CNR, INC.

AD A073479

DDC ACCESSION NUMBER

DATA SHEET

PHOTOGRAPH THIS SHEET

LEVEL: II

INVENTORY: 1

N00019-77-C-0190

Final Report - Nov 77

DOCUMENT IDENTIFICATION

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

Accession For
NTIS GRA&I [X]
DDC TAB [ ]
Unannounced [ ]
Justification

By

Distribution/

Availability Codes

Dist	Avail and/or special
A	

DISTRIBUTION STAMP

DDC
RECEIVED
SEP 4 1979
E

DATE ACCESSIONED

79 08 10 015

DATE RECEIVED IN DDC

PHOTOGRAPH THIS COPY

CNR, INC.
220 RESERVOIR STREET • NEEDHAM HEIGHTS, MASSACHUSETTS 02194 • (617) 449-4902

November 1977

FINAL REPORT

HF MODEM DESIGN FOR THE ADVANCED NARROWBAND DIGITAL VOICE TERMINAL (ANDVT)

Contract No. N00039-77-C-0190

Prepared for:

Department of the Navy
Naval Electronic System Command
Washington, D.C. 20360

Codes ELEX 3103, 3011, 904, 2601

and

Naval Research Laboratory
Washington, D.C. 20360

Code ~~5220~~ 7523 (3 Copies)

Prepared by:

CNR, Inc.
220 Reservoir Street
Needham, Massachusetts 02194

David Chase
Phillip A. Bello
Leslie W. Pickering
Robert W. Pinto
Lih-Jyh Weng
Ernest M. Bell

ABSTRACT

A major part of this study is the specification of a modem optimized for transmission of digitized speech over a HF channel. A tenth-order linear predictive coder (LPC-10), yielding a data rate of 2400 b/s, is used as the speech processor and a HF channel with an available bandwidth of 2.4 kHz is assumed. A modulation format which is frame synchronous to that of the speech processor allows coding to be applied to critical bits generated by the LPC-10 speech analyzer. In addition to the specification of modems for digitizing speech, an equally important part of this study is the development of new preambles which can operate at low signal-to-noise ratios in the presence of time-varying multipath profiles typical of HF channels. This task includes the specification of a waveform for signal presence detection, Doppler estimation, frame estimation, and the decoding of a KG synchronizing sequence. Complexity estimates, based on an implementation by a programmable processor, are obtained for the preamble and digitized speech modem specified in this study.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

SECTION 1

INTRODUCTION

A major part of this study is the specification of a modem optimized for transmission of digitized speech over a HF channel. A tenth-order linear predictive coder (LPC-10), yielding a data rate of 2400 b/s, is used as the speech processor and a HF channel with an available bandwidth of 2.4 kHz is assumed. A modulation format which is frame synchronous to that of the speech processor allows coding to be applied to critical bits generated by the LPC-10 speech analyzer. In addition to the specification of modems for digitizing speech, an equally important part of this study is the development of new preambles which can operate at low signal-to-noise ratios in the presence of time-varying multipath profiles typical of HF channels. This task includes the specification of a waveform for signal presence detection, Doppler estimation, frame estimation, and the decoding of a KG synchronizing sequence. Complexity estimates, based on an implementation by a programmable processor, are obtained for the preamble and digitized speech modem specified in this study.

An effective technique for signal presence detection and Doppler estimation, based on using four unkeyed parallel tones, is developed in Section 2. Frame detection is achieved by the use of Carley sequences composed of wideband pulses with raised cosine envelopes. The total time for this preamble is .449 second which is well within our design goal of 0.72 second. Also, successful operation is possible at an average signal-to-noise density ratio of 40 dB-Hz. This corresponds to a signal-to-noise ratio measured

in a 2.4 kHz bandwidth of only 6.2 dB. Appendix C discusses various definitions of SNR and develops simple relations between them.

Once Doppler and framing estimates are obtained a parallel tone format is used for the transmission of the KG sequences and the digitizing speech data. A new Doppler tracking algorithm, based on using all the keyed data tones is also developed in Appendix E and Section 5. Frame tracking is achieved by comparing energy levels in an early/late empty tone slot, as is done in conventional HF modems. It should be noted that this slot sync technique for tracking represents the only conventional technique used in this ANDVT modem. The inherent in-band diversity obtained from this tracking approach justifies its use.

In-band time and frequency diversity combining used in conjunction with a strong error correcting block is used in Section 3 for reception of the KG synchronization sequence. As an example, a combined KG sync sequence and net control sequence of up to 100 bits can be received by the use of a (250,100;46) block code with eighth-order in-band diversity. Assuming a 4-phase DPSK modem, with 39 tones keyed every 22.5 ms, a transmission time of 26 frames yields 1014 tones, of which 1000 are required for obtaining eighth-order diversity. If we allow one reference frame and an additional three frames for processing, the total time required for the reception of the KG sync sequence is .675 second (30 frames) which is well within the design goal of less than 1.53 seconds for the reception of this sequence. Successful reception of this preamble is obtained at a signal-to-noise density ratio of 35 dB-Hz even with just hard decoding. This number is quite compatible with the results for the preamble since the acceptable peak-to-average ratio for this 39 tone format should be about 5 dB greater than that required for the preamble.

In Section 4 specific modem designs for an LPC-10 speech processor are evaluated for several HF channel models. In addition to obtaining results in terms of bit error probabilities, speech tapes of the synthesizer speech after transmission over a HF channel are obtained. The major emphasis has been on comparing a conventicnal 16 tone channel, operating at a 75 f/s rate, to a 39 tone modem, operating at the same frame rate of the speech processor, i.e., 44.44 f/s. The 39 tone modem uses two (24,12;8) Golay codes to protect 24 critical bits in each frame of 54 bits. The performance results indicate quite clearly the superiority of the 39 tone modem over the more conventional 16 tone modem. Acceptable voice communication appears possible at an E_b/N_0 of around 11 dB, which corresponds to a signal-to-noise density ratio of around 45 dB-Hz. This number is several dB greater than required for the successful reception of preamble and KG sync sequence. This desired result indicates that the performance of the ANDVT modem will not be limited by signal acquisition requirements, as is the case for most conventional HF modems.

Some results have been obtained by using a 41 tone and a multiple rate coding approach which operates with a decoding delay of 10 frames (.225 second). The preliminary results to date indicate that the simpler Golay coding approach is superior to this multiple rate coding technique. However, further evaluation in this area is still required since multiple burst correction codes have not yet been evaluated, and the HF channel models have not yet been modified to include impulse noise.

The computational complexity of the modem operations is treated in Section 5 from the point of view of implementation on a programmable processor. The number of operations for each task has been determined for a model of a quite basic processor which certainly does not exceed the capabilities of the (undefined) Quintrell processors. Half-duplex operation appears well within the capabilities of the processor model assumed.

Several miscellaneous topics are discussed in Section 6. These include signal level optimization, a data transmission mode for the ANDVT, and a performance monitoring technique for this HF modem.

It is helpful to provide an example of a potential half-duplex format for the ANDVT which can be obtained from the detailed results given in the remainder of this report.

A four-tone preamble has been chosen for signal presence detection and Doppler offset estimation. The combined time required for signal presence detection and Doppler offset estimation is .359 second. After the Doppler offset is corrected, frame estimation is achieved by the use of wide band pulses to form an aperiodic sequence with good autocorrelation properties. A pair of Carley sequences with appropriate spacing is used for framing. A duration of .09 second (four frames at 22.5 ms per frame) is estimated as the time required for frame estimation. Thus the duration of the entire acquisition format is .449 second. Following the acquisition format, a parallel tone format with 39 tones, keyed at a rate of 44.44 f/s, can be used. The modulator is 4-phase differential phase shift keying (DPSK). The KG sync sequence and any net control information can be obtained in a time

period, including the processing for a (250,100;46) code, of .675 second. After the reception of the KG sync information, the 39 tone format may be used with two Golay codes to protect the critical LPC-10 bits.

Further studies and detailed simulations are expected to follow this initial effort. During this next phase more detailed results will be obtained and other formats in addition to that given on Fig. 1.1 will be evaluated.

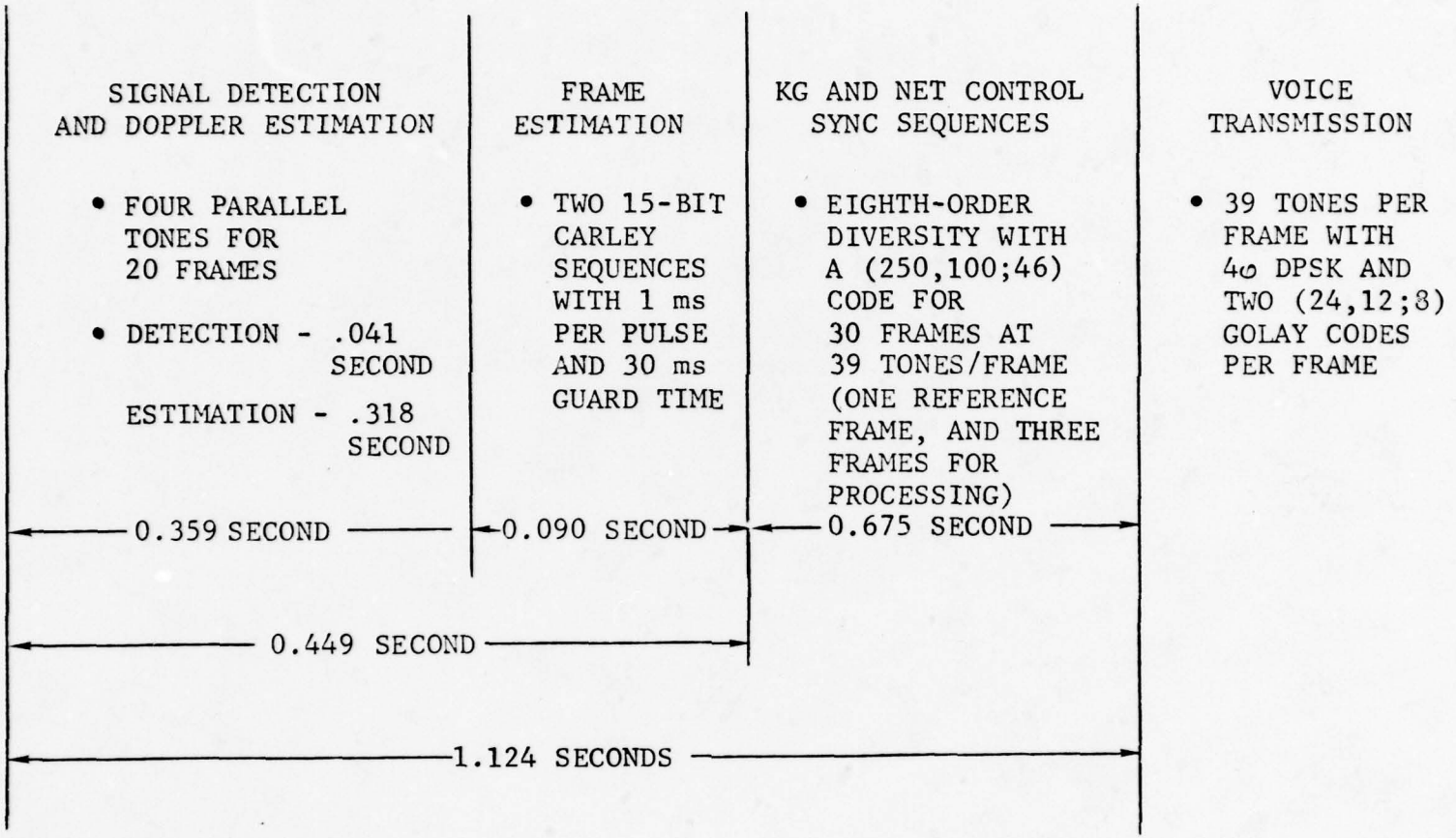


Figure 1.1 Example of a Potential Half-Duplex Format

SECTION 2

PREAMBLE DESIGN

2.1 Introduction

The preamble, provided at the beginning of the message just prior to the actual transmission of data, has three primary purposes.

1) To assert presence of the signal. This is done via comparative energy measurements made at the receiver. The operation is referred to as signal presence detection (see description in Section 2.2 and detailed analytical treatment in Section 2.4).

2) To make available to the receiver a signal that can be processed to obtain an estimate of Doppler shift and hence effect an initial Doppler correction. This operation is referred to as Doppler shift estimation (see overview description in Section 2.2 and detailed analytical treatment in Section 2.5).

3) To provide the receiver with a signal that can be processed in such a way as to initiate frame synchronization (see overview in Section 2.2 and detailed discussion in Section 2.6).

The sections to follow describe preliminary design specifications for the preamble and preamble processor at the receiver. Numerical values are provided for the preamble parameters that most seriously impact operation. Additionally, detailed descriptions of all processing operations are provided. The numerical values and procedures are summarized in Section 2.2 which is dedicated to a descriptive overview of the preamble; the design

guidelines, general structure, signal and processor timing, and specification of numerical values for relevant parameters are presented.

Subsequent sections describe the performance capabilities of the preamble and preamble processor. In particular, because part of the preamble consists of parallel tones, the problem of reducing the crest factor via phasing of the tones is discussed in Section 2.3.

The signal presence detector is described and analyzed in detail in Section 2.4. Similarly, performance of the Doppler estimation stage is analyzed in Section 2.5, and the sync preamble is analyzed in Section 2.6.

2.2 Description of Preamble and Processor

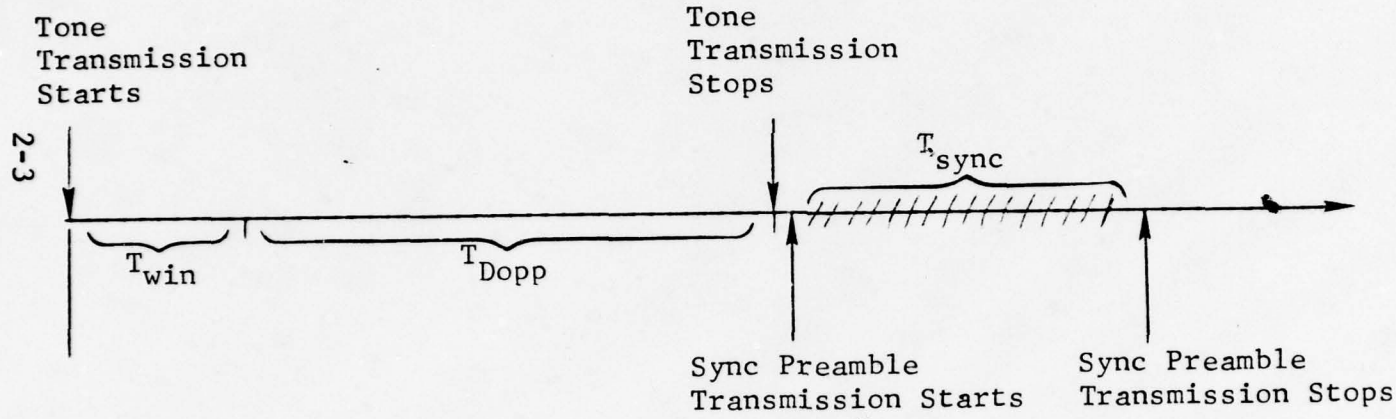
2.2.1 General Structure

The total time allotted to the preamble is divided into three sections corresponding to the three major tasks of the preamble. Similarly, the tasks are performed sequentially at the receiver. In the order performed, these tasks are:

1) Detection of signal presence
2) Doppler estimation
3) Frame sync estimation.

The general timing and nomenclature of the transmitted preamble are illustrated in Fig. 2.1.* Note that the same signal structure

*As prespecified, the preamble has been designed so that its overall length is no more than .72 second (32 frames at 22.5 ms/frame).



T_{win} ≡ Time Allotted to Detection (Length of Detection "Window")

T_{Dopp} ≡ Time Allotted to Doppler Processing

T_{sync} ≡ Duration of Sync Preamble

Figure 2.1 Timing of Transmitted Preamble

(four tones separated one from the other by approximately 600 Hz) is used for the detection preamble as for the Doppler section of the preamble. The tones are "on" for $T_{win} + T_{Dopp}$ seconds, T_{win} seconds allotted to detection of signal presence, T_{Dopp} seconds allotted to estimation of Doppler shift.

Because of dynamic limitations of equipment it is desirable to limit the crest factor (peak-to-average power ratio) of the four-tone composite signal. This can be achieved easily by ± 1 multipliers. Specifically, a phasing sequence $\{0,0,0,\pi\}$ on the tones yields a crest factor of 1.77. If no consideration is given to simplicity of implementation the "best" tone phases that were determined via numerical procedures described in [2.1] can be used. The sequence given there, $\{0,1.81,1.81,0\}$, results in a crest factor of 1.525. Specific details are provided in Section 2.3.

Processor timing is illustrated in Fig. 2.2. Note that sync processing is not allowed to commence while the tones are "on"; hence, there is a waiting period before sync processing begins. The specifics are clear from Fig. 2.2. The first line shows the preamble timing. The second and third lines show the timing of the preamble processor in two extreme cases of interest that are respectively, Case I (detection in zero time)* and Case II (detection in full allotted time). The most probable case, of course, is that detection will occur with some intermediate number of samples.

---

*This is an unattainable extreme, of course; the closest one can get to this extreme is through a threshold exceedance being obtained with the collection of a single sample.

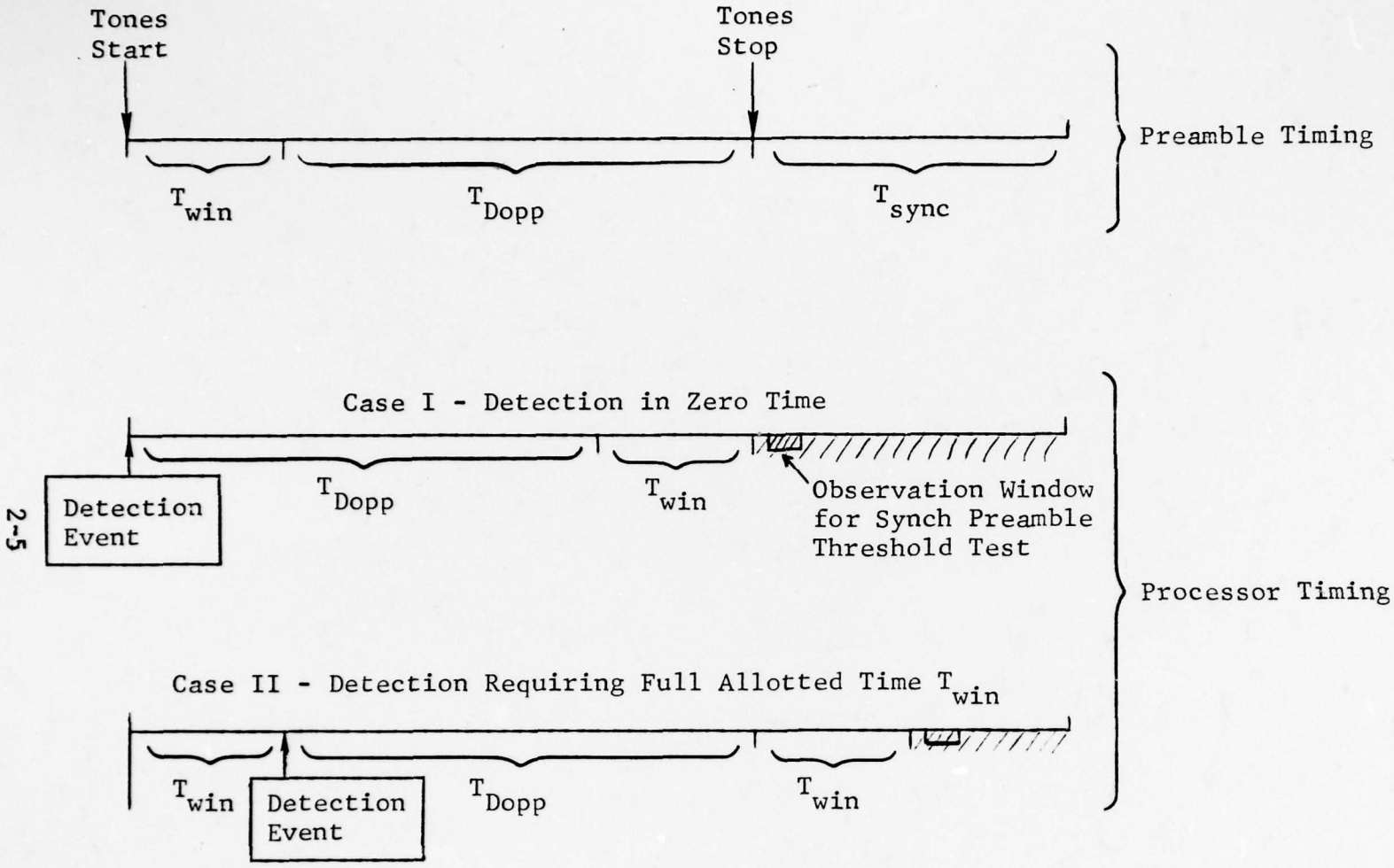


Figure 2.2 Processor Timing

It can be seen from the figure that the preamble timing is to a certain extent dominated by the requirement that the sync processing not start until the tone transmission has ended. From consideration of Case I, this requires a wait of T_{win} seconds after the Doppler processing has been completed. The same wait, in the event the individual detection event requires the full allotted time (Case II), means that the processor will have waited T_{win} seconds after the tone transmission has ended. Thus, transmission of the signals required for frame synchronization must continue for at least T_{win} seconds after the tone transmission has ended.

The problems relating to the detection of signal presence, Doppler estimation, and frame synch estimation are analyzed and described in detail in Sections 2.4, 2.5, and 2.6, respectively. The design implications of those extended developments are summarized respectively in Sections 2.2.2, 2.2.3, and 2.2.4 below.

2.2.2 Detection of Signal Presence

The four-tone structure of the preamble segment that is used for both detection and Doppler estimation was described above in Section 2.2.1.

In the receiver, a filter is centered at each of the four tone frequencies. This filtering operation can be done by frequency shifting the input separately so that each tone frequency appears at d.c. and uses a complex low pass filter. If recursive, this low pass filter must be second-order, at least, and a double pole is sufficient. Since the a priori range of Doppler shifts is ± 75 Hz, we specify a double-tuned filter

$$H(f) = \frac{1}{(1 + j2\pi f t_c)^2} \qquad (2.1)$$

where t_c is the time constant of each stage.* We additionally specify a filter half-power frequency of 75 Hz. Generally, when the input to the filter in (2.1) is white noise, it can be determined that sampling of the output every $\Delta = 3t_c$ seconds results in adjacent samples that have a correlation coefficient of .2 (see Section 2.5). The constraint on the half-power frequency results in a numerical specification of the value of Δ (see Section 2.5). This value is

$$\Delta = 4.1 \text{ ms} \qquad (2.2)$$

In order to detect signal presence, the filter output samples are squared and added and fed to a filter which averages over several successive samples. We call this latter filter the signal energy filter. Signal presence is denoted by the fact that this filter output exceeds a threshold. Because of the action of the radio's AGC, which is not under control of the modem, the total signal-plus-noise power appearing at the modem input will not vary much. Thus, a fixed threshold cannot be used in deciding signal presence. It is necessary to have other filters (we specify four) centered where no tones are transmitted to estimate noise level. The same computations are made as for the signal filters with the summation of squares and a low pass averaging filter. This latter filter is called the noise energy filter.

*Time has not permitted an extensive comparison of filter shapes.

The configuration of the signal presence detector is shown in Fig. 2.3. If we denote $S_k(\ell\Delta)$ as the ℓ'th time sample at the output of the k'th tone filter and $N_k(\ell\Delta)$ as the ℓ'th time sample at the output of the k'th noise filter the threshold test is given by

$$\frac{\sum_{\ell=1}^{N}\sum_{k=1}^{4}|S_k(\ell\Delta)|^2}{\sum_{\ell=1}^{N}\sum_{k=1}^{4}|N_k(\ell\Delta)|^2} \gtrless \alpha \qquad (2.3)$$

with threshold exceedance (>) resulting in a detection event. The operation is implemented as a moving window; continuous sampling of the arriving signal (or noise) energy takes place with only the N most recent time samples being used in the threshold test. If no signal is present and the N most recent time sampling instants provide samples that result in threshold exceedance a false alarm event takes place. If the same result is obtained when a signal <u>is</u> present a detection event takes place. This latter event may take place even though the tones are "on" for considerably less than N x Δ seconds. This is the cause of the fundamental uncertainty of signal arrival time discussed in the previous section.

To relate to the discussion of the previous section we note that

$$T_{win} = N \times \Delta \qquad (2.4)$$

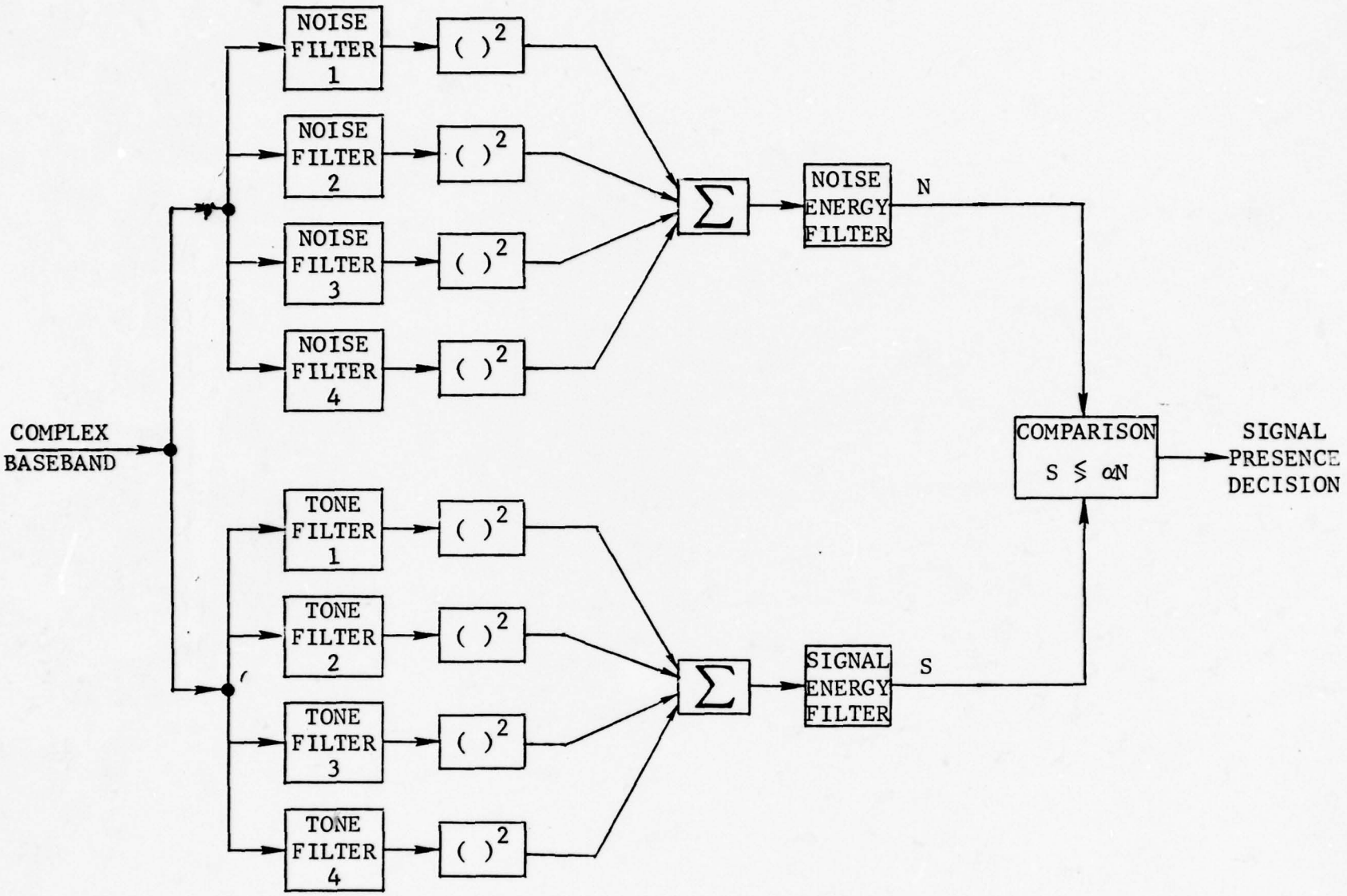


Figure 2.3 Signal Presence Detection

An extensive analysis of the signal presence detector, accompanied by performance curves, is provided in Section 2.4. As a tentative design goal we have chosen the bounds

$$P_{FA} \leq 10^{-5}$$
$$P_{ID} \leq 10^{-2} \tag{2.5}$$

where P_{FA} is the probability of false alarm and P_{ID} is the probability of incorrect dismissal. The specified performance levels are to be achieved with a signal-to-noise density ratio of

$$\frac{P}{N_0} = 40 \text{ dB-Hz} \tag{2.6}$$

where P is the total power in the four-tone signal and N_0 is the single-sided noise power density. The results are based on the assumption of independently fluctuating tones.*

As can be determined from the results of Section 2.5, the design specifications above are met with the numerical parameter values itemized in Table 2-1.

2.2.3 Doppler Shift Estimator

The composite four-tone signal used for detection of signal presence is used, immediately following the detection event, for estimation of Doppler shift. The four tone filters used previously are used unchanged as the initial stage of a three-stage adaptive Doppler estimator that utilizes the "current" estimate

*The validity of this assumption is questioned and discussed in Section 2.4.

TABLE 2-1

SIGNAL DETECTOR DESIGN PARAMETERS

Signal-to-Noise Density Ratio:

$$\frac{P}{N_0} = 40 \text{ dB-Hz}$$

Number of Time Samples:

$$N = 10$$

Corresponding Duration of Detection Window:

$$T_{win} = .041 \text{ second}$$

Achieved Performance:

$$P_{FA} = 10^{-5}$$

$$P_{ID} = .25 \times 10^{-2}$$

Threshold Setting:

$$\alpha = 2.65$$

of Doppler shift to effect a Doppler correction and a narrowing of the tone-filter bandwidth; this allows (via admission of less noise) an improvement in the subsequent estimate.

The algorithm for one-stage Doppler estimation is illustrated in Fig. 2.4.* Note that N in this context represents the number of times that the output of each tone filter is sampled. This number, as we show below, typically varies from stage to stage.

An extensive discussion of the single-stage Doppler estimator illustrated in Fig. 2.4 is provided in Section 2.5.1. Performance results are also provided in the form of an analytical expression for σ, the rms error in estimating the Doppler shift.

Multi-stage estimation of Doppler shift is discussed in Section 2.5.2 and performance is determined via a worst-case analysis described in that section.

The basic design goals and preliminary specifications of numerical parameters for the estimator are provided in Table 2-2.

Summarizing the operation of the three-stage estimator we note that 16 samples of the output of the originally configured tone filters (half-power frequency = 75 Hz) are used to make an initial estimate of Doppler shift. This estimate is used to effect a Doppler correction and the tone filter bandwidths are narrowed to a half-power frequency of 16.2 Hz. The outputs of the adapted filter [same basic structure given in (2.1)] are now

---

*Questions relative to estimation of the derivatives $\dot{x}_1$, $\dot{y}_1$, etc. are discussed in Section 2.5

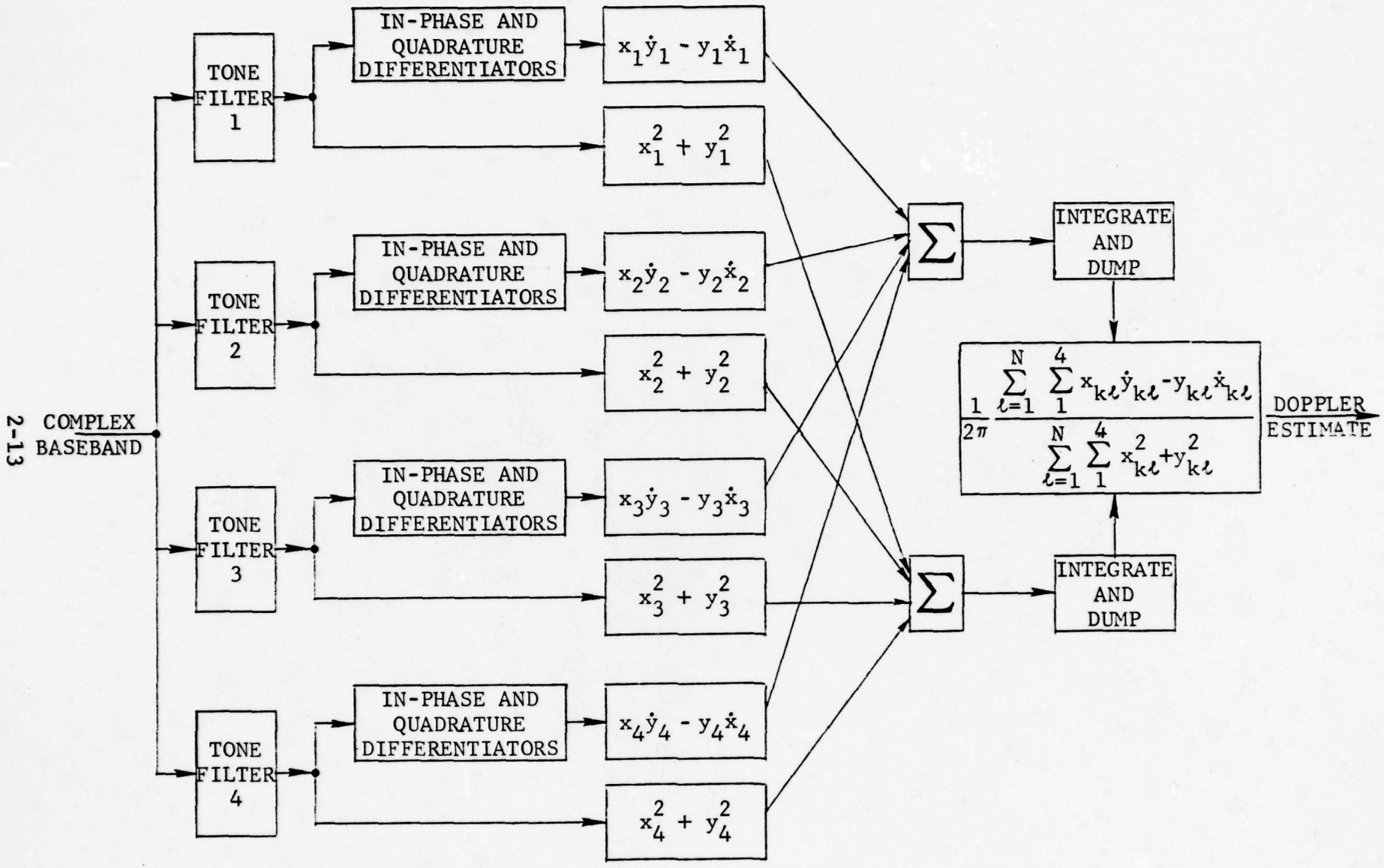


Figure 2.4 Illustration of a Doppler Estimation Technique

TABLE 2-2

DOPPLER ESTIMATOR DESIGN PARAMETERS

Signal-to-Noise Density Ratio:

$$\frac{P}{N_0} = 40 \text{ dB-Hz}$$

Design Goal for RMS Doppler Shift Estimation Error:

$$\sigma \leq .25 \text{ Hz}$$

Stage #1: N = 16 Δ = 4.1 ms ($f_{1/2}$ = 75 Hz)

Stage #2: N = 3 Δ = 19.0 ms ($f_{1/2}$ = 16.2 Hz)

Stage #3: N = 1 Δ = 88.2 ms ($f_{1/2}$ = 3.5 Hz)

Performance Achieved:

$$\sigma = .234 \text{ Hz}$$

Total Processing Time (Including Transient Build-Up):

$$T_{tot} = .318 \text{ second}$$

N ≡ Number of samples taken per stage

Δ ≡ Time between samples used in computation

$f_{1/2}$ ≡ Half-power frequency of adaptive filter

sampled three times at the new rate. The basic adaptation process described above is repeated resulting in tone filters narrowed to a 3.5 Hz half-power frequency. The outputs are sampled once in the final stage.

Note that the design goal for Doppler shift rms estimation error is $\sigma < .25$ Hz. It would be illusory to attempt to estimate Doppler to a resolution much finer than .25 Hz because of the occurrence of Doppler spread on the channel. Available measurements of HF channel Doppler spread [2.2] indicate that 20% of the measurements (expressed as a sigma width of the Doppler power spectrum) exceed this value.*

2.2.4 Frame Synch Estimation

In this section we describe the detailed structure of the frame synchronization preamble occupying the T_{synch} second segment of the preamble illustrated in Figs. 2.1 and 2.2.

The sync preamble consists of a pulse train designed both for high crest factor and low autocorrelation sidelobes. The pulse shape used in this pulse train is assumed to be of the Nyquist type, having normalized value unity at t=0 and zero value at time samples equal to multiples of Δ. Another requirement is that the pulse be bandlimited so that its spectrum lie within the flat portion of the HF terminal equipment. Further details and considerations are provided in Section 2.6. Because time did not permit a detailed comparison of various pulse shapes, we tentatively propose a pulse with a raised cosine spectrum of bandwidth $2/\Delta$. This satisfies the Nyquist requirement. The low pass spectrum is illustrated later in Fig. 2.17.

---

*Statistical measurements of Doppler spread are very limited. The measurements referred to were taken on a Fort Monmouth, New Jersey to Palo Alto, California path.

The remainder of this section relates to specifics of the pulse train timing and the discrete sequence used for synchronization.

In a general context the sync preamble timing is illustrated in Fig. 2.5. The length-K_1 segment of the preamble will be referred to as the correlation sequence. This discrete sequence is chosen both for high crest factor and low autocorrelation sidelobes. The specific choice of a length-15 Carley sequence (peak-to-sidelobe ratio = 15) is recommended for reasons described below in greater detail in Section 2.6. As shown in Fig. 2.5 the correlation sequence is repeated periodically with period K_2 ($K_2 \geq 2K_1 - 1$).*

The second line of Fig. 2.5 illustrates the timing of the output of the matched filter illustrated in Fig. 2.6. The correlation peaks and sidelobes (non-overlapping) are seen to repeat every K_2 samples. Note that an observation window of length K_2 will include one correlation peak.

For our initial recommendation we simplify the processor illustrated in Fig. 2.6 by eliminating the centroid calculation and replacing it with an operation that selects the largest peak which is then used to determine frame synch.

---

*Satisfaction of this inequality is required to guarantee that the output of the discrete matched filter consists ideally of repeated segments of the a periodic correlation function without overlap of samples, (see Section 2.6).

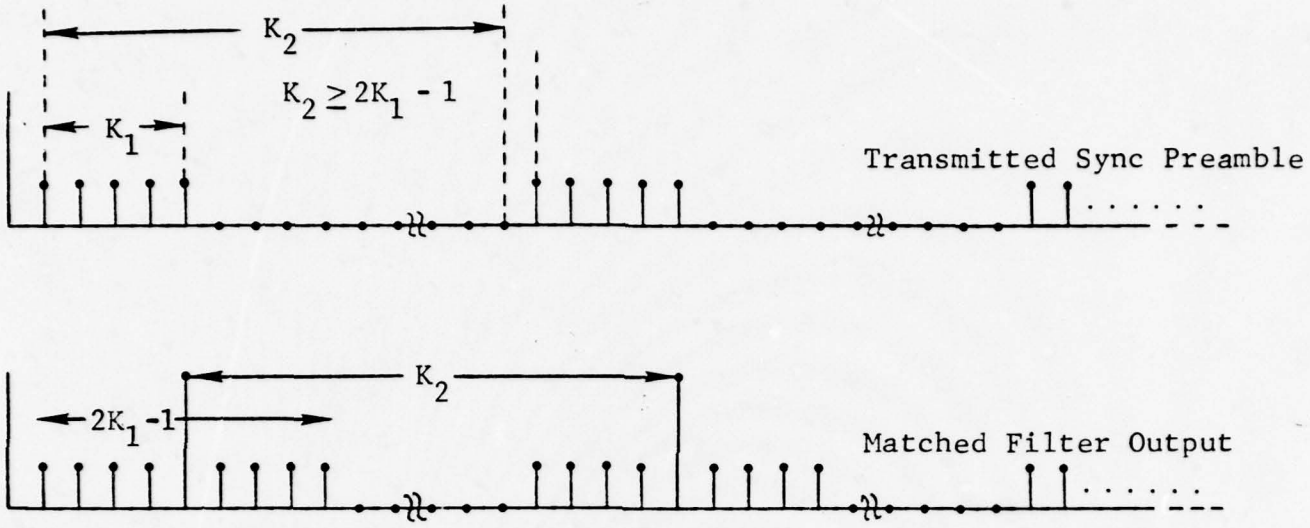


Figure 2.5 Transmitted Sync Preamble and Matched Filter Output (Ideal Channel With Zero Delay)

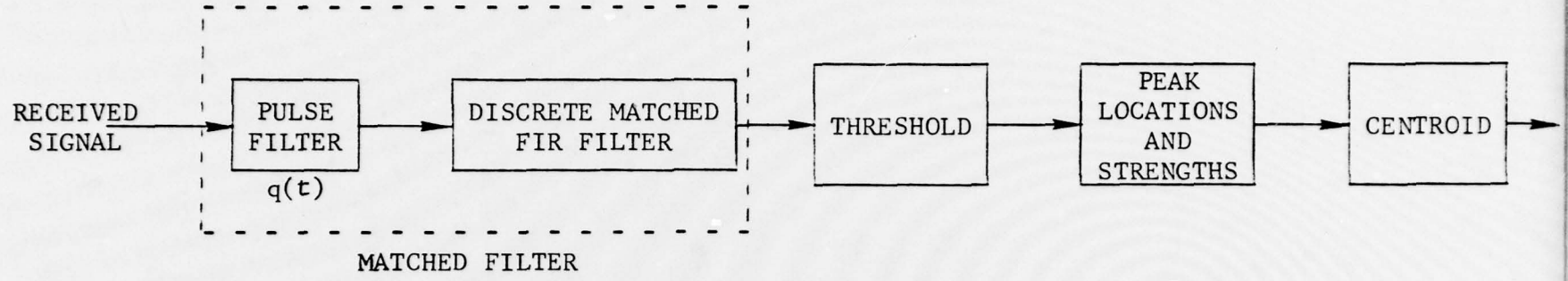


Figure 2.6 Sync Preamble Processing

The sequence of events defining the threshold test are as follows: Subsequent to the wait of T_{win} seconds after Doppler processing, K_2 samples of the matched filter output are observed. The receiver then formulates the decision variable $|S(t_i)|^2/\hat{\Gamma}$ $i = 1, K_2$, where $S(t)$ represents the received signal and $\hat{\Gamma}$ is an estimate of the reference noise level Γ given in (2.197).* Each of the K_2 samples are tested against a threshold ρ_0. The largest peak exceeding the threshold is chosen as the correlation peak to be used for synchronization.

It is helpful to itemize some features of this procedure:

1) Any one of the correlation peaks in the second line of Fig. 2.5 is suitable for initiation of clock signal provided the spacing between spikes is equal to an integer multiple of frame durations.

2) Consideration of Fig. 2.2 indicates that the processing time (subsequent to Doppler processing and including the T_{win} second "wait") is $T_{win} + (K_2 \times \Delta) + \delta_0$, where δ_0 is a safety margin (to allow for transients)that is small enough not to warrant concern in these preliminary general considerations.

3) The actual observation (search for the correlation peak) is made over $K_2 \times \Delta$ seconds subsequent to the waiting period. From Fig. 2.5 it is clear that observation over K_2 samples will yield one correlation peak in the matched filter output.

---

*Time limitations did not permit an evaluation of the accuracy to which Γ can be estimated. All indications point, however, to the conclusion that this quantity can easily be estimated to sufficient accuracy to have negligible impact on our conclusions of frame sync performance. See discussion in Section 2.6.2.2.

4) Because of periodic repetition n times the sync preamble is of total duration n x (K_2 x Δ) and (n-1) x (K_2 x Δ) must exceed, or at least equal, the waiting time T_{win}. This requirement is clear from consideration of Case II in Fig. 2.2 and guarantees, when detection requires the full alotted time T_{win}, that there will be at least K_2 samples of the sync preamble available subsequent to the "wait" of T_{win} seconds, i.e., there will be sufficient time available for observation of one fully developed spike in the matched filter output.

From the discussion above we see that, though the total sync preamble is long due to the n periodic repetitions of the basic probing sequence, the observation interval is shorter by a factor of n. Reduction of the observation interval, in accord with the discussion of Section 2.6, tends to minimize the effects of noise.

Considerations discussed above, apart from guard times for transients, indicate that data transmission can begin shortly after the end of the sync preamble. The requirement that the periodic sync preamble be synchronous with the data is satisfied if

$$K_2\Delta = mT_F \tag{2.7}$$

where m is an integer and T_F is the frame duration which is already specified to be 22.5 ms.

With the choice of a Carley sequence we have $K_1 = 15$ (see Fig. 2.3 and discussion in Section 2.6). With the further choice m = 2 and Δ = 1 ms we have $K_2 = 45$ which satisfies the requirement $K_2 \geq 2K_1 - 1$. For the detection window of duration $T_{win} = 41$ ms, item 4) above requires that $(n - 1) \times 45 \geq 41$ which is satisfied for n = 2 yielding a sync preamble of duration 2 x 45 = 90 ms in length. Note that this also satisfies the requirement that at least $K_2 = 45$ matched filter samples be available for processing after the $T_{win} = 41$ ms "wait".

Thus, the total duration of the sync preamble is given by

$$\text{Total Duration of Sync Preamble} = .09 \text{ sec} \qquad (2.8)$$

The above considerations and consideration of Fig. 2.5 indicate that data transmitted 15 ms after the last non-zero pulse in the sync preamble will be frame synchronous with the clock signal initiated by the output of the discrete matched filter.

To identify the <u>initial</u> frame boundary requires that the correlation peak be identified as being due to the first or second period of the correlation sequence. The ambiguity can be removed by processing for the second correlation

peak; this secondary processing can take place by threshold testing a very small number of matched filter outputs (ideally just one) because of the a-priori fixed time interval between the peaks.

The length of the observation interval is $K_2 \times \Delta =$ 45 ms. The length of this interval has importance relative to the effects of noise as discussed in Section 2.6. The numerical parameters describing the sync preamble and processor, as determined from the considerations of Section 2.6, are summarized in Table 2-3.

To be specific we have described just one way of implementing the sync preamble timing. There are a variety of different approaches one could use. As an example one can conceive of signal processing aimed at detecting the end of the tone transmission. This could considerably reduce the forementioned timing uncertainties and to some degree perform the function of the sync preamble itself.

2.3 Crest Factor Reduction

The parallel tones that constitute the Doppler preamble can be represented mathematically in terms of the complex envelope D(t). Since the N tones are harmonically related we have

TABLE 2-3

FRAME SYNC ESTIMATOR DESIGN PARAMETERS

Signal-to-Noise Density Ratio:

$$\frac{P}{N_0} = 40 \text{ dB-Hz}$$

Sampling Interval:

$\Delta = 1$ ms

Synchronization Sequence:

Length-15 Carley Sequence Repeated for Two Periods with Period $K_2 = 45$

Duration of Sync Preamble (Ignoring Transients):

.09 second

Correlation Peak Threshold Setting:

$\rho_0 = 15$

Start of Data Transmission:

15 ms After Last Sync Pulse

$$D(t) = \sum_{n=1}^{N} e^{j\theta_n} e^{j2\pi nFt} \tag{2.9}$$

The set of phases θ_n must be chosen to reduce the crest factor (peak/average power) of the resultant transmitted waveform.

The squared envelope of D(t) is given by

$$|D(t)|^2 = \sum_{n=1}^{N} \sum_{m=1}^{N} e^{j\theta_n} e^{-j\theta_m} e^{j2\pi(n-m)Ft}$$

$$= \sum_{p=-(N-1)}^{+(N-1)} R_p e^{j2\pi pFt} \tag{2.10}$$

where

$$R_p = \begin{cases} \sum_{m=1}^{N-p} e^{j\theta_{m+p}} e^{-j\theta_m} & p \geq 0 \\ \sum_{m=1-p}^{N} e^{j\theta_{m+p}} e^{-j\theta_m} & p < 0 \end{cases} \tag{2.11}$$

Note that R_p is the autocorrelation function of the phase weighting sequence. If this autocorrelation function had zero sidelobes, the crest factor would be unity.

The case of interest for us is N=4. A reasonable simple estimate for the desired phasing sequence can be generated by using ± 1 for the set $\{e^{j\theta_n}\}$ (see Frank [2.3])

$$\{\theta_n\} = \{0, 0, 0, \pi\} \tag{2.12}$$

The correlation function for this sequence is given by

$$\{R_p\} = \{-1, 0, 1, 4, 1, 0, -1\} \tag{2.13}$$

and the crest factor ρ which is defined

$$\rho \equiv \frac{\text{peak } |D(t)|^2}{\text{average } |D(t)|^2}$$

$$= \text{peak} \left(\frac{|D(t)|^2}{R_0} \right) \tag{2.14}$$

is given by

$$\rho = \text{peak } \{1 + \frac{1}{2} (\cos 2\pi Ft - \cos 6\pi Ft)\} \tag{2.15}$$

After some manipulation it can be demonstrated that the peak occurs at

$$2\pi Ft = \sin^{-1} \sqrt{\frac{2}{3}} \tag{2.16}$$

so that

$$\rho = 1.77 \tag{2.17}$$

A smaller crest factor can be obtained by using the "best phases discovered" reported by Nuttall [2.1]. As determined by numerical search techniques the tone phases are given by

$$\theta_n = \{0, 1.81, 1.81, 0\} \tag{2.18}$$

and result in a crest factor value

$$\rho = 1.525 \tag{2.19}$$

The resulting waveform envelope is shown in Fig. 2.7 (taken from [2.1]).

Though, to be specific, we recommend use of the second sequence, its use represents a gain of .64 dB over use of the first; in view of the simplicity of the first sequence (the last tone is simply reversed in polarity relative to the first three) consideration of its use should not be dropped completely.

2.4 Performance of Signal Presence Detector

The structure of the signal presence detector was presented earlier in Fig. 2.3. It should be recalled that the decision variable was chosen to be the ratio of tone filter energy output to noise filter energy output in order to avoid AGC effects.

When tones are being received, the output of the k'th tone filter $w_k(t)$ may be represented as the sum of signal and noise

$$w_k(t) = z_k(t) + n_k(t) \tag{2.20}$$

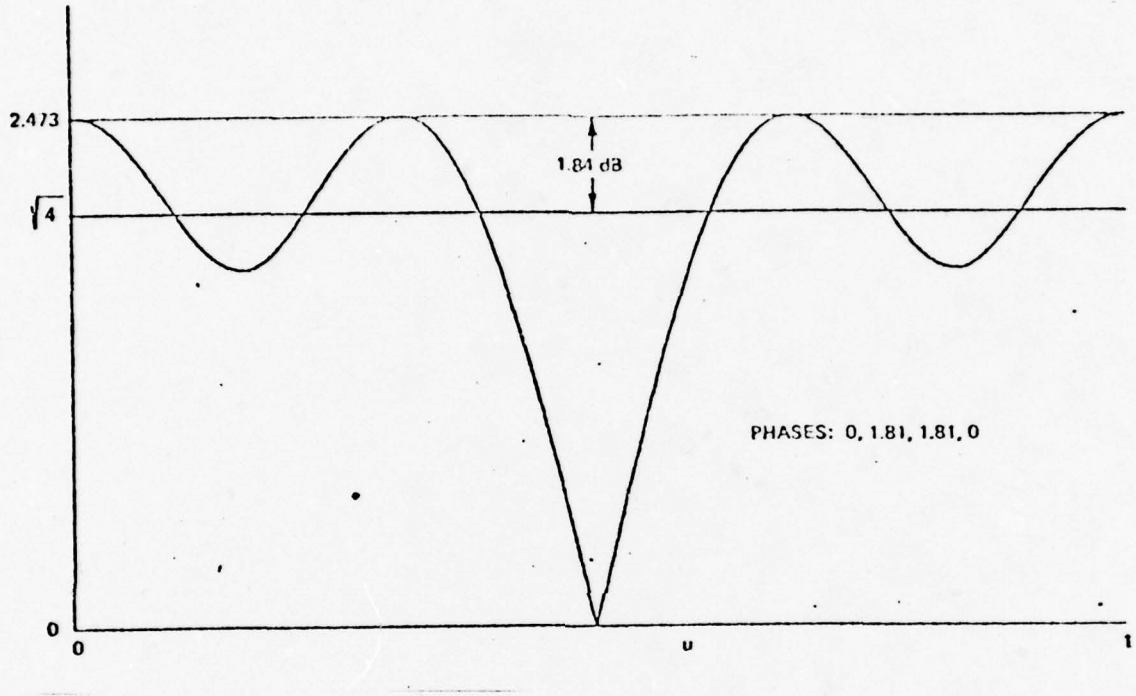


Figure 2.7 Waveform Envelope of Signal (Over One Period) Using Best Phases Discovered for Four Tones (Taken From Reference [2.1])

where $z_k(t)$ represents the received tone, and $n_k(t)$ is the noise at the tone filter output. At the noise filter output we have noise only [designated $\eta_k(t)$].

The threshold test is given by

$$\frac{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|w_k(\ell T)|^2}{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|\eta_k(\ell T)|^2} \underset{H_0}{\overset{H_1}{\gtrless}} \alpha \tag{2.21}$$

where H_1 is the hypothesis "signal present", H_0 is the hypothesis "signal absent", and α is a preset threshold. The summation in the numerator is over the K tone filters and N time samples, whereas the summation in the denominator is over the K noise filters and N time samples.

From (2.21) the probability of false alarm is given by

$$P_{FA} = \Pr\left[\frac{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|n_k(\ell T)|^2}{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|\eta_k(\ell T)|^2} > \alpha\right] \tag{2.22}$$

while the probability of incorrect dismissal is

$$P_{ID} = Pr\left[\frac{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|z_k(\ell T) + n_k(\ell T)|^2}{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|\eta_k(\ell T)|^2} < \alpha\right] \tag{2.23}$$

To calculate these quantities we first treat the signal as a deterministic quantity, i.e., we evaluate the probabilities P_{FA} and P_{ID} conditional on the set of values $z_k(\ell T)$. Some simplification is obtainable since we can assume, by design, that our sample interval exceeds the correlation time of the noise, i.e., the noise sample covariance takes the form

$$\overline{|n_k(\ell T)|^2} = \overline{|\eta_k(\ell T)|^2} = P_N \tag{2.24}$$

$$\overline{\eta_k^*(\ell T)\eta_s(rT)} = \overline{n_k^*(\ell T)n_s(rT)} = 0 \quad ; \quad k \neq s \tag{2.25}$$

$$\overline{\eta_k^*(\ell T)n_s(rT)} = 0 \tag{2.26}$$

These equations say simply that the noise samples from any two distinct filters are always uncorrelated but of equal energy. Coupled with the assumption of complex Gaussian statistics for the noise, we have shown that the ratio in (2.22) is a special case of the central F distribution while with $z_k(\ell T)$ fixed the ratio in (2.23) is a special case of the non-central F distribution. The specific forms of interest to us can be found in [2.4, pp. 9 and 13]. For the probability of false alarm, we get

$$P_{FA} = \frac{(2KN-1)!}{[(KN-1)!]^2} \sum_{j=0}^{KN-1} \binom{KN-1}{j} \frac{(-1)^j}{(KN+j)} \left(\frac{1}{1+\alpha}\right)^{KN+j} \tag{2.27}$$

and incorrect dismissal probability is given by

$$P_{ID}(\varepsilon) = \left(\frac{\alpha}{1+\alpha}\right)^{KN} \exp\left[\frac{-\lambda}{2(1+\alpha)}\right] \sum_{j=0}^{KN-1} \sum_{\ell=j}^{KN-1} \frac{1}{j!2^j}$$

$$\binom{KN+\ell-1}{\ell-j} \left(\frac{\alpha\lambda}{(1+\alpha)}\right)^j \left(\frac{1}{1+\alpha}\right)^{\ell} \tag{2.28}$$

where we have defined the signal energy term

$$\lambda = \frac{1}{\sigma^2} \sum_{\ell=1}^{N} \sum_{k=1}^{K} |z_k(\ell T)|^2 \tag{2.29}$$

where σ^2 is the noise variance $\sigma^2 = \overline{|n|^2}$. The probability in (2.28), of course, is a conditional probability conditioned on λ. Thus, we concentrate on computing the density function of λ and averaging $P_{ID}(\varepsilon)$ with respect to λ to obtain P_{ID}. With our assumption that the tone does not fluctuate during the signal detection interval,

$$\lambda = \frac{N}{\sigma^2} \sum_{k=1}^{K} |z_k|^2 \tag{2.30}$$

where we have dropped the argument (ℓT). Assuming that the $\{z_k\}$ are complex Gaussian, (2.30) represents a quadratic form in complex Gaussian variables. The characteristic and density functions of quadratic forms in complex variables have been derived by Turin [2.5].

We denote the density and characteristic functions of λ by $W(\lambda)$ and

$$F(jt) = \overline{e^{jt\lambda}} = \int e^{jt\lambda} W(\lambda)\, d\lambda \ , \tag{2.31}$$

respectively, In carrying out the average of (2.28) over λ, we note that the following typical average occurs:

$$I_r = \int \lambda^r \exp\left\{-\frac{\lambda}{2(1+\alpha)}\right\} W(\lambda)\, d\lambda \tag{2.32}$$

Comparing (2.31) and (2.32), we see immediately that

$$I_r = \left.\frac{\partial^r F(jt)}{\partial (jt)^r}\right|_{jt = -1/2(1+\alpha)} \equiv F^{(r)}\left(-\frac{1}{2(1+\alpha)}\right) \tag{2.33}$$

Then, using (2.33) in the average over (2.28), we see that

$$P_{ID} = \left(\frac{\alpha}{1+\alpha}\right)^{KN} \sum_{j=0}^{KN-1} \sum_{\ell=j}^{KN-1} \frac{1}{j!\,2^j} \binom{KN+\ell-1}{\ell-j} \left(\frac{\alpha}{1+\alpha}\right)^j \left(\frac{1}{1+\alpha}\right)^{\ell} F^{(j)}\left[-\frac{1}{2(1+\alpha)}\right] \tag{2.34}$$

For a generalized random variable with characteristic function F(jt) this is the general result. In our application, however, $Z_k(t)$ is assumed to be a complex Gaussian random variable. Thus, λ in (2.29) is a quadratic form in complex Gaussian random variables. From the work of Turin [2.5], the characteristic function of a quadratic form in complex Gaussian variables is given by

$$F(jt) = \frac{1}{\det [I - jtMQ]} \tag{2.35}$$

where Q is the matrix of the quadratic form q

$$q = \underset{\sim}{Z}^H \underset{\sim}{Q} \underset{\sim}{Z} \tag{2.36}$$

(the superscript H denotes conjugate transpose), Z is a column vector of complex Gaussian variables

$$\underset{\sim}{Z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix} , \tag{2.37}$$

and M is the moment matrix of the complex Gaussian variables

$$M = \overline{(\underset{\sim}{Z} - \overline{\underset{\sim}{Z}})(\underset{\sim}{Z} - \overline{\underset{\sim}{Z}})^H} \tag{2.38}$$

The elements of the matrix Q follow immediately from (2.30). They are

$$Q_{ij} = \frac{N}{\sigma^2} \delta_{ij} \tag{2.39}$$

To determine the moment matrix we assume that the HF channel is characterized by a discrete set of M independently-fluctuating paths having delays ξ_m, Doppler shifts ν_m, and complex amplitudes $g_m(t)$. The transfer function corresponding to this channel is

$$T(f,t) = \sum_{m=1}^{M} g_m(t) e^{-j2\pi f \xi_m} e^{j2\pi \nu_m t} \tag{2.40}$$

Consequently, the complex correlation between the k'th and ℓ'th received tone is just

$$\overline{z_\ell(t) z_k^*(t)} = \sum_{m=1}^{M} G_m e^{j2\pi (f_k - f_\ell)\xi_m} = S(f_k - f_\ell) \tag{2.41}$$

where f_k is the frequency location of the k'th tone,

$$G_m = \overline{|g_m(t)|^2} \tag{2.42}$$

is the strength of the m'th path, and S(f) is the frequency correlation function of the channel,

$$S(f) = \sum_{m=1}^{M} G_m e^{j2\pi f \xi_m} \tag{2.43}$$

Thus, the elements of the moment matrix M are determinable from knowledge of the frequency correlation function of the channel. If it can be assumed that the received tones fluctuate independently, the above formulation can be greatly simplified; the moment matrix then becomes diagonal

$$M = P_0 I \tag{2.44}$$

where I is the identity matrix and P_0 is the average received tone strength. In such a case,

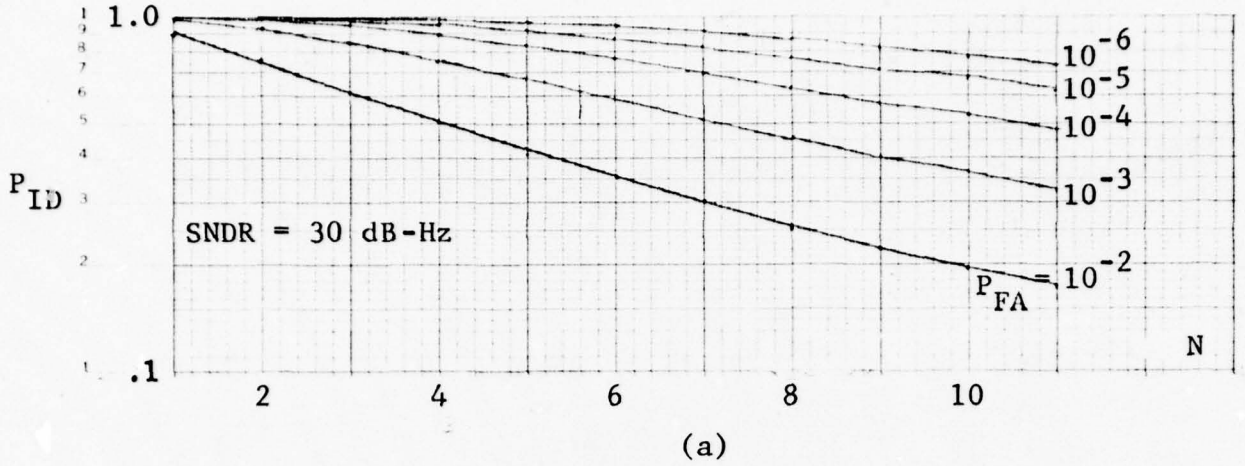
$$F(jt) = \frac{1}{\det\,[I - jtP_0 NI]} = \frac{1}{\left(1 - jtN\,\frac{P_0}{\sigma^2}\right)^K} \tag{2.45}$$

Manipulation of this expression indicates that the characteristic function derivatives to be used in (2.34) are given by

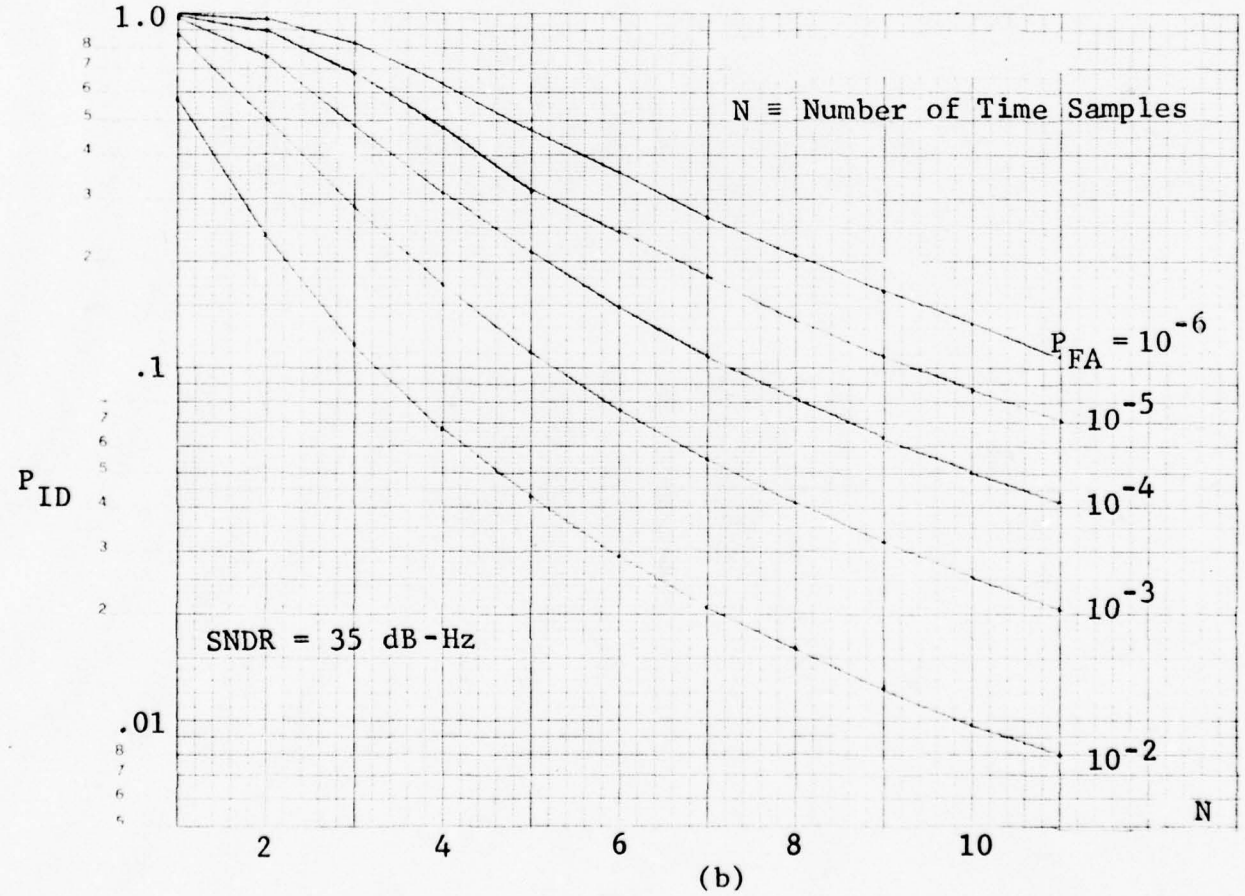
$$F^{(n)}\left(-\frac{1}{2(1+\alpha)}\right) = \frac{K(K+1)\cdots(K+n-1)}{\left[1 + \frac{NP_0}{2\sigma^2(1+\alpha)}\right]^{K+n}} \tag{2.46}$$

Utilizing the above formula, the signal presence detection capability of the preamble was determined. Numerical results, displayed as the probability of incorrect dismissal vs. number of time samples N are provided in Figs. 2.8 to 2.11. Each curve is parameterized to a fixed false alarm probability and each figure relates to one particular value of signal-to-noise density ratio, P/N_0.

To get some idea of the processing time required consider a sampling interval of 5 msec. (This particular choice approximates more highly motivated choices relating to the Doppler processing stage as discussed in Section 2.5.) Note that 20 time samples



(a)



(b)

Figure 2.8 Probability of Incorrect Dismissal (Four Independently Fading Tones) Vs. Number of Independent Noise Samples

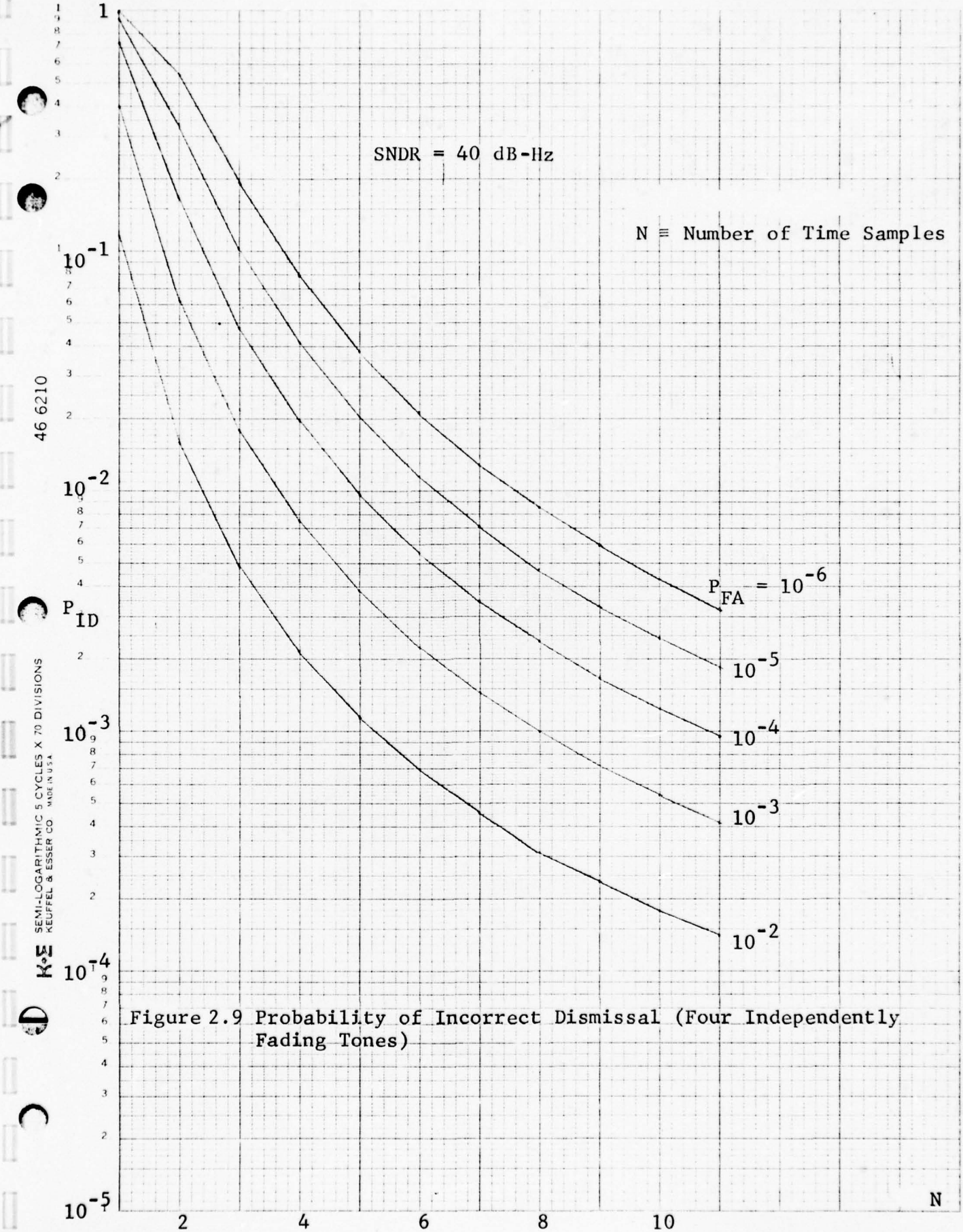


Figure 2.9 Probability of Incorrect Dismissal (Four Independently Fading Tones)

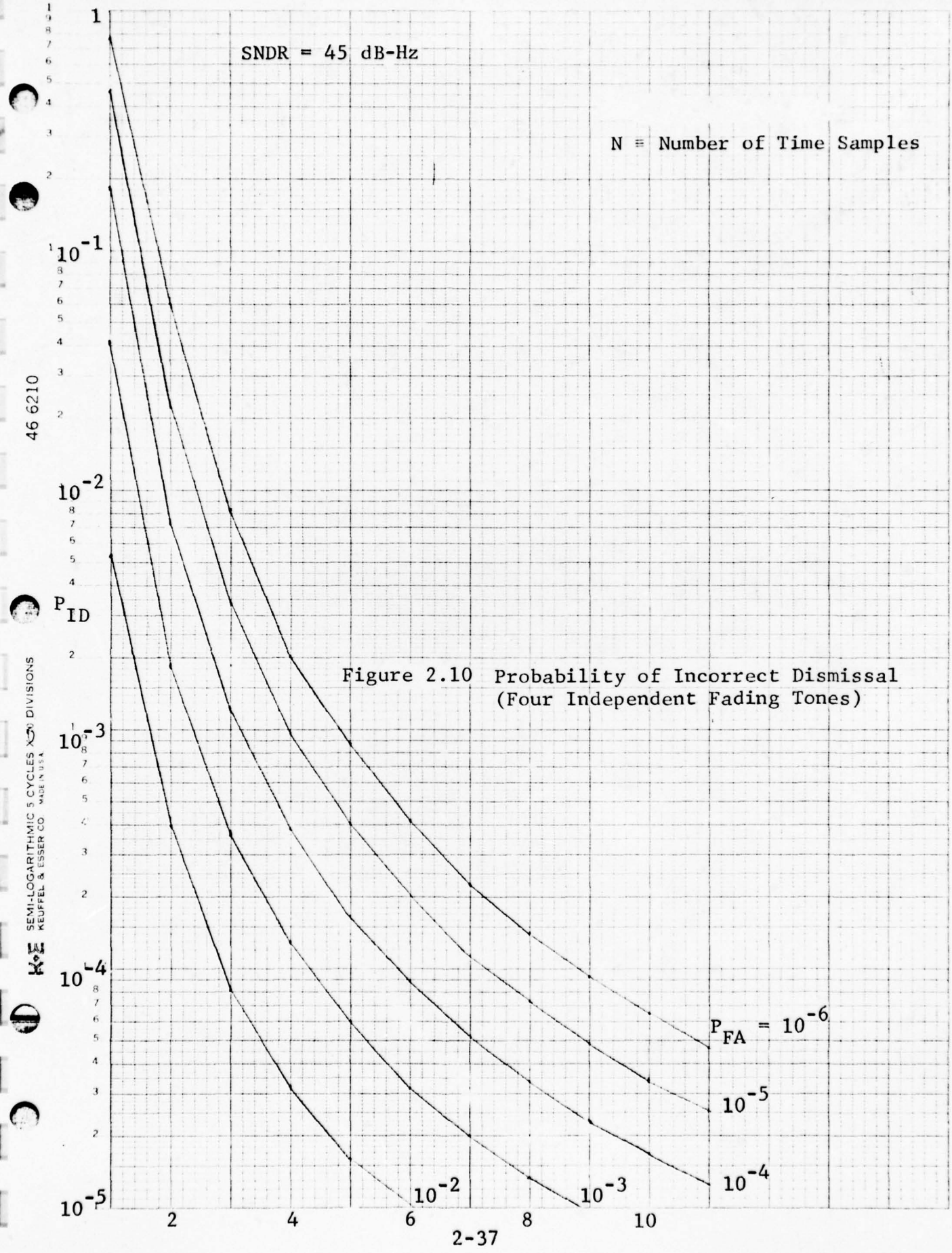
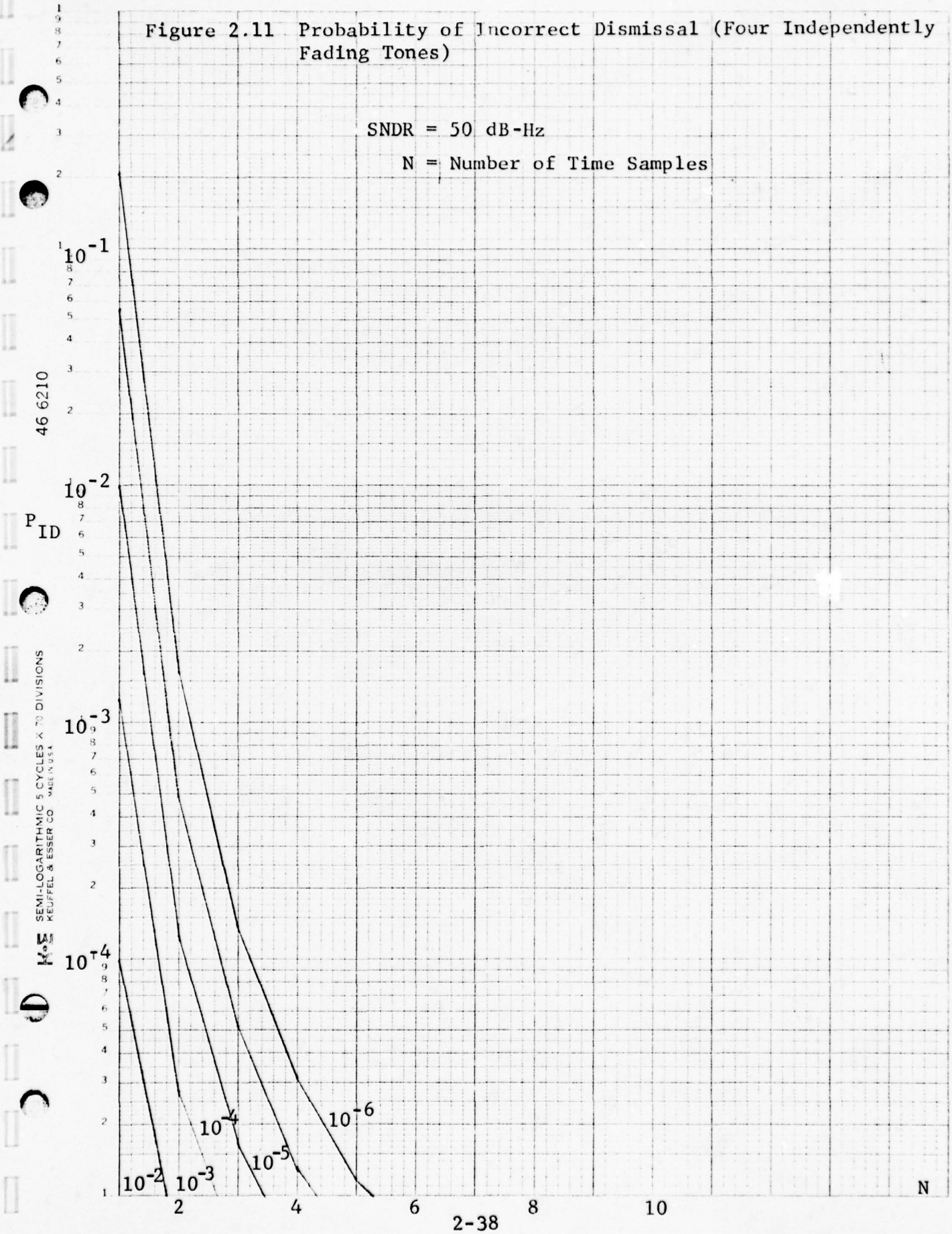


Figure 2.10 Probability of Incorrect Dismissal (Four Independent Fading Tones)

Figure 2.11 Probability of Incorrect Dismissal (Four Independently Fading Tones)



(i.e., N = 20) corresponds to .1 second of processing time. For the curves, four tones (K = 4) have been used and, as pointed out above, have been assumed to fluctuate independently.

