CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

ON SPECTRALLY EFFICIENT DIGITAL TRANSMISSION SYSTEMS

by

Kuang-Tsan Wu

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical Engineering

OTTAWA, Ontario, 1986

ISBN 0-315-33257-3

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

The University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

Two candidate modulation techniques, i.e., 256-QAM (Quadrature Amplitude Modulation) and 225-QPRS (Quadrature Partial-Response Signalling) are investigated in order to achieve a spectral efficiency of more than 6 b/s/Hz which is required for the transmission of a T1 carrier in a standard analog supergroup band and for many other applications.

A new data filtering strategy (chopping technique) for PRS/QPRS above the Nyquist rate, which enables a higher transmission rate than feasible with the conventional method, is introduced. It is demonstrated that, with the use of our technique for class-I 225-QPRS, an efficiency of 6.3 b/s/Hz can be achieved. The advantages of combining our technique with class-IV PRS are that a small bandwidth around dc can be used for a service channel and that a higher increase in spectral efficiency can be obtained in an FDM-SSB system. However, to achieve an efficiency of more than 6.3 b/s/Hz, 256-QAM is recommended.

To investigate the hardware feasibility of the concept of our new chopping technique, a 7-level class-IV PRS system operated above the Nyquist rate is designed and implemented. Measured results are shown to agree well with the simulated ones.

A quantitative evaluation of the effect of linear channel distortions on 256-QAM and 225-QPRS is presented to compare the sensitivities of these schemes and the numerical results could show modem designers and researchers how flat and how linear the filters' amplitude and phase responses should be.

To enable a rigorous comparison of the conventional method and our new method for PRS above the Nyquist rate, a truncated peak distortion is defined and a new method is presented to evaluate the error floors of these systems. This method is more general and could be applied also to the study of other transmission systems.

Since co-channel interference (CCI) is one of the major sources of performance impairments, its impact on the performance of 225-QPRS and 256-QAM is evaluated so that their potential for radio systems could be assessed.

Finally, in order to improve the performance of 256-QAM systems degraded by channel distortions, the use of an adaptive complex baseband equalizer is described and simulated.

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to Dr. K. Feher for the many forms of support, guidance, and helpful criticisms he provided in supervising this work. The financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and of Karkar Electronics, Inc., San Francisco is acknowledged. I am grateful to the Department of Electrical Engineering, University of Ottawa, for providing the facilities to carry out this thesis. I would like to thank the advisory committee, consisting of Dr. W. Steenaart, Dr. P. Galko, and Dr. I. Sasase for their constructive suggestions and criticisms.

I have benefited during the past 3 years from a close collaboration with all the members of Digital Communications Research Group at the University of Ottawa, including valuable discussions, comments and the limited but precious leisure time we spent together. They are Messrs. M.X. Chen, J. Coll, M. Hamze, D. Hatzinakos, A. Kucar, Z.Z. Lei, J.X. Li, D. Makrakis, T. Mathiopoulos, K. Nagaraj, H. Ohnishi, D. Prendergast, R.V. Scott, J.S. Seo, K. Sreenath, I. Tezcan, A. Yongacoglu, and Drs. M. Sato, M.S. Murthy, I. Sasase, R. Zakaravicius.

My special thank goes to Dr. J. C. Y. Huang. From him and the one-month work he arranged for me at Karkar Electronics, Inc., during November and December, 1984, I learned how to simulate adaptive tran-

sversal equalizers and decision feedback equalizers employed for high-level QAM and QPRS systems to combat channel distortions. I am thankful to Dr. D.D. Falconer, for the helpful discussion on equalization and for his introducing me the adaptive filtering theory.

I am indeed indebted to my family, especially my elder brother, Mr. Kuang-Hui Wu, for fostering my career.

Finally, I wish to thank my parents-in-law for their timely support and encouragement and my wife, Ling-Ling for typing the entire manuscript and for her patience and constant encouragement during my Ph.D. study. Our lovely daughter, Tsai-Yu (or Tati) has been contributing many to our daily life. At the age of three, she was so cooperative, patient and understanding during weekends and long evening hours while I was writing this thesis.

In addition, I acknowledge the examiners of the thesis defence - Dr. S. Mahmoud, Dr. E. Petriu, Dr. H. Sinnreich, Dr. W. Steenaart, and Dr. K. Feher for their comments and contributions to the final version of the thesis.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACI - adjacent channel interference

BER - bit error rate

CCI - co-channel interference

CCITT - international telegraphy & telephone consultative committee

C/N - carrier to noise power ratio

DAV - data above voice/video

DFE - decision feedback equalizer

DIV - data in voice

DUV - data under voice

FFT - fast Fourier transform

ISI - intersymbol interference

MLSD - maximum likelihood sequence detection

PRBS - pseudo random binary sequence

PRS - partial-response signalling

PSD - power spectral density

PSK - phase shift keying

QAM - quadrature amplitude modulation

QPRS - quadrature partial-response signalling

SG - supergroup

SSB - single side band

S/N - signal to noise power ratio

TWT - travelling wave tube

LIST OF SYMBOLS

$\{I_n\}$	i-channel data symbols
$\{Q_n\}$	q-channel data symbols
E	expectation operator
erfc(x)	$= \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2}\, dt$
F	Fourier transform operator
F^{-1}	inverse Fourier transform operator
f_b	bit rate
f_o	carrier frequency
f_s	symbol rate
N_o	one-sided power spectral density of the AWGN (additive white Gaussian noise)
Pr	Probability
P_e	the probability of error
Q(x)	$= \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2}\, dt$, $\quad \text{erfc}(x) = 2Q(\sqrt{2}\,x)$
T_s	symbol interval
T_b	bit interval
sgn(x)	$= 1$ if $x \geq 0$ $= -1$ if $x < 0$
$\delta(t)$	Dirac delta function, $\int_{-\infty}^{\infty} \delta(t)\, dt = 1$, $\delta(t) = 0$ for $t \neq 0$.

CONTENTS

Chapter I

INTRODUCTION

1.1 THE DEMAND FOR HIGHLY SPECTRUM-EFFICIENT MODULATION SCHEMES

The need to increase digital channel capacity over the past few years has led to the development of spectrally efficient modulation systems, such as 8-PSK, 9-QPRS, 16-QAM, 49-QPRS, and 64-QAM [1]-[14]. Due to the increasing need for more efficient utilization of spectrum in the digital radio world and especially in hybrid analog and digital channels, such as Data-In-Voice (DIV), Data-Under-Voice (DUV), and Data-Above-Voice/Video (DAV) [15], digital modulation schemes which achieve an even higher spectral efficiency are required. Before going any further, let us briefly mention the need for hybrid systems.

1.1.1 The Need for Hybrid Systems

It is well known that, on all continents, there were high investments in operational analog multiplex and radio systems, which were originally designed to carry only frequency-division multiplexed (FDM) voice signals. During the last two decades, data traffic has been growing rapidly. However, installation of a new digital transcontinental system would cost enormously in research, development, design, and installa-

tion efforts. Consequently, there is an increasing tendency to make use of existing analog FDM microwave systems as digital carriers.

The primary objectives of hybrid transmission techniques include minimal or no modification to existing hardware, minimal degradation to existing analog performance, and of course, efficient data transmission with a low probability of error.

With the recent developments of high capacity SSB analog microwave systems, the interest in hybrid microwave systems, particularly in DIV systems increases significantly [17]. SSB analog microwave systems have a capacity of up to 6000 toll-grade telephone voice channels in a 30 MHz radio frequency (RF) bandwidth. These SSB systems improve the efficiency of FDM FM microwave systems from 2400 voice channels to 6000 channels in the same bandwidth. There is a strong demand in addition to telephony voice transmission for data transmission on these long distance systems.

In [17], the following conclusion has been reached. For a number of applications, if it is required to carry data at a rate from 56 kb/s up to 8.448 Mb/s, it is more economical to modify an FDM FM, FDM SSB, or FDM baseband cable system into a hybrid system than to install a completely new, dedicated digital system. One specific DIV application for the transmission of a T1 carrier in a standard analog supergroup (SG), call it T1/SG, is described in the following section.

1.1.2 . T1/SG : The Required Spectral Efficiency and its Advantages

According to the International Telegraphy & Telephone Consultative Committee (CCITT) modulation plan [16], a supergroup (SG) contains five standard CCITT groups, equivalent to 60 voice channels. Each voice channel is a nominal 4-kHz voice channel occupying the 300-to-3400 Hz spectrum. The standard SG before translation occupies the frequency band 312 to 552 kHz. Consider the transmission of a T1 carrier (1.544 Mb/s rate data) in a standard SG band (T1/SG) [17]. In Fig.1.1, the power spectrum of a T1/SG DIV system is illustrated as an example, where one of the SG voice signals has been replaced by a digitally modulated T1 signal. It may be required to add bits for forward error correction (FEC), system control/monitoring and synchronization purposes. In this case, the transmission rate is 1.6 Mb/s. In order to transmit 1.6 Mb/s data in a 256 kHz SG band (including guard bands) a spectral efficiency of 1.6 Mb/s : 256 kHz=6.25 b/s/Hz is required. Moreover, to transmit 1.6 Mb/s data over CCITT standard SG through-connect 240 kHz filters requires a spectral efficiency of 1.6 Mb/s : 240 kHz=6.67 b/s/Hz. Thus it is apparent that a digital modem having a spectral efficiency of more than 6 b/s/Hz is needed for DIV applications.

The description of DIV approach would not be complete without a comment on its advantages. Major advantages of the T1/SG DIV approach include [17]:

1) An economical means to transmit one or more T1 rate data streams over a single FM or SSB radio system. Up to 100 separate T1

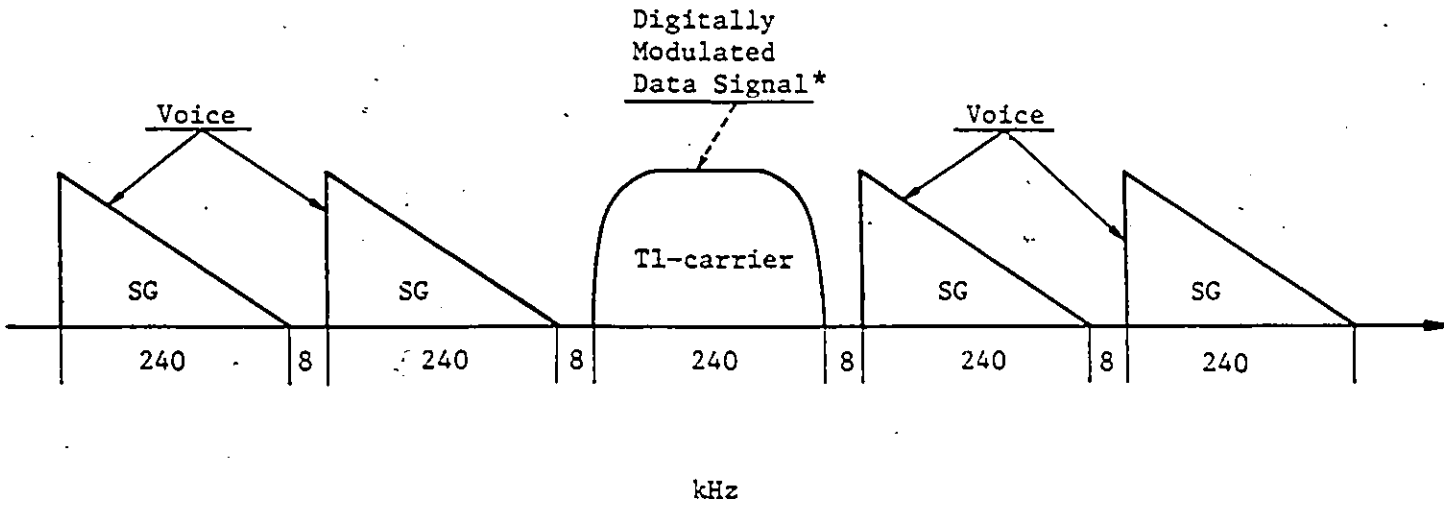


Fig.1.1 Illustrative power spectrum of a T1/SG DIV signal in a standard CCITT mastergroup.
A standard supergroup (SG) before translation occupies the frequency band 312 to 552 kHz.

* e.g., 256-QAM.

signals, having a total capacity of 154.44 Mb/s (100 times T1) could be transmitted over a 30 MHz SSB system.

ii) Flexible operational requirements - with this technique, it is simple to make changes. Data traffic capacity can be added to existing analog routes.

iii) BER performance of less than 10^{-8} can be achieved.

iv) Regenerative span over 1000 km - there is no need for regenerative modems for more than 1000 km. This is in sharp contrast with dedicated digital radio systems which require regenerative modems typically spaced at every 60 km.

v) 50% more efficient than 64-QAM digital radio - a spectral efficiency which is about 50% higher than currently attainable with 4.5 b/s/Hz 64-QAM radio systems is achieved.

vi) No selective fade problems - due to the narrow (240kHz) bandwidth of DIV modems, the selective fade caused problems are virtually non-existent. Adaptive equalizers easily compensate for transmission impairments.

1.1.3 Candidate Modulation Techniques

The spectral efficiency in terms of b/s/Hz and required carrier-to-noise ratio (C/N) at $P_e = 10^{-8}$ of various modulated systems are summarized in Fig.1.2 [14],[15]. A glance at Fig.1.2 has made M-ary PSK out of consideration for the aforementioned application since it requires a higher C/N as compared to QAM and QPRS systems in order to achieve a spectral efficiency of more than 3 b/s/Hz.

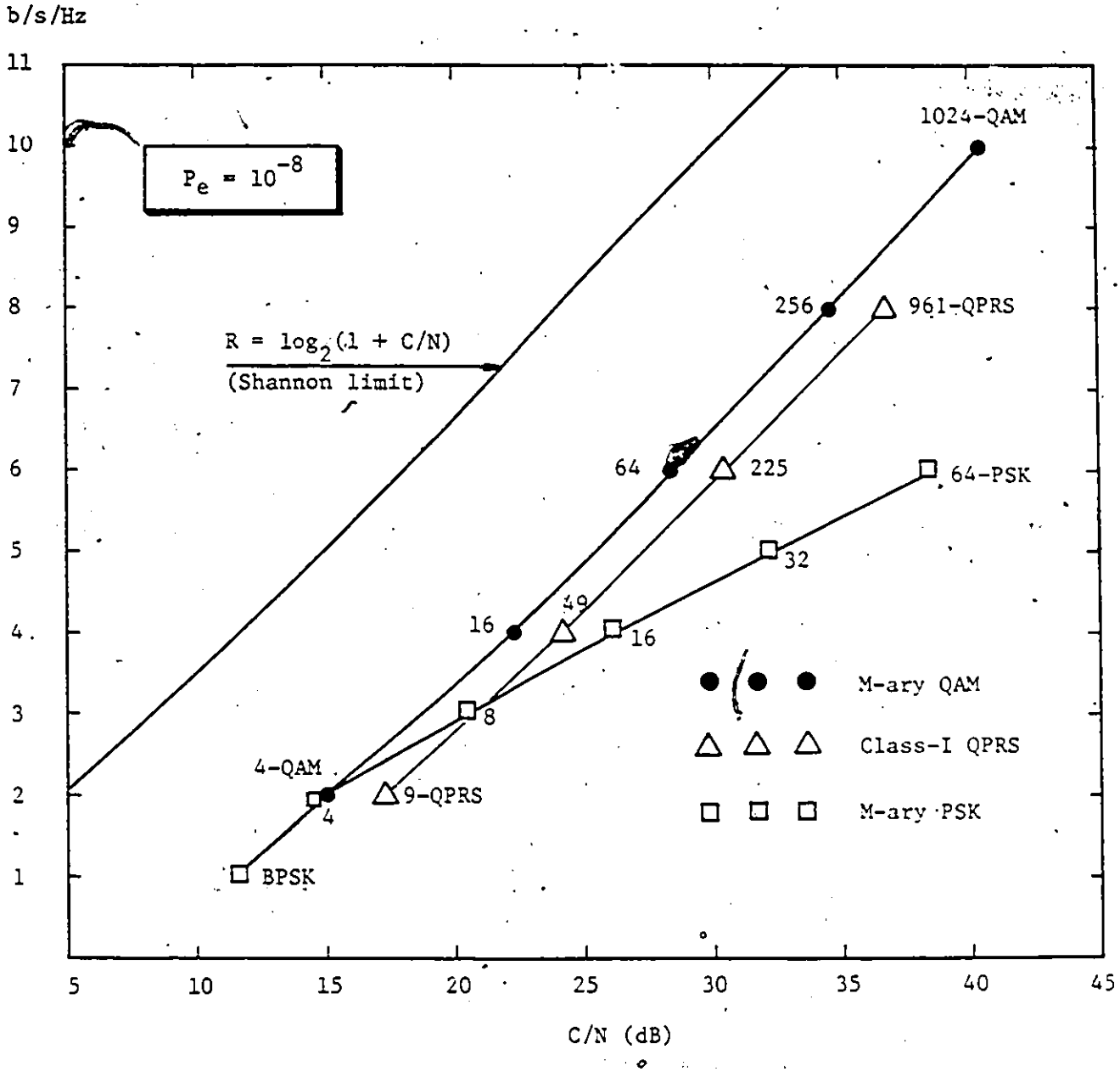


Fig.1.2 Spectral efficiency (b/s/Hz) and C/N requirement at $P_e = 10^{-8}$ of various modulation systems.
The average C/N is specified in the double-sided Nyquist bandwidth which equals the symbol rate. Ideal $\alpha = 0$ filtering has been assumed. (Adopted from Ref. [14], [15], and [17])

From Fig.1.2, we note that the theoretical spectral efficiency of 256-QAM modems is 8 b/s/Hz. An α=0.2 raised-cosine filtered 256-QAM could achieve a practical efficiency of 8 b/s/Hz : 1.2=6.67 b/s/Hz, which is required for the above-mentioned T1/SG DIV application.

An alternative modulation technique 225-QPRS achieves a spectral efficiency of 6 b/s/Hz assuming transmission at the Nyquist rate. It is well known that partial-response signalling (PRS) systems are speed tolerant [18]-[21], i.e., that it is possible to transmit data at a rate which is higher than the Nyquist rate. It should be noted that speed tolerance of PRS systems, reported in the literature [18]-[21], was evaluated for binary inputs, e.g., class-I PRS has a speed tolerance of 43%. In the above 1.6 Mb/s application example, can we use 225-QPRS to meet the efficiency requirement? To achieve 6.25 b/s/Hz by means of 225-QPRS requires an increased transmission rate of 6.25 : 6=1.042 that is 4.2% higher than the Nyquist rate.

From Fig.1.2, we see that the required carrier-to-noise ratio (C/N) for 225-QPRS at $P_e=10^{-8}$ is 30.5 dB, whereas 34.5 dB for 256-QAM. Thus, a potential advantage of 225-QPRS above the Nyquist rate is that an increase in carrier-to-noise ratio (C/N) immunity might be obtained as compared to 256-QAM. The performance of PRS/QPRS above the Nyquist rate with m-ary inputs rather than binary was not available before our investigation [22]. These led to the motivation of investigating the performance of multi-level PRS/QPRS above the Nyquist rate.

1.1.4 Signalling Above the Nyquist Rate

Signalling above the Nyquist rate was originally discussed for PRS systems for binary inputs [18]-[21]. Recently, a few schemes concerning faster-than-Nyquist signalling were addressed [23]-[25]. These schemes exploit other pulse shapes also, such as Nyquist pulses, not limited only to PRS pulses. However, there is a common feature in all the schemes mentioned so far. That is, signalling above the Nyquist rate is achieved by simply increasing the signalling rate with the originally designed transmission characteristic being kept fixed . As such, it introduces too much "undesired" intersymbol interference (ISI) which causes significant performance degradation by simple symbol-by-symbol detection. Maximum Likelihood Sequence Detection (MLSD) may be employed to overcome the consequent ISI in the best way possible [27]. However, the difficulty lies in the fact that the complexity of implementation may be beyond the capability of the present technology. For example, a number like 10^8 has been chosen as the number of states required by the Viterbi algorithm for decoding the Faster Binary Signalling (FBS) signal [24].

In this thesis, we do not insist on optimum receivers [26], and other advanced nonlinear receivers, such as MLSD and decision feedback equalizers (DFE) [27]-[32] for PRS/QPRS above the Nyquist rate because of the complexity involved in implementation. Instead, a symbol-by-symbol detection procedure is assumed because of easy implementation. We introduce a new method for PRS/QPRS above the Nyquist rate which allows the use of symbol-by-symbol detection and achieves a better performance than the conventional method.

1.2 THESIS ORGANIZATION

Following this introductory chapter, in Chapter 2 we discuss multi-level PRS/QPRS above the Nyquist rate. It is shown that the conventional method is not practical for 225-QPRS systems operated above the Nyquist rate. An improved efficiency PRS/QPRS data filtering strategy, which enables a higher transmission rate (more symbols/s/Hz in a normalized bandwidth) than feasible with conventional PRS/QPRS systems, is then introduced. Though the discussion in Chapter 2 emphasizes class-I (duobinary) PRS, it preserves sufficient generality to accommodate other classes of PRS.

In some systems, such as transformer coupled circuits, SSB modems, and carrier systems with carrier pilot tones, where reduced low-frequency components in the spectrum are required, class-IV (modified duobinary) PRS is more suitable than class-I. Thus, the study on the possibility of increasing the spectral efficiency above the Nyquist rate using class-IV PRS with multi-level inputs is presented in Chapter 3. We show that, by using our new technique, the performance of a class-IV 225-QPRS above the Nyquist rate is nearly as good as that of using class-I PRS. Additional advantages of class-IV PRS above the Nyquist rate using our new technique are that a service channel can be inserted and that a higher percentage increase in spectral efficiency can be obtained in an FDM SSB system.

In Chapter 4, we present the implementation of an experimental 7-level class-IV PRS system operated above the Nyquist rate using our new method. The performance is measured to confirm the concept and simulation accuracy.

The effect of channel and/or hardware imperfections on a 256-QAM modem is investigated in Chapter 5. The next chapter, Chapter 6, compares the performance of 256-QAM and improved efficiency 225-QPRS in distorted channels, based on the same spectral efficiency of 6.3 b/s/Hz. The effects of group delay and amplitude distortions with linear, parabolic, and sinusoidal characteristics on these two schemes are evaluated and compared.

In Chapter 7 we introduce and study error floors in digital transmission systems. Error floors are known in industry as one of the most severe problems of transmission systems. We first describe a new method to evaluate and compare the error floors of PRS systems operated above the Nyquist rate using the conventional and our new models. The concept is extended to evaluate error floors of PAM systems under severe distortions.

The discussion of co-channel interference (CCI) caused performance degradations on 64-QAM, 225-QPRS and 256-QAM is presented in Chapter 8.

Chapter 9 discusses the use of an adaptive baseband transversal equalizer for 256-QAM to combat channel distortions. Finally, Chapter 10 summarizes the results of the thesis and suggestions for further research.

Chapter II

MULTI-LEVEL PRS/QPRS ABOVE THE NYQUIST RATE

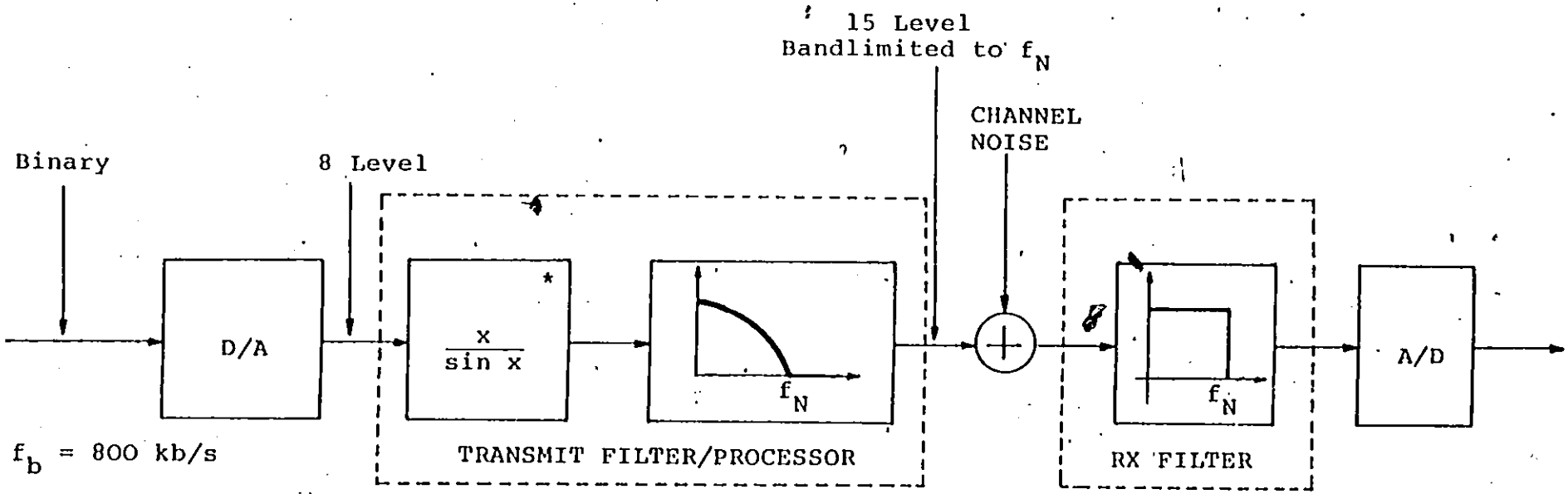
It is well known that partial-response signalling (PRS) systems are speed tolerant [18]-[21], i.e., that it is possible to transmit data at a rate which is higher than the Nyquist rate. Speed tolerance was defined [18]-[20] as the percentage increase over the Nyquist rate at which the smallest eye opening is zero with the transmission characteristic being kept fixed for binary inputs. Due to inherently higher intersymbol-interference (ISI) in multi-level systems, speed tolerance is expected to decrease as the number of input levels increases. However, it is not clear yet from the literature how speed tolerance would be changed for PRS systems with m-ary inputs rather than binary. And, as mentioned in Chapter 1, there is a need for highly spectrum-efficient modulation schemes. Therefore, it is important to study the performance of PRS/QPRS above the Nyquist rate in order that the feasibility of increasing the spectral efficiency for PRS/QPRS systems can be assessed. Since duobinary (class-I) PRS with binary inputs has a speed tolerance of 43% [18]-[20], which is the largest among various classes, it is logical to choose it as a starting point for studying PRS systems signalling above the Nyquist rate with m-ary inputs.

