JOINT CODING AND MODULATION DESIGNS FOR BANDLIMITED SATELLITE CHANNELS

by

Joseph Y. N. Hui

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREES OF
BACHELOR OF SCIENCE
and
MASTER OF SCIENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June, 1981

© Joseph Y. N. Hui

The author hereby grants to MIT permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author ....
Department of Electrical Engineering and Computer Science

Certified by ....
Thesis Supervisor (Academic)

Certified by ....
Company Supervisor (VI-A Cooperating Company)

Accepted by ....
Chairman, Departmental Committee on Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARCHIVES
JUL 30 1981
LIBRARIES

JOINT CODING AND MODULATION DESIGNS FOR
BANDLIMITED SATELLITE CHANNELS

by

Joseph Y. N. Hui

Submitted to the Department of Electrical Engineering and Computer Science on May 5, 1981, in partial fulfillment of the requirements for the degrees of Bachelor of Science and Master of Science.

Abstract

This thesis addresses the problem of severe bandwidth and power limitation of future satellite systems by a joint consideration of coding and modulation. Bandwidth economy is achieved by two methods. First, baseband pulses with rapid spectral roll-off and less than unity bandwidth-to-symbol-rate ratio are obtained by allowing controlled intersymbol interference (ISI). Second, phase shift keying (PSK) having more phases is used.

Convolutional encoding with maximum likelihood decoding is used to tradeoff some bandwidth economy acquired to overcome the power limitation. Lower bounds, tight in most cases, are derived for the minimum free Euclidean distance of the modulator output signals, after coding and inclusive of ISI. These bounds are used for searching good encoders.

Various encoder structures and modulation schemes are proposed. In combining a rate 2/3 encoder of six binary memories with 8ϕ-PSK modulation having controlled ISI, an E_b/N_o gain of 4-5 dB over uncoded QPSK is achieved simultaneously with marked spectral improvement.

System performances are evaluated theoretically and confirmed by simulation.

THESIS SUPERVISOR: Pierre A. Humblet
TITLE: Assistant Professor of Electrical Engineering

CO-OP COMPANY SUPERVISOR: Russell J. F. Fang
TITLE: Manager, Communication System Analysis, COMSAT Labs

ACKNOWLEDGMENTS

I would like to give my sincere thanks to Professor Pierre A. Humblet who, despite distance, contributed much more attention to my research than is expected for a Co-op thesis supervisor. His insistence on clarity of presentation improved both my communication skills and the lucidity of this thesis.

Special thanks also go to Dr. Russell Fang, who first introduced me to the subject and provided helpful supervision along the way. His keen sense of industrial application adds a new dimension to my academic pursuit, making the Co-op experience so worthwhile. I would also like to thank COMSAT Labs for providing this opportunity.

I am also indebted to Dr. Lin Nan Lee and Mr. Smith Rhodes, both of whom helped me to understand modulation and coding; Ms. Linda Mikisits for preparing some of the graphs; and Mr. and Mrs. Mark Eng for their hospitality during this period, thus providing the more humane part of my education.

Table of Contents

Page No.

Page No.

List of Figures

List of Figures (Continued)

List of Figures (Continued)

Chapter 1. POWER AND BANDWIDTH LIMITATIONS OF CURRENT SATELLITE COMMUNICATIONS SYSTEMS

In recent years, lowered cost and more diversified applications of satellite communications systems have dramatically increased the demand for communication traffic via satellite. Consequently, the existing spectral allocation for satellite services in the 6/4-GHz band has become extremely congested. This spectral congestion problem may be alleviated by designing satellites which would reuse the same frequency band for a multiple number of times or would operate in higher frequency bands such as 14/11 GHz or 30/20 GHz. Multiple frequency reuses can be realized by employing carrier waves with orthogonal sense of polarization or by employing a multiple-beam satellite antenna design. The imperfect isolation between satellite antenna beams, as well as nonideal polarization isolation, causes co-channel interference (CCI) that can be one of the major impairments in a satellite system reusing the same frequency spectrum for a multiple number of times. The co-channel interference can be especially severe if the reuse is achieved by orthogonal polarization, since rain can cause significant depolarization on the carrier waves.

Another source of interference is called adjacent channel interference (ACI), which is due to imperfect transponder frequency isolation.

Generally, the carrier-to-interference (CCI + ACI) ratio (C/I) cannot be improved by merely increasing the carrier power since the interference caused by nonideal isolation would increase as well. Therefore, the overall available carrier-to-thermal noise and interference power ratio [C/(N + I)] of a multiple-beam satellite system can be limited because of the presence of interference.

As a result, allowable power as well as bandwidth will be limited in future multibeam satellite systems such as the INTELSAT VI. The power limitation for avoiding interference can be particularly severe at 30/20 GHz where signals are more vulnerable to fades due to rain.

The Satellite Business Systems (SBS) is another example where there exists power and bandwidth limitations. Low equipment cost for High Power Amplifier (HPA) and antenna on the ground is crucial for the successful development of such systems. These cost considerations would limit the power available for transmission. The SBS system operates in the 14/11 GHz band, which is comparatively less congested than the 6/4 GHz band. However, bandwidth limitation is anticipated even in this new band when the number of users for such systems increases.

The purpose of this thesis is to address the transmission system design problem for satellite channels that are both bandwidth and power limited. Specifically, a joint forward error correction (FEC) coding and modulation system design approach is proposed.

The transmission system design problem becomes quite complex when real system constraints are included. The impairments caused by thermal noise, intersymbol interference (ISI), CCI, ACI, and channel nonlinearities must all be considered. Frequently, many of such impairments are mutually coupled. For example, in the Time Division Multiple Access (TDMA) system, the modulated signal must be filtered to minimize the interference into adjacent channel. Due to power and cost considerations, it is desirable to operate the HPA near saturation at the earth station. However, the spectral sidelobes of the filtered TDMA signal would regrow and spread after it is amplified by the nonlinear HPA. This spectral sidelobe regrowth can cause undesirable out-of-band emission (OBE) noise and adjacent channel interference. It can also cause additional intersymbol interference when the signal is further filtered by the satellite transponder filters and the earth station receive filters. Thus, the operating point of the earth station HPA is frequently backed off from its saturation power level in order to minimize the impairments caused by ISI and ACI or to limit the OBE noise. At the satellite transponder, the nonlinear Traveling Wave Tube Amplifier (TWTA) would further degrade the system performance and it often must also be backed off from its saturation power level in order to minimize ISI, ACI and other nonlinearity effects such as phase noise due to AM/PM conversion. The more robust approach of joint coding and modulation system design suggested in this thesis will hopefully combat such impairments present in a realistic power and bandwidth limited environment.

This thesis will consist of 8 chapters. Chapter 1 has given a description of the bandwidth and power limited satellite channels. Chapter 2 reviews previous work performed on the subject of coding and modulation system design for satellite channels. Chapter 3 models the transmitter and receiver structures and derives the likelihood ratio of signals in the presence of intersymbol interference. Chapter 4 deals with the problem of filter design which minimizes out-of-band emission. Chapter 5 presents several classes of encoders suitable for coded 8ϕ-PSK modulation in the absence of ISI. Chapter 6 discusses the subject of convolution encoder designs in the presence of intersymbol interference. Chapter 7 evaluates the overall system performance in theory as well as by system simulation to find the coding gain with respect to uncoded 4ϕ-PSK and bandwidth required for given bit error rate. The last chapter concludes the thesis and provides suggestions for further research.

CHAPTER 2. REVIEW OF PREVIOUS WORK

Most of the previous works dealing with bandwidth limitation or power limitation fail to address both subjects simultaneously. In this review we are going to cite merits of some approaches and reasons for rejecting others. After some reorganizing of the preferred approaches, we hope to form a unified framework for the course of research of this thesis.

We start by looking at some bandwidth efficient modulation techniques, then power efficient coding techniques and afterwards maximum likelihood detection which is often applicable for these bandwidth or power efficient techniques. Finally, we shall state our approach.

To speak about bandwidth efficiency, one must define what bandwidth is, which unfortunately does not have a universally satisfying definition. Recently, Amoroso [1] has given a rather comprehensive summary of various definitions for spectral bandwidth of a signal. Different pulse shape would be obtained in minimizing bandwidth under different definitions of bandwidth. One such example is described in the classical paper by Landau and Pollack [2] in which a prolate spheroidal function minimizes the width of the frequency band containing a specified fraction of the signal energy.

Lacking a universal criterion for bandwidth economy, modulation schemes are judged very often by inspecting their power spectra. Nevertheless, several modulations that are claimed to be bandwidth efficient have evolved in recent years.

Minimum shift keying (MSK) [3] is one such scheme using half-cosine as baseband pulse with the two quadrature channel pulse trains offset by half a pulse repetition interval. MSK can be viewed as a special case of continuous phase M-ary frequency shift keying (MFSK) with M = 2. Due to its constancy of envelope, MSK is more compatible with nonlinear satellite transmission mode. However, it can hardly be claimed as bandwidth efficient since the null-to-null bandwidth is 50 percent more than that of 4ϕ-PSK with rectangular pulse shape. In a recent paper [4], Rhodes proposed another constant envelope modulation called the frequency shift offset quadrature (FSOQ) modulation which is basically continuous phase 3FSK, with improved spectral property over MSK at the cost of slightly increased transmission complexity. Several other schemes use overlapped baseband pulses, such as overlapped raised cosine [6] and truncated sinc functions [7], which achieve spectral efficiency at the expense of slight envelope fluctuation and presence of intersymbol interference. The merit of all these schemes is that they can be treated as quadrature pulse amplitude modulation (QPAM) which is convenient for analyses and high speed implementation. However, these schemes do not improve power economy over QPSK using rectangular pulse-shaping.

One power efficient coding technique is described by Ungerboeck [10]. Redundancy is introduced by using a rate 2/3 convolutional encoder which takes in 2 bits, and maps its 3-bit output into the eight phases of 8ϕ-PSK. Ungerboeck was able to achieve 3-6 dB gain over uncoded 4ϕ-PSK, with the same information rate and spectral efficiency. It seems very appealing if the power efficiency of Ungerboeck's rate 2/3 coded 8-ϕ can be combined with the spectral efficiency of some of the QPAM schemes mentioned previously.

We shall prefer this combined coding and modulation over another well-known class of hybrid coding and modulation schemes, called correlated phase shift keying (CORPSK) or alternatively called trellis phase code, summarized in the paper by Muilwijk [8]. Typically, these schemes convolutionally encode the input bit sequence into multilevel phase positions, which are interpolated to generate a smooth phase function for phase modulation. For example, tamed frequency modulation (TFM) [5] achieves reduced bandwidth through convolutionally encoding three input bits into 8-phase positions. Anderson and Taylor [9] have shown that trellis phase codes can achieve substantial power improvement (2 to 4 dB) over 4ϕ-PSK, with reduced bandwidth at the same time. It is possible that the gain of CORPSK can be achieved by less complex coded QPAM schemes. In fact, we expect our combined coding and modulation approach to achieve better spectral and power efficiency than CORPSK.

A proper understanding of maximum likelihood (ML) decoding using the Viterbi algorithm is indispensable for optimal detection in case overlapped baseband pulse, or coded M-ary PSK, or trellis phase code is employed. ML detection for convolutional codes in the presence of non-stochastic channel impairments (such as bandlimiting, distortion by nonlinear elements or cross channel coupling) is a fairly well studied topic. ML estimation for band-limited linear channel has been investigated by Forney [11] and Ungerboeck [12]. A summary of these results is presented in [13]. Mesiya et al [14] treated the ML detection problem for the non-linear and bandlimited channel using a bank of matched filters. Hermann [15] subsequently evaluated numerically the degradation of the free Euclidean distance for uncoded signals transmitted in a bandlimited nonlinear channel and found the performance loss relative to the linear channel to be small. Hence there exists a substantial potential for receiver improvement by ML estimation relative to bit-by-bit detection. However, the question of pulse design is not addressed in these papers. Furthermore, the problem of code design with good free Euclidean distance is left out.

The framework formulated is as follows. First, we shall attempt to define bandwidth efficiency and obtain optimal pulse shapes according to these definitions. These pulses will be overlapped at baseband and coded M-ary PSK will be employed. Search algorithm for optimal encoder for M-ary PSK channel with controlled ISI will be investigated. ML detection is used for decoding.

We shall not treat the nonlinear satellite channel in our analysis, thus limiting ourselves to linear additive white Gaussian noise channel. The research of this thesis started with the practical problem stated in Chapter 1. Research direction is framed in this chapter and we shall proceed to offer a unified solution in later chapters.

CHAPTER 3. TRANSMITTER AND RECEIVER STRUCTURES FOR BASEBAND PULSES WITH CONTROLLED INTERSYMBOL INTERFERENCE

The bulk of the literature on channel coding is mostly concerned with reliable transmission over a binary symmetric channel. The encoding stage would take in a sequence of binary digits and deliver a binary sequence which is relatively immune to error occurrences. Unfortunately, such encoding techniques for binary symmetric channels are inadequate for most modulation schemes which use multilevel/phase signals. Until now, results for channel coding employing multilevel/phase signals are relatively few [10]. The problem of optimal code design for modulation schemes with controlled intersymbol interference (ISI) is scarcely addressed. However, growing demand for communications bandwidth in recent years has stimulated interest in modulation schemes employing expanded signal set and baseband pulses with controlled ISI. These novel modulation schemes necessitate the consideration of coding and modulation as an entity. For channels corrupted by Additive White Gaussian Noise (AWGN), we want to maximize the minimum free Euclidean distance amongst the coded channel waveforms. This chapter presents a unified mathematical model for M-ary PSK modulation technique with channel encoding for the AWGN channel. The modelling attempts to abstract the complicated satellite channel in a way that is mathematically tractable. The results obtained from this modeling will be tested by simulation in Chapter 7.

3.1 Transmitter Structure

The data source (Figure 3.1) puts out a sequence $\underline{u}$ consisting of binary u_k's which are statistically independent random variables with equal probability of being 0 or 1. The data stream is fed into a convolutional encoder to increase the message redundancy before transmission. The encoder output is mapped into a sequence $\underline{v}$ consisting of v_k's which are elements of an M-ary PSK channel symbol set. In the example shown in Figure 3.2 for 8ϕ-PSK, $v_k \varepsilon \{0, 1, 2, \ldots, 7\}$. For practical application, M is rarely greater than 8 when phase and timing jitter would then become a major limitation of system performance.

Throughout this thesis, convolutional encoders are employed due to their generally superior error correction capability compared to block coding as well as relative ease of decoding and code searching by the Viterbi algorithm. The problem of finding optimal encoder with multiphase signals will be explored in Chapter 4.

The low pass filter with impulse response $\sqrt{2E_s'}\, h'(t)$ (E_s' being the modulated signal power per symbol) provides the inphase and quadrature envelope functions $s_r(t)$ and $s_i(t)$ given by

$$s_r(t) = \sum_{k=-N}^{N} \sqrt{2E_s'}\, h'(t - kT) \cos \frac{2\pi v_k}{M}$$

$$s_i(t) = \sum_{k=-N}^{N} \sqrt{2E_s'}\, h'(t - kT) \sin \frac{2\pi v_k}{M}$$

The envelope functions then modulate the carriers $\cos 2\pi f_c t$ and $\sin 2\pi f_c t$ which are afterwards added to give the modulated signal

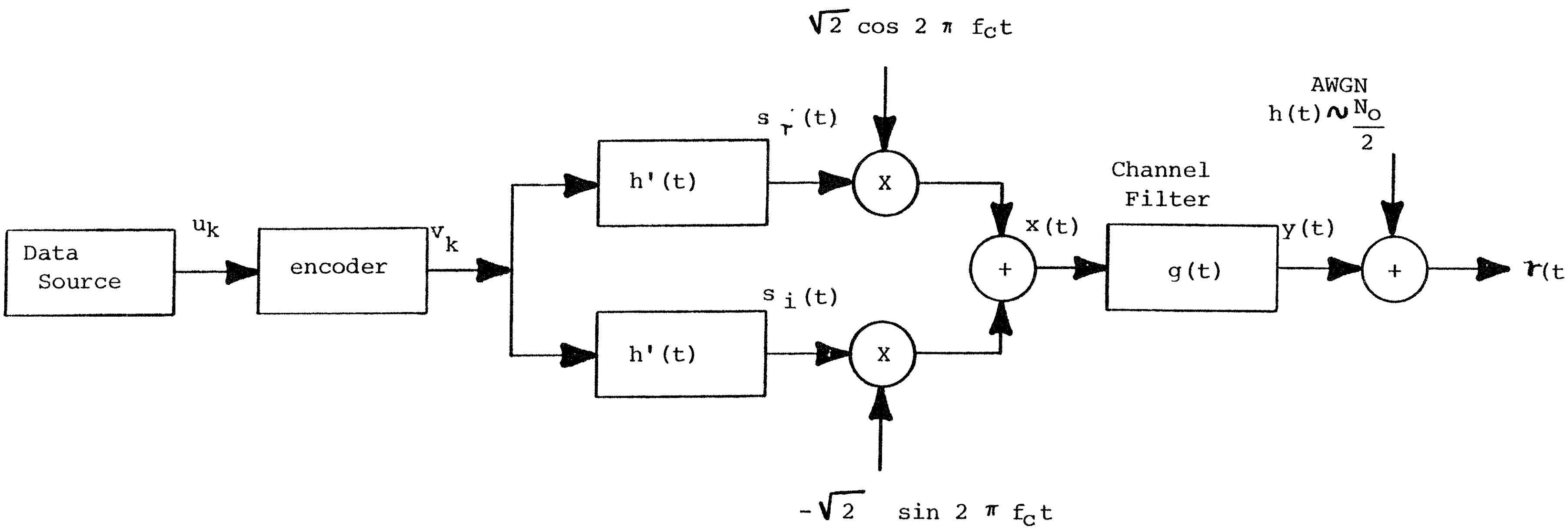


Figure 3.1. Transmitter Structure

$$x(t) = \sum_{k=-N}^{N} \sqrt{2E_s'}\ h'(t - kT) \left\{ \cos \frac{2\pi v_k}{M} \cos 2\pi f_c t - \sin \frac{2\pi v_k}{M} \sin 2\pi f_c t \right\}$$

$$= \sum_{k=-N}^{N} \sqrt{2E_s'}\ h'(t - kT) \quad \cos \left(2\pi f_c t + \frac{2\pi v_k}{M} \right)$$

$$= \sum_{k=-N}^{N} \frac{1}{2} \sqrt{2E_s'}\ h'(t - kT) \left\{ e^{j2\pi f_c t + j2\pi v_k/M} + (\cdot)^* \right\}$$

where $(\cdot)^*$ denotes the conjugate of the term before it. Defining

$$s(t) = \sum_{k=-N}^{N} \sqrt{2E_s'}\ h'(t - kT)\ e^{j2\pi v_k/M}$$

The modulated signal can be expressed as

$$x(t) = \frac{1}{2}\ s(t)\ e^{j2\pi f_c t} + \frac{1}{2}\ s^*(t)\ e^{-j2\pi f_c t}$$

and applying Fourier transformation gives

$$X(f) = \frac{1}{2}\ S(f - f_c) + \frac{1}{2}\ S^*(-f - f_c)$$

as shown in Figure 3.3.

Often times, the modulated signal is subsequently filtered to minimize adjacent channel interference. Therefore, the transmission channel is bandlimited and the band-pass filter with impulse response g(t) (Figure 3.1) is added to model the channel. Assume the Fourier transform of g(t) given by

$$G(f) = \int_{-\infty}^{\infty} g(t)\ e^{-j2\pi ft}\ dt$$

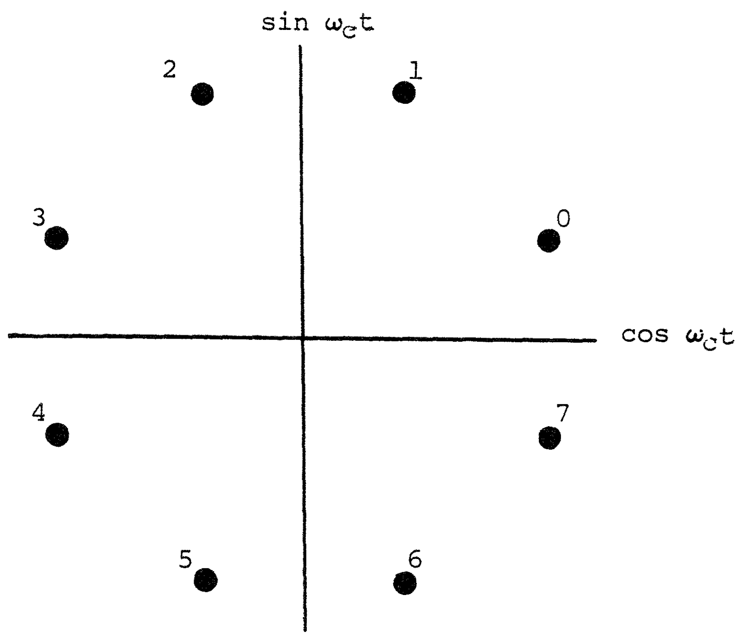


Figure 3.2. Channel Symbol Set for 8ϕ-PSK

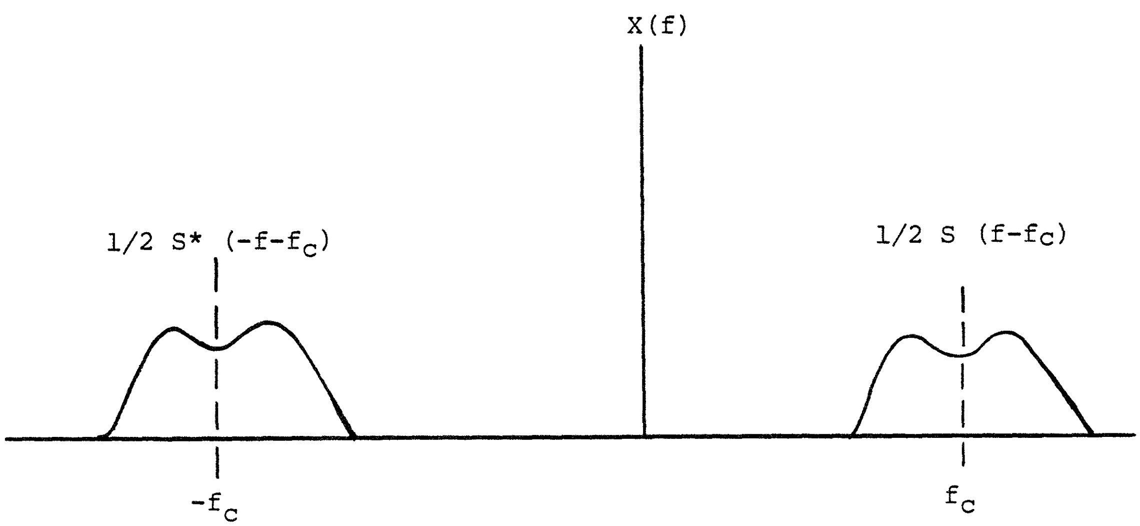


Figure 3.3. Fourier Transform of x(t)

to have a spectral shape as shown in Figure 3.4, and define $G_o(f)$ such that

$$G(f) = G_o(f - f_c) + G_o^*(-f - f_c) \text{ , and let}$$

$$g_o(t) = \int_{-\infty}^{\infty} G_o(f)\, e^{j2\pi ft}\, df$$

so that

$$g(t) = g_o(t)\, e^{j2\pi f_c t} + g_o^*(t)\, e^{-j2\pi f_c t}$$

$$= 2\, \mathrm{Re}\left[g_o(t)\, e^{j2\pi f_c t}\right]$$

Finally, the spectrum of the channel filtered signal and its inverse Fourier transform, assuming s(t) is low pass ($<f_c$) are respectively

$$Y(f) = \frac{1}{2}\, S(f - f_c)\, G_o(f - f_c) + \frac{1}{2}\, S^*(f - f_c)\, G_o^*(-f - f_c)$$

$$y(t) = \frac{1}{2}\left\{s(t) * g_o(t)\, e^{j2\pi f_c t} + (\cdot)^*\right\}$$

$$= \mathrm{Re}\left[s(t) * g_o(t)\, e^{j2\pi f_c t}\right]$$

in which

$$s(t)*g_o(t) = \sum_{k=-N}^{N} \sqrt{2E_s'}\, h'(t - kT)\, e^{j2\pi v_k/M} * g_o(t)$$

$$= \sum_{k=-N}^{N} [\sqrt{2E_s'}\, h'(t - kT) * g_o(t)]\, e^{j2\pi v_k/M}$$

Defining $\sqrt{2E_s}\, h(t - kT) = \sqrt{2E_s'}\, h'(t - kT) * g_o(t)$ gives

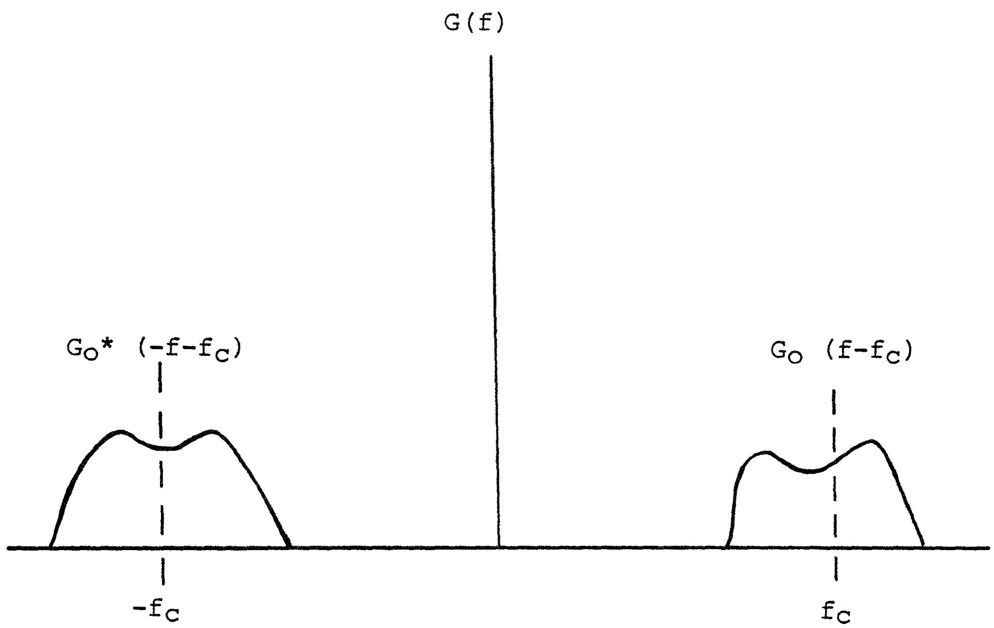


Figure 3.4. The Channel Filter

$$y(t) = \sum_{k=-N}^{N} \mathrm{Re}\left[\sqrt{2E_s}\ h(t - kT)\ e^{j2\pi f_c t + j2\pi v_k/M}\right]$$

$$= \sum_{k=-N}^{N} \frac{1}{2}\left\{\sqrt{2E_s}\ h(t - kT)\ e^{j2\pi f_c t + j2\pi v_k/M} + (\cdot)^*\right\}$$

Therefore, the two filters in the model can be combined by replacing $\sqrt{2E_s'}\ h'(t)$ with $\sqrt{2E_s}\ h(t)$ and omitting the channel filter with impulse response g(t) in Figure 3.1. As $g_o(t)$ can be complex in general, so can h(t). In the following discussion, we shall assume

$$\int_{-\infty}^{\infty} h(t)\ h^*(t) = 1$$

3.2 Metric Derivation

The signal y(t) is then corrupted by a zero mean white Gaussian noise n(t) with

$$E[n(t)\ n'(t)] = \frac{N_o}{2}\ \delta(t - t')$$

The received waveform is given by

$$r(t) = y(t) + n(t)$$

Note in particular that y(t) is real and consequently r(t) is also real.

A maximum likelihood receiver chooses the source sequence $\underline{u}$ that would most likely result in the waveform r(t) after noise corruption. For AWGN corruption, it is well known [17] that the logarithm of this likelihood is proportional to the negative value of the Euclidean distance between the received r(t) and the uncorrupted $y_u(t)$ generated by the sequence $\underline{u}$. Consequently, the metric for measuring distances amongst waveforms will be Euclidean, with Euclidean distance defined by

$$\left\| r(t) - y_u(t) \right\|^2 = \int_{-\infty}^{\infty} [r(t) - y_u(t)][r(t) - y_u(t)]^* \, dt$$

Therefore, the decision rule which minimizes the error probability, based on the entire received signal, is to choose $\underline{u}$ iff

$$\ln \frac{P[r(t) | y_u(t)]}{P[r(t) | y_{u'}(t)]} \geq 0 \text{ for all } u' \neq u$$

and for AWGN corruption, this likelihood ratio is equal to

$$-\frac{\left\| r(t) - y_u(t) \right\|^2 - \left\| r(t) - y_u'(t) \right\|^2}{N_o}$$

$$= \frac{2}{N_o} \int_{-\infty}^{\infty} [y_u(t) - y_{u'}(t)] \, r(t) \, dt$$

$$- \frac{1}{N_o} \int_{-\infty}^{\infty} [y_u^2(t) - y_{u'}^2(t)] \, dt$$

Hence, each input sequence $\underline{u}$ is associated with a λ_u for each received waveform $r(t)$ such that

$$\lambda_u = \frac{2}{N_o} \int_{-\infty}^{\infty} y_u(t) \, r(t) \, dt - \frac{1}{N_o} \int_{-\infty}^{\infty} y_u^2(t) \, dt$$

and the decision rule is to choose the u with the largest λ_u. For simplicity, the following derivations will drop the subscript u.

The remaining of this section is devoted to expressing λ in terms of a complex discrete sequence of sufficient statistics r. As we shall see later, the sufficient statistics r_k are generated by the demodulator as samples of the matched filter output.

The first term in the expression for λ is

$$\int_{-\infty}^{\infty} y(t)\, r(t)\, dt$$

$$= \int_{-\infty}^{\infty} \sum_{k=-N}^{N} \frac{1}{2} \left\{ \sqrt{2E_s}\, h(t - kT)\, e^{j2\pi f_c t + j2\pi v_k/M}\, r(t) \right.$$

$$\left. + (\cdot)^* \right\} dt$$

$$= \sum_{k=-N}^{N} \frac{1}{2} \sqrt{E_s} \left\{ e^{j2\pi v_k/M}\, r_k + (\cdot)^* \right\}$$

$$= \sum_{k=-N}^{N} \sqrt{E_s}\, \mathrm{Re}\left[e^{j2\pi v_k/M}\, r_k \right]$$

in which

$$r_k = \int_{-\infty}^{\infty} \sqrt{2}\, h(t - kT)\, e^{j2\pi f_c t}\, r(t)\, dt$$

is a complex number. The second term for λ is given by

$$\int_{-\infty}^{\infty} y^2(t)\, dt$$

$$= \int_{-\infty}^{\infty} \frac{1}{2} E_s \left\{ \sum_{k=-N}^{N} h(t - kT)\, e^{j2\pi f_c t + j2\pi v_k/M} + (\cdot)^* \right\}^2 dt$$

$$= \int_{-\infty}^{\infty} \frac{1}{2} E_s \sum_{k=-N}^{N} \sum_{\ell=-N}^{N} \Big\{ h(t - kT)\, h(t - \ell T)\, e^{j4\pi f_c t + j(v_k + v_\ell)2\pi/M}$$

$$+ (\cdot)^* \Big\} dt$$

$$+ \int_{-\infty}^{\infty} \frac{1}{2} E_s \sum_{k=-N}^{N} \sum_{\ell=-N}^{N} \Big\{ h(t - kT)\, h^*(t - \ell T)\, e^{j(v_k - v_\ell)2\pi/M}$$

$$+ (\cdot)^* \Big\} dt$$

We assumed previously s(t) to be low pass, or roughly speaking h(t) is slow-varying with respect to the carrier frequency. The first integral in the above expression can be shown to equal zero. Furthermore, define

$$h_{k-\ell} = \int_{-\infty}^{\infty} h(t - kT)\, h^*(t - \ell T)\, dt$$

$$= \int_{-\infty}^{\infty} \frac{1}{2\pi} \left| H(\omega) \right|^2 e^{-j\omega T(k-\ell)}\, d\omega$$

$$= h_{\ell-k}^{\;*}$$

Consequently, the second term for λ becomes

$$\frac{1}{2} E_s \sum_{k=-N}^{N} \sum_{\ell=-N}^{N} \Big\{ e^{j(v_k - v_\ell)2\pi/M}\, h_{k-\ell} + (\cdot)^* \Big\}$$

$$= E_s \sum_{k=-N}^{N} \sum_{\ell=-N}^{N} \mathrm{Re} \left[e^{j(v_k - v_\ell)2\pi/M}\, h_{k-\ell} \right]$$

The overall expression for λ is then given by

$$\lambda = 2\frac{\sqrt{E_s}}{N_o} \sum_{k=-N}^{N} \mathrm{Re}\left[e^{j2\pi v_k/M} r_k\right]$$

$$- \frac{\sqrt{E_s}}{N_o} \sum_{k=-N}^{N} \sum_{\ell=-N}^{N} \sqrt{E_s}\, \mathrm{Re}\left[e^{j(v_k - v_\ell)2\pi/M} h_{k-\ell}\right]$$

The second term in the above expression is a symmetrical quadratic form. The terms that are symmetrical to the diagonal are equal since

$$\mathrm{Re}\left[e^{j(v_\ell - v_k)2\pi/M} \cdot h_{\ell-k}\right]$$

$$= \mathrm{Re}\left[e^{j(v_\ell - v_k)2\pi/M} \cdot h_{\ell-k}\right]^*$$

$$= \mathrm{Re}\left[e^{j(v_k - v_\ell)2\pi/M} \cdot h_{\ell-k}^*\right]$$

$$= \mathrm{Re}\left[e^{j(v_k - v_\ell)2\pi/M} \cdot h_{k-\ell}\right]$$

Therefore, the double summation for λ can be separated into two terms which are twice the upper triangular quadratic form and the sum of the diagonal terms.

$$\lambda = 2\frac{\sqrt{E_s}}{N_o} \sum_{k=-N}^{N} \mathrm{Re}\left[e^{j2\pi v_k/M} r_k\right] - \frac{\sqrt{E_s}}{N_o} \sum_{k=-N}^{N} \sqrt{E_s}\, h_o$$

$$- \frac{\sqrt{E_s}}{N_o} \sum_{k=-N}^{N} 2\sqrt{E_s}\, \mathrm{Re}\left[e^{jv_k\pi/M} \sum_{\ell=1}^{k+N} e^{-jv_{k-\ell}\pi/M} h_\ell\right]$$

For sufficiently large ℓ, the intersymbol interference coefficient h_ℓ should be small. If we limit the intersymbol interference effect to s symbol, (i.e., $h_i = 0$ for $|i| \geqq s$), then

$$\lambda = 2\frac{\sqrt{E_s}}{N_o}\sum_{k=-N}^{N} \mathrm{Re}\left[e^{j2\pi v_k/M}\, r_k\right] - \frac{\sqrt{E_s}}{N_o}\sum_{k=-N}^{N}\sqrt{E_s}\, h_o$$
$$-\frac{\sqrt{E_s}}{N_o}\sum_{k=-N}^{N} 2\sqrt{E_s}\ \mathrm{Re}\ e^{j2\pi v_k/M}\sum_{\ell=1}^{s-1} e^{-jv_{k-\ell}\pi/M}\, h_\ell$$

Defining

$$\lambda_k = 2\mathrm{Re}\left[e^{j2\pi v_k/M}\, r_k\right] - \sqrt{E_s}\, h_o$$
$$-\ 2\sqrt{E_s}\ \mathrm{Re}\left[e^{j2\pi v_k/M}\cdot\sum_{\ell=1}^{s-1} e^{-j2\pi v_{k-\ell}/M}\quad h_\ell\right]$$

so that

$$\lambda = \frac{\sqrt{E_s}}{N_o}\sum_{k=-N}^{N}\lambda_k$$

The maximization of λ over all possible $\underline{u}$ is a dynamic programming problem which can be efficiently performed by the Viterbi algorithm [13]. The optimal receiver structure for this mathematical model follows immediately from the expression for λ.

3.3 Receiver Structure

As we have just seen, the sufficient statistics for detection is given by

$$r_k = \int_{-\infty}^{\infty} \sqrt{2}\, h(t - kT)\, e^{j2\pi f_c t}\, r(t)\, dt$$

which can be generated by a correlation receiver as shown in Figure 3.5. The received signal is multiplied by the in-phase and quadrature carriers before it is passed into filters matched to the transmit filter with impulse response h(t). The carriers can be recovered by conventional phase lock loop techniques. The filter output is sampled at appropriate instants with adequate timing synchronization. With the r_k's at hand, the most likely $\underline{u}$ can be estimated by trellis search using the Viterbi Algorithm. Consider the trellis diagram for a certain binary convolutional encoder of rate p/q and having γ binary memories in the absence of ISI as shown in Figure 3.6. The encoder shown has 2^γ states. Merging into every state are 2^p branches each associated with a channel symbol which is an element of the channel symbol set. In the presence of ISI, the channel "remembers" the past s-1 symbols. Consider adding (s - 1) shift registers to each queue of the encoder (or p.(s - 1) shift registers added altogether) as shown in Figure 3.7. Obviously, knowing the content of the shift registers for this convolutional encoder with extended memory is sufficient for calculating the present as well as the past (s - 1) channel symbols. The state of the system (encoder plus channel) is sufficiently represented by the content of the shift registers for the encoder with extended memory. The trellis diagram in the presence of ISI, therefore, consists of $2^{\gamma+p.(s-1)}$ states.

The probability of error P_e depends on the Euclidean distances between codeword waveforms. Asymptotically, P_e is determined by the minimum free Euclidean distance.

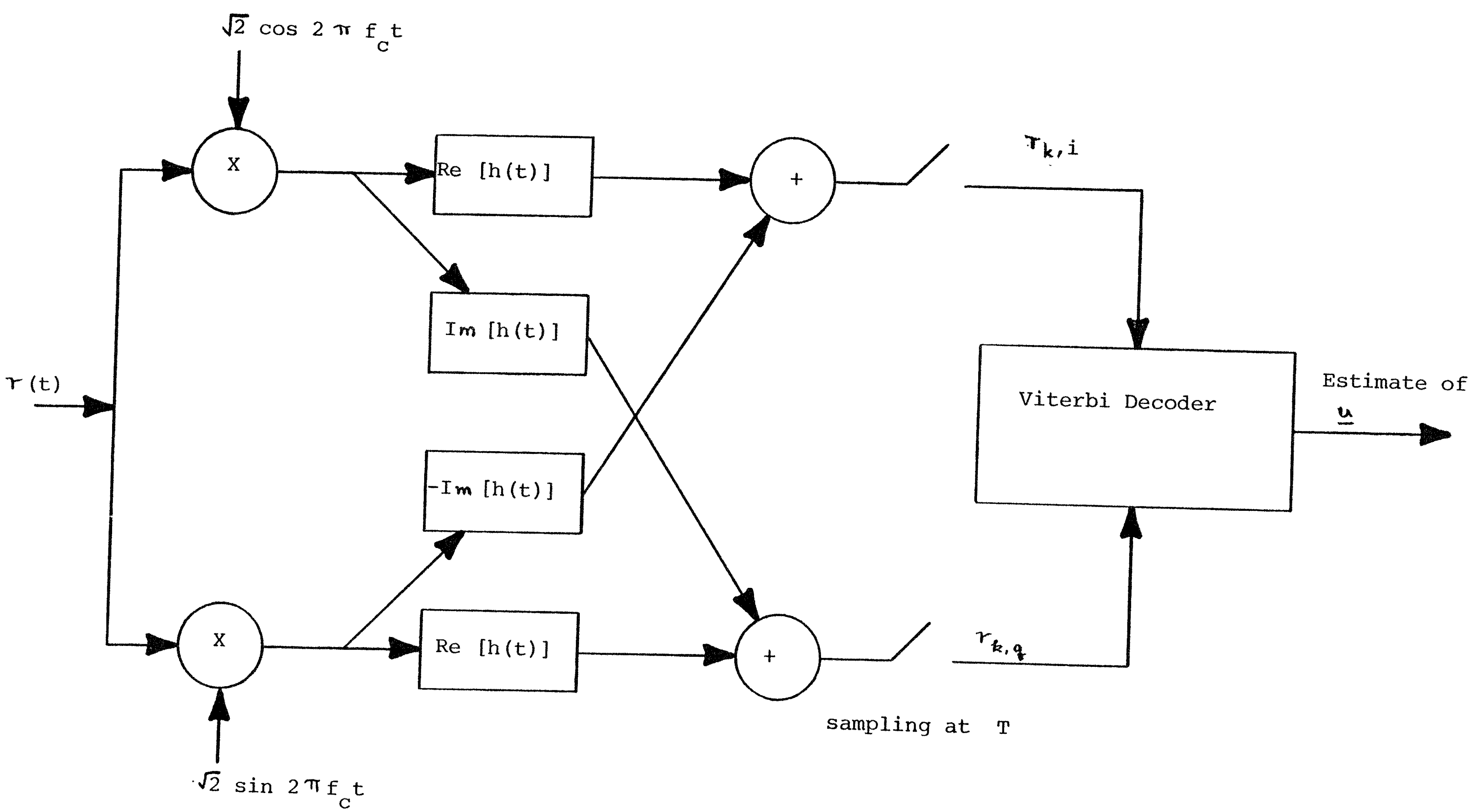


Figure 3.5. Demodulator Structure

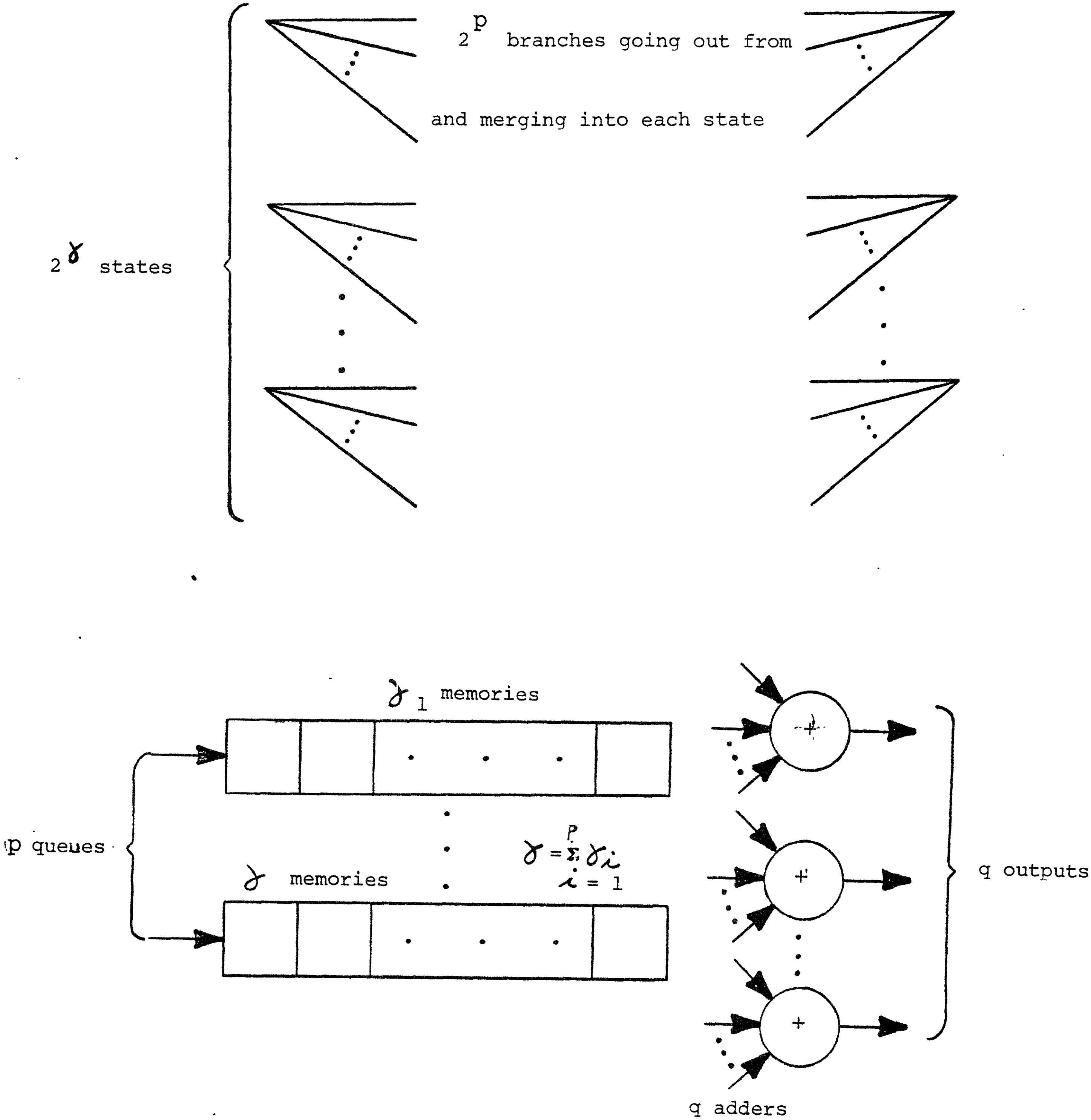


Figure 3.6. Trellis Diagram and Encoder Structure for Rate p/q Encoder With γ Memories

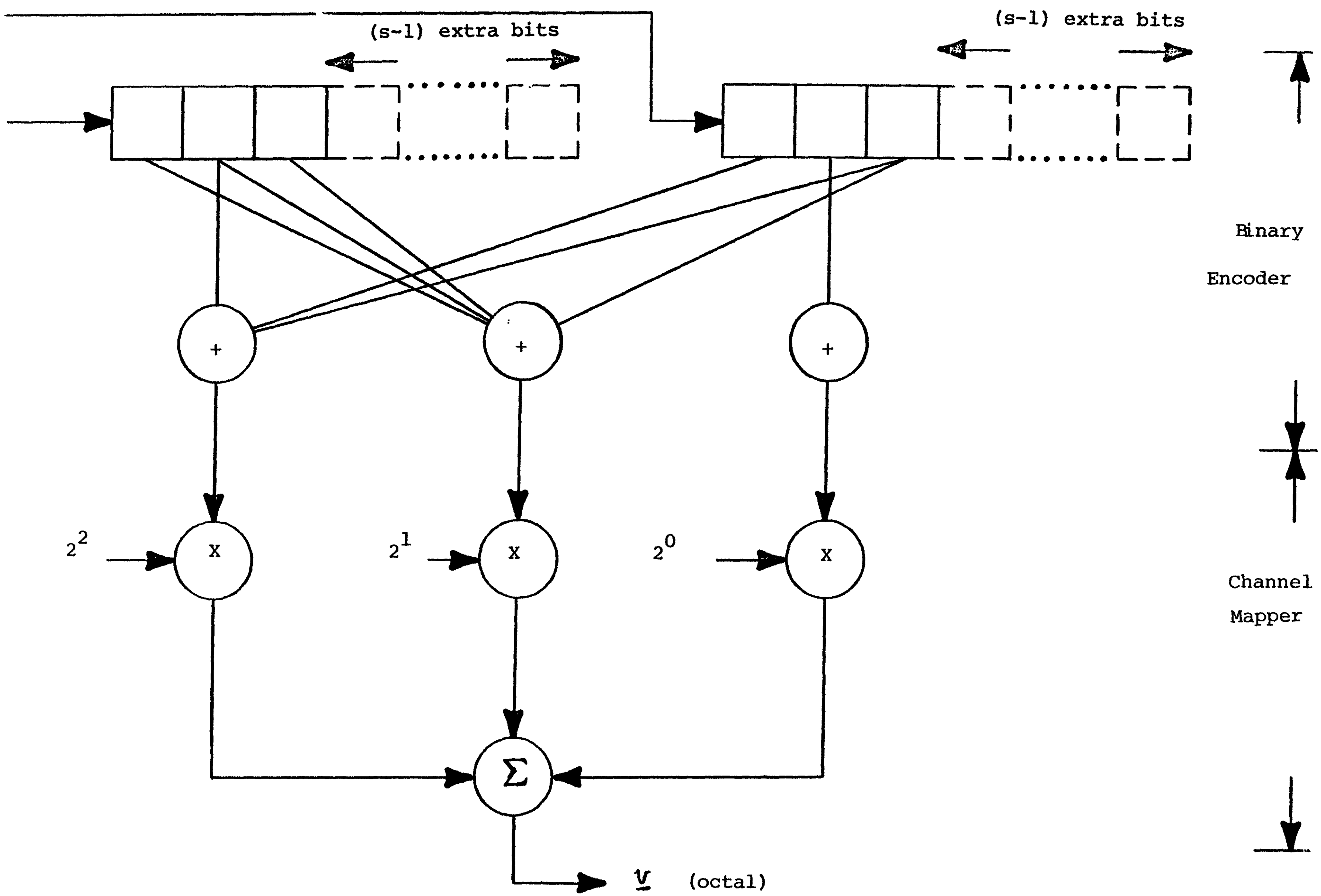


Figure 3.7. Configuration of a Rate 2/3 Convolutional Encoder with 2(s-1) Extra Bits to Denote Memory Due to ISI of Channel

3.4 Staggering Quadrature Components for Limiting Spectral Sidelobe Regrowth

Often times, one of the quadrature component is delayed by half a repetition interval T forming the so called staggered 4ϕ-PSK. For Time Division Multiple Access (TDMA) transmission, the transponder power amplifiers are operated near saturation and behave like band-pass soft limiters. To avoid adjacent channel interference, the signal is usually band-pass filtered before it is amplified. However, for unstaggered 4ϕ-PSK modulation, the nonlinear amplifier causes the spectral sidelobes to regrow to nearly their original unfiltered level. In this section, we attempt to explain this phenomenon and why staggering quadrature components may reduce spectral sidelobe regrowth.

