Coding For Frequency Hopped Spread Spectrum Satellite Communications

Final Report
Period Covered: October 13 1988 to March 31 1989

Prepared for
The Department of Communications of Canada
under DSS Contract No. 36001-8-3529/01-SS

Department of Electrical and Computer Engineering

THE UNIVERSITY OF VICTORIA
P.O. BOX 1700, VICTORIA, B.C. CANADA V8W 2Y2

Coding For Frequency Hopped Spread Spectrum Satellite Communications

Final Report
Period Covered: October 13 1988 to March 31 1989

Prepared for
The Department of Communications of Canada
under DSS Contract No. 36001-8-3529/01-SS

by

V.K. Bhargava, I.F. Blake[1] T.A. Gulliver, G. Li, Q. Wang and B. Weeks
Department of Electrical and Computer Engineering
University of Victoria
P.O. Box 1700
Victoria, B.C.
Canada V8W 2Y2

Scientific Authority: L.J. Mason

Technical Report ECE-89-1

April 15, 1989

[1]Professor Ian Blake is with the University of Waterloo and is a consultant to the University of Victoria on this project.

Abstract

A modified self-normalizing combiner is presented and its performance is analyzed under partial-band jamming. This is compared with other non-linear combining schemes.

Continuing on our previous work, the throughput performance of a coded Fast Frequency Hopped M-ary Frequency Shift Keying (FFH/MFSK) system with a fixed hop rate is evaluated using the cutoff rate argument. The analysis upperbounds the gains which can be realized using coding for various system parameters.

As a prelude to the study of coding for Slow Frequency Hopped Differential Phase Shift Keying (SFH/DPSK), we derive the probability distribution of DPSK in tone interference.

A comparison of two powerful DSP integrated circuit processors for implementation of a decoder for the (127,99) BCH code is presented as an Appendix.

Suggestions for future work include investigation of coding for SFH/DPSK; error control coding to alleviate very high error rate situations; implementation aspects of CODECS and a theoretical investigation of communications over an intentional interference channel.

Contents

List of Figures

Chapter 1

Introduction

In previous contracts, the use of various types of channel coding were studied to improve the jamming resistance of satellite communications using fast frequency hopping [1]. Coding for slow frequency hopping systems has received little attention up to this point.

In this annual report, we present the work performed during the period October 13, 1988 to March 31, 1989. The majority of our efforts has centered upon slow frequency hopping systems.

1.1 Plan of the Report

The plan of the report is as follows. In Chapter 2 a modified self-normalizing combiner is presented and its performance is analyzed under partial-band jamming.

Chapter 3 presents an analysis of the throughput performance of coded FFH/MFSK with a fixed hop rate based on the cutoff rate.

Chapter 4 is a prelude to the study of coding for slow frequency hopped DPSK (SFH/DPSK). The effects of jamming without coding are analyzed. The main result is the derivation of the probability distribution of DPSK in tone interference with applications to SFH/DPSK.

Chapter 5 provides some directions for future work in the areas of:

1. coding for SFH/DPSK,
2. coding for high channel error rate situations, and
3. identifying and analyzing various approaches to the general problem of communicating over an intentional interference channel.

As an adjunct to the report, a comparison of the Motorola DSP56000 and the Texas Instruments TMS320C25 Digital Signal Processors for implementation of a four error correcting BCH Error Control Code Decoder is presented as an Appendix.

Chapter 2

A Modified Self-Normalizing Combiner and Its Performance Analysis under Partial-Band Noise Jamming

2.1 Introduction

In frequency-hopped MFSK systems, diversity combining can be used to combat partial-band noise (PBN) jamming. Several forms of diversity combining have been proposed and their performance under jamming investigated [2,3,4,5]. These include linear combining, adaptive gain control combining, clipping combining, self-normalizing combining and product combining. Here these schemes are described briefly. We consider a noncoherent FFH/FSK system with a square-law detector and diversity, i.e., the outputs of the square-law envelope detectors are combined for different hops. We assume that the system is interfered by PBN jamming.

Linear Combining This scheme is depicted in Fig. 2.1. In this method, the detector outputs are combined by direct summing. Gong[3] and Lee et. al.[6] have shown that there is no diversity improvement for the square-law linear combining receiver on the

partial-band noise jamming channel.

Adaptive Gain Control (AGC) Combining The structure of this scheme is given in Fig. 2.2. It requires that side information regarding the noise level in each hop be available. The detector outputs for each hop are normalized with the noise variance of the hop. The normalized outputs of the detectors are then summed together. Some diversity gain can be obtained by this nonlinear combining scheme [7].

Self-Normalizing Combining This scheme is shown in Fig. 2.3. Instead of requiring side information, the outputs of the detectors for each hop are normalized with the sum of the outputs of the detectors in all channels for that hop.

Product Combining In this case, the product of the outputs of the detectors for all hops is used for the combining [4]. This scheme also does not require side information. Its structure is depicted in Fig. 2.4.

Clipping Combining This is shown in Fig. 2.5. The detector outputs are first passed through a clipper, (or soft limiter) then summed together. It is known that AGC combining has a better anti-PBN jamming performance than clipping combining. However, this performance advantage become small when the diversity is large [7].

Except for the first, all the above described combining strategies are nonlinear. It is known that linear combining cannot combat PBN jamming effectively. Instead, due to the noncoherent combining loss, the performance is degraded with increasing diversity. Thus our main interest is in nonlinear combining schemes [3].

In nonlinear combining, the AGC and clipper types have relatively better performance. However, the AGC combiner requires knowledge of the variance of the noise in each hop to the weight of the hop. To determine the clipping threshold in the clipper combiner, the bit energy to thermal noise ratio should be known [5]. Both of these combiners,

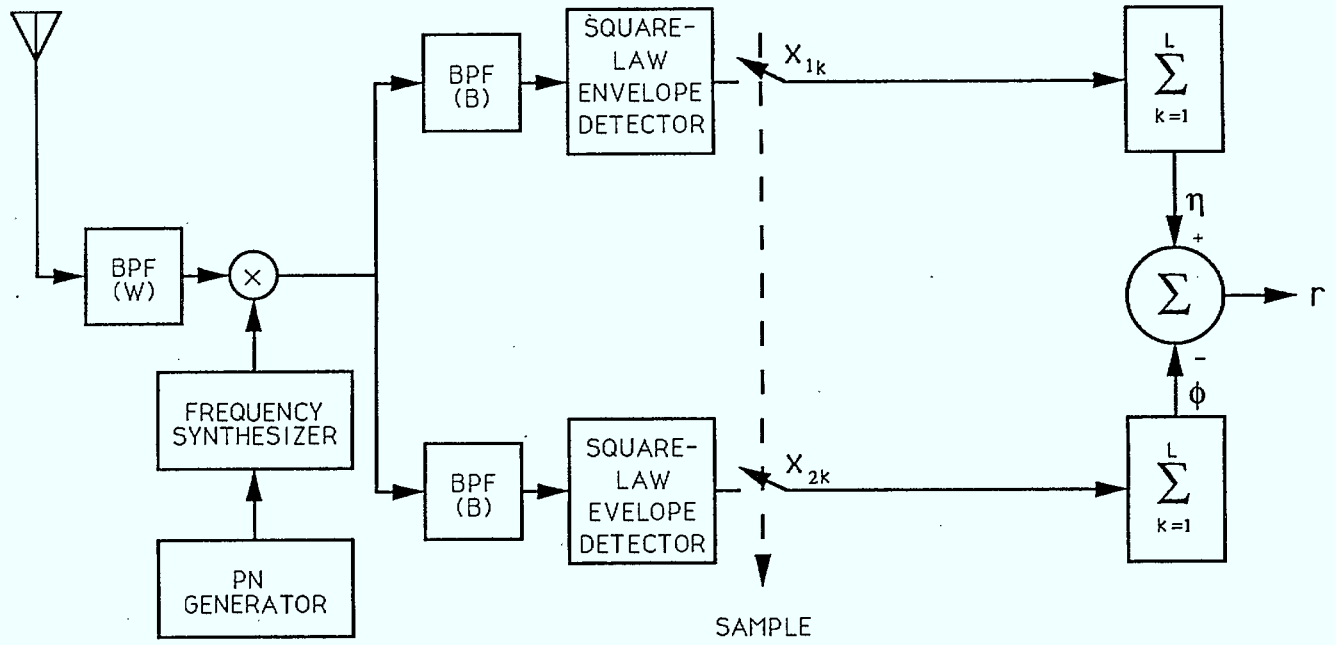

Figure 2.1: Linear Combining Scheme for a FFH/BFSK system with L hops/bit.

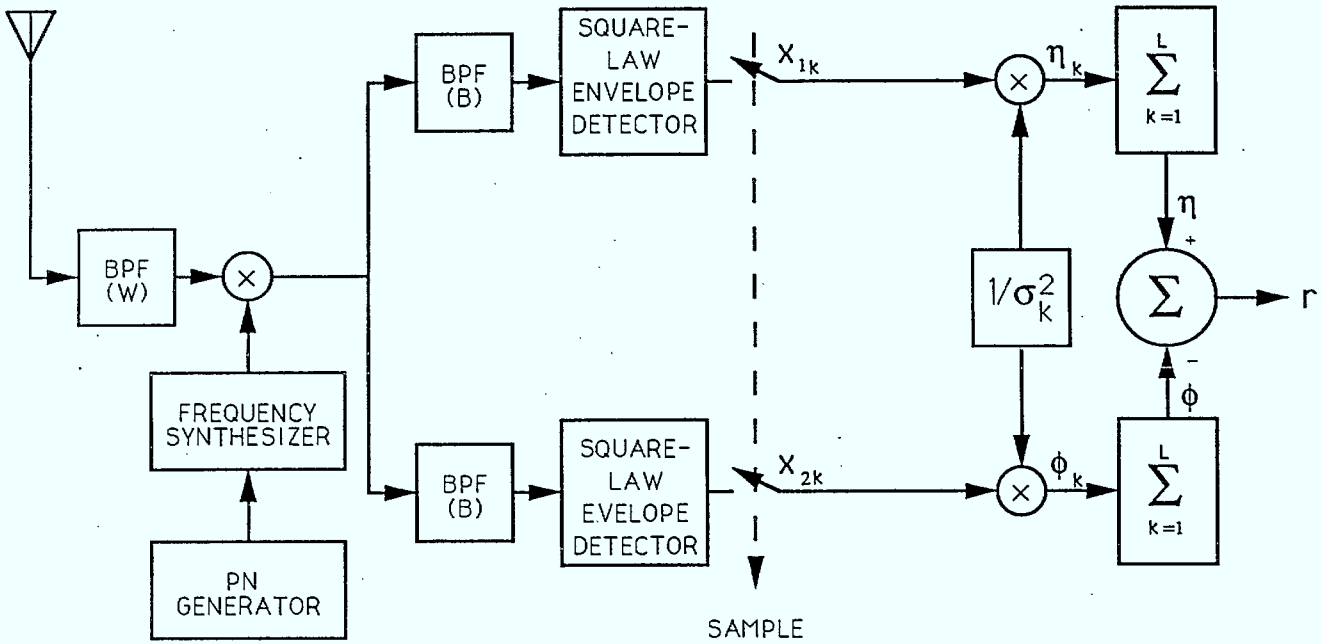

Figure 2.2: Adaptive Gain Control Combining Scheme for a FFH/BFSK system with L hops/bit.

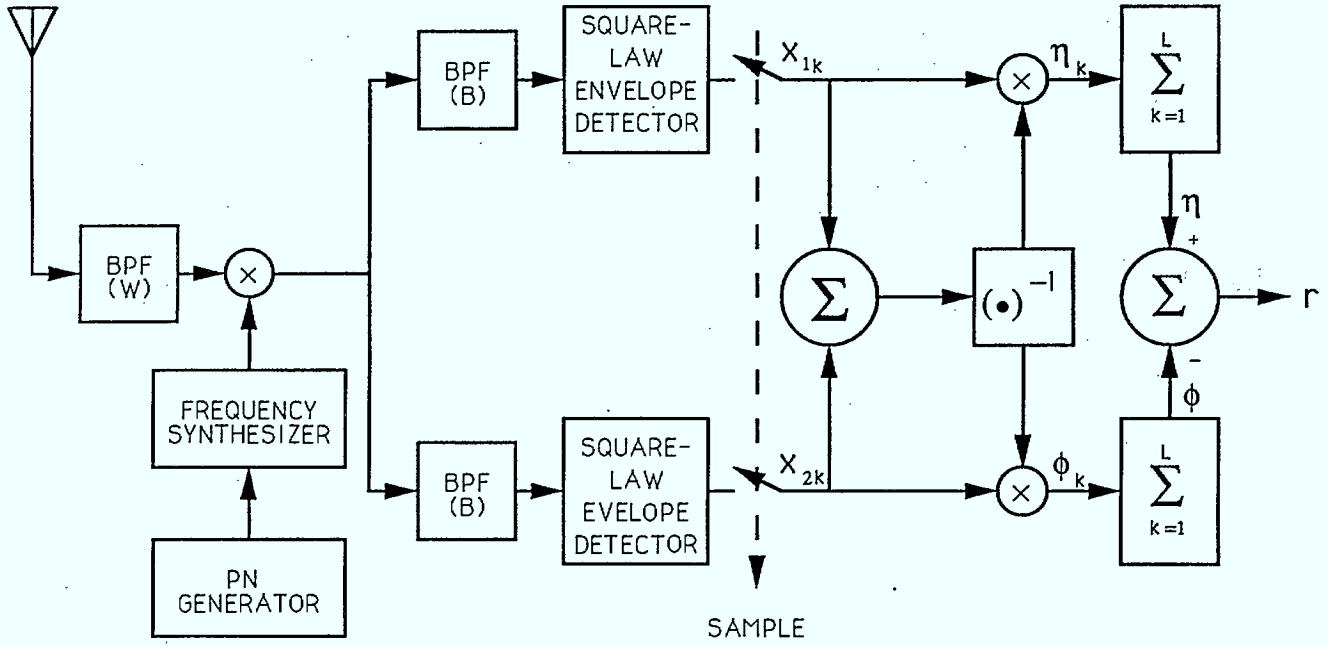

Figure 2.3: Self-Normalizing Combining Scheme for a FFH/BFSK system with L hops/bit.

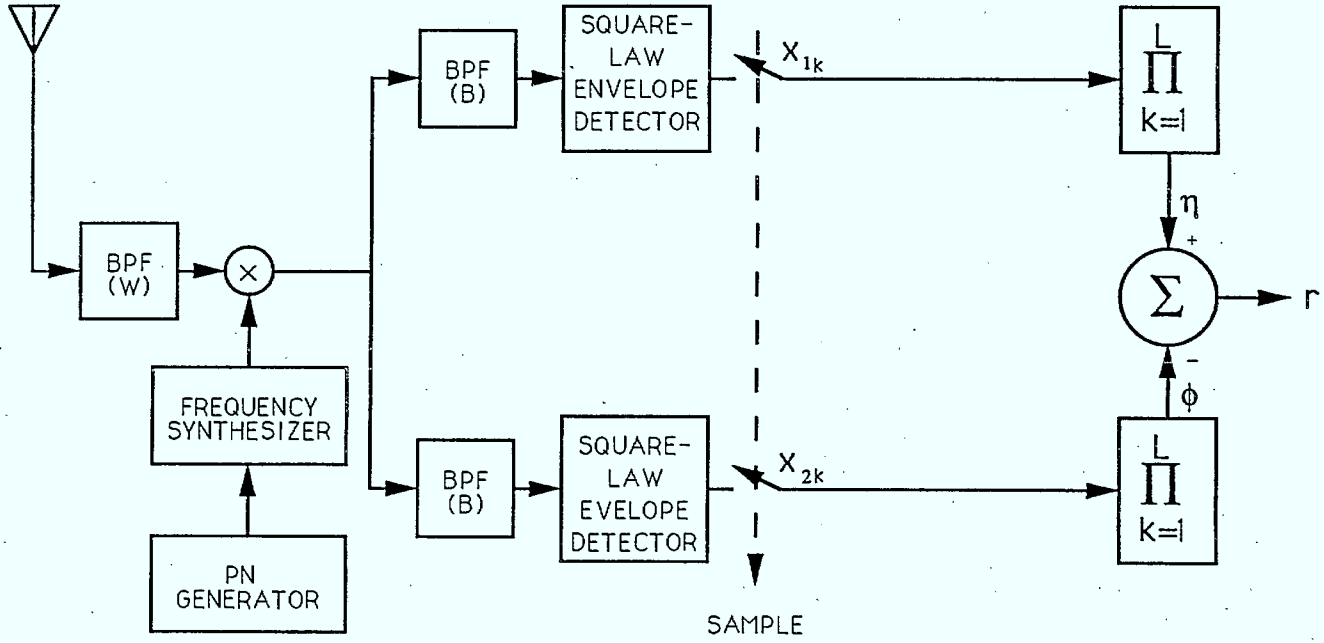

Figure 2.4: Product Combining Scheme for a FFH/BFSK system with L hops/bit.

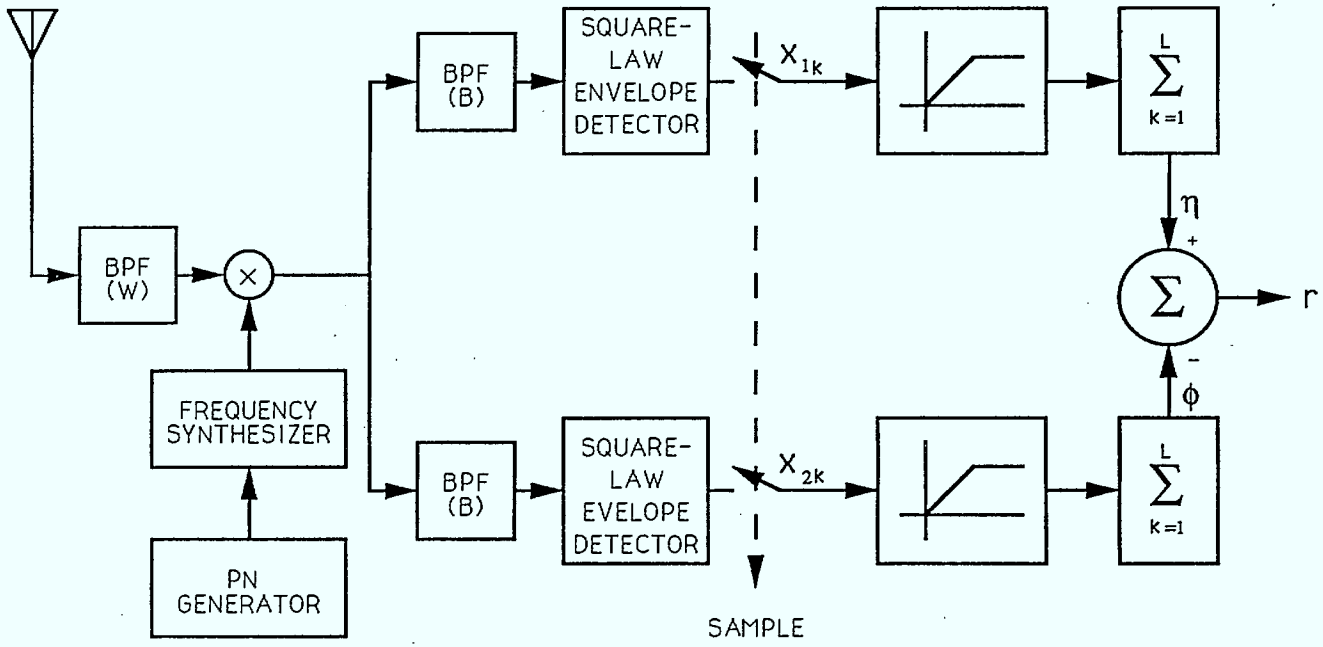

Figure 2.5: Clipping Combining Scheme for a FFH/BFSK system with L hops/bit.

therefore, require information which may not be available in practical situations. In order to implement these combiners, either some form of measurement or estimation must be employed. We can think of the self-normalizing scheme as an AGC type combiner with estimation of the variance of the noise in the channels.

If estimations are used, the quality of the estimation will determine the performance of the combiner. There may be some good approaches which can estimate the variance of the channel noise accurately. This in turn would make the performance of the corresponding combiner similar to that of the AGC combiner. To obtain such a method, a modified method of estimating the channel noise level is proposed and studied. Its performance under PBN jamming is analysed, and is compared with other nonlinear combining schemes.

2.2 Modified Self-Normalizing Combiner

The basic idea of an AGC combiner is to limit the influence of the large jamming energy which may be present in a hop. The weight of the hop output is small when interfered by a jammer, thus the contribution to the total sum is small, and the influence of the jamming is reduced.

In a true AGC combiner, the noise power in each channel is normalized to a unit. In a self-normalizing combiner, all channel outputs are less than one after normalizing. However, after comparison with the AGC combiner, we found that a drawback of this scheme is that it does not emphasize the influence of the hops which are not jammed.

In a self-normalizing scheme, the estimation of the variance of the channel noise is the sum of the outputs of M channels. To decide if a channel contains a signal, probably a better method to estimate the noise variance is to make use of the outputs of the other $M-1$ channels. Thus if the channel actually contains a signal, the outputs of the other $M-1$ channels are purely noise samples. If the channel under consideration does not contain a signal, the other $M-1$ channels will contain both signal and noise samples. Because of the

signal energy in the samples used for estimation, the estimation of the noise variance may tend to be larger than the true value. This causes the weight of the noise only channel to be smaller, which is desirable.

In the modified self-normalizing Combiner shown in Fig. 2.6, instead of normalizing with the sum of all M channel outputs, the channel output is normalized with the sum of the other $M-1$ outputs.

2.2.1 Some Intuitive Considerations

If there is a hop without jamming, and the signal to thermal noise ratio is large, the weight of this hop tends to be large, and so will make a large contribution to the sum. The AGC combining scheme has a similar property, where the small thermal noise variance makes the weight of a clean hop relatively large. The product scheme is also similar, though in a reverse way, i.e., if there is a clean hop, the combining output of the channels without a signal is near zero.

The proposed scheme also makes the fluctuation in data large when the number of channels, M, is small. This is an undesirable property. Further analysis is needed to determine whether the influence of this property is dominant or not.

2.3 Performance Analysis

In this Section we analyse the performance of the normalized combiner under PBN jamming. Both numerical methods and Monte Carlo simulation are used in the analysis.

First we show that for the BFSK case, with diversity $L = 2$, the modified self-normalizing combiner and the conventional self-normalizing combiner are equivalent. In fact, we show that the product combiner [4] is also equivalent to the above two combining schemes when $L = 2$.

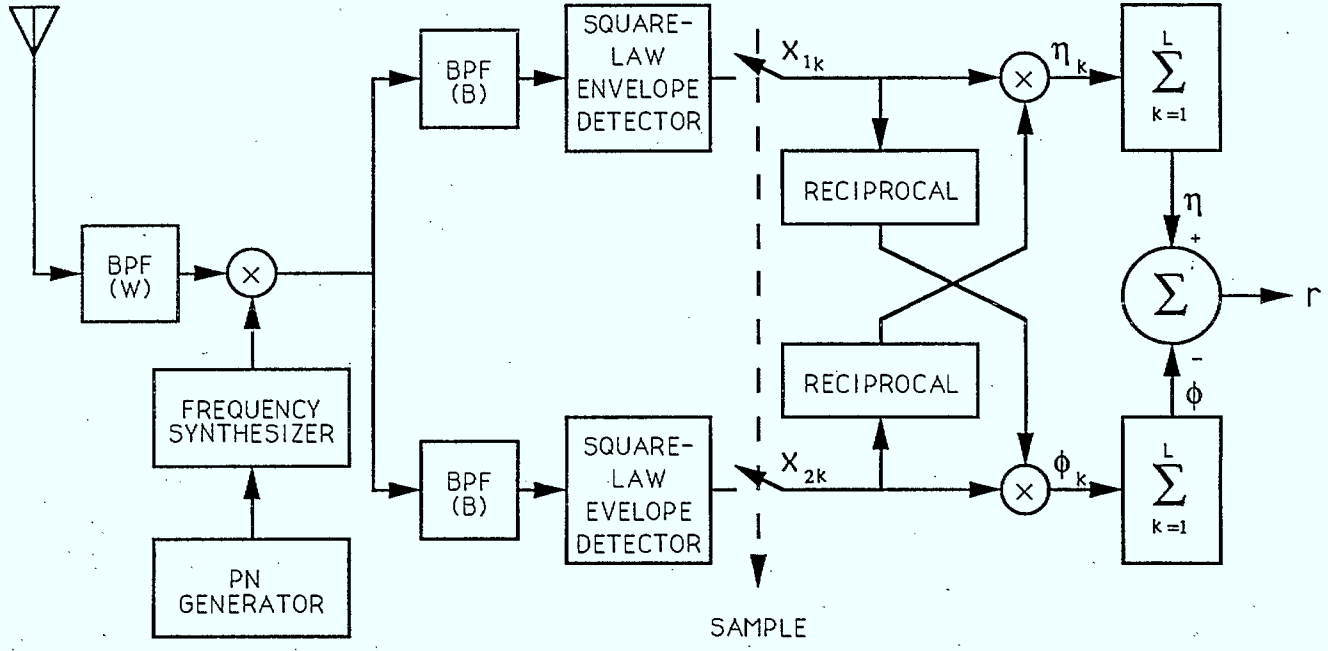

Figure 2.6: Modified Self-Normalizing Scheme for a FFH/BFSK system with L hops/bit.

2.3.1 Equivalence of Three Combiners for L = 2 in a BFSK System

Let x_{1k}, x_{2k} be outputs of square-law envelope detectors in two channels at the kth hop, respectively (see Fig. 2.6). When $L = 2$, let

$$\eta_1 = \frac{x_{11}}{x_{21}},$$

and

$$\eta_2 = \frac{x_{12}}{x_{22}}.$$

η_1 and η_2 are the weighted outputs of channel 1 respectively, in two hops. The outputs of the two channels in the proposed combiner are

$$\eta = \frac{x_{11}}{x_{21}} + \frac{x_{12}}{x_{22}} = \eta_1 + \eta_2,$$

and

$$\phi = \frac{x_{21}}{x_{11}} + \frac{x_{22}}{x_{12}} = \frac{1}{\eta_1} + \frac{1}{\eta_2} = \frac{\eta_1 + \eta_2}{\eta_1 \eta_2}.$$

The detection statistic r is

$$r = \eta - \phi = \eta_1 + \eta_2 - \frac{\eta_1 + \eta_2}{\eta_1 \eta_2} = \frac{\eta_1 + \eta_2}{\eta_1 \eta_2}(\eta_1 \eta_2 - 1).$$

The decision criteria is thus if

$$r \geq 0,$$

the transmitted signal is in channel 1, otherwise, it is in channel 2. However, $r \geq 0$ if and only if $\eta_1 \eta_2 - 1 \geq 0$. Hence we can restate this decision criteria as

$$\eta_1 \eta_2 - 1 \geq 0$$

if channel 1 has a signal. Otherwise it is in channel 2.

