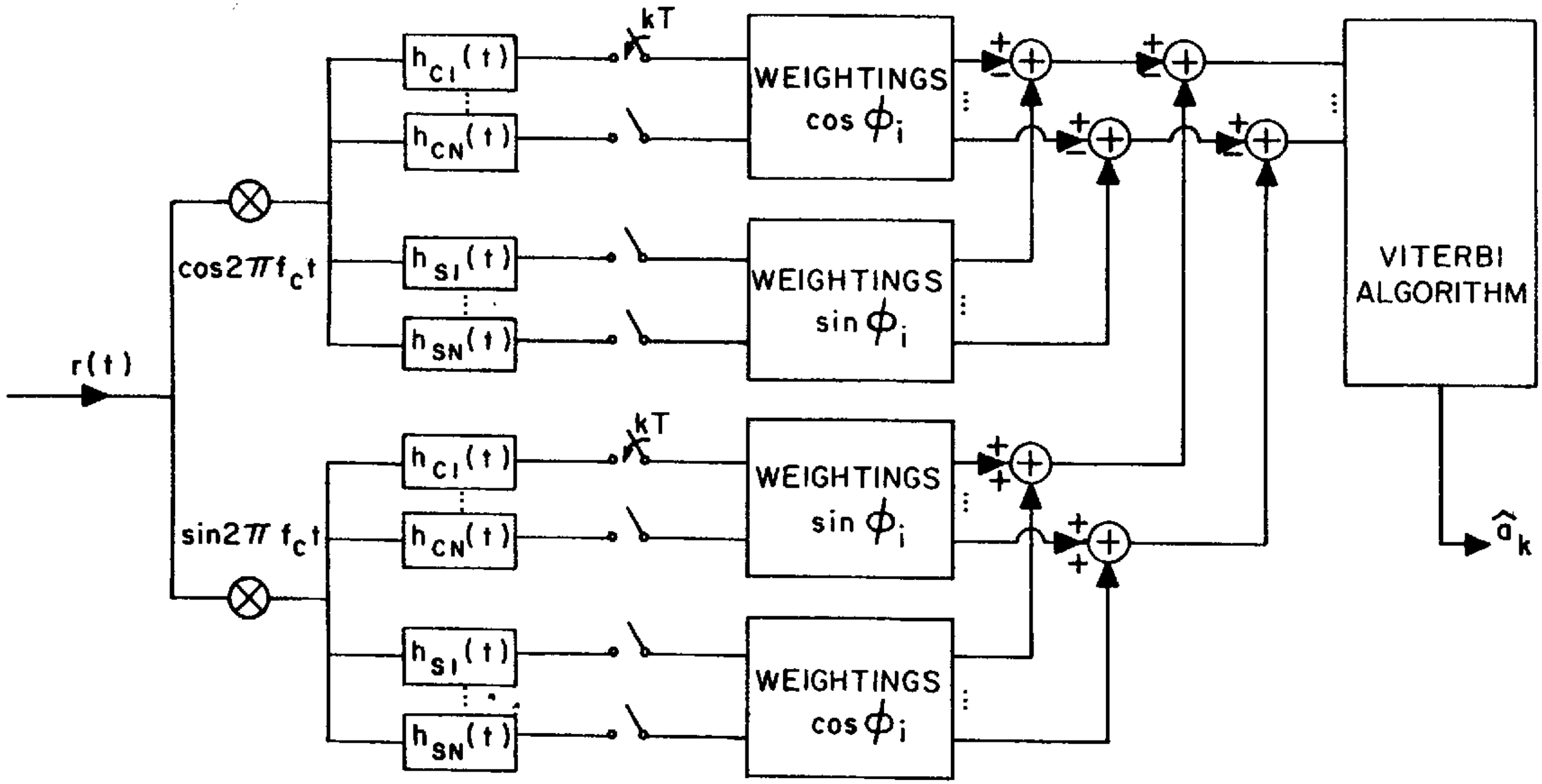

FIGURE 4 MLSE receiver block diagram.

CHAPTER 2

ALTERNATIVE DECODING APPROACH

The MLSE algorithm by definition identifies the signal (and corresponding sequence) most likely to have been the one transmitted. It does this by picking the signal for which, given the observed received signal, the noise looks the smallest. It forces decisions between distinct signals that share a common state. To correctly choose the transmitted signal over some alternate signal d_a away, it is necessary that the component of the noise in the direction of the alternate be less than $d_a/2$. In order that the transmitted signal is correctly chosen most of the time, the variance of the noise component in the direction of the closest alternates must be small compared to these closest separation distances. The smallest of these separation distances is d_{min}. Since the noise is white, it has equal variance in all directions in the signal space. If y_t is the transmitted signal, then the projection of $[\bar{r}-\bar{y}_t]_0^N$ in any given direction from y_t in the signal space should be less than $d_{min}/2$ most of the time. Here $\bar{r}$ represents the received signal vector as usual.

The MLSE algorithm is constrained to maintaining S_v survivors (one per state) at every time NT in order that the ML signal never be rejected. Consider the case when many of these survivor separation distances are necessarily large compared to d_{min}. Many of the survivor signals must be far from the received signal and have low likelihoods of being the transmitted signal. We should expect to be able to discard these survivors without seriously affecting the overall probability of rejecting the transmitted signal. We might test to see if one such survivor y_i is unlikely by examining the projection of $[\bar{r} - \bar{y}_i]_0^N$ in several directions. It makes sense to test the projections in the directions of other signals y_j; $[\bar{r} - \bar{y}_i]_{\mathcal{S}_{ij}^N}^N$ defines the projection of $[\bar{r}-\bar{y}_i]_0^N$ onto the subspace $\mathcal{S}_{ij}^N \triangleq [\bar{y}_j-\bar{y}_i]_0^N$.

Note that these projections are signed scalar quantities.

If these projections appreciably exceed $d_{min}/2$, y_i can be branded as unlikely. A general test for identifying those signals unlikely to be the transmitted signal is then, for each signal y_i:

$$\left|[\bar{r}-\bar{y}_i]_{\mathcal{S}_{ij}^N}^N\right| > R \text{ for any other } \bar{y}_j \text{ ?}$$

where R should be around $d_{min}/2$ to achieve close to the same probability of rejecting the transmitted signal as MLSE. Here $|\cdot|$ denotes absolute value.

A sequential[1] rule could be structured around this test. Such a rule has been presented by Vermuellen for dealing with intersymbol interference with PAM signalling [18]. The degree to which such a rule correctly identifies a signal as still likely to have been the transmitted one depends on the size of parameter R. By testing the projections in the directions of other signals, the rule forces survivors of the testing to be all pairwise less than 2R in separation. When many survivors of MLSE are separated by more than d_{min}, and we choose $R \sim d_{min}/2$, the mechanism for reduction of the survivor list becomes clear.

[1]It should be pointed out that 'sequential' is used here to describe any decoder that operates on sequences. The rules developed here allow no "backtracking" and bear little resemblance to the so called sequential decoder first proposed by Fano [19].

The rule development in this chapter is based on this hyperplane approach. A simple union bound on the probability of rejecting the transmitted signal is derived.

2.1 The Sequential Rule

To state Vermuellen's hyperplane approach:

> If for any j, $\left|[\bar{r}-\bar{y}_i]^N_{\mathcal{S}^N_{ij}}\right| > R$
>
> Then reject y_i at time NT
> where the y_j are all contenders at time NT.
> Extend all survivors of this test into all possible descendants at time (N+1)T for retesting.

This rule will be modified for our purposes. Instead of only considering the magnitude of the projection, its sign will be used as well (see Figure 5). If point P lies in the direction of $\bar{y}_j$ from $\bar{y}_i$, projection p_i will be positive; otherwise p_i will be negative. This modified rule will be adopted as basic sequential rule R1.

Rule R1:

> If for any j, $[\bar{r}-\bar{y}_i]^N_{\mathcal{S}^N_{ij}} > R$
> then REJECT y_i at time NT
> where the y_j are all contenders at time NT.
> Extend the survivors into all descendants as contenders to be tested at time (N+1)T.

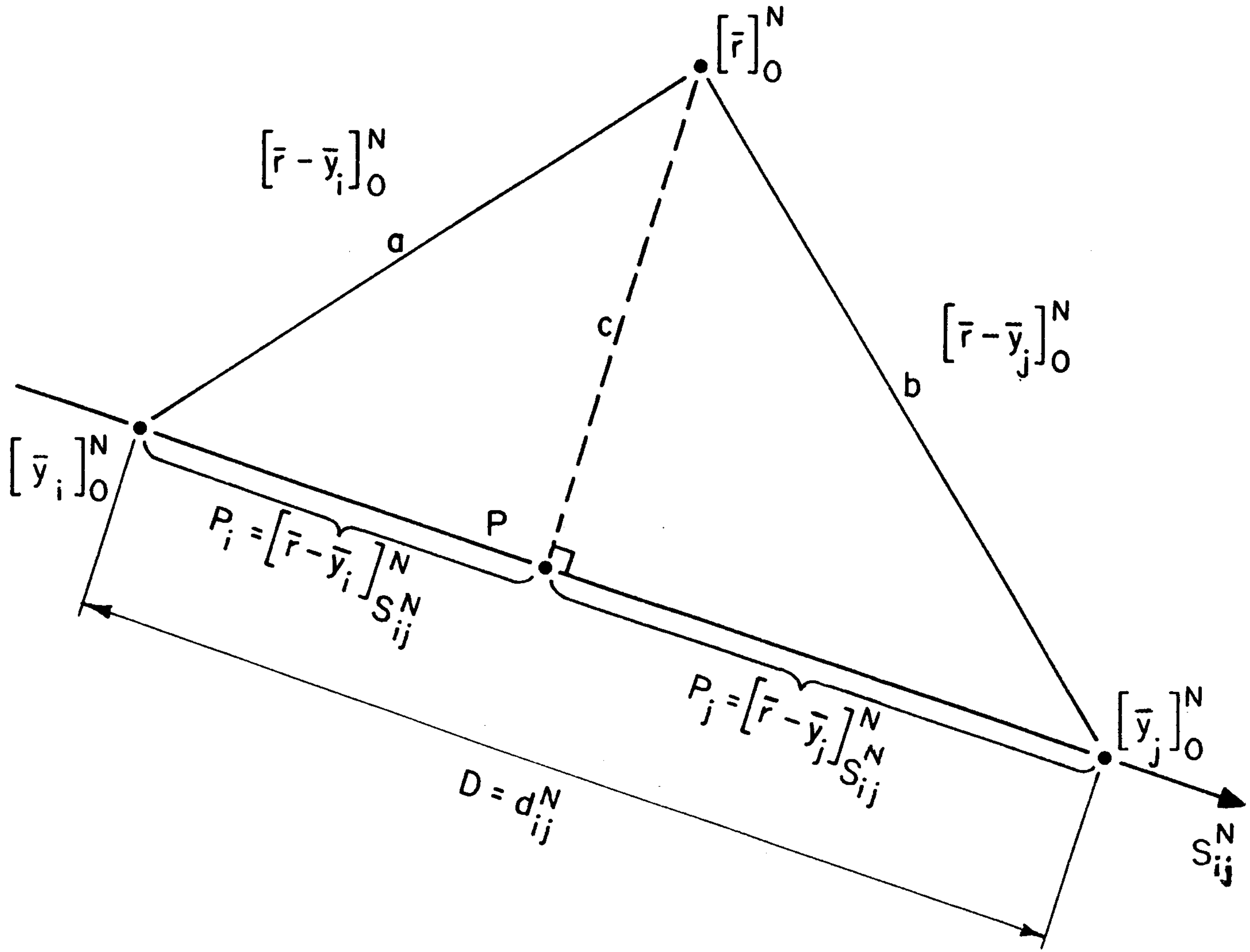

FIGURE 5 Projection operation geometry.

The decision regions are shown in Figure 6. Under this rule the transmitted signal cannot be rejected unless a noise component exceeds R in the direction of any of the other contenders. If a bound can be placed on the number of contenders, an upper bound on the probability of rejecting the transmitted signal can be produced. If R is chosen as $d_{min}/2$, then this bound is of the form $KQ(d_{min}/2\sigma)$, the same as the dominant term for MLSE at high SNR.

The other factor contributing to overall error rate is the length of error events under the rule. With MLSE, descendants of all possible states are in contention at every time NT. The transmitted signal path can only be rejected in favour of one that shares the same state. In effect, once a divergence from the transmitted path has occurred future sections of the actual transmitted path are immediately back in contention. This is not the case with the proposed alternate rule. The mechanism by which the decoded signal path (or state sequence) links back up with the transmitted signal path (or state sequence), once a divergence from the true path has occurred, is not immediately clear.

2.1.1 General Observations and Merging Property

In order to effect an unambiguous decoding using rule R1, we require that all survivors share a common history (merge) within a finite time before NT. As mentioned, all survivors must be pairwise less than 2R in separation. Define the distance developed between signals that have diverged from a common history for k time units as δ_{ij}^{k}, and the minimum of all these distances as

$$\zeta(k) = \min_{\forall(y_i, y_j)} (\delta_{ij}^{k})$$

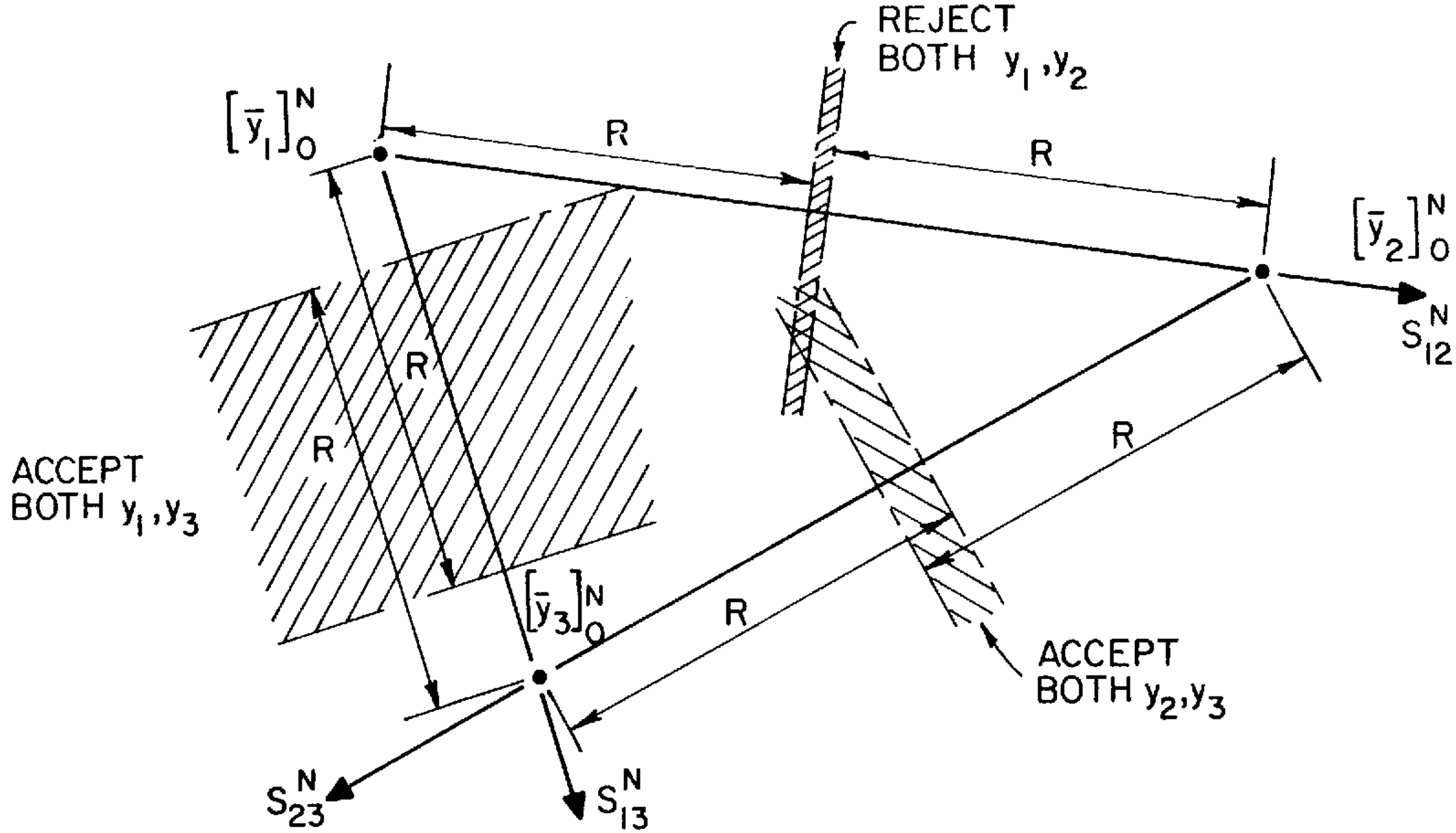

FIGURE 6 Decision regions for hyperplane rule R1.