For conventional 4ϕ-PSK (with rectangular pulse shaping), the transmission often times has phase shifts of π radians when the polarity is reversed (Figure 3.8). When the signal is filtered, envelope nulls occur in the regions where polarity is reversed, resulting in envelope fluctuations (Figure 3.9). The nonlinearity of power saturation or envelope limiting tends to restore these envelope nulls and consequently brings back the high frequency content of the signal that has been filtered out previously.

On the other hand, staggered 4ϕ-PSK makes $\pm\ \pi/2$ radians phase transitions only and avoids the phase shifts of π radians that cause large envelope fluctuations, thereby limiting the spectral sidelobe regrowth resulting from restoration of envelope nulls. Consequently, the staggered 4ϕ-PSK could induce less adjacent channel interference than conventional 4ϕ-PSK in transmission systems which do not suppress the spectral sidelobes by filtering the output of the nonlinear amplifier.

The demodulator for the staggered case consists also of matched filtering of the received signal as in the unstaggered case.

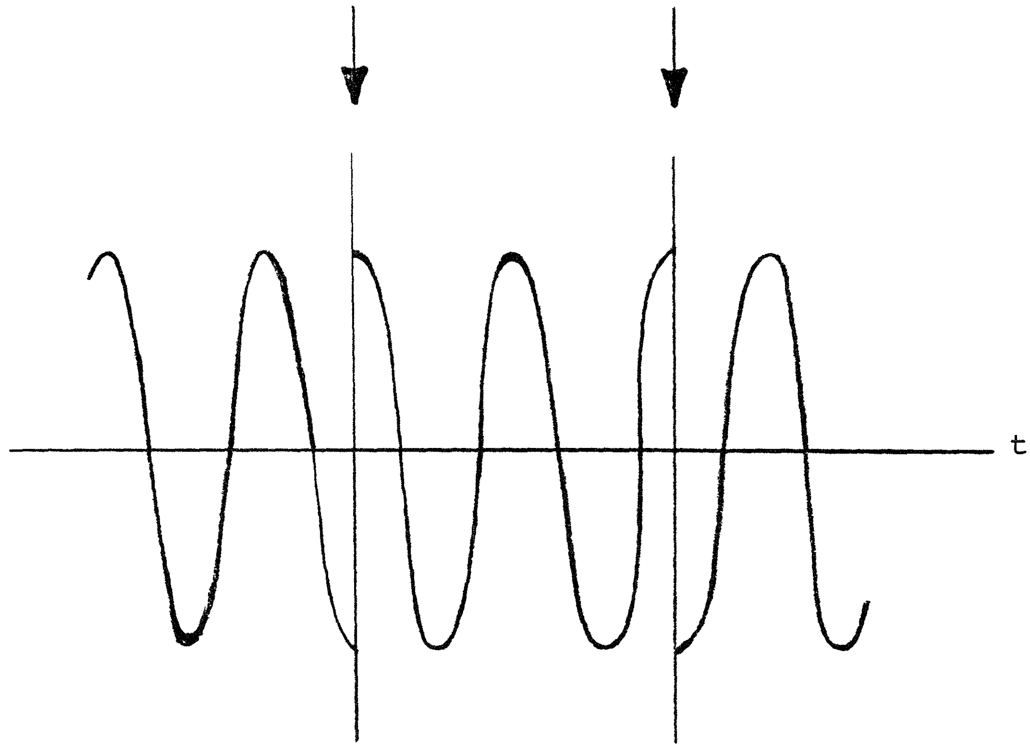


Figure 3.8. Reversed Polarity in 4ϕ-PSK

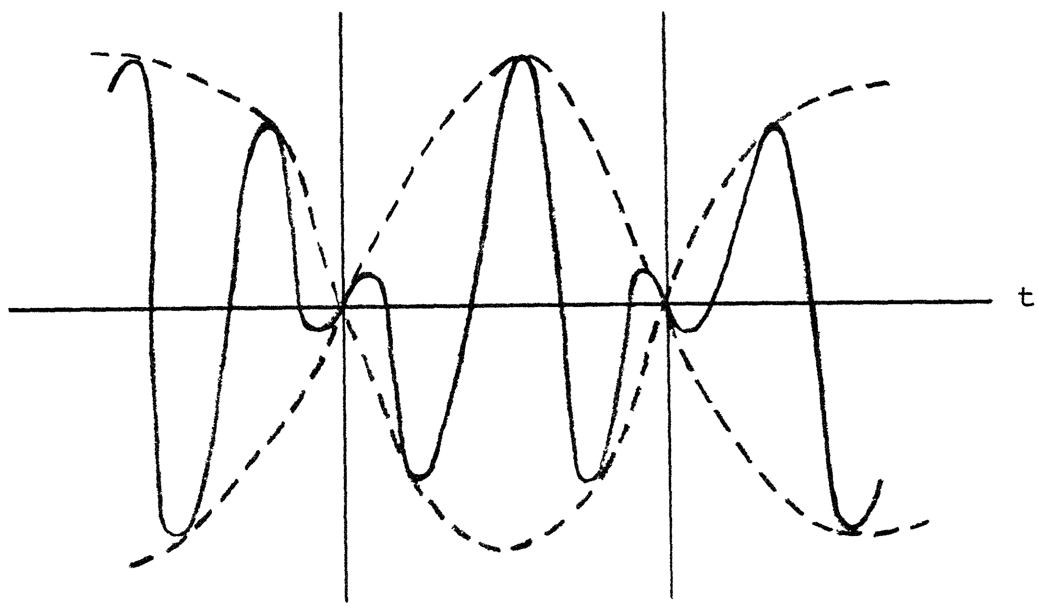


Figure 3.9. Envelope Null at Reversed Polarity
After Filtering

In a linear system, error performance and spectral shape are not changed for the staggered case. Therefore, the mathematical analyses in this thesis would refer to the unstaggered case.

This equivalence between the staggered and unstaggered cases will not be preserved for a nonlinear channel. Before moving onto the next chapter, it is worthwhile to look at cross-coupling interference that results from envelope limiting of the nonlinear TDMA transmission environment. The nonlinearity of power saturation introduces an interaction between the in-phase and quadrature components. This phenomenon is best explained by considering the effect of envelope limiting on filtered MSK (minimum shift keying) signals. MSK is a special form of staggered 4ϕ-PSK (d = T/2) that uses half-cycle sinusoidal pulse shaping. Due to its constant envelope, MSK can be useful for nonlinear transmission such as in TDMA satellite communication. Unfortunately, MSK has a spectral main lobe (the spectrum between the two nulls nearest to the zero frequency) which is 50 percent wider than 4ϕ-PSK with rectangular pulse shaping, which makes MSK infeasible if the transponder frequency spacing B_c is very tight relative to the quaternary symbol rate R_s. An example of such congested environment would be the INTELSAT-V TDMA system using 4ϕ-PSK. It is planned to transmit at 120 Mbit/s with a channel spacing of B_c = 80 MHz, giving a B_c/R_s ratio of 1.33. MSK communication requires a significantly larger B_c/R_s ratio in order that detection performance would not be significantly deteriorated by ACI.

To reduce the bandwidth required for MSK, it is necessary to filter the signal, which would then introduce intersymbol interference. The null of one of the quadrature modulation components shown in Figure 3.10 is smoothed as the filter removes the high frequency content of the sharp corner at the null. The removal of the null causes an envelope boost. Therefore, MSK can still experience significant envelope fluctuations if it is tightly

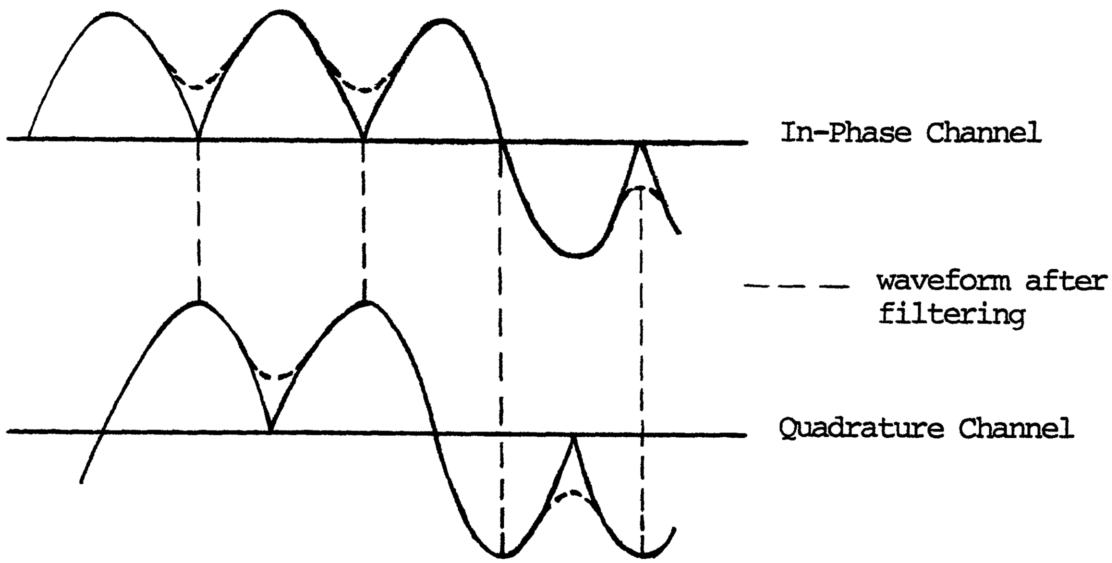


Figure 3.10. Removal of Nulls After Filtering

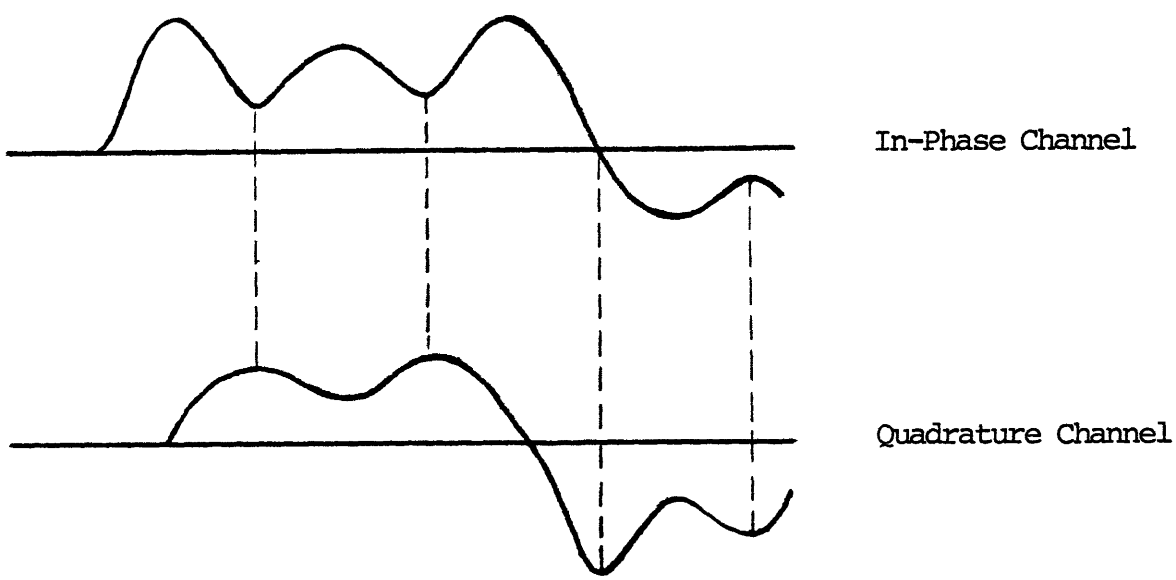


Figure 3.11. Channel Cross-Coupling After Nonlinear Amplifier

filtered. The peak power of the TWTA is shared by the two quadrature components in envelope limiting when the satellite transponders are operated at power saturation. As a result, the voltage level of one component at the location of the former null is increased at the expense of the decreased voltage level of the other component (Figure 3.11). This weakening of some of the pulses would increase the error rate of the system.

It is often over emphasized that constant envelope modulation schemes are more compatible with nonlinear amplifiers (such as TWTAs driven into saturation) than those non-constant envelope modulation schemes. The above example of filtered MSK signaling in a real system environment shows that channel bandwidth limitation inevitably brings in envelope fluctuations as well as intersymbol interference and cross-coupling interference. At small B_c/R_s ratio, signaling waveforms with constant envelopes prior to bandlimiting may perform no better than those with nonconstant envelopes. To optimize system performance, pulse shaping should be designed to match the channel characteristics, and the overly stringent condition of constancy of envelope should be relieved. The question of designing pulse shaping with improved spectral properties will be addressed in the next chapter. The degree of nonconstancy of envelope will be properly defined and quantified.

Chapter 4. FILTER DESIGN WITH SMALL OUT-OF-BAND EMISSION

4.1 INTRODUCTION

An uncertainty relationship between a function h(t) and its Fourier transform $H(\omega)$ states that the mean square time-spread

$$(\Delta t)^2 = \frac{\int_{-\infty}^{\infty} (t - t_o)^2 \left| h(t) \right|^2 dt}{\int_{-\infty}^{\infty} \left| h(t) \right|^2 dt}$$

and the mean square frequency-spread

$$(\Delta\omega)^2 = \frac{\int_{-\infty}^{\infty} (\omega - \omega_o)^2 \left| H(\omega) \right|^2 d\omega}{\int_{-\infty}^{\infty} \left| H(\omega) \right|^2 d\omega}$$

cannot be restricted too severely at the same time for any choice of t_o and ω_o. Specifically the product $(\Delta t)(\Delta\omega)$ is at least 1/2, and equality holds when h(t) is Gaussian, and t_o and ω_o are, respectively

$$\frac{\int_{-\infty}^{\infty} t \left| h(t) \right|^2 dt}{\int_{-\infty}^{\infty} \left| h(t) \right|^2 dt} \quad \text{and} \quad \frac{\int_{-\infty}^{\infty} \omega \left| H(\omega) \right|^2 d\omega}{\int_{-\infty}^{\infty} \left| H(\omega) \right|^2 d\omega}$$

This mathematical statement of the uncertainty principle can be put into a communication theory context. If we want to achieve high speed data transmission by using baseband pulses h(t) which are becoming more limited in time-spread Δt, it follows from the uncertainty principle that $H(\omega)$ would be broadened spectrally as $\Delta\omega$ increases.

In satellite communication, we would like to achieve as high a rate of data transmission as possible for the allocated spectral band, while out-of-band emission should be kept at an acceptably low level.

Ideally, one would like to have zero out-of-band emission, thereby implying the channel to be strictly band-pass in nature. However, such a sharp cut-off for pulse shaping results in infinite ringing of the time-domain signal which would be difficult to implement. Throughout this thesis, many baseband pulses considered are limited in time duration. Therefore, criteria have to be established to evaluate the out-of-band emission of these time-limited pulses.

Three such criteria may be cited as follows. First, the total energy that falls out of band may be computed and constrained to be less than a certain level. Second, the largest out-of-band sidelobe may be constrained to have a peak power less than the peak spectral density of the main lobe by a certain amount. The third method constraints the rate of roll-off of the spectrum.

The solution to the first approach is the well-known class of prolate spheroidal functions[2]. Ringing is observed for the subclass of time-limited prolate spheroidal functions.

The second method, even though crude in nature, is often times fairly robust and computationally economical for given pulse shaping. However, a mathematically tractable formulation for finding the pulse which is optimal in the sense of this criterion is rather unlikely.

The third criterion will predominate the discussion of this chapter due to its ease of formulation for the optimization. We shall observe that the optimized pulse generally does not ring, which perhaps is an advantage over the prolate spheroidal function. The rate of spectral roll-off is related to the moments of the spectrum; i.e.,

$$\int_{\infty}^{\infty} f^{2n} \left| H(f) \right|^2 df$$

for various values of n. Specifically, if the 2n-th moment is bounded, then $|H(f)|^2$ must decrease faster than f^{2n+1}.

In theory, one could achieve a high fraction of power in band and rapid roll-off out of band by using h(t) of long duration, yet avoiding ISI provided it satisfies Nyquist's criterion. However, it can be shown that the equivalent noise bandwidth (defined in Section 4.3) times the repetition interval T cannot be less than unity without introducing ISI, thus setting a limit to the bandwidth economy. Very often, technological problems make complete absence of ISI hard to achieve. First of all, the desired pulse shape may be difficult for implementation. Second, system performance would be sensitive to timing errors. Third, nonlinearity of the channel may introduce ISI at the sampling instants. Furthermore, such pulse shaping may produce a high variability in envelope, which is undesirable for satellite communication. The availability of the Viterbi algorithm for decoding the effect of ISI enables us to abandon the traditional approach of avoiding ISI by satisfying the Nyquist criterion.

The optimization of bandwidth economy must start from defining spectrum spread and time spread in a manner that reflects the characteristics of the communication system. In terms of these definitions, uncertainty principles which give a lower bound for the product of the spectrum spread and the time spread can be obtained. The optimal pulse shape is defined as the one which achieves the value of the lower bound. Consequently, the optimal pulse shape would be different under various definitions of the uncertainty principle. The degree of nonconstancy of envelope will be pictured by the filter loss defined later in the chapter.

4.2 PULSE SHAPE OPTIMIZATION

The time spread of h(t) is defined as the interval τ over which h(t) is nonzero. For symmetry's sake, this interval is assumed to be $[-\tau/2, \tau/2]$. A larger time spread for given repetition rate 1/T may give more nonzero h_i's

$$h_i \triangleq \int_{-\infty}^{\infty} h(t)\, h(t - iT)\, dt$$

consequently increases the complexity of the trellis decoding of the effect of ISI.

The spectrum spread is defined as the weighted moment of h(t), given by

$$Q\{h(t)\} = \int_{-\infty}^{\infty} \sum_{k=o}^{n} a_k f^{2k}\, H(f)\, H^*(f)\, df$$

in which H(f) is the Fourier transform of h(t). Each a_k is a non-negative weight for the 2k-th moment and a_n is assumed to be nonzero. The a_k's have proper dimensions so that Q{h(t)} has the same dimension as the 2n-th moment of H(f) H*(f).

Each term in the summation for Q{h(t)} is bounded if h(t) is (n-1) differentiable for all t. This follows from the fact that if h(t) is j-differentiable, then asymptotically

$$H(f)\, H^*(f) \leqq 0(1/f^{2j + 4})$$

and consequently

$$f^k H(f)\, H^*(f) \leqq 0\ (1/f^{2j-k+4})$$

As a result, the necessary condition for

$$\int_{-\infty}^{\infty} f^k \, H(f) \, H^*(f) \, df$$

to be bounded is

$$k < 2j + 3$$

The m-th power integral of h(t) is defined as

$$R\{h(t)\} = \int_{-\infty}^{\infty} h^m(t) \, dt \quad , \; m \geqq 1$$

$R\{h(t)\}$ for m = 2 corresponds physically to the energy of the pulse and for m = 1, the area under h(t). Without loss of generality, $R\{h(t)\}$ is assumed to be positive by employing - h(t) instead if $R\{h(t)\}$ turns out to be negative.

The equivalent bandwidth of h(t) for a given $A = \{a_k\}$ and m is defined as the bandwidth B_A of the bandpass filter

$$H_b(f) = \begin{cases} H_b(0) & \text{for } \; f \leqq B_A/2 \\ 0 & \text{otherwise} \end{cases}$$

which satisfies the conditions

$$Q\{h_b(t)\} = Q\{h(t)\}$$

$$R\{h_b(t)\} = R\{h(t)\}$$

Once h(t) is given, these two constraints determine the values of B_A and $H_b(0)$. An optimal h(t) is, by definition, the pulse shape that minimizes $B_A\tau$. The physical meaning of some special B_A's will be discussed after we obtain the necessary and sufficient conditions for optimality.

The two constraints are homogeneous in a sense that if $h_b(t)$ and h(t) satisfy the constraints, then so would $\alpha h_b(t)$ and

$\alpha h(t)$, leaving B_A unchanged. Therefore, optimal solutions are always defined up to a constant factor.

It can also be easily seen that B_A is a monotonically increasing function of the weighted moment of h(t) for a given $R\{h(t)\}$. Thus, the solution for B_A is always unique, hence, the optimal h(t), denoted by $h^0(t)$, is the solution to the following constrained optimization problem,

$$\min_h Q\{h(t)\}$$

subject to

$$R\{h(t)\} = C$$

If B_A for $h^0(t)$ is expressed as

$$B_A = \frac{\beta_A}{\tau} ,$$

then the following form of the uncertainty principle is obtained.

<u>For all h(t) of duration τ with equivalent B_A,</u>

<u>we have $B_A \tau \geqq \beta_A$</u>

The remainder of this section is devoted to finding $h^0(t)$.

To express the weighted moment in terms of h(t), Parseval's theorem is applied so that

$$Q\{h(t)\} = \int_{-\infty}^{\infty} \sum_{k=0}^{n} a_k \, H^*(f) \, f^{2k} \, H(f) \, df$$

$$= \int_{-\infty}^{\infty} \sum_{k=o}^{n} a_k \, H^*(f) \, F\{ \left(\frac{j}{2\pi} \right)^{2k} h^{(2k)}(t)\} \, dt$$

$$= \int_{-\tau/2}^{\tau/2} \sum_{k=o}^{n} (-1)^k a_k' \, h(t) \, h^{(2k)}(t) \, dt$$

in which

$$a_k' = (2\pi)^{-2k} a_k$$

In the process, we have assumed h(t) to be well behaved, that is, at least 2n-differentiable in the open interval $(-\tau/2, \tau/2)$. Furthermore, the fact that h(t) is (n-1)-differentiable at all t to keep $Q\{h(t)\}$ bounded implies

$$h^{(k)} \left(\pm \frac{\tau}{2}\right) = 0 \qquad \text{for} \quad 0 \leqq k \leqq n - 1$$

Using Lagrange multiplier techniques, we form the Lagrangian

$$G\{h(t)\} = \sum_{k=0}^{n} (-1)^n a_k' \, h(t) h^{(2k)}(t) + \lambda' h^m(t)$$

The necessary condition for G to be stationary is given by the generalized Euler's equation for calculus of variation problems [18], namely that

$$\frac{\partial G}{\partial h} - \frac{d}{dt} \frac{\partial G}{\partial h^{(1)}} + \ldots + (-1)^k \frac{d}{dt^k} \frac{\partial G}{\partial h^{(k)}} + \ldots + \frac{d}{dt^{2n}} \frac{\partial G}{\partial h^{(2n)}} = 0$$

which, after simplification, is reduced to the form

$$\sum_{k=o}^{n} (-1)^n a_k' \, h^{(2k)}(t) = \lambda h^{m-1}(t)$$

in which

$$\lambda = - \frac{1}{2} m \lambda'$$

Therefore $h^o(t)$, the optimal pulse shape, is the solution for a particular eigenvalue of the above 2n-th degree differential equation (nonlinear for $m \neq 1$ or 2) satisfying the 2n boundary conditions at $t = \pm \tau/2$.

Multiplying both sides of the above differential equation by $h(t)$ and afterwards integrating over $[-\tau/2, \tau/2]$, then for optimal pulse shape

$$Q\{h^{o}(t)\} = \lambda R\{h^{o}(t)\}$$

Since both $Q\{h(t)\}$ and $R\{h(t)\}$ are positive for all $h(t)$, λ is positive and the sufficient condition for $h(t)$ to be optimal is to possess the smallest λ possible.

4.3 SEVERAL PULSE SHAPES

We are going to solve the two cases of $m = 1$ and 2 for the above formulation. In both cases,

$$a'_k = \begin{cases} 1 & \text{if} \quad k = 2n \\ 0 & \text{otherwise} \end{cases}$$

As a result, specifying m and n is equivalent to specifying A, and therefore, we shall substitute the subscript A with the subscript m,n. The solutions are listed in this section, while the detailed derivation is shown in Appendix A. For both cases, the optimal pulse shapes are observed to be nonringing. All the pulses considered in this section are normalized to have unity energy.

The value $B_{1,0}$ is the equivalent noise bandwidth used so often for error performance analysis. This bandwidth, which we shall denote by B, is the width of the low-pass filter which has the same energy as $h(t)$, or in other words

$$B = \frac{\int_{-\infty}^{\infty} |H(f)|^2 \, df}{|H(0)|^2}$$

We are interested in evaluating the value of B for all pulse shape considered in this chapter due to three reasons. First, it has a physically appealing interpretation. Second, it is easily computable. Third, not all baseband pulses have finite 2nth moment which enable us to compare their BT products, a concept which will be introduced in the next section to describe spectral occupancy of the pulse shapes.

Case 1 m=1

This case corresponds to minimizing the 2n-th moment of a pulse shape with fixed energy. From Appendix A, we have

$$h_n^o(t) = A_n\left(1 - \frac{2t}{\tau}\right)^n \left(1 + \frac{2t}{\tau}\right)^n$$

in which

$$A_n = \frac{[(4n + 1)!]^{1/2}}{(2n)!} \frac{1}{\sqrt{\tau}}$$

We shall call these functions the beta functions, which are similar to the beta distribution found in probability theory. The plots of $h_n^o(t)$ for n from 0 to 4 are given in Figure 4.1.

The spectrum of these five beta functions are derived in Appendix A and are shown in Figure 4.2 through 4.6.

The values of $\beta_{1,n}$ are as follows:

n	0	1	2	3	4	5
$\beta_{1,n}$	1	5.24	8.96	12.40	15.72	18.95

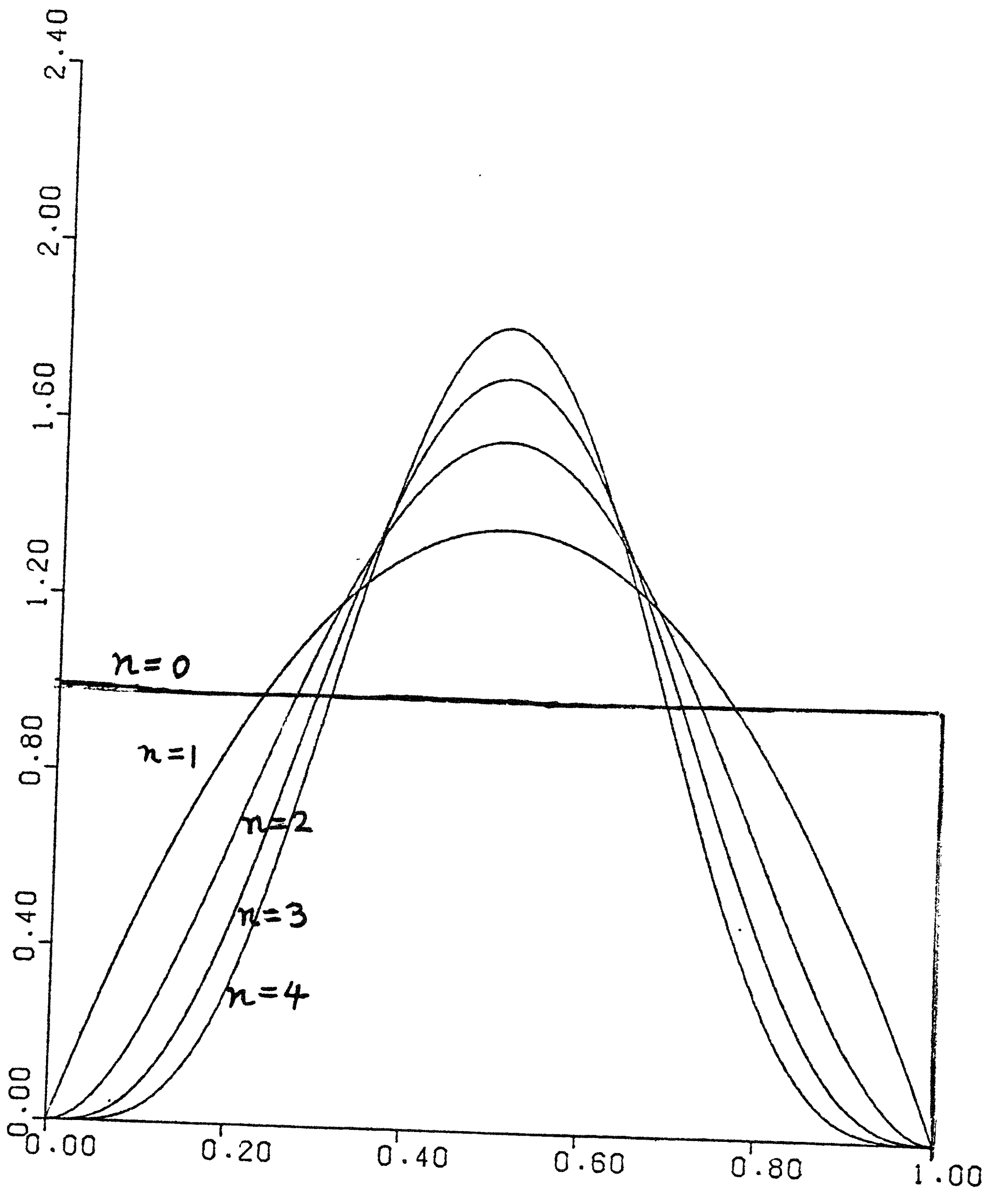


Figure 4.1. N-th Order Beta Functions

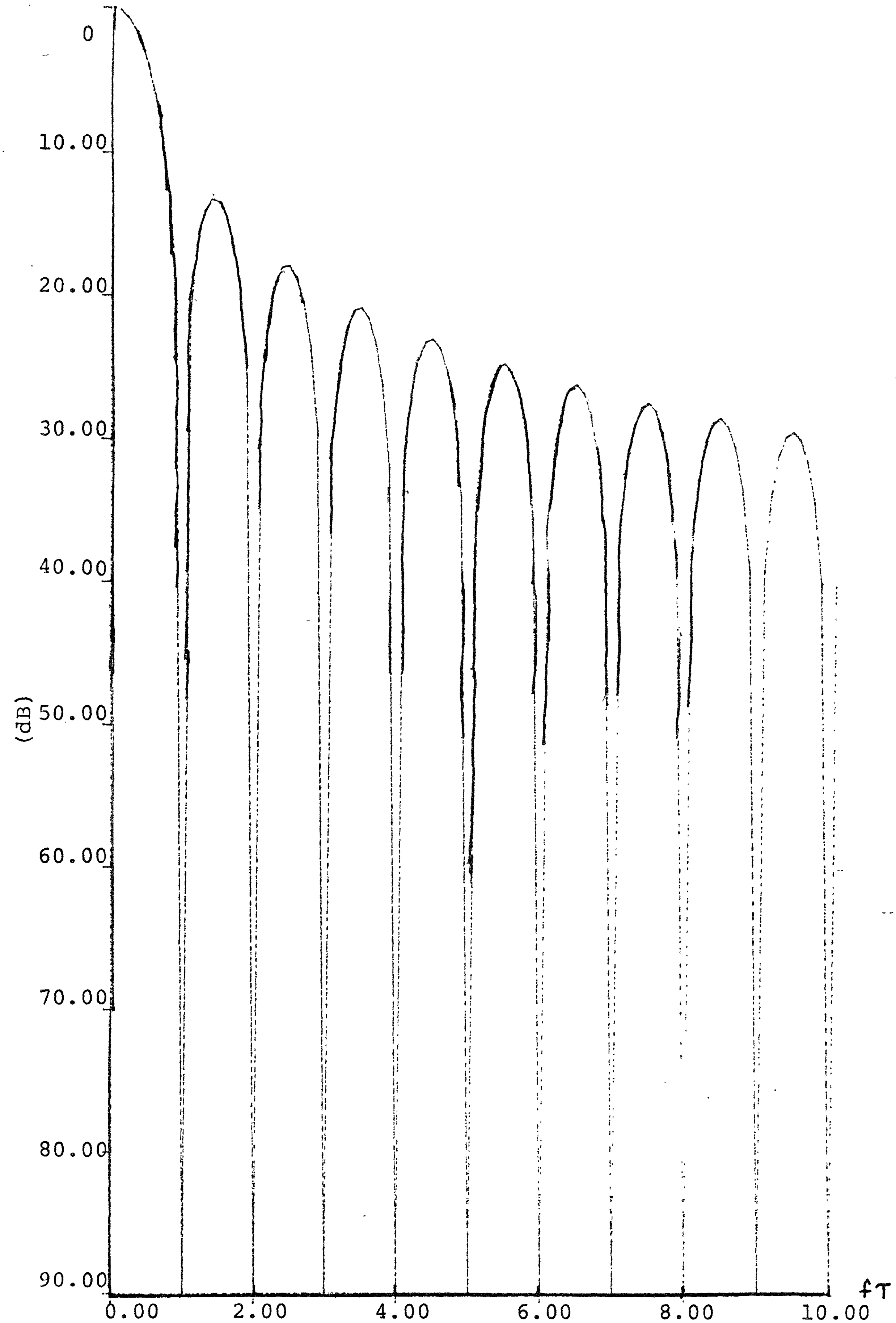


Figure 4.2. Power Spectrum of 0-th Order Beta Function
(Rectangular Pulse Shape, m = 1, n = 0)

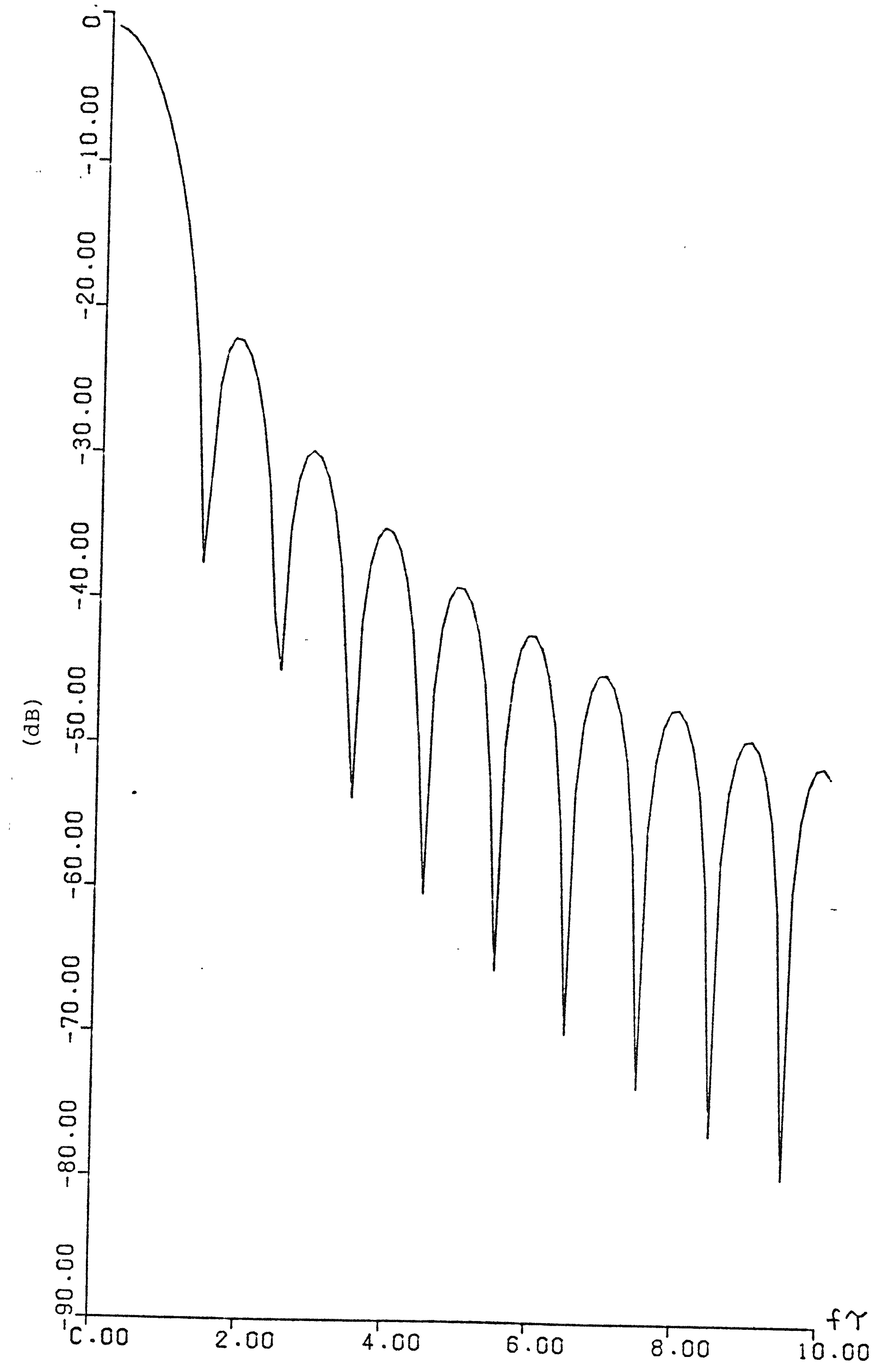


Figure 4.3. Power Spectrum of 1st Order Beta Function
(m = 1, n = 1)

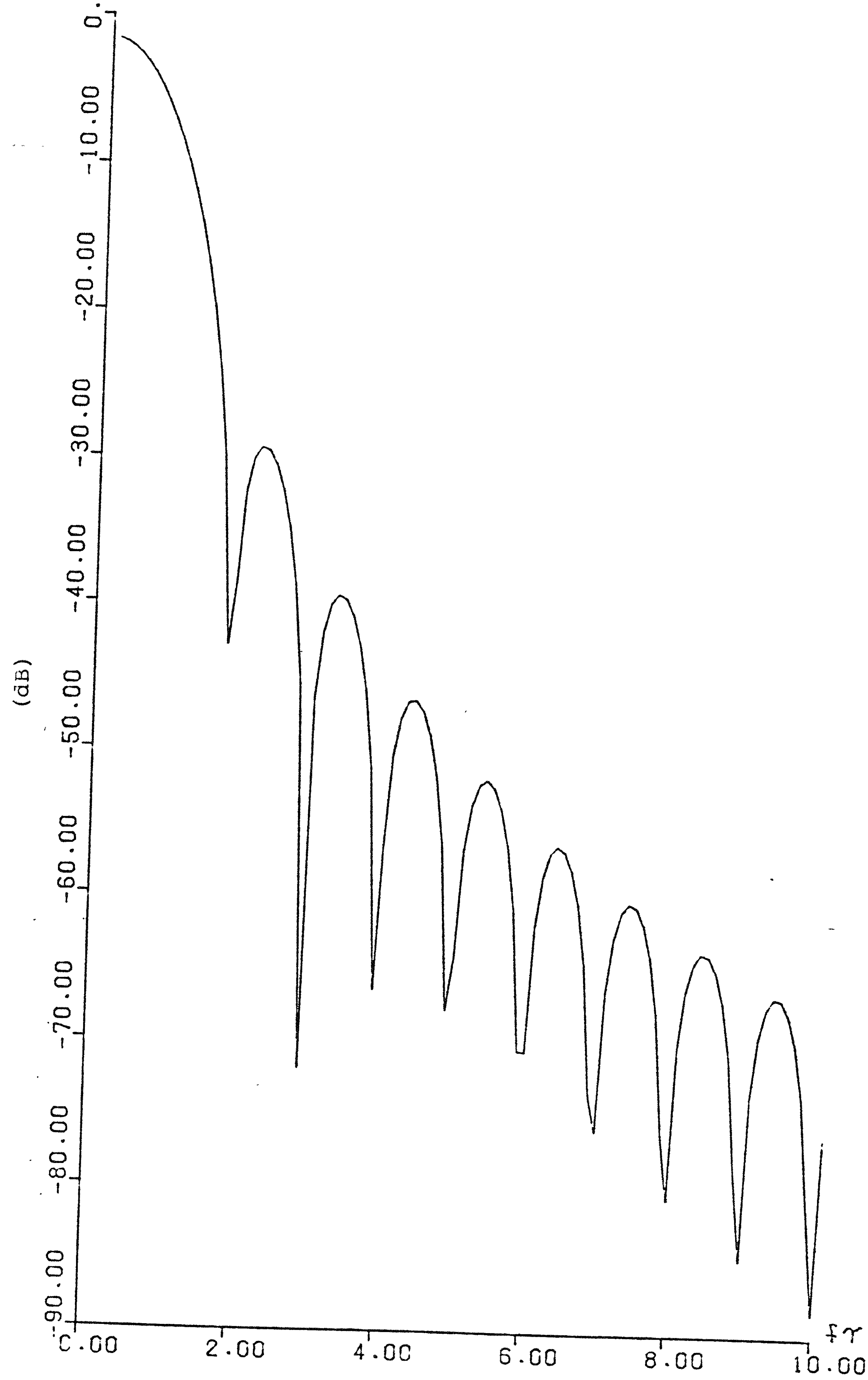


Figure 4.4. Power Spectrum of 2nd Order Beta Function
(m = 1, n = 2)

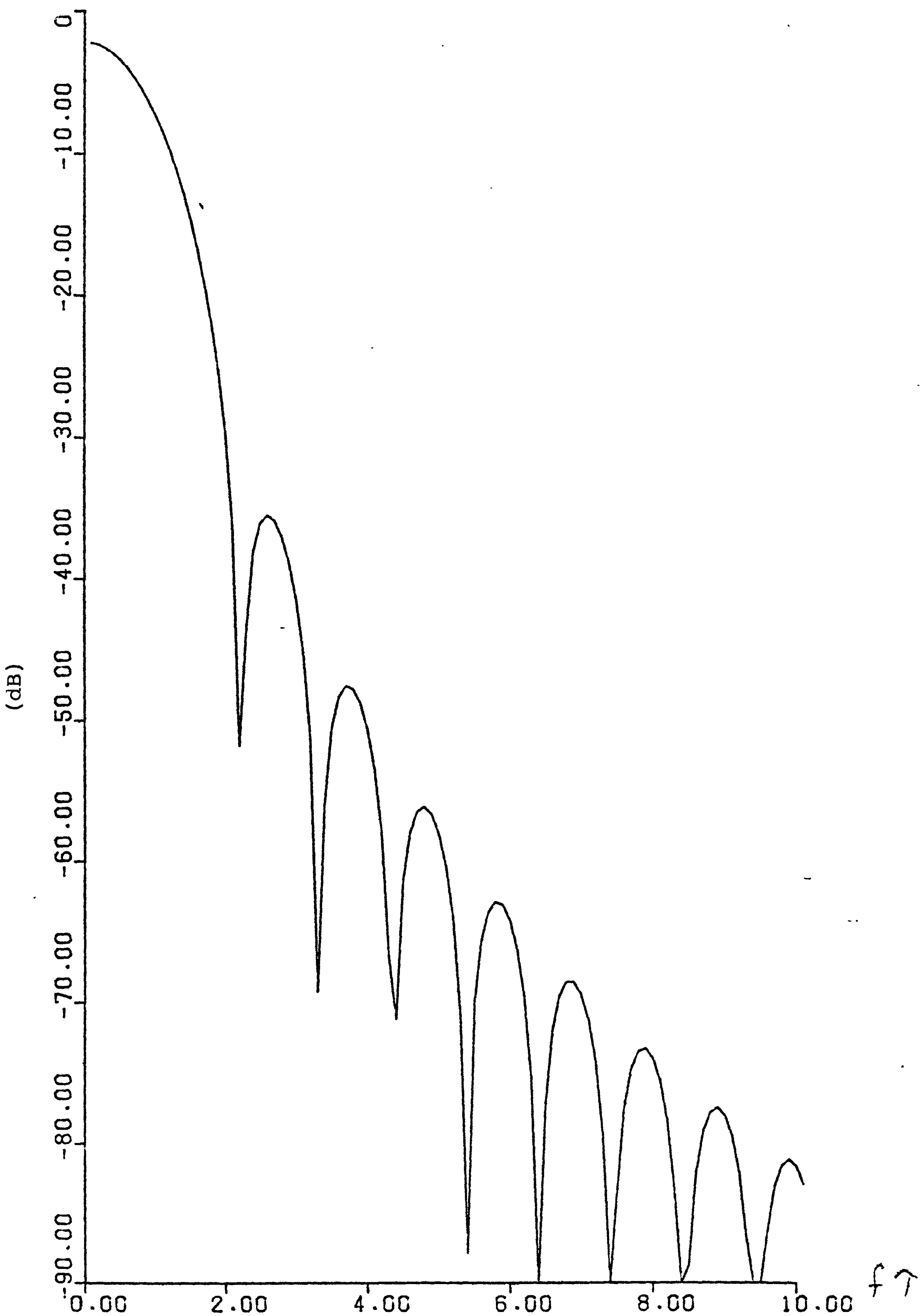


Figure 4.5 Power Spectrum of 3rd Order Beta Function
(m = 1, n = 3)

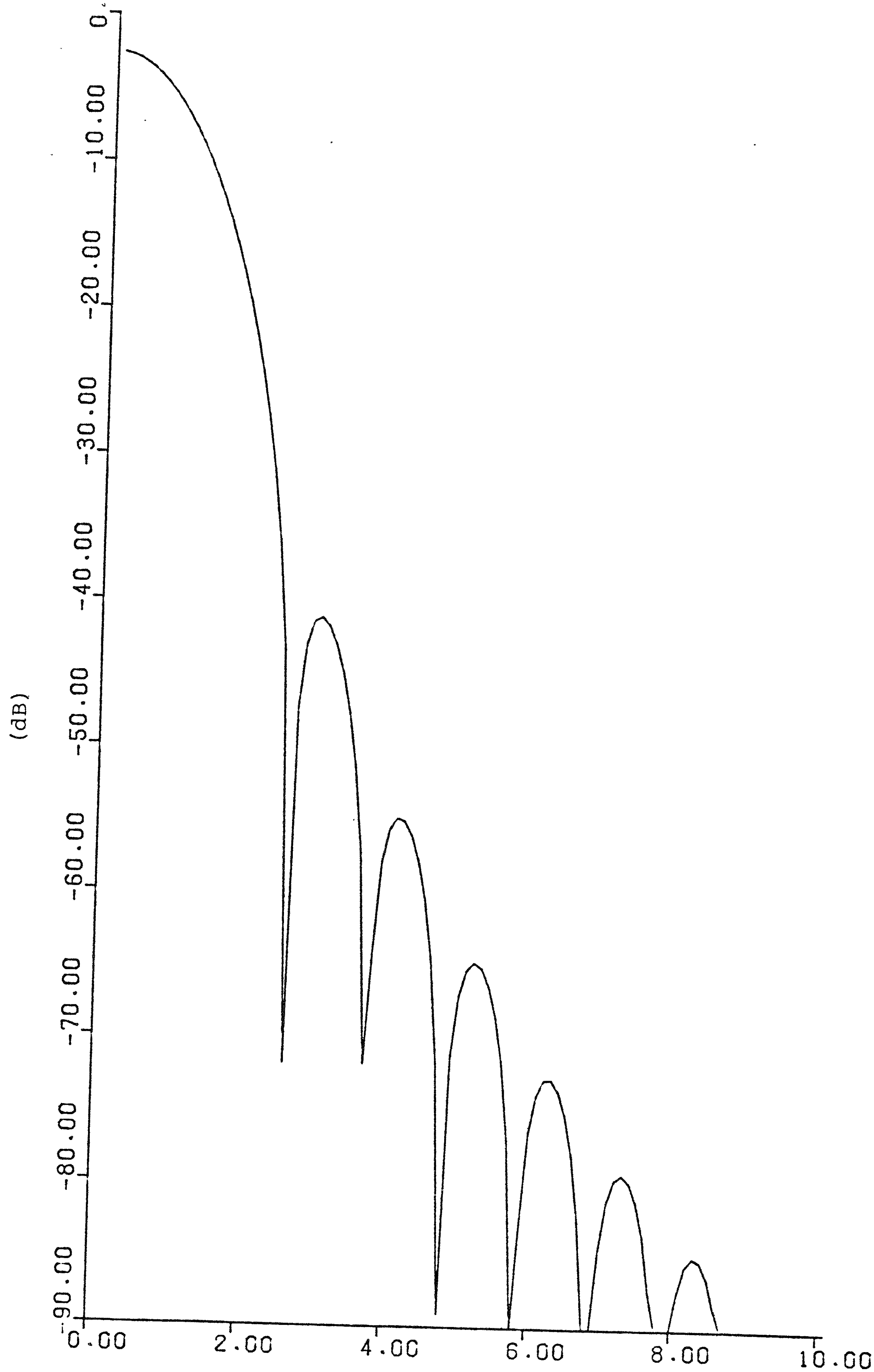


Figure 4.6. Power Spectrum of 4th Order Beta Function
($m = 1$, $n = 4$)

The $B\tau$ products are calculated to be

n	0	1	2	3	4
$B\tau$	1	1.200	1.429	1.630	1.814

Case 2 $\quad m = 2$

This case is physically important because it corresponds to minimizing the 2n-th moment of a pulse shape with fixed energy. For n = 1,

$$h_1^{o}(t) = \begin{cases} \sqrt{\frac{2}{\tau}} \cos \frac{\pi t}{\tau} & \text{for } -\tau/2 \leqq t \leqq \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

The half cosine pulse shape, when used with one quadrature staggered by T/2, forms the well-known minimum shift key (MSK) modulation. The spectrum of this pulse shape is

$$H_1^{o}(\omega) = \left(\frac{8\tau}{\pi^2}\right)^{1/2} \frac{\cos \frac{\omega\tau}{2}}{1 - (\omega\tau/\pi)^2}$$

From Appendix A, we have $\beta_{2,1} = 7.695$ and $B\tau = 1.235$

For n = 2

$$h_2^{o}(t) = 0.1863 \cosh \frac{4.73}{\tau} t + 1.4022 \cos \frac{4.73}{\tau} t$$

The value of $\beta_{2,2}$ is 11.9, and the $B\tau$ product is 1.450.

For m = 2 and n = 1, 2, $h_n^{o}(t)$ are plotted in Figures 4.7 and 4.8, and their Fourier transformations in Figures 4.9 and 4.10. The solution for general n is conjectured in Appendix A. For convenience sake, we shall call the solutions corresponding to m = 2 the trigonometric-hyperbolic functions.