In Section 2.1 two models for duobinary PRS systems above the Nyquist rate with m-ary inputs are described, with Model-A represent-

ing the conventional method as speed tolerance was discussed [18]-[20] and Modal-B standing for the new chopping method we propose. Based on a comparison from both the frequency and time domains, we show that Model-B is superior to Model-A. A few selected eye diagrams presented in Section 2.2, manifest the superiority of our new technique over the conventional method. In Section 2.3 experimental results are demonstrated to confirm that predicted by computer simulations. Evaluation of the probability of error for 225-QPRS above the Nyquist rate is given in Section 2.4. Results show that, by symbol-by-symbol detection, it is not practical to use the conventional method for 225-QPRS above Nyquist rate. However, with the use of our new technique (Model-B) for 225-QPRS, an increase of 5% above the Nyquist rate can be achieved with a small degradation. This increase is significant in some applications.

2.1 SYSTEM MODELS

Multi-level PRS and QPRS system block diagrams are shown in Fig.2.1(a) and (b). An illustrative example would be the transmission of an f_b= 800 kb/s rate data with a 15-level PRS system filtered at the Nyquist frequency of 133 kHz that is 6 b/s/Hz. In the QPRS Modem case the source rate at 1600 kb/s is serial-to-parallel converted and fed into the "inphase" and "quadrature" 800 kb/s channels of the modulator. In Fig.2.1(b), the transmit filter/processors and receive filters are identical with the ones illustrated in Fig. 2.1(a).



Also known as Duobinary encoder

* The $\frac{x}{\sin x}$ aperture equalizer is employed to convert the multi-level pulse stream into an "impulse stream".

Fig.2.1(a) Block diagram of a conceptual duobinary PRS baseband system.

Illustrative example : Source rate f_b=800 kb/s; Symbol rate of D/A output, assuming an 8-level output is used, is f_s=266 k symbol/s. In this case f_N=133 kHz. Thus, the spectral efficiency of the 8-level PAM signal, which gets converted to 15-level PRS is 800 kb/s : 133 kHz = 6 b/s/Hz.

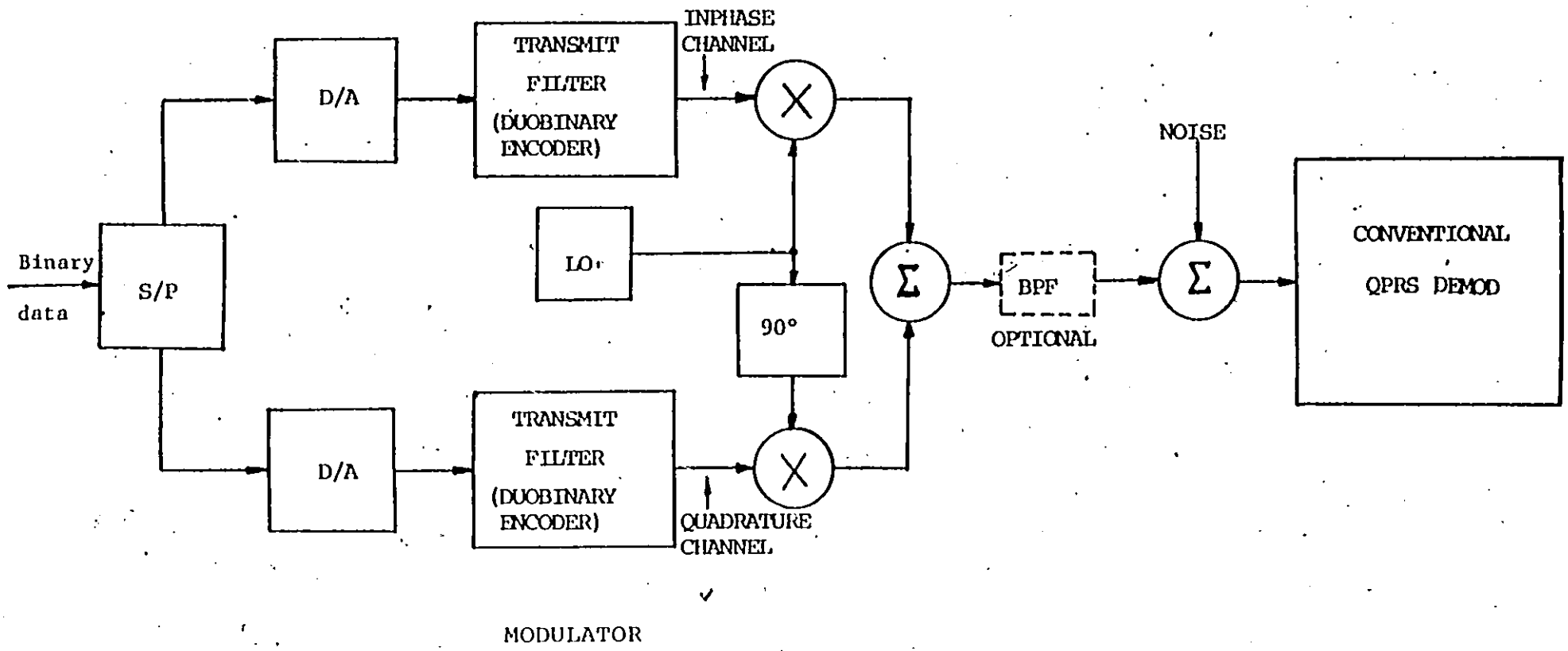


Fig.2.1(b) QPRS modem block diagram.
The duobinary encoders in the in-phase and quadrature channels are identical to the ones in Fig.2.1(a).

Transmission above the Nyquist rate by PRS can be achieved by increasing the signalling rate with the transmission characteristic being kept fixed, as is the case in which speed tolerance was evaluated in the published literature [18]-[20]. This conventional method will be called Model-A, as depicted in Fig.2.2. With Model-A, one transmits data at a rate of $f_s=(1+a)/T$ symbols/s over a bandwidth of 1/2T Hz, resulting in an efficiency of $2(1+a)$ symbols/s/Hz where 100a represents the percentage increase above the Nyquist rate. In fact, Model-A consists of forcing a symbol rate through a duobinary filter that is higher than the rate for which the filter is designed. As such, it will introduce serious "undesired" ISI and result in very significant performance degradations.

It will soon be clear that Model-A does not give good results concerning the probability of error, especially when multi-level inputs are used. Hence, we propose a new scheme called Model-B, as illustrated in Fig.2.3. Our Model-B consists essentially of defining a new (infinite) partial-response system in the following manner:

i) Define an ideal duobinary response for the new higher symbol rate.

ii) Truncate this response to achieve the desired spectral efficiency using an ideal brickwall filter ("chopping") which has a cutoff frequency at $f_N/(1+a)$ below the Nyquist frequency.

In Figs.2.1-2.3, we did not include the digital processing parts, such as precoding to avoid error propagation, Gray coding, etc. These pro-

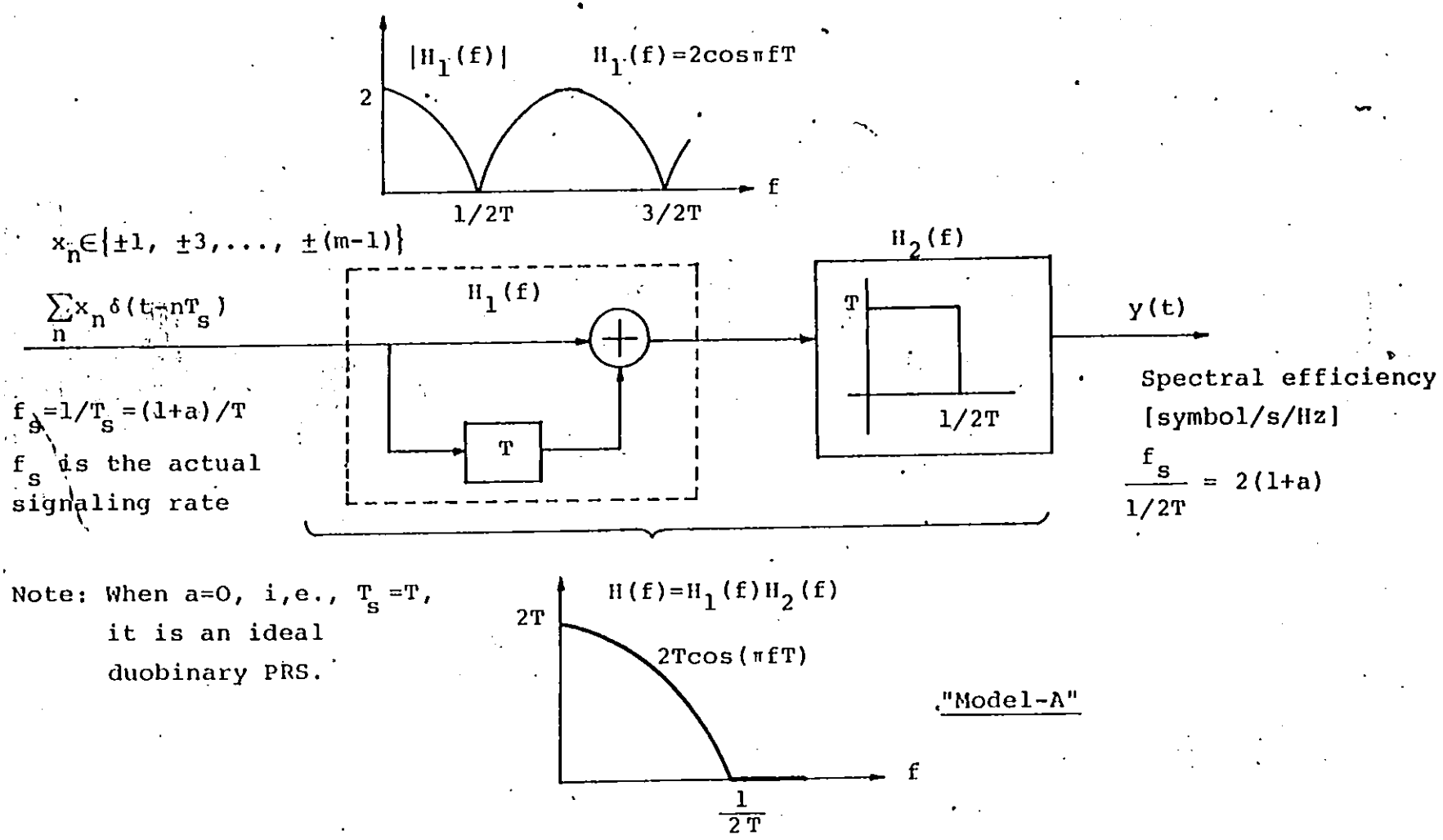
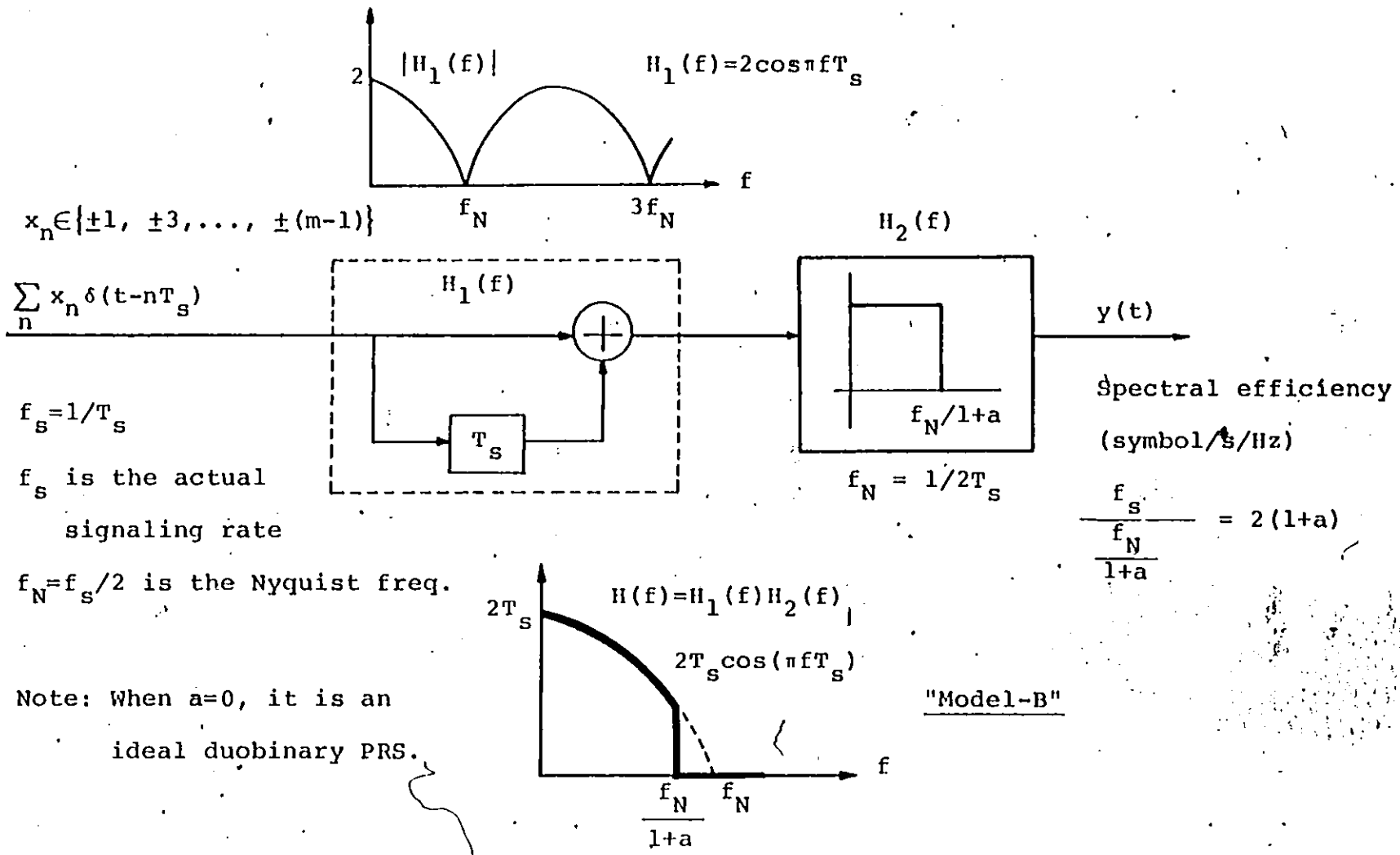


Fig.2.2 Model-A : Conventional system model for duobinary PRS above the Nyquist rate and cascaded transfer function. The spectral efficiency of this system is 2(1+a) symbol/s/Hz, i.e., 100a% higher than the so-called Nyquist rate of 2 symbol/s/Hz.



Note: When a=0, it is an ideal duobinary PRS.

"Model-B"

Fig.2.3 Model-B : New system model for duobinary PRS above the Nyquist rate and corresponding cascaded transfer function. The spectral efficiency of this system is 2(1+a) symbol/s/Hz, i.e., 100a% higher than the Nyquist rate of 2 symbol/s/Hz.

cessors were described in sufficient detail in [19]-[21], [53]. We shall first compare these two models from the frequency and time domains.

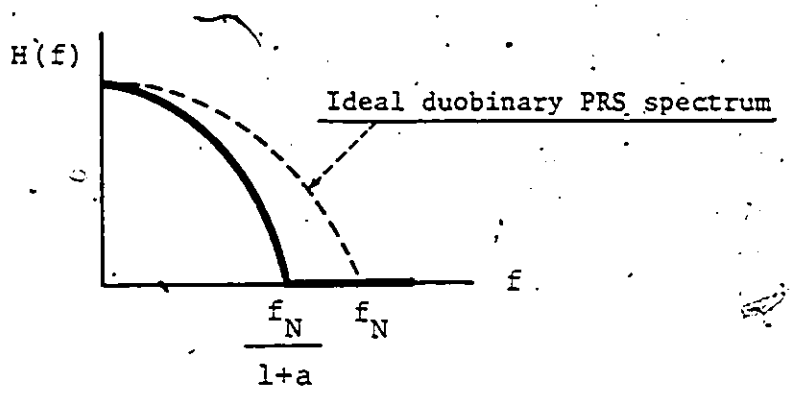
2.2 WHY IS OUR NEW MODEL-B BETTER THAN THE CONVENTIONAL MODEL-A?

In this section, we investigate the reason why our new Model-B is better than the conventional model-A for multi-level PRS/QPRS above the Nyquist rate.

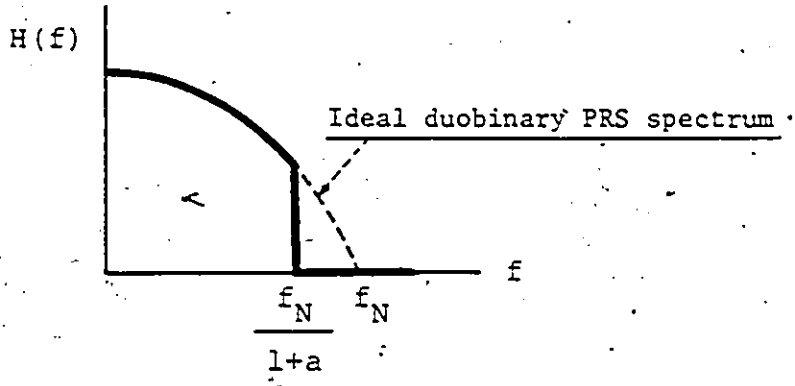
2.2.1 Frequency-Domain Comparison

An equivalent way of comparing operation above the Nyquist rate by PRS for the two models is to keep the signalling rate, f_S, the same but to reduce the bandwidth to less than f_N. Using this viewpoint, operation above the Nyquist rate in the Model-A scheme is illustrated in Fig.2.4(a) in which a scaled cosine shape is retained. Thus, only the frequency scale is changed, resulting in that all the frequency components suffer distortion. On the other hand, in our Model-B scheme shown in Fig.2.4(b), the old frequency response is simply "chopped off" at $f_N/(1+a)$. Most of the original frequency components are thus retained and the only frequency components that suffer distortion are the ones that have been cut off.

From this frequency-domain comparison, we would expect Model-B performs better, especially for multi-level inputs. The superiority of Model-B over Model-A can also be observed in the time domain.



(a) Conventional Model-A



(b) Our new Model-B

Fig.2.4 A frequency-domain comparison for the two models for duobinary PRS above the Nyquist rate.

2.2.2 Time-Domain Comparison

PRS systems operate by introducing a "controlled" or "desired" amount of ISI. When operated above the Nyquist rate, in addition to the "desired" ISI, there is introduced "undesired" ISI, which is responsible for the eventual deterioration of performance to below an acceptable level. We would expect Model-B to have less "undesired" ISI simply due to the fact that in Model-B most of the original frequency components are retained, as discussed earlier. To examine this ISI more rigorously, we shall consider the time domain behavior.

As a first step, expressions for impulse responses of Model-A and Model-B will be derived. Referring to Fig.2.2, the overall frequency response $H(f)$ of conventional Model-A is expressed as follows.

$$H(f) = \begin{cases} 2T\cos(\pi fT), & |f| \le 1/2T \\ 0 & \text{elsewhere.} \end{cases} \qquad (2.1)$$

The signaling rate f_s is related to T by (2.2).

$$f_s = 1/T_s = (1+a)/T \qquad (2.2)$$

The corresponding impulse response $h(t)$ can be obtained by taking the inverse Fourier transform of (2.1), i.e.,

$$\begin{aligned} h(t) &= \int_{-\infty}^{\infty} H(f)\, e^{j2\pi ft}\, df \\ &= 4T \int_0^{1/2T} \cos(\pi fT)\cos(2\pi ft)\, df \\ &= \mathrm{sinc}\left(\frac{t}{T}+\frac{1}{2}\right) + \mathrm{sinc}\left(\frac{t}{T}-\frac{1}{2}\right) \end{aligned} \qquad (2.3)$$

where

$$\mathrm{sinc}(x) = \sin(\pi x)/\pi x$$

For our Model-B case (see Fig.2.3) the overall frequency response H(f) can be written as (2.4).

$$H(f) = \begin{cases} 2T_S \cos(\pi f T_S), & |f| \le f_N'/(1+a) \\ 0 & \text{elsewhere.} \end{cases} \qquad (2.4)$$

where

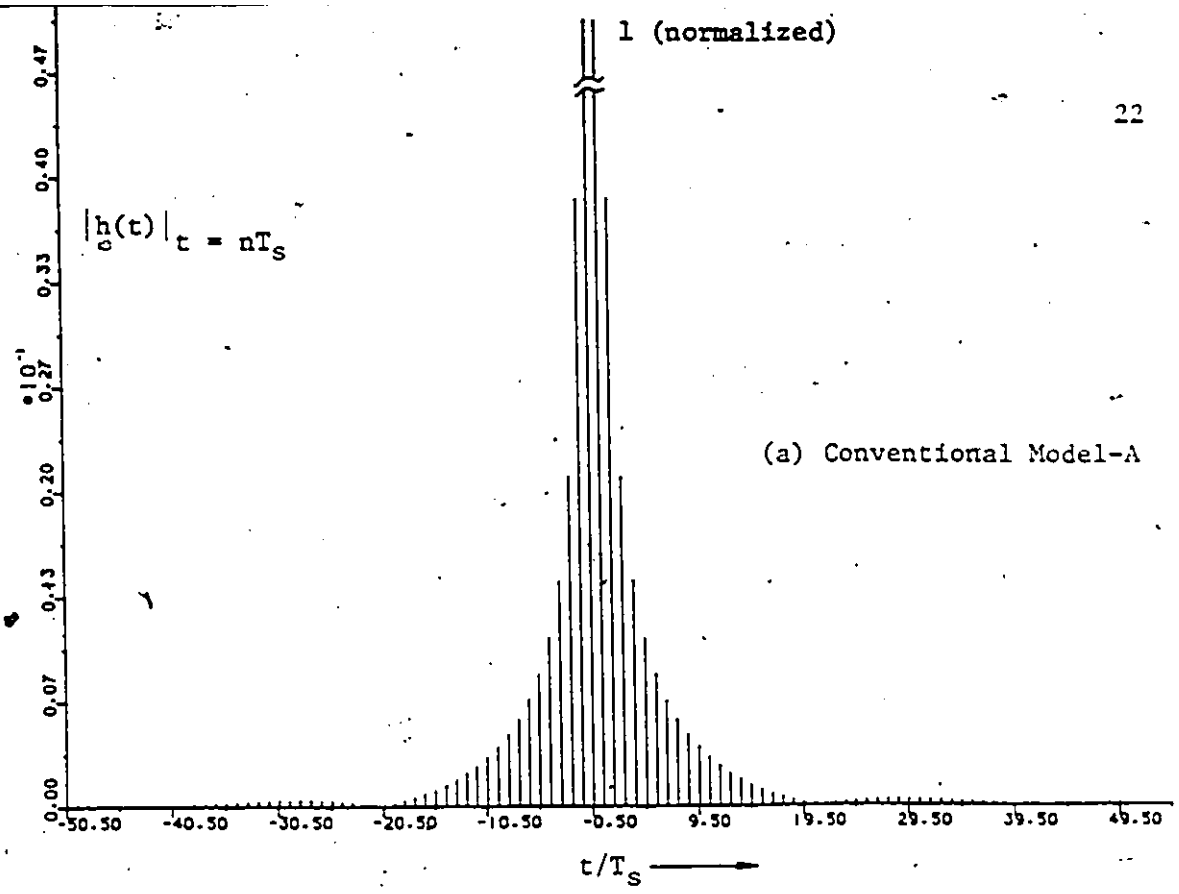
$f_N = 1/T_S$ is the Nyquist frequency of the signal.