Denote by k_{2R}, the maximum k such that $\zeta(k) \leq 2R$. We can then say that all survivors must have shared a common history no earlier than $k_{2R}T$ seconds prior to NT. A problem then arises for unbounded k_{2R}. Specifically, for $R > d_{min}/2$, k_{2R} is unbounded and it is possible to retain as separate survivors two signals whose state sequences become identical again after NT. To handle this event, extra measures would have to be introduced to force an optimal decision between such a pair.

Given high enough R, we expect the transmitted signal to survive the rule's testing most of the time. Eventually however, with probability one, the transmitted signal will be rejected. We note that:

(a) For every survivor given this rejection, we can find descendants that merge into the transmitted state sequence within a finite number of (usually few) intervals.

(b) For each subsequent pairwise test of the survivor descendants, when both do not survive, the one that does survive must be closer to the received signal to that time, i.e. it has the greater likelihood.

(c) Those survivor descendants that are "headed back" towards the transmitted signal path (in terms of state sequence), or are "paralleling" it, will generally be closer to the transmitted signal than those that are "headed away". Given reasonably high SNR, these descendants will almost surely have greater likelihood than those that are "headed away", and should be the survivors of the future tests.

These three points suggest that the signals retained by the rule would tend to link back up with the transmitted signal after only a few intervals. At lower SNR, once an error event has started, the rule is less likely to quickly zero back in on the transmitted signal path due to local noise fluctuations. In this case, the expected length of error events is greater.

Vermuellen [14] notes that there is a finite probability that all signals will be rejected under a hyperplane approach. In this situation, the practical modification is to choose the signal with greatest likelihood as sole survivor to time NT.

2.1.2 Bound On Probability Of Transmitted Signal Rejection

A simple argument may be used to bound the probability of rejecting the transmitted signal y_a before time LT under rule R1:

> If for any j, $[\bar{r}-\bar{y}_i]^N_{\mathcal{D}^N_{ij}} > R$
> then reject y_i at time NT
> where the y_j are all contenders at time NT

The number of contenders (other than y_a) at time NT is denoted by η. The actual signal is assumed to be in contention at t=0. We need to define several events:

Let: ε^j_N be the event that $[\bar{r}-\bar{y}_a]^N_{\mathcal{D}^N_{aj}} > R$ for contender $[y_j]^N_O$

$$\varepsilon_N \triangleq (\varepsilon^1_N \cup \varepsilon^2_N \cup \varepsilon^3_N \cup \ldots \cup \varepsilon^\eta_N);$$ i.e. the event for which $[y_a]^N_O$ would fail at least one of the tests

$$\rho \triangleq \varepsilon_1 \cup \varepsilon_2 \cup \varepsilon_3 \cup \ldots \cup \varepsilon_L$$

Let $\mathcal{R}$ be the event that y_a is rejected on or before time LT. Let us denote event probability by $P(\cdot)$,

$$P(\varepsilon_N) = P(\varepsilon_N^1 \cup \varepsilon_N^2 \cup \ldots \cup \varepsilon_N^{\eta})$$

A union bound on $P(\varepsilon_N)$ gives

$$P(\varepsilon_N) < P(\varepsilon_N^1) + P(\varepsilon_N^2) + \ldots + P(\varepsilon_N^{\eta}) \qquad (20)$$

These basic event probabilities are identical, and given by

$$P(\varepsilon_N^k) = P([\bar{r}-\bar{y}_a]^N_{\mathcal{S}^N_{ak}} \geq R) = Q(R/\sigma)$$

where $\sigma^2 = N_o/2$ is the variance of the independent noise vector components. The union bound gives $P(\varepsilon_N) < \eta Q(R/\sigma)$. If the number of contenders at any time is bounded by γ, then independent of the actual time NT,

$$P(\varepsilon_N) < \gamma Q(R/\sigma)$$

Event $\mathcal{R}$ is completely contained in event ρ, so $P(\mathcal{R}) < P(\rho)$.

$$P(\mathcal{R}) < P(\varepsilon_1 \cup \varepsilon_2 \cup \ldots \cup \varepsilon_L) \qquad (21)$$

Using another union bound,

$$P(\mathcal{R}) < P(\varepsilon_1) + P(\varepsilon_2) + \ldots + P(\varepsilon_L)$$

But $P(\varepsilon_i) < \gamma Q(R/\sigma)$ so that

$$P(\mathcal{R}) < L \cdot \gamma \cdot Q(R/\sigma) = KQ(R/\sigma) \qquad (22)$$

The bound on the probability of first rejecting the transmitted signal per unit time, P(e), is then $\gamma Q(R/\sigma)$, where γ bounds the number of contenders.

2.1.3 Expected Symbol Error Performance

As long as $R<d_{min}/2$, no extra measures would be needed to force decisions between unmerged survivors (see Section 2.1.1). In this case the bound of Section 2.1.2 is a bound on the probability of an error event starting at any time, given that the transmitted signal is initially in contention. Once rejection occurs, we expect the survivor descendants to link back up with the transmitted signal path after a short time. Once this merge of survivors has occurred, the noise during the common history portion of the contenders is irrelevant to the new projections tested under the rule. We can then apply the bound to the probability of starting another error event. As long as the total fraction of error event time is small, the bound of $\gamma Q(R/\sigma)$ is still accurate.

To get the average probability of symbol error, we need to multiply $\gamma Q(R/\sigma)$ by the expected number of errors per error event, K_ε. This gives

$$P(e) < K_\varepsilon \gamma Q(R/\sigma) = KQ(R/\sigma)$$

If $R>dmin/2$, P(e) will be dominated by $KQ(dmin/2\sigma)$ at high SNR, i.e. the probability of error in a forced decision between unmerged survivors of smallest separation.

Little may be said analytically about the length of error events, but we still expect their average length to be only a few digit intervals at reasonably high SNR. Simulation was used to verify this conjecture and the results are presented in Chapter 3.

2.1.4 Bound on The Maximum Number of Survivors - Choice of Parameter R

All survivors under rule R1 must be pairwise less then 2R in separation. This fact may be used to bound the maximum number of survivors at any time. The distance δ_{ij}^{k} between any two signals distinct for k intervals depends on the particular modulation. For the constant envelope modulations of interest, the distance depends on the difference in phase paths as discussed in Section 1.3.2. Using Equation 13, a bound or the number of signals all pairwise less than 2R in separation may be found by inspection of the phase tree.

For example, with binary input and modulation [(1+D)/2, h=1/2] (Figure 2(a)), the maximum number of survivors for $R=d_{min}/2$ is two. The maximum number of survivors will be a step-like function in the parameter R, rapidly increasing for R in excess of $d_{min}/2$.

The greater R is chosen, the less the probability of rejection of the transmitted signal. As mentioned, if R exceeds $d_{min}/2$, extra measures will be needed to force decisions between signals whose state sequences have remerged. In the limit as $R\rightarrow\infty$, all signals would survive the rule's testing and these extra measures would be equivalent to the Viterbi algorithm. Higher R will also generally mean a higher number of survivors and increased complexity. Choosing R in excess of $d_{min}/2$ does have an advantage. If R is much in excess of $d_{min}/2$, at higher SNR most of the rejections of the transmitted signal will be caused by forced decisions between d_{min} neighbours. In this case, most of the error events will be the same as those occurring in MLSE and the performance will be closer to MLSE. Choosing lower R will lead to more rejections of the transmitted signal for which the error event lengths could be longer.

If R is chosen less than $d_{min}/2$, the error exponent is adversely affected, although the complexity may be reduced. In addition fewer survivors should cause the link-up mechanism to be less efficient leading to longer error events. The choice of R is therefore a trade off between complexity and performance. The choice $R = d_{min}/2$ is of special interest as this defines the minimum complexity for which asymptotic optimality can be achieved.

2.2 Implementation and Complexity

The rule requires finding the lengths of projections of relevant vectors on the subspaces defined by every pair of contenders. Consider the three vectors involved in any pairwise test, the received vector $[\bar{r}]_0^N$ and the two contenders vectors, $[\bar{y}_i]_0^N$ and $[\bar{y}_j]_0^N$. These three points are contained in a single plane as shown in Figure 5. The projection operation is equivalent to dropping a perpendicular from $[\bar{r}]_0^N$ to the line passing through $[\bar{y}_i]_0^N$ and $[\bar{y}_j]_0^N$. This line is simply $\mathcal{S}_{ij}^N$. The point of projection is called P.

If y_i is under test, $p_i = [\bar{r}-\bar{y}_i]_{\mathcal{S}_{ij}^N}^N$ is of interest and, similarly, p_j is of interest for y_j. Simple geometry gives

$$p_i = \frac{a^2-b^2+D^2}{2D} \tag{23}$$

and

$$p_j = \frac{b^2-a^2+D^2}{2D}$$

where

$$a^2 = ||[\overline{r}-\overline{y}_i]_O^N||^2,\ b^2=||[\overline{r}-\overline{y}_j]_O^N||^2,\ \text{and } D=d_{ij}^N$$

It can be shown that these relations hold regardless of the relative positions of the three vectors as long as the projections p_i, p_j are allowed to have sign as well as magnitude. As mentioned earlier, if point P lies in the direction of $\overline{y}_j$ from $\overline{y}_i$, then p_i is considered positive, otherwise it is negative.

The testing of the projections is $p_i > R?$, $p_j > R?$. These become for y_i, $(a^2-b^2) > 2DR-D^2?$, and for y_j, $(b^2-a^2) > 2DR-D^2?$. Vermuellen [14] provides the equivalent relationships in terms of the squares of these quantities. Dropping the qualifiers $[\cdot]_O^N$ for convenience,

$$a^2=(\overline{r}-\overline{y}_i)\cdot(\overline{r}-\overline{y}_i)$$

$$= ||\overline{r}||^2+||\overline{y}_i||^2 - 2\overline{r}\cdot\overline{y}_i \qquad (24)$$

Similarily,

$$b^2=||\overline{r}||^2+||y_j||^2 - 2\overline{r}\cdot\overline{y}_j$$

so that

$$a^2 - b^2 = 2(\overline{r}\cdot\overline{y}_j - \overline{r}\cdot\overline{y}_i)$$

Now $L_i(N) = \overline{r}\cdot\overline{y}_i$ is simply the correlation of the received signal and signal y_i to time NT. Recall Eq. 11,

$$L_i(N) = \sum_{\ell=1}^{N} \int_{(\ell-1)T}^{\ell T} r(t)\cdot y_i(t)\, dt$$

This is simply the sum of the appropriate matched filter outputs obtained in exactly the same way as described for MLSE in Chapter 1. In any given comparison we need only subtract the accumulated likelihood (correlation) and compare to a pre-computed quantity, $(2DR-D^2)/2$. The flow chart for implementation of this rule is given in Figure 7.

Define S_R as the bound on the maximum number of survivors. With S_R survivors, there can be up to $C_R=M\cdot S_R$ contenders. If all tests are performed, this means up to $\binom{C_R}{2}=C_R\cdot(C_R-1)/2$ tests. Each test involves one subtraction, one complementation, and two binary comparisons (1A, 1C, 2B.C.). In addition there are C_R additions needed to find the metrics (likelihoods). Even with a low number of survivors, say four, with binary input there are eight contenders, and $\binom{8}{2} = 28$ pairwise tests. Total calculation per cycle is then (36A, 28C, 56 B.C). This far exceeds the complexity of full Viterbi decoding for the modulation $[(1+2D+D^2)/4,\ h=\frac{1}{2}]$. This modulation has a maximum of four survivors with $R=d_{min}/2$. Although the survivor list has been reduced from Viterbi decoding, the number of operations has actually increased. In addition if $R>d_{min}/2$ extra complexity is introduced by choosing between any unmerged survivors. (see Section 2.1.1).

Even considering the computations needed for the average number of survivors, there is no gain over the Viterbi algorithm. In addition, accomodating a flexible number of survivors must introduce extra overhead in the final programming code. Clearly, simpler algorithms are needed.

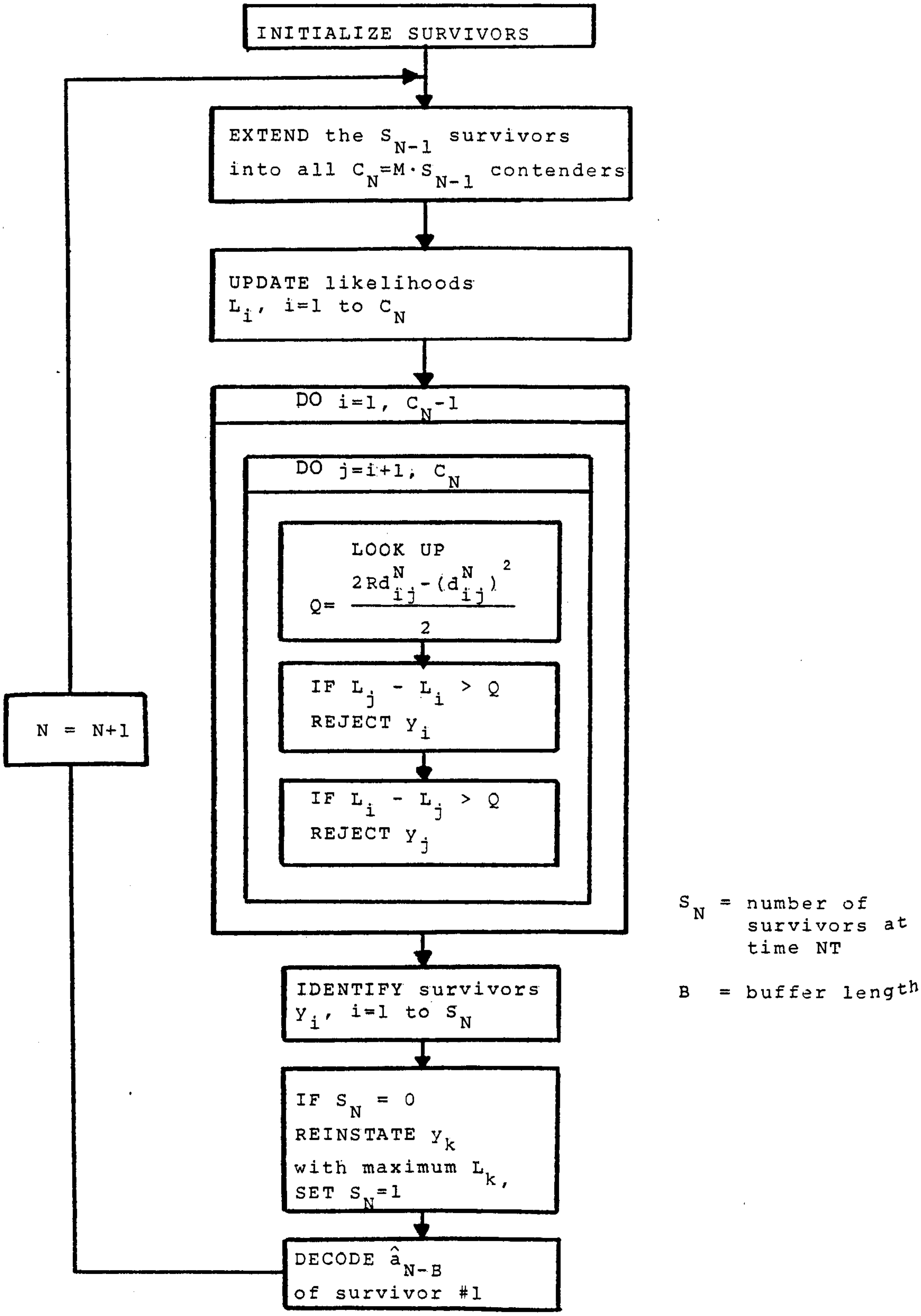

Figure 7 Rule R1 implementation flow chart

2.3 Rule Modifications

The following series of changes to the basic rule R1 lead to an ultimate rule that is much simpler.