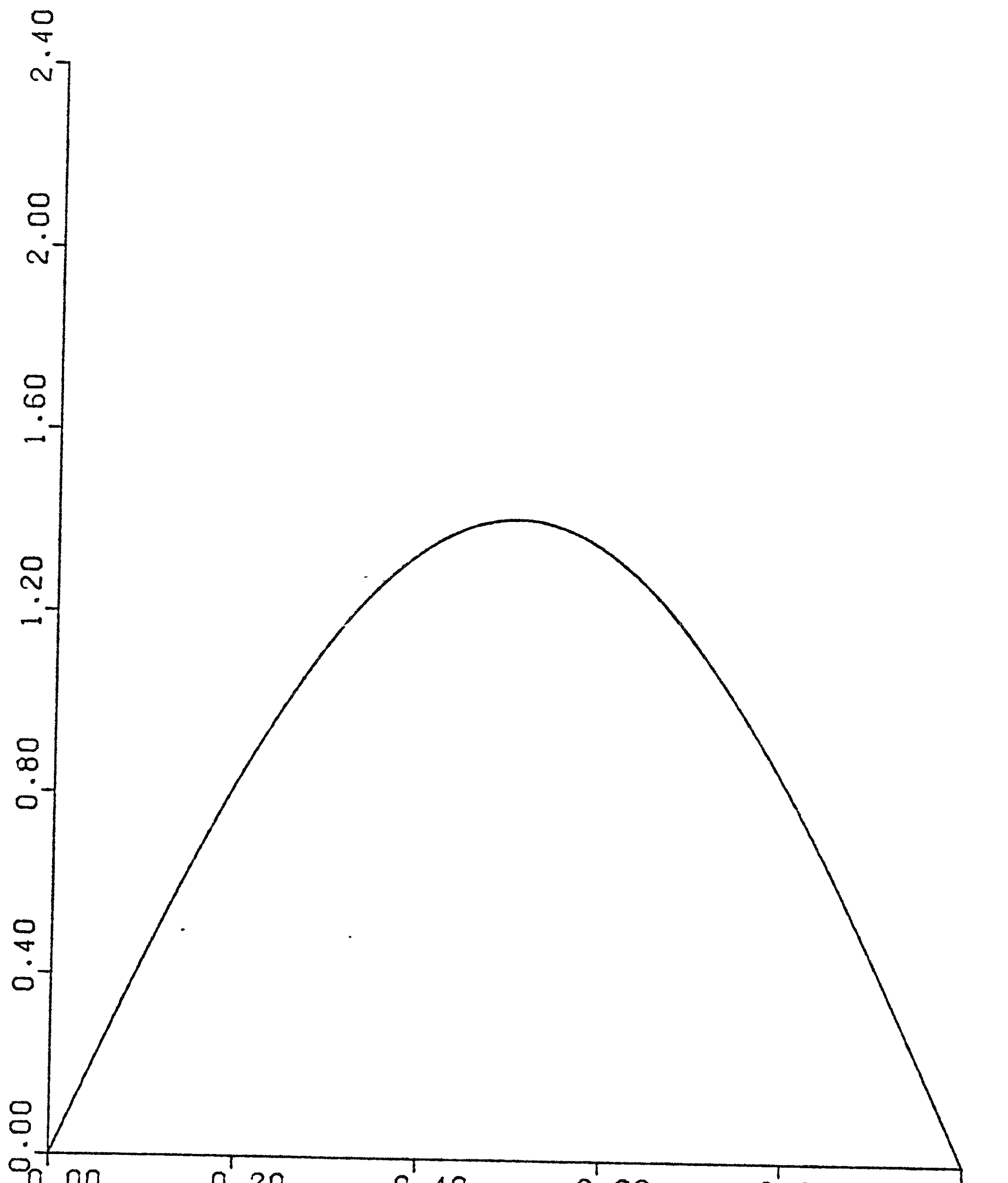


Figure 4.7. Half-Cosine Pulse Shaping
(m = 2, n = 1)

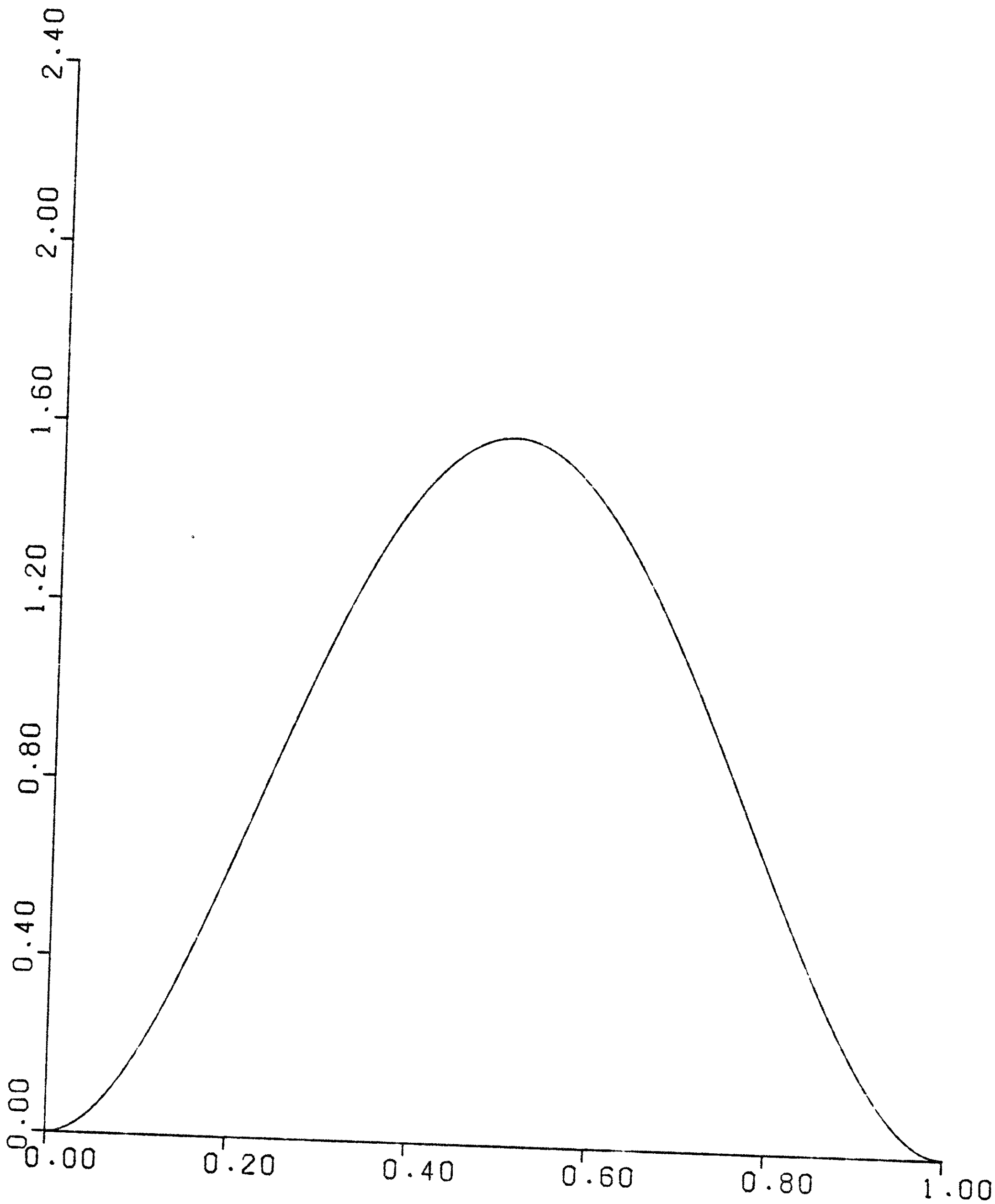


Figure 4.8. 2nd Order Trigonometric-Hyperbolic Function
(m = 2, n = 2)

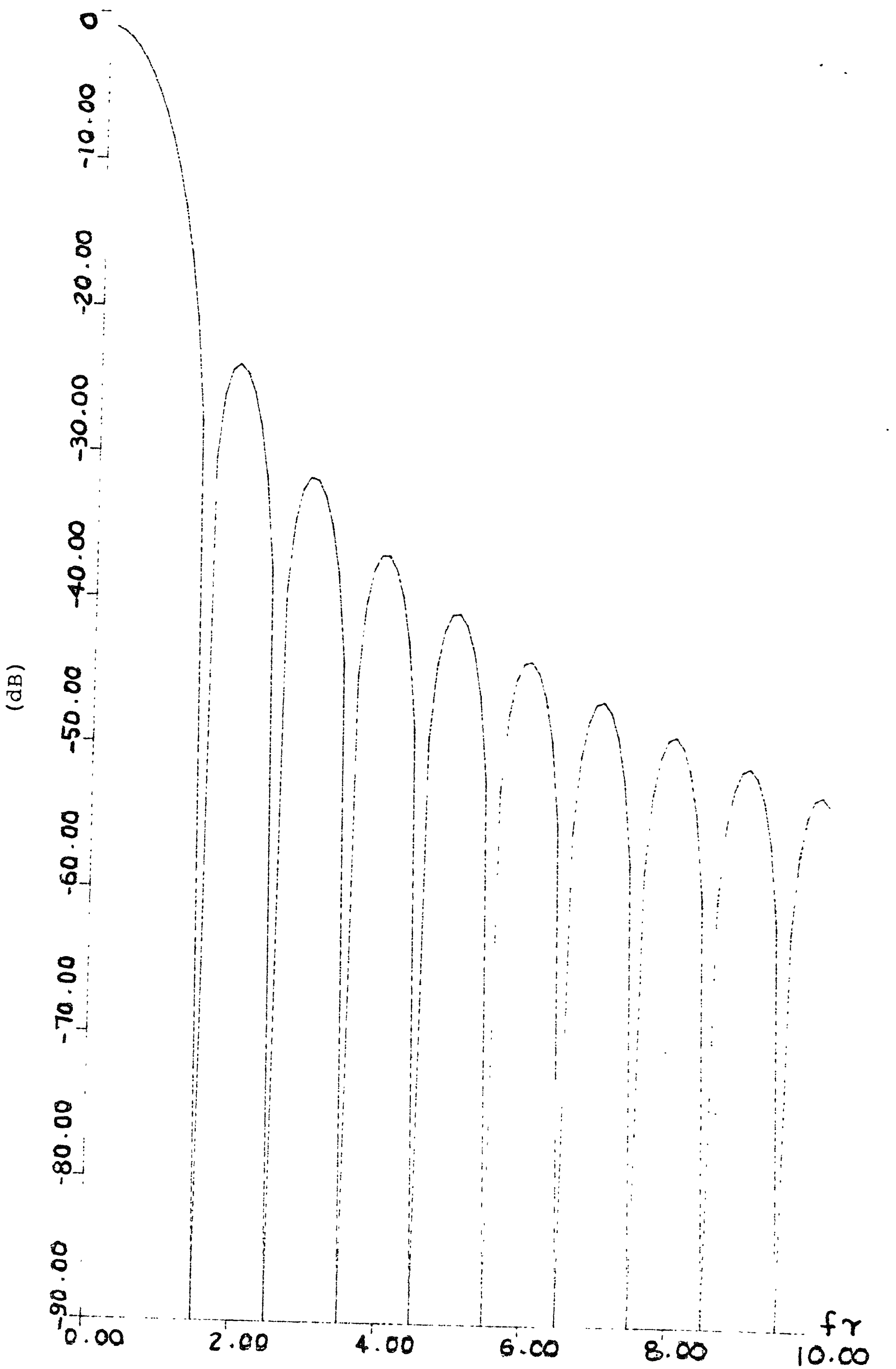


Figure 4.9. Power Spectrum of Half-Cosine Pulse Shaping
($m = 2$, $n = 1$)

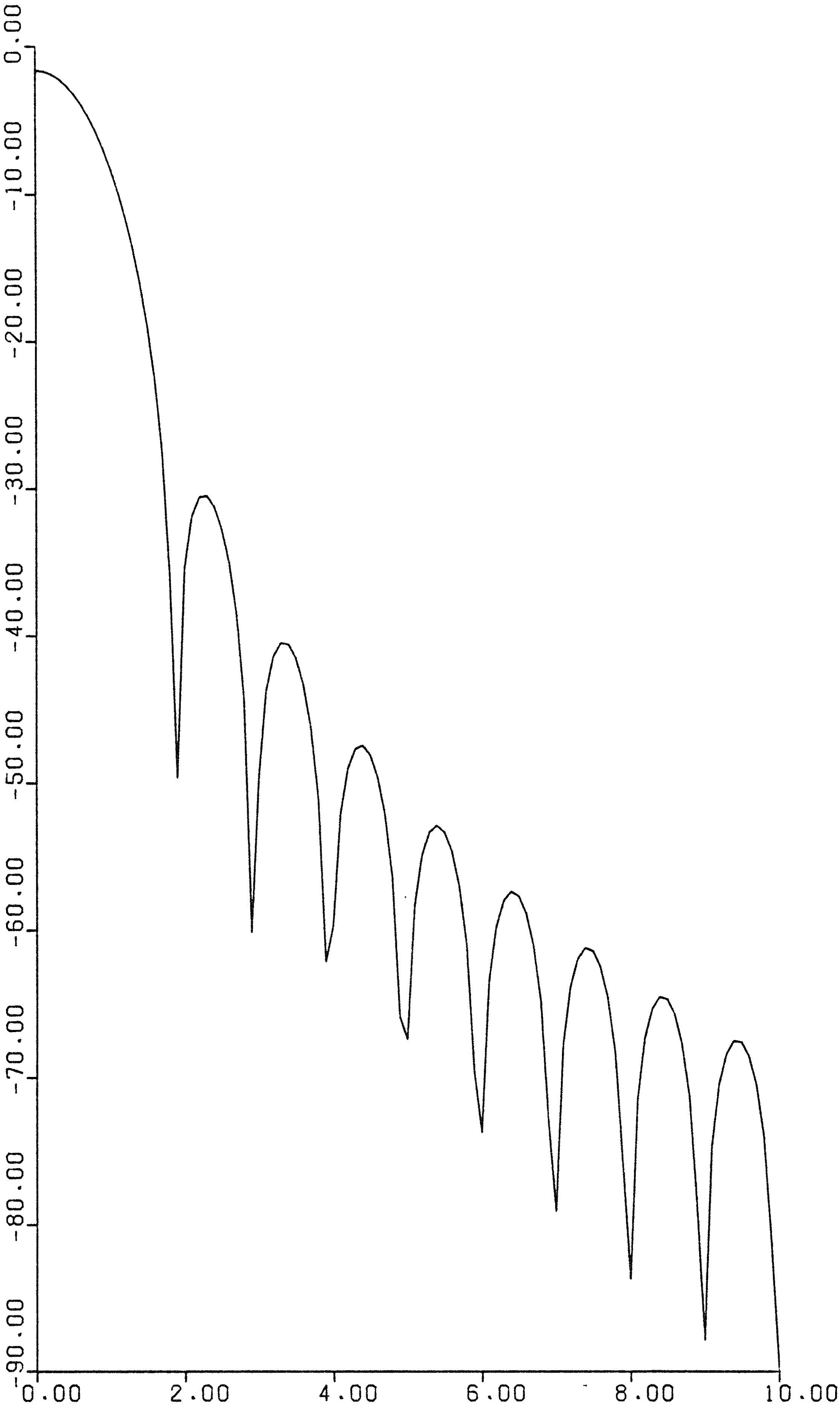


Figure 4.10. Power Spectrum of 2nd Order Trigonometric-Hyperbolic Function (m = 2, n = 2)

To the list of optimal pulse shapes found so far, we shall add two more classes of pulse shapes suggested for usage [6,7] due to their generally good spectral properties as well as their nonringing nature.

Case 3 overlapped raised - cosine pulse (Figure 4.11)

$$h(t) = \begin{cases} \left(\frac{2}{3\tau}\right)^{1/2} \left(1 - \sin \frac{\pi t}{\tau}\right), & -\tau/2 \leqq t \leqq \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

with spectrum (Figure 4.12)

$$H(\omega) = \left(\frac{2\tau}{3}\right)^{1/2} \frac{\cos \omega\tau/2}{1 - (\omega\tau/\pi)^2} \quad \frac{\sin \tau/2}{\omega\tau/2}$$

and

$$B\tau = 1.5$$

Case 4 Truncated n-th power sinc functions

The functions (Figure 4.13) considered are

$$h_n(t) = \begin{cases} \left(\frac{2\pi}{\alpha_{2n}\tau}\right)^{1/2} \left(\frac{\sin 2\pi t/\tau}{2\pi t/\tau}\right)^n & -\tau/2 \leq t \leq \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

in which

$$\alpha_n = \int_{-\pi}^{\pi} \left(\frac{\sin x}{x}\right)^n dx$$

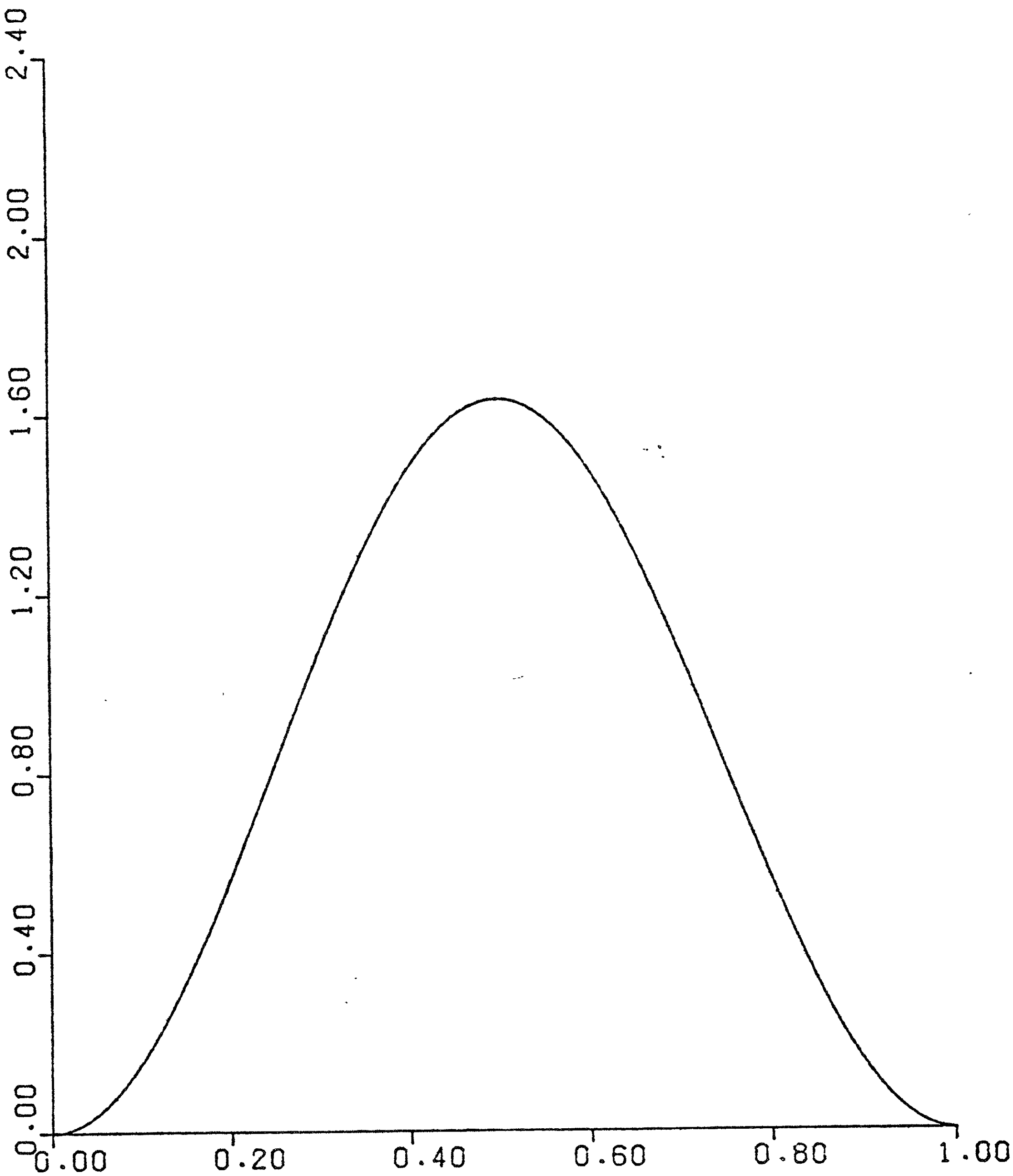


Figure 4.11. Raised-Cosine Pulse Shaping

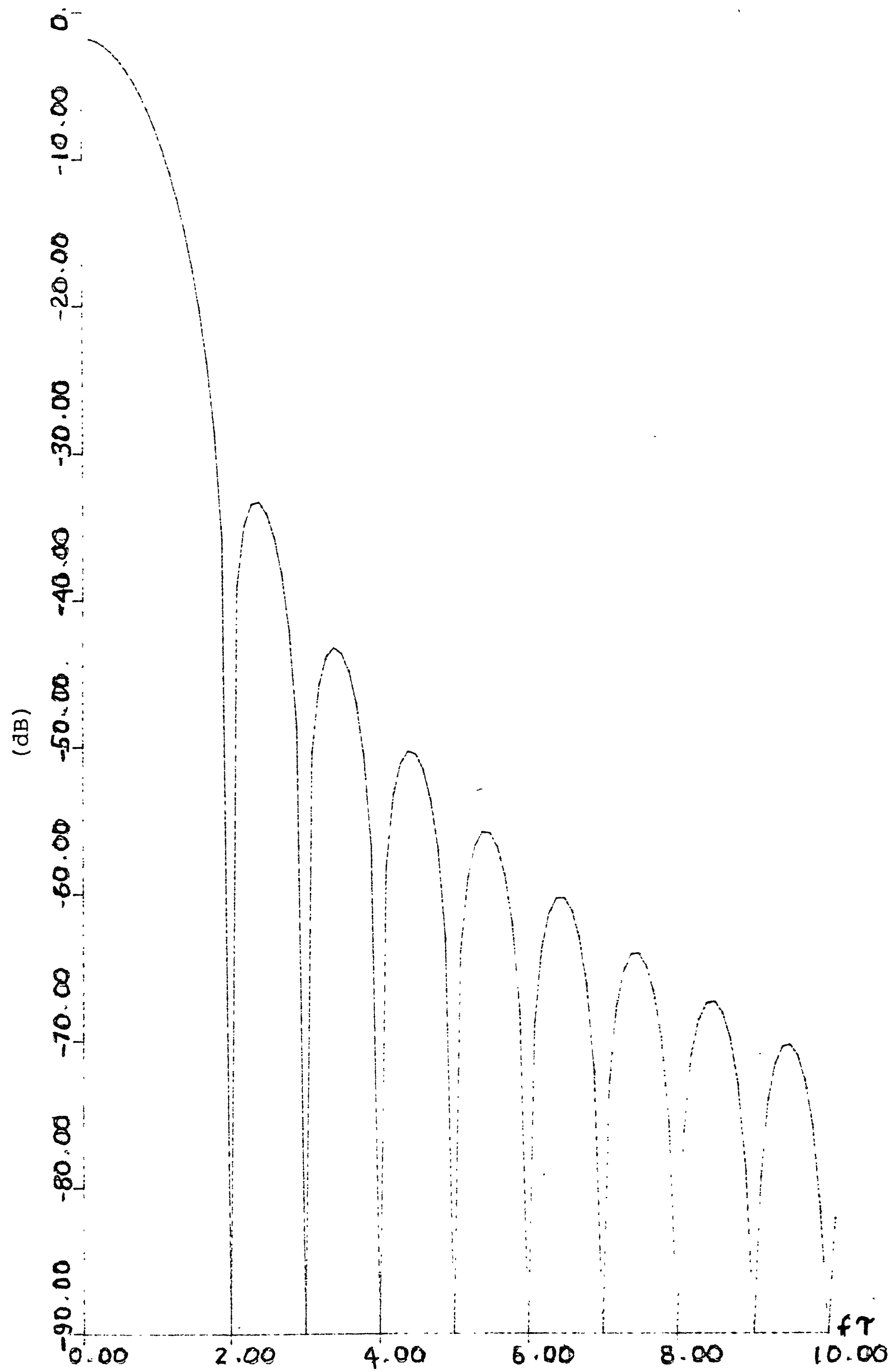


Figure 4.12. Power Spectrum of Raised Cosine

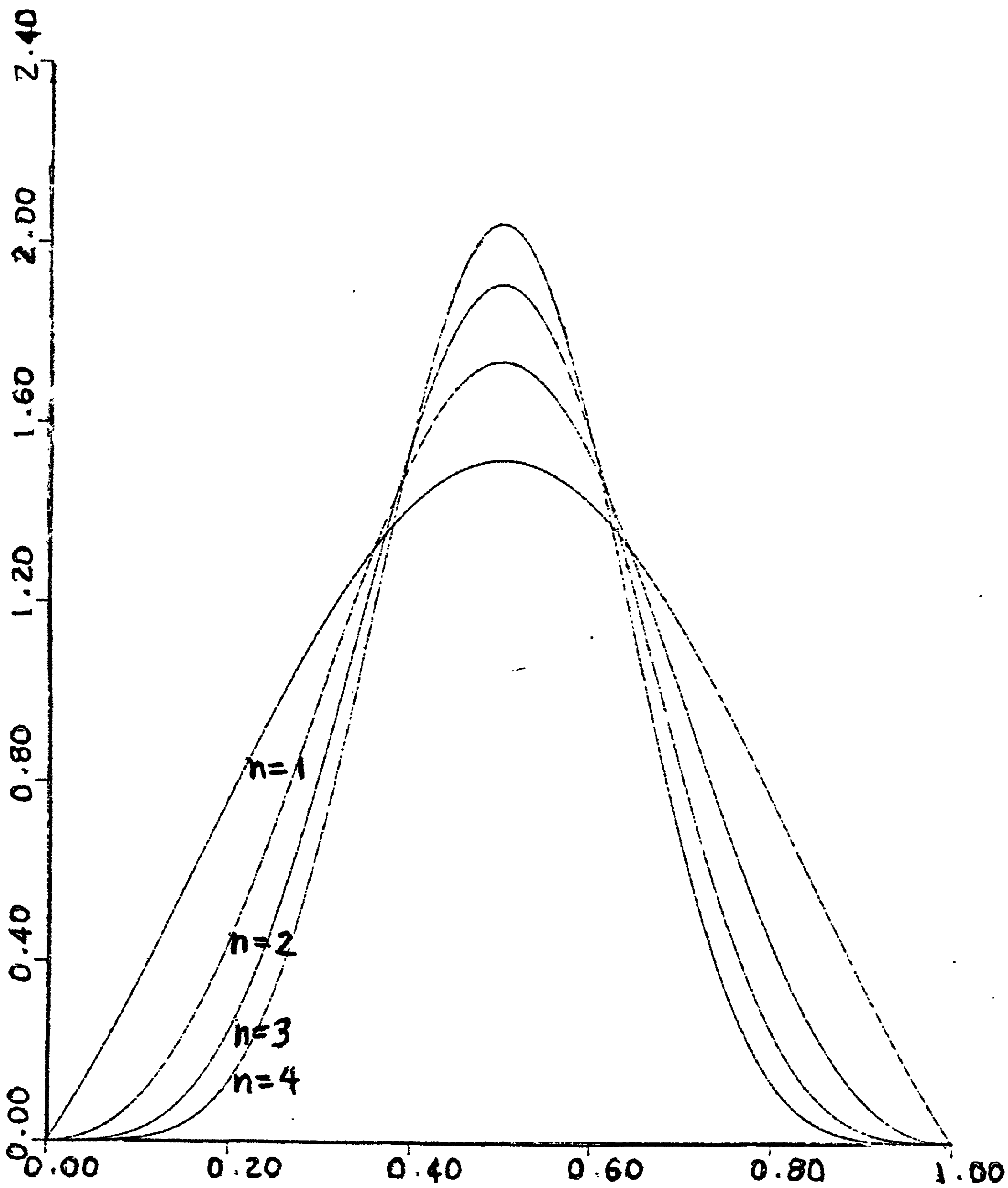


Figure 4.13. N-th Order Truncated Sinc Functions

are computed numerically with values

n	α_n
1	3.70387
2	2.73815
3	2.37926
4	2.08822
5	1.88265
6	1.72765
7	1.60546
8	1.50595

The power spectra for n = 1 to 4 are also computed numerically and shown in Figures 4.14 - 4.17.

The $B\tau$ product for $h_n(t)$ is equal to

$$\frac{\alpha_{2n}}{\alpha_n^2} 2\pi$$

For n = 1 to 4, they are respectively,

n	1	2	3	4
$B\tau$	1.254	1.750	1.918	2.170

So far, the discussion has not addressed the question of how much overlapping of the pulse shape is tolerable during transmission. We shall present an overall design procedure for pulse signalling in the next two sections.

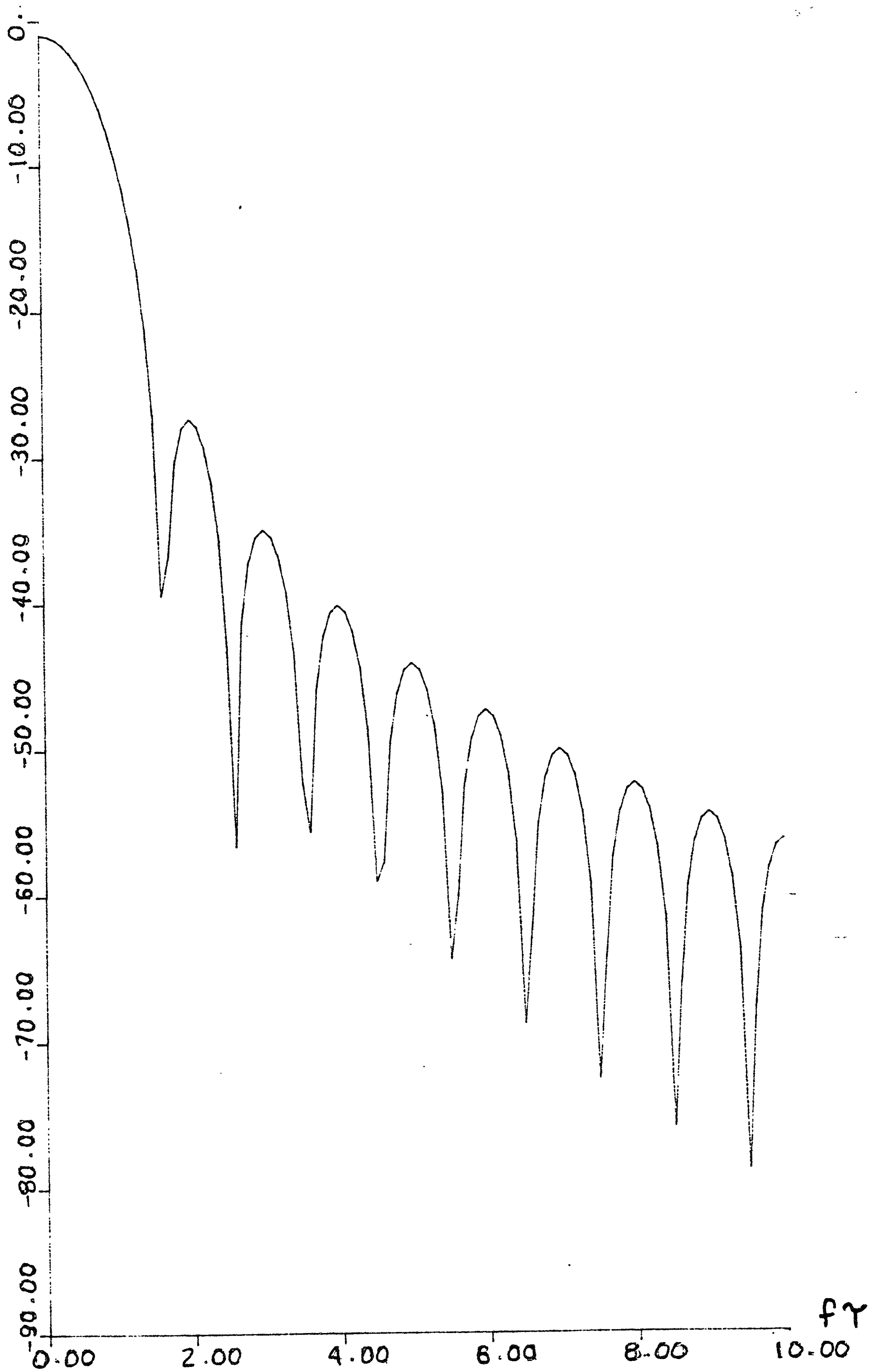


Figure 4.14. Power Spectrum of 1st Order Truncated Sinc Function

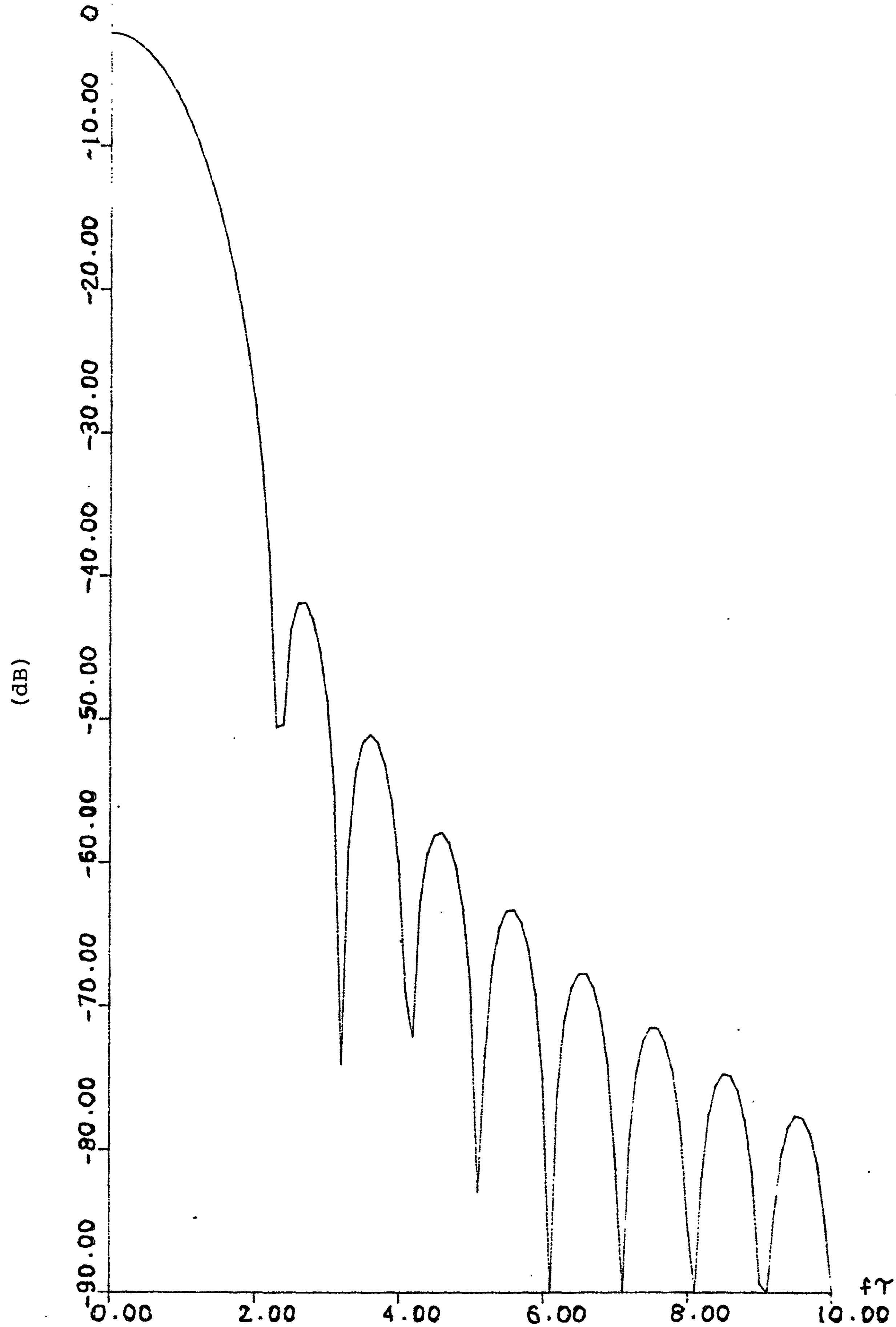


Figure 4.15. Power Spectrum of 2nd Order Truncated Sinc Function

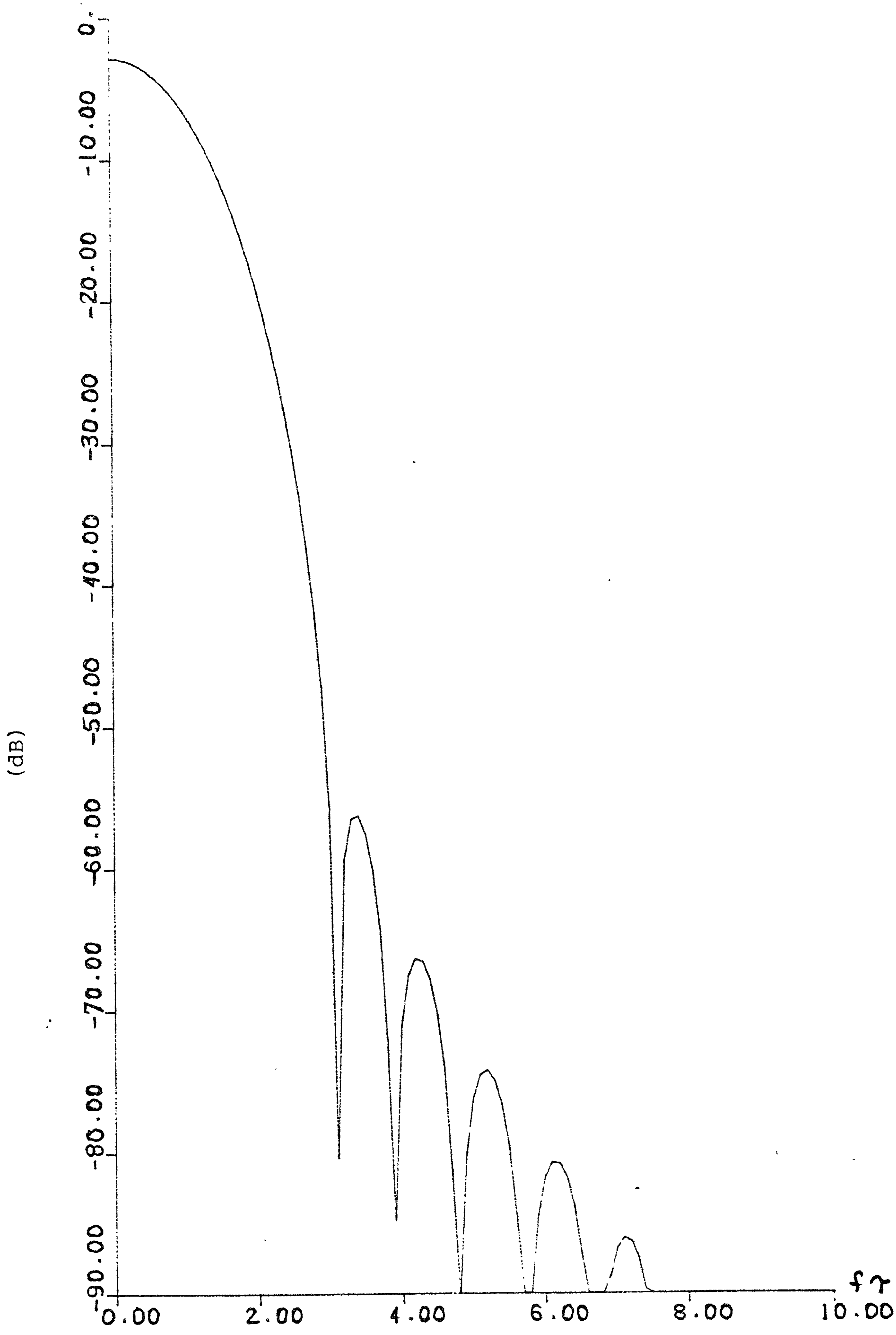


Figure 4.16. Power Spectrum of 3rd Order Truncated Sinc Function

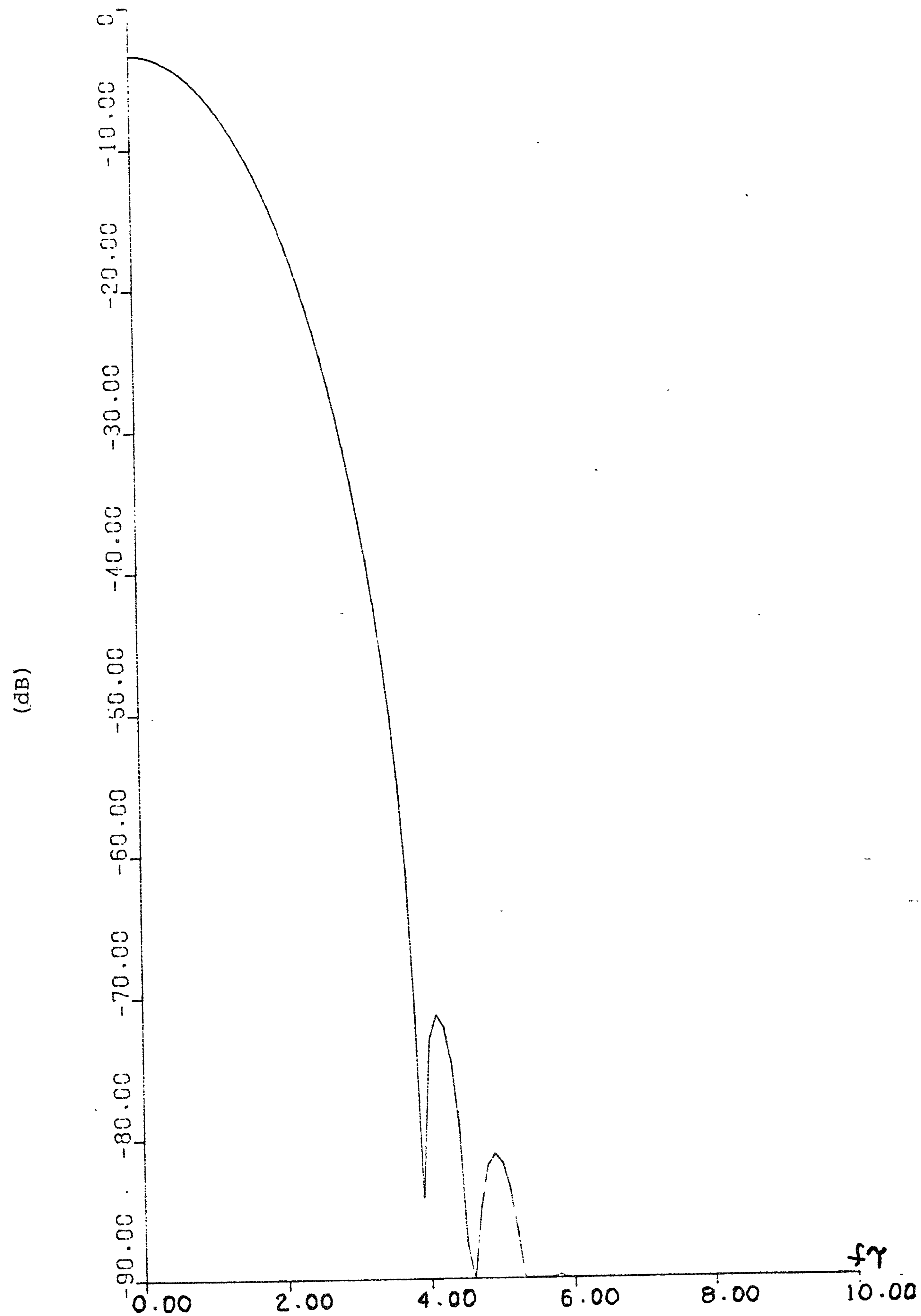


Figure 4.17. Power Spectrum of 4th Order Truncated Sinc Function

4.4 BT PRODUCT

Consider the signal

$$s(t) = \sum_{k=o}^{N-1} a_k \sqrt{E_s}\, h(t - kT)$$

in which each a_k numerically represents a channel symbol, and for $k \neq \ell$, a_k and a_ℓ are uncorrelated so that

$$E[a_k\, a_\ell] = \begin{cases} 0 & k \neq \ell \\ 1 & k = \ell \end{cases}$$

It can then be proved that the average power spectrum, defined as

$$\lim_{N\to\infty} \frac{1}{N} \left| S(f) \right|^2$$

is independent of the pulse repetition rate. Roughly speaking, this implies that the spectral occupancy of a given pulse shape is independent of the pulse repetition rate 1/T. Applying Parseval's theorem to the above result also shows that the average energy per repetition interval of s(t) is equal to E_s and independent of T.

The BT product measures the bandwidth efficiency of the signaling, in a sense that a smaller BT would use less bandwidth for a given T. Defining resolution θ as the ratio T/τ, the BT product is related to the $B\tau$ product by

$$BT = B\tau \cdot \theta$$

The BT product serves as a criterion for comparison amongst various pulse shapes. A few examples would convince us of its usefulness. The rectangular pulse shape has BT = 1. The enlarged spectral mainlobe of the half-cosine pulse shape for MSK is reflected by an increased BT of 1.235. For raised cosine pulse shape, BT = 1.5θ.

shape, BT = 1.5θ. Typically, θ = 1/2 giving BT = 0.75 which reflects the spectral improvements (at the cost of ISI and increased envelope fluctuation).

4.5 FILTER LOSS

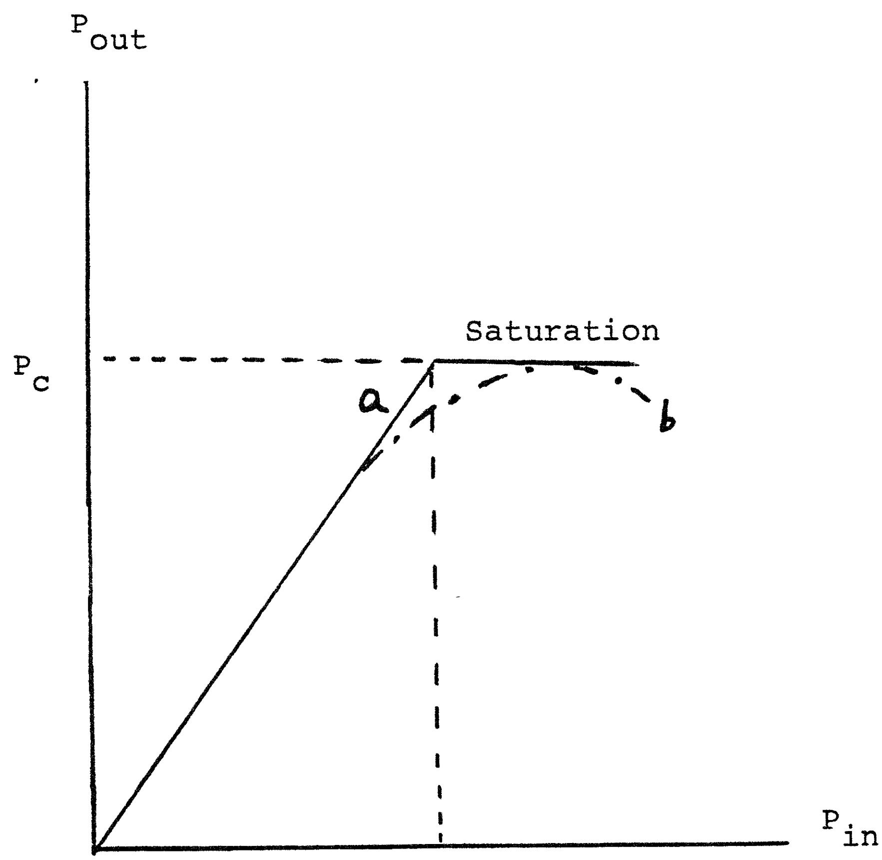
For a given h(t), a higher repetition rate increases ISI as well as introduces more levels to s(t). The higher rate of data transmission is achieved at the cost of more envelope fluctuation and a higher peak power requirement. In some satellite communication systems such as TDMA transmission, the peak power rather than the average power is limited. Assuming the nonlinear amplifier with a characteristics shown in curve a of Figure 4.18, which is linear up to the peak power required by s(t), then the filter loss is defined as the ratio of the maximum power of the signal that still ensures no nonlinear distortion to the power available from the amplifier. Expressed in dB, the filter loss is

$$f_{\ell} = -10 \log \frac{E_s}{P_c T}$$

In which E_s is related to the maximum rms power of the carrier P_c by

$$\max_t s(t) = \sqrt{P_c}$$

The remaining discussion illustrates how the dual concepts of BT product (which measures spectral efficiency) and filter loss (which measures nonconstancy of envelope) can be applied for pulse shapes considered in the previous section.



Curve a = An ideal clipping nonlinearity

Curve b = Typical nonlinearity of amplifier

Figure 4.18. Characteristics of Nonlinearity

The time domain pulse shapes of the beta functions, the trigonometric-hyperbolic functions and the truncated n-th power sinc functions become narrower as n increases. For large n, these functions would resemble impulses with unity energy. Since an impulse has a white spectrum, the main lobe of these functions should become wider as n increases. The spectra plotted for these functions confirm our speculation. For a given value of θ, an enlarged main lobe would require more spectral bandwidth for transmission.

On the other hand, the amount of ISI for given θ decreases as the functions become more impulse-like. This enables us to use a smaller resolution θ without introducing excessive filter loss.

A computer program which evaluates filter loss as a function of resolution yields the plots shown in Figure 4.19 - 4.23. In these plots, the upper curve represents the filter loss when the in-phase and quadrature channels are not staggered. The lower curve shows the filter loss when the two channels are offset by T/2. The minima of these curves shift to the left as n increases, allowing the use of smaller θ for a given level of filter loss. It is also observed that filter loss for the staggered case is always less than that for the unstaggered case. If this observation is true for all h(t), then a constant envelope pulse modulation with staggered quadratures, when filtered, would give less filter loss and envelope fluctuation than the unstaggered case. Less envelope fluctuation makes the modulation more compatible with a limiting nonlinearity.