For the original self-normalizing combiner, the corresponding outputs are

$$\eta = \frac{x_{11}}{x_{11} + x_{21}} + \frac{x_{12}}{x_{12} + x_{22}} = \frac{\eta_1}{1 + \eta_1} + \frac{\eta_2}{1 + \eta_2}$$

and

$$\phi = \frac{x_{21}}{x_{11}+x_{21}} + \frac{x_{22}}{x_{12}+x_{22}} = \frac{1}{1+\eta_1} + \frac{1}{1+\eta_2}.$$

The detection statistic r is

$$r = \frac{\eta_1}{1+\eta_1} + \frac{\eta_2}{1+\eta_2} - \left(\frac{1}{1+\eta_1} + \frac{1}{1+\eta_2}\right) = \frac{2(\eta_1\eta_2 - 1)}{(1+\eta_1)(1+\eta_2)},$$

and therefore the decision criteria is also

$$\eta_1\eta_2 - 1 \geq 0$$

then channel 1 has the signal, otherwise it is in channel 2. since the decision criteria are the same, the two combiners are equivalent.

For the product combining scheme, the detection statistic r is

$$\begin{aligned} r &= x_{11}x_{12} - x_{21}x_{22} \\ &= x_{21}x_{22}\left(\frac{x_{11}}{x_{21}}\frac{x_{12}}{x_{22}} - 1\right) \\ &= x_{21}x_{22}(\eta_1\eta_2 - 1), \end{aligned}$$

and the decision still depends on whether

$$\eta_1\eta_2 - 1 \geq 0.$$

Therefore all three schemes are equivalent, and their performance without coding should be identical. However, in a receiver with coding or using soft decisions, this may not be the case.

2.3.2 The BFSK Case

The joint distribution probability density function of the outputs of the square-law envelope detector in two channels is

$$p_{x_1,x_2}(\alpha,\beta) = \begin{cases} \frac{1}{4\sigma_k^4}\exp\left\{-\frac{\alpha+\beta}{2\sigma_k^2} - \rho_k\right\} I_0\left(\sqrt{\frac{2\alpha\rho_k}{\sigma_k^2}}\right), & \text{if } \alpha,\beta \geq 0\ ; \\ 0, & \text{otherwise;} \end{cases} \tag{2.1}$$

where we assume channel 1 contains the signal. This signal has a SNR of

$$\rho_k = \frac{A^2}{2\sigma_k^2},$$

where A is the signal amplitude and

$$\sigma_k^2 = \begin{cases} N_O B & \text{with probability } 1-\gamma, \\ (N_O + J_O/\gamma)B & \text{with probability } \gamma. \end{cases}$$

Let

$$\eta_k = \frac{x_{1k}}{x_{2k}},$$

and

$$\phi_k = \frac{x_{2k}}{x_{1k}}.$$

η_k, ϕ_k, $k = 1, 2, \cdots, L$ are the weighted outputs of channels 1 and 2, respectively. After some derivations (see Section 2.5 for details), we obtain the following probability density function for η_k and ϕ_k,

$$p_{\eta_k}(\eta) = \begin{cases} \frac{1+\eta+\eta\rho_k}{(1+\eta)^3} e^{-\frac{1}{1+\eta}\rho_k} & \eta \geq 0; \\ 0 & \text{otherwise;} \end{cases} \tag{2.2}$$

and

$$p_{\phi_k}(\phi) = \begin{cases} \frac{1+\phi+\rho_k}{(1+\phi)^3} e^{-\frac{\phi}{1+\phi}\rho_k} & \phi \geq 0; \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

After combining, the outputs of the two channels are

$$\eta = \sum_{k=1}^{L} \eta_k,$$

and

$$\phi = \sum_{i=k}^{L} \phi_k.$$

η and ϕ are then compared, and the detection decision made. The probability density function of η and ϕ are Lth order convolutions of $p_{\eta_k}(\eta)$, $p_{\phi_k}(\phi)$, respectively, i.e.,

$$p_\eta(\eta) = \bigodot_{k=1}^{L} p_{\eta_k}(\eta),$$

and

$$p_\phi(\phi) = \bigodot_{k=1}^{L} p_{\phi_k}(\phi),$$

where $\odot$ means continuous convolution.

For $L = 1$, the conditional error rate $p_e(\gamma \mid \rho_1)$ is

$$p_e(\gamma \mid \rho_1) = Pr(\eta_i \leq 1) = \int_0^1 p_\eta(\eta)d\eta = \frac{1}{2}e^{-\rho_1},$$

where

$$\rho_1 = \begin{cases} \frac{E_b}{N_O} & \text{not jammed with probability } 1-\gamma; \\ \frac{E_b}{N_O + J_O/\gamma} & \text{jammed with probability } \gamma. \end{cases}$$

Thus the total error probability is

$$Pe(\gamma) = \frac{1-\gamma}{2}e^{-\frac{E_b}{2N_O}} + \frac{\gamma}{2}e^{-\frac{E_b}{2(N_O+J_O/\gamma}}.$$

This is the same as [2, (15)], which is intuitive, since when $L = 1$ the normalization has no influence on the decision. This result however, is useful since it verifies that the derived formula is correct.

For $L = 2$, we have shown that the performance is the same as that of a conventional self-normalizing combiner. For $L > 2$, the conditional probability of error is

$$p_e(\gamma \mid \rho_1, \rho_2, \cdots, \rho_L) = 1 - \int_0^\infty \left[\int_0^\eta p_\phi(\phi)d\phi\right]^{L-1} p_\eta(\eta)d\eta,$$

and the total error rate is

$$\begin{aligned} p_e(\gamma) \quad = \quad & \textstyle\sum_{k=0}^{L} \binom{L}{k} (1-\gamma)^k \gamma^{L-k} p_e(\gamma \mid \rho_1, \rho_2, \cdots, \rho_k = \frac{E_b}{LN_O}, \\ & \rho_{k+1}, \cdots, \rho_L = \frac{E_b}{L(N_O+J_O/\gamma)}) \end{aligned}$$

For moderate L, $p_e(\gamma)$ can be computed numerically using a reasonably powerful computer.

For a simple performance comparison under some typical conditions, Monte Carlo simulation can be used. The time requirements for multiple numerical integration are enormous, so simulation is used to obtain some quick results. The modified self-normalizing

scheme was compared with the other schemes using simulation.

Simulation results for $L = 4$ and $M = 2$ are given in Fig. 2.7 and Fig. 2.8, where the probability of error, p_e, is plotted versus bit energy to jamming noise density ratio, E_b/J_O. The bit energy to thermal noise density ratio, E_b/N_O, was chosen as 12.31 dB, for which a bit error rate of 10^{-4} can be achieved without jamming, for $L = 1$. Five types of combining schemes, linear, AGC, self-normalizing, modified self-normalizing, and product combining, were tested. The results for $\gamma = 0.1$ and $\gamma = 0.01$ are given in Figs. 2.7 and 2.8 respectively.

From the two Figures, we can see that the anti-jamming performance of the nonlinear combining schemes are similar. However, the tendency of the modified self-normalizing scheme to create large fluctuations in the data does have a negative influence, which causes a higher probability of error. The reason for these fluctuations is the small number of data values used in the estimation when M is small. We expect that the modified self-normalizing scheme would perform much better when M is large. While the differences between these nonlinear combining schemes are not large, it is not known if this will be the case when coding is used.

2.3.3 The MFSK Case

For MFSK, bound analysis and simulation seem to be good choices for comparison. Simulation results for $M = 4$, $L = 4$, $\gamma = 0.1$ and $\gamma = 0.01$ are given in Figs. 2.9 and 2.10, respectively. For comparison with different M, the bit energy to thermal noise density ratio, E_b/N_O, was chosen such that a bit error rate (BER) of 10^{-4} can be achieved without jamming for $L = 1$. It is known that [12, vol. 1, Ch. 4, Eq. 4.76]

$$P_b \leq \frac{1}{4} M e^{-\frac{E_b log_2 M}{2N_O}},$$

and when $P_b \leq 10^{-4}$, the bound is very tight. Thus for $P_b \leq 10^{-4}$, we have

$$P_b \approx \frac{1}{4} M e^{-\frac{E_b log_2 M}{2N_O}},$$

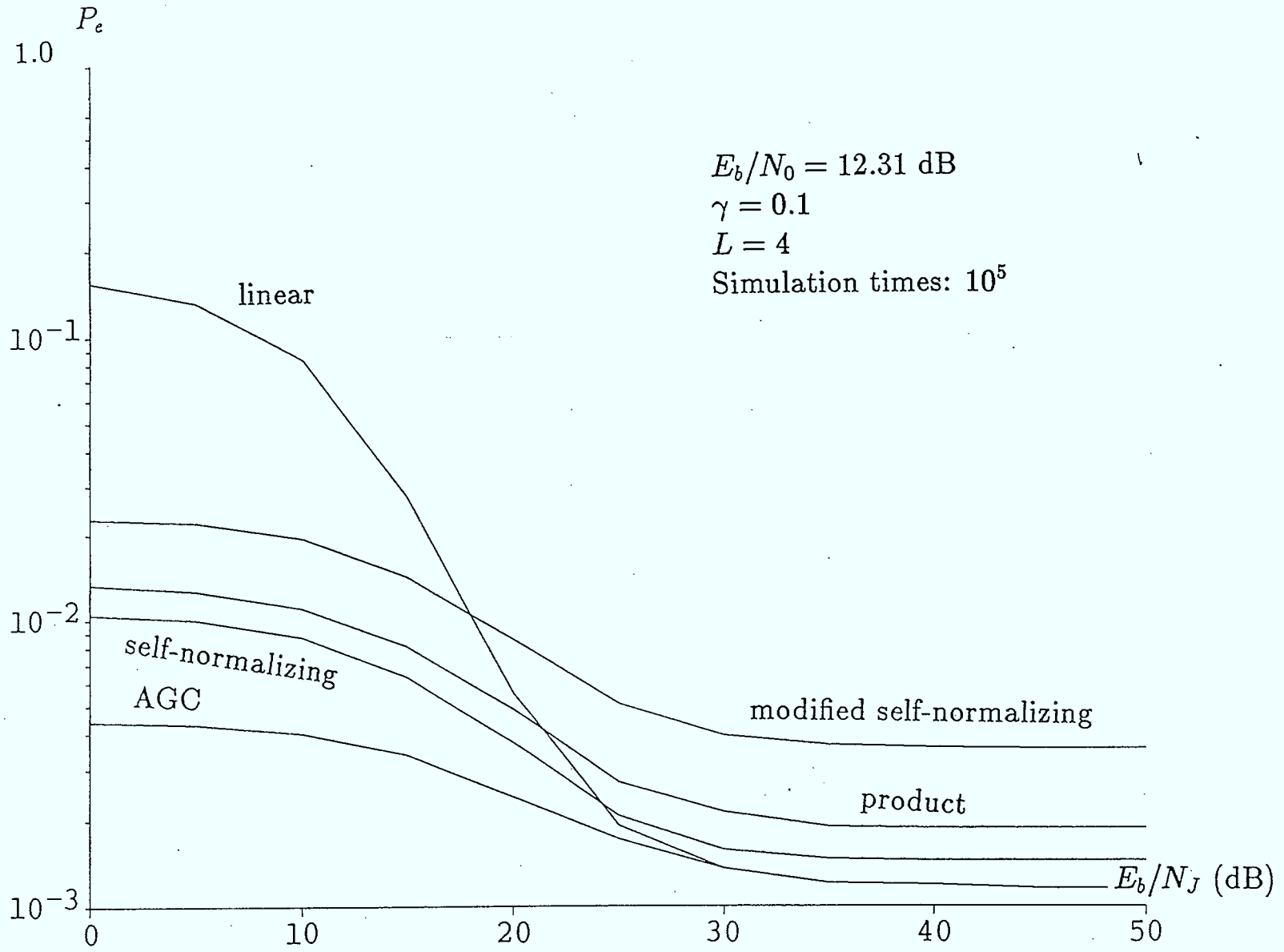

Figure 2.7: Simulation Results for the FFH/BFSK Receiver with Combiners Under Partial Band Jamming, $M = 2, L = 4$ and $\gamma = 0.1$.

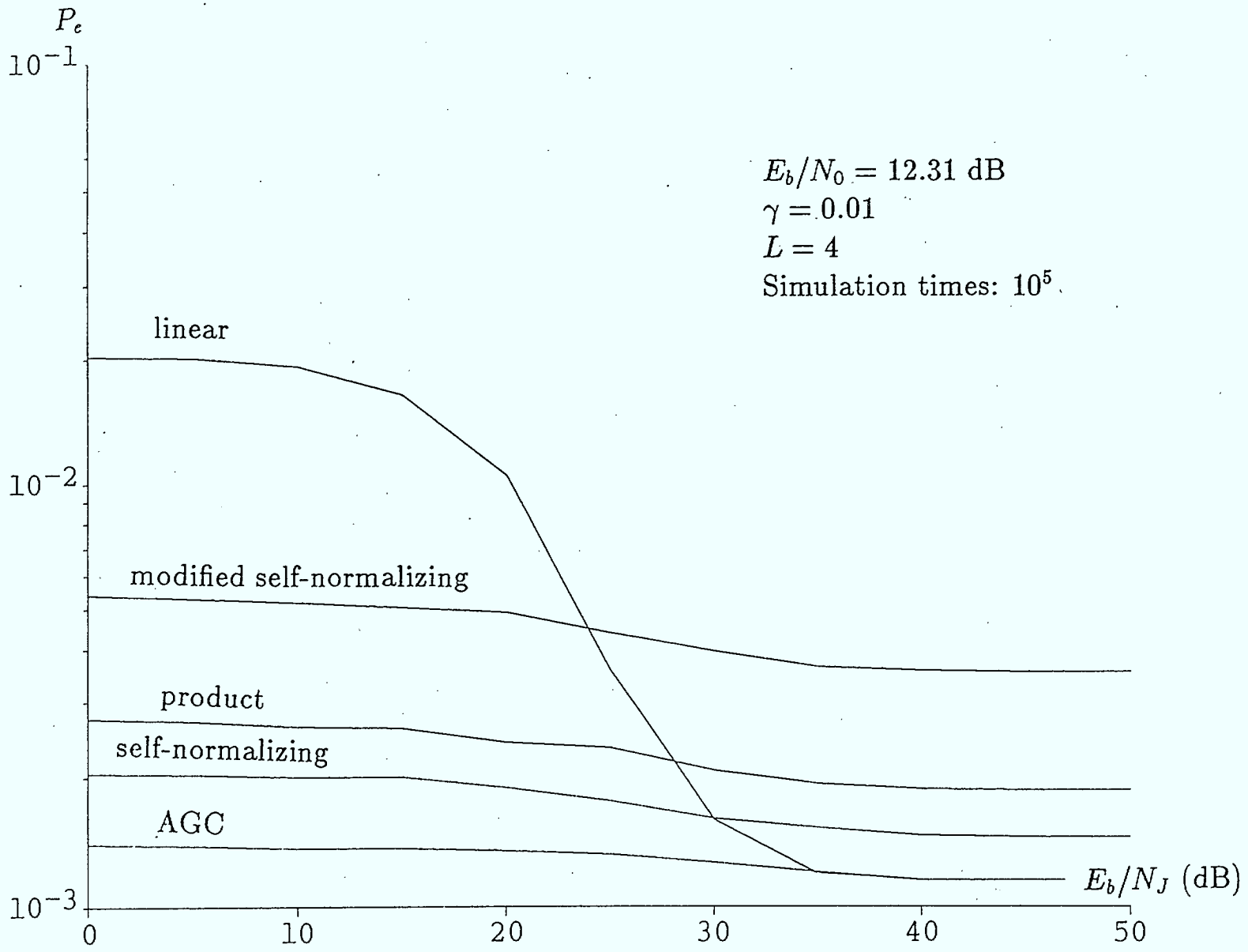

Figure 2.8: Simulation Results for the FFH/BFSK Receiver with Combiners Under Partial Band Jamming, $M = 2, L = 4$ and $\gamma = 0.01$.

hence

$$\frac{E_b}{N_O} = \frac{2}{log_2 M} ln\left(\frac{M}{4P_b}\right).$$

For $P_b = 10^{-4}$, $M = 4$ and 8, the corresponding E_b/N_O are 9.64 dB and 8.20 dB, respectively. Therefore E_b/N_O was chosen to be 9.64 dB. The results indicate that the differences between the anti-jamming performances of the four schemes, i.e., modified self-normalizing, product combining, self-normalizing, and AGC combining, are not very large. In fact, the differences in the performance of the modified self-normalizing and product combining schemes is quite small, and the performance of the self-normalizing scheme is quite close to that of AGC combining. AGC combining can be viewed as a lower bound of the anti-jamming performance for the self-normalizing type combining schemes, since the AGC scheme uses the exact noise variance in each hop.

Simulation results for $M = 8$, $L = 4$, $\gamma = 0.1$, and $\gamma = 0.01$ are given in Figs. 2.11 and 2.12, respectively. In Fig. 2.10, the performance of the self-normalizing technique and its modified version are quite close to that of AGC combining when E_b/J_O is low. In general, the performance of the product, self-normalizing and modified self-normalizing techniques approach that of AGC combining as M increases. As well, the performance of linear combining is superior when the jamming is weak, i.e., when the main factor determining the BER is the thermal noise (Gaussian white noise). In this case, the linear combining of the outputs of the square-law envelope detector is almost optimum. However, when E_b/J_O is high, the performance of self-normalizing combining is closer to that of AGC combining than its modified version.

2.4 Comments

Some nonlinear combining schemes without coding have been analysed. Schemes without coding are compared first to determine if there is an outstanding nonlinear combining scheme which has a much better anti-jamming performance than the others.

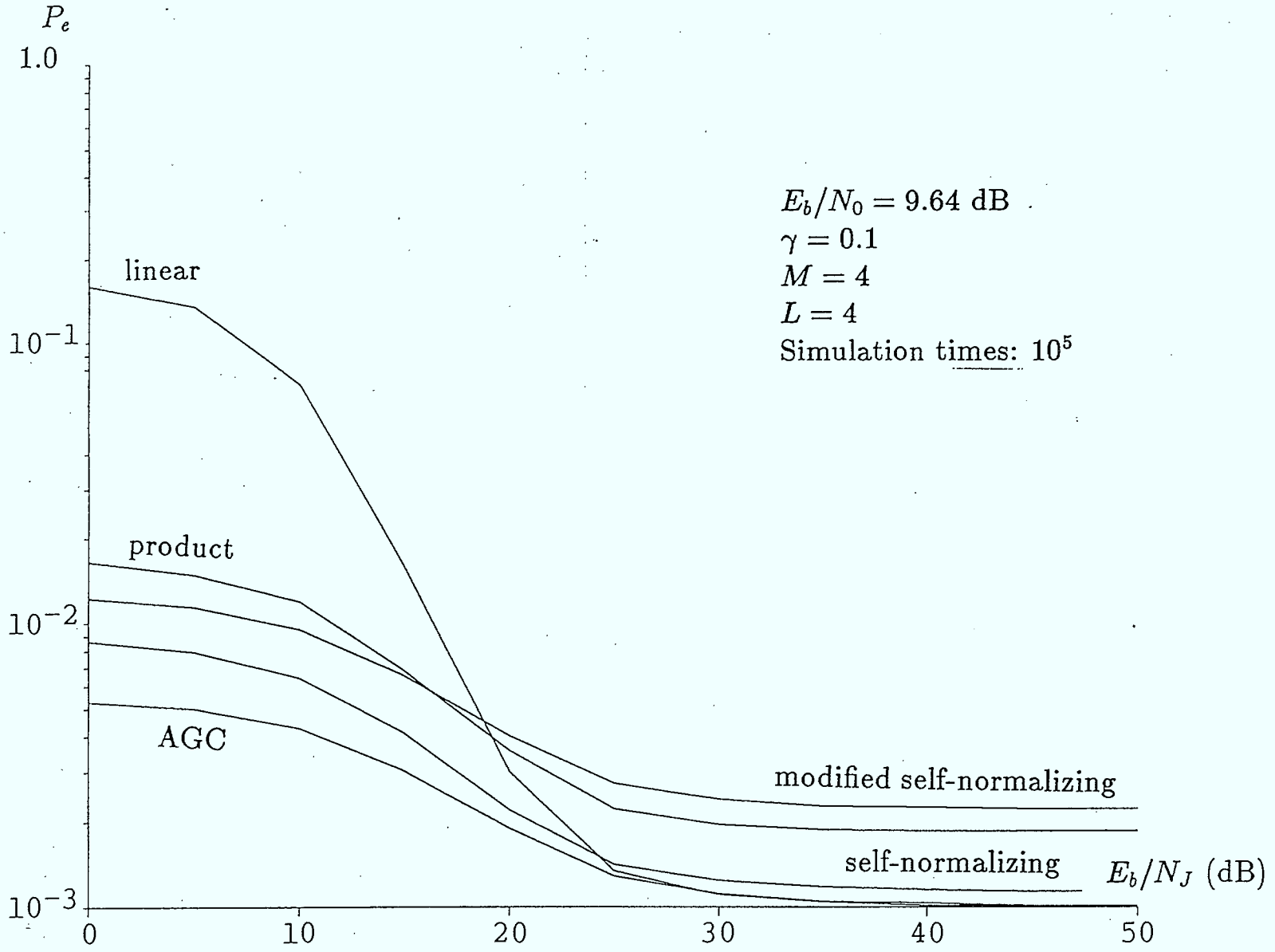

Figure 2.9: Simulation Results for the FFH/BFSK Receiver with Combiners Under Partial Band Jamming, $M = 4, L = 4$ and $\gamma = 0.1$.

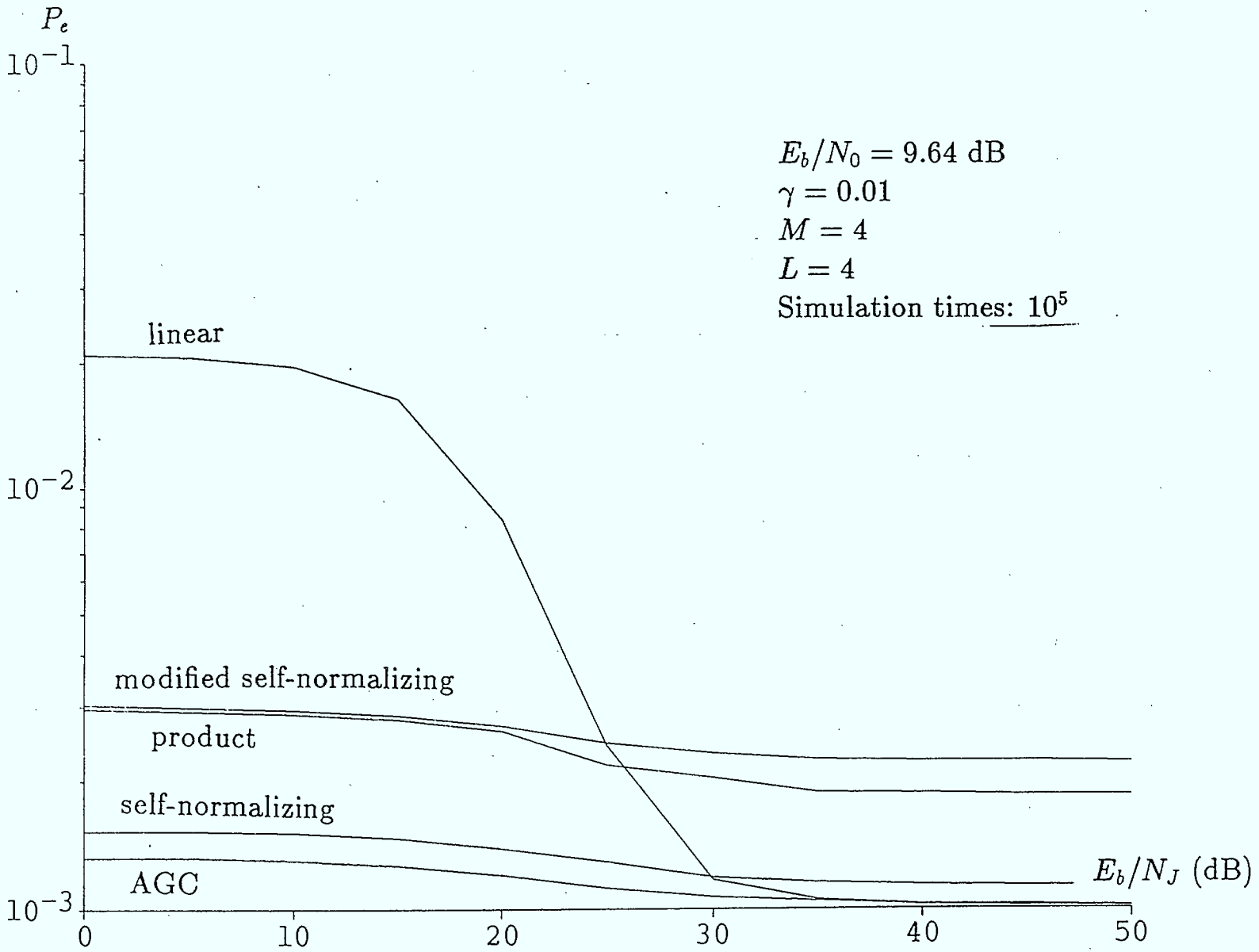

Figure 2.10: Simulation Results for the FFH/BFSK Receiver with Combiners Under Partial Band Jamming, $M = 4, L = 4$ and $\gamma = 0.01$.

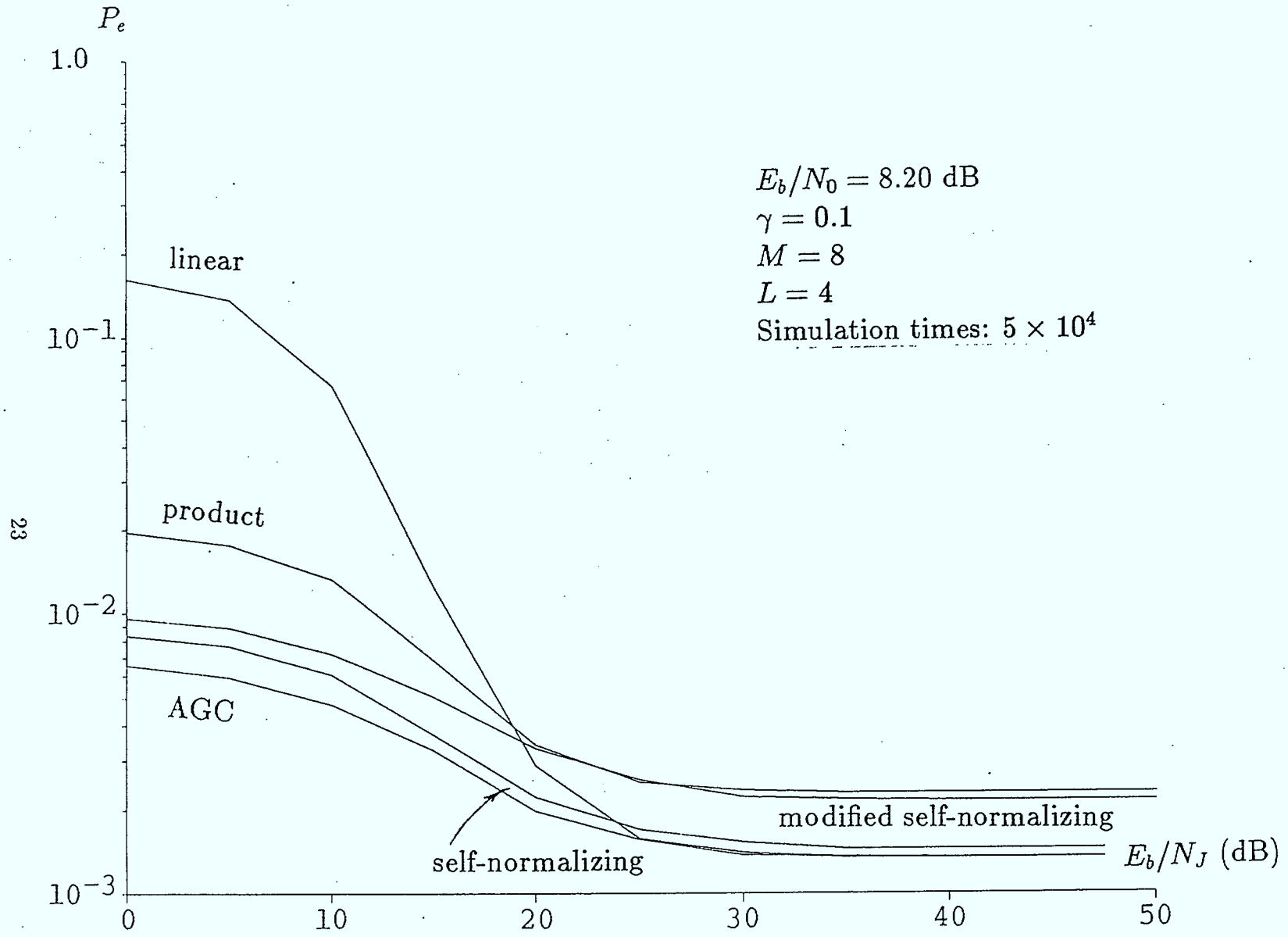

Figure 2.11: Simulation Results for the FFH/BFSK Receiver with Combiners Under Partial Band Jamming. $M = 8$, $L = 4$ and $\gamma = 0.1$.