During any interval of testing, there will generally be signals separated by more than 2R. In a test involving such a pair, at least one must be rejected. Suppose that instead of parameter R, for such a test we use a revised parameter $R^*=d_{ij}^N/2$, where d_{ij}^N is simply the distance separating the signals.

MODIFICATION #1:

> In any pairwise test involving signals y_i, y_j with $d_{ij}^N > 2R$, replace R by $R^*=d_{ij}^N/2$.

Now one signal must still be rejected, but there is no longer the possibility of losing both. This should enhance the reliability of the rule. When the transmitted signal is involved in such a test, the larger effective R should decrease the probability of rejection. This modification does not increase the maximum number of survivors as other tests will still use parameter R when the signal pair separation is less than 2R.

This simple change actually decreases the complexity of many of the tests. The test value against which the likelihood difference is compared, $(2d_{ij}^N R-(d_{ij}^N)^2)/2$, becomes $(2d_{ij}^N\cdot(d_{ij}^N/2)-(d_{ij}^N)^2)=0$ when $d_{ij}^N > 2R$. The tests become for y_i

$$L_j(N) - L_i(N) > 0 \ ? \qquad (25)$$

and for y_j

$$L_i(N) - L_j(N) > 0 \ ?$$

This is simply selecting the signal segment with greater likelihood to that time.

MODIFICATION #2

> As soon as $[y_k]_0^N$ fails any pairwise test, drop $[y_k]_0^N$ from any subsequent tests of remaining signals.

Now the survivors at the end of testing may depend on the order in which the testing is done. However, the modified rule must still retain all those survivors kept by rule R1 (it can't possibly reject any signals that R1 doesn't reject). In addition, it is no longer possible to reject all signals, and any extra provisons for such an event are unnecessary. Although this should reduce the number of tests on any contender group, the number of survivors with this change may be greater than that for rule R1, and the savings in computation is not immediately clear. Given that this modification is adopted, further steps can be taken to reduce the number of computations.

In any test involving signals separated by more than 2R, only one can possibly survive. If we arrange to perform these tests first, given modification #2 a more rapid reduction in the contender list size should be possible. A core of no more than S_R contenders, with pairwise separations all less than 2R, would rapidly be reached. Subsequent tests on this core could identify the survivors. Alternatively, it seems easier to accept all of these contenders as survivors. There would be at most S_R survivors and $M \cdot S_R$ contenders every interval, but the actual numbers would still depend on the actual noise and transmitted signal path.

Suppose instead of processing until all remaining signals are less than 2R apart, that we test only until they are S_R signals remaining, and accept all as survivors. Clearly, the rule's performance cannot be hurt by stopping the rejection process early.

MODIFICATION #3

> Process those signal pairs with separations greater than 2R until there are S_R remaining. Accept all S_R as survivors.

Now all the pairwise tests simply reject the signal with lower likelihood. There will be a fixed number of survivors, S_R, and a fixed number of contenders, $M \cdot S_R$, every interval. The fixed structure provided by this approach is a great advantage. Still, unless a consistently ordered approach is possible, the extra manipulations involved in selecting the pairs with separation greater than 2R may offset any gains.

Fortunately, for the bandwidth efficient modulations of interest, a consistently ordered approach is possible. This ordering allows a reduction to S survivors, where $S > S_R$.

2.4 Rules With Structured Processing Order

2.4.1 Ordering Approach and Implementation

The modifications of Section 2.3 are best incorporated by ordering the contender signals according to their input digit sequences. The input sequences are ordered in the following way. If we add (M-1) to each digit and divide the result by two, the input digits are mapped into a shifted set (ascending from zero). For example $\{-1,1\}$ maps into $\{0,1\}$. Now this digit sequence is interpreted "base M" with the most recent digit being the least significant digit. The contenders are then ordered sequentially in terms of their "base-M" sequence numbers. It should be pointed out that this ordering is actually very simple and achieved almost implicitly in practice. The modulation must have the following property for this ordering to be exploited:

> All signals with sequence numbers separated by S or more, have distance separations of 2R or more.

This condition is readily met by many of the modulations of interest for $R<d_{min}/2$. Given this property, pairwise testing proceeds from the outside-in on the contender list. The contenders separated by S or more positions in the list will have distance separations greater than 2R. This arrangement can be thought of as a stack with pointers for the top and bottom which identify the pair for likelihood comparison. The top and bottom signals are compared, the signal with the lower likelihood is rejected, and the pointer aligned with it is shifted one position inward on the list to identify a new test pair. This is repeated until there are S survivors.

Specifically, the case $R=d_{min}/2$ is of interest. For many modulations, a convenient relationship exists between the length of the frequency pulse h(t) and a minimum value of S for $R=d_{min}/2$. These modulations have the following properties, where L is the duration of h(t):

(1) The minimum distance, d_{min}, is developed between signals that diverge from a common state and converge again over the shortest possible route, and over an interval of length (L+1)T.

(2) Where these signals converge their sequence numbers are separated by $M^{(L-1)}$.

(3) All signals with sequence numbers separated by more than $M^{(L-1)}$ are separated by more than d_{min} in distance

For these modulations, we choose the number of survivors as $S_L = M^{(L-1)}$.

The most general form of this type of rule follows the same ordered processing and keeps some general number of survivors S. This will be called rule R2.

Rule R2:

> Order contending signals according to their input digit sequence numbers. Always comparing the outside pair, reject the signal with lower likelihood until S survivors remain at time NT. Extend the S survivors into all M•S contenders for retesting at time (N+1)T.

A flow chart for implementation of rule R2 is shown in Figure 8. The following provides an explanation of the terms used in the flow chart:

(I) indicates the position in the contender or survivor list

LKCONT(I) = likelihood of contender I

LKSURV(I) = likelihood of survivor I

$CORRL_N(I)$ = correlation of signal for contender I with received signal for interval N

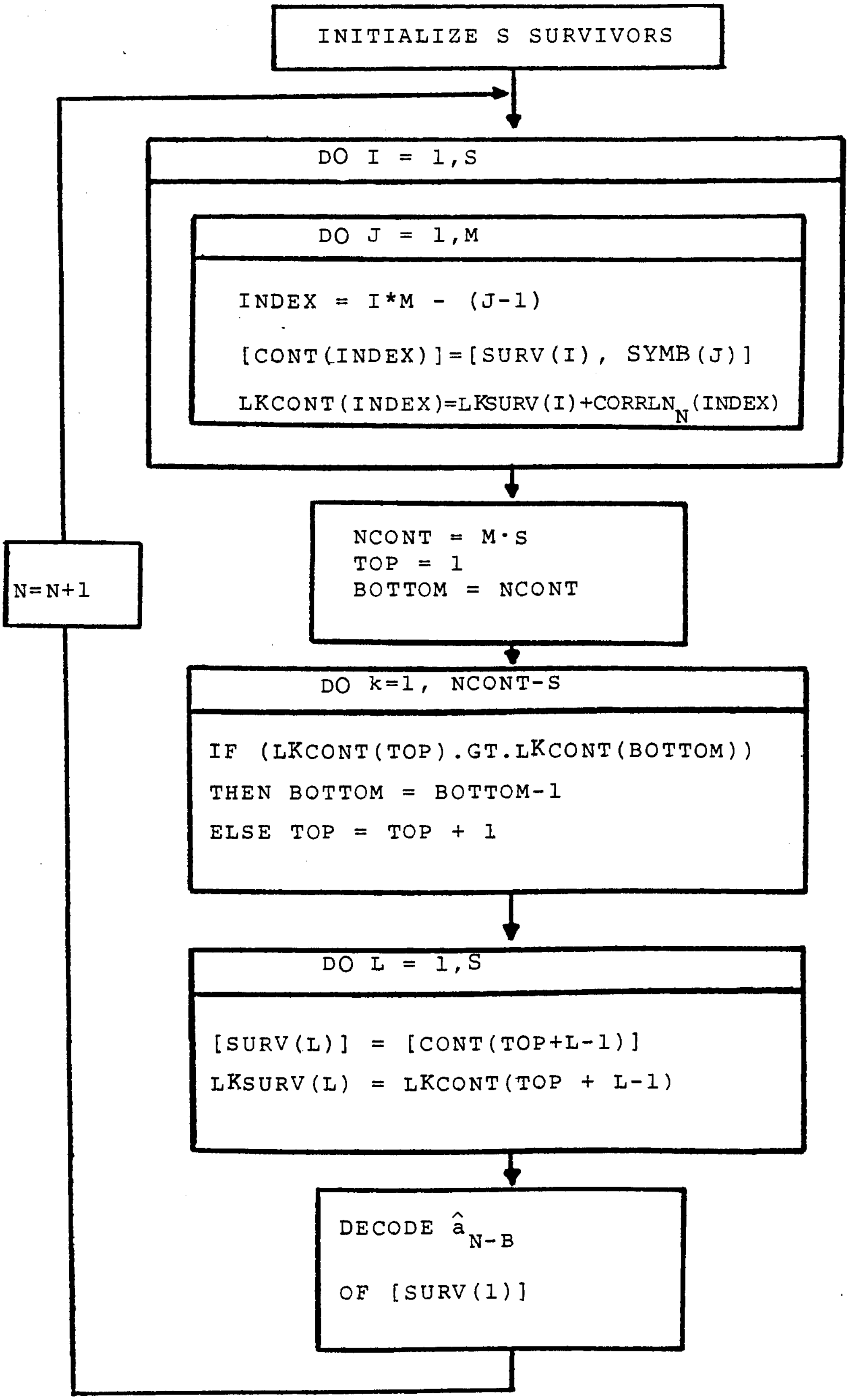

FIGURE 8 Rule R2 implementation flow chart
See text, Section 2 for explanation
of symbols

[CONT(I)] = input digit sequence of contender I

[SURV(I)] = input digit sequence of survivor I

[SURV(I),SYMB(J)] = input digit sequence of survivor I appended with the J'th digit of the input alphabet. Digit -(M-1) is the first digit and +(M-1) is the M'th digit

A typical progression of the rule for S=2 and modulation [(1+D)/2, $h=\frac{1}{2}$], which illustrates the implicit ordering, is shown in Figure 9.

The likelihoods are derived recursively as before from the outputs of filters matched to the appropriate signal sections. Rule R2 requires M•S additions to update these likelihoods and (M-1)•S binary comparisons to reduce the contenders back to S survivors. Since Viterbi decoding with S_V states requires $M \cdot S_V$ additions for likelihoods and $(M-1)S_V$ binary comparisons, the complexity reduction for rule R2 is by a factor of S_V/S. For $S=S_L=M^{(L-1)}$ the complexity reduction factor is k where the modulation index is h=n/k; n and k are smallest integers (see Section 1.3.3).

2.4.2 Performance

The most general form of the rules developed, R2, keeps S survivors at every time NT. If, following the ordered processing, all comparisons involve signals separated by 2R or more, performance should be as good as rule R1 with parameter R. Error event probability should still be overbounded by $KQ(R/\sigma)$ for $R<d_{min}/2$, and by $KQ(d_{min}/2\sigma)$ for $R>d_{min}/2$. In fact due to the higher R* parameter used

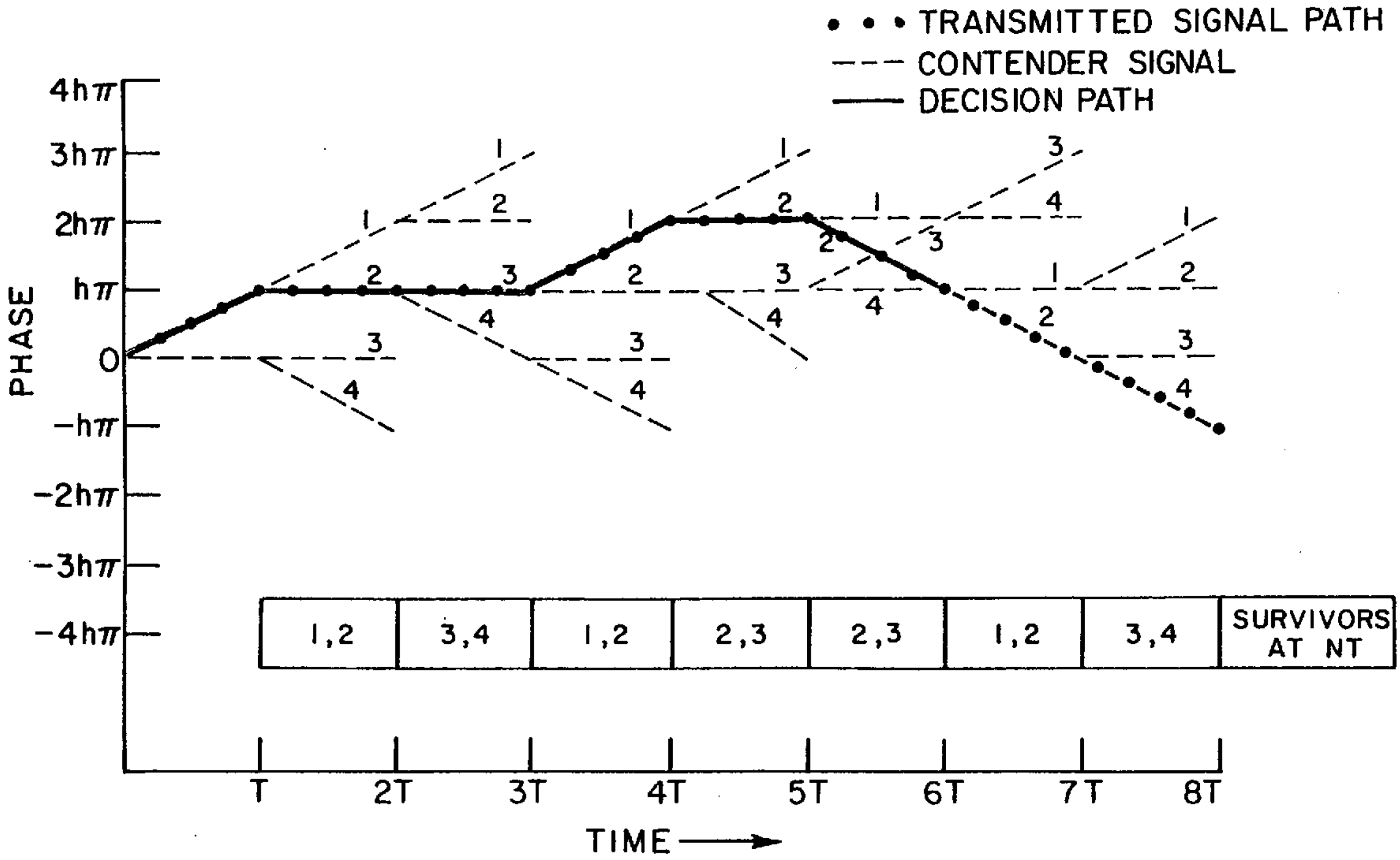

FIGURE 9 Typical progression of rule R2.
Two survivor processing with [(1+D)/2,h] and binary input is shown. Numbers indicate position in the contender list.

in comparisons of signals separated by more than 2R, we expect performance to be even better than rule R1. In addition the larger number of survivors should lead to shorter error events.