Finally, the analyses in this chapter suggest the following procedures for designing pulse shapings. The proper pulse shaping is chosen by tailoring the spectrum according to the width of the main lobe and the rate of roll-off that we desire for the

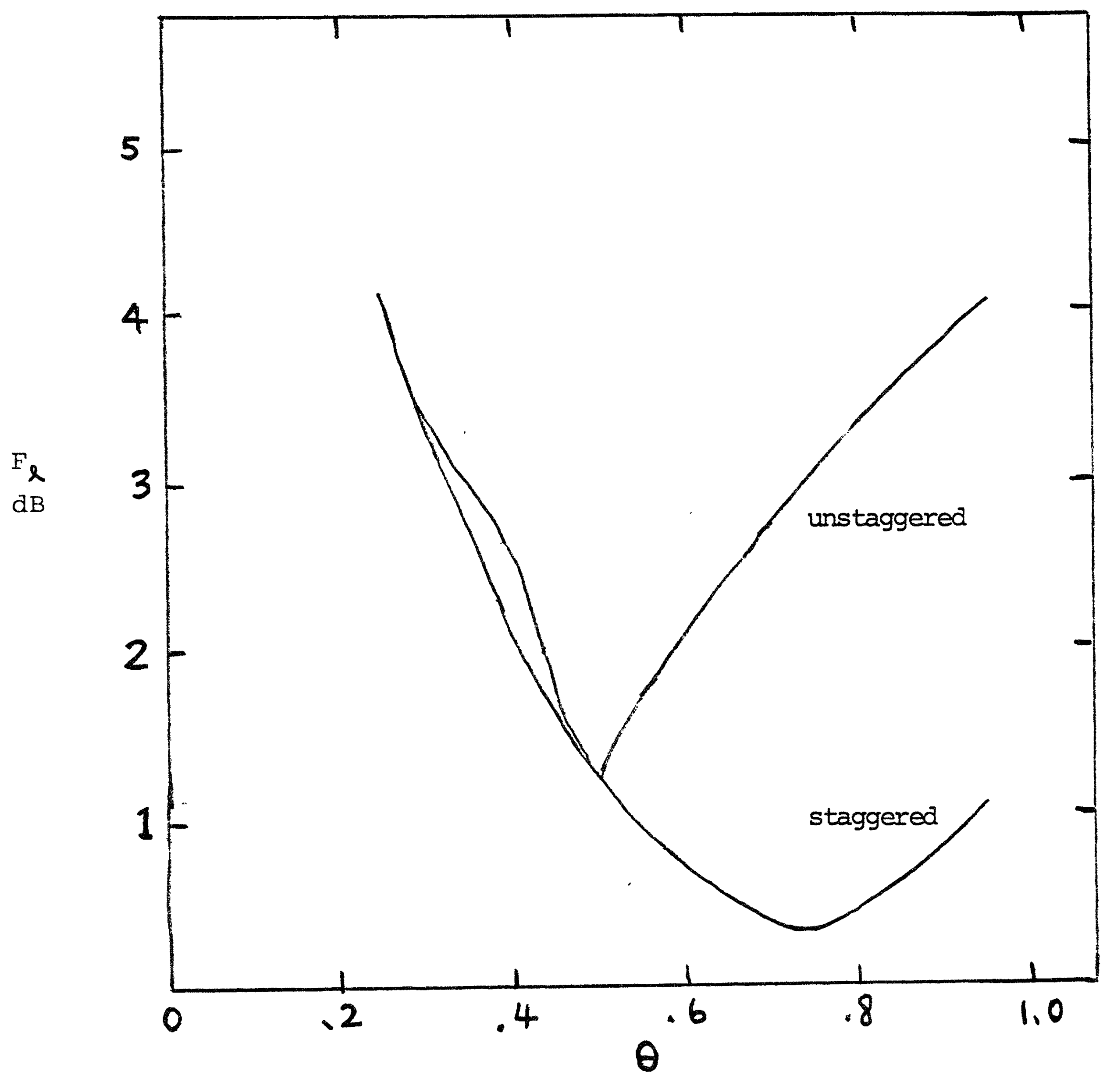


Figure 4.19. F_ℓ vs θ For Overlapped Raised-Cosine

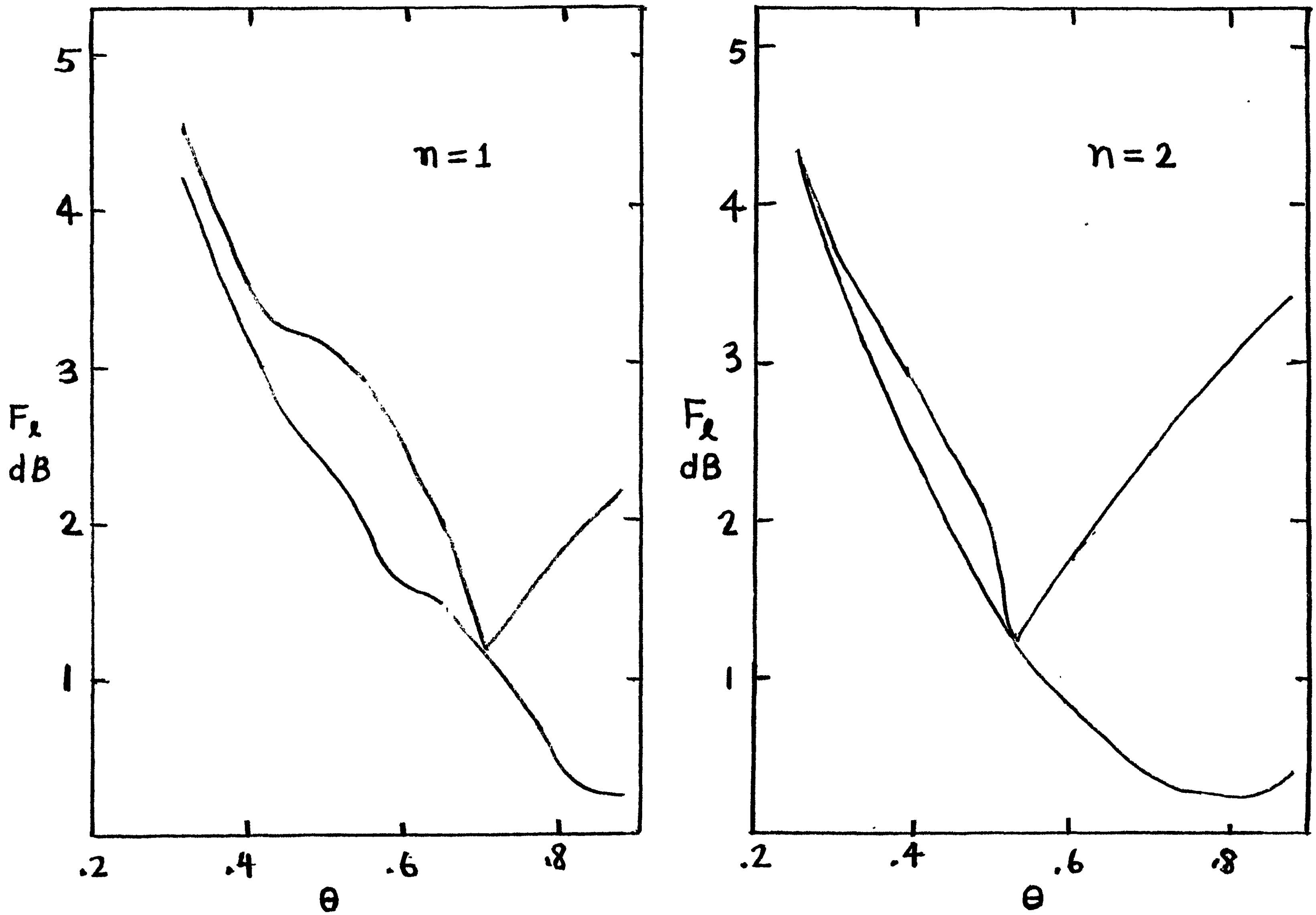


Figure 4.20. F_ℓ vs θ for the 1st and 2nd Order Beta Functions
(m = 1, n = 1, 2)

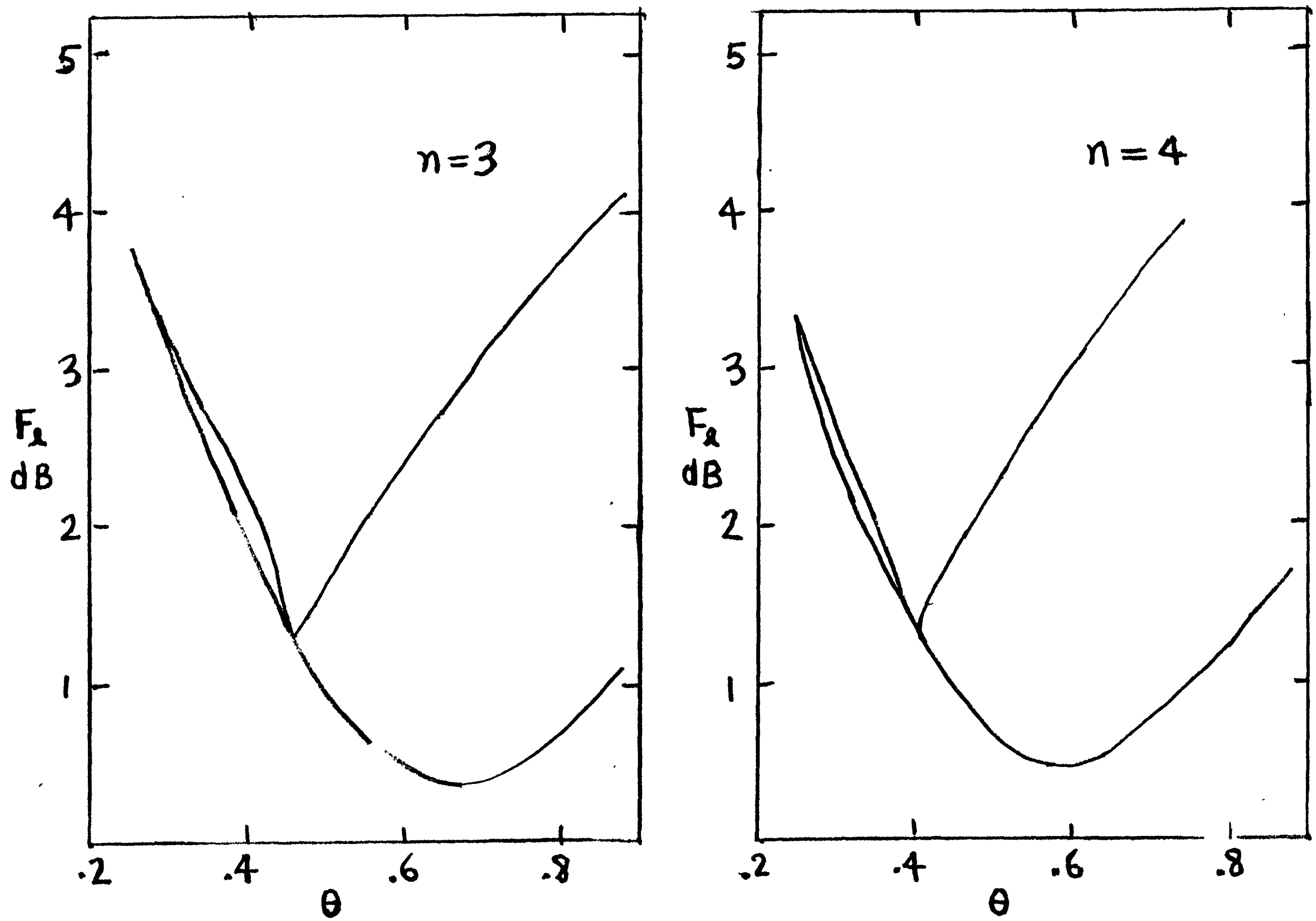


Figure 4.21. F_ℓ vs θ for the 3rd and 4th Order Beta Functions
(m = 1, n = 3, 4)

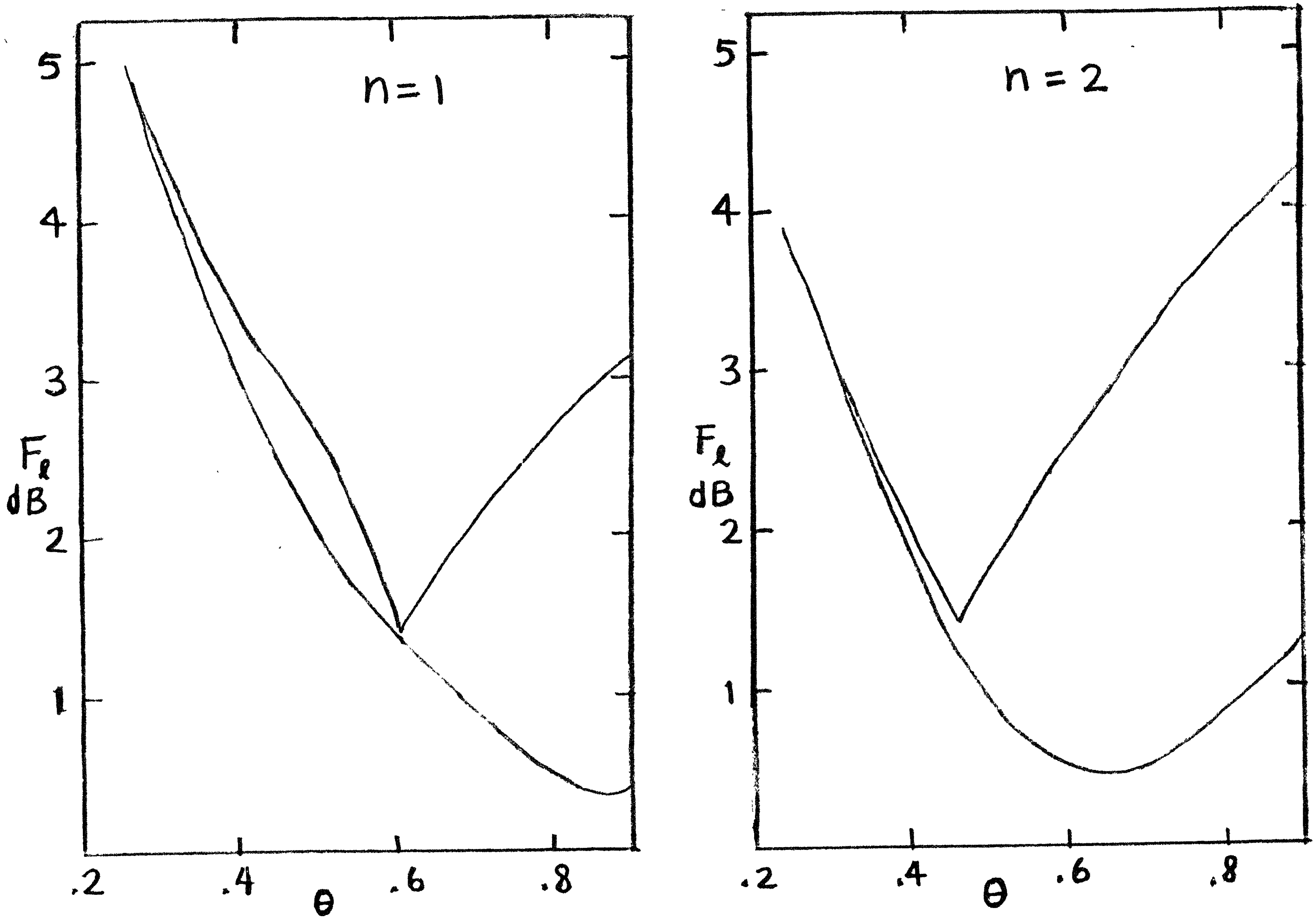


Figure 4.22. F_ℓ vs θ for the 1st and 2nd Order Truncated Sinc Functions

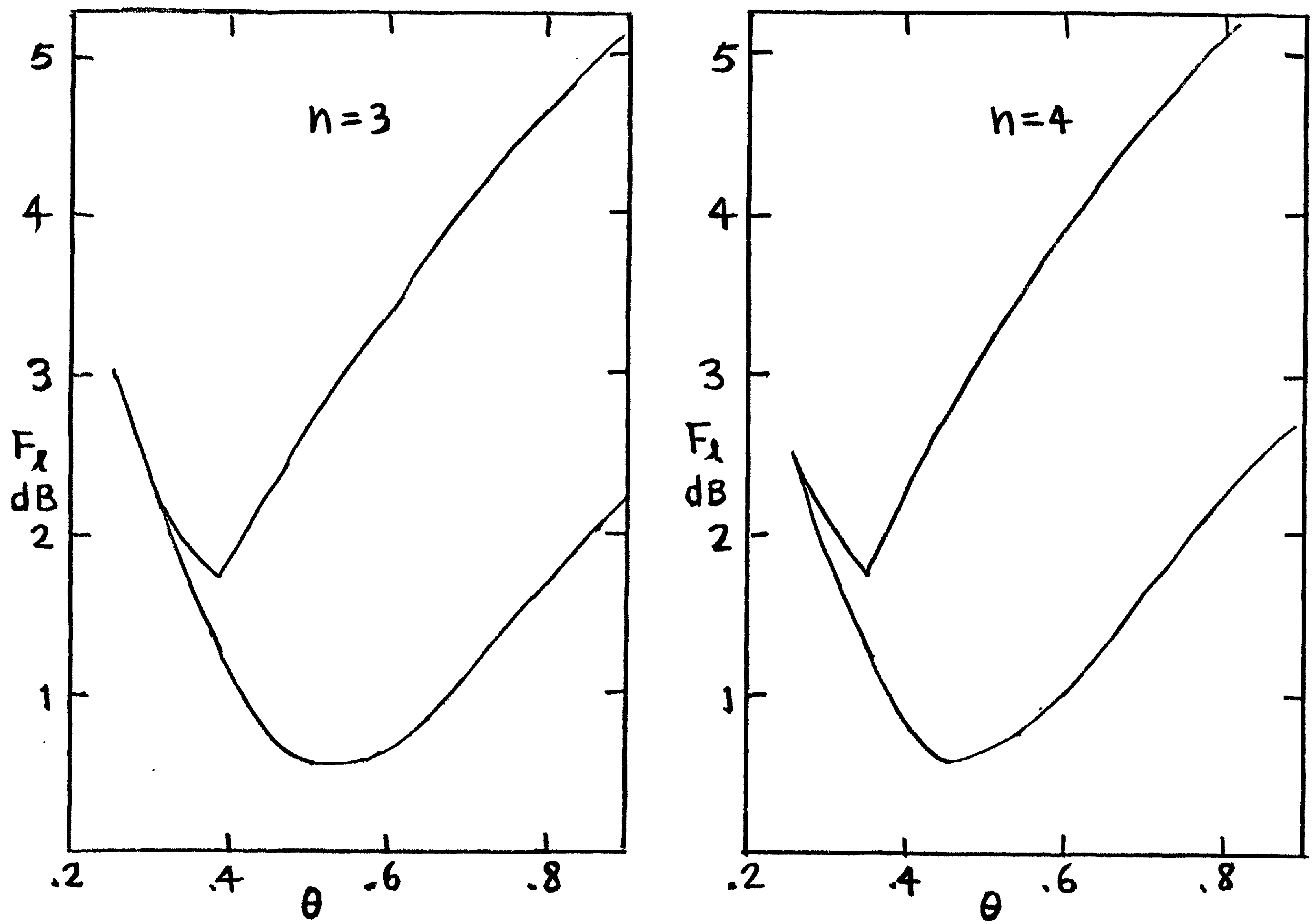


Figure 4.23. F_ℓ vs θ for the 3rd and 4th Order Truncated Sinc Functions

channel to reasonably avoid adjacent channel interference and out-of-band emission. The level of filter loss that can be tolerated by the transmission system is determined, which gives an idea of what θ should be used from the filter loss versus resolution plots. A comparison of various pulse shapes is given in Figure 4.24.

System performance with and without channel coding for these pulses which introduce ISI will be considered in Chapter 6.

Asymptotic Rate of Roll-Off of $\lvert H(\omega)\rvert^2$	Pulse Shape	β	$B\tau$	BT and (θ) For Staggered Components at F_ℓ (dB) = 0.5	1.0	1.5
$0(1/f^2)$	$Beta^0$	1.00	1.00	F_ℓ = 6 dB, BT = θ for $\frac{1}{2} < \theta < 1$		
$0(1/f^4)$	$Beta^1$	5.24	1.20	0.96 (0.80)	0.89 (0.74)	0.79 (0.66)
	TH^1	7.70	1.24			
	TS^1	-	1.25	0.97 (0.78)	0.85 (0.68)	0.73 (0.58)
$0(1/f^6)$	$Beta^2$	8.96	1.43	0.96 (0.67)	0.83 (0.58)	0.72 (0.5)
	TH^2	11.90	1.45			
	TS^2	-	1.75	1.05 (0.60)	0.86 (0.49)*	0.77 (0.44)*
	RC	-	1.5	0.93 (0.62)	0.81 (0.54)	0.70 (0.47)*
$0(1/f^8)$	$Beta^3$	12.40	1.63	0.96 (0.59)	0.80 (0.49)*	0.72 (0.44)*
	TS^3	-	1.92	1.00 (0.52)	0.81 (0.42)*	0.71 (0.37)*
$0(1/f^{10})$	$Beta^4$	15.72	1.81	0.98 (0.54)	0.80 (0.44)*	0.71 (0.39)*
	TS^4	-	2.17	1.00 (0.46)*	0.80 (0.37)*	0.72 (0.33)*

*h_2 is nonzero

$Beta^n$, TH^n, TS^n - nth order Beta, trigonometric-hyperbolic, and truncated sinc functions

RC - Raised-cosine function

Figure 4.24. A Comparison of Various Pulse Shapings

Chapter 5. CONVOLUTIONAL ENCODERS FOR MULTIPHASE MODULATION WITHOUT INTERSYMBOL INTERFERENCE

The spectrum of uncoded M-ary PSK for constant symbol rate (1/T) is independent of the number of phases employed by the symbol, but E_b/N_o required for a given bit error rate increases significantly as M increases. To ensure reliable transmission, message redundancy may be introduced, at the expense of reducing the bit data rate R_b, or by using an expanded set of phases with coding. The second approach of coded phase is particularly attractive for satellite communication since a coding gain with respect to uncoded 4ϕ-PSK of several dBs may be achieved without reducing R_b, with a spectrum similar to 4ϕ-PSK. Most of the known binary convolutional codes with good minimum free Hamming distance can be used for 2ϕ-PSK and 4ϕ-PSK, when the free Euclidean distance between signals is proportional to the free Hamming distance of the codes for the signals. Unfortunately, this proportionality is no longer preserved when more than 4 phases are used with coding.

This chapter explores several convolutional encoder structures that may be employed for coded phase, and investigates their free Euclidean distance properties. Specifically, coded 8-ϕ is addressed due to its attractiveness for implementation. Otherwise, most of the results may be extended with some efforts for the general M-ary case.

5.1 DEFINITIONS

The encoder inputs a sequence $\underline{u}$ and outputs a sequence $\underline{w}$, consisting of u_k's and w_k's respectively which are elements of the set U, over which the operators multiplication $\cdot$ and addition

$\oplus$ are well-defined. The encoder is characterized by the generator matrix G with entries which are also elements of U, so that functionally

$$\underline{w} = \underline{u}\, G$$

The encoder output sequence $\underline{w}$ is distinctively mapped, by the mapping M, onto the channel sequence $\underline{v}$ consisting of v_k's which are elements of the channel symbol set V so that $\underline{v} = M(\underline{w})$. The modulator F then distinctively maps each sequence $\underline{v}$ onto a time function f(t).

A channel encoding scheme S is therefore completely specified by the triple (G, M, F), namely, the discrete encoder G with its associated U and operators, the modulation F which generates the physical waveform and the channel symbol mapper M which links G and F.

For AWGN, knowing the Euclidean distances between waveforms is sufficient for performance evaluation. The square Euclidean distance D between the channel sequences, $\underline{v}^1$ and $\underline{v}^2$ is given by

$$D[\underline{v}^1, \underline{v}^2] = \left\| f(t, \underline{v}^1) - f(t, \underline{v}^2) \right\|^2$$

$$= \int_{-\infty}^{\infty} [f(t, \underline{v}^1) - f(t, \underline{v}^2)][f(t, \underline{v}^1) - f(t, \underline{v}^2)]^* \, dt$$

The minimum free square Euclidean distance d_f for S is defined to be the minimum D between all distinct input sequences. The encoder G is said to be optimal for given M and F if d_f is maximized for the class of encoders of equivalent complexity (described by such as rate and number of memories of the encoder). In this thesis, we are mostly concerned with finding the optimal G in this sense, for the various M and F proposed. An equally interesting problem, but much less understood, is to find the M which would result in good d_f for given F and specified algebraic structures of G. Take for example, if binary convolutional encoder is used for 8ϕ-PSK modulation in the absence of ISI, we would be interested to see whether Gray mapping, or straight binary mapping, or any other mapping which maps a binary output sequence into an octal sequence is the best.

Furthermore, S is said to be invariant if and only if for all input sequences $\underline{a}$, $\underline{b}$ and $\underline{c}$

$$D[M((\underline{a} \oplus \underline{c})G), M((\underline{b} \oplus \underline{c})G)]$$
$$= D[M(\underline{a}\ G), M(\underline{b}\ G)]$$

If (U, $\oplus$, $\cdot$) is a ring, then invariance implies

$$D[M(\underline{a}\ G), M(\underline{b}\ G)]$$
$$= D[M(\underline{O}\ G)\ , M((-\underline{a} \oplus \underline{b})\ G)]$$

in which $-\underline{a}$ consists of additive inverses for the elements of $\underline{a}$.

For an invariant S, it follows that

$$d_f = \min_{\underline{b}} D[M(\underline{a}\ G), M(\underline{b}\ G)]$$

for any $\underline{a}$, making the evaluation for d_f much simpler. Code searching thus becomes much easier when the distance between two codewords depends only upon the difference sequence $(-\underline{a} \oplus \underline{b})$ between the two input sequences.

Two schemes S_1 and S_2 are said to be equivalent, denoted by

$$S_1 = (G_1, M_1, F_1) = (G_2, M_2, F_2) = S_2$$

if and only if they generate the same set of waveforms for all possible input sequences.

Using the terms defined, three channel encoding schemes for F being 8ϕ-PSK without ISI will be described in the following sections.

5.2 BINARY ENCODERS WITH STRAIGHT BINARY MAPPING

The first scheme referred to as the binary encoding scheme, is defined as follows,

$$U = \{0, 1\}$$

with multiplication and addition defined by

$\oplus$	0	1		$\cdot$	0	1
0	0	1		0	0	0
1	1	0		1	0	1

The channel symbol set is given by

$$V = \{0, 1, 2, \ldots, 7\}$$

and G is a binary rate 2/3 convolutional encoder.

M maps each triple output (A, B, C) of the rate 2/3 convolutional encoder G into an octal v using a straight binary conversion, so that

$$v = 4A + 2B + C$$

F uses the octal sequence $\underline{v}$ for 8ϕ-PSK without ISI.

An alternative M' which maps the triple output (A', B', C') of another binary convolutional encoder G' into an octal v' using Gray mapping can be specified as

$$v' = 4a + 2b + c$$

in which

$$a = A'$$

$$b = A' \oplus B'$$

$$c = A' \oplus B' \oplus C'$$

Obviously, (G, M, F) and (G', M', F) would be equivalent if for the same input sequence $\underline{u}$, the output octal sequences $\underline{v}$ and $\underline{v}'$ are equal, which implies

$$a = A$$

$$b = B$$

$$c = C$$

and consequently

$$A = A'$$

$$B = A' \oplus B'$$

$$C = A' \oplus B' \oplus C'$$

The transformation given by these equations enables us to convert each G into an equivalent G' and vice versa. Therefore, the discussion on these binary convolutional encoder will assume the use of straight binary mapping for the remainder of the chapter.

It is noteworthy that this description of S is similar to the formulation of rate 2/3 coded 8ϕ-PSK by Ungerboeck [10]. However, the method of bounding d_f is quite different and the bound we are going to get is tighter than Ungerboeck's. As a consequence, we are able to find codes with better d_f.

The S defined is unfortunately not invariant. Consider

$$\underline{w} = \underline{u}\, G$$

$$\underline{\varepsilon} = \underline{e}\, G$$

$$\text{hence } \underline{w} \oplus \underline{\varepsilon} = (\underline{u} \oplus \underline{e})\, G$$

Let $w_k = (w_{k,1}, w_{k,2}, w_{k,3})$ and $\varepsilon_k = (\varepsilon_{k,1}, \varepsilon_{k,2}, \varepsilon_{k,3})$ be groups of three bits that are mapped into an octal, that is

$$M(w_k) = 4w_{k,1} + 2w_{k,2} + w_{k,3}$$

Defining

$$d_i = D[i, 0] \qquad i = 0, 1, 2, 3, 4$$

The square Euclidean distances

$$D[M(w_k \oplus \varepsilon_k) \quad , \quad M(w_k)]$$

are tabulated in Figure 5.1, which shows that these distances are equal to

$$D[M(\varepsilon_k) \quad , \quad 0]$$

regardless of the values of w_k if $M(\varepsilon_k) = 0, 1, 2, 4, 5, 6$; and for $M(\varepsilon_k) = 3$ or 7, may be equal to d_1 for some w_k.

A lower bound for the Euclidean distance between two channel sequences can be stated as follows:

For any

$$\underline{v}^1 = M(\underline{w}^1)$$

$$\underline{v}^2 = M(\underline{w}^2)$$

we have

$$D[\underline{v}^1, \underline{v}^2] \geq D[M_b(\underline{\varepsilon}), \underline{0}]$$

in which the error sequence

$$\underline{\varepsilon} = \underline{w}^1 \oplus \underline{w}^2$$

and M_b is the mapping that replaces occurrences of 3's in ε_k into 1's. In other words

$w_k \oplus \varepsilon_k$ D	ε_k $(M(\varepsilon_k))$							
	000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)
w_k $(M(w_k))$								
000 (0)	000 d_0	001 d_1	010 d_2	011 d_3	100 d_4	101 d_3	110 d_2	111 d_1
001 (1)	001 d_0	000 d_1	011 d_2	010 d_1	101 d_4	100 d_3	111 d_2	110 d_3
010 (2)	010 d_0	011 d_1	000 d_2	001 d_1	110 d_4	111 d_3	100 d_2	101 d_3
011 (3)	011 d_0	010 d_1	001 d_2	000 d_3	111 d_4	110 d_3	101 d_2	100 d_1
100 (4)	100 d_0	101 d_1	110 d_2	111 d_3	000 d_4	001 d_3	010 d_2	011 d_1
101 (5)	101 d_0	100 d_1	111 d_2	110 d_1	001 d_4	000 d_3	011 d_2	010 d_3
110 (6)	110 d_0	111 d_1	100 d_2	101 d_1	010 d_4	011 d_3	000 d_2	001 d_3
111 (7)	111 d_0	110 d_1	101 d_2	100 d_3	011 d_4	010 d_3	001 d_2	000 d_1

Figure 5.1. Effect of w_k on $D[M(w_k \oplus \varepsilon_k), M(\varepsilon_k)]$

$$M_b(\varepsilon_k) = \begin{cases} M(\varepsilon_k) & \text{if } M(\varepsilon_k) \neq 3 \\ 1 & \text{if } M(\varepsilon_k) = 3 \end{cases}$$

A lower bound for d_f follows immediately

$$d_f \geq \min_{\underline{\varepsilon}} D[M_b(\underline{\varepsilon}), \underline{0}] = d_b$$

in which $\underline{\varepsilon}$ is any encoder output sequence.

This bound enables us to search for good codes using the Viterbi algorithm which compute the value of d_b at each stage of the algorithm. Furthermore, the encoder with maximum d_b is optimal since it can be shown that the lower bound is tight, which is to say,

Theorem

For every $\underline{\varepsilon}$, there exist a

$$\underline{w} = \underline{u}\, G$$

such that

$$D[M(\underline{w}), M(\underline{w} \oplus \underline{\varepsilon})] = D[M_b(\underline{\varepsilon}), \underline{0}]$$

Proof

Forney [16] has shown that every convolutional encoder is equivalent to a feedback systematic encoder with structure shown in Figure 5.2, in the sense that both generate the same set of codewords. Therefore, it suffices to prove the theorem if a $\underline{w}$ can be generated by an equivalent convolutional encoder so that the bound can be satisfied with equality.

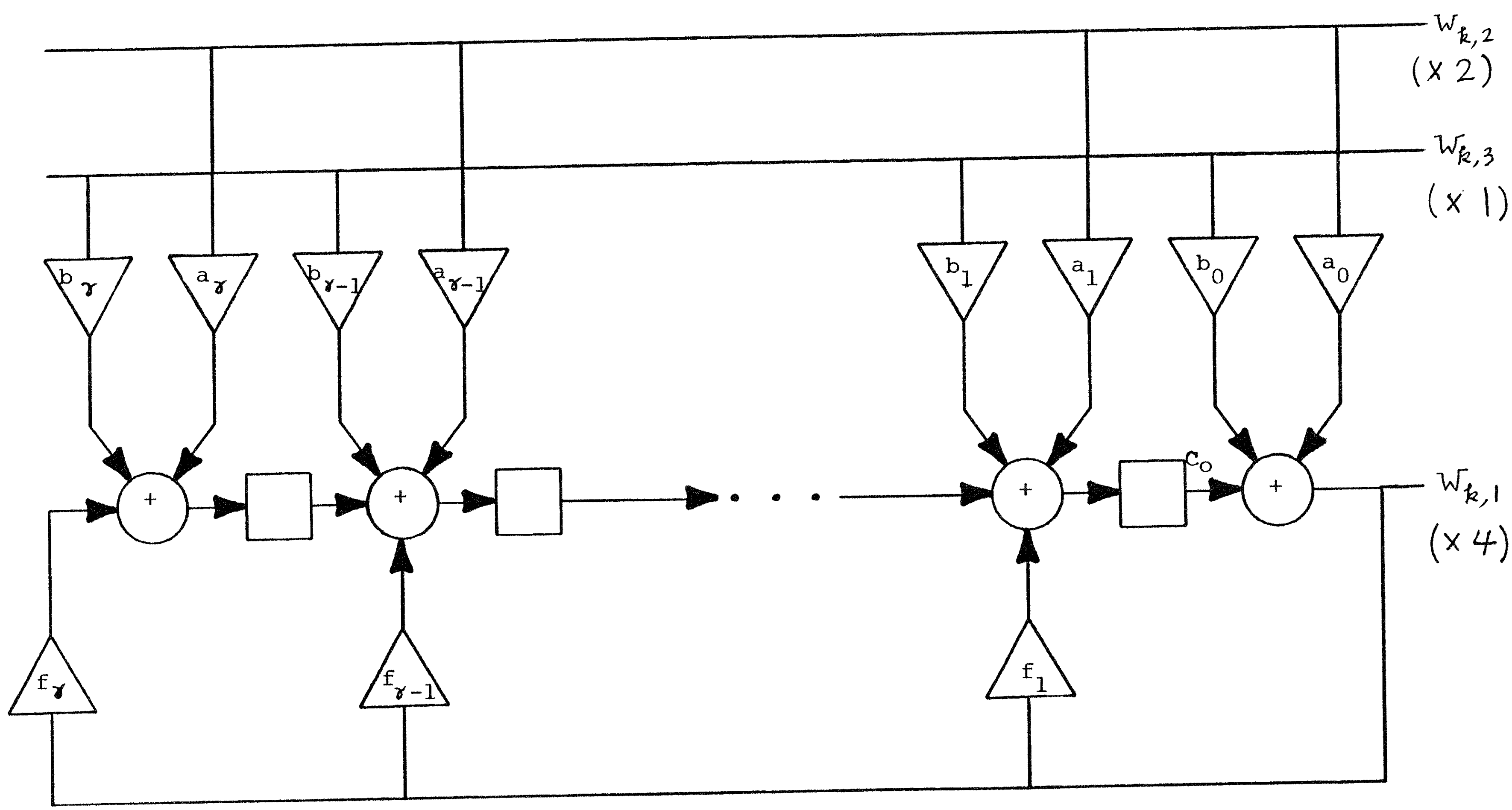


Figure 5.2. A Feedback Convolutional Encoder

The outputs $w_{k,2}$, $w_{k,3}$ in Figure 5.2 are basically unconstrained. From Figure 5.1, it is seen that for every ε_k, we can choose $w_{k,2}$ and $w_{k,3}$ in such a way that

$$D[M(w_k), M(w_k \oplus \varepsilon_k)] = D[M_b(\varepsilon_k), 0]$$

Q.E.D.

A computer program was written to perform the code search. The search algorithm and program notes are described in Appendix B. The $\gamma = 4$ and $\gamma = 6$ encoders found are shown in Figures 7.2 and 7.3.

5.3 OCTAL ENCODERS WITH IDENTITY MAPPING

In this section, an invariant encoder is introduced with

$$U = V = \{0, 1, 2, \ldots, 7\}$$

over which addition is defined by

$$a \oplus b = (a + b) \text{ modulo } 8$$

and multiplication is defined by

$$a \cdot b = (a \times b) \text{ modulo } 8$$

The addition and multiplication tables are given in Figure 5.3.

M is the identity mapping

$$M(w_k) = w_k$$

For the sake of convenience, the notation M will be harmlessly left out in the following discussion.

It should be noted that, unlike the binary encoder in the previous section, the $(U, \oplus, \cdot)$ defined is only a ring and not a field, and hence most of the results concerning canonical encoders in [16] do not apply.

⊕	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

Figure 5.3 Addition and Multiplication Tables for Octal Convolutional Encoder

Theorem

This encoder is invariant for 8ϕ-PSK in the absence of ISI.

Proof

For

$$\underline{v} = \underline{u}\, G$$

$$\underline{\varepsilon} = \underline{e}\, G$$

it follows that

$$(\underline{u} \oplus \underline{e})\, G = (\underline{u}\, G) \oplus (\underline{e}\, G)$$

$$= \underline{v} \oplus \underline{\varepsilon}$$

Since for F being 8ϕ-PSK without ISI

$$D[\underline{v} \oplus \underline{\varepsilon}, \underline{v}] = D[\underline{\varepsilon}, \underline{0}]$$

the invariance of this scheme follows immediately. Q.E.D.

The Viterbi algorithm can be applied in a straight forward manner to search for optimum octal rate 2/3 convolutional encoders. Such encoder with γ octal memories will have 8^{γ} states. Furthermore, a rate p/q encoder (p and q relatively prime) will have 8^{p} branches going into each state. While $8^{p}-1$ comparisons have to be made at each state, 3p information bits are being decoded at each stage of the Viterbi algorithm. There are altogether $(\gamma + p)q$ multiplicative taps in the encoder, each can take on 8 possible values. The large number of possible tap combinations makes exhaustive code searching computationally very consuming for $\gamma > 3$.

To anticipate the performance of these encoders, an upper bound on the minimum free distance achievable will occupy our attention for the rest of this section.

The following is basically a modified Plotkin bound [19] for octal encoders.

Theorem

Let $\underline{m}$ be an input octal sequence and $\underline{v}^m = \underline{m}G$ consisting of octals v_n^m be the corresponding output octal sequence. It follows then, the set of v_n^m for all m and a given n either has

1. v_n^m all zeros or
2. an equal number of $v_n^m = 0$ and $v_n^m = 4$, and also an equal number of $v_n^m = 1$, $v_n^m = 3$, $v_n^m = 5$ and $v_n^m = 7$

Proof

Let the n-th column of G be g_n. Consequently,

$$v_n^m = \underline{m}\, g_n$$

Now either

a) $v_n^m = 0$ for all $\underline{m}$

or b) There exists an m' with $v_n^{m'} = 4$.

Then for every $\underline{m}$ which gives $v_n^m = 0$, we have an

$$\underline{m}^* = \underline{m} \oplus \underline{m}'$$

such that

$$v_n^{m*} = (\underline{m} \oplus \underline{m}')\, g_n$$
$$= (\underline{m}\, g_n) \oplus (\underline{m}'\, g_n)$$
$$= 0 \oplus 4$$
$$= 4$$

since the mapping of $\underline{m}'$ onto $\underline{m}*$ is a one to one onto mapping and furthermore, only those $\underline{m}$ s with $v_n^m = 0$ are mapped onto $\underline{m}*$ s with $v_n^{m*} = 4$. There is therefore an equal number of $v_n^m = 0$ and $v_n^m = 4$.

or c) There is no $\underline{m}$ such that $v_n^m = 1, 3, 5,$ or 7 in which case the number of $v_n^m = 1, 3, 5,$ and 7 are all zero and hence equal.

or d) There exists an $\underline{m}'$ with $v_n^{m'} = 1$ (3 or 5 or 7). It can be readily shown that the set of codewords partitioned according to their value of v_n^m gives 8 equal-sized cosets. Hence, the claim in 2 is true. Q.E.D.

If F is such that the signal energies are normalized so that

$$D(0, 4) = 2$$

then from Figure 3.2, the square distances are

$$D(0, 1) = D(0, 7) = 1 - \sqrt{2}/2$$

$$D(0, 2) = D(0, 6) = 1$$

$$D(0, 3) = D(0, 5) = 1 + \sqrt{2}/2$$

It follows immediately from the above theorem that the average weight of v_n^m within the n-th column is 1.

The upper bound for the minimum distance of octal convolution encoders can be stated as, Theorem

For a rate p/q octal convolutional encoder (p and q relatively prime) with p queues (each consisting of K memory elements) and q modulo 8 adders, the minimum free square Euclidean distance is upper bounded by

$$d_f \leq \min_{L} \frac{q(L + K)\ 8^{pL}}{8^{pL}-1}$$

Proof

There are 8^{pL} information octal sequence of length pL. Each corresponding code sequence is of length q(L + K). The total weight of all 8^{pL} codewords is $q(L + K)8^{pL}$ since each octal in a codeword has an average weight of 1 within a not-all-zero column. The average weight of a nonzero code word is therefore

$$\frac{q(L + K)8^{pL}}{8^{pL}-1}$$

The minimum free distance must be less than the average distance for all L. Therefore, by minimizing over L, we obtained an upper bound for d_f.

Q.E.D

The minimum is observed to occur always at L = 1. A fair approximation to this upper bound is q(K + 1).

For rate 1/2 octal convolutional encoders, we have

K	0	1	2	3	4	5
Upper bound	2.29	4.57	6.86	9.14	11.43	13.71

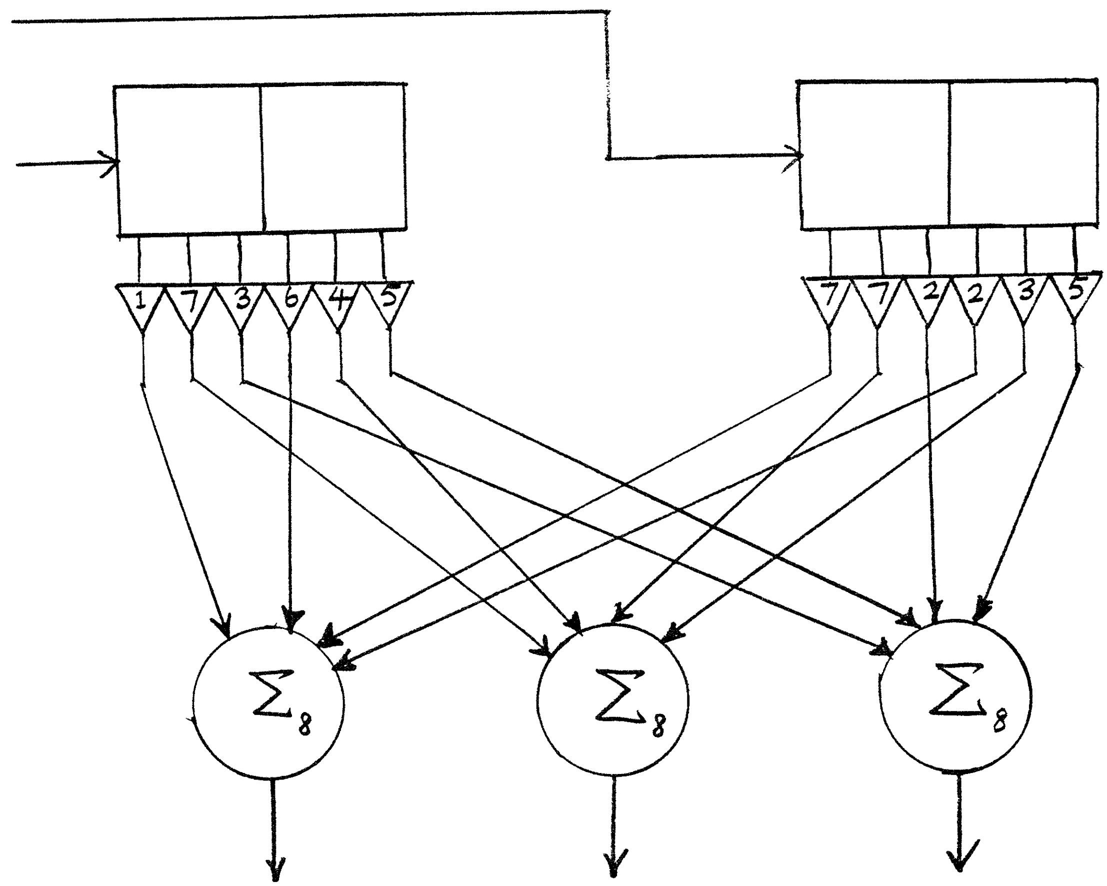
For rate 2/3 octal convolutional encoders, we have

K	0	1	2	3	4	5
Upper bound	3.05	6.09	9.14	12.19	15.24	18.29

Furthermore, if the p queues are of unequal length, then the upper bound is equal to the case of K for which K is the number of octal memory elements in the shortest queue. From the computer search for optimal code, the bound for rate 1/2 encoders is fairly tight for K = 0, 1, 2. The upper bound for the rate 2/3 encoders seems to suggest a far superior performance than the binary encoders which have an equal number of states. Unfortunately, the code search turned out codes which achieve a d_f much less than the upper bound. A non-exhaustive code search for the 64 states (K = 1) octal encoder gave a code shown in Figure 5.4, with $d_f = 3.172$ which equals the d_f for the best binary encoder with 6 binary memory found in the previous section. The algorithms for searching octal encoders of rates 1/2 and 2/3 are discussed in Appendix B. Some rate 1/2 encoders found are listed in Figure 5.5.

5.4 GF(8) ENCODERS

The modulo 8 encoder has a multiplication table which is not homogeneous, in a sense that certain elements of U occur more frequently and in a structured manner in the table. Furthermore, (U, $\oplus$, $\cdot$) does not form a field since not every nonzero element of U has a multiplicative inverse. Based on these observations, we suggested a class of encoders for which (U, $\oplus$, $\cdot$) is a Galois field with 8 elements. The addition and multiplication tables shown in Figure 5.6 are generated as follows. Each element of U can be represented either as a binary triple

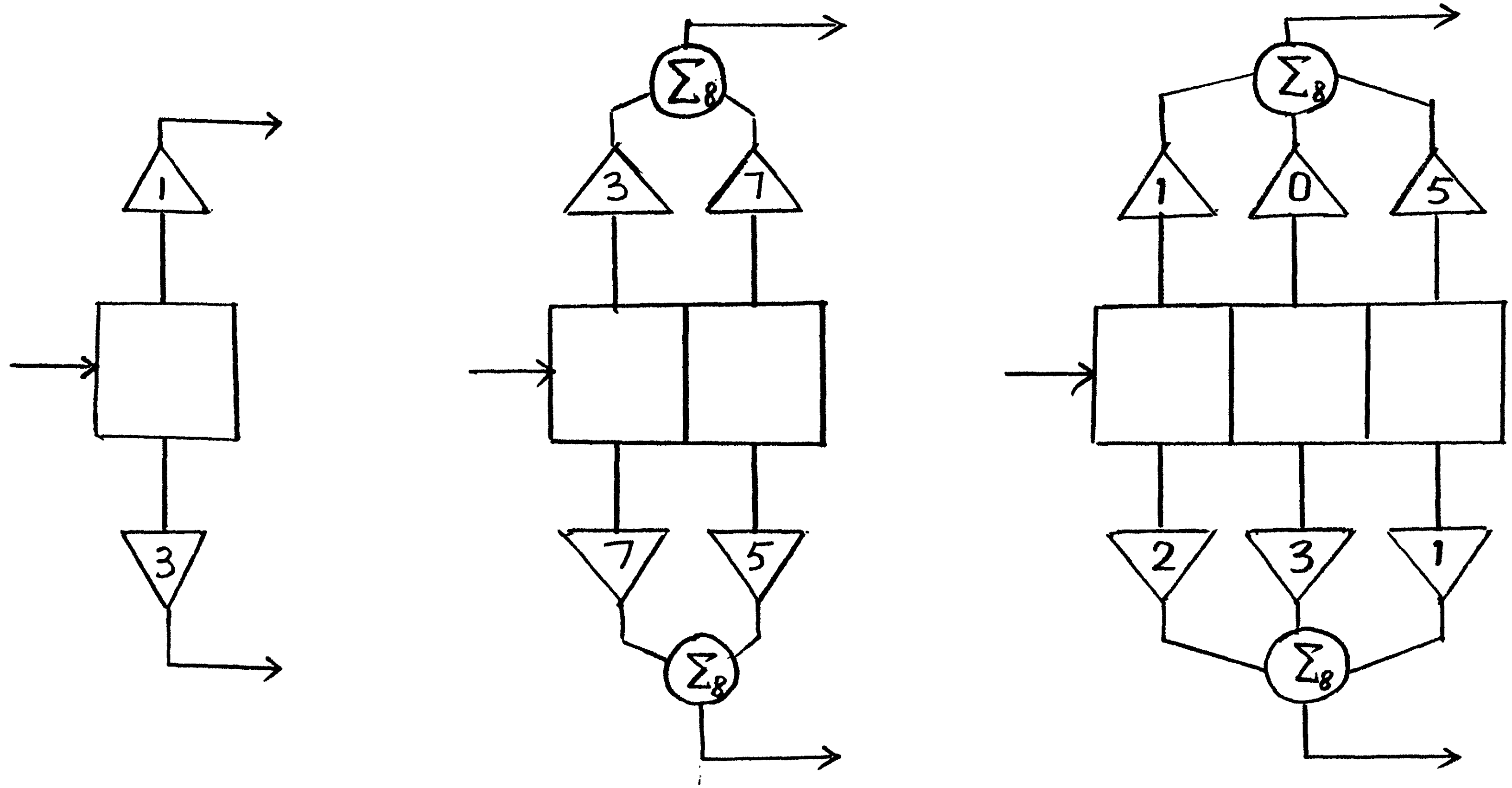


Subgeneratorss: 1 6 7 2

1	6	7	2
7	4	7	3
3	5	2	5

$d_f = 3.172$ Asymptotic Coding Gain = 5.0 dB

Figure 5.4. A Rate 2/3 Coded 8φ Octal Convolutional Encoder with 2 Octal Memories



$\gamma = 0$ $\quad d_f = 2.0$

$\gamma = 1$ $\quad d_f = 4.0$

$\gamma = 2$ $\quad d_f = 5.0$ *

*search incomplete

Figure 5.5. Rate 1/2 Coded 8ϕ Octal Convolutional Encoders with 0, 1, and 2 Octal Memories

⊕	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	3	1	7	5
3	0	3	6	5	7	4	1	2
4	0	4	3	7	6	2	5	1
5	0	5	1	4	2	7	3	6
6	0	6	7	1	5	3	2	4
7	0	7	5	2	1	6	4	3

Figure 5.6 Addition and Multiplication Tables for GF(8) Encoders

$$v = (a, b, c) \qquad a, b, c \in GF(2)$$

or as a polynomial

$$a\, t^2 + b\, t + c$$

The sum of two elements is expressed by the sum of their respective polynomials, or equivalently, the exclusive OR of the elements in binary representation. Their product is given by the product of their respective polynomials modulo

$$t^3 + t + 1$$

which itself cannot be factorized.