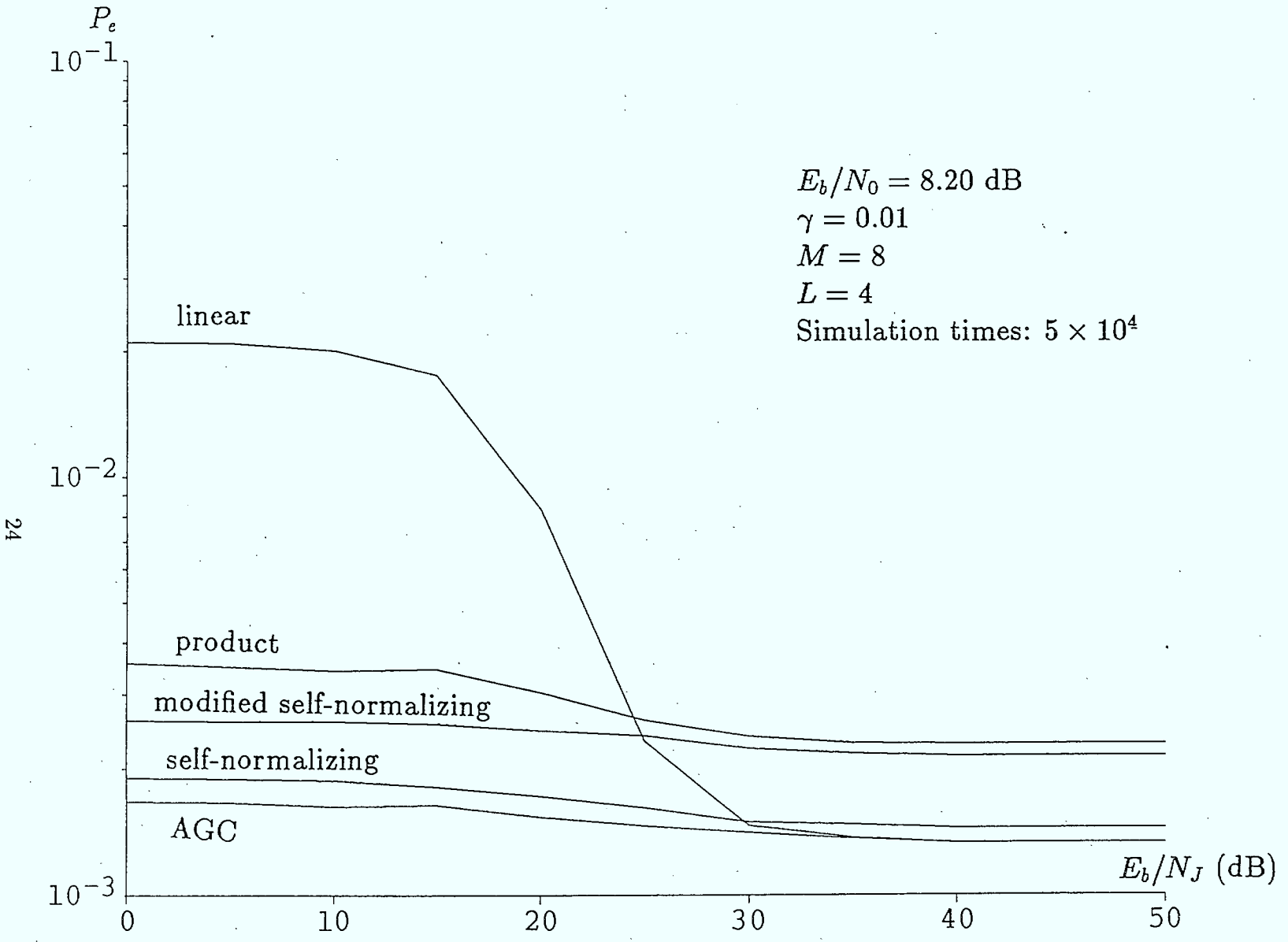

Figure 2.12: Simulation Results for the FFH/BFSK Receiver with Combiners Under Partial Band Jamming, $M = 8, L = 4$ and $\gamma = 0.01$.

2.5 Derivation of (2.2) and (2.3)

The joint probability of x_1 and x_2 is given in 2.1 and let

$$\phi = \frac{x_2}{x_1},$$

then, $p_\phi(\phi)$, the probability density function of ϕ, $p_\phi(\phi)$, is

$$\begin{aligned} p_\phi(\phi) &= \int_{-\infty}^{\infty} \mid \alpha \mid p_{x_1,x_2}(\alpha, \phi\alpha) d\alpha \\ &= \int_0^{\infty} \alpha \frac{1}{4\sigma_k^4} \exp\left[-\frac{(1+\phi)\alpha}{2\sigma_k^2} - \rho_k\right] I_0\left(\sqrt{\frac{2\alpha\rho_k}{\sigma_k^2}}\right) d\alpha. \end{aligned}$$

Substituting $\rho_k = \frac{A^2}{2\sigma_k^2}$ into above expression, we obtain

$$p_\phi(\phi) = \int_0^{\infty} \frac{1}{4\sigma_k^4} \alpha \exp\left[-\frac{(1+\phi)\alpha + A^2}{2\sigma_k^2}\right] I_0\left(\frac{A\sqrt{\alpha}}{\sigma_k^2}\right) d\alpha.$$

Let

$$\tilde{\alpha}^2 = (1+\phi)\alpha,$$

and

$$\tilde{A} = \frac{A}{\sqrt{1+\phi}}.$$

Then

$$\begin{aligned} p_\phi(\phi) &= \int_0^{\infty} \frac{1}{2\sigma_k^4} \frac{1}{(1+\phi)^2} \tilde{\alpha}^3 \exp\left[-\frac{\tilde{\alpha}^2 + (1+\phi)\tilde{A}^2}{2\sigma_k^2}\right] I_0\left(\frac{\tilde{A}\tilde{\alpha}}{\sigma_k^2}\right) d\tilde{\alpha} \\ &= \frac{1}{2\sigma_k^2} \frac{1}{(1+\phi)^2} \exp\left(-\frac{\phi\tilde{A}^2}{2\sigma_k^2}\right) \int_0^{\infty} \frac{1}{\sigma_k^2} \tilde{\alpha}^3 \exp\left(-\frac{\tilde{\alpha}^2 + \tilde{A}^2}{2\sigma_k^2}\right) I_0\left(\frac{\tilde{A}\tilde{\alpha}}{\sigma_k^2}\right) d\tilde{\alpha} \\ &= \frac{1}{2\sigma_k^2(1+\phi)^2} \exp\left(-\frac{\phi\tilde{A}^2}{2\sigma_k^2}\right) F(\tilde{A}, \sigma_k^2), \end{aligned}$$

where

$$F(\tilde{A}, \sigma_k^2) = \int_0^{\infty} \frac{1}{\sigma_k^2} \tilde{\alpha}^3 \exp\left(-\frac{\tilde{\alpha}^2 + \tilde{A}^2}{2\sigma_k^2}\right) I_0\left(\frac{\tilde{A}\tilde{\alpha}}{\sigma_k^2}\right) d\tilde{\alpha},$$

is the second order moment of the Rician distribution [8], and

$$F(\tilde{A}, \sigma_k^2) = 2(\sigma_k^2 + \frac{\tilde{A}^2}{2}).$$

Therefore

$$\begin{aligned} p_\phi(\phi) &= \frac{1}{2\sigma_k^2}\frac{1}{(1+\phi)^2}\exp\left(-\frac{\phi\tilde{A}^2}{2\sigma_k^2}\right)2(\sigma_k^2+\frac{\tilde{A}^2}{2}) \\ &= \frac{1+\phi+\rho_k}{(1+\phi)^3}e^{-\frac{\phi}{1+\phi}\rho_k}, \end{aligned}$$

and because $\eta = \phi^{-1}$,

$$\begin{aligned} p_\eta(\eta) &= \frac{1}{\eta^2}p_\phi(\eta^{-1}) \\ &= \frac{1+\eta+\eta\rho_k}{(1+\eta)^3}e^{-\frac{1}{1+\eta}\rho_k}. \end{aligned}$$

Chapter 3

An Analysis of the Throughput Performance of Coded FFH/MFSK with a Fixed Hop Rate Based on the Cutoff Rate

3.1 Introduction

In previous work [1], the performance of various error correcting (EC) codes in an FFH/MFSK system was evaluated under the condition of a fixed hop rate. Continuing from this work, we analyse the throughput performance of the coded system using the cutoff rate argument. This analysis is intended to upperbound the improvement that can be realized through the use of coding for various system parameters. In this Chapter, we attempt to present the results in a more or less self-contained manner, while more information can be found in Chapter 3 of [1].

A fixed hop rate is a practical requirement for satellite communications when multiple users access the same onboard dehopper. This constant hop rate is determined by many factors, such as the response time of a potential repeat-back jammer and the synchronization capability of communication receivers. That is, the hop rate should be high

enough to avoid repeat-back jammers but low enough to avoid synchronization problems. As in [1], the Chernoff union bound method is used for the performance evaluation.

Suppose that the average signal power S is fixed. Since the hop rate R_h is fixed, so is the energy per hop $E_h = S/R_h$. In this case the term optimum diversity is meaningless, because the diversity factor, L (the number of hops per M-ary symbol), is no longer an independent parameter. Specifically, L is given by

$$L = \frac{rK}{R_b/R_h} = \frac{R'}{R_b/R_h}, \tag{3.1}$$

where r is the EC code rate and $M = 2^K$. R' is the code rate in data bits per M-ary symbol before diversity. Note that (3.1) must satisfy the restriction $L \geq 1$. This means that R_b/R_h cannot exceed the upper limit R'. For a given bit error rate (BER), the information bit rate R_b reflects the throughput of the system. A larger R_b means a larger throughput.

E_b/J_O is determined by

$$E_b/J_O = \frac{S}{R_b J_O} = \frac{S/R_h}{(R_b/R_h)J_O} = \frac{E_h/J_O}{R_b/R_h}. \tag{3.2}$$

For a fixed hop rate, E_b/J_O depends on R_b/R_h and E_h/J_O (which is fixed as mentioned above). Thus we will use E_h/J_O as a basic parameter to evaluate the system performance rather than E_b/J_O. This results in two system performance criteria. One is R_b/R_h, reflecting the system throughput, and the other is the more traditional BER, or P_b. Note that to determine R_b/R_h for a fixed E_h/J_O, P_b must be fixed. In fact, as mentioned above, only for a given P_b can R_b/R_h reflect the throughput in a meaningful way. On the other hand, to determine P_b, R_b/R_h and therefore L must be given. This method of evaluating the system performance is equivalent to the P_b versus E_b/J_O format, for a given R_b/R_h as given in (3.2). Another useful format is P_b versus R_b/R_h for a given E_h/J_O, which shows explicitly the tradeoff between them. It is not difficult to see that these formats present the same results in different ways. In this Chapter, we focus on the R_b/R_h versus E_h/J_O format, which is consistent with a cutoff rate analysis given later in the Chapter.

System assumptions are the same as in Chapters 1 and 2 of [1], with the critical exception that there is no optimum diversity. In Section 3.2, we present the basic formulas for performance evaluation. The results for uncoded systems (but with diversity) can be found in [1]. A theoretical analysis of a coded system using the cutoff rate argument is given. The purpose of this Chapter is to show quantitatively the improvement that can be gained using Error Correction coding, (as opposed to diversity).

We consider two types of worst case (WC) intelligent but non-repeat-back jamming, namely partial band noise and multitone interference. For partial band noise (PBN) jamming, J is restricted to a fraction ρ $(0 < \rho \leq 1)$ of the full spread spectrum bandwidth, but in this band the power spectral density is increased to J_O/ρ. Multitone jamming (MT) includes band multitone jamming and independent multitone jamming. It has been shown that worst case multitone jamming tends to have a single jamming tone per jammed band[9], using equal power tones. We consider only this type of worst case multitone jamming. In this case the jammer has one parameter to optimize, namely the ratio of signal power of one hop to the power of the jamming tone, denoted as α.

3.2 An Analysis of Coded Systems Based on the Cutoff Rate

It is useful to see how, in general, EC coding can improve performance over the uncoded case (but with diversity). This analysis is based on the cutoff rate of a channel. The use of the cutoff rate has been proposed and well argued in [10,11]. In this case, the cutoff rate R_0 is given by [12],

$$R_0 = \log_2 M - \log_2[1 + (M-1)D^L], \tag{3.3}$$

where D is given in (3.4) through (3.7) as follows.

$$D = \begin{cases} \frac{4e^{-1}}{E_h/J_O} & \text{PBN} \quad \frac{E_h}{J_O} \geq 3; \\ \frac{e^{-\frac{\lambda}{\lambda+1}\frac{E_h}{J_O}}}{1-\lambda^2} & \text{PBN} \quad \frac{E_h}{J_O} < 3; \\ \frac{1}{E_h/J_O} & \text{MT}, \quad K=1, \quad \frac{E_h}{J_O} \geq 2; \\ \frac{1}{2} & \text{MT}, \quad K=1, \quad \frac{E_h}{J_O} < 2; \\ \frac{\beta K}{E_h/J_O} & \text{MT}, \quad K \geq 2, \quad \frac{E_h}{J_O} \geq \alpha_0 M; \\ \frac{1}{E_h/J_O}\left[\frac{\alpha_{wc}(M-2)}{1-\alpha_{wc}}\right]^{1-\alpha_{wc}} & \text{MT}, \quad K \geq 2, \quad \frac{E_h}{J_O} < \alpha_0 M, \end{cases} \tag{3.4}$$

with λ equal to

$$\lambda = \frac{1}{2}\left[\sqrt{1+3\frac{E_h}{J_O}+\frac{1}{4}\left(\frac{E_h}{J_O}\right)^2} - \frac{1}{2}\frac{E_h}{J_O} - 1\right]. \tag{3.5}$$

Note that D should be raised to the power L to get D^L. β is given in Table 2.1 of [1]. The worst case α, α_{wc}, is then

$$\alpha_{wc} = \begin{cases} \alpha_0 & E_h/J_O \geq \alpha_0 M; \\ \frac{E_h/J_O}{M} & E_h/J_O < \alpha_0 M. \end{cases} \tag{3.6}$$

with α_0 given in Table 2.1 of [1]. The worst case ρ, denoted as ρ_{wc}, is given by

$$\rho_{wc} = \begin{cases} \frac{3}{E_h/J_O} & E_h/J_O \geq 3; \\ 1 & E_h/J_O < 3. \end{cases} \tag{3.7}$$

If a code is so powerful (we call it a very powerful code) that the cutoff rate is achieved, we have

$$R' = R_0. \tag{3.8}$$

Using (3.1), (3.3) and (3.8), we have

$$\frac{R_b}{R_h} = \frac{1}{L}\{\log_2 M - \log_2[1+(M-1)D^L]\}. \tag{3.9}$$

Now we show that R_b/R_h in (3.9) as a function of L is maximized when $L = 1$. This is a sensible result, since it means that if a very powerful EC code is used, no diversity is necessary to add more redundancy.

Thus, when a very powerful code is used, the maximum throughput R_b/R_h that can be achieved is in effect the cutoff rate of the M-ary channel without using diversity given by

$$\frac{R_b}{R_h} = \log_2 M - \log_2[1 + (M-1)D]. \tag{3.10}$$

Note that for the case $L = 1$, $R_b/R_h = R'$ always. Thus we can plot R_b/R_h vs. E_h/J_O without concern that R_b/R_h will exceed its upper limit R'. Using (3.10), the throughput performance of a coded system is plotted in Figs. 3.1 and 3.2 respectively for WC PBN jamming and WC MT jamming. Comparing these Figures with the results for an uncoded system, (but with diversity)[1], we make the following conclusions:

1. As was observed for uncoded systems, for all E_h/J_O, PBN is the worst case jamming for $K = 1$ (binary) and MT is the worst for $K \geq 2$ (nonbinary). Results similar to these were observed for systems with fixed data rates[1].

2. As reported in Section 3.2 of [1], under worst case jamming (PBN for $K = 1$ and MT for $K \geq 2$), the optimum K is an increasing function of E_h/J_O and $K = 1$ is never optimum. It is interesting to note that the regions of E_h/J_O for the optimum K are basically the same as those given for the uncoded systems. The regions are 4.8 dB $< E_h/J_O <$ 8.4 dB, when $K = 2$ is optimum, 8.4 dB $< E_h/J_O <$ 13.1 dB, when $K = 3$ is optimum, 13.1 dB $< E_h/J_O <$ 18.0 dB, when $K = 4$ is optimum, and $E_h/J_O >$ 18.0 dB, when $K = 5$ is optimum. The optimum K increases as E_h/J_O increases. This result differs from that for systems with a fixed data rate, for which an increase in K above 2 always gives a poorer performance under MT jamming. These results indicate that we can determine (or estimate) the best K for a particular E_h/J_O. Note that $K = 1$ never gives the best R_b/R_h.

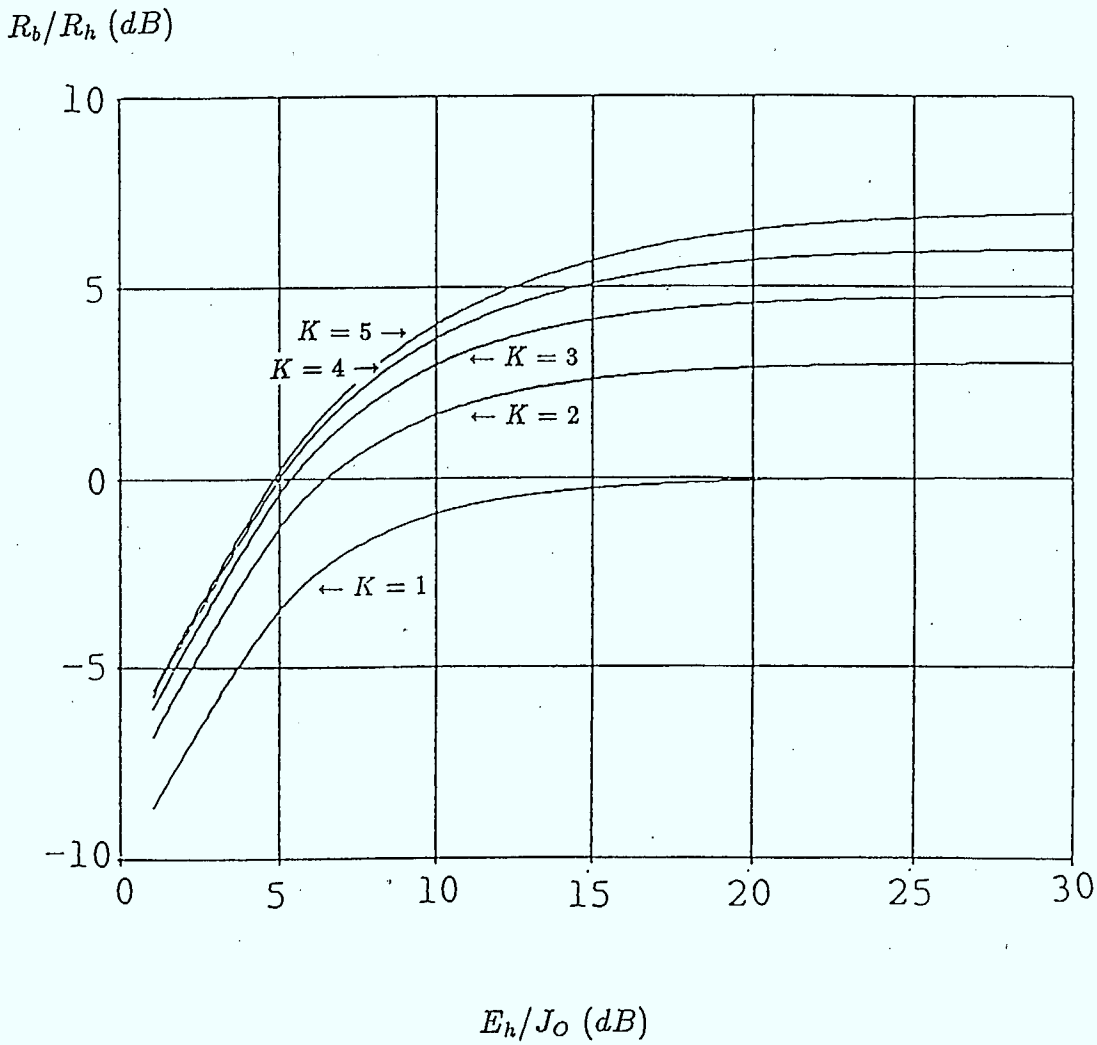

Figure 3.1: Maximum throughput performance of a coded system using a very powerful code with MFSK for $K = 1$ to 5 ($M = 2^K$), and fixed hop rates under WC PBN jamming.

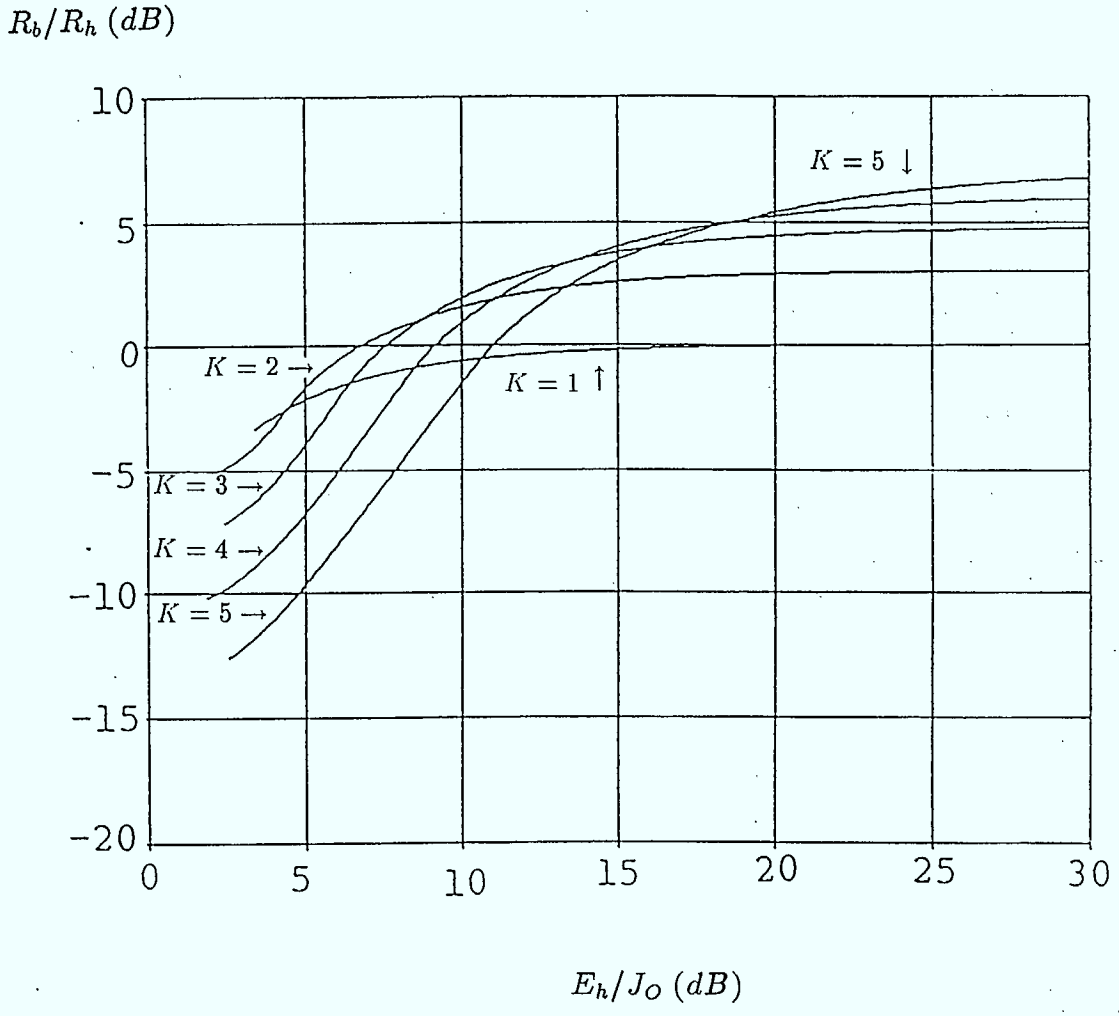

Figure 3.2: Maximum throughput performance of a coded system using a very powerful code with MFSK for $K = 1$ to 5 ($M = 2^K$), and fixed hop rates under WC MT jamming.

3. EC coding can provide a few dB gain in R_b/R_h over systems using only diversity. This is shown in Fig. 3.3. These curves represent the difference between the results for the coded system shown in Fig. 3.1 and 3.2 and the results for the uncoded system shown in Fig. 3.7 of [1]. Note that this coding gain decreases to zero as E_h/J_O increases. This seems to suggest that for "large" E_h/J_O, the uncoded system using simple diversity can perform nearly as well as the coded system. It is seen from Fig. 3.3, however, that for E_h/J_O up to 30 dB, the coding gain is still significant, i.e., greater than 2 dB.