The assumption of independence is one that should be examined more closely. This can be done by appealing to the representation of the frequency correlation function in (2.44) and expanding the exponential. Retaining terms up to second-order and setting the mean delay

$$\bar{\xi}_m = \sum_{m=1}^{M} G_m \xi_m \tag{2.47}$$

equal to zero we define the frequency correlation coefficient

$$\rho(f) = \frac{S(f)}{S(0)} \tag{2.48}$$

and the rms delay spread

$$\sigma_\xi = \left(\frac{\sum_{m=1}^{M} G_m \xi_m^2}{\sum_{m=1}^{M} G_m} \right)^{1/2} \tag{2.49}$$

and obtain after some simple substitutions the approximation

$$\rho(\Delta) \approx 1 - 2\pi^2 \Delta^2 \sigma_\xi^2 \tag{2.50}$$

which holds when Δ is not much greater than zero (i.e., there is still substantial correlation). Thus, to have enough decorrelation such that the inequality

$$\rho \leq \rho_0 \tag{2.51}$$

is satisfied the inequality

$$2\sigma_\xi \geq \frac{1}{\pi\Delta}\sqrt{2(1-\rho_0)} \tag{2.52}$$

For the preamble tone separation (about .6 KC) the above expansion indicates that decorrelation to a value of ρ less than .75 requires a multipath spread, $2\sigma_\xi$, in excess of .36 ms. This is typical of multipath spreads on HF links. In fact, on a link for which we have data, the Fort Monmouth-to-Palo Alto path, [2.2] there is a 60% chance that the cited value will be met or exceeded.

To get a complete picture of the consequences of correlation we have calculated detection results for the situation wherein all the tones fade in step. For this case we have

$$\lambda = \frac{KN}{\sigma^2}|z|^2 \tag{2.53}$$

and

$$Q_{11} = \frac{KN}{\sigma^2} \tag{2.54}$$

The characteristic function of the one term quadratic form is obtained simply from (2.46) by setting $K = 1$. This case is truly a worst case situation and, as one would expect, performance is considerably inferior to that presented earlier. This is illustrated in Fig. 2.12. For this degenerate case it appears to be roughly true that the processing time required to obtain a performance level comparable to that in Fig. 2.9 is about four times as great as it was for the independent tone case.

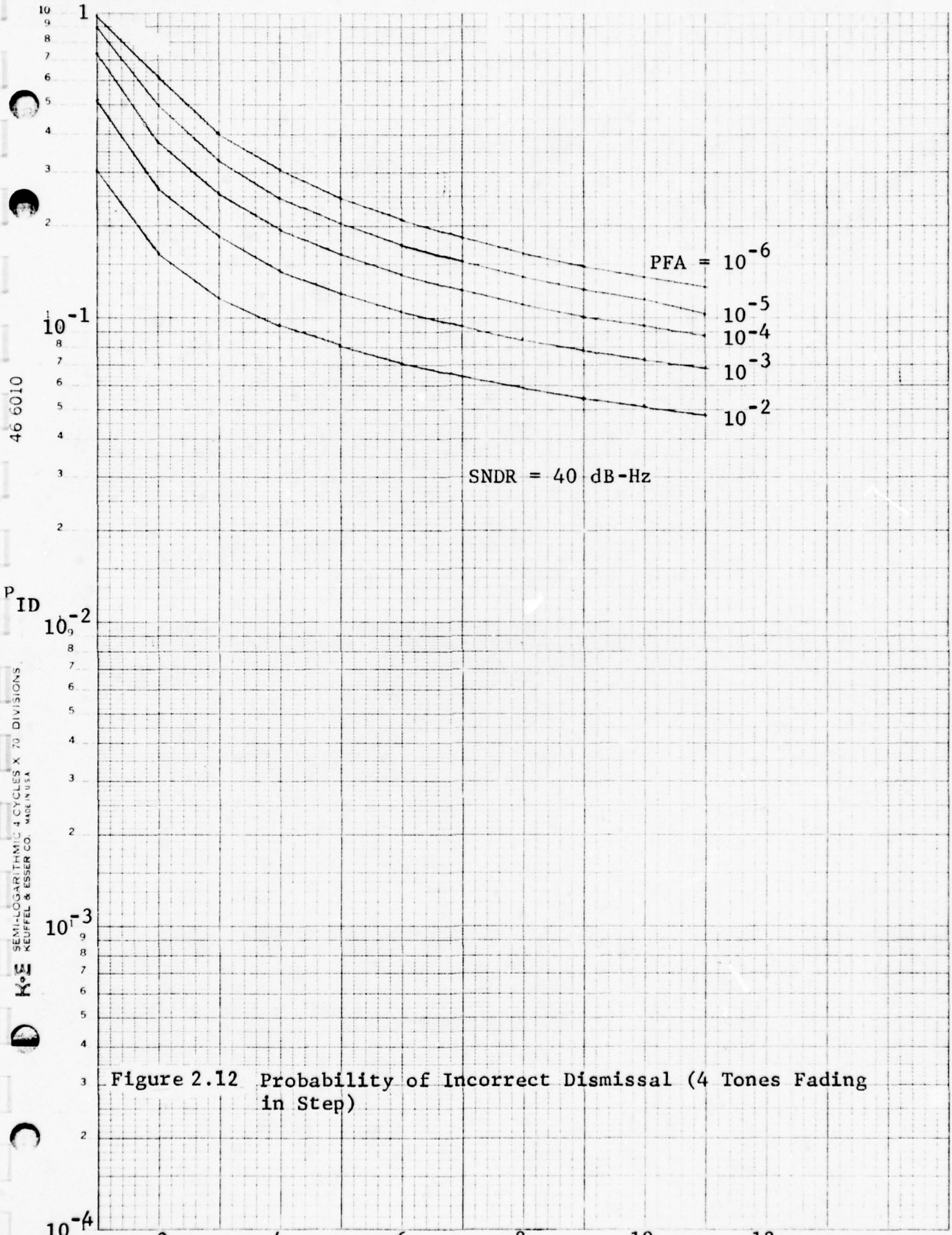


Figure 2.12 Probability of Incorrect Dismissal (4 Tones Fading in Step)

The true case is somewhere inbetween the two extreme cases presented with a definite bias towards the earlier high performance situation. It would be of interest to examine the possible case wherein there is a high correlation between adjacent tones only. Time does not permit such considerations, here. It can be reasonably speculated, however, that high correlation between adjacent tones would, to obtain the same performance, require a processing time falling somewhere inbetween the time T_0 to process uncorrelated tones and $2T_0$.

Though the specific threshold settings required to obtain the performance in Figs. 2.8 to 2.11 are of no interest from the viewpoint of performance, they are of interest relative to design. The values of α are tabulated in Table 2-4.

2.5 Doppler Estimation Performance

The four-tone structure described for the detection process in the previous section is also used for Doppler estimation. As described in Section 2.2.1 the tones are on for a total time

$$T_{tone} = T_{win} + T_{Dopp} \tag{2.55}$$

where T_{win} is the duration of the moving window used in the detection process* and T_{Dopp} is the fixed time interval required to do the Doppler estimation as discussed later in this section. With high probability detection will take place in T_{Det} seconds where $T_{Det} < T_{win}$. The time T' remaining in the tone is

*The duration referred to is just the amount of time it takes to collect the N time samples of the energy comparison detector in Fig. 2. .

TABLE 2-4

THRESHOLD VALUES (α)

P_{FA} / 2KN	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
8	6.03	12.05	22.71	41.64	75.3
16	3.37	5.20	7.61	10.79	15.01
24	2.66	3.74	5.00	6.51	8.32
32	2.32	3.09	3.95	4.92	6.02
40	2.11	2.73	3.38	4.09	4.88
48	1.98	2.49	3.02	3.59	4.20
56	1.88	2.32	2.77	3.24	3.74
64	1.80	2.19	2.59	2.99	3.42
72	1.74	2.09	2.44	2.80	3.17
80	1.69	2.01	2.33	2.65	2.98
88	1.65	1.95	2.24	2.53	2.82

$$T' = T_{tone} - T_{Det} = T_{win} - T_{Det} + T_{Dopp} \tag{2.56}$$

After Doppler processing the time remaining is

$$T'' = T_{win} - T_{Det} \leq T_{win} \tag{2.57}$$

seconds and hence, one must wait this length of time before commencing sync estimation.

Some of the issues raised above will be discussed in the later section (Section 2.7) on sync estimation. In this section we concentrate on determining performance of the Doppler shift estimator and on the fixed amount of time T_{Dopp} that must be allocated to the Doppler estimation.

2.5.1 Doppler Estimation Error for Single Stage

Doppler estimation takes place immediately after signal presence detection, i.e., immediately after the threshold exceedance indicated in (2.3). Based upon the analysis in [2.6, Section 5] one may postulate near-optimal estimation algorithms for utilizing the signal filter outputs to estimate Doppler shift. In the reference, the following algorithm was found to be near-optimal:

$$\hat{\nu} = \frac{1}{2\pi} \frac{\sum_{k=1}^{K} R_k^2 \dot{\theta}_k}{\sum_{k=1}^{K} R_k^2} \tag{2.58}$$

where R_k and $\dot{\theta}_k$ denote the envelope and phase derivative of the signal at the k'th tone filter output, respectively, and K is the number of tones. A form equivalent to (2.58) is:

$$\hat{\nu} = \frac{1}{2\pi} \frac{\sum_{k=1}^{K} x_k \dot{y}_k - y_k \dot{x}_k}{\sum_{k=1}^{K} x_k^2 + y_k^2} \tag{2.59}$$

where x_k and y_k are the in-phase and quadrature components and $\dot{x}_k$ and $\dot{y}_k$ are the corresponding derivatives of the k'th tone filter output, respectively. This complex representation assumes a reference tone frequency for each transmitted tone.

Equations (2.58) and (2.59) yield an estimate of Doppler corresponding to a single time sample of envelopes and phase derivatives (or in-phase components, quadrature components, and their derivatives) at the output of the signal filter. However, several time samples could be used in a Doppler estimate. Since the fading will be slow compared to the estimation time, successive tone filter output samples will differ primarily in that the noise components will fluctuate. Thus, the utility of having several output samples is primarily one of allowing the noise to be averaged out. Thus we may modify (2.58) and (2.59) to

$$\hat{\nu} = \frac{1}{2\pi} \frac{\sum_{\ell=1}^{N} \sum_{k=1}^{K} R_{k\ell}^2 \dot{\theta}_{k\ell}}{\sum_{\ell=1}^{N} \sum_{k=1}^{K} R_{k\ell}^2} \tag{2.60}$$

$$\hat{\nu} = \frac{1}{2\pi} \frac{\sum_{\ell=1}^{N} \sum_{k=1}^{K} x_{k\ell}\dot{y}_{k\ell} - y_{k\ell}\dot{x}_{k\ell}}{\sum_{\ell=1}^{N} \sum_{k=1}^{K} x_{k\ell}^2 + y_{k\ell}^2} \tag{2.61}$$

where the added ℓ subscript denotes the ℓ'th time sample and N is the number of time samples. Note that the denominators in (2.60) and (2.61) would be identical to the output of the signal energy filter if integrate-and-dump implementation were used over N samples.

Since the tone filters must have a bandwidth of the order of 150 Hz and since the tones may be ± 75 Hz relative to nominal center frequencies, a sampling rate somewhat faster than 150 samples/second would be desirable for the filter output. However, there is an added complication in the fact that it is necessary to compute $\{\dot{\theta}_{k\ell}\}$ or $\{\dot{x}_{k\ell}, \dot{y}_{k\ell}\}$ from the filtered tone. This would seem to require a sampling rate high compared to 75 Hz, or at least several samples close together in time at a high rate.

For example, the simplest approximation to the derivative of x(t) is given by

$$\hat{\dot{x}}(t) \approx \frac{x(t+\tau) - x(t)}{\tau} \tag{2.62}$$

Here τ must be small compared to the correlation time of x(t) or small compared to one cycle of maximum Doppler of 75 Hz. On the other hand, it must not be made too small or, due to the ambient noise, the derivative $\dot{x}(t)$ will be estimated poorly. More accurate algorithms than (2.62) are available to estimate the derivatives

which involve several delayed samples. In any case, an analysis and/or simulation is required to fully optimize the selection of algorithms. Figure 2.13 presents a block diagram of the processing described above to estimate Doppler shift.

Generally it can be said that the problem here is that the number of time samples N to be used in estimating the Doppler shift $\hat{\nu}$ must be optimized. On the one hand, making N large reduces the effect of noise fluctuations. On the other hand, making N too large can result in an averaging time over which the tone fluctuation becomes noticeable. This would violate the ground rules on which the algorithm was developed. The above estimation procedure, as discussed at the end of this section, is modified to become adaptive by using the Doppler estimate to make a Doppler correction and subsequently reducing the bandwidth of the tone filters to improve the estimate. Before making this step, however, future developments are simplified by determining the performance of a Doppler estimator based on (2.60) or (2.61).

Equations (2.60) and (2.61) present two equivalent versions of the Doppler estimator of interest, and Eqs. (2.58) and (2.59) present estimates, previously evaluated in [2.6], based upon a single sample of the filter outputs. Due to the difficulty of an exact analysis, we present here an approximate performance analysis of the multi-sample algorithm based upon a high SNR approximation. This approximation is useful here because Doppler estimation takes place only after signal detection has occurred, i.e., after

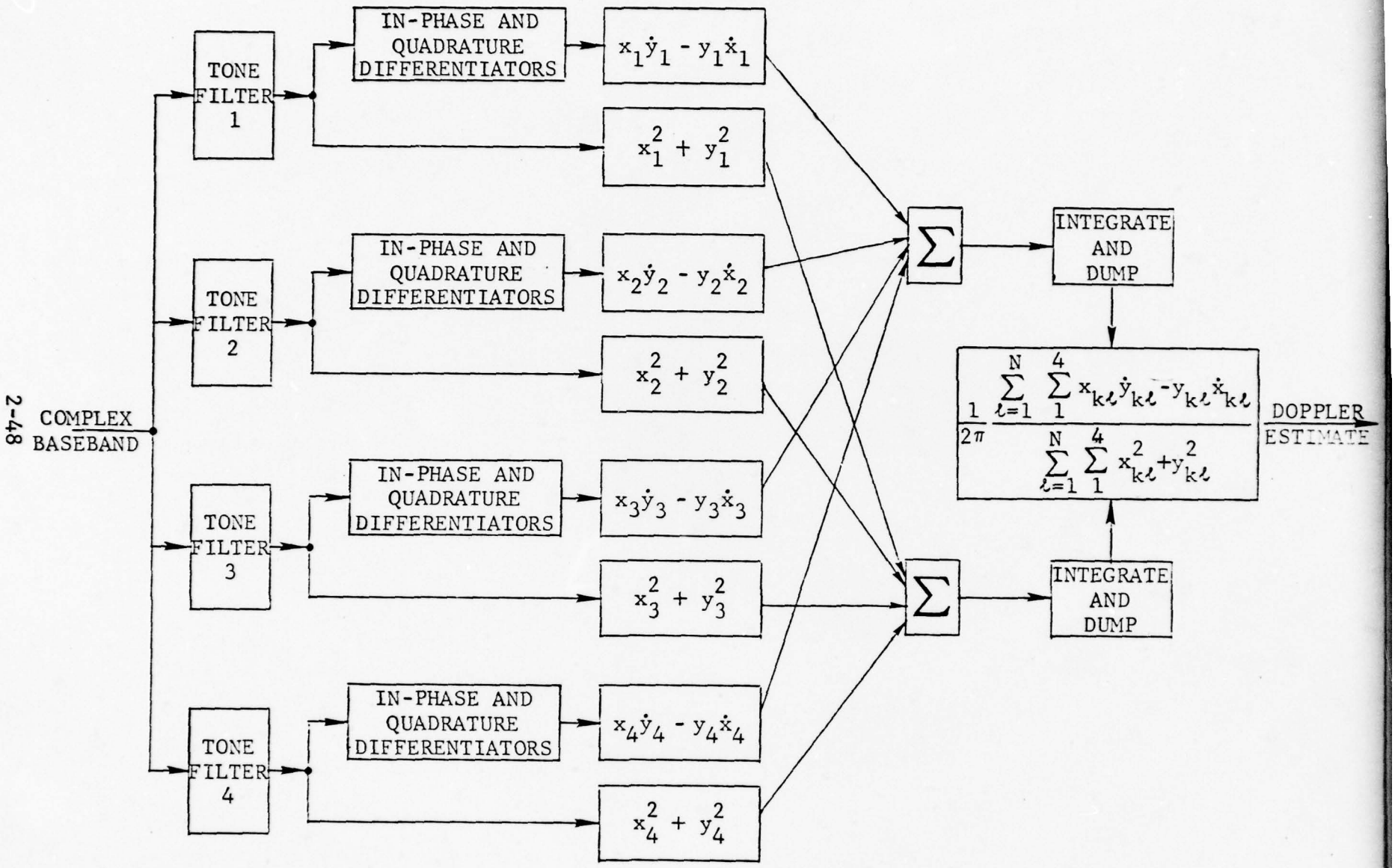


Figure 2.13 Illustration of a Doppler Estimation Technique

$$\sum_{\ell=1}^{N}\sum_{k=1}^{K}|w_k(\ell T)|^2 > \alpha \sum_{\ell=1}^{N}\sum_{k=1}^{K}|\eta_k(\ell T)|^2 \tag{2.63}$$

where

$$w_k(t) = z_k(t) + n_k(t) \tag{2.64}$$

In this expression $z_k(t)$ represents the received tone and $n_k(t)$ is the noise at the filter output. The rms bandwidth (Doppler spread) of $z_k(t)$ is assumed small enough so that, over the time interval for Doppler estimation, $z_k(t)$ differs little from a cissoid. Mathematically, in the vicinity of the time instant t_0, we may represent $z_k(t)$ in the form

$$z_k(t) = R_k(t_0)e^{j\theta_k(t_0)}e^{j(t-t_0)\dot{\theta}_k(t_0)}e^{j2\pi\nu_0(t-t_0)} \tag{2.65}$$

where $R_k(t)$ and $\theta_k(t)$ are the envelope and phase of $z_k(t)$, respectively, $\dot{\theta}_k(t)$ is the phase derivative or instantaneous angular frequency of $z_k(t)$ with respect to ν_0, and ν_0 is the centroid of the power spectrum of $z_k(t)$. ν_0 may also be called the mean Doppler shift on the "carrier" frequency of $z_k(t)$.

The Doppler estimate to be analyzed is repeated below [in somewhat altered notation from (2.60)]:

$$\hat{\nu} = \frac{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|w_k(\ell T)|^2\dot{\gamma}_k(\ell T)/2\pi}{\sum_{\ell=1}^{N}\sum_{k=1}^{K}|w_k(\ell T)|^2} \tag{2.66}$$

where

$$\gamma_k(t) = \measuredangle w_k(t)$$

Setting $t_0 = 0$ in (2.65) for simplicity, we see that

$$w_k(t) = R_k e^{j\theta_k} e^{j(\dot{\theta}_k + 2\pi\nu_0)t} + n_k(t) \tag{2.67}$$

$$= z_k(t)[1 + m_k(t)] \tag{2.67}$$

where the noise is given by

$$m_k(t) = \frac{n_k(t)}{R_k} e^{-j\theta_k} e^{-j(\dot{\theta}_k + 2\pi\nu_0)t} \tag{2.68}$$

For large SNR,

$$\overline{|m_k(t)|^2} = \frac{\overline{|n_k(t)|^2}}{R_k^2} \ll 1 \tag{2.69}$$

Note that $m_k(t)$ is essentially a scaled, frequency-shifted, version of the additive noise $n_k(t)$, where the frequency shift is

$$\nu_k = \nu_0 + \frac{\dot{\theta}_k}{2\pi} \tag{2.70}$$

Thus, the power spectrum of $m_k(t)$, $P_{m_k}(f)$, is related to that of $n_k(t)$ and $P_n(f)$ by the relationship

$$P_{m_k}(f) = \frac{1}{R_k^2} P_n(f + \nu_k) \tag{2.71}$$

where we have considered $\dot{\theta}_k$ and, thus, ν_k fixed.

If we represent $m_k(t)$ in terms of its quadrature components

$$m_k(t) = U_k(t) + jV_k(t) \tag{2.72}$$

then the high SNR quadrature approximation is given by

$$\not\arg\,[1 + m_k(t)] \approx V_k(t) \tag{2.73}$$

so that

$$\gamma_k(t) \approx (\dot{\theta}_k + 2\pi\nu_0)t + V_k(t) \tag{2.74}$$

and

$$\frac{\dot{\gamma}_k(t)}{2\pi} = \nu_0 + \frac{\dot{\theta}_k}{2\pi} + \frac{\dot{V}_k(t)}{2\pi} \tag{2.75}$$

Using (2.75) in (2.66),

$$\hat{\nu} - \nu_0 \approx \frac{\sum_{\ell=1}^{N}\sum_{k=1}^{K} |w_k(\ell T)|^2 \dot{\theta}_k/2\pi}{\sum_{\ell=1}^{N}\sum_{k=1}^{K} |w_k(\ell T)|^2} + \frac{1}{2\pi} \frac{\sum_{\ell=1}^{N}\sum_{k=1}^{K} |w_k(\ell T)|^2 \dot{V}_k(\ell T)/2\pi}{\sum_{\ell=1}^{N}\sum_{k=1}^{K} |w_k(\ell T)|^2} \tag{2.76}$$

We may further simplify (2.76) by noting from a high SNR approximation that

$$|w_k(t)|^2 \approx R_k^2[1 + 2U_k(t)] \tag{2.77}$$

It is not difficult to see that the noise component in the squared envelope of $w_k(t)$ constitutes a second-order effect in the output noise of the Doppler estimate (2.72). Thus, in (2.76) we replace $|w_k(t)|^2$ by R_k^2, yielding

$$\hat{\nu} = \nu_0 + \frac{\sum_{k=1}^{K} R_k^2 \dot{\theta}_k/2\pi}{\sum_{k=1}^{K} R_k^2} + \frac{\sum_{k=1}^{K} R_k^2 \left(\frac{1}{N} \sum_{\ell=1}^{N} \dot{V}_k(\ell T)/2\pi\right)}{\sum_{k=1}^{K} R_k^2} \tag{2.78}$$

The first term on the right-hand side of (2.78) represents the true Doppler and the second and third terms are noise terms. The second term is the estimation error due to the fading characteristics of the channel, while the third term represents the estimation error due to the additive noise.

Computation of the mean value of the Doppler estimator (2.78) shows it to be unbiased. In actual fact, the mean value of the unapproximated estimator (2.66) is biased, but the bias becomes small at high SNR. This bias is given by $-\nu_0(\overline{|n|^2}/\overline{|z|^2} + \overline{|n|^2})$, and would be around 1% of ν_0 for a 20 dB SNR. Because of our high SNR approximation, this bias does not appear in (2.78). However, the variance of the estimate (2.78) should be acceptable as a measure of the variance of the estimate (2.66) at high SNR.

We focus our attention on determining the variance of the noise terms in (2.74) conditioned on the set $\{R_k;\ k = 1, \ldots, K\}$ being available. From Rice [2.7] the joint density function of R_k, θ_k, and $\dot{\theta}_k$ is given by

$$W(R_k, \theta_k, \dot{\theta}_k) = \frac{R_k^2}{P_0} \frac{1}{\pi^2} \frac{1}{B_0\sqrt{\pi P_0}} \exp\left\{-\frac{R_k^2}{P_0}\right\} \exp\left\{-\frac{\frac{\dot{\theta}}{2\pi}^2}{P_0 B_0^2/4R_k^2}\right\} \tag{2.79}$$

where

$$P_0 = \overline{|R_k|^2} = \overline{|z_k|^2} \tag{2.80}$$

$$B_0 = 2\sqrt{\frac{\int (\nu - \nu_0)^2 P(\nu)\ d\nu}{\int P(\nu)\ d\nu}} \tag{2.81}$$

$$\nu_0 = \frac{\int \nu P(\nu)\ d\nu}{\int P(\nu)\ d\nu} \tag{2.82}$$

and $P(\nu)$ is the power spectrum of $z_k(t)$.

From these equations we see that the conditional density function of $\dot{\theta}_k$, given R_k, is Gaussian, zero mean, and with variance $\pi^2 P_0 B_0^2/2R_k^2$. In addition, if we assume that the tones fluctuate independently, the $\dot{\theta}_k$ will be independent. As a result,

$$E\left[\left(\frac{\sum_{k=1}^{K} R_k^2 \dot{\theta}_k}{2\pi \sum_{1}^{K} R_k^2}\right)^2 \Bigg/ \{R_k;\ k = 1, \ldots, K\}\right] = \frac{P_0 B_0^2}{8 \sum_{1}^{K} R_k^2} \tag{2.83}$$

We consider now the last term in (2.23). Note that the sampling interval T has been selected larger than the correlation interval of the complex Gaussian noise process. As a result, the set $\{\dot{V}_k(\ell T);\ k = 1, \ldots, K;\ \ell = 1, \ldots, N\}$ of random variables are statistically independent and of common variance. This variance is given by

$$\overline{[\dot{V}_k(t)]^2} = \frac{1}{2}\,\overline{|\dot{m}_k(t)|^2}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} (2\pi f)^2 P_{m_k}(f)\, df$$

$$= \frac{2\pi^2}{R_k^2}\int_{-\infty}^{\infty} f^2 P_n(f + \nu_k)\, df \tag{2.84}$$

where we have made use of (2.71).

Assuming the noise has a symmetrical spectrum and defining the rms bandwidth of the noise as

$$B_N = 2\sqrt{\frac{\int f^2 P_n(f)\, df}{\int P_n(f)\, df}} \tag{2.85}$$

we see that the variance of $\dot{V}_k(t)$ conditioned on R_k and $\dot{\theta}_k$ is given by

we see that the variance of $\dot{V}_k(t)$ conditioned on R_k and $\dot{\theta}_k$ is given by

$$\overline{[\dot{V}_k(t)]^2} = \frac{2\pi^2}{R_k^2}\left(\frac{B_N^2 P_N}{4} + \nu_k^2 P_N\right) \tag{2.86}$$

where

$$P_N = \int P_n(f)\, df = \overline{|n(t)|^2} \tag{2.87}$$

To simplify the calculations, we neglect the term $\dot{\theta}_k$ contained within ν_k, and thus replace ν_k by ν_0 to obtain the approximation

$$\overline{[\dot{V}_k(t)]^2} = \frac{2\pi^2 P_N}{R_k^2}\left(\frac{B_N^2}{4} + \nu_0^2\right) \tag{2.88}$$

The neglect of $\dot{\theta}_k$ is justified because it may be shown to contribute a component to the estimator variance which is small compared to the second term in (2.78), as long as the SNR is high.

Using (2.33) and the previously noted observation that the $\{\dot{V}_k(\ell T)\}$ are statistically independent leads to the result

$$E\left\{\left[\frac{\sum R_k^2 \frac{1}{N}\sum_{\ell=1}^{N} \dot{V}_k(\ell T)/2\pi}{\sum R_k^2}\right]^2 \Bigg/ \{R_k;\ k = 1, \ldots, K\}\right\} = \frac{1}{N}\,\frac{P_N B_N^2/8 + P_N \nu_0/2}{\sum_{k=1}^{K} R_k^2} \tag{2.89}$$

Thus, the variance of the Doppler estimator is given approximately at high SNR (conditioned on $\sum_{k=1}^{K} R_k^2$) by

$$\sigma_\nu^2 = \frac{AP_0}{\sum R_k^2} \tag{2.90}$$

where

$$A = \left[\frac{B_0^2}{8} + \frac{1}{N} \frac{P_N}{P_0} \left(\frac{B_N^2}{8} + \frac{\nu_0^2}{2} \right) \right] \tag{2.91}$$

For $N = 1$, this expression should approximate the variance of the single-sample estimation at high SNR evaluated in [2.6]. Indeed, examining Eq. (23) of Appendix B in [2.6] [use Eq. (2.6) for $\nu_N = 0$], we find that they agree at high SNR.

It remains now to average σ_ν^2 over $\sum_{k=1}^{K} R_k^2$ subject to the decision inequality (2.8). Because of the high SNR approximation, we replace the inequality (2.53) by

$$\frac{1}{P_0} \sum_{k=1}^{K} R_k^2 > \beta \tag{2.92}$$

where the random variable β is the quadratic form

$$\beta = \frac{\alpha}{NP_0} \sum_{\ell=1}^{N} \sum_{k=1}^{K} |\eta_k(\ell T)|^2 \tag{2.93}$$

The random variable

$$y = \frac{1}{P_0} \sum_{k=1}^{K} R_k^2 \tag{2.94}$$

has the χ^2 probability density and distribution function with 2K degrees of freedom:

$$W_K(y) = \frac{1}{(K-1)!} y^{K-1} e^{-y} \quad ; \quad y > 0 \tag{2.95}$$

$$F_K(Y) = P(y < Y) = 1 - e^{-Y} \sum_{j=0}^{K-1} \frac{1}{j!} Y^j \tag{2.96}$$

The desired estimation variance averaged over y, subject to the inequality (2.37) but with β fixed, is given by

$$E(\sigma_\nu^2 / y > \beta;\beta) = A \frac{\int_\beta^\infty \frac{1}{y} W_k(y)\, dy}{\int_\beta^\infty W_k(y)\, dy} \tag{2.97}$$

But,

$$\frac{1}{y} W_k(y) = \frac{1}{K-1} W_{K-1}(y) \quad ; \quad K \geq 2 \tag{2.98}$$

so that

$$E(\sigma_\nu^2 / y > \beta;\beta) = \frac{A}{K-1} \frac{\sum_{j=0}^{K-2} \frac{1}{j!} \beta^j}{\sum_{j=0}^{K-1} \frac{1}{j!} \beta^j} \quad ; \quad K > 2 \tag{2.99}$$

The final step in the evaluation of the estimator variance is the average of (2.44) over β. We note that β has a χ^2 density function with 2KN degrees of freedom:

$$W(\beta) = \left(\frac{NP_0}{\alpha P_N}\right)^{KN} \frac{1}{(KN-1)!} \beta^{KN-1} \exp\left\{-\beta \frac{NP_0}{\alpha P_N}\right\} \; ; \quad \beta > 0 \tag{2.100}$$

The mean and variance of β are:

$$\overline{\beta} = \left(\frac{\alpha P_N}{NP_0}\right) KN \tag{2.101}$$

$$\sigma_\beta^2 = \overline{\beta^2} - \overline{\beta}^2 = \left(\frac{\alpha P_N}{NP_0}\right)^2 (KN)(KN+1) - \left(\frac{\alpha P_N}{NP_0}\right)^2 K^2N^2$$

$$= \left(\frac{\alpha P_N}{NP_0}\right)^2 KN \tag{2.102}$$

and the ratio of standard deviation to mean is:

$$\frac{\sigma_\beta}{\overline{\beta}} = \frac{1}{\sqrt{KN}} \tag{2.103}$$

Since KN will be much bigger than 1 in applications, it does not appear that significant errors will be introduced in the final average over σ_ν^2 if we replace β by its average value, i.e.,

$$\sigma_\nu^2 \approx \frac{A}{K-1} \frac{\sum_{j=0}^{K-2} \frac{1}{j!} \left(\alpha K \frac{P_N}{P_0}\right)^j}{\sum_{j=0}^{K-1} \frac{1}{j!} \left(\alpha K \frac{P_N}{P_0}\right)^j} \quad ; \quad K > 1 \qquad (2.104)$$

It has been assumed that the Doppler shift remains strictly constant during the measurement period. In actual practice, the Doppler shift may vary at a rate of 3.5 Hz/second. The resultant variation during the measurement period will result in some averaged Doppler estimate. Clearly it is desirable to keep the measurement time as small as possible.

The result given by (2.104) and (2.91) is of primary importance in evaluating the operation of the multi-stage processor described in the following section.

2.5.2 Multi-Stage Doppler Estimator

Applying the reasonable high SNR approximation to (2.104) and combining with (2.91) we conclude from the basic theoretical considerations of the previous section that a single-stage estimate of Doppler shift based on reception of a 4-tone preamble yields an estimate with variance given approximately by

$$\sigma^2 = \frac{P_N}{3NP_0}\left(\frac{B_N^2}{8} + \frac{\nu_0^2}{2}\right) + \frac{B_0^2}{24} \qquad (2.105)$$

where P_0 and P_N are the signal power and noise power, respectively at the tone filter output, B_N is the rms bandwidth of the tone filter, ν_0 is the Doppler shift on the incoming signal, B_0 is the rms Doppler spread of the channel, and N is the number of independent time samples of each output tone used in the Doppler

estimate. Denoting the tone filter transfer function by H(f) we make use of the definitions

$$B_F = \frac{\int |H(f)|^2 \, df}{|H(0)|^2} \tag{2.106}$$

$$B_N = \frac{4 \int f^2 |H(f)|^2}{\int |H(f)|^2 \, df} \tag{2.107}$$

and write

$$P_N = B_F N_0 \tag{2.108}$$

$$P_0 = \frac{P}{4} |H(\nu_0)|^2 \tag{2.109}$$

where N_0 is the single-sided noise power density and P is the total input power in the four tones.

In the subsequent discussion we neglect the term $B_0^2/24$ which will usually be small compared to noise contributions.

It is convenient to relate B_N and B_F and the filter time constant t_c to the half-power frequency of the filter H(f). To momentarily avoid considering a particular filter we write

$$B_F = af_{1/2} \qquad B_N = bf_{1/2} \qquad t_c = c/f_{1/2} \tag{2.110}$$

where a, b, and c can be determined for the particular filter under consideration. For the double-tuned filter discussed later t_c will be the time it takes the impulse response of one section to decay to e^{-1} times its original value.

Choosing $3t_c$ as a reasonable time interval between samples to achieve independence of noise samples at the filter output we use $N = T/(3t_c)$ in (2.105) and perform the other obvious substitutions to get

$$\sigma^2 = \frac{2ac}{sT|H(\nu_0)|^2}\left(\frac{b^2 f_{1/2}^2}{4} + \nu_0^2\right)$$

$$s = \frac{P}{N_0} \tag{2.111}$$

We now consider a worst-case situation wherein the half-power frequency has been chosen equal to some maximum design Doppler shift value, and ν_0 the Doppler shift on the channel actually takes on that value. We have

$$\sigma^2 = \frac{4ac}{sT}\left(\frac{b^2}{4} + 1\right) f_{1/2}^2 \tag{2.112}$$

As a particular possibility we consider a double-tuned filter

$$H(f) = \frac{\alpha^2}{(j2\pi f + \alpha)^2} \tag{2.113}$$

Each of the cascaded sections of this filter has an impulse response

$$h_0(t) = \alpha e^{-\alpha t} U(t) \tag{2.114}$$

The overall impulse response for the double-tuned filter is

$$h(t) = h_0(t) \otimes h_0(t)$$

$$= \alpha^2 t e^{-\alpha t} U(t) \qquad (2.115)$$

For the quantity t_c in (2.110) we choose the time constant of a single section, i.e.,

$$t_c = 1/\alpha \qquad (2.116)$$

Though, this quantity is not the time-constant of the overall filter in the usual sense, it serves our purposes here and is referred to as the time constant in the remainder of our present discussion.

In view of our choice of $3t_c$ as the sampling interval in our derivation of (2.111) it is of interest to determine the extent of correlation between samples of noise at the filter output taken $3t_c$ seconds apart. Since the noise is flat across the band the correlation coefficient is given simply by

$$\rho(\tau) = \frac{\int h(\tau + n)h^*(n)\, dn}{\int |h(n)|^2\, dn} \qquad (2.117)$$

Substituting (2.115) and evaluating the integrals yields

$$\rho(\tau) = [1 + \alpha|\tau|]e^{-\alpha|\tau|} \qquad (2.118)$$

Evaluating this expression at $3t_c$ yields

$$\rho(3t_c) = .2 \tag{2.119}$$

Thus, the choice of $3t_c$ as a sampling interval seems justified on the basis of providing reasonably uncorrelated noise samples at the filter output.

Now the constants in (2.110) can easily be determined from (2.106) and (2.107). We get

$$a = 2.441 \qquad b = 3.108 \qquad c = .1024 \tag{2.120}$$

Hence, using (2.102)

$$\sigma^2 = \frac{3.414}{sT} f_{1/2}^2 \tag{2.121}$$

In addition to the usual inverse dependence on signal-to-noise ratio s and observation time T, (2.121) shows the dependence on the half-power frequency $f_{1/2}$. Obviously, as $f_{1/2}$ is increased more noise is allowed to contribute to the estimate and the estimate is degraded.

In order to compensate for the $f_{1/2}$ dependence described above we have considered a multi-stage estimation process wherein there is a Doppler correction made at the end of each stage with a corresponding reduction in filter bandwidth, i.e., $f_{1/2}$. The Doppler correction, of course, is based on the estimate performed in the preceding stage. To take into account the error

in this estimate the half-power frequency of the current stage is chosen to be equal to three standard deviations of the estimate in the previous stage.

For a three-stage estimator the sequence of filter configurations is shown in Fig. 2.14. We have used f_i to denote the i'th stage half-power frequency. Note that the σ's are calculable from (2.121) and the f_i are chosen according to the rule $f_i = 3\sigma_{i-1}$.

Figure 2.14 also illustrates the tone configuration. At the first stage a worst-case Doppler shift of 75 Hz is assumed; the tone is estimated with variance σ_1^2. As part of our worst-case consideration a $3\sigma_1$ error is assumed for the stage 1 estimate. This estimate is used to perform the Doppler correction which just carries over the $3\sigma_1$ error into stage 2. In stage 2 the half-power frequency of the filter has been changed to $f_2 = 3\sigma_1$. In this stage the tone frequency is estimated with variance σ_2^2 and pushing through the worst-case choice we again assume a three-sigma error (this time $3\sigma_2$).

To summarize our approach to getting numerical results for multi-stage estimation of Doppler shift we have assumed a multi-stage operation that uses the previous stage estimate to make a Doppler correction. During the Doppler correction preceding the ℓ'th stage, the half-power frequency is changed from $f_{\ell-1}$ to $f_\ell = 3\sigma_{\ell-1}$. With this fact one can generate a recursion relation from (2.121) that can be manipulated to show that the estimate variance at the output of the L'th stage is given by

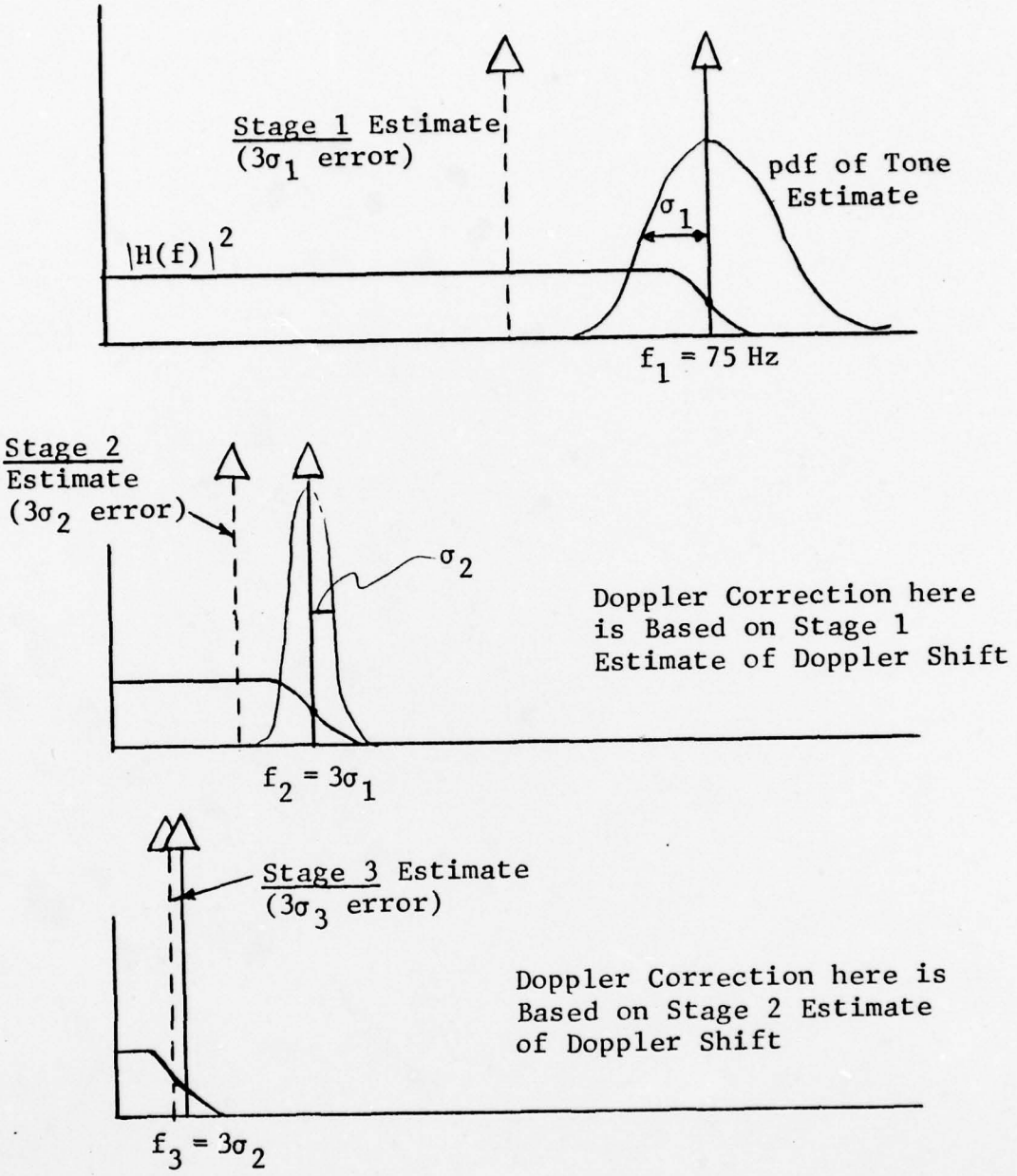


Figure 2.14 Three-Stage Adaptive Filter for Doppler Shift Estimation (Order of Occurrence is Top-to-Bottom) "Worst-Case" Performance is Illustrated

$$\sigma_L^2 = \left(\frac{3.414}{s}\right)^L \frac{9^{L-1} f_1^2}{\left[\prod_{\ell=1}^{L} T_\ell\right]} \tag{2.122}$$

where T_ℓ is the processing time for the ℓ'th stage.

Recall that our design goal is to minimize total processing time to achieve a given level of estimator performance. In doing this we fix the estimator variance to some desired design value and encounter the mathematical problem of solving the equation

$$T_1 T_2 \ \ldots \ T_L = \text{const} \tag{2.123}$$

for the set of values $\{T_i;\ i = 1,L\}$ that minimizes the sum

$$T_{tot} = T_1 + T_2 + \ldots \ T_L \tag{2.124}$$

This is achieved by using a basic theorem of analysis that states that the geometric mean of the elements in a set $\{a_1 a_2 \ \ldots \ a_i\}$ is less than the arithmetic mean of the elements unless the elements are all equal, i.e., defining

$$G(a) \equiv \left(\prod_{i=1}^{n} a_i\right)^{1/n} \tag{2.125}$$

and

$$M(a) = \frac{1}{n} \sum_{i=1}^{n} a_i \tag{2.126}$$

we have

$$G(a) \leq M(a) \tag{2.127}$$

with equality holding only if

$$a_1 = a_2 = \cdots a_n \tag{2.128}$$

Since G(a) is constant in our context and since minimizing the sum in (2.124) above is equivalent to minimizing the arithmetic mean, the inequality (2.127) implies directly that the sum in (2.124), subject to the fixed product constraint in (2.123), is minimized by choosing all the T_i equal.

Thus, each stage processes for equal time intervals of duration T. From (2.122) this value is given by

$$T = 9 \left(\frac{3.414}{s}\right)\left[\left(\frac{f_1}{3\sigma}\right)^2\right]^{1/L} \tag{2.129}$$

Sampling rates, of course, will reduce appreciably as we move to later stages. Using the fact that the sampling interval* in each stage is assumed to be three time constants we have from (2.110) and (2.120)

$$N_\ell = \frac{Tf_\ell}{.3072} \tag{2.130}$$

where

$$f_\ell = 3\sigma_{\ell-1} \tag{2.131}$$

*By sampling interval we here refer to the time between samples used in the computation. This would be implemented as an integer multiple of the fundamental sampling interval.

From the expressions provided above, the data of Table 2-5 has been compiled using an s value (signal-to-noise density ratio) of 40 dB/Hz. In this table T_ℓ represents the duration of each stage as determined from (2.129); T_{tot} is just this quantity multiplied by the number of stages. The number of samples taken during each stage is determined from round-off of (2.130). (For those stages in which round-off has resulted in a zero value, the value has been replaced by unity and the stage number marked with an asterisk.)

All of the quantities to the left of the double line in Table 2-5 are directly determinable from the expressions presented above. To the right of the double line two corrections are represented:

a) Integer sample correction

b) Transient build-up time correction.

The first correction derives from the fact that our solution for T does not necessarily represent an integer number of samples; to correct this we have adjusted T to an appropriate value T_ℓ for each stage, T_ℓ being the sampling period for that stage (three time constants) multiplied by the integer determined from round-off of (2.130). With this correction we have calculated [using (2.122) now] the new estimator standard deviation and total processing time. These are shown in the first two columns to the right of the double line in Table 2-5. In the third column we show the processing time that is obtained when the transient build-up time between stages is taken into account. This is obtained by adding to T_{tot} one sampling interval (three time constants) for each stage after the first.

Among other things, we see from this table that a three-stage processor can be used to reduce estimation error to sigma

TABLE 2-5

PERFORMANCE OF MULTI-STAGE DOPPLER ESTIMATOR FOR DESIGN VALUES OF ESTIMATOR STANDARD DEVIATION, σ = 1 Hz, .75 Hz, .5 Hz, and .25 Hz
SNDR = 40 dB-Hz

	Number of Stages	T (sec)	T_{tot} (sec)	Samples Per Stage					Sampling Interval for Each Stage					Integer Sample Correction		Addition of Transient Build-Up Time Gives T_{tot} (sec)
				#1	#2	#3	#4	#5	#1	#2	#3	#4	#5	σ (Hz)	T_{tot} (sec)	
1 Hz	1	1.921	1.921	469	–	–	–	–	.0041	–	–	–	–	.1	1.921	1.921
	2	.0768	.1536	19	4	–	–	–	.0041	.0205	–	–	–	.962	.1597	.1802
	3	.0263	.0788	6	2	1	–	–	.0041	.0120	.0350	–	–	.938	.0835	.1305
	4*	.0154	.0615	4	2	1	–	–	.0041	.0092	.0205	.0458	–	.445	.1010	.1764
	5*	.0111	.0557	3	1	1	1	1	.0041	.0078	.0148	.0283	.0538	.281	.1170	.2217
.75 Hz	1	3.414	3.414	833	–	–	–	–	.0041	–	–	–	–	.750	3.412	3.412
	2	.1024	.2048	25	4	–	–	–	.0041	.0236	–	–	–	.780	.197	.221
	3	.0318	.0955	8	2	1	–	–	.0041	.0132	.0424	–	–	.703	.1016	.157
	4*	.0177	.6710	4	2	1	1	–	.0041	.0098	.0236	.0568	–	.359	.1165	.2068
	5	.0125	.0625	3	2	1	1	1	.0041	.0083	.0167	.0036	.0677	.149	.1467	.2730

TABLE 2-5 (Continued)

PERFORMANCE OF MULTI-STAGE DOPPLER ESTIMATOR FOR DESIGN VALUES OF ESTIMATOR STANDARD DEVIATION, σ = 1 Hz, .75 Hz, .5 Hz, and .25 Hz
SNDR = 40 dB-Hz

	Number of Stages	T (sec)	T_{tot} (sec)	Samples Per Stage					Sampling Interval for Each Stage					Integer Sample Correction		Addition of Transient Build-Up Time Gives T_{tot} (sec)
				#1	#2	#3	#4	#5	#1	#2	#3	#4	#5	σ (Hz)	T_{tot} (sec)	
.5 Hz	1	7.681	7.681	1879	–	–	–	–	.0041	–	–	–	–	.5	7.681	7.681
	2	.1536	.3073	38	5	–	–	–	.0041	.0290	–	–	–	.512	.3005	.3294
	3	.0417	.1251	10	3	1	–	–	.0041	.0151	.0556	–	–	.419	.1418	.2125
	4*	.0217	.0869	5	2	1	1	–	.0041	.0109	.0290	.0770	–	.237	.1482	.2651
	5*	.0147	.0735	4	2	1	1	1	.0041	.0090	.0196	.0428	.0937	.086	.1904	.3554
.25 Hz	1	30.73	30.73	7501	–	–	–	–	.0041	–	–	–	–	.25	30.73	30.73
	2	.3073	.6145	78	8	–	–	–	.0041	.0410	–	–	–	.242	.6349	.6759
	3	.0662	.1986	16	3	1	–	–	.0041	.0190	.0882	–	–	.234	.2108	.3181
	4*	.0307	.1229	8	2	1	1	–	.0041	.0130	.0410	.1295	–	.111	.2292	.4126
	5*	.0194	.0969	5	2	1	1	1	.0041	.0103	.0258	.0649	.1631	.039	.2949	.5590

values of 1, .75, .5, and .25 Hz in .1305, .157, .2125, and .3181 second, respectively.

Because of channel Doppler spread (largest typical value approximately 1 Hz) it seems illusory to attempt estimates of much higher precision than the cases cited above.

Some indication of the dependence of Doppler estimator performance on SNDR is given by Fig. 2.15. It can be seen that there is no substantial variation of estimator performance within a few dB-Hz ($\pm$ 2) of 40 dB-Hz.

2.6 Frame Sync Estimation

In this section we discuss the segment of the preamble utilized for frame synchronization. The structure of this segment, referred to as the sync preamble, will be described in detail in Section 2.6.1 below. Integration of the sync preamble into the preamble as a whole was discussed in Section 2.

The performance of the frame sync estimator can be degraded by several factors:

1) Doppler correction error
2) Terminal equipment distortion
3) Poor threshold setting
4) Additive noise
5) Multipath.

2.6.1 Basic Elements of Sync Preamble

The overall structure and timing of the sync preamble was described in Section 2. The basic signal element of the sync preamble was seen to be a pulse train designed both for high crest factor and low autocorrelation side lobes. It is assumed that in the receiver Doppler correction is carried out prior to receipt

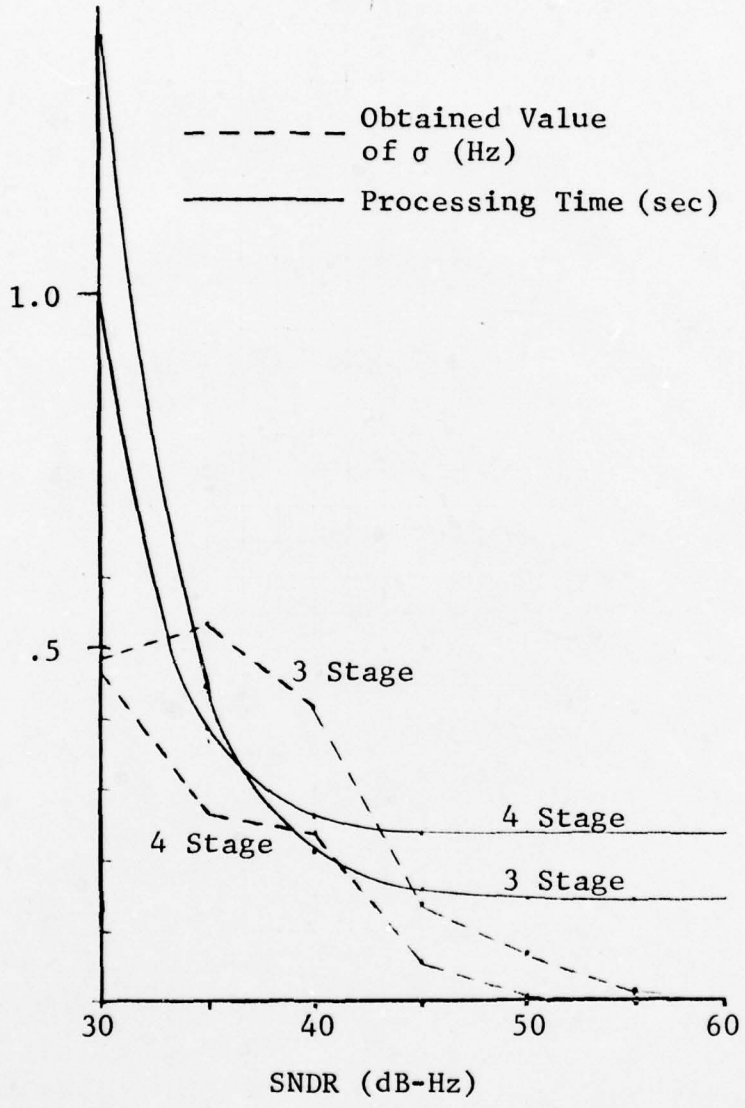


Figure 2.15 Performance of Doppler Estimator for Design Value of σ = .5 Hz (σ ≡ RMS Doppler Shift Estimation Error)

of the sync preamble, and that the latter preamble duration is small compared to the reciprocal Doppler error, so that ambiguity function design of the sync preamble is unnecessary. To test the validity of this assumption the effect of Doppler error on the sync estimation is evaluated in Section 2.6.3.

The pulse train is assumed generated by a sequence of two operations. First, a discrete sequence is generated which has a desirable discrete autocorrelation function (i.e., low side lobes). This discrete sequence is fed to a pulse-shaping filter which is designed to have the following properties:

1) The spectrum of the pulse is kept within the "flat" passband of the channel terminal equipment to prevent ringing or pulse broadening.

2) The pulse shape is designed to preserve, as much as possible, the good crest factor of the discrete sequence, and to provide low side lobes at the receiver matched filter output.

In practice, the pulse shaping filter will include any zero-order holds or anti-alias filters present in the D/A conversion process. The major part of the pulse shaping can be done in the computer or by an external analog filter.

We consider first possible discrete sequences with good autocorrelation properties. Selection of the discrete sequence is influenced in part by the anticipated receiver processing. To simplify the latter, the discrete sequence should ideally contain numbers that allow the matched filtering in the receiver to use few multiplies. A sequence containing only the numbers ± 1 would avoid the use of a multiplier entirely. Use of $\pm j$ weights should also avoid the use of a multiplier since multiplication by $\pm j$ involves interchanging the real and imaginary parts of

words with possible sign changes. The inclusion of multiples of 45^{o} in the phase weights would introduce the requirement to have available the input in-phase and quadrature components weighted by $\pm 1/\sqrt{2}$.

If weights of ± 1 and $\pm j$ are used, only relatively short sequences are known which have low side lobes. From the paper by Frank [2.3], one may find a 64 sample sequence containing $\pm 1, \pm j, \left(\pm \frac{1}{\sqrt{2}} \pm j \frac{1}{\sqrt{2}} \right)$ weightings with very low side lobes in its autocorrelation function. This sequence is the entry N = 8 in Table I of [2.3]. The ratio of autocorrelation peak to side lobe for this waveform is 25. A sequence of length-15 is given by Carley [2.9], containing ± 1, $\pm j$ weightings, which has an autocorrelation function peak-to-sidelobe ratio of 15. The length-64 and -15 waveforms are listed here in Fig. 2.16 together with the longest Barker sequence known [2.10] (length-13) containing ± 1's as elements. This latter sequence has an autocorrelation function peak-to-sidelobe ratio of 13.

The transmitted pulse train complex envelope z(t), may be represented as the convolution of an impulse train having areas corresponding to the selected discrete sequence with a continuous filter p(t) providing the pulse shape, i.e.,

$$z(t) = p(t) \otimes \sum_{0}^{M-1} \gamma_m \delta(t - m\Delta)$$

$$= \sum_{0}^{M-1} \gamma_m p(t - m\Delta) \qquad (2.132)$$

where $\{\gamma_m;\ m = 0,1, \ldots, M-1\}$ is the discrete sequence, $\delta(\cdot)$ is the unit impulse, and Δ is the time between samples.

LENGTH-13 SEQUENCE (Barker [1])	1	1	1	1	1	-1	-1	1	1	-1	1	-1	1

LENGTH-15 SEQUENCE (Carley [2])	1	-1	1	j	-j	-1	j	-j	-j	1	j	j	1	1	1

LENGTH-64 SEQUENCE (Frank [1])

1	1	1	1	1	1	1	1	1	$e^{j\frac{\pi}{4}}$	j	$je^{j\frac{\pi}{4}}$	-1	$-e^{j\frac{\pi}{4}}$	-j	$e^{-j\frac{\pi}{4}}$
1	j	-1	-j	1	j	-1	-j	1	$je^{j\frac{\pi}{4}}$	-j	$e^{j\frac{\pi}{4}}$	-1	$e^{-j\frac{\pi}{4}}$	j	$-e^{j\frac{\pi}{4}}$
1	-1	1	-1	1	-1	1	-1	1	$-e^{j\frac{\pi}{4}}$	j	$e^{-j\frac{\pi}{4}}$	-1	$e^{j\frac{\pi}{4}}$	-j	$je^{j\frac{\pi}{4}}$
1	-j	-1	j	1	-j	-1	j	1	$e^{-j\frac{\pi}{4}}$	-j	$-e^{-j\frac{\pi}{4}}$	-1	$je^{j\frac{\pi}{4}}$	j	$e^{j\frac{\pi}{4}}$

Figure 2.16 Sequences of Length-13, -15, and -64 with Low Autocorrelation Sidelobes

Since the γ_m have unit magnitude, the discrete sequence may be said to have a unity crest factor. The pulse shape p(t) is assumed to be of the Nyquist type, having the normalized value unity as $t = 0$ and zero value at time samples equal to multiples of Δ. Such a pulse shape assures that z(t) has constant magnitude at the time instants $t = m\Delta$, m an integer. The values inbetween will generally not be constant envelope. However, if p(t) is chosen to drop to zero rapidly for t outside the interval $-\Delta < t < \Delta$, the values inbetween will be primarily some linear combination of adjacent values, and thus quite limited in their fluctuations. Another requirement on p(t) is that it be bandlimited since the spectrum of the pulse must lie within the flat portion of the HF terminal equipment channel. A raised cosine spectrum of bandwidth $2/\Delta$ satisfies the Nyquist requirement. The low pass spectrum that would be used is shown in Fig. 2.17. The shaded regions are frequency guard bands placed where the transceiver passband is not flat. If a 2 kHz wide flat passband is available, one may select Δ = 1 ms. A length-64 discrete sequence would occupy 64 ms and the analog pulse train an additional few milliseconds, or around three frames for a typical 22.5 ms frame duration.

The raised cosine spectrum is only given as an illustration, and other shapes are possible. It must be realized that the so-called "flat" part of the transceiver will have some "bumps" and produce pulse shape deterioration. The following section will include a discussion of the interaction between receiver processing and pulse shape in producing low side lobes at the matched filter output.

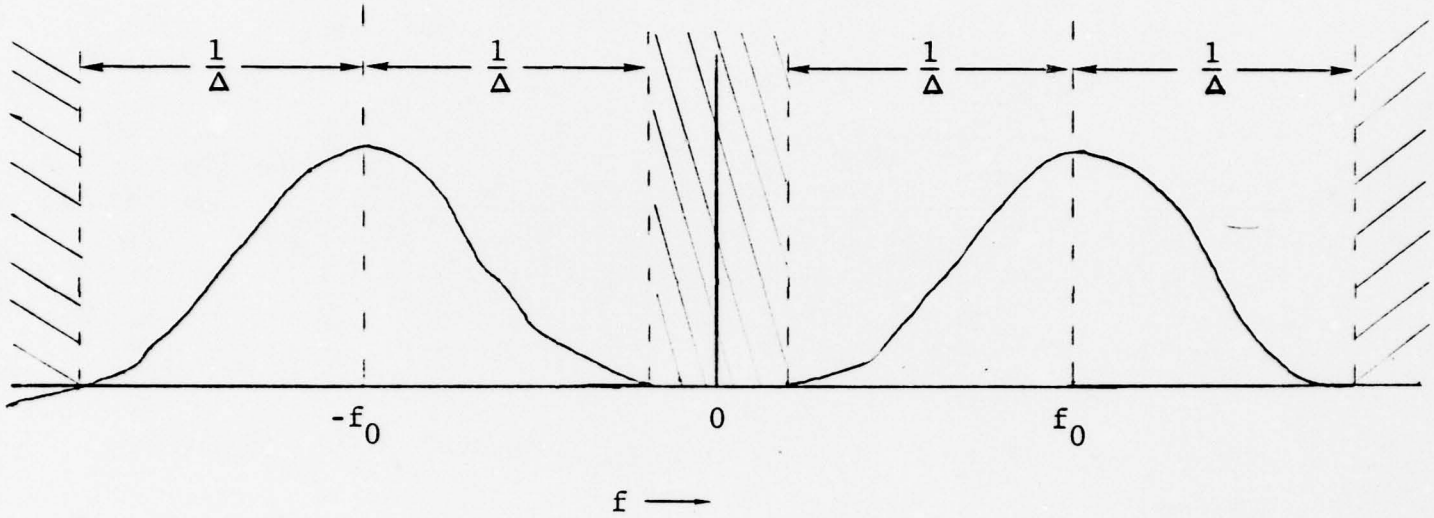


Figure 2.17 Transmitted Pulse Spectrum for Sync Preamble Pulse Train

Signal processing of the synch preamble is illustrated in Fig. 2.18. Estimation of frame sync takes place, of course, after signal detection, Doppler estimation, and Doppler correction. The estimation of frame sync is carried out by utilization of a "matched" filter followed by a multipath centroid detector which locates all peaks above a given threshold and forms the centroid of their locations.

As indicated in Fig. 2.18, the general structure of the matched filter would be the cascade of a pulse matched filter matched to the transmitted pulse shape and a discrete FIR filter matched to the M-length discrete sequence formed in the transmitter. Referring to (2.1), $\{\gamma_m;\ m = 0,1, \ldots, M-1\}$ denotes the discrete sequence and $p(t)$ denotes the transmitted pulse yielding a transmitted signal with complex envelope equal to $z(t)$ in (2.1).

Since the received pulse filter may not be matched exactly to the transmitted pulse, we denote the pulse filter impulse response by $q^*(-t)$, and the convolution of $p(t)$, $q^*(-t)$ becomes

$$r(t) = \int q^*(\tau)p(\tau + t)\, d\tau \tag{2.133}$$

When the receive pulse filter is truly matched, $q(t) = p(t)$ yielding

$$r(t) = \int p^*(\tau)p(\tau + t)\, d\tau \tag{2.134}$$

At the receive pulse filter output, the pulse train is given by

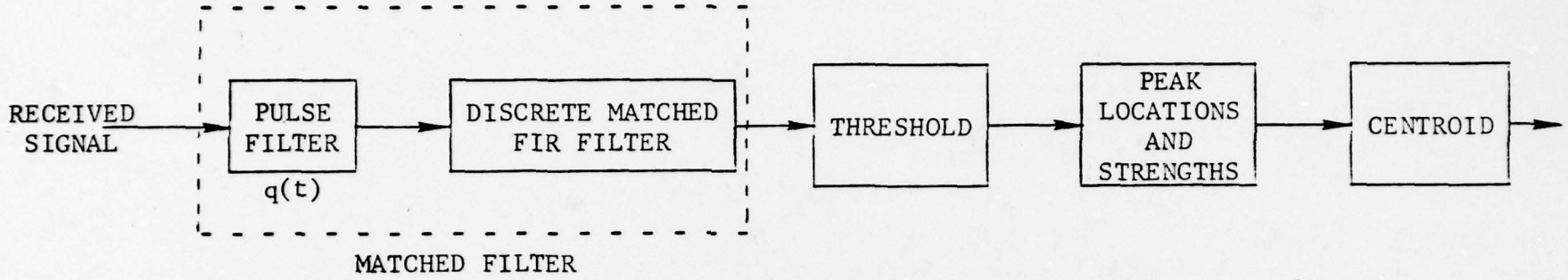


Figure 2.18 Sync Preamble Processing

$$w_1(t) = z(t) \otimes q^*(-t)$$

$$= \sum_{m=0}^{M-1} \gamma_m r(t - m\Delta) \tag{2.135}$$

The discrete matched filter has the impulse response

$$\Sigma\, \gamma^*_{-m} \delta(t - m\Delta) \tag{2.136}$$

(apart from an unimportant delay) which yields at its output the pulse train

$$w(t) = \Sigma\, C_m r(t - m\Delta) \tag{2.137}$$

where C_m is the discrete autocorrelation of the discrete sequence $\{\gamma_m\}$

$$C_m = \Sigma\, \gamma^*_\ell \gamma_{\ell+m} \tag{2.138}$$

As discussed above, $\{\gamma_m\}$ is selected so that $\{C_m\}$ has low side lobes. If $q^*(-t)$ is selected so that $r(t)$ is of the Nyquist form with zeroes at $t = m\Delta$ and normalized unit value at $t = 0$, then the good side lobes in C_m will be preserved in $w(t)$ at the sampling instants $t = m\Delta$, i.e.,

$$w(\ell\Delta) = C_\ell \tag{2.139}$$

Moreover, if $r(t)$ is small outside the interval $-\Delta < t < \Delta$, there will be limited fluctuation between sample points.

To make the complexity of the receiver pulse filter as small as possible, a recursive filter realization should be used. However, such a realization conflicts with the previous specification that the transmitted pulse be bandlimited. An approximation is clearly necessary. One may also entertain the idea of removing the receive pulse filter entirely with probably only a couple of dB loss in SNR performance for a considerably simpler receiver processing.

With proper filter and sequence design and a close enough Doppler correction, the matched filter output from an ideal single path channel will consist of a single narrow pulse plus low-level side lobe hash. Multiple paths will yield multiple pulses with low-level side lobe hash. Because of the previous signal detection and signal and noise energy measurement operations and the known relation between the transmitted Doppler preamble and sync preamble strengths, it should be possible to set a threshold well above the rms noise level (and certainly above the hash level) so that resolvable path components will be cleary visible. Algorithms can be developed for finding the location and strength of maxima, $\{\xi_m, \alpha_m\}$, above the threshold, and computing the centroid

$$\bar{\xi} = \Sigma\, \alpha_m \xi_m / \Sigma\, \alpha_m \tag{2.140}$$

From this centroid one may determine the frame sync to within an error determined primarily by the multipath spread but also by the width of r(t) and the SNR.

There are clearly certain pitfalls in the above operations. In determining the pulse threshold, one must consider the action of the AGC when the Doppler preamble is followed by the sync

preamble, because this procession may produce gain changes. In selecting maxima, one must consider that two or more paths, which are close together and not resolved, may produce subsidiary maxima that would confuse the computation. In fact, to simplify the processing, it may be sufficient to use only the largest peak in determining frame sync. This estimation procedure should produce a timing error within the range of protection of the time guard bands.

2.6.2 Performance of Sync Preamble

In this section we discuss some sources of degradation in sync preamble operation. These are seen to be Doppler correction error (Section 2.6.2.1), poor threshold setting and additive noise (Section 2.6.2.2). Terminal equipment distortion and effects of multipath sidelobes are not treated here. These latter disturbances will require an increase in the threshold setting to achieve the required false peak detection probability in comparison with the absence of these disturbances.

2.6.2.1 Effect of Doppler Correction Error

A sync preamble designed using the simple criterion that the set $\{\gamma_m;\ m = 0,1, \ldots, M-1\}$ should be chosen to provide desired correlation properties discussed in connection with (2.137) may not provide suitable protection against Doppler. Because of the inevitable (but predictably small) Doppler shift remaining after Doppler correction, sequences of one length may be preferable over sequences of another for reasons that do not derive strictly from correlation considerations.

To analyze this situation we assume that a Doppler correction error denoted F exists. In such a case the complex representation of the received sync preamble for a unity gain channel is given by

$$z(t) = e^{j2\pi Ft} \sum_{m=0}^{M-1} \gamma_m p(t - m\Delta) \tag{2.141}$$

This may be rewritten in the altered form:

$$z(t) = \sum_{m=0}^{M-1} \gamma_m e^{j2\pi mF\Delta} p(t - m\Delta) e^{j2\pi F(t - m\Delta)} \tag{2.142}$$

which may be interpreted as the transmission of an impulse train with complex amplitudes $\{\gamma_m \exp(j2\pi mF\Delta)\}$ through a pulse-shaping filter with impulse response $p(t) \exp(j2\pi Ft)$. The dominant effect concerns the "mismatching" of the impulse complex amplitudes from their correct values $\{\gamma_m\}$. The slight frequency shift of the pulse filter produced by the factor $\exp(j2\pi Ft)$ multiplying $p(t)$ will produce negligible mismatch.

The receiver may be regarded as the cascade of a pulse matched filter and a discrete filter matched to the transmitted discrete sequence. Using the same notation as in the previous section, we see that the receive pulse filter output is given by:

$$w_1(t) = z(t) \otimes q^*(-t)$$

$$= \sum_{m=0}^{M-1} \gamma_m e^{j2\pi fm\Delta} r(t - m\Delta, F) \tag{2.143}$$

where

$$r(t,F) = \int p(\tau) q^*(\tau - t) e^{j2\pi F\tau} \, d\tau \tag{2.144}$$

But since $p(\tau)$ is of duration very much less than $\frac{1}{F}$, the product $2\pi F\tau$ will be very small, and we can use

$$r(t,F) \approx r(t,0) = r(t) \tag{2.145}$$

Thus,

$$w_1(t) \approx \sum_{m=0}^{M-1} \gamma_m e^{j2\pi fm\Delta} r(t - m\Delta) \tag{2.146}$$

and the discrete matched filter has the output

$$w(t) = \sum_{m=-(M-1)}^{M-1} A_m(F) r(t - m\Delta, F) \tag{2.147}$$

where

$$A_m(F) = \begin{cases} \sum_{\ell=m}^{M-1} \gamma_\ell \gamma^*_{\ell-m} e^{j2\pi F\ell\Delta} & \ldots\ 0 \le m \le M-1 \\ \sum_{\ell=0}^{M-1+m} \gamma_\ell \gamma^*_{\ell-m} e^{j2\pi F\ell\Delta} & \ldots\ -(M-1 \le m \le 0 \end{cases} \tag{2.148}$$

is the ambiguity function of the discrete sequence $\{\gamma_m;\ m = 0, M-1\}$. When the Doppler error F is zero, this expression reduces to the autocorrelation function of the discrete sequence, i.e.,

$$A_m(0) = C_m \tag{2.149}$$

If F is too large, the side lobes of w(t) will be degraded; and if F is sufficiently large, even the main pulse lobe will be reduced. To determine sensitivity to Doppler correction error, $A_m(F)$ must be evaluated for any candidate pulse train. A conservative requirement for small degradation is that $2\pi\ell F\Delta$ be much less than unity for the maximum value of $\ell = M-1$, i.e.,

$$2\pi(M-1)F\Delta \ll 1 \tag{2.150}$$

then the exponential in (2.148) will differ little from unity. For $\Delta = 1$ ms, $M - 1 = 63$, we see that (2.150) leads to

$$F \ll 2.53 \text{ Hz} \tag{2.151}$$

which is a stringent specification on Doppler correction error.

When the channels contain L paths with relative delays ξ_ℓ, relative Doppler shifts ν_ℓ, and amplitudes g_ℓ, the matched filter output becomes

$$w(t) = \sum_{\ell=1}^{L} g_\ell \sum_k A_k(F + \nu_\ell) r(t - k\Delta - \xi_k) \tag{2.152}$$

Note that the relative Doppler shifts between multipath components can result in output pulse degradation due to the ambiguity function being evaluated at $F + \nu_\ell$. This degradation persists even if $F = 0$, and, for a given waveform, can only be reduced by shortening the duration of the sync preamble. An alternate route is to use Doppler-shift tolerant discrete sequences. The development of such sequences requires the use of computer-aided numerical design techniques, and is not attempted here because of time limitations.

To pin down the effect of the Doppler correction error on frame sync estimation we have calculated values of the ambiguity function for three sequences of interest:

1) Length-13 — Barker [2.10]

2) Length-15 — Carley [2.9]

3) Length-64 — Frank [2.3].

Results are illustrated in Figs. 2.19 through 2.21 for F in the range 0 to 2 Hz. For each sequence we note from the F = 0 entry that the desired autocorrelation properties (low side lobes) are present. For the length-13, length-15, and length-64 sequences the values of main-to-side lobe ratio ρ are 13, 15, and 24.5, respectively in the absence of Doppler offset. Clearly, with F = 0 the third choice would be the best of the three were there no residual Doppler shift.

In view of the achievable precision in estimating Doppler, as described earlier, and the Doppler spread on HF channels (a reasonable maximum would be about 1 Hz) it seems reasonable to design for F = 1 Hz. In this case the length-13, length-15, and length-64 codes have main-to-side lobe ratios of 13, 14.99, and 11.97, the length-15 code being clearly the most suitable choice. A further examination of the figures indicates that a design choice based on F = 1 Hz is still the leading contender at F = 2 Hz and it would seem that the crossover point (where the length-13 code begins to surpass the length-15 code) is far beyond the range of Doppler offsets that need to be considered.