Asymptotic performance is governed by the smallest separation between any signals which could come into a likelihood comparison under the rule's ordered processing. The performance at lower SNR and moderate error rates is more complicated. It depends not only on the separation distances involved in the signal comparisons, but also on the corresponding probabilities of these comparisons. Suppose we choose $S_L=M^{(L-1)}$ and apply this rule to a modulation for which not all signal pairs which come into comparisons are separated by d_{min} or more. Clearly asymptotic optimality is not possible. Consider however the case when the probability of the transmitted signal coming into comparison with a signal somewhat less than d_{min} away is small. The overall effect of such a comparison on the error rate can then be very small at moderate error rates. This is exactly the case for modulation $[(1+D+D^2)/3,\ h=\frac{1}{2}]$. Here M=2, and $S_L = 2^{(3-1)}=4$. Signals with the sequence pair (XXX1-1-111,xxx-111-1-1) or equivalently (xxx10011, xxx01100), have base -2 sequence numbers (since their common history xxx) of 19 and 12 respectively, and can come into a comparison. Their separation distance is -.6 dB from d_{min}, however the probability of the transmitted signal taking one of these paths is only 1/16. The overall effect of this one suboptimal pair should be small at error rates around 10^{-3}. This is verified in Chapter 3. This observation means that at moderate error rates, in some cases we can choose S less than that strictly required for asymptotic performance and save on complexity.

Under rule R2, not all survivors are guaranteed to be less than 2R in separation, and the merging mechanism is not obvious. Still, we expect that it would be very improbable that two signals that have been distinct for some time, and therefore have a large separation distance, would be both maintained as survivors under the rule. Inspection of the typical rule progression reveals that for this to occur, the noise would have to assume a large value, placing the received signal about mid way between such signals. This is then a very unlikely event. There is still the question of what happens to signals that have diverged and converged again to share the same state sequence. As previously mentioned such a pair is a problem with rule R1. With rule R2 however, the sequence number separation of such a pair must necessarily increase with time, even though their separation distance may not. Eventually a decision will be forced between these signals and merging is guaranteed in this case. All in all, we should then expect no problem with merging as long as a reasonable buffer length is allowed in the decoder.

The performance of R2 compared to MLSE is of special interest. In this case the number of survivors S may be chosen to meet (or approximate) the performance of R1 with $R=d_{min}/2$. Tracking of the MLSE signal should be superior to that of rule R1. This is due to the higher effective R* parameter implied in comparing likelihoods of signals separated by more than d_{min}. If all such comparisons involve signal separations much in excess of d_{min}, then at high SNR the tracking of the MLSE signal should be almost perfect. This distance structure is typical of the modulations with lower modulation indices.

2.5 Best S Signals Approach

The previous rules have relied on the special relationship between signal separation distances and corresponding input digit sequence number separations. For some modulations, especially at higher modulation indices, this relationship breaks down. In such cases another approach is possible:

> Keep the S signals with highest likelihoods to time NT

With a little thought, it is clear that with $S>S_R$ this approach will keep as survivors all those signals that would be retained by rule R1 as survivors with parameter R. The performance must therefore be at least as good. Again the problem of unambiguous decoding arises with this approach. Not all survivors will automatically merge and extra provisions would be needed to force optimal decisions between such signals. This rule generally will have much greater complexity than rule R2, so it has not been simulated.

2.6 Summary

This chapter has described decoding approaches which can maintain a survivor signal list at every time NT reduced in size from that of Viterbi decoding. The final rule R2, developed from the basic hyperplane approach, is the simplest in terms of computational complexity. A fixed order processing approach was required to reduce the computations below those required for Viterbi decoding. Under rule R2, the complexity reduction is by a factor of S_V/S where S_V is the number of states in the Viterbi implementation, and S is the number of survivors with rule R2.

As developed all the alternate rules have the same sequential nature as MLSE. The minimum weight and length of error events is therefore identical to that for MLSE decoding. The signal likelihoods are derived identically to Viterbi decoding. The receiver structure differs only in the processing algorithm. The same quadrature demultiplexing and the same matched filters are required.

Rule R2 with $S=S_L=M^{(L-1)}$ is of special interest since asymptotic optimality should be achievable with minimum complexity. Simulations are carried out in Chapter 3 to verify expected performance with $S=S_L$.

CHAPTER 3

SIMULATION OF DECODING RULES APPLIED TO PARTIAL RESPONSE FM

In this chapter simulations of basic rule R1 and modified rule R2 applied to partial response FM are presented. Error rate performance was obtained by Monte Carlo simulation over a range of signal to noise ratios (SNR). To achieve reasonable statistical accuracy, run lengths were chosen to yield at least 25 error events, 15 events at the very highest SNR values.

The object was to verify the expected error event rates and to assess the lengths of typical error events under these rules. Overall symbol error rates are compared to the basis curve $Q(SNR^{\frac{1}{2}})$. For the purposes of the simulation, signal to noise ratio is defined by $SNR \triangleq d^2_{min}/2N_o$, where d_{min} is the minimum distance for the modulation of interest, and $N_o/2$ is the double sided power spectral density of the Gaussian white noise. This choice performs a normalization so that a rule's performance on all modulations may be readily compared to MLSE. A perfectly coherent receiver is assumed.

The modulations considered have a basic rectangular frequency pulse of duration T, and polynomials $(1+D)/2$, $(1+2D+D^2)/4$ and $(1+D+D^2)/3$ over a range of modulation indices less than unity. To simplify things, only a binary input alphabet is considered.

The occurrence of long error events for certain modulation indices is explained and revised approaches developed to deal with these cases.

To summarize the rules:

RULE R1: If for any j, $[\overline{r}-\overline{y_i}]^N_{\delta} N_{ij} > R$ then reject y_i at time NT where the y_j are all contenders at NT, that is all extensions of the survivors from (N-1)T. If all signals fail the tests, keep the one with greatest likelihood as sole survivor to NT.

RULE R2: At NT, all contenders are ordered in a list according to their input digit sequences. Always comparing the outside signal pair, reject the contender with lower likelihood until S remain. Accepts all S as survivors at NT. Extend the survivors into all possible contenders for retesting at (N+1)T.

Only the cases $R=d_{min}/2$ for rule R1 and $S=S_L$ for rule R2 were considered. For (1+D)/2, $S_L=2$ and for the second order polynomials, $S_L=4$.

3.1 Simulation Method

A random input digit sequence was generated to simulate the usual information source. In order to determine the signals corresponding to the digit sequences, "pseudo-states" were derived. These pseudo states are formed by combining the $2^{(L-1)}$ bit histories with all possible phases which may be reached at any time kT. The true phase states are a <u>subset</u> of these phases (see Section 1.3.3). Similarily, the Markov states are a subset of the pseudo-states.

The advantage of using these pseudo-states is that, for checking the simulations, the phase of the signal may be quickly related to the pseudo-state. In addition, the pseudo-states encompass all possible Markov states regardless of the initial state of the transmitted signal. Of course, only the Markov subset of the pseudo-states will actually occur for any given simulation.

Signals of duration T are uniquely associated with each state transision. Using the signal space ideas, all possible duration-T signals are represented as vectors in signal space. The vector coordinates were calculated using all the pairwise separation distances of the duration-T signal sections. Any signal over an interval of many digit intervals, NT, is simply the sum of the appropriate section vectors. Additive Gaussian white noise is simulated by adding a noise vector to the vector representing the transmitted signal section for each interval. As the noise is white, these noise vectors should be independent from interval to interval as well as have independent components during each interval. The noise vector components were derived by forming independent samples of an approximately Gaussian random variable by a mathematical relation described in Section 3.1.3.

The relevant likelihoods are formed recursively as the sum of likelihoods due to each interval. The interval likelihood is simply the inner (dot) product of the signal segment vector and the received signal segment vector. These likelihoods are used to arrive at the survivor signals for each interval according to the specific decoding rule simulated.

3.1.1 Vector Formulations:

The number of dimensions needed to completely describe all possible duration-T signals depends on the particular modulation. These signals are not all linearly independent. The general form of such a signal emerging from phase θ_n is $s_i = A\cos(\theta_n + \phi_i(t))$, where $\phi_i(t)$ depends on the bit history and current input bit. Expanding

$$s_i = A\cos\theta_n \cos\phi_i(t) - A\sin\theta_n \sin\phi_i(t)$$

$$= A\cos\theta_n \cos\phi_i(t) - A\sin\theta_n \cos(\phi_i(t) - \pi/2) \quad (26)$$

All signals can be represented as linear combinations of the basis signals $A\cos\phi_i(t)$ and $A\cos(\phi_i(t)-\pi/2)$. If there are ν distinct paths $\phi_i(t)$, then 2ν basis signals are needed

For example, modulation $[(1+2D+D^2)/4,\ h=\frac{1}{2}]$ has only five distinct polynominal sums, and therefore only five distinct phase paths $\phi_i(t)$; $\nu=5$. Ten basis signals are needed.

The vector coordinates for these basis signals can be calculated using all the pairwise separations. To perfectly represent the 2ν basis signals, up to 2ν dimensions may be needed (a Gram - Schmidt orthogonalization could be done). However, sufficiently accurate representation is usually possible with a lower number of dimensions. A program was written to calculate vector coordinates for the basis signal vectors with a minimum number of dimensions to achieve a maximum component "residual" of 1% of the length of the vector. The vector length is $\sqrt{E}$, where E is the energy per bit. This was done as follows.

The energy per bit E is normalized to unity. One signal is chosen and placed a distance of unity away from the origin along the first coordinate axis. Each remaining signal is tentatively constrained to dimension two, and the tentative component in this new dimension is calculated. The signal with the largest component is selected to be constrained to this dimension and keep this representation. The process repeats, adding one more dimension each iteration until the largest tentative component in the new dimension is less than .01 for all signals not completely represented in the current dimensionality. All signals then retain their present representations in the current dimensionality. The pairwise separations were re-calculated using the vector coordinates for this reduced-dimension representation to verify the accuracy.

The distances between the duration-T signal segments s_i, s_j are calculated according to equation 13,

$$\frac{d_{ij}^2}{2E} = \frac{1}{T} \int_0^T \left[1-\cos\Delta\phi_{ij}(t)\right] dt$$

where E is the energy per bit and $\Delta\phi_{ij}(t)$ is the difference in phase over the interval. For the modulations of interest, this difference phase is a linear function of time, and the distances are readily calculated in terms of the initial and final phases. If $\Delta\theta_{ij}$ is the initial phase difference between s_i and s_j, and β_{ij} is the net phase change between them over the interval,

$$\frac{d_{ij}^2}{2E} = 1- \frac{\cos\Delta\theta_{ij}}{\beta_{ij}} \sin\beta_{ij} - \frac{\sin\Delta\theta_{ij}}{\beta_{ij}} \cdot(\cos\beta_{ij}-1) \quad (27)$$

A program was written to calculate these distances.

3.1.2 Approach Used For Rule Rl

To simulate this basic rule, the quantity $2d^N_{ij}R-(d^N_{ij})^2$ is needed for every possible test pair (i,j). Since all survivors under this rule share a common history within a few intervals before the tests, their base-2 input sequences can be used to index a memory location where this quantity can be found. For $[(1+D)/2,\ h=\frac{1}{2}]$, a three bit history is needed and for $[(1+2D+D^2)/4,\ h=\frac{1}{2}]$, a four bit history is needed. The distances between all possible contender signal pairs were calculated by a special program.

3.1.3 Simulation Program Details

A flow chart showing the program used is given in Figure 10. The transmitted, contender and survivor bit sequences were stored in 36 bit words. Random input bits were selected from a random generator of period $2^{18}-1$. The noise vector components were calculated by summing twelve samples from a uniform distribution to simulate a Gaussian random variable, and scaling to achieve the required variance. The uniform samples were provided by a standard recursive relation.

The relation yields an 18 bit word between zero and unity. The samples are effectively independent due to the low correlation between them. To achieve an SNR of X dB, where SNR = $d^2_{min}/2N_o$, the noise component variance ($\sigma^2=N_o/2$) must be given by

$$\sigma^2 = \frac{d^2_{min}}{4\cdot 10^{(X/10)}}$$

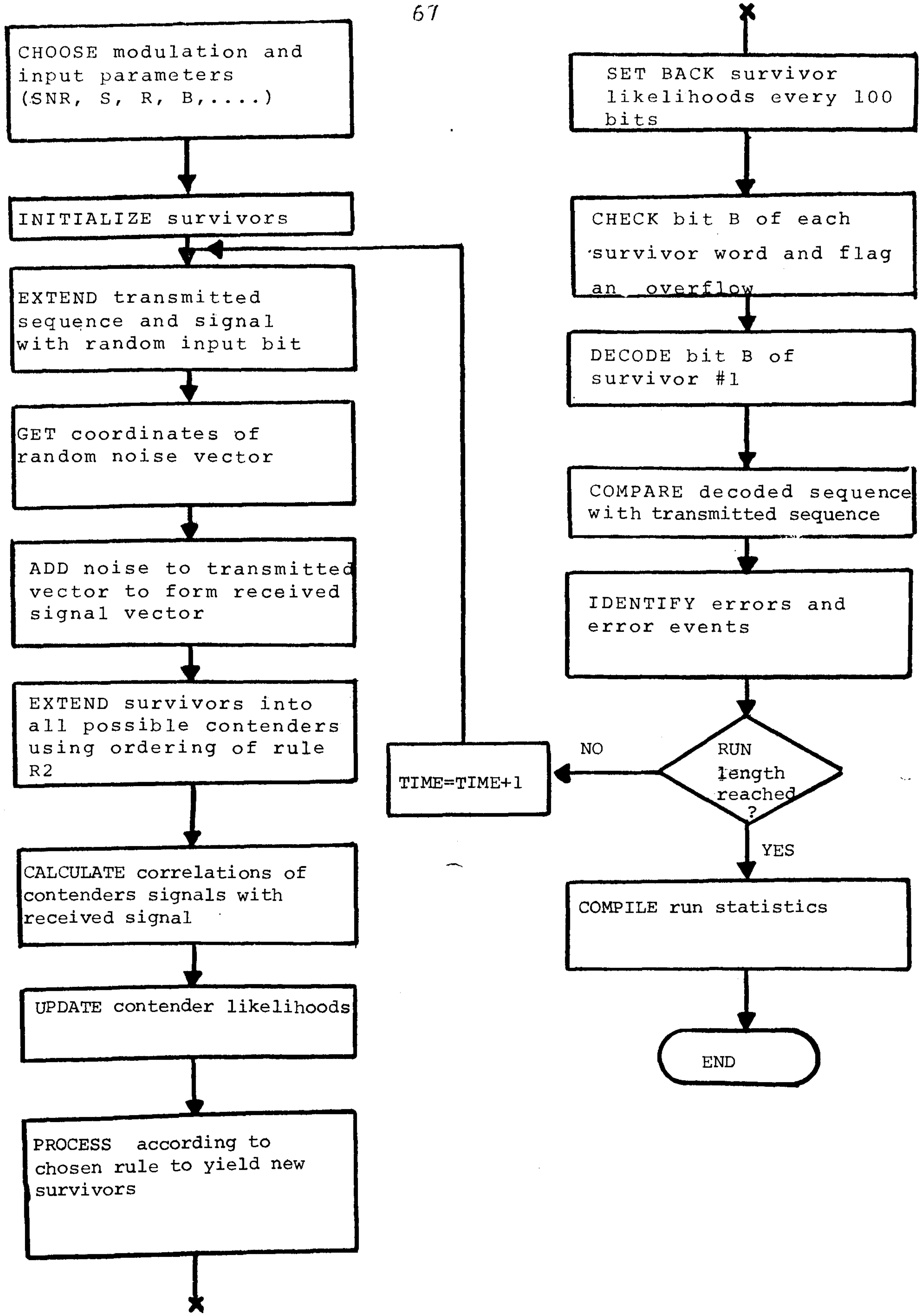

FIGURE 10 Simulation program flow chart

Since the sum of the twelve uniform samples yields a variance of unity, this sum must be multiplied by a scale of factor SF,

$$SF = \frac{d_{min}}{2 \cdot 10^{(X/20)}} \tag{28}$$

Decoding with buffer length B was achieved by extracting bit B of the word containing the digit sequence for survivor number "one" (an arbitrary choice that does not affect the decoding). The decoded bit sequence was compared with the transmitted bit sequence to identify error events as well as bit errors. Printouts showed a number of specific error events and provided statistics for the error events for the entire run. These statistics included maximum and average length and bit errors as well as total number of error events and mean time to error event.