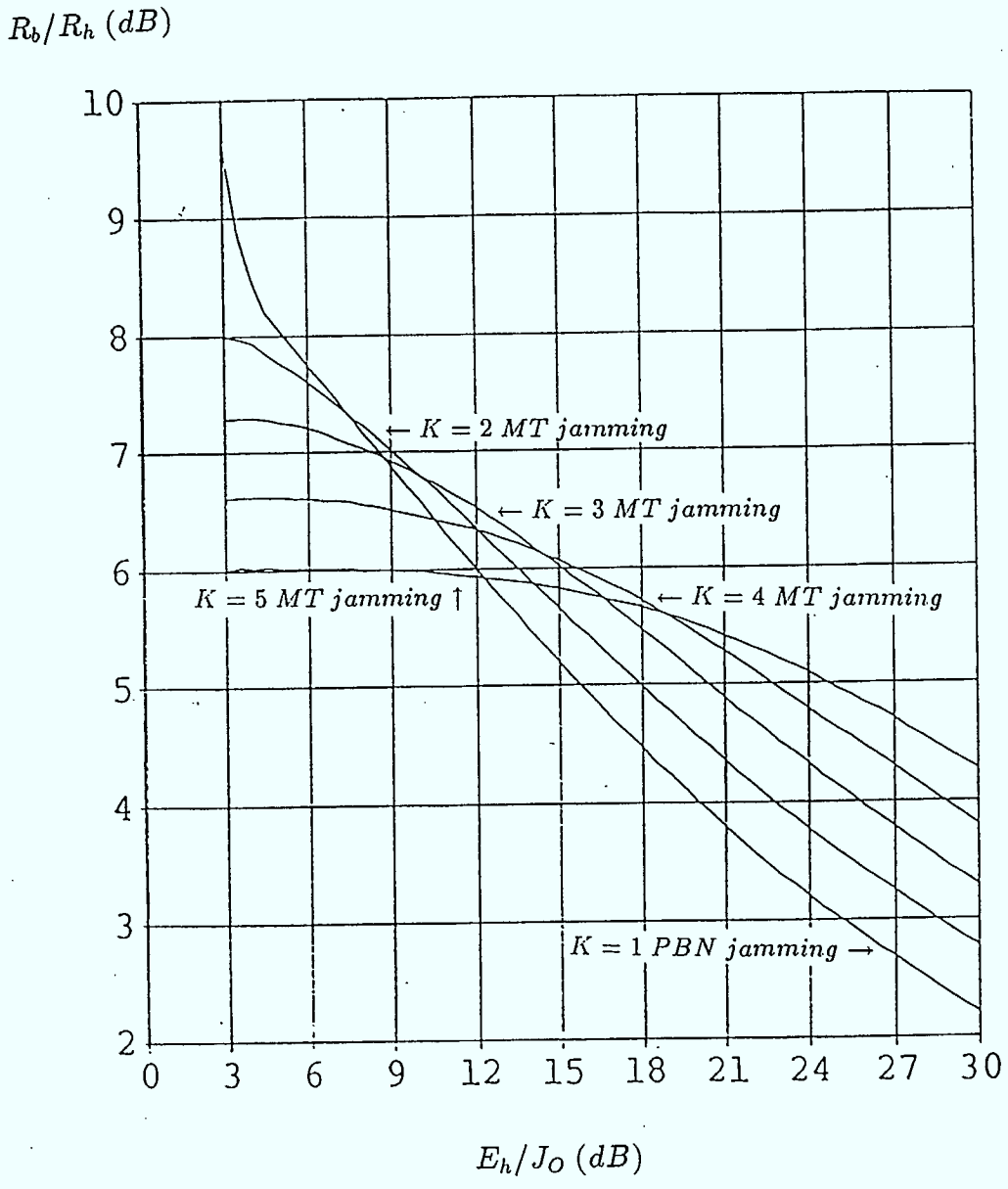

Figure 3.3: Throughput performance gain over an uncoded system (but with diversity) of a coded system using a very powerful code with MFSK for $K = 1$ to 5 ($M = 2^K$), and fixed hop rates under WC jamming.

Chapter 4

Probability Distribution of DPSK in Tone Interference and Applications to SFH/DPSK

4.1 Introduction

In previous work, the performance of fast frequency hopped (FFH) systems has been considered (see, e.g., Chapter 2 of [12] and [13,9,14] and their references). In an FFH system, the information bit rate R_b is relatively low so that an M-ary symbol can be transmitted over one or more hops. If R_b is very high relative to the hop rate R_h, FFH is impossible. For example, it may be required to transmit $1.5Mbit/s$ information at $R_h = 20khop/s$. In this case, slow frequency hopping (SFH) must be used.

In slow frequency hopping there are several transmitted symbols during one hop. This chapter is concerned with SFH/DPSK where the transmitted symbols are modulated in the form of differential PSK. Differential PSK is used because the hop period in SFH is usually not long enough to allow the receiver to recover the carrier phase, and to maintain the phase coherence between different hops at the transmitter. Thus genuine coherent detection is usually impossible. However, since there are many symbols transmitted over one hop, differential coherent detection is possible. Because differential coherent detection

outperforms noncoherent detection such as that used in FFH/MFSK, it is a logical choice for an SFH system.

In this Chapter, both binary and nonbinary (4-ary, 8-ary, etc.) DPSK, i.e., general M-ary DPSK, are considered. Anticipated interference may be both Gaussian noise and tone jamming. Unlike FFH/MFSK, little has been published on SFH/DPSK in the literature. There are many basic questions yet to be answered. In this paper we focus on the effects of jamming and hence coded systems are not considered. The intent is to study the effects of jamming against SFH/DPSK, specifically to provide some tools for the analysis of such a system.

At the receiving end of an uncoded SFH/DPSK system, the differential phase between two consecutive received symbols is detected, and this is used to decide which information symbol was transmitted. Houston[15] and Simon[16] (which is also a part of Chapter 4 in [12]) have analysed the performance of SFH/DPSK under multiple continuous tone jamming for a specific set of signal phases and equally spaced decision regions. Recently, Gong analysed the performance of a specific binary SFH/DPSK scheme in both tone and noise interference[17].

If the jamming tone over a jammed hop is continuous, i.e., the amplitude and initial phase are constant over a particular hop, then the received symbols over that hop are subject to an interference which is highly correlated from symbol to symbol. Recently, Winters has suggested that in correlated *noise*, the performance of DPSK depends on the set of signal phases and decision regions[18]. In order to minimize the demodulator output symbol error rate, we must consider the dependence of the performance of SFH/DPSK, under highly correlated tone jamming, on the signal phases and decision regions.

While the probability distribution of a received differential phase in Gaussian noise has been widely studied and well documented[19,20], no general results have been published on the probability distribution of DPSK in tone interference. Therefore we will derive in

the next section the general probability distribution of a received DPSK signal interfered by continuous tone jamming. By "continuous tone jamming" we mean that a jamming tone interferes with two consecutively transmitted DPSK symbols (with the same amplitude, frequency and initial phase). When DPSK symbols are jammed by a single tone, the jamming tone is assumed to have the same frequency as the DPSK carrier frequency. In Section 4.3, we apply the results obtained in Section 4.2 to the evaluation of SFH/DPSK systems.

4.2 Probability Distributions of the Received DPSK Signal under Tone Jamming

In complex form, the transmitted DPSK signal in the i-th signalling interval is represented by

$$\mathbf{S}^{(i)} = Ee^{j(2\theta+\theta_T^{(i-1)})},$$

where $\theta_T^{(i-1)}$ is the total accumulated phase in the $(i-1)$-th signalling interval and 2θ is the differential phase transmitted in the i-th signalling interval with $0 \leq \theta < \pi$. The jamming tone is represented by

$$\mathbf{J} = Ie^{j\theta'_J},$$

where θ'_J is a random phase uniformly distributed in an interval of length 2π. Let β denote the ratio of the amplitude of the jamming tone to that of the signal tone,

$$\beta = \frac{I}{E}.$$

The received signals, (on which a decision on the transmitted differential phase in the i-th signalling interval is to be based), are represented by $\mathbf{Y}^{(i-1)}$ and $\mathbf{Y}^{(i)}$. The received differential phase is then

$$\Psi = \arg(\mathbf{Y}^{(i)}\mathbf{Y}^{(i-1)^*}),$$

where the phase angle function arg has a main value in the range $(-\pi, \pi]$ and the asterisk denotes complex conjugation.

Simon [16] derived the probability of $\Psi - 2\theta$ within equally spaced decision regions for a *specific* set of θ. The final results are very complicated. This seems to suggest that for *any* θ, the derivation of the probability of Ψ or $\Psi - 2\theta$, (or equivalently the probability distribution), over *any* region would be prohibitively complex. However, we have found that unlike Ψ or $\Psi - 2\theta$, $\Gamma = \Psi - \theta$ has some symmetry that can be utilized to simplify the derivation significantly, as is shown below.

Under continuous tone jamming, we have

$$\mathbf{Y}^{(i-1)} = Ee^{j\theta_T^{(i-1)}} + Ie^{j\theta'_J},$$

and

$$\mathbf{Y}^{(i)} = Ee^{j(2\theta+\theta_T^{(i-1)})} + Ie^{j\theta'_J}.$$

We define

$$R_1 = |\mathbf{Y}^{(i-1)}|,$$

and

$$R_2 = |\mathbf{Y}^{(i)}|.$$

To analyse the bit error rate (BER) performance under strong tone jamming and negligibly low system thermal noise, only the probability distribution of Γ is required. Otherwise we must consider the joint probability distribution of Γ, R_1 and R_2 as will be seen later. To clarify the derivation procedure, we first derive the probability distributions of Γ, R_1 and R_2, separately, and then consider the joint one.

4.2.1 Probability Distribution of the Differential Phase under Continuous Tone Jamming

We first consider the probability distribution of Γ, and those of Ψ and $\Psi - 2\theta$. For Γ, we have

$$\begin{aligned}
\Gamma &= \arg[\mathbf{Y}^{(i)}(\mathbf{Y}^{(i-1)})^* e^{-j\theta}] \\
&= \arg[(e^{j(2\theta+\theta_T^{(i-1)})} + \beta e^{j\theta'_J})(e^{-j\theta_T^{(i-1)}} + \beta e^{-j\theta'_J})e^{-j\theta}] \\
&= \arg[(e^{j2\theta} + \beta e^{j(\theta_T^{(i-1)}+2\theta-\theta'_J)} + \beta e^{j(\theta'_J-\theta_T^{(i-1)})} + \beta^2)e^{-j\theta}] \\
&= \arg[e^{j\theta} + \beta^2 e^{-j\theta} + \beta(e^{j(\theta_T^{(i-1)}+\theta-\theta'_J)} + e^{-j(\theta_T^{(i-1)}+\theta-\theta'_J)})] \\
&= \arg[e^{j\theta} + \beta^2 e^{-j\theta} + 2\beta\cos\theta_J]
\end{aligned} \tag{4.1}$$

where

$$\theta_J = \theta'_J - \theta_T^{(i-1)} - \theta.$$

Since θ'_J can be assumed to be uniformly distributed over $(\theta_T^{(i-1)} + \theta - \pi,\ \theta_T^{(i-1)} + \theta + \pi]$, θ_J is uniformly distributed over $(-\pi,\ \pi]$. Suppose $\beta > 0$ and denote

$$\mathbf{U} = \frac{\cos\theta(1+\beta^2)}{2\beta} + j\frac{\sin\theta(1-\beta^2)}{2\beta},$$

and

$$\mathbf{V} = \mathbf{U} + \cos\theta_J.$$

Then we have

$$\Gamma = \arg(\mathbf{V}).$$

It is clear that θ_J does not change the value of the imaginary part of $\mathbf{V}$. Consequently, $Im(\mathrm{V})$ is equal to $Im(\mathrm{U})$. Let

$$\Phi = \arg(\mathbf{U}).$$

Obviously if $\theta = 0$, Γ is always equal to 0 and the probability density function (PDF) of Γ is

$$p_\Gamma(\gamma) = \delta(\gamma),$$

where $\delta(x)$ is the Dirac delta function.

If $\beta = 1$, $\mathbf{U} = \cos\theta$ and $\mathbf{V} = \cos\theta + \cos\theta_J$. Consider $\cos\theta + \cos\theta_J > 0$, i.e., $\cos\theta_J > -\cos\theta$. Then we have $|\theta_J| < \arccos(\cos(\pi - \theta)) = \pi - \theta$. Thus Γ equals 0 with probability $1 - \theta/\pi$ and π with probability θ/π. For $0 < \theta < \pi$,

$$p_\Gamma(\gamma) = (1 - \frac{\theta}{\pi})\delta(\gamma) + \frac{\theta}{\pi}\delta(\gamma - \pi).$$

Now we assume $\theta \neq 0$ and $\beta \neq 1$, i.e., $\sin(\Phi) \neq 0$. Suppose $\beta < 1$ (i.e., $Im(\mathbf{V}) \geq 0$). Then $0 < \Phi < \pi$ and

$$Prob\{-\pi < \Gamma < 0\} = 0.$$

Now we calculate

$$Pr(\gamma) = Prob\{0 < \Gamma \leq \gamma\},$$

where $0 < \gamma < \pi$. As shown in Fig. 4.1, the intermediate variable d is defined as

$$d = |\mathbf{U}|(\sin(|\Phi|)\cot|\gamma| - \cos\Phi). \tag{4.2}$$

Noting that $|\cos\theta_J| \leq 1$, we have a symmetric region of θ_J centered at 0 in which $0 < \Gamma \leq \gamma$. Specifically, for $-1 < d < 1$, the corresponding θ_J is in $[-\theta_\Gamma, \theta_\Gamma]$, where

$$\theta_\Gamma(\gamma) = \arccos(d),\ |d| \leq 1.$$

For $d \leq -1$, the corresponding θ_J can be anywhere in $(-\pi, \pi]$. For $d \geq 1$, there is no such θ_J that may result in $0 < \Gamma < \gamma$. Then we have

$$Pr(\gamma) = \begin{cases} 1, & d \leq -1; \\ \frac{\theta_\Gamma}{\pi}, & -1 < d < 1; \\ 0, & d \geq 1. \end{cases} \tag{4.3}$$

The cumulative distribution function (CDF) of Γ is then

$$P_\Gamma(\gamma) = \begin{cases} 0, & \gamma \leq 0; \\ Pr(\gamma), & 0 < \gamma < \pi; \\ 1, & \gamma \geq \pi. \end{cases} \tag{4.4}$$

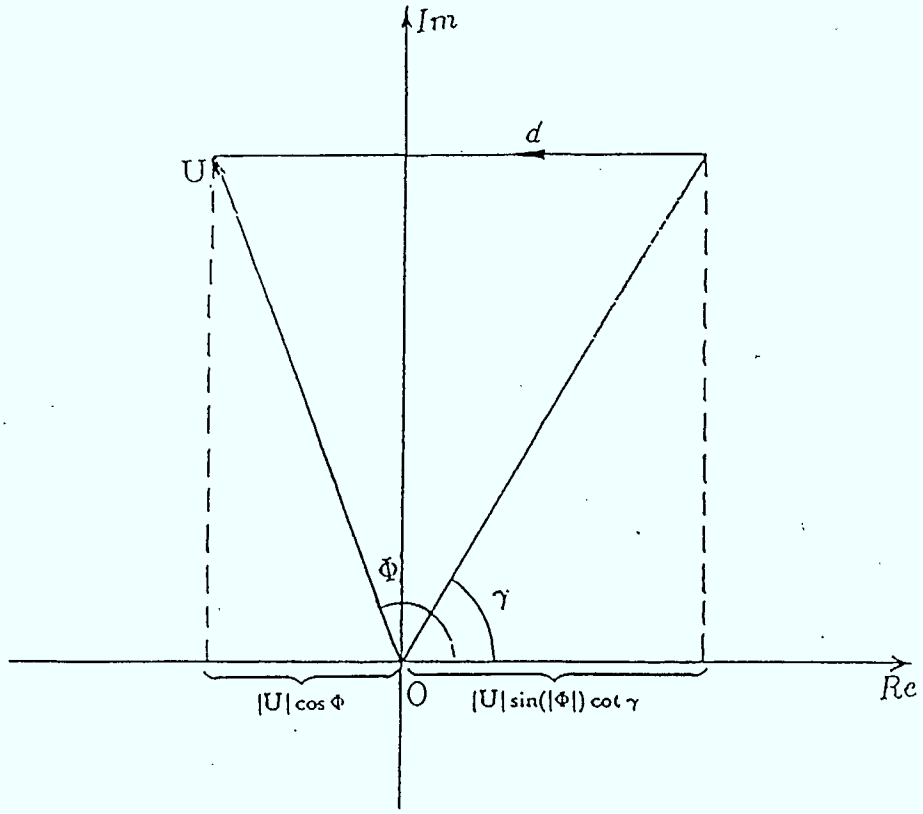

Figure 4.1: Illustration of the relation between the intermediate variable d and U and γ.

Note that, from (4.2), for $\gamma \geq 0$, we have

$$\cot\gamma = \frac{d}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|,$$

and

$$\frac{\partial d}{\partial \gamma} = -|\mathbf{U}|\sin(|\Phi|)\csc^2\gamma.$$

Then the PDF of Γ is

$$p_\Gamma(\gamma) = \begin{cases} \frac{|\mathbf{U}|\sin(|\Phi|)\csc^2\gamma}{\pi\sqrt{1-d^2}}, & \operatorname{arccot}(\frac{1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|) < \gamma < \operatorname{arccot}(\frac{-1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|); \\ 0, & \text{elsewhere.} \end{cases} \tag{4.5}$$

Suppose $\beta > 1$ (i.e., $Im(\mathbf{V}) < 0$). Then $-\pi < \Phi < 0$ and

$$Prob\{0 < \Gamma \leq \pi\} = 0.$$

By symmetry, and noting the term $|\Phi|$ in (4.2), we have

$$Prob\{-\gamma \leq \Gamma < 0\} = Pr(\gamma), \tag{4.6}$$

where $0 < \gamma < \pi$, (note Fig. 4.1). Then the CDF of Γ is

$$P_\Gamma(\gamma) = \begin{cases} 0, & \gamma \leq -\pi; \\ 1 - Pr(-\gamma), & -\pi < \gamma < 0; \\ 1, & \gamma \geq 0, \end{cases} \tag{4.7}$$

and the PDF of Γ for $0 < \theta < \pi$ is

$$p_\Gamma(\gamma) = \begin{cases} \frac{|\mathbf{U}|\sin(|\Phi|)\csc^2\gamma}{\pi\sqrt{1-d^2}}, & -\operatorname{arccot}(\frac{-1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|) < \gamma < -\operatorname{arccot}(\frac{1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|); \\ 0, & \text{elsewhere.} \end{cases} \tag{4.8}$$

Note that using the absolute value of γ in (4.2) is only for conciseness in (4.8), where we actually have

$$d = |\mathbf{U}|(\sin(|\Phi|)\cot(-\gamma) - \cos\Phi).$$

Using the CDFs or PDFs given above, we can calculate the arbitrary probability

$$Pr_\Gamma(\gamma_1, \gamma_2) = Prob\{\gamma_1 < \Gamma \leq \gamma_2\},$$

where $\gamma_1 < \gamma_2$ and both are main valued bounds, (both are in the main value interval of arg). To use Pr_Γ to calculate the probability distribution of Ψ or $\Psi - 2\theta$, all we need do is shift the specified region and convert it into one or two pairs of main-valued bounds for use in Pr_Γ. For example, for main-valued bounds b_1 and b_2, we want to calculate

$$\begin{aligned} P_1 &= Prob\{b_1 < \Psi - 2\theta \leq b_2\} \\ &= Prob\{b_1 + \theta < \Psi - \theta \leq b_2 + \theta\} \\ &= Prob\{b_1 + \theta < \Gamma \leq b_2 + \theta\}. \end{aligned} \tag{4.9}$$

By adding multiples of 2π to $b_1 + \theta$ and $b_2 + \theta$, respectively, we can obtain a pair of bounds bm_1 and bm_2 (corresponding to $b_1 + \theta$ and $b_2 + \theta$, respectively) such that both bm_1 and bm_2 are in $(-\pi, \pi]$. If $bm_1 < bm_2$, they are main-valued bounds and

$$P_1 = Pr_\Gamma(bm_1, bm_2).$$

If $bm_1 > bm_2$,

$$P_1 = Pr_\Gamma(bm_1, \pi) + Pr_\Gamma(-\pi, bm_2).$$

In terms of PDFs, we can obtain the PDFs of Ψ and $\Psi - 2\theta$ by periodically extending p_Γ and then shifting by $\pm\theta$. The periodic extension of $p_\Gamma(\gamma)$, (with period 2π), is

$$\tilde{p}_\Gamma(\gamma) = \sum_{l=-\infty}^{+\infty} p_\Gamma(\gamma - l2\pi).$$

Then the PDF of Ψ is

$$p_\Psi(\psi) = \begin{cases} \tilde{p}_\Gamma(\psi - \theta), & -\pi < \psi \leq \pi; \\ 0, & \text{elsewhere,} \end{cases} \tag{4.10}$$

and the PDF of $\Psi - 2\theta$ is

$$p_{\Psi-2\theta}(\psi_2) = \begin{cases} \tilde{p}_\Gamma(\psi_2 + \theta), & -\pi < \psi_2 \leq \pi; \\ 0, & \text{elsewhere.} \end{cases} \tag{4.11}$$

For later use in deriving the joint probability distribution, we define

$$H_\Gamma(\theta_J) = \begin{cases} 1, & \text{if } \Gamma \leq \gamma; \\ 0, & \text{otherwise.} \end{cases} \tag{4.12}$$

For $\theta = 0$,

$$H_\Gamma(\theta_J) = \begin{cases} 1, & \text{if } \gamma \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

For $\beta = 1$ and $0 < \theta < \pi$,

$$H_\Gamma(\theta_J) = \begin{cases} 1, & \text{if } \gamma \geq \pi; \\ \Pi(\frac{\theta_J}{\pi - \theta}), & \text{if } 0 \leq \gamma < \pi; \\ 0, & \text{otherwise.} \end{cases} \tag{4.14}$$

Here $\Pi(x)$ is the rectangular function which is equal to 1 if $|x| \leq 1$ and 0 otherwise. In the above equation, it is implied that $\arg(\mathbf{X}) = 0$ if $|\mathbf{X}| = 0$. In this case we have

$$\frac{\partial H_\Gamma}{\partial \gamma} = \begin{cases} \delta(\gamma), & \text{if } |\theta_J| \leq \pi - \theta; \\ \delta(\gamma - \pi), & \text{otherwise.} \end{cases} \tag{4.15}$$

For $\beta \neq 1$ and $0 < \theta < \pi$, from (4.3) through (4.8), it is clear that for $|d| < 1$, (or $\gamma_1 < \gamma < \gamma_2$ where $\gamma_1 = \text{arccot}(\frac{1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|)$, $\gamma_2 = \text{arccot}(\frac{-1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|)$ for $\beta < 1$, and $\gamma_1 = -\text{arccot}(\frac{-1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|)$, $\gamma_2 = -\text{arccot}(\frac{1}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|)$ for $\beta > 1$), we have

$$H_\Gamma(\theta_J) = \begin{cases} \Pi(\frac{\theta_J}{\theta_\Gamma}), & \text{if } \beta < 1; \\ \Pi^-(\frac{\theta_J}{\theta_\Gamma}), & \text{if } \beta > 1. \end{cases} \tag{4.16}$$

Here $\Pi^{-}(x)$ is a function which is equal to 1 if $|x| \geq 1$ and 0 otherwise. Then we have

$$\frac{\partial H_\Gamma}{\partial \gamma} = (\delta(x+\theta_\Gamma) + \delta(x-\theta_\Gamma))\frac{\partial \theta_\Gamma}{\partial \gamma} c(\beta), \tag{4.17}$$

where

$$c(\beta) = \begin{cases} 1, & \text{if } \beta < 1; \\ -1, & \text{if } \beta > 1. \end{cases}$$

We can also write the inverse function of θ_Γ

$$\theta_{\Gamma^{-1}}(\theta_\gamma) = \begin{cases} \operatorname{arccot}(\frac{\cos\theta_\gamma}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|), & \text{if } \beta < 1; \\ -\operatorname{arccot}(\frac{\cos\theta_\gamma}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi|), & \text{if } \beta > 1. \end{cases} \tag{4.18}$$

4.2.2 Probability Distribution of the Amplitude under Continuous Tone Jamming

In this section we consider the probability distribution of R_1 and R_2, in particular as a function of θ_J. They will be used in deriving the joint probability distribution.