2.6.2.2 Effect of Additive Noise and Threshold Setting

Before directly addressing the problem of the effect of additive noise we develop a necessary formalism for the received signal. Recall from (2.141) that the transmitted frame sync preamble is a pulse train with complex envelope

$$z(t) = \Sigma \gamma_m p(t - m\Delta) \qquad (2.153)$$

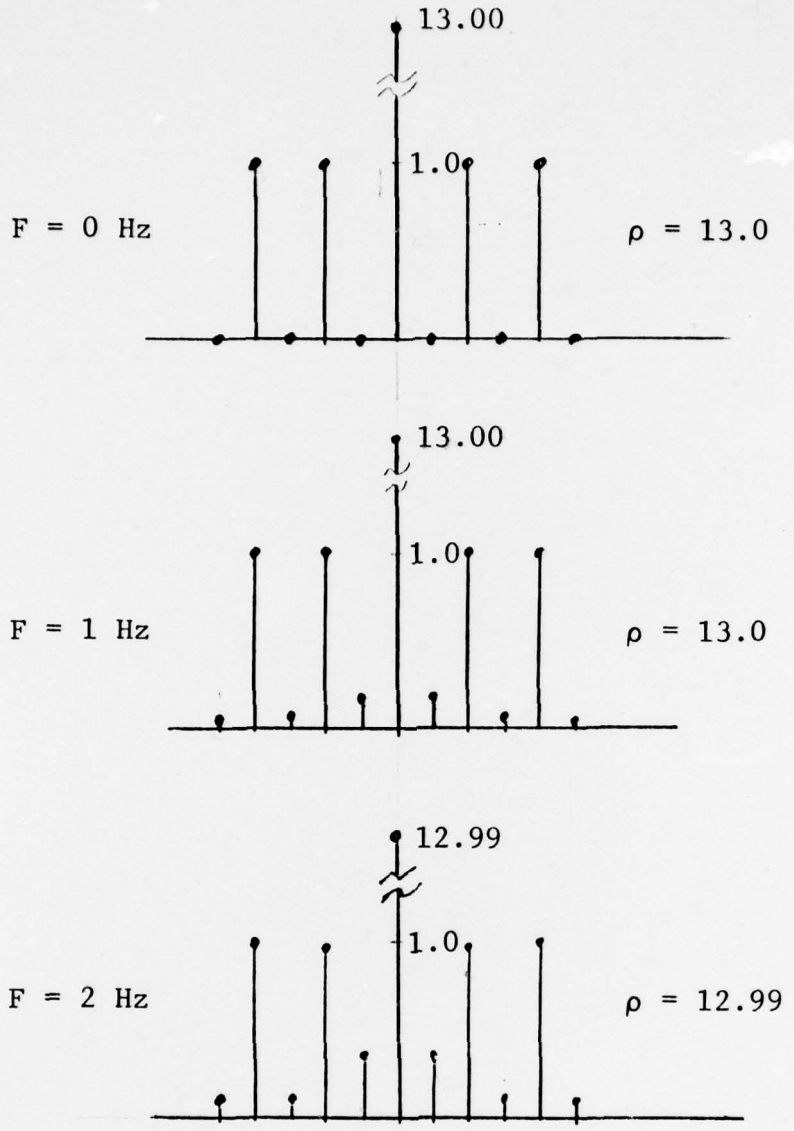


Figure 2.19 Ambiguity Function for Length-13 Sequence
$\rho \triangleq$ Main Lobe/Peak Side Lobe (Only Central Lobes of Importance are Shown)

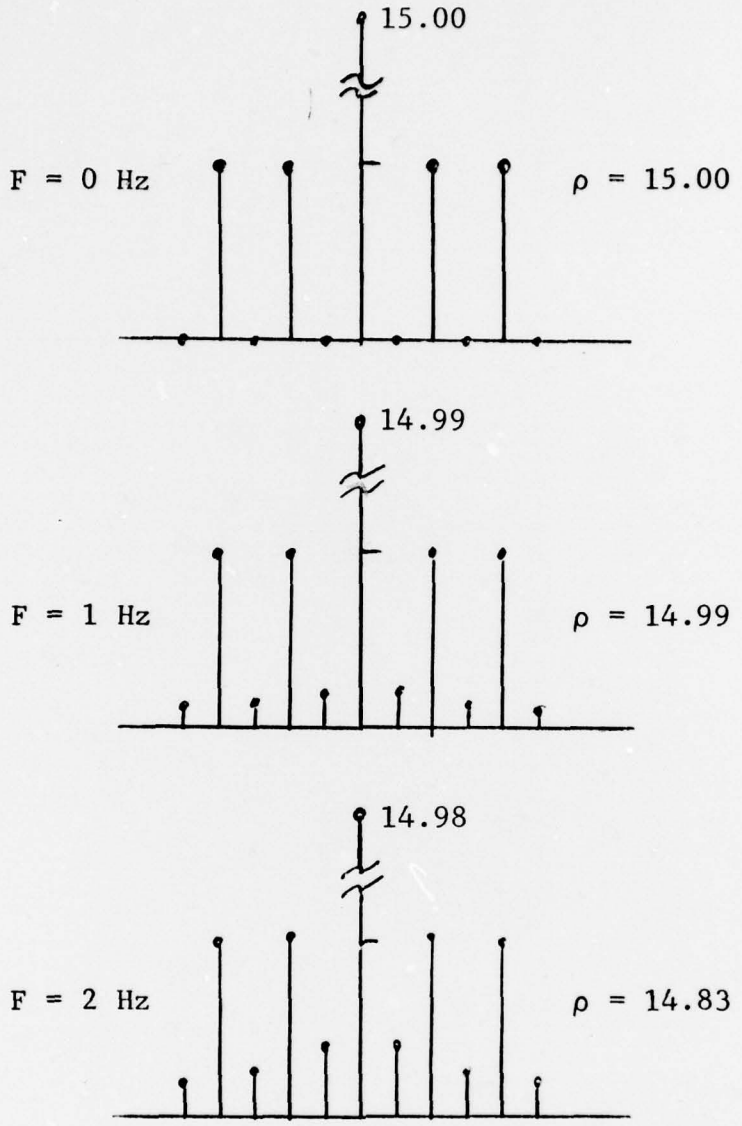


Figure 2.20 Ambiguity Function for Length-15 Sequence
$\rho \triangleq$ Main Lobe/Peak Side Lobe (Only Central Lobes of Importance are Shown)

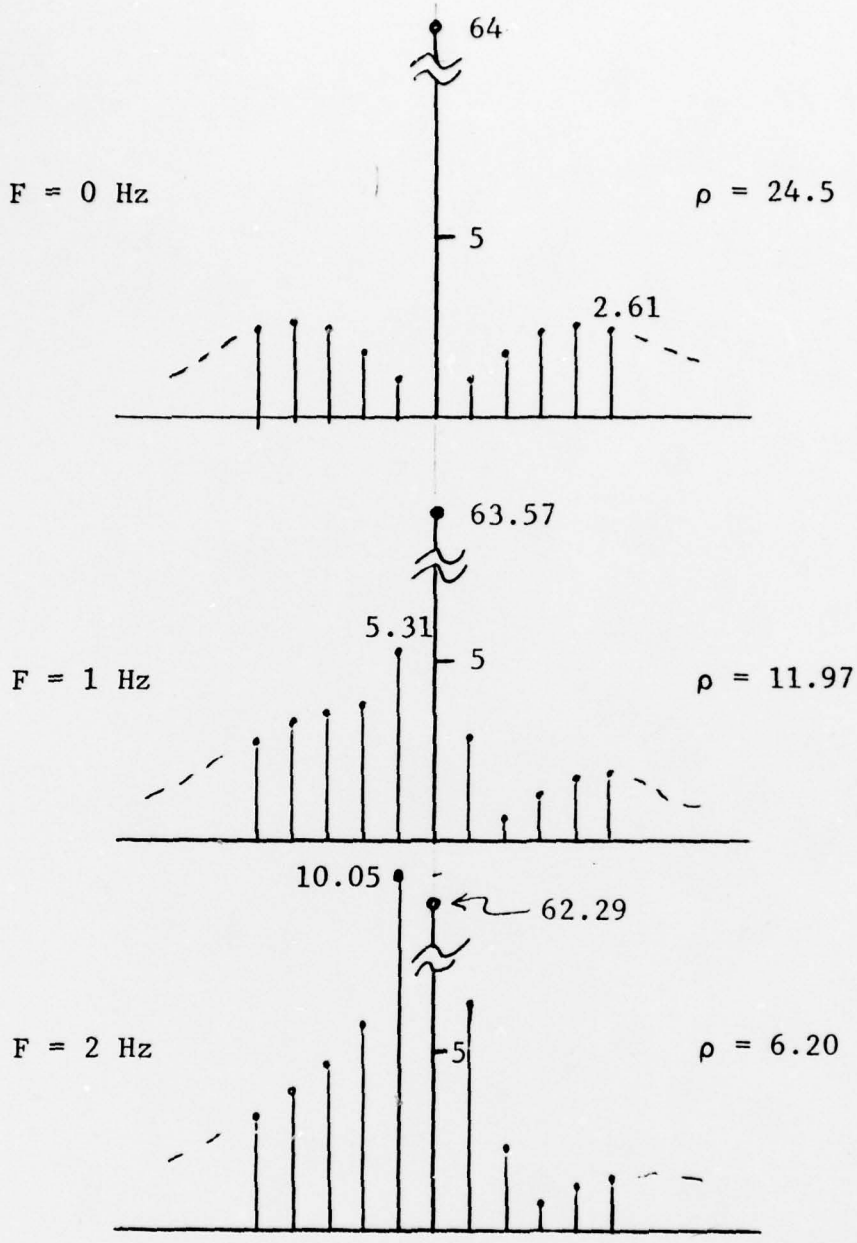


Figure 2.21 Ambiguity Function for Length-64 Sequence
$\rho \triangleq$ Main Lobe/Peak Side Lobe (Only Central Lobes of Importance are Shown)

The channel is modeled by a small set of discrete paths having amplitudes $\{g_p;\ p = 1,2, \ldots, P\}$ and delays $\{\xi_p;\ p = 1,2, \ldots, P\}$. Consequently, the received signal is given by

$$w(t) = \Sigma\, g_p z(t - \xi_p) + n(t) \tag{2.154}$$

where $n(t)$ is an assumed additive white noise.

An ideal matched filter receiver is assumed. The output of the matched filter is thus given by

$$u(t) = \int z^*(\tau - t)w(\tau)\, d\tau$$

$$= \Sigma\, g_p C(t - \xi_p) + N(t) \tag{2.155}$$

where $C(\tau)$ is the aperiodic autocorrelation function of $z(t)$,

$$C(\tau) = \int z^*(t)z(t + \tau)\, dt \tag{2.156}$$

and $N(t)$ is the additive noise at the matched filter output,

$$N(t) = \int z^*(\tau - t)n(\tau)\, d\tau \tag{2.157}$$

The autocorrelation function of $n(t)$ is given by

$$\overline{n^*(t)n(t+\tau)} = 2N_0\delta(\tau) \tag{2.158}$$

where N_0 is the one-sided power density of the real noise. Using (2.158), we readily find that

$$\overline{|N(t)|^2} = 2N_0 \int |z(t)|^2\, dt \tag{2.159}$$

The preamble autocorrelation function has been designed to have low side lobes, as discussed earlier. Using the 15 pulse length Carley sequence, the side lobe level will be 1/15 the level of the peak. This peak value is given by

$$C(0) = \int |z(t)|^2 \, dt \tag{2.160}$$

We now turn to a direct consideration of the additive noise. Additive noise can have two effects; it can randomly move the location of the correlation peaks, and it can produce new peaks exceeding the threshold. We consider each of these effects in order.

To determine the effect of noise on movement of the correlation peak, it is sufficient first to consider a single path with zero delay and complex amplitude g. Then the received signal is

$$u(t) = gC(t) + N(t) \tag{2.161}$$

In the vicinity of the tone peak location in the absence of noise, t = 0, we can represent C(t)and N(t) by Taylor series:

$$C(t) \approx C(0)\left[1 + \frac{\dot{C}(0)}{C(0)}\,t + \frac{\ddot{C}(0)}{C(0)}\,\frac{t^2}{2}\right] \tag{2.162}$$

$$N(t) \approx N(0) + \dot{N}(0)t \tag{2.163}$$

where we have retained terms to second order only (the noise is assumed small compared to signal). The envelope of the received pulse is given by

$$|\mu(t)| = |gC(t) + N(t)|$$

$$\approx |g|C(0) \left| 1 + \frac{N(0)}{gC(0)} + \left[\frac{\dot{C}(0) + \frac{\dot{N}(0)}{g}}{C(0)} \right] t + \frac{\ddot{C}(0)}{C(0)} \frac{t^2}{2} \right|$$

$$\approx |g|C(0) \left[1 + \mathrm{Re}\left\{ \frac{N(0)}{gC(0)} \right\} + \mathrm{Re}\left\{ \frac{\dot{N}(0)}{gC(0)} \right\} t + \frac{\ddot{C}(0)}{C(0)} \frac{t^2}{2} \right]$$

(2.164)

where we have assumed the perturbation terms are small compared to unity and have made use of the fact that $\dot{C}(0)$ is imaginary and $\ddot{C}(0)$ is real. In particular,

$$\frac{\ddot{C}(0)}{C(0)} = -(2\pi)^2 \frac{\int f^2 |Z(f)|^2 \, df}{\int |Z(f)|^2 \, df} \tag{2.165}$$

where $Z(f)$ is the preamble spectrum. Defining the rms bandwidth of $z(t)$ as

$$W = 2 \sqrt{\frac{\int f^2 |Z(f)|^2 \, df}{\int |Z(f)|^2 \, df}} \tag{2.166}$$

we see that

$$\frac{\ddot{C}(0)}{C(0)} = -\pi^2 W^2 \tag{2.167}$$

Completing the square in the last approximation in (2.164),

$$\mu(t) \approx |g|C(0) \left[1 + \mathrm{Re} \left\{ \frac{\dot{N}(0)}{gC(0)} \right\} + \epsilon^2 \frac{\pi^2 W^2}{2} - \frac{\pi^2 W^2}{2} (t - \epsilon)^2 \right] \tag{2.168}$$

where

$$\epsilon = \frac{1}{\pi^2 W^2} \mathrm{Re} \left\{ \frac{\dot{N}(0)}{gC(0)} \right\} \tag{2.169}$$

In the rewritten form (2.168), it is seen by inspection that the maximum has been shifted from $t = 0$ to $t = \epsilon$.

The rms value of this shift will now be computed:

$$\overline{\epsilon^2} = \frac{1}{\pi^4 W^4} \frac{1}{2} \frac{\overline{|\dot{N}(0)|^2}}{|g|^2 C^2(0)} \tag{2.170}$$

where use has been made of the following relation for a complex Gaussian variable N:

$$\overline{\mathrm{Re}^2\{N\}} = \frac{1}{2} \overline{|N|^2} \tag{2.171}$$

From (2.157) and (2.158) we see that

$$\overline{|\dot{N}(0)|^2} = 2N_0 \int |\dot{z}(t)|^2 \, dt \tag{2.172}$$

Utilizing Parseval's theorem, we find that

$$\overline{|\dot{N}(0)|^2} = 2N_0(2\pi)^2 \int f^2 |Z(f)|^2 \, df$$

$$= 2N_0\pi^2W^2C(0) \tag{2.173}$$

Substituting this expression in (2.170) gives

$$\overline{\epsilon^2} = \frac{N_0}{\pi^2W^2|g|^2C(0)} \tag{2.174}$$

The received signal energy over the single path is given by

$$\mathcal{E} = \frac{1}{2}|g|^2 \int |z(t)|^2 \, dt$$

$$= \frac{1}{2}|g|^2C(0) \tag{2.175}$$

In terms of this energy,

$$\overline{\epsilon^2} = \frac{1}{\pi^2W^2\frac{2\mathcal{E}}{N_0}} \tag{2.176}$$

Defining an average power for the received signal,

$$P = \frac{\mathcal{E}}{T} \tag{2.177}$$

where T is the duration of the preamble,

$$\sqrt{\overline{\epsilon^2}} = \frac{1}{\sqrt{2}\,\pi W} \cdot \frac{1}{\sqrt{T\frac{P}{N_0}}} \tag{2.178}$$

To be specific, we assume a raised cosine spectrum for the preamble:

$$Z(f) = \cos^2 (\pi f/B) \quad ; \quad |f| < \frac{B}{2} \tag{2.179}$$

The rms bandwidth W for this spectrum is given by

$$W^2 = \frac{4 \int_{-B/2}^{B/2} f^2 \cos^4 (\pi f/B)\, df}{\int_{-B/2}^{B/2} \cos^4 (\pi f/B)\, df}$$

$$= \frac{4B^2}{\pi^2} \frac{\int_0^{\pi/2} x^2 \cos^4 x\, dx}{\int_0^{\pi/2} \cos^4 x\, dx} \tag{2.180}$$

Upon evaluating the integrals in (2.180), we find

$$W^2 = \frac{4B^2}{\pi^2} (.197) \tag{2.181}$$

$$\sqrt{2}\, \pi W = 1.26\ B \tag{2.182}$$

so that

$$\sqrt{\overline{\epsilon^2}} = \frac{1}{1.26\ B} \frac{1}{\sqrt{T \frac{P}{N_0}}} \tag{2.183}$$

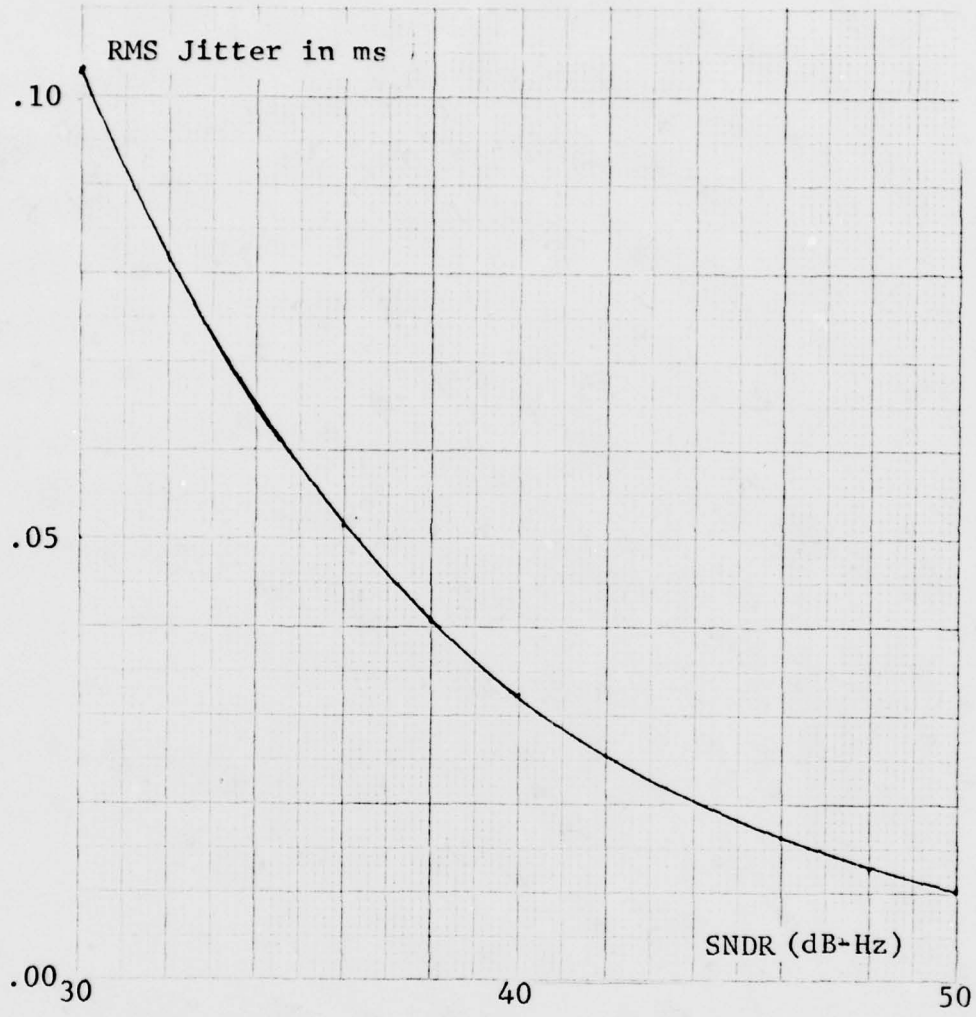
The dependence of this expression on P/N_0 is illustrated in Fig. 2.22 over a range of interest. Note, at 40 dB-Hz that the rms jitter of the peak is only .03 ms, which is negligible. It may thus be concluded that the jitter of the correlation peaks caused by additive noise in the ranges of interest is small enough to be of second-order effect since the guardband available will be of the order of 4 ms and the multipath spread (separation between correlation peaks for all paths) can be a substantial fraction of 4 ms.

We now turn to consideration of the spurious threshold exceedances that may occur due to noise.

There exists a finite probability that noise peaks will exceed the threshold used to define correlation peaks. It is of interest to determine the height of the threshold required above the additive noise in order to make the probability very small that one noise peak will exceed the threshold in the vicinity of the expected a priori region of correlation peak arrivals.

To make the subsequent calculations simple, we shall assume that the spectrum of the preamble is flat over B Hz instead of being a raised cosine shape. This will result in a probability of threshold exceedance somewhat larger than would be true for the raised cosine shape, which is a conservative approach to system design.

We assume that the matched filter output (complex envelope) is sampled at a rate of B samples per second (a somewhat higher sampling rate would probably be used in the actual system, but this will not appreciably change our results). An uncertainty



Sync Preamble Duration = 15 ms

Available Bandwidth = 2 kHz

Figure 2.22 RMS Jitter of Correlation Peak (in ms) as a Function of Signal-to-Noise Density Ratio. Pulse Spectrum is a Raised Cosine

region of duration T_0 is assumed to exist wherein the correlation peaks from the various paths would be expected to arrive. The size of this uncertainty region has impact on the sync preamble's vulnerability to noise; the tendency is toward degradation of performance as the region is widened. This was one of the major motivating factors for the particular preamble timing discussed in Section 2.2. We recall from arguments given there that the actual observation (search for the correlation peak) is made over $K_2 \times \Delta$ seconds (also see Figs. 2.2 and 2.5). From the specifications of that earlier section ($K_2 = 45$ and $\Delta = 1$ ms) we will have occasion to later use the specific value $T_0 = 45$ ms. For the time being, however, we will retain T_0 as a general unspecified variable.

With a flat spectrum of duration B for the matched filter, the output noise will also have a flat spectrum of duration B and samples taken at a rate of B/second will be statistically independent. In a duration T_0 there will be

$$M = BT_0 \tag{2.184}$$

independent noise samples.

We are interested in the probability p that one or more noise instantaneous power samples exceed a threshold. Along these lines we define

$$\gamma = |N(t)|^2$$

$$\Gamma = \bar{\gamma} = \overline{|N(t)|^2}$$

$$= 2N_0 \int |z(t)|^2 \, dt \tag{2.185}$$

the last equality following from (2.159). We are interested in the probability that one sample or more of γ/Γ exceeds a fixed threshold ρ_0. This probability must satisfy the inequality $p \leq p_0$, where

$$p_0 = \sum_{m=1}^{M} P_r\left(\frac{\gamma(t_j)}{\Gamma} > \rho_0\right)$$

$$= MP_r\left(\frac{\gamma}{\Gamma} > \rho_0\right) \tag{2.186}$$

The bound is very tight as long as $p_0 << 1$ which may be assumed here.

The value of p_0 is easily determined by noting that the assumption of complex Gaussian statistics for the noise implies that γ is exponentially distributed, i.e., its pdf is

$$p(\gamma) = \frac{1}{\Gamma} e^{-\frac{\gamma}{\Gamma}} \tag{2.187}$$

which yields

$$p_0 = Me^{-\rho_0} \tag{2.188}$$

A numerical example of interest here utilizes B = 2 kHz, $\rho_0 = 15$, and $T_0 = 45$ ms, the values currently under consideration for the preamble. From (2.184) we have M = 90. Thus, $p_0 = 2.75 \times 10^{-5}$.

The above behavior is not surprising; as one increases the threshold, the probability of threshold exceedance by noise alone decreases exponentially. The threshold cannot be placed too high, however, because the correlation peaks of the paths, at least the strongest path, must exceed the threshold. Recall that the decision variable in the absence of sync preamble is simply the normalized noise variable γ/Γ. It is of interest to determine the value of the decision variable at the correlation peak in the absence of noise. Along these lines we note that the squared correlation peak for the p'th path is given by

$$S_p = |g_p|^2 \int |z(t)|^2 \, dt \Big)^2 \tag{2.189}$$

so that

$$\frac{S_p}{N} = \frac{|g_p|^2 \int |z(t)|^2 \, dt}{2N_0} = \frac{\mathcal{E}_p}{N_0} = T \frac{P_p}{N_0} \tag{2.190}$$

where $\mathcal{E}_p$ is the preamble energy received over the p'th path, and P_p is the corresponding average preamble power. Assuming T = 15 ms and P_p/N_0 = 40 dB-Hz, S_p/N = 150. The peak power in the side lobes of the output pulse will be down by a factor of 15^2 = 225, and thus well in the ambient noise. For the example cited in setting the noise threshold, the correlation peak will exceed the threshold by 150 - 15 = 135, which is large enough to make the probability of noise cancelling the peak negligible. Reducing the value of P_p/N_0 by a factor of 5 (to around 33 dB) results in a correlation peak/noise power ratio of 30 and a threshold exceedance of 15, which is still large enough to make the probability of noise cancelling a signal pulse negligible.

The considerations described above indicate that the effects of noise (jitter of the correlation peak and decision error) will be minimal. Additionally, they have presented us with some reasonable preliminary design specifications.

1) Formulate decision variables $|S(t_i)|^2/\hat{\Gamma}$ $i = 1, M$ where $S(t)$ represents the sync preamble received in noise and $\hat{\Gamma}$ is an estimate of the reference noise level Γ given in (2.185).

2) Test each of the M samples against a threshold ρ_0. As an initial choice we use the specific value $\rho_0 = 15$.

There are three questions relative to the above comments that we have not been able to pursue in depth because of time limitations.

The first relates to the estimate of reference noise level in 1) above. Presumably, this estimate could derive from the measurements made during the processing for detection of signal presence. It could also derive from measurements made subsequent to that point if collection of more noise measurements were found to provide a decided advantage. In any event, it should be clear from the nature of the decision variable that the basic probabilistic conclusions described above would not be greatly affected by any but very large fractional errors in estimating Γ. There seem to be several avenues for guaranteeing that such large errors will be extremely improbable.

The second consideration relates to the AGC which must be kept from changing appreciably when the sync portion of the preamble enters the receiver. If necessary, to hold the receiver AGC

constant, one may transmit out-of-band tones that do not get through the matched filter, a procedure that may also be employed following the Doppler estimation procedure.

The third question relates to the effect of multipath and equipment distortion on the sync preamble peak-to-sidelobe ratio and on threshold setting. The existence of two or more paths may result in superposition of sidelobes and spurious threshold exceedances. In such a case it is felt that the high performance levels described here, with some modification of operating parameter values, could still be attained. This would be achieved via several avenues. As an example we point out that the length of the synchronization sequence has in no way been optimized. We have chosen from a very small set. There is every reason to believe that going to longer sequences would allow greater flexibility in threshold setting and substantially increase performance. Further examination of multipath effects (and methods for overcoming them) is recommended.

Though of importance, the AGC and noise level questions raised above, once corrected in the straightforward manner suggested, are not expected to impact preamble performance to the same extent as the fundamental considerations emphasized in this section.

SECTION 2

REFERENCES

[2.1] A. H. Nuttall, "Phasing and/or Clipping Tones for a Peak Amplitude Limitation," NUSC Technical Report 4921, May 1975.

[2.2] R. A. Shepard and J. B. Lomax, "Frequency Spread in Ionospheric Radio Propagation," IEEE Trans. on Comm. Tech., Vol. 15, No. 2, April 1967, pp. 268 - 275.

[2.3] R. L. Frank, "Polyphase Codes With Good Nonperiodic Correlation Properties," IEEE Trans. on Information Theory, January 1963, pp. 43 - 45.

[2.4] J. Omara and T. Kailath, "Some Useful Probability Distributions," Technical Report No. 7050-6, Stanford Electronic Laboratories, Stanford University, September 1965.

[2.5] G. L. Turin, "The Characteristic Function of Hermitian Quadratic Forms in Complex Normal Variables," Biometrika, Vol. 47, Pts. 1, 2, June 1960, pp. 149 - 201.

[2.6] D. Chase, et al., "Franklin Study," CNR Report No. 16-13, 14 October 1975. Final Report on Contract DAAB03-74-C-0494.

[2.7] S. O. Rice, "Statistical Properties of a Sine Wave Plus Random Noise,"BSTJ, Vol. 27, January 1948, p. 127.

[2.8] G. H. Hardy, et al., Inequalities, Cambridge University Press, 1959, p. 17.

[2.9] G. Carley, Unpublished Report, M.I.T., 1963; Mentioned by Golomb and Sholz in "Generalized Barker Sequences," PGIT, January 1965, pp. 533 - 537.

[2.10] R. H. Barker, "Group Synchronizing of Binary Digital Systems," Communications Theory, London; Butterworth, 1953, pp. 273 - 287.

SECTION 3

CODING FOR A KG SYNCHRONIZATION SEQUENCE

In this section, good block codes are evaluated for the protection of the KG synchronizing sequence. This KG sync sequence may actually be composed of the combination of a KG synchronizing sequence and a small number of bits used for net control. Sequences of lengths 100, 128, and 150 are assumed. The number of tones used to code the sequence varies from 250 to 4000 which (assuming approximately 40 tones per frame) covers the range of 6 to 100 frames. With 22.5 ms per frame, coding delays in the range of .135 second to 2.25 seconds are considered. While the actual specification limits the KG decoding delay to 68 frames, or 1.53 seconds, performance curves* for delays up to 2.25 seconds are still of interest.

Since the modulation is 4-phase DPSK with 2 bits per tone, the number of tones used is given by $\frac{L \cdot N}{2}$, where N is the block code length and L is the order of diversity combining for each coded bit. The best BCH codes are chosen for the simple reason that they can be decoded easily. The codes and diversities used are shown in Table 3-1. Note that the total number of bits used to protect the 128-bit KG sequence is a multiple of 252 rather than a multiple of 250 which is used for the 100-bit KG and 150-bit KG sequences.

---

*In this section, performance curves for bit error probabilities are presented. The analyses for computing the bit and block error rate are given in Appendix D.

TABLE 3-1

CODES FOR SYNCHRONIZATION SEQUENCE (ℓ = 1, 2, 4, 8)

Length of Sequence in Bits	Number of Tones	Code	Order of Diversity
100	250	(125,100;8)*	4
100	250	(250,100;46)†	2
100	250	(500,100;120)	1
100	500ℓ	(125,100;8)*	8ℓ
100	500ℓ	(250,100;46)†	4ℓ
100	500ℓ	(500,100;120)	2ℓ
100	500ℓ	(1000,100;341)	ℓ
128	252	(252,128;37)	2
128	252	(504,128;110)‡	1
128	504ℓ	(252,128;37)	4ℓ
128	504ℓ	(504,128;110)‡	2ℓ
128	504ℓ	(1008,128;253)	ℓ
150	250	(250,150;27)	2
150	250	(500,150;96)**	1
150	500ℓ	(250,150;27)	4ℓ
150	500ℓ	(500,150;96)**	2ℓ
150	500ℓ	(1000,150;245)	ℓ

*Obtained from (125,103;8) code

†Obtained from (250,101;46) code

‡Obtained from (504,131;110) code

**Obtained from (500,155;96) code.

To illustrate the effects of fading bandwidth and frame rate on the KG sequence performance, the hard decoding results are used for comparison. The results plotted in Figs. 3.1 through 3.7 are limited to 500 tones and hard decoding. Table 3-2 shows the different values for these seven figures. As can be seen from these figures, the best codes are the codes with length 250 or 252 using fourth-order diversity for all cases. The signal-to-noise ratio per received tone (E_t/N_0) in dB for each code to reach a bit error rate of 10^{-7} for two frame rates and three different rms fading bandwidths are given in Table 3-3. In the worst case, i.e., B = 5 Hz at a frame rate of 44.44 frames/second, the (250,150;27) code needs an additional 1.7 dB to reach 10^{-7} when compared to the case B = 0. Therefore, the effect of the rms fading bandwidth B on the performance is quite small. This is especially true for the stronger codes.

In Fig. 3.8, the 1000 or 1008 tone KG codes are compared for the slowly fading channel only. It is expected that the performance is fairly independent of the rms fading bandwidth B. In this case, the codes with lengths of 250 and 252 again yield the best performance.

In the remainder of this section, the results are extended to cover both soft and hard decodings, and the number of tones per block range from 250 (or 252) to 4000 (or 4032). From the results given in Figs. 3.1 through 3.7 we note that, for a fixed fading bandwidth, the results for a frame rate = 44.44 frames/second are inferior to those that are obtained for a frame rate = 75 frames/second. Hence, in the following discussion,

TABLE 3-2

FRAME RATES AND RMS FADING BANDWIDTH
FOR FIGURES 3.1 THROUGH 3.7

Figure	Frame Rate (frames/sec)	RMS Fading Bandwidth (Hz)
3.1	44.44 or 75	0
3.2	44.44	1
3.3	44.44	3
3.4	44.44	5
3.5	75	1
3.6	75	3
3.7	75	5

TABLE 3-3

SNR PER RECEIVED TONE REQUIRED TO REACH AN ERROR RATE OF 10^{-7} USING FOURTH-ORDER DIVERSITY AND HARD DECODING

RMS Fading Bandwidth (Hz)	Frame Rate (frames/sec)	(250,100;46) Code	(252,128;37) Code	(250,150;27) Code
0	44.44 or 75	4.6 dB	5.4 dB	6.5 dB
1	44.44	4.7 dB	5.5 dB	6.6 dB
3	44.44	5.0 dB	5.9 dB	7.1 dB
5	44.44	5.8 dB	6.7 dB	8.2 dB
1	75	4.7 dB	5.4 dB	6.5 dB
3	75	4.8 dB	5.6 dB	6.7 dB
5	75	5.0 dB	5.9 dB	7.1 dB

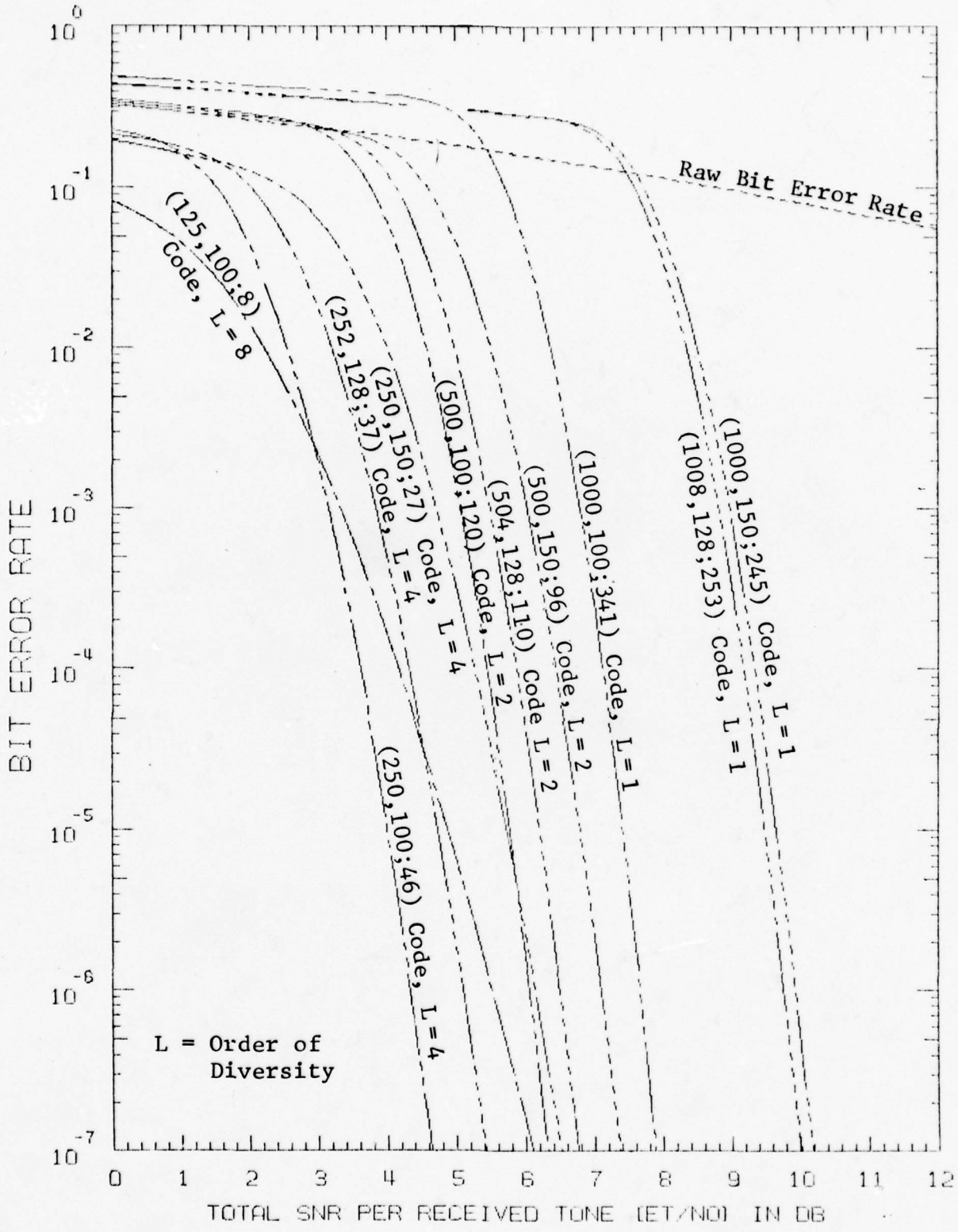


Figure 3.1 Performance of 500- or 504-Tone KG Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (Hard Decoding)

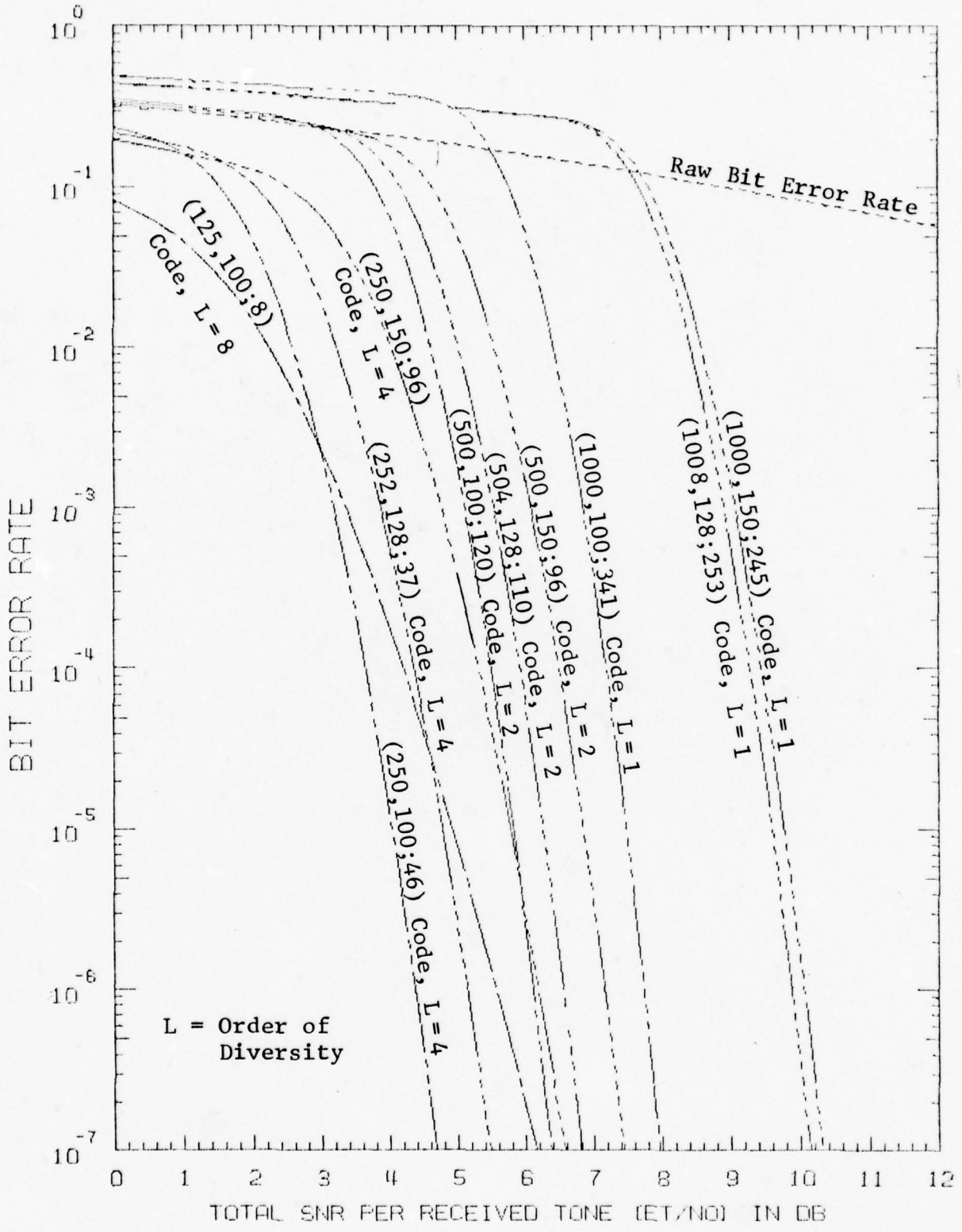


Figure 3.2 Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 1 Hz, Hard Decoding)

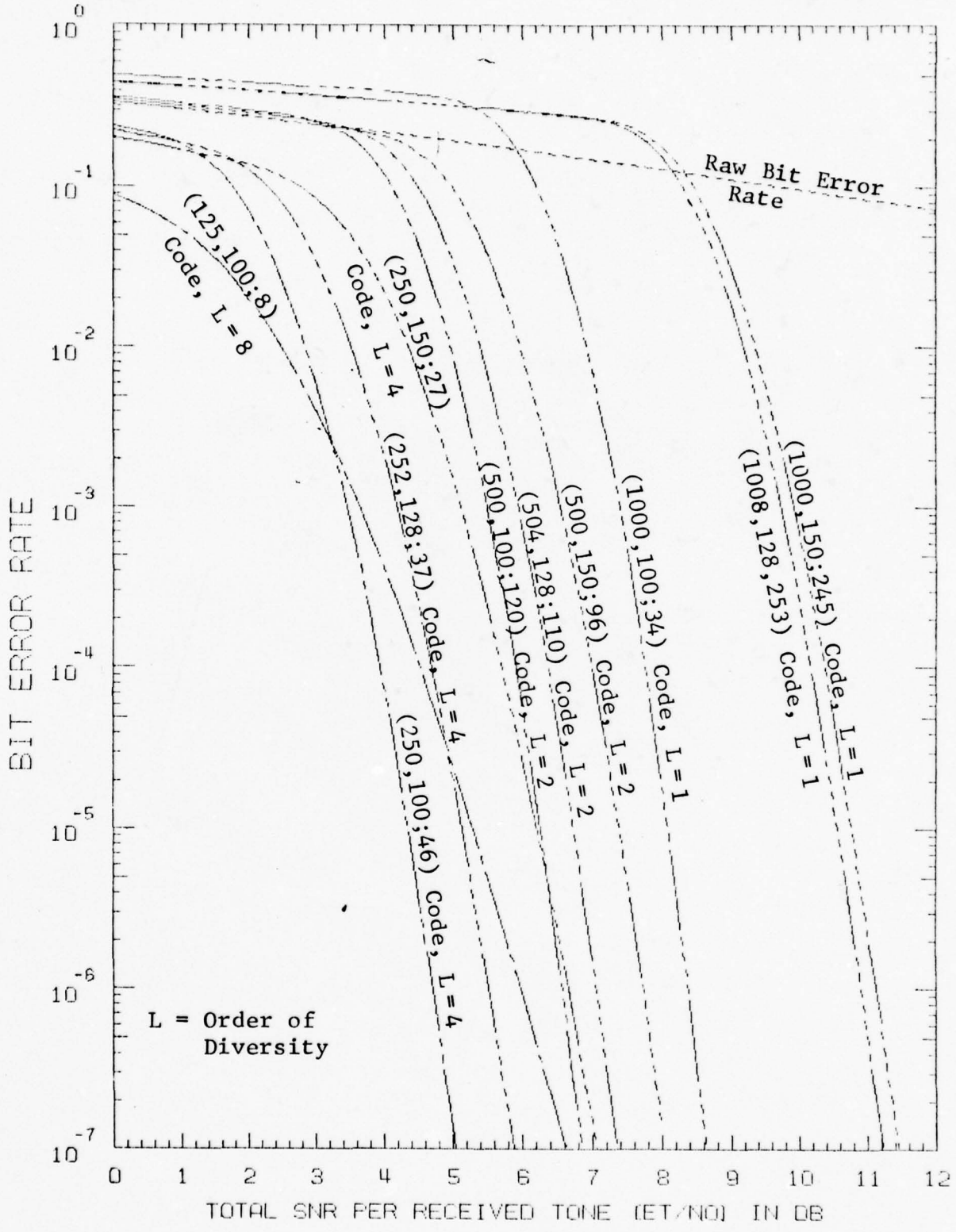


Figure 3.3 Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 3 Hz, Hard Decoding)

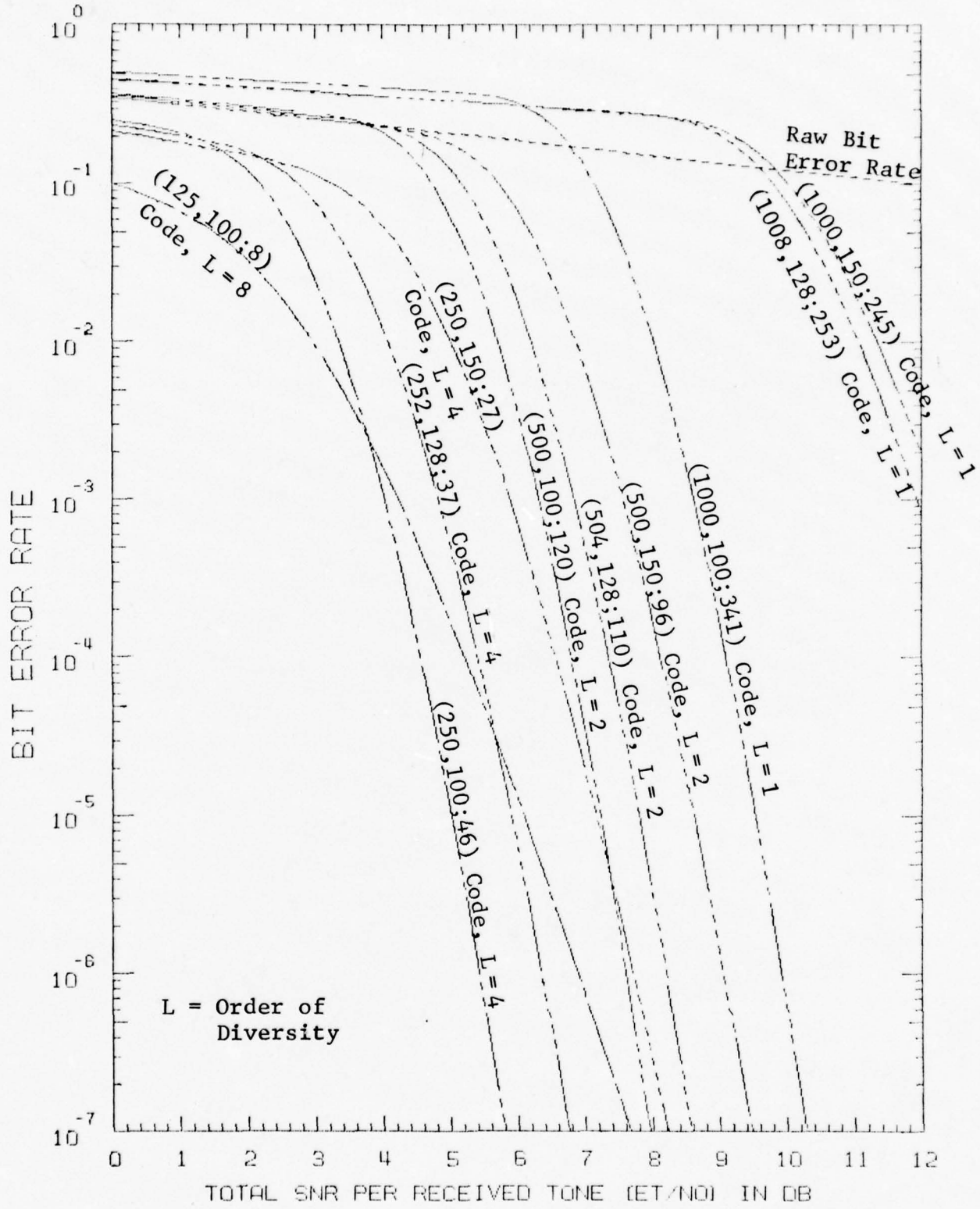


Figure 3.4 Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5, Hard Decoding)

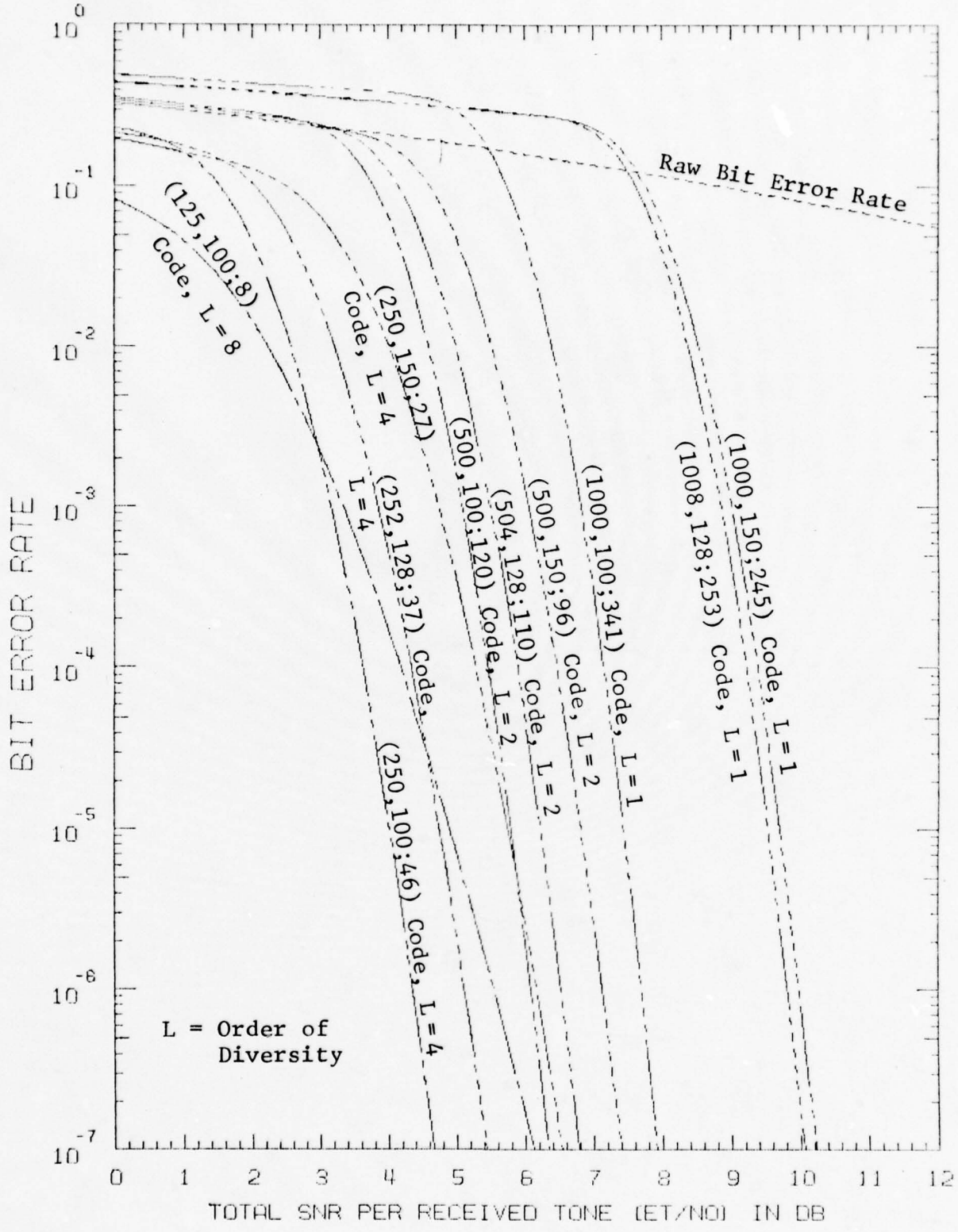


Figure 3.5 Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 75 frames/sec, B = 1 Hz, Hard Decoding)

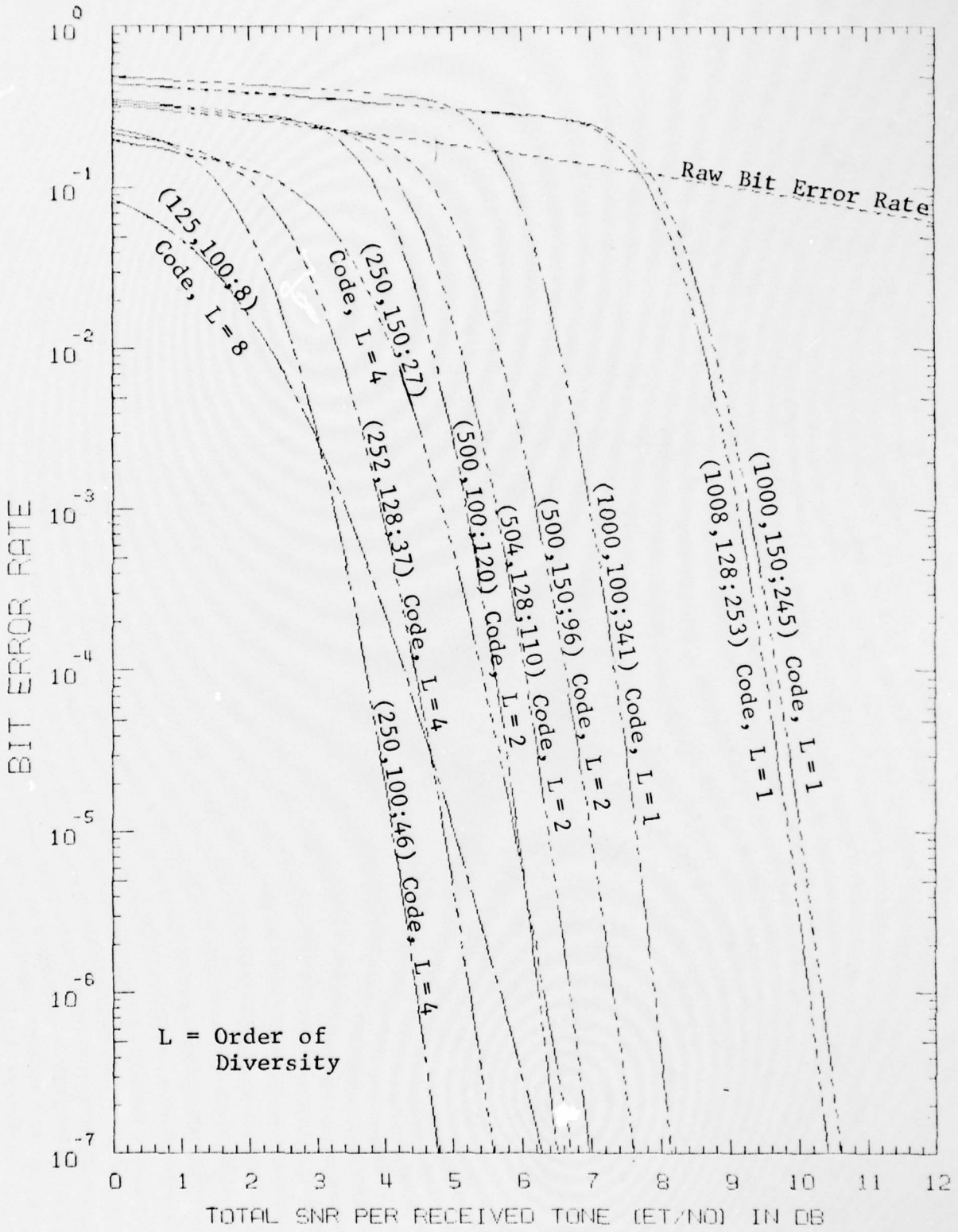


Figure 3.6 Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 75 frames/sec, B = 3 Hz, Hard Decoding)

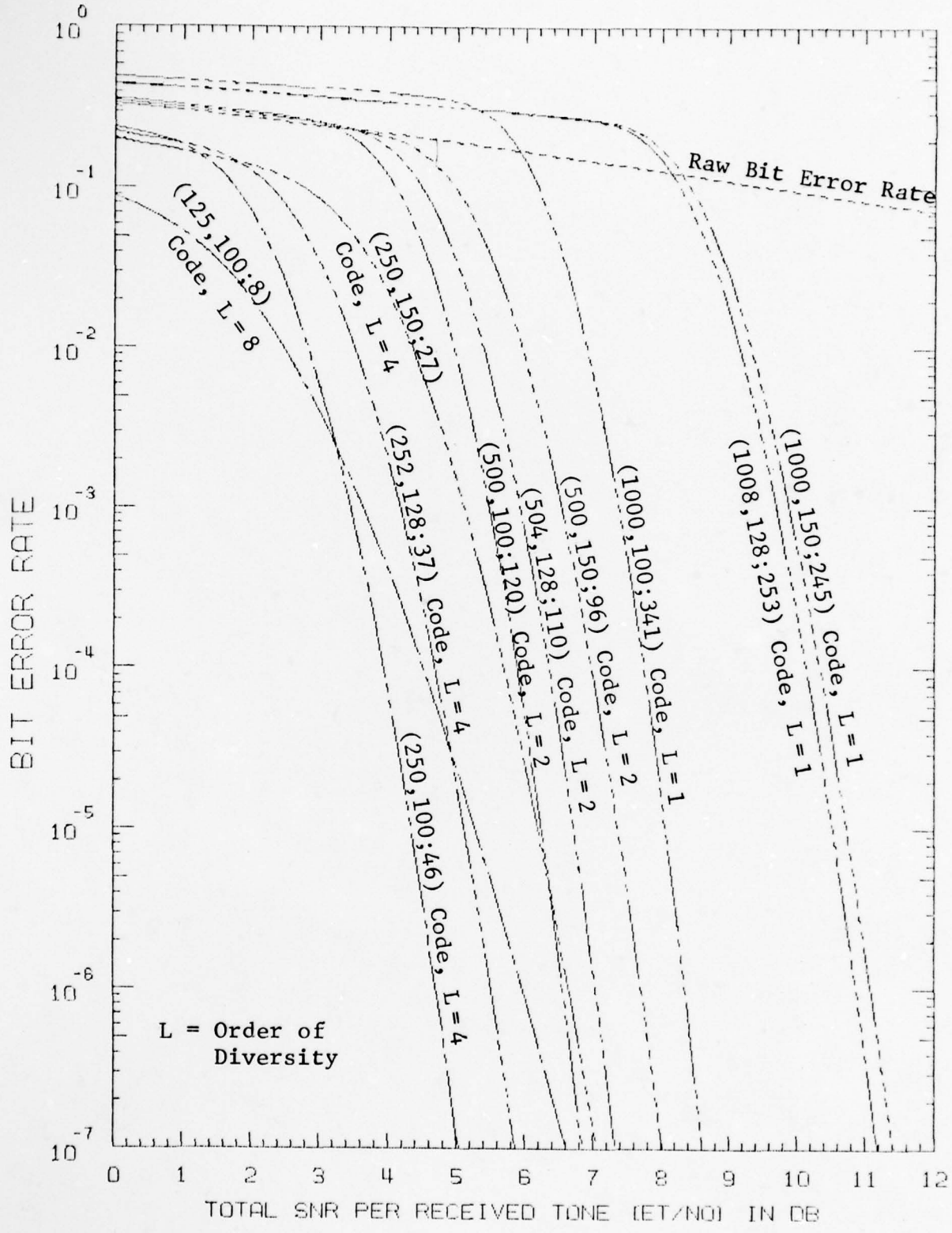


Figure 3.7 Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 75 frames/sec, B = 5 Hz, Hard Decoding)

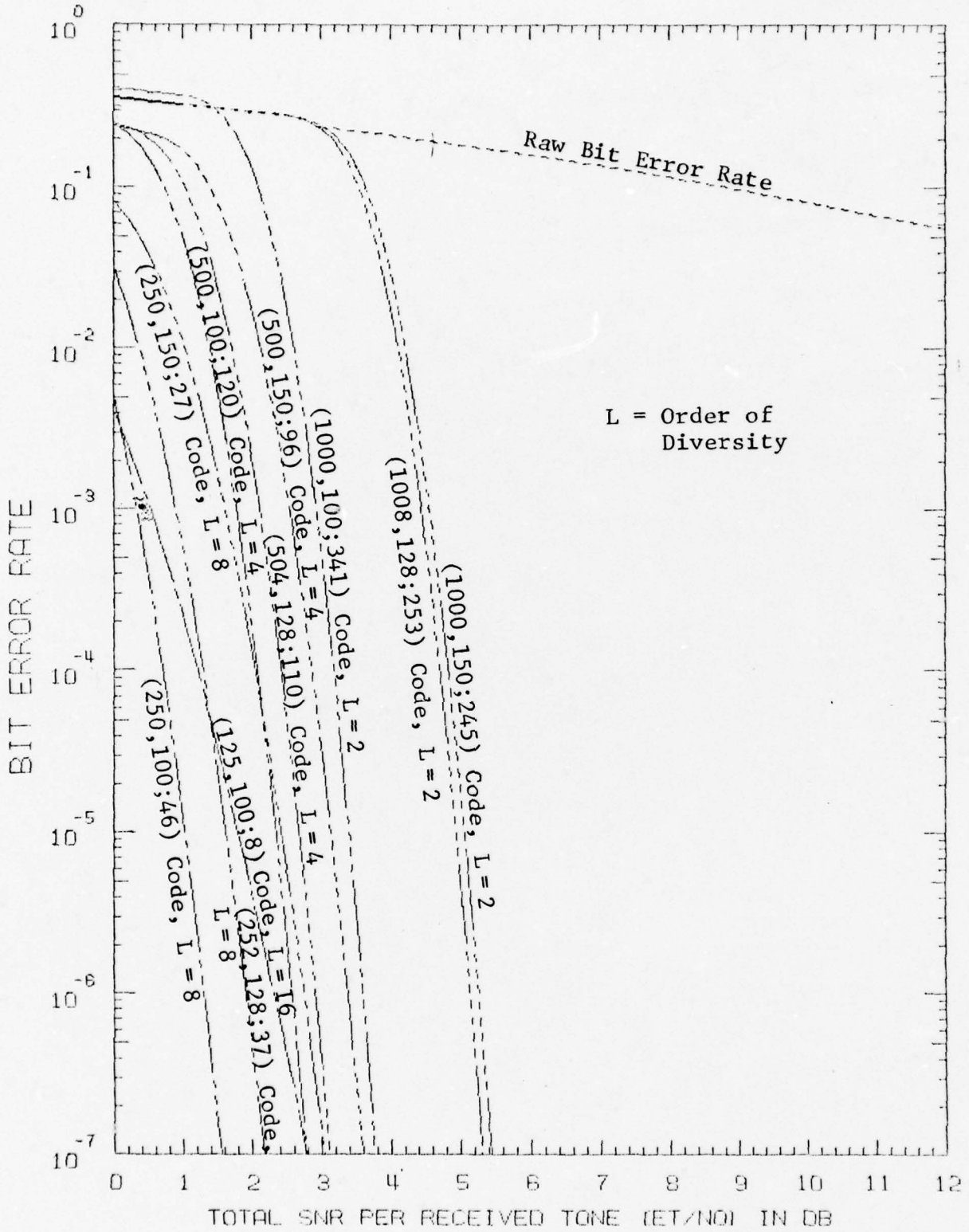


Figure 3.8 Performance of 1000- or 1008-Tone KG Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (Hard Decoding)

only the worst case will be considered. Also, when the number of tones/code block becomes larger, a large amount of diversity combining is used for each bit in a code block. As a result, the performance is fairly independent of the rms Doppler bandwidth B. Thus, only the case B = 5 Hz, the worst case, is shown when the number of tones/block $\geq$ 1000.

Table 3-4 lists all the pertinent information for Figs. 3.9 through 3.26.

For hard decoding, the three codes of code length 250 or 252 yield the best performance. This is also true for most soft decoding cases. [The noted exceptions are the soft decoding estimates for the (1000,100;341) code. For the case of 4000 or 4032 tones, the (1000,100;341) code outperforms the (250,100;46) code by 0.5 dB, and the (500,150;96) code edges the (250,150;96) code by 0.3 dB at a bit error rate of 10^{-7}. However, it is much easier to soft decode the shorter code than the longer code. Also, a better algorithm can be applied to the shorter codes to compensate for the 0.5 to 0.3 dB estimated loss in performance.] Therefore, the three codes of length 250 or 252 are recommended to be used for KG codes. In Table 3-5, the signal-to-noise ratios per tone (E_t/N_0) required to reach a bit error rate of 10^{-7} are listed. The required signal-to-noise ratios per received tone (E_t/N_0) for the KG sequences to reach a bit error rate of 10^{-7} vs. the number of tones employed for the KG sequences are plotted in Fig. 3.27 on the top horizontal scale. For the same delay, a savings of 2 dB in signal-to-noise ratio results if soft decoding is employed instead of hard decoding. Note that

TABLE 3-4

PARAMETERS FOR FIGURES 3.9 THROUGH 3.26

Figure	Tones/Code Block	RMS Doppler Bandwidth (Hz)	Decoding Algorithm
3.9	250 or 252	0	Hard
3.10	250 or 252	1	Hard
3.11	250 or 252	3	Hard
3.12	250 or 252	5	Hard
3.13	250 or 252	0	Soft
3.14	250 or 252	1	Soft
3.15	250 or 252	3	Soft
3.16	250 or 252	5	Soft
3.17	500 or 504	0	Soft
3.18	500 or 504	1	Soft
3.19	500 or 504	3	Soft
3.20	500 or 504	5	Soft
3.21a & b	1000 or 1008	5	Hard
3.22	1000 or 1008	5	Soft
3.23	2000 or 2016	5	Hard
3.24	2000 or 2016	5	Soft
3.25	4000 or 4032	5	Hard
3.26	4000 or 4032	5	Soft

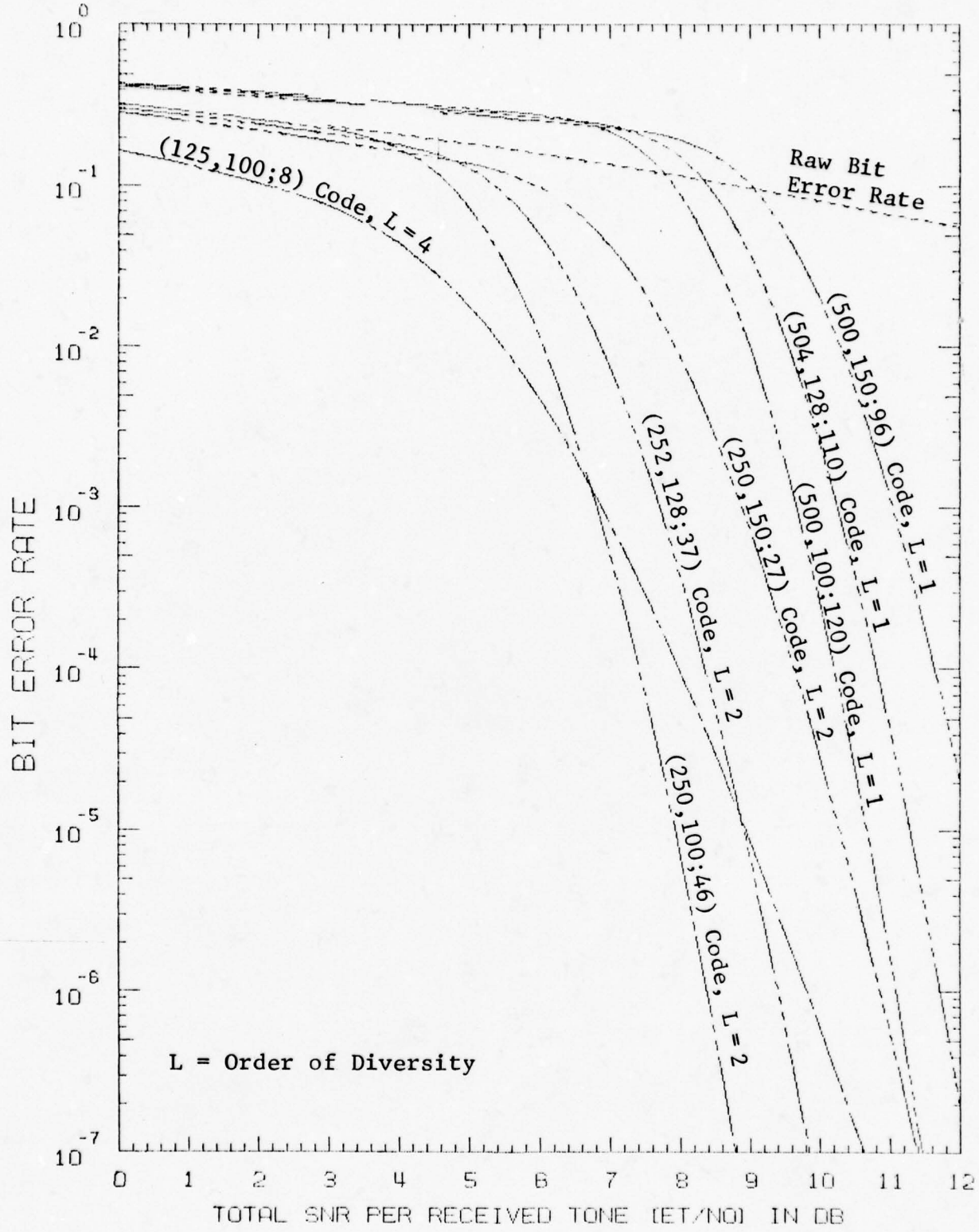


Figure 3.9 Performance of 250- or 252-Tone KG Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (Hard Decoding)

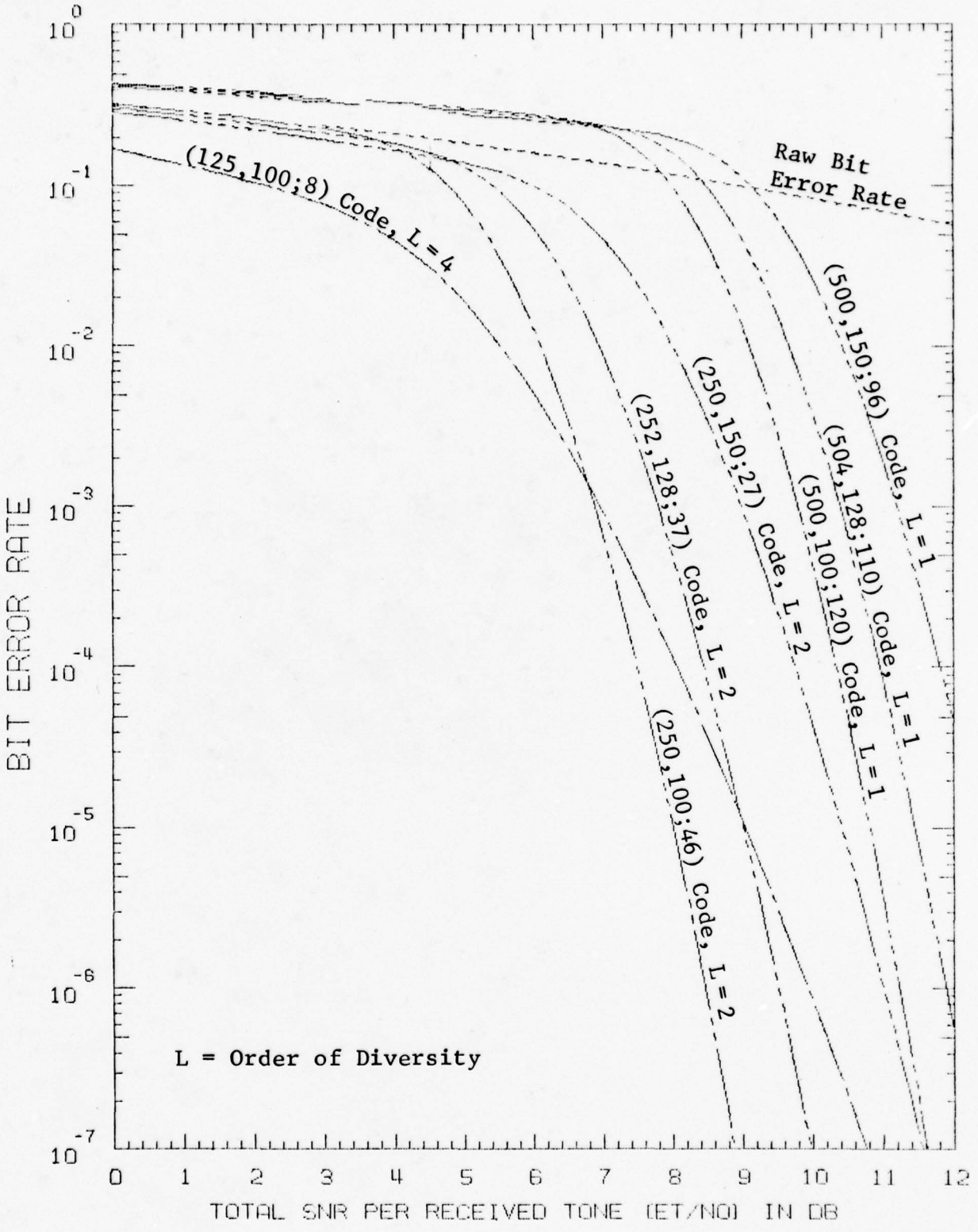


Figure 3.10 Performance of 250- or 252-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 1 Hz, Hard Decoding)

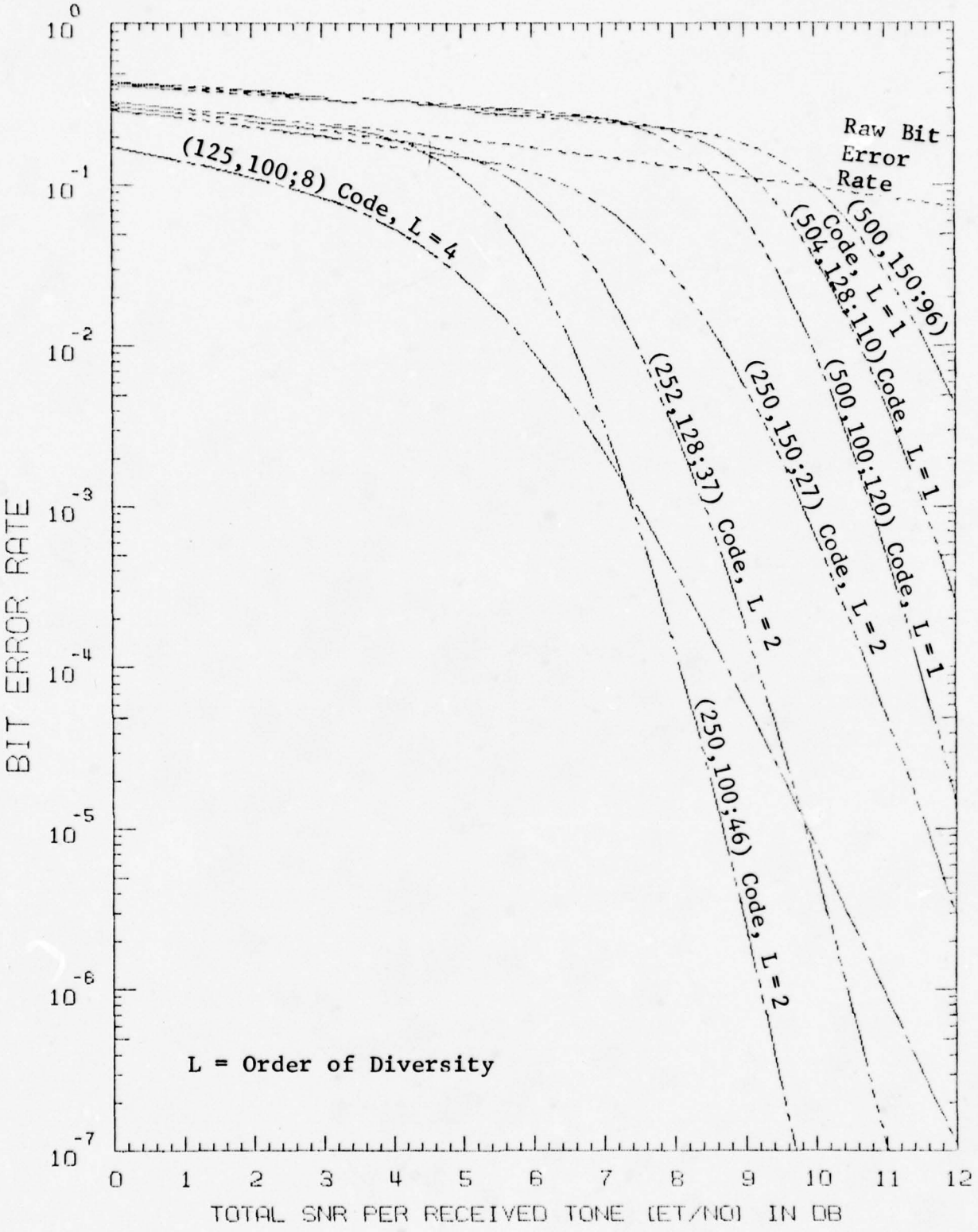


Figure 3.11 Performance of 250- or 252-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 3 Hz, Hard Decoding)

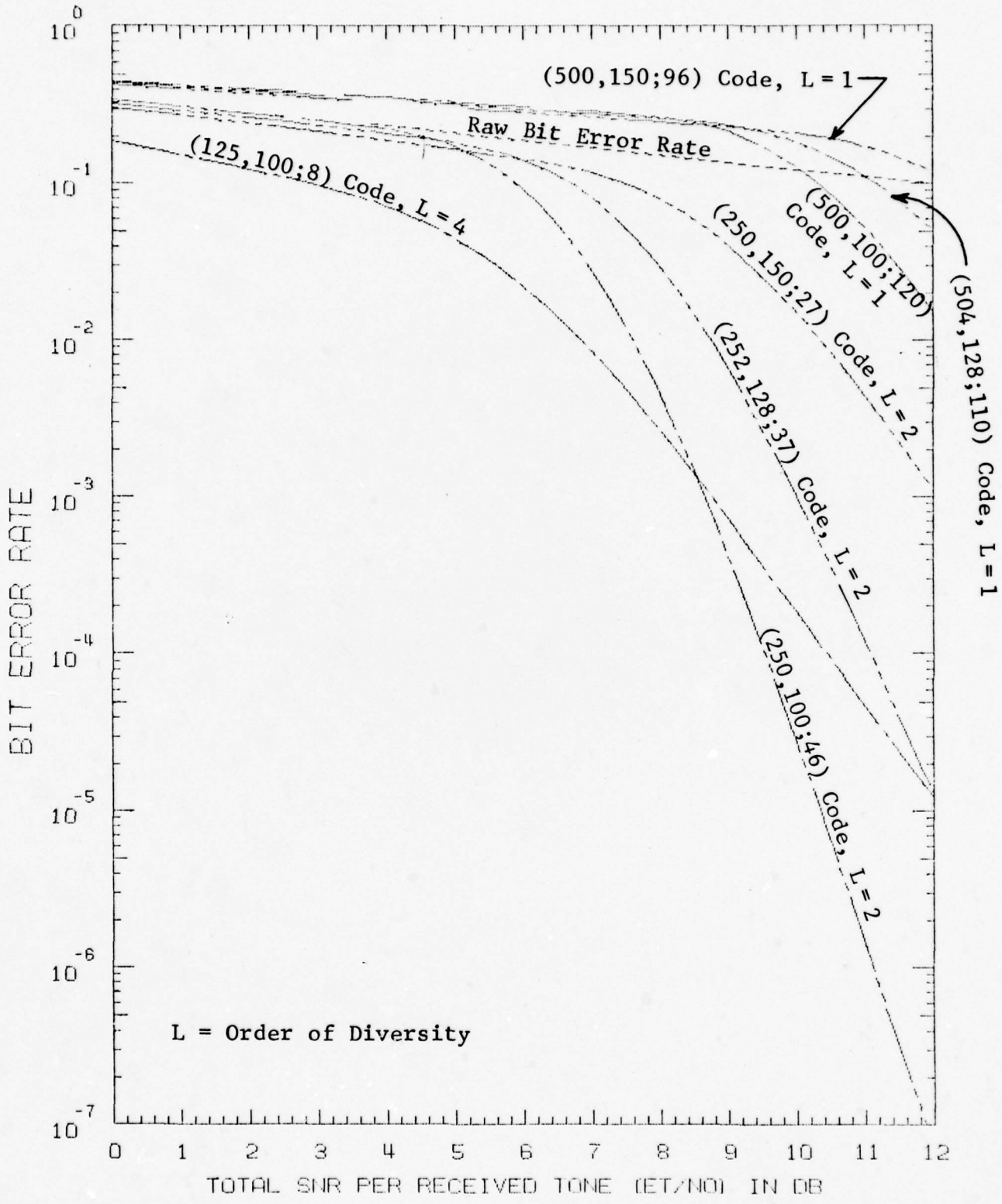


Figure 3.12 Performance of 250- or 252-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Hard Decoding)