As the signal likelihoods continue to grow with time, it is necessary to set them back to lower values occasionally. This was done by subtracting the lowest likelihood from all the survivor likelihoods every one hundred bits.

3.2 Results And Discussion

Error event rates and symbol (bit) error rates are presented in this section. Where relevant, both rates will be shown. In general, error event rates are more indicative of the basic performance. True behaviour can be masked by only considering overall bit error rates. At low SNR where there are many error events, and where these events are long, a "bias" is introduced into the overall rates. Clearly, an error event cannot start where one is already in progress. The overall error event rate will tend to look lower than what would be expected from the value $Q(d/\sigma)$.

To partly remove the effect of the "bias" error event rates are defined as the reciprocal of the mean time to error event (MTTEE). This measure is an average of the time from the end of one event to the start of the next.

As a basis for comparison, the curve $Q(SNR^{\frac{1}{2}}) \equiv Q(d_{min}/2\sigma)$ is shown along with the data points. It should be recalled that MLSE error event rates can be asymptotically equal to $\frac{1}{2}Q(d_{min}/2\sigma)$ in some cases.

Table 1 contains quantities relevant to the simulations. Buffer lengths were chosen so that overflows had a negligible effect on the results.

3.2.1 Basic Hyperplane Rule R1

Only the case $R=d_{min}/2$ was considered when verifying the expected asymptotic optimality of this rule. Only modulations $[(1+D)/2, h=\frac{1}{2}]$ and $[(1+2D+D^2)/4, h=\frac{1}{2}]$ were simulated. The results are shown in Figure 11.

The maximum number of survivors S^*_{max} was found to be equal to the theoretical value, S_R, $R = d_{min}/2$. Merging within the expected number of bit intervals was verified. No overflows occurred with buffer lengths of three and four bits.

Modulation	$S_R, R=d_{min}/2$	S^*_{max}	Buffer	$P(e)_{max}/Q(d_{min}/2\sigma)$
$(1+D)/2, h=\frac{1}{2}$	2	2	3	3
$(1+2D+D^2)/4, h=\frac{1}{2}$	4	4	4	7

$P(e)_{max}$ = bound on error event probability

Table 1

Quantities relevant to MLSE or alternate decoding of partial response modulations of interest.
P = number of phase states, S_v = number of Viterbi states,
F = number of matched filters, ℓ_{min} = minimum error event length,
e_{min} = number of errors in minimum length event.

Modulation		P	S_v	F	$d^2_{min}/2E$	ℓ_{min}	e_{min}
(1+D)/2	h=1/2	4	4	8	1.73	3	2
	1/4	8	8		0.49		
	1/6	12	12		0.22		
	4/7	7	14		2.14		
	3/8	16	16		1.05		
$(1+2D+D^2)/4$	h=1/2	4	8	12	1.45	4	2
	1/4	8	16		0.40		
	3/5	10	20		1.98		
	4/7	7	28		1.83		
	3/8	16	32		0.86		
$(1+D+D^2)/3$	h=1/2	4	8	8	1.34	4	2
	1/4	8	16		0.36		
	3/5	10	20		1.87		

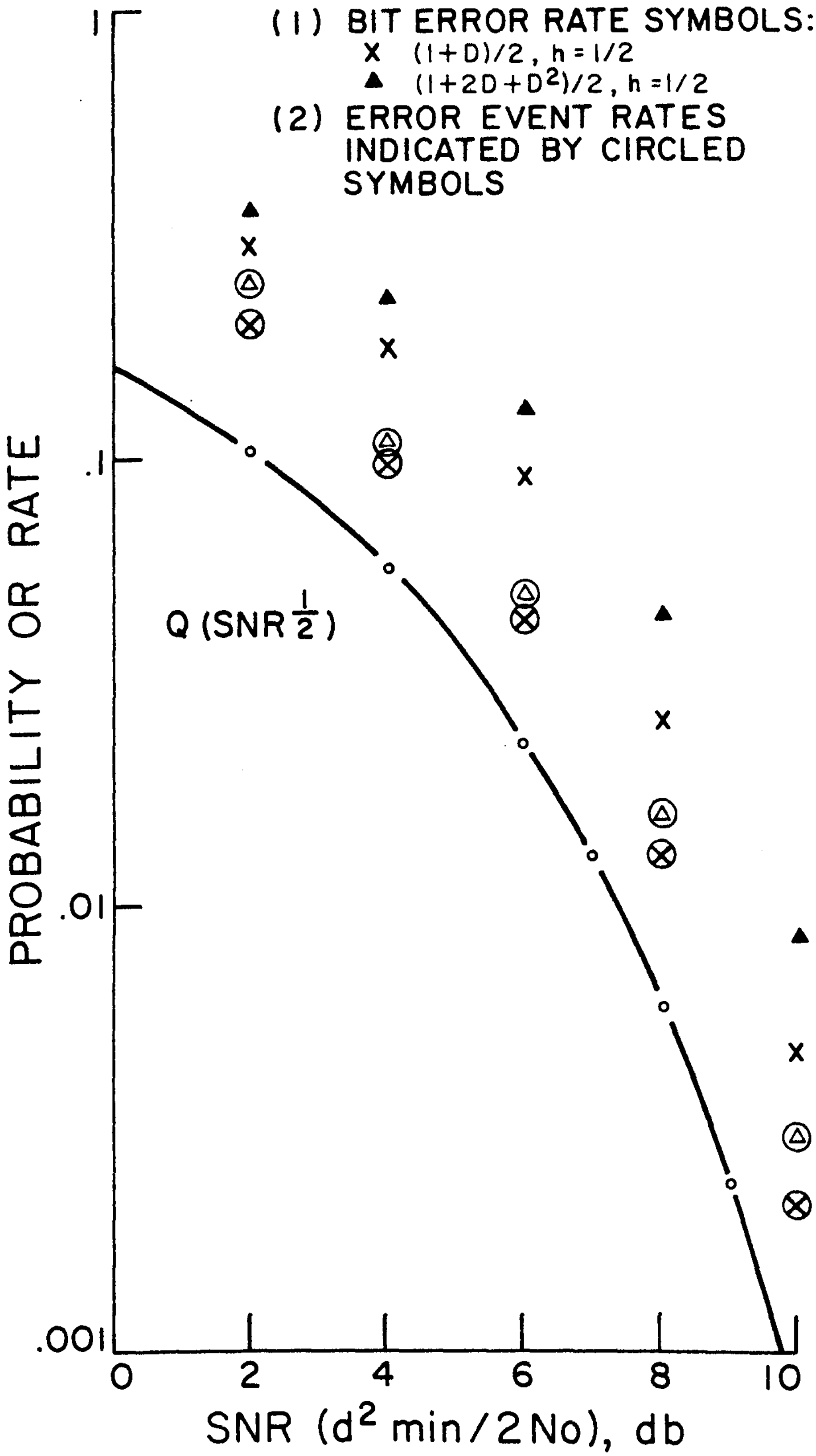

FIGURE 11 Error rates for rule R1.

In all cases, the bound on the error event probability, $\gamma Q(d_{min}/2\sigma)$, was found to hold. The average length of error events increased slowly with decreasing SNR as expected, but was always less than 1.5 times the minimum event length, ℓ_{min}. Even at high SNR, long error events occurred, three to five times ℓ_{min}. With a low number of survivors, long link-up times can occur.

3.2.2 Rule R2, Modulation Index h=1/n, $S=S_L$

Simulations were run on all three PRS polynomials at h=1/2 and h=1/4, and on (1+D)/2 at h=1/6. The number of survivors was taken as $S_L=2^{(L-1)}$, i.e. two survivors for (1+D)/2 and four survivors for the second order polynomials. A digit sequence buffer length of eight bits was used for (1+D)/2 and twelve bits for the second order polynomials. The results from these simulations should be indicative of performance at h=1/n for n any integer. The results are shown in Figures 12(a), (b).

Although the effect of buffer length was not specifically investigated, it was found that overflow occurred only very rarely and had negligible effect. Even an eight bit buffer worked very well for the second order polynomials.

The bit error rates seem to be asymptotically equal to a constant (1 or 2) times $Q(d_{min}/2\sigma)$. This is achieved even at low SNR. Even though there are two bit errors per event, for the second order polynomials a minimum distance alternate is only in contention about half the time. This leads to an overall bit error rate around $Q(d_{min}/2\sigma)$ at high SNR.

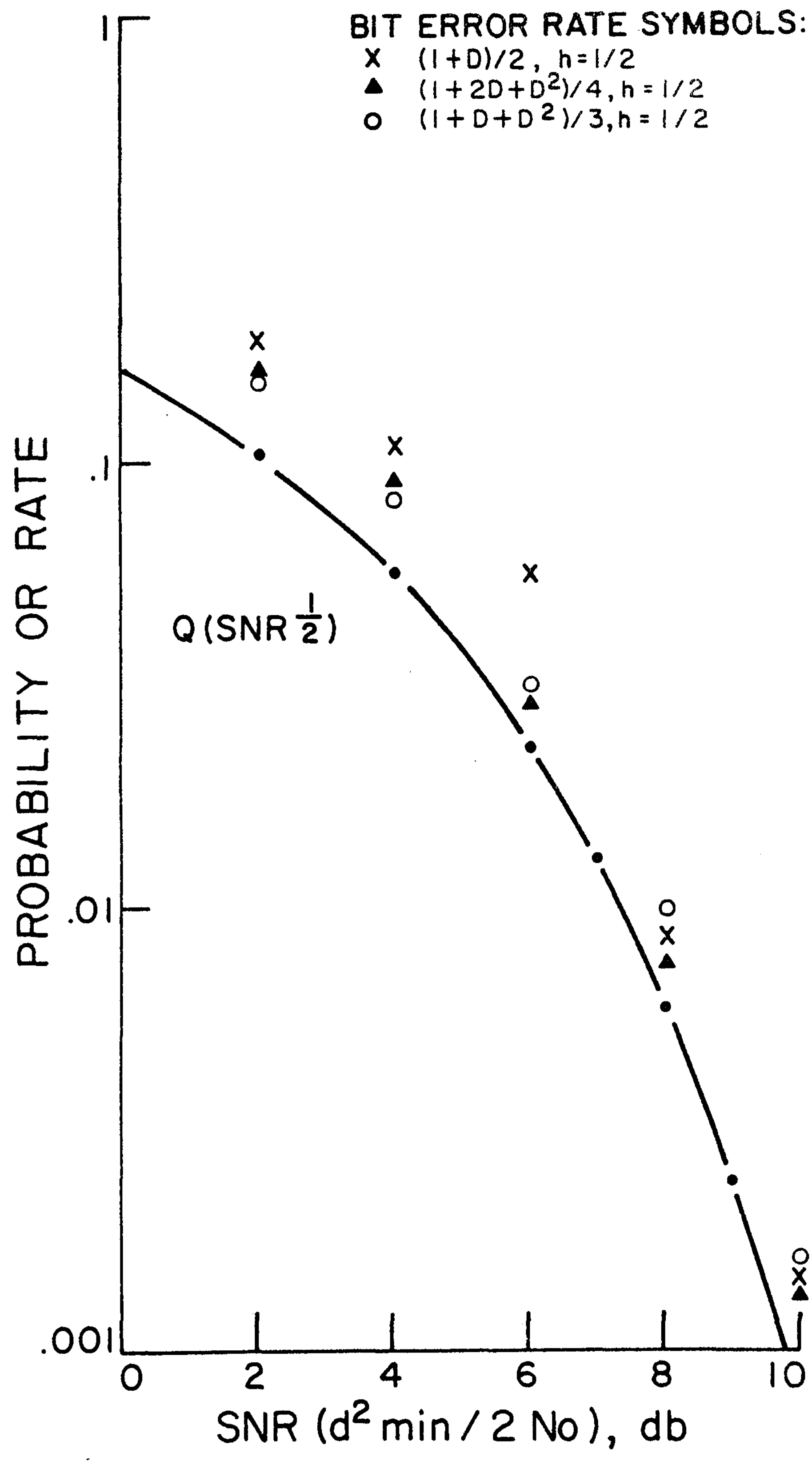

FIGURE 12(a) Error rates for rule R2, modulation index h=1/n, $S=S_L$.

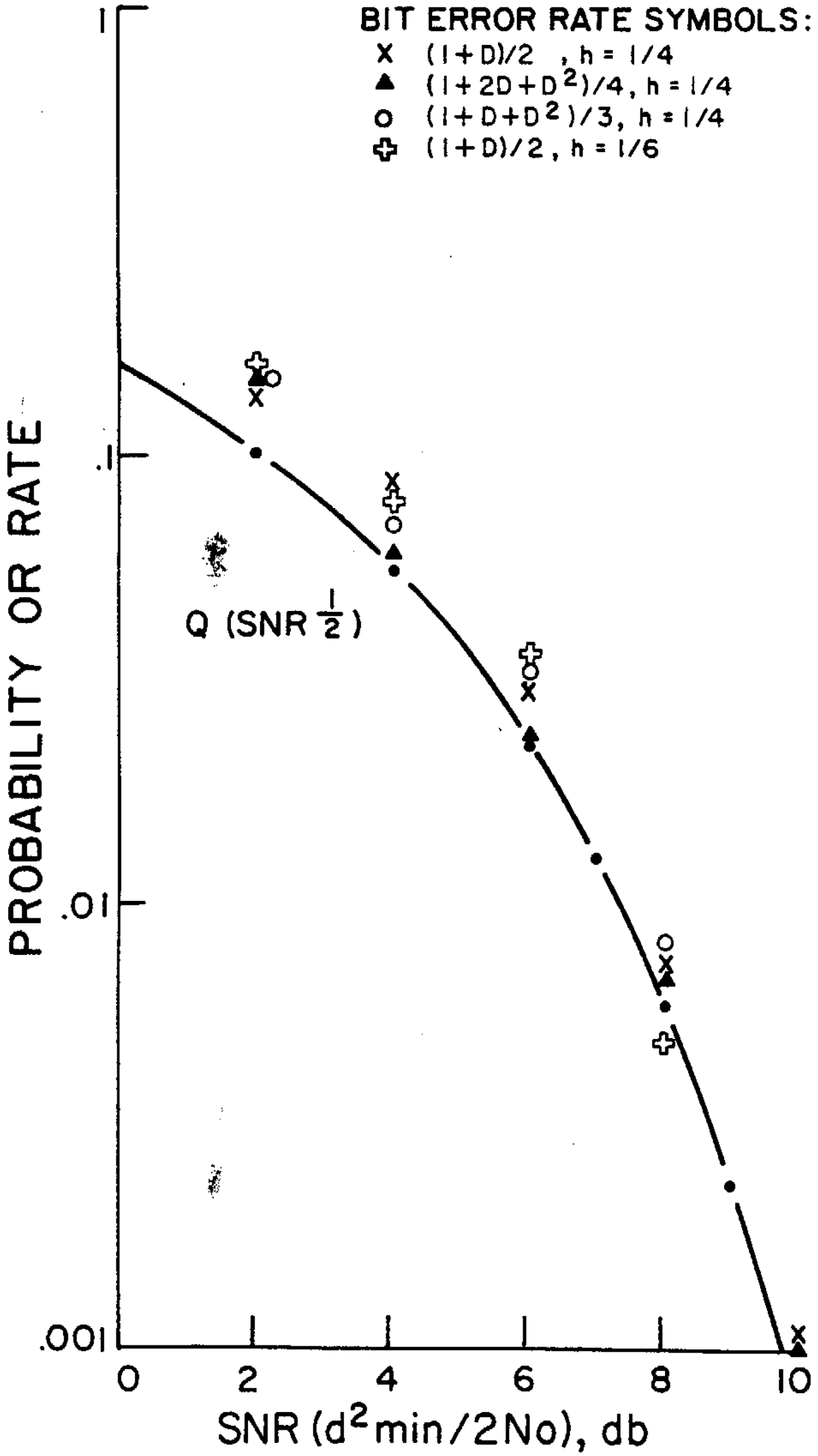

FIGURE 12(b) Error rates for rule R2, modulation index h=1/n, $S=S_L$.