Note that R_1 and R_2 are nonnegative. For R_1 we have

$$\begin{aligned} R_1^2 &= |\mathbf{Y}^{(i-1)} e^{-j(\theta+\theta_T^{(i-1)})}|^2 \\ &= |E(e^{j\theta_T^{(i-1)}} + \beta e^{j\theta'_J}) e^{-j(\theta+\theta_T^{(i-1)})}|^2 \\ &= |E(e^{-j\theta} + \beta e^{j(\theta'_J - \theta_T^{(i-1)} - \theta)})|^2 \\ &= |E(e^{-j\theta} + \beta e^{j\theta_J})|^2 \\ &= E^2[(\cos\theta + \beta\cos\theta_J)^2 + (-\sin\theta + \beta\sin\theta_J)^2] \\ &= E^2[1 + \beta^2 + 2\beta\cos(\theta+\theta_J)]. \end{aligned} \tag{4.19}$$

If $r_1 \geq (1+\beta)E$, $Prob\{R_1 \leq r_1\} = 1$. If $r_1 \leq |1-\beta|E$, $Prob\{R_1 \leq r_1\} = 0$. If $|1-\beta|E < r_1 < |1+\beta|E$, then we have

$$Prob\{R_1 \leq r_1\} = Prob\{\cos(\theta_J + \theta) \leq \cos\theta_{R_1}\},$$

where

$$\theta_{R_1} = \arccos[(\frac{r_1^2}{E^2} - 1 - \beta^2)\frac{1}{2\beta}].$$

This implies a symmetrical region of θ_J centered at $\pi - \theta$ with a width of $\pi - \theta_{R_1}$ on either side in which $R_1 \leq r_1$. Thus the CDF of R_1 is

$$P_{R_1}(r_1) = \begin{cases} 1, & r_1 \geq (1+\beta)E; \\ 1 - \frac{\theta_{R_1}}{\pi}, & |1-\beta|E < r_1 < |1+\beta|E; \\ 0, & r_1 \leq |1-\beta|E. \end{cases} \tag{4.20}$$

The PDF of R_1 is

$$p_{R_1}(r_1) = \begin{cases} -\frac{1}{\pi}\frac{\partial\theta_{R_1}}{\partial r_1} = \frac{2r_1}{\pi E^2\sqrt{4\beta^2 - (\frac{r_1^2}{E^2} - 1 - \beta^2)^2}}, & |1-\beta|E < r_1 < (1+\beta)E; \\ 0, & \text{elsewhere.} \end{cases} \tag{4.21}$$

Similar to $H_\Gamma(\theta_J)$, we define

$$H_{R_1}(\theta_J) = \begin{cases} 1, & \text{if } R_1 \leq r_1; \\ 0, & \text{otherwise.} \end{cases} \tag{4.22}$$

For $|1-\beta|E < r_1 < (1+\beta)E$ and $-\pi < \theta_J \leq \pi$, we have

$$H_{R_1}(\theta_J) = \sum_{l=-\infty}^{+\infty} \Pi(\frac{\theta_J - (\pi - \theta) - l2\pi}{\pi - \theta_{R_1}}), \; -\pi < \theta_J \leq \pi; \tag{4.23}$$

and

$$\begin{aligned} \frac{\partial H_{R_1}}{\partial r_1} \quad = \quad & \textstyle\sum_{l=-\infty}^{+\infty}[\delta(\theta_J - (\pi-\theta) - l2\pi + \pi - \theta_{R_1}) + \delta(\theta_J - (\pi-\theta) - l2\pi - (\pi - \theta_{R_1}))]\times \\ & (-\frac{\partial\theta_{R_1}}{\partial r_1}) \\ & -\pi < \theta_J \leq \pi. \end{aligned} \tag{4.24}$$

We can also write the inverse function of θ_{R_1},

$$\theta_{R_1^{-1}}(\theta_{r_1}) = E\sqrt{1+\beta^2+2\beta\cos(\theta_{r_1})}. \tag{4.25}$$

Similarly, for R_2 we have

$$\begin{aligned} R_2^2 &= |\mathbf{Y}^{(i)}e^{-j(\theta+\theta_T^{(i-1)})}|^2 \\ &= |E(e^{j(\theta_T^{(i-1)}+2\theta)}+\beta e^{j\theta'_J})e^{-j(\theta+\theta_T^{(i-1)})}|^2 \\ &= |E(e^{j\theta}+\beta e^{j(\theta'_J-\theta_T^{(i-1)}-\theta)})|^2 \\ &= |E(e^{j\theta}+\beta e^{j\theta_J})|^2 \\ &= E^2[(\cos\theta+\beta\cos\theta_J)^2+(\sin\theta+\beta\sin\theta_J)^2] \\ &= E^2[1+\beta^2+2\beta\cos(\theta-\theta_J)]. \end{aligned} \tag{4.26}$$

As well, if $r_2 \geq (1+\beta)E$, $Prob\{R_2 \leq r_2\} = 1$. If $r_2 \leq |1-\beta|E$, $Prob\{R_2 \leq r_2\} = 0$. If $|1-\beta|E < r_2 < |1+\beta|E$, then we have

$$Prob\{R_2 \leq r_2\} = Prob\{\cos(\theta_J-\theta) \leq \cos\theta_{R_2}\},$$

where

$$\theta_{R_2} = \arccos[(\frac{r_2^2}{E^2}-1-\beta^2)\frac{1}{2\beta}].$$

This implies a symmetrical region of θ_J centered at $\theta+\pi$ with a width of $\pi-\theta_{R_2}$ on either side in which $R_2 \leq r_2$. It can be seen that the CDF and PDF of R_2 are the same as those of R_1. That is, the CDF of R_2 is

$$P_{R_2}(r_2) = P_{R_1}(r_2). \tag{4.27}$$

The PDF of R_2 is

$$p_{R_2}(r_2) = p_{R_1}(r_2). \tag{4.28}$$

As was done above, we define

$$H_{R_2}(\theta_J) = \begin{cases} 1, & \text{if } R_2 \leq r_2; \\ 0, & \text{otherwise.} \end{cases} \tag{4.29}$$

For $|1-\beta|E < r_2 < (1+\beta)E$ and $-\pi < \theta_J \leq \pi$, we have a slightly different result from H_{R_1},

$$H_{R_2}(\theta_J) = \sum_{l=-\infty}^{+\infty} \Pi(\frac{\theta_J - (\pi+\theta) - l2\pi}{\pi - \theta_{R_2}}),\ -\pi < \theta_J \leq \pi, \tag{4.30}$$

and

$$\begin{aligned}\frac{\partial H_{R_2}}{\partial r_2} &= \sum_{l=-\infty}^{+\infty}[\delta(\theta_J - (\pi+\theta) - l2\pi + \pi - \theta_{R_2}) + \delta(\theta_J - (\pi+\theta) - l2\pi - (\pi - \theta_{R_2}))]\times \\ &\quad (-\frac{\partial \theta_{R_2}}{\partial r_2}),\ -\pi < \theta_J \leq \pi.\end{aligned} \tag{4.31}$$

We can also write the inverse function of θ_{R_2},

$$\theta_{R_2^{-1}}(\theta_{r_2}) = E\sqrt{1+\beta^2+2\beta\cos(\theta_{r_2})}. \tag{4.32}$$

4.2.3 The Joint Distribution and the Expectation

Using H_Γ, H_{R_1} and H_{R_2} defined previously, we have the joint CDF of Γ, R_1 and R_2,

$$P_{\Gamma,R_1,R_2}(\gamma, r_1, r_2) = \frac{\int_{-\pi}^{+\pi} H_\Gamma(\theta_J) H_{R_1}(\theta_J) H_{R_2}(\theta_J) d\theta_J}{2\pi}. \tag{4.33}$$

The joint PDF, $p_{\Gamma,R_1,R_2}(\gamma, r_1, r_2)$, has a somewhat unconventional form. Considering

$$p_{\Gamma,R_1,R_2}(\gamma, r_1, r_2)d\gamma dr_1 dr_2 = Prob\{\gamma < \Gamma \leq \gamma + d\gamma, r_1 < R_1 \leq r_1 + dr_1, r_2 < R_2 \leq r_2 + dr_2\},$$

we can see that $p_{\Gamma,R_1,R_2}(\gamma, r_1, r_2)$ is nonzero only over a line (or several lines) in the three dimensional space which consists of values of γ, r_1 and r_2. In fact, it can be shown that over these lines, the PDF assumes infinite values.

For the analysis of the BER performance, all we need know is the expectation of

$G(\Gamma, R_1, R_2)$ where G is assumed to be an arbitrary continuous function. This will be shown later. When $0 < \theta < \pi$ and $\beta \neq 1$, the expectation is given by

$$\begin{aligned}
\overline{G} &= \int\int\int_{-\infty}^{+\infty} G(\gamma, r_1, r_2) p_{\Gamma,R_1,R_2}(\gamma, r_1, r_2) d\gamma dr_1 dr_2 \\
&= \int \int_{|1-\beta|E}^{(1+\beta)E} dr_1 dr_2 \int_{\gamma_1}^{\gamma_2} G(\gamma, r_1, r_2) \frac{\partial^3 P_{\Gamma,R_1,R_2}}{\partial\gamma \partial r_1 \partial r_2} d\gamma \\
&= \frac{1}{2\pi} \int \int_{|1-\beta|E}^{(1+\beta)E} dr_1 dr_2 \int_{\gamma_1}^{\gamma_2} d\gamma G(\gamma, r_1, r_2) \int_{-\pi}^{\pi} \frac{\partial H_\Gamma(\theta_J)}{\partial\gamma} \frac{\partial H_{R_1}(\theta_J)}{\partial r_1} \frac{\partial H_{R_2}(\theta_J)}{\partial r_2} d\theta_J \\
&= \frac{1}{2\pi} \int \int_{|1-\beta|E}^{(1+\beta)E} dr_1 dr_2 \int_{-\pi}^{\pi} \frac{\partial H_{R_1}(\theta_J)}{\partial r_1} \frac{\partial H_{R_2}(\theta_J)}{\partial r_2} d\theta_J \times \\
&\quad \int_{\gamma_1}^{\gamma_2} d\gamma G(\gamma, r_1, r_2)(\delta(\theta_J + \theta_\Gamma) + \delta(\theta_J - \theta_\Gamma))(\frac{\partial\theta_\Gamma}{\partial\gamma}) c(\beta) \\
&= \frac{1}{2\pi} \int \int_{|1-\beta|E}^{(1+\beta)E} dr_1 dr_2 \int_{-\pi}^{\pi} \frac{\partial H_{R_1}(\theta_J)}{\partial r_1} \frac{\partial H_{R_2}(\theta_J)}{\partial r_2} d\theta_J \times \\
&\quad \int_0^{\pi} d\theta_\Gamma G(\theta_{\Gamma^{-1}}(\theta_\Gamma), r_1, r_2)(\delta(\theta_J + \theta_\Gamma) + \delta(\theta_J - \theta_\Gamma))^1 \\
&= \frac{1}{2\pi} \int \int_{|1-\beta|E}^{(1+\beta)E} dr_1 dr_2 \int_{-\pi}^{\pi} \frac{\partial H_{R_1}(\theta_J)}{\partial r_1} \frac{\partial H_{R_2}(\theta_J)}{\partial r_2} G(\theta_{\Gamma^{-1}}(|\theta_J|), r_1, r_2) d\theta_J{}^2 \\
&= \frac{1}{2\pi} \int_{|1-\beta|E}^{(1+\beta)E} dr_2 \int_{-\pi}^{\pi} d\theta_J \int_0^{\pi} d\theta_{R_1} \times \\
&\quad G(\theta_{\Gamma^{-1}}(|\theta_J|), \theta_{R_1^{-1}}(\theta_{R_1}), r_2)(\delta(\arg(e^{j(\theta_J+\theta)}) + \theta_{R_1}) + \delta(\arg(e^{j(\theta_J+\theta)}) - \theta_{R_1})) \\
&= \frac{1}{2\pi} \int_{|1-\beta|E}^{(1+\beta)E} dr_2 \int_{-\pi}^{\pi} d\theta_J G(\theta_{\Gamma^{-1}}(|\theta_J|), \theta_{R_1^{-1}}(|\arg(e^{j(\theta_J+\theta)})|), r_2)^3 \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} G(\theta_{\Gamma^{-1}}(|\theta_J|), \theta_{R_1^{-1}}(|\arg(e^{j(\theta_J+\theta)})|), \theta_{R_2^{-1}}(|\arg(e^{j(\theta_J-\theta)})|)) d\theta_J.
\end{aligned} \tag{4.34}$$

From (4.18), (4.25) and (4.32) we can slightly simplify (4.34) to

$$\overline{G} = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(\theta_{\Gamma^{-1}}(\theta_J), \theta_{R_1^{-1}}(\theta_J + \theta), \theta_{R_2^{-1}}(\theta_J - \theta)) d\theta_J. \tag{4.35}$$

[1] See Appendix A.
[2] See Appendix A.
[3] See Appendix A.

If $\theta = 0$, it is clear that Γ is independent of R_1 and R_2. Thus

$$p_{\Gamma,R_1,R_2}(\gamma, r_1, r_2) = \delta(\gamma) p_{R_1,R_2}(r_1, r_2),$$

where $p_{R_1,R_2}(r_1, r_2)$ is the joint PDF of R_1 and R_2. Note that the joint CDF of R_1 and R_2 is given by

$$P_{R_1,R_2}(r_1, r_2) = \frac{\int_{-\pi}^{+\pi} H_{R_1}(\theta_J) H_{R_2}(\theta_J) d\theta_J}{2\pi}.$$

Then following a procedure similar to that of deriving (4.35), we have

$$\overline{G} = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(0, \theta_{R_1^{-1}}(\theta_J), \theta_{R_2^{-1}}(\theta_J)) d\theta_J. \tag{4.36}$$

When $\theta = 0$, $\Phi = 0$ and $|\mathrm{U}| = \frac{1+\beta^2}{2\beta}$, and $\lim_{x\to 0} \cot x - 1/\sin x = 0$, (for $\beta \neq 1$), so (4.18) gives

$$\theta_{\Gamma^{-1}}(\theta_\gamma) \equiv 0. \tag{4.37}$$

When $\beta = 1$, (4.37) is also valid except for $\theta_\gamma = (2k+1)\pi$, where k is an integer. Since G is a continuous function, (4.35) also applies when $\theta = 0$. From the point of view of numerical computation, (4.36) provides a good approximation when $\theta \approx 0$.

For $\beta = 1$ and $0 < \theta < \pi$ using (4.15), we similarly have

$$\begin{aligned}
\overline{G} &= \tfrac{1}{2\pi} \textstyle\int_0^{\pi} d\gamma \int_{-\pi}^{\pi} G(\gamma, \theta_{R_1^{-1}}(\theta_J + \theta), \theta_{R_2^{-1}}(\theta_J - \theta)) \frac{\partial H_\Gamma(\theta_J)}{\partial \gamma} d\theta_J \\
&= \tfrac{1}{2\pi} [\textstyle\int_{-(\pi-\theta)}^{\pi-\theta} d\theta_J G(0, \theta_{R_1^{-1}}(\theta_J + \theta), \theta_{R_2^{-1}}(\theta_J - \theta)) \\
&\quad + \textstyle\int_{-\pi}^{-(\pi-\theta)} d\theta_J G(\pi, \theta_{R_1^{-1}}(\theta_J + \theta), \theta_{R_2^{-1}}(\theta_J - \theta)) \\
&\quad + \textstyle\int_{\pi-\theta}^{\pi} d\theta_J G(\pi, \theta_{R_1^{-1}}(\theta_J + \theta), \theta_{R_2^{-1}}(\theta_J - \theta))] \\
&= \tfrac{1}{2\pi} [\textstyle\int_{-(\pi-\theta)}^{\pi-\theta} d\theta_J G(0, \theta_{R_1^{-1}}(\theta_J + \theta), \theta_{R_2^{-1}}(\theta_J - \theta)) \\
&\quad + \textstyle\int_{-\theta}^{\theta} d\theta_J G(\pi, \theta_{R_1^{-1}}(\theta_J - \pi + \theta), \theta_{R_2^{-1}}(\theta_J - \pi - \theta))].
\end{aligned} \tag{4.38}$$

Now we examine the relation between (4.35) and (4.38). When $\beta = 1$, $\mathbf{U} = \cos\theta$ and $\Phi = 0$ or π. If $\cos\theta \geq 0$, $\Phi = 0$ and from (4.18),

$$\frac{\cos\theta_\gamma}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi| = (\frac{\cos\theta_\gamma}{\cos\theta} + 1)\frac{1}{\sin|\Phi|} + \cot|\Phi| - \frac{1}{\sin|\Phi|}.$$

Then except for the single point $\theta_\gamma = \pi - \theta$, we have when $\beta \to 1$

$$\lim_{\Phi\to 0} \theta_{\Gamma^{-1}}(\theta_\gamma) = \begin{cases} 0, & \text{if } |\theta_\gamma| < \pi - \theta; \\ \pi, & \text{if } \pi > |\theta_\gamma| > \pi - \theta. \end{cases} \tag{4.39}$$

If $\cos\theta < 0$, $\Phi = \pi$ and from (4.18),

$$\frac{\cos\theta_\gamma}{|\mathbf{U}|\sin|\Phi|} + \cot|\Phi| = (\frac{\cos\theta_\gamma}{-\cos\theta} - 1)\frac{1}{\sin|\Phi|} + \cot|\Phi| + \frac{1}{\sin|\Phi|}.$$

Note $\lim_{x\to\pi} \cot x + 1/\sin x = 0$. Then we can get (4.39) again. In conclusion, (4.38) is the limit form of (4.35) when $\beta \to 1$ and $\beta = 1$ is a continuous point. For numerical computation, (4.38) may be used to provide a good approximation for $\beta \approx 1$.

4.3 Performance of SFH/MDPSK under Multitone Jamming

In this section we consider the performance of uncoded SFH/MDPSK under multitone jamming. The transmitted M-ary DPSK signal has M possible differential phases $2\theta_i$ for $i = 1, \cdots, M$, with equal probability of transmission. The signal is hopped over N frequencies and is jammed with probability ρ. When the signal is jammed, it has the probability distribution calculated in Section 4.2. We assume that there are enough symbols per hop so that the energy loss due to the first dummy symbol of each hop is negligible. We assume that all jamming tones have equal power $I^2/2$. With a total jamming power J available, the number of jammed frequency slots is

$$Q = \frac{J}{I^2/2} = \frac{J}{S\beta^2},$$

where $S = E^2/2$ is the signal power. Suppose the hop frequency spacing is $1/T_s$, where T_s is the M-ary symbol period. Then the total number of hop frequency slots with total spread spectrum bandwidth W_{ss} is

$$N = \frac{W_{ss}}{1/T_s} = W_{ss} T_b \log_2 M,$$

where T_b is the bit period. Then

$$\rho = \frac{Q}{N} = \frac{J/(S\beta^2)}{W_{ss} T_b \log_2 M} = \frac{1}{\log_2 M \beta^2 E_b / J_O}, \tag{4.40}$$

where J_O is the equivalent broadband jamming power spectral density given by

$$J_O = J/W_{ss}.$$

E_b is the signal energy per bit. Therefore β is defined as

$$\beta = \frac{1}{\sqrt{\log_2 M \rho E_b / J_O}}. \tag{4.41}$$

Note that $\rho \leq 1$ is a constraint, which implies $\beta \geq \frac{1}{\sqrt{\log_2 M E_b / J_O}}$. The above result is available in [12]. As well, since Q and N are integers, ρ is not continuous as it appears to be. Nevertheless, when N is large we may assume that ρ is continuous for computational simplicity.

A decision region is specified for each of the M phases representing the M-ary signal. The probability that the received phase falls outside the decision region is the symbol error probability conditioned on the transmission of that signal. The sum of all M such conditional probabilities divided by M and averaged over the jamming state (whether a hop is jammed or not) is the average symbol error probability P_s. Specifically, when there is AWGN with one-side spectral density N_O, (system thermal noise), which is not negligible,

$$\begin{aligned} P_s &= \rho P_{s1} + (1-\rho) P_{s2} \\ &= \rho (P_{s1} - P_{s2}) + P_{s2} \\ &= \frac{1}{\log_2 M E_b / J_O \beta^2} (P_{s1} - P_{s2}) + P_{s2}, \end{aligned} \tag{4.42}$$

$$\beta \geq \frac{1}{\sqrt{\log_2 M E_b / J_O}}.$$

where P_{s1} and P_{s2} are the symbol error rates (SER) conditional on that hop being jammed or not, respectively. P_{s1} is a function of both E_b/J_O and E_b/N_O, and P_{s2} is a function of E_b/N_O only. P_{s1} can be calculated by considering the signal as first being jammed and then further contaminated by the additive noise. Then we can use (4.35) and (4.38) to compute P_{s1}. Let G_i be the SER conditional on $\Gamma = \gamma$, $R_1 = r_1$ and $R_2 = r_2$. Let b_{i1} and b_{i2}, with $b_{i1} < b_{i2}$, be the bounds determining the decision region for differential phase $2\theta_i$. b_{i1} and b_{i2} lie within the particular 2π interval of interest (not necessarily $(-\pi, \pi]$). Then if $2\theta_i$ is transmitted, we have the conditional SER [19]

$$G_i(\gamma, r_1, r_2) = \begin{cases} F(b_{i1}) - F(b_{i2}), & b_{i1} - \theta_i < \gamma < b_{i2} - \theta_i; \\ 1 - F(b_{i2}) + F(b_{i1}), & b_{i1} - \theta_i > \gamma \text{ or } \gamma > b_{i2} - \theta_i, \end{cases} \tag{4.43}$$

where

$$F(b) = \frac{W \sin(\gamma + \theta_i - b)}{4\pi} \int_{-\pi/2}^{\pi/2} dt \frac{e^{-[U - V \sin t - W \cos(\gamma + \theta_i - b) \cos t]}}{U - V \sin t - W \cos(\gamma + \theta_i - b) \cos t}, \tag{4.44}$$

and

$$U = \frac{1}{2}(\eta_2 + \eta_1),\ V = \frac{1}{2}(\eta_2 - \eta_1),\ W = \sqrt{\eta_1 \eta_2},$$

and

$$\eta_1 = \frac{r_1^2 T_b \log_2 M}{2N_O},\ \eta_2 = \frac{r_2^2 T_b \log_2 M}{2N_O}.$$

Then we can write

$$G_i(\gamma, r_1, r_2) = G_i^*(\gamma, \frac{r_1^2 T_b}{2N_O}, \frac{r_2^2 T_b}{2N_O}). \tag{4.45}$$

For $\beta \neq 1$, from (4.35), we have

$$\begin{aligned} P_{s1} \quad = \quad & P_{s1}(\beta, \tfrac{E_b}{N_O}) = \tfrac{1}{M 2\pi} \textstyle\sum_{i=1}^{M} \int_{-\pi}^{\pi} d\theta_J \times \\ & G_i^*(\theta_{\Gamma^{-1}}(\theta_J)|_{\theta=\theta_i}, \tfrac{E_b}{N_O}(1 + \beta^2 + 2\beta \cos(\theta_J + \theta_i)), \tfrac{E_b}{N_O}(1 + \beta^2 + 2\beta \cos(\theta_J - \theta_i))) \end{aligned} \tag{4.46}$$

For $\beta = 1$, from (4.38), we have

$$P_{s1} = \frac{1}{M2\pi}\sum_{i=1}^{M}[\int_{-(\pi-\theta_i)}^{\pi-\theta_i} d\theta_J G_i^*(0, \frac{E_b}{N_O}(1+\beta^2+2\beta\cos(\theta_J+\theta_i)), \frac{E_b}{N_O}(1+\beta^2+2\beta\cos(\theta_J-\theta_i)))$$
$$+ \int_{-\theta_i}^{\theta_i} d\theta_J G_i^*(\pi, \frac{E_b}{N_O}(1+\beta^2-2\beta\cos(\theta_J+\theta_i)), \frac{E_b}{N_O}(1+\beta^2-2\beta\cos(\theta_J-\theta_i)))] \tag{4.47}$$

Similarly, we have

$$P_{s2} = P_{s2}(\frac{E_b}{N_O}) = \frac{1}{M}\sum_{i=1}^{M} G_i^*(\theta_i, \frac{E_b}{N_O}, \frac{E_b}{N_O}). \tag{4.48}$$

Note that for a given E_b/N_O, P_{s2} is a constant and P_{s1} is the function of β.

To determine the worst case ρ, ρ_{wc}, which maximizes P_s for a given E_b/J_O and E_b/N_O, we rewrite (4.42) as

$$\frac{E_b}{J_O}(P_s - P_{s2}(\frac{E_b}{N_O})) = \frac{1}{\log_2 M\beta^2}(P_{s1}(\beta, \frac{E_b}{N_O}) - P_{s2}(\frac{E_b}{N_O})) = \zeta(\beta, \frac{E_b}{N_O}). \tag{4.49}$$

Suppose that, with the constraint $\beta \geq 1/\sqrt{\log_2 M E_b/J_O}$, $\beta = \beta_{wc}$ gives the maximum $\zeta(\beta, \frac{E_b}{N_O}) = \zeta_{max}$. Then

$$\rho_{wc} = \frac{1}{\log_2 M E_b/J_O \beta_{wc}^2}, \tag{4.50}$$

and the worst case SER is

$$P_{s_{wc}} = \frac{\zeta_{max}}{E_b/J_O} + P_{s2}(\frac{E_b}{N_O}) \tag{4.51}$$

Note that, in general, β_{wc} and ζ_{max} are functions of both E_b/N_O and E_b/J_O. In two special case which are commonly encountered, β_{wc} and ζ_{max} are functions of E_b/N_O only.

If $\beta_{wc} = 1/\sqrt{\log_2 M E_b/J_O}$ for a *range* of E_b/J_O (which may occur for small E_b/J_O when, e.g., ζ is a function of β with a single maximum), then $\rho_{wc} = 1$ which corresponds to full band multitone jamming and

$$P_{s_{wc}} = P_{s1}(\frac{1}{\sqrt{\log_2 M E_b/J_O}}, \frac{E_b}{N_O}). \tag{4.52}$$

If $\beta_{wc} = \beta_{wc}(E_b/N_O)$ for a *range* of E_b/J_O (which may occur for large E_b/J_O when, e.g., ζ is a function of β with a single maximum), then

$$\rho_{wc} = \frac{1}{\log_2 M \beta_{wc}^2} \frac{1}{E_b/J_O}, \tag{4.53}$$

which corresponds to an inverse linear function of E_b/J_O with a slope (or the vertical shift in the logarithmic scale) dependent on E_b/N_O, and

$$P_{s_{wc}} = \frac{\zeta_{max}}{E_b/J_O} + P_{s2}(\frac{E_b}{N_O}) \tag{4.54}$$

which corresponds to a similar inverse linear function *plus* a floor SER due to the AWGN.

If the the system thermal noise can be neglected, then $P_{s2} = 0$ and

$$P_{s1} = 1 - \frac{1}{M} \sum_{i=1}^{M} \int_{b_{i1}}^{b_{i2}} \tilde{p}_\Gamma(\gamma)|_{\theta=\theta_i} d\gamma. \tag{4.55}$$

The function $P_{s_{wc}}$ may be optimized with respect to the signal phases and decision regions. For example, for binary DPSK (M=2), we can have: (1) phase $2\theta_1 = \pi/2$ corresponding to 0 and phase $2\theta_2 = 3\pi/2$ corresponding to 1; reasonable decision regions in this case are $[0, \pi]$ for 0 and $(-\pi, 0)$ for 1; or (2) phase $2\theta_1 = 0$ corresponding to 0 and phase $2\theta_2 = \pi$ corresponding to 1; reasonable decision regions are $[-\psi_1, \psi_1]$ for 0 and the rest of the phasor plane for 1. In scheme (2), without thermal noise, $\psi_1 = 0$ would make $P_s = 0$ because when 0 is transmitted, the continuous jamming tone could never alter the transmitted differential phase. Thus in this case inclusion of the thermal noise in the analysis is indispensable and the desirable ψ_1 is greater than 0. The peculiarity in scheme (2) does not exist for similar signal phase schemes with $M > 2$.