Similarly, M is the identity map and F is 8ϕ-PSK without ISI.

For all possible octal input sequences $\underline{m}$, we have either

1. $v_n^m = 0$ or
2. The numbers of $\underline{m}$ such that

$$v_n^m = i$$

are equal for all $i \in U$

<u>Proof</u>

If there is an $\underline{m}$ such that

$$v_n^m = i$$

is nonzero, then the sequence

$$\underline{v}^{m^j} = j(i)^{-1}\, \underline{v}^m$$

in which

$$i(i)^{-1} = 1$$

has

$$v_n^{m^j} = j$$

Furthermore, $\underline{v}^{m^j}$ is a codeword and is being generated by

$$\underline{m}^j = j(i)^{-1}\ \underline{m}$$

Therefore, there exist a $\overline{1}$ to 1 correspondence between every codeword with

$$v_n^m = i$$

and

$$v_n^m = j$$

from which statement 2 follows

Q.E.D.

The (G, M, F) thus defined is not an invariant system. In fact, it satisfies the same lower bound, namely

$$d_f \geqq \min_{\varepsilon} D[M_b(\underline{\varepsilon}),\ \underline{0}]$$

for the free Euclidean distance between two codewords as in the case of binary encoding, since both use exclusive OR as the operator for addition. We suspect that the bound is also tight, perhaps through a similar but more elaborate argument than that used for the binary encoder. If the bound is indeed tight, the average value of

$$D[M_b(v_n^m),\ 0]$$

in which M_b substitutes every

$$v_n^m = 3$$

with

$$v_n^m = 1$$

is 1 - $\sqrt{2}/8$ or 0.823. The minimum free Euclidean distance is therefore upper bounded by the same bound as for modulo 8 encoder times 0.823.

A computer program was written for searching rate 2/3 GF(8) encoders and is shown in Appendix B. Exhaustive search for encoders with just 1 or 2 octal memories is virtually impossible. The d_b resulting from our nonexhaustive search was rather disappointing, and further investigation into the GF(8) encoder was suspended.

Chapter 6. CONVOLUTIONAL ENCODER DESIGNS FOR BASEBAND PULSES WITH CONTROLLED INTERSYMBOL INTERFERENCE

The concern of this chapter is to design good, and if possible optimal, binary convolutional encoders G for modulation F being 2ϕ-PSK or 4ϕ-PSK with controlled ISI. Previously, Viterbi [13] has shown, using a code ensemble performance argument, that for duo-binary antipodal signaling, performance loss is less than 1 dB relative to the case of signaling without ISI. Using the notation developed in Chapter 3, duo-binary signaling corresponds to having

$$h_0 = 1 \quad , \quad h_1 = \frac{1}{2}$$

and

$$h_i = 0 \text{ for } |i| \geq 2$$

This result is complemented in this chapter by calculating the asymptotic performance for specific codes by finding or lower-bounding the minimum free Euclidean distance achieved by the code.

It should be noted that we were unable to extend the results in this chapter to find bounds for d_f for 8ϕ-PSK with ISI.

6.1 BOUNDS ON d_f IN THE PRESENCE OF ISI

The following discussion is based on using coded 2ϕ-PSK (or equivalently coded 2-PAM). The results will be extended to 4ϕ-PSK, which can be generalized as 2 orthogonal 2-PAM streams in the absence of channel crosstalk.

The signaling scheme is defined as follows:

G: A binary input binary output rate p/q convolutional encoder.

M: The set $U = \{0, 1\}$ is mapped onto the set $V = \{1, -1\}$ given by

$$0 \to 1, \; 1 \to -1$$

It is noteworthy that

$$M(u_1 \oplus u_2) = M(u_1)\, M(u_2)$$

F: The channel symbol sequence $\underline{v}$ generates the waveform

$$y(t) = \sum_k h(t - kT)\, v_k$$

The factor $\sqrt{E_s}$ has been left out in the expression since it is immaterial to our discussion.

The square Euclidean distance between the channel symbol sequences $\underline{v}$ and $\underline{v}'$ is

$$D[\underline{v}, \underline{v}'] = \int_{-\infty}^{\infty} \left[\sum_k h(t - kT)(v_k - v_k')\right]^2 dt$$

$$= \int_{-\infty}^{\infty} \sum_{k,\ell} h(t - kT)\, h(t - \ell T)(v_k - v_k')(v_\ell - v_\ell')\, dt$$

$$= \sum_{k,\ell} h_{k-\ell}\, (v_k - v_k')(v_\ell - v_\ell')$$

$$= h_0 \sum_k (v_k - v_k')^2$$

$$+ 2 \sum_{i=1}^{s} h_i \sum_k (v_k - v_k')(v_{k-i} - v_{k-i}')$$

in which

$$h_{k-\ell} = h_{\ell-k} = \int_{-\infty}^{\infty} h(t - kT)\, h(t - \ell T)\, dt$$

and the last expression is obtained by rearranging the order of summation and assuming

$$h_i = 0 \qquad \text{for } |i| > s$$

Our objective now is to express $D[\underline{v}, \underline{v}']$ explicitly in terms of the sequence $\underline{v}$ and the difference sequence between $\underline{v}$ and $\underline{v}'$. Recall that exclusive -OR in U is equivalent to multiplication in V, we may define the difference sequence $\underline{\varepsilon}$ consisting of ε_k and the delta sequence $\underline{\delta}(i)$ consisting of $\delta_k(i)$ by

$$v_k' = \varepsilon_k\, v_k$$

and

$$\delta_k(i) = v_k\, v_{k-i}$$

so that we have

$$\begin{aligned}(v_k - v_k')(v_{k-i} - v_{k-i}') &= (v_k - \varepsilon_k v_k)(v_{k-i} - \varepsilon_{k-i}\, v_{k-i}) \\ &= v_k v_{k-i}(1 - \varepsilon_k)(1 - \varepsilon_{k-i}) \\ &= \delta_k(i)(1 - \varepsilon_k)(1 - \varepsilon_{k-i})\end{aligned}$$

Since $\delta_k(i)$, ε_k and ε_{k-i} can take on the values ±1 only, we have

$$(1 - \varepsilon_k)(1 - \varepsilon_{k-i}) \geqq (v_k - v_k')(v_{k-i} - v_{k-i}') \geqq -(1 - \varepsilon_k)(1 - \varepsilon_{k-i})$$

By appropriately choosing the upper bound or lower bound with respect to the sign of h_i, we obtain the following upper and lower bounds for D.

$$h_0(1 - \varepsilon_k)^2 + 2 \sum_{i=1}^{s} |h_i| \sum_k (1 - \varepsilon_k)(1 - \varepsilon_{k-i})$$

$$\geq D\ [\underline{v}, \underline{v}']$$

$$\geq h_0(1 - \varepsilon_k)^2 - 2 \sum_{i=1}^{s} |h_i| \sum_k (1 - \varepsilon_k)(1 - \varepsilon_{k-i})$$

Note in particular that in the construction of the bounds, no reference is made to G, and the conclusion drawn should be treated rather as a property of the M and F used.

The lower bound for D, which depends on a single sequence $\underline{\varepsilon}$ only, can be used in the Viterbi algorithm to lower bound the free Euclidean distance.

The remaining of this section will examine the tightness of this lower bound under various circumstances.

Defining

$$A_0(\underline{\varepsilon}) = \sum_k (1 - \varepsilon_k)^2$$

and

$$A_i(\delta, \underline{\varepsilon}) = \sum_k \delta_k(i)(1 - \varepsilon_k)(1 - \varepsilon_{k-i})$$

in which δ denotes the set of $\underline{\delta}(i)$, so that we may express

$$D[\underline{v}, \underline{v}'] = A_0(\underline{\varepsilon})\ h_0 + 2 \sum_i A_i(\delta, \underline{\varepsilon})\ h_i$$

The tightness of the bound depends on the degree of freedom to choose δ so that as many as possible of the inequalities used in lower bounding $D[\underline{v}, \underline{v}']$ become strict equalities. The restrictions on δ to achieve tightness of the bound are as follows,

I. For $h_i < 0$, we want

$$A_i(\delta, \underline{\varepsilon}) = \sum_k (1 - \varepsilon_k)(1 - \varepsilon_{k-i})$$

constraining

$$\delta_k(i) = 1$$

for k's satisfying

$$\varepsilon_k = \varepsilon_{k-i} = -1$$

II. For $h_i > 0$, we want

$$A_i(\delta, \underline{\varepsilon}) = \sum_k -(1 - \varepsilon_k)(1 - \varepsilon_{k-i})$$

constraining

$$\delta_k(i) = -1$$

for k's satisfying

$$\varepsilon_k = \varepsilon_{k-i} = -1$$

III. Each $\underline{v}$ defining a δ must be a codeword of G.

A $\underline{v}$ that will satisfy constraint I is $\underline{v} = \underline{1}$ when $\underline{\delta}(i) = \underline{1}$ which corresponds to feeding an all-zero sequence into G, hence constraint III is also satisfied. Therefore, the lower bound is always tight if h_is are negative for nonzero i.

A few $\underline{\delta}(i)$ which satisfy constraint II are

$$\underline{\delta}(i) = \underline{-1}$$

$$\underline{\delta}(i) = \underline{\varepsilon}$$

$$\underline{\delta}(i) = T^i\underline{\varepsilon}$$

or

$$\underline{\delta}(i) = \underline{\varepsilon}T^i\underline{\varepsilon} \qquad (\text{i.e., } \delta_k(i) = \varepsilon_k\varepsilon_{k-i})$$

in which the operator T^i delays the $\underline{\varepsilon}$ sequence by i places. Note that constraint III may not be satisfied.

Less nonzero h_is would impose less restrictions on δ, making the lower bound fairly tight. We expect the lower bound to be strictly tight if there is only one nonzero h_i besides h_o. In case there are many nonzero h_is, attention should be paid to the largest h_i. Mathematically, it is not possible to have many large h_is.

This section is wrapped up by considering the example of rate ½ coded 4ϕ with only h_0 and h_1 nonzero (all h_i's are real). The 2 outputs of the binary rate ½ encoder are regarded as two PAM data streams, each modulating the carriers cos $2\pi f_c t$ and sin $2\pi f_c t$ which are mutually orthogonal. If h_1 is negative, the bound is tight by the previous discussion. If h_1 is positive, we know that $\delta(1) = \underline{\varepsilon}$ (for each stream) would satisfy II. It remains to show that the $\underline{v}$ which defines $\delta(1)$ is a codeword, hence satisfying III. Instead, we will first prove a stronger statement for rate ½ 4ϕ-PSK

Theorem

For every codeword $\underline{\varepsilon}$ and $\underline{v}$ having
$D[\underline{\varepsilon} \cdot \underline{v}, \underline{v}] = A_0(\underline{\varepsilon}) + 2A_1(\{\underline{\delta}(1)\}, \underline{\varepsilon})\, h_1$
there exists a codeword, $\underline{v}'$, obtained from $\underline{v}$ by a 1 to 1 onto mapping such that
$D[\underline{\varepsilon} \cdot \underline{v}', \underline{v}'] = A_0(\underline{\varepsilon}) - 2A_1(\{\underline{\delta}(1)\}, \underline{\varepsilon})\, h_1$

Proof

Defining $\underline{\delta}'(1)$ by

$$\delta_k'(1) = v_k' \, v_{k-1}'$$

Consider obtaining $\underline{\delta}'(1)$ from $\underline{\delta}(1)$ from the following 1-1 onto mapping

$$\underline{\delta}'(1) = \underline{\delta}(1) \cdot \underline{\varepsilon}$$

Consequently

$$D[\underline{\varepsilon} \cdot \underline{v}', \underline{v}'] = A_0(\underline{\varepsilon}) + 2A_1\{[\underline{\delta}'(1)], \underline{\varepsilon}\} h_1$$

$$= A_0(\underline{\varepsilon}) + 2 \sum_k \delta_k'(1)(1 - \varepsilon_k)(1 - \varepsilon_{k-1}) h_1$$

$$= A_0(\underline{\varepsilon}) + 2 \sum_k \delta_k(1) \varepsilon_k (1 - \varepsilon_k)(1 - \varepsilon_{k-1}) h_1$$

$$= A_0(\underline{\varepsilon}) - 2A_1(\{\delta(1)\}, \underline{\varepsilon}) h_1$$

To show $\underline{v}'$ is a codeword, we note that $\underline{\delta}(1)$ is a codeword, hence $\underline{\delta}'(1)$ is also a codeword. Therefore

$$\underline{v}' = \underline{\delta}'(1) \cdot T\underline{\delta}'(1) \cdot T^2\underline{\delta}'(1) \cdot T^3\underline{\delta}'(1) \ . \ . \ .$$

is also a codeword formed by multiplying the delayed versions of the codeword $\underline{\delta}'(1)$.

Q.E.D.

This theorem tells us that the distribution of the coefficients for h_1 is symmetrical about 0. Furthermore, we know that feeding an all-zero sequence [when $\underline{\delta}(1) = \underline{1}$] generates the largest A_1 for given $\underline{\varepsilon}$. Therefore, the smallest A_1 is $-A_1(\{\underline{1}\}, \underline{\varepsilon})$ when

$$\underline{\delta}'(1) = \underline{1} \cdot \underline{\varepsilon} = \underline{\varepsilon}$$

The next section will demonstrate how to search for optimal rate ½ encoder with positive h_1.

6.2 CODE SEARCHING

The minimum free square Euclidean distance is given by

$$d_f = \min_{(\underline{\varepsilon}_i, \underline{\varepsilon}_q)} \sum_k \Big\{ (1 - \varepsilon_{k,i})^2 h_0 - 2(1 - \varepsilon_{k,i})(1 - \varepsilon_{k-1,i}) h_1 + (1 - \varepsilon_{k,q})^2 h_0 - 2(1 - \varepsilon_{k,q})(1 - \varepsilon_{k-1,q}) h_1 \Big\}$$

in which the subscript i and q corresponds to quantities associated with the in-phase and quadrature channels. Let λ_k denotes each term in the summation. The minimization of

$$\lambda = \sum_k \lambda_k$$

can be performed using the Viterbi algorithm. The states are defined by using the extended state concept described in Section 3.3. The state information at time k would be sufficient to determine ε_k and ε_{k-1} for the two orthogonal channels, and consequently λ_k.

The computer program first of all sets up tables showing the possible state transitions and the value of λ_k associated with each transition. Then the minimum-free distance path is trellis searched until every state has accumulated a metric greater than the minimum-free distance found so far. This stopping result is based on the fact that λ_k is always nonnegative since

$$(1 - \varepsilon_k)^2 h_0 - 2(1 - \varepsilon_k)(1 - \varepsilon_{k-1}) h_1$$

equals zero if

$$\varepsilon_k = 1$$

and equals

$$4h_0 - 4(1 - \varepsilon_{k-1}) h_1$$

$$\geqq 4h_0 - 8h_1$$

$$> 0$$

if

$$\varepsilon_k = -1$$

provided

$$h_1 < \frac{1}{2} h_0$$

In fact, it can be proved that $h_1 \leqq \frac{1}{2} h_0$ if $s = 1$. We shall have a small digression here to provide the proof.

Let E be a set of positive integers such that $i \varepsilon E$ iff $h_i \neq 0$. Now

$$h_i = \int_{-\infty}^{\infty} h(t)\, h(t - iT)\, dt$$

Applying Parseval's theorem and assuming real h(t) give

$$h_i = \int_{-\infty}^{\infty} |H(f)|^2 \cos 2\pi ifT\, df$$

and consequently breaking up the integral into intervals of [n/T - 1/2T, n/T + 1/2T] and afterwards through a change of variable, we have

$$h_i = \int_{-1/(2T)}^{1/(2T)} \left[\sum_{n=-\infty}^{\infty} \left| H\left(f + \frac{n}{T}\right) \right|^2 \right] \cos 2\pi ifT\, df$$

The folded spectrum inside the square bracket is named S(f) so that overall,

$$h_i = \int_{-1/(2T)}^{1/(2T)} S(f) \cos 2\pi i f T \, df$$

Now if $h_i = 0$, we must have S(f) orthogonal to $\cos 2\pi i f T$ in $[-1/2T, 1/2T]$. Since $\cos 2\pi i f T$ are mutually orthogonal for nonzero i's, it follows immediately that

$$S(f) = 2T\left\{\frac{h_0}{2} + \sum_{i \varepsilon E} h_i \cos 2\pi i f T\right\}$$

The set of h_is must satisfy

$$S(f) \geqq 0 \qquad . . . *$$

since $|H(f)|^2$ is real and positive for all f.

Now if $E = \{1\}$, the maximum value of h_1 satisfying * is $1/2\ h_0$ so that

$$S(f) = T\, h_0\, (1 - \cos 2\pi f T)$$

which is a raised cosine spectrum. Q.E.D.

It follows immediately that

$$\lambda_k > 0 \qquad \text{for } \tfrac{1}{2} > h_1 > 0$$

if and only if

$$\lambda_k > 0 \qquad \text{for } h_1 = 0$$

This result has an important consequence concerning code catastrophe. The necessary and sufficient condition for a code to be noncatastrophic in the absence of ISI is that there is no zero weight path from some nonzero state back to itself. Therefore, a noncatastrophic code in the absence of ISI would also be noncatastrophic in the presence of ISI ($h_1 < \frac{1}{2}$) when s equals 1.

The computer program watches out for loops of zero weight to exclude code catastrophe. Optimal codes with up to 7 binary memories for various ranges of h_1 which are listed in Figures 6.1 - 6.6. These codes are represented by two subgenerator polynomials shown in Figure 7.1 for code #1 in Figure 6.1.

Code #1: 1 0 1

1 1 1

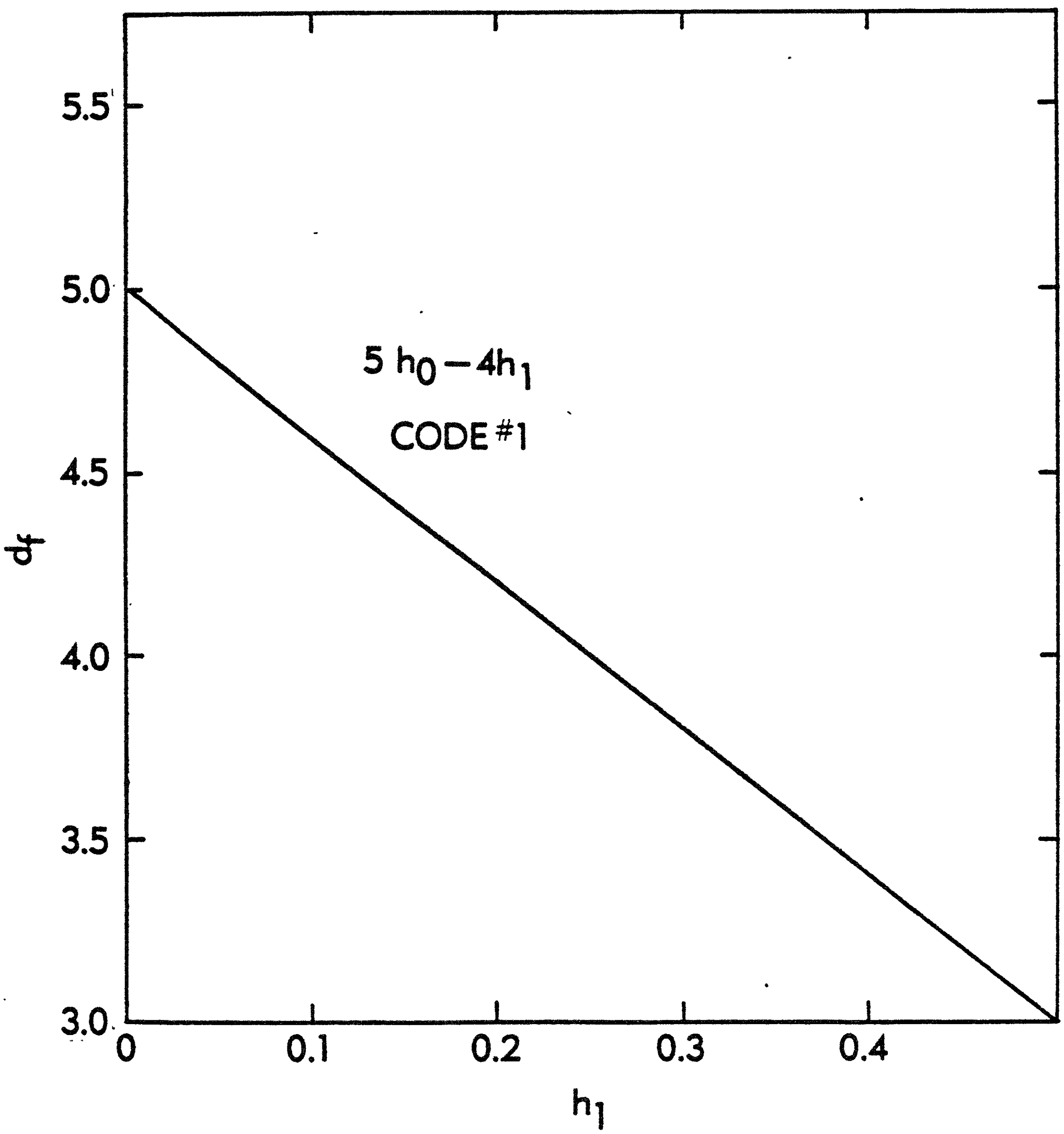


Figure 6.1. d_f vs h_1 for $\gamma = 2$

Code #2 : 1 0 1 1
1 1 1 1

Code #2' : 1 0 0 1
1 0 1 1

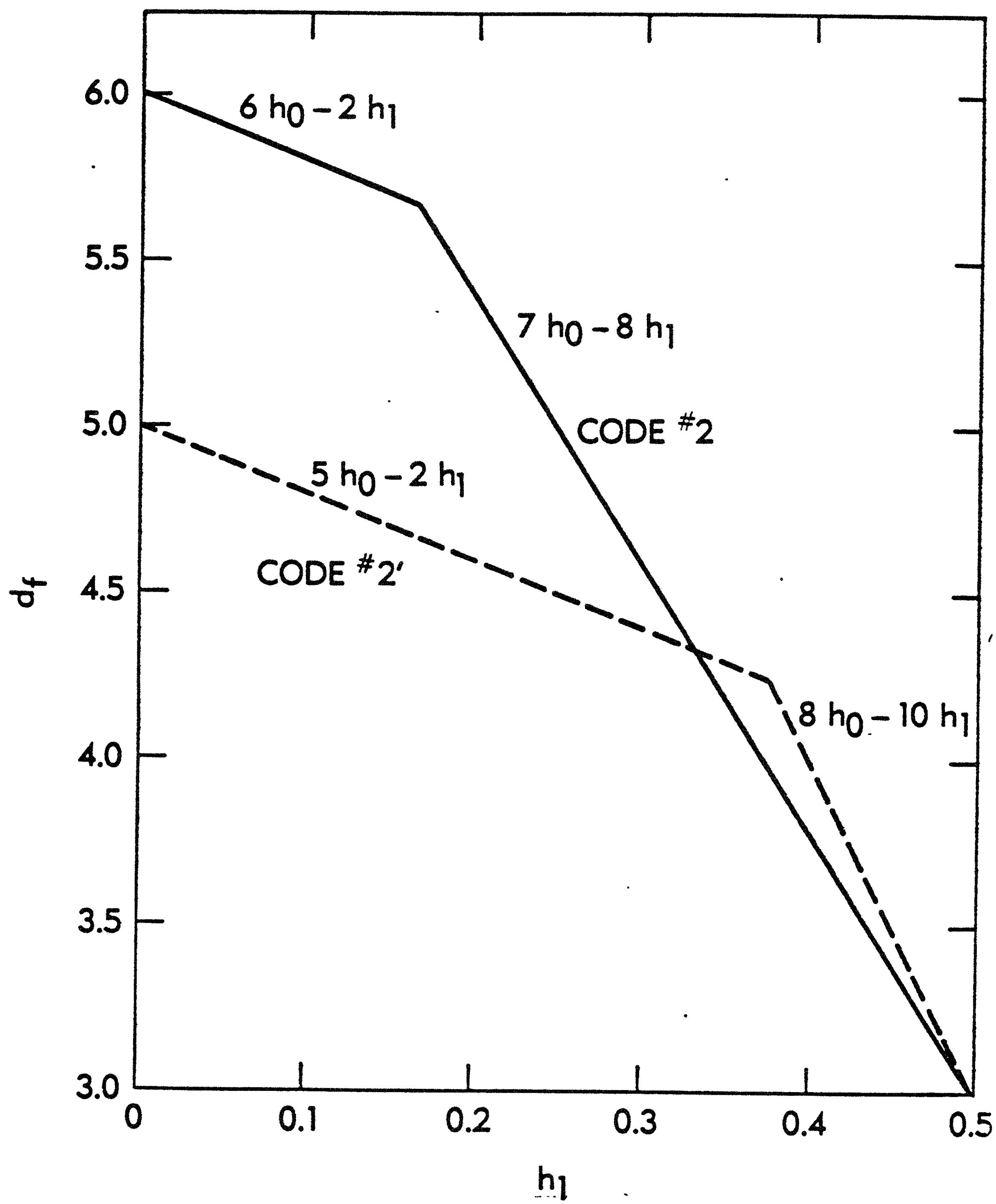


Figure 6.2. d_f vs h_1 for $\gamma = 3$

Code #3: 1 0 0 1 1
1 0 1 1 1

Code #3': 1 0 0 1 1
1 0 1 0 1

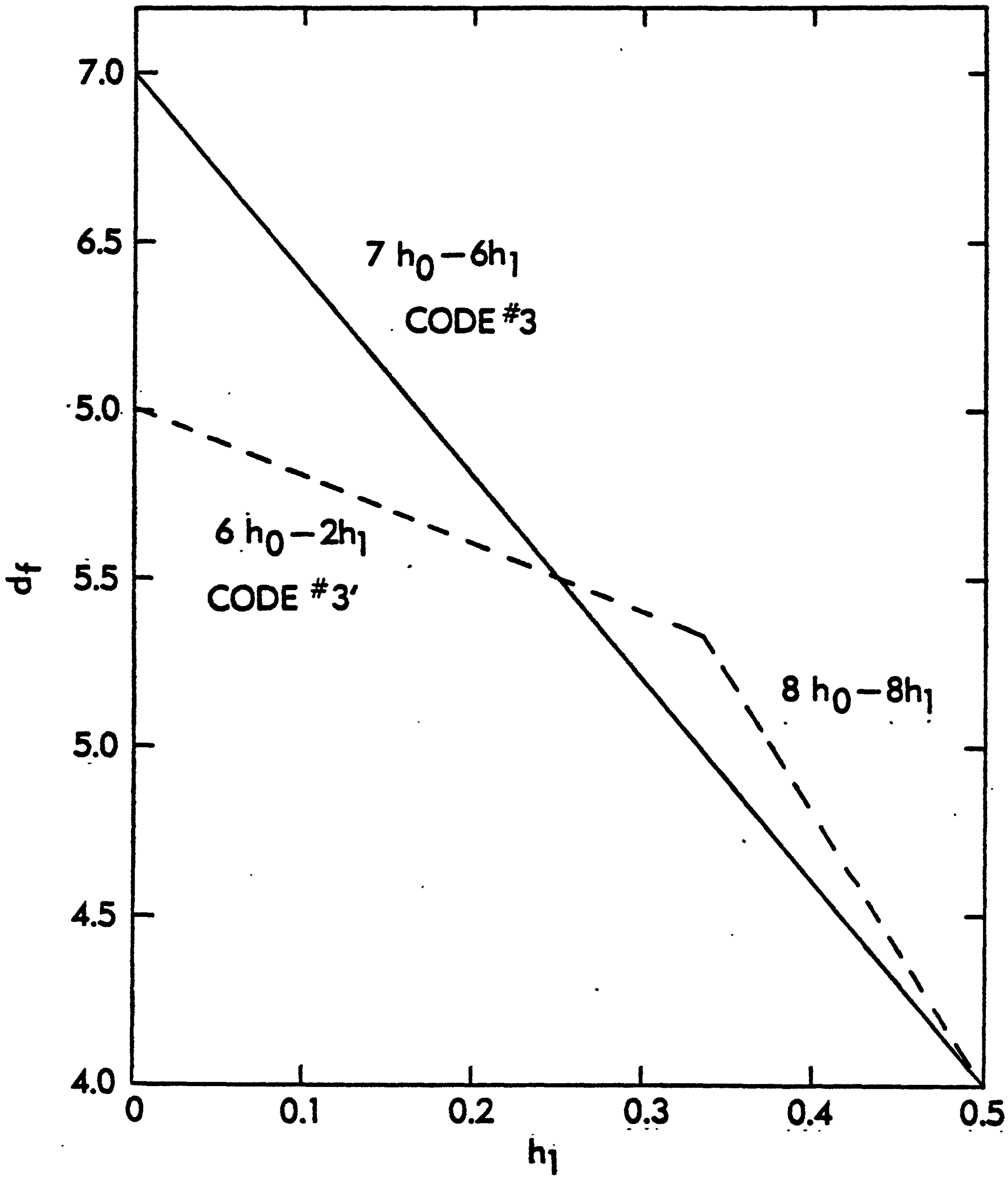


Figure 6.3. d_f vs h_1 for $\gamma = 4$

Code #4: 1 0 1 1 0 1
1 0 1 1 1 1

Code #4': 1 0 0 1 0 1
1 0 1 0 0 1

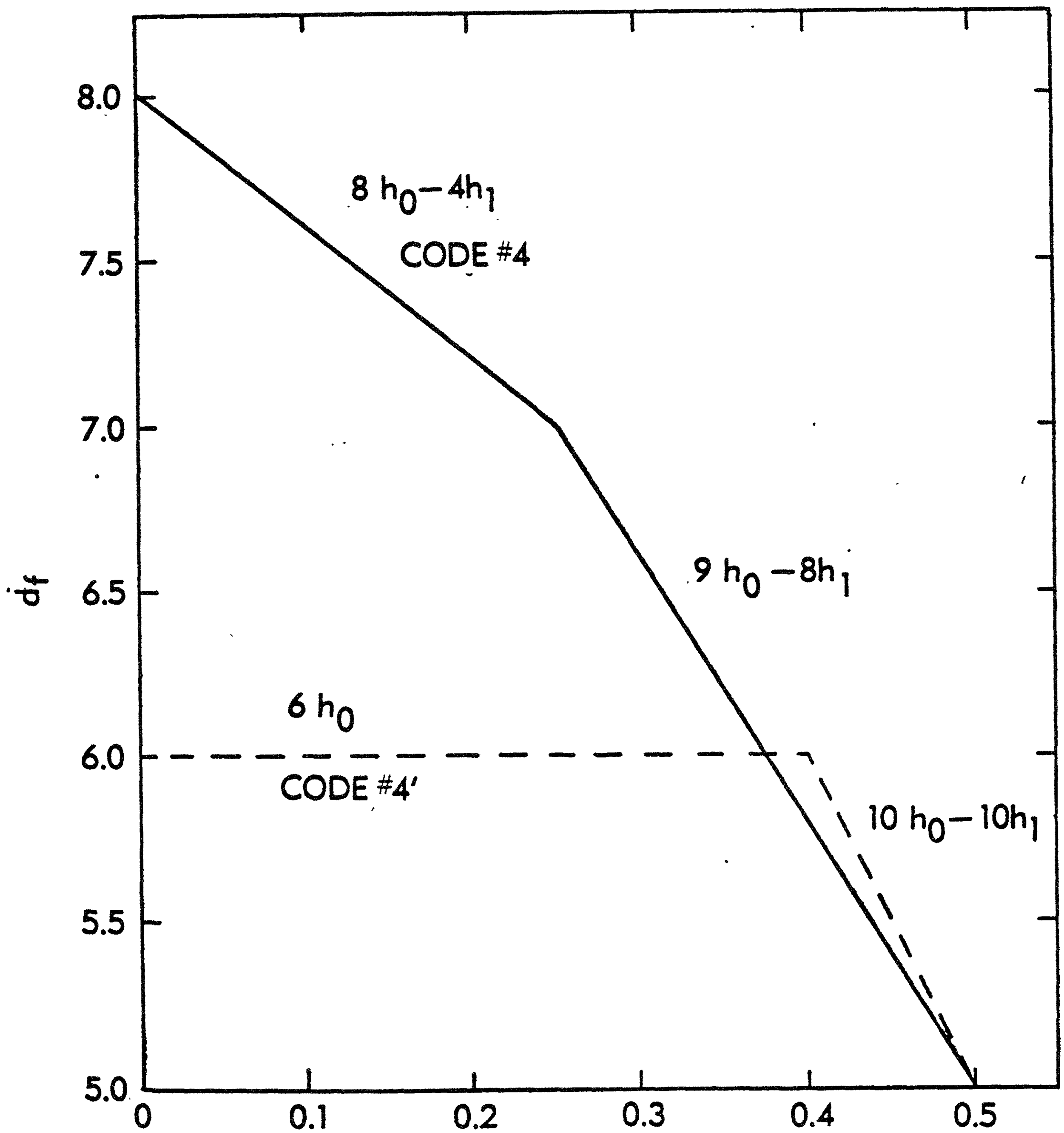


Figure 6.4. d_f vs h_1 for $\gamma = 5$

Code #5: 1 0 1 1 0 1 1
1 1 1 1 0 0 1

Code #5': 1 0 0 0 1 0 1
1 0 1 1 1 0 1

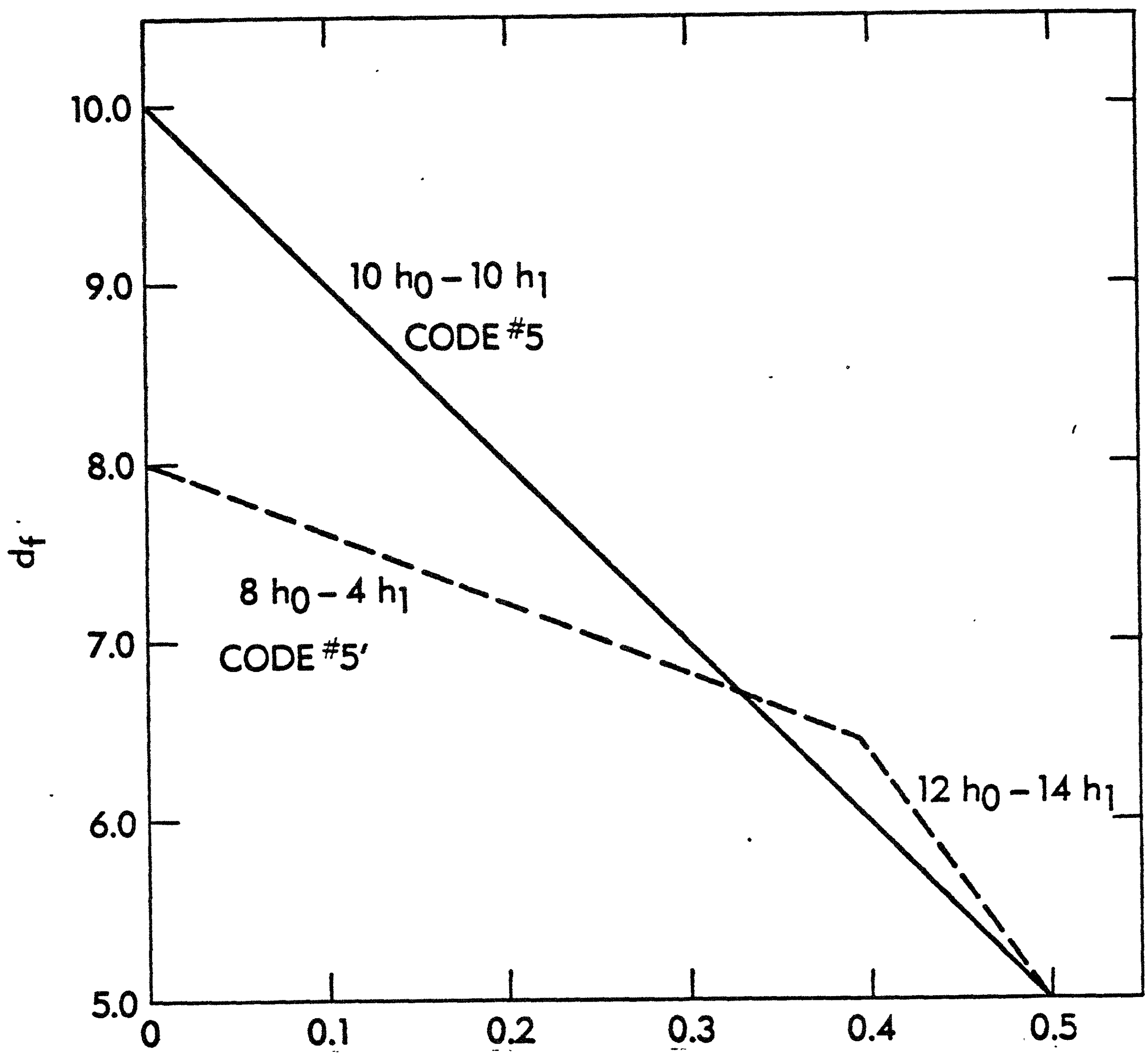


Figure 6.5. d_f vs h_1 for $\gamma = 6$

Code #6: 1 0 0 1 1 0 1 1
1 1 1 1 0 1 0 1

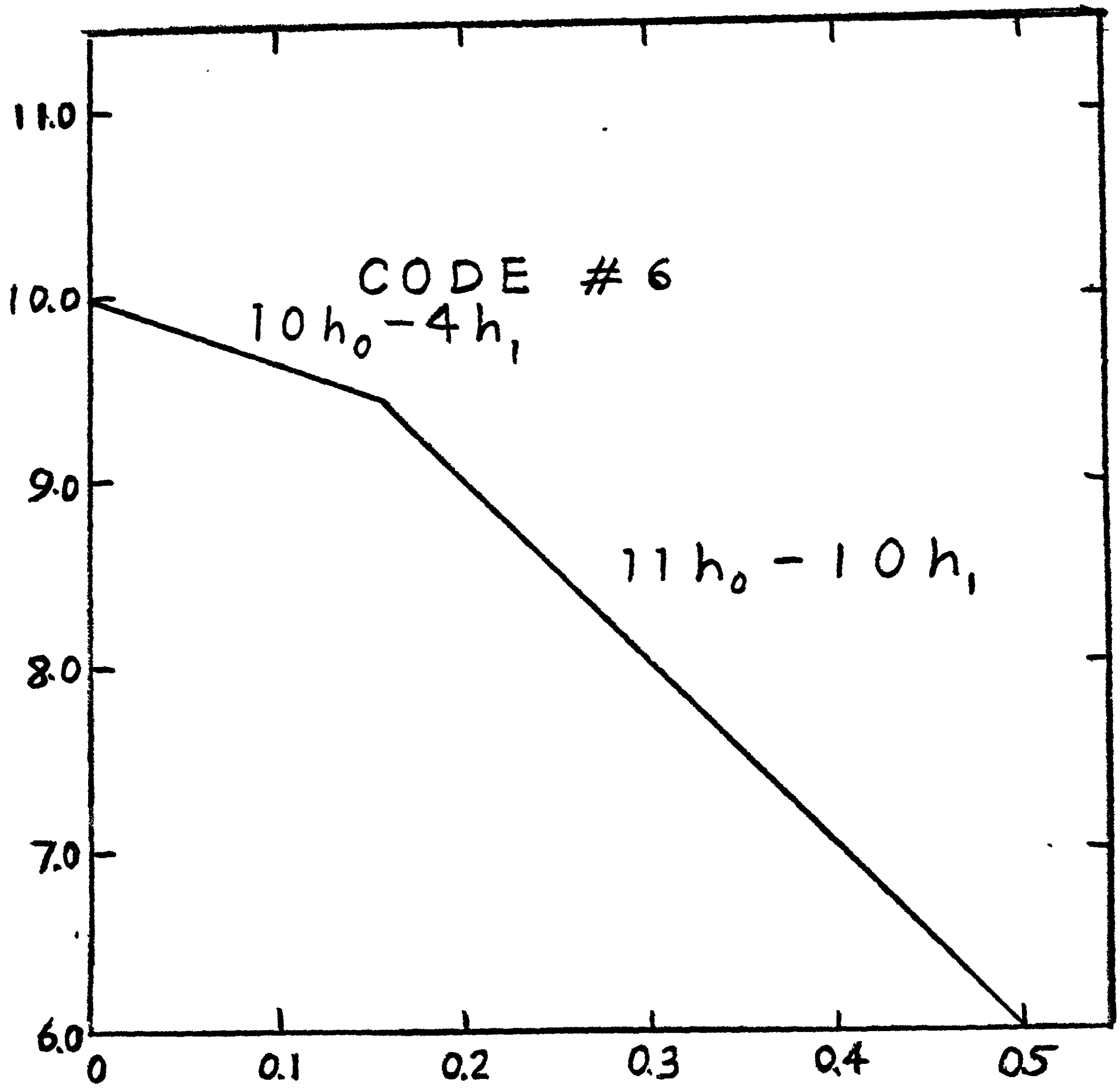


Figure 6.6. d_f vs h_1 for $\gamma = 7$

Chapter 7. PERFORMANCE EVALUATION

In this chapter, the E_b/N_o gain for binary rate 2/3 coded 8ϕ and rate 1/2 coded 4ϕ over uncoded 4ϕ-PSK is investigated. The minimum free distance gives the gain at high E_b/N_o while system performance at moderate E_b/N_o is pictured by simulation. The deterioration in asymptotic performance due to intersymbol interference predicted by the previous chapter will be tested experimentally. It is assumed that only $h_0 = 1$ and h_1 are nonzero in the following discussion. Furthermore, h_1 is considered to be real, which implies no cross-coupling of the channel.

7.1 THEORETICAL RESULTS

At high E_b/N_o, error occurrence is dominated by the minimum free distance paths. Therefore, signaling schemes with the same minimum Euclidean separation will have comparable asymptotic error performance. Euclidean separation for a given scheme can be enhanced by increasing signaling power. Consequently, asymptotic performance gain for a certain scheme over another is the reduction in E_b/N_o (in dB) which maintains the same minimum Euclidean separation. E_b is related to E_s by

$$R_s \, E_b = E_s$$

in which R_s is the number of information bits for each repetition interval.

a. Asymptotic Performance of Rate 2/3 Coded 8ϕ Without ISI.

In Chapter 5, d_f for the best binary rate 2/3 coded 8-ϕ encoders of up to 6 memories were found. D[0, 2], which is also the free square Euclidean distance for uncoded 4ϕ, is normalized to be 1. Therefore, if the same E_b is used for coded 8-ϕ as in uncoded 4ϕ, the gain in free square Euclidean distance, denoted by $d_{f,8\phi}/d_{f,4\phi}$, is equal to the d_f listed for the codes. Consequently, the coding gain of coded 8ϕ over uncoded 4ϕ is given by

$$G_{8\phi/4\phi} = 10 \log (d_{f,8\phi}/d_{f,4\phi}) = 10 \log d_f$$

These values are tabulated as follows,

γ	$d_{f,8\phi}/d_{f,4\phi}$ (same E_b)	$G_{8\phi/4\phi}$ (dB)
2	2.000	3.0
3	2.293	3.6
4	2.586	4.1
5	2.879	4.6
6	3.172	5.0

Theory for evaluating the minimum free distance in the presence of ISI for rate 2/3 coded 8ϕ is still lacking.

b. Asymptotic Performance of Rate 1/2 Coded 4ϕ With ISI.

For rate 1/2 coded 4ϕ, asymptotic performance deterioration evaluated theoretically in Chapter 6 is quite noticeable if ISI is present. The coding gain can be referenced with respect to uncoded signaling, either without ISI, or in the presence of ISI the effect of which is trellis decoded, or in the presence of ISI the effect of which is not trellis decoded.

The asymptotic E_b/N_o gains for the rate 1/2 encoders (R_s = 1) listed unprimed in Figure 6.2 - 6.7 over uncoded 4ϕ PSK without ISI can be calculated from their free square Euclidean distance.

G(dB)	h_1/h_o equals 0.0	0.1	0.2	0.3	0.4	0.5
γ						
2	3.98	3.62	3.22	2.79	2.17	1.76
3	4.77	4.62	4.31	3.60	2.76	1.76
4	5.44	5.05	4.62	4.15	3.62	3.01
5	6.02	5.80	5.56	5.19	4.62	3.98
6	6.99	6.53	6.02	5.44	4.77	3.98
7	6.99	6.81	6.53	6.02	5.44	4.77

in which

$$G = 10 \log\left(\frac{d_{f,\ \text{coded } 4\phi}}{d_{f,\ \text{uncoded } 4\phi}} \cdot \frac{1}{2}\right)$$

It has been shown in some cases [13] that the asymptotic exponent of the bit error probability without coding is not deteriorated, relative to the case of no ISI, if the ISI effect is trellis decoded. If the effect of ISI is not trellis decoded, the asymptotic error occurrences are due mainly to the weakest pulses as a result of destructive ISI. If the receive filter is matched to the weakest pulse, then the equivalent free distance of such a scheme can be shown to be

$$d^{o}_{f,\ \text{uncoded } 4\phi} = \text{energy of weakest pulse} = E_s\left(1 - 2\frac{h_1}{h_o}\right)$$

In the case of no coding, trellis decoding the effect of ISI would asymptotically recover a loss (with respect to any non-trellis decoding) which is upper bounded by

$$-10 \log \left(1 - 2 \frac{h_1}{h_o}\right)$$

No explicit expression was obtained regarding the equivalent 'free distance' for decoders which only trellis decode the effect of the encoder but not the effect of ISI.

It is rather unlikely that h_1 exceeding 0.2 would be adopted for satellite communication. For overlapped raised-cosine [6] pulse shaping with coding (h_1 equals 1/6 for $T = \frac{1}{2} \Upsilon$), asymptotic performance loss relative to the case of no ISI is about 0.5 dB, which is typical for the other pulse shapings considered in this thesis.

c. Non-Asymptotic Loss of ISI

The non-asymptotic loss due to ISI should be less than the asymptotic loss due to two reasons. First, many of the distances between pairs of codewords are larger than they would be in the absence of ISI, due to reinforcement by ISI. These improved Euclidean separations would reduce error occurrences non-asymptotically but have little effect asymptotically. Second, only a few information sequences $\underline{u}$ can make the error sequence $\underline{\varepsilon}$ to achieve the value of the lower bound for the Euclidean distance. In effect, the occurrence of minimum distance paths is much less frequent than in the case of no ISI.

When $h_0 = 1$ and $h_1 = 1/2$ are the only nonzero h_i, asymptotic loss can be seen from the table in part b above to be about 2 dB or 3 dB relative to the case of coding without ISI. Previously, Viterbi has shown by a random coding argument that the average loss over a code ensemble in such a case would have been about 1 dB for antipodal transmission relative to the case of no ISI. It may be unfair to compare our results with Viterbi's since ours are based on 4ϕ-PSK rather than antipodal signaling. Also, the non-asymptotic performance loss should be significantly less than 2 dB or 3 dB for 4ϕ-PSK with ISI due to reasons mentioned previously.

Codes with good Hamming distance for 4ϕ-PSK usually have a larger deterioration of free distance once ISI is introduced than those with mediocre Hamming distance. This turns out to be the case when d_f is evaluated during the code search. Therefore, a random coding argument may not properly reflect the deterioration on good codes due to ISI. The simulation performed in the next section does seem to suggest a 2 dB loss projected asymptotically, rather than the diminishing loss asymptotically for antipodal signaling as suggested in Figure 5-11 of [13].

7.2 COMPUTER SIMULATION

Two computer programs listed in Appendix D which optimally decode rate 1/2 coded 4ϕ with $\gamma = 2$ and rate 2/3 coded 4ϕ with $\gamma = 4, 6$ in the presence of ISI were implemented on the IBM 3032 machine. Two additional programs which do not trellis decode the effect of ISI (i.e., an ordinary Viterbi decoder which would be optimal without ISI) were also written to compare their performance loss relative to the optimal decoders.