The average event length was found to increase slowly with decreasing SNR, but was always less than 1.2 times ℓ_{min}. At high SNR, the average event length is virtually equal to ℓ_{min}, although not all events are necessarily of weight d_{min}. The maximum event length was less than two times ℓ_{min}.

As expected, many of the rejections of the transmitted signal were in favour of a direct d_{min} alternate. These optimal decisions indicate the rule is tracking the ML signal through d_{min} error events. For (1+D)/2 at h=1/4 and h=1/6, and for the second order polynomials, especially at lower h, tracking of the ML signal was especially good. In these cases virtually all error events were due to optimal decisions between minimum distance alternates. The performance in this case is essentially equivalent to MLSE. This behaviour is readily explained in terms of the distances between signals involved in likelihood comparisons. Those signals that are not d_{min} direct alternates are separated by distances much in excess of d_{min}. At high SNR, the probability of rejecting the transmitted signal for anything other than a direct d_{min} alternate is very small.

Although there is a sequence pair (XXX10011, XXX01100) that has signal separation .6 dB less than d_{min} for modulation $[(1+D+D^2)/3,\ h=\frac{1}{2}]$, this had negligible effect on the overall error rate at the rates tested. This is due to the relatively low probability (1/16) of the transmitted signal taking such a path. (see Section 2.4.2).

Those error events that were not of minimum weight could be grouped into two types. The first type showed a decision path that diverged from the true path and then reconverged with a net phase traversal of zero radians. The other type showed a net phase deviation of 2π. These types are indicated in Figure 13 for modulation $[(1+D)/2, h=\frac{1}{2}]$. The type of error event depends not only on the noise, but on the transmitted signal path at the time of rejection. The probability of rejection itself depends on the actual transmitted signal path (effective R parameter, see Section 2.4.2).

If the transmitted signal is rejected when it is showing maximum phase change over consecutive intervals, there will be no d_{min} weight path for the rule to take to link up again. In this case, the decision path may parallel the actual signal or diverge 2π to achieve linkup. In both instances, the ability to link up depends on the transmitted signal path after rejection. This provides a mechanism for the longer error events. Whether the decision path diverges 2π or parallels the true signal depends on the relative distances separating the "diverging" or "paralleling" survivor descendants from the transmitted signal path. Given moderately high SNR, the survivor descendants that are closer to the transmitted signal path are more likely to be retained by the rule.

3.2.3 Rule R2, Modulation Index $h=m/p$, $S=S_L$

Simulations were done for various combinations of polynomials and modulation indices. For $(1+D)/2$ with $h>1/2$, $S_L=2$ is suboptimum as the signals that "wrap around the back of the phase cylinder" are somewhat closer than d_{min}. In these cases $S=8$ would be necessary for optimality. Only a small loss should be incurred by keeping only two survivors. Buffer lengths were the same as for R2, $h=1/n$. The simulation results are shown in Figure 14. Table 2 summarizes the error event length statistics.

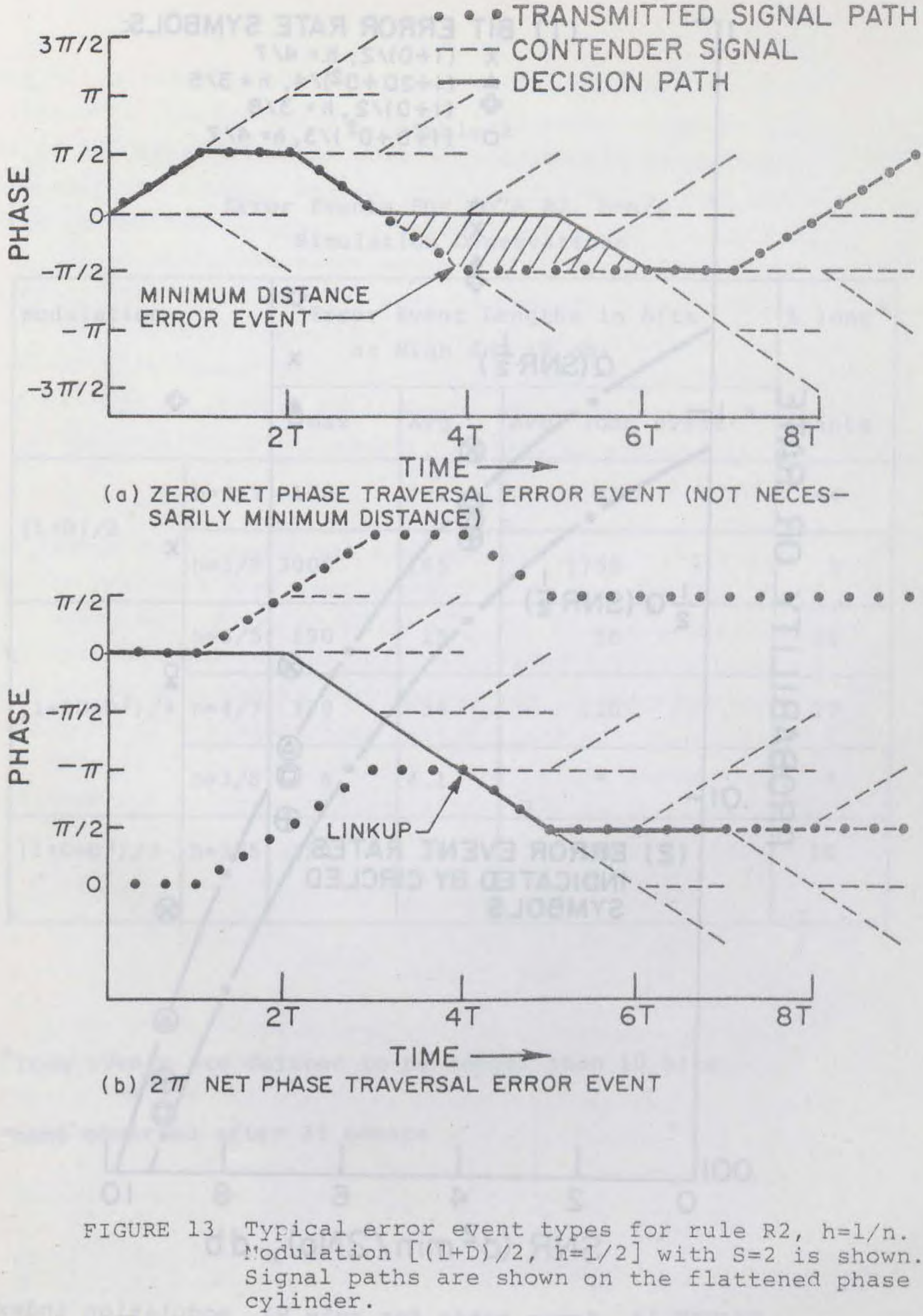

FIGURE 13 Typical error event types for rule R2, h=1/n. Modulation [(1+D)/2, h=1/2] with S=2 is shown. Signal paths are shown on the flattened phase cylinder.

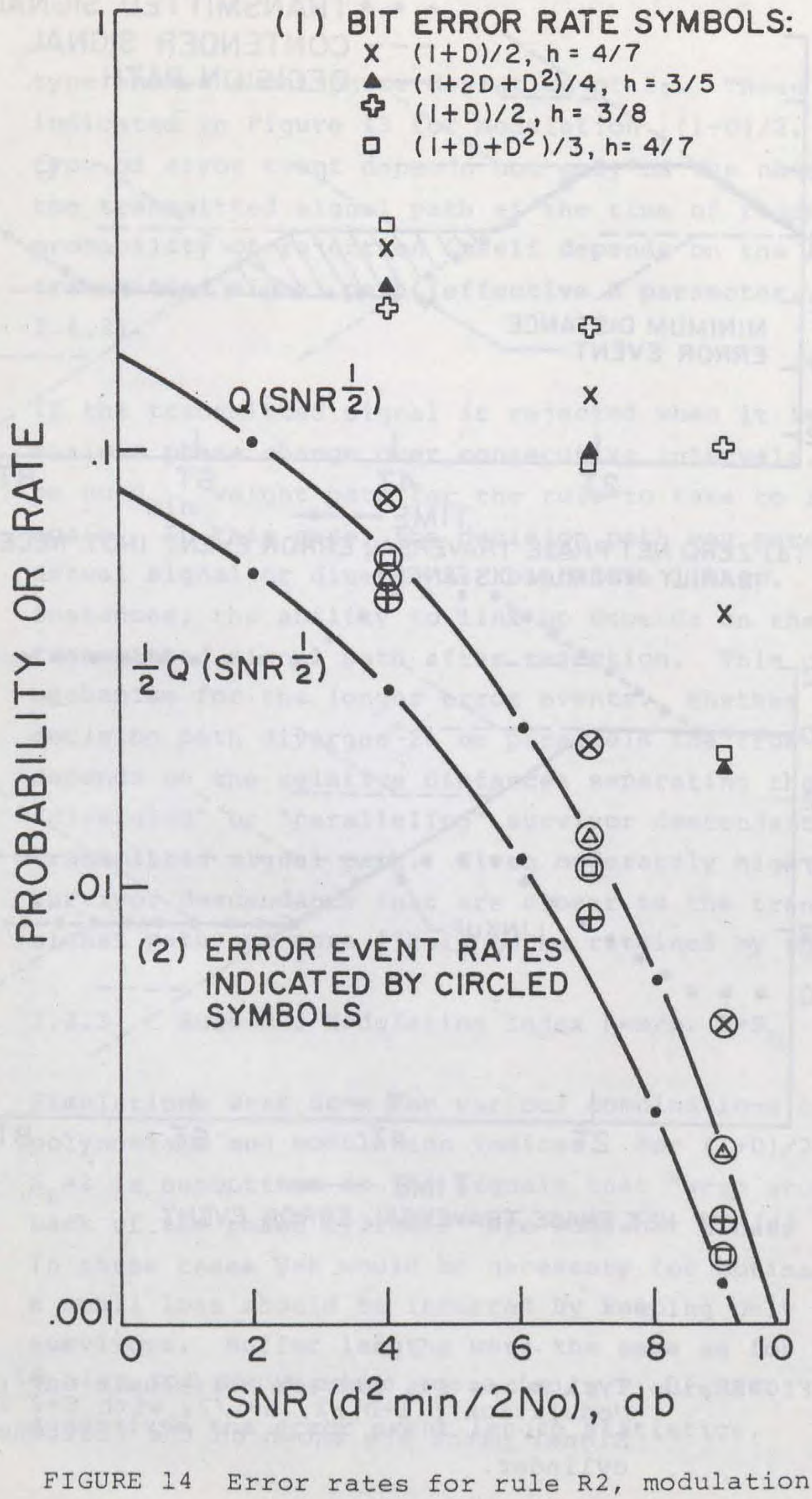

FIGURE 14 Error rates for rule R2, modulation index h=m/p, $S=S_L$.

Table 2

Error Events For Rule R2, h=m/p
Simulation Observations

Modulation		Error Event Lengths in Bits at High SNR (9 dB)			% long[a] events
		Max	Avg.	Avg. long event[a]	
(1+D)/2	h=4/7	170	22	45	44
	h=3/8	3000	165	1750	5
$(1+2D+D^2)/4$	h=3/5	190	15	50	24
	h=4/7	370	34	120	27
	h=3/8	6	4.1	*	*
$(1+D+D^2)/3$	h=3/5	220	39	110	30

[a]long events are defined to be longer than 10 bits

*none observed after 35 events

Although the error event rate seems to asymptotically equal to a constant times $Q(d_{min}/2\sigma)$ for the optimal S_L cases, the symbol error probability is very high. This is due to the occurrence of very long error events (some up to thousands of bits!). Given the success of the rule for h=1/n, this seems puzzling until the error event mechanism is investigated. The events that were to blame were those in which the rule's decision path diverged from the transmitted path and achieved a false "lock" with the 2π-shifted version of the transmitted signal. A typical event of this nature is shown in Figure 15 for modulation [(1+D)/2,h=4/7].