Following the similar steps, the impulse response h(t) of Model-B can be derived as follows.

$$\begin{aligned} h(t) &= \int_{-\infty}^{\infty} H(f)\, e^{j2\pi ft}\, df \\ &= 4T_S \int_0^{f_N/(1+a)} \cos(\pi f T_S) \cos(2\pi ft)\, df \\ &= \frac{1}{1+a}\left\{\text{sinc}\left[\frac{1}{1+a}\left(\frac{t}{T}+\frac{1}{2}\right)\right] + \text{sinc}\left[\frac{1}{1+a}\left(\frac{t}{T}-\frac{1}{2}\right)\right]\right\} \end{aligned} \qquad (2.5)$$

Note that in the above derivations, the center of the impulse response has been chosen as the time origin. Thus for the ideal duobinary PRS, i.e., a = 0 for either Model-A or Model-B, the only two nonzero samples are at $t = \pm T_S/2$, respectively.

The discrete-time impulse response $|h(t)|_{t=nT_S}$ is of interest since it provides insight into the behavior of the undesired ISI. We computed $|h(t)|_{t=nT_S}$ by using the equations (2.3) and (2.5) for the two models. For illustrative purposes, we plot $|h(t)|_{t=nT_S}$ of Model-A and Model-B in Figs.2.5 and 2.6 for the same percentage increases of 5% and 10% respectively. It is easily seen that, the undesired ISI's in our new Model-B case are much smaller than those of Model-A, especially



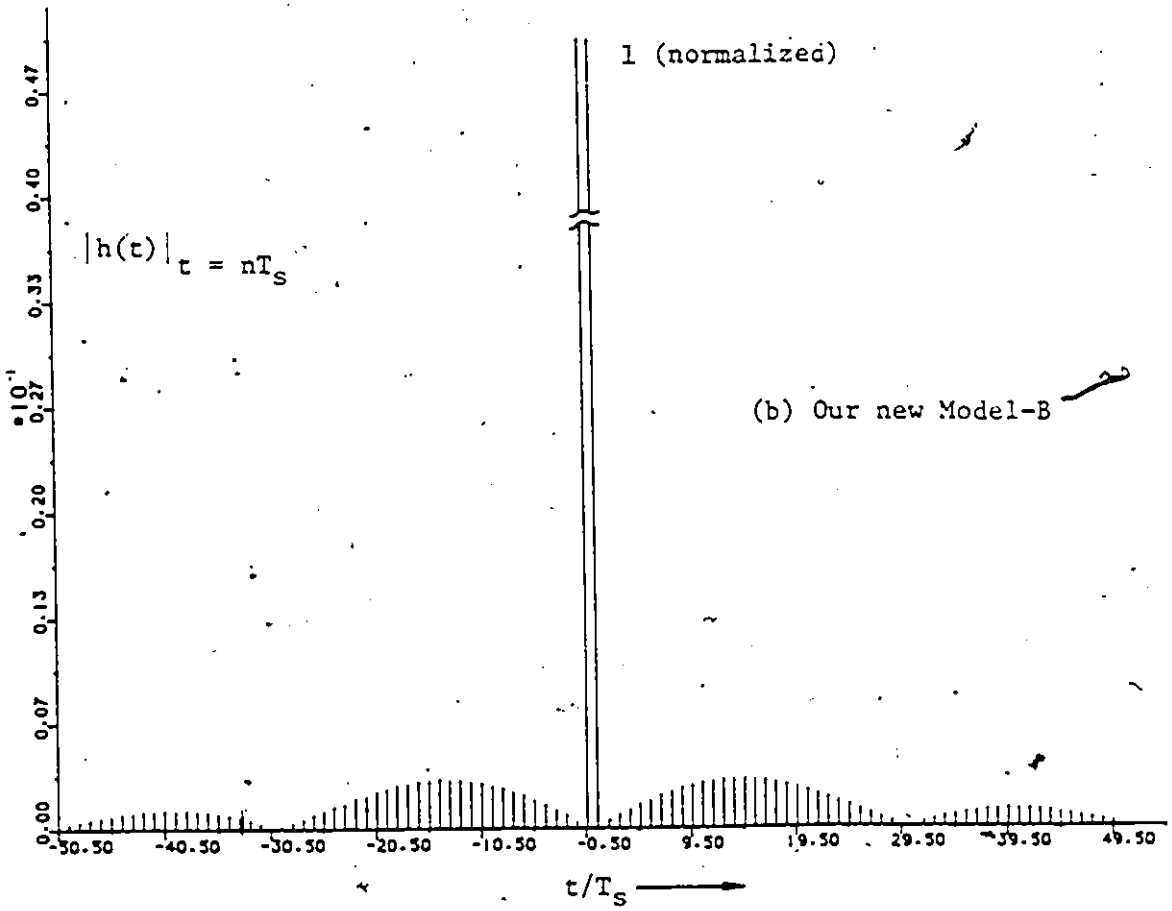


Fig.2.5 Discrete-time impulse response of duobinary (class-I) PRS with a 5% increase above the Nyquist rate. (a) Conventional Model-A, (b) Our new Model-B.

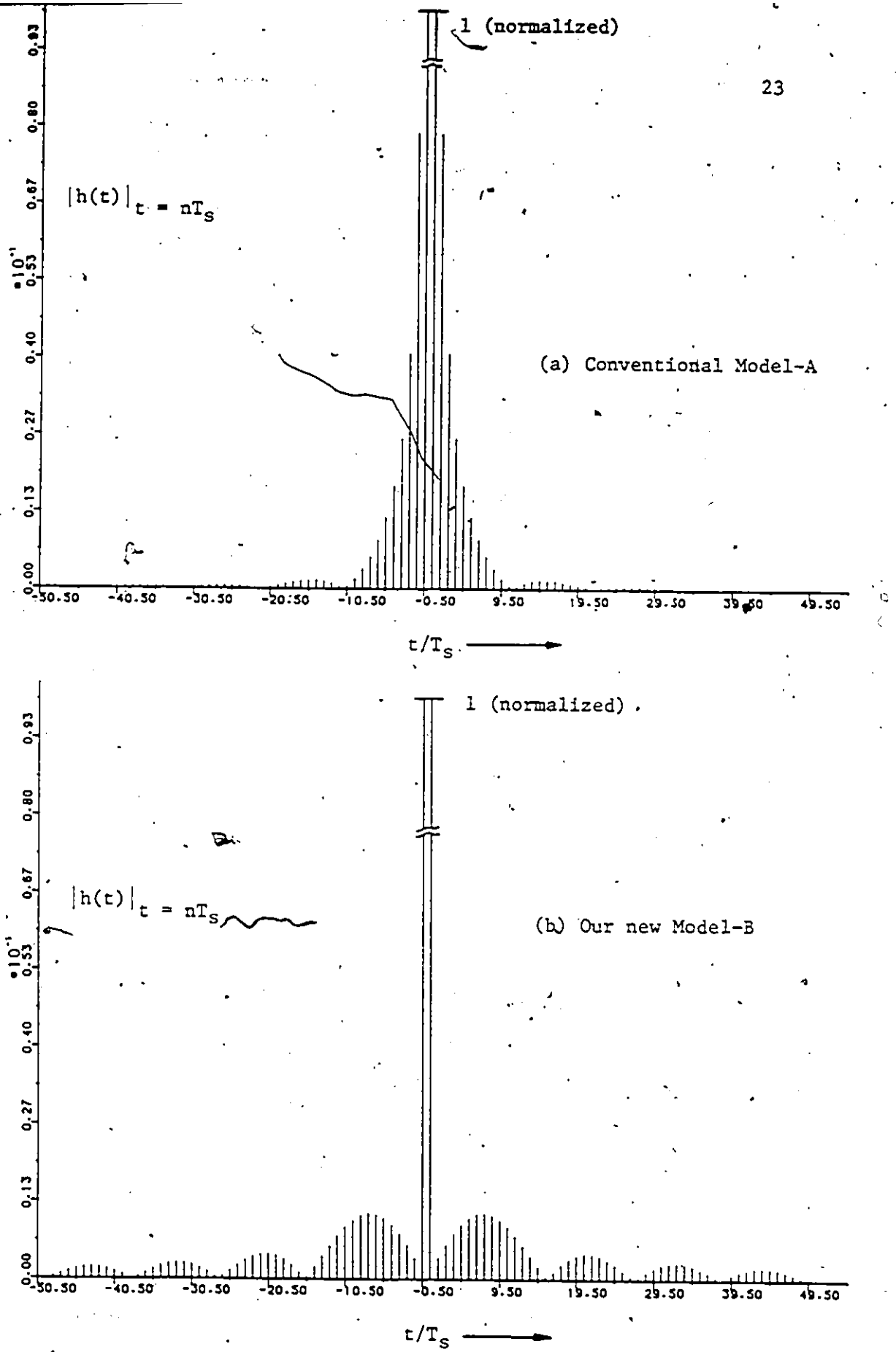


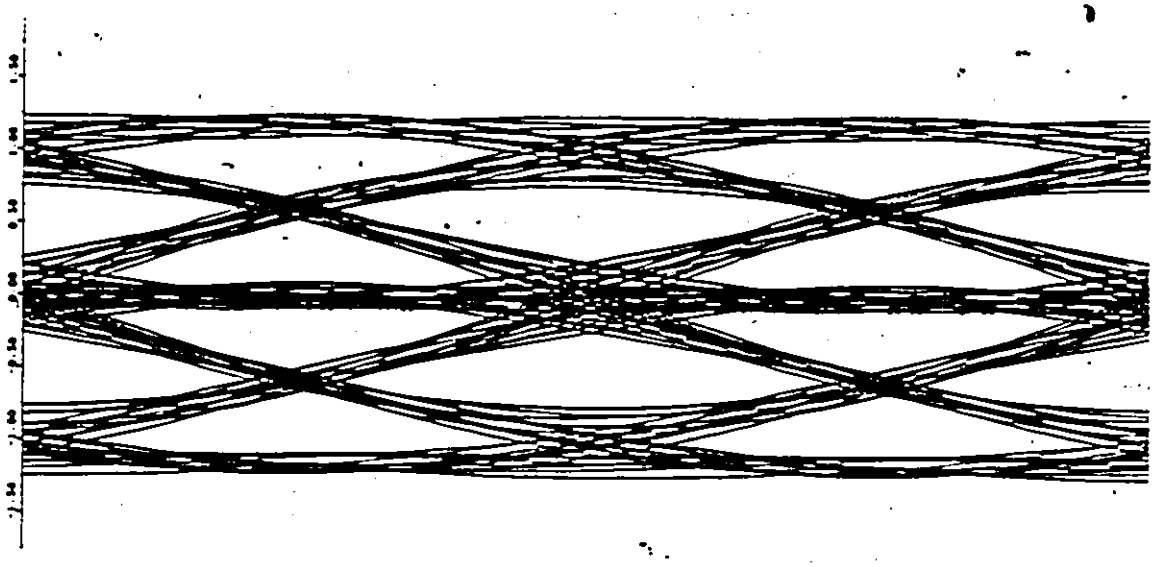
Fig.2.6 Discrete-time impulse response of duobinary (class-I) PRS with a 10% increase above the Nyquist rate. (a) Conventional Model-A, (b) Our new Model-B.

for those near the two desired ISI samples. This confirms our conjecture from the frequency-domain comparison. A more rigorous comparison of the two models which leads to concepts, such as truncated peak distortion and system error floor [22], will be treated in Chapter 7.

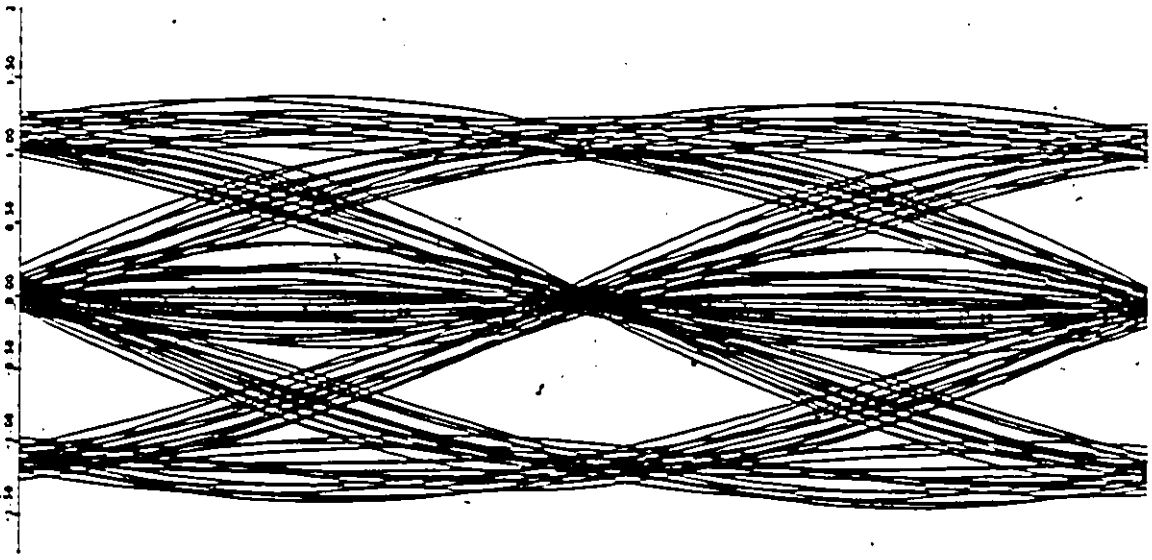
2.3 COMPARISON OF EYE DIAGRAMS

Since eye diagrams show a good indication of system performance, we shall first compare a few typical eye diagrams, generated by computer simulations. The attention will be focused on the vertical eye opening because it gives the noise margin at the optimum sampling instants. The eye diagrams, displayed in Figs.2.7-2.9, represent 9-, 49-, and 225-QPRS, which consist of modulating in quadrature two baseband 3-, 7-, and 15-level PRS systems, respectively, using Model-A or Model-B. The choice of these eye diagrams is, of course, arbitrary but proper for comparison. Fig.2.7(a) and(b) show the eye diagrams of 9-QPRS for Model-A and Model-B with an increase of 20% above the Nyquist rate. Both show that the eye is open. However, the eye opening in Fig.2.7(b) is much larger than that of Fig.2.7(a), indicating clearly that a larger noise margin exists by using our Model-B.

In the case of 49-QPRS with an increase of 10% above the Nyquist rate, Fig.2.8(b) (Model-B) gives a robust eye opening whereas in Fig.2.8(a) the eye is seen about to close. It shows that Model-B is better than Model-A as the number of input levels increases. This trend becomes clearer as we look into 225-QPRS with an increase of 4.2% above the Nyquist rate. While Fig.2.9(b) shows a reasonably good eye

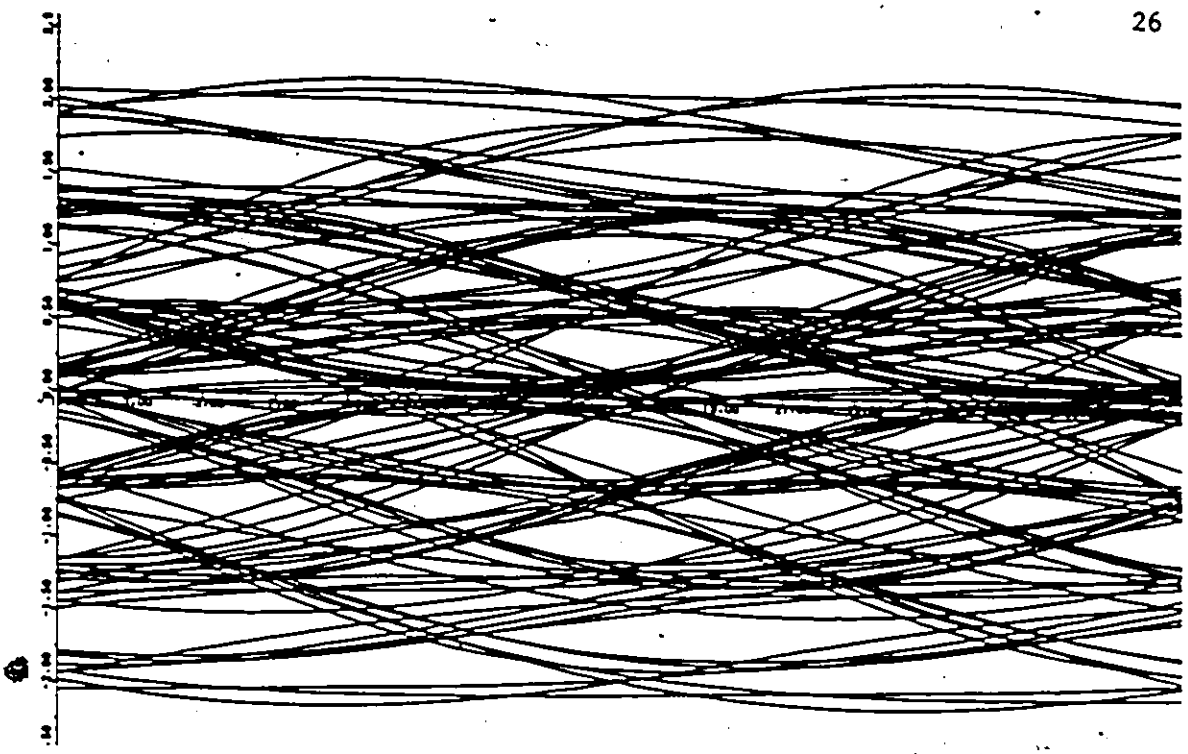


(a) Using the conventional Model-A.

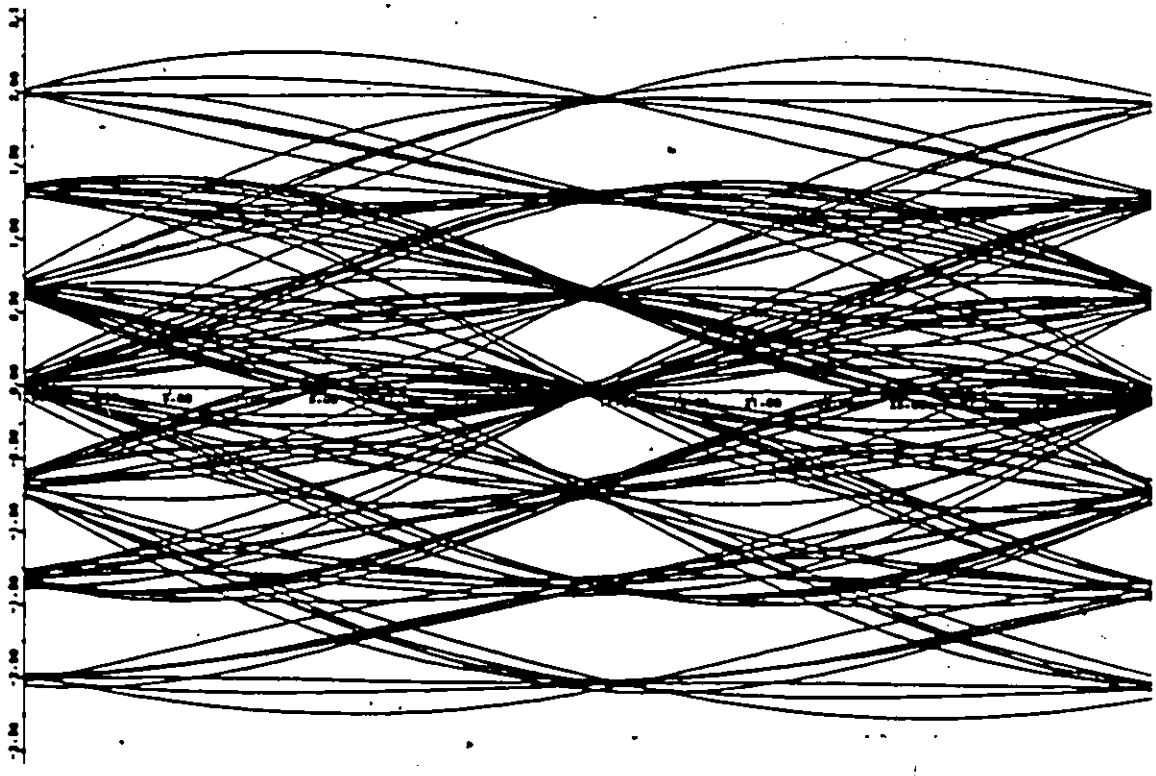


(b) Using our new Model-B.

Fig.2.7 Eye diagrams of 3-level PRS/9-QPRS with a 20% increase above the Nyquist rate. Spectral efficiency: 2.4 b/s/Hz.

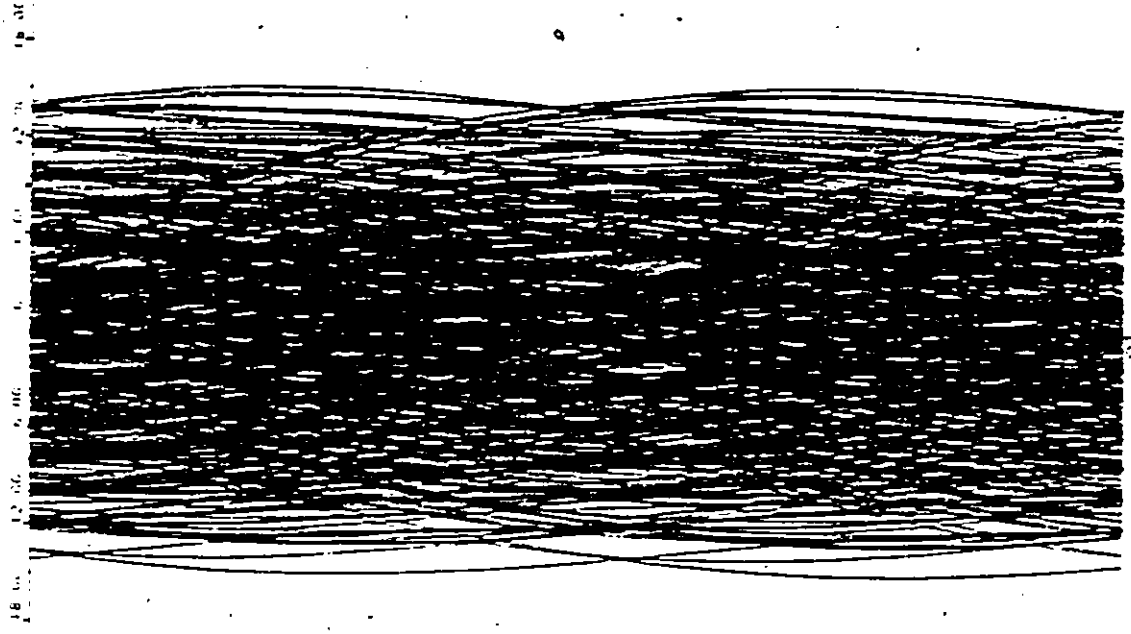


(a) Using the conventional Model-A.

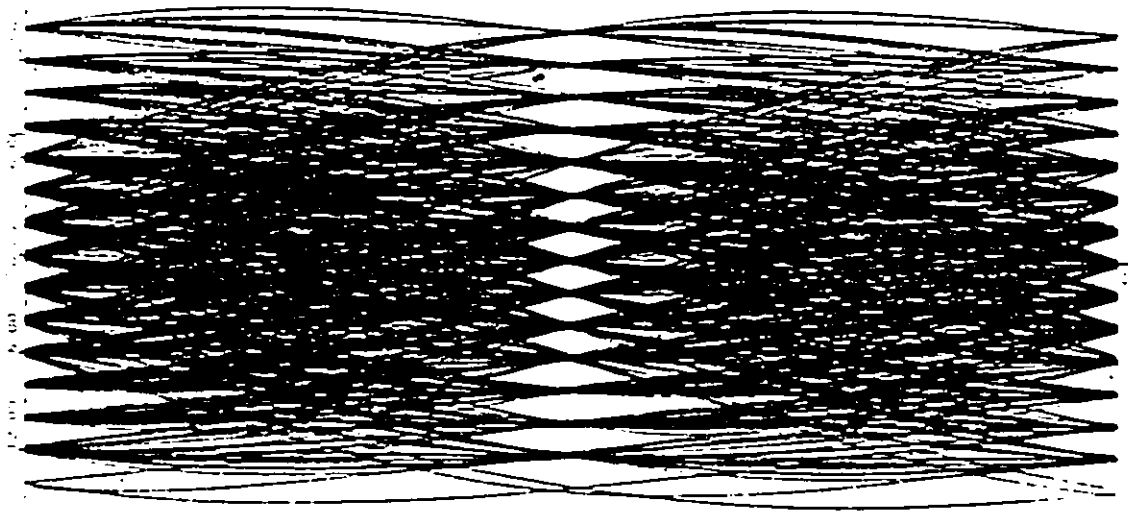


(b) Using our new Model-B.

Fig.2.8 Eye diagrams of 7-level PRS/49-QPRS with a 10% increase above the Nyquist rate. Spectral efficiency: 4.4 b/s/Hz.



(a) Using the conventional Model-A.



(b) Using our new Model-B.

Fig.2.9 Eye diagrams of 15-level PRS/225-QPRS with a 4.2% increase above the Nyquist rate. Spectral efficiency: 6.25 b/s/Hz.

opening, in Fig.2.9(a) the eye is completely closed, meaning that in this case symbol-by-symbol detection fails to decode the signal for the conventional Model-A.

From the study of eye diagrams, we found that speed tolerance of duobinary PRS has reduced from 43% for binary inputs to about 4% for 8-level inputs by using the conventional Model-A. On the other hand, with the use of our new Model-B it is possible to achieve a higher spectral efficiency for PRS/QPRS systems.