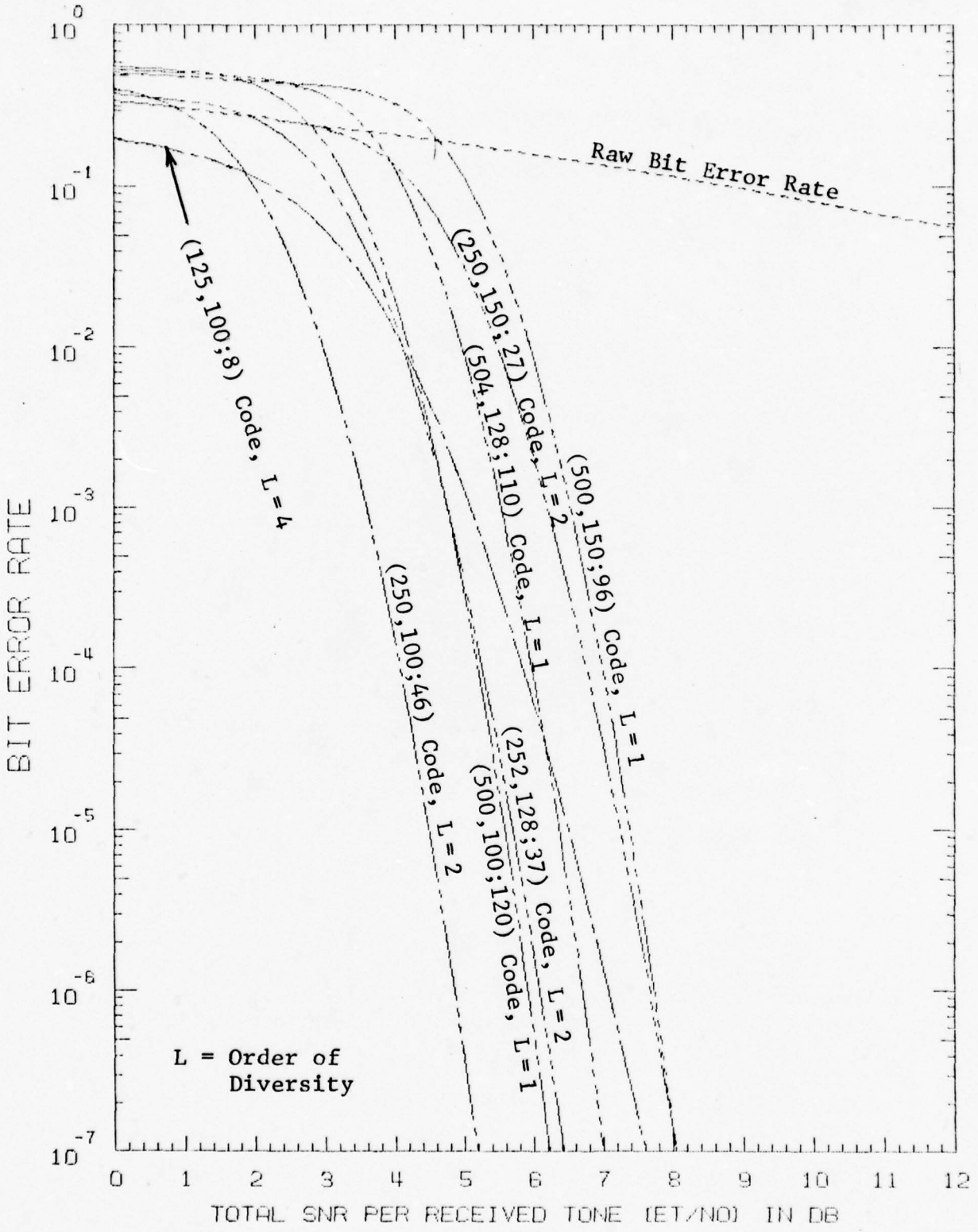


Figure 3.13 Estimated Performance of 250- or 252-Tone KG Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (Soft Decoding)

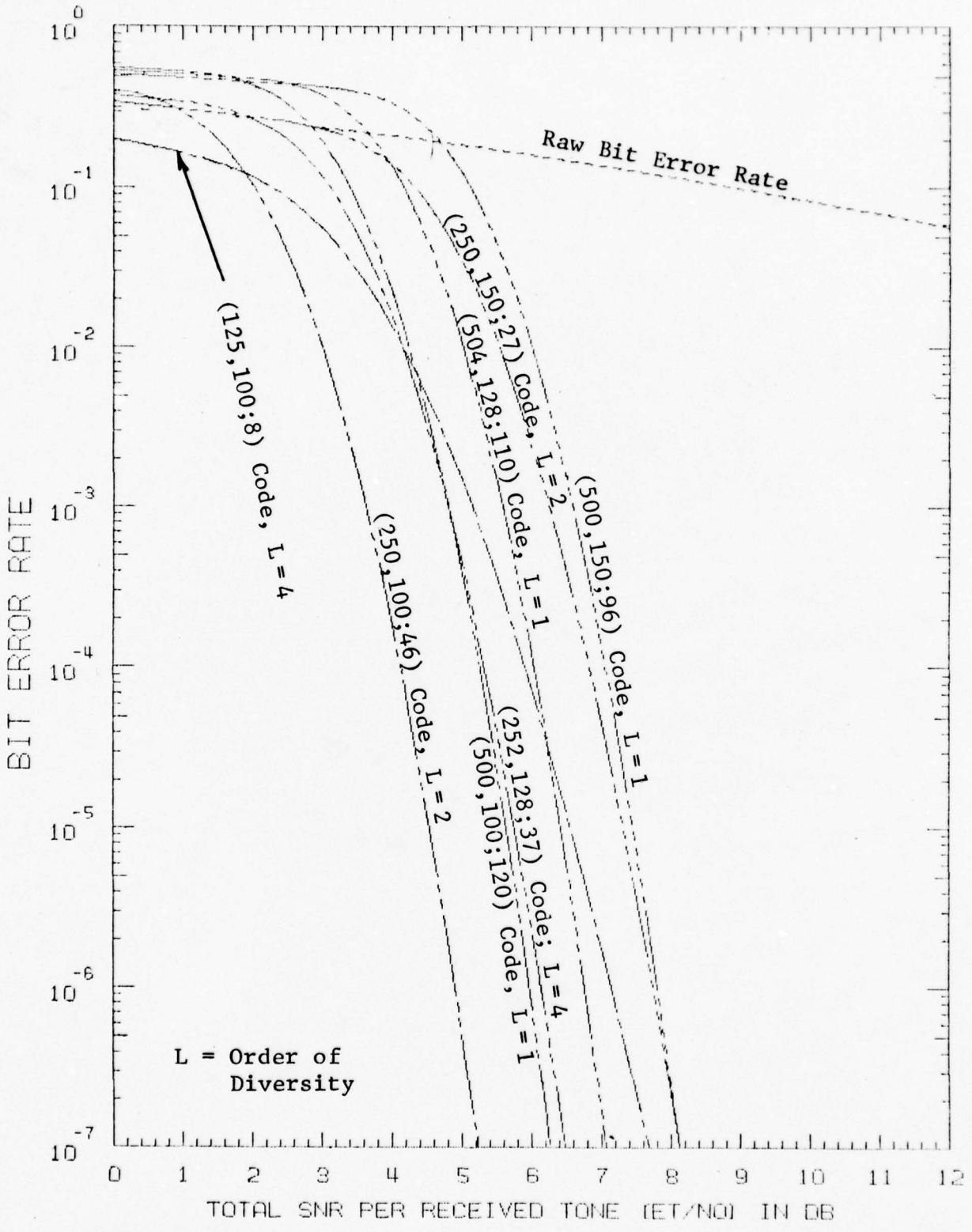


Figure 3.14 Estimated Performance of 250- or 252-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 1 Hz, Soft Decoding)

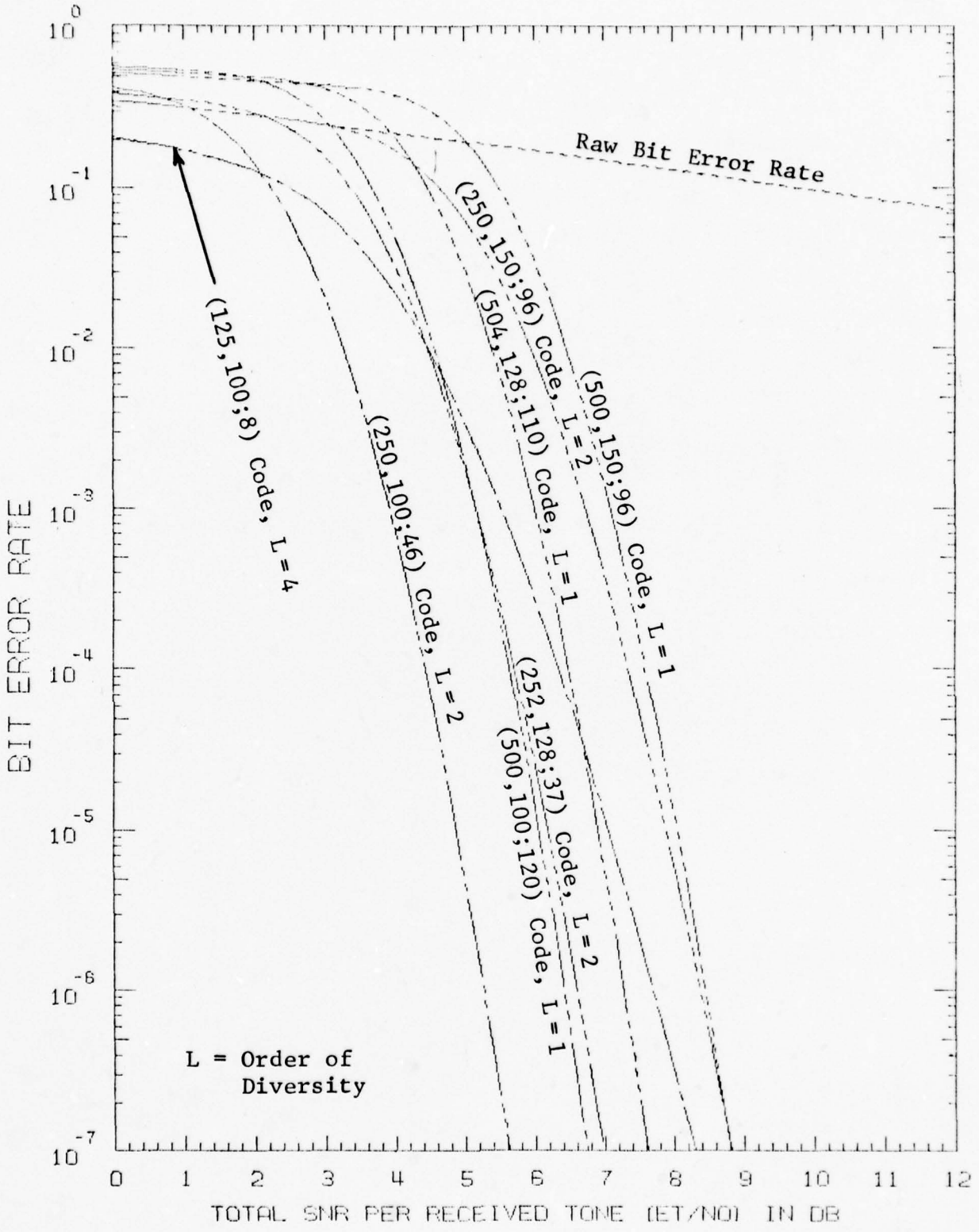


Figure 3.15 Estimated Performance of 250- or 252-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 3 Hz, Soft Decoding)

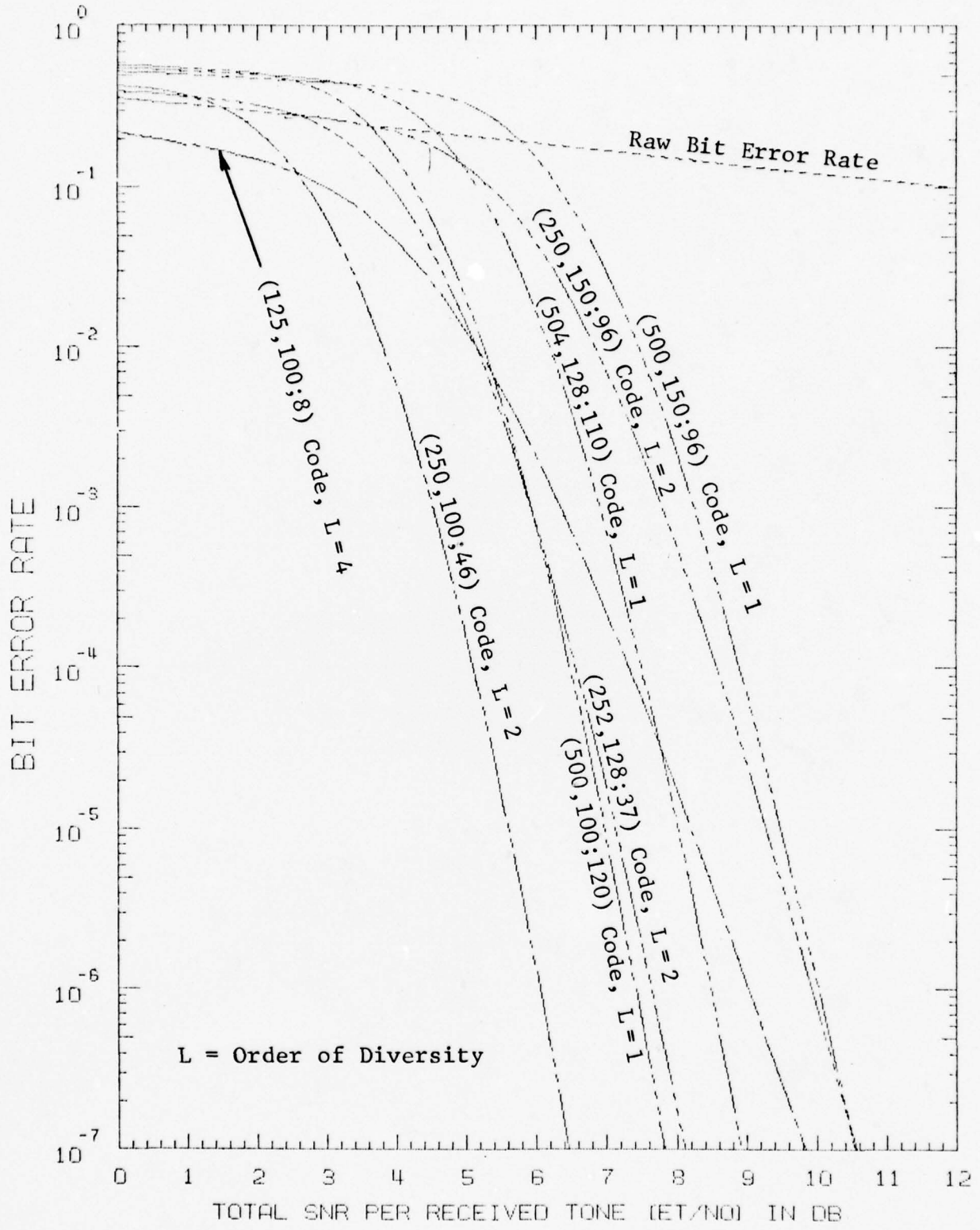


Figure 3.16 Estimated Performance of 250- or 252-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Soft Decoding)

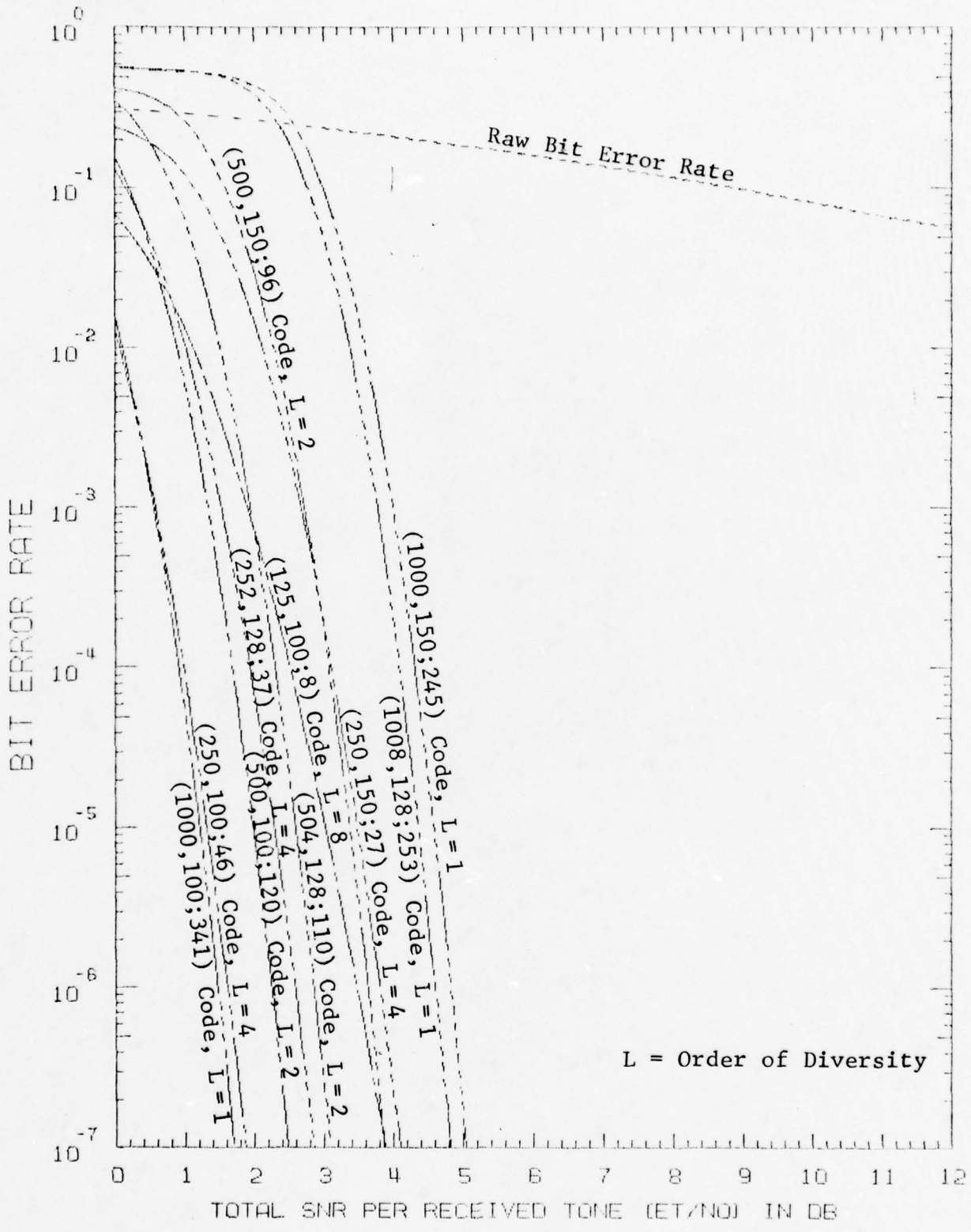


Figure 3.17 Estimated Performance of 500- or 504-Tone KG Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (Soft Decoding)

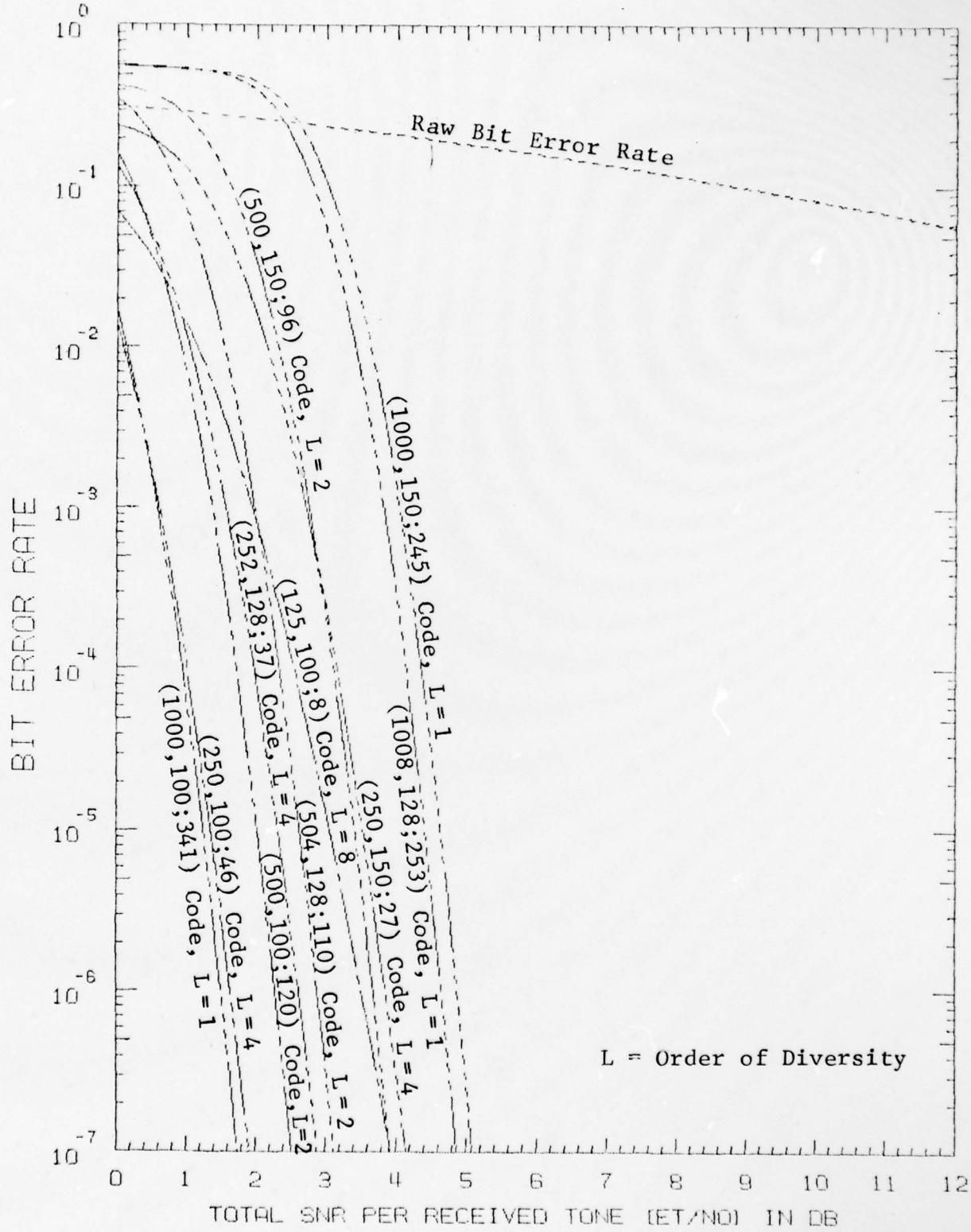


Figure 3.18 Estimated Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 1 Hz, Soft Decoding)

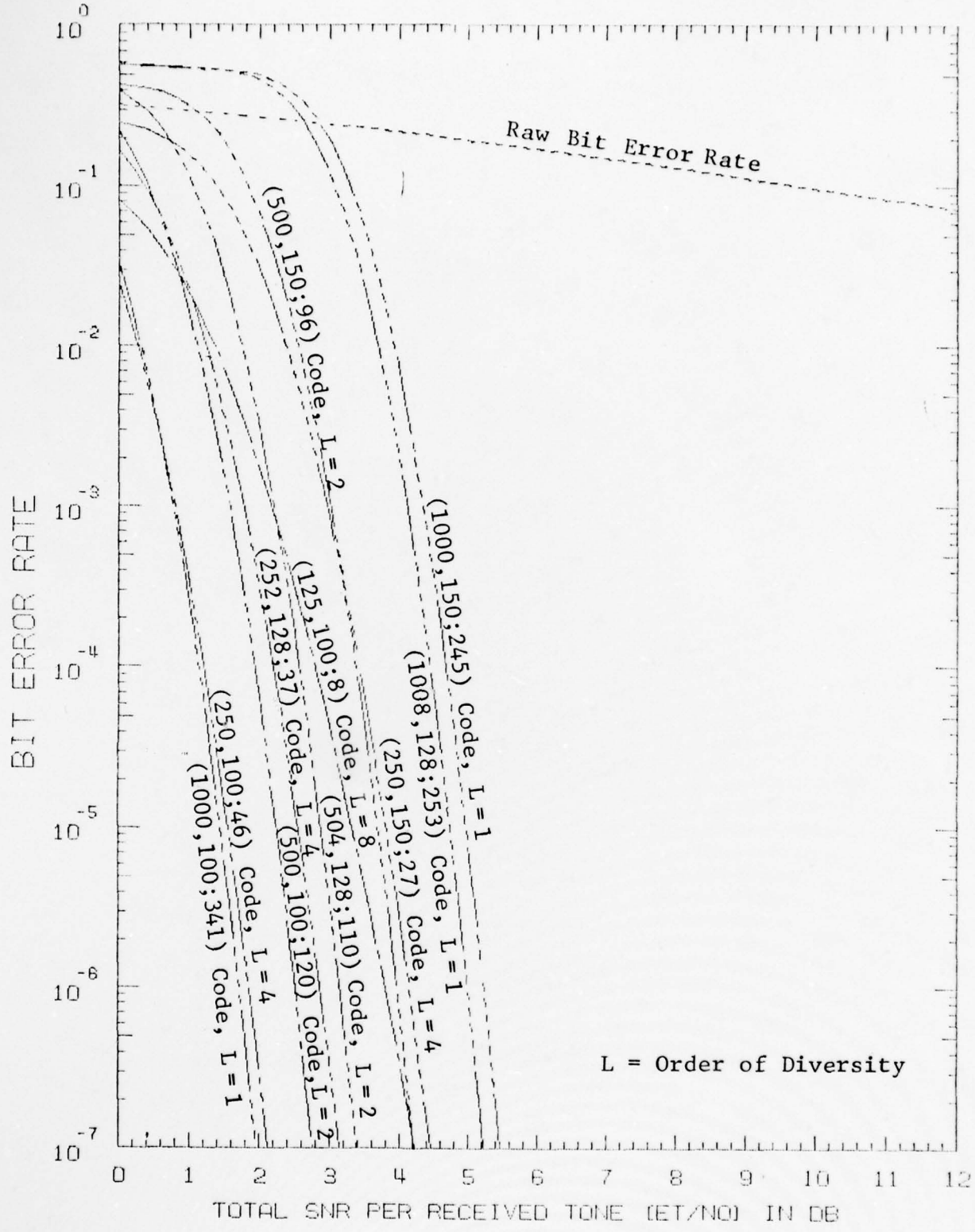


Figure 3.19 Estimated Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 3 Hz, Soft Decoding)

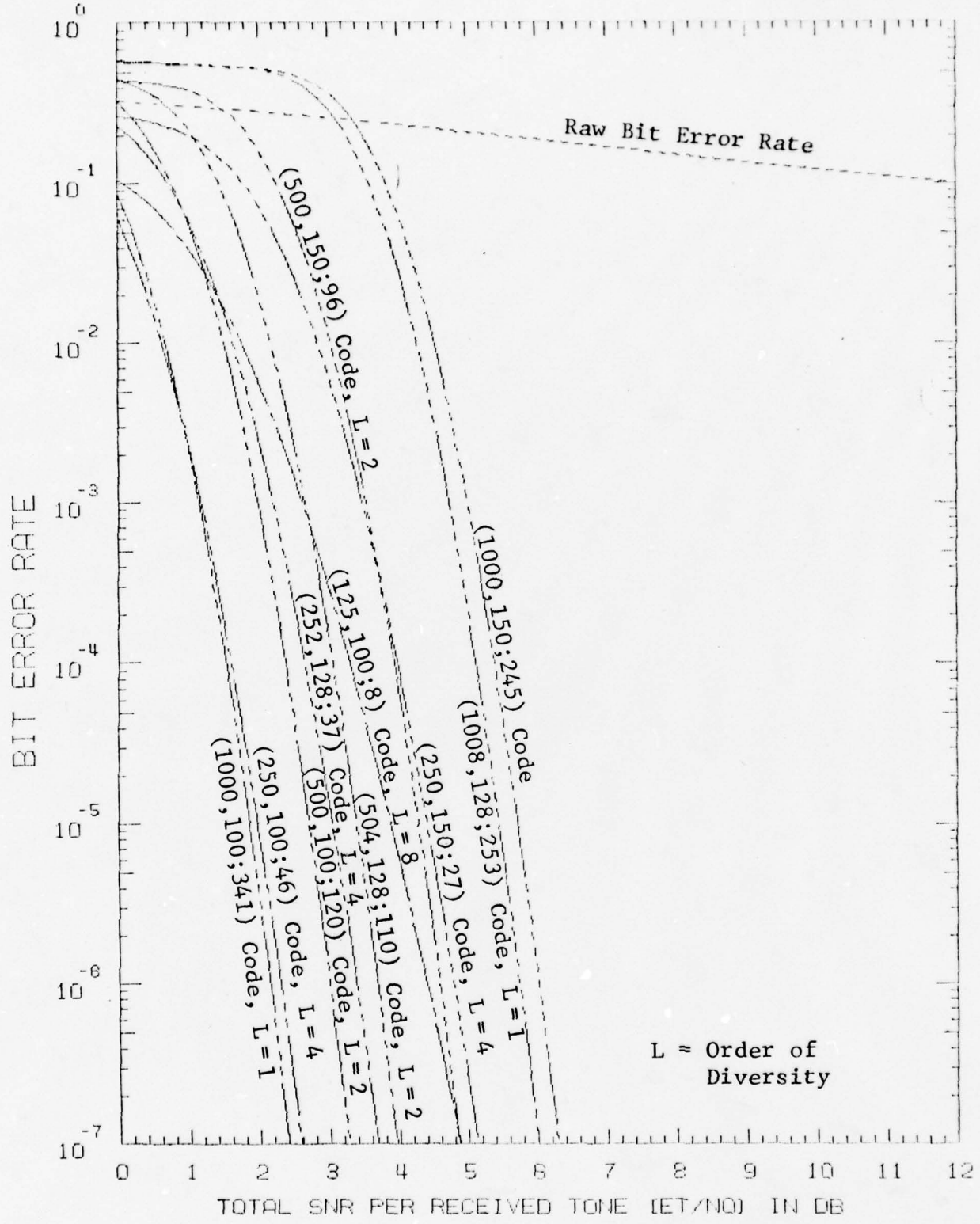


Figure 3.20 Estimated Performance of 500- or 504-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Soft Decoding)

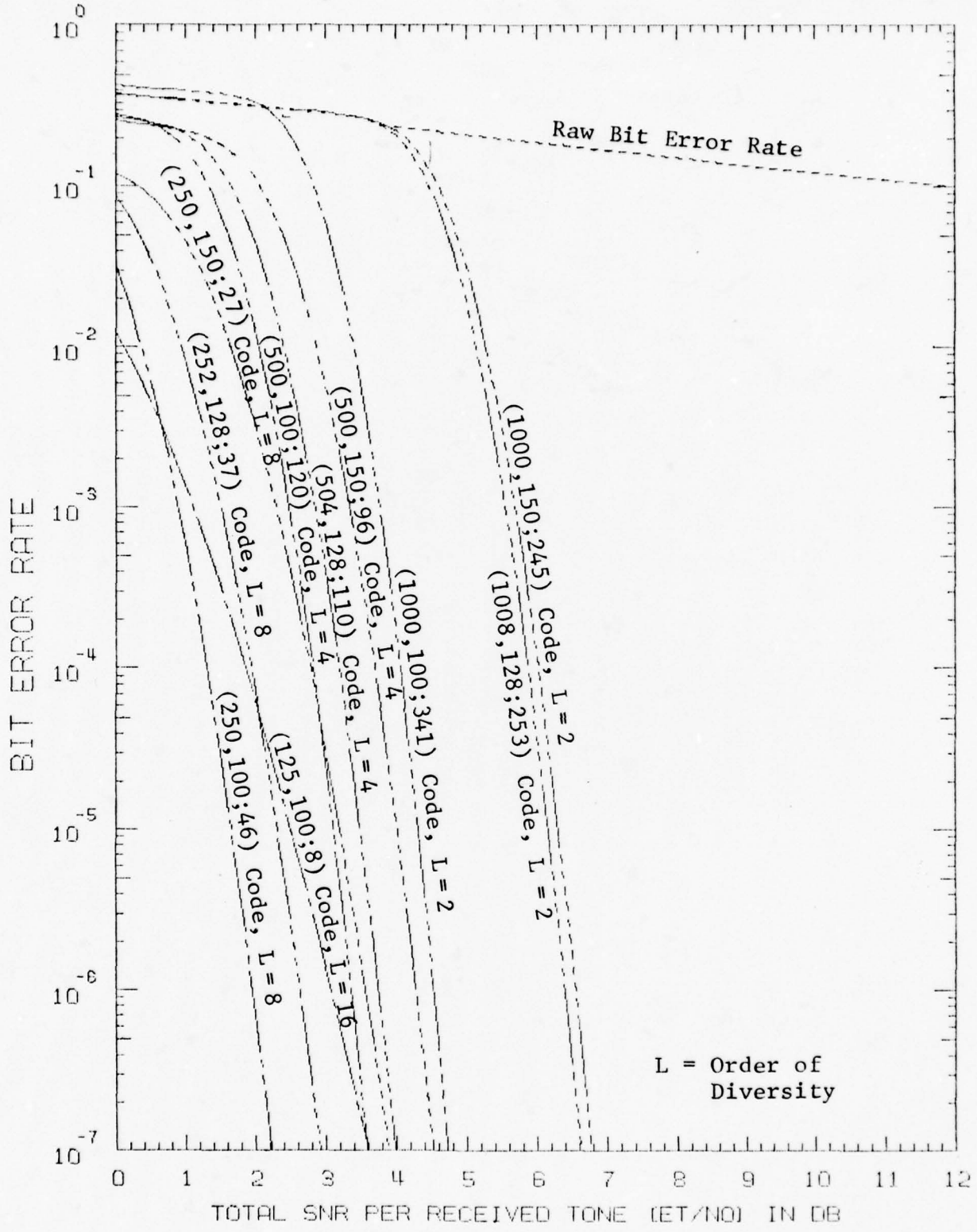


Figure 3.21a Performance of 1000- or 1008-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Hard Decoding)

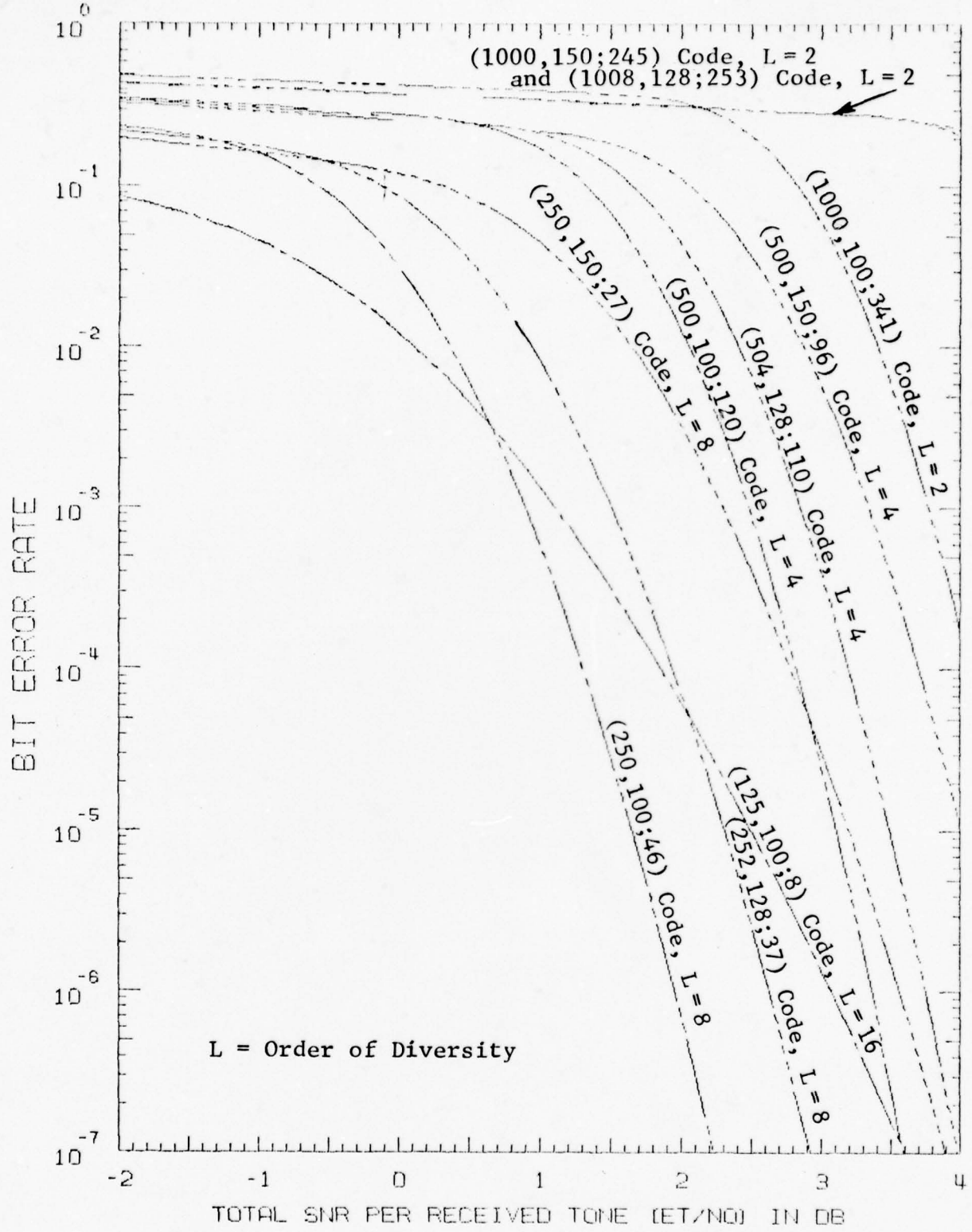


Figure 3.21b Performance of 1000-or 1008-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Hard Decoding)

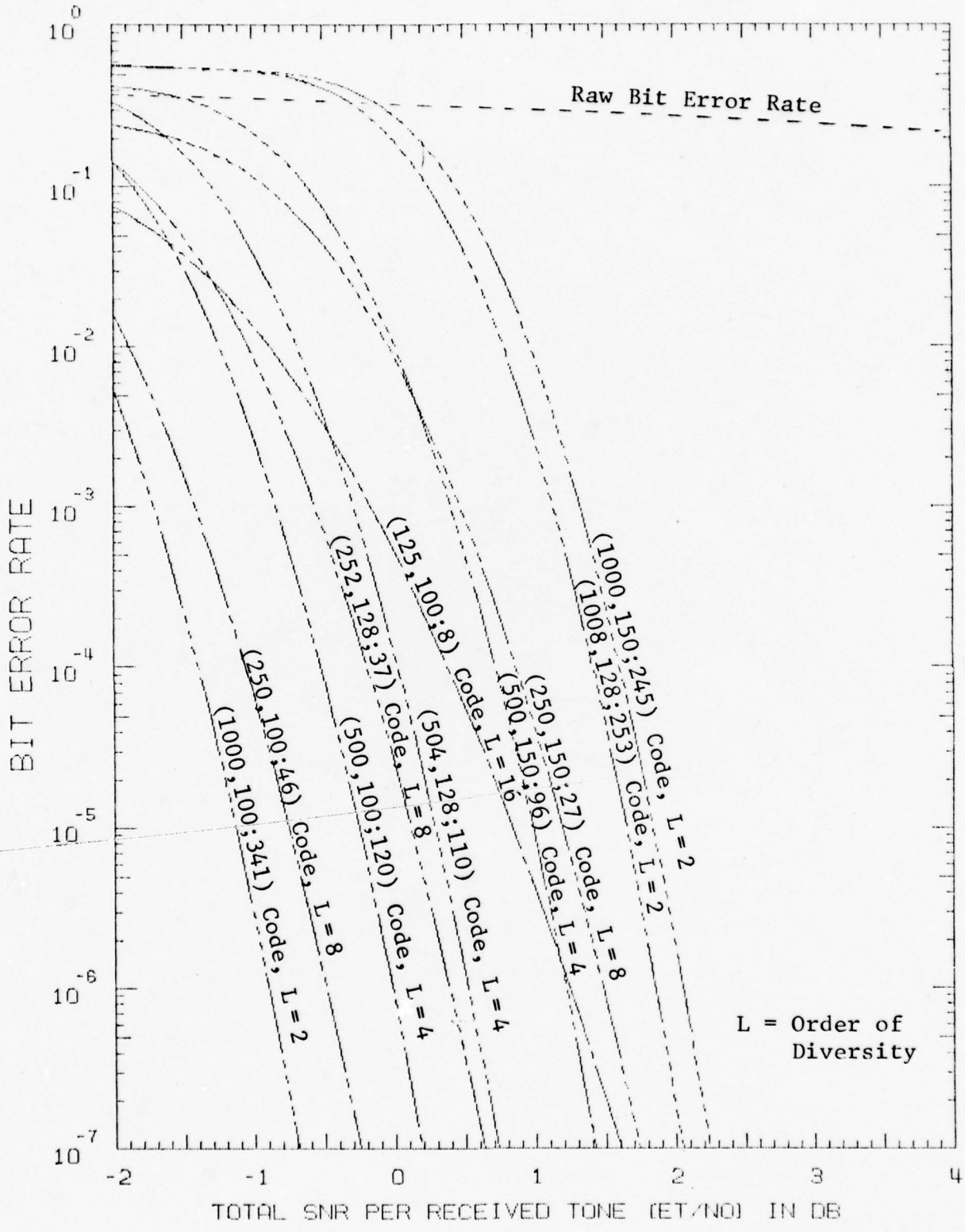


Figure 3.22 Estimated Performance of 1000- or 1008 Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Soft Decoding)

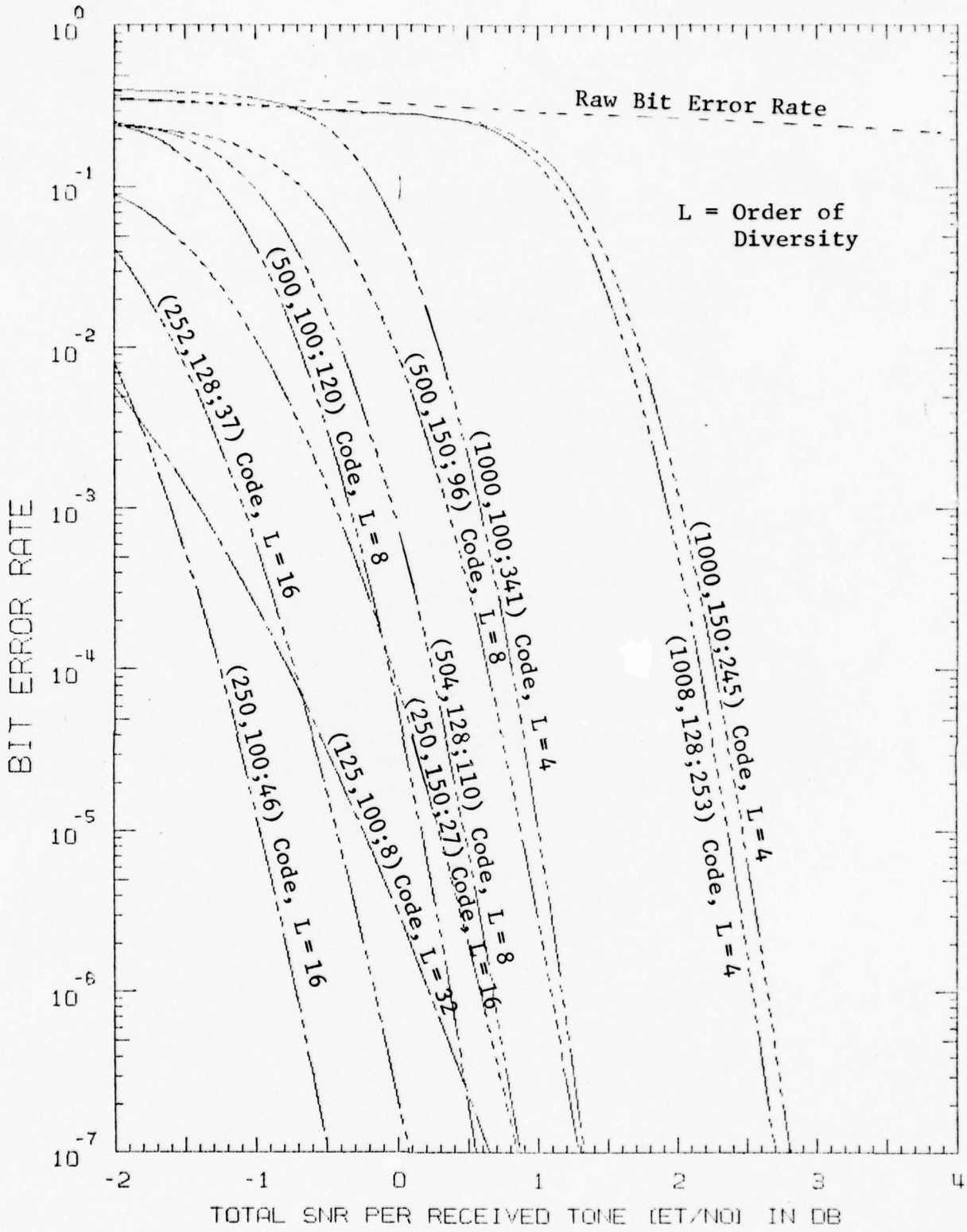


Figure 3.23 Performance of 2000- or 2016-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Hard Decoding)

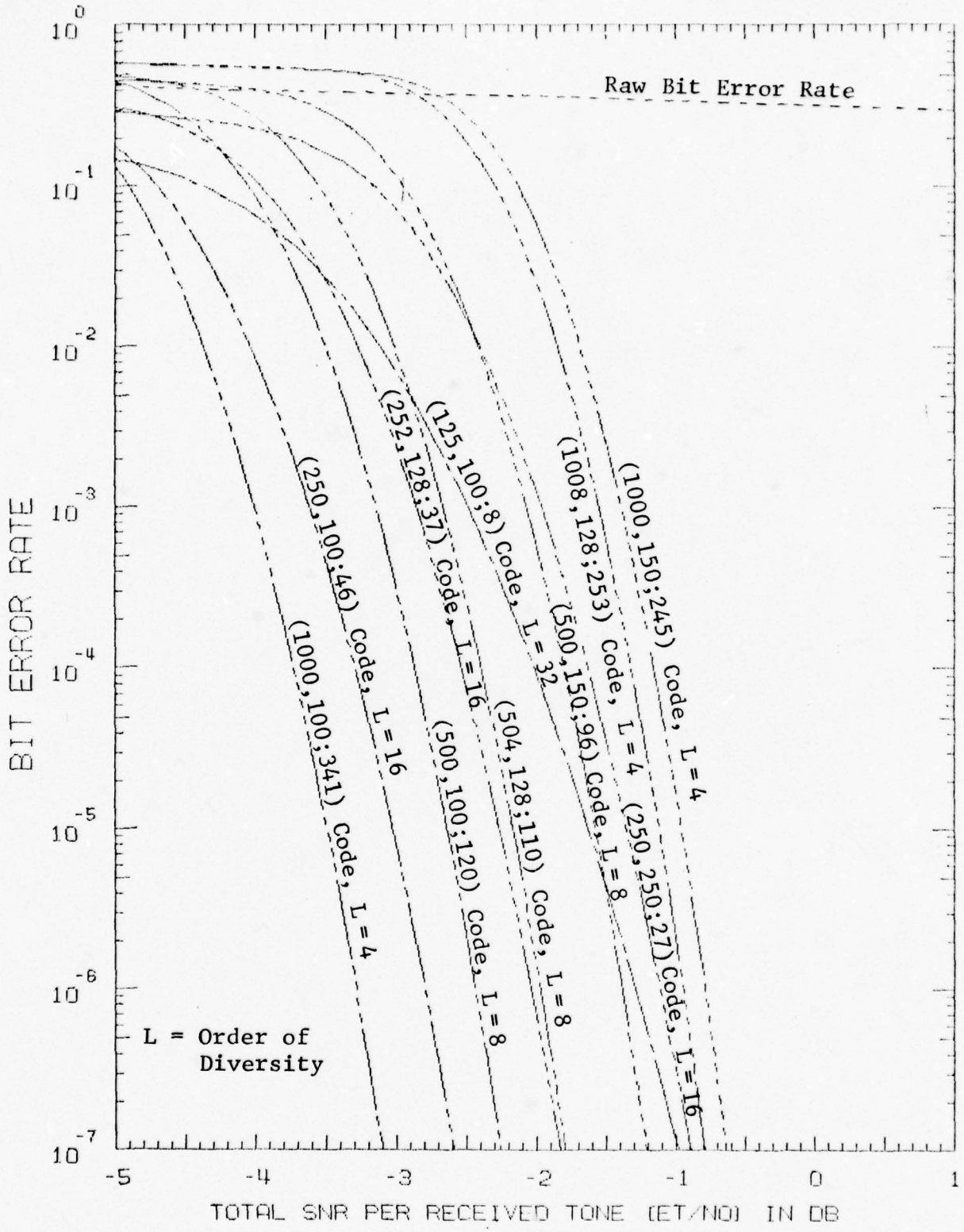


Figure 3.24 Estimated Performance of 2000- or 2016 Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Soft Decoding)

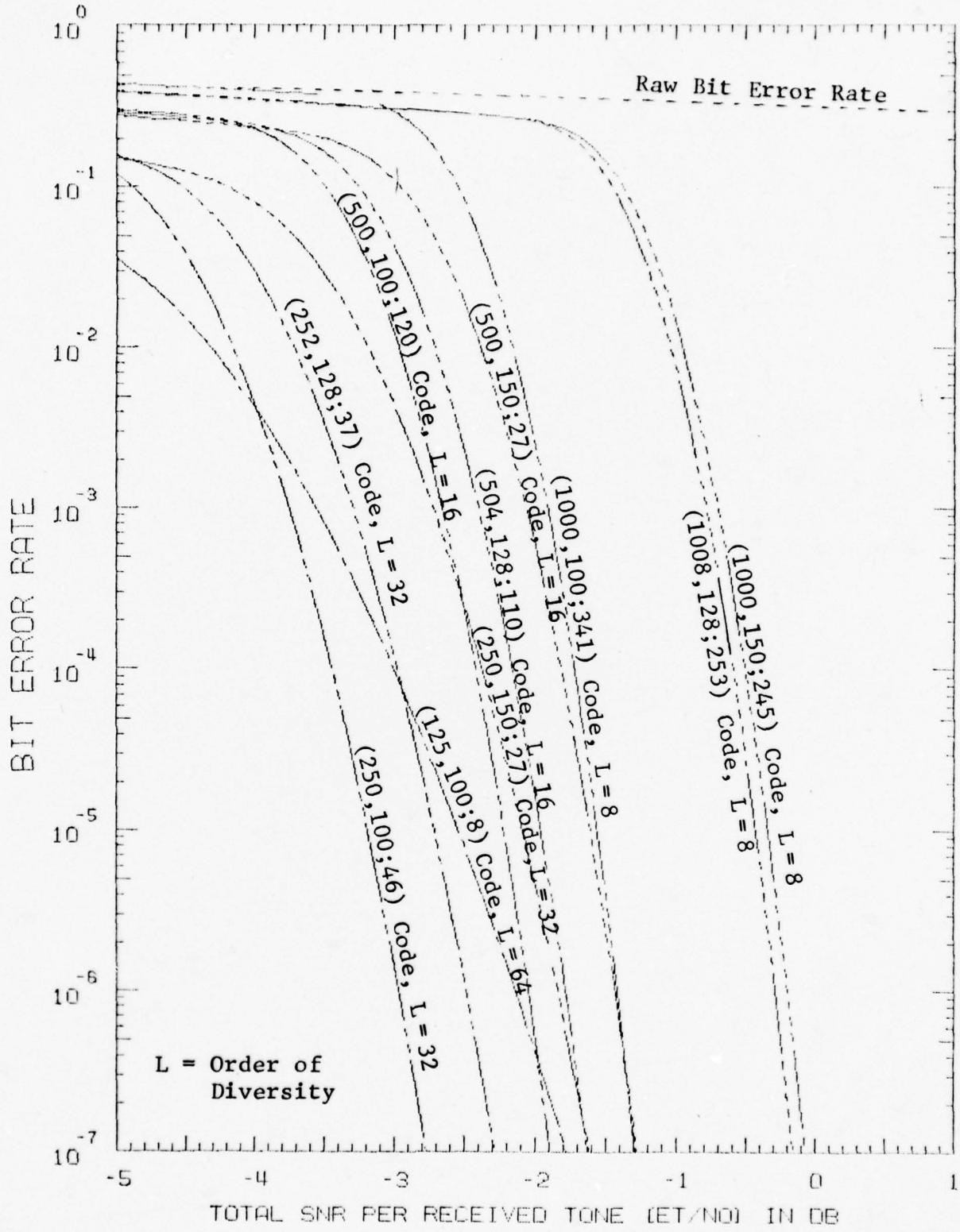


Figure 3.25 Performance of 4000- or 4032-Tone KG Codes over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Hard Decoding)

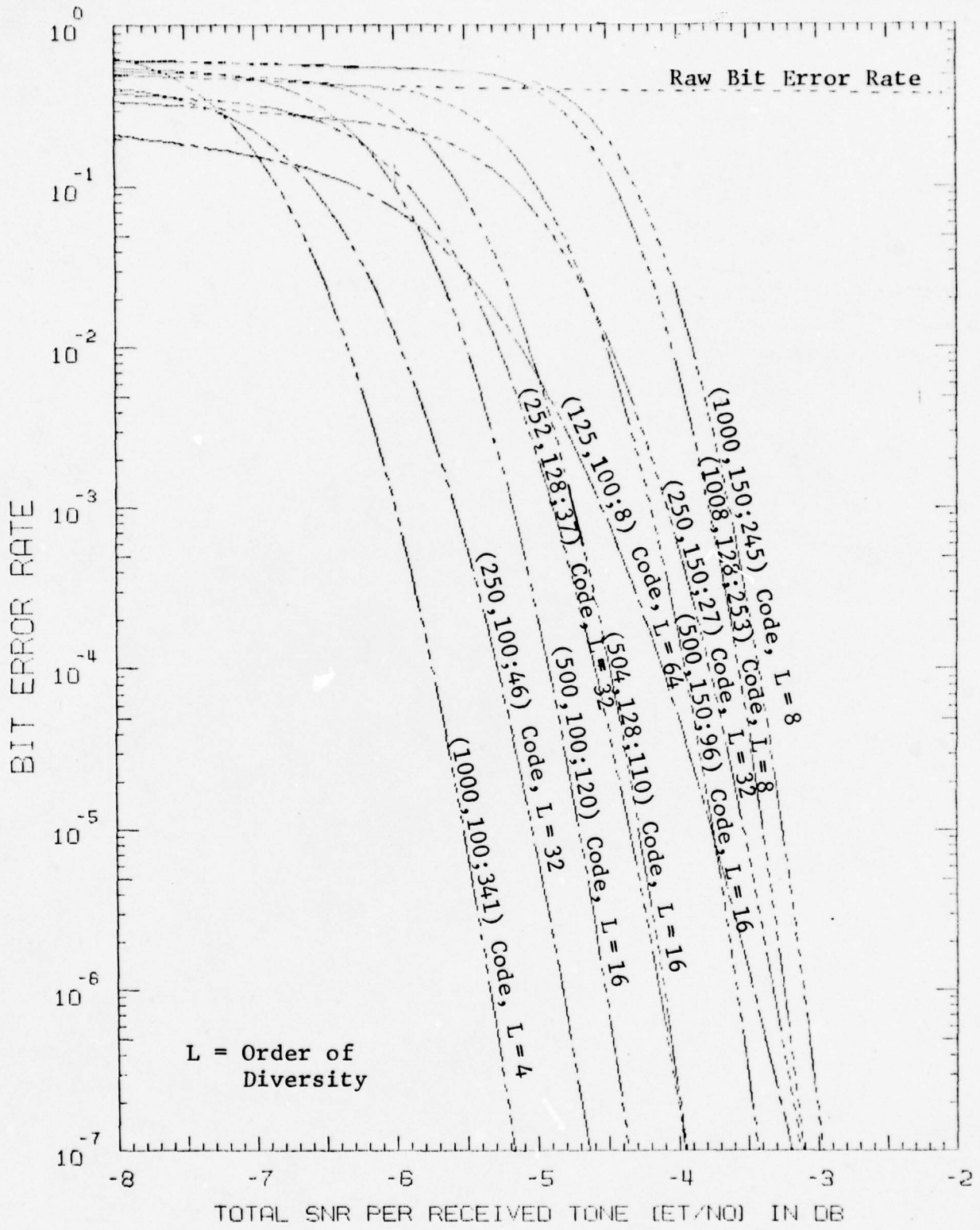


Figure 3.26 Estimated Performance of 4000- or 4032-Tone KG Coding over a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, B = 5 Hz, Soft Decoding)

TABLE 3-5

REQUIRED SIGNAL-TO-NOISE RATIO PER TONE (E_t/N_0) IN dB TO REACH A BIT ERROR RATE OF 10^{-7} FOR A TIME-VARYING HF CHANNEL WITH 4-PHASE DPSK MODULATION (FRAME RATE = 44.44 FRAMES/SEC, RMS DOPPLER BANDWIDTH = 5 Hz)

Number of Tones Per Code Block	(250,100;46) Code		(252,128;37) Code		(250,150;27) Code	
	Hard Decoding	Soft Decoding	Hard Decoding	Soft Decoding	Hard Decoding	Soft Decoding
250 or 252	11.9	6.5	*	8.1	*	10.5
500 or 504	5.8	2.6	6.7	3.7	8.2	5.1
1000 or 1008	2.2	-0.2	2.9	0.6	3.9	1.7
2000 or 2016	-0.5	-2.6	0.1	-1.8	0.8	-1.1
4000 or 4032	-2.8	-4.6	-2.3	-4.0	-1.6	-3.1

*Cannot reach a bit error rate of 10^{-7} due to irreducible errors.

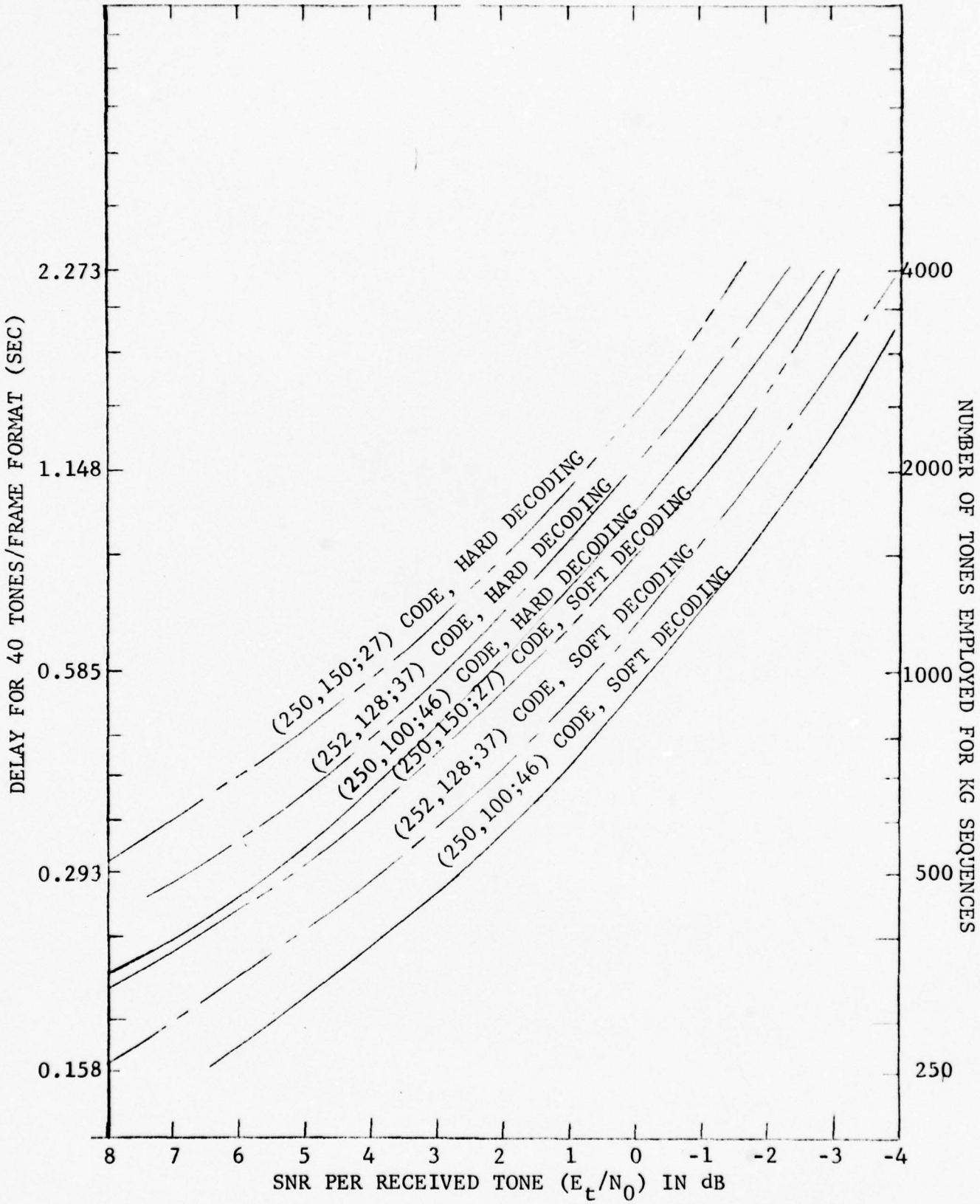


Figure 3.27 Required Signal-to-Noise Ratio for the KG Sequence to Reach a Bit Error Rate of 10^{-7} for a Time-Varying HF Channel with 4-Phase DPSK Modulation (Frame Rate = 44.44 frames/sec, Bandwidth = 5 Hz)

the delay required is far less than 1.53 seconds, the maximum allowable delay, if the number of tones is limited to 2000. Thus, there is sufficient time to implement one of the more sophisticated soft decoding algorithms to obtain the best result possible for the KG sequences.

SECTION 4

MODEM SIMULATION

Conventional parallel tone modems operating on the HF channel generally suffer significant performance degradation under conditions in which large multipath spread can be expected. These modems do not take advantage of in-band signal diversity during instances of large multipath. As a result, the information on a tone which is selectively eliminated during a fade cannot be recovered from the information in other tones which have not been so catastrophically affected.

Modems which employ coding* attempt to explicitly use in-band diversity by adding redundant information to the transmitted tone array. In order to demonstrate the usefulness of coding on the HF channel, three 2400-bit per second modems will be defined to form the basis of a comparative study:

(1) A 16 parallel tone, 4-phase DPSK modem will be used when transmissions not involving coding are to be tested. Each tone is separated by 110 Hz from adjacent tones. The modem has a frame duration of 13.33 ms.

(2) A 39 parallel tone, 4-phase DPSK modem will be used for simulations involving the Golay code (referred to as ANDVT-G). Each tone is separated from adjacent tones by 55.56 Hz. The modem has a frame duration of 22.5 ms.

*Information concerning code selection is given in Appendix B.

(3) A 41 parallel tone, 4-phase DPSK modem will be used for simulations involving a multiple rate coding structure (referred to as ANDVT-MRC). Each tone is separated from adjacent tones by 35.56 Hz. The modem has a frame duration of 22.5 ms.

Each of these modems will be evaluated using two HF channel models. (A detailed description of the HF channel model can be found in Appendix A.)

Model 1 - The HF channel is represented by two equi-strength paths separated by 1 ms. Each path is assumed to have a double-sided Doppler bandwidth of 1.0 Hz. This channel is representative of channel behavior when operating near the maximum usable frequency (MUF).

Model 2 - The HF channel is represented by a 6-path equi-strength split ray model with a total multipath spread of 2.6 ms. Each path is assumed to be resolvable into two components separated by 0.1 ms. The double-sided Doppler bandwidth per path is 1.5 Hz. This channel is representative of channel behavior when operating significantly below the maximum usable frequency (MUF).

A functional block diagram of the HF channel simulation appears in Figure 4.1. After the input data sequence has been encoded, the resulting vector is modulated, transmitted over a synthetic channel, and demodulated. The exclusive-OR ($\oplus$) of the

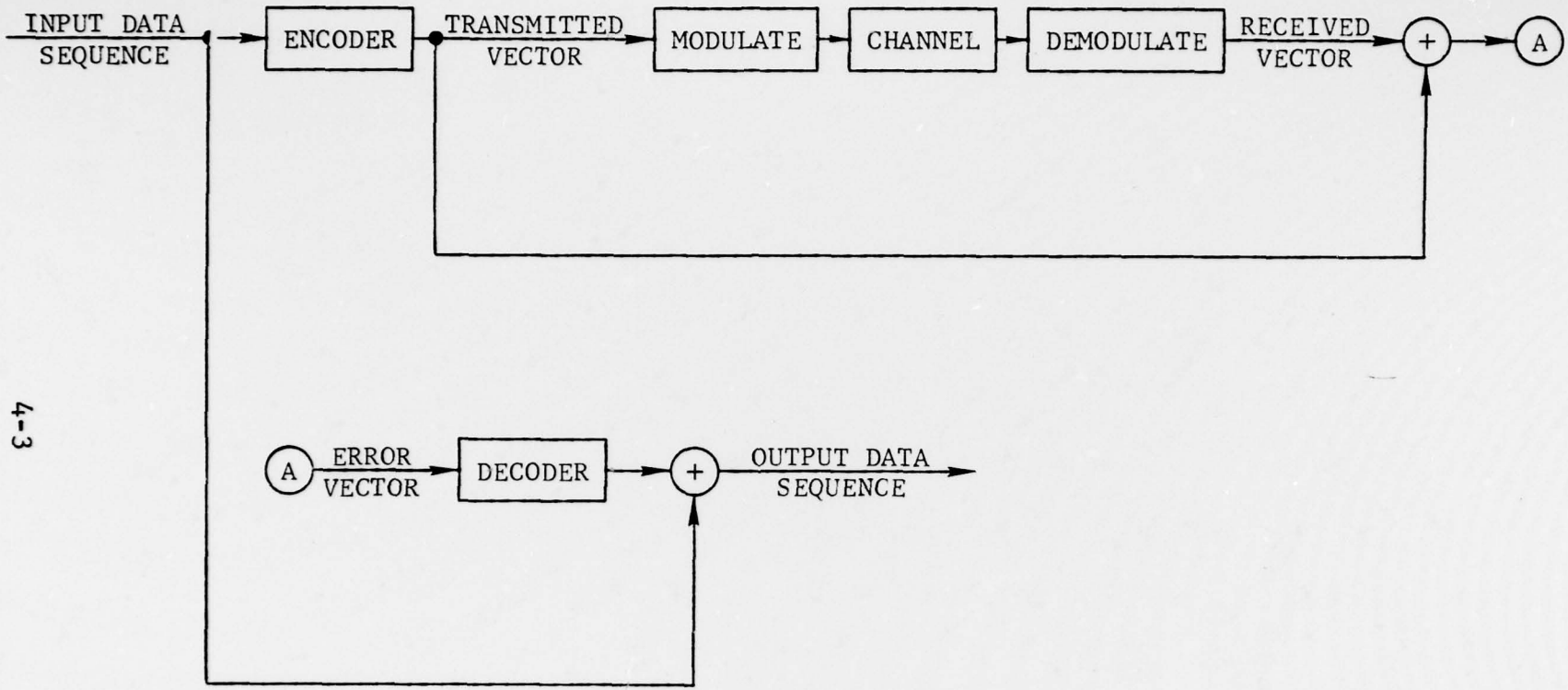


Figure 4.1 Simulation Block Diagram

transmitted vector and received vector yields a bit-by-bit indication of the errors incurred during data transmission. An exclusive-OR operation, using the decoded error vector and the input data sequence, yields the output data sequence.

The simulation program generically diagrammed in Figure 4.1 is capable of emulating several coding schemes (e.g., no coding, Golay code, BCH code, multiple rate code) as well as most parallel tone modems of practical interest. Throughout the development of the simulation, considerable emphasis has been placed on maintaining flexibility in order to insure the usefulness of these programs as a research tool.

The data to be transmitted over the HF channel consists of LPC-10 quantized parameters. The parameter content of each frame analyzed by the LPC-10 processor appears in Figure 4.2. There are 54 information bits for each analyzed frame. For the LPC-10 analyzer, a frame consists of 22.5 ms of data. Since the 16-tone modem uses 4-DPSK modulation, only 32 bits of information are conveyed by a given set of tones. As a result, a frame of LPC-10 data is not synchronous with the 16 parallel tone modem. Each of the other modems (ANDVT-G and ANDVT-MRC) uses a tone complement which insures frame synchronism for the particular coding structure employed.

4.1 ANDVT Modem with Golay Code Bit Protection

A 39 parallel tone modem using 4 DPSK can convey up to 78 bits per frame. Since 54 information bits comprise the LPC-10 quantized parameters, there are 24 residual bits which can be used for the parity bits of the Golay codes employed. A Golay code with overall parity has dimensions (24,12;8). By using two Golay codes to

LPC-10 QUANTIZED PARAMETERS	PROTECTED BITS	
	CASE 1 (RECOMMENDED FOR GOLAY CODE)	CASE 2
PITCH - 6	5	5
V/UV - 1	1	1
RMS - 5	4	3
R(1) - 5	4	3
R(2) - 5	4	3
R(3) - 4	3	2
R(4) - 4	3	2
R(5) - 4	0	1
R(6) - 4	0	1
R(7) - 4	0	1
R(8) - 4	0	1
R(9) - 4	0	1
R(10) - 3	0	0
SYNC - 1	0	0
54	24	24

LEGEND: V/UV → VOICED/UNVOICED PARAMETER
R(i) → i^{th} REFLECTION COEFFICIENT

Figure 4.2 Golay Code Protection Pattern for LPC-10 Quantized Parameters

protect 24 bits of the LPC-10 quantized parameters, the full complement of available bits (54 + 2 x 12) in the ANDVT-G modem is utilized.

For the purposes of simulation, the LPC-10 quantized parameters have been restructured. An attempt has been made to order the bits comprising these parameters according to their relative importance in synthesized speech quality. The resulting ordering is portrayed in Figure 4.3. The two Golay codes employed by the modem are given the task of protecting the first 24 bits of the ordered bit sequence. A summary of the parameters to which the protected bits belong appears as Case 1 in Figure 4.2. Case 2 was attempted, however, speech tapes indicated that Case 1 offers superior performance.

Once the bits to be protected have been encoded, all the bits must be placed on the 39-tone array. In order to benefit from any available in-band diversity, care is exercised in maintaining a large tone separation between bits representing the same codeword. Of particular importance is the assurance that two bits from one codeword do not occupy the same tone (since both bits may be lost during a fade).

The mapping between the 78 transmitted bits and the 39 tone array is detailed in Figure 4.4. For convenience, this mapping will be referred to as the modulation format bit assignments. The first 12 bits of the ordered bit sequence (Figure 4.3) are protected by the Golay code in FIELD 1 of Figure 4.4. The second set of 12 bits in the ordered bit sequence is protected by the Golay code in FIELD 2 of Figure 4.4. The remaining 30 bits in the ordered sequence are unencoded. Collectively, they comprise FIELD 3 in Figure 4.4.

P → PITCH

A → AMPLITUDE (RMS)

V → VOICE/UNVOICE

S → SYNCHRONIZATION

$R^{(i)}$ → i'th REFLECTION COEFFICIENT

α_j → j'th MOST SIGNIFICANT BIT OF THE PARAMETER ALPHA

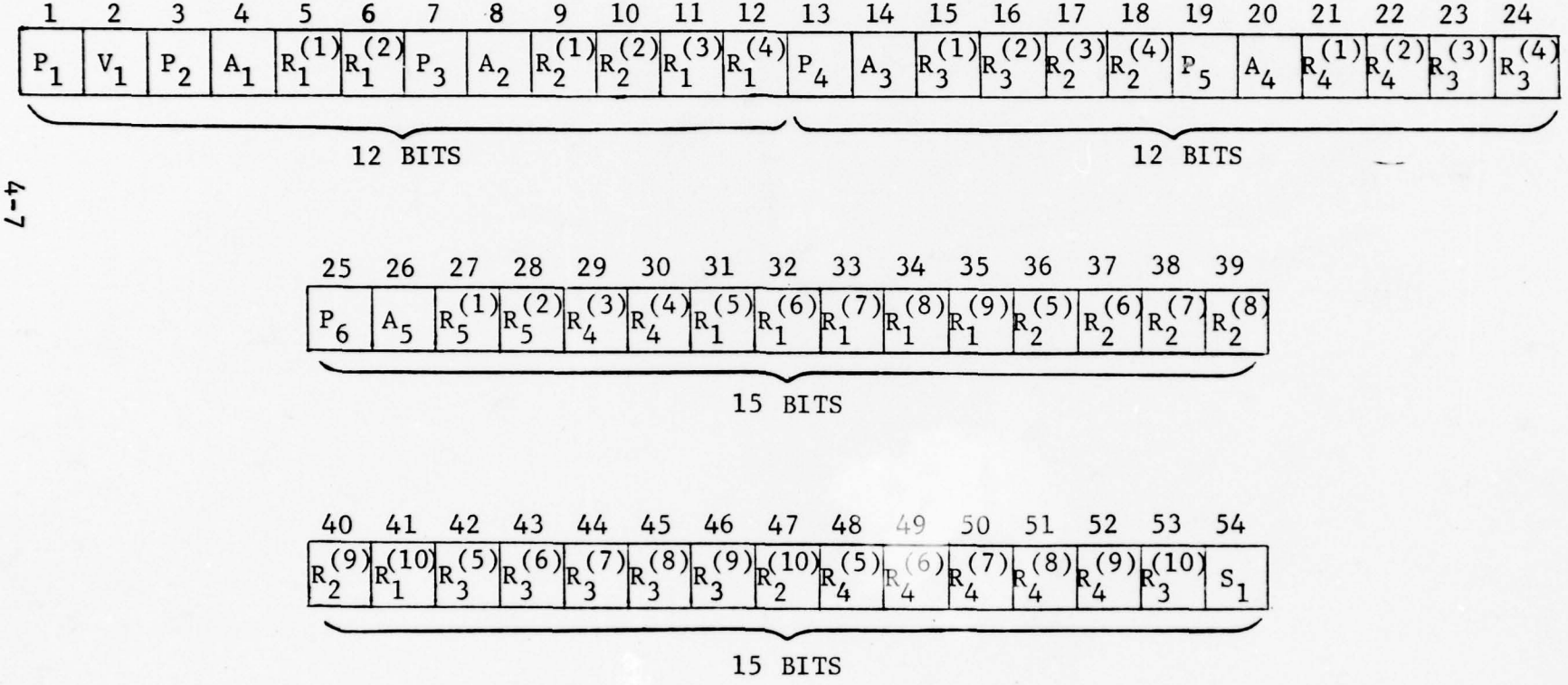
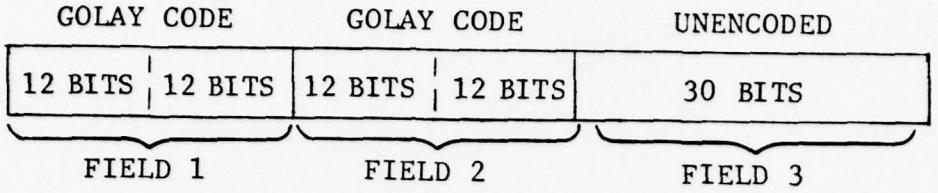


Figure 4.3 Ordered Sequence of Bits Comprising the LPC-10 Quantized Parameters



$F_j^i \Longrightarrow$ j'th BIT OF THE i'th FIELD

BIT 2	F_{14}^2	F_{14}^3	F_{15}^3	F_{15}^2	F_{16}^3	F_{17}^3	F_{16}^2	F_{18}^3	F_{24}^1	F_{17}^2	F_{19}^3	F_{23}^1	F_{18}^2	F_{20}^3	F_{22}^1	F_{19}^2	F_{21}^3	F_{21}^1	F_{20}^2	F_{22}^3
BIT 1	F_1^1	F_1^2	F_1^3	F_2^1	F_2^2	F_2^3	F_3^1	F_3^2	F_3^3	F_4^1	F_4^2	F_4^3	F_5^1	F_5^2	F_5^3	F_6^1	F_6^2	F_6^3	F_7^1	F_7^2
TONE NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

BIT 2	F_{20}^1	F_{21}^2	F_{23}^3	F_{19}^1	F_{22}^2	F_{24}^3	F_{18}^1	F_{23}^2	F_{25}^3	F_{17}^1	F_{24}^2	F_{26}^3	F_{16}^1	F_{27}^2	F_{28}^3	F_{15}^1	F_{29}^2	F_{30}^3	F_{14}^1
BIT 1	F_7^3	F_8^1	F_8^2	F_8^3	F_9^1	F_9^2	F_9^3	F_{10}^1	F_{10}^2	F_{10}^3	F_{11}^1	F_{11}^2	F_{11}^3	F_{12}^1	F_{12}^2	F_{12}^3	F_{13}^1	F_{13}^2	F_{13}^3
TONE NO.	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39

Figure 4.4 Modulation Format Bit Assignment (Golay Code)

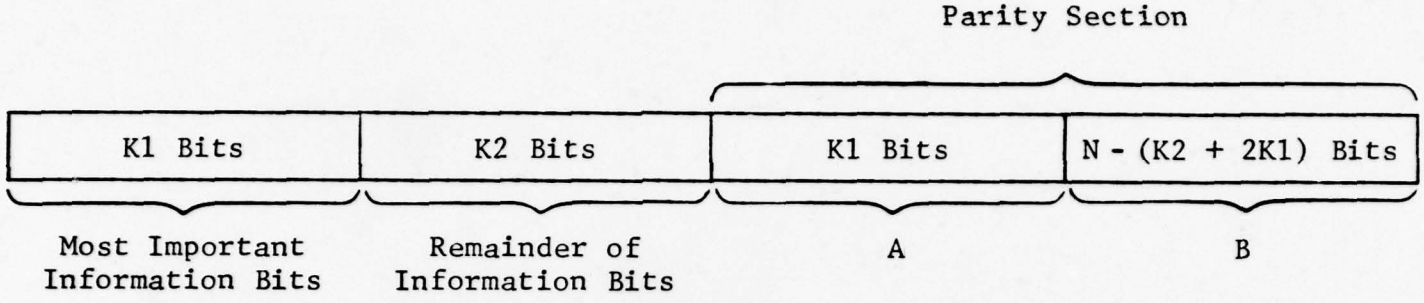
4.2 ANDVT Modem with Multiple Rate Code Bit Protection

Simulations have also been conducted for a modem (ANDVT-MRC) which uses a multiple rate coding structure. This coding approach places redundant information across time frame boundaries as well as the tone array. A detailed discussion of this time domain coding technique can be found in [4.1].