This inability to diverge from the transmitted signal path and achieve link up 2π radians away is a function of the modulation index. If link up is to occur, the state sequence of the rule's decision path must merge with the state sequence of the transmitted signal path at the time of link-up; the phase states must therefore be identical. If the phase state of the transmitted signal at time nT is ϕ_n, and that of the rule's decision path is θ_n, then for linkup at nT:

$$\phi_n - \theta_n \equiv \left[\sum_{k=d}^{n-L} \Delta a_k g(LT) \right]_{mod 2\pi} = 0 \qquad (29)$$

Here Δa_k represents the difference between the input digits of the transmitted and decision path at time kT, $\Delta a_k = a_{tk} - a_{dk}$, and dT is the time at which the two paths diverge. The set $\{-2(M-1),\ldots,-2, 0, 2, \ldots, 2(M-1)\}$ contains all possible values for each Δa_k. This means that

$$\sum_{k=d}^{n-L} \Delta a_k g(LT) = 2m \cdot g(LT)$$

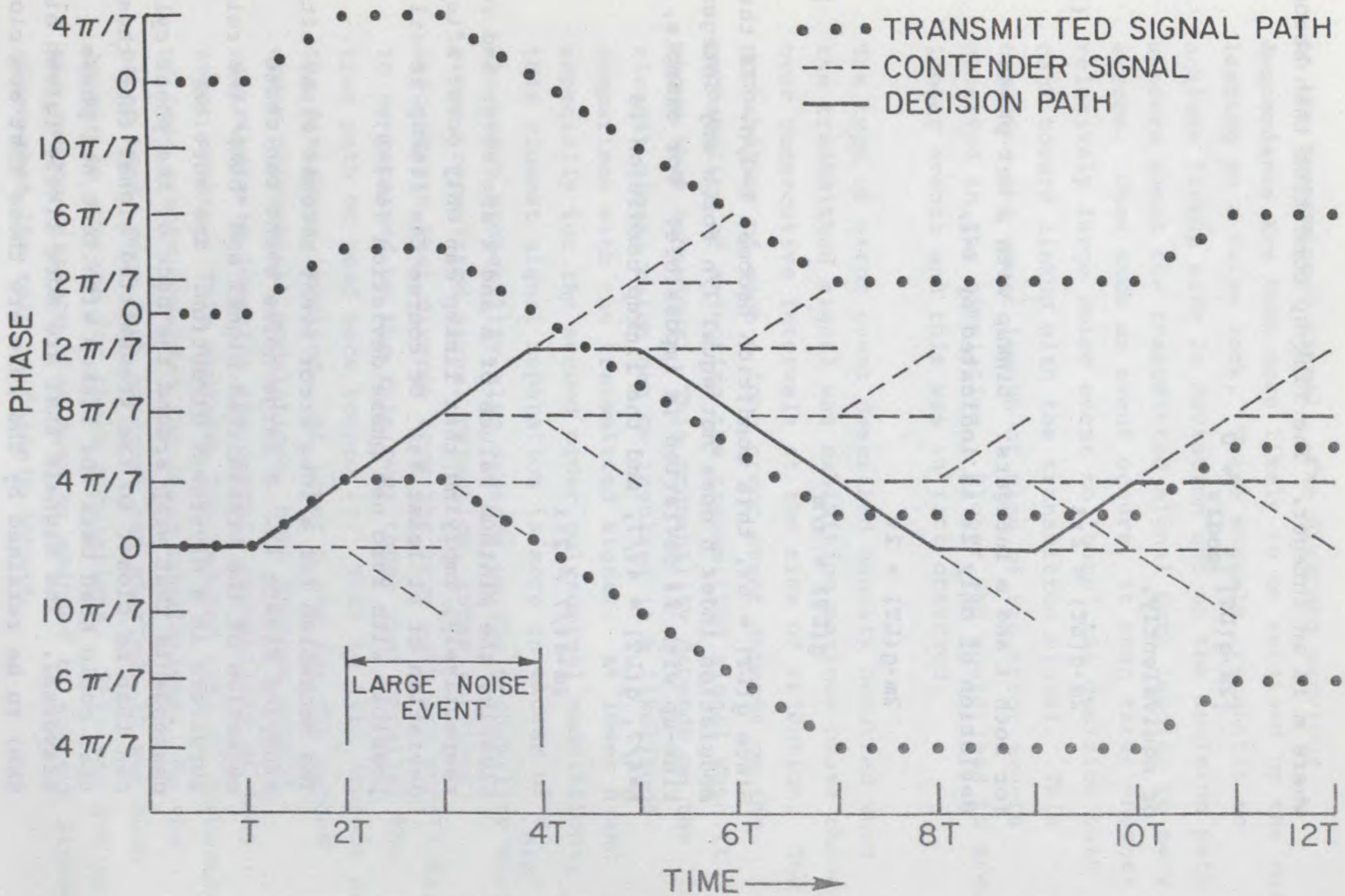

FIGURE 15 Typical false lock error event for rule R2, h=m/p. Modulation [(1+D)/2, h=4/7] with S=2 and binary input is shown. Signal paths are shown on the flattened phase cylinder.

where m is an integer. The link-up condition then becomes

$$[2m \cdot g(LT)]_{\mathrm{mod}\,2\pi} = 0$$

or equivalently,

$$2m \cdot g(LT) = 2\pi\ell \qquad (30)$$

for both ℓ and m integers. Linkup with a net phase deviation of only 2π is indicated by $\ell=1$,

$$2m \cdot g(LT) = 2\pi$$

$$g(LT) = \pi/m \qquad (31)$$

Since $g(LT) = h\pi$, this condition becomes $h=1/n$. If the modulation index h does not equal $1/n$ for n any integer, link-up with 2π deviation is impossible. For example, with $h=4/7$, $g(LT) = 4\pi/7$, and the linkup condition is

$$2m \cdot 4\pi/7 = 2\pi\ell$$

Clearly, the minimum values of m and ℓ are seven and four respectively, implying that linkup can only occur after a deviation of at least 8π. Of course the linkup is still possible with zero net phase deviation ($m=\ell=0$).

The mechanism for a long error event becomes clear after studying Figure 15. A large noise event can cause rejection of the transmitted signal and "bump" the rule's survivors in a divergent direction. The survivor descendants that wrap around the back of the phase cylinder can then be closer to the transmitted signal than those that could head back for linkup with zero net phase traversal. The signals that are more likely (given high SNR) to be retained by the rule are those that are closer to the transmitted signal. The diverged survivor descendants are then more likely to be retained by the rule leading to a false lock. These signals are unable to achieve linkup with 2π deviation and so the decision path wanders about the transmitted signal path leading to many errors. Once such an event occurs, it then takes another relatively large noise event to "bump" the decision path back toward linkup with the transmitted signal. This explains the great length of many of the error events observed in the simulation. Higher SNR would lead to even longer events and this was in fact observed.

The type of error event described usually occurred when the transmitted signal was exhibiting maximum phase change over consecutive intervals at the time of rejection. The probability of such a long error event depends on the probability of rejecting the transmitted signal when it shows this behaviour. This probability depends on the closest signal which can be involved in a likelihood comparison with the transmitted signal. At lower h and especially for the second order polynominal modulations, this closest signal separation is more in excess of d_{min}. The event probability also depends on the probability that given transmitted signal rejection, those survivor descendants that diverge from the true path are more likely to be retained by the rule over those that parallel the true path or head back toward it. This in turn depends on the distances involved. At lower h and for the second order polynominal modulations, both factors should contribute to lowering the overall probability of long error events. This explains the reduced rate of occurrence of long events observed for these modulations. On the other hand, once such an event had started, it was much longer on average for these modulations. This was due to the larger distance that must be spanned to achieve linkup given that the rule is falsely locked.

Even though error event rates seem to be asymptotically optimal, the occurrence of long error events can lead to poor overall symbol error probability. There is no mechanism to limit the length of these events (other than high noise itself). Ways of trying to deal with this problem are covered in the next section.

3.3 Expanded Algorthms for h=m/p

There are two general approaches which may be identified to deal with the long error event problem:

(1) Monitor progress and take corrective action when it is determined that a long error event is likely in progress (false lock).

(2) Provide expanded processing that operates continuously and tends to limit the length of error events.

It was found that it was possible to identify the occurrence of long error events with reasonably high probability. This can be done by monitoring the trend of the decision path likelihood. When the rule is off track, the decision path shows lower correlation with the transmitted signal. If the change in decision path likelihood is observed over a long enough interval, a fairly reliable indication of false lock can be provided. When detected, expanded processing could be used to try to bring the decision path back on track.

Instead of following this monitoring approach, the fixed approach which includes expanded processing at all times was investigated. The same expanded processing could be introduced when required by the monitored approach.

Initial attempts were made at using an increased number of survivors for rule R2 while still maintaining a complexity reduction over the Viterbi algorithm Although this could reduce the probability of these long events occurring (higher effective R parameter), it was found that the length of the error events was not significantly reduced. Other attempts were made at deliberately introducing extra contenders that diverged from the current decision path in hope that these might achieve linkup should a false lock be in progress. Neither method was effective as they both faced the same problem: once the rule gets bumped off track, those survivor descendants that are headed back for linkup are further from the transmitted signal path, will tend to have lower likelihoods, and will be discarded by the rule.

3.3.1 Expanded Processing Using Shifted Survivors

The approach finally settled on was arrived at after inspection of the false lock phenomenon shown on the phase tree (or flattened phase cylinder). Once false lock occurs, if the survivors could only be shifted in phase in the appropriate direction (up or down), a linkup could be achieved. This linkup would be effected by essentially violating the modulation structure during the error event.

The necessary phase shift is simply $k \cdot (2\pi/p)$ where p is the number of phase states and k an integer. For example, for $h=4\pi/7$, a shift of $\pm k \cdot (2\pi/7)$ is required. The integer k depends on how far the decision path has diverged from the transmitted signal. The most likely false lock occurs after a divergence around 2π and the necessary shift is $\pm 2\pi/p$. Even if higher shifts are required, successive shifts of $\pm 2\pi/p$ should eventually lead to linkup.

This shifting idea is fine once we know that an error event is in progress. Obviously, this is never known with certainty. In addition, the required direction of shift is not known. It becomes necessary to choose between the shifted and unshifted survivor signals. This choice must be made in a manner that does not seriously affect the overall probability of error events. The basic approach is this:

(1) Shift the rule R2 survivors by $\pm 2\pi/p$ to form an expanded signal set. Assign bit histories and likelihoods to these shifted survivors identical to those of the unshifted survivors.

(2) Process the shifted and unshifted survivor descendants in separate groups using rule R2. Each group will generate a decision path where the survivors have merged. This expanded processing should continue until "sufficient" distance has been developed between the two separate decision paths.

(3) Decide on which decision path is more likely to be on the track of the transmitted path and keep the survivors corresponding to this decision path. Reject the other group.

(4) Repeat this shifting on the new survivor group and continue the expanded processing in a cyclic manner.

The idea is that if a false lock is in progress, and the proper shift is made, the shifted survivors will spawn descendants that should quickly link up with the transmitted signal path. The decision path of this shifted group should, after a long enough interval, achieve much higher correlation with the transmitted signal (higher likelihood) than the decision path of the unshifted survivors descendants. Picking the path with higher likelihood should set the rule back on track.

Conversely, if the shift occurs when an error event is not in progress, the unshifted (correct) survivor decision path should develop the higher likelihood and would still be likely to be retained by this approach. The probability of rejecting the proper survivor group and causing an error event sets the criterion for choosing the length of the expansion interval. This probability depends on the distance developed between the two decision paths. If we want this probability to be roughly the same as the error event probability for rule R2 alone, this distance must be sufficiently large.

3.3.2 Performance And Implementation Considerations

The following assumes that rule R2 is achieving asymptotically optimal error event rates. Extensions to the case when a deliberately sub-optimal number of survivors is chosen are clear.

The smallest distance between any two possible decision paths over the expansion interval is the quantity of interest for high SNR performance. This distance d_c depends on the modulation and the particular transmitted signal path. Suppose we are using enough survivors for rule R2 to achieve the optimal error event rate at high SNR. If the expansion interval is chosen long enough that this distance d_c exceeds d_{min}, the probability of rejecting the correct decision path should be no greater than $Q(d_{min}/2\sigma)$, and the overall error event performance of the approach should still be "optimal" at high SNR.

At practical, moderate values of SNR, a compromise on the expansion interval length is possible. Guaranteeing a certain distance between any two possible decision paths from shifted survivors may require a long expansion time. There will be worst case paths that remain close that would force a long expansion time. However, if these worse case paths have only very low probabilities of occurring, a shorter expansion interval should only slightly affect the overall probability of error events at moderate SNR. For example, if 95% of all possible decision path pairs develop at least d_{min} separation after E intervals of expansion, even if some were considerably closer than d_{min}, the overall error event probabiltiy can still be around $Q(d_{min}/2\sigma)$ at moderate SNR. To be more exact. each separation distance has a probability of decision error that should be weighted by the probability of that distance occurring in a choice between decision paths. The simplest way of determining the minimum expansion interval to maintain a desired error event rate may be to resort to simulation.

If the expansion interval is not too long, shifting can occur fairly often, and a long error event should be interrupted fairly early. The overall effect should be to reduce the length of error events from rule R2. Since the required shift direction is unknown, one strategy is to alternate the shifts at every expansion time. This approach will be called rule R3. Alternatively, shifts in both directions $\pm 2\pi/p$ can be made at every expansion time and three separate groups processed. This approach will be called rule R4. Rule R4 should tend to catch error events earlier than rule R3. Both rules were simulated. Their implementation flow charts are shown in Figure 16.

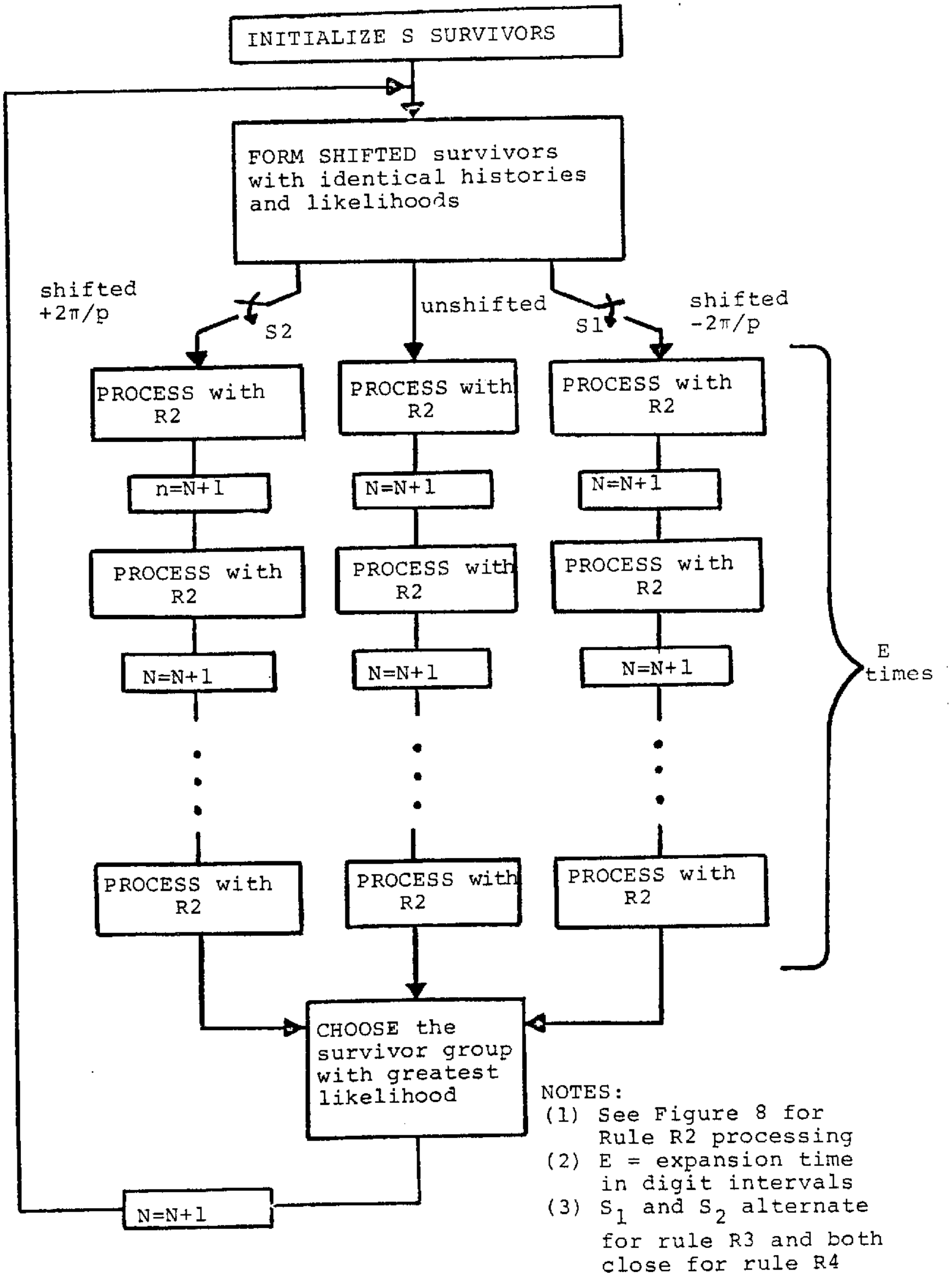

FIGURE 16 Rules R3 and R4 implementation flow chart

There are other considerations in implementing this shifting approach. To decide between the decision paths from the separate survivor groups, the likelihoods of the paths at the end of the expansion interval should be compared. These merged path likelihoods are not readily available. If the expansion interval is long enough, the likelihood of the maximum likelihood survivor from each group should work just as well. For the simulations, a simpler approach was used; the likelihoods of the first and last survivors from each group were averaged and taken to represent the decision path likelihoods. Another consideration is that a longer buffer length is needed due to the holding of distinct survivors over the expansion interval.