2.4 EXPERIMENTAL RESULTS FOR A 15-LEVEL PRS USING OUR NEW MODEL-B

We implemented an above-the-Nyquist-rate transmission system using our Model-B for a 15-level PRS system [33] according to Fig.2.10. The circuit of Fig.2.10 operates as follows. The incoming binary data, at a bit rate of f_b = 800 kb/s, is converted into an eight state symbol at a symbol rate of $f_s = f_b/3$ = 266.66 kBaud by the serial-to-parallel (S/P) converter. The symbol is time aligned by the first set of three "D" type flip-flops. The second set of three "D" type flip-flops provides a unit symbol delay between two successive symbols. These two successive 8 state symbols are then added to provide one 15-state PRS symbol at the symbol rate. Note that the "D" type flip-flops and the binary adder form a digital cosine filter as $H_1(f)$ in Fig.2.3, which can be expressed explicitly as (2.5).

$$\left|H_1(f)\right| = 2\left|\cos(\pi f T_s)\right| \qquad |f| < \infty \tag{2.6}$$

where

$T_s = 1/f_s$ is the signaling symbol interval.

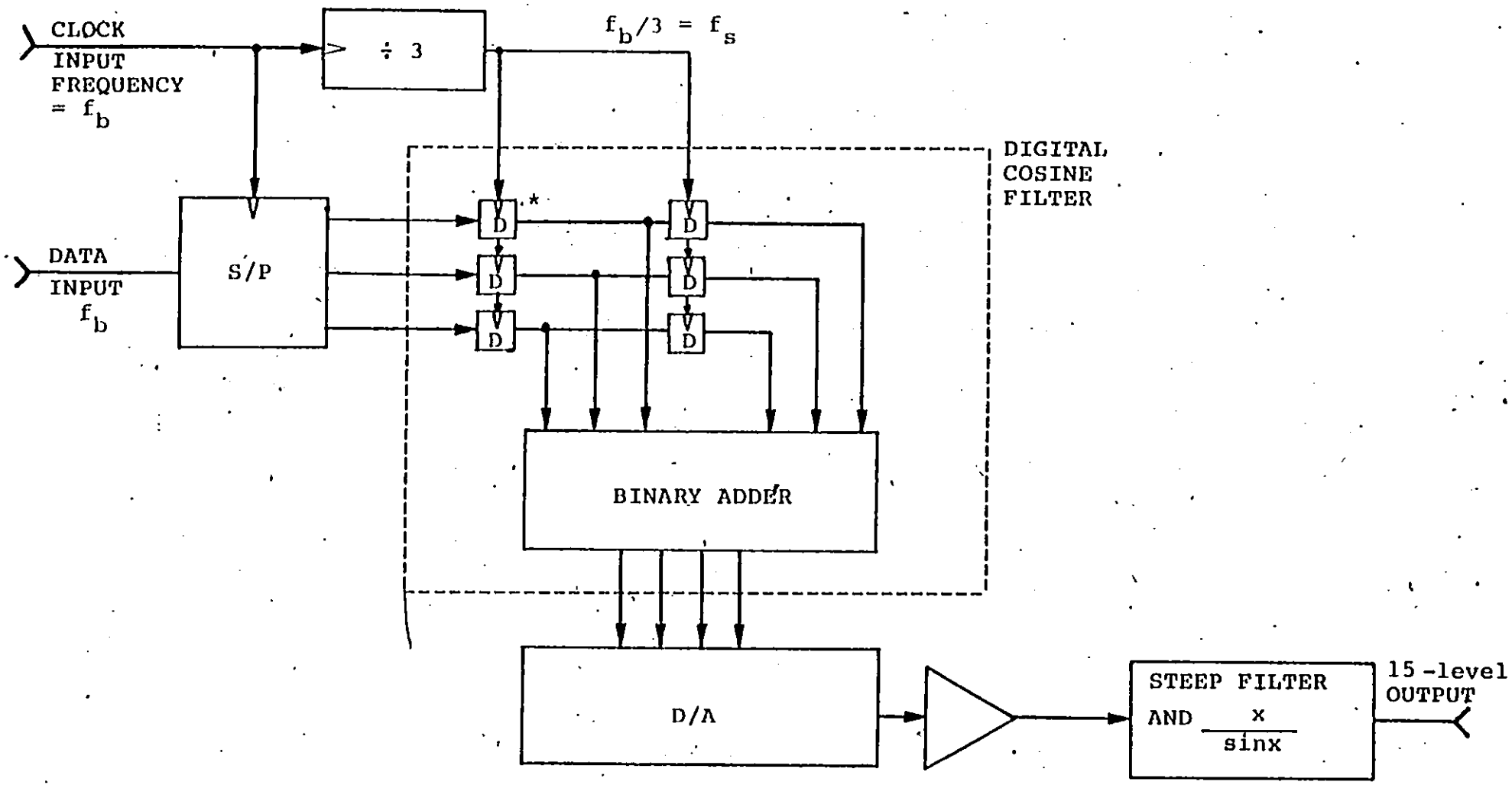


Fig.2.10 Hardware implementation of 15-level duobinary PRS above the Nyquist rate using our new "Model-B".

The 15-state digital symbol is then converted into a 15-level analog symbol by the D/A converter. The amplifier serves as a buffer for the filter and voltage offset for the D/A converter.

The wideband 15-level PRS signal is then sent through a steep filter which has a 50 dB attenuation at $f_c = f_N/(1+a) = 128$kHz, where $f_N = f_S/2 = 133.33$ kHz. It is to be noted that, in practice, non-return to zero (NRZ) rectangular pulses are used instead of impulses. And for a rectangular pulse p(t) defined by

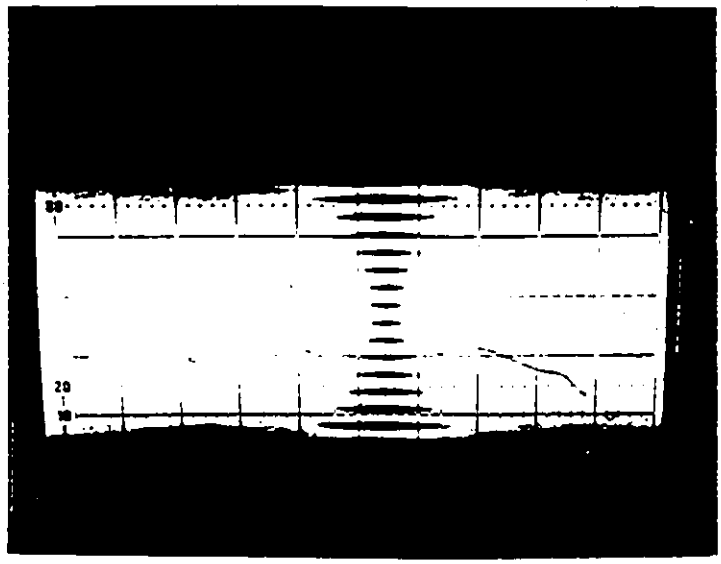
$$p(t) = \begin{cases} A & |t| \leq T_S/2, \\ 0 & \text{elsewhere.} \end{cases} \tag{2.7}$$

The Fourier transform of p(t) is

$$P(f) = F\{p(t)\} = AT_S \frac{\sin(\pi f T_S)}{\pi f T_S} \tag{2.8}$$

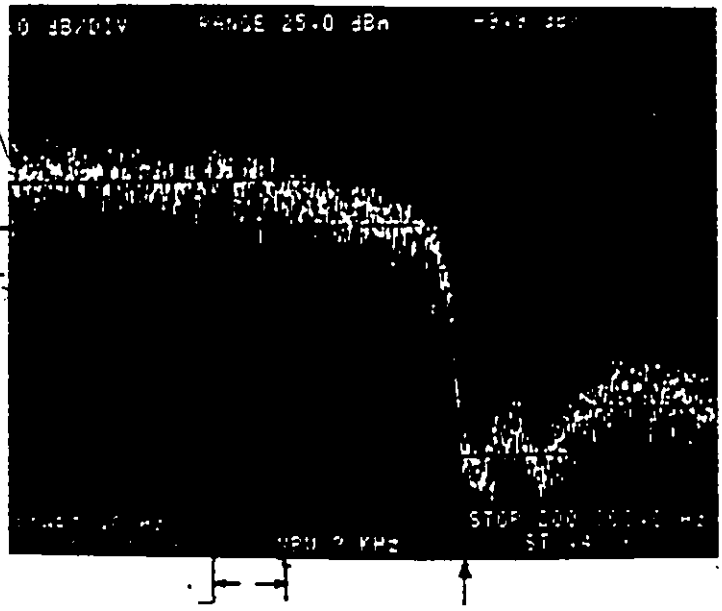
where $T_S = 1/f_S$ is the symbol interval.

Thus, the final transmit filter contains a compensating factor of x/sin(x) to account for the nonreturn-to-zero (NRZ) rectangular pulse (instead of impulse) induced sin(x)/x shaping of the spectrum. The measured eye diagram and corresponding power spectral density for f_b = 800 kb/s and f_c = 128 kHz are shown in Fig.2.11. In this case the spectral efficiency is 800:128 = 6.25 b/s/Hz, i.e., an increase of 4.2% above the Nyquist rate. We note that the measured eye diagram is open as predicted in the simulation result of Fig.2.9(b).



H: 500 ns/div

V: 100 mV/div



H: 20 kHz/div

V: 10 dB/div

Fig.2.11 Measured baseband eye diagram and corresponding power spectral density of a 15-level PRS signal using our new Model-B. This eye diagram corresponds to that of the I (or Q) channel of a 225-QPRS modem operated at f_b = 1.6 Mb/s. The spectral efficiency of the baseband system and of the quadrature modem is 800 kb/s ÷ 128 kHz = 6.25 b/s/Hz, i.e., a 4.2% increase above the Nyquist rate of 6 b/s/Hz. This efficiency is defined at the 50 dB attenuation point. In this measurement, a Karkar Electronics, Inc. patented α = .01 filter was used (Ref. U.S. Patent 3271705).

2.5 PROBABILITY OF ERROR OF 225-QPRS ABOVE THE NYQUIST RATE

Essentially 225-QPRS above the Nyquist rate can be generated by modulating in quadrature two baseband 15-level PRS systems using Model-A or Model-B. A FORTRAN program was written to evaluate the performance. The simulation method is very similar to that used in evaluating the performance of high-level QAM systems [8], [36].

Two baseband 15-level PRS signals are represented in a complex form. We assume the duobinary PRS filter shaping is fully at the transmitter whereas the receive filter merely bandlimits the channel noise which is assumed to be additive, white and Gaussian. In Appendix A, we show that this results in about a 1 dB degradation in C/N with respect to the case in which the filter shaping is optimally divided between the transmitter and the receiver. The well-known method for precoding [19]-[21] has been used to prevent error propagation. A perfect synchronization is assumed and symbol-by-symbol decision is used in the simulation.

Results are plotted in Fig.2.11. The ideal curve in Fig.2.11 represents the P_e performance corresponding to the optimum distribution of filters and is plotted according to the following equation.

$$P_e = \frac{63}{64}\ \mathrm{erfc}\left(\sqrt{\frac{1}{42}\ \left(\frac{\pi}{4}\right)^2 \frac{C}{N}}\ \right) \qquad (2.9)$$

Eq. (2.9) is obtained by substituting M = 8 into (A.22) and replacing S/N by C/N, since 225-QPRS is equivalent to a 15-level PRS signal in

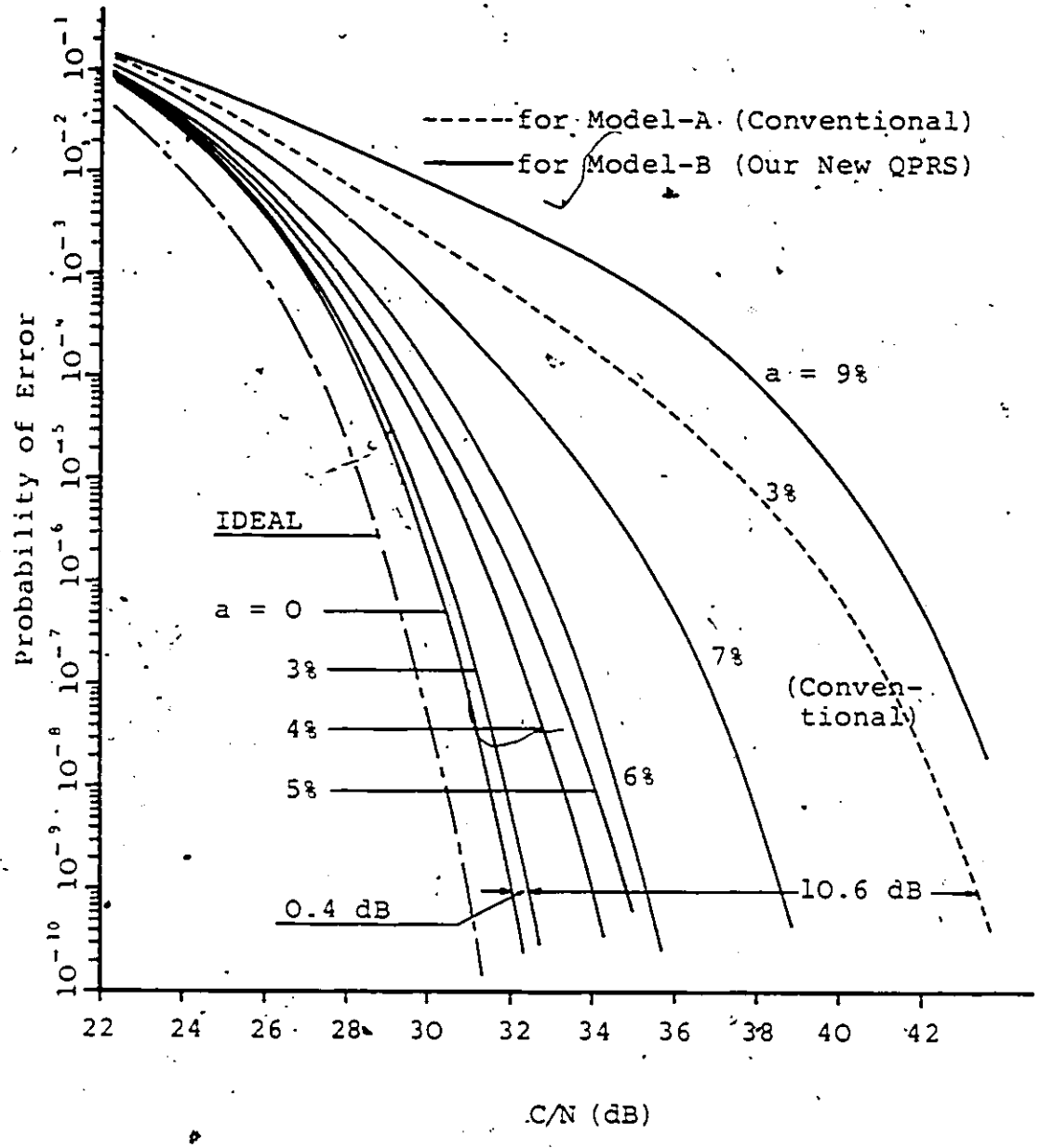


Fig.2.12 Probability of error performance of 225-QPRS above the Nyquist rate versus C/N, using Model-A or Model-B.

TABLE 2.1 Required C/N (dB) at $P_e = 10^{-9}$ for 225-QPRS above the Nyquist rate.

% above Nyquist rate	Spectral efficiency (b/s/Hz)	Model-A (conventional) C/N (dB)	Model-B (new scheme) C/N (dB)
0	6	32	32
3	6.18	43.3	32.4
4	6.24	Not possible	34.
5	6.3	Not possible	34.9
6	6.36	Not possible	35.3

baseband. Table 2.2 summarizes the required C/N at $P_e = 10^{-9}$ for 225-QPRS for various percentage increases above the Nyquist rate using Model-A or Model-B. The significant advantage of our Model-B is evident from Fig.2.11 and Table 2.2. For example, for 225-QPRS transmission, 3% above the Nyquist rate with conventional Model-A 11 dB degradation, while with our new Model-B only a 0.4 dB degradation is observed.

2.6 CONCLUSIONS

We conclude that, by simple symbol-by-symbol receiver structures without MLSD or other advanced nonlinear equalizers, the conventional Model-A is inappropriate for multi-level PRS/QPRS systems, such as 225-QPRS, operated above the Nyquist rate. However, with our new Model-B for 225-QPRS, an increase of up to 5% above the Nyquist rate can be achieved by simple symbol-by-symbol detection with a small degradation. This increase has very significant applications, such as the transmission of a T1 carrier in one standard analog supergroup band, as previously mentioned in Chapter 1.

Chapter III

IMPROVED EFFICIENCY 15-LEVEL MODIFIED DUOBINARY PRS ABOVE THE NYQUIST RATE

An improved efficiency technique for duobinary (class-I) PRS above the Nyquist rate with multi-level inputs has been described in Chapter 2 and [22],[33]. However, duobinary PRS is not suitable for use in some systems, such as transformer coupled circuits, SSB modems, and carrier systems with carrier pilot tones, where reduced low-frequency components in the spectrum are required. On the other hand, modified duobinary (class-IV) PRS is ideally suited for the above-mentioned applications [34], since it has no energy at dc and a small amount at low frequencies. Thus, the objective of this chapter is to study the possibility of increasing the spectral efficiency above the Nyquist rate using class-IV PRS with multi-level inputs.

3.1 SPEED TOLERANCE OF CLASS-IV PRS WITH MULTI-LEVEL INPUTS

The conventional method for class-IV PRS above the Nyquist rate transmission is illustrated in Fig.3.1 and will be called Model-A. Referring to Fig.3.1, the original system is designed to transmit 1/T symbols/s over a bandwidth of 1/2T Hz, resulting in a spectral efficiency of 2 symbols/s/Hz, i.e., the so-called Nyquist rate. Transmission above the Nyquist rate is conventionally achieved by simply increasing the

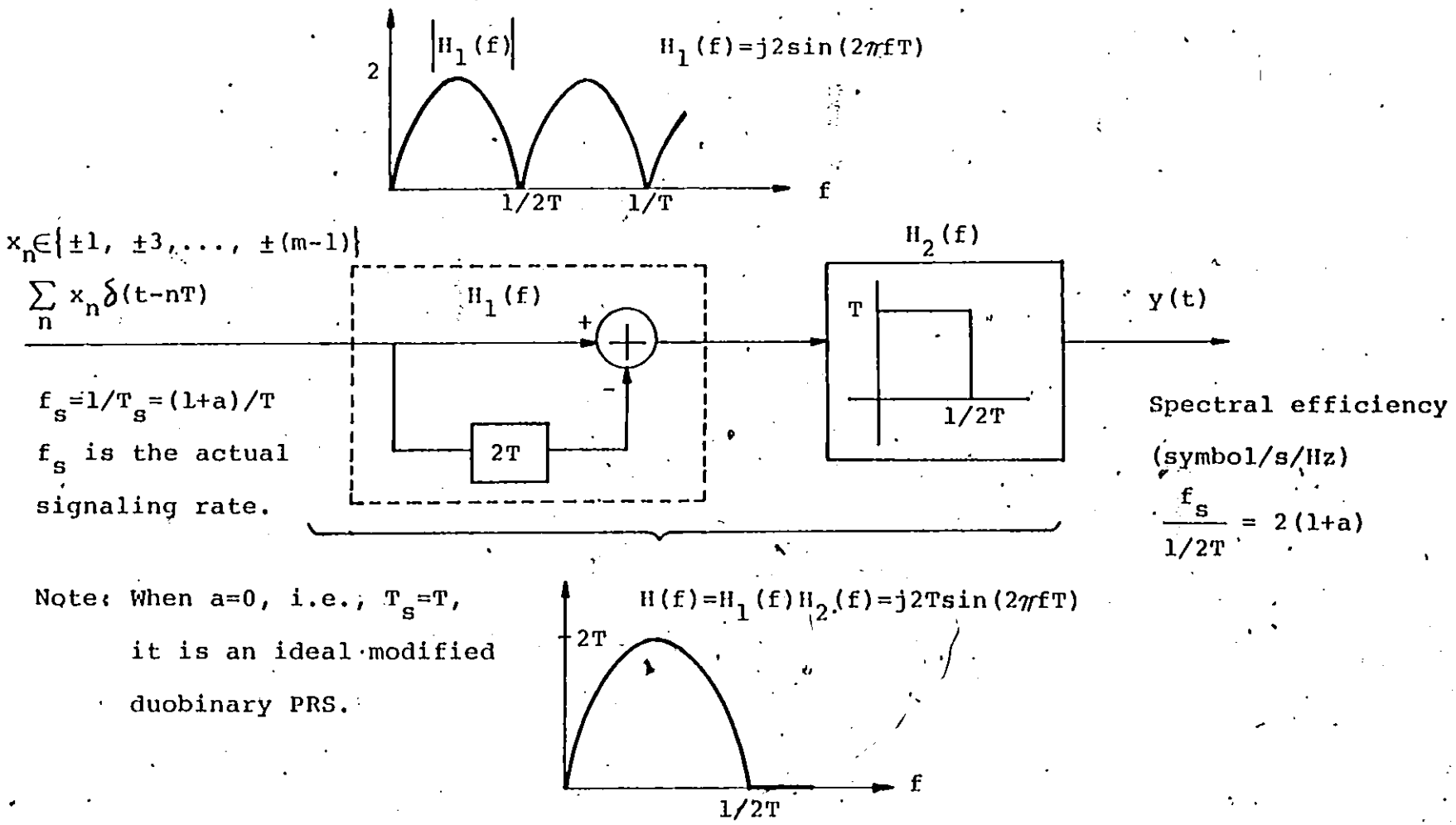


Fig.3.1 Conventional system Model: Model-A for class-IV (modified duobinary) PRS above the Nyquist rate and the cascaded transfer function. The spectral efficienty of this system is 2(1+a) symbol/s/Hz, i.e., 100a% higher than the so-called Nyquist rate of 2 symbol/s/Hz.

signalling rate to $(1+a)/T$ symbols/s while keeping the transmission characteristic fixed, as is the case in which speed tolerance was evaluated [18]-[20]. Thus, the spectral efficiency of Model-A is $(1+a)/T : 1/2T=2(1+a)$ symbol/s/Hz, where the parameter, "a" represents the percentage increase above the Nyquist rate.

The speed tolerance for binary inputs is defined as the percentage increase over the Nyquist rate which will just cause overlap between adjacent levels ("eye closure") [19]. In the case of m-ary inputs, the speed tolerance can be defined as the percentage increase over the Nyquist rate which will just cause at least one of the multi-level eyes to be closed. Based on this definition, we evaluated the speed tolerance of class-IV PRS with m-ary inputs by computer simulation using "worst possible" data sequences [18],[19]. The results are summarized in Table 3.1 where the well-known 16% speed tolerance of class-IV PRS for binary inputs is confirmed and the results for 4- and 8- level inputs are new.

We notice that the speed tolerance of class-IV PRS has dramatically reduced to about 1% for 8-level inputs. This is mainly due to inherently higher ISI in multi-level systems. Hence, by simple symbol-by-symbol receiver structures, it is nearly impossible for using class-IV PRS to increase the spectral efficiency above the Nyquist rate by the conventional method.

TABLE 3.1 SPEED TOLERANCE OF CLASS-IV (MODIFIED DUOBINARY) PRS FOR MULTI-LEVEL INPUTS.

Number of input levels	Speed tolerance
2	16%
4	3%
8	1%

Note: The speed tolerance of 16% for binary inputs is confirmed to agree with that in the published literature [19], [20].
Speed tolerances for 4- and 8-level inputs are our new results.