For the purposes of this simulation, it is only necessary to understand the impact of the codeword structure on the LPC-10 bit protection choices. Toward that end, a cursory description of a multiple rate coding technique seems appropriate.

In Figure 4.5, a typical multiple rate codeword is depicted. In general, each multiple rate codeword contains the modulo-2 (exclusive-OR) addition of several inner codewords. With little loss in generality, assume that each inner codeword is the result of a BCH encoding of a set of information bits. The inner codewords forming a multiple rate codeword will have been encoded at more than one rate (the ratio of the number of information bits in the codeword to the total length of the codeword). This simulation only considers code structures in which two code rates are needed.

In Figure 4.5, there is a total of K1+K2 information bits in the codeword. The information section which is to receive the greatest protection contains K1 bits. The other information section contains K2 bits; it is afforded a somewhat smaller protection level. In an ordinary single rate implementation, the information bits are encoded to form an N-bit codeword with rate (K1+K2)/N. In this case, both information sections receive the same level of protection.



A — Contains the exclusive-OR of some parity bits from the low rate encoding of this code word with the K1 information bits from a selected set of previous codewords.

B — Contains the exclusive-OR of some parity bits from the low rate encoding of this code word with the parity bits from the high rate encoding of a selected set of previous codewords.

High rate code description → $(N, 2K1 + K2, d_h)$

Low rate code description → $(N, K1 + K2, d_\ell)$

Figure 4.5 Contents of a Multiple Rate Codeword

The multiple rate code structure incorporates the K1 most important bits into a codeword with rate (2K1 + K2)/N. This higher rate encoding is accomplished by appending the K1 most important bits to K1 + K2 zeros. The resulting conglomerate becomes the 2K1 + K2 information bits of the high rate codewords. After encoding, the high rate vector is added (modulo-2). to a predefined set of future codewords. Observe that the K1 most important information bits which have been included in high rate codewords appear in the parity section of the low rate encoding of future information blocks. The number of times that the high rate code is fed forward is equivalent to the burst error-correcting capability of the multiple rate code. The description in Figure 4.5 and the structure depicted in Figure 4.6 typify the feed-forward character of this time domain coding approach.

The resulting admixture of different rate codewords can be decoded in the low rate mode in the absence of errors or in the high rate mode when a past block of K1 bits is unknown. Of course, the low rate code (N, K1 + K2) must be a subcode of the high rate code (N, 2K1 + K2). This is generally not a problem for BCH codes since they can be selected to insure that the generator polynomial for the high rate code is a multiple of the generator polynomial for the low rate code.

During the decoding process, the K1 most important information bits can potentially be recovered from any of the codewords which they influence. If all the information from previous codewords has not been removed, then the decoding must be attempted in the high rate mode. When the information from all but one previous frame has been removed from a codeword, and when a successful high rate decode can be achieved, the K1 most important

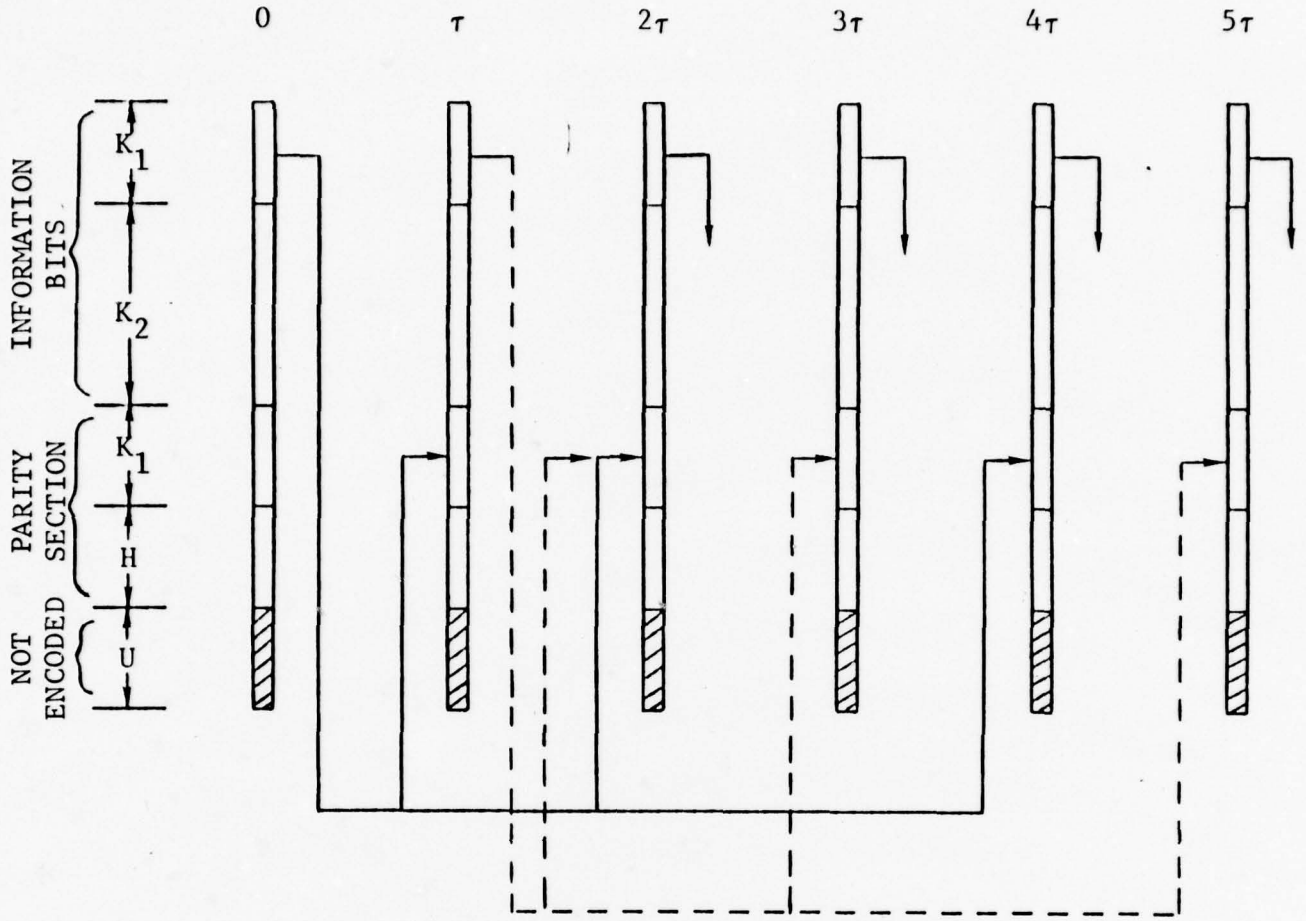


Figure 4.6 Structure of a Triple Burst Correcting Multiple Rate Code (K_1 information bits are principally protected by the high rate code and the multiple repeats; K_2 information bits are not protected by multiple repeats; H bits are the parity check bits of the high rate code; and there are U unprotected bits. τ is a delay constrained to be an integral multiple of a frame duration.)

information bits from the previous frame can be recovered. A low rate decode can be tried only when the information from all previous codewords has been removed. The K2 information bits in a previous codeword can never be recovered if they have not been correctly decoded initially. Thus, they are not as well protected as the K1 most important information bits.

As previously mentioned, each frame of the LPC-10 quantized parameters consists of 54 information bits. If an LPC-10 frame is to remain synchronous with a modem frame when a multiple rate code structure is employed, then 54 - (K1 + K2) bits must be transmitted unencoded. The presence of unencoded bits is reflected in the structure of Figure 4.6. With two bits per tone in a 4-DPSK modem, [N + 54 - (K1 + K2)]/2 tones are required. Since the ANDVT-MRC is presently conceived as a 41 parallel tone modem, the number of parity bits in the low rate code [N - (K1 + K2)] is constrained to be 28.

Initially, two multiple rate codes are being considered. Each code will be defined by a descriptor of the form $(N,k_\ell;d_\ell) \rightarrow (N,k_h;d_h)$; where k_ℓ (k_h) are the number of information bits in the low (high) rate code, and where d_ℓ (d_h) is the minimum distance of the low (high) rate code. The values of K1 and K2 for a $(43,15;12) \rightarrow (43,30;6)$ code are, respectively, 15 and 0. Similarly, the values of K1 and K2 for a $(64,36;12) \rightarrow (64,51;6)$ code are, respectively, 15 and 21. A summary of the parameters to which the protected bits belong appears in Figure 4.7.

After each multiple rate codeword has been formed, the resulting vector along with the unencoded bits must be placed in the 41 tone array. The modulation format bit assignments for the $(43,15;12) \rightarrow (43,30;6)$ multiple rate code are clearly

LPC-10 Quantized Parameters	One Level of Protection (43,15;12) → (43,30;16) MRC Most Important Bits	Two Levels of Protection (64,36;12) → (64,51;16) MRC		
		First Level	Second Level	Total
Pitch – 6	4	4	2	6
U/UV – 1	1	1	0	1
RMS – 5	3	3	2	5
R(1) – 5	3	3	2	5
R(2) – 5	2	2	3	5
R(3) – 4	1	1	3	4
R(4) – 4	1	1	3	4
R(5) – 4	0	0	2	2
R(6) – 4	0	0	1	1
R(7) – 4	0	0	1	1
R(8) – 4	0	0	1	1
R(9) – 4	0	0	1	1
R(10) – 3	0	0	0	0
Sync – 1	0	0	0	0
54	15	15	21	36

Figure 4.7 List of Protected Bits for LPC-10 Multiple Rate Code Structure

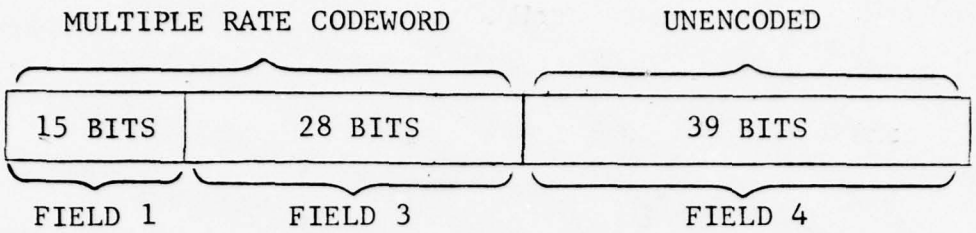
defined in Figure 4.8. Similar bit assignments for the (64,36;12) → (64,51;6) multiple rate code are contained in Figure 4.9.

Ordinarily, more than one bit from the same codeword is not placed on a single transmitted tone. This restriction is obviously not applicable, however, when the length of the codeword exceeds the number of tones available. The procedure followed in the preparation of Figures 4.8 and 4.9 involves maximizing the tone separation of the bits within each of the defined fields. When a successful decode does not occur, the procedure minimizes the variance of the number of bit errors per frame for a given coding field.

4.3 Simulation Results

For each of the two HF channel models described in the introduction to Section 4, simulations have been conducted using the following: 1) a 16 parallel tone modem without coding; 2) the ANDVT modem with the Golay code; and 3) the ANDVT modem with the (64,36;12) → (64,51;6) multiple rate code. A principal product of these simulations has been the production of a set of voice tapes which document the performance of these modems as a function of E_b/N_0 (signal-to-noise ratio per information bit). In Appendix C the relationship between E_b/N_0 and E_t/N_0 (signal-to-noise ratio per tone) has been explored.

Of more immediate interest are the statistical summaries detailing modem performance as a function of E_b/N_0. In Figure 4.10 comparative bit error rates are presented for the 16 parallel and the ANDVT modem using the Golay code; a two-path HF channel is assumed. Similar results appear in Figure 4.11 for a 6-path split ray HF channel model. The raw error rate before decoding for the

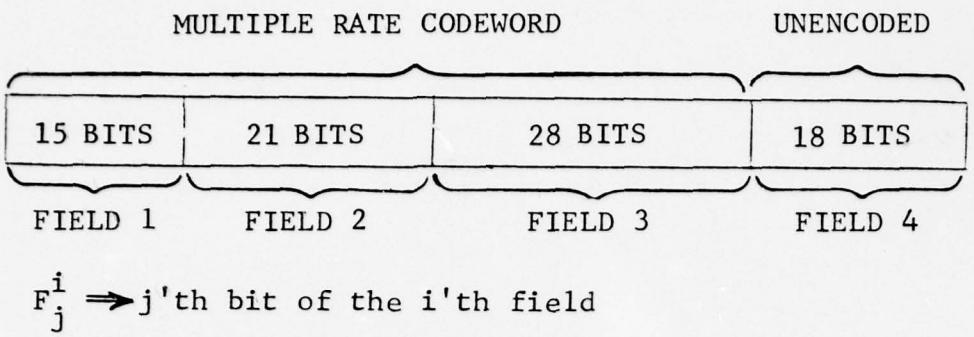


$F_j^i \Rightarrow$ j'th bit of the i'th field

BIT 2	F_1^1	F_{16}^3	F_1^4	F_{17}^3	F_2^1	F_2^4	F_{18}^3	F_{13}^3	F_3^4	F_4^4	F_{19}^3	F_4^1	F_5^4	F_{20}^3	F_6^4	F_7^4	F_{21}^3	F_{14}^3	F_6^1	F_8^4
BIT 1	F_1^4	F_{20}^4	F_1^3	F_{21}^4	F_{22}^4	F_2^3	F_{14}^1	F_{23}^4	F_3^1	F_3^3	F_{24}^4	F_{25}^4	F_4^3	F_{26}^4	F_5^1	F_5^3	F_{27}^4	F_{28}^4	F_{29}^4	F_6^3
TONE NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

BIT 2	F_{22}^3	F_9^4	F_{10}^4	F_{23}^3	F_8^1	F_{11}^4	F_{24}^3	F_{15}^3	F_{12}^4	F_{13}^4	F_{25}^3	F_{10}^1	F_{14}^4	F_{26}^3	F_{15}^4	F_{16}^4	F_{27}^3	F_{12}^1	F_{17}^4	F_{28}^3	F_{18}^4
BIT 1	F_{30}^4	F_7^1	F_7^3	F_{31}^4	F_{32}^4	F_8^3	F_{15}^1	F_{33}^4	F_9^1	F_9^3	F_{34}^4	F_{35}^4	F_{10}^3	F_{36}^4	F_{11}^1	F_{11}^3	F_{37}^4	F_{38}^4	F_{12}^3	F_{39}^4	F_{13}^1
TONE NO.	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41

Figure 4.8 Modulation Format Bit Assignments, Multiple Rate Code (43,15;12) $\Rightarrow$ (43,30;6)



BIT 2	F_{16}^3	F_1^1	F_1^2	F_{17}^3	F_2^1	F_2^2	F_{18}^3	F_{13}^3	F_1^4	F_3^2	F_{19}^3	F_4^1	F_2^4	F_{20}^3	F_4^2	F_3^4	F_{21}^3	F_{14}^3	F_6^1	F_5^2
BIT 1	F_{11}^2	F_9^4	F_1^3	F_{12}^2	F_{10}^4	F_2^3	F_{14}^1	F_{13}^2	F_3^1	F_3^3	F_{14}^2	F_{11}^4	F_4^3	F_{12}^4	F_5^1	F_{15}^2	F_5^3	F_{16}^2	F_{13}^4	F_6^3
TONE NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

BIT 2	F_{22}^3	F_4^4	F_6^2	F_{23}^3	F_8^1	F_5^4	F_{24}^3	F_7^2	F_{15}^3	F_6^4	F_{25}^3	F_{10}^1	F_8^2	F_{26}^3	F_7^4	F_9^2	F_{27}^3	F_{12}^1	F_8^4	F_{10}^2	F_{21}^2
BIT 1	F_{14}^4	F_7^1	F_7^3	F_{15}^4	F_8^3	F_{17}^2	F_{15}^1	F_{16}^4	F_9^1	F_9^3	F_{18}^2	F_{17}^4	F_{10}^3	F_{19}^2	F_{11}^1	F_{11}^3	F_{18}^4	F_{20}^2	F_{12}^3	F_{28}^3	F_{13}^1
TONE NO.	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41

Figure 4.9 Modulation Format Bit Assignments, Multiple Rate Code (64,36;12) ⇒ (64,51;6)

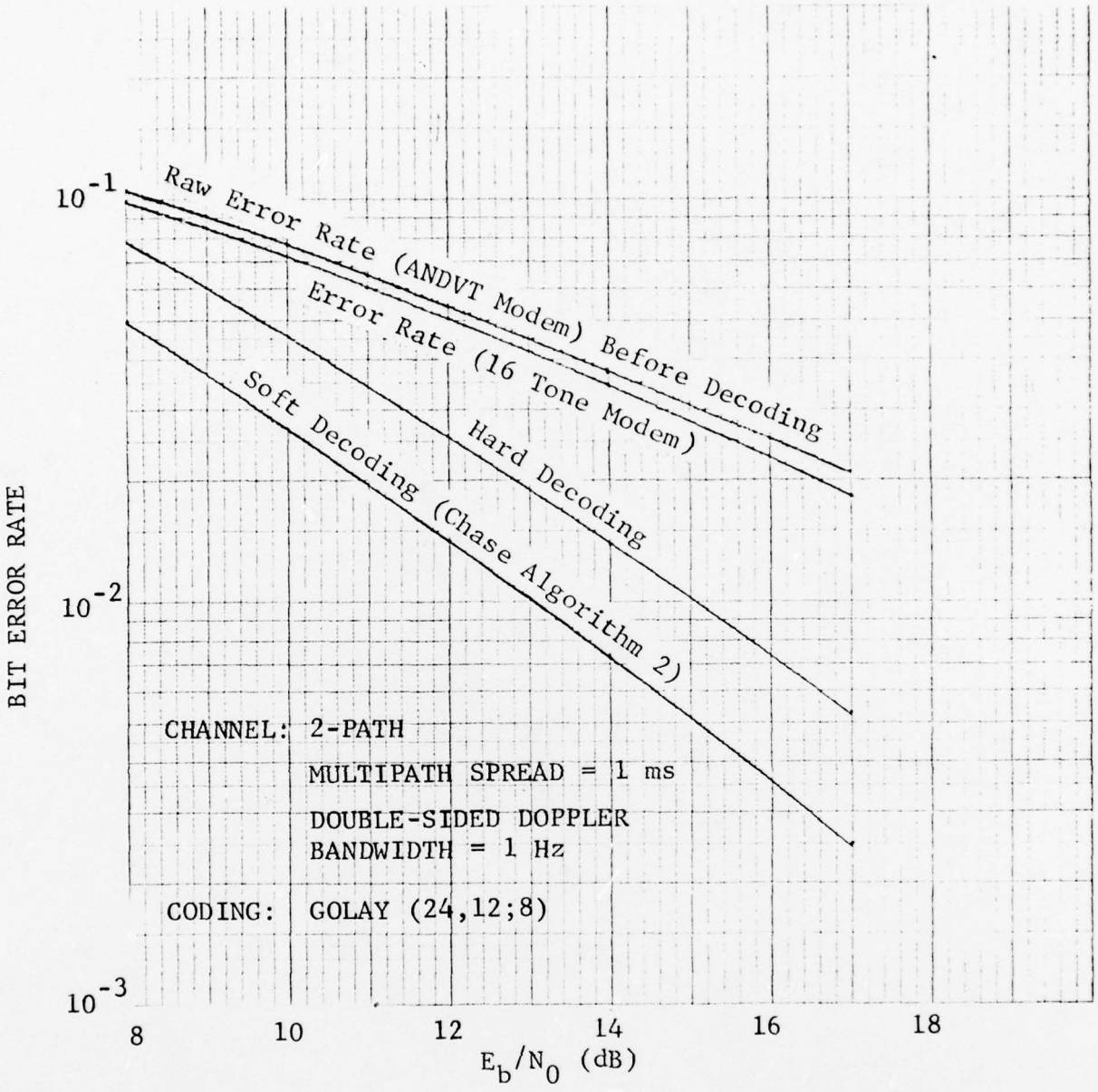


Figure 4.10 Simulated Bit Error Rates for the ANDVT Modem and a 16 Parallel Tone Modem (2-Path Model, Golay Code)

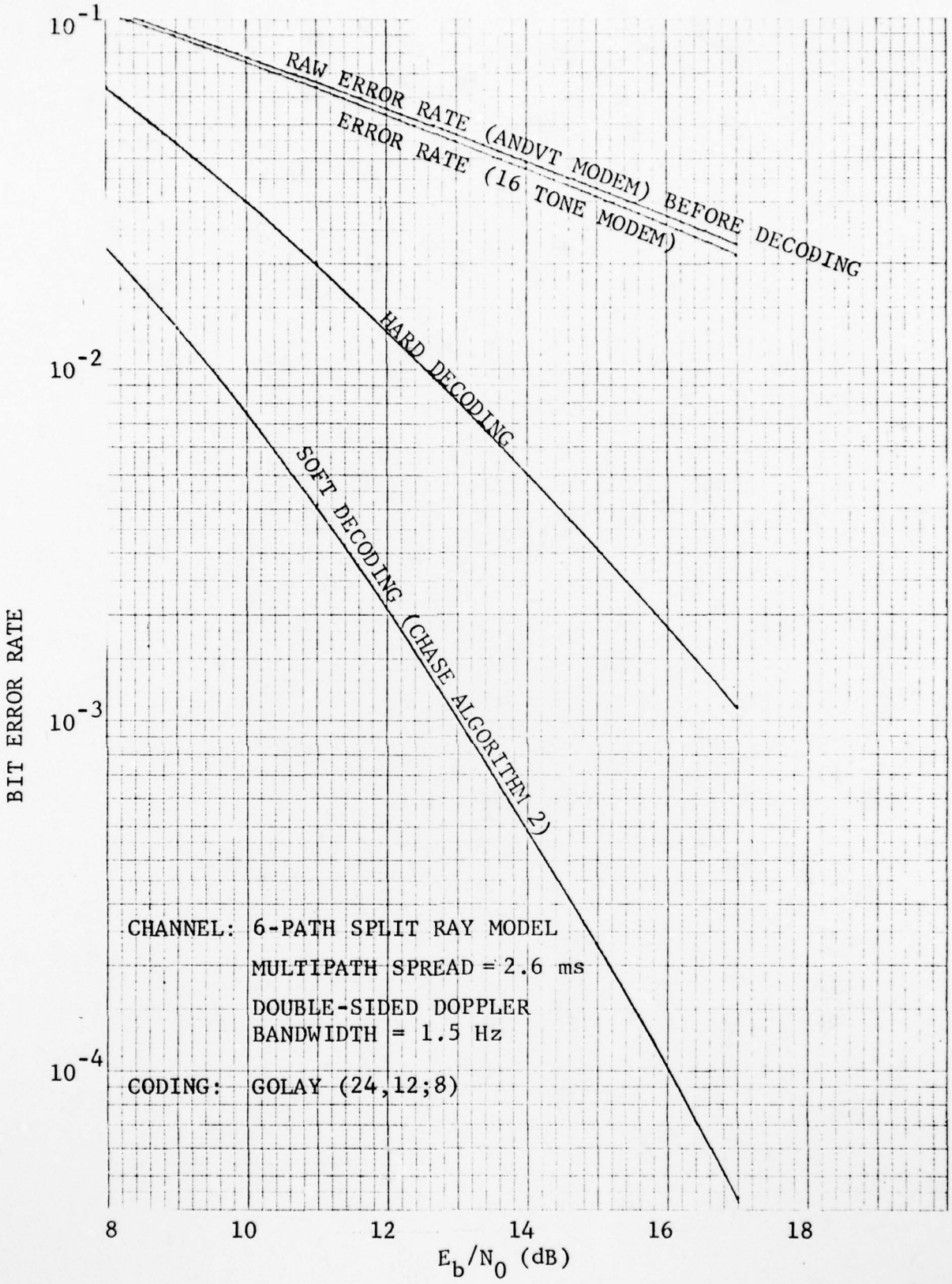


Figure 4.11 Simulated Bit Error Rates for the ANDVT Modem and a 16 Parallel Tone Modem (6-Path Split Ray Model, Golay Code)

ANDVT modem applies to the 54 bits which comprise the LPC-10 quantized parameters. The curves which present results for the Golay hard decoder and the Golay soft decoder (Chase algorithm 2) obviously apply only to the 24 bits which are protected by the Golay code. Coding gains, relative to raw bit error rate, are significantly larger for the 6-path split ray model than for the 2-path model. This is directly attributable to the greater in-band diversity for the 6-path split ray model.

Bit error rates for a 16 parallel tone modem and the ANDVT modem employing a multiple rate code are plotted in Figure 4.12; a 2-path HF channel is assumed. Figure 4.13 contains comparative bit error rates for conditions under which a 6-path split ray HF channel model is appropriate. In all the multiple rate code simulations conducted, a burst error-correcting capability of 1 has been assumed for the code. In addition, the coding has been implemented with a decoding delay equivalent to the duration of ten LPC-10 frames. With reference to the structure in Figure 4.6, the delay parameter τ has a value equal to the duration of ten LPC-10 frames.

Statistics are kept separately for the bits which receive first and second levels of protection from the multiple rate code. The annotation on the curves of Figures 4.12 and 4.13 is consistent with the notation of Section 4.2; K1 denotes the set of bits which receive the first level of protection, while K2 denotes the set of bits which receive the second level of protection. Again, relative coding gains are larger for the 6-path split ray HF model than for the 2-path HF model. This is dramatically exemplified by comparing the performance of the soft decoder acting on the bits receiving the second level of protection with the performance of the hard decoder acting on the bits receiving the first level of protection.

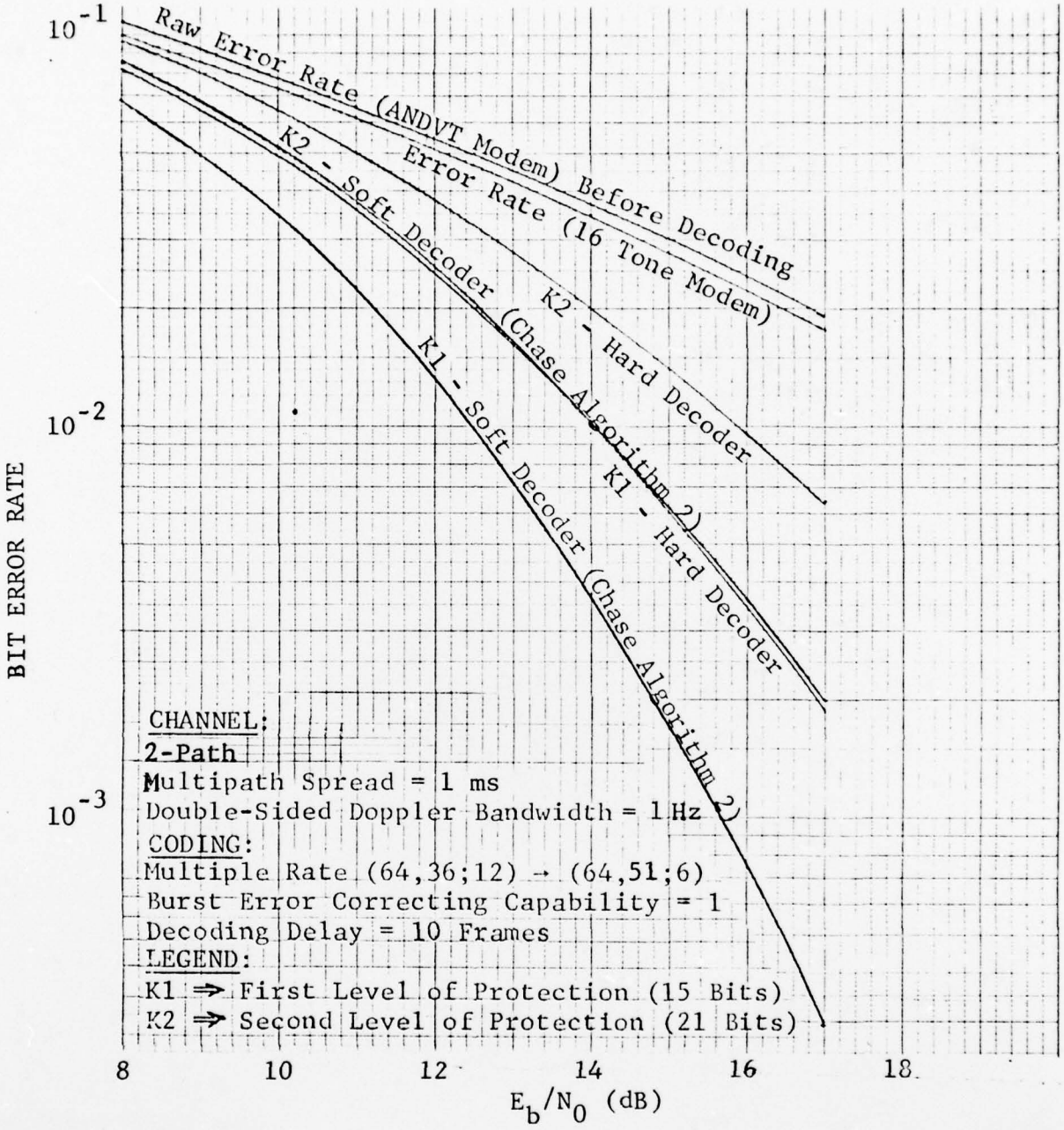


Figure 4.12 Simulated Bit Error Rates for the ANDVT Modem and a 16 Parallel Tone Modem (2-Path Model, Multiple Rate Code)

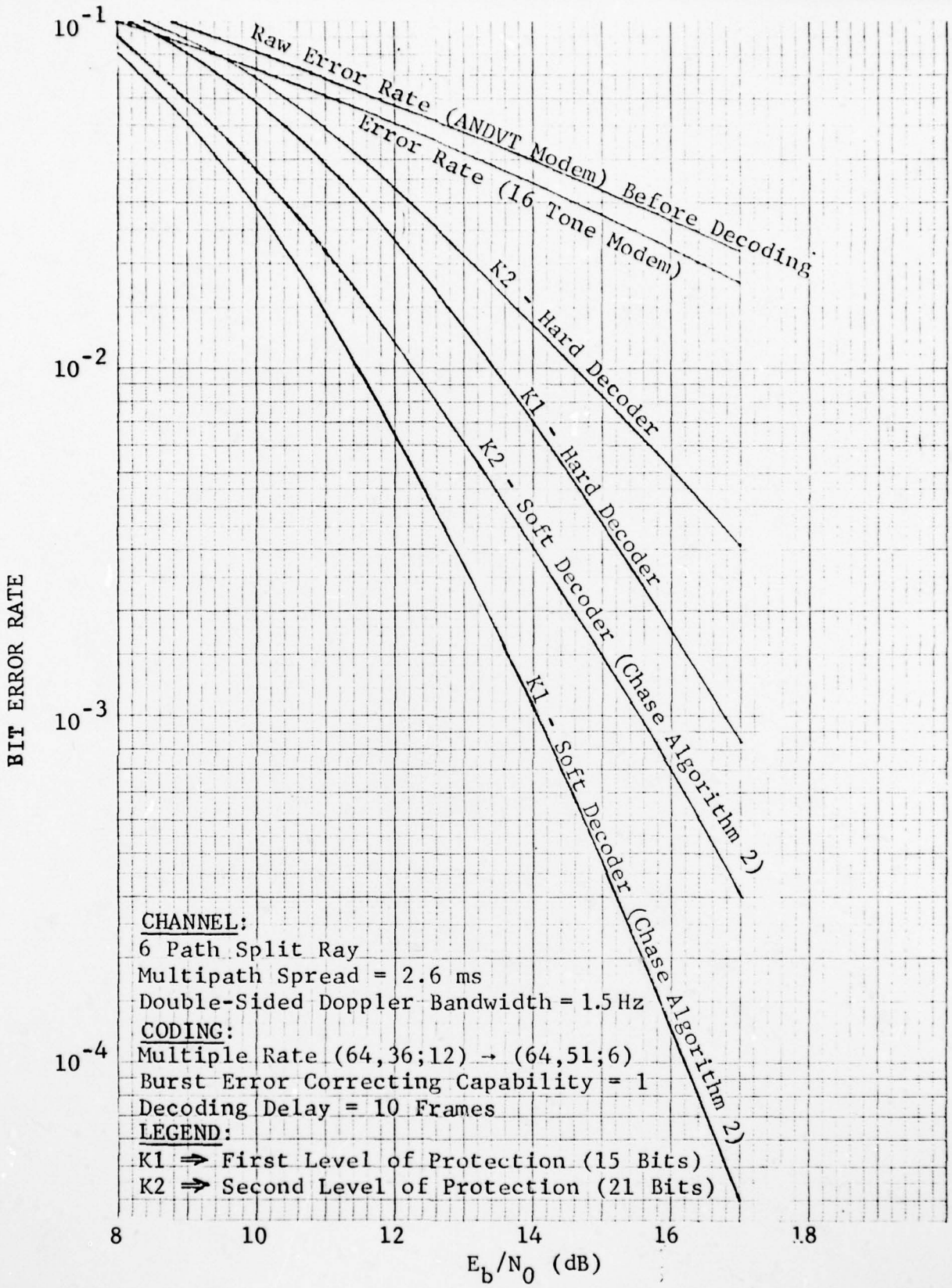


Figure 4.13 Simulated Bit Error Rates for the ANDVT Modem and a 16 Parallel Tone Modem (6-Path Split Ray Model, Multiple Rate Code)

It is of some interest to determine which of the two principal coding structures (Golay, multiple rate) performs best in conjunction with the ANDVT modem. From Figures 4.9 through 4.13 it is clear that the Golay code outperforms the multiple rate code at low E_b/N_0. However, at higher E_b/N_0, soft decoding the bits which receive the multiple rate codes' first level of protection produces lower bit error rates than soft decoding the bits protected by the Golay code. The crossover point occurs at a lower E_b/N_0 for the 2-path HF model than for the 6-path split ray model. No conclusions can be drawn from these observations, however, since the two coding structures protect different numbers of bits. The final measure of relative value must be speech-listening quality.

REFERENCES

[4.1] D. Chase and L. J. Weng, "Multiple Burst Error Correction Techniques for Slowly Fading Channels," IEEE Trans. on Information Theory, Vol. IT-22, No. 5, September 1976, pp. 505 - 513.

[4.2] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," IEEE Trans. on Information Theory, Vol. IT-18, No. 1, January 1972.

SECTION 5

ANDVT IMPLEMENTATION STUDY

5.1 Introduction

This section describes the results of a study of the implementation of the ANDVT modem by a programmable processor. This study was undertaken to satisfy the contract requirement that the proposed modem algorithms be checked for implementation practicality and has influenced the algorithms selected. The study focusses on obtaining estimates of the processor time required to perform each of the modem tasks. The lack of timely information concerning candidate processors necessitated the use of a hypothetical processor model. This model was selected to be simple enough so that it would not exceed the capabilities of the candidate processors and yet to be general enough so that the performance of the candidate processors could be easily related to that of the model. A description of this model may be found in Section 5.3.

The lack of knowledge of the input/output and operating systems precludes the assumption of a sophisticated multi-tasking operating environment. The study has thus been limited to a single task mode appropriate to a half-duplex device. Since the modulator portion of the modem is much less complex than the demodulator, the peak processing time, and thus the limits to performance, will lie with the demodulator implementation. Because of this and the limited amount of time for the accomplishment of this study, only the demodulator implementation has been studied in detail.

In the sequel the modulator will be mentioned only in regards to the demodulator functions of which it performs the inverse function.

The remaining sections will briefly review the operation of the modem, describe the assumptions made in the study, and present the results of the study. Section 5.2 provides the review of the modem, Section 5.3 describes the processor model, and Section 5.4 presents the study of the demodulator algorithms. A summary of the results and an estimate of the memory requirements is given in Section 5.5.

5.2 Modem Review

5.2.1 Signaling Format

The ANDVT modem is a multi-tone modem with a frame time compatible with the LPC-10 speech processor. The tone modulation is four-phase DPSK. The transmitted phase at the beginning of a frame is defined as:

$$\theta_k = \theta_{k-1} + \theta_s$$

where

θ_k = tone phase at beginning of current frame

θ_{k-1} = tone phase at beginning of previous frame

θ_s = phase shift due to data.

The values for θ_s given the two data bits x_1x_0 modulating the tone are specified in Table 5-1. The tone library may consist of about 40 tones (depending on the error correcting coding format) and an empty frequency slot among the active tones. This empty slot is used by the receiver to maintain proper frame synchronization. No Doppler reference tone is present because the receiver will recover Doppler tracking information from the data phase.

In initiating communications, the transmitter will transmit a preamble to allow synchronization of the receiver. The preamble consists of three sequences: a four-tone Doppler reference sequence, a frame sync sequence of wideband pulses, and a phase reference frame with about 40 tones. The four-tone Doppler reference sequence announces the presence of a signal to the receiver and allows the receiver to estimate the Doppler offset of the incoming signal. The frame sync sequence consists of a wideband signal modulated by a sequence whose autocorrelation function exhibits a high peak-to-sidelobe ratio. The detection of these peaks allows the receiver to find the proper locations of the frame boundaries. The phase reference frame establishes the starting phase of the DPSK modulation. The differential phase changes of the successive frame begins the transmission of data.

5.2.2 Sampling Rate Considerations

For a digital implementation of a multi-tone HF modem it is desirable that the chosen sampling rate result in an integral number of samples over both the frame time and the integration time. For the ANDVT modem frame time of 22.5 ms, the acceptable range of integration times is approximately 16 ms (a 1.5 dB loss

TABLE 5-1

TRANSMITTER DIFFERENTIAL PHASE

Information Bits		θ_s
x_1	x_0	
0	0	45^o
0	1	135^o
1	0	315^o
1	1	225^o

in signal power) to 20 ms (a 2.5 ms guard time). An additional constraint due to the planned use of a 64 point DFT is that the integration time contain 64 samples.

Let the number of samples/frame be 64 + n

T_I = Integration time

$$= \frac{64}{64 + n} 22.5 \text{ ms}$$

but

$$16 \text{ ms} \leq T_I \leq 20 \text{ ms}$$

$$\therefore \quad 8 \leq n \leq 26$$

The following relations can also be developed:

T_G = Guard time = $\frac{n}{64 + n} 22.5$ ms

f_s = Sampling rate of analytic baseband signal = $\frac{64 + n}{22.5 \text{ ms}}$

Δf = Tone spacing = $f_s/64$

N = Number of tones = $BW/\Delta f$

BW = Channel bandwidth.

The quantities are ennumerated in Table 5-2 for the allowable values of n and three values of BW.

5.2.3 Modulator Functions

A flow chart of the modulator is given in Fig. 5.1. At the start of a message four signal sequences must be sent to allow the demodulator to acquire, synchronize, and properly decode the message. These are:

- Doppler preamble
- Frame sync preamble
- Phase reference frame
- Cryptographic key and net control frames.

These are followed by a sequence of data frames.

5.2.3.1 Doppler Preamble

The Doppler preamble consists of four unmodulated tones. It announces the start of a message and provides a reference whereby the amount of Doppler offset can be measured at the receiver. One period of this sequence is generated by an inverse DFT operation on the desired spectrum. This is then repeated for the duration of the Doppler preamble.

5.2.3.2 Frame Sync Preamble

The exact time at which the receiver declares signal presence will vary with the channel conditions and the AGC parameters of the radio. Consequently the signal must contain a reference from which the receiver can recover the boundaries of the succeeding frames. This function is accomplished by the frame sync preamble. The frame sync preamble consists of a complex sequence

TABLE 5-2

ALLOWABLE SAMPLING RATES

n	T_I	T_G	f_s	Δf	N: 2400 Hz	N: 2700 Hz	N: 3000 Hz
8	20	2.5	3200.00	50.00	48	54	60
9	19.73	2.77	3244.44	50.69	47	53	59
10	19.46	3.04	3288.89	51.39	46	52	58
11	19.20	3.30	3333.33	52.08	46	51	57
12	18.95	3.55	3377.78	52.78	45	51	56
13	18.70	3.80	3422.22	53.47	44	50	56
14	18.46	4.04	3466.67	54.17	44	49	55
15	18.23	4.27	3511.11	54.86	43	49	54
16	18.00	4.50	3555.56	55.56	43	48	54
17	17.78	4.72	3600.00	56.25	42	48	53
18	17.56	4.94	3644.44	56.94	42	47	52
19	17.35	5.15	3688.89	57.64	41	46	52
20	17.14	5.36	3733.33	58.33	41	46	51
21	16.94	5.56	3777.78	59.03	40	45	50
22	16.74	5.76	3822.22	59.72	40	45	50
23	16.55	5.95	3866.67	60.42	39	44	49
24	16.36	6.14	3911.11	61.11	39	44	49
25	16.18	6.32	3955.56	61.81	38	43	48
26	16.00	6.50	4000.00	62.50	38	43	48

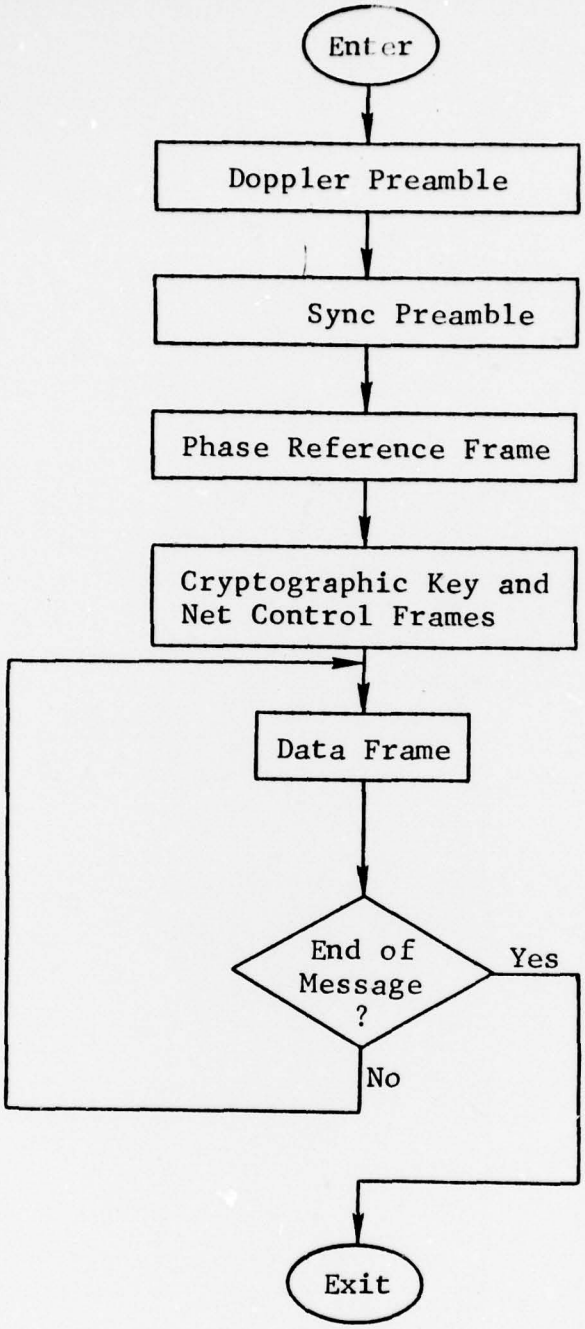


Figure 5.1 Modulator Flow Chart

modulating a wideband pulse. The modulating sequence consists of a few repetitions of a sequence selected to have an autocorrelation function with a high peak-to-sidelobe ratio. The receiver uses a pulse filter matched to the wideband pulse shape to recover the sequence. The correlation can then be computed to locate the frame sync boundary.

5.2.3.3 Phase Reference Frame

The modulation format described in Section 5.2.1 is four-phase DPSK. Since the information is carried by the frame-to-frame phase difference, the first frame must provide the reference for the next frame. The 64-point, complex FFT routine of the demodulator is used to perform an FFT^{-1} and produce 128 real samples in the integration time of the frame. This sequence is extended partially to fill out the frame. The initial values for tone phase are selected to minimize the ratio of peak-to-average power in the transmitted signal.

5.2.3.4 Cryptographic Key and Net Control Frames

A binary sequence specifying the cryptographic key and the net control designator are heavily encoded for error correction purposes and modulate a number of frames as per Section 5.2.1. The codes used are described in Section 3 of this report. They are easily implemented using a linear feedback shift register to encode the parity section.

5.2.3.5 Data Frames

Each data frame lasts for one frame time (22.5 msec) and contains the encrypted speech parameters for the corresponding speech frame. The most critical bits of the speech parameters

are protected by a block code prior to the modulation of the tone phases. The code improves the quality of the reconstructed speech in the presence of channel errors. After each frame the modulator checks for end of message and decides whether or not to continue transmitting the data frames.

5.2.3.6 $\underline{\text{FFT}^{-1}}$ Implementation

The modulator shares the FFT routine of the demodulator and uses it to translate from the desired tone phases to the real samples representing the signal in the time domain. While the demodulator uses an FFT routine, the duality of the discrete Fourier transform indicates that an inverse FFT can be implemented by transforming the reverse sequence of the frequency components and dividing by N, the number of samples.

Let

$$X(k) = \text{DFT}\ \{x(n)\}$$

where

$X(k)$ = frequency sample
$x(n)$ = time samples.

Then

$$x(n) = \text{DFT}^{-1}\ \{X(k)\} = \frac{1}{N}\ \text{DFT}\ \{X(-k)\}$$

Figure 5.2 shows a flow graph of the means by which the 128 time samples are produced using the 64-point DFT^{-1}. The variables in this graph are:

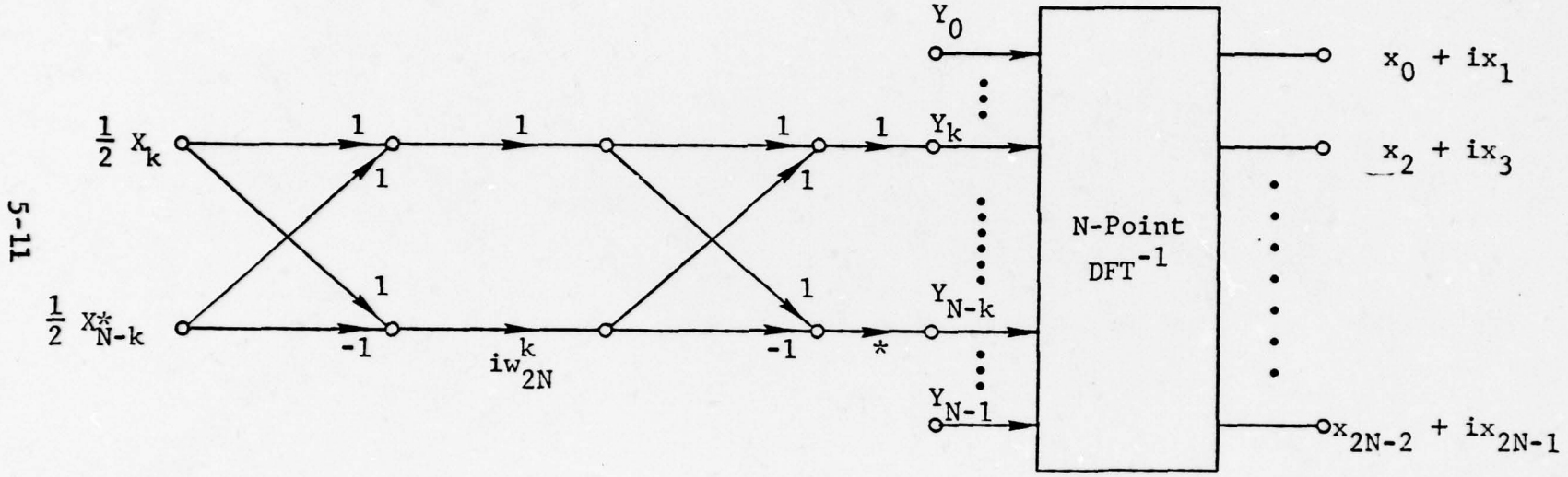


Figure 5.2 Flow Graph of Processing for Real Output DFT

X_k = Frequency samples

X_k^* = Conjugate of X_k

$W_{2N} = e^{-i2\pi/2N}$

x_n = Time samples

$i = \sqrt{-1}$

This method uses the fact that the spectrum of a real sequence (of length 2N) has conjugate symmetry ($X_{2N-k} = X_k^*$) to reduce the DFT size requirement from 2N points to N points. In this fashion both the real and imaginary parts of the output contain information. This technique is used to increase the commonality of the modulator and demodulator and reduce the storage requirements.

5.2.4 Demodulator Functions

A flow chart of the demodulator is given in Fig. 5.3. The boxes indicate the various tasks the demodulator performs. These are:

- Signal presence detection
- Doppler estimation
- Frame sync estimation
- Phase reference frame processing
- Cryptographic key and net control frame processing
- Data frame processing.

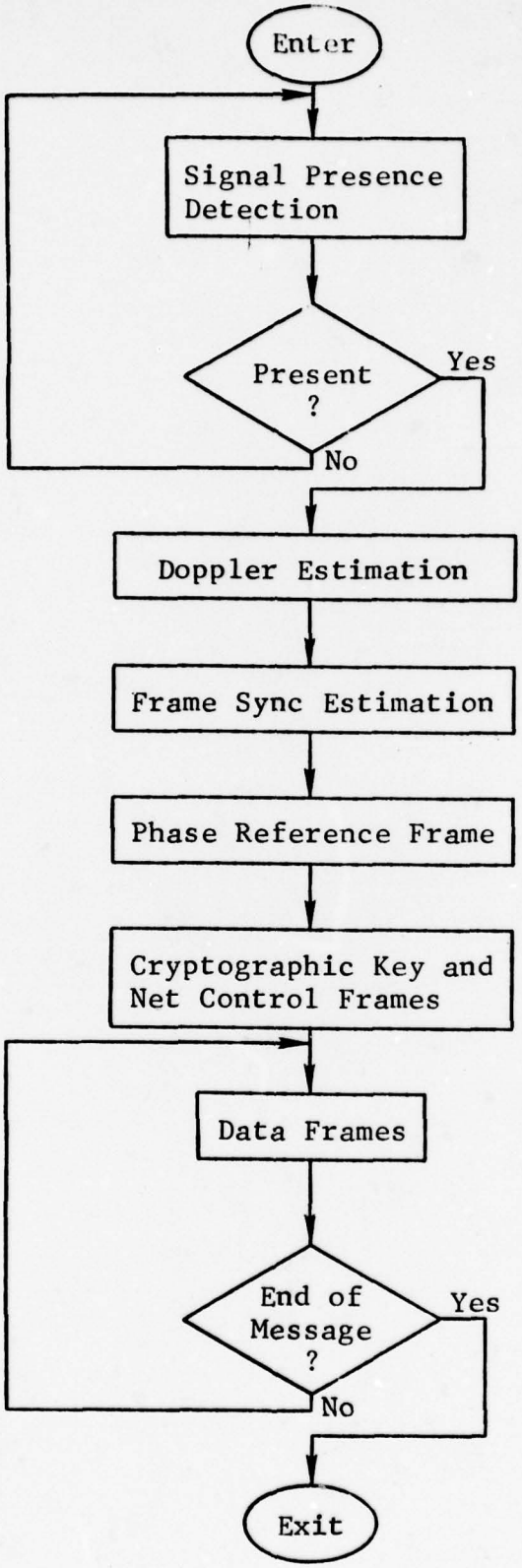


Figure 5.3 Demodulator Flow Chart

While quiescent the demodulator monitors the spectrum of the incoming signal to determine the presence of a signal. When a signal is present the demodulator begins the processing of the preamble. The first portion of the preamble consists of four unmodulated tones. These tones are processed to provide an estimate of the Doppler offset of the signal. This estimate is used in the following stages to remove the Doppler offset prior to processing. The second stage of the preamble is used to estimate the location of the frame boundaries in the frame-oriented processing which follows. This allows the modulated tones in these frames to be orthogonal over the integration time. The first frame-oriented signal is the phase reference frame. This frame provides the tone phase reference needed to demodulate the next frame in the DPSK format. The next set of frames contain a cryptographic key encoded for error correction. These frames are demodulated and decoded and sent to the deciphering unit. The frames which follow are of uniform format and contain the speech parameters. The most important bits are protected by an error correcting code.

5.2.4.1 Signal Presence Detection

Signal presence detection is accomplished by looking for the Doppler preamble. The signal presence detector consists of eight filters and a threshold detector. Four of the filters are centered on the transmitted tones (with enough bandwidth to pass the Dopplered tone) and four are centered to pick up only noise. The total energy at the outputs of the "noise" filters is summed over successive time instants and compared with "signal" energy similarly collected at the outputs of the tone filters. Signal presence is declared when the ratio of the two sums ("signal"-to-"noise") exceeds a design threshold.

5.2.4.2 Doppler Estimation

The Doppler estimator uses a three-step iterative procedure to generate an estimate of the Doppler offset. The four noise filters of the signal presence detector are discontinued while the four tone filters remain. The rate of change of the phase of each tone indicates the Doppler offset. To minimize the effect of noise, an energy weighted average over time and the tones is formed. In the first stage the filters are centered on the transmitted frequencies, with a large enough bandwidth to enclose all expected Doppler offsets. The estimate of the first stage is used to compute new filter coefficients. These coefficients specify filters whose center frequencies match those measured in phase 1, but with a smaller bandwidth. These filters are used in phase 2 to re-estimate the Doppler offset error. This estimation causes the filter to again be repositioned and narrowed for phase 3. The result obtained from phase 3 is taken as the final result of the Doppler estimation process. In the remaining processing this estimate is used to remove the Doppler offset by multiplying a rotating phasor times the complex samples of the received signal.

5.2.4.3 Frame Sync Estimation

The frame sync preamble consists of a complex sequence modulating a wideband pulse. The frame sync estimator first filters the incoming signal by a filter matched to the wideband pulse. This recovers the complex sequence. The incoming sequence is correlated with a local replica of the transmitted sequence.

The peaks of this correlation function indicate the positions of the frame boundaries in the following signal. A threshold is compared to the correlation. All values above the threshold are used to compute a magnitude weighted average of the correlation peak location. The sequence is repeated twice and the two averages obtained are combined to produce the estimate of frame boundary position.

5.2.4.4 Frame-Oriented Processing

The frame-oriented processing encompasses the phase reference frame, the cryptographic key and net control frames, and the data frames. Figure 5.4 presents a flow chart of the operations which occur during each frame.

The input samples are acquired in either an interrupt-driven background mode or are stored in a FIFO in the A/D system. These are complex samples $[s(kT) + j\hat{s}(kT)]$ of the input process $s(t)$. It has been assumed throughout this study that the Hilbert transform $\hat{s}(t)$ is done external to the processor. The complex samples have been chosen because Doppler correction can be accomplished simply by multiplying by a rotating complex phasor at the Doppler offset frequency. Following this, the 64 samples occupying the integration time are transformed by a 64-point complex FFT algorithm to measure the data tone phases.

If the current frame is the phase reference frame then the phases are stored and the processing for this frame ends. Otherwise, the measured tone phases are projected onto the phases from the last frame and thresholded by zero to accomplish the tone demodulation. As the tone demodulation is accomplished the

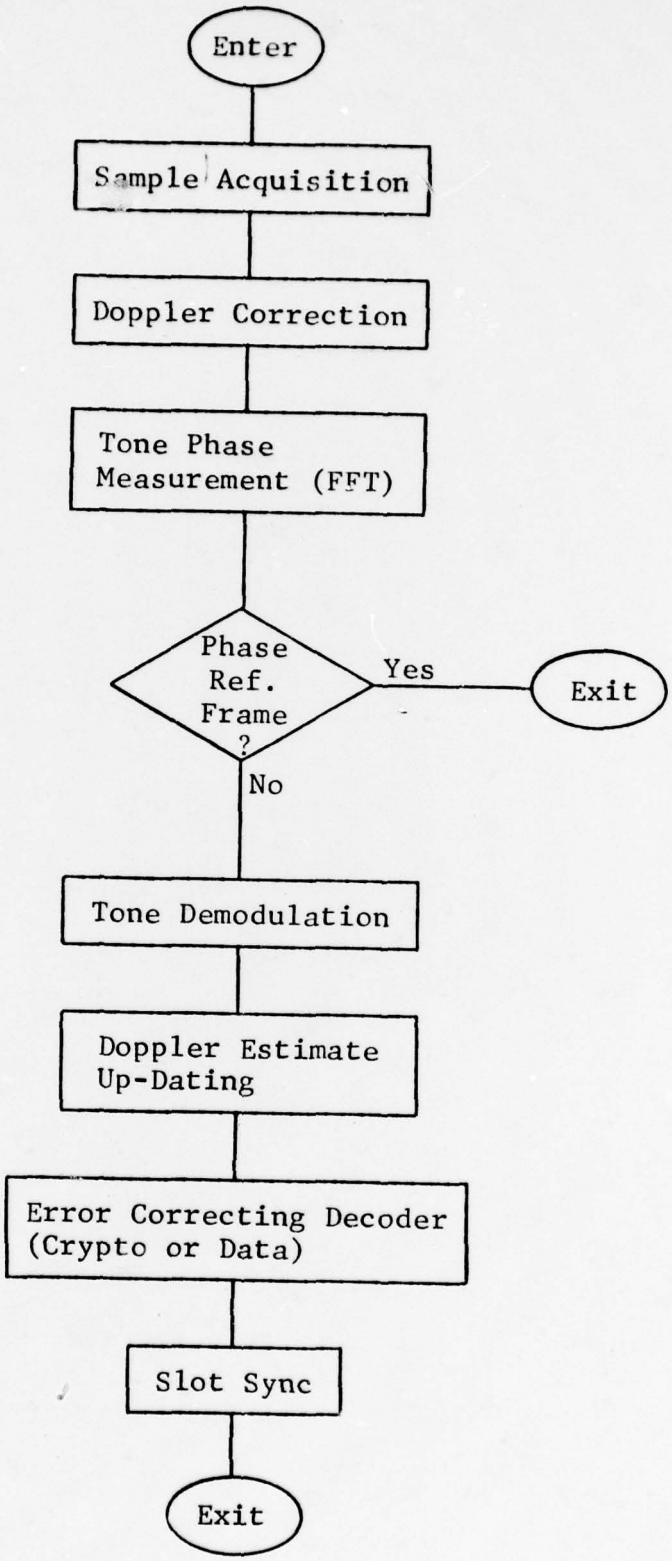


Figure 5.4 Demodulator Frame Activity Flow Chart

decisions are used to compare the received phases to the expected received phase. The mean of the angle difference indicates an error in the Doppler estimate. An amplitude weighted average of the Doppler estimate error is formed and is used to up-date the Doppler estimate used by the next frame.

The demodulated bits are then processed by an error correcting decoder. Two decoders are present in the demodulator, one for the cryptographic key and net control sequence, the other for the speech parameter data frames. The cryptographic key is given more protection than the speech parameters because its correctness is essential to the intelligibility of the entire message.

The final task is the slot sync tracking function. The DFT is used to measure the energy in a blank tone slot for both an early and a late integration gate. (These early and late gates overlap both the guard time and the integration time.) The energy difference provides a measure of the centering of the integration time between the adjacent phase transition points. This value is thus averaged and used to control the sample clock to maintain the optimum position of the integration time.

5.2.4.5 End-of-Message

Following each data frame the demodulator must check for end-of-message. If found, the demodulator is returned to the signal presence detection mode. A method of accomplishing this is to compare the energy in the active tone slots of the FFT to the energy in the empty tone slots. A low ratio can be taken as an indication of the loss of signal. The exact point at which the loss is declared will be influenced by the fading characteristics of the channel and the AGC characteristics of the radio, and deserves further study.

5.3 Processor Model

This section presents the processor model used in the implementation study. The model was chosen to be simple and general and was based on the AMD 2900 micro-processor family to match the Quintrell processors under consideration. Figure 5.5 presents the block diagram of the processor model. The AMD 2900 processor accepts input from the D-buss and places its output on the Y-buss. The D-buss handles data from the input devices, the data memory, and the fast multiplier. The Y-buss delivers data to the output devices, the data memory, and the fast multiplier. The micro-program sequencer contains the program memory and controls the operation of the entire system. The following assumptions have been made:

1) Modem functions are micro-programmed
2) The data memory contains both RAM and ROM
3) Constants can be supplied to the ALU from the micro-store
4) Overlapped operation:
   a) A memory reference and an ALU operation can be hidden behind a multiplication
   b) An ALU operation can be hidden behind a memory reference
5) Samples of the signal and its Hilbert transform are available.

Flow charts of the demodulator algorithms are presented in Section 5.4. The time required to perform these flow charts will be expressed in terms of the three quantities t_{ALU}, t_{memory}, and t_{mult}, the times for the execution of an ALU operation, a memory operation, and a multiply operation, respectively. Each block in the flow chart is labelled with the 4-tuple (R; n_{ALU}, n_{memory}, n_{mult}) where:

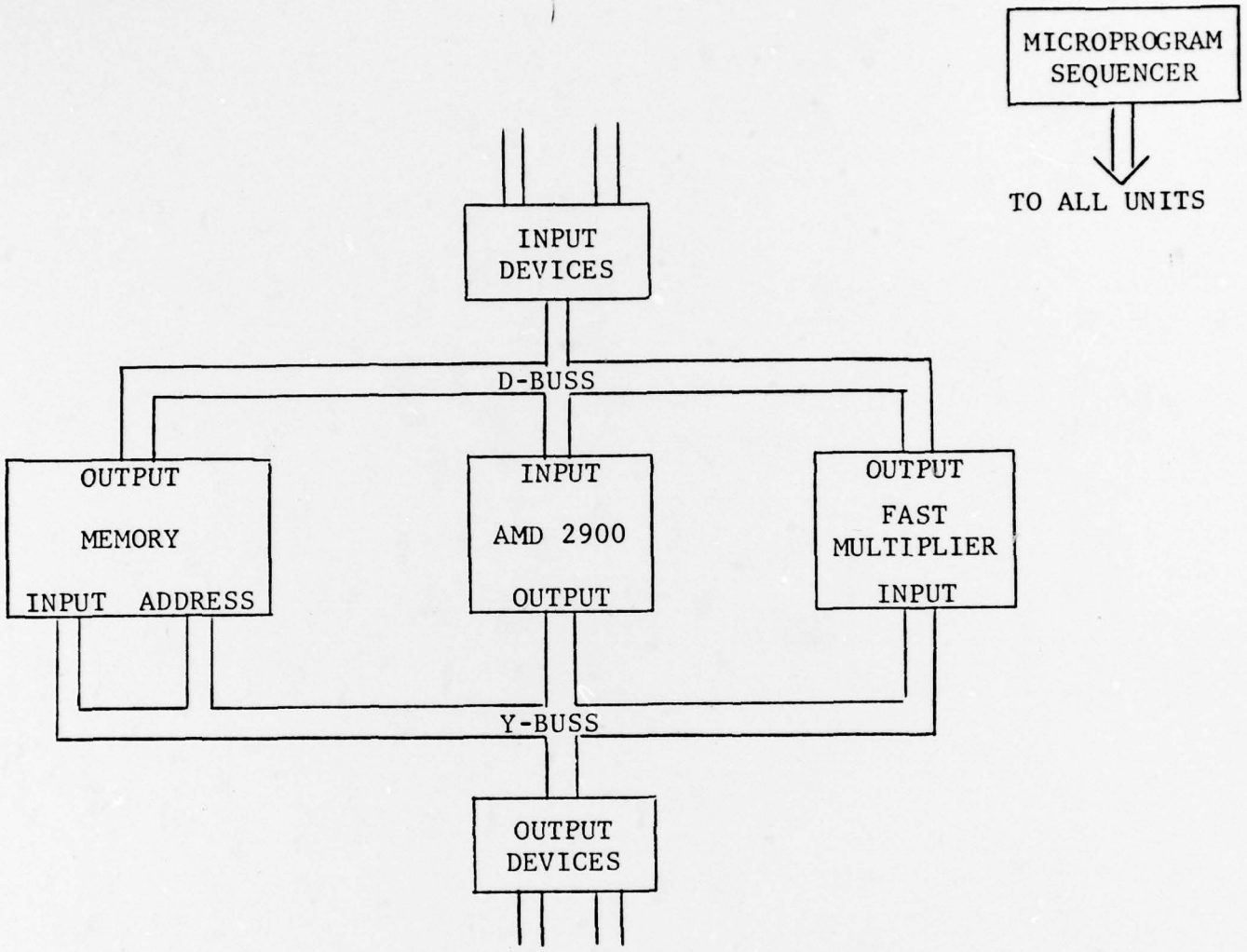


Figure 5.5 Processor Model

R = Number of times the block is executed in the flow chart

n_{ALU} = Number of ALU cycles (shifts, adds, branching, etc.) for one execution of the block

n_{memory} = Number of data memory cycles in one execution of the block

n_{mult} = Number of multiplications in one execution of the block.

In assigning (R; n_{ALU}, n_{memory}, n_{mult}), overlapped operation is accounted for by not counting memory cycles that can be hidden in a concurrent multiplication. The total execution time is thus:

$$t = t_{ALU} \sum_i R_i n_{ALU,i} + t_{memory} \sum_i R_i n_{memory,i} + t_{mult} \sum_i R_i n_{mult,i}$$

5.4 Demodulation Algorithms

5.4.1 Signal Presence Detection

Figure 5.6 shows the proposed structure for determining signal presence. The energy in the frequency slots covering the four Doppler acquisition tones is compared to a threshold set by the energy in four empty frequency slots. The filters are described by the transfer function

$$H(z) = \left(\frac{z}{z - a}\right)^2$$

This filter can be realized by the block diagram of Fig. 5.7. An estimate of the number of operations needed to implement a single filter is 7 ALU operations, 6 memory operations, and 8 multiply operations. This assumes concurrency of operation in that a memory operation and an ALU operation can be hidden behind a multiply

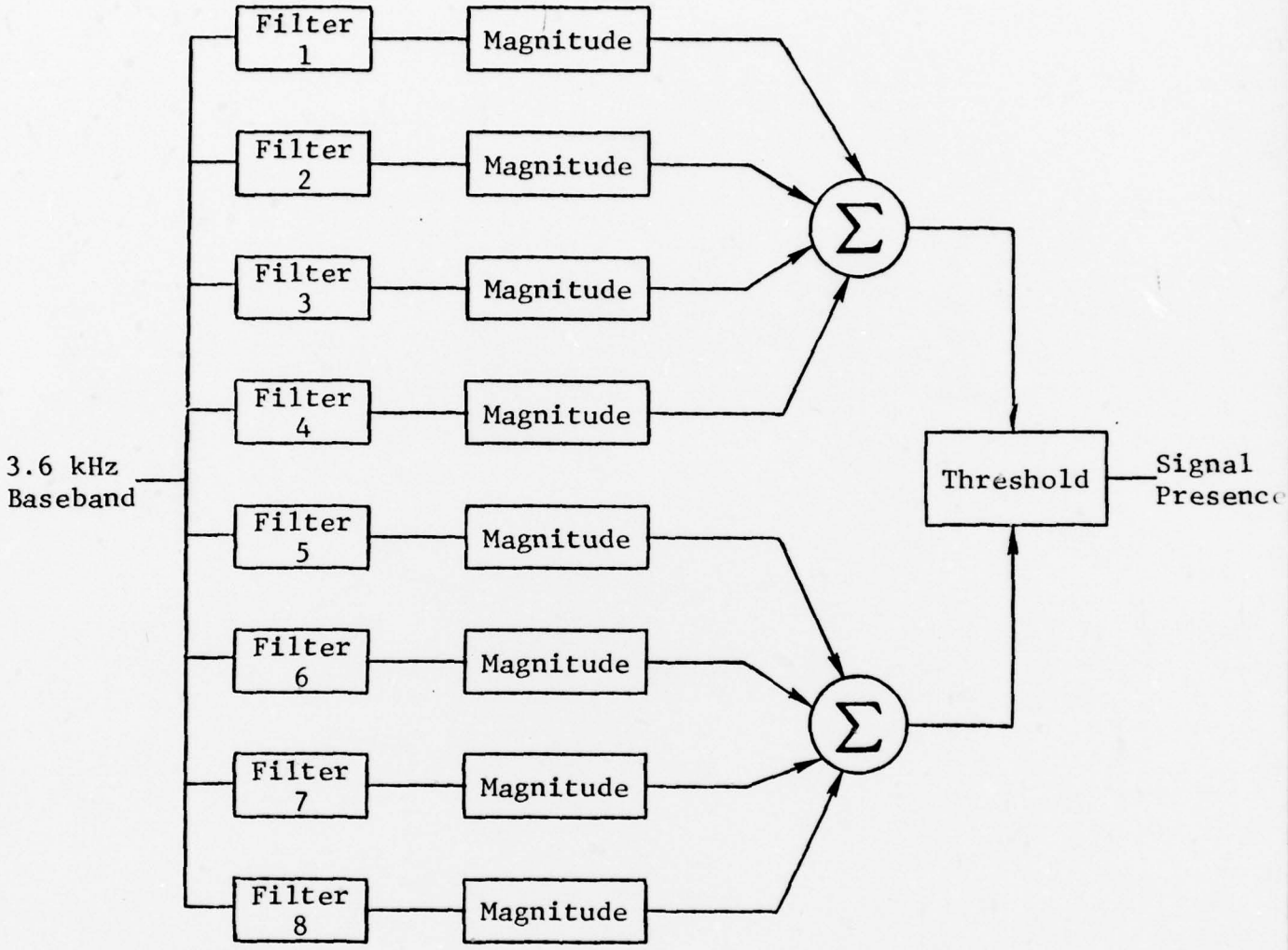


Figure 5.6 Signal Presence Detector

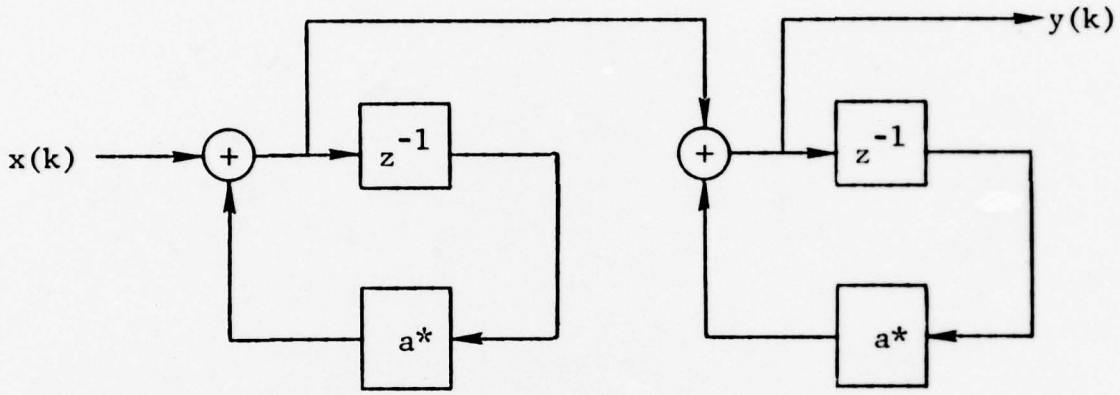


Figure 5.7 Bandpass Filter

*a is complex for the bandpass filter.

operation, and that an ALU operation can be hidden behind a memory operation. The further assumption is made that one sub-routine will perform all eight filters by operating on successive offsets in arrays containing the filter coefficients and state variables. The time to execute the filters for each sample value is

$$t = 56\ t_{ALU} + 48\ t_{memory} + 64\ t_{mult}$$

An estimate of the number of operations needed to perform the magnitude operation on a filter output is 4 ALU operations, one memory operation, and 2 multiply operations. The number of operations to perform the eight magnitude operations and the two summations is approximately 40 ALU operations, 8 memory operations, and 16 multiply operations. A simple threshold operation which requires the threshold to be exceeded N times consecutively before declaring signal presence can be done in about 10 ALU operations and 2 memory operations. Thus, the total time required per 8 magnitude operations and one threshold operation is

$$t = 50\ t_{ALU} + 10\ t_{memory} + 16\ t_{mult}$$

The filter operations must be done for each sample but an M to 1 sample rate reduction is possible after the filters. Thus the magnitude and threshold operations can be performed on every M'th filter output and the required computations spread over the time remaining after the filter operations during the M samples. The total execution time per sample for the detection of signal presence is then

$$\boxed{t_{detect} = \left(56 + \frac{50}{M}\right) t_{ALU} + \left(48 + \frac{10}{M}\right) t_{memory} + \left(64 + \frac{16}{M}\right) t_{mult}}$$

5.4.2 Doppler Estimation

The structure of the Doppler estimator is shown in Fig. 5.8. The four filters each isolate one of the Doppler reference tones. Each filter output is mixed with a reconstructed version of the transmitted tone to produce a signal which contains only a tone at the Doppler frequency. From these four signals an amplitude weighted average of the rate of change of the phase is computed. This value can be scaled to produce the Doppler frequency. These estimates can be averaged, if desired, to reduce the effects of noise.

The processing to determine the Doppler frequency requires the values of two derivatives per tone. These derivatives must be approximated digitally. An implementation in which the derivative is approximated by

$$\dot{x}(k) = x(k) - x(k-1)$$

has been studied. The input to the processor is modeled as

$$x(k) = \cos(k\Delta\theta)$$

$$y(k) = \sin(k\Delta\theta)$$

The output of the processor is $\widehat{\Delta\theta}$, the estimate of $\Delta\theta$. The values of $\widehat{\Delta\theta}$ and $\widehat{\Delta\theta}/\Delta\theta$ have been tabulated for various values of $\Delta\theta$ in Table 5-3.