There is what may be a major drawback to using these shifted survivor approaches. The major requirement is that a sufficient distance be developed between any transmitted signal path and the decision path growing out of survivors shifted from the transmitted path. The shifted survivor decision path will invariably be quite close in phase to any transmitted signal path for the duration of the expansion interval. If the phase of the carrier is not known with high accuracy, the development of separation distance over the expansion time may be illusory. In this situation, the decision between the paths cannot be made accurately, and this shifted survivor approach may cause a catastrophic deterioration in error event probability. The approach that only takes action when a long error event is likely to be in progress may have a considerable advantage in practical use.

In any event, it is still useful to examine these shifted survivor approaches as they demonstrate that, at least in theory, a receiver that occasionally violates the modulation structure can still perform well.

The complexities of rules R3 and R4 are, respectively, at least two and three times that of rule R2. This assumes that the shifted survivor groups use the same number of survivors as rule R2. In addition there are extra operations required in deciding between the groups at the end of the expansion interval, and in initializing the shifted survivors at the beginning of the expansion interval.

3.3.3 Simulation Results For Rule R3 And R4, h=m/p

To compare the performance of the new rules with that of rule R2, simulations were carried out on the same modulations for an identical transmitted sequence and identical noise values. This was done by starting the random number routines at the same point. Buffer lengths were 32 bits. Along with Figure 17, Table 3 summarizes the relevant results.

Both rules tended to limit the length of the error events found for rule R2 while essentially achieving the same asymptotic error event rate. The error event rates for rules R3 and R4 were found to be almost identical for the expansion time used. Rule R4 provided only a slight improvement in limiting the error event lengths.

The most probable length of the long error events seems to be close to the length of the expansion interval chosen. In addition, there were longer events about two, three and four times this length, but these occurred mostly at lower SNR.

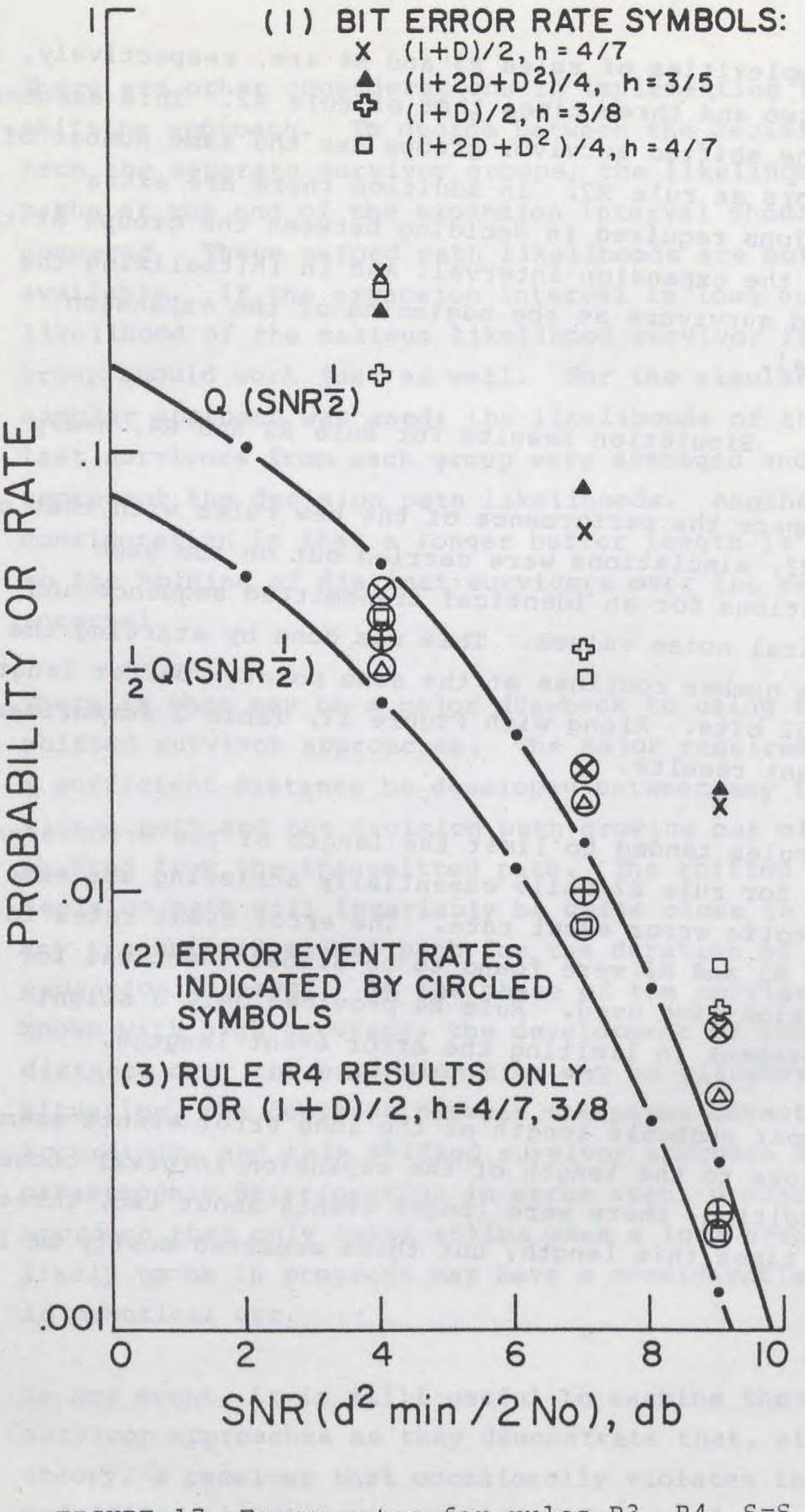

FIGURE 17 Error rates for rules R3, R4, $S=S_L$

Table 3

Error Events For Rules R3, R4; h=m/p.
Simulation Observations. E=expansion interval

Modulation		E	Error Event Lengths in Bits at High SNR (9 dB)						% long[a] events	
			Max		Avg.		Avg. long event[a]			
			R3	R4	R3	R4	R3	R4	R3	R4
(1+D)/2	h=4/7	10	29	22	5.7	4.8	25	20	14	7
	h=3/8	15	30	13	5.1	4.2	19	12	12	10
$(1+2D+D^2)/4$	h=4/7	15	25	--	7.4	--	15	--	32	--
	h=3/5	15	40	--	8.0	--	27	--	20	--
	h=3/8	30	27	--	4.4	--	17	--	8	--
$(1+D+D^2)/3$	h=3/5	15	33	--	7.6	--	24	--	22	--

[a]long events are defined to be longer than 10 bits

CHAPTER 4

CONCLUSIONS

Decoding approaches that are less complex than MLSE can be applied to bandwidth efficient constant envelope modulations. These approaches can achieve asymptotic (high SNR) performance almost identical to MLSE or can trade off performance for greater complexity reduction. These approaches maintain a list of survivor signals at every time that is reduced in size from that kept for Viterbi decoding. When some of the Viterbi survivor separation distances necessarily exceed d_{min}, a list reduction is possible while still maintaining asymptotic optimality. The degree to which reduction is possible depends on the modulation index.

The sequence decoders extend survivors into all possible descendants at each digit interval to form a set of contenders. These contenders are processed to reduce their number to a few survivors. Decisions are forced between contenders signals on the basis of their likelihoods. Asymptotic performance (as well as computational complexity) is governed by the decision rule's effective parameter R. For asymptotic optimality $R > d_{min}/2$. The lower R is chosen, the lower the complexity. As long as all decisions are between signals separated by d_{min} or more, asymptotic optimality of error event probability is possible. Complexity reduction over Viterbi decoding only seems possible by exploiting a fixed processing order. For many modulations an almost implicit ordering based on input digit sequences can be used to select appropriate signal pairs for likelihood comparisons. The rule developed with this approach was called rule R2. A general number of survivors, S, can be kept with this rule. The implicit ordering can only be exploited for low enough modulation indices (≤ 1). The choice of S depends on desired performance and complexity. Merging seems fast and overflow was not a problem with reasonable decoder buffer lengths.

A complexity reduction by a factor of S_V/S can be achieved with rule R2, where S is the number of survivors chosen and S_V is the number of states per interval in the Viterbi decoder.

For modulation indices $h=1/n$, a complexity reduction by a factor of n is possible. This reduction was demonstrated for the modulations considered for n=2, 4, 6 with high SNR performance almost identical to that expected for MLSE. As described in Section 2.4.2 modulation $[(1+D+D^2)/3,\ h=1/n]$ cannot achieve asymptotic optimality with only $S_L=4$ survivors but it still performed very well at the lowest error rates considered (10^{-3}).

The ability of the rule to link back up with the transmitted signal once it gets off track depends on the particular transmitted signal path. Error events of lengths two or more times the minimum event length are then possible even at high SNR, although the longer the event, the more improbable.

For modulation indices $h \neq 1/n$, the reduced survivor approach can lead to a false lock phenomenon and very long error events. Although the event probability can be asymptotically optimal, overall performance can be poor due to the long error events. If $h=n/k$, n and k least integers, the complexity reduction can be by a factor of k. The percentage of these long error events can be very small for higher SNR and lower modulation indices. In addition, a larger number of survivors can reduce the probability of these long error events. At high SNR and low h, these events can be extremely long implying a near catastrophic decoder failure when such an event occurs.

Expanded algorithms based on rule R2 can limit the length of these long events while maintaining error event rates nearly identical to rule R2. Although the maximum error event lengths are greatly reduced, "longer" error events are still possible. These expanded algorithms require roughly two to three times the operations required for rule R2. The complexity reduction for $h=n/k$ is then roughly by a factor of $k/2$ to $k/3$.

The results of the investigations may be used to make careful generalizations about the applicability of reduced complexity approaches to modulations other than those simulated. The complexity reduction by factor n for $h=1/n$ should hold for the case $M=4$ given that the same general modulation properties described in Section 2.4.1 hold. For example, modulation $[(1+2D+D^2)/4,\ h=1/n]$ appears to be a good candidate. The modulations studied seem fairly representative of schemes with frequency pulse lengths of two and three digit intervals. Other modulations may have close to the same pulse shapes but with more smoothing. The decoding approaches studied would then seem to be applicable to a broader class of constant envelope phase modulations [14]. The 3RC modulation is just a smoothed version of $[(1+2D+D^2)/4]$. The decoding approaches should carry over nicely to this modulation.

The phenomenon of false lock and long error events for $h \neq 1/n$ seems to be only a property of not considering all the possible states which MLSE does. Any reduced survivor approach should almost certainly encounter the same problem for these modulation indices.

There is one aspect of implementation complexity that has not been addressed. This is the acquisition of the matched filter outputs. The Viterbi algorithm knows ahead of time which filter outputs it will need for any given interval. Buffers may be used to store outputs from the filters for delayed processing.

The reduced computation approaches do not know ahead of time which filter outputs will be needed. These depend on the actual decision path taken. It may therefore be necessary to read in and store more filter outputs than are actually used in the computation in order to allow the same delayed processing. Waiting for A/D conversions is clearly not a viable option. The ultimate speed increase of the reduced computation approaches strongly depends on the acquisition-time of the necessary filter outputs. If direct memory access techniques (DMA) are used, acquisition time may not pose any speed constraints. In this case, the overall speed increase may be directly proportional to the computational reduction factor.

For those modulations where the fixed order processing is not viable, the 'best-S' approach can still be used. If there are more than just a few survivors necessary however, this approach may have no gain over full Viterbi decoding. The 'best-S' decoder will still suffer from the problem of long error events for modulations indices $h \neq 1/n$. It may be possible to find efficient ways of ordering the processing so that rule R2 can be used on these modulations.

Further research could focus on an efficient microprocessor implementation of the decoders presented here. The sensitivity of the decoders to phase jitter, timing errors and the other usual impairments could be studied. Further investigation is needed into ways of coping with the false lock problem for modulation indices $h \neq 1/n$.

It is very possible that the ordered processing of the decoders presented here could be utilized in reduced complexity decoders for linear modulations and codes.

REFERENCES

[1] S. Pasupathy, "Minimum Shift Keying: A spectrally efficient modulation", IEEE Communications Magazine, 1979, pp 14-22.

[2] M. K. Simon, "A generalization of minimum-shift-keying (MSK) - type signalling based upon input data symbol pulse shaping", IEEE Trans. on Comm., vol. COM-24, pp. 845 - 856, August 1976.

[3] T. Aulin and C. E. Sundberg, "M-ary CPFSK type of signalling with input data symbol pulse shaping - Minimum distance and spectrum", ICC'79 Conf Rec Boston, Mass, June 1979, pp. 42.3.1 - 42.3.6.

[4] G. S. Deshpande and P. H. Wittke, "Correlative encoded digital FM", IEEE Trans. on Comm., vol. COM-29, pp. 156 - 162, February 1981.

[5] J. B. Anderson, C. E. Sundberg, T. Aulin, and N. Rydbeck, "Smoothed phase modulation codes: Power vs bandwidth, NTC'80 Conf. Rec., Houston, Texas, December 1980, pp. 14.6.1 - 14.6.5.

[6] D. Muilwijk, "Correlative phase shift keying - A class of constant envelope modulation techniques, IEEE Trans on Comm., vol. COM-29, no.3, pp. 226 - 236, March 1981.

[7] T. Aulin, and C. E. Sundberg, "Continuous phase modulation - Part I: Full response signalling", IEEE Trans. on Comm., vol. COM-29, no. 3, pp. 196 - 209, March 1981.

[8] T. Aulin, N. Rydbeck and C. E. Sundberg, "Continuous phase modulation - Part II: Partial response signalling", IEEE Trans on Comm, vol. COM-29, no. 3, pp. 210 - 225, March 1981.

[9] H. Miyakawa, H. Harashima, Y. Tanaka, "A new digital modulation scheme - Multi-mode binary CPFSK", Proc. 3rd Int. Conf. Digital Satellite Comm., Kyoto, Japan, November 1975, pp. 105-112.

[10] J. B. Anderson and D. P. Taylor, "A bandwidth efficient class of signal space codes", IEEE Trans. on Info. Theory, vol. IT-24, no. 6, pp. 703 - 712, November 1978.

[11] R. Debuda, "Fast FSK signals and their demodulation", Can. Elect. Eng. J., vol. 1, no. 1, pp. 28 - 34, 1976.

[12] G. D. Forney, "The Viterbi algorithm", Proc. IEEE, vol. 61, pp. 268 - 278, March 1973.

[13] T. A. Schonoff, H. E. Nicholls, and H. M. Gibbons, "Use of the MLSE algorithm to demodulate CPFSK", ICC'78 Conf. Rec., Toronto, Canada, June 1978, pp. 25.4.1 - 25.4.5.

[14] T. Aulin, N. Rydbeck, and C. E. Sundberg, "Transmitter and receiver structures for M-ary partial response FM", 1980 Int. Zurich Seminar on Digital Comm., Zurich Switzerland, March 1980, pp. A2.1 - A2.6.

[15] T. Aulin, "Viterbi detection of continuous phase modulated signals", NTC'80 Conf. Rec., Houston, Texas, December 1980, pp. 14.2.1 - 14.2.7.

[16] J. B. Anderson, "Simulated error performance of multi-h phase codes", ICC'80 Conf. Rec, Seattle, Washington, June 1980, pp. 26.4.1 - 26.4.4.

[17] F. de Jager and C. B. Dekker, "Tamed Frequency Modulation, a novel method to achieve spectrum economy in digital transmission", IEEE Trans. on Comm., vol. COM-26, no. 5, pp. 534 - 542, May 1978.

[18] F. L. Vermuellen, "Low complexity decoders for channels with intersymbol interference", Ph.d. thesis, Info. Systems Lab., Stanford University, CA., August 1975.

[19] R. M. Fano, "A heuristic discussion of probabilistic decoding", IEEE Trans. on Info. Theory, vol. IT-9, pp. 64-74, April 1963.