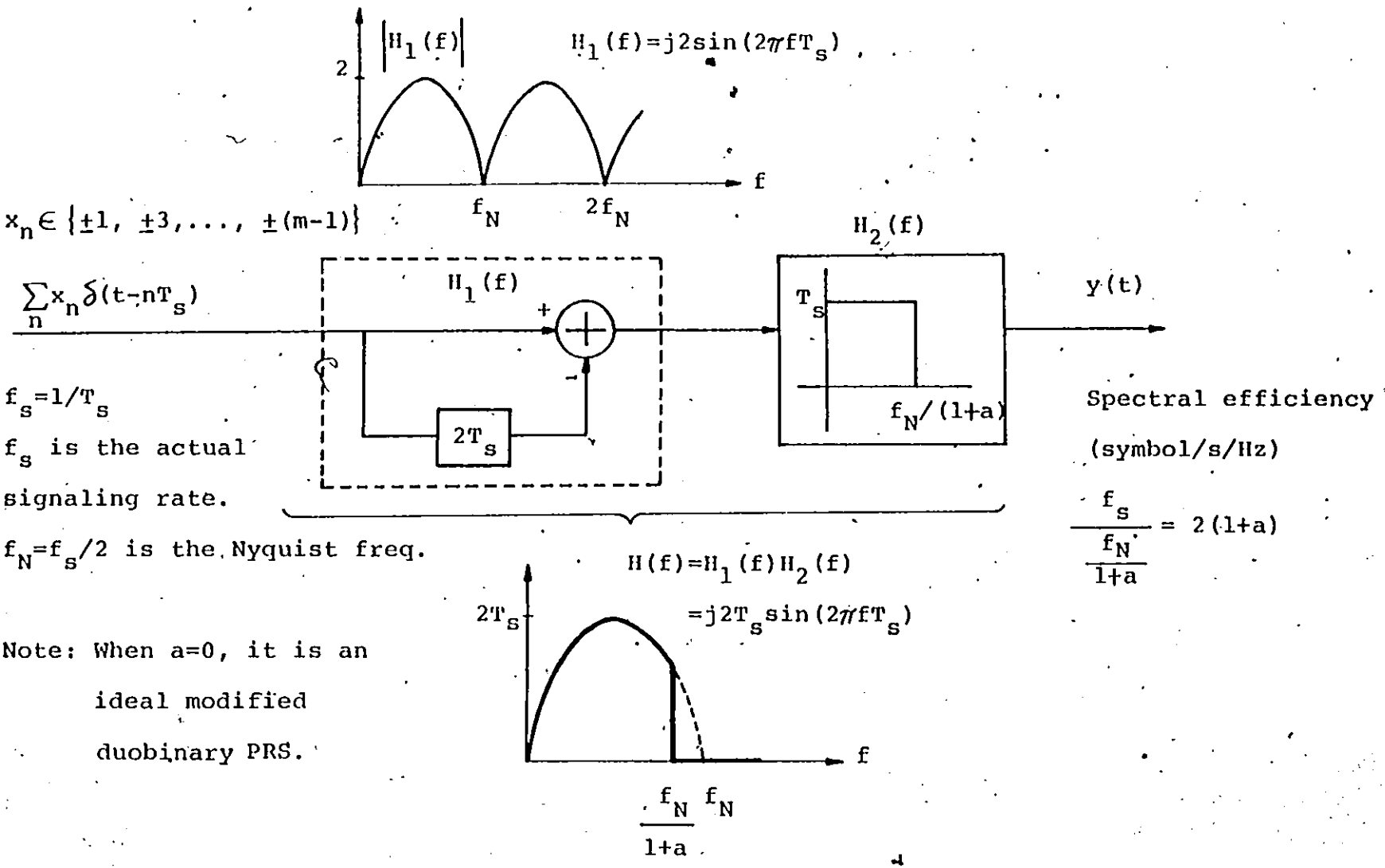
3.2 A NEW TECHNIQUE FOR CLASS-IV PRS ABOVE THE NYQUIST RATE

An improved efficiency technique for multi-level duobinary (class-I) PRS above the Nyquist rate has been proposed in Chapter 2. The merit of this technique lies in the following fact. For the duobinary system, there is only a small amount of energy at frequencies near the Nyquist frequency (which is defined as half the signaling symbol rate) so that part of it can be chopped off without causing too much performance degradation. Since the class-IV PRS system has a spectral characteristic near the Nyquist frequency very similar to that of duobinary, we expect the chopping technique can be applied.

Fig.3.2 shows a new system model [35], called Model-B, using the chopping technique for class-IV PRS above the Nyquist rate. It represents a scheme in which the original class-IV PRS filter is followed by an ideal brickwall filter with a cutoff frequency at $f_N/(1+a)$, where $f_N=1/2T_s$ is the Nyquist frequency of the signal. The spectral efficiency of this new model is $1/T_s$: $f_N/(1+a)=2/(1+a)$ symbols/s/Hz, i.e., the same as that of the conventional Model-A. We shall concentrate on a 15-level system, i.e., with 8-level inputs since it has a potential to achieve a spectral efficiency of more than 6 b/s/Hz.

From Fig.3.2, the overall frequency response of the new Model-B for class-IV PRS above the Nyquist rate can be expressed as follows.

$$H(f) = \begin{cases} j2T_s \sin(2\pi f T_s) & |f| < f_N/(1+a), \\ 0 & \text{elsewhere.} \end{cases} \qquad (3.1)$$



Note: When a=0, it is an ideal modified duobinary PRS.

Fig.3.2 New system model: Model-B for class-IV (modified duobinary) PRS above the Nyquist rate with m-ary inputs and the corresponding transfer function. This also corresponds to the system model of the equivalent baseband in-phase or quadrature channel of a quadrature-modulated system.

And the corresponding impulse response can be easily obtained by a straightforward manipulation, i.e.,

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

$$= -\int_{-f_N/(1+a)}^{f_N/(1+a)} 2T_s \sin(2\pi fT_s) \sin(2\pi ft)\, df \qquad (3.2)$$

$$= \frac{1}{1+a}\left\{\text{sinc}\left(\left(\frac{t}{T_s}+1\right)/(1+a)\right) - \text{sinc}\left(\left(\frac{t}{T_s}+1\right)/(1+a)\right)\right\}$$

Before proceeding to deal with the probability of error, we shall present a few typical eye diagrams, plotted by computer simulations. An ideal eye diagram of a 15-level class-IV PRS signal at the Nyquist rate is shown in Fig.3.3 for reference. The eye diagram for the new Model-B with 8-level inputs and a=4%, i.e., with a spectral efficiency of 6.24 b/s/Hz is demonstrated in Fig.3.4. Though ISI exists at the optimum sampling instants, the quality of this eye pattern is excellent. For comparison, the eye diagram presented in Fig.3.5, with the same efficiency of 6.24 b/s/Hz using the conventional Model-A, is completely closed.

It is also important, at this moment, to compare the eye diagram shown in Fig.3.4 for class-IV PRS with that in Fig.2.9 (b) for class-I PRS. We observe that the vertical eye openings in both cases are quite comparable. This implies that, when signalling above the Nyquist rate using our new chopping technique, class-I and class-IV PRS will have similar performance in terms of the probability of error.

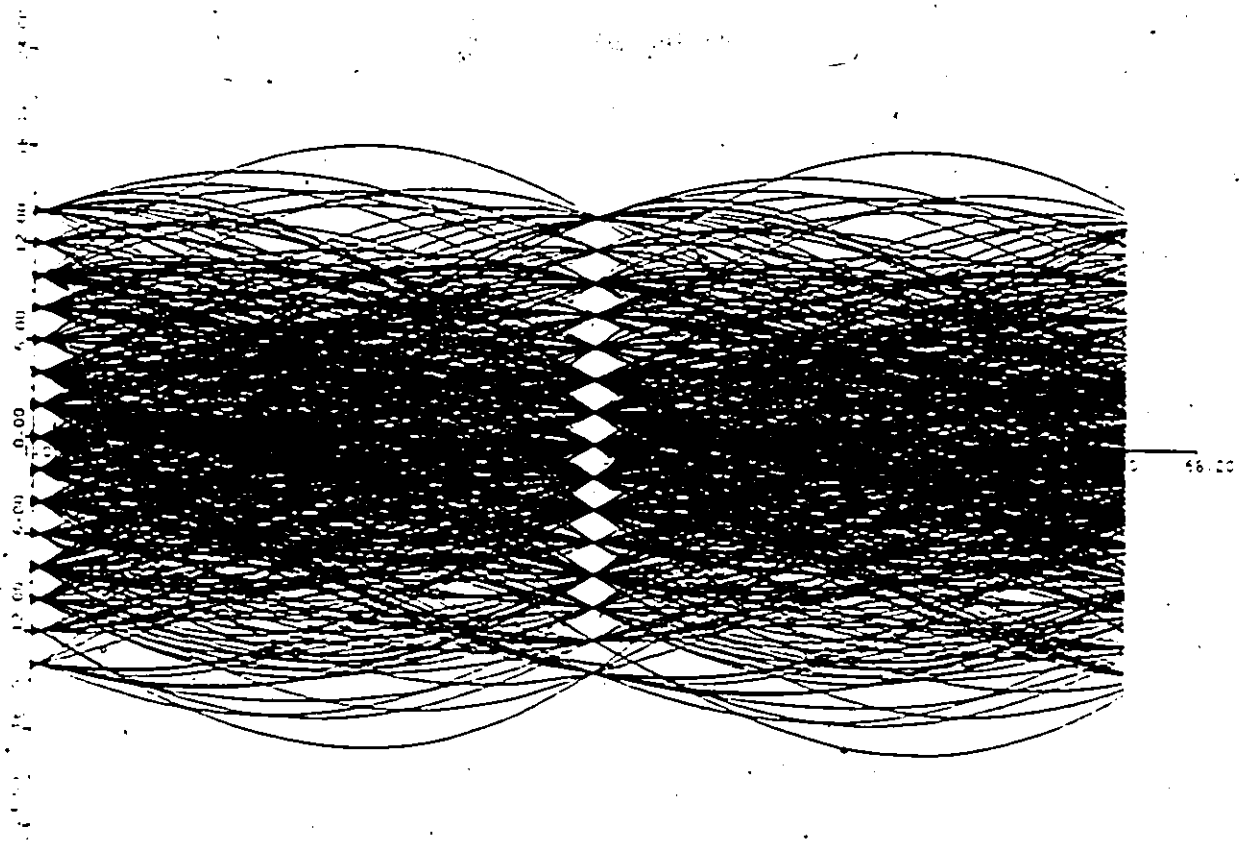


Fig.3.3 Eye diagram of an ideal 15-level class-IV (modified duobinary) PRS signal. Spectral efficiency : 6 b/s/Hz.

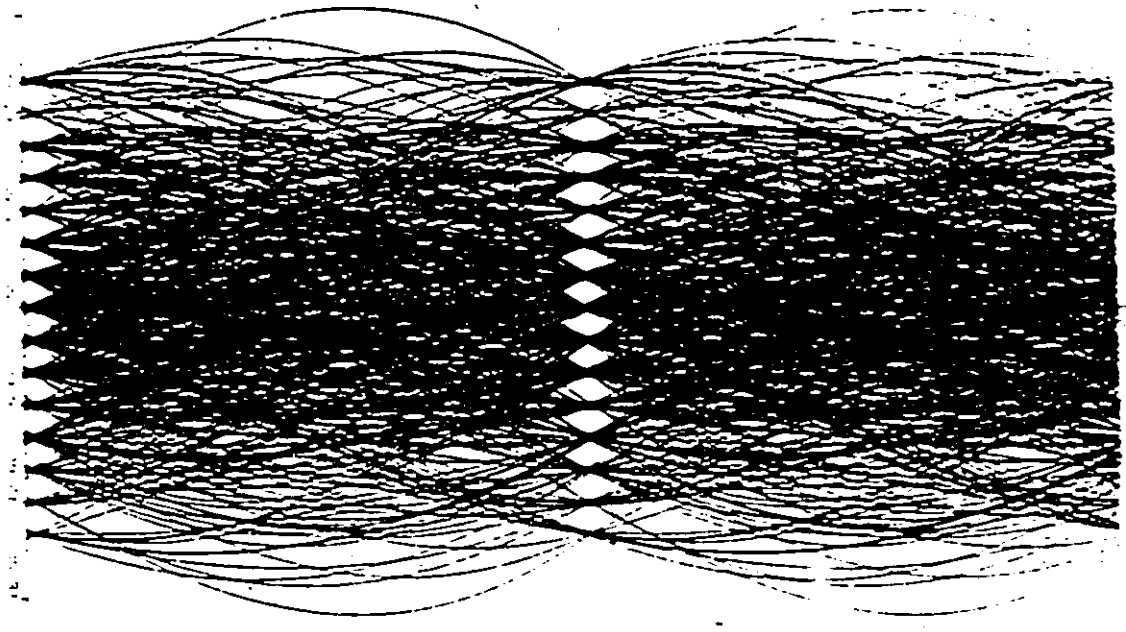


Fig.3.4 Eye diagram of an improved efficiency class-IV 15-level PRS/225-QPRS with a 4% increase above the Nyquist rate. Spectral efficiency : 6.24 b/s/Hz.

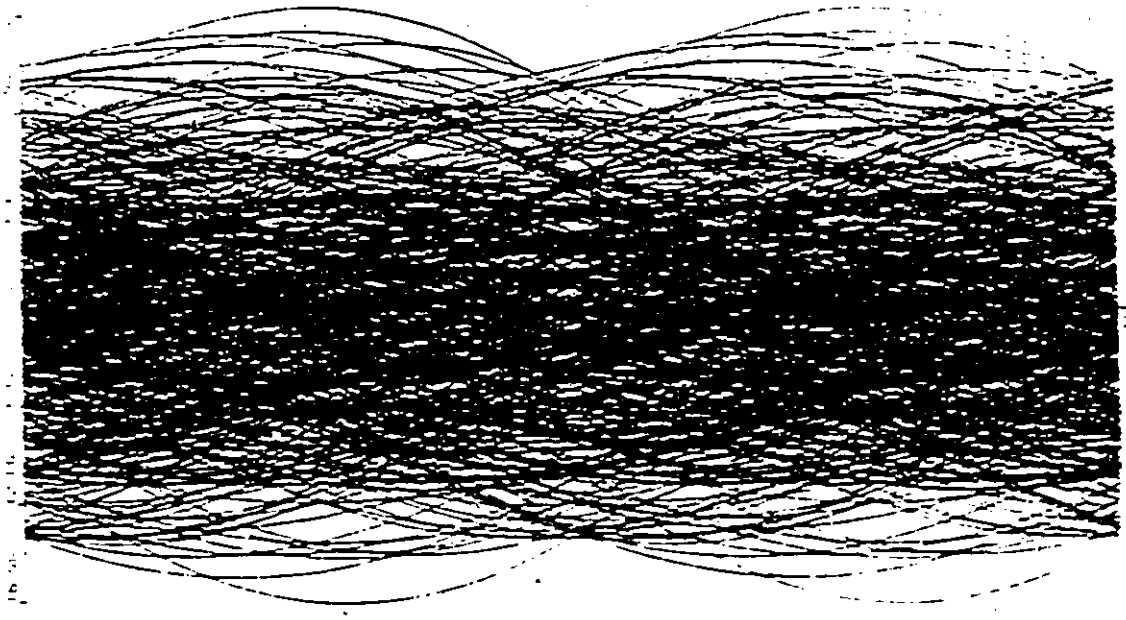


Fig.3.5 Eye diagram of a conventional class-IV 15-level PRS/225-QPRS with a 4% increase above the Nyquist rate. Spectral efficiency : 6.24 b/s/Hz.

3.3 PERFORMANCE OF IMPROVED EFFICIENCY CLASS-IV 15-LEVEL PRS/225-QPRS ABOVE THE NYQUIST RATE

In this section, we evaluate the performance of an improved efficiency class-IV 15-level baseband and of modulated 225-QPRS using the new technique described in Section 3.2. Essentially, a class-IV 225-QPRS above the Nyquist rate can be generated by modulating in quadrature two baseband 15-level systems using Model-B in Fig.3.2. A FORTRAN program was written to evaluate the probability of error. As an illustrative example, a bit rate of 1.6 Mb/s is used in the simulation, i.e., with 800 kb/s in both in-phase and quadrature channels. The simulation is performed in a way very similar to that described in Chapters 2 and 5. Again, a perfect synchronization is assumed and symbol-by-symbol detection is used. Results are plotted in Fig.3.6. Table 3.2 summarizes the required C/N at $P_e=10^{-9}$ for various percentage increases above the Nyquist rate using the improved efficiency class-IV PRS.

The significant advantage of the new technique is evident from Table 3.2. For example, for class-IV 225-QPRS with a 3% increase above the Nyquist rate using our new Model-B, only a 0.6 dB degradation is observed. Also listed in Table 3.2 is the corresponding result of improved class-I 225-QRRS in Chapter 2 for comparison.

We notice that, with our new Model-B for 225-QPRS above the Nyquist rate for an increase of up to 5%, the performance of using class-IV PRS is nearly as good as that of using class-I PRS. For example, for the same 4% increase using class-IV PRS, there is only a 0.2 dB loss in

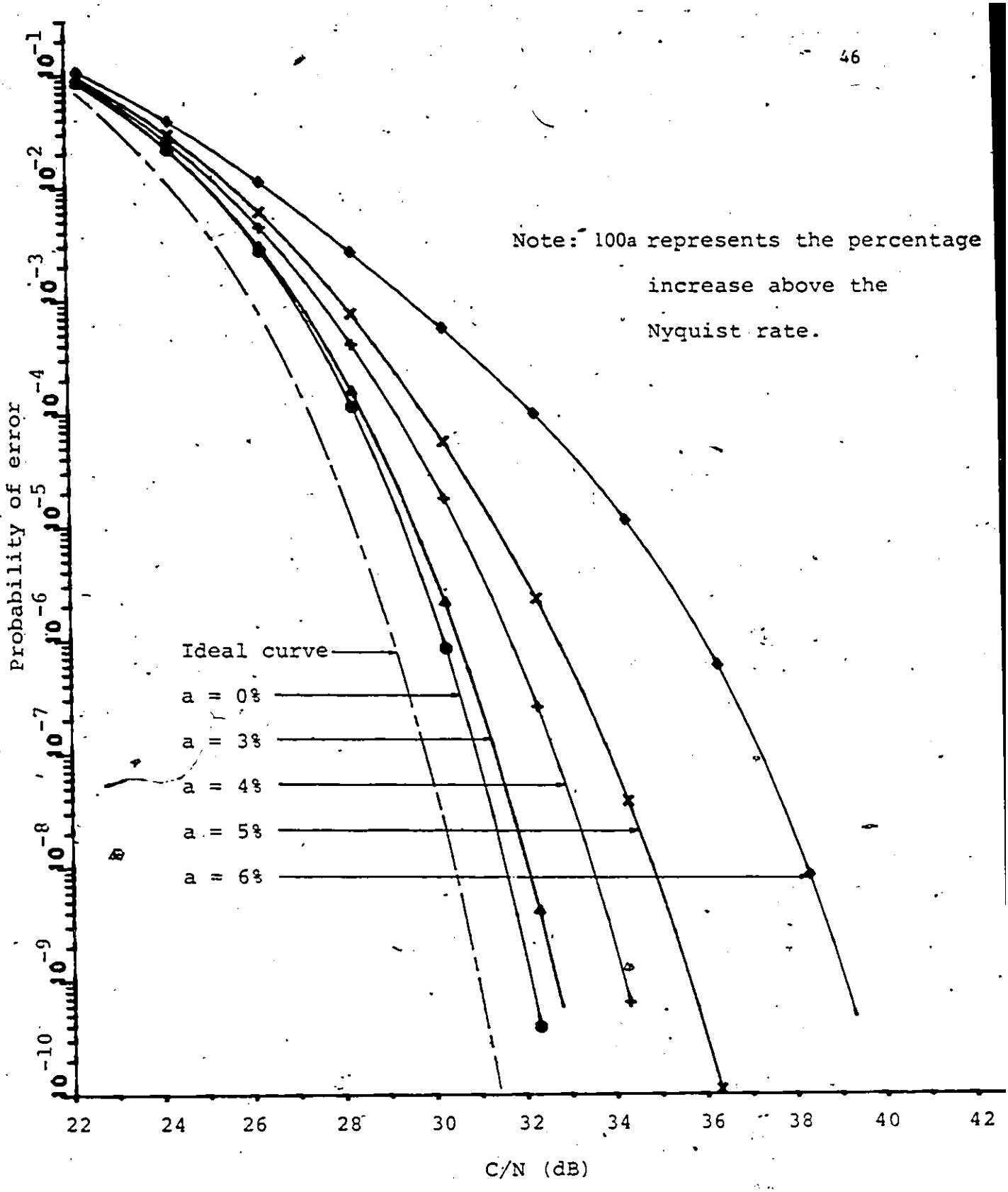


Fig.3.6 Probability of error of our improved efficiency class-IV 15-level PRS/225-QPRS above the Nyquist rate versus carrier-to-noise ratio (C/N), using our Model-B.

TABLE 3.2 REQUIRED C/N AT $P_e = 10^{-9}$ for 225-QPRS ABOVE THE NYQUIST RATE USING OUR NEW METHOD.

% above the Nyquist rate	Spectral Efficiency (b/s/Hz)	Class-IV 225-QPRS C/N (dB)	Class-I * 225-QPRS C/N (dB)
0	6	32	32
3	6.18	32.6	32.4
4	6.24	34.2	34
5	6.3	35.6	34.9
6	6.36	39.1	35.3

* From Table 2.1.

C/N at $P_e=10^{-9}$ with respect to that of using class-I PRS. This is surprising since it might be tempting to say that, for transmission above the Nyquist rate, duobinary is better than modified duobinary for the following fact. The speed tolerance of duobinary is 43% for binary inputs whereas it is only 16% for modified duobinary. It is true when the conventional method is used. However, it is not true, especially for m-ary inputs, by using the chopping technique described in section 3.2. In addition, modified duobinary has some other advantages. For example, it has no energy at dc, thus a carrier pilot can be inserted for easy carrier synchronization. Also, it has a very small amount of energy at low frequencies near dc so that a service channel can be added; this will be described in the next section.

3.4 AN EFFICIENT METHOD FOR SERVICE CHANNEL TRANSMISSION

An unusual and interesting aspect of class-IV PRS is that there is no energy at dc and a relatively small amount at low frequencies. Thus, in addition to increasing the spectral efficiency by chopping at the high-frequency end, a small bandwidth at the low-frequency end of the modified duobinary spectrum can be carved out for accommodating a service channel [21]. In this case, the advantage is having a service channel completely independent of the main traffic bit stream. That is, should the data stream be lost, the service (order) channel may still exist. It should be mentioned that a service channel is very important in control, supervision and servicing of a system.

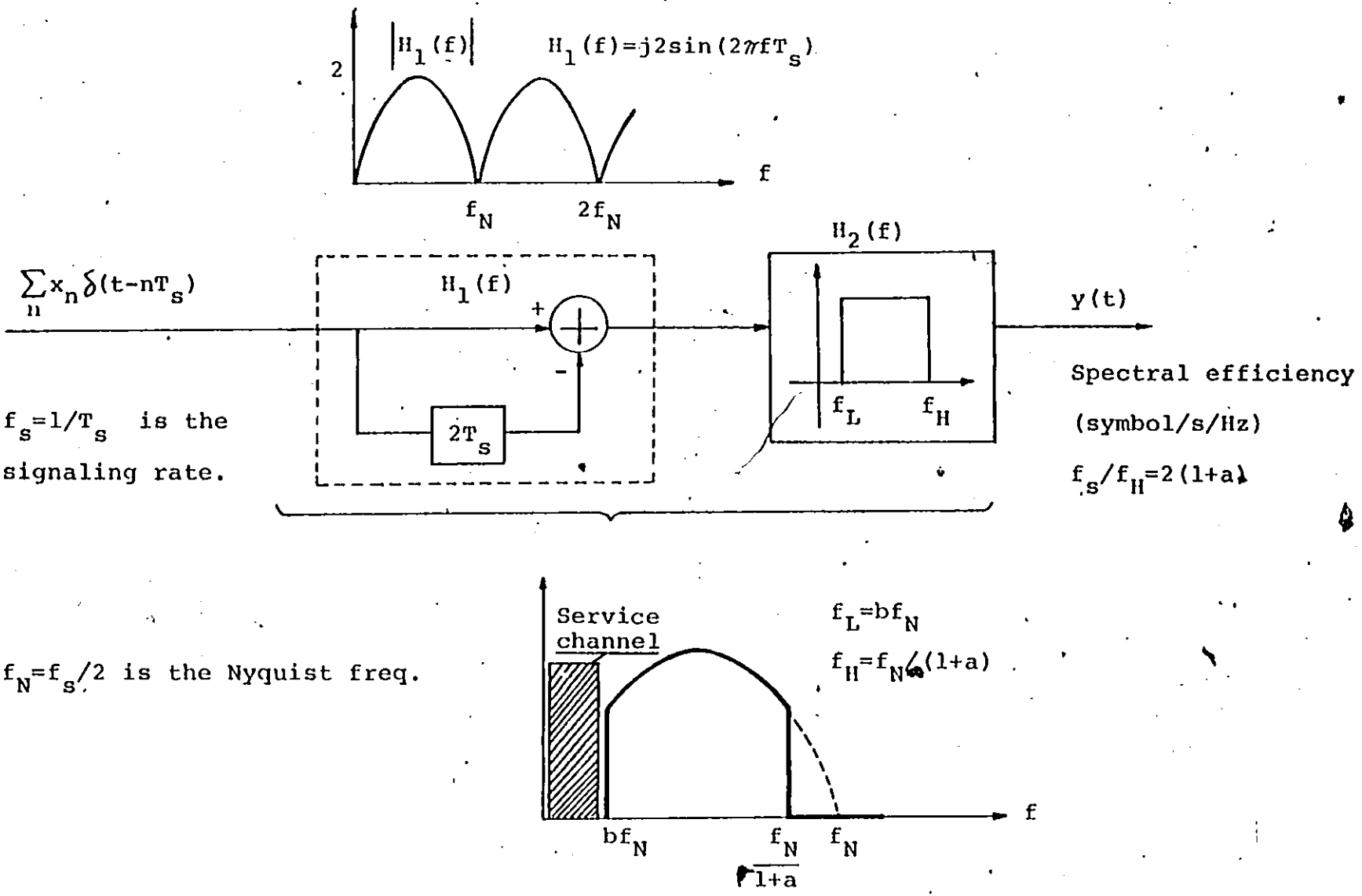


Fig.3.7 System model for an improved efficiency class-IV PRS above the Nyquist rate and for service channel transmission.

This particular application is illustrated in Fig.3.7, in which the modified duobinary filter is followed by an ideal bandpass filter with $f_L(=bf_N)$ and $f_H(=f_N/(1+a))$ representing the cutoff frequencies at the low end and at the high-frequency end respectively. A service channel can then be inserted between dc and f_L.