To compare system performances for different M we must convert P_s into an equivalent bit error rate (BER), P_b. For a small signal to noise ratio, we can use an orthogonal model which results in

$$P_b \approx \frac{M}{2(M-1)} P_s.$$

If a Gray code is used, so that the Hamming distance of the binary representation of adjacent differential signal phases is 1, we may have, for a large signal to noise ratio[22],

$$P_b \approx \frac{1}{\log_2 M} P_s.$$

4.3.1 Performance Results

We now present some numerical results based on the analysis in the preceding Sections. Performance is measured by the worst case BER for the DPSK signalling scheme, and the specific E_b/J_O and E_b/N_O. Two configurations were given in the preceding section for binary DPSK. The first of these had $2\theta_1 = 0$ and $2\theta_2 = \pi$. Fig. 4.2 shows the worst case BER performance of this scheme for E_b/J_O(dB) from 2 to 28, and E_b/N_O(dB) = 4, 5, 6, 8, 10 and 28. This Figure is the same as Fig. 3 in [17]. Note the error floor due to the noise level. The corresponding plot of E_b/N_O(dB) vs BER for E_b/J_O(dB) = 0, 2, 4, 6, 10, 15 and 20 is given in Fig. 4.3. This plot also shows the error floor, this time due to the fixed E_b/J_O. This figure differs from Fig. 2 in [17] because it uses S/J instead of E_b/J_O, which is used in the Figures in this report. The meaning of S/J is unclear to the authors. The worst case jamming parameter vs E_b/J_O is given in Fig. 4.4. This is the same as Fig. 4 in [17]. The second signalling scheme for binary DPSK is $2\theta_1 = \pi/2$ and $2\theta_2 = 3\pi/2$. The BER performance of this scheme, vs E_b/J_O is given in Fig. 4.5. Comparison with the first scheme shows that $2\theta_1 = 0$ is superior, as indicated in the previous section. Asymptotically, they are identical in performance, but for large E_b/J_O, the first scheme is better, i.e., when $E_b/J_O >> E_b/N_O$. The two previous schemes were evaluated with symmetric decision regions. If we now take the first scheme and modify the decision regions so that $b_{11} = -\pi/4$ and $b_{12} = \pi/4$, and $b_{21} = \pi/4$ and $b_{22} = 7\pi/4$, we get the result shown in Fig. 4.6. The performance of this scheme is worse than those with equal decision regions when E_b/N_O is large, but superior when E_b/N_O is small. Thus the choice of the best DPSK signalling scheme, and decision regions, is dependent upon the relative strength of the noise and tone

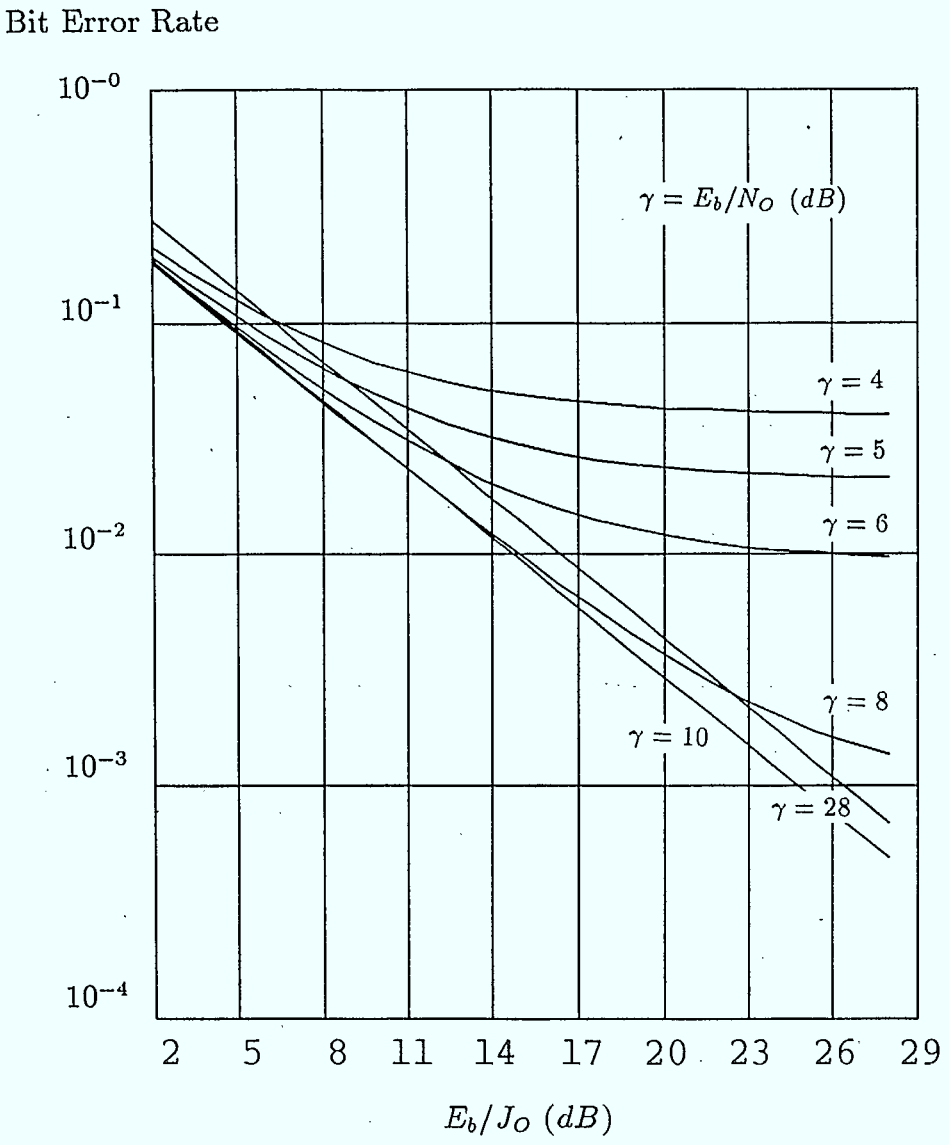

Figure 4.2: Worst Case BER vs E_b/J_O(dB) for E_b/N_O(dB) = 4, 5, 6, 8, 10 and 28, for binary DPSK with $2\theta_1 = 0$ and $2\theta_2 = \pi$. Decision regions are equal and symmetric.

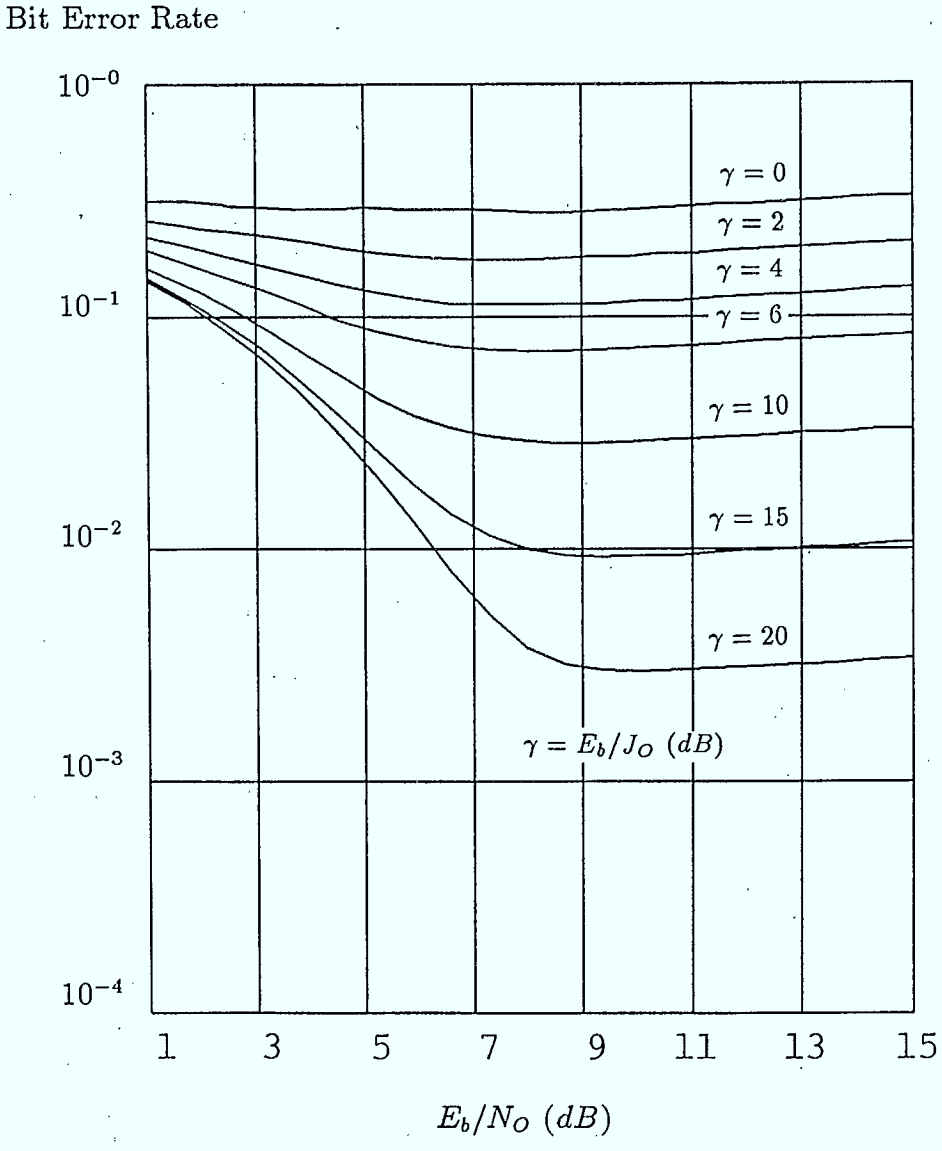

Figure 4.3: Worst Case BER vs E_b/N_O(dB) for E_b/J_O(dB) = 0, 2, 4, 6, 10, 15 and 20, for binary DPSK with $2\theta_1 = 0$ and $2\theta_2 = \pi$. Decision regions are equal and symmetric.

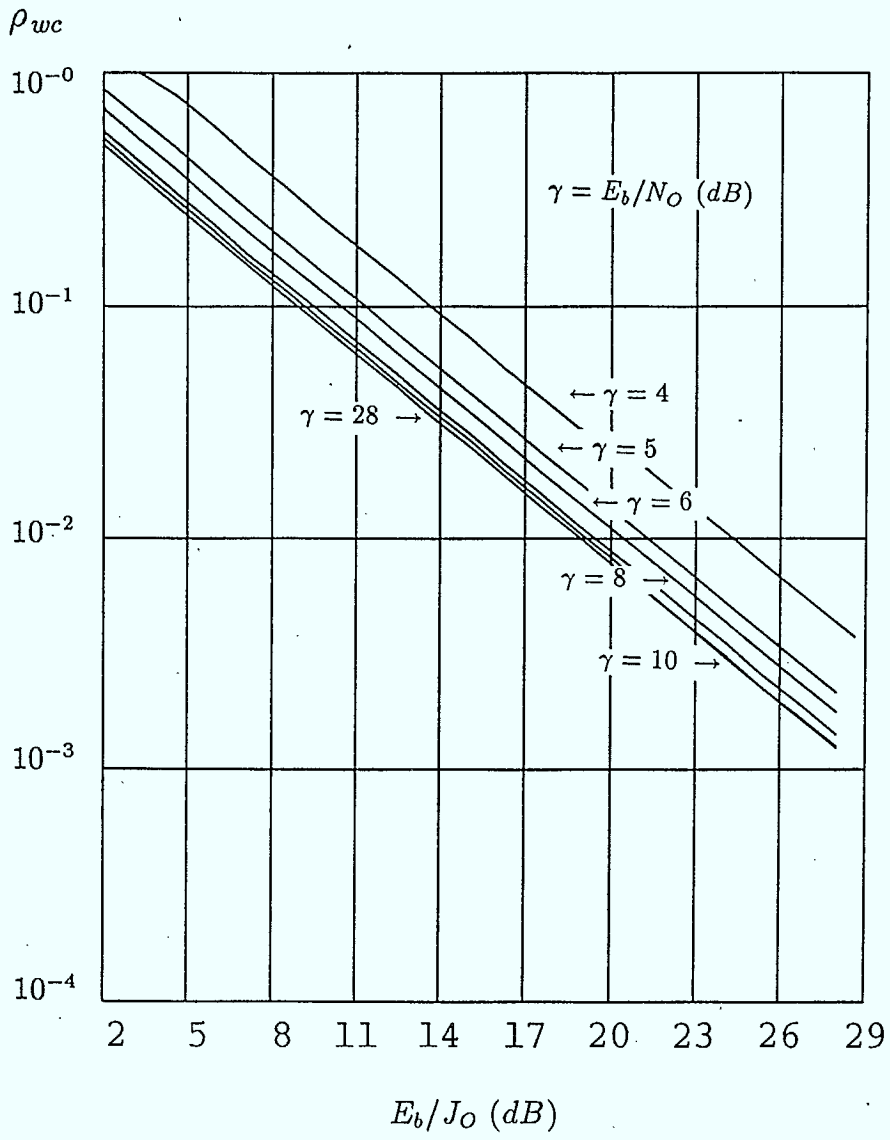

Figure 4.4: ρ_{wc} vs E_b/J_O(dB) for E_b/N_O(dB) = 4, 5, 6, 8, 10 and 28, for binary DPSK with $2\theta_1 = 0$ and $2\theta_2 = \pi$. Decision regions are equal and symmetric.

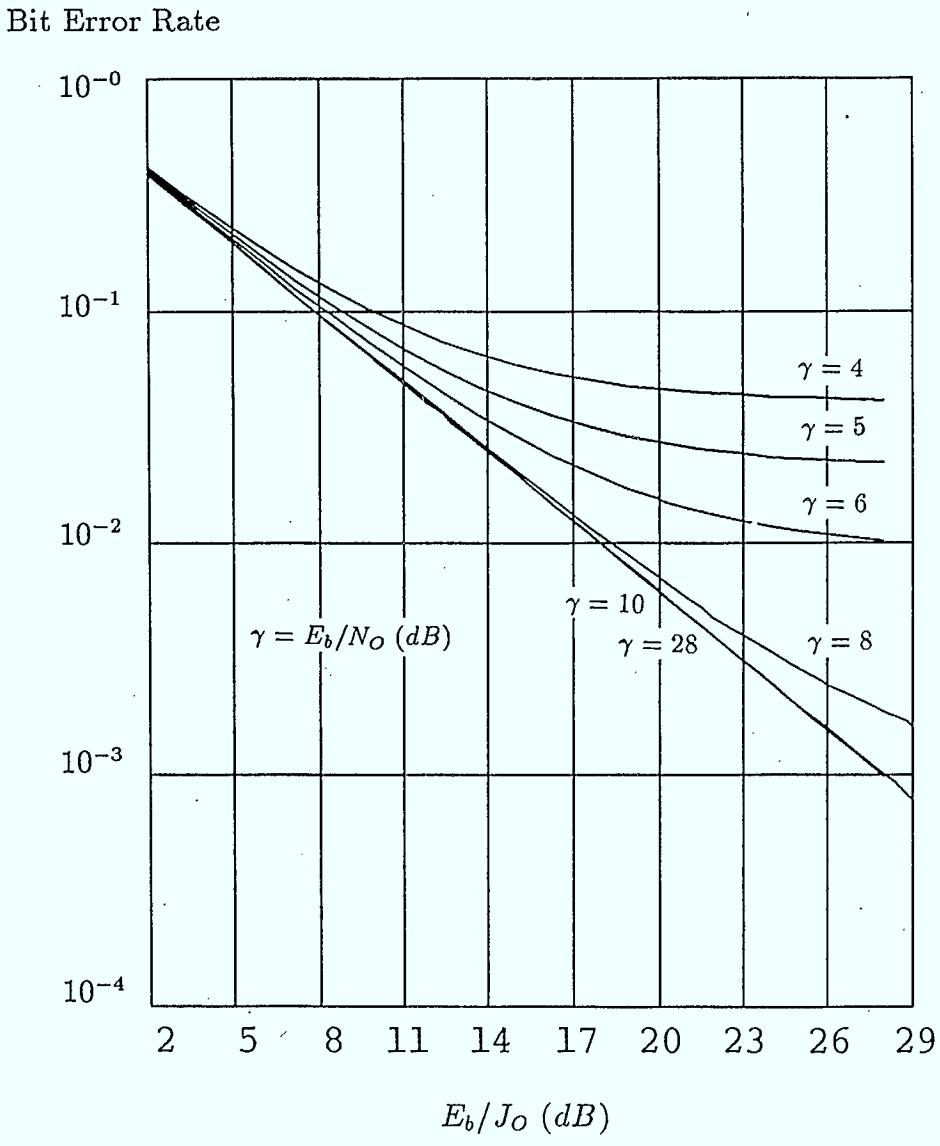

Figure 4.5: Worst Case BER vs E_b/J_O(dB) for E_b/N_O(dB) = 4, 5, 6, 8, 10 and 28, for binary DPSK with $2\theta_1 = \pi/2$ and $2\theta_2 = 3\pi/2$. Decision regions are equal and symmetric.

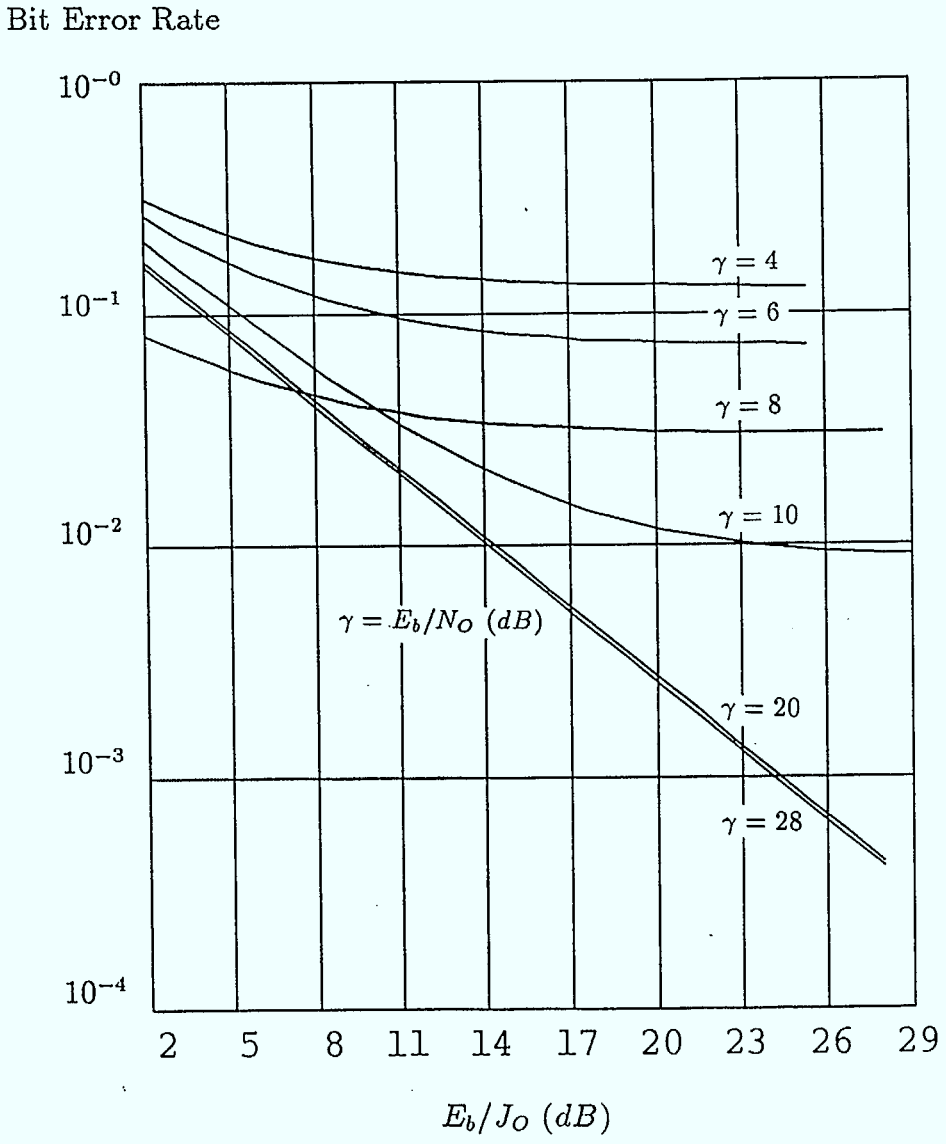

Figure 4.6: Worst Case BER vs E_b/J_O(dB) for E_b/N_O(dB) = 4, 5, 6, 8, 10 and 28, for binary DPSK with $2\theta_1 = 0$ and $2\theta_2 = \pi$. The decision region boundaries are $\pi/4$ and $-\pi/4$.

jamming.

For $M = 4$ we look at two symmetric signalling schemes with equal decision regions. The first has $2\theta_1 = 0$, and the second has $2\theta_1 = \pi/4$. Figs. 4.7 and 4.8 give the BER performance of these schemes, respectively, for $E_b/J_O = 2$ to 28, and $E_b/N_O(\text{dB}) = 4$, 6, 8, 10, 20 and 28. From these Figures, it is clear that choosing $2\theta_1 = 0$ is best when E_b/J_O is large, as was the case for binary DPSK. As expected, the performance of 4-ary DPSK is better than binary DPSK.

4.4 Concluding Remarks

This chapter has addressed some basic problems associated with SFH/DPSK. General probability distributions are derived for arbitrary DPSK signals. Applying these distributions, we have evaluated the performance of SFH/DPSK under both tone jamming and system thermal noise for $M = 2$ and 4. The performance results indicate that choosing $2\theta_1 = 0$ is best under tone jamming, and equal and symmetrical decision regions are best when noise is predominant, with the choice of $2\theta_1$ arbitrary.

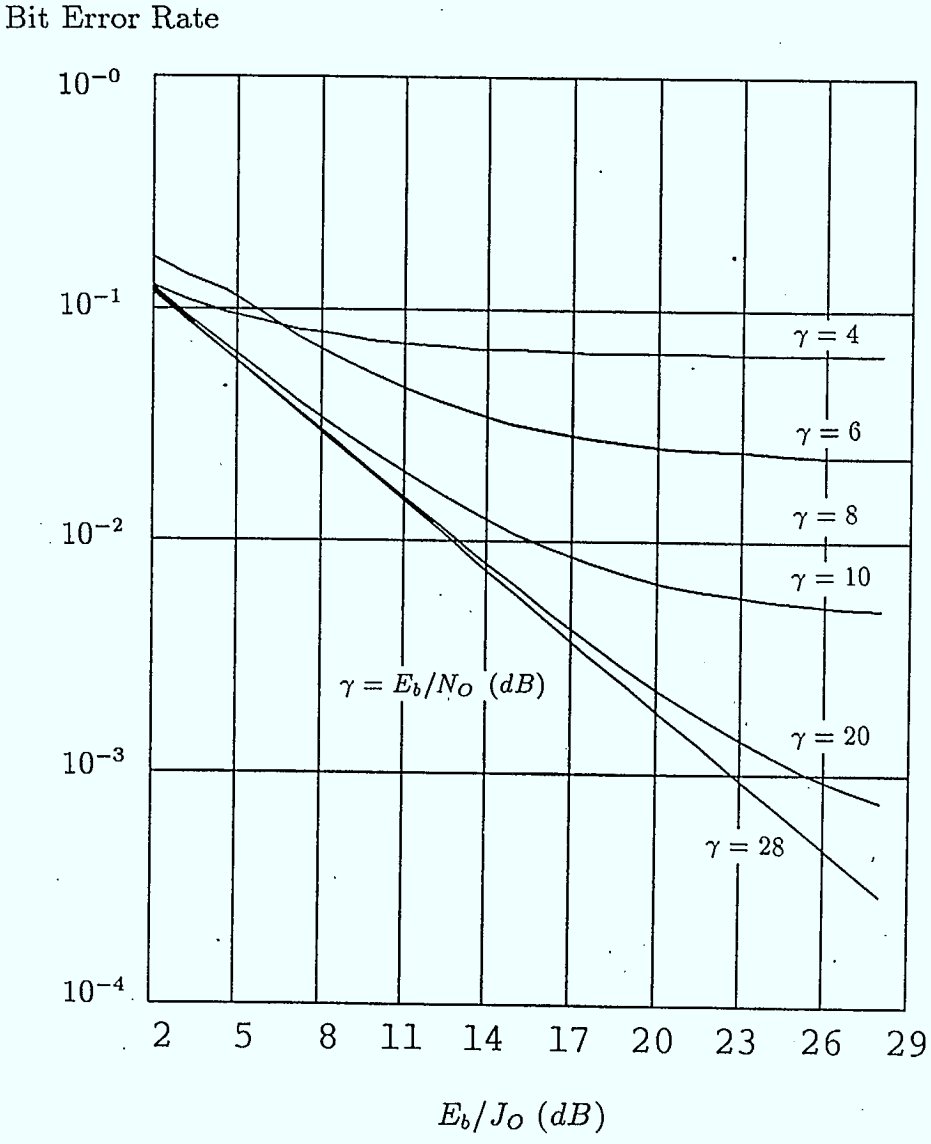

Figure 4.7: Worst Case BER vs E_b/J_O(dB) for E_b/N_O(dB) = 4, 5, 6, 8, 10 and 28, for 4-ary DPSK with $2\theta_1 = 0$. The decision regions are equal and symmetric.

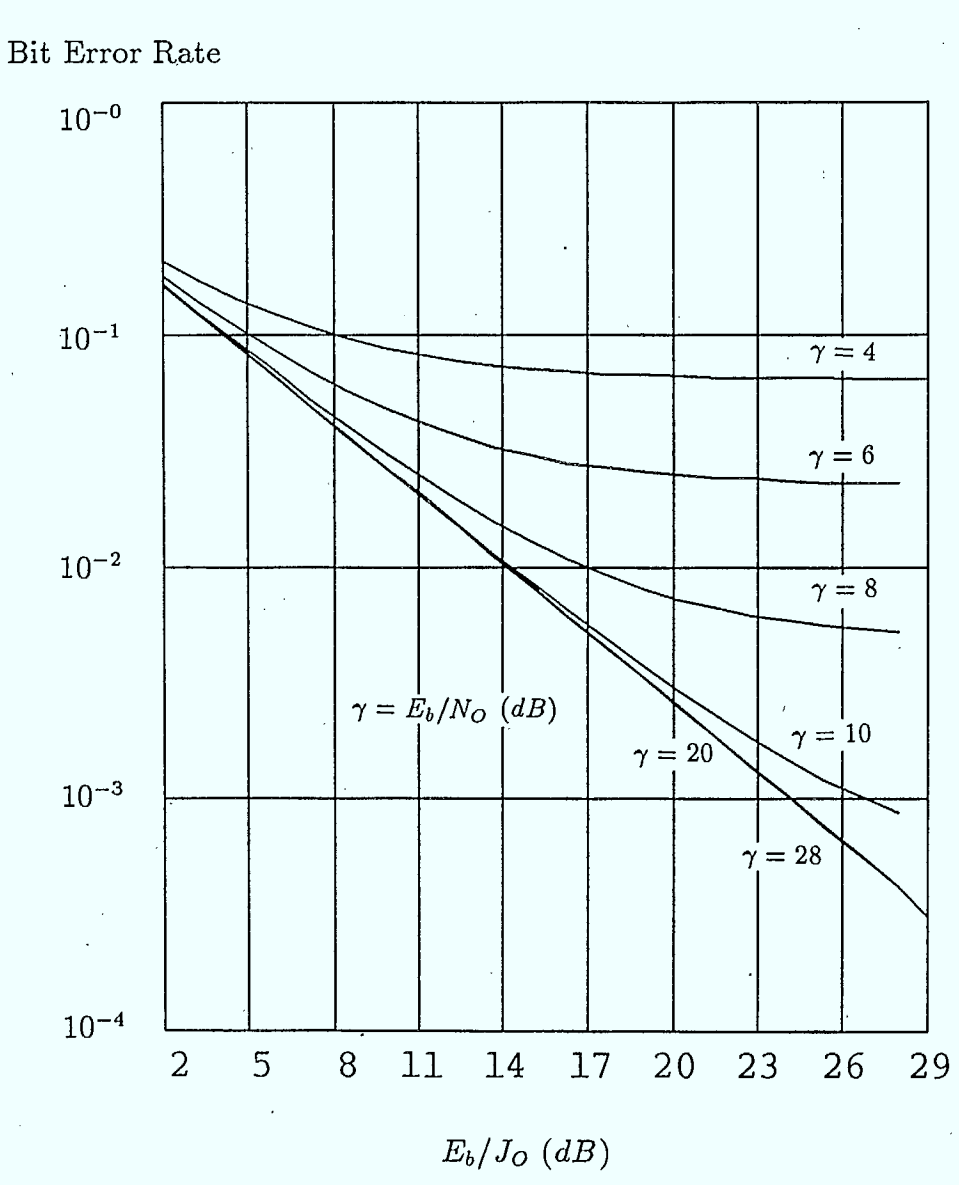

Figure 4.8: Worst Case BER vs E_b/J_O(dB) for E_b/N_O(dB) = 4, 5, 6, 8, 10 and 28, for 4-ary DPSK with $2\theta_1 = \pi/4$. The decision regions are equal and symmetric.

Chapter 5

Suggestions for Future Work

5.1 Coding for Slow Frequency Hopping Systems

To analyse the performance of coded SFH/DPSK, we will first investigate some error exponent type of bounds. These bounds will facilitate the performance evaluation of a specific coding scheme. They may also provide information on the expected implementation complexity to meet certain performance requirements. They can also be used to optimize system parameters such as the code rate.