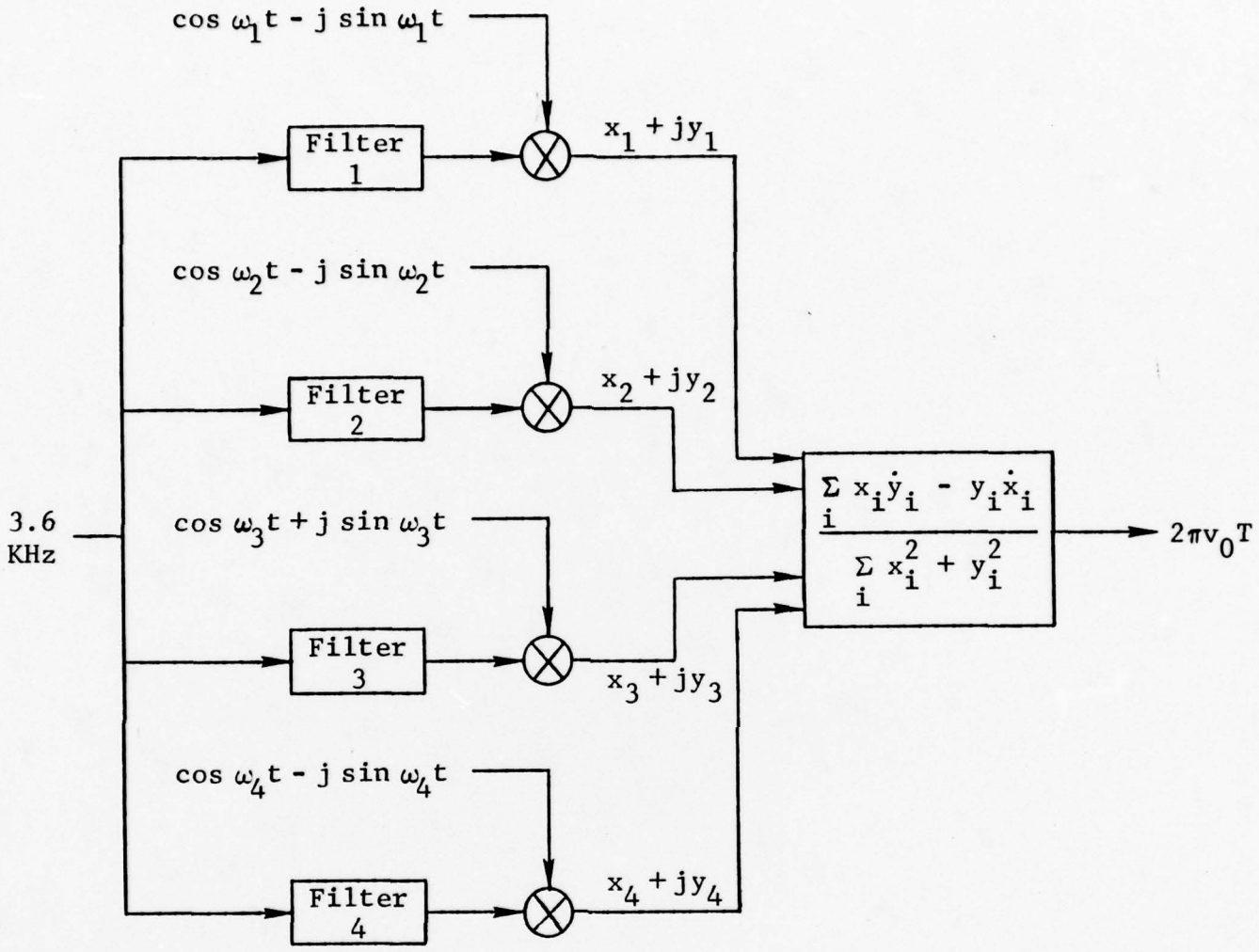


Figure 5.8 Doppler Estimator

TABLE 5-3

DOPPLER ESTIMATION ERROR

$\Delta\theta$	$\widehat{\Delta\theta}$	$\widehat{\Delta\theta}/\Delta\theta$
0^o	0^o	(1.000)
5^o	4.994^o	.999
10^o	9.949^o	.995
15^o	14.829^o	.989
20^o	19.596^o	.980
25^o	24.214^o	.969
30^o	28.648^o	.955
35^o	32.864^o	.939
40^o	36.829^o	.921
45^o	40.514^o	.900

The values of $\frac{\widehat{\Delta\theta}}{\Delta\theta}$ are fit excellently by

$$\frac{\Delta\hat{\theta}}{\Delta\theta} = 1 - \frac{.1}{(45)^2}(\Delta\theta)^2$$

$$\therefore \quad \widehat{\Delta\theta} = \Delta\theta - \frac{.1}{(45)^2}(\Delta\theta)^3$$

To keep the estimation error small, the sampling rate at the input to the processor must be high enough, relative to the maximum expected Doppler, so that $\Delta\theta$ is in the range of accurate estimation. This restricts the amount of sampling rate reduction that can occur after the multipliers in Fig. 5.8. No sampling rate reduction and a 3600 Hz sampling rate results in a .21 Hz error in the estimation of a 75 Hz Doppler frequency.

In implementing the Doppler estimator of Fig. 5.8 it is necessary to up-date the filters with each sample. The remainder of the processing can be done in a background manner by storing two successive outputs of each filter and processing them in the time leftover from sampling and filtering operations. The four filters are the same as the upper four filters of Fig. 5.6 and can be implemented in 28 ALU operations, 24 memory operations, and 32 multiply operations. The time required is

$$t = 28\ t_{ALU} + 24\ t_{memory} + 32\ t_{mult}$$

Since the derivatives required in the estimation process are approximated by the difference of two samples, only two successive output values from each filter are necessary to produce an estimate of the Doppler frequency. Also, since the absolute phase of each tone after the multiplier is of no consequence, only one of each pair of filter outputs needs to be multiplied by a complex phasor to remove the tone center frequency. If the output of the i'th filter is represented as $x_i(k) + jy_i(k)$, then the following equations define the Doppler estimator:

$$\dot{x}_i(k) = x_i(k) - x_i(k-1) \cos \omega_i T + y_i(k-1) \sin \omega_i T$$

$$\dot{y}_i(k) = y_i(k) - x_i(k-1) \sin \omega_i T - y_i(k-1) \cos \omega_i T$$

$$2\pi v_0 T = \frac{\sum_i (x_i(k)\dot{y}_i(k) - y_i(k)\dot{x}_i(k))}{\sum_i (x_i^2(k) + y_i^2(k))}$$

An estimate of the number of operations required to implement this is 146 ALU operations, 28 memory operations, and 32 multiplier operations. The time required to execute these instructions is

$$t = 146\ t_{ALU} + 28\ t_{memory} + 32\ t_{mult}$$

If a Doppler frequency estimate is made every P samples, the execution time per sample for the filters and the estimator is given by

$$\boxed{t_{estimate} = \left(28 + \frac{146}{P}\right) t_{ALU} + \left(24 + \frac{28}{P}\right) t_{memory} + \left(32 + \frac{32}{P}\right) t_{mult}}$$

5.4.3 Doppler Correction

The input to the processor is samples of the baseband audio signal, s(t), and its Hilbert transform $\hat{s}(t)$. The sum $r(t) = s(t) + j\hat{s}(t)$ represents the positive frequencies of s(t). The phase-keyed tones present in r(t) contain a Doppler offset which must be removed prior to demodulation. This can be accomplished by multiplying by the complex phasor $e^{-j2\pi v_0 t}$, where v_0 is the Doppler frequency. The signal $r_c(t) = r(t)e^{-j2\pi v_0 t}$ is used as the received, Doppler-corrected signal in the frame sync interval and in the demodulation interval.

The complex phasor $e^{-j2\pi v_0 t}$ is constructed from a look-up table containing 64 samples of one cycle of a cosine. The table is addressed by a pointer representing the product $v_0 t$. This

pointer is incremented each sample interval by a quantity related to the Doppler estimate. Because the table length is a power of 2, the pointer can be maintained within proper bounds simply by ignoring the carry out of the most significant bit. The top six bits of the pointer can be used to address an entry in the table and its neighbor. The remaining bits (10 in a computer with 16 bit words) are used to perform a linear interpolation between the two table values. This results in the real part of $e^{-j2\pi v_0 t}$. The process is repeated with a quarter cycle shift in the table look-up to find the imaginary part of $e^{-j2\pi v_0 t}$. The remaining step is the calculation of $r_c(t) = r(t)e^{-j2\pi v_0 t}$.

Figure 5.9 shows a flow chart for the Doppler correction algorithm. The variables used and their meanings are as follows:

θ = Equal to the current angle of the complex phasor used for Doppler correction.

$\Delta\theta$ = The sample-to-sample change in θ. $\Delta\theta$ is proportional to the Doppler frequency.

ITP = The fractional part of θ. ITP is used to interpolate between values from the cosine look-up table.

L1 and L2 = The indices for the cosine look-up table which bracket the desired value.

A = Real part of $e^{-j2\pi v_0 t}$.

B = Imaginary part of $e^{-j2\pi v_0 t}$.

r = Complex sample of the input.

r_c = Doppler corrected sample.

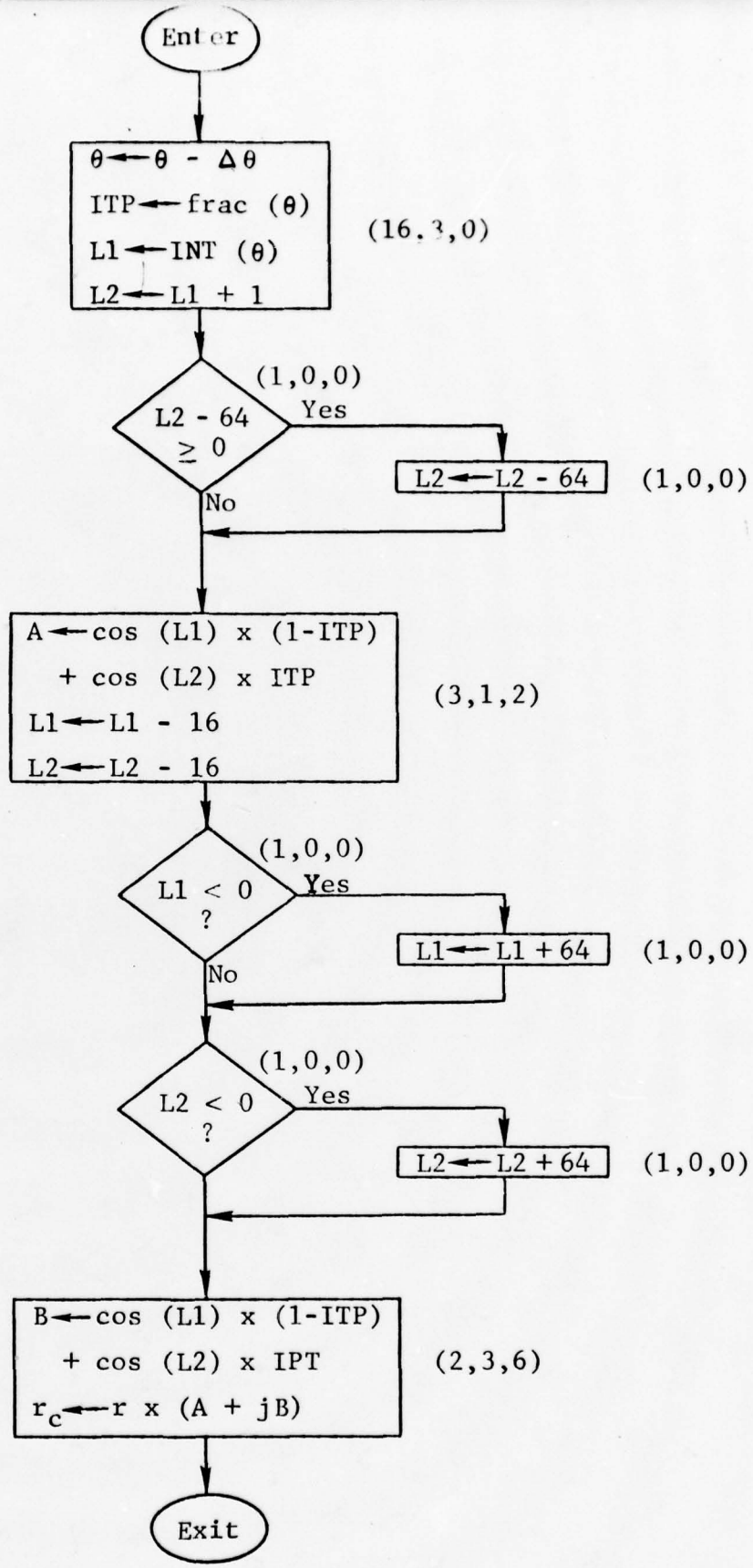


Figure 5.9 Doppler Correction Flow Chart

Beside each block of the flow chart is a 3-tuple (n_{ALU}, n_{memory}, n_{mult}) giving, respectively, the number of ALU operations, the number of memory operations, and the number of multiply operations to execute the block. In the worst case all blocks may be executed for a particular sample, requiring 27 ALU operations, 7 memory operations, and 8 multiply operations. The processor time to execute these operations is the per sample Doppler correction time t_{DC}:

$$t_{DC} = 27\ t_{ALU} + 7\ t_{memory} + 8\ t_{mult}$$

Doppler correction must take place during the sync preamble and during data transmission. This means that t_{DC} is a per sample overhead to the other processing occurring during these times.

5.4.4 Frame Sync Matched Filter

Frame sync is acquired by forming a weight average based on the correlation of the received signal with a local sequence. The elements of this sequence are ± 1 and $\pm j$. The sequence is 1 -1 1 j -j -1 j -j -j 1 j j 1 1 1. The correlation is formed by passing the received sequence through a FIR filter whose taps are the complex conjugates of the sequence terms. The algorithm takes advantage of the fact that no multiply operations are necessary when multiplying by ± 1 or $\pm j$.

Figure 5.10 shows a block diagram of the frame sync detector. The matched filter is implemented by a FIR filter with N taps. The magnitude squared of the output of the FIR filter is compared to a threshold. If the threshold is exceeded, the present term is used in the centroid calculation to determine the time of peak matched filter output. The phase reference frame will start a fixed number of samples following the peak of the matched filter output.

Figure 5.11 shows a flow chart implementing this function. This flow chart incorporates an ending criterion not mentioned in the introduction. The procedure ends and produces its result when a limit on the variable L is exceeded. L indicates the number of times the threshold has not been exceeded following the last time the threshold was exceeded. The variables used and their meanings are as follows:

k = Time index relative to which location of the matched filter peak is found, k does not start until the threshold on the matched filter output is first exceeded

L = A negative value of ℓ indicates that the threshold has not yet been exceeded, a non-negative value of ℓ indicates the number of times the threshold has not been exceeded since the last time it was exceeded

A = Numberator of centroid sum

B = Denominator of centroid sum

I = Loop counter for FIR filter

F = Filter output

r(I) = Delayed sample values used in the FIR filter

M = Magnitude squared of FIR filter output

Δ = Number of samples between the peak of the FIR filter output and the start of the phase reference frame

T = The number of samples to be skipped before the start of the phase reference frame.

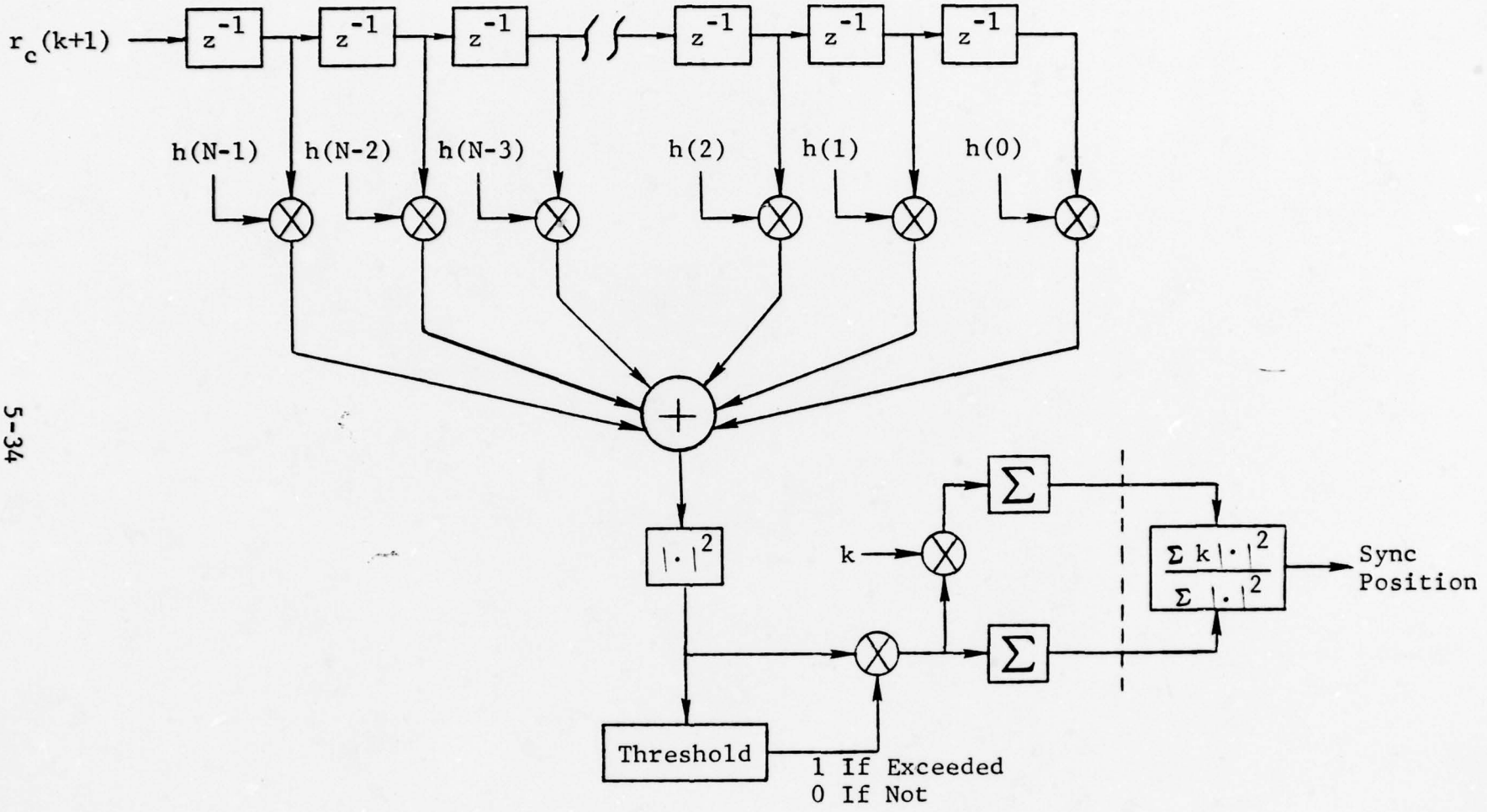


Figure 5.10 Frame Sync Matched Filter

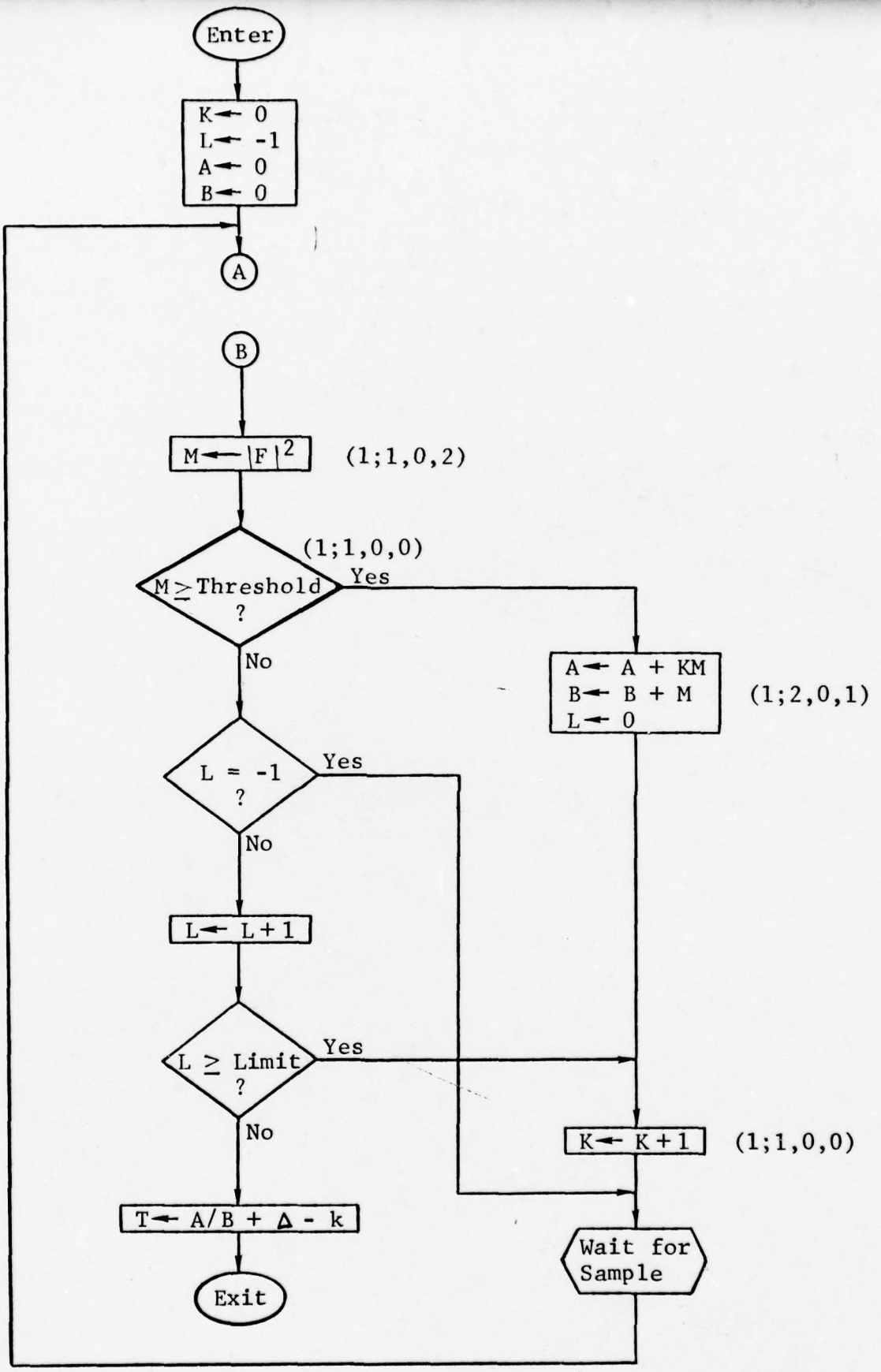


Figure 5.11a Frame Sync Flow Chart

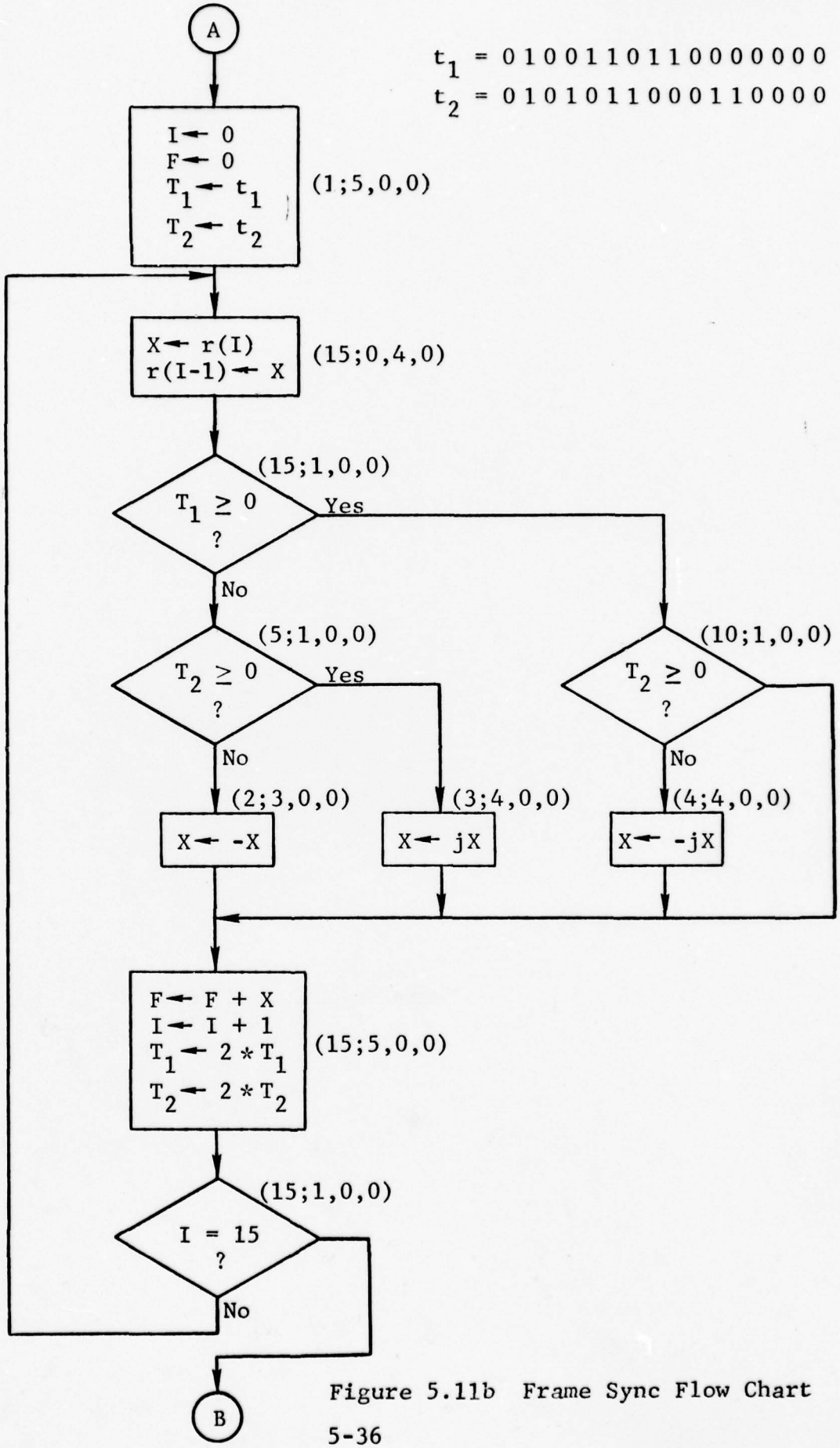


Figure 5.11b Frame Sync Flow Chart

The blocks in Fig. 5.11 along the longest per sample path are labelled with the 4-tuple (R; n_{ALU}, n_{memory}, n_{mult}). The number of operations required is 164 ALU operations, 60 memory operations, and 3 multiply operations. The time required per sample is

$$t_{MF} = 164\ t_{ALU} + 60\ t_{memory} + 3\ t_{mult}$$

5.4.5 64-Point FFT

Two algorithms for computing a 64-point DFT have been studied. The first is a radix-2, decimation-in-frequency algorithm and the second is a radix-4, decimation-in-frequency algorithm.

5.4.5.1 Radix-2 FFT

The flow chart of Fig. 5.12a and Fig. 5.12b gives an algorithm for the computation of a 64-point DFT by the radix-2, decimation-in-frequency technique. The algorithm is in place and accepts input in normal order and produces output in a bit reversed ordering. The (R; n_{ALU}, n_{memory}, n_{mult}) assignments shown assume a fixed-point representation of the numbers and a memory assignment of one word for each real quantity and two words for each complex quantity.

The looping structure of this flow chart allows the detection of all instances of multiplication by 1 or j and replaces them with addition to save execution time. The innermost loop (Fig. 5.12b) successively performs adjacent butterfly operations. The middle loop successively sequences through groups of butterflies

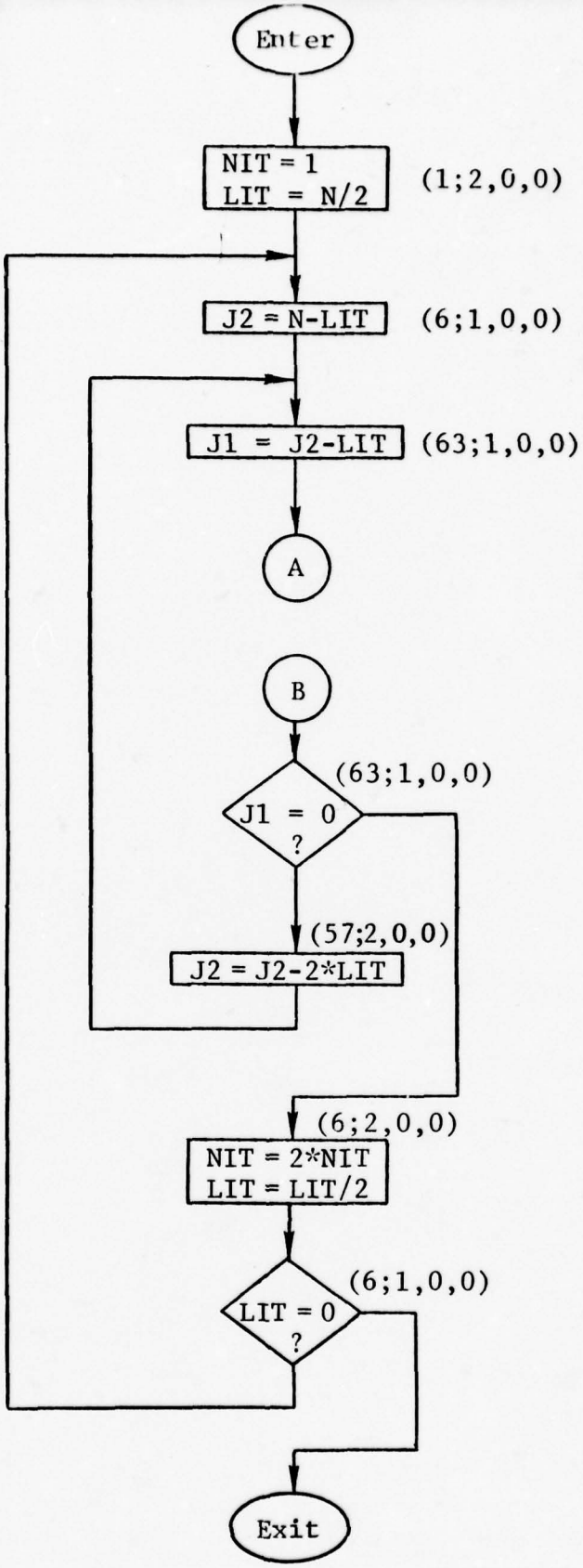


Figure 5.12a Radix-2 FFT

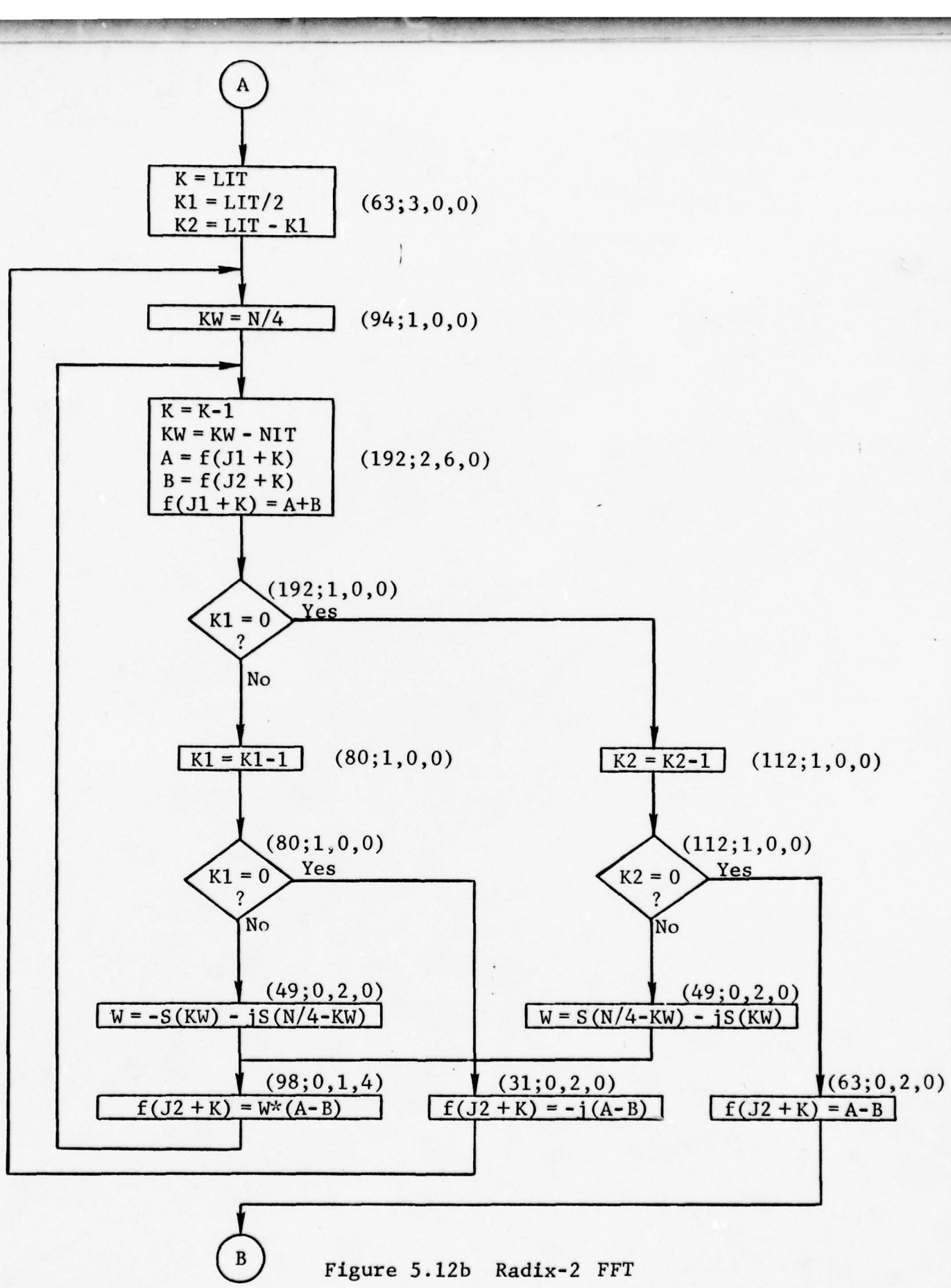


Figure 5.12b Radix-2 FFT

and the outer loop counts the six iterations necessary to compute the DFT. This algorithm requires a table of 15 samples of a sine wave. Tabulating the values in the flow chart results in an execution time of

$$t = 1509\ t_{ALU} + 1634\ t_{memory} + 392\ t_{mult}$$

5.4.5.2 Radix-4 FFT

The flow chart of Fig. 5.13a and Fig. 5.13b gives an algorithm for the computation of a 64-point DFT by the radix-4, decimation-in-frequency technique. The input and output ordering and memory allocation are the same as for the radix-2 algorithm.

The looping structure of the flow chart allows the replacement of all multiplication by 1 with addition. The remainder of the looping structure is the same as for radix-2, with minor modifications due to butterfly size. This algorithm requires a table of 64 samples of a sine wave. Tabulating the values in the flow chart results in an execution time of:

$$t = 2477\ t_{ALU} + 768\ t_{memory} + 324\ t_{mult}$$

5.4.6 Data Demodulation From Tone Phases

The 4-phase DPSK modulation planned for the ANDVT is diagrammed in Fig. 5.14 below. The solid vector r(k-1) is the tone phase measured during frame k-1. The dotted vectors indicate the possible expected tone phases during frame k, conditioned on the values of the two data bits x_1x_0. The two bits can be demodulated by examining the signs of the dot and cross products of r(k) and r(k-1). Specifically:

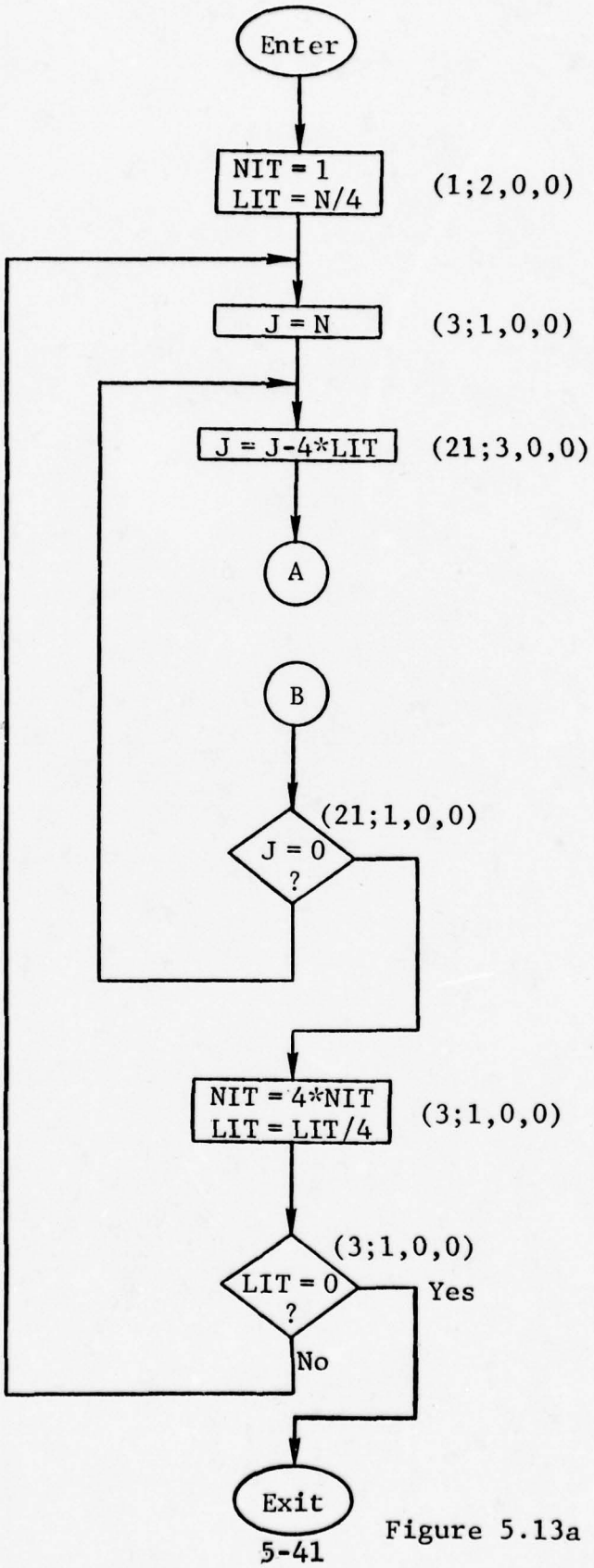


Figure 5.13a Radix-4 FFT

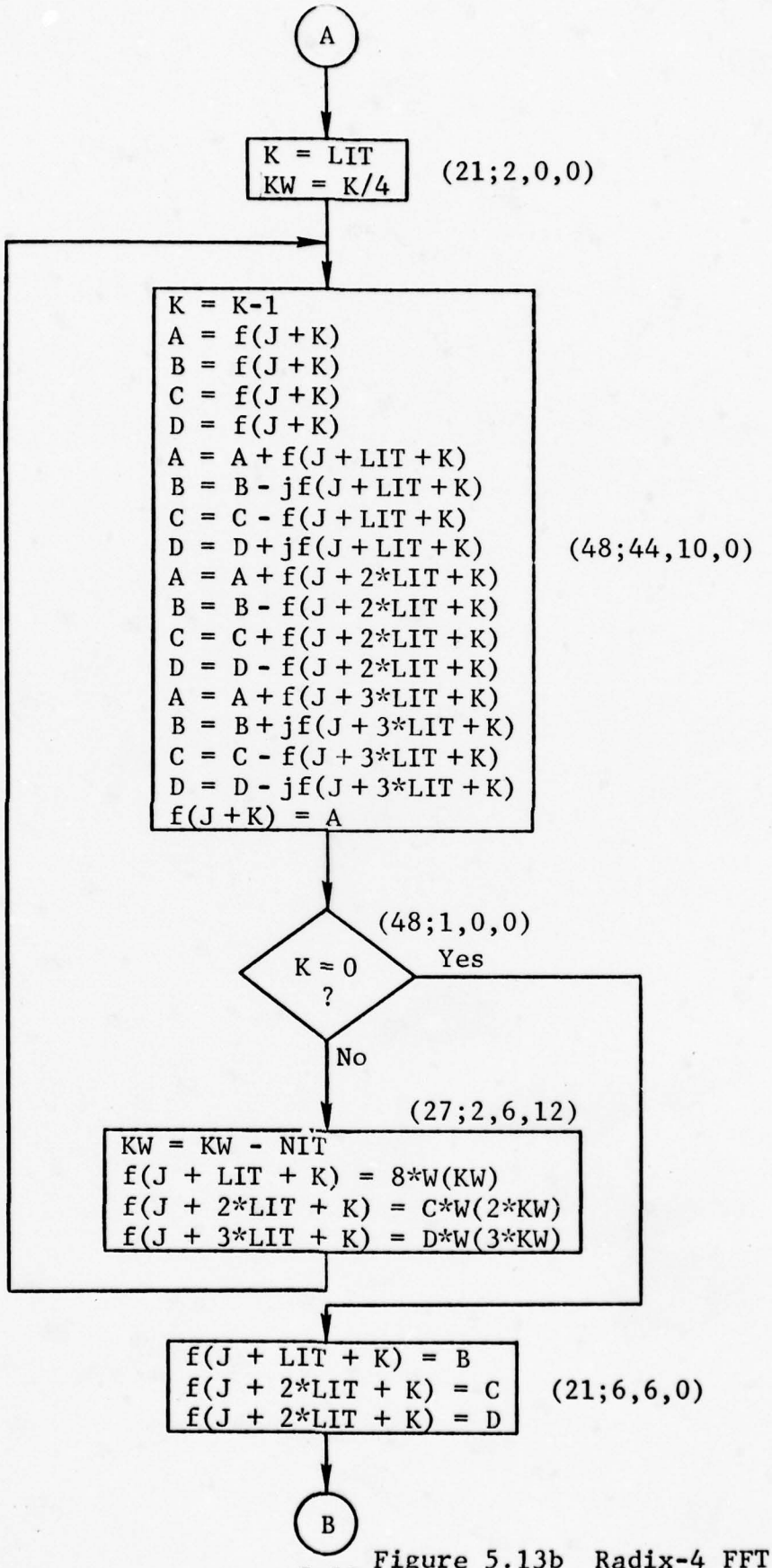


Figure 5.13b Radix-4 FFT

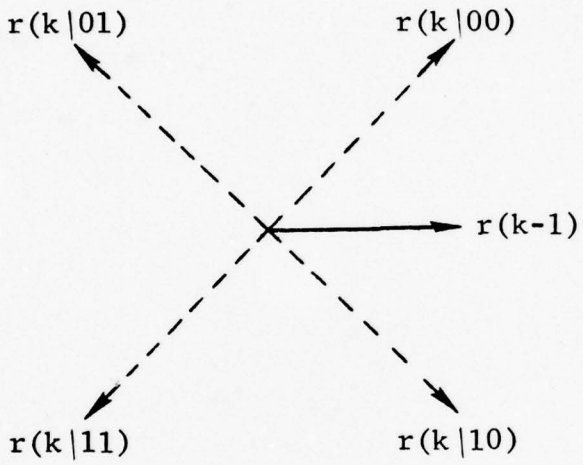


Figure 5.14 Receiver Tone Phases

$$x_1 = \text{sgn}\ (r(k-1) \times r(k))$$

$$x_0 = \text{sgn}\ (r(k-1) \cdot r(k))$$

where

$$\text{sgn}\ (x) = \begin{cases} 0 & x \geq 0 \\ 1 & x < 0 \end{cases}$$

Figure 5.15 shows a simple flow chart for demodulating the data from the tone phases. The flow chart also incorporates moving the new tone phases into the memory occupied by the old tone phases to be ready to demodulate the next frame. Each flow chart block is labelled with (R; n_{ALU}, n_{memory}, n_{mult}). For demodulation of L tone phases, the processor time required is approximately:

$$\boxed{t_{DEMOD} = (2 + 11L)\ t_{ALU} + 4L\ t_{memory} + 4L\ t_{mult}}$$

5.4.7 Doppler Tracking During Data Transmission

One method of tracking changes in the Doppler offset frequency is to include an unkeyed reference tone in the tone library. The tone is surrounded by frequency guard bands to allow the receiver to separate it from the data tones and determine its Doppler offset. This method suffers two drawbacks. First, a frequency selective fade could attenuate the reference tone and cause Doppler tracking to be inaccurate for the duration of the fade. Second, the number of tones proposed for the ANDVT may not leave sufficient bandwidth for the reference tone and its guard bands. A solution to the first drawback is to include multiple reference tones and their guard bands. However, this makes the second drawback even more critical.

Another method of tracking changes in the Doppler offset frequency uses the information present in the data tone phasors. The frame-to-frame phase change of a tone includes a component due to the data and a component proportional to the error in the

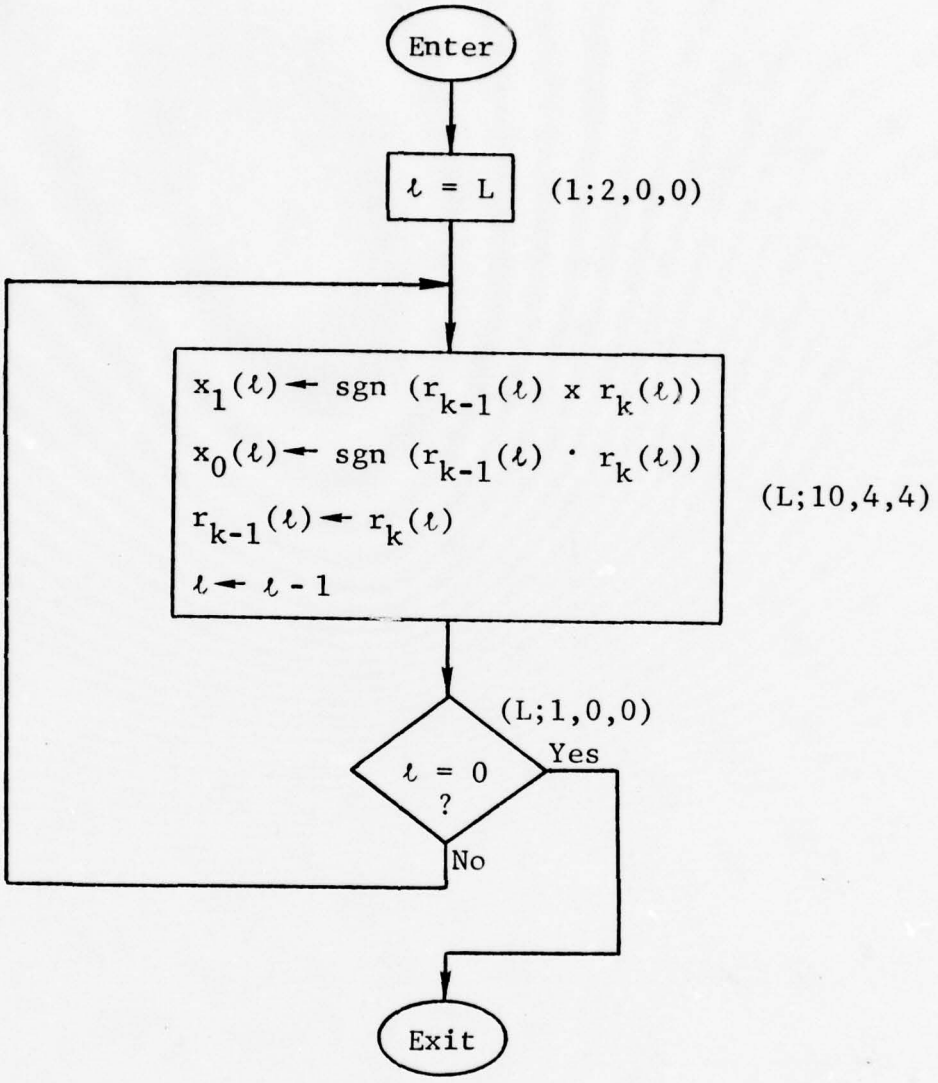


Figure 5.15 Demodulation Flow Chart

Doppler measurement. The Doppler component of the phase change is the same for all tones in a given frame and causes the tone phasors to precess from frame-to-frame. Two techniques of extracting the Doppler component are discussed below. The first, referred to as data averaging, measures the phase change on each tone and averages a sufficient number so that the data component in the average is small. The second, referred to as data feedback, uses the results of the demodulation process to cancel the data component of the phase change and to produce a measurement requiring less averaging.

Data Averaging

The frame-to-frame phase difference, at the transmitter, of a data tone can be either 135^{o}, 45^{o}, -45^{o}, or -135^{o}. If the transmitted data is uniformly distributed then each of the possible phase differences will occur with probability .25. The mean of the phase difference $\Delta\theta$ is $\overline{\Delta\theta} = 0$ and the variance is $\sigma^2_{\Delta\theta} = 10125(^{o})^2$.

If the receiver does not have a correct estimate of the Doppler then the frame-to-frame phase difference at the receiver is $\Delta\phi = \Delta\theta + \theta_D$, where θ_D is an angle proportional to the error in the receiver Doppler estimate. The mean and variance of $\Delta\phi$ are $\overline{\Delta\phi} = \theta_D$ and $\sigma^2_{\Delta\phi} = 10125(^{o})^2$. If the $\Delta\phi$ from n independent tones are averaged, the variance of the average is reduced to $\sigma^2_{\Delta\phi}/n$.

For Doppler frequency errors of 1 Hz or less, the maximum value of θ_D is 8.1^{o}. If the 3-sigma point of an average over n tones is set to 8.1^{o}, the required value of n is 1389. If there are 43 tones/frame, then the averaging must occur over 33 frames.

Since the Doppler may change at a rate of 3.5 Hz/sec, the actual Doppler error may have changed by 2.6 Hz during the measurement interval. Approximately half of the 2.6 Hz change would be indicated by the averaged measurement. It can be seen that the amount of averaging required to produce what is still a noisy estimate of the Doppler error has limited the rate of change of Doppler frequency which can be tracked by this method. If the 3-sigma point of the average is set to 1^{o} (a much less noisy measurement), then n = 91125 or 2120 frames, an impractical amount of averaging.

Data Feedback

The data feedback technique uses the result of the demodulation process to remove the data component of the tone phase change. Averaging is still desirable, however, to remove the effects of additive noise, round-off noise, tone fading, and demodulation error.

Figure 5.16 shows the four possible phasors received during frame k with an error in the Doppler estimate, relative to the received phasor from frame k - 1. The data feedback technique uses the demodulated data to rotate the phasor r(k - 1) to the position expected for r(k) if there were no Doppler error. The angle θ_D is then measured and an amplitude weighted average computed over the set of data tones. If we define:

$$A_i = \text{Re}\ \{r_i(k)r_i^*(k-1)\}$$

$$B_i = \text{Im}\ \{r_i(k)r_i^*(k-1)\}$$

$$C_i = r_i(k)r_i^*(k)$$

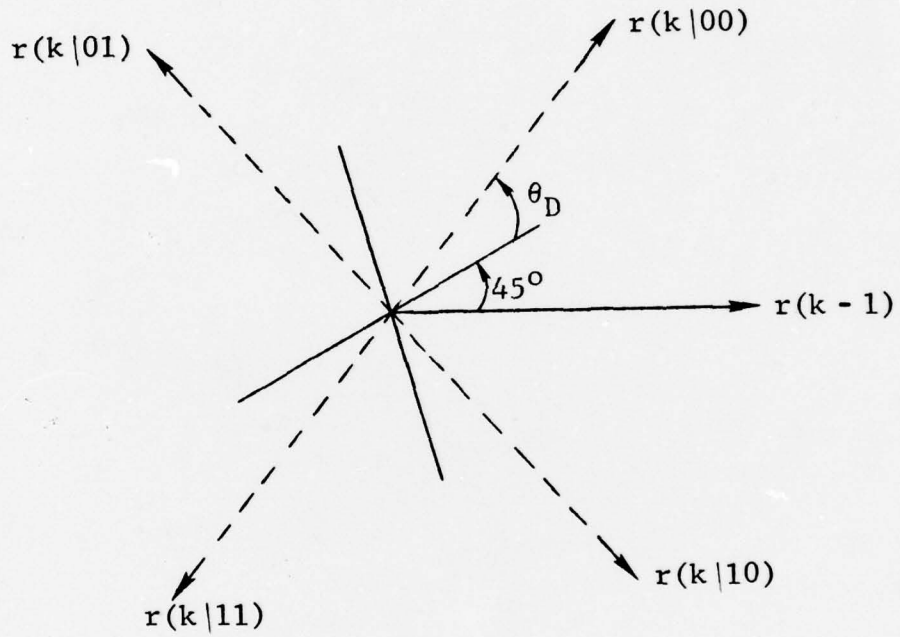


Figure 5.16 Receiver Phasors With Doppler Error

$$a_i = \begin{cases} 1 & A_i \geq 0 \\ -1 & A_i < 0 \end{cases}$$

$$b_i = \begin{cases} 1 & B_i \geq 0 \\ -1 & B_i < 0 \end{cases}$$

Then if T_f is the frame time, the Doppler error is

$$\Delta v = \frac{1}{2\sqrt{2}\,\pi T_f} \frac{\sum_i \{a_i B_i - b_i A_i\}}{\sum_i C_i}$$

It should be noted that the derivation of (6) involved the approximation of a derivative by

$$\dot{x}(k) = \frac{x(k) - x(k-1)}{T}$$

An interesting feature of the estimator defined in (6) is that the A_i and B_i are the decision variables used to demodulate the data. The notation used here differs from that of Section 5.4.6 so the comparisons for the demodulator are restated as:

$$x_{0i} = \begin{cases} 0 & A_i \geq 0 \\ 1 & A_i < 0 \end{cases}$$

$$x_{1i} = \begin{cases} 0 & B_i \geq 0 \\ 1 & B_i < 0 \end{cases}$$

Since the quantities in the numerator of (6) are computed in the course of demodulation, the additional computations required to complete the estimator are simply those necessary to perform the summations, compute the terms in the denominator, and perform the division. An estimate of this is (104 + 8L) ALU operations, 4 memory operations, and 2L multiply operations for L data tones. The amount of processor time in excess of the demodulation time is

$$\boxed{t_{track} = (104 + 8L)\, t_{ALU} + 4t_{memory} + 2Lt_{mult}}$$

5.4.8 Slot Sync

To maintain a proper frame sync position in the presence of clock drift and changes in the multipath environment, it is necessary to track the position of the phase keying and adjust the receiver clock to keep it within the guard time. This serves to ensure that the keyed tones are orthogonal during the integration time. One method of accomplishing this is slot sync. If an integration gate covers a phase keying point, the phase discontinuity causes crosstalk between the tones. The slot sync technique measures the energy in an empty frequency slot of an early integration time and also measures the energy in the same frequency slot of a late integration time. The difference of these energies provides a discriminator function to indicate the centering of the data integration time within the frame time.

This technique requires the measurement of a tone amplitude in each of two integration times. The FFT is not efficient in this application because the amplitude of only one tone is desired per set of samples. An efficient method of obtaining a small number of tone amplitudes is the Goertzel algorithm for the DFT. The Goertzel algorithm has two advantages over the direct form of the DFT. First, the number of real multiply operations is half that of the direct DFT, and, second, only a single coefficient is required rather than the N required by the direct DFT. This saves table look-up time during program execution.

Algorithm Description

The Goertzel algorithm is a digital filter approach to obtaining the tone phasor. Figure 5.17 shows a block diagram of the filter used. The N samples are input to the filter and the output at t = NT is y(NT) = F(kΩ), the value of the DFT for the k'th tone. Since the output of the filter is needed only at n = N, the operations to the right of the dotted line need only be performed for that sample time.

Figure 5.18 shows a flow chart implementing the Goertzel algorithm with N = 64. The blocks are labelled with (R; n_{ALU}, n_{memory}, n_{mult}). The number of operations to execute the flow chart once is 646 ALU operations and 132 multiply operations. This must be done twice for slot sync and a few additional finishing operations to implement the discriminator, approximately 4 memory operations and 6 multiply operations. The total time for the slot sync operation is given approximately by:

$$t_{slot} = 1292\ t_{ALU} + 4\ t_{memory} + 270\ t_{mult}$$

5.4.9 Golay Soft-Decoder

This section describes an implementation of a soft-decision decoder for the (24,12) Golay code with overall parity. This decoder selects the four least reliable bits in the code word and performs a hard-decode operation for each of the 16 trial error patterns generated by these four bits. The decoded word with the lowest analog weight is selected as the output of the

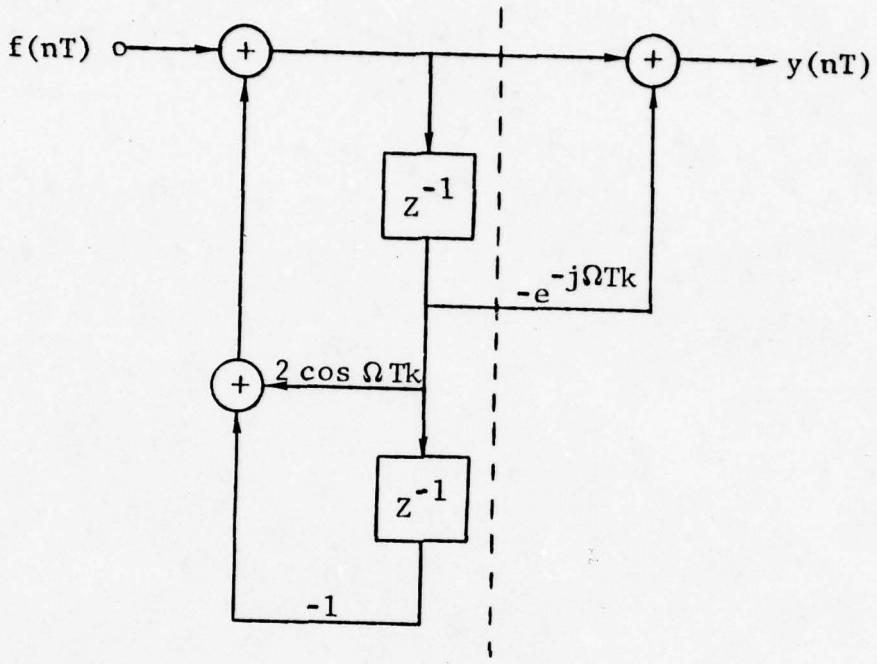


Figure 5.17 Digital Filter for Goertzel Algorithm

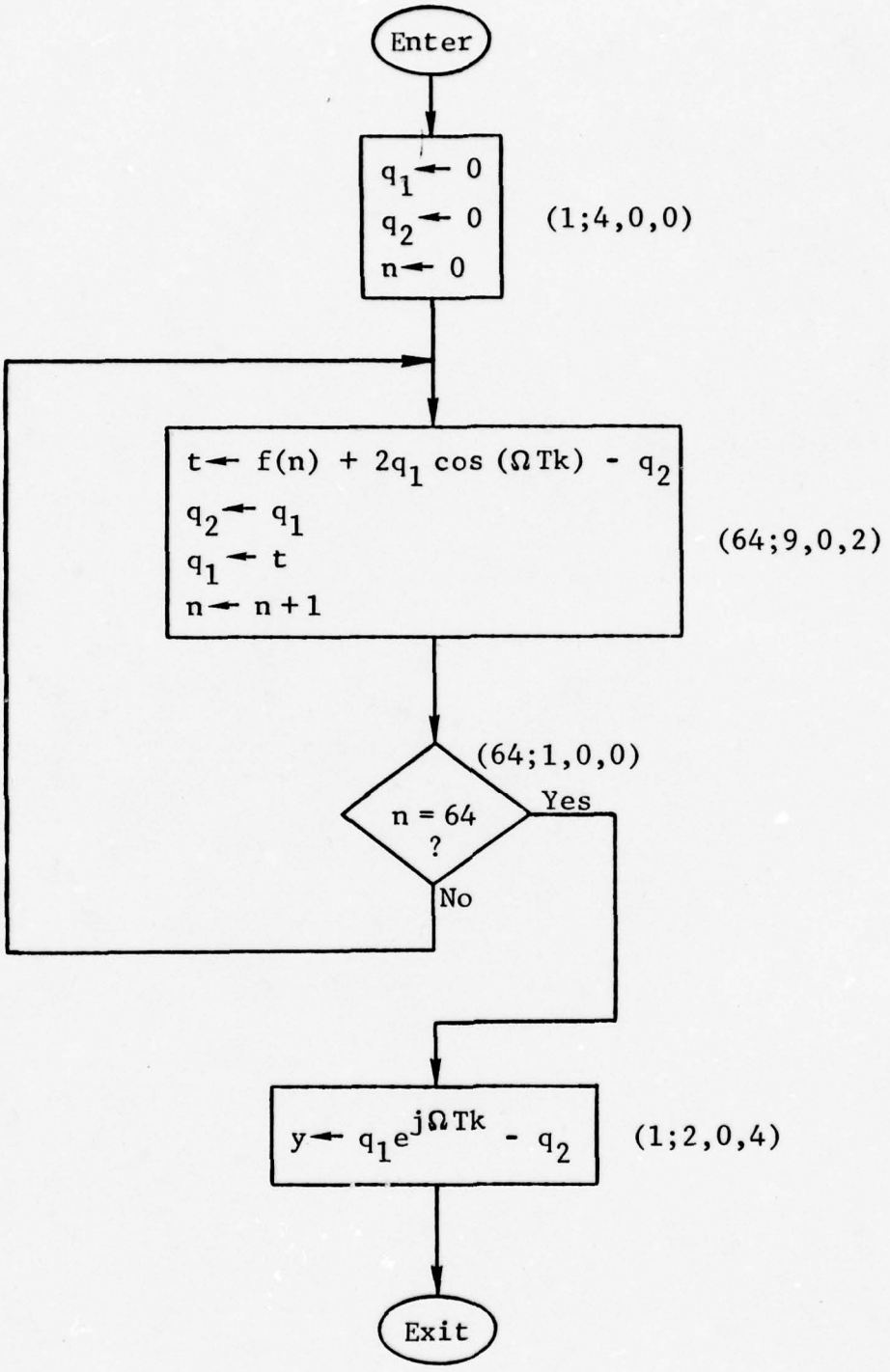


Figure 5.18 Goertzel Algorithm Flow Chart

decoder. The absolute values of the decision statistics defined in the first two equations of Section 5.4.7 are used in the selection of the least reliable bits and in the computation of the analog weights.

The hard-decoding algorithm chosen is a table lookup technique. The syndrome addresses a 1-bit wide table whose content indicates the presence or absence of an error in the rightmost bit position. The syndrome is successively modified to correspond to a 1-bit cyclic shift of the code word and the table indicates the presence or absence of an error. This is performed 23 times to check the validity of the entire Golay code word. The parity of the decoded word is used to check the validity of the overall parity bit. The modified error-trapping procedure of Kasami was compared to the table lookup method and found to require either a much larger table or to require significantly more processor time.

Figure 5.19 shows a flow chart of the soft-decoding process. Four arrays of 24 words and two lookup tables are used in this implementation. The four arrays are the received bit array, the weight array, the error pattern array, and the old error pattern array. The received bit array and the weight array are set by the demodulation routines. The error pattern array holds the error pattern when each of the trial error patterns is decoded. The old error pattern array holds the error pattern which resulted in the lowest weight. One of the lookup tables is used by the hard decoder to determine the presence of an error in the rightmost bit position. The remaining lookup

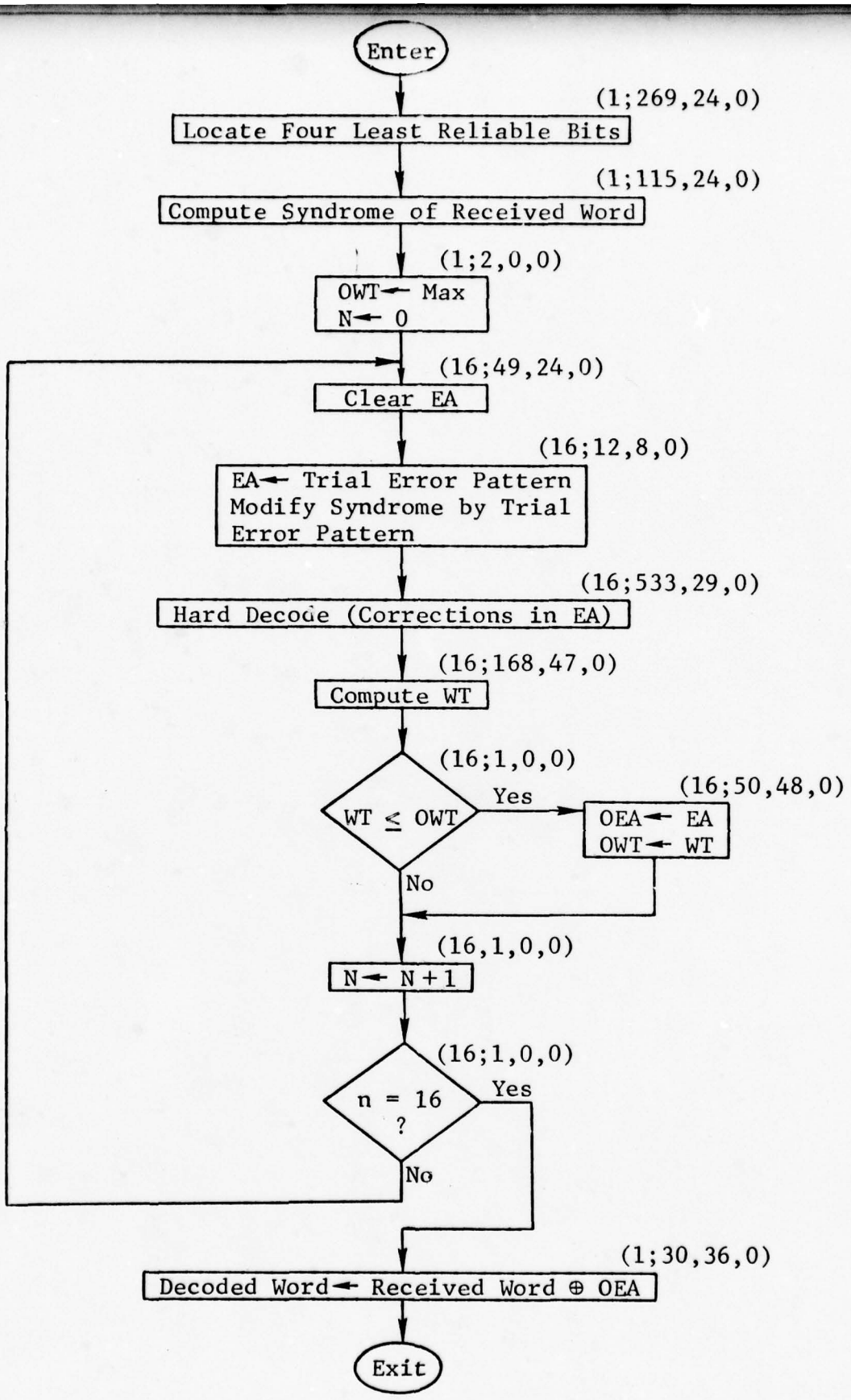


Figure 5.19 Soft Decoding Flow Chart

table contains the syndrome modifications used to compute the starting syndrome from the received syndrome given the trial error pattern.

The flow chart starts by locating the four least reliable bits, computing the syndrome of the received word, and setting the variable which holds the weight of the best error pattern (OWT) to a maximum (insuring that the first hard decode is stored in the best error pattern array (OEA)). The loop counter N is also set to zero. Each pass through the loop causes a new trial error pattern to be generated, and a hard decode performed using it. The weight (WT) of the error pattern is compared to OWT to determine if the present contents of the EA should replace the contents of OEA. Following the sixteenth pass through the loop, the decoded word is taken as the bit-by-bit exclusive-OR of the received word with OEA.

Each block in Fig. 5.19 is labelled to indicate the number of times the block is executed, the number of ALU operations, the number of memory operations, and the number of multiply operations, respectively. Tabulating these numbers results in a count of 13456 ALU operations and 2580 memory operations. The time required to execute this operation is

$$t_{SD} = 13456\ t_{ALU} + 2580\ t_{memory}$$

Various stopping rules may be employed to speed up the above processing time. These stopping rules, which yield no loss in performance, may be used in the actual implementation, if required.

5.4.10 BCH Decoder for KG Sequence

The cryptographic key and net control sequences are protected by a (249,101;45) BCH code.* The single code word has a total of 249 bits, 101 of which are information bits containing the cryptographic key and net control sequences. The minimum distance of the code is 45, allowing 22 or fewer errors to be corrected. A flow chart for the decoding of this code is given in Fig. 5.20. The decoding operation consists of three main steps: syndrome calculation, calculation of the error location polynomial $\sigma(X)$, and the Chien search to locate the errors. If the order of $\sigma(X)$ is ≤ 22 and the number of errors found in the Chien search equals the order of $\sigma(X)$ then the decoded output $V(X)$ is the modulo-2 sum of the received sequence $R(X)$ and the error sequence $E(X)$ from the Chien search. Otherwise we take $R(X)$ as the decoded output.

The decoding process begins with the computation of the syndromes S_i, $i = 1,2,3, \ldots, 44$. The odd syndromes are computed as:

$$S_i = \sum_{j=0}^{248} r_j \alpha^{ij}$$

The even syndromes can be determined more easily by the relation:

$$S_{2i} = (S_i)^2$$

The terms α^{ij} are field elements in the Galois field $GF(2^8)$.

A flow chart for the calculation of the syndromes is given in Figs. 5.21a and 5.21b. This algorithm uses a table look-up to find the α^{ij} used in the summation for the odd syndromes. Figure

*The complexity of this code is equivalent to that of the (250,100;46) code considered in Section 3, and the performance is essentially equivalent. The above code may be selected in place of the (250,100;46) code.

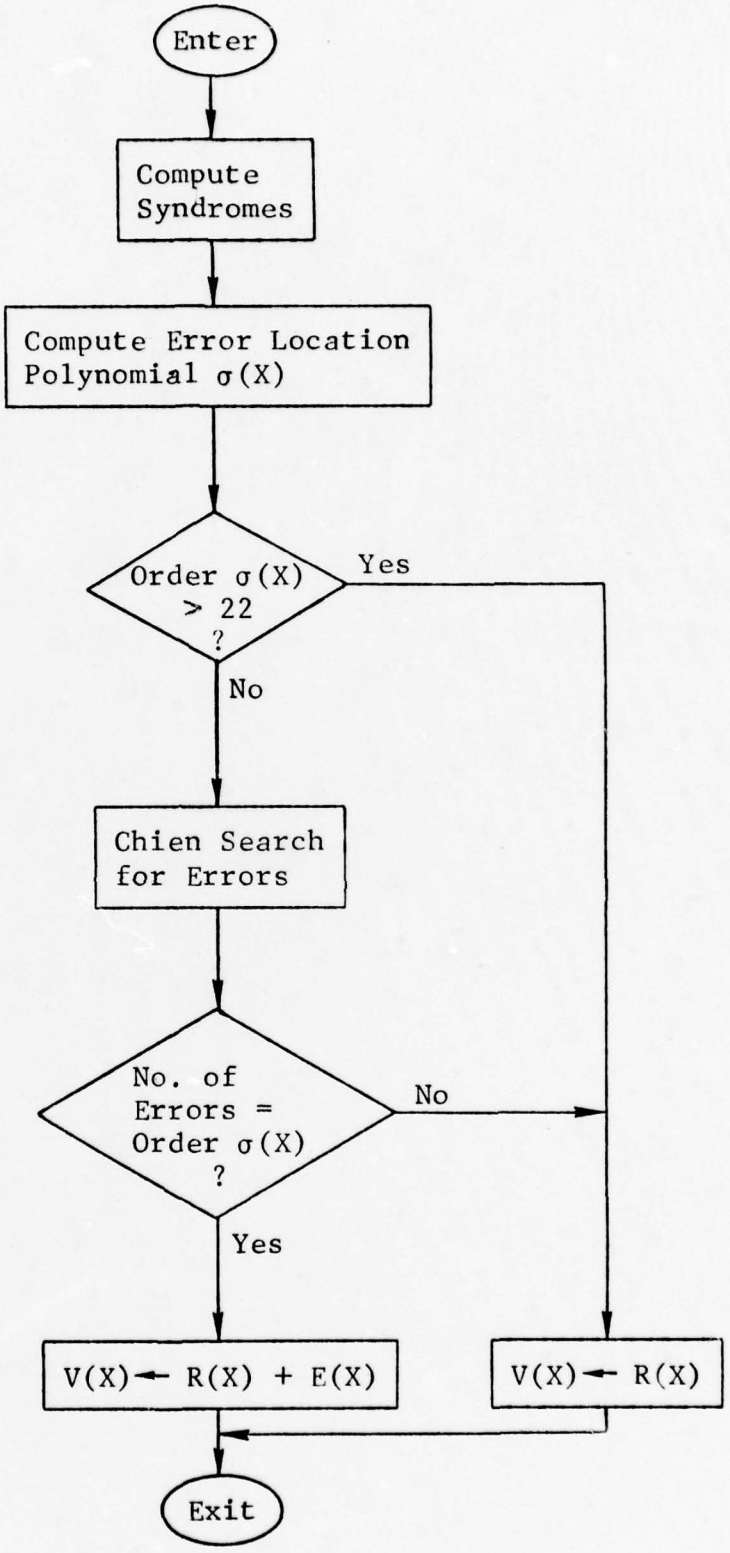


Figure 5.20 BCH Decoding Algorithm

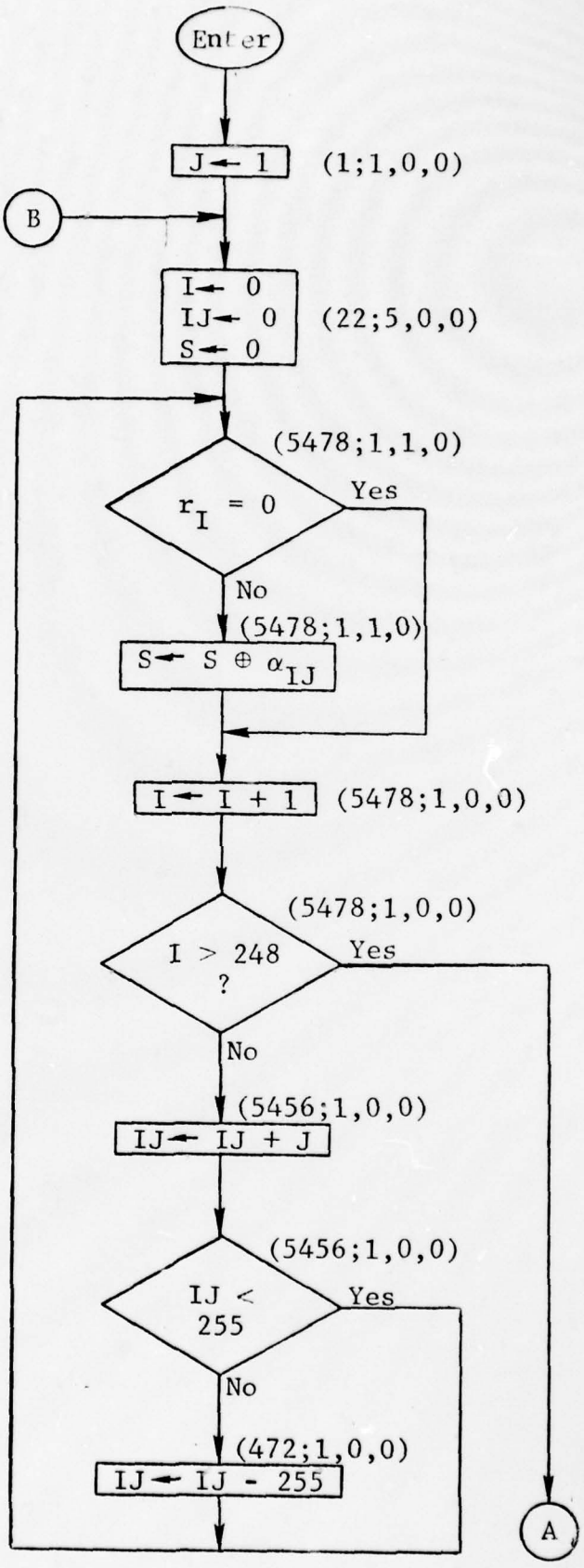


Figure 5.21a Syndrome Computation

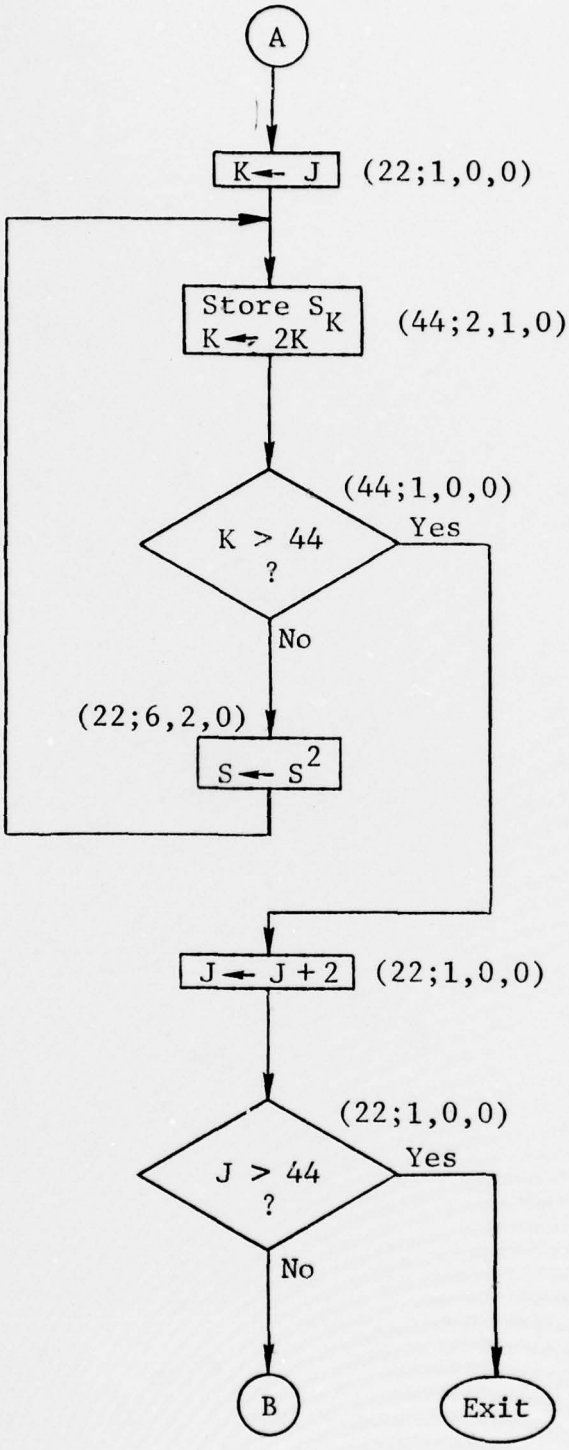


Figure 5.21b Syndrome Computation

5.21b is the part which stores the syndromes in memory and computes the required even syndromes. Each block is labelled with $(R; n_{ALU}, n_{mem}, n_{mult})$. Tabulating these numbers results in a count of 33737 ALU operations and 11044 memory operations for the syndrome calculations.