We studied the effect of the location of f_L on the performance of a 15-level PRS/225-QPRS with $f_H=f_N/(1+a)$, a=0.04, i.e., with a 4% increase above the Nyquist rate. The simulation result is shown in Fig.3.8. It is seen that, as long as b<0.03, the degradation is in fact negligible. This indicates that, under the condition of a 4% increase above the Nyquist rate for the improved efficiency class-IV 225-QPRS, a bandwidth of 3% f_N at the low-frequency end can be used for service channel transmission, which is not achievable by duobinary PRS.

It should be particularly noted that at the low-frequency end of the class-IV PRS spectrum not only a service channel can be used by using our chopping technique but also an increase in spectral efficiency is obtained in an FDM SSB system. For example, from the results shown in Fig.3.8, a 7% (i.e., with 3% of f_N cut at the low-frequency end and 4% at the high-frequency end) increase in spectral efficiency with respect to the Nyquist rate can be obtained by SSB modulation with a small degradation.

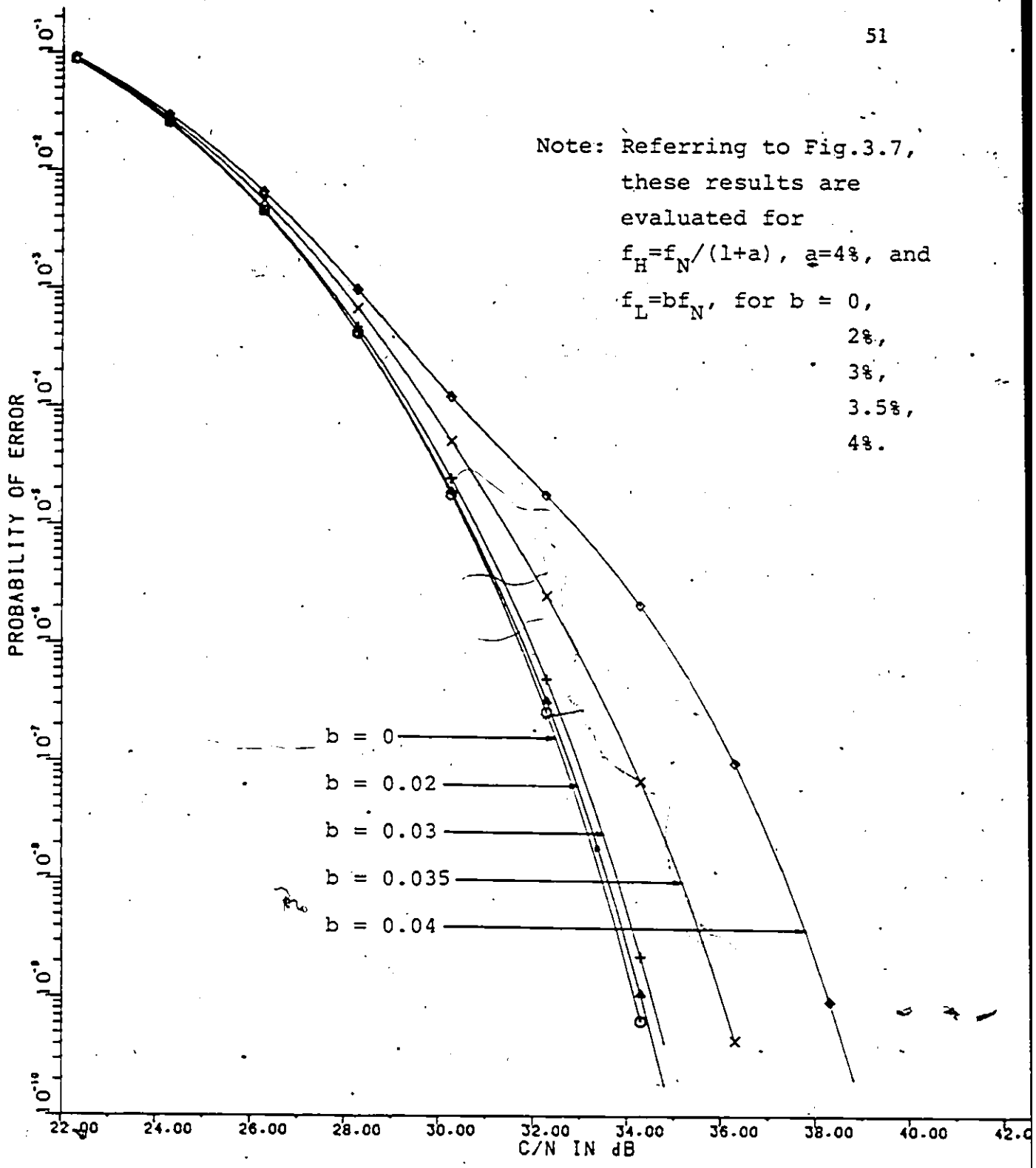


Fig.3.8 Performance of an improved efficiency class-IV 15-level PRS/225-QPRS with a 4% increase above the Nyquist rate and with various bandwidths carved out for a service channel.

3.5 CONCLUSIONS

An improved efficiency technique for modified duobinary (class-IV) PRS above the Nyquist rate transmission has been described. By using this technique, an increase of 4% or more above the Nyquist rate with class-IV 225-QPRS, i.e., a spectral efficiency of 6.24 b/s/Hz or more can be achieved by simple symbol-by-symbol detection and the performance is nearly comparable to that of duobinary (class-I) PRS. Furthermore, it is shown that with a 4% increase above the Nyquist rate for the improved class-IV 225-QPRS, an additional 3% of the Nyquist bandwidth can be carved out for service channel transmission with a negligible degradation. In particular, a 7% increase in spectral efficiency can be obtained by SSB-modulated class-IV 15-level PRS using our technique.

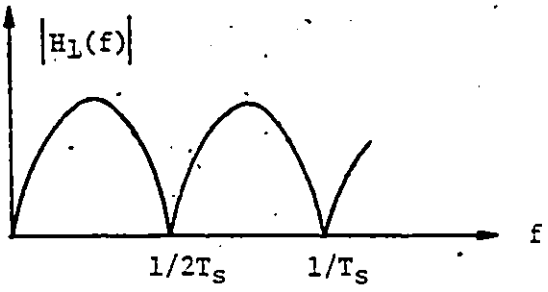
Chapter IV

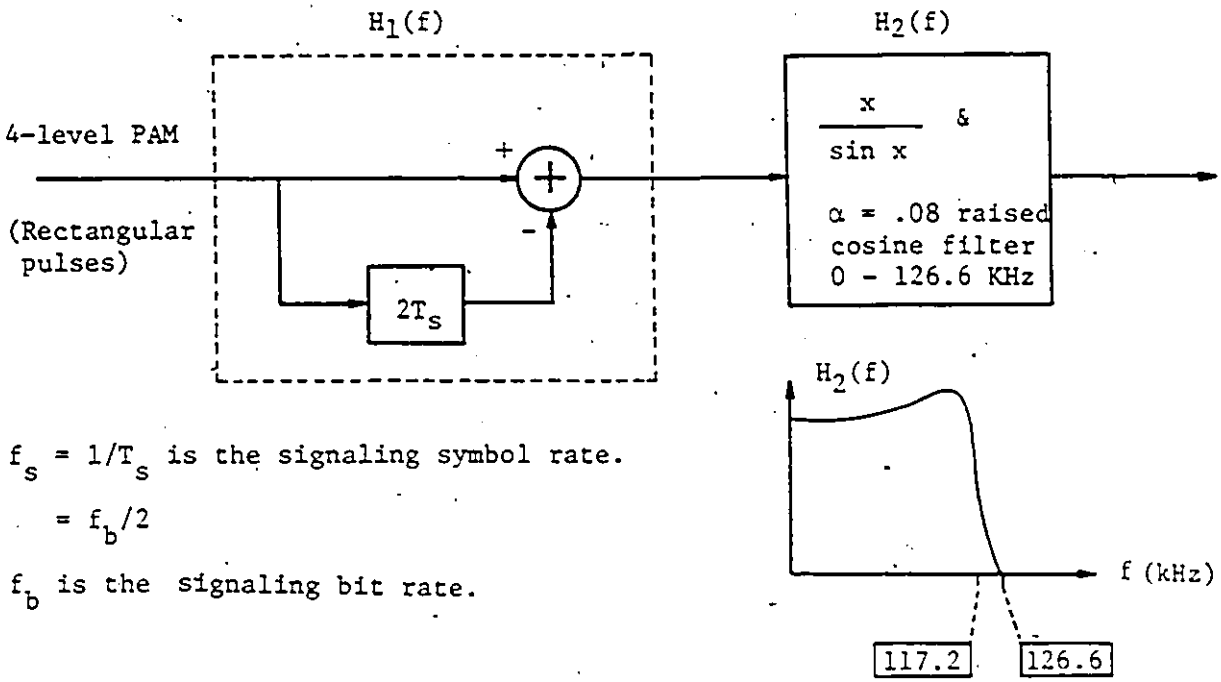
IMPLEMENTATION AND PERFORMANCE EVALUATION OF AN IMPROVED EFFICIENCY 7-LEVEL CLASS-IV PRS SYSTEM

In order to perform practical research and investigate practical hardware feasibility of the new concept previously presented for PRS/QPRS above the Nyquist rate, we designed, by using our new Model-B, a 7-level class-IV PRS system which is the equivalent base-band of a 49-QPRS modem. Detailed description of the hardware design is presented. Several measured eye diagrams and corresponding power spectral densities are compared for various percentage increases of signaling rate above the Nyquist rate. Measured bit error rate (BER) performance versus signal-to-noise ratio (S/N) shows a good agreement with that predicted by computer simulations.

4.1 SYSTEM MODEL

Referring to Fig.4.1, which is redrawn here from Fig.3.2 for convenience, it is easy to see that the function of increasing the signalling rate while keeping the bandlimiting filter $H_2(f)$ unchanged is equivalent to that of keeping the signalling rate fixed while changing the cutoff frequency of $H_2(f)$. We adopted the former approach because of hardware simplicity. It should be noted that the ideal brickwall filter $H_2(f)$ in Fig.3.2 is practically replaced by a steep filter which approximates





$f_s = 1/T_s$ is the signaling symbol rate.

$= f_b/2$

f_b is the signaling bit rate.

Illustrative bit rates relative to the nominal one:

f_b = 468.8 kHz --- the norminal bit rate for which the filter $H_2(f)$ is designed.

= 468.8 x 1.05 = 492.2 kHz --- a 5% increase relative to the nominal bit rate.

= 468.8 x 1.1 = 515.7 kHz --- a 10% incerase relative to the nominal bit rate.

Fig.4.1 System model for the experimental 7-level class-IV PRS system operated above the Nyquist rate using our Model-B.

an $\alpha = 0.08$ raised-cosine filter in Fig.4.1. The choice of this practical filter will be explained in section 4.2.3, where the measured filter characteristics will be presented as well.

4.2 DESCRIPTION OF HARDWARE IMPLEMENTATION

A system block diagram of our experimental 7-level class-IV PRS above the Nyquist rate using our new Model-B is depicted in Fig.4.2. We elaborate the system operation in the following paragraphs.

4.2.1 Encoder Operation

Referring to Fig.4.2, the input to the system with a bit rate f_b is a usual serial binary data stream consisting of 1 and 0. A serial-to-parallel (S/P) converter is used to group every two serial bits into a four-state symbol (i.e., one of the numbers 0, 1, 2, 3) represented by two parallel bits at a symbol rate $f_s = f_b/2$. In order to ensure that an error in an adjacent level of the received signal results in only a bit error, the output of the S/P converter is Gray-coded. The precoding operation for preventing error propagation [21],[53] is then performed on the Gray-coded data stream, x_i, which consists of elements from the four numbers 0, 1, 2, 3. Let y_i denote the output data sequence of the precoder. The relationship between y_i and x_i is described by (4.1).

$$y_i = (x_i + y_{i-2}) \text{ Mod } 4 \qquad (4.1)$$

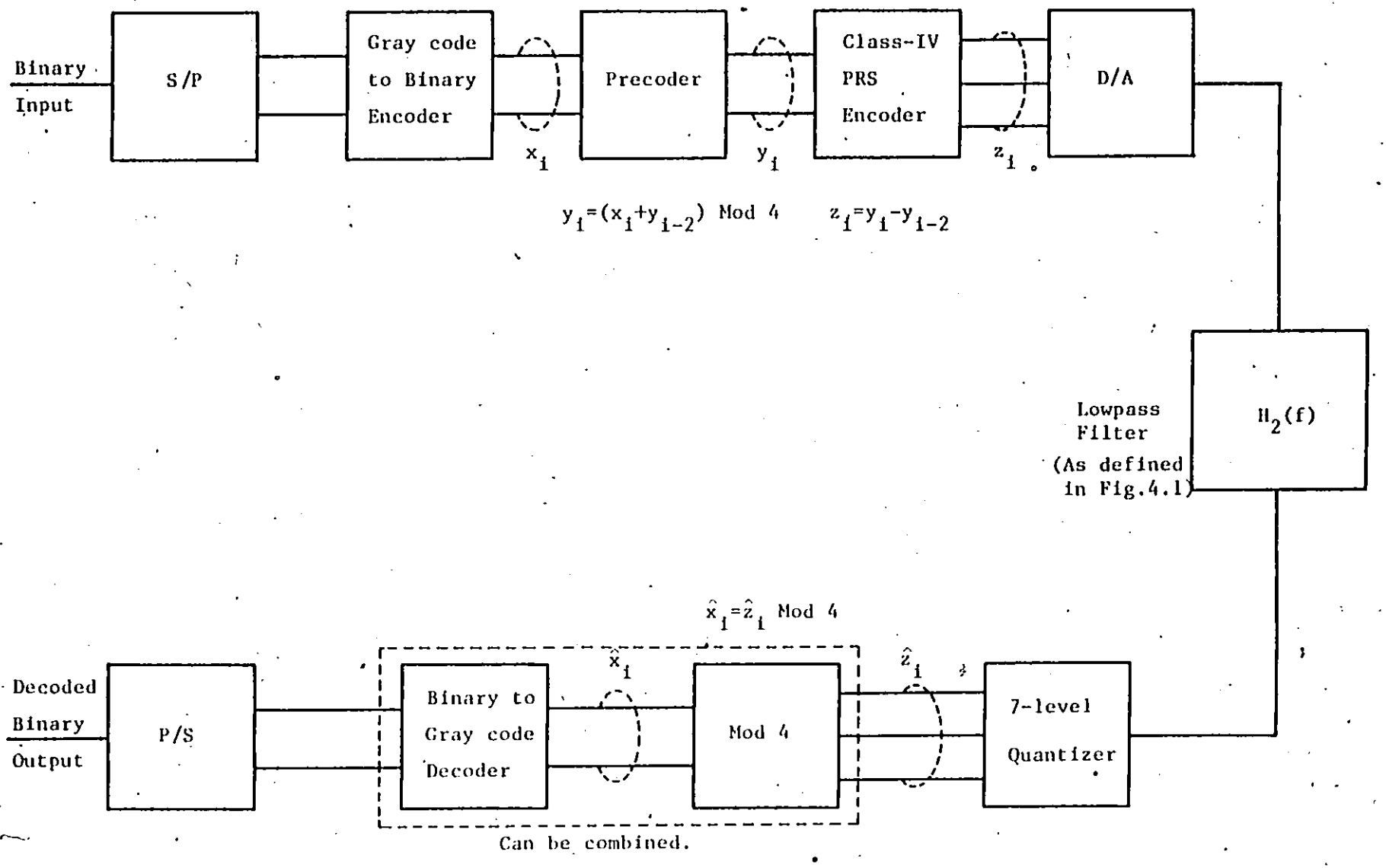


Fig.4.2 System block diagram of the experimental 7-level Class-IV PRS above the Nyquist rate using our new Model-B.

The next two blocks labelled class-IV PRS encoder and D/A converter convert the binary form (y_i) into a 7-level waveform in a digital manner. The function of the class-IV PRS encoder is to process the parallel binary signals as follows.

$$z_i = y_i - y_{i-2} \tag{4.2}$$

where y_{i-2} indicates a 2-symbol-interval-delayed version of y_i.

In effect, this is a combinational logic type digital filter that can conceptually be represented in the frequency domain as

$$\left|H_1(f)\right| = 2\left|\sin(2\pi f T_s)\right| \quad \text{for} \quad -\infty < f < \infty \tag{4.3}$$

To limit the bandwidth of the digital filter, a steep filter was employed as $H_2(f)$ in Fig.4.1. The filter characteristics will be described in Sec.4.2.3. In the next section, we describe in detail the operation of the encoder circuit.

4.2.2 Detailed Description of the Encoder Circuit

The following description refers to the circuit diagram of a 7-level class-IV PRS encoder depicted in Fig.4.3(a) and (b). The symbol-rate clock is derived through U5 by dividing the bit-rate clock by 2. The incoming binary data stream consisting of 1 and 0 which are represented by +5V and 0V, respectively, is S/P converted by U1 and the first two D flip-flops of U2 which are time aligned by the symbol-rate clock.

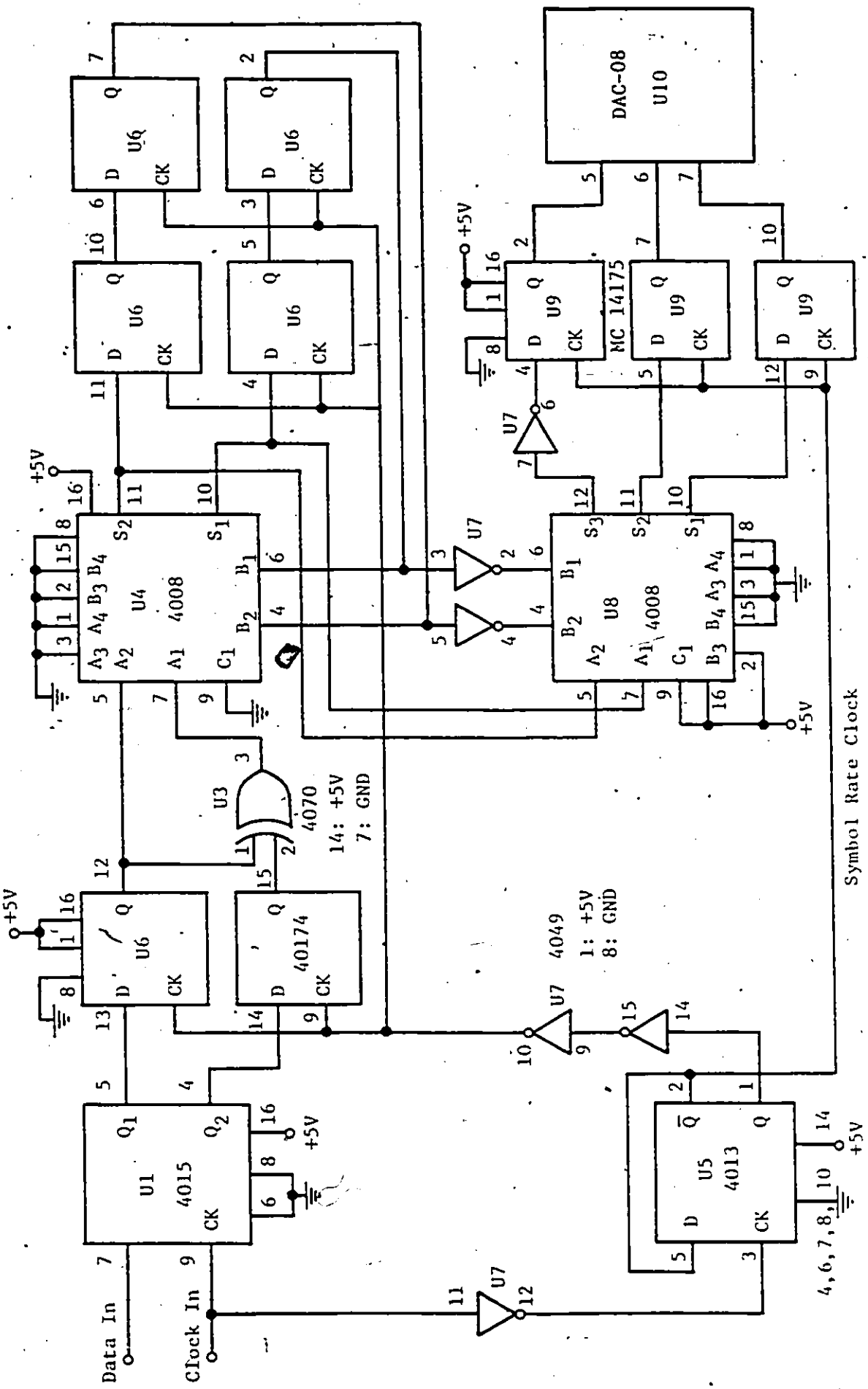


Fig.4.3(a) Circuit diagram of a 7-level class-IV PRS encoder.

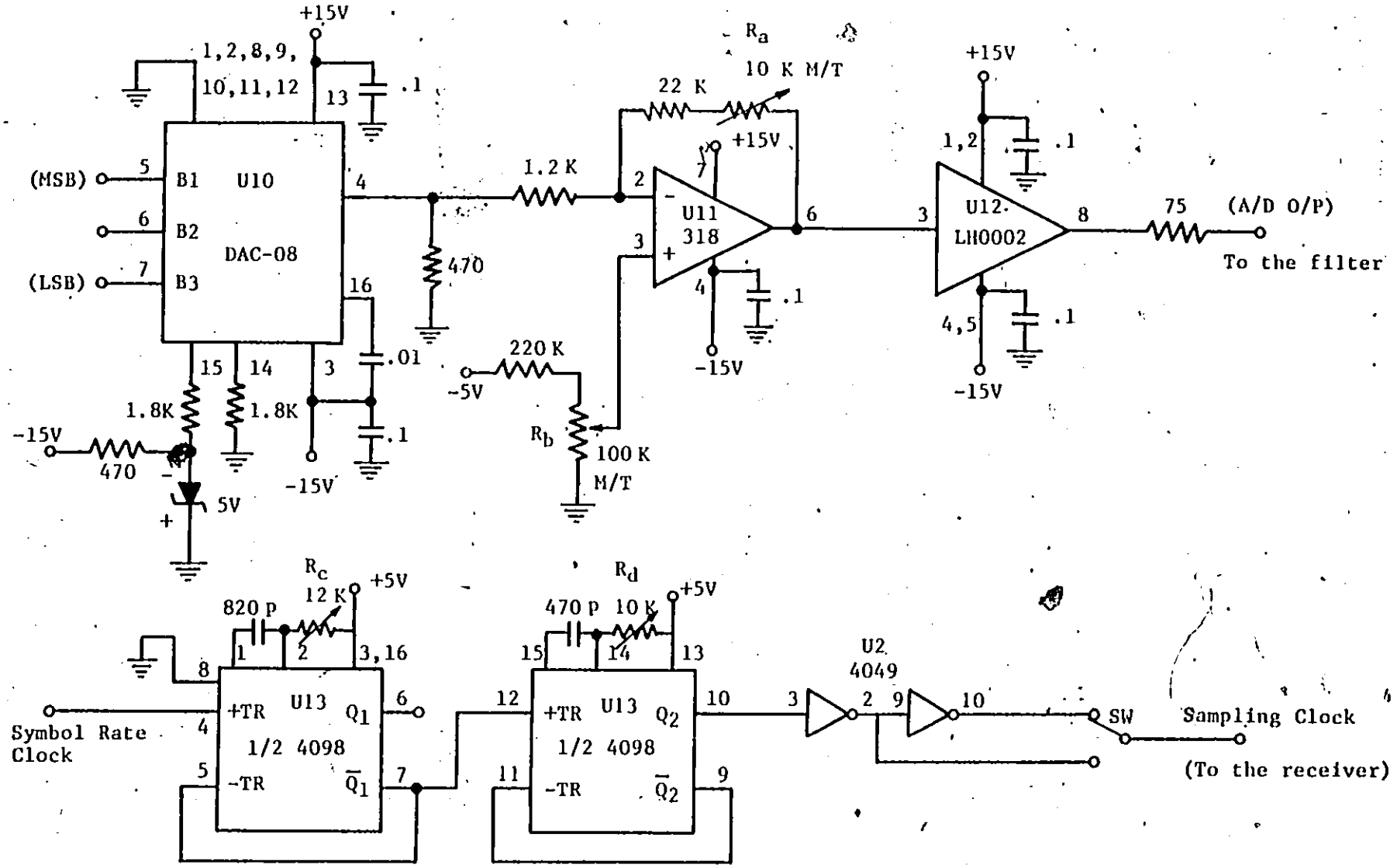


Fig.4.3(b) Output stage of the encoder and the sampling clock.
(Continuation of Fig.4.3(a))

U3 performs the gray-code operation on the aligned 2 parallel bits. Table 4.1 illustrates the gray-coding in detail.

After the gray coding, U4 and the remaining four D flip-flops of U2 carry out the precoding operation to prevent error propagation (see also Fig.4.2). In this case U4 is used as a 2-bit adder and because of the Mod 4 operation we simply discard the MSB of the result, i.e., the output is still 2 bits.