The programs each contain an encoder which takes in a random binary sequence. The sufficient statistics sequence $\{r_k\}$ obtained by demodulation is fed into the decoder and the decoded sequence is compared with the properly delayed input sequence. The channel is asumed to be AWGN. As we shall see, the physical waveform does not have to be generated in order to find $\{r_k\}$.

Recall that each r_k is given by

$$r_k = \int_{-\infty}^{\infty} \sqrt{2}\, e^{j2\pi f_c t}\, r(t)\, h(t - kT)\, dt$$

in which

$$r(t) = y(t) + n(t)$$

where

$$y(t) = \frac{1}{2} \sqrt{2E_s} \sum_k \left\{ h(t - kT) e^{j2\pi f_c t + jv_k \pi/M} + (\cdot)^* \right\}$$

and $n(t)$ is a zero mean uncorrelated white Gaussian process with

$$E[n(t)\, n(t')] = \frac{N_o}{2} \delta(t - t')$$

The expression for r_k can be reduced, by the baseband assumption and assuming non-zero h_o and h_1 only, to the form

$$r_k = y_k + n_k$$

in which

$$y_k = \sqrt{E_s} \{\cos \frac{2\pi v_k}{M} + h_1 \cos \frac{2\pi v_{k-1}}{M} + h_1 \cos \frac{2\pi v_{k+1}}{M}\}$$

$$- \sqrt{E_s}\, j\{\sin \frac{2\pi v_k}{M} + h_1 \sin \frac{2\pi v_{k-1}}{M} + h_1 \sin \frac{2\pi v_{k+1}}{M}\}$$

and

$$n_k = n_{k,i} + jn_{k,q}$$

has $n_{k,i}$, $n_{k,q}$ being zero mean Gaussian random variables of variance $\sigma^2 = N_o/2$. All $n_{k,i}$'s and $n_{k,q}$'s are uncorrelated, except for consecutive $n_{k,i}$'s or consecutive $n_{k,q}$'s when

$$E[n_{k,i}\, n_{k-1,i}] = E[n_{k,q}\, n_{k-1,q}] = \frac{N_o\, h_1}{2}$$

The value of r_k can be readily generated for known $\underline{v}$:

Let us consider the generation of a zero mean real Gaussian sequence $\{u_k\}$ with

$$E[u_i\, u_j] = \begin{cases} \sigma^2 & \text{if } i = j \\ \sigma^2\, h_1 & \text{if } |i - j| = 1 \\ 0 & \text{otherwise} \end{cases}$$

Suppose $\{t_k\}$ is sequence of uncorrelated, zero mean and real Gaussian random variables each of variance 1. It can be readily shown that

$$u_k = \sigma\, \beta(t_k + \alpha t_{k-1})$$

in which

$$\alpha = \frac{1}{2h_1} - \left\{\left(\frac{1}{2h_1}\right)^2 - 1\right\}^{1/2}$$

$$\beta = (1 + \alpha^2)^{-1/2}$$

does has the desired mean, variance and correlation with other u_k's. Using this technique, the sequences $\{n_{k,i}\}$, $\{n_{k,q}\}$ and subsequently $\{r_k\}$ can be generated for the decoder.

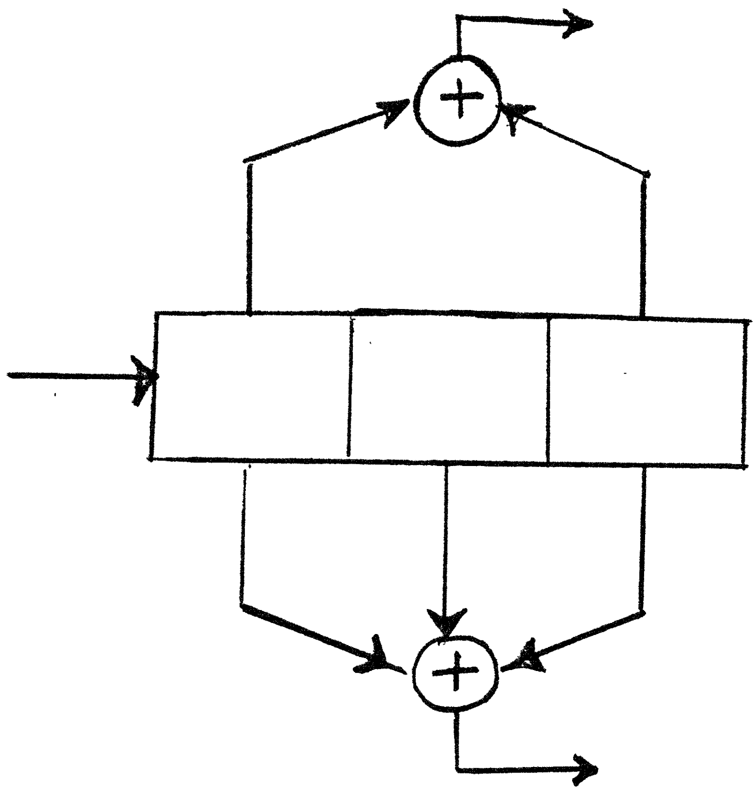
The encoder with extended memory discussed in Section 3.3 is used to define the states of the decoder. Specifically, a state is defined as the contents of the memories as well as the bits shifted out of the end of each queue at the previous instant. Possible state transitions and the associated branch metric are tabulated. In some versions of the program, quantization of values involved in the decoding is available. However, either real values or very fine quantization is used in the simulation. Since extensive simulation is rather expensive, we have not made enough runs to picture the effect of quantization. Such knowledge, however, is rather valuable from an implementation standpoint.

A survivor at each state is retained by choosing the branch transition merging into the state which makes the accumulated metric of that state a maximum. The survivor of each state is stored in a table. At each decoding stage the survivor which has accumulated the largest metric is traced back 100 state transitions to obtain the decoded information (1 bit for rate 1/2 coded 4ϕ and 2 bits for rate 2/3 coded 8ϕ).

The encoders shown in Figures 7.1 - 7.3 were simulated for various values of h_1. Computer programs which simulate the performance of uncoded 4ϕ-PSK and 8ϕ-PSK in the absence of ISI were also written. The bit error performances are shown in Figures 7.4 - 7.6. Specifically, Figure 7.4 gives the simulation results for rate 2/3 coded 8ϕ over the AWGN channel or INTELSAT V channel, without using controlled ISI. The simulations generating curves 4, 5, and 6 are performed by S. Lebowitz, assuming perfect timing and phase recovery and sufficient quantization in decoding. Figure 7.5 simulates the performance of rate 2/3 coded 8ϕ ($\gamma = 4$) with controlled ISI over an AWGN channel and compares the performances with and without extended state Viterbi decoding. Figure 7.6 is analogous to Figure 7.5, except the code studied is rate 1/2 coded 4ϕ with $\gamma = 2$.

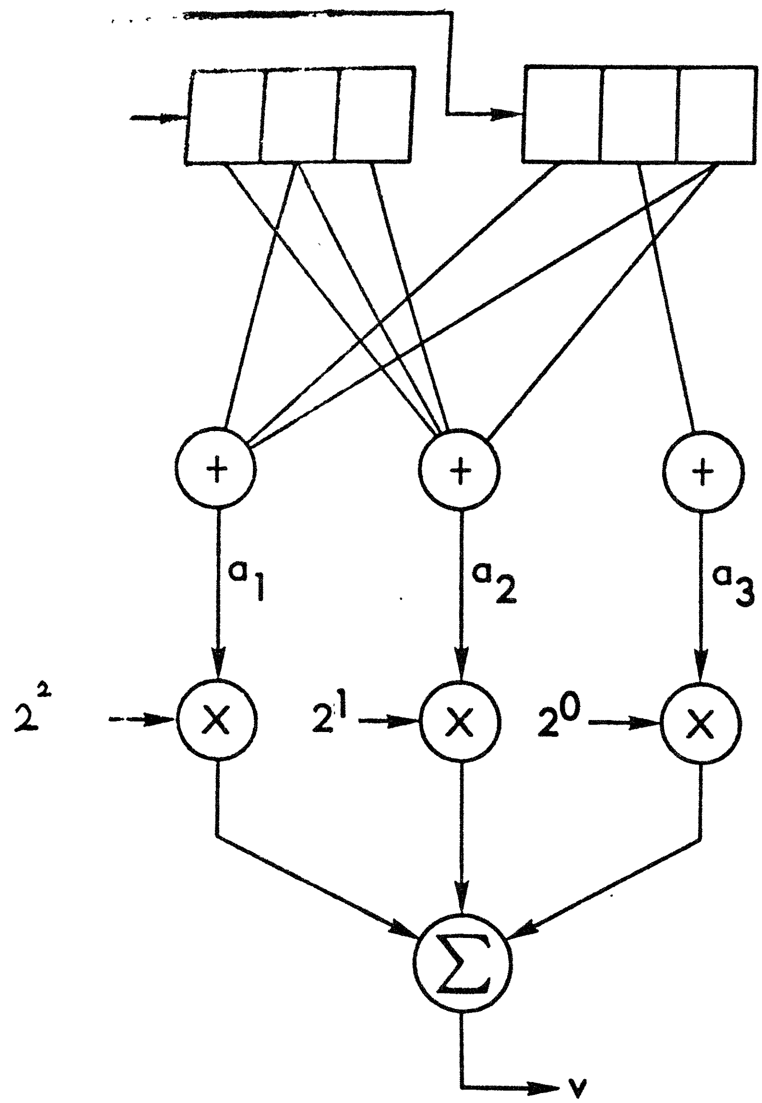
One rather surprising result of Figure 7.4 is that for $\gamma = 6$ and BER = 10^{-5}, the coding gain of 4.3 dB in the INTELSAT V channel is significantly higher than the coding gain of 3.7 dB for the AWGN channel. This demonstrates the robustness of the code against real-live channel impairments.

From Figures 7.5 and 7.6, it is seen that system performance deterioration is quite noticeable if the decoder does not trellis decode the effect of ISI.



Subgenerators: 1 0 1
1 1 1

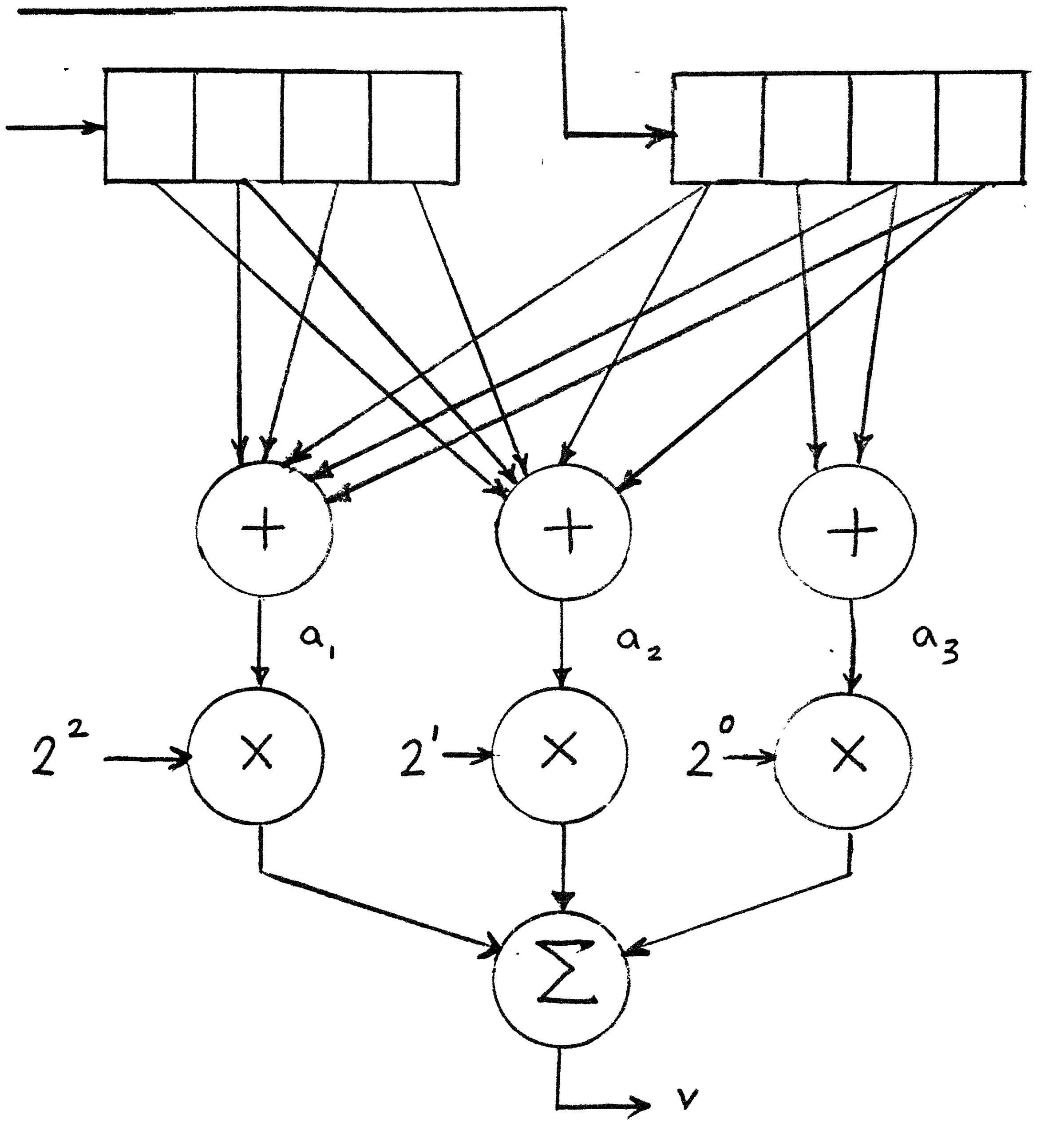
Figure 7.1. A Rate 1/2 $\gamma = 2$ Encoder



Subgenerators: 0 1 0 1 0 1
1 1 1 0 0 1
0 0 0 0 1 0

d_f = 2.586 Asymptotic = 4.1 dB coding gain.

Figure 7.2. A Rate 2/3 γ = 4 Encoder



Subgenerators: 0 1 1 0 1 0 1 1

1 1 0 1 1 0 0 1

0 0 0 0 0 1 1 0

d_f = 3.172 Asymptotic coding gain = 5.0 dB

Figure 7.3. A Rate 2/3 γ = 6 Encoder

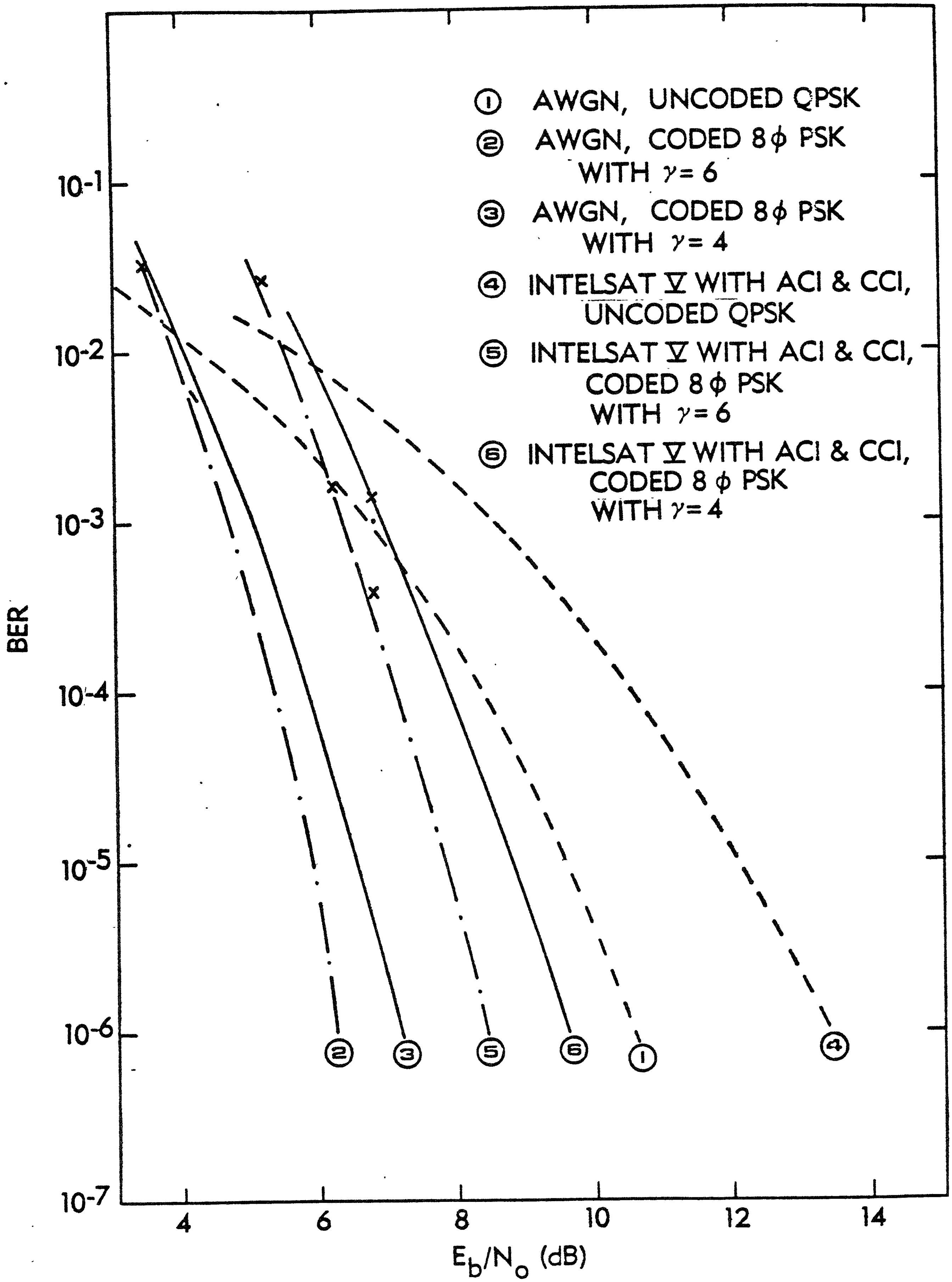


Figure 7.4. Performance of Rate 2/3 Coded 8Ø over AWGN and INTELSAT V Channel

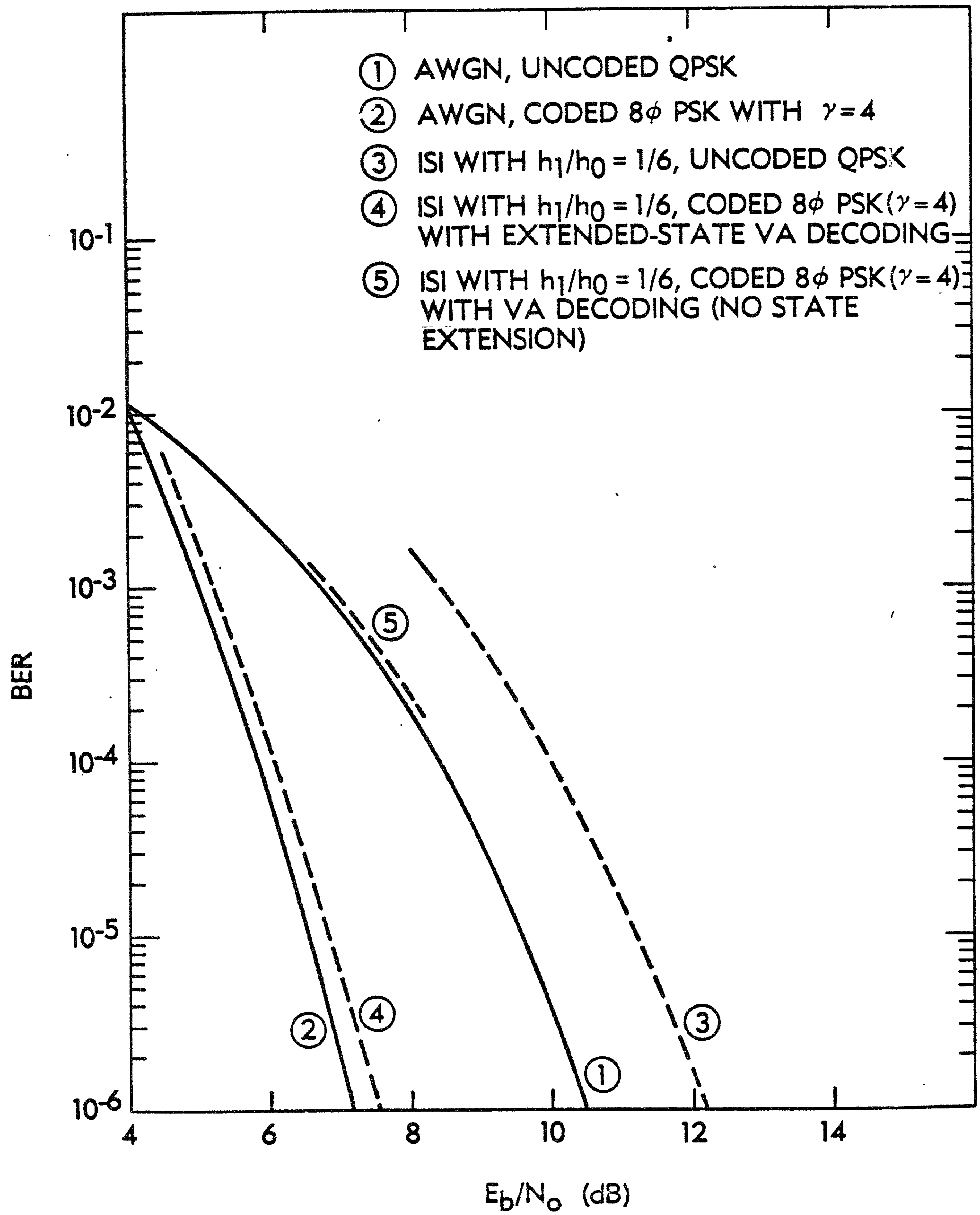


Figure 7.5. Performance of Rate 2/3 Coded 8ϕ over AWGN Channel with Controlled ISI

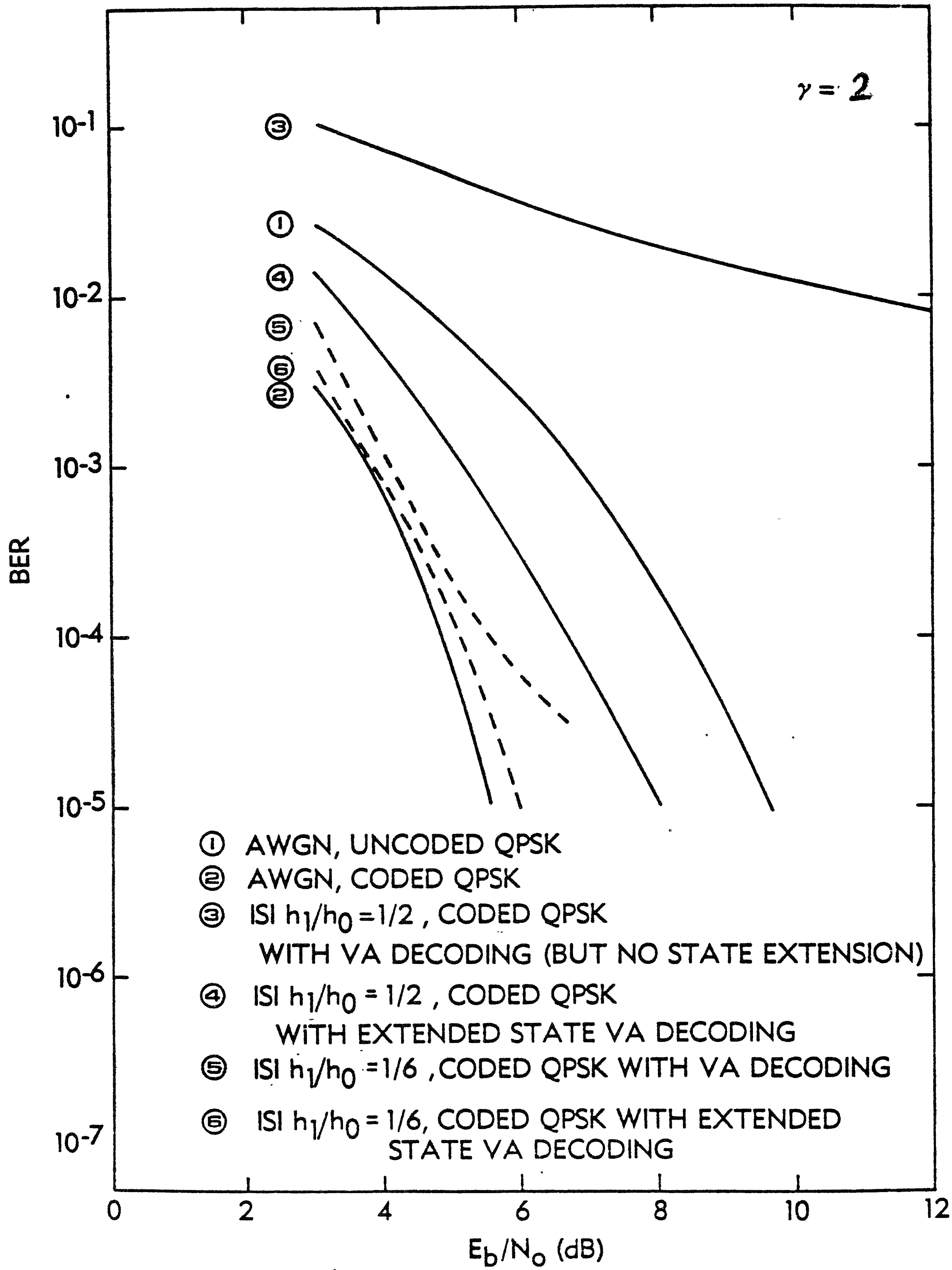


Figure 7.6. Performance of Rate 1/2 Coded 4∅ over AWGN Channel with Controlled ISI

The simulation results for rate 1/2 coded 4ϕ in Figure 7.6 seem to agree fairly well with the theoretical prediction of Chapter 6. On the other hand, it is surprising to see that the effect of ISI hardly deteriorates system performance for rate 2/3 coded 8ϕ in Figure 7.5. A plausible explanation is that the information sequence $\underline{u}$ which makes the error sequence $\underline{\varepsilon}$ achieve the lower bound in the absence of ISI may not bring about any reduction in free distance when ISI is present. In essence, there are two mismatched patterns of variance with respect to $\underline{u}$, one due to the fact that the (G, M, F) is not invariant even without ISI, and another due to the variance brought about by ISI. This complication also explains our failure to effectively lower bound the Euclidean free distance between any two input sequences.

The rate 2/3 decoders with 4 memories and 6 memories theoretically have 4.1 dB and 5.0 dB asymptotic coding gain in the absence of ISI over uncoded 4ϕ-PSK, which is comparable to the coding gain of 3.0 dB and 3.7 dB at BER = 10^{-5} from the simulation. Even though theoretically the 6 memory code has a 0.9 dB asymptotic gain over the 4 memory code, it is noteworthy that for the 6 memory case, the number and the length of the minimum distance paths are increased. In fact, a typical error event has about 10-15 errors for the $\gamma = 6$ decoder, compare to 3-6 errors for the case of $\gamma = 4$. However, we do expect the decoders to achieve their asymptotic coding gain at high enough E_b/N_o.

At each E_b/N_o, ten thousand to half a million information bits were decoded to obtain a reasonable average of the bit error probability. Therefore, simulation can be statistically reliable only for BER greater than 10^{-4}. An extensive study on the system performance for the various decoders and values of h_1 used is very expensive due to heavy computation requirements. Results presented in this thesis are for the purpose of illustration rather than as an extensive evaluation of performance.

The computer programs are documented in Appendix D.

Chapter 8. CONCLUSION AND SUGGESTION FOR FURTHER RESEARCH

The major contribution of this thesis is unifying modulation and coding as a single entity. While binary encoding which maximizes Hamming distance is a fairly mature field, principles for designing codes for modulation schemes and channel characteristics in general are still very lacking. The conceptualization of a transmission system as a triple (G, M, F) offers a convenient formulation, and this thesis serves as an example for a more general framework.

Chapter 4 offers a unified approach for pulse design, taking into account the spectral roll-off requirements and restrictions imposed by the band limiting and nonlinear channel. In effect, less than unity channel bandwidth per symbol rate can be realized (for INTELSAT V, BW/SR = 80 MHz/60 MHz = 1.33).

The bounding techniques for minimum free square Euclidean distance used in Chapter 5 and 6 can be used in general for variant schemes. Using such techniques, we addressed the methods of searching for optimal code for multi-phase PSK and for modulations with controlled ISI. We have also demonstrated the robustness of rate 2/3 coded 8-ϕ against the INTELSAT V channel impairments through simulation.

The main theme of Chapter 5 is left unanswered, namely, which (G, M, F) for multiphase modulation is the best for schemes of similar complexity. While we are satisfied with the simplicity and performance of the binary encoders found, we suppose octal and GF(8) encoders with better distance properties can be discovered if more powerful rejection rules are adopted in the code searching. A generalized concept of complexity (in terms of decoder complexity, inevitable decoding delay etc.) required for a certain level of system performance is needed for meaningful comparison of various classes of encoders, the formulation of which by itself is a complex subject.

Trellis decoding the effect of ISI requires increased complexity for the decoder. An interesting question is whether we would be better off if the same complexity is used for an encoder with longer constraint length. Simulation results for pulses with significant ISI ($h_1 > 0.1$) seem to favor the encoder which also trellis decode the effect of ISI.

For implementation purposes, we would like to know the effect of quantization and path memory length on error performance in the decoding process. There is a strong need in satellite communication to advance the state of art of implementing very high speed hardwares for Viterbi algorithm decoding.

The Viterbi algorithm is the optimal (in the maximum likelihood sense) scheme for decoding, at the cost of exponential increase of complexity with constraint length. Sequential decoding algorithms on the other hand reduces decoder complexity at the expense of increased delay, memory and computation requirements. Between the two extreme, a reduced state decoding algorithm, if one ever exists, seems to be a good compromise. We suspect that the increased complexity (4 fold for rate 2/3 coded 8ϕ with nonzero h_o, h_1) due to ISI can be reduced by certain manner of ignoring or combining some state or state transitions. Success in the treatment of the ISI case may bring insight concerning reduced state Viterbi decoding for a long constraint length encoder. Naturally, the ignoring of states introduced by nonzero h_2, h_3, etc. is a trivial example of this reduced state approach. Reduced state decoding for a given encoder would inevitably deteriorate performance, but there may be gain compared with full state decoding of the same complexity. The success of the reduced state approach depends very much upon the 'distinctiveness' of the states. In the case of ISI, the diminishing 'distinctiveness' as ISI is reduced may enable us to conglomerate states together.

In conclusion, we would like to see how the problem of severe bandwidth and power limitation has been dealt with in this thesis by proposing specific schemes for implementation which acquires good performance improvement with reasonable increase of complexity. Pulse shapings are to be chosen from those suggested in Chapter 4 with BT product of around 0.8 when filter loss is less than 1 dB. It may be possible at the cost of increased filter loss and introducing a nonzero h_2 to obtain a BT product of about 0.7 with $1/3 < \theta < 1/2$ for the 4th order beta and truncated sinc functions. In all cases, the quadrature component should be staggered with respect to the in-phase component to decrease filter loss. For very small earth terminals with severely limited transmission power which would necessitate the use of a low rate coding scheme, the rate 1/2 encoders are recommended. For transmission systems such as TDMA which is power limited in order to reduce ACI and OBE, the rate 2/3 coded 8ϕ with 4 binary memory would offer a 3 - 4 dB gain due to coding. The full benefit of these transmission schemes cannot be fully estimated until they are simulated in a more realistic system environment.

APPENDIX A. Analyses of Pulse Optimization for $m = 1, 2$

From Section 4.2, $h_n^{o}(t)$ is the solution associated with the smallest λ for the following boundary value problem,

$$\frac{d^{2n}}{dt^{2n}} h(t) = (-1)^n \lambda \, h^m(t)$$

subject to

$$\int_{-\tau/2}^{\tau/2} h^2(t) = 1$$

$$h(t) = 0 \qquad \text{for} \qquad t > \tau/2$$

$$h^{(k)}(\pm \tau/2) = 0 \qquad \text{for} \qquad 0 \leqq k \leqq n-1$$

Notice that the energy of h(t) has been normalized.

Case 1. $m = 1$

The solution of the differential equation is the beta function, a well-known distribution in probability theory

$$h_n(t) = \frac{1}{\tau} \frac{(2n+1)!}{n!\ n!} \left(1 - \frac{2t}{\tau}\right)^n \left(1 + \frac{2t}{\tau}\right)^n \qquad t \leqq \tau/2$$

$$= 0 \qquad \text{otherwise}$$

with

$$\int_{-\tau/2}^{\tau/2} h_n(t)\, dt = 1$$

Normalizing the energy gives

$$h_n^{o}(t) = \frac{h_n(t)}{\left\{\int_{-\tau/2}^{\tau/2} [h_n(t)]^2 \, dt\right\}^{1/2}}$$

To evaluate the denominator, we observe that

$$1 = \int_{-\tau/2}^{\tau/2} h_{2n}(t) \, dt$$

$$= \frac{1}{\tau} \frac{(4n + 1)!}{(2n)!(2n)!} \left(\frac{\tau}{1} \frac{n! \; n!}{(2n + 1)!}\right)^2 \int_{-\tau/2}^{\tau/2} [h_n(t)]^2 \, dt$$

Consequently,

$$h_n^{o}(t) = A_n\left(1 - \frac{2t}{\tau}\right)^n \left(1 + \frac{2t}{\tau}\right)^n$$

in which

$$A_n = \frac{[(4n + 1)!]^{1/2}}{(2n)!} \frac{1}{\sqrt{\tau}}$$

The plots of $h_n^0(t)$ for n from 0 to 4 are given in Figure 4.1.

The value of $B_{1,n}$ can be derived from its definition. For $h_b(t)$,

$$Q\{h_b(t)\} = \frac{2}{2n + 1} \left(\frac{B_{1,n}}{2}\right)^{2n + 1} \left|H_b(0)\right|^2$$

$$R\{h_b(t)\} = \int_{-\infty}^{\infty} h_b(t) \, dt$$

$$= H_b(0)$$

since

$$Q\{h_b(t)\} = Q\{h_n{}^o(t)\}$$

$$R\{h_b(t)\} = R\{h_n{}^o(t)\}$$

therefore by eliminating $H_b(0)$, we have

$$\beta_{1,n} = \tau\ B_{1,n}$$

$$= 2\tau \left\{ (n + \tfrac{1}{2}) \frac{Q\ \{h_n^o(t)\}}{R\{h_n{}^o(t)\}^2} \right\}^{\frac{1}{2n+1}}$$

$$= 2\tau\ \{(n + \tfrac{1}{2})\ \lambda\ R\{h_n{}^o(t)\}^{-1}\}^{\frac{1}{2n+1}}$$

The eigenvalue is given by

$$\lambda = (-1)^n \frac{d^{2n}}{dt^{2n}} h_n{}^o(t)$$

The highest order term in t of $h_n{}^o(t)$ can be found by binormial expanding $h_n{}^o(t)$ to be

$$A_n(-1)^n \left(\frac{2t}{\tau}\right)^{2n}$$

and subsequently

$$\lambda = A_n\ (2n)! \left(\frac{2}{\tau}\right)^{2n}$$

on the otherhand

$$R\{h_n{}^o(t)\} = \int_{-\tau/2}^{\tau/2} h_n{}^o(t)\ dt$$

$$= A_n \frac{n!\ n!}{(2n+1)!}\ \tau$$

Substituting the values of λ and $R\{h_n{}^o(t)\}$ into the expression for $\beta_{1,n}$ gives

$$\beta_{1,n} = 2\left\{\frac{1}{\sqrt{2}} \frac{(2n+1)!}{n!}\right\}^{\frac{2}{2n+1}}$$

The values of $\beta_{1,n}$ are as follows:

n	0	1	2	3	4	5
$\beta_{1,n}$	1	5.24	8.96	12.40	15.72	18.95

The spectrum of $h_n{}^o(t)$ is

$$H_n{}^o(\omega) = A_n \int_{-\tau/2}^{\tau/2} \left(1 - \frac{2t}{\tau}\right)^n \left(1 + \frac{2t}{\tau}\right)^n e^{-j\omega t}\, dt$$

$$= A_n \tau \int_{-1/2}^{1/2} \left(\frac{1}{2} - t\right)^n \left(\frac{1}{2} + t\right)^n e^{\beta t}\, dt \;, \quad \beta = -j\omega\tau$$

$$= A_n \tau \sqrt{\pi}\, \beta^{-n-(1/2)} n!\; I_{n+(1/2)}\left(\frac{\beta}{2}\right)$$

in which

$$I_{n+(1/2)}\left(\frac{\beta}{2}\right)$$

$$= (\pi\beta)^{-1/2} \left\{ e^{\beta/2} \sum_{k=0}^{n} \frac{(-1)^k (n+k)!}{k!(n-k)!\; \beta^k} \right.$$

$$\left. + (-1)^{n+1} e^{-\beta/2} \sum_{k=0}^{n} \frac{(n+k)!}{k!(n-k)!\; \beta^k} \right\}$$

is a special case of the modified Bessel functions of the first kind. After a considerable amount of computation, it can be shown that

$$H_0{}^{o}(\omega) = (\tau)^{1/2} \left(\frac{\omega\tau}{2}\right)^{-1} \sin \frac{\omega\tau}{2}$$

$$H_1{}^{o}(\omega) = (5!\tau)^{1/2} \frac{1}{2!} (\omega\tau)^{-2} \left\{ 2\left(\frac{\omega\tau}{2}\right)^{-1} \sin \frac{\omega\tau}{2} -2 \cos \frac{\omega\tau}{2} \right\}$$

$$H_2{}^{o}(\omega) = (9!\tau)^{1/2} \frac{2!}{4!} (\omega\tau)^{-3} \left\{ 6\left(\frac{\omega\tau}{2}\right)^{-2} \sin \frac{\omega\tau}{2} \right.$$

$$\left. -6 \left(\frac{\omega\tau}{2}\right)^{-1} \cos \frac{\omega\tau}{2} - 2 \sin \frac{\omega\tau}{2} \right\}$$

$$H_3{}^{o}(\omega) = (13!\ \tau)^{1/2} \frac{3!}{6!} (\omega\tau)^{-4} \left\{ 30\left(\frac{\omega\tau}{2}\right)^{-3} \sin \frac{\omega\tau}{2} \right.$$

$$\left. - 30 \left(\frac{\omega\tau}{2}\right)^{-2} \cos \frac{\omega\tau}{2} -12 \left(\frac{\omega\tau}{2}\right)^{-1} \sin \frac{\omega\tau}{2} + 2 \cos \frac{\omega\tau}{2} \right\}$$

$$H_4{}^{o}(\omega) = (17!\ \tau)^{1/2} \frac{4!}{8!} (\omega\tau)^{-5} \left\{ 210 \left(\frac{\omega\tau}{2}\right)^{-4} \sin \frac{\omega\tau}{2} \right.$$

$$- 210 \left(\frac{\omega\tau}{2}\right)^{-3} \cos \frac{\omega\tau}{2} -90 \left(\frac{\omega\tau}{2}\right)^{-2} \sin \frac{\omega\tau}{2}$$

$$\left. + 20 \left(\frac{\omega\tau}{2}\right)^{-1} \cos \frac{\omega\tau}{2} + 2 \sin \frac{\omega\tau}{2} \right\}$$

The $B\tau$ product for $h_n^{\,o}(t)$ is given by

$$\frac{1}{\left(\int_0^{\tau} h_n^{\,o}(t)\ dt\right)^2}\ \tau$$

$$= \frac{(2n+1)!}{(4n+1)!}\ \left\{\frac{n!}{(2n)!}\right\}^2$$

For n from 0 to 4,

n	0	1	2	3	4
$B\tau$	1	1.200	1.429	1.630	1.814

The spectrum of these five beta functions are shown in Figure 4.2-4.6.

Case 2 $\qquad$ m = 2

The energy of $h_b(t)$, by Parseval's theorem, is given by

$$R\{h_b(t)\} = \int_{-\infty}^{\infty} h_b^{\,2}(t)\ dt$$

$$= \int_{-B_{2,n}/2}^{B_{2,n}/2} |H_b(f)|^2\ df$$

$$= |H_b(0)|^2\ B_{2,n}$$

This, together with the expression for $Q\{h_b(t)\}$ which is the same as the one for m = 1, gives

$$\beta_{2,n} = \tau\ B_{2,n}$$

$$= 2\tau \left\{ (n + \frac{1}{2}) \frac{Q\{h_n{}^o(t)\}}{R\{h_n{}^o(t)\}} \right\}^{1/2n}$$

$$= 2\tau\ \{(n + \frac{1}{2})\ \lambda\}^{1/2n}$$

The differential equation to be solved is

$$\frac{d^{2n}}{dt^{2n}} h(t) = (-1)^n\ \lambda h(t)$$

For n = 1,

$$h_1{}^o(t) = \begin{cases} \sqrt{\frac{2}{\tau}} \cos \frac{\pi t}{\tau} & \text{for } -\tau/2 \leqq t \leqq \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

The half cosine pulse shape, when used with one quadrature staggered by T/2, forms the well-known minimum shift key (MSK) modulation. The spectrum of this pulse shape is

$$H_1{}^o(\omega) = \left(\frac{8\tau}{\pi^2}\right)^{1/2} \frac{\cos \frac{\omega\tau}{2}}{1 - (\omega\tau/\pi)^2}$$

with

$$B\tau = \frac{1}{\left|H_1{}^o(0)\right|^2} \cdot \tau = \frac{\pi^2}{8} = 1.235$$

The value of λ is π^2/τ^2, consequently giving

$$\beta_{2,1} = 2\sqrt{\frac{3}{2}}\ \pi = 7.695$$

For $n = 2$, the eigenvalues of the differential equation are $\sqrt[4]{\lambda}$, $-\sqrt[4]{\lambda}$, $j\sqrt[4]{\lambda}$, $-j\sqrt[4]{\lambda}$. Defining

$$\alpha = \frac{\tau}{2}\sqrt[4]{\lambda}$$

so that we may express

$$h_2^{o}(t) = A_1 \cosh\frac{2\alpha t}{\tau} + A_2 \sinh\frac{2\alpha t}{\tau} + A_3 \cos\frac{2\alpha t}{\tau} + A_4 \sin\frac{2\alpha t}{\tau}$$

The four boundary conditions give

$$\begin{cases} A_1 \cosh\alpha + A_2 \sinh\alpha + A_3 \cos\alpha + A_4 \sin\alpha = 0 \\ A_1 \cosh\alpha - A_2 \sinh\alpha + A_3 \cos\alpha - A_4 \sin\alpha = 0 \end{cases}$$

$$\Rightarrow \begin{cases} A_1 \cosh\alpha + A_3 \cos\alpha = 0 \\ A_2 \sinh\alpha + A_4 \sin\alpha = 0 \end{cases}$$

and

$$\begin{cases} A_1 \sinh\alpha + A_2 \cosh\alpha - A_3 \sin\alpha + A_4 \cos\alpha = 0 \\ -A_1 \sinh\alpha + A_2 \cosh\alpha + A_3 \sin\alpha + A_4 \cos\alpha = 0 \end{cases}$$

$$\Rightarrow \begin{cases} A_2 \cosh\alpha + A_4 \cos\alpha = 0 \\ A_1 \sinh\alpha - A_3 \sin\alpha = 0 \end{cases}$$

In matrix form, we have

$$\begin{pmatrix} \cosh\alpha & \cos\alpha & 0 & 0 \\ \sinh\alpha & -\sin\alpha & 0 & 0 \\ 0 & 0 & \sinh\alpha & \sin\alpha \\ 0 & 0 & \cosh\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} A_1 \\ A_3 \\ A_2 \\ A_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

should there be a nontrivial solution, either one of the sub-determinants

$$\Delta_1 = \begin{vmatrix} \cosh\alpha & \cos\alpha \\ \sinh\alpha & -\sin\alpha \end{vmatrix}, \quad \Delta_2 = \begin{vmatrix} \sinh\alpha & \sin\alpha \\ \cosh\alpha & \cos\alpha \end{vmatrix}$$

must equals zero, giving

$$\tan \alpha = \pm \tanh \alpha$$

The smallest positive solution for this transcendental equation is $\alpha = 2.365$ when $\Delta_1 = 0$, for which case $\Delta_2 \neq 0$ would give A_2 and A_4 the trivial solution. Since

$$\frac{A_1}{A_3} = \frac{\sin \alpha}{\sinh \alpha}$$

we may assume

$$A_1 = k \sin \alpha$$

$$A_3 = k \sinh \alpha$$

in which k normalizes the energy of $h_2^{\,o}(t)$. After some numerical computation, we have

$$h_2^{\,o}(t) = 0.1863 \cosh \frac{4.73}{\tau} t + 1.4022 \cos \frac{4.73}{\tau} t$$

The value of $\beta_{2,2}$ is 11.9, and the $B\tau$ product is 1.45.

The solution for general n is suspected to be an even function of the form

$$h_n^{\,o}(t) = \sum_{k=1}^{\ell} A_k \cosh \alpha_k t \cos \beta_k t + B_k \sinh \alpha_k t \sin \beta_k t$$

in which

$$\alpha_k + j\beta_k \quad (\alpha_k \geqq 0,\ \beta_k \geqq 0)$$

is one of the 2n-th root of $(-1)^n \lambda$ (the smallest λ of course), ℓ the number of such roots (in the first quadrant and on the positive real as well as imaginary axes) and A_k, B_k are found by matching boundary conditions. For convenience sake, these pulse shapes will be called trigonometric-hyperbolic functions.

For m = 2 and n = 1, 2, $h_n^{\,o}(t)$ are plotted in Figures 4.7, 4.8, and their Fourier transforms in Figures 4.9, 4.10.

Appendix B. Code Searching Algorithms for Rate 2/3 Coded 8ϕ

The encoders are represented by subgenerators such as

$$g_i^a = (g_{i,0}^a,\ g_{1,1}^a\ .\ .\ .\ g_{i,\ell}^a) \qquad i = 1,2$$

in which $g_{i,j}^a$ is the tap gain from the j-th register of the i-th queue to the adder A (Figure B.1). ℓ denotes the number of memories of a queue. Each subgenerator will be interchangeably expressed by its integer representation, such as

$$g_i^a = g_{i,0}^a\ m^{\ell-1} + g_{i,1}^a\ m^{\ell-2} + \ .\ .\ .\ + g_{i,\ell}^a$$

where m is the number of elements in the set V. The addition of two subgenerators is given by the element-wise adding of the two subgenerators. A subgenerator is larger than another subgenerator by virtue of its integer representation. Two encoders are said to be similar if they have the same minimum free Euclidean distance.

B.1 BINARY ENCODERS WITH STRAIGHT BINARY MAPPING

Let queue 1 has $\ell = \eta$ memories and queue 2 has $\ell = \mu$ memories. Then the encoder has a total of $\gamma = \eta + \mu$ memories. There are $3(\gamma + 2)$ taps and investigating each possible tap combination becomes prohibitive for $\gamma > 4$. A number of rejection rules, based on the structural similarity of encoders and conjectures about tap patterns for good encoders, would serve to limit the computation requirements effectively.

In the code searching algorithm, the subgenerators are incremented by nested loops, from the innermost to the outermost according to the order

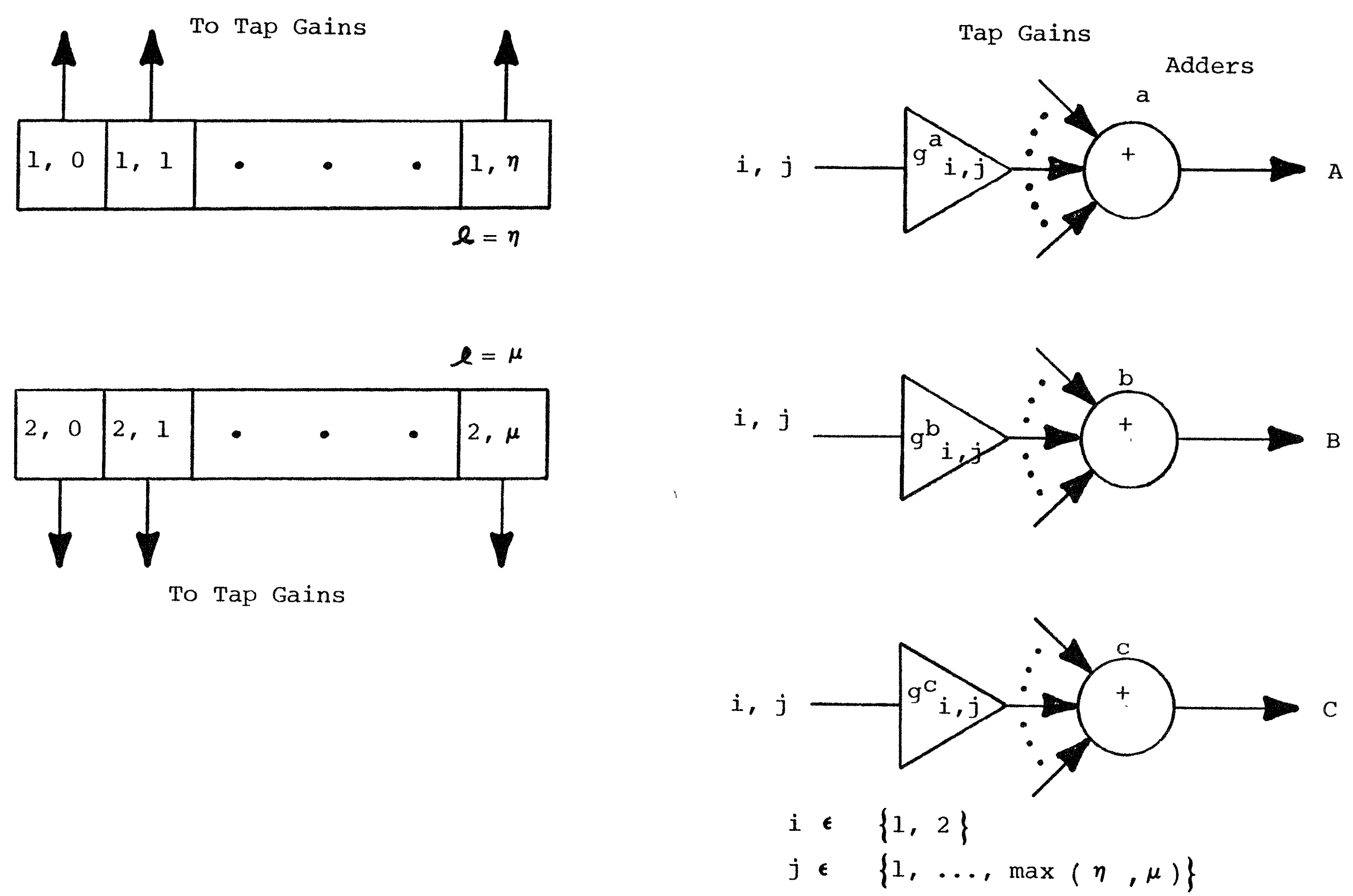


Figure B.1. A General Convolutional Rate 2/3 Encoder

$$g_2^a,\ g_1^a,\ g_2^b,\ g_1^b,\ g_2^c,\ g_1^c\ .$$

In other words, g_2^a is incremented the most often and g_1^c the least often. The rejection rules used include,

Rule 1: The first and last register of any queue must each be connected to at least one of the adders. If this condition is not satisfied, the encoder can be rejected since there is an equivalent encoder with shorter constraint length.