The second stage of the decoding procedure is the calculation of the error locating polynomial $\sigma(X)$. The roots of $\sigma(X)$ give the locations of the errors. $\sigma(X)$ is found by an iterative procedure using the syndromes. A flow chart of this iterative procedure is given in Fig. 5.22. This implementation assumes a specific memory layout for the polynomials $\sigma(X)$, $\sigma'(X)$, and $\tau(X)$. For $\sigma(X)$ this is: $0\ 1\ \sigma_1\ \sigma_2\ \sigma_3\ \ldots\ \sigma_{43}$ in consecutive memory locations. For $\sigma'(X)$ this is $\sigma'_1\ \sigma'_2\ \sigma'_3\ \ldots\ \sigma'_{43}$ in consecutive memory locations. For $\tau(X)$ this is $0\ 0\ \tau_0\ \tau_1\ \tau_2\ \ldots\ \tau_{42}$ in consecutive memory locations. These layouts are necessary to keep the loop structures simple within the various blocks of Fig. 5.22.

The boxes in Fig. 5.22 are labelled with the 3-tuple $(n_{ALU}, n_{mem}, n_{mult})$ where each is the total number of ALU operations, memory operations, or multiplication operations, respectively, that is required in the box for the entire execution of the flow chart. This has been done because the amount of effort in many boxes varies with the index k and the usual notation is inappropriate. Summing all of these entries results in 19374 ALU operations and 8343 memory operations for the determination of the error locating polynomial.

The last significant amount of computation in the error correction procedure is the Chien search. The Chien search computes the value of $\sigma(X)$ for each α^i corresponding to a bit in the received code word. $\sigma(\alpha^i) = 0$ indicates that an error has been

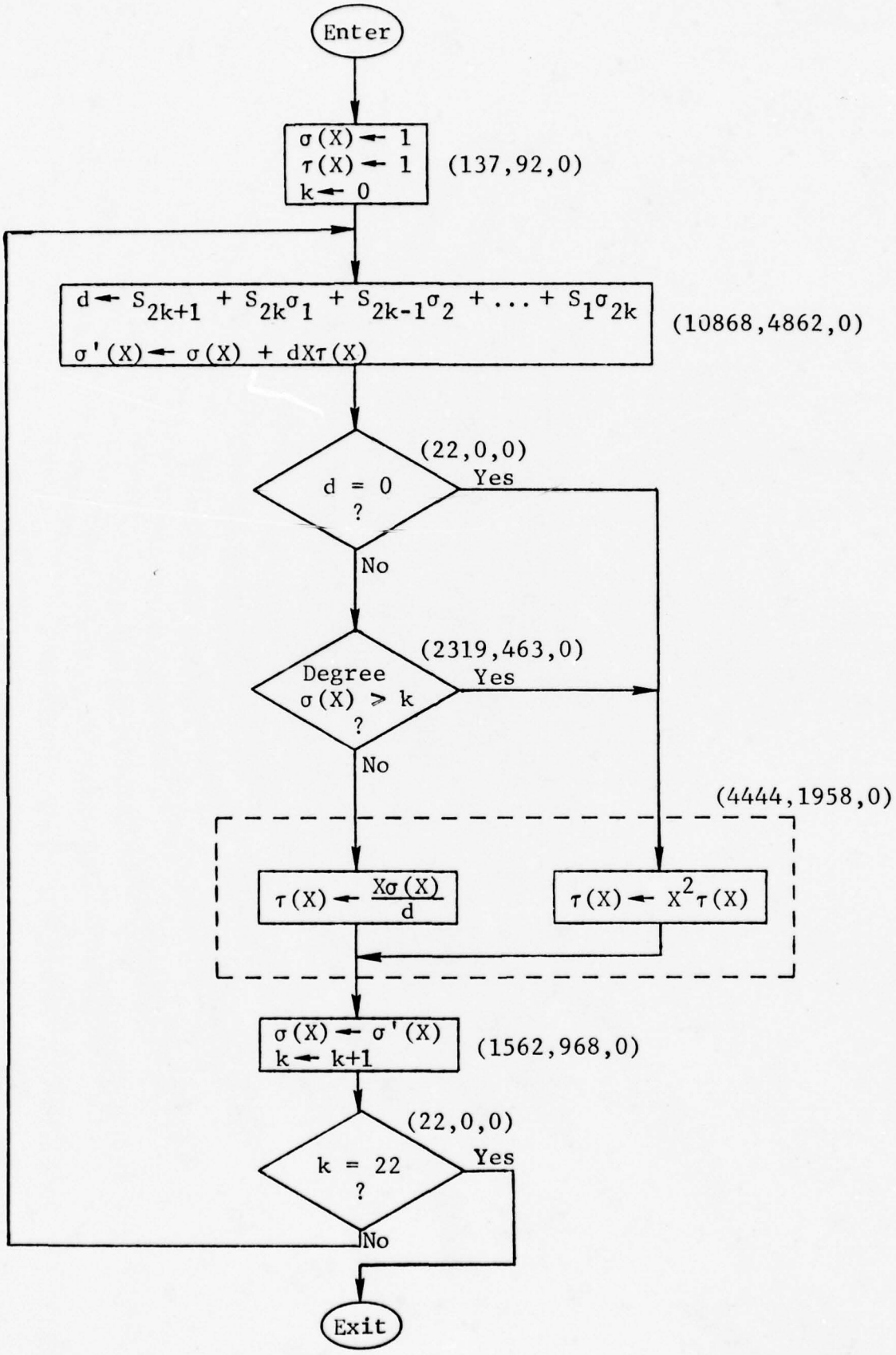


Figure 5.22 Iterative Procedure for σ(X)

found. The flow chart for this operation is given in Figs. 5.23a and 5.23b. The loop of Fig. 5.23a computes the log of each coefficient of $\sigma(X)$ and performs a correction due to the fact that the (249,101;45) code is a shortened (255,107;45) code. The correction simply serves to advance the Chien search past the missing bits. Figure 5.23b performs the actual search by advancing each term of the polynomial by the correct power of α and summing the non-zero terms. If the term $\Sigma = \sigma(\alpha^i)$ is zero then a 1 is entered in the error array and the error count is incremented, otherwise a 0 is entered in the error array. The blocks in the flow chart are labelled with $(R; n_{ALU}, n_{mem}, n_{mult})$, tabulating these results in 51726 ALU operations and 16749 memory operations.

The total of the operation counts for the three major portions of the decoding is 104837 ALU operations and 36136 memory operations.

5.5 Demodulator Processor Time Summary

Table 5-4 presents a summary of the study of the implementation of the demodulator on a programmable processor. The tasks listed are the algorithms described in the subsections of Section 5.4. The formulas given for signal presence detection and for Doppler estimation include all operations. To find the processor time required per sample for frame sync it is necessary to add the times for Doppler correction and for the frame sync mathched filter. The time for complete data demodulation is found by summing the total Doppler correction time, the 64-point FFT time, the demodulation time, the Doppler tracking time and the slot sync time. The total Doppler correction time is the per sample Doppler correction time times the number of samples in a frame. If two Golay code words are included then twice the Golay soft-decode time must be added.

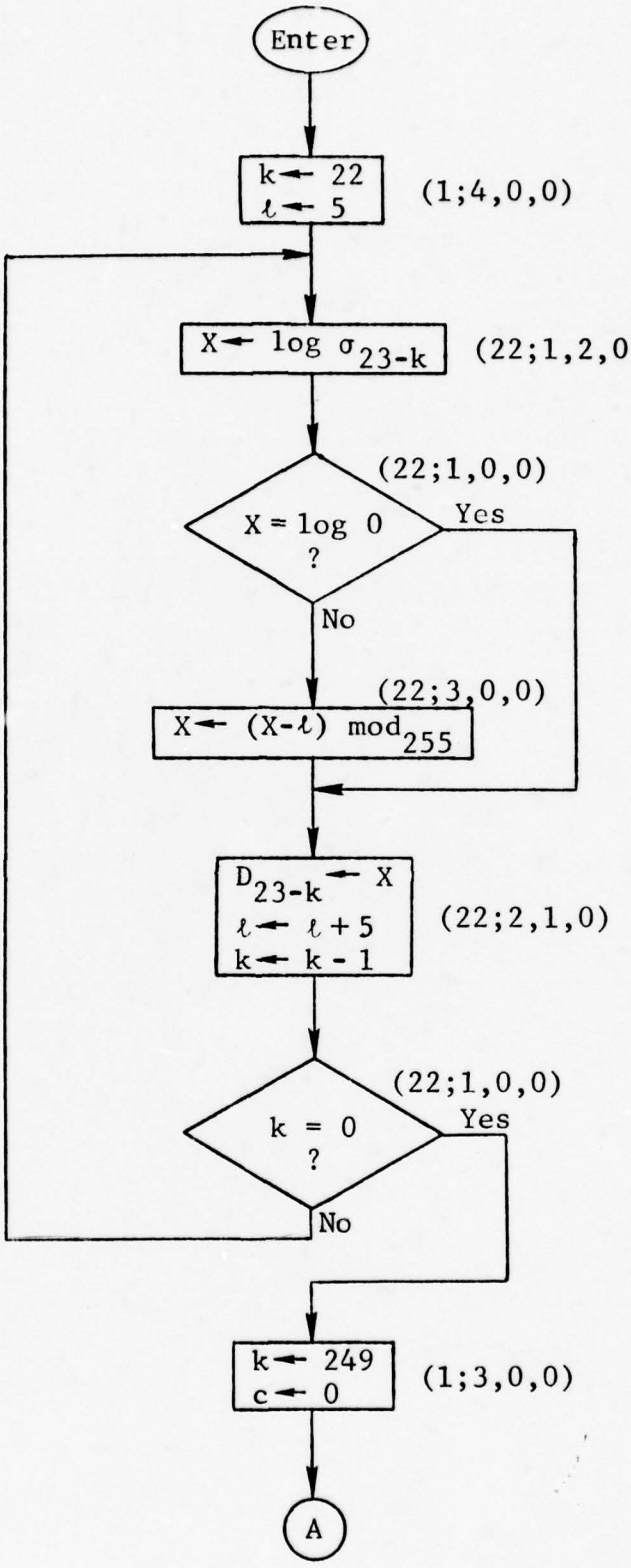


Figure 5.23a Chien Search

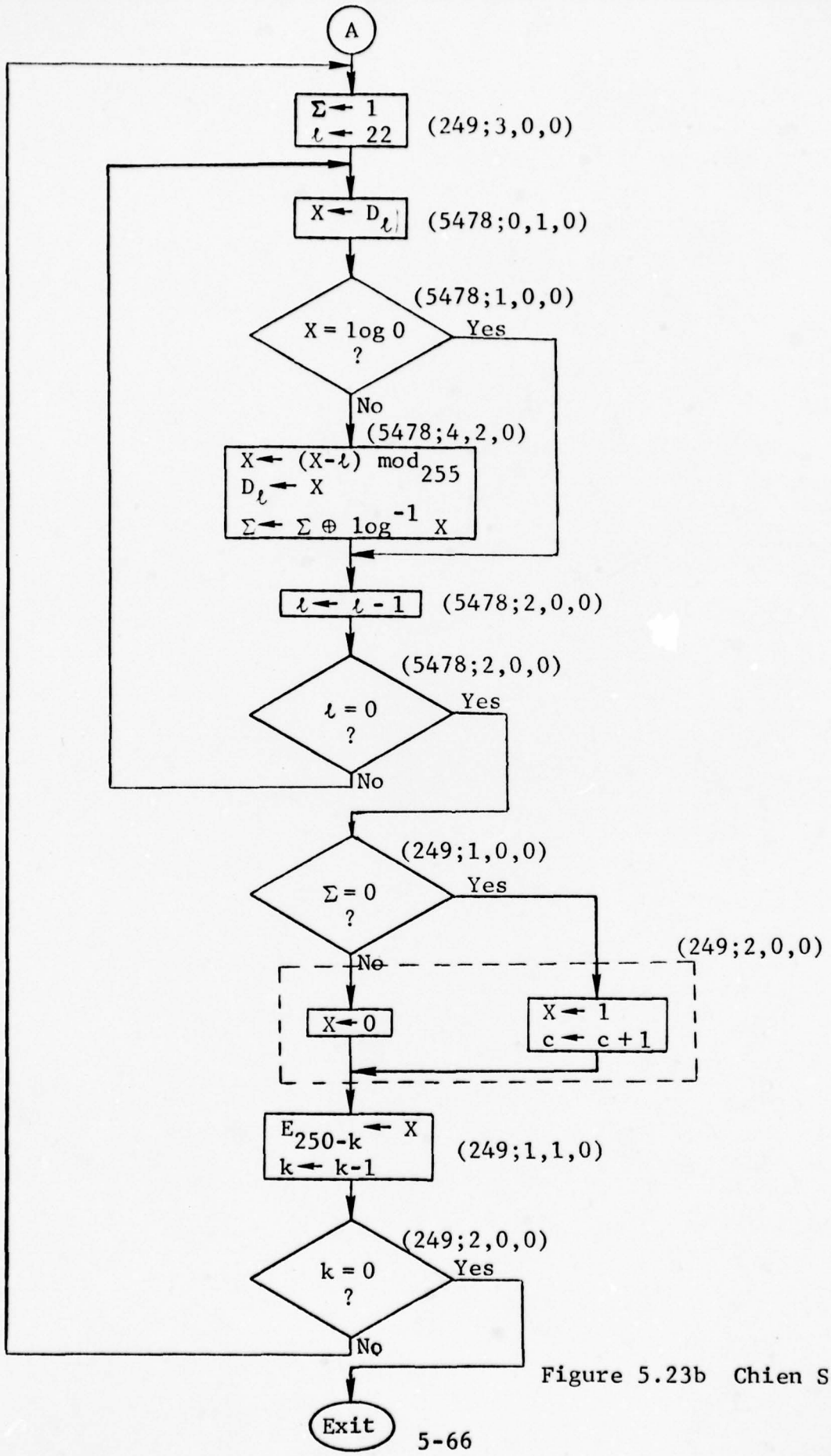


Figure 5.23b Chien Search

TABLE 5-4

DEMODULATOR IMPLEMENTATION SUMMARY

TASK	PROCESSOR TIME
SIGNAL PRESENCE DETECTION (PER SAMPLE)	$\left(56 + \frac{50}{M}\right)t_{ALU} + \left(48 + \frac{10}{M}\right)t_{mem} + \left(64 + \frac{16}{M}\right)t_{mult}$
DOPPLER ESTIMATION (PER SAMPLE)	$\left(28 + \frac{146}{P}\right)t_{ALU} + \left(24 + \frac{28}{P}\right)t_{mem} + \left(32 + \frac{32}{P}\right)t_{mult}$
DOPPLER CORRECTION (PER SAMPLE)	$27t_{ALU} + 7t_{mem} + 8t_{mult}$
FRAME SYNC MATCHED FILTER (PER SAMPLE)	$164t_{ALU} + 60t_{mem} + 3t_{mult}$
64-POINT FFT (TOTAL)	$1509t_{ALU} + 1634t_{mem} + 392t_{mult}$
DEMODULATION (TOTAL)	$(4+11L)t_{ALU} + 4L\,t_{mem} + 4L\,t_{mult}$
DOPPLER TRACKING (TOTAL)	$(104+8L)t_{ALU} + 4t_{mem} + 2L\,t_{mult}$
SLOT SYNC (TOTAL)	$1292t_{ALU} + 4t_{mem} + 270t_{mult}$
16 ITERATION (24,12) GOLAY SOFT DECODE (TOTAL)	$13456t_{ALU} + 2580t_{mem}$
BCH DECODER FOR KG SEQUENCE (TOTAL)	$104837t_{ALU} + 36136t_{mem}$

The memory requirements for the modem have been examined briefly. A data memory of .5 K of ROM and 1 K of RAM is sufficient. For the processor model of Section 5.3 a 2 K microprogram memory will be adequate.

Table 5-5 lists the assumptions made and presents an example of the use of the formulas in Table 5-4. The assumptions relating to the processor time are believed to represent the TRW Quintrell processor. This table indicates that all demodulator functions can be accomplished within the speed constraints of the TRW processor.

TABLE 5-5

PROCESSOR UTILIZATION EXAMPLE

Assume:

- 3600 Hz Sampling
- 39 Data Tones
- t_{ALU} = 350 ns
- t_{memory} = 350 ns
- t_{mult} = 700 ns
- M = P = 16

Signal Presence Detection:	8.3×10^{-5} Sec/Sample 30% Processor Utilization
Doppler Estimation:	4.6×10^{-5} Sec/Sample 16% Processor Utilization
Frame Sync: (Doppler Correction and Frame Sync Matched Filter)	9.8×10^{-5} Sec/Sample 35% Processor Utilization
Data Demodulation: (Doppler Correction + 64-Point FFT + Demodulation + Doppler Tracking + Slot Sync)	4.0×10^{-3} Sec/Frame 18% Processor Utilization
Data Demodulation and Decoding: (Data Demodulation + 2 Golay Soft Decoders)	1.5×10^{-2} Sec/Frame 67% Processor Utilization
BCH Decoder for KG Sequence	5.0×10^{-2} Sec

SECTION 6

ADDITIONAL MODEM DESIGN CONSIDERATIONS

6.1 Introduction

In this section several topics which are related to the ANDVT modem are discussed.

Section 6.2 outlines design considerations for optimizing the overall waveform, with particular emphasis on minimizing the effects of the high peak-to-average power ratio present in multiple tone HF modems.

Considerations for providing the ANDVT modem with a data transmission mode of operation are discussed in Section 6.2.

Finally, in Section 6.3 the application of coding for performance monitoring is discussed.

6.2 Signal Level Optimization

For an N-tone HF modem with phase shift keying the transmitted signal is given by

$$x(t) = \sum_{i=1}^{N} A \cos (\omega_i t + \theta_i) \qquad (6.1)$$

where θ_i represents the information and ω_i represents the frequency of the i'th tone which is assumed orthogonal over an integration period of T second. The peak value of this signal is NA and the rms value of $\sqrt{N}$ A. Thus the peak-to-rms ratio in terms of N is given by

$$\text{peak-to-rms ratio} = 20 \log \sqrt{N} \text{ dB} \qquad (6.2)$$

which for a 40 tone modem is 16 dB. If one designs the transmitted signal to be distortion free, regardless of the particular phase relationships between the individual tones, the transmitter must operate at a mean value which is 16 dB below its peak value. Clearly, this is an undesirable mode of operation.

The problem is further compounded by the preamble design (see Section 2) which allows for a peak-to-rms ratio of less than 5 dB. Note that during signal presence detection and Doppler estimation, the four tone preamble can be designed with a crest factor of 1.77 (2.48 dB) using 180^{o} phase increments and using optimum phase adjustemtns a crest factor of 1.52 (1.84 dB) can be obtained. During frame estimation, using raised cosine pulses, a peak-to-rms value of 2.67 (4.26 dB) is obtained. If cosine shaped pulses are used, a peak-to-rms value of only 3 dB results. It is desirable to introduce some intentional clipping in the transmitted signal during the occurrence of the multiple tone signal to maximize the transmitted power. (Note that HF radios with some peak power limitations are assumed.) In conventional modems the received signal is intentionally clipped to maximize the dynamic range and thus, the clipping of the transmitted signal should be somewhat less than that required by the receiver.

A heuristic discussion of these issues, based on a combination of theoretical arguments and experimental results, is given in the Codem I final report [6.1]. Following these arguments one may conclude that several dB of intentional clipping at the transmitter is desirable. Since the preamble can operate at a peak-to-rms ratio well below 5 dB, it would be desirable to operate the 40 tone modem with at least 11 dB of intentional clipping at the transmitted signal. The actual level of acceptable clipping (which is expected to be somewhat less than 11 dB) can be determined

experimentally during the expected follow-on to this present contract. The clipping nonlinearity at the transmitter, and the effects of this nonlinearity when digitized speech is transmitted over a HF channel model, make this problem quite intractable mathematically. Fortunately this problem is tractable by detailed simulations which include the entire HF modem, the HF channel model, and the speech digitizer.

6.3 A Data Transmission Mode of Operation for ANDVT

It is desirable to have a data transmission mode for data rates of 2400, 1200, and 600 b/s, in addition to the speech transmission mode of operation. While a frame rate of 44.44 f/s (frame of 22.5 ms duration) is desirable when speech is encoded at this frame rate, the question arrises as to what frame rate should be used for data transmission. Conventional Tadil-A modems have a 16 tone four-phase DPSK format, and a frame rate of 75 f/s. Thus a data rate of 2400 b/s is achieved before coding is attempted. If a 40-tone four-phase DPSK modem format and a frame rate of 44.44 fps is used, a maximum data rate of 3555.55 b/s can be achieved. This format enables us to consider coding even at a data rate of 2400 b/s. The possibility exists that once a code is chosen for the digitized speech signal the same code may be modified for data transmission applications. However, at this point a final decision on the code for ANDVT has not been made. The remainder of this section presents preliminary trade-off results for convolutional codes and the (24,12) block codes used with four-phase DPSK modems at frame rates of 75 f/s and 44.44 f/s.

Hard and soft decoding is considered for both the block codes and convolutional codes. The soft decision performance is obtained by a somewhat optimistic estimate obtained by assuming a factor of two increase in the number of errors that can be corrected [6.2]. Since decoding delay is constrained only by complexity, code bits can be made independent with sufficient interleaving, which will be assumed. Figure 6.1 illustrates the benefits of coding, in particular soft decoding, for a four-phase DPSK modulation format. The Doppler variations are assumed negligible on this figure and thus the performance is the same for 75 f/s and 44.44 f/s modems. Note that the bit error probability is illustrated in terms of the signal-to-noise ratio per tone (E_t/N_0) which is closely related to the signal-to-noise ratio per bit (E_b/N_0), (see Appendix C). In terms of E_b/N_0 the 44.44 f/s modem has a slight advantage (1.67 dB), since the guard time ratio is reduced and no extra tone is used for Doppler tracking.

Figures 6.2 through 6.6 illustrate the effects of Doppler variations for a two-sided rms Doppler spread $B = 2\sigma$ of 1, 2, 3, 4, and 5 Hz, when a frame rate of 44.44 f/s is utilized. While Doppler spreads are typically below 2 Hz, the importance of coding for operating in the presence of Doppler is evident from these figures.

Figures 6.7 through 6.11 illustrate the same curves when a modem with a frame rate of 75 f/s is assumed. This modem is less sensitive to Doppler spreads. However, when soft decoding is used and $B \leq 2$ Hz, the performance difference between a 75 f/s and 44.44 f/s modem is negligible.

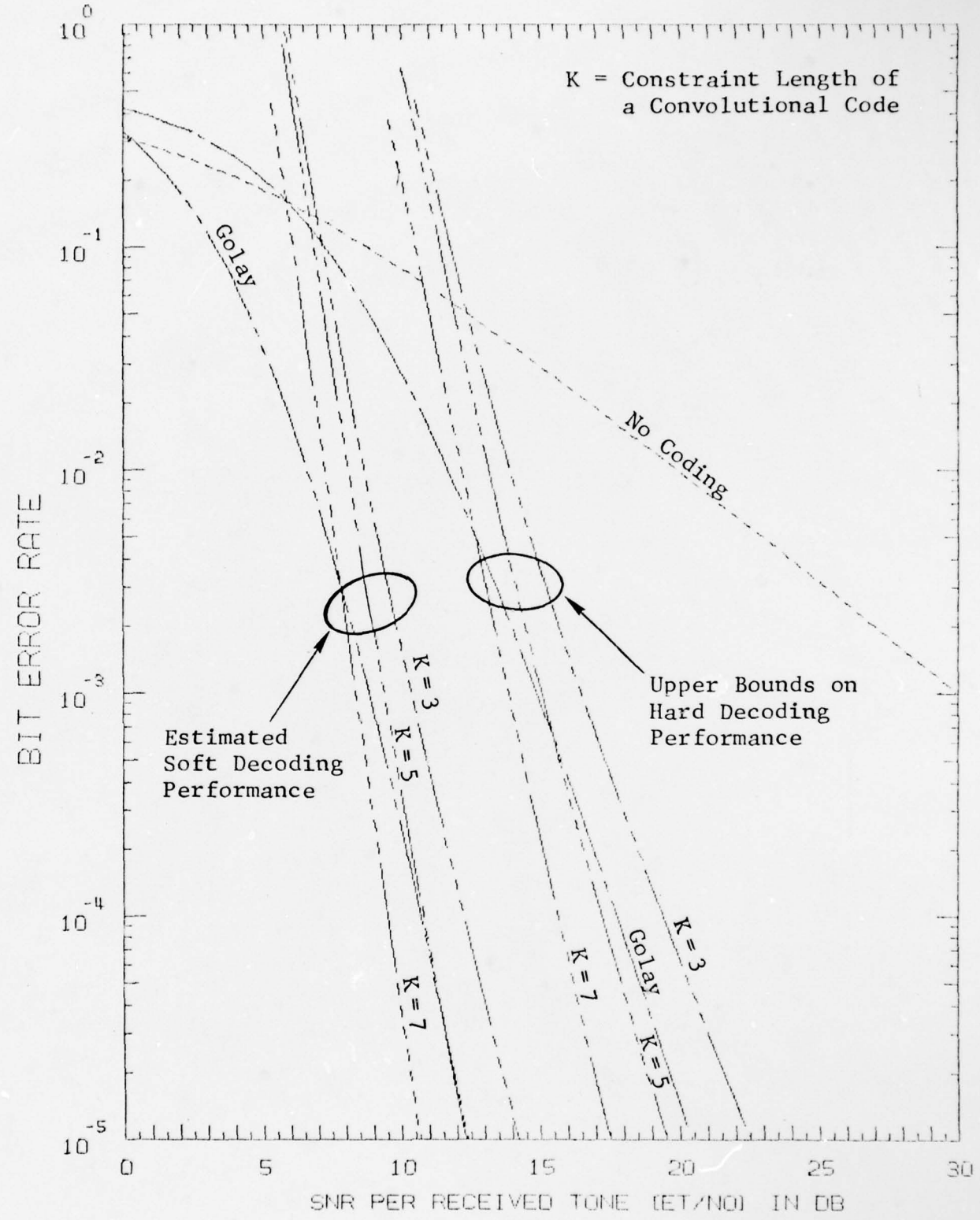


Figure 6.1 Performance of Rate-1/2 Codes Over a Slowly Varying Rayleigh Channel With 4-Phase DPSK Modulation

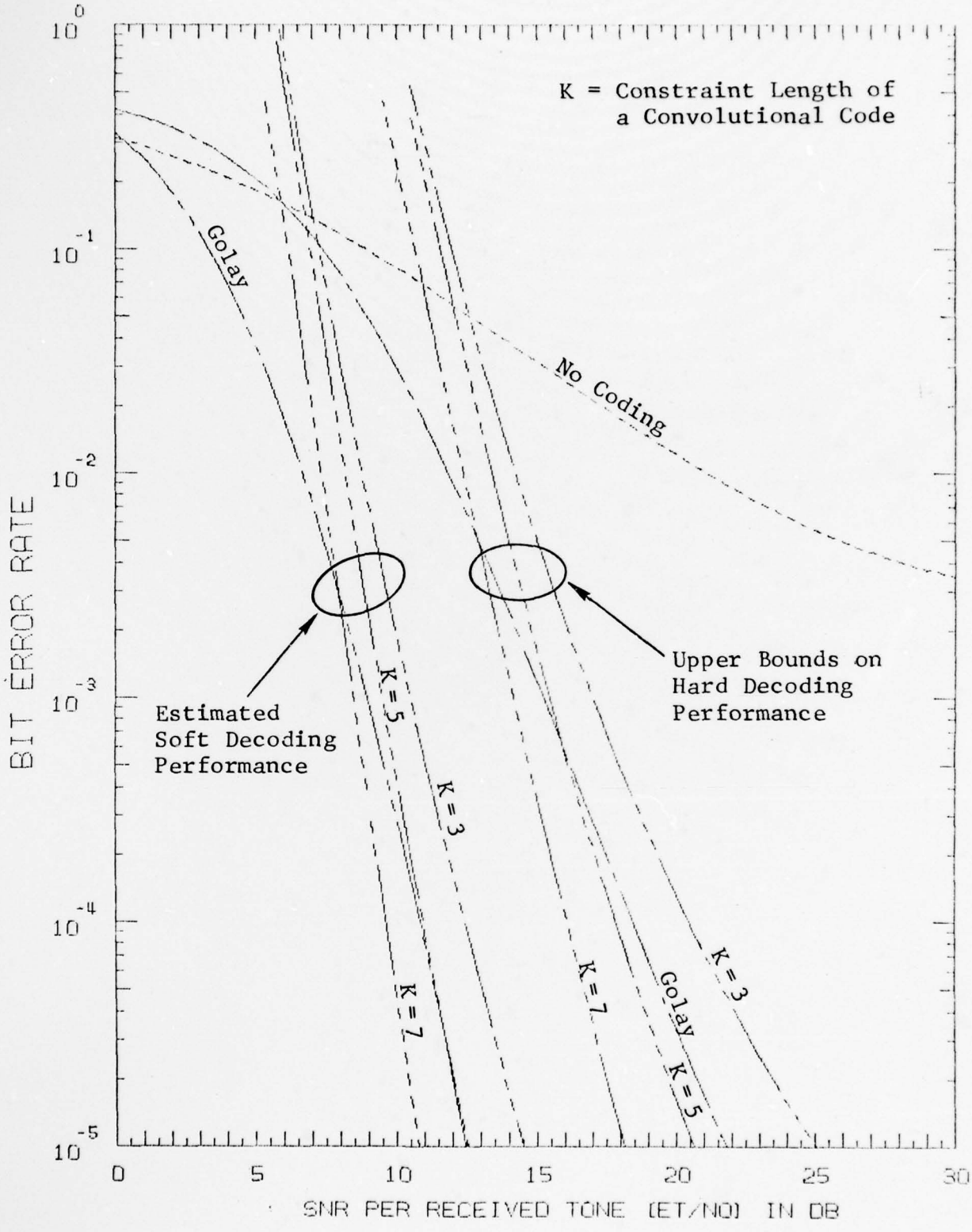


Figure 6.2 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 1 Hz, Frame Rate = 44.44 Frames/Sec.)

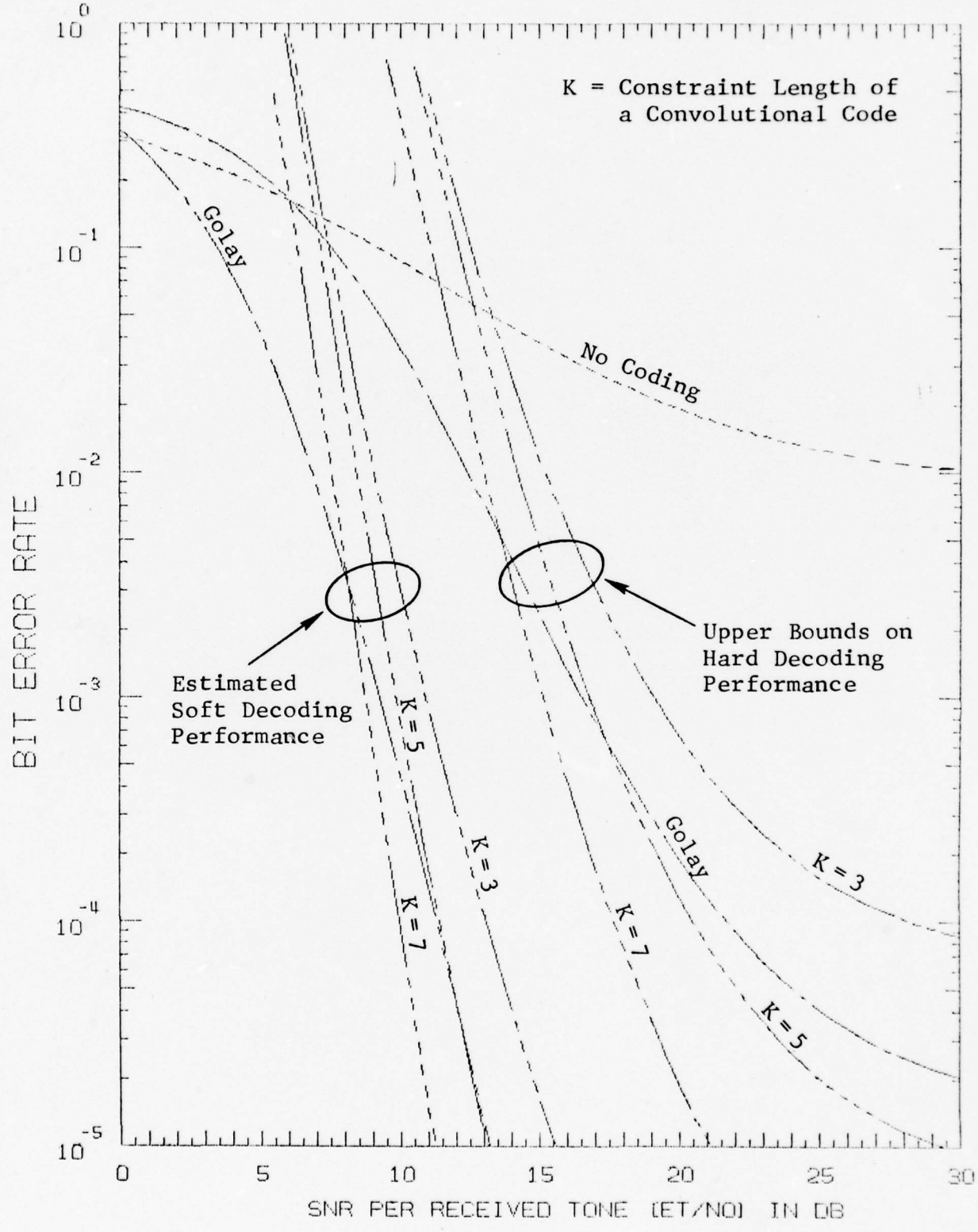


Figure 6.3 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 2 Hz, Frame Rate = 44.44 Frames/Sec.)

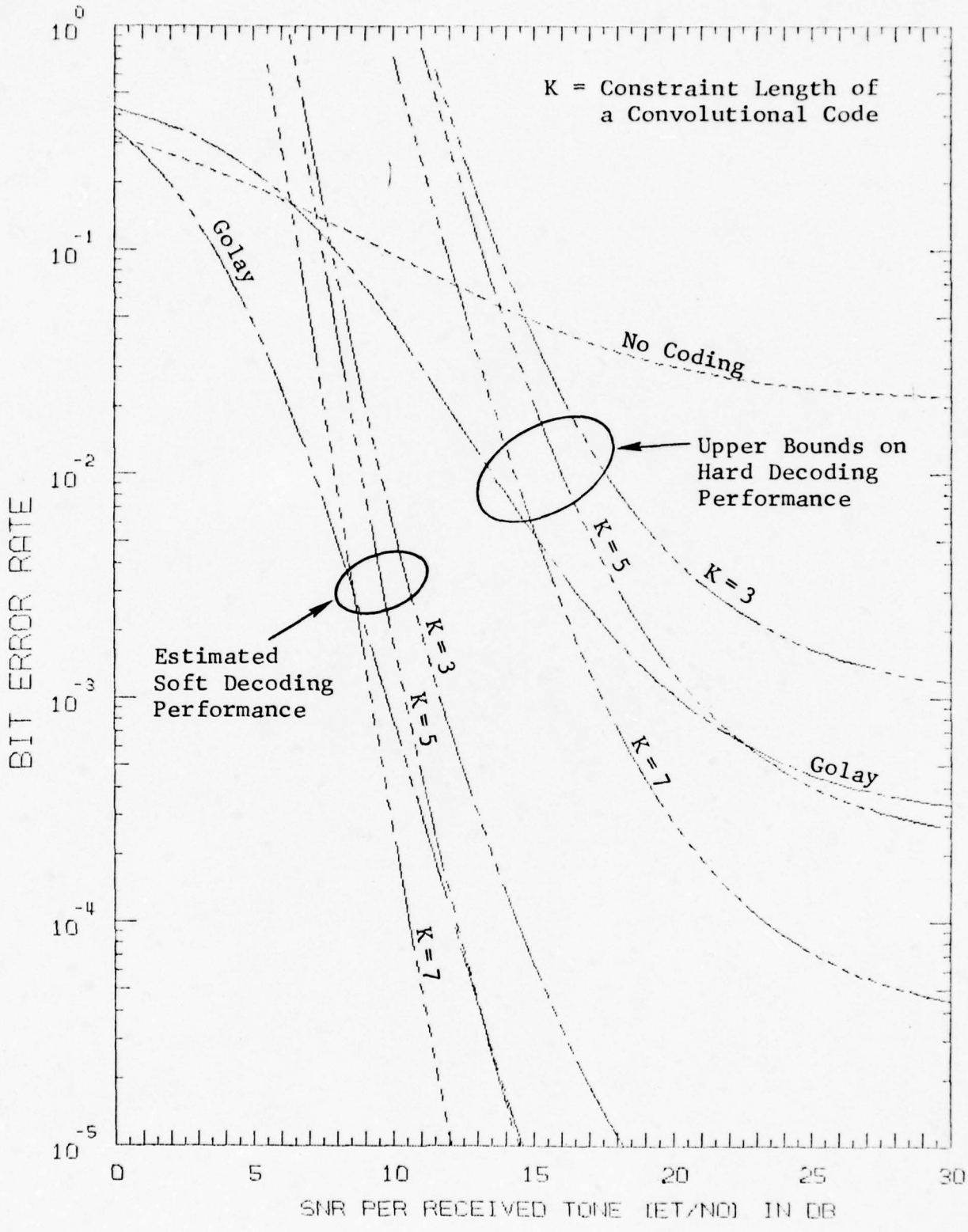


Figure 6.4 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 3 Hz, Frame Rate = 44.44 Frames/Sec.)

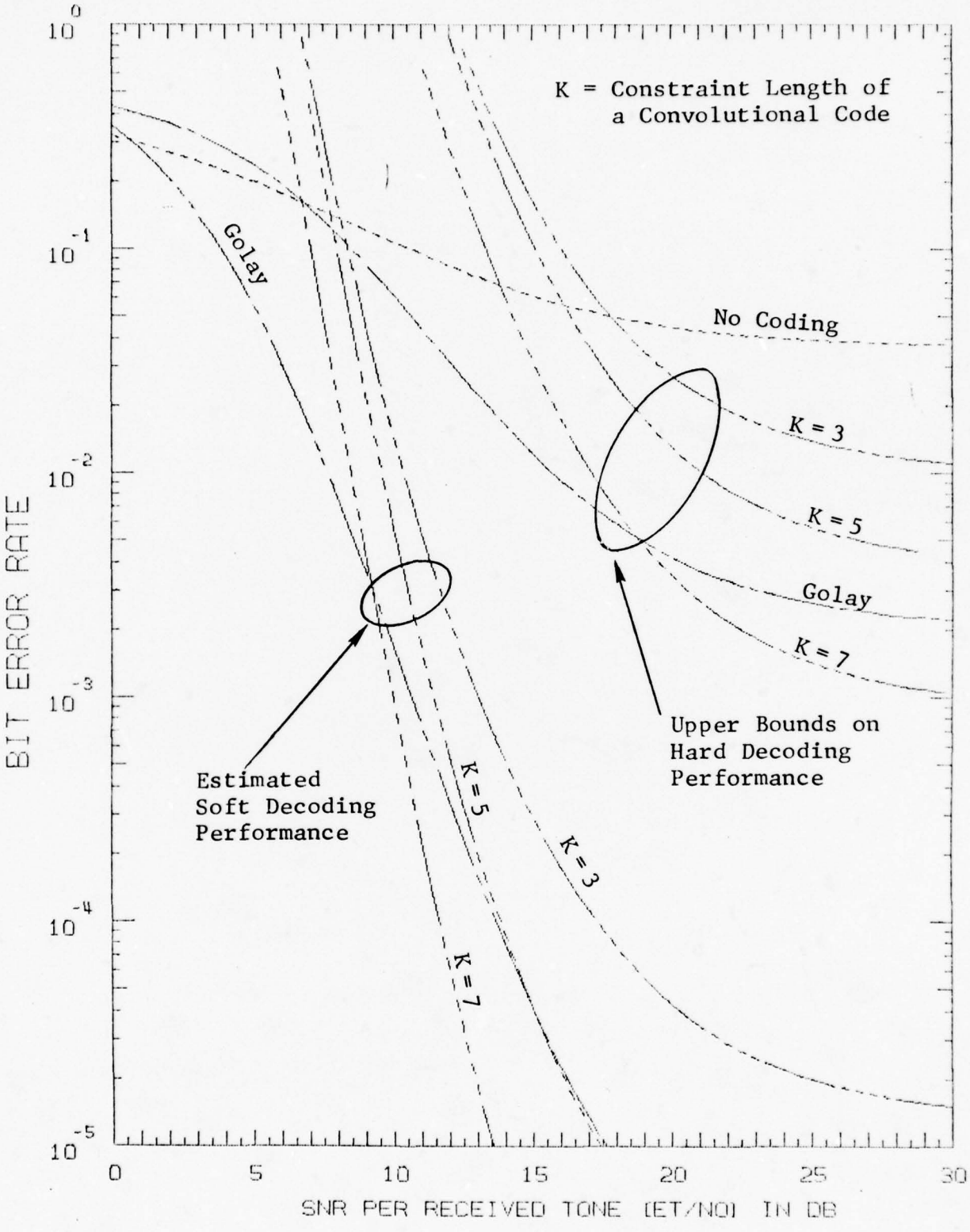


Figure 6.5 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 4 Hz, Frame Rate = 44.44 Frames/Sec.)

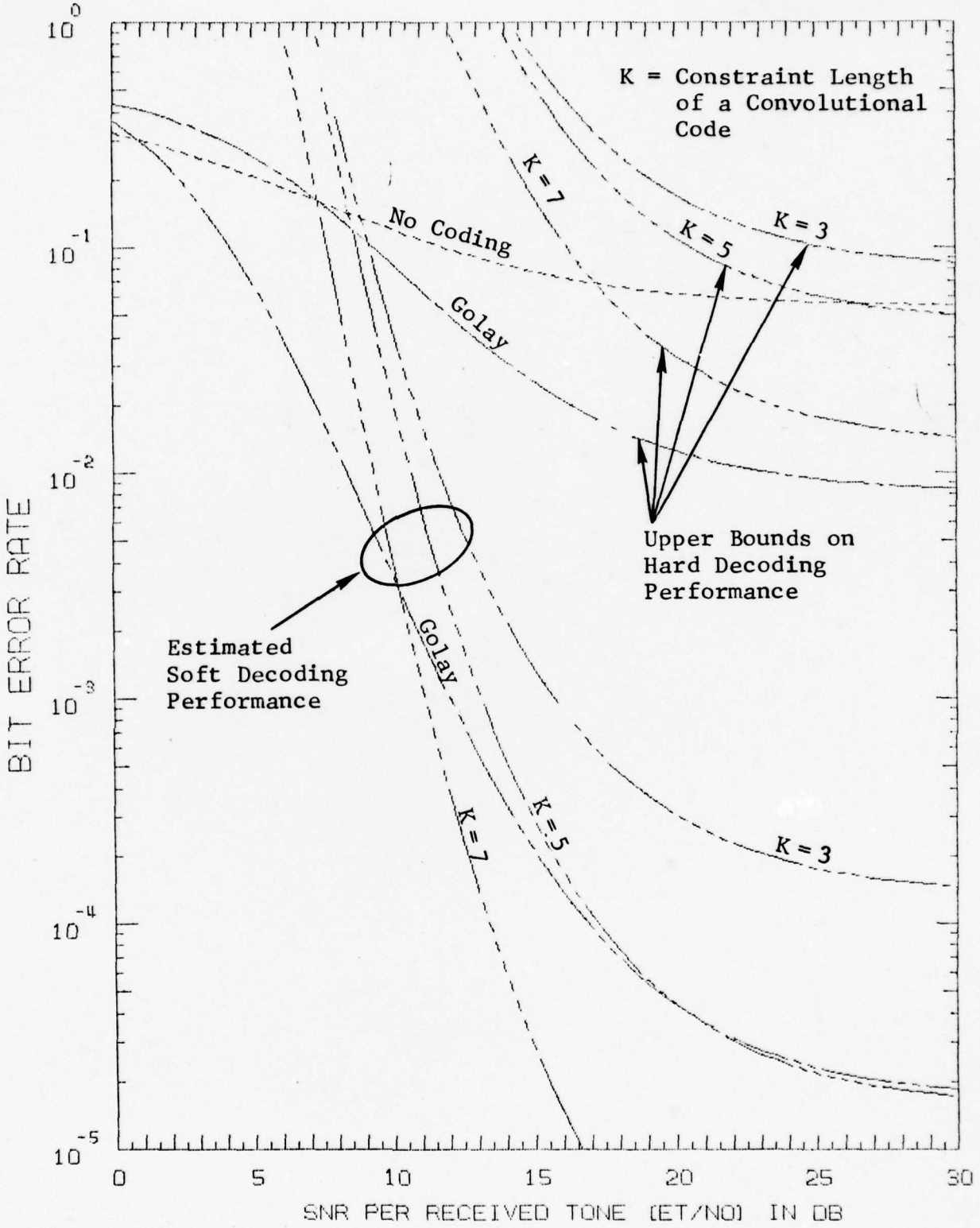


Figure 6.6 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 5 Hz, Frame Rate = 44.44 Frames/Sec.)

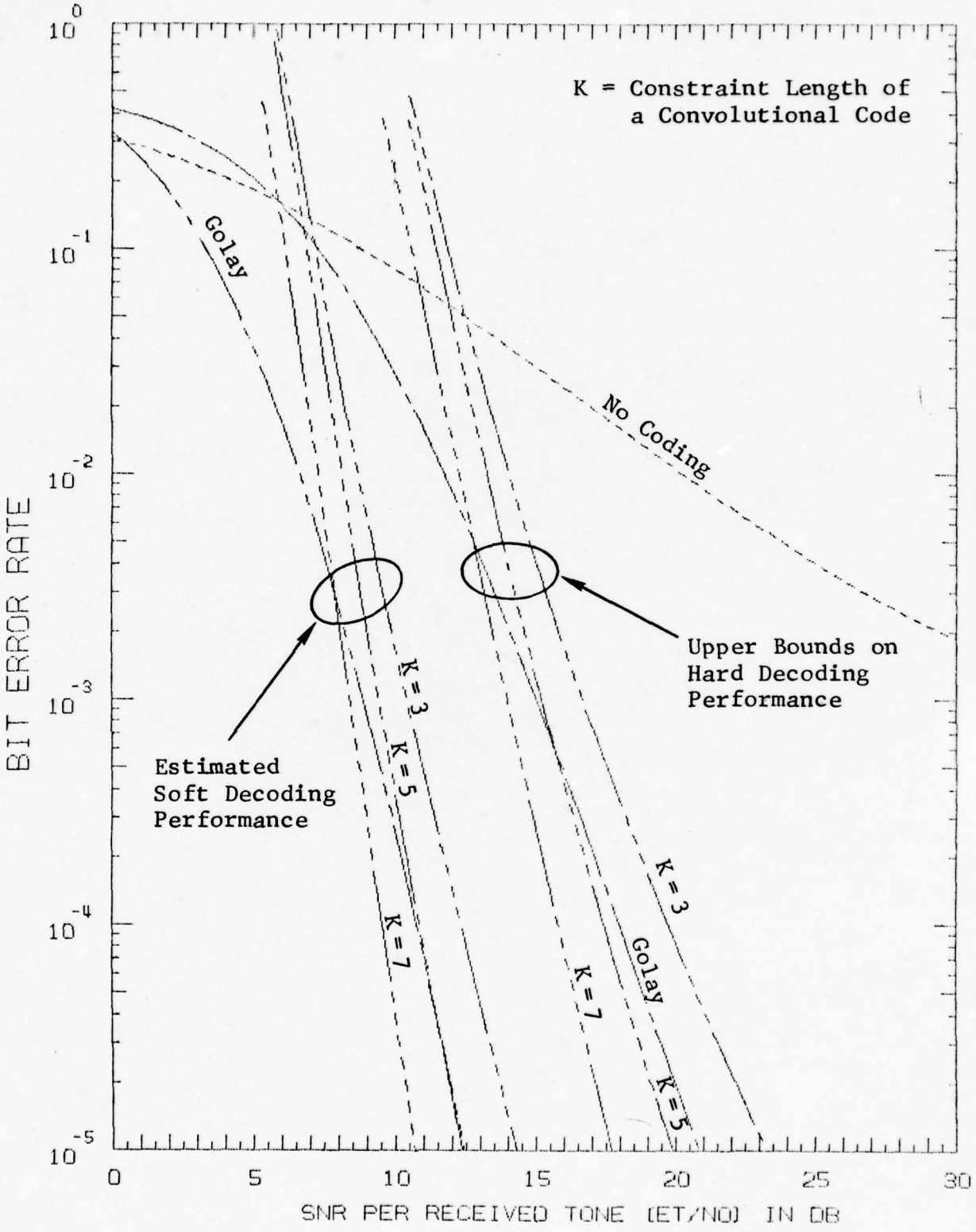


Figure 6.7 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 1 Hz, Frame Rate = 75 Frames/Sec.)

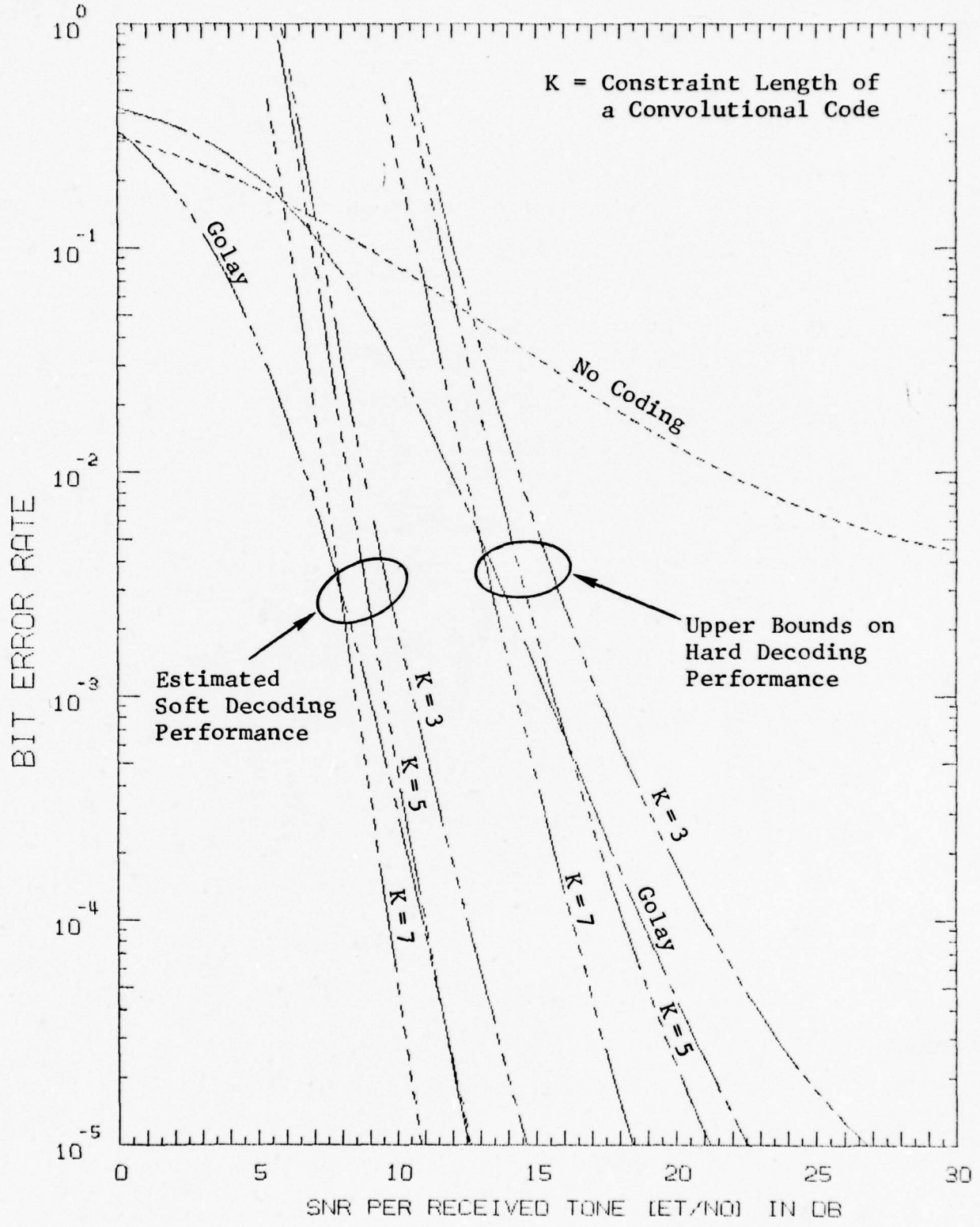


Figure 6.8 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 2 Hz, Frame Rate = 75 Frames/Sec.)

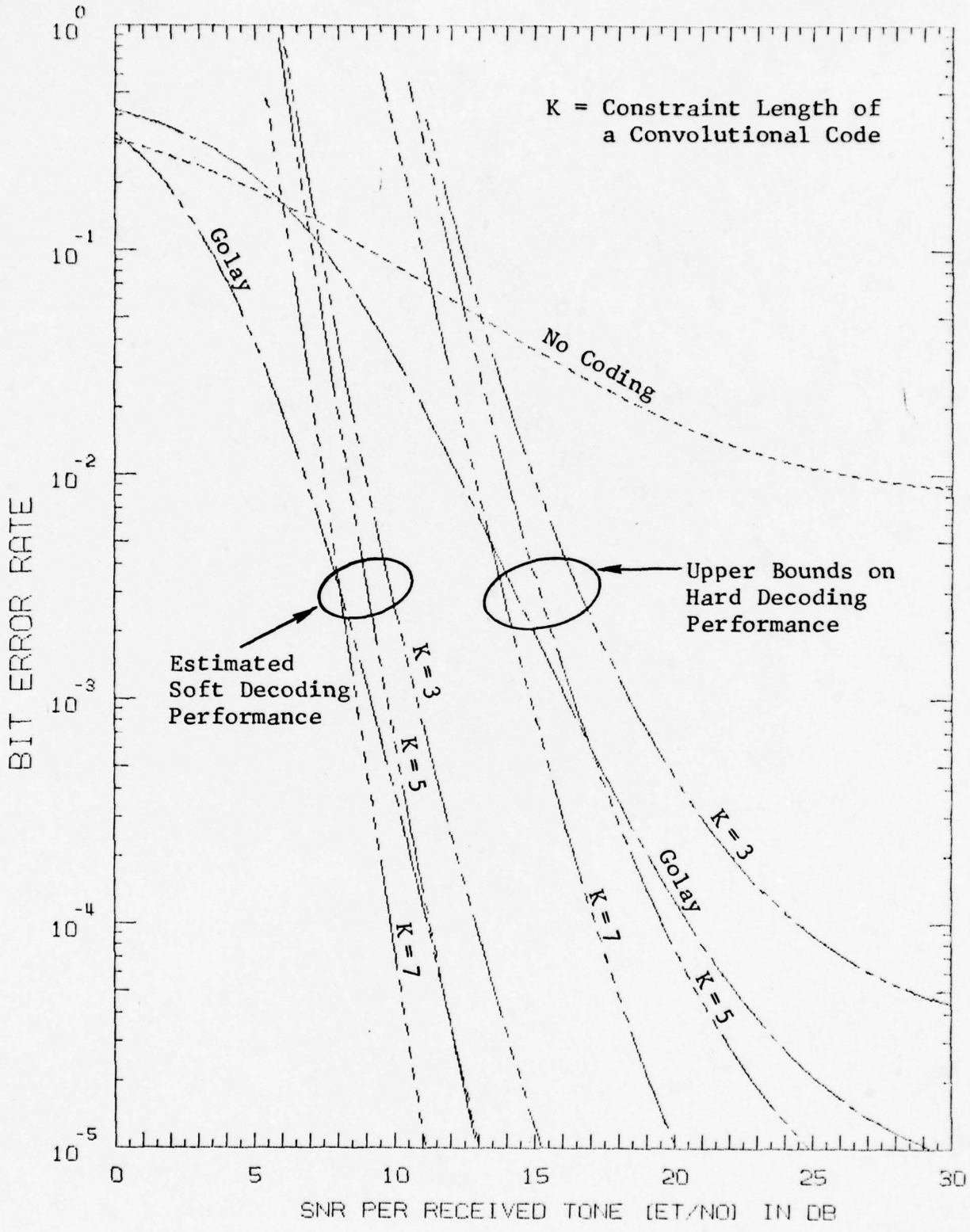


Figure 6.9 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 3 Hz, Frame Rate = 75 Frames/Sec.)

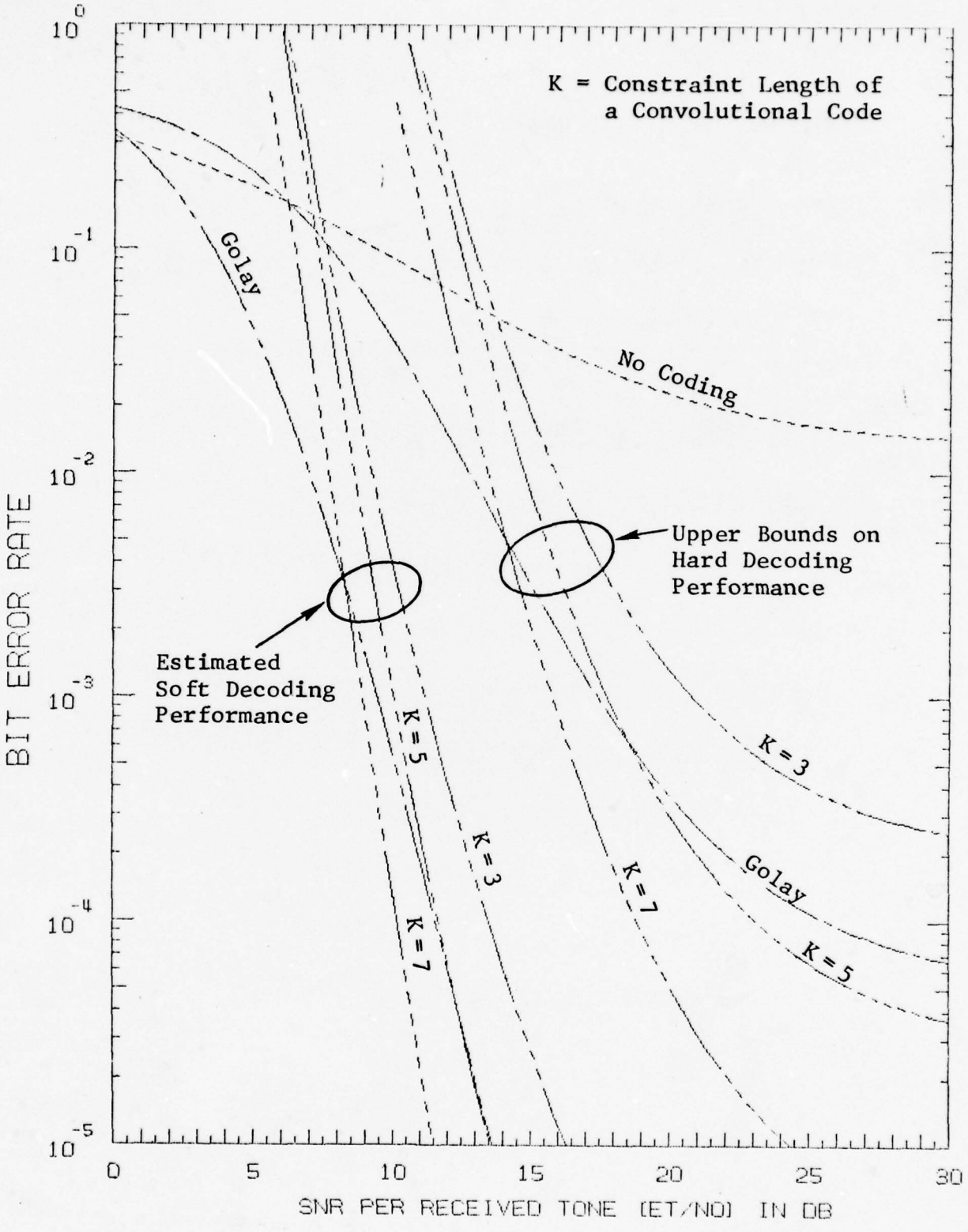


Figure 6.10 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 4 Hz, Frame Rate = 75 Frames/Sec.)

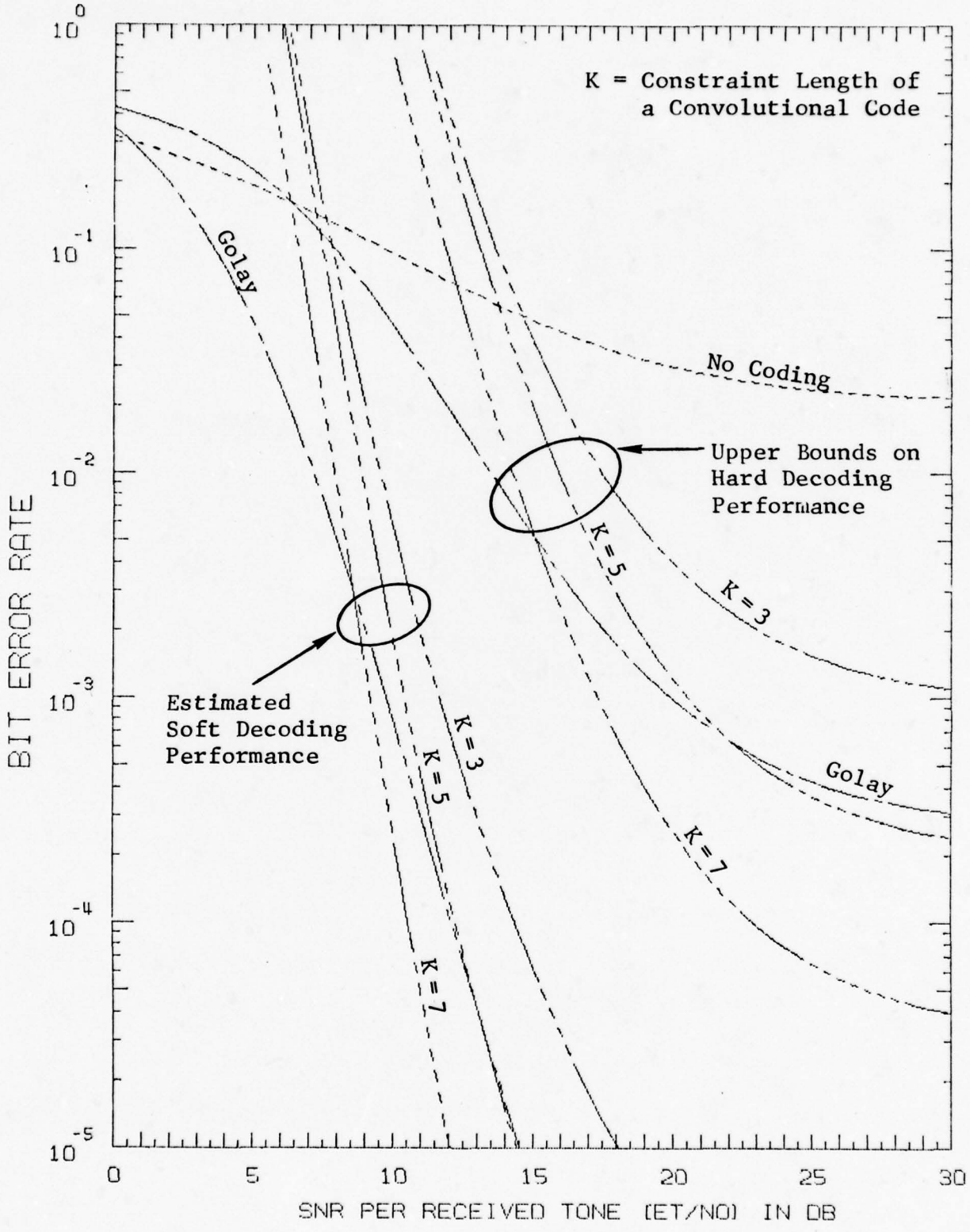


Figure 6.11 Performance of Rate-1/2 Codes Over a Time-Varying Rayleigh Channel With 4-Phase DPSK Modulation (B = 5 Hz, Frame Rate = 75 Frames/Sec.)

Since the major disadvantage of using a 44.44 f/s modem is its increased sensitivity to Doppler spreads and since this disadvantage is overcome with soft decoding, it is anticipated that the data mode for ANDVT will also use a frame rate of 44.44 f/s. Further studies in this direction should be undertaken after the coding for the speech digitization is finalized. However, preliminary results indicate that if sufficient interleaving is used the K = 7 convolutional code is superior to the Golay code. For lower data rates this convolutional code can readily be modified [6.2]. Comparisons between this code and a multiple rate structure is of interest, especially if a multiple rate code is selected for speech transmissions.

6.4 Performance Monitoring

When error correction coding is not used in a system, performance monitoring is achieved by observation of certain modulator features of the received signal. As an example, for a four-phase DPSK format, performance monitoring can be achieved by notnoting that the receiver phase of each tone is likely to fall in a window centered about the four possible phases transmitted. An even simpler example is the measurement of the ratio of "in-band" received power to "out-of-band" noise and setting a threshold for a loss of signal based on this ratio.

When coding is utilized, Fig. 6.12 illustrates a quite simple approach for performance monitoring which is accurate enough to even predict the channel (raw) error rate when this number is below 10^{-2}. Under these conditions, the decoder data is highly reliable and thus by comparing the raw data to the re-encoded data

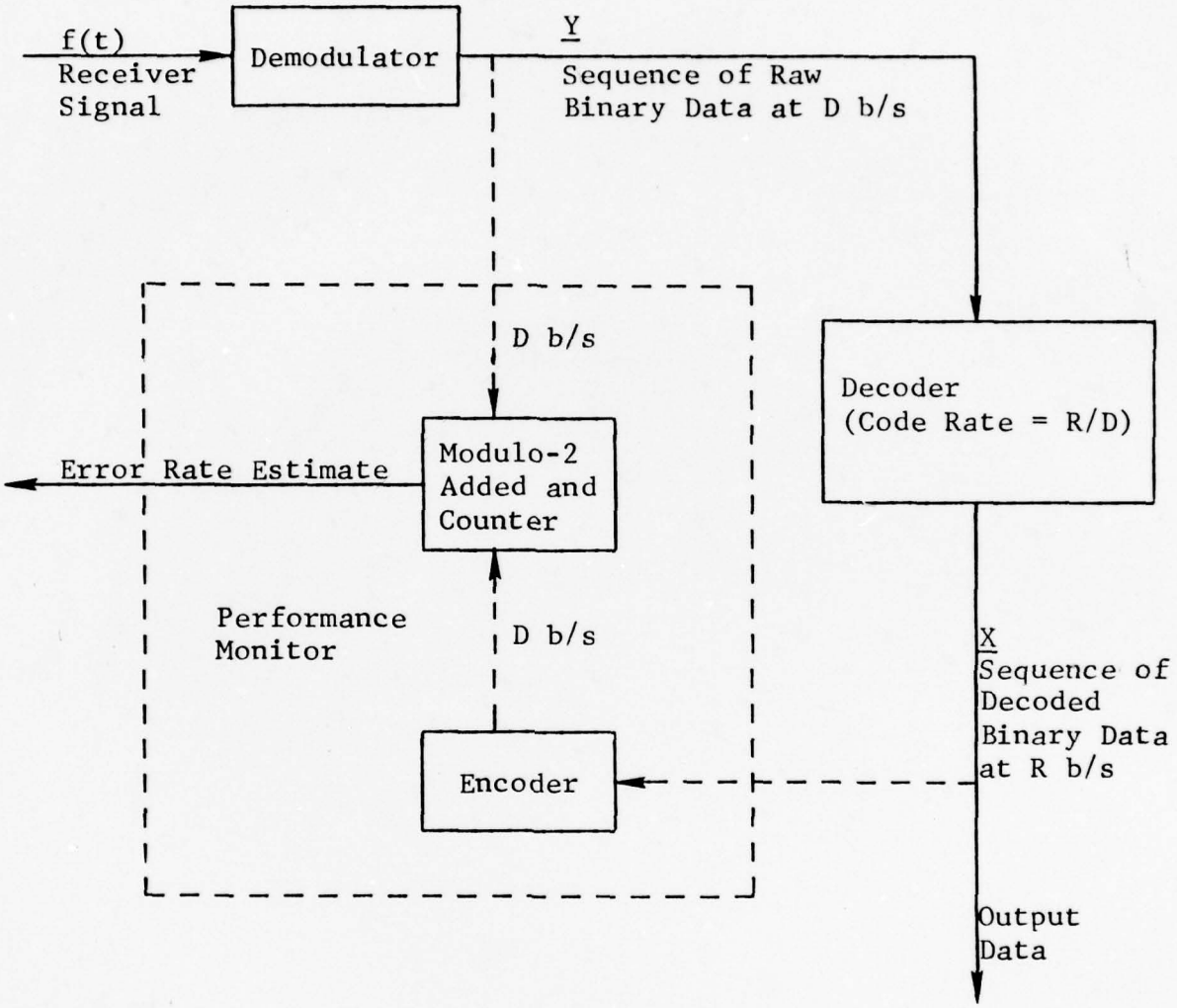


Figure 6.12 Performance Monitoring Based on Error Correction Coding

an accurate estimate of the channel error rate is possible. For poor channel error rates, i.e., above 10^{-2}, the decoded data may be unreliable and the channel error rate estimate may also be unreliable and, in fact, indicate a higher error rate than is actually present. This feature is still quite acceptable for most applications.

Other more sophisticated performance monitors are obtainable by including channel measurement (soft decision) information in the performance monitoring algorithm, such as the use of the analog weight of the estimated error pattern [6.3]. Techniques of this nature can be applied if the concept illustrated on Fig. 6.12 proves to be inadequate for a specific application.

SECTION 6

REFERENCES

[6.1] D. Chase, et al., "Codem I Data Transmission System," Final Report, Vol. I: Theory, General Atronics Corp., Report No. 2125-2211-28, July 1972.

[6.2] D. Chase, et al., "Robust Waveforms for HF Communications," Final Report, CNR, Inc., Contract No. N00173-76-C-0168, November 15, 1976.

[6.3] D. Chase, "A Class of Algorithms for Decoding Block Codes With Channel Measurement Information," IEEE Trans. on Information Theory, Vol. IT-18, No. 1, January 1972.

APPENDIX A

HF CHANNEL MODEL

Ideally, the response of one path of the HF channel to a sinusoidal signal of the form

$$x(t) = A \cos(\omega t + \theta) \tag{1}$$

is

$$y(t) = A\, r_j \cos(\omega t + \theta + \varphi_j) + n(t) \tag{2}$$

where r_j is the time-varying amplitude of the path (Rayleigh distributed), φ_j is a phase due to Doppler, time delay, etc., and n(t) is a complex white Gaussian noise.

Assuming a transmitted signal,

$$x(t) = \sqrt{2P} \cos(\omega t + \theta) \tag{3}$$

the received signal over a multipath channel is

$$y(t) = \sqrt{2P} \sum_{j=1}^{N} r_j \cos(\omega t + \theta + \varphi_j) + n(t) \tag{4}$$

where the r_j are independent Rayleighs, and $E\left[r_j^2\right] = P_j$, the power of the j^{th} path and $\sum_{j=1}^{N} P_j = 1$; and n(t) is the total noise on all paths.

The in-phase and quadrature matched filters at the receiver form the quantities

$$r_c = \frac{1}{T} \int_0^T \sqrt{2}\, y(t) \cos \omega t \, dt \tag{5}$$

$$r_s = -\frac{1}{T} \int_0^T \sqrt{2}\, y(t) \sin \omega t \, dt \tag{6}$$

Thus

$$r_c = \frac{1}{T} \sqrt{2P}\, \sqrt{2} \sum_{j=1}^{N} r_j \int_0^T \cos \omega t \, \cos(\omega t + \theta + \varphi_j) dt + n_s$$

$$= \frac{1}{T}\, 2\sqrt{P} \sum_{j=1}^{N} r_j \frac{T}{2} \cos(\theta + \varphi_j) + n_c$$

$$= \sum_{j=1}^{N} r_j \sqrt{P} \cos(\theta + \varphi_j) + n_c \tag{7}$$

and

$$r_s = \sum_{j=1}^{N} \sqrt{P}\, r_j \sin(\theta + \varphi_j) + n_s \tag{8}$$

Note that

$$E\left[r_c^2\right] + E\left[r_s^2\right] = P \tag{9}$$

and

$$E\left[n_c^2\right] = E\left[n_s^2\right] = \frac{N_0}{2T} \tag{10}$$

Thus γ, the signal-to-noise ratio per tone, is given by:

$$\gamma = \frac{P}{2\left(\frac{N_0}{2T}\right)} = \frac{PT}{N_0} \equiv \frac{E}{N_0} \tag{11}$$

For an array of tones,

$$y(t) = \sum_{i=1}^{N_{TONE}} \sum_{j=1}^{N_{PATH}} \sqrt{2P}\, r_j \cos(\omega_i t + \theta_i + \varphi_{ij}) + n_i(t) \tag{12}$$

Note that for this model, r_j is assumed constant over all frequencies of interest while φ_{ij} is frequency-dependent. r_c and r_s have the same form for each tone with the r_j's the same in each case.

The simplicity of Eq. (12) suggests that the HF channel model is well-suited for computer program implementation. Define $\{\Delta\theta_i\}_{i=1}^{N_{TONE}}$ to be the set of information-bearing phase differences for the tone array. The quantity,

$$\theta_n(k) = \sum_{j=1}^{k} \Delta\theta_n(j) \tag{13}$$

which is the actual transmitted phase angle at the k^{th} iteration, is formed.

The received phase angle for the n^{th} tone over the j^{th} path at the k^{th} iteration, $\psi_n(k)$, is

$$\psi_{jn}(k) = \theta_n(k) + \tan^{-1}\left[\frac{Y_j(k)}{X_j(k)}\right] + 2\pi\left[(kT - \tau_j)\nu_j - (n f_{sep} + f_0)\tau_j\right] \tag{14}$$

where $X_j(k)$ and $Y_j(k)$ are the real and imaginary channel gains over the j^{th} path at the k^{th} iteration; τ_j is the time delay of the j^{th} path, ν_j is the Doppler offset, $(n f_{sep} + f_0)$ is the distance of the n^{th} tone from the carrier, and T is the integrate time for the matched filters. Each $\{X_j(kT)\}_k$ and each $\{Y_j(kT)\}_k$ is obtained by filtering a sequence of independent, unit variance Gaussian random variables. The filter used is a second-order recursive digital filter equivalent to the cascade of two identical single pole filters. If w(kT) is the input sequence and z(kT) is the output, the filter is defined by

$$z(kT) = A_1 z(kT - T) + A_2 z(kT - 2T) + A_3 w(kT) \tag{15}$$

where

$$A_1 = 2e^{-\pi BT} \tag{16}$$

$$A_2 = -e^{-2\pi BT} \tag{17}$$

$$A_3 = \sqrt{\frac{P}{2}} \frac{(1 - e^{-2\pi BT})^{3/2}}{(1 + e^{-2\pi BT})^{1/2}} \tag{18}$$

and B is the Doppler (double-sided) bandwidth of the path.

This filter is initialized prior to forming Z(T) by loading Z(0) and Z(-T) with two Gaussian random variables having a correlation coefficient given by

$$\rho = \frac{2e^{-\pi BT}}{1 + e^{-2\pi BT}} \tag{19}$$

The received power is:

$$r_j(k) = \sqrt{X_j^2(k) + Y_j^2(k)} \tag{20}$$

Thus, overall,

$$r_{cn}(k) = \sum_{j=1}^{N_{PATH}} \sqrt{X_j^2(k) + Y_j^2(k)} \cos \psi_{jn}(k) + n_{cn}(k) \tag{21}$$

and

$$r_{sn}(k) = \sum_{j=1}^{N_{PATH}} \sqrt{X_j^2(k) + Y_j^2(k)} \sin \psi_{jn}(k) + n_{sn}(k) \tag{22}$$

The decision statistics appropriate to this processing are:

$$\ell_{cn} = r_{cn}(k)\, r_{cn}(k-1) + r_{sn}(k)\, r_{sn}(k-1) \tag{23}$$

$$\ell_{sn} = r_{sn}(k)\, r_{cn}(k-1) - r_{cn}(k)\, r_{sn}(k-1) \tag{24}$$

where ℓ_{cn} (ℓ_{sn}) is the component of the n^{th} tone at frame number k in-phase (quandrature) with the n^{th} tone at frame number (k - 1).

APPENDIX B

CODING CONSIDERATION FOR VOICE PROTECTION

In this appendix, some insight into the code selection for both multiple-rate codes and block codes is given. Unfortunately, mathematical results are obtained for a simplified HF channel model which does not allow for correlation between code digits. Thus, even though firm conclusions cannot be drawn from these results, trade-offs in performance can still be ascertained. Note that only single-burst-correcting and double-burst-correcting multiple-rate codes are considered. The technique used in this appendix can be applied to general multiple-burst-correcting multiple-rate codes; however, the computation becomes increasingly tedious and lengthy as the burst-correcting capability increases.