5.1.1 Reed-Solomon Codes

Continuing with our work on the probability distribution of SFH/DPSK, we will study the coded symbol error rate. One coded symbol may consist of several channel symbols. Under tone jamming, channel symbol errors occur in a correlative way which complicates the calculation of the coded symbol error rate. The use of codes such as Reed-Solomon codes may be quite effective in such channels.

5.1.2 Coding with Deep Interleaving

Random error correcting codes are usually used to combat bursty errors through interleaving. Since jammed SFH/DPSK signals can have long error bursts, the interleaving

used should have a large depth. Unfortunately, interleaving all coded symbols over one hop is costly in terms of complexity, interleaver cost and decoding delay. The benefits of this scheme are improved performance and robustness as far as error burst lengths are concerned. Thus it may be used as a benchmark to compare with in assessing other coded systems.

5.1.3 Long Error Correcting Codes to Correct Both Burst and Random Errors

In this technique, a long codeword is continually transmitted over several hops without interleaving. Then the longest error burst is limited to the number of bits per hop, and the code should be designed to correct bursts of that length. If the error bursts are short and frequent, however, the burst error correcting code performance is degraded. Thus the task in this section is to design a combined code that can efficiently correct both burst and random errors. This type of coding is also attractive if the jamming signal level is close to the level of system thermal noise, which causes random errors.

5.1.4 Diversity and Coding

This is described in section 5.4.1.

5.2 Error Correcting Codes for a High Channel Error Rate

In order to correct errors with an error rate as high as 10^{-1}, low rate codes will be considered. Low rate codes can provide larger minimum distances (or free distances for convolutional codes), which result in increased error correcting power. To verify the correctness of decoded information, which may be control commands, an error detection code can be used as an inner code concatenated with an error correction outer code. At the receiving end, error correction is performed first, and then the decoded output is tested by the error detection circuit. This will generate a highly reliable command whenever errors

can be detected.

An alternative to this method could be to repeat a coded command several times. After error correction decoding, the copies of the same command are compared and a majority vote taken to decide which is the most likely command transmitted.

5.3 Implementation of CODECs

We will concentrate on the design of Reed-Solomon CODECs and the implementation of a Galois Field processor. In addition, implementation using Xilinx technology will continue.

5.4 Communications Over An Intentional Interference Channel

In this section we propose various approaches to the general problem of communicating over an intentional interference channel. These will be pursued to provide a framework in which the work for particular systems of interest may be evaluated. The approaches taken here are quite theoretical and are intended to provide insights into certain aspects of the general problem, insights that can hopefully be of use to the central consideration of the work. Several lines of work have been identified that might be of interest and use and two of these are introduced here. Section 5.4.1 considers the problem of diversity versus coding, a problem that has been of considerable interest to the literature in this area for some time. Numerous papers report results on the problem, although most consider a large number of coding and diversity schemes and give probability of error curves and reach conclusions based on these. A more "universal" approach is proposed here. Subsection 5.4.2 has a few comments on the problems of modeling and evaluating the appropriateness of jamming models. It is felt that such an area of research might be of use to test the robustness of system models shown to perform acceptably well in the more conventional channel models.

5.4.1 Diversity versus Coding

Typically a spread spectrum system will operate in an environment that requires the use of large amounts of redundancy for successful communication. Such redundancy can be achieved by coding or diversity (repetition coding) or a combination of the two (concatenated coding). Numerous studies have evaluated the performance of a variety of specific diversity and coding schemes and some of these are reported in the volumes of Simon et. al. [12]. The problem is of such complexity that it is difficult, at best, to gain insight.

In order to alleviate the problems that some of these studies have, an information-theoretic approach is sometimes used [23] where the minimum SNR to achieve either capacity or computational cut-off rate in the presence of partial band jamming, say, is computed. This approach appears to be very interesting although it is not clear how closely the intuition achieved from such a study matches reality. Some relevant work on this approach that is in progress [24] uses the simplified interference channel model of Chase and Ozarow [25] in an attempt to gain further insight. This work includes consideration of the coherent and noncoherent systems and the effects of both coding and diversity. Although this study was not motivated by the present contract, the results should prove interesting to it and will hopefully suggest lines of inquiry of more direct interest to this work.

Another possible approach to the diversity/coding problem is studied in the work of Chase [26], which considered code combining for a packet network application. While this approach was not felt to be of interest to spread spectrum situations, many of the observations mentioned there are most relevant. In particular, he compared the minimum distance obtainable from a "pure" code (no diversity) to that obtainable from a repetition code (diversity) with a given fixed code of some rate. For convolutional codes he noted that the well known rate $1/2, K = 7$ convolutional code has a free distance of 10 and achieves the upper bound. With diversity 8 a code with rate $1/166$ and free distance 80 is obtained while the maximum free distance possible is 82. For diversity 64 the code rate is $1/128$ and

the free distance is 640 as opposed to a maximum possible of 658. Similar observations are made for binary block codes. The implications of these observations are that, for such low rate codes, there is little to be gained from "pure" coding since the combination of diversity and coding will be so much easier to decode and the performance difference negligible.

This approach can be modified as follows. An asymptotic form of the Varshamov-Gilbert bound shows that, for large values of block length n, there exists an infinite sequence of codes with rate at least R and minimum distance d as long as

$$R \geq 1 - H(d/n), 0 \leq R \leq 1, 0 \leq d/n \leq 1/2, \tag{5.1}$$

where $H(x) = -xlog_2(x) - (1-x)log_2(1-x), 0 \leq x \leq 1$, [21]. For binary convolutional codes of rate R and constraint length K we have the Heller upper bound on the free distance,

$$d_f \leq \min_h \frac{2^h}{2^h - 1} \frac{K + h - 1}{2R}. \tag{5.2}$$

Assuming for the moment that codes exist that meet these bounds, these expressions allow an accurate determination of the trade-off between the minimum distance of "pure" and concatenated codes. It should be possible to take these trade-offs and translate them into performance trade-offs in a direct manner. It should be noted that the Varshamov-Gilbert lower bound is an existence result only and most well known classes of codes fall well short of it. However, it is expected that the performance differences between actual codes with and without diversity will closely follow the performance differences of the assumed codes with and without diversity. For schemes using diversity, of course, the effect of the combining loss will have to be considered. Hopefully this can be done in a reasonable manner to yield some useful insights into the problem. This approach should provide, at the least, one more tool to consider in the complex analysis of these systems.

5.4.2 Interference Channel Modeling

The problem of designing systems to operate in an intentional interference environment is complex. The system performance is determined by its performance in the

worst-case scenario and thus the systems must be robust, operating in an acceptable manner in all possible environments. The modeling of these environments is thus an important aspect of the problem.

The usual partial band noise model assumes the jammer has available a noise source of two-sided power spectral density $J_O/2$. The jammer is able to adjust the noise source to achieve a power spectral density of $J_O/2\rho$ over a fraction ρ of the band for any ρ, $0 < \rho \leq 1$. It is sometimes assumed in the simplest form of this conventional jamming model, that when the transmission is not jammed, perfect reception is made. This last assumption is often modified by including a background or thermal noise component to the total received noise and this complicates matters further.

The model of Chase and Ozarow mentioned in the previous section [25] appears to be simpler than the conventional one, yet intuitively useful. It assumes that, when there is no jamming, which happens with probability $1 - \rho$, signal plus noise is received where the noise has a power spectral density of $J_O/2$. In the presence of jamming, which happens with probability ρ, the received signal consists of noise only. The assumption is that in the presence of jamming the noise completely masks the signal. This model has the advantage that only one noise power spectral density is involved and yet accounts for the effects of the jamming. As well, this model appears to be simpler than the conventional one, yet it has intuitive appeal. It would be of interest to reconsider the performance of systems evaluated for the conventional model to determine the performance sensitivity to such model changes. The lack of robustness of such systems might have serious implications. As mentioned in the previous section, the Chase-Ozarow investigation [25], which considers the problem only from an information-theoretic point of view, should prove interesting.

There has been a considerable amount of recent work on the modeling channels, inspired by the interference channel model. This includes channels with block interference [27] and game theoretic approaches to jamming situations [28]. More recently there has

been growing interest in the study of Arbitrarily Varying Channels (AVC's) (e.g. [29],[30]). Again, much of the interest of this work has been of an information-theoretic nature, proving coding bounds and error rate exponents etc. It may be however that some of this work will have practical implications and it is intended to monitor this area for future consideration.

5.4.3 Comments

Section 5.4 of has attempted to outline some approaches to the investigation of the problem of coding and diversity on intentional interference channels with the view that the results obtained will be of use to the systems of interest in this contract. It is also intended that the approaches of subsections 5.4.1 and 5.4.2 be somehow aligned when they have progressed far enough so that coding and diversity can be considered for a wider variety of channel models than is presently the case.

Bibliography

[1] Wang, Q., Gulliver, T.A., Bhargava, V.K. and Little, W.D., "Coding for Frequency Hopped Spread Spectrum Satellite Communications", *Final Report prepared for the Department of Communications of Canada under DSS Contract No. 27ST.36001-6-3539*, April, 1988.

[2] L.E. Miller, J.S. Lee, and A.P. Kadrichu, "Probability of Error Analyses of BFSK Frequency-Hopping System with Diversity Under Partial-Band Jamming Interference – Part III: Performance of a Square-Law Self-Normalizing Soft Decision receiver," *IEEE Trans. Commun.*, Vol. COM-34, no.7, July 1986, pp. 669-675.

[3] K.S. Gong, "Performance of Diversity Combining Techniques for FH/MFSK in Worst Case Partial Band Noise and Multi-Tone Jamming," *Proc. IEEE Milcom,* 1983, pp.17-21.

[4] R. Viswanathan and K. Taghizadeh, "Diversity Combining in FH/BFSK Systems to Combat Partial Band Jamming," *IEEE Trans. Commun*, Vol. COM-36, no. 9, Sept. 1988, pp. 1062-1069.

[5] C.M. Keller and M.B. Pursley, "Clipped Diversity Combining for Channels with Partial-Band Interference — Part I: Clipped-Linear Combining," *IEEE Trans. Commun.,* Vol. COM-35, no.12, Dec. 1987, pp.1320-1328.

[6] Lee, J.S., French, R.H. and Miller, L.E., "Probability of Error Analysis of a BFSK Frequency Hopping System with Diversity Under Partial Band jamming Interference - Part I: Performance of Square-Law Linear Combining Soft Decision Receivers", *IEEE Trans. Commun.*, Vol. COM-32, No. 6, June 1984, pp. 645-653.

[7] Lee, J.S., Miller, L.E. and Kim, Y.K., "Probability of Error Analysis of a BFSK Frequency Hopping System with Diversity Under Partial Band jamming Interference - Part II: Performance of Square-Law Nonlinear Combining Soft Decision Receivers", *IEEE Trans. Commun.*, Vol. COM-32, No. 12, Dec. 1984, pp. 1243-1250.

[8] A.D. Whalen, *Detection of Signals in Noise,* Academic Press, New York, 1971.

[9] Bird, J.S. and Felstead, E.B., "Antijam Performance of Fast Frequency-Hopped M-ary NCFSK – An Overview", *IEEE J. on Selected Areas Commun.*, Vol. SAC-4, No. 2, Mar. 1986, pp 216-233.

[10] Wozencraft, J.M. and Jacobs, I.M., *Principles of Communication Engineering*, Wiley, New York, 1965.

[11] Massey, J.L., "Coding and Modulation in Digital Communications", *Proc. International Zurich Seminar*, 1974.

[12] Simon, M.K., Omura, J.K., Scholtz, R.A. and Levitt, B.K., *Spread Spectrum Communications*, in Three Volumes, Computer Science Press, Rockville, MD, 1985.

[13] Lee, J.S., French, R.H. and Miller, L.E., "Error Correcting Codes and Nonlinear Diversity Combining Against the Worst Case Partial-Band Noise Jammer of Frequency-Hopping MFSK Systems", *IEEE Trans. Commun.*, Vol. COM-36, No. 4, April 1988, pp. 471-478.

[14] Wang, Q., Gulliver, T. A., Bhargava, V. K. and Felstead, E. B., "Coding for Fast Frequency Hopped Noncoherent MFSK Spread Spectrum Communications Under Worst Case Jamming", *Proc. IEEE MilCom*, 1988, pp. 15.4.1 - 15.4.7.

[15] Houston, S. W., "Modulation Techniques for Communication, Part 1: Tone and Noise Jamming Performance of Spread Spectrum M-ary FSK and 2, 4-ary DPSK Waveforms", *NAECON'75 Record*, pp 51-58.

[16] Simon, M.K., "The Performance of M-ary DPSK/FH in the Presence of Partial-Band Multitone Jamming", *IEEE Trans. Commun.*, Vol. COM-30, No. 5, May 1982, pp. 953-958.

[17] Gong, K.S., "Performance Analysis of FH/DPSK in Additive White Gaussian Noise (AWGN) and Multitone Jamming", *Proc. IEEE MilCom*, 1988, pp. 53.4.1 - 53.4.7.

[18] Winters, J. K., "On Differential Detection of M-ary DPSK with Intersymbol Interference and Noise Correlation", *IEEE Trans. Commun.*, Vol. COM-35, No. 1, Jan. 1987, pp. 117-120.

[19] Pawula, R. F., Rice, S. O. and Roberts, J. H., "Distribution of the Phase Angle Between Two Vectors Perturbed by Gaussian Noise", *IEEE Trans. Commun.*, Vol. COM-30, No. 8, Aug. 1982, pp. 1828-1841.

[20] Pawula, R. F., "On the Theory of Error Rates for Narrow-Band Digital FM", *IEEE Trans. Commun.*, Vol. COM-29, No. 11, Nov. 1981, pp. 1634-1643.

[21] MacWilliams, F.J. and Sloane, N.J.A., *The Theory of Error Correcting Codes,* North-Holland Publishing Co., Amsterdam, 1977.

[22] Ziemer, R.E. and Peterson, R.L., *Digital Communications and Spread Spectrum Systems*, MacMillan, New York, NY, 1985.

[23] Omura, J.K. and Levitt, B.K., "Coded Error Probability Evaluation for Antijam Communication Systems", *IEEE Trans. Commun.,*, Vol. COM-30, 1982, pp. 896-903.

[24] Tong, T.Y., "Diversity and Coding for an Interference Model", Master's thesis, University of Waterloo, in progress.

[25] Chase, D. and Ozarow, L.H., "Capacity Limits for Binary Codes in the Presence of Interference", *IEEE Trans. Commun.,* Vol. COM-27, 1979, pp. 441-448.

[26] Chase, D., "Code Combining - A Maximum-Likelihood Decoding Approach for Combining an Arbitrary Number of Noisy Packets", *IEEE Trans. Commun.,* Vol. COM-33, 1985, pp. 385-393.

[27] McEliece, R.J. and Stark, W., "Channels with Block Interference", *IEEE Trans. Infor. Theory,* Vol. IT-30, 1984, pp. 44-53.

[28] McEliece, R.J. and Rodemich, E.R., "A Study of Optimal Abstract Jamming Strategies vs. Noncoherent MFSK", *Proc. IEEE Milcom*, 1983.

[29] Csiszar, I. and Narayan, P., "Arbitrarily Varying Channels with Constrained Inputs and States", *IEEE Trans. Infor. Theory,* Vol. IT-34, 1988, pp. 27-34.

[30] Hughes, B. and Narayan, P., "The Capacity of a Vector Gaussian Arbitrarily Varying Channel", *IEEE Trans. Infor. Theory,* Vol. IT-34, 1988, pp. 995-1003.

[31] Blazek, Z., Little, W.D., Bhargava, V.K. and Gulliver, T.A., *Software Implementation of a (127,99) BCH CODEC*, Technical Report ECE 87-4, University of Victoria, 1987.

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

[33] Michelson, A.M. and Levesque, A.H., *Error-Control Techniques for Digital Communications*, John Wiley & Sons, Inc., New York, 1985. .

[34] Glover, N., *Practical Error-Correction Design for Engineers*, Data Systems Technology, Corp., Broomfield, 1982.

[35] *DSP56000 Digital Signal Processor User's Manual*, Motorola Inc., 1986.

[36] Kloker, K.L., "The Motorola DSP56000 Digital Signal Processor", *IEEE MICRO*, Dec. 1986, pp. 29-46.

[37] *Second-Generation TMS320 User's Guide*, Texas Instruments Incorporated, 1987.

[38] Frantz, G.A., Lin, K.-S., Reimer J.B., and Bradley, J., "The Texas Instruments TMS320C25 Digital Signal Microcomputer," *IEEE MICRO*, Dec. 1986, pp. 10-28.

Appendix A

Notes on the Derivation of (4.34)

1. Here we use the integral transformation specified by

$$\theta_\Gamma(\gamma) = arccos(d), \gamma_1 \leq \gamma \leq \gamma_2,$$

with d given in (4.2).

For $\beta < 1, C(\beta) = 1$, and $\theta_\Gamma(\gamma_1) = 0, \theta_\Gamma(\gamma_2) = \pi$. Thus

$$\begin{aligned} & \int_{\gamma_2}^{\gamma_1} d\gamma G(\gamma, r_1, r_2)(\delta(\theta_J + \theta_\Gamma) + \delta(\theta_J - \theta_\Gamma))\frac{\partial\theta_\Gamma}{\partial\gamma}C(\beta) \\ = & \int_0^\pi d\theta_\Gamma G(\theta_{\Gamma^{-1}}(\theta_\Gamma), r_1, r_2)(\delta(\theta_J + \theta_\Gamma) + \delta(\theta_J - \theta_\Gamma)). \end{aligned} \tag{A.1}$$

For $\beta > 1$, $C(\beta) = 1$ and $\theta_\Gamma(\gamma_1) = \pi, \theta_\Gamma(\gamma_2) = 0$. Thus (A.1) is still valid.

2. For $0 < |\theta_J| \leq \pi$, it is obvious that

$$\begin{aligned} & \int_0^\pi d\theta_\Gamma G(\theta_{\Gamma^{-1}}(\theta_\Gamma), r_1, r_2)(\delta(\theta_J + \theta_\Gamma) + \delta(\theta_J - \theta_\Gamma)) \\ = & G(\theta_{\Gamma^{-1}}(|\theta_J|), r_1, r_2), \end{aligned} \tag{A.2}$$

because either $\theta_J + \theta_\Gamma = 0$ or $\theta_J - \theta_\Gamma < 0$ can be satisfied for some $0 \leq \gamma \leq \pi$, but not both. For $\theta_J = 0$, the right hand side of (A.2) will be $2G(\theta_{\Gamma^{-1}}(|\theta_J|), r_1, r_2)$. Since the integration with respect to θ_J is over a continuous integrand, changing the value of the integrand at a single point of θ_J, i.e., $\theta_J = 0$, will not affect the integration result. Thus (A.2) is valid for all $|\theta_j| \leq \pi$.

3. Using exactly the same arguments as in item 2, we have

$$\int_0^{\pi} d\theta_{R_1} G(\theta_{\Gamma^{-1}}(|\theta_J|), \theta_{R_1^{-1}}(\theta_{R_1}), r_2)(\delta(arg(e^{j(\theta_J+\theta)}) + \theta_{R_1}) + \delta(arg(e^{j(\theta_J+\theta)}) - \theta_{R_1}))$$

$$= \quad G(\theta_{\Gamma^{-1}}(|\theta_J|), \theta_{R_1^{-1}}(|arg(e^{j(\theta_J+\theta)})|), r_2)$$

In this case the *arg* function is used to ensure that the angles concerned are in the range $(-\pi, \pi]$, because θ_{r_1} varies from 0 to π.

Appendix B

A Comparison of the Motorola DSP56000 and the Texas Instruments TMS320C25 Digital Signal Processors for Implementing the (127,99) BCH Code

This Appendix presents algorithms for efficient microprocessor implementations of a decoder. The ability to implement time critical steps in these algorithms is the basis for comparing the DSP56000 and TMS320C25. The DSP56000's comparatively general purpose architecture and certain unique features provide a higher bit rate decoder than can be implemented on the TMS320C25. Assembly language programs were written and then tested for performance and timing using IBM PC based simulators of the processors. A complete decoder was implemented on the DSP56000, achieving an average bit rate in excess of 1 million bits per second.

B.1 Introduction

This Appendix investigates and compares the Motorola DSP56000 and the Texas Instruments TMS320C25 digital signal processors based on their use in implementing a four error correcting (127,99) BCH code decoder. This code was chosen as a good compromise between performance and implementation complexity [31]. Previous to the implementation outlined in this Appendix, two decoders had already been created. The first decoder was implemented in software using the C programming language. The second decoder was implemented in hardware using the XILINX programmable logic device with peripheral memory and logic integrated circuits.

Section B.2 provides some background on the theory of error control coding and digital signal processors. Section B.3 outlines the high level operation of a decoder and presents the decoding algorithms chosen and developed for a digital signal processor implementation. The time critical operations of the decoding algorithms are pinpointed. Section B.4 compares the Motorola DSP56000 and Texas Instruments TMS320C25 on the basis of how well they can perform the time critical operations of the decoding algorithms. Section B.5 presents performance results of the complete decoder which was implemented on the Motorola DSP56000. Results presented in Sections B.4 and B.5 are based on code tested for both processors with IBM PC based simulators. Section B.6 gives the conclusions of the Appendix.

B.2 Background

B.2.1 Error Control Coding Theory

Error control coding is implemented by adding redundant information (parity bits) to a message before transmission. This redundant information is used by the receiver to detect and/or correct errors in the received message. In this Appendix, the decoder implementation corrects errors as opposed to just detecting errors. The number of errors which

can be corrected depends on the error control coding scheme used and is proportional to the amount of redundant information added to a message. A (127,99) BCH code will correct all one, two, three and four bit errors in a single 127 bit received message (which includes the parity bits). A simple block diagram of a digital transmission system incorporating a (127,99) BCH error control code is given in Fig. B.1. The error control coding is accomplished in the encoder block of the transmitter and the decoder block of the receiver.

In a system using (127,99) BCH coding, the encoder receives a message vector m containing 99 bits. The encoder uses the message vector to compute 28 parity bits. The encoder outputs a code vector c containing $127 = 99 + 28$ bits, the parity bits having been tagged onto the end of the message vector. Encoded messages are transmitted one vector at a time. If no errors occur during transmission, the received vector r output by the receiver demodulator is identical to the code vector c. In this case, the decoder simply removes the 28 parity bits from the received vector and outputs the message vector m.

If errors occur during transmission, the received vector r is not identical to the code vector c, as some bits are flipped where errors have occurred. As long as the number of errors is less than or equal to four, the decoder corrects the flipped bits in the message portion of the received vector and outputs the message vector m. (Errors in the 28 parity bits are included in the total number of errors, but are not corrected since the parity bits are not output from the decoder.) If the number of errors is greater than four, the decoder erroneously corrects the received vector. This erroneous correction may or may not be detected. The error correcting power of a decoder should in general be sufficient to make the probability of undetected erroneous correction almost zero.

Implementing the encoder is a trivial problem. The decoder implementation, however, is nontrivial, and increases in complexity with the number of correctable errors. The decoder typically requires significantly more time than the encoder to process data when correcting errors. The maximum achievable data rate for a system is limited by the maxi-

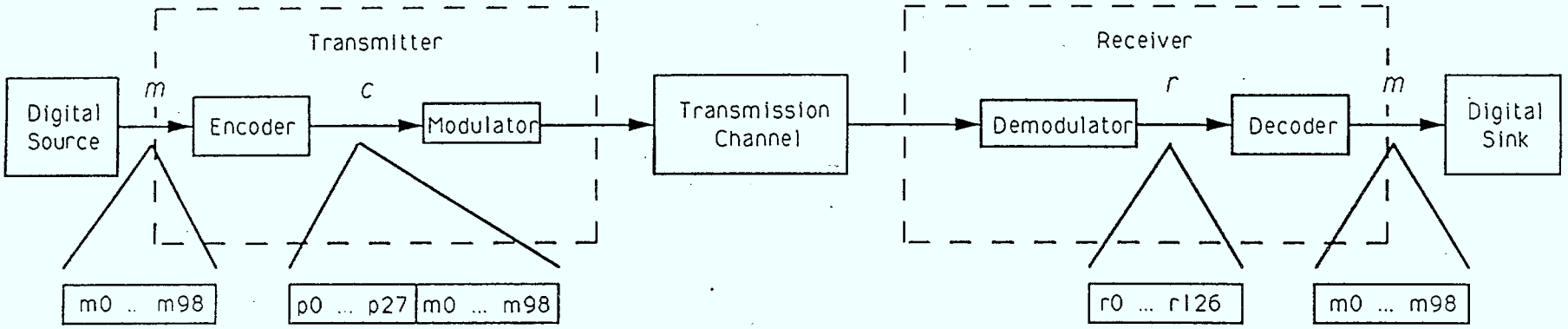

Figure B.1: Block Diagram of a Digital Transmission System Incorporating the (127,99) BCH Error Correcting Code.

mum rate at which the decoder can correct errors. Thus optimizing the decoder for speed is a primary goal in implementation. In general, implementing a decoder is a practical way to evaluate the usefulness of any technology or device for error control coding applications.

The reader is referred to [32] or [33] for further information on error control coding in general and BCH codes in particular.

B.2.2 Digital Signal Processors

Digital signal processors (DSP's) are special purpose microprocessors which have been designed specifically for implementing signal processing functions such as digital filters and Fast Fourier Transforms (FFT). Important features distinguishing DSP's, such as the Motorola DSP56000 and Texas Instruments TMS320C25, from most general purpose microprocessors are fast instruction cycles, parallel and/or pipelined operation, internal memory (RAM and ROM), multiple data/program buses, DSP oriented addressing modes and a fast multiplication circuit. All of these features except the fast multiplication circuit can be exploited in implementing a decoder.

There exist faster and more powerful DSP's than the DSP56000 and TMS320C25. These include the Motorola DSP96002 and the Texas Instruments TMS320C30. However, this comparison serves to identify which decoding algorithms are computationally intensive, and what hardware features are desirable for decoder implementation.

B.3 Decoder Operation and Algorithms

The decoder for a (127,99) BCH code performs the high level algorithm presented in Fig. B.2. The key steps of this algorithm are:

1. computing the syndromes,

2. forming the error locator polynomial, and

3. finding the roots of the error locator polynomial.

Input and output, and correcting the locations of the erroneous bits are comparatively trivial steps and will not be discussed further here. It is the above three steps that constitute the bulk of the decoding process.

```
while (receiving transmitted code vectors)
        input 127 bit received vector
        detect errors by computing syndromes
        if (errors detected)
                locate errors by forming error locator
                        polynomial and finding its roots
                correct errors in received vector
        end if
        output (corrected) 99 bit message vector
end while
```

Figure B.2: The High Level Decoder Algorithm

All three steps are based on mathematical calculations in a 128 element finite field (Galois Field), denoted as GF(2^7). Individual elements in the finite field are represented by a unique 7 bit vector. There are 128 unique 7 bit vectors and thus 128 elements in the finite field. Only 3 finite field mathematical operations are required for the decoder implementation, addition, multiplication and division. The important feature of a finite field, and operations in the finite field, is that the result of any calculation is always one of the elements in the finite field. The reader is referred to [32] or [33] for further information on finite field (Galois Field) theory.