Rule 2: Time reversal does not change the distance properties of an encoder. Therefore, an encoder is similar to another encoder with reversed subgenerators. A reversed subgenerator is given by

$$g_{i\ rev} = g_{i,\ell}\ 2^{\ell} + g_{i,(\ell-1)} 2^{\ell-1} + \ldots g_{i,0}$$

If the outermost nonzero subgenerator g_i satisfies $g_i > g_{i\ rev}$, the encoder defined by the loop indices has a time reversed version which has been considered previously. Therefore, that value of g_i can be skipped.

Rule 3: Ungerboeck [10] conjectured that for good encoders, the adder C, which outputs the least significant bit, is not connected to the present inputs, which is to say

$$g_{1,0}^c = g_{2,0}^c = 0$$

The state of the encoder determines the value of C and hence which of the sets {0, 2, 4, 6} or {1, 3, 5, 7} v(=4A + 2B + C) belongs to. If we adopt this restriction on the encoder, the taps $g_{1,\eta}^c$ and $g_{2,\mu}^c$ can also be set to zero by the time reversal argument. Such a restriction is rather difficult to justify but it was adopted in our program for its 16 fold reduction in computation requirement.

Rule 4: An encoder with $\eta = \mu$ is equivalent to the encoder obtained by exchanging the two queues. In other words, it is immaterial to exchange the subgenerators g_1^a, g_1^b, g_1^c with g_2^a, g_2^b, g_2^c in that order. By consideration of the order of looping in the program, an encoder can be rejected if

$$g_1^a \, 2^{\eta+1} + g_2^a > g_1^a + g_2^a \, 2^{\eta+1}$$

Rule 5: Consider encoders generated by exclusive-ORing subgenerators. If v = (A, B, C) is the output of the encoder shown in Figure B.1, we may obtain v' = (A ⊕ B, B, C) by replacing the subgenerator g_1^a by $g_1^a \oplus g_1^b$ and g_2^a by $g_2^a \oplus g_2^b$. In particular, we will consider the following transformations.

i. (A, B, C) → (A ⊕ B, B, C)

or by expressing v and v' by straight binary conversion,

v	0	1	2	3	4	5	6	7
→v'	0	1	6	7	4	5	2	3

ii. (A, B, C) → (A ⊕ C, B, C)

or equivalently

v	0	1	2	3	4	5	6	7
→v'	0	5	2	7	4	1	6	3

iii. (A, B, C) → (A, B ⊕ C, C)

or equivalently

v	0	1	2	3	4	5	6	7
→v'	0	3	2	1	4	7	6	5

iv. $(A, B, C) \rightarrow (A \oplus B, B \oplus C, C)$

or equivalently

v	0	1	2	3	4	5	6	7
$\rightarrow v'$	0	3	6	5	4	7	2	1

v. $(A, B, C) \rightarrow (A \oplus C, B \oplus C, C)$

or equivalently

v	0	1	2	3	4	5	6	7
$\rightarrow v'$	0	7	2	5	4	3	6	1

vi. $(A, B, C) \rightarrow (A \oplus B \oplus C, B, C)$

or equivalently

v	0	1	2	3	4	5	6	7
$\rightarrow v'$	0	5	6	3	4	1	2	7

vii. $(A, B, C) \rightarrow (A \oplus B \oplus C, B \oplus C, C)$

or equivalently

v	0	1	2	3	4	5	6	7
$\rightarrow v'$	0	7	6	1	4	3	2	5

For all these transformations, each element of the triple is replaced by adding itself to those elements to the right. In other words, only B or C can be added to A, and C to B. Using the free Euclidean distance bound obtained in Section 5.2, it is observed that for these transformations, $M_b(v) = M_b(v')$ for $v = 0$, 2, 4, 6 and the corresponding v'. For $v = 1$ (or 3,7), v' can become 5 (recall that $M_b(5) = 1.707$ while $M_b(1) = M_b(3) = M_b(7) = 0.293$) under some transformations. Therefore, the above transformations (8 altogether if we include the identity transformation $(A, B, C) \rightarrow (A, B, C)$) actually represent four different cases of whether $v = 1$, 3, 5 or 7 is transformed into 5. Thus, computation can be reduced by a factor of 2. In fact, we cut the computation requirement by a factor of 8 by ignoring all transformations other than the identity transformation. This restriction is justified experimentally when removing the restriction for encoders of a small number of memories did not yield encoders with better minimum free distance.

Rule 6: The best d_f discovered so far is remembered and an encoder is rejected immediately when a free distance of less than d_f is revealed.

There may be some other hidden symmetries which would give additional rejection rules. The best encoder found may not be optimal (though we strongly suspect that would not be the case) since some of the rejection rules have not been rigorously proven.

A computer search program, which evaluates the Euclidean distance of rate 2/3 coded 8ϕ encoders is listed on the following pages.

```
      CODE8P.FORT
C     THIS PROGRAM CHECKS THE FREE EUCLIDAAN DISTANCE OF ANY BINARY
C     RATE 2/3 ENCODER USING THE VITERBI ALGORITHM.
C
C     ILINK STORES WHERE THE DIFFERENT BRANCHES MERGING INTO A STATE
C     COME FROM. RMET STORES THE OCTAL OUTPUT V ASSOCIATED WITH EACH
C     BRANCH.IOUT STORES THE TRIPLE (A,B,C). DIST STORES THE METRIC AT
C     EACH STATE,WHEREAS DNEW SERVES AS A TEMPORARY STORAGE FOR DIST.
C     ITAP(I,J) IS THE JTH INPUT TO THE ADDER I. ISTORE IS A TEMPORARY
C     STORAGE FOR CONVERTING DECIMALS TO BINARY NUMBERS.
C
      DIMENSION RMET(64,4),DIST(64),ITAP(3,8),DNEW(64),ILINK(64,4)
      DIMENSION IOUT(3),EUCLBD(8)
C
C     EUCLBD IS THE BOUND FOR THE EUCLIDEAN FREE DISTANCE FOR EACH
C     CHANNEL SYMBOL V
C
      DATA EUCLBD/0.,.293,1.,.293,2.,1.707,1.,.293/
      COMMON ISTORE(8)
C
C     DMIN IS THE MINIMUM FREE DISTANCE FOUND SO FAR.THE CONSTRAINT
C     LENGTH AND THE SUBGENERATOR POLYNORMIAL (SUBSEQUENTLY CONVERTED
C     TO BINARY FORM) IS REQUESTED.
C
      DMIN=-1.
      WRITE(6,122)
 122  FORMAT(1X,'INPUT N AND U, THE NUMBER OF MEMORIES IN EACH QUEUE')
      READ*,KN,KU
      K=KN+KU
      WRITE(6,121)
 121  FORMAT(1X,'INPUT THE TAP GAINS TO THE ADDER A,IN DECIMAL FORM')
      READ*,I1
      WRITE(6,123)
 123  FORMAT(1X,'INPUT THE TAP GAINS TO THE ADDER B')
      READ*,I2
      WRITE(6,124)
 124  FORMAT(1X,'INPUT THE TAP GAINS TO THE ADDER C')
      READ*,I3
      CALL CBIN(I1)
      DO 3 I=1,8
      ITAP(1,I)=ISTORE(I)
 3    CONTINUE
      CALL CBIN(I2)
      DO 5 I=1,8
      ITAP(2,I)=ISTORE(I)
 5    CONTINUE
      CALL CBIN(I3)
      DO 42 I=1,8
      ITAP(3,I)=ISTORE(I)
 42   CONTINUE
```

```
C
C       THE TABLES FOR THE ILINK AND RMET ARE BEING FILLED. EACH I
C       DENOTE ONE OF THE STATES. EACH J DENOTES ONE OF THE BRANCHES
C       GOING INTO THE STATE I. THE METRIC TABLE DIST IS INITITIALIZED
C       WITH LARGE VALUES.
C
        IK=2**K
        DO 6 I=1,IK
        DIST(I)=100.
        DO 7 J=1,4
C
C       THE PREVIOUS STATE LINKED BY THE J-TH BRANCH IS FOUND. THE
C       ENCODER OUTPUTS ARE FOUND AND USED TO COMPUTE THE OCTAL V
C       ASSOCIATED WITH THE BRANCH TRANSITION
C.
        IQ1=(I-1)/2**KU
        IQ2=I-1-IQ1*2**KU
        IQ1=IQ1*2+(J-1)/2
        IQ2=IQ2*2+J-1-((J-1)/2)*2
        ILINK(I,J)=MOD(IQ1,2**KN)*2**KU+MOD(IQ2,2**KU)+1
        IOUT(1)=0
        IOUT(2)=0
        IOUT(3)=0
        IQ=IQ1*2**(KU+1)+IQ2
        CALL CBIN(IQ)
        IA=7-K
        DO 8 L1=1,3
        DO 9 L2=IA,8
        IOUT(L1)=IOUT(L1)+ITAP(L1,L2)*ISTORE(L2)
   9    CONTINUE
   8    CONTINUE
        IOUT(1)=MOD(IOUT(1),2)
        IOUT(2)=MOD(IOUT(2),2)
        IOUT(3)=MOD(IOUT(3),2)
        IOCTAL=IOUT(1)*4+IOUT(2)*2+IOUT(3)
        RMET(I,J)=EUCLBD(IOCTAL+1)
   7    CONTINUE
   6    CONTINUE
C
C       TRELLIS SEARCH FOR MINIMUM DISTANCE PATH
C       DOO DENOTES THE MINIMUM DISTANCE AMONGST PARALLEL TRANSITIONS.
C       DSHORT IS THE SHORTEST EUCLIDEAN SEPARATION FOUND SO FAR.
C       DLEASP REPRESENT THE SMALLEST METRIC AMONGST ALL THE STATES AT
C       A DECODING STAGE. ICOUNT IS THE NUMBER OF STAGES THE ALGORITHM
C       HAS GONE THROUGH.
C
```

```
      DOO=100.
      DO 46 J=2,4
      IF (ILINK(1,J).EQ.1) DOO=AMIN1(DOO,RMET(1,J))
 46   CONTINUE
      DSHORT=DOO
      DIST(1)=0
      DNEW(1)=1000.
      ICOUNT=1
 17   DLEAST=100.
      ICOUNT=ICOUNT+1
      IF (ICOUNT.EQ.100) GO TO 41
C
C     A SURVIVOR IS CHOSEN AMONGST THE 4 BRANCHES GOING INTO A STATE
C
      DO 18 I=2,IK
      DNEW(I)=DIST(ILINK(I,1))+RMET(I,1)
      DO 19 J=2,4
      DNEW(I)=AMIN1(DIST(ILINK(I,J))+RMET(I,J),DNEW(I))
 19   CONTINUE
C
C     THE STATE WITH THE SMALLEST METRIC IS FOUND.
C
      DLAAST=AMIN1(DLEAST,DNEW(I))
 18   CONTINUE
C
C     THE METRIC TABLE IS BEING UPDATED
C
      DO 20 I=1,IK
      DIST(I)=DNEW(I)
 20   CONTINUE
C
C     THE THREE BRANCHES (J=2 TO 4) THAT MERGES INTO THE ALL ZERO
C     STATE IS COMPARED TO SEE WHICH ONE GIVES THE SHORTEST DRUN AT
C     THAT STAGE. IF DRUN IS LESS THAN THE SHORTEST FREE DISTANCE OF
C     THE ENCODER (DSHORT) FOUND SO FAR, DSHORT WOULD BE UPDATED.
C
      DRUN=1000.
      DO 21 J= 2,4
      DRUN=AMIN1(DIST(ILINK(1,J))+RMET(1,J),DRUN)
 21   CONTINUE
      IF (DSHORT-DRUN.GT.-0.00001) GO TO 22
      GO TO 24
 22   DSHORT=DRUN
C
C     IF DSHORT IS LESS THAN THE DMIN FOUND FOR PREVIOUS ENCODERS, THEN
C     THE ENCODER CONSIDERED RIGHT NOW IS NO GOOD. IF EVERY STATE HAS A
C     METRIC (THE SMALLEST OF WHICH IS DLEAST) LARGER THAN DSHORT, THEN
C     IT IS NOT NECESSARY TO GO TO FURTHER STAGES TO FIND THE MINIMUM
C     FREE DISTANCE. DMIN FOR THE ENCODER IS EQUAL TO DSHORT.
C
```

```
 24     IF((DMIN-DSHORT).GT.0.00001) GO TO 41
        IF ((DSHORT-DLEAST).GT.0.00001) GO TO 17
        IF (DSHORT.LT.DOO) DMIN=DSHORT
        WRITE(6,125)
 125    FORMAT(1X,'THE TAP GAINS FOR YOUR ENCODER FOR THE ADDERS A,B,C
      & ARE RESPECTIVELY:')
        DO 126 L3=1,3
 126    WRITE(6,127) (ITAP(L3,L4),L4=IA,8)
 127    FORMAT(1X,10I3)
 41     WRITE(6,128) DMIN
 128    FORMAT(1X,'THE MINIMUM FREE DISTANCE IS',1X,F10.5)
        STOP
        END
C
C       THE SUBROUTINE CBIN CONVERTS A DECIMAL NUMBER INTO A BINARY
C       NUMBER.
C
        SUBROUTINE CBIN(IDEC)
        COMMON ISTORE(8)
        IQUOT=IDEC
        DO 1 I=1,8
        ISTORE(I)=IQUOT/2**(8-I)
        IQUOT=IQUOT-ISTORE(I)*2**(8-I)
 1      CONTINUE
        RETURN
        END
```

```
call code8p
TEMPNAME ASSUMED AS MEMBERNAME
INPUT N AND U, THE NUMBER OF MEMORIES IN EACH QUEUE
?
2  2
INPUT THE TAP GAINS TO THE ADDER A,IN DECIMAL FORM
?
21
INPUT THE TAP GAINS TO THE ADDER B
?
57
INPUT THE TAP GAINS TO THE ADDER C
?
2
THE TAP GAINS FOR YOUR ENCODER FOR THE ADDERS A,B,C    ARE RESPECTIVELY:
  0  1  0  1  0  1
  1  1  1  0  0  1
  0  0  0  0  1  0
THE MINIMUM FREE DISTANCE IS     2.58600
READY
```

B.2 OCTAL CONVOLUTIONAL ENCODERS

A similar set of rejection rules can probably be deduced for the octal convolutional encoders. However, the plain fact that there are $8^{3(\gamma+2)}$ ($\sim 7 \times 10^{10}$ for $\gamma = 2$) possible tap combinations would deny exhaustive search even if the rejection rules are powerful enough to reduce the effort by four or five orders of magnitude. Instead, we shall employ a different tactic for code searching, which randomizes the code search within a small class of promising candidates. This technique enables us to obtain an encoder with reasonable d_f within a much shorter period of computation. This technique can be used similarly for searching other types of convolutional encoders. The randomization avoids a lot of computation waste due to equivalence patterns. Imagine tasting a large variety of cookies in a box. By picking at random, it is rather unlikely that one would repeatedly taste the same flavor, though it is also unlikely that one would be able to pick the best flavor. On the other hand, a systematic picking may coincide with the way cookies of the same flavor are arranged.

The class of encoders which will be considered consists of those encoders which achieves the largest minimum free distance when the input error sequence is restricted to have one nonzero entry only. This restricted d_f achieved is usually very close to the upper bound derived in Section 5.3.

Since the error sequence has only one nonzero entry, we may restrict our attention to the tap gains of only one of the queues. The tap gains of the other queue can be generated independently and similarly. The tap gains of concern for $\gamma = 2$ are $(g^a_{1,0}, g^a_{1,1}, g^b_{1,0}, g^b_{1,1}, g^c_{,1,0}, g^c_{1,2})$. The restricted d_f would not be altered by conjugating any element of this 6-tuple (the conjugate of i is 8-i) or by pairwise interchanging any two of the values. By a thorough computer search, the only 6-tuples which achieve a maximum restricted d_f with each element of the 6-tuple having a value from 1 to 4 and

$$g^a_{1,0} \leqq g^a_{1,1} \leqq g^b_{1,0} \leqq g^b_{1,1} \leqq g^c_{1,0} \leqq g^c_{1,1}$$

are

(1, 1, 1, 3, 3, 3)

(1, 1, 2, 2, 3, 3)

(1, 1, 2, 2, 3, 4)

(1, 1, 2, 3, 3, 4)

(1, 2, 2, 3, 3, 4)

The code search algorithm picks any two (or the same) 6-tuples at random and exchange randomly two of the entries within each chosen 6-tuples, conjugating the entries during the exchange. The two 6-tuples now defines an encoder. To further reduce the candidates of encoders, error sequences with one nonzero entry fed simultaneously into each of the two queues are passed into the encoder. Again, only those encoders with the maximum achieved d_f for the double error sequences are retained. The remaining encoders are then trellis searched for the unrestricted d_f. Through such a process, a large proportion of encoders is rejected since they cannot survive the occurrences of these error sequences which most likely induce the minimum free distance.

The documented computer algorithm is listed on the following pages.

octal23.fort

```
C       THIS PROGRAM SEARCHES FOR GOOD RATE 2/3 ENCODERS WITH 2 OCTAL
C       MEMORIES.
C
C       DIST STORES THE CUMULATED METRIC OF EACH OF THE 64 STATES, AND
C       DNEW IS USED AS A TEMPORARY STORAGE FOR DIST. THERE ARE 64 BRANCH
C       GOING INTO EACH STATE, AND THE ENCODER OUTPUT OF EACH BRANCH IS
C       DENOTED BY RMET. THE ENCODER OUTPUT IS A FUNCTION OF THE TAP GAIN
C       WHICH IS STORED IN ITAP.THE THREE OCTAL OUTPUT OF THE OCTAL
C       ENCODER IS GIVEN BY IOUT. EUC(I) DENOTES THE EUCLIDEAN DISTANCES
C       BETWEEN THE CHANNEL SYMBOL SEQUENCE I AND 0. ICODE IS THE POOL OF
C       SUBGENERATORS WHERE THE ITAPS GET THEIR VALUES.
C
        DIMENSION RMET(64,64),DIST(64),ITAP(4,3),ICODE(10,6)
        DIMENSION IREG(4),IOUT(3),DNEW(64),EUC(8),IC(2)
        DOUBLE PRECISION DSEED
        COMMON ISTORE(4)
        DATA EUC /0.,.293,1.,1.707,2.,1.707,1.,.293/
        DATA ICODE /1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,2,
     &              1,2,2,2,2,1,2,2,2,2,3,2,2,3,3,3,2,2,3,3,
     &              3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,3,3,4,4,4/
        WRITE(6,222)
  222   FORMAT('  ENTER A SEED FOR THE RANDOM NUMBER GENERATOR')
        READ*,DSEED
C
C       BY INVOKING THE RANDOM NUMBER GENERATOR, TWO SUBGENARATORS
C       ARE PICKED FROM THE IC(1) AND IC(2) ROWS OF THE POOL OF SUBGEN-
C       ERATORS. THEN THE IC3 AND IC3 LOCATIONS OF EACH SUBGENERATORS
C       ARE EXCHANGED AND CONJUGATED AT THE SAME TIME. THUS THE POOL OF
C       SUBGENERATORS IS CONSTANTLY VARIED.
C
  50    IC(1)=GGUBFS(DSEED)*9.99999+1
        IC(2)=GGUBFS(DSEED)*9.99999+1
        IF (IC(1).EQ.IC(2)) GO TO 50
        DO 51 I=1,2
        IC3=GGUBFS(DSEED)*5.99999+1
        IC4=GGUBFS(DSEED)*5.99999+1
        ITEMP=ICODE(IC(I),IC3)
        ICODE(IC(I),IC3)=8-ICODE(IC(I),IC4)
        ICODE(IC(I),IC4)=8-ITEMP
  51    CONTINUE
C
C       THE ENCODER PICKED RANDOMLY IS THEN TESTED WITH ERROR SEQUENCES
C       WITH ONE ERROR FED SIMULTANEOUSLY INTO EACH QUEUE OF THE ENCODER.
C       ANY ERROR PATH HAVINC DISTANCE LESS THAN 3.9 IS REJECTED.
C
```

```
      SMPAIR=10.
      DO 52 J1=1,7
      DO 53 J2=1,4
      SMTEMP=0.
      DO 54 J3=1,6
      SMTEMP=SMTEMP+EUC(MOD(J1*ICODE(IC(1),J3)+J1*ICODE(IC(2),J3),8)+1)
   54 CONTINUE
      IF (SMTEMP.LT.3.9) GO TO 50
      SMPAIR=AMIN1(SMPAIR,SMTEMP)
   53 CONTINUE
   52 CONTINUE
C
C     ITAP IS NOW READ FROM ICODE. THE VITERBI ALGORITHM WILL BE USED
C     TO FIND THE MINIMUM DISTANCE OF THE ENCODER.
C
      ICODNM=ICODNM+1
      ITAP(1,1)=ICODE(IC(1),1)
      ITAP(2,1)=ICODE(IC(1),2)
      ITAP(3,1)=ICODE(IC(2),1)
      ITAP(4,1)=ICODE(IC(2),2)
      ITAP(1,2)=ICODE(IC(1),3)
      ITAP(2,2)=ICODE(IC(1),4)
      ITAP(3,2)=ICODE(IC(2),3)
  3   ITAP(4,2)=ICODE(IC(2),4)
      ITAP(1,3)=ICODE(IC(1),5)
      ITAP(2,3)=ICODE(IC(1),6)
      ITAP(3,3)=ICODE(IC(2),5)
      ITAP(4,3)=ICODE(IC(2),6)
      DSHORT=1000.
C
C     THE ENCODER IS SIMULATED SO THAT THE OUTPUTS AS A FUNCTION OF THE
C     CONTENT OF THE SHIFT REGISTERS IS FOUND.
C
      DO 6 I=1,64
      DIST(I)=100.
      DO 7 J=1,64
      IOUT(1)=0
      IOUT(2)=0
      IOUT(3)=0
      IREG(1)=(I-1)/8
      IREG(2)=(J-1)/8
      IREG(3)=(I-1)-IREG(1)*8
      IREG(4)=(J-1)-IREG(2)*8
      DO 8 L1=1,3
      DO 9 L2=1,4
      IOUT(L1)=MOD(IOUT(L1)+IREG(L2)*ITAP(L2,L1),8)
    9 CONTINUE
    8 CONTINUE
      RMET(I,J)=EUC(IOUT(1)+1)+EUC(IOUT(2)+1)+EUC(IOUT(3)+1)
    7 CONTINUE
    6 CONTINUE
```

```
C
C       THIS IS THE BEGINNING OF THE TRELLIS SEARCH. ICOUNT COUNTS THE
C       NUMBER OF STAGES THE VITERBI ALGORITHM HAS PERFORMED.
C
        DIST(1)=0.
        DNEW(1)=1000.
        ICOUNT=0
 17     DLEAST=100.
        ICOUNT=ICOUNT+1
        IF (ICOUNT.EQ.10) GO TO 50
C
C       THE 64 PATHS MERGING INTO A STATE IS COMPARED AND THE SURVIVOR IS
C       PICKED. THE SURVIVOR GOING BACK TO THE ZERO STATE GIVES ONE OF
C       THE FREE DISTANCES (DRUN), WHICH IS COMPARED WITH THE PREVIOUS
C       MINIMUM FREE DISTANCES (DSHORT) AND UPDATES DSHORT IF DRUN IS
C       SMALLER THAN DSHORT. DLEAST REGISTERS THE SMALLEST ACCUMULATED
C       METRIC OF THE 64 STATES, AND IF DLEAST IS GREATER THAN DSHORT,
C       TRELLIS SEARCH FOR THE MINIIUM DISTANCE PATH OF THE ENCODER HAS
C       BEEN ACCOMPLISHED.
C
        DO 18 I=2,64
        DNEW(I)=DIST(1)+RMET(I,1)
        DO 19 J=2,64
        DNEW(I)=AMIN1(DIST(J)+RMET(I,J),DNEW(I))
 19     CONTINUE
        DLEAST=AMIN1(DLEAST,DNEW(I))
 18     CONTINUE
        DO 20 I=1,64
        DIST(I)=DNEW(I)
 20     CONTINUE
        DRUN=1000.
        DO 21 I=2,64
        DRUN=AMIN1(DIST(I)+RMET(1,I),DRUN)
 21     CONTINUE
        IF (DSHORT-DRUN.LT.0.000001) GO TO 24
        DSHORT=DRUN
C
C       DMIN IS THE FREE DISTANCE OF THE BEST ENCODER FOUND SO FAR. IF
C       DSHORT IS LESS THAN DMIN, THEN THE PRESENT ENCODER CAN BE
C       ABANDONED. AFTER WE FINISH THE TRELLIS SEARCH FOR THE ENCODER
C       AND THE MINIMUM FREE DISTANCE IS LARGER THAN DMIN, THUS WE HAVE
C       FOUND A BETTER ENCODER,WHICH IS PRINTED AT THE TERMINAL.
C.
```

```
24      IF ((DMIN-DSHORT).GT.0.00001) GO TO 50
        IF ((DSHORT-DLEAST).GT.0.00001) GO TO 17
        DMIN=DSHORT
        WRITE(6,240)
240     FORMAT('     X------X------X---------X---------X--------X')
        DO 224  I=1,3
        WRITE(6,227) I
227     FORMAT('  TAP GAINS TO THE ADDER ',I5,' ARE:')
        WRITE(6,223) (ITAP(L3,I),L3=1,4)
223     FORMAT(4I5)
224     CONTINUE
        WRITE(6,225) DMIN
225     FORMAT('  MINIMUM FREE DISTANCE= ',F10.5)
        WRITE(6,226) ICODNM
226     FORMAT('  NUMBER OF ENCODERS TRELLIS-SEARCHED SO FAR= ',I6)
        IF (ICODNM.LT.10000) GO TO 50
        STOP
        END
```

```
call octal23
TEMPNAME ASSUMED AS MEMBERNAME
 ENTER A SEED FOR THE RANDOM NUMBER GENERATOR
?
1234.d0
   X-----X-----X-----X-----X-----X
 TAP GAINS TO THE ADDER     1 ARE:
   1    7    1    6
 TAP GAINS TO THE ADDER     2 ARE:
   6    6    5    6
 TAP GAINS TO THE ADDER     3 ARE:
   4    5    3    1
 MINIMUM FREE DISTANCE=    2.05100
 NUMBER OF ENCODERS TRELLIS-SEARCHED SO FAR=      4
   X-----X-----X-----X-----X-----X
 TAP GAINS TO THE ADDER     1 ARE:
   1    7    4    2
 TAP GAINS TO THE ADDER     2 ARE:
   3    2    3    3
 TAP GAINS TO THE ADDER     3 ARE:
   2    5    7    6
 MINIMUM FREE DISTANCE=    2.05100
 NUMBER OF ENCODERS TRELLIS-SEARCHED SO FAR=      9
   X-----X-----X-----X-----X-----X
 TAP GAINS TO THE ADDER     1 ARE:
   6    7    1    2
 TAP GAINS TO THE ADDER     2 ARE:
   3    6    5    5
 TAP GAINS TO THE ADDER     3 ARE:
   1    4    4    6
 MINIMUM FREE DISTANCE=    2.29300
 NUMBER OF ENCODERS TRELLIS-SEARCHED SO FAR=     18
@!
READY
```

```
      OCTAL12.FORT'
C     THIS PROGRAM SEARCHES EXHAUSTIVELY FOR OPTIMAL RATE 1/2 OCTAL
C     CONVOLUTIONAL ENCODERS.
C
C     DIST STORES THE CUMULATED METRIC OF THE STATES OF THE ENCODER
C     WITH K-1 OCTAL MEMORIES,AND DNEW IS USED AS A TEMPORARY STORAGE
C     FOR DIST. THE EUCLIDEAN DISTANCE OF THE OUTPUT OF THE J-TH
C     BRANCH MERGING INTO THE I-TH STATE IS GIVEN BY RMET(I,J). THE
C     ENCODER OUTPUT (IOUT)  IS A FUNCTION OF THE TAP GAINS, WHICH ARE
C     STORED IN ITAP. EUC(I) DENOTES THE EUCLIDEAN DISTANCES BETWEEN
C     THE CHANNEL SYMBOL I AND O. ILINK(I,J) DENOTES THE PREVIOUS
C     STATE CONNECTING TO THE STATE I THROUGH THE J-TH BRANCH MERGING
C     INTO STATE I.
C
      DIMENSION RMET(64,8),DIST(64),ITAP(3,2),EUC(8)
      DIMENSION ILINK(60,8),IOUT(2),DNEW(64)
      DATA EUC /0.,.293,1.,1.707,2.,1.707,1.,.293/
      COMMON ISTORE(3),K
      WRITE(6,130)
130   FORMAT('  PUT IN THE NUMBER OF OCTAL MEMORIES OF THE ENCODER.')
      READ*,M
      K=M+1
      IF (M.EQ.2) WRITE(6,139)
C
C     IL AND IU REPRESENTS THE SUBGENERATORS FOR THE ENCODER. IL AND
C     ARE CONVERTED TO OCTAL REPRESENTATION AND STORED AS TAP GAINS.
C
      IL=8**(K-1)
      IU=8**K-1
      DMIN=0.
      DO 2 I1=1L,IU
      CALL OCTAL(I1)
      DO 3 I=1,3
      ITAP(I,1)=ISTORE(I)
3     CONTINUE
      DO 4 I2=1,IU
      CALL OCTAL(I2)
      DO 5 I=1,3
      ITAP(I,2)=ISTORE(I)
5     CONTINUE
      DSHORT=1000.
C
C     THE OCTAL EJCODER IS SIMULATED TO GIVE THE STATE TRANSITION TABLE
C     ILINK AND THE OCTAL OUTPUT IOUT, AS WELL AS THE METRIC OF EACH
C     TRANSITIONS (RMET).
C
```

```
      DO 6 I=1,IL
      DIST(I)=100.
      DO 7 J=1,8
      ILINK(I,J)=(I-1)*8+(J-1)
      CALL OCTAL(ILINK(I,J))
      IOUT(1)=0
      IOUT(2)=0
      DO 8 L1=1,2
      DO 9 L2=1,3
      IOUT(L1)=MOD(IOUT(L1)+ISTORE(L2)*ITAP(L2,L1),8)
 9    CONTINUE
 8    CONTINUE
      RMET(I,J)=EUC(IOUT(1)+1)+EUC(IOUT(2)+1)
 7    CONTINUE
 6    CONTINUE
C
C     THIS IS THE BEGINNING OF THE TRELLIS SEARCH. ICOUNT COUNTS THE
C     THE NUMBER OF STAGES THE VITERBI ALGORITHM HAS PERFORMED.
C
      DIST(1)=0.
      DNEW(1)=1000.
      ICOUNT=0
 17   DLEAST=100.
      ICOUNT=ICOUNT+1
      IF (ICOUNT.EQ.10) GO TO 4
C
C     THE 8 PATHS MERGING INTO A STATE IS COMPARED AND THE SURVIVOR IS
C     PICKED. THE SURVIVOR GOING INTO THE ZERO STATE GIVES ONE OF THE
C     FREE DISTANCES (DRUN), WHICH IS COMPARED WITH THE PREVIOUS MINIMUM
C     FREE DISTANCES (DSHORT) AND UPDATES DSHORT IF DRUN IS SMALLER THAN
C     DSHORT. DLEAST REGISTERS THE SMALLEST ACCUMULATED METRIC OF THE
C     STATES, AND IF DLEAST IS GREATER THAN DSHORT, TRELLIS SEARCH FOR
C     THE MINIMUM DISTANCE PATH OF THE ENCODER HAS BEEN ACCOMPLISHED.
C
      DO 18 I=2,IL
      DNEW(I)=DIST(MOD(ILINK(I,1),IL)+1)+RMET(I,1)
      DO 19 J=2,8
      DNEW(I)=AMIN1(DIST(MOD(ILINK(I,J),IL)+1)+RMET(I,J),DNEW(I))
 19   CONTINUE
      DLEAST9AMIN1(DLEAST,DNEW(I))
 18   CONTINUE
      DO 20 I=1,IL
      DIST(I)=DNEW(I)
 20   CONTINUE
      DRUN=1000.
      DO 21 I=2,8
      DRUN=AMIN1(DIST(MOD(ILINK(1,I),IL)+1)+RMET(1,I),DRUN)
 21   CONTINUE
      IF (DSHORT-DRUN.LT.0.000001) GO TO 24
      DSHORT=DRUN
```

```
C
C       DMIN IS THE FREE DISTANCE OF THE BEST ENCODER FOUND SO FAR. IF
C       DSHORT IS LESS THAN DMIN, THEN THE PRESENT ENCODER CAN BE
C       ABANDONED. AFTER WE FINISH THE TRELLIS SEARCH FOR THE ENCODER
C       HENCE THE MINIMUM FREE DISTANCE OF THE ENCODER IS AT LEAST AS
C       LARGE AS DMIN, THE SUBGENERATORS OF THE ENCODER ARE PRINTED.
C
 24     IF ((DMIN-DSHORT).GT.0.00001) GO TO 4
        IF ((DSHORT-DLEAST).GT.0.00001) GO TO 17
        DMIN=DSHORT
        WRITE(6,138)
 138    FORMAT('     X-----X------X------X-------X------X')
        IK=3-M
        DO 134 L4=1,2
        WRITE(6,135) L4
 135    FORMAT('  THE TAPS TO THE ADDER ',I4,' ARE:')
        WRITE(6,133) (ITAP(L3,L4),L3=IK,3)
 133    FORMAT(3X,5I5)
 134    CONTINUE
        WRITE(6,136) DMIN
 136    FORMAT(' THE MINIMUM DISTANCE OF THE ENCODER IS ',F10.5)
 4      CONTINUE
 2      CONTINUE
        STOP
        END
        SUBROUTINE OCTAL(IDEC)
        COMMON ISTORE(3),K
        IQUOT=IDEC
        DO 1 I=1,3
        ISTORE(I)=IQUOT/8**(3-I)
        IQUOT=IQUOT-ISTORE(I)*8**(3-I)
 1      CONTINUE
        RETURN
        END
```

```
call octal12
TEMPNAME ASSUMED AS MEMBERNAME
 PUT IN THE NUMBER OF OCTAL MEMORIES OF THE ENCODER.
?
1
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 THE TAPS TO THE ADDER    2 ARE:
       0    1
THE MINIMUM DISTANCE OF THE ENCODER IS    0.58600
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 THE TAPS TO THE ADDER    2 ARE:
       0    2
THE MINIMUM DISTANCE OF THE ENCODER IS    1.29300
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 THE TAPS TO THE ADDER    2 ARE:
       0    3
THE MINIMUM DISTANCE OF THE ENCODER IS    2.00000
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 THE TAPS TO THE ADDER    2 ARE:
       0    5
THE MINIMUM DISTANCE OF THE ENCODER IS    2.00000
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 THE TAPS TO THE ADDER    2 ARE:
       1    3
THE MINIMUM DISTANCE OF THE ENCODER IS    2.29300
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 THE TAPS TO THE ADDER    2 ARE:
       1    5
THE MINIMUM DISTANCE OF THE ENCODER IS    2.29300
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 THE TAPS TO THE ADDER    2 ARE:
       2    3
THE MINIMUM DISTANCE OF THE ENCODER IS    3.00000
   X-----X-----X-----X-----X-----X
 THE TAPS TO THE ADDER    1 ARE:
       1    0
 T!
READY
```

```
      GF8CODE.FORT'
C     THIS PROGRAM SEARCHES EXHAUSTIVELY FOR RATE 2/3 GF(8) CONVOLUT
C     -IONAL ENCODERSWITH 1 OCTAL MEMORY.
C
C     DIST STORES THE CUMULATED METRIC OF EACH OF THE EIGHT STATES,
C     AND DNEW IS USED AS A TEMPORARY STORAGE FOR DIST. THERE ARE
C     64 BRANCHES GOING INTO EACH STATE, AND THE ENCODER OUTPUT OF
C     EACH BRANCH IS DENOTED BY RMET. THE ENCODER OUPUT IS A FUNCTION
C     OF THE TAP GAINS WHICH ARE STORED IN ITAP. THE THREE OCTAL OUTPUT
C     OF THE ENCODER IS GIVEN BY IOUT. EUC(I) DENOTES THE EUCLIDEAN
C     DISTANCE BOUND BETWEEN I AND 0.
C
      DIMENSION RMET(8,64),DIST(8),ITAP(3,3),EUCLID(8),ILINK(8,64)
      DIMENSION IREG(3),IOUT(3),DNEW(8),IADD(8,8),IMULT(8,8)
C
C     IADD AND IMULT ARE THE ADDITION AND MULTIPLICATION TABLE FOR
C     GF(8)
C
      DATA IADD /0,1,2,3,4,5,6,7,1,0,3,2,5,4,7,6,
     &           2,3,0,1,6,7,4,5,3,2,1,0,7,6,5,4,
     &           4,5,6,7,0,1,2,3,5,4,7,6,1,0,3,2,
     &           6,7,4,5,2,3,0,1,7,6,5,4,3,2,1,0/
      DATA IMULT/0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,7,
     &           0,2,4,6,3,1,7,5,0,3,6,5,7,4,1,2,
     &           0,4,3,7,6,2,5,1,0,5,1,4,2,7,3,6,
     &           0,6,7,1,5,3,2,4,0,7,5,2,1,6,4,3/
      DATA EUCLID /0.,.293,1.,.293,2.,1.707,1.,.293/
      COMMON ISTORE(3)
      DMIN=0.
C
C     I1,I2,I3 REPRESENTS THE SUBGENERATORS OF THE ENCODER,WHICH ARE
C     CONVERTED TO OCTAL REPRESENTATION.
C
      DO 2 I1=1,255
      CALL OCTAL(I1)
      DO 3 I=1,3
      ITAP(I,1)=ISTORE(I)
 3    CONTINUE
      DO 4 I2=1,255
      CALL OCTAL(I2)
      DO 5 I=1,3
      ITAP(I,2)=ISTORE(I)
 5    CONTINUE
      DO 41 I3=1,255
      CALL OCTAL(I3)
      DO 42 I=1,3
      ITAP(I,3)=ISTORE(I)
 42   CONTINUE
      DSHORT=1000.
```

```
C
C       THE GF(8) ENCODER IS SIMULATED TO GIVE THE STATE TRANSITION TABLE
C       ILINK AND THE OCTAL OUPUT IOUT, AS WELL AS THE METRIC OF EACH
C       TRANSITIONS IMET.
C
        DO 6 I91,8
        DIST(I)=100.
        DO 7 J=1,64
        IOUT(1)=0
        IOUT(2)=0
        IOUT(3)=0
        IREG(1)=(J-1)/8
        IREG(2)=(I-1)
        IREG(3)=J-1-IREG(1)*8
        ILINK(I,J)=J-((J-1)/8)*8
        DO 8 L1=1,3
        DO 9 L2=1,3
        IOUT(L1)=IADD(IOUT(L1)+1,IMULT(IREG(L2)+1,ITAP(L2,L1)+1)+1)
 9      CONTINUE
 8      CONTINUE
        DO 10 L1=1,3
 10     CONTINUE
        RMET(I,J)=EUCLID(IOUT(1)+1)+EUCLID(IOUT(2)+1)+EUCLID(IOUT(3)+1)
 7      CONTINUE
 6      CONTINUE
C
C       THE 64 PATHS MERGING INTO A STATE IS COMPARED AND THE SURVIVOR IS
C       PICKED. THE SURVIVOR GOING INTO THE ZERO STATE GIVES ONE OF THE
C       FREE DISTANCES (DRUN), WHICH IS COMPARED WITH THE PREVIOUS
C       MINIMUM FREE DISTANCES (DSHORT) AND UPDATES DSHORT IF DRUN IS
C       SMALLER THAN DSHORT. DLEAST REGISTERS THE SMALLEST ACCUMULATED
C       METRIC OF THE STATES, AND IF DLEAST IS GREATER THAN DSHORT,
C       TRELLIS SEARCH FOR THE MINIMUM DISTANCE PATH OF THE ENCODER
C       HAS BEEN ACCOMPLISHED
C
        DIST(1)=0.
        DNEW(1)=1000.
        ICOUNT=0
 17     DLEAST=100.
        ICOUNT=ICOUNT+1
        IF (ICOUNT.EQ.10) GO TO 41
        DO 18 I=2,8
        DNEW(I)=DIST(1)+RMET(I,1)
 43     DO 19 J=2,64
        DNEW(I)=AMIN1(DIST(ILINK(I,J))+RMET(I,J),DNEW(I))
 19     CONTINUE
        DLEAST=AMIN1(DLEAST,DNEW(I))
 18     CONTINUE
```

```
      DO 20 I=1,8
      DIST(I)=DNEW(I)
 20   CONTINUE
      DRUN=1000.
      DO 21 I=2,8
      DO 44 J=1,8
      DRUN=AMIN1(DIST(I)+RMET(1,I+(J-1)*8),DRUN)
 44   CONTINUE
 21   CONTINUE
      IF (DSHORT-DRUN.LT.-0.000001) GO TO 24
      DSHORT=DRUN
C
C     DMIN IS THE FREE DISTANCE OF THE BEST ENCODER FOUND SO FAR. IF
C     DSHORT IS LESS THAN DMIN, THEN THE PRESENT ENCODER CAN BE
C     ABANDONED. AFTER WE FINISH THE TRELLIS SEARCH FOR THE ENCODER
C     HENCE THE MINIMUM FREE DISTANCE OF THE ENCODER IS AT LEAST AS
C     FODD AS DMIN, THE SUBGENERATORS OF THE ENCODER ARE PRINTED.
C
 24   IF ((DMIN-DSHORT).GT.0.00001) GO TO 41
      IF ((DSHORT-DLEAST).GT.0.00001) GO TO 17
      IF (DSHORT.LT.3.00001) DMIN=DSHORT
      WRITE(6,141)
 141  FORMAT('        X------X------X------X------X------X')
      DO 142 L4=1,3
      WRITE(6,143) L4
 143  FORMAT('  THE TAPS TO THE ADDER ',I5,'  ARE:')
      WRITE(6,144) (ITAP(L3,L4),L3=1,3)
 144  FORMAT(5X,3I5)
 142  CONTINUE
      WRITE(6,145) DMIN
 145  FORMAT('   THE FREE DISTANCE OF THIS CODE = ',10F5)
 41   CONTINUE
 4    CONTINUE
 2    CONTINUE
      STOP
      END
      SUBROUTINE OCTAL(IDEC)
      COMMON ISTORE(3)
      IQUOT=IDEC
      DO 1 I=1,3
      ISTORE(I)=IQUOT/8**(3-I)
      IQUOT=IQUOT-ISTORE(I)*8**(3-I)
 1    CONTINUE
      RETURN
      END
```