The class-IV PRS encoder is digitally fulfiled by U8, where the 2's complement subtraction is used. Due to different delays introduced in digital logic circuits, the output of U8 may have glitches. Therefore, three D flip-flops of U9 have been used to clock the output of U8 in order to ensure a glitch-free signal. Note that due to the specific correlation introduced by class-IV PRS, the 3 parallel bits at the output of U8, and of course U9, have only 7 different combinations. As a result, Table 4.2 summarizes the number mapping.

Due to the inherent structure, U10 maps a largest binary number (i.e., 111 in the present case) at its input into a most negative level. However, with the aid of the inverted amplifier (U11), finally a most positive level corresponds to the largest binary number, which is also illustrated in Table 4.2.

A dc balancing adjustment is provided by a 100 k multi-turn (M/T) potentiometer located at pin 3 of U11. A current amplifier (U12) is required to drive a 75 Ω load, which is the input impedance of the α = 0.08 raised-cosine filter with x/sin(x) equalization. Thus the wideband

TABLE 4.1 GRAY-CODE OPERATION WITH REFERENCE TO Fig.4.3(a).

U1		U4	
Q_1	Q_2	A_2	A_1
0	0	0	0
0	1	0	1
1	1	1	0
1	0	1	1

TABLE 4.2 THE NUMBER MAPPING WITH REFERENCE TO Fig.4.3(a) & (b).

	U8			U10			U11
	S_3	S_2	S_1	B_1	B_2	B_3	pin 6
3	0	1	1	1	1	1	-----
2	0	1	0	1	1	0	----- positive voltage
1	0	0	1	1	0	1	-----
0	0	0	0	1	0	0	----- 0
-1	1	1	1	0	1	1	-----
-2	1	1	0	0	1	0	----- negative voltage
-3	1	0	1	0	0	1	-----
	{2's complement representation}						

signal will be bandlimited by the steep filter to achieve the desired spectral efficiency.

The symbol-rate clock triggers a set of two one-shots to generate the required sampling clock, which will be hard-wired to the decoder. The first one-shot provides a delay which can be adjusted by the 5 k M/T potentiometer.

4.2.3 Filter Characteristics and Signalling Above the Nyquist Rate

In the analysis of our new Model-B for PRS above the Nyquist rate (see Figs.2.3 and 3.2), a brickwall filter (i.e., $\alpha = 0$) was used. A question may arise as to how an ideal brickwall filter can be realized. This can be solved by noting that the idea of using a brickwall filter is to cut off part of the energy below the Nyquist frequency of the signal so as to achieve a higher spectral efficiency. Thus, what we really need is a steep filter which has a cutoff frequency below the Nyquist frequency. In our experiment, a practical steep filter was employed, which approximates an $\alpha = 0.08$ raised-cosine filter and was designed by Karkar Electronics, Inc., San Francisco. First, let us clarify some parameters which are related to the filter characteristics. The measured amplitude characteristics of the $\alpha = .08$ raised-cosine filter without and with x/sin(x) equalization are shown in Figs.4.4 and 4.5, respectively. The x/sin(x) equalizer is to compensate for the rectangular input pulse induced sin(x)/x shaping of the spectrum. This filter has at least 35 dB attenuation at and beyond 126.6 kHz. The 3 dB cutoff frequency is

$$f_{3dB} = 126.6/(1+\alpha) = 126.6/1.08 = 117 \text{ kHz}. \qquad (4.4)$$

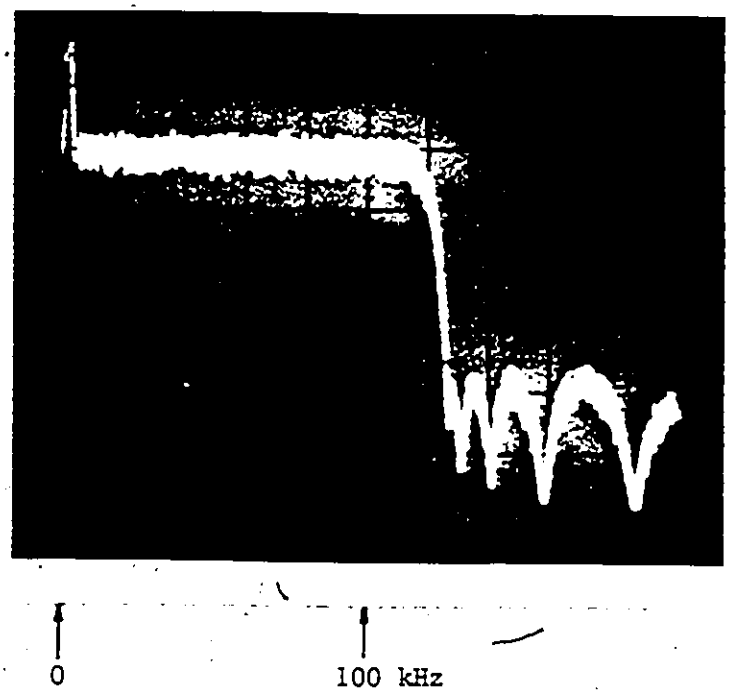


Fig.4.4 Measured amplitude characteristic of an approximate α = .08 raised-cosine filter.

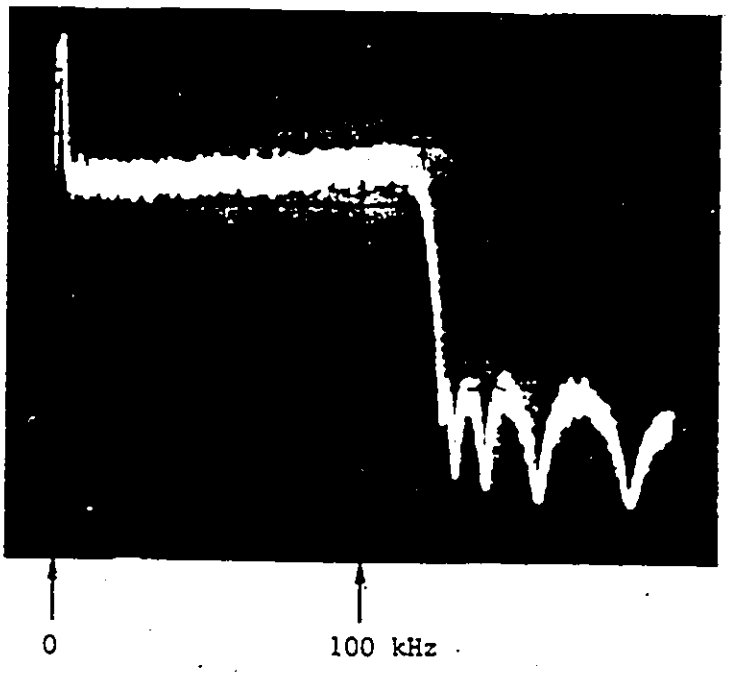


Fig.4.5 Measured amplitude characteristic of an approximate α = .08 raised-cosine filter with x/sin(x) equalization.

Thus, the filter was originally designed for transmitting data at a nominal symbol rate of

$$f_{sn} = 2f_{3dB} = 234.4 \text{ kBaud.} \tag{4.5}$$

For 4-level PAM signalling (or equivalently 7-level PRS), the nominal bit rate is

$$f_{bn} = 2f_{sn} = 468.8 \text{ kb/s.} \tag{4.6}$$

To investigate the performance of signalling above the Nyquist rate, we simply increase the input signalling rate higher than the nominal rate. Due to the nature of our hardware design, the delay elements (i.e., U4 in Fig.4.3(a)) yield a delay which is exactly equal to the symbol interval of the new signalling rate. Since the filter is fixed, as we increase the input signalling rate, the frequency components near the Nyquist frequency of the signal will be cut. Higher the increase in signalling rate, the more frequency components below the Nyquist frequency will be cut, resulting in a higher spectral efficiency. This explains that the function of the experimental system is equivalent to that of Fig.4.1.

4.2.4 Decoder Operation

With reference to Fig.4.2, the receiver employs, in a straightforward manner, 6 slicers (threshold detectors) to quantize the received signal into one of the 7 levels, $\hat{z}_i$. In the absence of noise, $\hat{z}_i$ is identical to z_i. The parallel binary outputs of the slicers are combined by

using standard logic gates which perform the modulo-4 operation and Gray code decoding to yield the original parallel bits. Finally, a parallel-to-serial (P/S) conversion takes place at the last stage to generate a replica of the transmitted binary input data.

An example of the precoding and decoding operations for class-IV PRS is illustrated in Table 4.1 for 4-level signalling. Detailed description of the decoder circuit follows.

4.2.5 Detailed Description of the Decoder Circuit

Referring to Fig.4.6, the received signal, r(t), is passed through a 7-level quantizer consisting of six voltage comparators and six D flip-flops. The threshold voltages $V_1, V_2, \ldots, V_6$ are set properly in the middle of each eye of the eye diagram. Denote by r_k the sampled value of r(t) at $t = kT_s + \tau$, i.e.,

$$r_k = r(t)\big|_{t = kT_s + \tau} \tag{4.7}$$

where

k is an integer,

τ is the sampler's phase, which is to be adjusted to the optimum sampling instants.

The output of 74174 depends on the location of r_k, as illustrated by Table 4.4.

TABLE 4.3 Four-level signal transmission with class-IV PRS.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	⋯
x_i	0	2	1	2	3	0	3	3	2	1	1	2	1	⋯
y_i	0	2	1	0	0	0	3	3	1	0	2	2	3	⋯
z_i	0	2	1	-2	-1	0	3	3	-2	-3	1	2	1	⋯
$\hat{x}_i$	0	2	1	2	3	0	3	3	2	1	1	2	1	⋯

$y_i = (x_i + y_{i-2}) \text{ Mod } 4, \quad z_i = y_i - y_{i-2}$

$\hat{x}_i = \hat{z}_i \text{ Mod } 4 = z_i \text{ Mod } 4 = x_i$ (assume no noise)

TABLE 4.4 DECODING TABLE WITH REFERENCE TO Fig.4.6.

	74174						Decoded symbol $\hat{r}_k$	$\hat{r}_k$ Mod 4	7420	
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6			$\hat{Q}_1$	$\hat{Q}_2$
$V_1 < r_k$	1	1	1	1	1	1	3	3 (1 1)	1	0
$V_2 < r_k < V_1$	0	1	1	1	1	1	2	2 (1 0)	1	1
$V_3 < r_k < V_2$	0	0	1	1	1	1	1	1 (0 1)	0	1
$V_4 < r_k < V_3$	0	0	0	1	1	1	0	0 (0 0)	0	0
$V_5 < r_k < V_4$	0	0	0	0	1	1	-1	3 (1 1)	1	0
$V_6 < r_k < V_5$	0	0	0	0	0	1	-2	2 (1 0)	1	1
$r_k < V_6$	0	0	0	0	0	0	-3	1 (0 1)	0	1

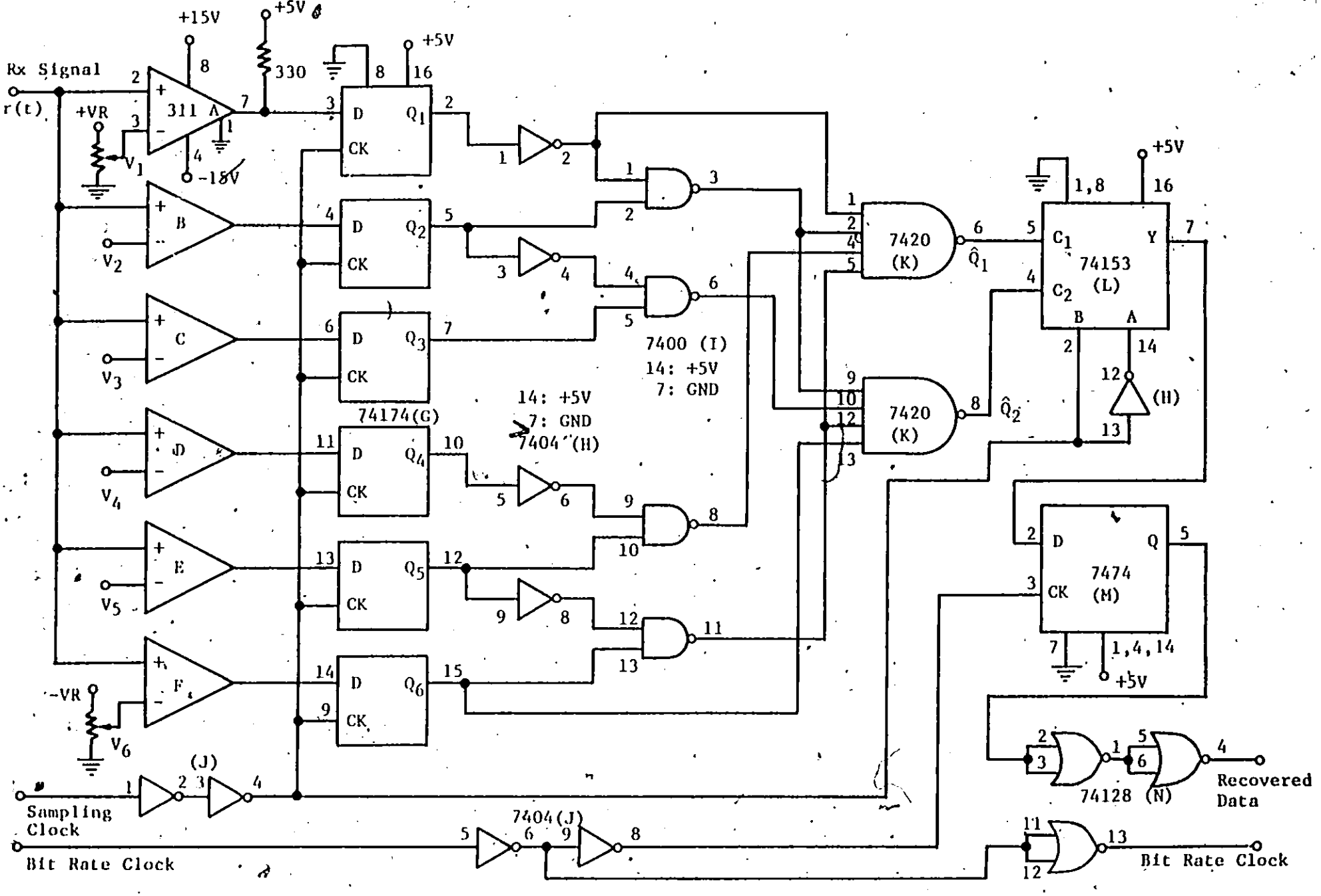


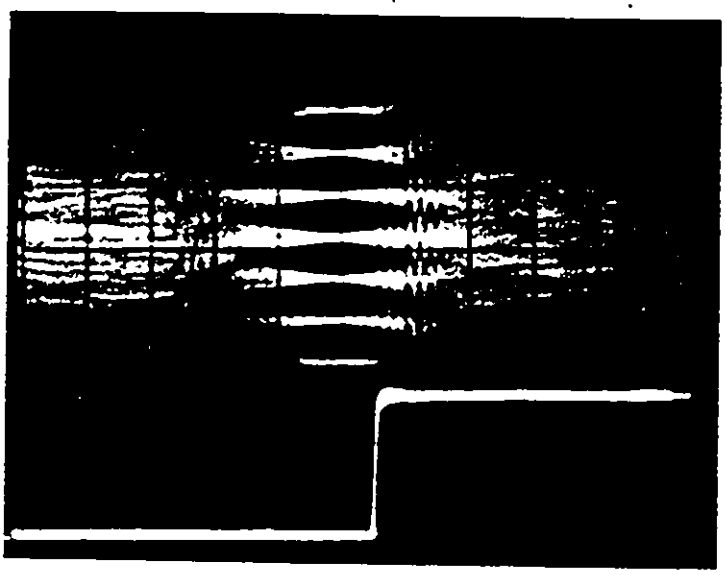
Fig.4.6 Circuit diagram of a 7-level class-IV PRS decoder.

Following 74174, the combinational logic gates 7404, 7400, and 7420 are combined to yield the desired output $\hat{Q}_1$ and $\hat{Q}_2$ by performing the decoding procedure which includes Mod 4 operation and gray-code decoding (see also Fig.4.2). With the aid of Table 4.4, the circuit connections among 7420, 7400, and 74174 are very straightforward.

Finally, 74153 converts the 2 parallel bits into a serial bit stream through the correct timing provided by the sampling clock. Again 7474 is employed to yield a clean recovered data stream and 74128 is used to drive a 50 Ω load, which may be the input impedance of an error analyzer. This completes the description of the circuit operation.

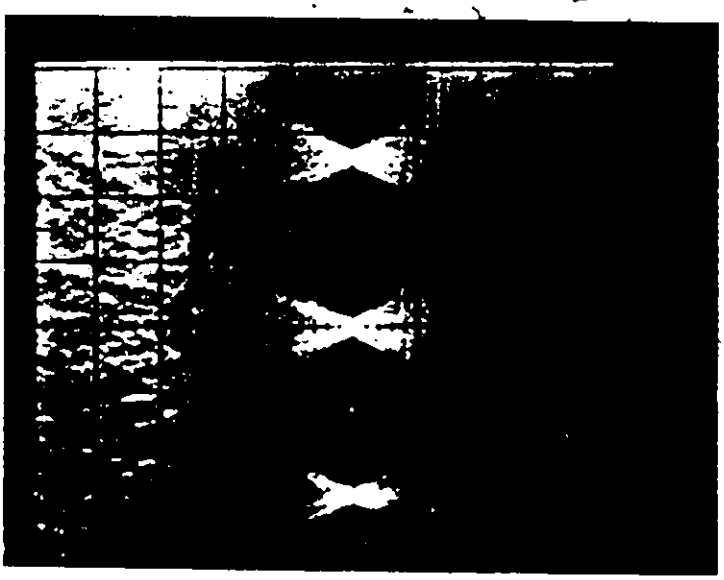
4.3 EXPERIMENTAL RESULTS AND DISCUSSIONS

A typical measured eye diagram and the sampling clock are shown in Fig.4.7 for the 7-level class-IV PRS at the Nyquist rate. It is worth noting that the optimum sampling instant seems not to be at the middle of the eye, as observed from Fig.4.7. According to our experience from the experimental evaluation of the BER performance, we found that the optimum sampling instant which results in minimum BER is indeed located at a point which is to the right of where the maximum eye opening is. This is because that the commercially available comparator IC's which are used in our circuit (311 in Fig.4.6) have a certain amount of response time (typically 300 ns from the data manual) which causes a delay for a stable output. The precise position of the optimum sampling instant can be located by adjusting the sampling clock while looking at an error analyzer until it gives a minimum reading of BER.



V:
Upper trace:
200 mV/div
Lower trace:
2 V/div
H: 283 ns/div

Fig.4.7 Measured eye diagram of the 7-level class-IV PRS signal and the sampling clock.



V: 50 mV/div
H: 283 ns/div
f_b = 473.6 kb/s.

Fig.4.8 Close-up of the inner two eyes of the 7-level class-IV PRS signal.

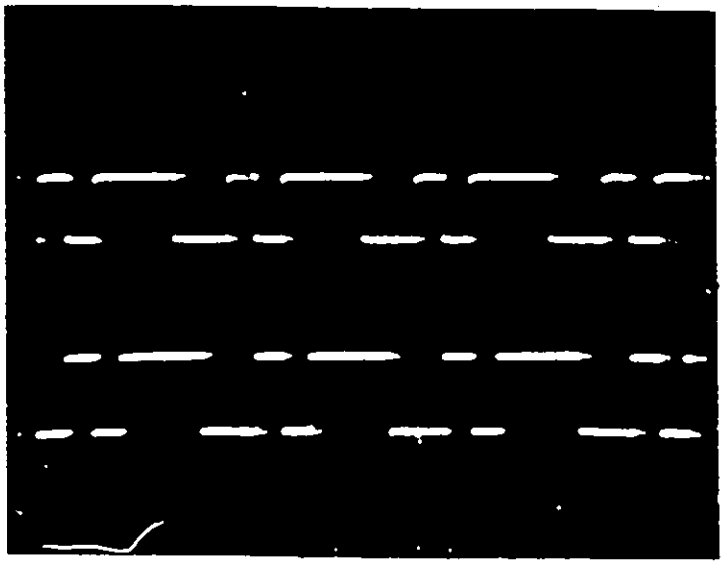


Fig.4.9 Illustrative transmitted and decoded data for a PRBS pattern of 1110010.

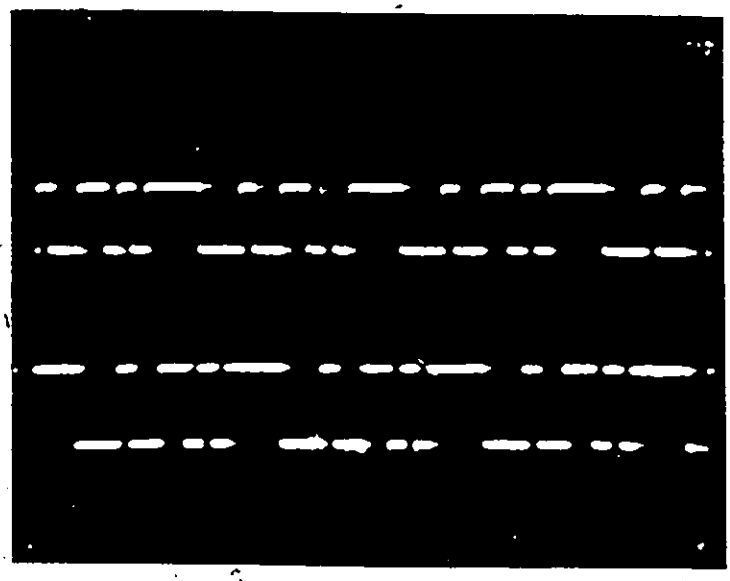


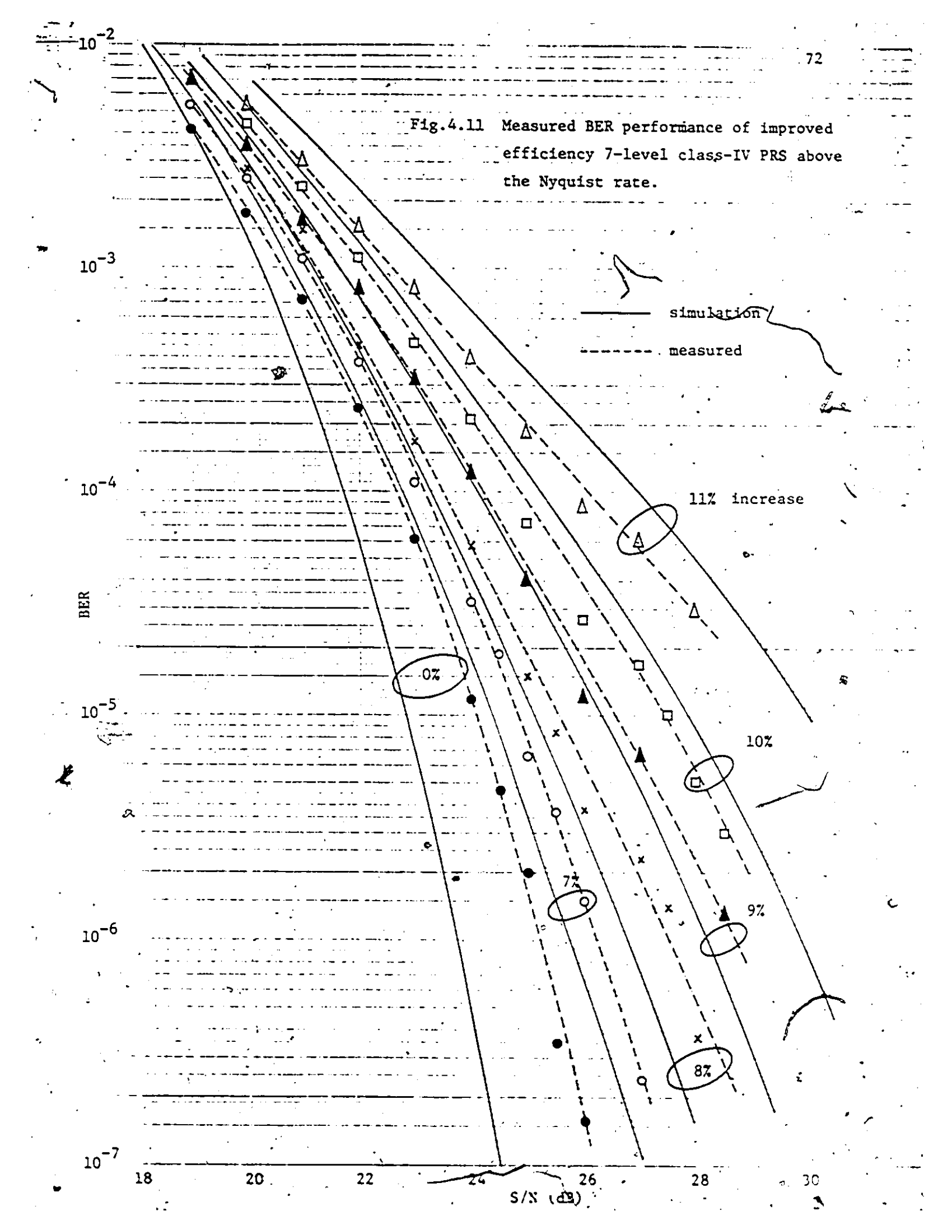
Fig.4.10 Illustrative transmitted and decoded data for a PRBS pattern of 100110101111000.