B.1 Performance of Single-Burst-Correcting and Double-Burst-Correcting Multiple-Rate Codes

Performances of single-burst-correcting and double-burst-correcting multiple rate codes are derived under the assumption that there are no undetected errors and that blocks are independent. Both the high rate code and the low rate codes are taken into consideration. The performance curves for some multiple-rate codes suitable for 41 tones per frame format are computed. The results are compared to the Golay (24,12;8) code and the low rate code employed as a block code. Only the hard decoding algorithm is used in obtaining the results for multiple rate codes.

B.1.1 General Error Probabilities

Let P_h be the probability that a high rate code block contains uncorrectable errors and P_ℓ the corresponding probability for the low rate code. When bits within a code block are independent, we have

$$P_h = \sum_{i=e_h+1}^{N} \binom{N}{i} p^i (1-p)^{N-i} \tag{1}$$

and

$$P_\ell = \sum_{i=e_\ell+1}^{N} \binom{N}{i} p^i (1-p)^{N-i} \tag{2}$$

where e_h and e_ℓ are the respective error-correcting capabilities of the high rate code and the low rate code, and p is the bit error rate. Let α_h and α_ℓ be the conditional bit error rate for the high rate code and the low rate code, respectively, given the code block is in error. For the case where bits are independent within a code block, we have

$$\alpha_h = \frac{\frac{1}{N} \sum_{i=e_h+1}^{N} \binom{N}{i} i p^i (1-p)^{N-i}}{\sum_{i=e_h+1}^{N} \binom{N}{i} p^i (1-p)^{N-i}} \tag{3}$$

and

$$\alpha_\ell = \frac{\frac{1}{N} \sum_{i=e_\ell+1}^{N} \binom{N}{i} i p^i (1-p)^{N-i}}{\sum_{i=e_\ell+1}^{N} \binom{N}{i} p^i (1-p)^{N-i}} \tag{4}$$

It is more convenient to define

$$\beta_h = \alpha_h P_h \tag{5}$$

and

$$\beta_\ell = \alpha_\ell P_\ell \tag{6}$$

Then β_h and β_ℓ are given, respectively, by the numerators of Eqs. (3) and (4).

B.1.2 Single-Burst-Correcting Multiple-Rate Code

A single-burst-correcting multiple-rate code can be characterized by a two-state Markov chain as shown in Fig. 1. The states are h and ℓ. When the chain is in state h, a code block is decoded as a high rate code; when it is in state ℓ, a code block is decoded as a low rate code. A state is determined if the previous state and transition probability are known. The transition (probability) matrix for the single Markov chain is

$$M = \begin{bmatrix} P_h & 1 - P_h \\ P_\ell & 1 - P_\ell \end{bmatrix} \tag{7}$$

Let w_h and w_ℓ be the steady-state distribution for states h and ℓ, respectively. Then

$$[w_h \quad w_\ell] \begin{bmatrix} P_h & 1 - P_h \\ P_\ell & 1 - P_\ell \end{bmatrix} = [w_h \quad w_\ell] \tag{8}$$

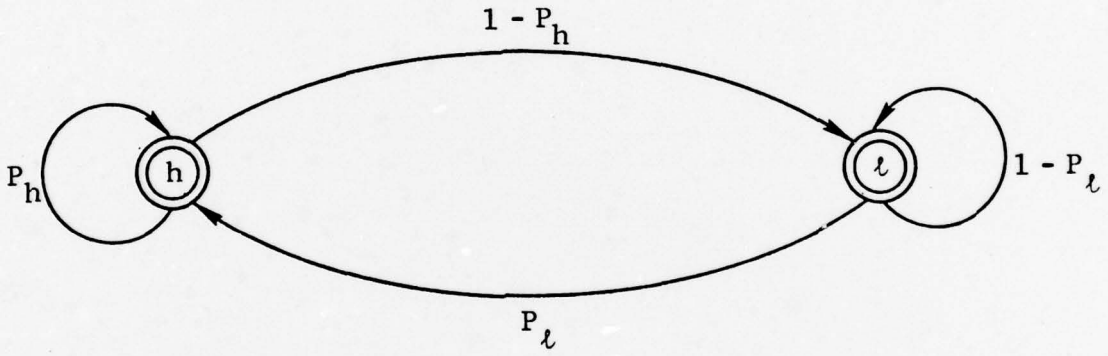


Figure 1 A Markov Chain Model for a Single-Burst-Correcting Multiple-Rate Code

Solving Eq. (8), we have

$$w_h = \frac{P_\ell}{1 - P_h + P_\ell} \tag{9}$$

and

$$w_\ell = \frac{1 - P_h}{1 - P_h + P_\ell} \tag{10}$$

Recall that the multiple-rate code is an $(N, K_1 + K_2; D_\ell) \rightarrow (N, 2K_1 + K_2; D_h)$ code. The most significant K_1 bits are protected by the multiple-rate code structure; and the second most significant K_2 bits are protected by either the low rate code or the high rate code, depending on the state. For the K_1 most significant bits, an erroneous block results if two consecutive blocks are in error; thus, the bit error rate for the K_1 most significant bit is:

$$\begin{aligned} P_{bit} &= w_h P_h^2 \alpha_h + w_\ell P_\ell P_h \alpha_\ell \\ &= w_h P_h \beta_h + w_\ell P_h \beta_\ell \\ &= \frac{P_\ell P_h \beta_h + (1 - P_h) P_h \beta_\ell}{1 - P_h + P_\ell} \end{aligned} \tag{11}$$

and the bit error rate for the second most significant K_2 bits is given by

$$P'_{bit} = w_h \cdot p_{hb} + w_\ell \cdot p_{\ell b} \tag{12}$$

where p_{hb} and $p_{\ell b}$ are the decoded bit error rates, respectively, for the high rate code and the low rate code when employed as block codes. At high signal-to-noise ratios, $w_h \to 0$, $w_\ell \to 1$; thus,

$$p_{bit} \to p_h \beta_\ell \tag{13}$$

and

$$p'_{bit} \to p_{\ell b} \tag{14}$$

B.1.3 Double-Burst-Correcting Multiple-Rate Code

Whether a block is decoded as a low rate code or as a high rate code depends on the decoded results of the previous two blocks for the double-burst-correcting multiple-rate code. As a result, five states are needed to describe the Markov chain: Hh, Hℓ, $\bar{H}$h, Lℓ, and $\bar{L}$h. The second letter of the state shows that the present code block is decoded as the low rate code (ℓ) or the high rate code (h). The first letter denotes the conditions of the previous block:

Letter	Condition for Previous Block: Code	Condition for Previous Block: Containing Burst?
H	High Rate	No
$\overline{H}$	High Rate	Yes
L	Low Rate	No
$\overline{L}$	Low Rate	Yes

Note that it is impossible to have state $\overline{H}\ell$ and $\overline{L}\ell$ because, when the previous block is known to contain a burst, the present block must be decoded as the high rate code. Also, it is impossible to have state Lh because the state Lh implies that the previous two blocks are decoded correctly; hence, the present block must be decoded as a low rate code. Therefore, only ℓ is allowed to follow L. The state diagram is shown in Fig. 2. Let w_{Hh}, $w_{\overline{Hh}}$, $w_{H\ell}$, $w_{\overline{Lh}}$, and $w_{L\ell}$ be the corresponding steady-state distributions for states Hh, $\overline{H}$h, Hℓ, $\overline{L}$h, and Lℓ. Then

$$(w_{Hh}\ w_{\overline{Hh}}\ w_{H\ell}\ w_{\overline{Lh}}\ w_{L\ell}) \begin{bmatrix} 0 & P_h & 1-P_h & 0 & 0 \\ 1-P_h & P_h & 0 & 0 & 0 \\ 0 & 0 & 0 & P_\ell & 1-P_\ell \\ 1-P_h & P_h & 0 & 0 & 0 \\ 0 & 0 & 0 & P_\ell & 1-P_\ell \end{bmatrix}$$

$$= (w_{Hh}\ w_{\overline{Hh}}\ w_{H\ell}\ w_{\overline{Lh}}\ w_{L\ell}) \qquad (15)$$

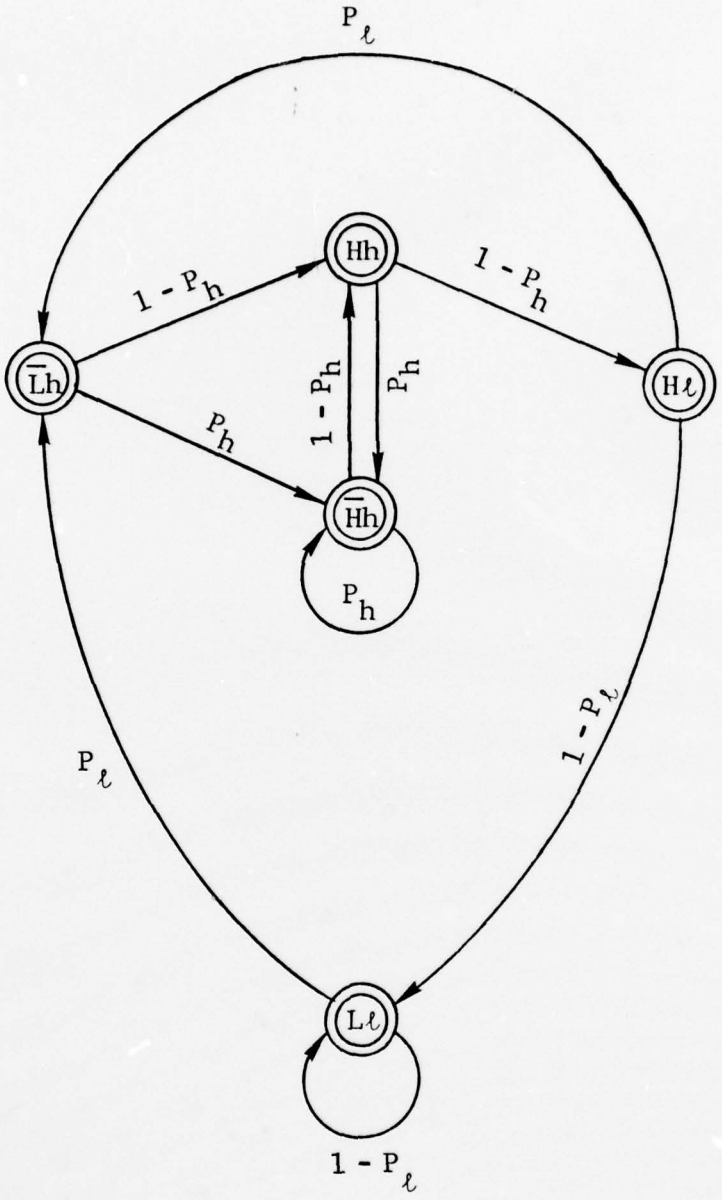


Figure 2 A Markov Chain Model for a Double-Burst-Correcting Multiple-Rate Code

Solving Eq. (15), we have

$$w_{Hh} = P_\ell \cdot (1 - P_h)/[P_\ell + (1 - P_h)(1 - P_h + P_\ell)] \qquad (16)$$

$$w_{\overline{Hh}} = P_\ell[1 - (1 - P_h)^2]/[P_\ell + (1 - P_h)(1 - P_h + P_\ell)] \qquad (17)$$

$$w_{H\ell} = w_{\overline{Lh}} = P_\ell(1 - P_h)^2/[P_\ell + (1 - P_h)(1 - P_h + P_\ell)] \qquad (18)$$

$$w_{L\ell} = (1 - P_\ell)(1 - P_h)^2/[P_\ell + (1 - P_h)(1 - P_h + P_\ell)] \qquad (19)$$

Recall that the most significant K_1 bits of blocks containing a burst can be correctly recovered if they are followed by, or adjacent to, two error-free blocks. In Table 1, all possible cases resulting in an erroneous block being present are listed (a "1" indicates the block contains a burst and a "0" denotes the block is correct). Since there is a burst at time 0, all blocks at time τ must be decoded as high rate codes. Also, due to the error patterns shown in the table, all code blocks at time 2τ and 3τ are also decoded as high rate codes. For cases 1, 2, and 3, the joint probability time τ to time 3τ is

$$T = 2P_h^2(1 - P_h) + P_h^3 = 2P_h^2 - P_h^3 \qquad (20)$$

For cases 4 through 8, the joint probability from time τ to time 3τ is

$$S = P_h(1 - P_h)^2 + 3P_h^2(1 - P_h) + P_h^3 = P_h(1 - P_h) + T \qquad (21)$$

TABLE 1

BLOCK ERROR PATTERNS RESULTING IN A DECODED BLOCK ERROR AT TIME 0

Time →	$-\tau$	0	τ	2τ	3τ	State at Time 0
Case 1	0	1	1	0	1	$H\bar{h}$
Case 2	0	1	1	1	0	$H\ell$ or
Case 3	0	1	1	1	1	$L\ell$
Case 4	1	1	0	1	0	
Case 5	1	1	0	1	1	$\bar{H}h$
Case 6	1	1	1	0	1	or
Case 7	1	1	1	1	0	$\bar{L}h$
Case 8	1	1	1	1	1	

The bit error rate for the K_1 most significant bits is

$$P_{bit} = (w_{Hh}P_h\alpha_h + w_{H\ell}P_\ell\alpha_\ell + w_{L\ell}P_\ell\alpha_\ell)T + (w_{Hh} + w_{\overline{Lh}})P_h\alpha_h S$$

$$= [w_{Hh}\beta_h + (w_{H\ell} + w_{L\ell})\beta_\ell]T + (w_{\overline{Hh}} + w_{\overline{Lh}})\beta_h S$$

$$= [(w_{Hh} + w_{\overline{Hh}} + w_{\overline{Lh}})\beta_h + (w_{H\ell} + w_{L\ell})\beta_\ell](2P_h^2 - P_h^3)$$

$$+ (w_{\overline{Hh}} + w_{\overline{Lh}})\beta_h P_h(1 - P_h) \qquad (23)$$

The bit error rate for the K_2 second most significant bits is

$$P'_{bit} = (w_{Hh} + w_{\overline{Hh}} + w_{\overline{Lh}})P_{hb} + (w_{H\ell} + w_{L\ell})P_{\ell b} \qquad (24)$$

At high signal-to-noise ratio,

$$\left.\begin{array}{l} w_{H\ell} \to 0,\ w_{\overline{Hh}} \to 0,\ w_{H\ell} \to 0,\ w_{\overline{Lh}} \to 0 \\ \text{and} \\ w_{L\ell} \to 1 \end{array}\right\} \qquad (25)$$

Thus,

$$P_{bit} \to P_\ell\alpha_\ell(2P_h^2 - P_h^3) = P_h^2(2 - P_h)\beta_\ell \qquad (26)$$

and

$$P'_{bit} \to P_{\ell b} \qquad (27)$$

Comparing Eqs. (26) and (27) to Eqs. (13) and (14), we observe that p'_{bit} is about the same for both single- and double-burst-correcting multiple-rate codes at high signal-to-noise ratios; but p_{bit} for the double-burst-correcting multiple-rate code would be much smaller than that for the single-burst-correcting code at high signal-to-noise ratios.

B.1.4 Performance Curves for Multiple-Rate Codes

Performance curves for single- and double-burst-correcting multiple-rate codes are plotted in Figs. 3 through 8. The relevant information concerning each of the figures is given in Table 2.

In Table 3 the performance of various coding schemes are compared at a signal-to-noise ratio per received tone (E_t/N_0) of 15 dB.

B.1.5 Recommendations

1) When a block code is employed, either the (64,36;12) code or its shortened version may provide a better performance than the two Golay (24,12;8) codes used in parallel when hard decoding is employed. In Section B.2 further comparison of these two codes will be presented.

2) The (43,15;12) → (43,30;6), (64,36;12) → (64,51;6), and (64,36;12) → (64,45;8) multiple rate codes appear to be attractive for voice protection when hard decoding is employed. These codes warrant simulation for soft decoding performance.

TABLE 2

RELEVANT INFORMATION FOR FIGURES 3 THROUGH 8

Figure	Single- and Double-Burst-Correcting Multiple-Rate Codes	K_1	K_2	Block Codes Compared
3	(43,15;12) → (43,30;6)	15	0	(24,12;8) & (43,15;12)
4	(64,36;12) → (64,51;6)	15	21	(24,12;8) & (64,36;12)
5	(64,36;12) → (64,45;8)	9	27	(24,12;8) & (64,36;12)
6	(64,36;12) → (64,39;10)	3	33	(24,12;8) & (64,36;12)
7	(82,54;9) → (82,68;5)	14	40	(24,12;8) & (82,54;9)
8	(82,54;9) → (82,61;7)	7	47	(24,12;8) & (82,54;9)

TABLE 3

COMPARISON OF VARIOUS CODING APPROACHES USING HARD DECODING FOR 41 TONES/FRAME FORMAT AT A SIGNAL-TO-NOISE RATIO PER RECEIVED TONE = 15 dB FOR A SLOWLY FADING HF CHANNEL USING 4-PHASE DPSK MODULATION

Coding Scheme		Performance		
		K_1/Bit Error Rate	K_2/Bit Error Rate	K_3/Bit Error Rate
(24,12;8) Block Code		$24/9.17 \times 10^{-4}$	$30/2.93 \times 10^{-2}$	—
(43,15;12) Block Code		$15/2.47 \times 10^{-4}$	$39/2.93 \times 10^{-2}$	—
(64,36;12) Block Code		$36/1.31 \times 10^{-3}$	$18/2.93 \times 10^{-2}$	—
(82,54;9) Block Code		$54/1.08 \times 10^{-2}$	—	—
(43,15;12) → (43,30;6)	SBC	$15/3.12 \times 10^{-5}$	$39/2.93 \times 10^{-2}$	—
	DBC	$15/1.07 \times 10^{-5}$	$39/2.93 \times 10^{-2}$	—
(64,36;12) → (64,51;6)	SBC	$15/3.86 \times 10^{-4}$	$21/1.60 \times 10^{-3}$	$18/2.93 \times 10^{-2}$
	DBC	$15/3.08 \times 10^{-4}$	$21/1.99 \times 10^{-3}$	$18/2.95 \times 10^{-2}$
(64,36;12) → (64,45;8)	SBC	$9/1.41 \times 10^{-4}$	$27/1.42 \times 10^{-3}$	$18/2.93 \times 10^{-2}$
	DBC	$9/4.61 \times 10^{-5}$	$27/1.54 \times 10^{-3}$	$18/2.93 \times 10^{-2}$
(64,36;12) → (64,39;10)	SBC	$3/4.47 \times 10^{-5}$	$30/1.34 \times 10^{-3}$	$18/2.93 \times 10^{-2}$
	DBC	$3/5.07 \times 10^{-6}$	$30/1.37 \times 10^{-3}$	$18/2.93 \times 10^{-2}$
(82,54;9) → (82,68;5)	SBC	$14/3.58 \times 10^{-3}$	$40/1.36 \times 10^{-2}$	—
	DBC	$14/4.12 \times 10^{-3}$	$40/1.71 \times 10^{-2}$	—
(82,54;9) → (82,61;7)	SBC	$7/1.53 \times 10^{-3}$	$47/1.18 \times 10^{-2}$	—
	DBC	$7/9.16 \times 10^{-4}$	$47/1.29 \times 10^{-2}$	—

K_1 = Number of most significant bits, K_2 = Number of second most significant bits, K_3 = Number of third most significant bits

MRC = Multiple-Rate Code

SBC = Single-Burst-Correcting MRC

DBC = Double-Burst-Correcting MRC

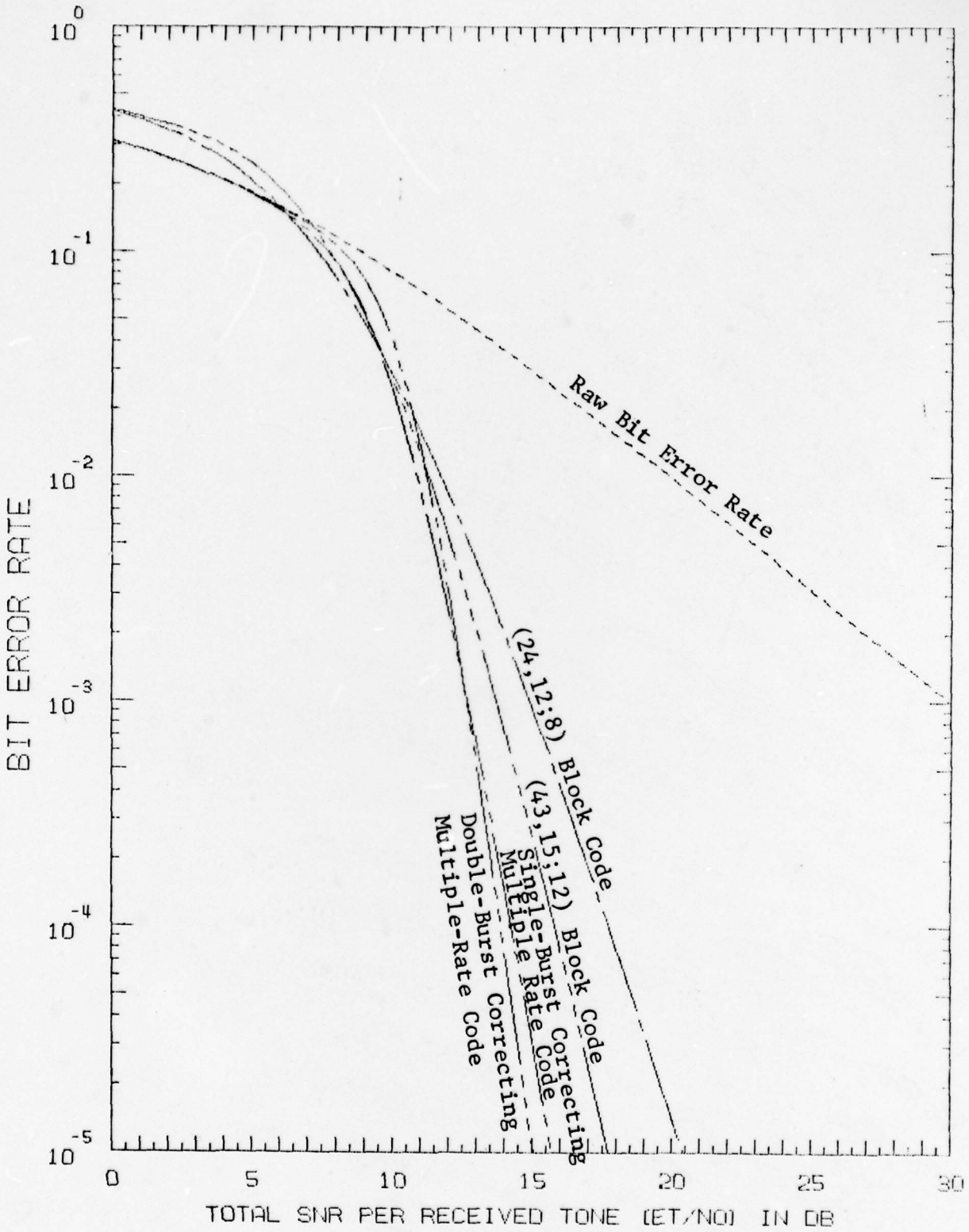


Figure 3 Performance Comparisons of a Hard Decoded (43,15;12) → (43,30;6) Multiple-Rate Code with Block Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (41 Tones/Frame)

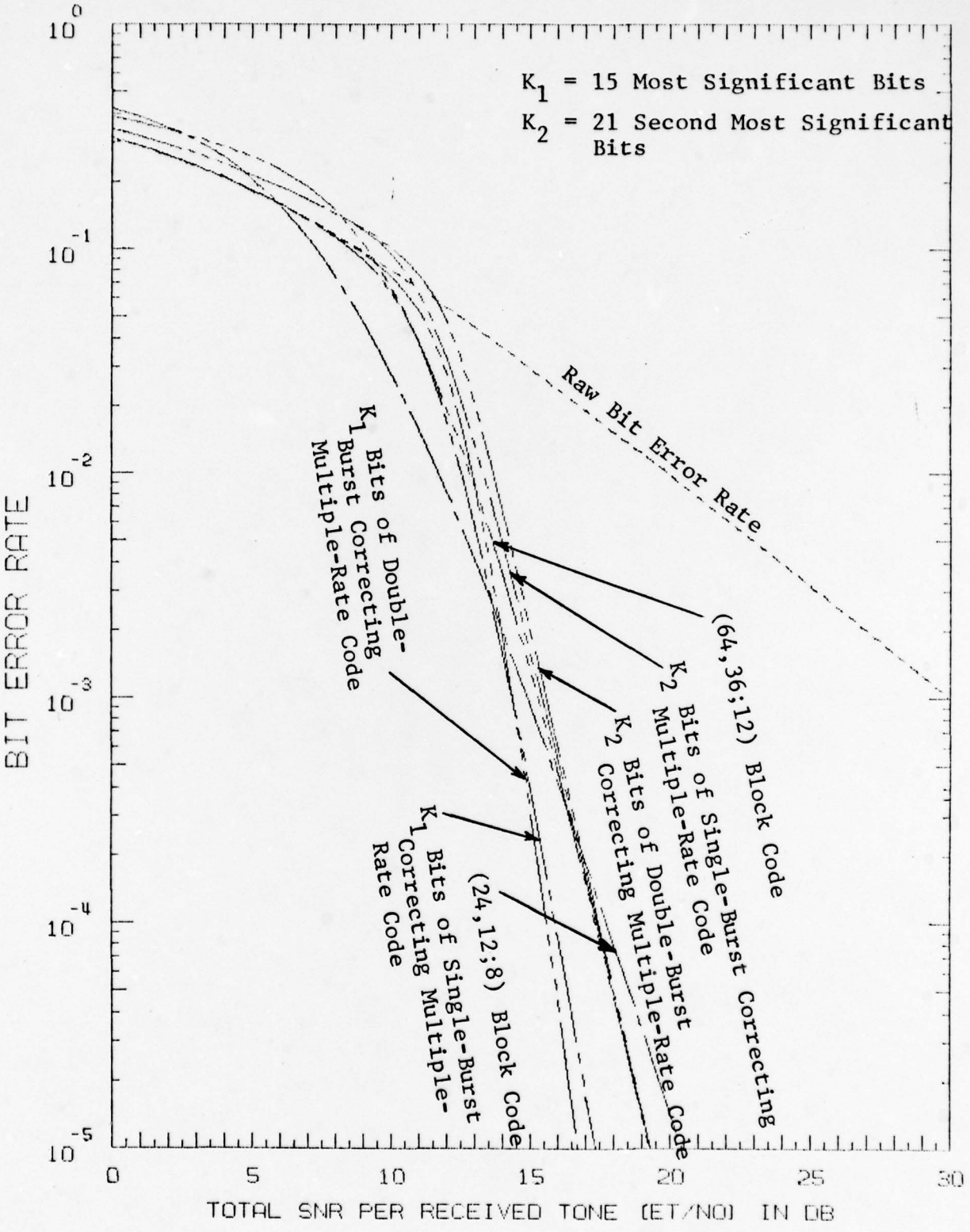


Figure 4 Performance Comparisons of a Hard Decoded (64,36;12) → (64,51;6) Multiple-Rate Code with Block Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (41 Tones/Frame)

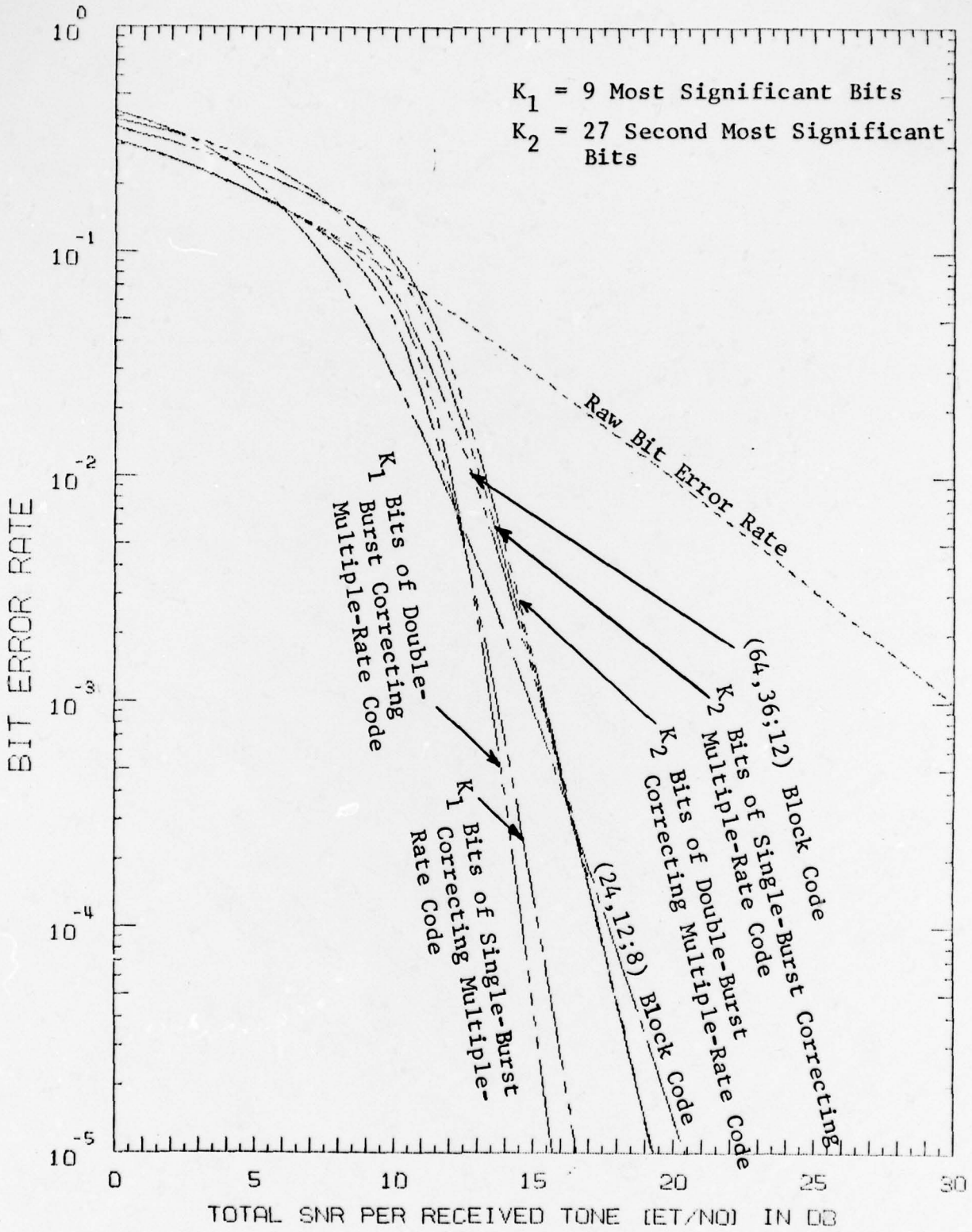


Figure 5 Performance Comparisons of a Hard Decoded (64,36;12) → (64,45;8) Multiple-Rate Code with Block Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (41 Tones/Frame)

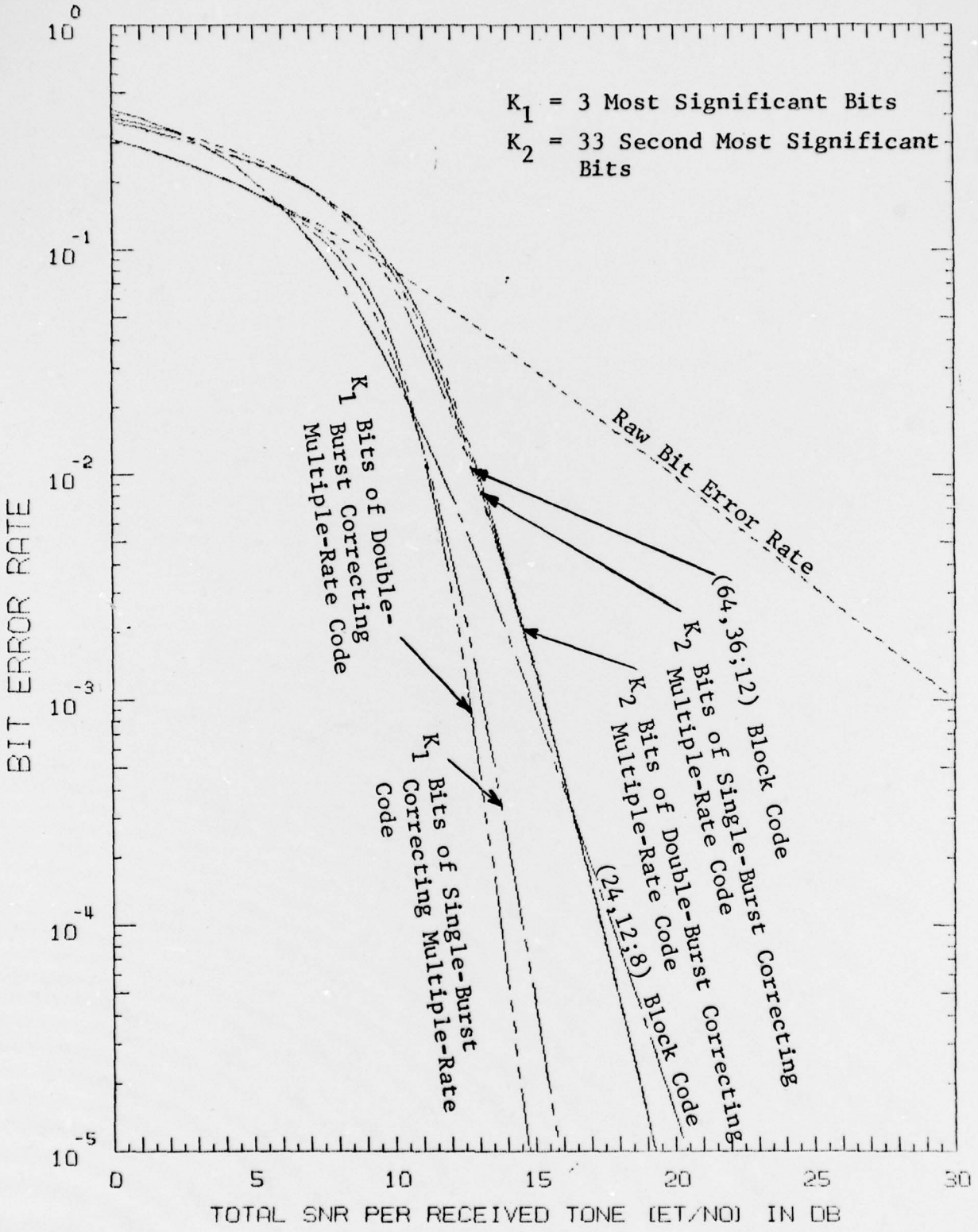


Figure 6 Performance Comparisons of a Hard Decoded (64,36;12) → (64,39;10) Multiple-Rate Code with Block Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (41 Tones/Frame)

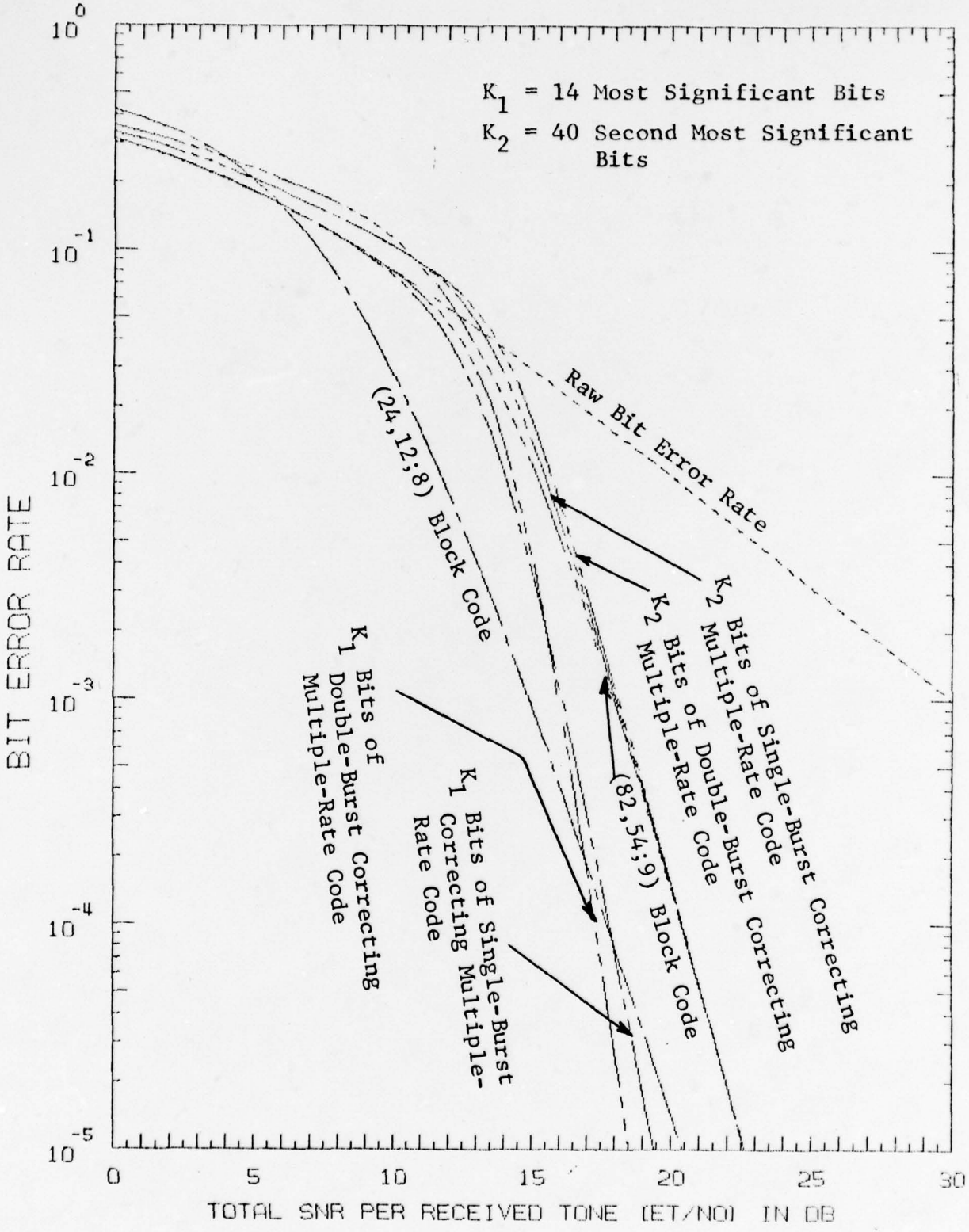


Figure 7 Performance Comparisons of a Hard Decoded (82,54;9) → (82,68;5) Multiple-Rate Code with Block Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (41 Tones/Frame)

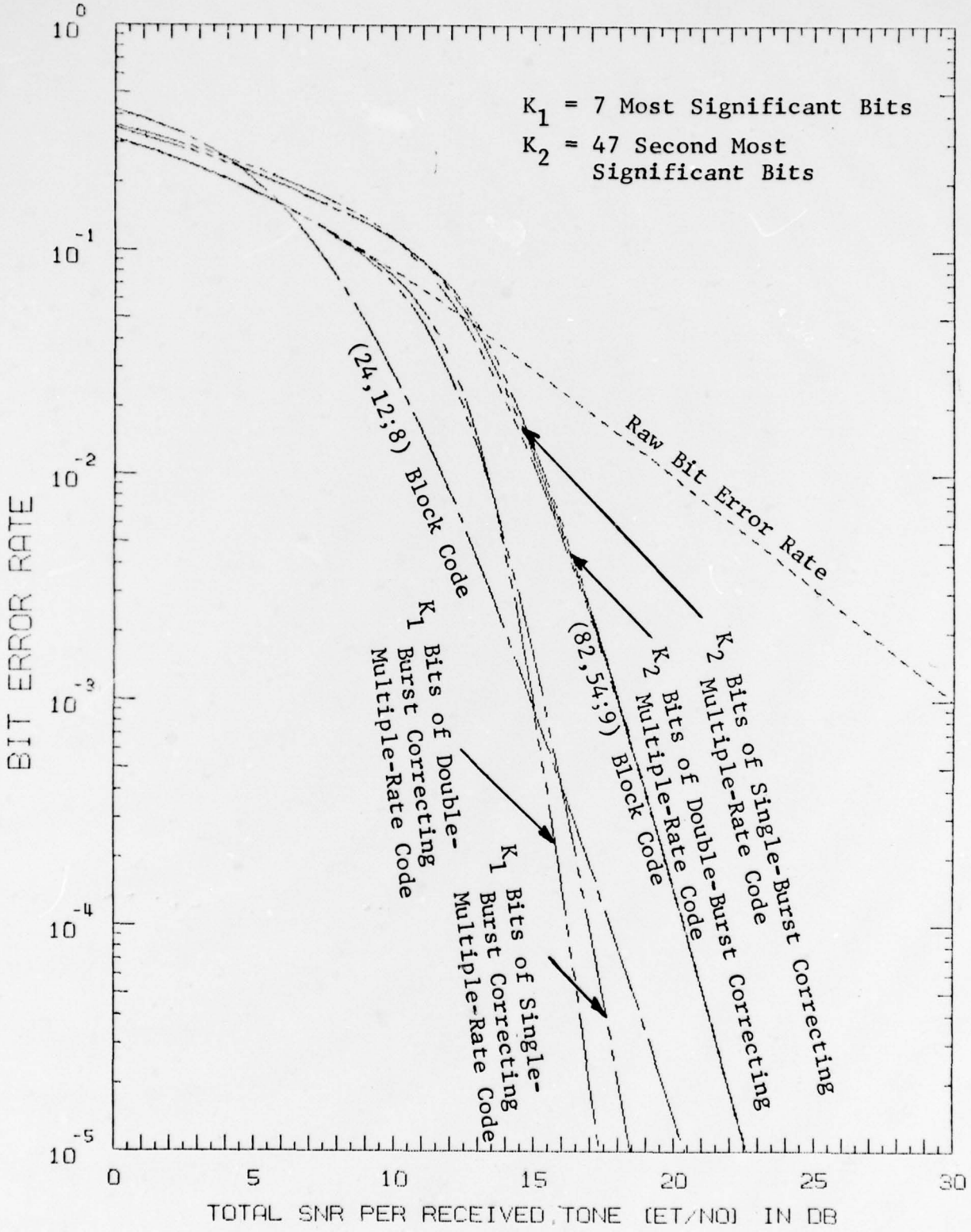


Figure 8 Performance Comparisons of a Hard Decoded (82,54;9) → (82,61;7) Multiple-Rate Code with Block Codes over a Slowly Fading HF Channel with 4-Phase DPSK Modulation (41 Tones/Frame)

B.2 Performance Comparisons of Shortened and Unshortened BCH (64,36;12) Codes with the Golay (24,12;8) Code for Voice Protection

In view of the closed hard decoding performance results obtained in Section B.1 for the (64,36;12) code and the Golay code, shortened and unshortened BCH (64,36;12) codes are evaluated and compared to the performance of two side-by-side Golay (24,12;8) codes. The BCH codes used are the (52,24;12) code, (55,27;12) code, (58,30;12) code, (61,33;12) code, and (64,36;12) code.

The hard decoded results are shown in Fig. 9. The (58,30;12) code performs approximately the same as the Golay code for a decoded bit error rate of 10^{-3}; but the BCH code provides the protection of 30 bits vs. 24 bits in the case of the Golay code. The two shorter BCH codes outperform the Golay code for a decoded bit error rate of 10^{-3}, and the two longer BCH codes protect more bits (33 bits and 36 bits vs. 24 bits for Golay codes). Therefore, for the hard decoding case, all five of these codes are strong competition for the Golay codes.

For the soft decoding case, none of these five BCH codes are expected to be comparable to the Golay code, as shown in Fig. 10. Therefore, when soft decoding is employed, the Golay code would probably yield the best performance. Detailed simulations for realistic HF channels are still required before firm conclusions can be drawn on the effectiveness of these coding techniques for digital speech transmissions.

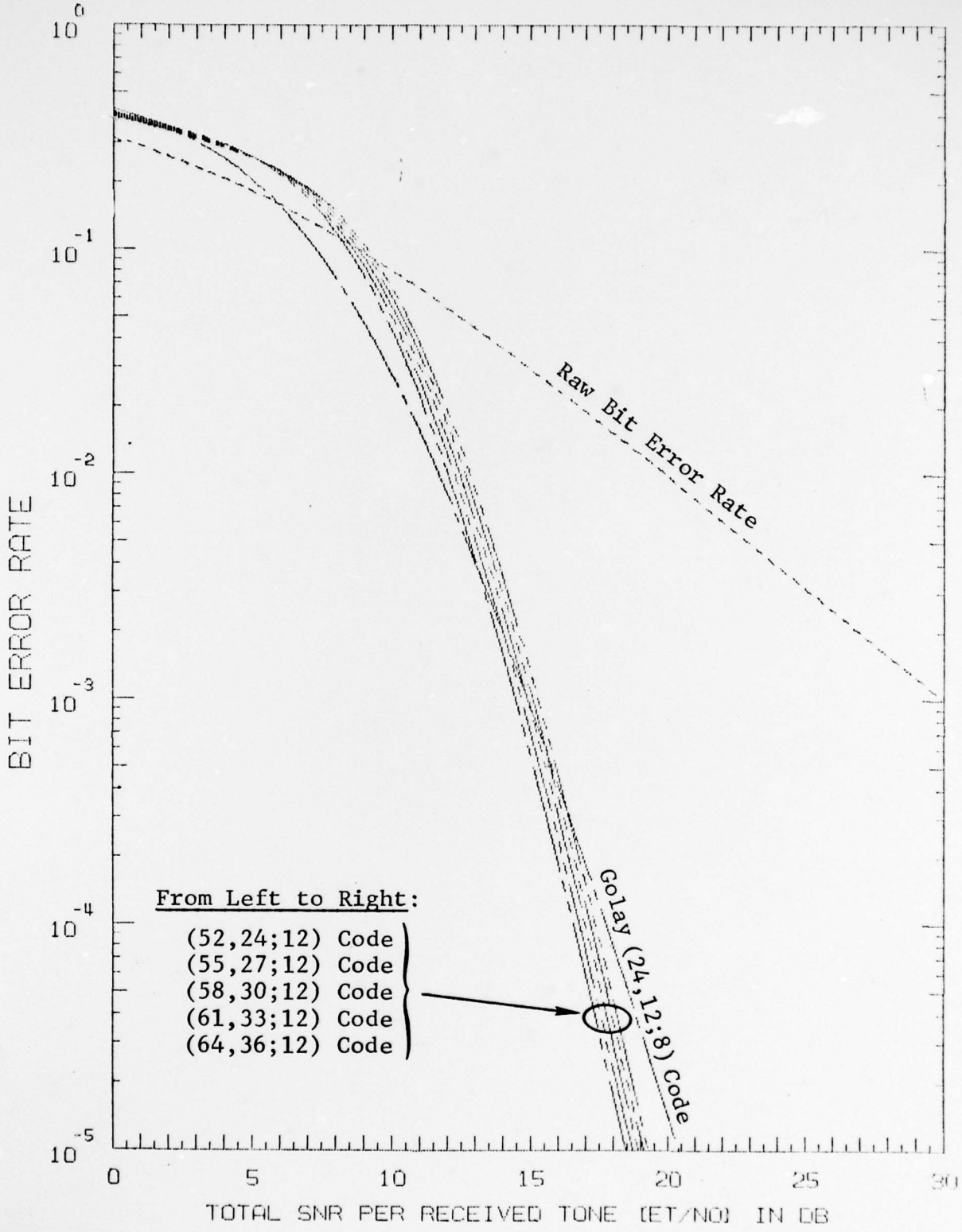


Figure 9 Performance Comparison of Hard Decoded Shortened and Unshortened BCH (64,36;12) Codes and the Golay (24,12;8) Code over a Slowly Fading HF Channel with 4-Phase DPSK Modulation

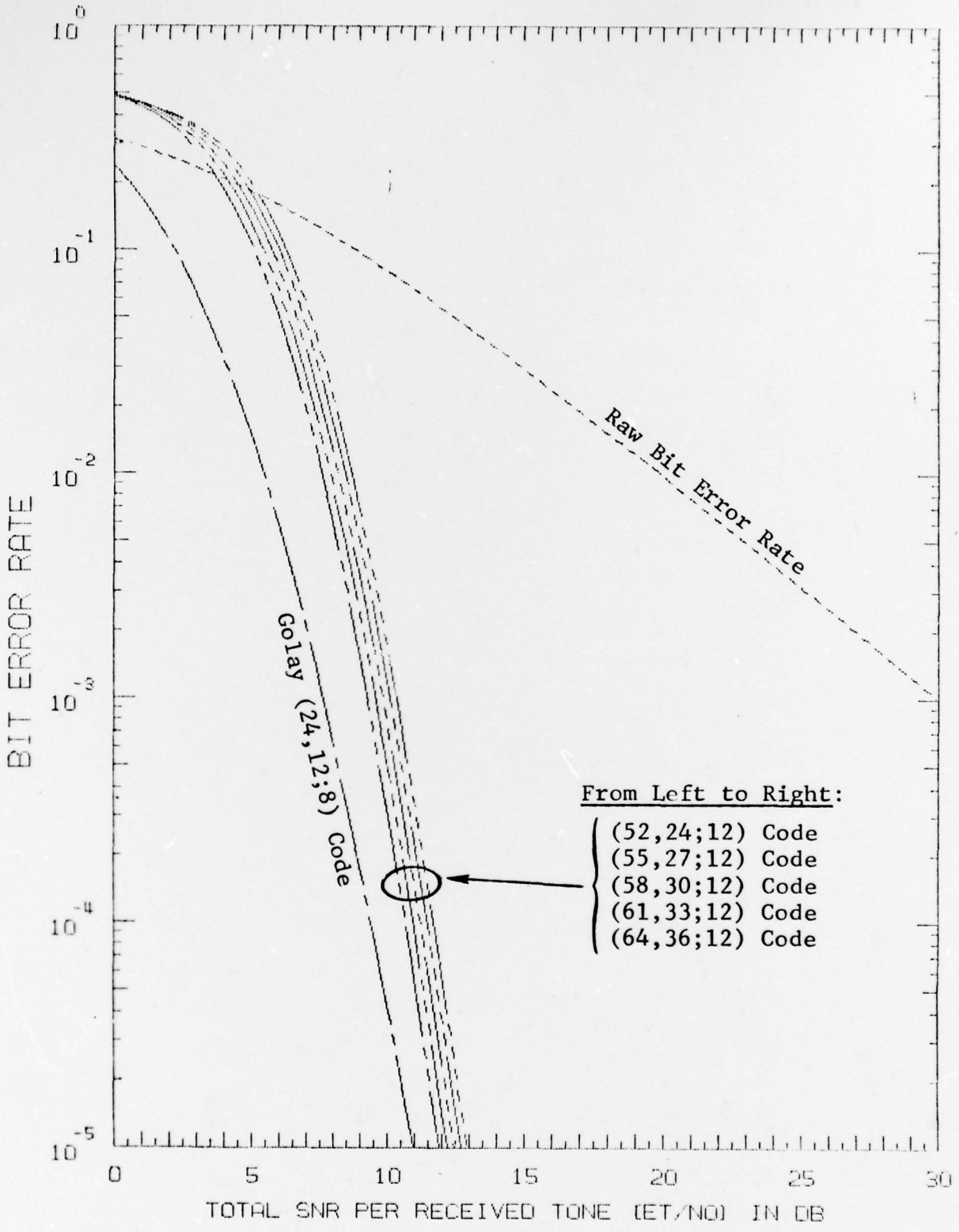


Figure 10 Estimated Performance of Soft Decoded Shortened and Unshortened BCH (64,36;12) Codes and the Golay (24,12;8) Code over a Slowly Fading HF Channel with 4-Phase DPSK Modulation

APPENDIX C

SIGNAL-TO-NOISE RELATIONSHIPS FOR HF MODEMS

It is quite important to establish proper definitions of signal-to-noise since this report compares results for different modulation formats such as the conventional 16-tone modems and an ANDVT modem designed with 39 tones. The modulation format for the preamble design is based on using only four parallel tones followed by a wideband pulse sequence. Thus, any definition of SNR must also be suitable for preamble evaluation.

In all cases it is meaningful to obtain results in terms of the ratio of the total received power to noise power density. This signal-to-noise density ratio (SNDR) is given by the ratio of the total received power, P, to the noise power per Hz of bandwidth, denoted by N_0, as

$$\text{SNDR} = \frac{P}{N_0} \tag{1}$$

For preamble evaluation in Section 2 results are given in terms of this SNDR. The above is called a signal-to-noise density ratio since the denominator is the noise density, rather than the more conventional noise power in the bandwidth of the signal. To avoid the ambiguity which results from different bandwidth definitions one may define the signal-to-noise ratio per information bit

$$\frac{E_b}{N_0} = \frac{P}{N_0 R} \tag{2}$$

where R is the actual data rate of the modem. For speech transmission R = 2400 b/s and thus Eqs. (1) and (2) can be related by

$$\frac{P}{N_0} = \left(\frac{E_b}{N_0} + 33.8\right)\text{dB} \text{ - for } R = 2400 \text{ b/s} \tag{3}$$

For digitized speech the modulated signal is approximately contained in a bandwidth of 2400 Hz, and thus, E_b/N_0 can also be viewed as the ratio of signal-to-noise power measured in a bandwidth of 2400 Hz.

In addition to obtaining results in terms of P/N_0 and E_b/N_0, which are easily related, it is customary for parallel tone evaluations to present results in terms of the average signal-to-noise ratio per tone defined by

$$\frac{E_t}{N_0} = \frac{E[r^2]}{E[n_c^2] + E[n_s^2]} \tag{4}$$

where r is the tone amplitude and n_c and n_s are the cosine and sine noise components at the output of the integrate-and-dump circuits. This SNR is also closely related to the signal-to-noise power measured in a bandwidth equal to that of the modulated signal.

To relate E_t/N_0 to E_b/N_0 we first note that a multiple tone received signal can be written as

$$y(t) = \sum_{i=1}^{N} r_i \cos(\omega_i t + \theta_i) + n(t) \tag{5}$$

where n(t) is assumed to be a white Gaussian noise process normalized such that $E[n(t)n(t+\tau)] = (N_0/2)\delta(\tau)$.

The cosine and sine components per tone are given by

$$y_{c_i} = 2\int_0^T y(t) \cos \omega_i t \, dt = r_i \cos \theta_i + n_{c_i} \tag{6}$$

$$y_{s_i} = 2\int_0^T y(t) \sin \omega_i t \, dt = r_i \sin \theta_i + n_{s_i}$$

Note, the multiple tones are orthogonal over the integration time T, which is less than a baud interval by the allowable guard time.

The average signal energy per tone is given by

$$E_t = E[r_i^2 \cos^2 \theta_i + r_i^2 \sin^2 \theta_i] = E[r_i^2] \tag{7}$$

and the energy per noise component is given by

$$E[n_{c_i}^2] = E[n_{s_i}^2] = N_0/2 \tag{8}$$

The total received power can be written as

$$P = \frac{NE_t}{T} \tag{9}$$

where N is the total number of tones and T is the integration time per tone. Thus, we can relate (1) and (2) to E_t/N_0 by the following:

$$\frac{P}{N_0} = \frac{N}{T}\left(\frac{E_t}{N_0}\right) \tag{10}$$

and

$$\frac{E_b}{N_0} = \frac{N}{TR}\left(\frac{E_t}{N_0}\right) \tag{11}$$

For a conventional 16 tone modem* operating at 2400 b/s, and including a Doppler correction tone of twice the amplitude, ($N_{effective} = 20$), equation (11) gives the relationship between E_b/N_0 and E_t/N_0 as

$$\frac{E_b}{N_0} = \frac{20}{\frac{1}{110}\ 2400}\left(\frac{E_t}{N_0}\right) \tag{12}$$

or

$$\frac{E_b}{N_0} = \left[\frac{E_t}{N_0} - .37\right] \text{ dB - 16 tone modem} \tag{13}$$

For an ANDVT modem operating at 2400 b/s with two Golay codes, using 39 tones and an integration of 18 ms, we have

$$\frac{E_b}{N_0} = \frac{39}{18 \times 10^{-3} \times 2400}\left(\frac{E_t}{N_0}\right) \tag{14}$$

---

*D. Chase, "A Combined Coding and Modulation Approach for Communication Over Dispersive Channels," IEEE Trans. on Communications, Vol. COM-21, No. 3, March 1973, page 159.

or

$$\frac{E_b}{N_0} = \left[\frac{E_t}{N_0} - .44\right] \text{ dB - 39 tone ANDVT modem} \qquad (15)$$

Thus, in this case, a comparison based on E_t/N_0 is essentially equivalent (adjustment is only .07 dB) to the fiar comparison based on E_b/N_0.

If the ANDVT modem is used without coding a data rate of 2400 bps can be achieved with 27 tones which are keyed every 22.5 ms with one of four phases (2 bits/tone). In this case the relationship between E_b/N_0 and E_t/N_0 is given by

$$\frac{E_b}{N_0} = \frac{27}{18 \times 10^{-3} \times 2400}\left(\frac{E_t}{N_0}\right) \qquad (16)$$

$$\frac{E_b}{N_0} = \left[\frac{E_t}{N_0} - 1.04\right] \text{ dB - 27 tone ANDVT modem} \qquad (17)$$

Note, by comparing Eqs. (13) and (17), without coding the ANDVT modem offers a 1.67 dB advantage in E_b/N_0 when compared to the 16-tone modem. This gain is due to the fact that the loss due to guard time is smaller for the ANDVT modem and also due to the Doppler tracking algorithm for the ANDVT modem which does not require an external tone.

The ANDVT modem results are all presented in terms of E_b/N_0 and the preamble results are presented in terms of P/N_0. Equation (3) results relate these two measures. The results for the KG synchronizing sequence are presented in terms of E_t/N_0,

which has the desirable property that guard time and code redundancy does affect this SNR measure. As a consequence of the low data rate used for the KG sync sequence, E_b/N_0 is considerably greater than E_t/N_0 for this case, and thus, it is more meaningful to make comparisons based on P/N_0. Equation (10) relates P/N_0 to E_t/N_0, which for a 39-tone preamble and integration of 18 ms yields

$$\frac{P}{N_0} = [E_t/N_0 + 33.58] \text{ dB - 39-tone KG preamble} \qquad (18)$$

In summary, the preamble, KG sync sequence, and multiple tone modems can be compared in terms of P/N_0, but by convention E_b/N_0 and E_t/N_0 are sometimes used. Equation (3) provides the appropriate conversion for the ANDVT modem results given in Section 4, and Eq. (18) provides the conversion for the KG sync results given in Section 3.

APPENDIX D

BIT AND BLOCK ERROR RATE COMPUTATION FOR LINEAR BLOCK CODES

A detailed analyses for obtaining a lower and an upper bound on decoded bit error rates for linear block codes are presented. The tightness of these bounds are discussed. An optimistic approximation to the bit error rate of soft-decoded linear block codes is then given. Finally, the block error rate expression for hard-decoded linear block codes and an approximation to soft-decoded block error rate is obtained.

D.1 A Lower Bound on Hard-Decoded Bit Error Rate for Linear Block Codes

Let p_b denote the decoded bit error rate for an (N,K;d) block code and let

$$e = \lfloor (d-1)/2 \rfloor \tag{1}$$

Then the expected number of errors per decoded block is Np_b. This expected number of decoded errors can be lower bounded by

$$Np_b > \sum_{i=e+1}^{d-1} \binom{N}{i} i p^i (1-p)^{N-i} + d \sum_{i=d}^{d+e} \binom{N}{i} p^i (1-p)^{N-i}$$

$$+ \sum_{i=d+e+1}^{N} \binom{N}{i} (i-e) p^i (1-p)^{N-i} \tag{2}$$

where p is the raw bit error rate. This lower bound is obtained by first noting that all possible patterns of e or fewer errors are corrected since the block code is an e-error correcting code. Error patterns with e+1 to d-1 errors are assumed to be detected, thus no additional errors are introduced by the decoder. When there are d to d+e errors, a lower bound is achieved by assuming that these patterns are all decoded into d errors. Finally, if there are i > d+e errors, the decoder is assumed, optimistically, to leave i-e decoded errors. Therefore, Eq. (1) is strictly an inequality. Dividing both sides by N, Eq. (2) becomes

$$p_b > \frac{1}{N} \sum_{i=e+1}^{d-1} \binom{N}{i} i p^i (1-p)^{N-i} + \frac{d}{N} \sum_{i=d}^{d+e} \binom{N}{i} p^i (1-p)^{N-i}$$

$$+ \frac{1}{N} \sum_{i=d+e+1}^{N} \binom{N}{i} (i-e) p^i (1-p)^{N-i} \qquad (3)$$

which provides the lower bound on the decoded bit error rate for hard decoding.

D.2 An Upper Bound on Hard-Decoded Bit Error Rate for Linear Block Codes

In this section, an upper bound on the hard decoded bit error rate for linear block codes will be obtained. This bound is independent of the weight distribution of a block code. The tightness of the bound is to be discussed.

For the case that d is odd, the simple bound is:

$$p_b \leq \frac{1}{N} \sum_{i=e+1}^{N} \binom{N}{i}(i+e)p^i(1-p)^{N-i} \quad \text{for} \quad d \text{ odd} \tag{4}$$

This can be seen as follows: whenever there are e+1 or more errors, the decoder is assumed to produce extra e errors. However, the decoder, in reality, produces e or fewer errors. Thus Eq. (4) is indeed an upper bound. It can be seen from Eq. (3) and Eq. (4) that

$$(p_b)_{\text{lower bound}} > \frac{d-e}{d} (p_b)_{\text{upper bound}} \quad \text{for} \quad d \text{ odd} \tag{5}$$

In other words, for d odd,

$$\frac{d-e}{d} (p_b)_{\text{upper bound}} \leq p_b < (p_b)_{\text{upper bound}} \tag{6}$$

where

$$\frac{1}{2} < \frac{d-e}{d} = \frac{e+1}{2e+1} \leq \frac{2}{3} \tag{7}$$

When the number of code words of minimum weight d is known to be w_d, the upper bound given in Eq. (4) can be tightened to:

$$p_b \leq \frac{\Delta d + (1-\Delta)(e+1)}{N} \binom{N}{e+1} p^{e+1}(1-p)^{N-e-1}$$

$$+ \frac{1}{N} \sum_{i=e+2}^{N} \binom{N}{i}(i+e)p^i(1-p)^{N-i} \quad \text{for} \quad d \text{ odd} \tag{8}$$

where

$$\Delta = w_d \binom{d}{e+1} / \binom{N}{e+1} \tag{9}$$

Note that when there are e+1 errors in a code word, there is a probability Δ that the pattern is not detected, which results in d decoded errors, and a probability 1 - Δ that the pattern is detected and the e+1 errors remain unchanged. The expression for Δ is obtained by noting that, for each code word of minimum distance d away from the transmitted code word, $\binom{d}{e+1}$ patterns of e+1 errors are undetected. [There are $\binom{d}{e+1}$ possible error patterns of weight e+1.] If Eq. (8) is employed, the upper bound is very tight, since the first term, which is the dominating term for small values of p, is the exact expression for the decoded bit error rate. Unfortunately, for most linear codes, the value of w_d is unknown. Therefore, Eq. (4) is employed to obtain all of the performance curves in this report.

For the case that d is even, the upper bound on the decoded bit error rate is given by

$$p_b \leq \frac{e+1}{N} \binom{N}{e+1} p^{e+1} (1-p)^{N-e-1} + \frac{1}{N} \sum_{i=e+2}^{N} \binom{N}{i} (i+e) p^i (1-p)^{N-i}$$

$$\text{for} \quad d \text{ even} \tag{10}$$

The first term of Eq. (10) is due to the fact that all possible $e+1 = \frac{d}{2}$ errors can be detected for even d. Equation (10) is as tight as Eq. (8) for odd d. However, there is no need to know the value of w_d in evaluating Eq. (10), since the dominating term

of Eq. (10) is identical to the dominating term of Eq. (3), the lower bound. Thus, for small values of p, the upper and lower bounds are almost indistinguishable. [Also, Eq. (6) can be easily shown to be valid for even d when Eq. (10) is employed to compute the upper bound.]

D.3 An Estimate of Soft Decoding Performance for Block Codes

Unlike the case of hard decoding, the exact performance of soft-decoded block codes cannot be bounded accurately. Note that the performance depends not only on the code structure and the raw bit error rate but also on the more detailed channel characteristics, such as the confidence level of each received bit. However, a less accurate estimate than those obtained for the hard decoding can be easily computed by assuming that all the possible hard error patterns of weight d-1 or less can be corrected. Since a soft decoder can usually only correct a fraction of hard error patterns with weight d-1 or less, this estimate obtained for the soft decoding may be somewhat too optimistic for some codes. Nevertheless, this is a simple and easy way to obtain an approximation to lower bounds for the code performance. Furthermore, asymptotically, an optimum soft decoder can correct (d-1) errors [1]. [Note that for a more conservative estimate of the soft decoding performance the code is assumed to be able to correct d-i or fewer hard errors with

$$1 \leq i < \left\lfloor \frac{d-1}{2} \right\rfloor \tag{11}$$

A properly chosen value of i may yield a good estimate for the soft decoding performance for a meaningful range of signal-to-noise ratios. One way to determine i is to simulate the

performance by computer at relatively low signal-to-noise ratios. Once the value i is determined, it is rather simple to arrive at the estimate of soft decoding performance at higher signal-to-noise ratios.]

D.4 Block Error Rate for Linear Block Codes

For hard decoding, a block error results if, and only if, there are e+1 or more errors per code block. Thus the block error rate P_b is given by

$$P_b = \sum_{i=e+1}^{N} \binom{N}{i} p^i (1-p)^{N-i} \tag{12}$$

For small values of raw bit error rate p,

$$P_b \approx \binom{N}{e+1} p^{e+1} (1-p)^{N-e-1} \approx \frac{N}{e+1} p_b \tag{13}$$

An optimistic estimate for soft-decoded block error rate is given by

$$P_b \approx \sum_{i=d}^{N} \binom{N}{i} p^i (1-p)^{N-i} \tag{14}$$

which is obtained by assuming that all d-1 or fewer hard errors can be corrected. At high signal-to-noise ratio

$$P_b \approx \binom{N}{d} p^d (1-p)^{N-d} \approx \frac{N}{d} p_b \tag{15}$$

APPENDIX D

REFERENCE

[1] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," IEEE Transactions on Information Theory, Vol. IT-18, No. 1, January 1972.

APPENDIX E

DOPPLER TRACKING DURING DATA TRANSMISSION

E.1 Introduction

In order to track out the Doppler offset while data is being transmitted, a Doppler discriminator is needed to provide an appropriate error signal for driving the voltage-controlled oscillator that tracks the Doppler. Conventional approaches use an unmodulated tone and conventional frequency discriminator for a Doppler discriminator. This approach wastes channel capacity and power and, in addition, does not take advantage of the signal diversity present in the received waveform. In this appendix we shall discuss a Doppler discriminator that removes these objections. However, this discriminator introduces new requirements; the frame sync is assumed to be operating and the average error rate needs to be sufficiently below 1/2.

E.2 Formulation of Received Signal and Decision Variables

The transmitted signal consists of a set of frequency-division-multiplexed subcarriers, each carrying 4-DPSK data. Thus, the k'th modulated subcarrier can be represented in the (complex envelope) form

$$d_k(t) = e^{j2\pi kFt} \sum e^{j\varphi_{k\ell}} \text{ Rect } \left(\frac{t - \ell T}{T}\right) \tag{1}$$

where F is the frequency spacing between subcarriers, T is the pulse duration,

$$\text{Rect}\left(\frac{t}{T}\right) = \begin{cases} 1 \; ; \; |t| \leq \frac{T}{2} \\ \\ 0 \; ; \; |t| > \frac{T}{2} \end{cases} \tag{2}$$

and, in accordance with the 4-DPSK format,

$$e^{j\varphi_{k,\ell+1}} \; e^{-j\varphi_{k,\ell}} = \pm \frac{1}{\sqrt{2}} \pm j\frac{1}{\sqrt{2}} \tag{3}$$

The received signal corresponding to the transmission of $d_k(t)$ will be distorted by the multipath in the vicinity of the transitions between pulses. However, in the receiver, a synchronized time gate passes only the central, nondistorted portion of each pulse. Any fading present at the subcarrier location still appears on the received gated pulse. Thus, if T(f,t) is the time-variant transfer function of the channel, the complex fading at the subcarrier frequency kF is given by T(kF,t). Thus, after the time gate, the received modulated subcarrier (in the absence of noise) can be represented in the form

$$w_k(t) = T(kF,t) \; e^{j2\pi kFt} \sum e^{j\varphi_{k\ell}} \text{ Rect}\left(\frac{t - \ell T}{T_g}\right) \tag{4}$$

where T_g is the duration of the time gate.

The received noise after the time gate is given by

$$m(t) = \sum \text{Rect}\left(\frac{t - \ell T}{T_g}\right) \; n(t) \tag{5}$$

where n(t) is the received noise before the time gate.

The integrate-and-dump operation for the p^{th} baud of the k^{th} subcarrier is given by

$$S_{kp} = \int_{pT-T_g/2}^{pT+T_g/2} e^{-j2\pi kFt} \; [w(t) + m(t)] \; dt \tag{6}$$

where

$$w(t) = \sum_{k=1}^{K} w_k(t) \tag{7}$$

where K is the number of subcarriers.

Because the subcarrier separation frequency F is set equal to the reciprocal gate duration,

$$F = \frac{1}{T_g} \tag{8}$$

the integration in (6) separates out a signal contribution due to the p^{th} band and k^{th} subcarrier above; to wit,

$$S_{kp} = \int_{pT-T_g/2}^{pT+T_g/2} T(kF,t) \; e^{j\varphi_{kp}} \; dt + \int_{pT-T_g/2}^{pT+T_g/2} e^{-j2\pi kFt} \; n(t) \; dt \tag{9}$$

This may be expressed in the form

$$S_{kp} = e^{j\varphi_{kp}} U_{kp} \tag{10}$$

where

$$U_{kp} = T_{kp} + N_{kp} \tag{11}$$

in which

$$T_{kp} = \int_{pT-T_g/2}^{pT+T_g/2} T(kF,t)\, dt \tag{12}$$

and

$$N_{kp} = \int_{pT-T_g/2}^{pT+T_g/2} e^{-j2\pi kFt}\, e^{-j\varphi_{kp}}\, n(t)\, dt \tag{13}$$

Assuming white complex Gaussian noise and complex Gaussian fading statistics, T_{kp}, N_{kp}, and U_{kp} are complex Gaussian variables with

$$\overline{T^*_{kp} N_{rs}} = 0 \tag{14}$$

$$\overline{N^*_{kp} N_{rs}} = \begin{cases} 0 & ; k \neq p,\ r \neq s \\ 2 N_0 T_g & ; k = p,\ r = s \end{cases} \tag{15}$$

where N_0 is the real one-sided noise power density.

Under the reasonable assumption that T(kF,t) will change little over one gate duration, we see that

$$T_{kp} \approx T(kF, pT) \tag{16}$$

The SNR at the output of the integrate-and-dump is

$$\rho = \frac{\overline{|T_{kp}|^2}}{\overline{|N_{kp}|^2}} = \frac{P}{K N_0 T_g} \tag{17}$$

where P is the average received signal power for all tones and K is the number of tones.

The decision variable used in 4-DPSK modulation is

$$\begin{aligned} D_{kp} &= S^*_{kp} S_{k,p+1} = R_{kp} + j Q_{kp} \\ &= e^{j(\varphi_{k,p+1} - \varphi_{k,p})} U^*_{kp} U_{k,p+1} \end{aligned} \tag{18}$$

where R_{kp} and Q_{kp} are the real and imaginary part of D_{kp}, respectively.

Let the input quaternary data be encoded into two binary variables associated with the sign of the real and imaginary parts of $\exp(j\varphi_{k,p+1} - j\varphi_{k,p})$ [see (3)]. Thus,

$$e^{j(\varphi_{k,p+1} - \varphi_{k,p})} = \frac{1}{\sqrt{2}} (r_{kp} + j\, q_{kp}) \tag{19}$$

where

$$r_{kp} = \pm 1 \tag{20}$$

$$q_{kp} = \pm 1 \tag{21}$$

Two pairs of binary decisions, $\hat{r}_{kp}$ and $\hat{q}_{kp}$, are made, as follows:

$$\hat{r}_{kp} = \text{Sgn}[R_{kp}] \tag{22}$$

$$\hat{q}_{kp} = \text{Sgn}[Q_{kp}] \tag{23}$$

where

$$\text{Sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0 \end{cases} \tag{24}$$

Under the ideal conditions of very slow fading and no noise,

$$E_{kp} = U^*_{kp} U_{k,p+1} \longrightarrow |T(kF,pT)|^2 \text{ ; slow fading and no noise} \tag{25}$$

and we see that

$$\hat{r}_{kp} = r_{kp} \tag{26}$$

$$\hat{q}_{kp} = q_{kp} \tag{27}$$

i.e., no errors. More generally, E_{kp} has a nonzero phase angle α_{kp},

$$E_{kp} = |E_{kp}| \, e^{j\alpha_{kp}} \tag{28}$$

and

$$\hat{r}_{kp} = \text{Sgn}\left[r_{kp} \cos \alpha_{kp} - q_{kp} \sin \alpha_{kp}\right] \tag{29}$$

$$\hat{q}_{kp} = \text{Sgn}\left[r_{kp} \sin \alpha_{kp} + q_{kp} \cos \alpha_{kp}\right] \tag{30}$$

The phase α_{kp} is a random variable due both to the additive noise and the channel fading. In addition, it has a nonzero average value due to the presence of nonzero instantaneous frequency of the received subcarrier. Only if $|\alpha_{kp}|$ exceeds 45^{o} will errors occur.

Using (16) and noting that the noise components of U_{kp} and $U_{k,p+1}$ are statistically independent (15), we can show that the mean value of α_{kp} is given by

$$\overline{\alpha_{kp}} \approx 2\pi \nu T \tag{31}$$

where ν is the uncompensated Doppler shift of the received subcarrier. A typical value of residual Doppler error may be 1 Hz. With T = 22.5 ms, this produces a value of $\overline{\alpha_{kp}}$ of 0.14 radian, or around 8^{o}.

E.3 Doppler Discriminator

The proposed Doppler discriminator utilizes the decision variables R_{kp} and Q_{kp}. In essence, a quantity proportional to the Doppler error is estimated on each pair of bauds for all subcarriers and averaged over several successive pairs of bauds. The basic estimate for a single pair is given by

$$\epsilon_{kp} = \left(|Q_{kp}| - |R_{kp}|\right) \text{Sgn}[R_{kp}] \text{ Sgn}[Q_{kp}] \tag{32}$$

Let us analyze this estimate. Using (28), (18), (19), (22), and (23),

$$\epsilon_{kp} = \frac{1}{\sqrt{2}} E_{kp}\Big(|r_{kp} \sin\alpha_{kp} + q_{kp} \cos\alpha_{kp}| - |r_{kp} \cos\alpha_{kp} - q_{kp} \sin\alpha_{kp}|\Big) \hat{r}_{kp}\, \hat{q}_{kp} \tag{33}$$

Assuming that no error has been committed, $|\alpha_{kp}| < 45^{o}$, we see that

$$|r_{kp} \sin\alpha_{kp} + q_{kp} \cos\alpha_{1]}| = \begin{cases} r_{kp} \sin\alpha_{kp} + q_{kp} \cos\alpha_{kp} \; ; \; q_{kp} > 0 \\ \\ -r_{kp} \sin\alpha_{kp} - q_{kp} \cos\alpha_{kp} ; \; q_{kp} < 0 \end{cases} \tag{34}$$

$$|r_{kp} \cos\alpha_{kp} - q_{kp} \sin\alpha_{kp}| = \begin{cases} r_{kp} \cos\alpha_{kp} - q_{kp} \sin\alpha_{kp} \; ; \; r_{kp} > 0 \\ \\ -r_{kp} \cos\alpha_{kp} + q_{kp} \sin\alpha_{kp} ; \; r_{kp} < 0 \end{cases} \tag{35}$$

Using (34) and (35) in (33) and assuming no error in binary decisions, we find that

$$\epsilon_{kp} = \sqrt{2}\, E_{kp} \sin\alpha_{kp} \tag{36}$$

In the absence of noise and assuming a reasonable Doppler error,

$$\epsilon_{kp} \approx \sqrt{2}\, |T(kF, pT)|\, |T\Big(kF, (p+1)T\Big)|\; 2\pi\, T\, \nu \tag{37}$$

Summing over all frames we obtain

$$\epsilon_p = \sum_{k=1}^{K} \epsilon_{kp} = \sqrt{2} \sum_{k=1}^{K} E_{kp} \sin \alpha_{kp} \tag{38}$$

Note that weak subcarriers have a corresponding small contribution to the error signal. The actual error signal used would involve a time average of ϵ_p over the fading with an appropriate discrete filter,

$$\langle \epsilon \rangle_p = \sum h_\ell \, \epsilon_{p-\ell} \tag{39}$$

where $\{h_\ell\}$ is the discrete filter impulse response.

A detailed analysis of the effects of bit errors has not been conducted due to time limitations. At this point it is believed that, with reasonable amounts of averaging time, high error rates, well beyond those at which intelligibility has been lost, can be tolerated.