A number of algorithms exist for the decoding steps. Some are oriented to hardware implementation, while others are better for software. The algorithms given in this Appendix are considered the best for a (127,99) BCH code implemented on a microprocessor. The reader is referred to [31] for a thorough discussion of the relative merit of different algorithms

useful for a software implementation. The algorithms for the three decoding steps follow.

B.3.1 The Syndrome Computation Algorithm

The first step in the decoding process is to compute the four syndromes (symptoms of errors), S_1, S_3, S_5 and S_7. If any of the syndromes are nonzero, errors have been detected. The syndromes are 7 bit finite field elements and are computed using the following equations:

$$\begin{aligned} S_1 &= r_0 + r_1\alpha + r_2\alpha^2 + \cdots + r_{126}\alpha^{126}, \\ S_3 &= r_0 + r_1\alpha^3 + r_2\alpha^6 + \cdots + r_{126}\alpha^{378}, \\ S_5 &= r_0 + r_1\alpha^5 + r_2\alpha^{10} + \cdots + r_{126}\alpha^{630}, \\ S_7 &= r_0 + r_1\alpha^7 + r_2\alpha^{14} + \cdots + r_{126}\alpha^{882}, \end{aligned}$$

where r_i is bit i in the received code vector, and α^i is a 7 bit element of the finite field.

An efficient algorithm for computing the syndromes based on the above equations uses table lookup. The individual syndromes are concatenated to form a single 28 bit syndrome word S. Similarly, the coefficients α^i for each bit r_i are concatenated to form 28 bit syndrome masks, M_i. The M_i are explicitly defined as follows:

$$\begin{aligned} M_0 &= [\alpha^0, \alpha^0, \alpha^0, \alpha^0], \\ M_1 &= [\alpha, \alpha^3, \alpha^5, \alpha^7], \\ M_2 &= [\alpha^2, \alpha^6, \alpha^{10}, \alpha^{14}], \\ &\vdots \\ M_{126} &= [\alpha^{126}, \alpha^{378}, \alpha^{630}, \alpha^{882}]. \end{aligned}$$

These syndrome masks are stored in a 127×28 bit table. The algorithm for computing the syndrome word S is given in Fig. B.3. The bit index i runs from 126 down to 0

because in most cases, if the message is being transmitted serially to the decoder, the first bit received is r_{126}. The addition of the 28 bit masks is an extension of finite field addition. After the syndrome word S is computed, the individual syndromes S_1, S_3, S_5 and S_7 are extracted.

```
S = 0
for (i = 126 down to 0)
        if (r_i = 1)
                S = S + M_i
        end if
end for
```

Figure B.3: The Syndrome Computation Algorithm

B.3.2 The Error Locator Polynomial Algorithm

If any of the syndromes computed for a received vector are nonzero, bit errors have been detected. The next step in the decoding process is to form the error locator polynomial. This is done by computing the coefficients σ_i of the error locator polynomial:

$$\sigma(x) = \sigma_4 + \sigma_3 x + \sigma_2 x^2 + \sigma_1 x^3 + x^4.$$

The roots of the error locator polynomial $\sigma(x)$ are the indexes of the erroneous bit locations in the received vector. For example, if one root is the finite field element α^{87}, then the bit r_{87} in the received vector is in error and must be flipped.

The quickest algorithm for computing the error locator polynomial coefficients for a four error correcting BCH code is Peterson's direct solution [33]. The algorithm consists of two steps:

1. determine an estimate of the range of errors that are contained in the received vector.

2. based on this number and the syndrome values, use the appropriate set of equations to compute the polynomial coefficients.

All of these calculations use the the previously computed syndromes.

The first step is to compute the following determinant:

$$determinant = S_3(S_1^3 + S_3) + S_1(S_1^5 + S_5).$$

If the determinant is zero, there are one or two bit errors, and if the determinant is nonzero, there are three or four bit errors.

The second step is to compute the σ_i. Three sets of equations are available:

1. for one or two bit errors detected,
2. for three or four bit errors detected with $S_1 \neq 0$,
3. for three or four bit errors detected with $S_1 = 0$.

The equations for each case are presented in Table B.1.

After computing the σ_i, the exact number of bit errors in the received vector is determined as follows:

$$\begin{aligned} number\ of\ errors &= 4\ if\ \sigma_4 \neq 0, \\ number\ of\ errors &= 3\ if\ \sigma_4 = 0\ and\ \sigma_3 \neq 0, \\ number\ of\ errors &= 2\ if\ \sigma_4 = \sigma_3 = 0\ and\ \sigma_2 \neq 0, \\ number\ of\ errors &= 1\ if\ \sigma_4 = \sigma_3 = \sigma_2 = 0\ and\ \sigma_1 \neq 0. \end{aligned}$$

This covers all possible cases for the computed σ_i.

B.3.3 The Polynomial Root Finding Algorithm

Unfortunately, there are no elegant algorithms for finding the roots of a polynomial defined over a finite field. The standard algorithm is the Chien search, which is really just

1. Two error correcting formulas:

$$\begin{aligned}
\sigma_1 &= S_1, \\
\sigma_2 &= \frac{S_3 + S_1^3}{S_1}, \\
\sigma_3 &= 0, \\
\sigma_4 &= 0.
\end{aligned}$$

2. Four error correcting formulas:

$$\begin{aligned}
\sigma_1 &= S_1, \\
\sigma_2 &= \frac{S_1(S_7 + S_1^7) + S_3(S_1^5 + S_5)}{S_3(S_1^3 + S_3) + S_1(S_1^5 + S_5)}, \\
\sigma_3 &= (S_1^3 + S_3) + S_1\sigma_2, \\
\sigma_4 &= \frac{(S_5 + S_1^2 S_3) + (S_1^3 + S_3)\sigma_2}{S_1}.
\end{aligned}$$

3. Simplified four error correcting formulas for $S_1 = 0$:

$$\begin{aligned}
\sigma_1 &= S_1, \\
\sigma_2 &= \frac{S_5}{S_3}, \\
\sigma_3 &= S_3, \\
\sigma_4 &= \frac{S_5^2 + S_3 S_7}{S_3^2}.
\end{aligned}$$

Table B.1: Formulas for the Error Locator Polynomial Coefficients

an exhaustive search. Each element of the finite field is substituted into the error locator polynomial to determine whether or not it is a root. The efficiency of this algorithm can be increased by:

1. only testing those finite field elements which, if a root, specify an erroneous bit in the 99 message bits of the 127 bit received vector,
2. nested evaluation of the polynomial,
3. degrading the polynomial after a root is found,
4. using a lookup table for one and two error cases.

An algorithm including the above optimization is given in Fig. B.4. Note that i is the bit index corresponding to the received vector and α^i is the finite field element corresponding to the received bit r_i. If α^i is found as a root of $\sigma(x)$, r_i is in error. It is only necessary to degrade $\sigma(x)$ from degree four to three and from degree three to two. The equations for degrading $\sigma(x)$ are presented in Table B.2.

The algorithm presented in Fig. B.4 also includes correction of the bits in error. For implementation purposes, it is more efficient to simply store each error location found and continue the search. When all of the error locations have been found and stored, a bit correction routine uses the stored error locations to determine which bits to flip in the 99 message bits of the 127 bit received vector. This variation of the algorithm is implied by the high level decoding algorithm presented at the beginning of this Section.

B.3.4 Time Critical Components of the Algorithms

Of the three decoding steps, the first and the third are the most computationally intensive. This is because both syndrome computation and polynomial root finding are done on a per bit basis. The implementations of both algorithms must minimize execution time. The syndrome computation algorithm can be considered on an entire codeword basis when

```
i = 126
while (number of errors > 2) and (i > 27)
        β = σ(α^i)
        if (β = 0)
                correct r_i
                degrade σ(x)
                number of errors = number of errors - 1
        end if
        i = i - 1
end while
if (i > 27)
        lookup error locations two and one
        if (error location two > 27)
                correct r_two
        end if
        if (error location one > 27)
                correct r_one
        end if
end if
```

Figure B.4: The Algorithm to Find the Roots of the Error Locator Polynomial

1. Formulas for degrading $\sigma(x)$ from degree 4 to 3:

$$\begin{aligned}
\sigma'_1 &= \sigma_1 + \beta, \\
\sigma'_2 &= \sigma'_1\beta + \sigma_2, \\
\sigma'_3 &= \sigma'_2\beta + \sigma_3, \\
\sigma'_4 &= 0.
\end{aligned}$$

2. Formulas for degrading $\sigma(x)$ from degree 3 to 2:

$$\begin{aligned}
\sigma'_1 &= \sigma_1 + \beta, \\
\sigma'_2 &= \sigma'_1\beta + \sigma_2, \\
\sigma'_3 &= 0, \\
\sigma'_4 &= 0.
\end{aligned}$$

Table B.2: The Polynomial Coefficient Degradation Formulas

attempting to minimize its execution time. The execution time of the root finding algorithm is primarily dependent on how fast finite field multiplies and additions are performed. To adequately compare implementation alternatives, it is generally sufficient to compare how fast syndrome computation, multiplication and addition can be performed.

B.4 Comparison of the Motorola DSP56000 and Texas Instruments TMS320C25

Both the Motorola DSP56000 and the Texas Instruments TMS320C25 processors are designed to efficiently implement standard digital signal processing functions such as digital filters and FFTs. When compared on this basis, the processors achieve comparable processing rates. Neither processor is specifically designed for error correcting decoder implementations. A very basic problem with both processors, as well as with any general purpose microprocessor, is that the mathematical instructions operate on fixed point numbers (on the TMS320C25, integers) as opposed to finite field elements. As discussed previously, the fast implementation of finite field mathematical operations is essential to fast decoding. Implementation alternatives using lookup tables provide the fastest finite field mathematical operations on both processors. Before comparing the DSP56000 and TMS320C25 on the basis of implementing the time critical components of the decoding algorithms, it is useful to provide a brief overview and comparison of their architectures.

B.4.1 The Processor Architectures

This section presents those features of the DSP56000 and TMS320C25 architectures which are relevant to a decoder implementation.

DSP56000

The DSP56000 can execute up to 10.25 million instructions per second based on an instruction cycle time of 97.5 ns. Most instructions take only one instruction cycle. This fast instruction cycle time is achieved by pipelining the fetch, decode and execution operations for a single instruction. During instruction execution the program controller, address arithmetic logic unit (address ALU) and data ALU operate in parallel.

The DSP56000 uses a 24 bit program and data word size. The data ALU has potentially eight independent 24 bit registers. Two of the registers (A and B) are available as 24 bit accumulators to hold the results of 24 bit data ALU operations.

The DSP56000 has three separate 64K-word memory spaces, program memory, X data memory and Y data memory. Some on board memory is provided in each of the memory spaces. Internal program memory consists of 2K words of ROM. Internal X and Y data memories both include 256 words of RAM and 256 words of ROM, so they provide a total of 1K words of internal data memory. Internal program and data memory are important, as they allow zero wait state memory accesses and thus instructions can be performed in one instruction cycle. External memory can also provide zero wait states, but is expensive. Internal memory accesses are performed using three internal 16 bit address buses and four internal 24 bit data buses. This is another advantage of internal memory, allowing program, X data and Y data to be moved in parallel internally. External memory (program, X or Y data) is accessed using a single external 16 bit address bus and a single external 24 bit data bus.

Memory addressing is primarily indirect, using any of the eight 16 bit address registers in the address ALU. Seven indirect addressing modes are provided with three types of address update arithmetic. Each address register has a dedicated index register and address arithmetic register. Four of the addressing modes allow memory to be read or written and the address register to be updated in the same instruction cycle. This is important

for moving through data in an array. Circular buffers are supported in hardware by modulo address update arithmetic. Some direct addressing is supported, but only for memory mapped I/O and writing or storing control registers. This can prove to be a limitation when implementing several distinct variables that are used in the same program section but are not suited to being stored in an array.

The instruction set for the DSP56000 is in many ways close to that of a reduced instruction set computer (RISC), having only 62 basic instructions. The data ALU implements all of the standard fixed point arithmetic instructions and logical instructions. Two important features of the instruction set are parallel data moves and hardware DO loops. Most of the instructions involving the data ALU allow up to two data words to be moved simultaneously with the execution of the instruction. This is facilitated by the multiple internal address and data buses mentioned previously. Nine types of parallel data moves are provided involving registers and/or memory. A single instruction is provided for setting up the execution of program loops. A set of hardware registers and a system stack provide all the necessary control for executing the looping operation as well as nesting loops. Thus no time is lost in loops executing a "decrement counter and branch" instruction.

Both serial and 24 bit parallel input and output (I/O) are provided on the DSP56000. Asynchronous and synchronous serial I/O are implemented using a serial communications interface (SCI) and/or a synchronous serial interface (SSI). The SCI provides asynchronous rates up to 320K bits per second and synchronous rates up to 2.5M bits per second. The SSI provides synchronous rates up to 5M bits per second. Parallel I/O is memory mapped to the high address segment of Y data memory. A single input or output takes a minimum of two instruction cycles. The reader is referred to [35] and [36] for detailed information on the DSP56000.

TMS320C25

The TMS320C25 can execute up to 10 million instructions per second based on an instruction cycle time of 100 ns. Most instructions take only one instruction cycle. This fast instruction cycle time is achieved by pipelining the fetch, decode and execution operations for a single instruction. During instruction execution, the program controller, central arithmetic logic unit (CALU) and auxiliary register arithmetic unit (ARAU) operate in parallel.

The TMS320C25 uses a 16 bit program and data word size. The CALU has one 32 bit accumulator register and two registers used in conjunction with the hardware multiplier. The lack of registers is somewhat compensated by the fact that memory operands can be used as input in CALU instructions. Some instructions involving the accumulator operate only on 16 bits while others operate on the full 32 bits. The groupings of 16 bit and 32 bit instructions does not always make sense. For example, the logical instructions And, Or and Exclusive Or operate only on 16 bits of the accumulator but Logical Shift Left and Logical Shift Right operate on the full 32 bits. This can be a problem in implementing certain algorithms requiring logical operations and shifts on the same 16 bit word. Having only one accumulator is also a definite restriction in that two separate mathematical operations on different data can not be implemented conveniently in parallel.

The TMS320C25 has two separate 64K-word memory spaces: program memory and data memory. Some on board memory is provided in both of the memory spaces. Internal program memory consists of 4K words of ROM. Internal data memory consists of 544 words of RAM. Of this RAM, 256 words can be configured to be part of the program memory space instead of the data memory space. Internal memory accesses are performed using two internal 16 bit address buses and two internal 16 bit data buses. One of each is used for program memory accesses and the other of each is used for data memory accesses. External memory (both program and data) are accessed using a single external 16 bit address bus

and a single external 16 bit data bus.

Memory addressing can be either direct or indirect. In the case of direct addressing, memory is paged such that 128 words of data memory are addressable at any one instant. Indirect addressing is provided using eight 16 bit auxiliary registers which are also used for looping control. Seven addressing modes are provided which include two types of address update arithmetic. One of the auxiliary registers is available as an index register and must be shared between the other seven auxiliary registers. The ARAU performs all address updates, operating in parallel with the CALU. Six of the addressing modes (which encompass both types of address arithmetic) allow memory to be read or written and the auxiliary register to be updated all in the same instruction cycle.

The instruction set for the TMS320C25 contains 133 instructions, over half of which involve the CALU. For many operations, one or two instructions using immediate addressing mode exist as well as a separate instruction performing the same operation with direct or indirect addressing. Separate instructions exist for performing similar or even identical operations on separate operands, as opposed to one instruction operating on several possible operands. For example, instead of there just being a single data movement instruction, several data movement instructions exist which are specific to certain registers in the processor. These aspects of the instruction set alone account for the comparatively large number of instructions. The CALU implements all of the standard fixed point arithmetic instructions and logical instructions. A number of CALU instructions incorporate significant parallelism to speed up digital signal processing algorithms.

Both serial and parallel I/O are provided on the TMS320C25. Synchronous serial I/O is implemented in an on board serial port which can run at rates up to 5M bits per second. Parallel I/O is supported by the IN and OUT instructions which use the external 16 bit address port and 16 bit data port. A single parallel input or output typically takes two instruction cycles. The reader is referred to [37] and [38] for detailed information on

the TMS320C25.

B.4.2 Implementation of the Time Critical Components of the Decoding Algorithms

As was explained in Section B.3.4, syndrome computation, finite field multiplication and finite field addition are the time critical computations of the decoding algorithms. Of these, finite field addition is the simplest to implement. Finite field additions can be implemented as the Exclusive Or function, which is available on both processors as a one cycle instruction. The word sizes for both processors are larger than the 7 bit word size required to represent a finite field element and are thus sufficient to implement addition with the Exclusive Or instruction.

The remainder of this section will compare implementations of the syndrome computation algorithm and multiplication on the two processors.

Implementation of the Syndrome Computation Algorithm

Recalling the syndrome computation algorithm presented in Section B.3.1, the critical operations of this algorithm are executing a loop, testing successive bits of a 127 bit vector, accumulating a 28 bit syndrome word and looking up successive 28 bit masks to add to the accumulating syndrome word. The accumulation/addition operation equates to a 28 bit Exclusive Or. Testing successive bits of a 127 bit vector breaks down to successively loading a subvector of size equal to the word size of the processor into an accumulator and shifting out one bit at a time to test. A fast implementation requires that one accumulator be dedicated to this task throughout the syndrome computation. Another accumulator of word size greater than or equal to 28 bits is required to hold the accumulating 28 bit syndrome word.

Summarizing the above, the efficient implementation of the syndrome computation algorithm requires two accumulators: one of arbitrary length for shifting, and one of at least

28 bits in length for Exclusive Or operations. Neither processor provides a 28 bit Exclusive Or operation. Thus the 28 bit syndrome word must be broken up into two subwords which are computed separately using two separate submask tables. Breaking up the syndrome word into subwords increases the requirement for the number of separate accumulators to three if the complete syndrome is to be computed efficiently in a single loop. Further details and results of implementing syndrome computation on the two processors follows. Timings are normalized on a per message bit basis as opposed to a per received vector bit basis, since it is the rate of meaningful/non-redundant data transmission which is of primary concern.

DSP56000

The two 24 bit accumulators combined with parallel data moves allow the 28 bit syndrome word to be computed in a single loop almost as efficiently as if a 28 bit or greater word size was available on the DSP56000 for the Exclusive Or operation. Accumulator A is used to perform shifts/bit tests on a received subvector as well as accumulate 7 bits of the syndrome. This is done by swapping the subvector and 7 bits of syndrome in and out of Accumulator A with parallel data moves. Accumulator B is used to accumulate the remaining 21 bits of syndrome. The DSP56000 also provides very efficient implementation of the looping operation and fetching syndrome masks with the hardware DO loop and parallel data moves respectively.

The memory requirements for performing the syndrome computation with table lookup is for two 127 word tables. These can be set up in internal ROM. One table contains 7 bit submasks, while the other table contains 21 bit submasks.

The syndrome computation program on the DSP56000 takes a different amount of time to process a received 1 bit than a received 0 bit. In the case of the received vector being all 1's, syndrome computation is performed at a rate of 0.91 μs/message bit. In the case of an all 0's received vector, syndrome computation is performed at a rate of 0.54 μs/message bit. On average, if a received vector contains a near equal number of 0's and 1's, syndrome

computation is performed at a rate of 0.73 μs/message bit.

TMS320C25

The single accumulator on the TMS320C25 only provides 16 bit Exclusive Or operations. An efficient implementation requires two separate loops to compute the syndrome subwords. The upper 16 bits of the accumulator and a single word in memory are used to perform shifts/bit tests on a received subvector while the lower 16 bits of the accumulator is used to accumulate 14 bits of the syndrome.

The memory requirements for performing the syndrome computation with table lookup is for two 127-word tables. These can be set up in internal RAM. Both tables contain 14 bit submasks.

For any given received vector, the program on the TMS320C25 performs syndrome computation at a rate of 2.91 μs/message bit.

Implementation of Finite Field Multiplication

Finite field multiplication is best implemented using log and antilog tables as follows, (in a manner similar to integer arithmetic),

$$a \cdot b \;=\; antilog((log(a) + log(b)) \; mod \; 127).$$

This requires 3 table lookups, 1 addition and 1 modulus. The addition operation is just integer addition. The log and antilog are performed using table lookups. The modulus operation can be avoided by constructing a double length antilog table or using modulo addressing. A double length antilog table simply repeats the first 127 antilog table values in the second 127 table locations. The addition of $log(a)$ and $log(b)$ will never overflow the second section of the table.

A problem arises with multiplication by zero. Multiplication by zero gives a result of zero but $log(0)$ is undefined for a finite field as it is for integer arithmetic. The solution is to build an even longer antilog table, just over twice the size of that already required.

The added section forming this table is initialized to zero. If log(0) returns a value which just indexes the zeroed section of the antilog table, then multiplication by zero is realized.

An efficient implementation on the DSP56000 uses modulo addressing with a zero extended antilog table. The log table requires 128 words of memory and the antilog table requires 255 words of memory. The low 127 words of the antilog table can just be the 127-word syndrome submask table which contains the 7 bit submasks as opposed to the 21 bit submasks. This requires that the appropriate portion of the 28 bit syndrome masks be used for the 7 bit submasks. Recalling the definition of the syndrome masks M_i given in Section 3.2, the first elements of each mask vector should form the 7 bit submasks.

An efficient implementation on the TMS320C25 uses normal linear addressing with a double length zero extended antilog table. The log table requires 128 words of memory and the antilog table requires 506 words of memory. Both syndrome mask tables contain 14 bit masks and therefore cannot be designed to form the lower portion of the antilog table.

Multiplication can be implemented with comparable efficiency on the two processors, disregarding memory usage. When memory usage is considered, the TMS320C25 requires more internal data memory than is available. Syndrome computation requires 254 words of memory and multiplication requires 634 words. The total requirement, 888 words, exceeds the 544 words of internal data RAM available. Fast external RAM is required to optimize an implementation. The DSP56000 requires 254 words for syndrome computation and only an additional 256 words of memory for multiplication. The 512 words of internal ROM split between X and Y memory is sufficient to hold all of the tables.

B.4.3 Comparison Summary

Based primarily on the speed of computing syndromes and secondarily on the total memory requirements for table lookups used in computing syndromes and performing finite field multiplies, the DSP56000 is a better processor than the TMS320C25 for implementing a (127,99) BCH code decoder. Comparing syndrome computation speed alone, the DSP56000

is almost four times faster than the TMS320C25. Because of this, no work beyond implementing syndrome computation and multiplication has been done for the TMS320C25. A complete decoder has been implemented for the DSP56000. Performance results are given for this decoder in the next section.

B.5 Performance Results for Decoder Implementation on Motorola DSP56000

A complete (127,99) BCH error control code decoder has been implemented on the Motorola DSP56000 using the algorithms presented in Section B.3. The decoder performance is measured in message bit rate (bits per second or bps) as opposed to raw received bit rate since received data contains redundant parity bits. In other words, the message bit rate is the data rate achievable at the decoder output, not input. The average performance statistics are:

- 1.4 Mbps when no errors detected/corrected,
- 1.1 Mbps when correcting 1 or 2 errors,
- 250 Kbps - 1 Mbps when correcting 3 or 4 errors.

The range of rates at which 3 or 4 errors are corrected is due to the exhaustive search algorithm used to locate the first 1 or 2 errors before performing table lookup of the remaining 2 error locations. The worst case is when all of the erroneous bits are in the parity bits of the received vector. All 99 message bits are tested to see if any one is in error. This is the worst case decoding rate of 250 Kbps. The probability of occurrence of errors causing this worst case decoding rate is likely quite low. Otherwise, a more powerful decoder would be used.

B.6 Conclusions and Recommendations

The Motorola DSP56000 is a better processor than the Texas Instruments TMS320C25 for the purpose of implementing an error correcting decoder. The worst case message bit rate for a complete four error correcting (127,99) BCH code decoder on the DSP56000 is 250 Kbps, only slightly slower than the rate of 340 Kbps at which the TMS320C25 could implement just the first of the three steps in the decoding process. Three key differences between the processors stand out as the main reasons why the DSP56000 can be used to implement a significantly faster decoder than one implemented on the TMS320C25.

The first key difference lies in the overall architecture of the two processors. Although the DSP56000 is specifically designed for digital signal processing, its architecture is still quite general purpose and not restricted to only being useful for implementing DSP or similar applications. The architecture of the TMS320C25 is much more specialized. In optimizing certain features of the architecture for important DSP applications, generality has been sacrificed. In particular, the specific architecture of this central arithmetic logic unit is detrimental to implementing an efficient decoder.

The second key difference between the two processors is the hardware DO loop available on the DSP56000. This allows an efficient implementation of the key program loops of the decoder with no instruction overhead for controlling loops. The TMS320C25 does not have an equivalent feature for efficient looping through a block of program code.

The third key difference between the two processors is the parallel data move operations available on the DSP56000. The ability to move data around in parallel with arithmetic operations enhances the performance of the decoder implemented on the DSP56000. The TMS320C25 does implement a number of instructions which move data in parallel with an operation in the arithmetic logic unit, but the instructions are very application specific.

The DSP56000 could be used to implement a (127,99) BCH decoder with a throughput in excess of 1 Mbps. The implementation described in this Appendix could easily be

extended to include data buffering and make use of the on board serial ports. The primary external logic required for the decoder is a 16K $\times$ 16 bit ROM to store the one/two error lookup table.

Considering that the performance of the DSP56000 as a decoder is hampered by its 24 bit word size, it would be worthwhile to investigate implementing the same decoder on a general purpose 32 bit microprocessor. Such a processor in conjunction with zero wait state external RAM would likely yield a faster decoder than that implemented on the DSP56000.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section B.)

University of Victoria, Department of Electrical and Computer Engineering, P.O. Box 1700, Victoria, BC V8W 2Y2

2. SECURITY CLASSIFICATION (overall security classification of the document. including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

Coding for Frequency Hopped Spread Spectrum Satellite Communications (U)

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Bhargava, Vijay K.; Blake, Ian F.; Gulliver, T. Aaron; Li, Gang; Wang, Qiang and Weeks, Brent.

5. DATE OF PUBLICATION (month and year of publication of document)

April 1989

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

104

6b. NO. OF REFS (total cited in document)

38

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Annual Technical Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Communications Research Centre
P.O. Box 11490, Station "H"
Ottawa, ON, K2H 8S2

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

D 6470

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

36001-8-3529/01-SS

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

Technical Report No. ECE 89-1

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availabilty (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

A modified self-normalizing combiner is presented and its performance is analyzed under partial-band jamming. This is compared with other non-linear combining schemes.

Continuing on our previous work, the throughput performance of a coded Fast Frequency Hopped M-ary Frequency Shift Keying (FFH/MFSK) system with a fixed hop rate is evaluated using the cutoff rate argument. The analysis upperbounds the gains which can be realized using coding for various system parameters.

As a prelude to the study of coding for Slow Frequency Hopped Differential Phase Shift Keying (SFH/DPSK), we derive the probability distribution of DPSK in tone interference.

A comparison of two powerful DSP integrated circuit processors for implementation of a decoder for the (127,99) BCH code is presented as an Appendix.

Suggestions for future work include investigation of coding for SFH/DPSK; error control coding to alleviate very high error rate situations; implementation aspect of CODECS and a theoretical investigation of communications over an intentional interference channel.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible, keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

-Frequency hopping
-Spread spectrum
-Error-correcting codes
-Satellite communications
-M-ary FSK modulation
-DPSK modulation
-Self-normalizing combiner