Appendix C. Program for Searching Optimal Rate 1/2 Coded 4ϕ with Nonzero h_1

```
      ISICODE.FORT'
C     THIS PROGRAM SEARCHES FOR OPTIMAL RATE 1/2 BINARY CONVOLUTIONAL
C     ENCODER FOR QPSK WITH NONZERO H1 ONLY.
C
C     IDIST STORES THE COEFFICIENT OF H0 AND H1 OF THE ACCUMULATED
C     METRIC OF THE ENCODER PLUS CHANNEL,WHILE IDNEW IS A TEMPORARY
C     STORAGE FOR IDIST. THERE ARE TWO BRANCHES MERGING INTO STATE I,
C     AND THE ENCODER OUTPUT OF BOTH BRANCHES ARE THE SAME THUS CAN
C     BE STORED IN IMET(I,3). AT STATE I, THE COEF. OF H1 ASSOCIATED
C     WITH EACH BRANCH IS STORED IN IMET(I,1) AND IMET(I,2). THE
C     ENCODER OUTPUTS IOUT AND PREVIOUS ENCODER OUTPUT IPREV ARE
C     FUNCTIONS OF THE TAP GAINS ITAP. THE ACCUMULATED EUCLIDEAN METRIC
C     OF EACH STATE IS RMET WHICH CAN BE CALCULATED FROM IMET. ICH,ITR,
C     IBR ARE ASSOCIATED WITH CHECKING CODE CATASTROPHE.
C
      DIMENSION IMET(256,3),RMET(256,2),IDIST(256,2),DIST(256)
      DIMENSION ITAP(8,2),ICH1(50),ITR(200),IBR(200)
      DIMENSION ILINK(256,2),IDNEW(256,2),IOUT(2),IPREV(2),DNEW(256)
      COMMON ISTORE(8)
      WRITE(6,151)
 151  FORMAT('  ENTER THE NUMBER OF MEMORIES OF THE ENCODER.')
      READ*(K
      WRITE(6,152)
 152  FORMAT('' ENTER THE VALUE OF THE FIRST CORRELATION COEF,H1 ')
      READ*,H1
      WRITE(6,153)
 153  FORMAT(' GUESS A LOWER BOUND FOR DMIN TO START WITH. ')
      READ*,DMIN
      IL=2**K+1
      IU=2**(K+1)-1
C
C     I1 AND I2 REPRESENT THE SUBGENERATORS OF THE ENCODER, WHICH ARE
C     CONVERTED INTO BINARY REPRESENTATION AND STORED AS THE TAP GAINS.
C
      DO 2 I1=IL,IU,2
      CALL CBIN(I1)
      DO 3 I=1,8
      ITAP(I,1)=ISTORE(I)
 3    CONTINUE
      DO 4 I2=IL,IU,2
      CALL CBIN(I2)
      DO 5 I=1,8
      ITAP(I,2)=ISTORE(I)
 5    CONTINUE
      DSHORT=1000.
```

```
C
C      THE EXTENDED STATE OF THE SYSTEM, DENOTED BY ISTATE, DETERMINES
C      THE PRESENT AS WELL AS PREVIOUS OUTPUT OF THE ENCODER. IMET CAN
C      BE CALCULATED FROM THESE OUTPUTS.
C
       ISTATE=2**(K+1)
       DO 6 I=1,ISTATE
       DIST(I)=100.
       IDIST(I,1)=0
       IDIST(I,2)=0
       IOUT(1)=0
       IOUT(2)=0
       CALL CBIN(I-1)
       IA=8-K
       DO 8 L1=1,2
       DO 9 L2=IA,8
       IOUT(L1)=IOUT(L1)+ITAP(L2,L1)*ISTORE(L2)
  9    CONTINUE
       IOUT(L1)=MOD(IOUT(L1),2)
  8    CONTINUE
       IMET(I,3)=IOUT(1)+IOUT(2)
       DO  7 J=1,2
       ILINK(I,J)=MOD(I-1,2**K)*2+J
       IPREV(1)=0
       IPREV(2)=0
       DO 41 IN=IA,7
       ISTORE(IN)=ISTORE(IN+1)
 41    CONTINUE
       ISTORE(8)=J-1
       DO 81 L1=1,2
       DO 91 L2=IA,8
       IPREV(L1)=IPREV(L1)+ITAP(L2,L1)*ISTORE(L2)
 91    CONTINUE
       IPREV(L1)=MOD(IPREV(L1),2)
 81    CONTINUE
       IMET(I,J)=IOUT(1)+IOUT(2)+IPREV(1)+IPREV(2)-
     &           MOD(IOUT(1)+IPREV(1),2)-MOD(IOUT(2)+IPREV(2),2)
       RMET(I,J)=IMET(I,3)-H1*IMET(I,J)
  7    CONTINUE
  6    CONTINUE
C
C      THIS IS THE BEGINNING OF THE TRELLIS SEARCH. ICOUNT COUNTS THE
C      NUMBER OF STAGES THE VITERBI ALGORITHM HAS PERFORMED.
C
       DO 96 I=1,50
 96    ICH1(I)=-1
       DIST(1)=0.
       DNEW(1)=1000.
       ICOUNT=1
```

```
 17     DLEAST=100.
        ILEAST=100
        ICOUNT=ICOUNT+1
        IF (ICOUNT.EQ.50) GO TO 4
C
C       THE 2 PATHS MERGING INTO A STATE IS COMPARED AND THE SURVIVOR IS
C       PICKED. THE SURVIVOR GOING BACK TO THE ZERO STATE GIVES ONE OF
C       THE FREE DISTANCE BOUNDS (DRUN), WHICH IS COMPARED WITH THE
C       PREVIOUS MINIMUM FREE DISTANCES (DSHORT) AND UPDATES DSHORT IF
C       DRUN IS SMALLER THAN DSHORT. AT THE SAME TIME, THE COEFFICIENTS
C       OF H0 AND H1 ASSOCIATED WITH THE MINIMUM FREE DISTANCE IS STORED
C       AS IH0SH AND IS1SH RESPECTIVELY.
C
        DO 18 I=2,ISTATE
        D1=DIST(ILINK(I,1))+RMET(I,1)
        D2=DIST(ILINK(I,2))+RMET(I,2)
        IF (D1.GE.D2) GO TO 82
        DNEW(I)=D1
        IDNEW(I,1)=IDIST(ILINK(I,1),1)+IMET(I,3)
        IDNEW(I,2)=IDIST(ILINK(I,1),2)+IMET(I,1)
        GO TO 18
  82    DNEW(I)=D2
        IDNEW(I,1)=IDIST(ILINK(I,2),1)+IMET(I,3)
        IDNEW(I,2)=IDIST(ILINK(I,2),2)+IMET(I,2)
  18    CONTINUE
        DO 20 I=1,ISTATE
        DIST(I)=DNEW(I)
        IDIST(I,1)=IDNEW(I,1)
        IDIST(I,2)=IDNEW(I,2)
C
C       DLEAST REGISTERS THE SMALLEST ACCUMULATED METRIC OF THE STATES,
C       AND IF DLEAST IS GREATER THAN DSHORT, TRELLIS SEARCH FOR THE
C       MINIMUM DISTANCE PATH OF THE ENCODER HAS BEEN ACCOMPLISHED.
C       DMIN IS THE FREE DISTANCE OF THE BEST ENCODER FOUND SO FAR.
C       OF DSHORT IS LESS THAN DMIN, THEN THE PRESENT ENCODER CAN BE
C       ABANDONED.
C
        DLEAST=AMIN1(DLEAST,DIST(I))
        IF (I.EQ.1) GO TO 20
        ILEAST=MIN0(ILEAST,IDIST(I,1))
 20     CONTINUE
        DRUN=DIST(ILINK(1,2))+RMET(1,2)
        IH0=IDIST(ILINK(1,2),1)+IMET(1,3)
        IH1=IDIST(ILINK(1,2),2)+IMET(1,2)
        IF (DRUN.LT.20) ICH1(IH0)=MAX0(ICH1(IH0),IH1)
        IF (DSHORT-DRUN.LT.0.00001) GO TO 24
        DSHORT=DRUN
        IH0SH=IH0
        IH1SH=IH1
 24     IF ((DMIN-DSHORT).GT.-0.00001) GO TO 4
```

```
C
C       AFTER WE FINISH THE TRELLIS SEARCH AND HENCE DSHORT IS AT LEAST
C       AS LARGE AS DMIN, WE PRINT OUT THE CODE,THE COEFFICIENTS OF HO
C       AND H1 ASSOCIATED WITH THE MINIMUM DISTANCE PATH (IHOMIN AND
C       IH1MIN, AS WELL AS THE VALUE OF DMIN.
C
        IF (DLEAST.LT.DSHORT) GO TO 17
        IHOMIN=IHOSH
        IH1MIN=IH1SH
        WRITE(6,161)
 161    FORMAT('      X------X------X------X------X------X ')
        DO 162 J=1,2
        WRITE(6,163) J
 163    FORMAT('    THE TAP GAINS TO THE ADDER  ',I5,' ARE')
        WRITE(6,164) (ITAP(I,J),I=IA,8)
 164    FORMAT(3X,8I3)
 162    CONTINUE
        WRITE(6,165) DSHORT,IHOMIN,IH1MIN
 165    FORMAT(' MIN. FREE DISTANCE=   ',F10.5,' = ',I2,' HO - ',I3,' H1')
        WRITE(6,166)
 166    FORMAT('  OTHER FREE DISTANCES:')
        WRITE(6,167)
 167    FORMAT(5X,'COEF. OF HO',5X,'-VE OF COEF. OF H1')
        DO 168 I=1,20
        IF (ICH1(I).EQ.-1) GO TO 168
        WRITE(6,169) I,ICH1(I)
 169    FORMAT(7X,I2,15X,I3)
 168    CONTINUE
C
C       CODE CATASTROPHE IS FOUND BY CHECKING IF LOOPS OF ZERO WEIGHT
C       EXISTS
C
        IK1=0
        DO 44 I=2,ISTATE
        DO 45 J=1,2
        IF (RMET(I,J).GT.0.00001) GO TO 45
        IK1=IK1+1
        ITR(IK1)=I
        IBR(IK1)=J
  45    CONTINUE
  44    CONTINUE
        DO 97 IK2=1,IK1
        DO 98 IK3=1,IK1
        IF (ITR(IK2).NE.ILINK(ITR(IK3),IBR(IK3))) GO TO 98
        ITRACK=ITR(IK2)
        ITRBR=IBR(IK2)
```

```
      DO 99 IK4=1,IK1
      IF (ILINK(ITRACK,ITRBR).NE.ITR(IK4)) GO TO 99
      ITRACK=ITR(IK4)
      ITRBR=IBR(IK4)
      IF (ITRACK.NE.ITR(IK2)) GO TO 99
      WRITE(6,101)
101   FORMAT(1X,'HOWEVER, THIS IS A CODE CATASTROPHE.')
      GO TO 4
 99   CONTINUE
 98   CONTINUE
 97   CONTINUE
      DMIN=DSHORT
 4    CONTINUE
 2    CONTINUE
      STOP
      END
      SUBROUTINE CBIN(IDEC)
      COMMON ISTORE(8)
      IQUOT=IDEC
      DO 1 I=1,8
      ISTORE(I)=IQUOT/(2**(8-I))
      IQUOT=IQUOT-ISTORE(I)*2**(8-I)
  1   CONTINUE
      RETURN
      END
```

```
call isicode
TEMPNAME ASSUMED AS MEMBERNAME
 ENTER THE NUMBER OF MEMORIES OF THE ENCODER.
?
4
 ENTER THE VALUE OF THE FIRST CORRELATION COEF,H1
?
.2
GUESS A LOWER BOUND FOR DMIN TO START WITH.
?
5
    X-----X-----X-----X-----X-----X
  THE TAP GAINS TO THE ADDER      1 ARE
    1  0  0  0  1
  THE TAP GAINS TO THE ADDER      2 ARE
    1  0  1  1  1
MIN. FREE DISTANCE=     5.20000 =  6 HO -   4 H1
 OTHER FREE DISTANCES:
    COEF. OF HO     -VE OF COEF. OF H1
       6                 4
       8                 8
HOWEVER, THIS IS A CODE CATASTROPHE.
    X-----X-----X-----X-----X-----X
  THE TAP GAINS TO THE ADDER      1 ARE
    1  0  0  0  1
  THE TAP GAINS TO THE ADDER      2 ARE
    1  1  1  0  1
MIN. FREE DISTANCE=     5.20000 =  6 HO -   4 H1
 OTHER FREE DISTANCES:
    COEF. OF HO     -VE OF COEF. OF H1
       6                 4
       8                 8
      10                10
HOWEVER, THIS IS A CODE CATASTROPHE.
    X-----X-----X-----X-----X-----X
  THE TAP GAINS TO THE ADDER      1 ARE
    1  0  0  0  1
  THE TAP GAINS TO THE ADDER      2 ARE
    1  1  1  1  1
MIN. FREE DISTANCE=     5.20000 =  6 HO -   4 H1
 OTHER FREE DISTANCES:
    COEF. OF HO     -VE OF COEF. OF H1
       6                 4
       7                 8
       8                 4
       9                10
      10                14
    X-----X-----X-----X-----X-----X
  THE TAP GAINS TO THE ADDER      1 ARE
    1  0  0  1  1
  THE TAP GAINS TO THE ADDER      2 ARE
    1  0  1  0  1
MIN. FREE DISTANCE=     5.60000 =  6 HO -   2 H1
 OTHER FREE DISTANCES:
    COEF. OF HO     -VE OF COEF. OF H1
       6                 2
       8                 8
      10                12
```

```
   X-----X-----X-----X-----X-----X
  THE TAP GAINS TO THE ADDER      1 ARE
   1  0  0  1  1
  THE TAP GAINS TO THE ADDER      2 ARE
   1  0  1  1  1
MIN. FREE DISTANCE=      5.80000 =  7 HO -    6 H1
 OTHER FREE DISTANCES:
    COEF. OF HO      -VE OF COEF. OF H1
       7                  6
       8                  8
       9                 10
      11                 12
READY
```

Appendix D. Program for Decoders

```
      R12DECOD.FORT'
C     THIS IS AN OPTIMAL DECODER FOR VITERBI DECODING A RATE 1/2
C     ENCODER WITH UP TO 6 MEMORIES IN THE PRESENCE OF SEVERE ISI.
C
      DIMENSION ITERM2(128,2),ILINK(128,2),IDIST(128),IOUT(2)
      DIMENSION IQRT(128),IPATH(128,100),ITAP(2,6),ITERM1(4),INPUT(128)
      DIMENSION IDNEW(128),RCOS(4),RSIN(4),ISTORE(100),R(2)
      DOUBLE PRECISION DSEED
      COMMON IREG(6),K
      NR=2
      WRITE(6,141)
  141 FORMAT(1X,'INPUT THE NUMBER OF MEMORIES OF THE ENCODER')
      READ*,MEMORY
      K=MEMORY+1
      WRITE(6,142)
  142 FORMAT(1X,'INPUT THE FIRST ISI COEFFICIENT H1')
      READ*,H1
      WRITE(6,143)
  143 FORMAT(1X,'INPUT THE SCALING FACTOR FOR PHE QUANTIZER')
      READ*,FACTOR
      WRITE(6,144)
  144 FORMAT(1X,'INPUT THE LENGTH OF THE SURVIVOR TO BE STORED')
      READ*,IMEM
      WRITE(6,145)
  145 FORMAT(1X,'INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR')
      READ*,DSEED
      WRITE(6,146)
  146 FORMAT(1X,'HOW MANY BITS YOU WANT TO RUN FOR EACH ROUND?')
      READ*,NTOTAL
C
C     ALPHA AND BETA ARE EVALUATED FOR GENERATING RANDOM SEQUENCES
C     WITH CORRELATED CONSECUTIVE ELEMENTS SHOWN IN CHAPTER 7 OF
C     THE THESIS.
C
      ALPHA=0.
      IF (ABS(H1).GT..0001) ALPHA=.5/H1-SQRT((.5/H1)**2-1)
      BETA=1./SQRT(1+ALPHA**2)
C
C     ITAP(I,J) IS THE TAP GAIN FROM THE J-TH REGISTER (INPUT INCLUDED)
C     TO THE I-TH ADDER. RCOS AND RSIN ARE EVALUATED AND STORED SO
C     THEIR VALUE CAN BE RETRIEVED WITHOUT COMPUTATION WHEN NEEDED.
C
      WRITE(6,147)
  147 FORMAT(1X,'INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER A')
      READ *,(ITAP(1,I),I=1,K)
      WRITE(6,148)
  148 FORMAT(1X,'INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER B')
```

```
      READ *,(ITAP(2,I),I=1,K)
      PI=3.1415926
      DO 15 I11=1,4
      RCOS(I11)=COS((I11-1)*PI/2.+PI/4.)
      RSIN(I11)=SIN((I11-1)*PI/2.+PI/4.)
   15 CONTINUE
C
C     THE TABLES FOR DECODING IS GOING TO BE SET UP. EACH STATE IS
C     REPRESENTED BY A NUMBER ISTATE. NSTATE IS THE TOTAL NUMBER OF
C     STATES,INCLUDING THOSE DUE TO ISI.
C
      NSTATE=2**K
      DO 1 ISTATE=1,NSTATE
      CALL CBIN(ISTATE-1)
C
C     THE OUTPUT (IOUT) OF THE ENCODER OF EACH STATE IS CALCULATED.
C     THE CHANNEL SYMBOL IQRT IS THEN OBTAINED BY GRAY MAPPING.
C
      DO 2 I1=1,2
      IOUT(I1)=0
      DO 3 I2=1,K
      IOUT(I1)=IOUT(I1)+ITAP(I1,I2)*IREG(I2)
    3 CONTINUE
      IOUT(I1)=MOD(IOUT(I1),2)
    2 CONTINUE
      IQRT(ISTATE)=IOUT(1)+IOUT(2)+1
      IF ((IOUT(1).EQ.1).AND.(IOUT(2).EQ.0)) IQRT(ISTATE)=4
C
C     THE INPUT INTO THE ENCODER CORRESPONDING TO EACH STATE IS
C     COMPUTED. THEN THE PREVIOUS STATE OF THE ENCODER WHICH IS LINKED
C     TO THE PRESENT STATE BY THE THE BRANCH IB IS FOUND.
C
      INPUT(ISTATE)=IREG(1)
      DO 4 IB=1,2
      DO 61 I=2,K
      ILINK(ISTATE,IB)=ILINK(ISTATE,IB)+IREG(I)*(2**(K-I+1))
   61 CONTINUE
      ILINK(ISTATE,IB)=ILINK(ISTATE,IB)+IB
    4 CONTINUE
    1 CONTINUE
C
C     THE FOLLOWING CALCULATES THE CORRECTION TERM (ITERM2) FOR EACH
C     STATE IN THE PRESENCE OF ISI AS GIVEN BY THE FORMULA FOR THE
C     THE METRIC IN CHAPTER 3 OF THE THESIS. IN THE EXPRESSION,IRD
C     PERFORMS A ROUNDING FUNCTION AFTER SCALING BY FACTOR**2
C
      DO 5 ISTATE=1,NSTATE
      DO 6 I4=1,2
      ITERM2(ISTATE,I4)=IRD(H1*COS((IQRT(ILINK(ISTATE,I4))-IQRT(ISTATE)
     &                             )*PI/2.)*(FACTOR**2))
```

```
      6    CONTINUE
      5    CONTINUE
     71    WRITE(6,150)
    150    FORMAT('INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB')
           READ*,DBNO
           WRITE(6,151)
    151    FORMAT(1X,'IF THE DECODED SHOULD (OR SHOULD NOT) DACODE ISI,
         & ENTER 1 (OR 0)')
           READ*,IMPRVE
           RTEMP=SQRT(.125*10**(-DBNO/10.))
           RNOISE=RTEMP
C
C          STATE METRIC (IDIST), THE PAST NOISE SAMPLES (PNSE) , THE
C          CONTENT OF THE REGISTERS OF THE ENCODER (IREG), THE PRESENT AND
C          PAST CHANNEL SYMBOL OUTPUT (IQ) ARE INITIALIZED TO BE ZERO.
C
           DO 75 I=1,NSTATE
     75    IDIST(I)=0
           IE=0
           INBITS=0
           PNSE1=0.
           PNSE2=0.
           DO 21 I=1,K
           IREG(I)=0
     21    CONTINUE
           IQ1=0
           IQ2=0
           INDEX=1
     18    DO 62 I=2,K
           IREG(K-I+2)=IREG(K-I+1)
     62    CONTINUE
C
C          THE FOLLOWING LINES REPRESENTS AN ENCODER. A RANDOM NUMBER
C          GENERATOR GIVES THE INPUT TO THE ENCODER (IREG(1)). THE PREVIOUS
C          CHANNEL SYMBOLS (IQ1 AND IQ2) ARE ADVANCED AND THE ENCODER PUTS
C          OUT A NEW IQ1.
C
           IREG(1)=GGUBFS(DSEED)+.5
           DO 19 I1=1,2
           IOUT(I1)=0
           DO 20 I2=1,K
           IOUT(I1)=IOUT(I1)+ITAP(I1,I2)*IREG(I2)
     20    CONTINUE
           IOUT(I1)=MOD(IOUT(I1),2)
     19    CONTINUE
           IQ3=IQ2
           IQ2=IQ1
           IQ1=IOUT(1)+IOUT(2)
           IF ((IOUT(1).EQ.1).AND.(IOUT(2).EQ.0)) IQ1=3
```

```
C
C      THE CORRELATED NOISE SEQUENCES ARE GENERATED ACCORDING TO THE
C      TECHNIGUE MENTIONED IN CHAPTER 7 OF THE THESIS. ICR AND ICI
C      FORM THE REAL AND IMAGINARY PART OF THE SUFFICIENT STATISTICS.
C      ITERM1 CORRESPONDS TO THE FIRST TERM OF THE EXPRESSION FOR THE
C      METRIC.
C
       CALL GGNML(DSEED,NR,R)
       CNSE1=(R(1)+ALPHA*PNSE1)*BETA
       CNSE2=(R(2)+ALPHA*PNSE2)*BETA
       PNSE1=R(1)
       PNSE2=R(2)
       ICR= IRD(((COS(IQ2*PI/2+PI/4)+H1*(COS(IQ1*PI/2+PI/4)+
     &        COS(IQ3*PI/2+PI/4)))/2+CNSE1*RNOISE)*FACTOR**2)
       ICI=-IRD(((SIN(IQ2*PI/2+PI/4)+H1*(SIN(IQ1*PI/2+PI/4)+
     &        SIN(IQ3*PI/2+PI/4)))/2+CNSE2*RNOISE)*FACTOR**2)
       DO 7 I5=1,4
       ITERM1(I5)=IRD((RCOS(I5)*ICR-RSIN(I5)*ICI)*2)
   7   CONTINUE
C
C      THE FOLLOWINGS SIMULATE A DECODER WHICH INPUTS THE SUFFICIENT
C      STATISTICS ICR AND ICI (OR ANY INPHASE AND QUADRATURE SAMPLED
C      VOLTAGES OF THE DEMODULATOR) AND TRELLIS SEARCH FOR THE
C      MAXIMUM LIKELIHOOD SEQUENCE. IDLARG IS THE LARGEST METRIC FOR
C      THE STATES AT A STAGE OF DECODING.
C
       IDLARG=-10000000
       DO 8 ISTATE=1,NSTATE
C
C      FOR EACH STATE,THERE ARE TWO BRANCHES (IBRCH) MERGING INTO IT.
C      THE SURVIVOR IS CHOSEN AND THE STATE METRIC IS UPDATED AND STORED
C      TEMPORARILY IN IDNEW. THE PREVIOUS STATE IN THE PATH OF THE
C      SURVIVOR IS STORED IN IPATH.
C
       IDMRGE=IDIST(ILINK(ISTATE,1))-ITERM2(ISTATE,1)*IMPRVE
       IBRCH=1
       ITEMP=IDIST(ILINK(ISTATE,2))-ITERM2(ISTATE,2)*IMPRVE
       IF (IDMRGE.GE.ITEMP) GO TO 9
       IBRCH=2
       IDMRGE=ITEMP
   9   IDNEW(ISTATE)=IDMRGE+ITERM1(IQRT(ISTATE))
       IPATH(ISTATE,INDEX)=ILINK(ISTATE,IBRCH)
C
C      THE STATE WITH THE LARGEST METRIC (ILARGE) IS FOUND AND STORED.
C
```

```
      IF (IDNEW(ISTATE).LE.IDLARG) GO TO 8
      IDLARG=IDNEW(ISTATE)
      ILARGE=ISTATE
   8  CONTINUE
C
C     THE SURVIVOR WITH THE LARGEST METRIC IS TRACED BACK A NUMBER OF
C     STATES TO FIND THE DECODED INFORMATION SEQUENCE.IPOINT SERVES AS
C     A POINTER TRACING FROM ONE STATE TO ANOTHER. THE LOCATION OF
C     STORAGE FOR ILARGE AT THE PRESENT DECODING STAGE IS POINTED TO
C     BY THE POINTER CALLED INDEX,WHICH IS INCREMENTED BY MODULO
C     ARITHMETICS. THE INPUT TO THE ENCODER IS STORED BY A CIRCULAR
C     STRUCTURE CALLED ISTORE, SO THAT IT MAY BE RETRIEVED LATER FOR
C     COMPARISON WITH THE DECODED SEQUENCE.
C
      IPOINT=ILARGE
      ITRACE=ISTORE(INDEX)
      ISTORE(INDEX)=IDELAY
      IDELAY=IREG(1)
      DO 10 I7=1,INDEX
      IPOINT=IPATH(IPOINT,INDEX+1-I7)
  10  CONTINUE
      IF (INDEX.EQ.IMEM) GO TO 17
      ITIMES=IMEM-INDEX
      DO 11 I8=1,ITIMES
      IPOINT=IPATH(IPOINT,IMEM-I8+1)
  11  CONTINUE
  17  IF ((INBITS.GT.IMEM).AND.(ITRACE.NE.INPUT(IPOINT))) IE=IE+1
C     UPDATE DISTANCE TABLE
      DO 12 ISTATE=1,NSTATE
      IDIST(ISTATE)=IDNEW(ISTATE)
  12  CONTINUE
      INDEX=INDEX+1
      IF (INDEX.GT.IMEM) INDEX=1
C
C     IE IS THE NUMBER OF BIT ERRORS MADE. THE BIT ERROR PROBABILITY
C     IS COMPUTED. FOR EVERY 10000 BITS,THE BER WOULD BE PRINTED UNTIL
C     THE DECODER HAS DECODED THE REQUIRED NUMBER OF BIT (NTOTAL).
C
      INBITS=INBITS+1
      IF (MOD(INBITS,10000).NE.0) GO TO 68
      BER=IE*1./INBITS
      WRITE(6,180) INBITS,IE,BER
 180  FORMAT(1X,I6,' BITS ARE DECODED,ERRORS=',I5,'BER=',F7.6)
  68  IF (INBITS.LT.NTOTAL) GO TO 18
      WRITE(6,270)
 270  FORMAT('      X------X------X------X------X------X')
      IF (1.EQ.1) GO TO 71
      STOP
      END
```

```
C
C       CBIN CONVERTS A DECIMAL NUMBER INTO A BINARY NUMBER
C
        SUBROUTINE CBIN(IDEC)
        COMMON IREG(6),K
        IQUOT=IDEC
        DO 16 I=1,K
        IREG(I)=IQUOT/2**(K-I)
        IQUOT=IQUOT-IREG(I)*2**(K-I)
   16   CONTINUE
        RETURN
        END
C
C       IRD PERFORMS A ROUNDING FUNCTION.
C
        FUNCTION IRD(RE)
        IF (RE.GE.0.) IRD=RE+.5
        IF (RE.LT.0.) IRD=RE-.5
        RETURN
        END
```

```
call r12decod
TEMPNAME ASSUMED AS MEMBERNAME
INPUT THE NUMBER OF MEMORIES OF THE ENCODER
?
2
INPUT THE FIRST ISI COEFFICIENT H1
?
.2
INPUT THE SCALING FACTOR FOR THE QUANTIZER
?
10
INPUT THE LENGTH OF THE SURVIVOR TO BE STORED
?
100
INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR
?
2345
HOW MANY BITS YOU WANT TO RUN FOR EACH ROUND?
?
50000
INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER A
?
1 0 1
INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER B
?
1 1 1
INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB
?
3
IF THE DECODER SHOULD (OR SHOULD NOT) DECODE ISI,      ENTER 1 (OR 0)
?
1
 10000 BITS ARE DECODED,ERRORS=   56BER=.005600
 20000 BITS ARE DECODED,ERRORS=   95BER=.004750
 30000 BITS ARE DECODED,ERRORS=  123BER=.004100
 40000 BITS ARE DECODED,ERRORS=  172BER=.004300
 50000 BITS ARE DECODED,ERRORS=  214BER=.004280
     X-----X-----X-----X-----X-----X
INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB
?
4
IF THE DECODED SHOULD (OR SHOULD NOT) DECODE ISI,      ENTER 1 (OR 0)
?
1
 10000 BITS ARE DECODED,ERRORS=   10BER=.001000
 20000 BITS ARE DECODED,ERRORS=   26BER=.001300
 30000 BITS ARE DECODED,ERRORS=   38BER=.001267
 40000 BITS ARE DECODED,ERRORS=   45BER=.001125
 50000 BITS ARE DECODED,ERRORS=   54BER=.001080
     X-----X-----X-----X-----X-----X
NPUT THE ENERGY PER BIT TO NOISE RATIO IN DB
?
@!
READY
```

```
      .R23DECOD.FORT'
C     THIS IS AN OPTIMAL DECODER FOR VITERBI DECODING A RATE 2/3
C     CODE WITH UP TO 6 MEMORIES IN THE PRESENCE OF ISI.
C
      DIMENSION ITERM2(256,4),ILINK(256,4),IDIST(256),IOUT(3)
      DIMENSION IMET(256),IPATH(256,150),ITAP(3,6),ITERM1(8),R(2)
      DIMENSION IDNEW(256),RCOS(8),RSIN(8),ISTORE(2,150),INPUT(256,2)
      DOUBLE PRECISION DSEED
      COMMON IREG(8),K
      NR=2
      WRITE(6,140)
 140  FORMAT(1X,'INPUT N AND U, THE NUMBER OF MEMORIES IN EACH QUEUE')
      READ*,KN,KU
      K=KN+KU+2
      WRITE(6,141)
 141  FORMAT(1X,'INPUT THE FIRST ISI COEFFICIENT')
      READ*,H1
      WRITE(6,142)
 142  FORMAT(1X,'INPUT THE SCALING FACTOR FOR THE QUANTIZER')
      READ*, FACTOR
      WRITA(6,143)
 143  FORMAT(1X,'INPUT THE LENGTH OF THE SURVIVOR TO BE STORED')
      READ*,IMEM
      WRITE(6,144)
 144  FORMAT(1X,'INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR')
      READ*,DSEED
      WRITE(6,145)
 145  FORMAT(1X,'HOW MANY BITS YOU WANT TO RUN FOR EACH ROUND')
      READ*,NTOTAL
C
C     ALPHA AND BETA ARE EVALUATED FOR GENERATING RANDOM SEQUENCES
C     WITH CORRELATED CONSECUTIVE ELEMENTS SHOWN IN CHAPTER 7 OF
C     THE THESIS.
C
      ALPHA=0.
      IF (ABS(H1).GT.0.00001) ALPHA=.5/H1-SQRT((.5/H1)**2-1)
      BETA=1./SQRT(1+ALPHA**2)
      PI=3.1415926
C
C     ITAP(I,J) IS THE TAP GAIN FROM THE J-TH REGISTER (INPUT INCLUDED)
C     TO THE I-TH ADDER. RCOS AND RSIN ARE EVALUATED AND STORED SO
C     PHAT THEIR VALUE MAY BE RETRIEVED LATER WITHOUT COMPUTATION
C     WHEN NEEDED.
C
```

```
      WRITE(6,147)
 147  FORMAT(1X,'INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER A')
      READ*,(ITAP(1,I),I=1,K)
      WRITE(6,148)
 148  FORMAT(1X,'INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER B')
      READ*,(ITAP(2,I),I=1,K)
      WRITE(6,149)
 149  FORMAT(1X,'INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER C')
      READ*,(ITAP(3,I),I=1,K)
      DO 15 I11=1,8
      RCOS(I11)=COS((I11-1)*PI/4.+PI/8.)
      RSIN(I11)=SIN((I11-1)*PI/4.+PI/8.)
  15  CONTINUE
C
C     THE TABLES FOR DECODING IS GOING TO BE SET UP. EACH STATE
C     IS REPRESENTED BY A NUMBER ISTATE.NSTATE IS THE TOTAL NUMBER OF
C     STATES, INCLUDING THOSE DUE TO ISI.
C
      NSTATE=2**K
      DO 1 ISTATE=1,NSTATE
      CALL CBIN(ISTATE-1)
      DO 2 I1=1,3
      IOUT(I1)=0
      DO 3 I2=1,8
      IOUT(I1)=IOUT(I1)+ITAP(I1,I2)*IREG(I2)
   3  CONTINUE
      IOUT(I1)=MOD(IOUT(I1),2)
   2  CONTINUE
      IMET(ISTATE)=IOUT(1)*4+IOUT(2)*2+IOUT(3)+1
C
C     THE INPUT TO THE ENCODER CORRESPONDING TO EACH STATE IS
C     COMPUTED. THEN THE PREVIOUS STATE OF THE ENCODER WHICH IS LINKED
C     TO THE PRESENT STATE BY THE BRANCH (I3) IS FOUND.
C
      INPUT(ISTATE,1)=IREG(1)
      INPUT(ISTATE,2)=IREG(2+KN)
      DO 4 I3=1,4
      ILINK(ISTATE,I3)=2-MOD(I3,2)
      DO 160 I6=1,KU
 160  ILINK(ISTATE,I3)=IREG(K-I6+1)*2**I6+ILINK(ISTATE,I3)
      ILINK(ISTATE,I3)=((I3-1)/2)*2**(KU+1)+ILINK(ISTATE,I3)
      DO 161 I6=1,KN
 161  ILINK(ISTATE,I3)=IREG(K-KU-I6)*2**(KU+I6+1)+ILINK(ISTATE,I3)
   4  CONTINUE
   1  CONTINUE
C
C     THE FOLLOWING CALCULATES THE CORRECTION TERM (ITERM2) FOR EACH
C     STATE IN THE PRESENCE OF ISI AS GIVEN BY THE SECOND TERM OF THE
C     FORMULA FOR THE METRIC DERIVED IN CHAPTER 3 OF THE THESIS. IRD
C     PERFORMS A ROUNDING FUNCTION AFTER SCALING BY FACTOR**2
C
```

```
      DO 5 ISTATE=1,NSTATE
      DO 6 I4=1,4
      ITERM2(ISTATE,I4)=IRD(H1*COS((IMET(ILINK(ISTATE,I4))-IMET(ISTATE)
     &                          )*PI/4.)*FACTOR**2)
    6 CONTINUE
    5 CONTINUE
   71 WRITE(6,150)
  150 FORMAT(1X,'INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB')
      READ*,DBNO
      RTEMP=SQRT(.0625*10**(-DBNO/10.))
      RNOISE=RTEMP
C
C     STATE METRIC(IDIST),THE PAST NOISE SAMPLES (PNSE), THE CONTENT
C     OF THE REISTERS OF THE ENCODER(IREG), THE PRESENT AND PAST
C     CHANNEL SYMBOL OUTPUT (IOCT) ARE INITIALIZED TO BE ZERO.
C
      DO 79 I=1,NSTATE
   79 IDIST(I)=0.
      PNSE1=0.
      PNSE2=0.
      IE=0
      INBITS=0
      DO 21 I=1,K
      IREG(I)=0
   21 CONTINUE
      IOCT1=0
      IOCT2=0
      INDEX=1
C
C     THE FOLLOWING REPRESENTS AN ENCODER. A RANDOM NUMBER GENERATES
C     THE INPUTS TO THE ENCODER (IREG(1) AND IREG(KN+2)). THE PREVIOUS
C     CHANNEL SYMBOLS (IOCT1 AND IOCT2) ARE ADVANCED AND THE ENCODER
C     PUTS OUT A NEW IOCT1
C
   18 IF (KN*EQ.0) GO TO 165
      DO 162 I=1,KN
  162 IREG(KN+2-I)=IREG(KN+1-I)
  165 DO 163 I=1,KU
  163 IREG(K+1-I)=IREG(K-I)
      IREG(1)=GGUBFS(DSEED)+.4999
      IREG(KN+2)=GGUBFS(DSEED)+.4999
      DO 19 I1=1,3
      IOUT(I1)=0
      DO 20 I2=1,8
      IOUT(I1)=IOUT(I1)+ITAP(I1,I2)*IREG(I2)
   20 CONTINUE
      IOUT(I1)=MOD(IOUT(I1),2)
   19 CONTINUE
      IOCT3=IOCT2
      IOCT2=IOCT1
      IOCT1=IOUT(1)*4+IOUT(2)*2+IOUT(3)
```

```
C
C       THE CORRELATED NOISE SEQUENCES (CNSE1 AND CNSE2) ARE GENERATED
C       ACCORDING TO THE TECHNIQUE MENTIONED IN CHAPTER 7 OF THE THESIS.
C       ICR AND ICI FORM THE REAL AND IMAGINARY PART OF THE SUFFICIENT
C       STATISTICS. ITERM1 CORRESPONDS TO THE FIRST TERM OF THE
C       EXPRESSIKN FOR THE METRIC.
C
        CALL GGNML (DSEED,NR,R)
        CNSE1=(PNSE1*ALPHA+R(1))*BETA
        CNSE2=(PNSE2*ALPHA+R(2))*BETA
        PNSE1=R(1)
        PNSE2=R(2)
        ICR= IRD(((COS(IOCT2*PI/4+PI/8)+H1*(COS(IOCT1*PI/4+PI/8)+
     &       COS(IOCT3*PI/4+PI/8)))/2+RNOISE*CNSE1)*FACTOR**2)
        ICI=-IRD(((SIN(IOCT2*PI/4+PI/8)+H1*(SIN(IOCT1*PI/4+PI/8)+
     &       SIN(IOCT3*PI/4+PI/8)))/2+RNOISE*CNSE2)*FACTOR**2)
        DO 7 I5=1,8
        ITERM1(I5)=IRD((RCOS(I5)*ICR-RSIN(I5)*ICI)*2)
    7   CONTINUE
C
C       THE FOLLOWING SIMULATE A DECODER WHICH INPUTS THE SUFFICIENT
C       STATISTICS ICR AND ICI (OR ANY INPHASE AND QUADRATURE SAMPLED
C       VOLTAGES OF THE DEMODULATOR) AND TRELLIS SEARCH FOR THE MAXIMUM
C       LIKELIHOOD SEQUENCE. IDLARG IS THE LARGEST METRIC FOR THE STATES
C       AT A STAGE OF DECODING.
C
        IDLARG=-10000000
        DO 8 ISTATE=1,NSTATE
C
C       FOR EACH STATE, THERE ARE FOUR BRANCHES (I6) MERGING INTO IT.
C       IDMERGE IS THE METRIC OF THE SURVIVOR,WHICH LAST BRANCH IS IBRCH.
C       THE SURVIVOR IS STORED IN THE TABLE IPATH. THE STATE METRIC IS
C       THEN UPDATED.
C
        IDMRGE=IDIST(ILINK(ISTATE,1))-ITERM2(ISTATE,1)
        IBRCH=1
        DO 9 I6=2,4
        ITEMP=IDIST(ILINK(ISTATE,I6))-ITERM2(ISTATE,I6)
        IF (IDMRGE.GE.ITEMP) GO TO 9
        IBRCH=I6
        IDMRGE=ITEMP
    9   CONTINUE
        IDNEW(ISTATE)=IDMRGE+ITERM1(IMET(ISTATE))
        IPATH(ISTATE,INDEX)=ILINK(ISTATE,IBRCH)
C
C       THE STATE WITH THE LARGEST METRIC(ILARGE) IS FOUND AJD STORED.
C
        IF (IDNEW(ISTATE).LE.IDLARG) GO TO 8
        IDLARG=IDNEW(ISTATE)
        ILARGE=ISTATE
    8   CONTINUE
```

```
C
C       THE SURVIVOR WITH THE LARGEST METRIC IS TRACED BACK A NUMBER OF
C       STATES TO FIND THE DECODED INFORMATION SEQUENCE. IPOINT SERVES
C       AS A POINTER TRACING FROM ONE STATE TO ANOTHER. THE LOCATION OF
C       STORAGE FOR ILARGE AT THE PRESENT DECODING STAGE IS POINTED TO
C       BY THE POINTER CALLED INDEX, WHICH IS INCREMENTED BY MODULO
C       ARITHMETICS. THE INPUT TO THE ENCODER IS STORED BY A CIRCULAR
C       STRUCTURE CALLED ISTORE, SO THAT IT MAY BE RETRIEVED LATER FOR
C       COMPARISON WITH THE DECODED SEQUENCE.
C
        IPOINT=ILARGE
        ITR1=ISTORE(1,INDEX)
        ITR2=ISTORE(2,INDEX)
        ISTORE(1,INDEX)=IDLAY1
        ISTORE(2,INDEX)=IDLAY2
        IDLAY1=IREG(1)
        IDLAY2=IREG(4)
        DO 10 I7=1,INDEX
        IPOINT=IPATH(IPOINT,INDEX+1-I7)
  10    CONTINUE
        IF (INDEX.EQ.IMEM) GO TO 17
        ITIMES=IMEM-INDEX
        DO 11 I8=1,ITIMES
        IPOINT=IPATH(IPOINT,IMEM-I8+1)
  11    CONTINUE
  17    IF ((ITR1.NE.INPUT(IPOINT,1)).AND.(INBITS.GT.IMEM*2)) IE=IE+1
        IF ((ITR2.NE.INPUT(IPOINT,2)).AND.(INBITS.GT.IMEM*2)) IE=IE+1
C
C       THE DISTANCE TABLE IS UPDATED
C
        DO 12 ISTATE=1,NSTATE
        IDIST(ISTATE)=IDNEW(ISTATE)
  12    CONTINUE
C
C       THE INDEX AND THE COUNT FOR NUMBER OF DECODED BITS ARE
C       INCREMENTED. IE IS THE NUMBER OF BIT ERRORS MADE. THE BIT ERROR
C       PROBABILITY IS COMPUTED* FOR EVERY 10000 BITS, THE BER WOULD BE
C       PRINTED UNTIL THE DECODER HAS DECODED THE REQUIRED NUMBER OF
C       BITS (NTOTAL).
C
        INDEX=INDEX+1
        IF (INDEX.GT.IMEM) INDEX=1
        INBITS=INBITS+2
        IF (MOD(INBITS,10000).NE.0) GO TO 75
        BER=IE*1./INBITS
        WRITE(6,180) INBITS,IE,BER
 180    FORMAT(1X,I6,' BITS ARE DECODED,ERROR=',I5,' BER=',F7.6)
  75    IF (INBITS.LT.NTOTAL) GO TO 18
```

```
      WRITE(6,270)
 270  FORMAT('      X-----X------X------X-------X-------X')
      IF (1.EQ.1) GO TO 71
      STOP
      END
C
C     THE SUBROUTINE CBIN CONVERTS A DECIMAL NUMBER INTO A BINARY
C     NUMBER
C
      SUBROUTINE CBIN(IDEC)
      COMMON IREG(8),K
      IQUOT=IDEC
      DO 16 I=1,K
      IREG(I)=IQUOT/2**(K-I)
      IQUOT=IQUOT-IREG(I)*2**(K-I)
  16  CONTINUE
      RETURN
      END
C
C     IRD PERFORMS A ROUNDING FUNCTION.
C
      FUNCTION IRD(RE)
      IF (RE.GE.0.) IRD=RE+.5
      IF (RE.LT.0.) IRD=RE-.5
      RETURN
      END
```

```
call r23decod
TEMPNAME ASSUMED AS MEMBERNAME
INPUT N AND U, THE NUMBER OF MEMORIES IN EACH QUEUE
?
2 2
INPUT THE FIRST ISI COEFFICIENT
?
.166667
INPUT THE SCALING FACTOR FOR THE QUANTIZER
?
10
INPUT THE LENGTH OF THE SURVIVOR TO BE STORED
?
100
INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR
?
456
HOW MANY BITS YOU WANT TO RUN FOR EACH ROUND
?
50000
INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER A
?
0 1 0 1 0 1
INPUT THE TAP GAINS IN A BIN. SEQUENCE FOR ADDER B
?
1 1 1 0 0 1
INPUT THE TAP GAINS IN A BIN. SEQUENCE FKR ADDER C
?
0 0 0 0 1 0
INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB
?
5
 10000 BITS ARE DECODED,ERROR=    0 BER=.0
 20000 BITS ARE DECODED,ERROR=   12 BER=.000600
 30000 BITS ARE DECODED,ERROR=   12 BER=.000400
 40000 BITS ARE DECODED,ERROR=   32 BER=.000800
 50000 BITS ARE DECODED,ERROR=   42 BER=.000840
    X-----X-----X-----X-----X-----X
INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB
?
6
 10000 BITS ARE DECODED,ERROR=   13 BER=.001300
 20000 BITS ARE DECODED,ERROR=   13 BER=.000650
 30000 BITS ARE DECODED,ERROR=   13 BER=.000433
 40000 BITS ARE DECODED,ERROR=   13 BER=.000325
 50000 BITS ARE DECODED,ERROR=   13 BER=.000260
    X-----X-----X-----X-----X-----X
INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB
?
6
 10000 BITS ARE DECODED,ERROR=    0 BER=.0
 20000 BITS ARE DECODED,ERROR=    0 BER=.0
 30000 BITS ARE DECODED,ERROR=    0 BER=.0
 40000 BITS ARE DECODED,ERROR=    0 BER=.0
 50000 BITS ARE DECODED,ERROR=    0 BER=.0
    X-----X-----X-----X-----X-----X
```

Appendix E. Miscellaneous Programs

```
      PSK.FORT'
C     THIS PROGRAM EVALUATES THE BIT ERROR RATE OF 4-PSK BY SIMULATION.
C
      DIMENSION R(2)
      NR=2
      WRITE(6,21)
 21   FORMAT(1X,'INPUT THE ENERGY PER BIT TO NOISE LEVEL (IN DB).')
      READ*,EBNO
      WRITE(6,22)
 22   FORMAT(1X,'HOW MANY BITS YOU WANT TO SIMULATE?')
      READ*,NTOTAL
      WRITE(6,23)
 23   FORMAT(1X,'INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR?')
      READ*,DSEED
      AVAR=SQRT(.0625*10**(-EBNO/10.))
      WRITE(6,27) AVAR
 27   FORMAT(' INPUT THE VALUE',F8.7,' BACK INTO THE PROGRAM.')
      READ*,FACTOR
      NERROR=0
      NBITS=0
 3    CALL GGNML (DSEED,NR,R)
      A1=R(1)
      A2=R(2)
      RCR= .5+A1*FACTOR
      RCI=A2*FACTOR
      IF (RCR.LT.0) GO TO 1
      IF (ABS(RCI).GT.ABS(RCR)) NERROR=NERROR+1
      GO TO 2
  1   IF (ABS(RCI).GT.ABS(RCR)) NERROR=NERROR+1
      IF (ABS(RCI).LT.ABS(RCR)) NERROR=NERROR+2
  2   NBITS=NBITS+2
      IF (MOD(NBITS,10000).NE.0) GO TO 3
      PERROR=NERROR*1./NBITS
      WRITE(6,26) NBITS,PERROR
      IF (NBITS.LT.NTOTAL) GO TO 3
 26   FORMAT(' BER FOR ',I7,' BITS IS ',F8.7)
 25   STOP
      END
```

```
call psk
TEMPNAME ASSUMED AS MEMBERNAME
INPUT THE ENERGY PER BIT TO NOISE LEVEL (IN DB).
?
2.5
HOW MANY BITS YOU WANT TO SIMULATE?
?
30000
INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR?
?
3345
INPUT THE VALUE.1874736 BACK INTO THE PROGRAM.
?
.1874
BER FOR   10000 BITS IS .0294000
BER FOR   20000 BITS IS .0285000
BER FOR   30000 BITS IS .0285000
READY
```

```
      'PSK8.FORT'
C     THIS PROGRAM EVALUATES THE BIT ERROR RATE OF 8-PSK.
C
      DIMENSION R(2)
      NR=2
      WRITE(6,21)
 21   FORMAT('  INPUT THE ENERGY PER BIT TO NOISE RATIO (IN DB)')
      READ*,EBNO
      WRITE(6,22)
 22   FORMAT(' WHAT IS THE TOTAL NUMBER OF BITS YOU WANT TO SIMULATE?')
      READ*,NTOTAL
      WRITE(6,23)
 23   FORMAT('  INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR.')
      READ*,DSEED
      AVAR=SQRT(.0625*10**(-EBNO/10.)*.666667)
      WRITE(6,24) AVAR
 24   FORMAT(' INPUT THE VALUE ',F8.7,' INTO THE PROGRAM.')
      READ*,FACTOR
      NE=0
      NBITS=0
      S1=COS(3.1415926/8)/2
      S2=SIN(3.1415926/8)/2
 3    CALL GGNML (DSEED,NR,R)
      A1=R(1)
      A2=R(2)
      RCR= S1+A1*FACTOR
      RCI= S2+A2*FACTOR
      ANGLE=ATAN2(RCI,RCR)/(3.1415926/4.)
      IF (ANGLE.LT.0) ANGLE=ANGLE+8
      IANGLE=ANGLE
      IF ((IANGLE.EQ.1).OR.(IANGLE.EQ.3).OR.(IANGLE.EQ.7)) NE=NE+1
      IF ((IANGLE.EQ.2).OR.(IANGLE.EQ.4).OR.(IANGLE.EQ.6)) NE=NE+2
      IF (IANGLE.EQ.5) NE=NE+3
      NBITS=NBITS+3
      IF (MOD(NBITS,30000).NE.0) GO TO 3
      PERROR=NE*1./NBITS
      WRITE(6,26) NBITS,PERROR
 26   FORMAT(' FOR ',I9,' THE BER IS ',F8.7)
      IF (NBITS.LT.NTOTAL) GO TO 3
      STOP
      END
```

```
call psk8
TEMPNAME ASSUMED AS MEMBERNAME
 INPUT THE ENERGY PER BIT TO NOISE RATIO (IN DB)
?
6
WHAT IS THE TOTAL NUMBER OF BITS YOU WANT TO SIMULATE?
?
100000
 INPUT THE SEED FOR THE RANDOM NUMBER GENERATOR.
?
4540
INPUT THE VALUE .1023045  INTO THE PROGRAM.
?
.1023          bits
FOR      30000 THE BER IS .0200000
FOR      60000 THE BER IS .0194833
FOR      90000 THE BER IS .0195889
FOR     120000 THE BER IS .0200000
READY
```

```
'FLTRLOSS.FORT'
C       THIS PROGRAM EVALUATES THE FILTER LOSS OF PQLSES WHICH ARE
C       SYMMETRICAL ABOUT THE CENTER. THE MAXIMUM AMPLITUDE IS FOUND
C       CONSTRUCTIVELY SUPERIMPOSING SHIFTED VERSIONS OF THE PULSE.
C       THEN THE FILTER LOSS IS CALCULATED ACCORDING TO THE FORMULA
C       GIVEN IN THE THESIS.
C
        DIMENSION PULSE(300,8),AMP(300)
  8     READ *,(PULSE(I,1),I=1,16)
        DO 11 I=17,300
 11     PULSE(I,1)=0.
        DO 1 I=1,16
        PULSE(I+16,1)=PULSE(16-I+1,1)
  1     CONTINUE
        DO 2 J1=4,16
        DO 9 K1=2,8
        DO 10 K2=1,300
        PULSE(K2,K1)=0
 10     CONTINUE
  9     CONTINUE
        DO 3 J2=2,8
        DO 4 J3=1,32
        PULSE(J3+2*J1*(J2-1),J2)=PULSE(J3,1)
  4     CONTINUE
  3     CONTINUE
        DO 5 J2=1,300
        SUM=0.
        DO 6 J3=1,8
        SUM=SUM+PULSE(J2,J3)
  6     CONTINUE
        AMP(J2)=SUM
  5     CONTINUE
        AMAXU=0.
        AMAXS=0.
        DO 7 J2=1,250
        R1=2*(AMP(J2)**2)
        R2=AMP(J2)**2+AMP(J2+J1)**2
        IF (R1.GT.AMAXU) AMAXU=R1
        IF (R2.GT.AMAXS) AMAXS=R2
  7     CONTINUE
        FL1= 10*ALOG10(AMAXU*J1/31.)
        FL2= 10*ALOG10(AMAXS*J1/31.)
        WRITE(6,*) J1,FL1,FL2
  2     CONTINUE
        READ *,IC
        IF (IC.EQ.1) GO TO 8
        STOP
        END
```

REFERENCES

[1] F. Amoroso, "The Bandwidth of Digital Data Signals," IEEE Communication Magazine, Vol. 18, Nov. 1980.

[2] H. J. Landau, H. O. Pollak, "Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty - II," Bell System Tech. J., 40, pp. 65-84, 1961.

[3] S. A. Gronemeyer, A. L. McBride, "MSK and Offset QPSK Modulation," IEEE Trans. on Comm., Vol. COM-24, Aug. 1976.

[4] S. Rhodes, "FSOQ, A Bandwidth Efficient Modulation Technique," NTC, 1980, 51.5.

[5] F. de Jager, C. B. Dekker, "Tamed Frequency Modulation, A Novel Method to Achieve Spectrum Economy in Digital Transmission," IEEE Trans. on Comm., Vol. COM-26, May 1978.

[6] M. C. Austin, M. U. Chang, "Quadrature Overlapped Raised-Cosine Modulation," ICC '80 Conference Record, pp. 2.7.1-5.

[7] F. Amoroso, "The Use of Quasi-Bandlimited Pulses in MSK Transmission," IEEE Trans. on Comm., Vol. COM-27, Oct. 1979.

[8] D. Muilwijk, "Correlative Phase Shift-Keying, A Bandwidth and Power Efficient Constant Envelope Modulation," ICC '80 Conference Record, pp. 26.6.1-8.

[9] J. B. Anderson, D. P. Taylor, "A Bandwidth-Efficient Class of Signal-Space Codes," IEEE Trans. Inform. Theory, Vol. IT-24, pp. 703-712, Nov. 1978.

[10] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," unpublished, May 10, 1977.

[11] G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," IEEE Trans. Inform. Theory, Vol. IT-18, May 1972.

[12] G. Ungerboeck, "Adaptive Maximum-Likelihood Receiver for Carrier Modulated Data-Transmission System," IEEE Trans. on Comm., Vol. COM-22, May 1974.

[13] A. J. Viterbi, J. K. Omura, Principles of Digital Communication and Coding, New York: McGraw Hill, 1979.

[14] M. F. Mesiya, P. J. McLane, L. L. Campbell, "Maximum Likelihood Sequence Estimation of Binary Sequences Transmitted Over Bandlimited Channels," IEEE Trans. on Comm., Vol. COM-25, July 1977.

[15] G. F. Herrmann, "Performance of Maximum-Likelihood Receiver in the Nonlinear Satellite Channel," IEEE Trans. on Comm., Vol. COM-26, March 1978.

[16] G. D. Forney, "Convolutional Codes I: Algebraic Structure," IEEE Trans. Inform. Theory, Vol. IT-16, Nov. 1970.

[17] J. M. Wozencraft, I. M. Jacobs, Principle of Communication Engineering, New York, Wiley, 1965.

[18] I. M. Gelfand, S. M. Fomin, Calculus of Variations, Englewood Cliffs, N. J., Prentice-Hall, 1963.

[19] R. G. Gallager, Information Theory and Reliable Communication, Wiley, 1968.