Fig.4.8 shows the close-up of the inner two eyes. Two decoded data patterns are illustrated in Fig.4.9 and 4.10 for PRBS lengths of 7 and 15 bits, respectively.

The BER performances versus S/N were measured for various percentage increases above the Nyquist rate. The signal and noise powers were measured at the output of the lowpass filter just before the threshold detectors. The measured results are shown in dotted curves in Fig.4.11. To confirm the accuracy of our simulation method, we simulated the experimental 7-level class-IV PRS system and the simulated results are also given in Fig.4.11 by the solid curves for comparison. We found that the measured results are basically in good agreement with the simulated ones.

A comparison at the 10^{-6} error rate indicates the experimental result is 1.5 dB away from the theory for the 7-level class-IV PRS at the Nyquist rate. The 1.5 dB degradation is mainly due to practical imperfections in filters and circuit layout since the experimental hardware is in breadboard rather than printed circuit form. As observed from Fig.4.11, for the increase of up to 9% above the Nyquist rate, the measured results are slightly worse than those by simulation. This is in general due to hardware imperfections. However, the situation is reverse when the increase is higher than 9%. This can be explained as follows. In the hardware measurements, the threshold voltages were always adjusted to the middle of each corresponding eye, whereas they were fixed and evenly spaced in the simulateion. As the signaling rate is increased higher, the 6 eyes become uneven as can be seen from the

Fig.4.11 Measured BER performance of improved efficiency 7-level class-IV PRS above the Nyquist rate.

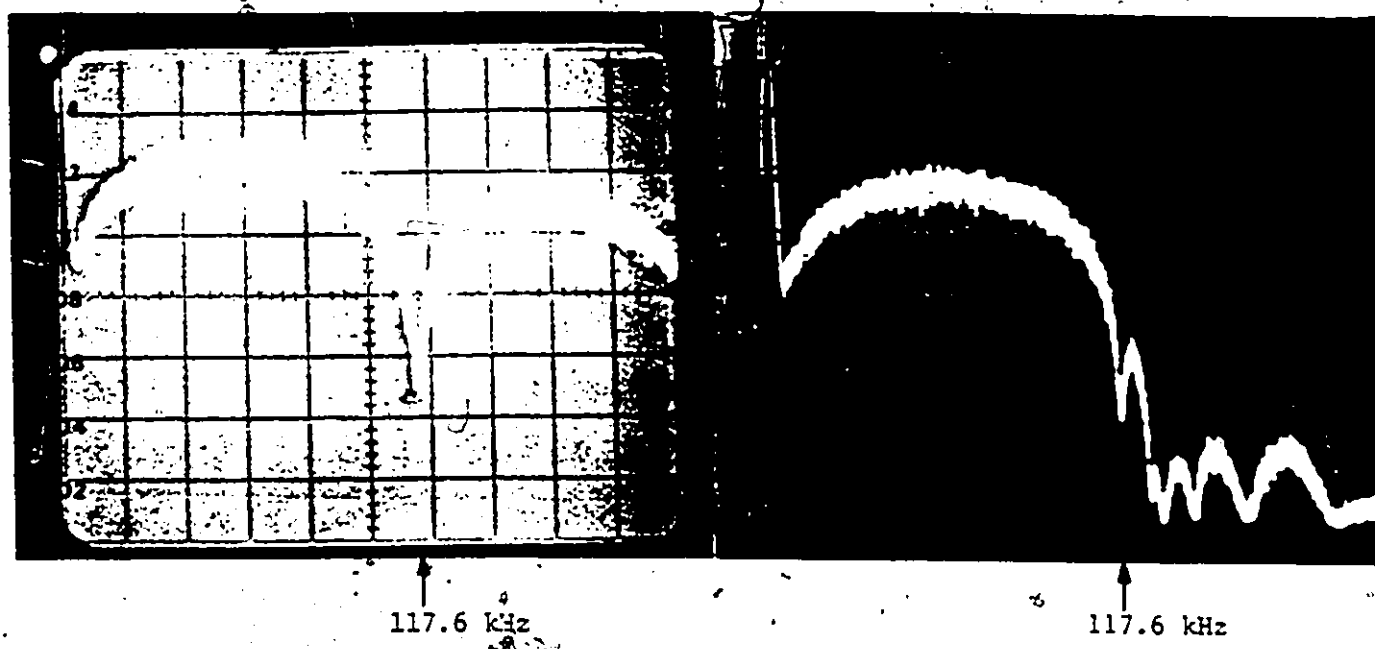


eye diagrams. The original threshold voltages are no longer at the optimum positions. Thus the measured results are found better than the simulated ones when the increase is higher than 9%.

We also notice from Fig.4.11 that for an increase of up to 9% above the Nyquist rate for the 7-level class-IV PRS using our new chopping technique, the degradation is reasonably small. Illustrative examples of the measured eye diagrams and corresponding PSD's are presented in Figs.4.12-4.14 for 0%, 8%, and 12% increases above the Nyquist rate, respectively. Note that all of the PSD's measured before the lowpass filter show nulls at dc and the Nyquist frequency of the signal. This confirms the frequency-domain behavior of the class-IV PRS.

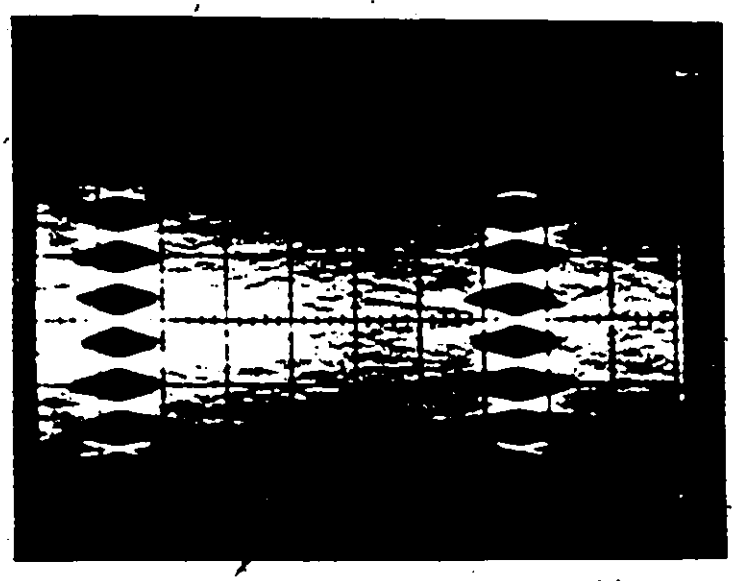
As pointed out previously, the wideband spectrum of the signal is bandlimited by the lowpass filter whose characteristics are fixed. As the signalling rate is increased higher, the more frequency components below the Nyquist frequency will be cut. This can be observed from the PSD's measured after the LPF in Figs.4.12-14. As seen from the measured eye diagrams in Figs.4.12-4.14, the "undesired" ISI increases as the percentage increase above the Nyquist rate increases, which eventually causes performance degradation to an unacceptable level.

Fig.4.15 shows the photo of the experimental set-up in the Digital Communications Laboratory, Department of Electrical Engineering, University of Ottawa.



(a) PSD before the lowpass filter.

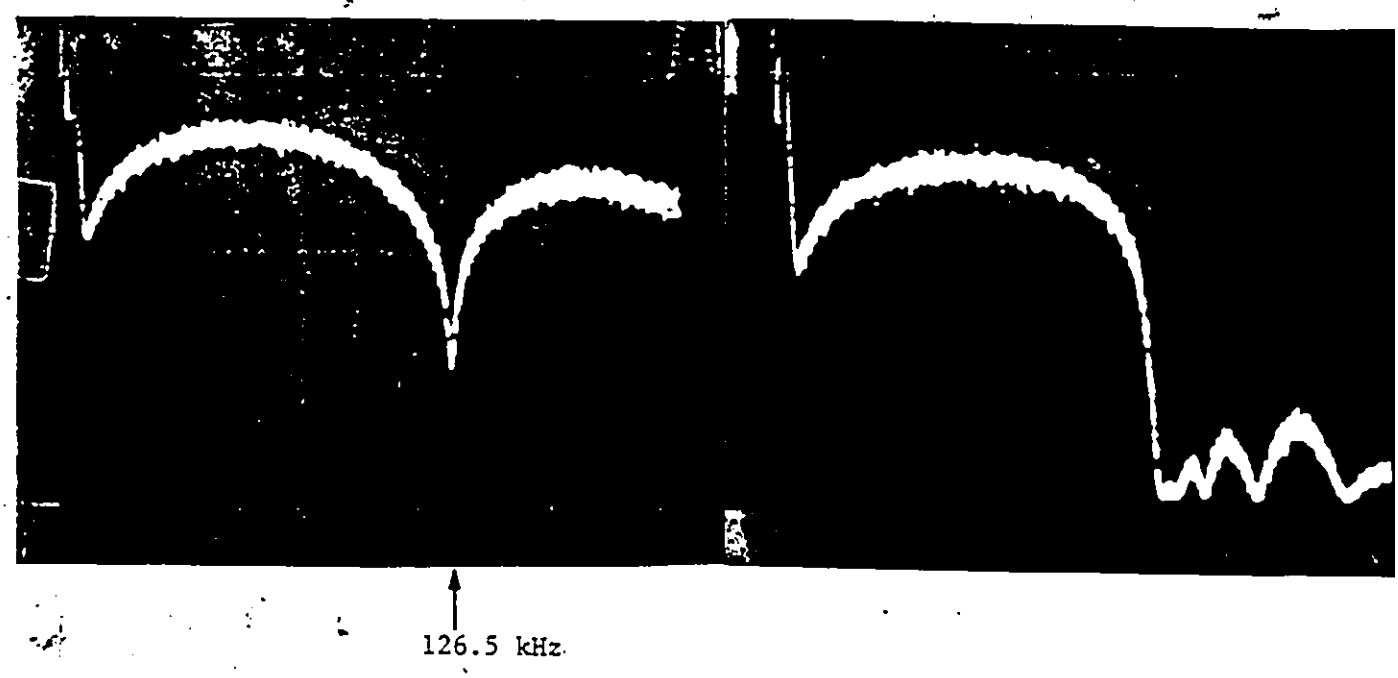
(b) PSD after the lowpass filter.



(c) The received eye diagram.

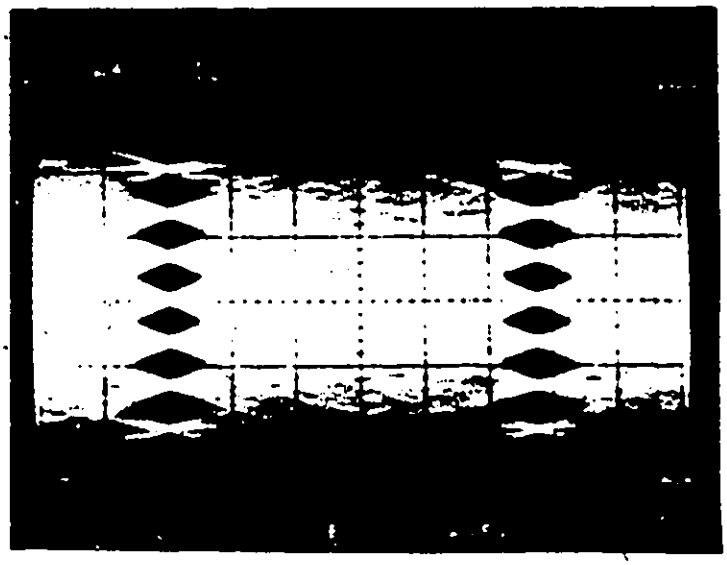
Fig.4.12 Measured PSD's and corresponding eye-diagram of our 7-level class-IV PRS signal with f_b = 470.3 kb/s.

V: 10 dB/div
H: 20 kHz/div



(a) PSD before the lowpass filter.

(b) PSD after the lowpass filter.



V: 200 mV/div

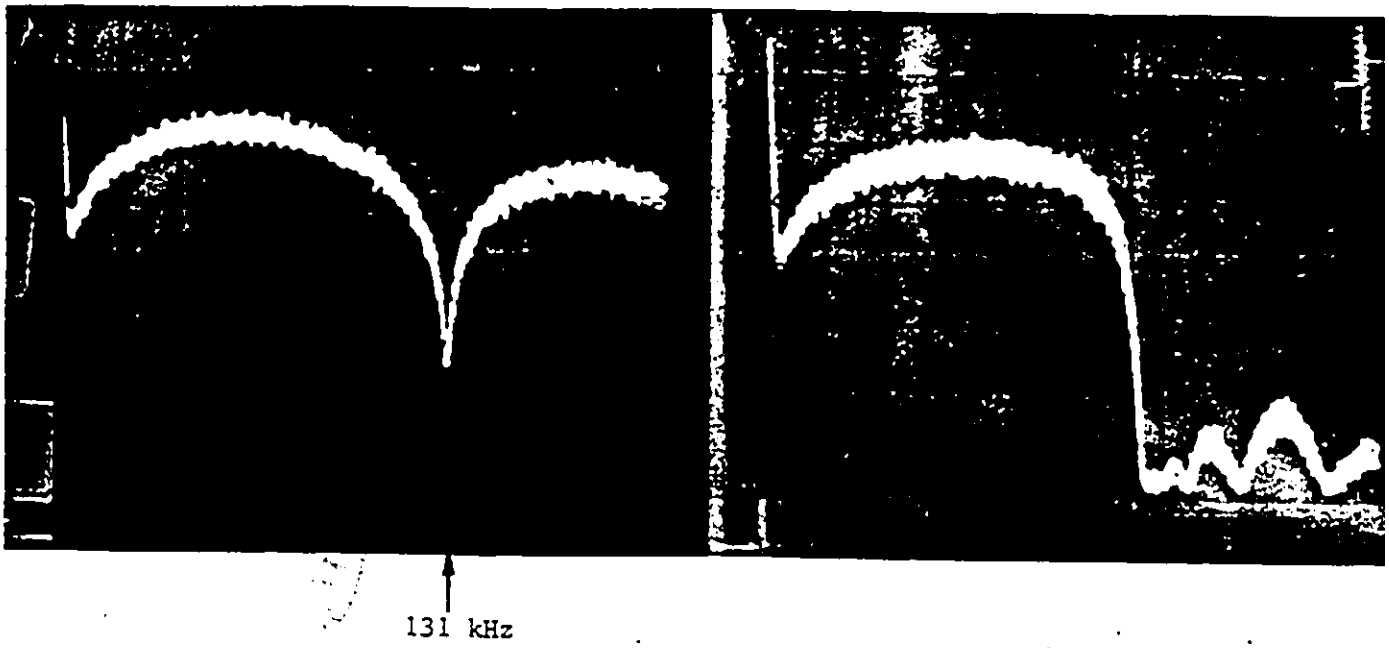
H: 685 ns/div

(c) The received eye diagram.

Fig.4.13 Measured PSD's and corresponding eye diagram of our 7-level class-IV PRS signal with f_b = 506.2 kb/s (an 8% increase above the Nyquist rate).

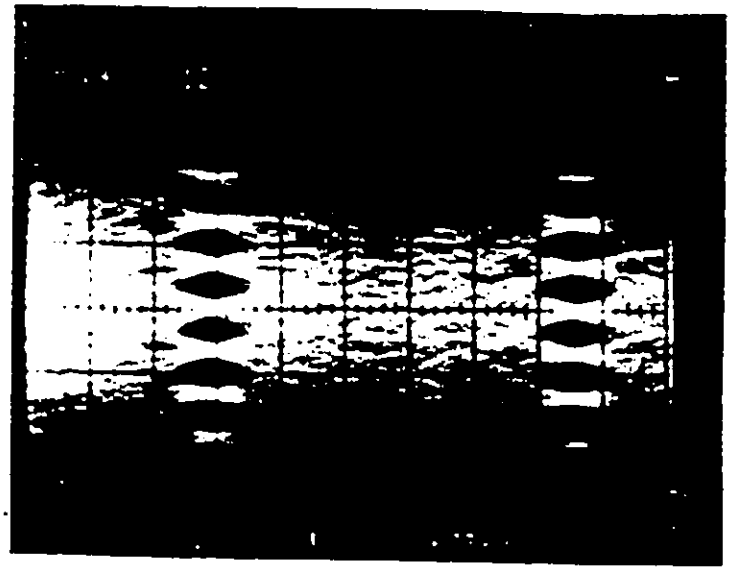
V: 10 dB/div

H: 20 kHz/div



(a) PSD before the lowpass filter.

(b) PSD after the lowpass filter.



V: 200 mV/div

H: 673 ns/div

(c) The received eye diagram.

Fig.4.14 Measured PSD's and corresponding eye diagram of our 7-level class-IV PRS signal with f_b = 524 kb/s (a 12% increase above the Nyquist rate).

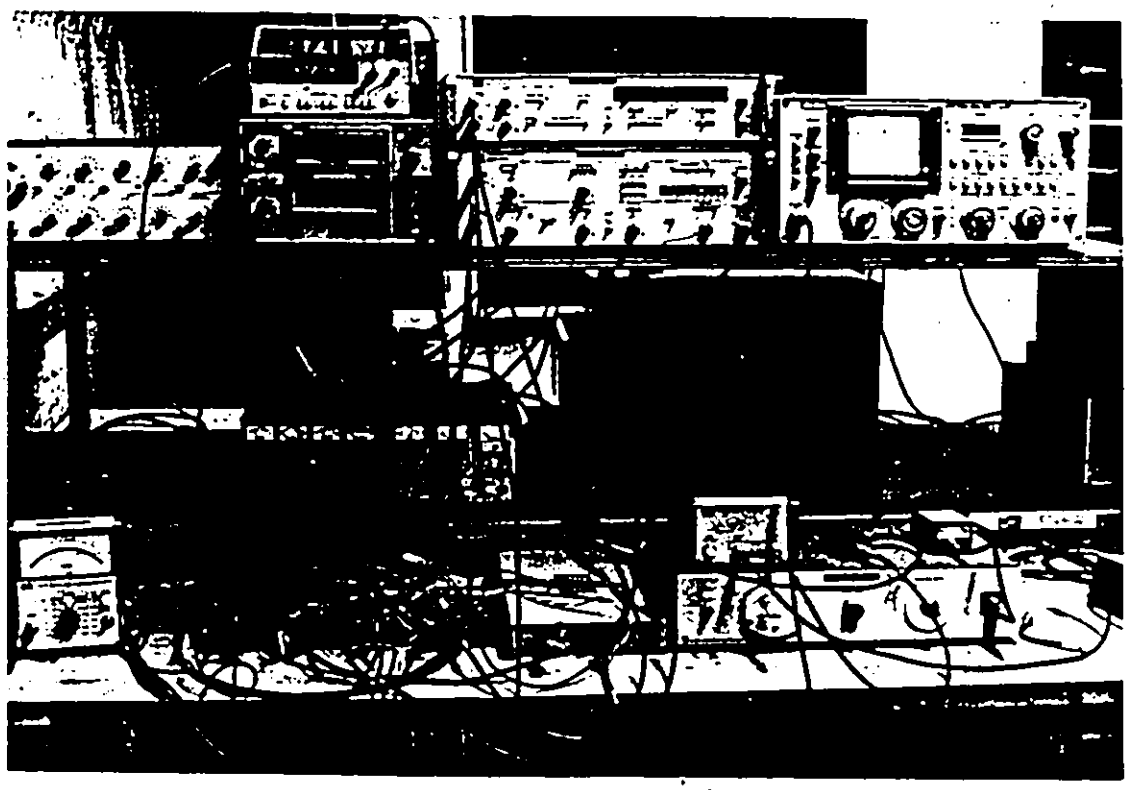


Fig.4.15 Photograph of the experimental set-up in the Digital Communications Laboratory, Department of Electrical Engineering, University of Ottawa.

4.4 CONCLUSIONS

To verify the new concept previously presented for multi-level PRS above the Nyquist rate, a 7-level class-IV PRS baseband system has been designed and implemented. System design principles were highlighted and detailed circuit operation was described. Practical problems encountered during the experiment and the valuable experience learned from the measurement were also discussed. The measured BER performance of the improved efficiency 7-level PRS was confirmed to agree well with the simulated results.

Chapter V

256-QAM MODEM PERFORMANCE IN DISTORTED CHANNELS

In Chapter 1, after introducing the demand for highly spectrum-efficient modulation schemes, it was concluded that 256-QAM is one of the candidates. Essentially, 256-QAM is equivalent to a 16-level pulse-amplitude-modulation (PAM) in baseband. Multi-level signals are more sensitive to various impairments because the same impairment (e.g., filter imperfection) yields higher ISI in multi-level systems than in a binary system. As a result, filters specifications which are suitable for a binary system do not necessarily guarantee a satisfactory performance for a 16-level system. Thus it is important to investigate the effect of channel distortions [37], [38] and/or hardware imperfections on 256-QAM.

The objective of this chapter is to quantitatively evaluate the effects of group delay and amplitude distortions with linear, parabolic, or sinusoidal characteristic on the performance of a 256-QAM modem. The first part of this chapter will be devoted to a brief description of our simulation method. Simulated results for a 120 Mb/s system will then be presented. Finally, in order to efficiently apply the simulated results to systems with different bit rates, a simple scaling procedure will be described.

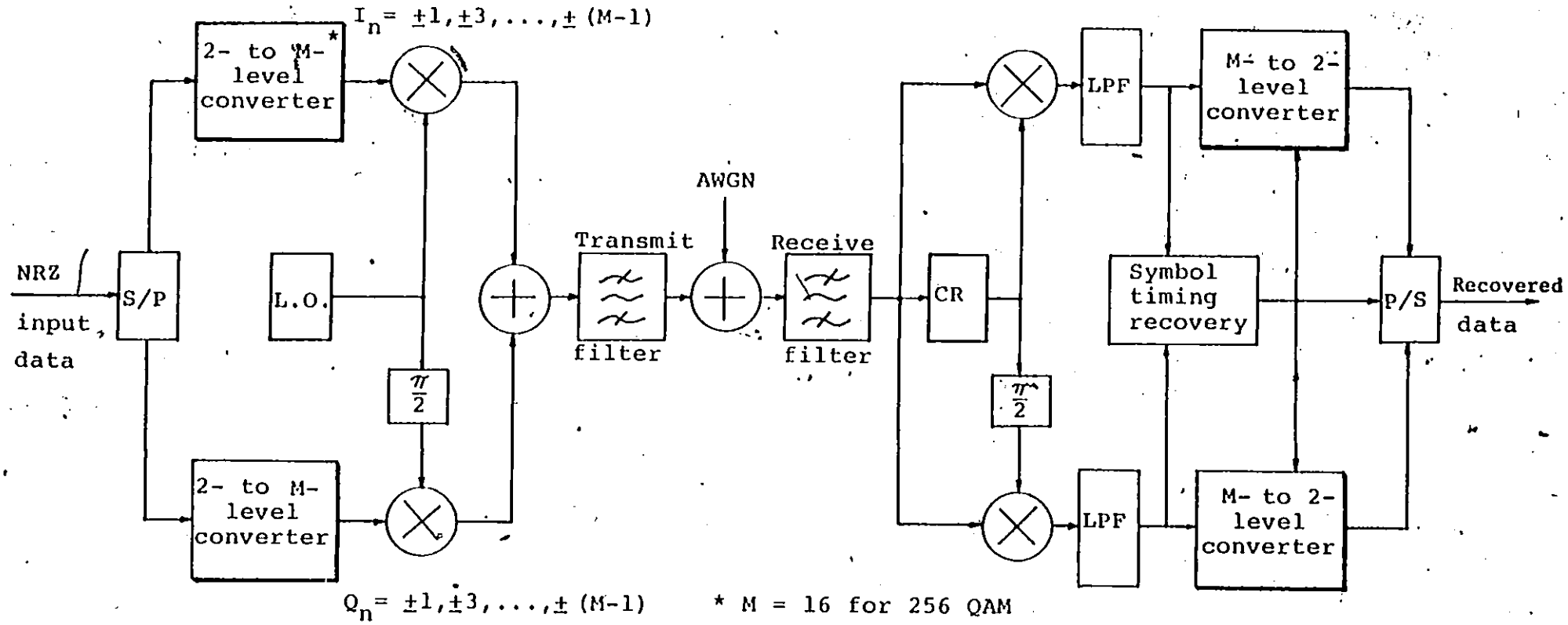


Fig.5.1 256 and/or 64-QAM system block diagram.

5.1 DESCRIPTION OF COMPUTER SIMULATIONS

Fig.5.1 shows the block diagram of a 256-QAM system. The modulated 256-QAM signal can be represented by (5.1).

$$s(t) = \mathrm{Re}\{ \sum_{n=0}^{\infty} (I_n + jQ_n)\, g(t-nT_s)\, e^{j2\pi f_o t} \} \qquad (5.1)$$

where

Re{ } denotes real part of the complex-valued quantity in the brackets,

f_o is the carrier frequency,

$1/T_s$ is the symbol rate; for 256-QAM $T_s = 8T_b$;

$1/T_b$ is the bit rate,

$g(t)$ is a pulse defined by

$$g(t) = \begin{cases} 1 & \text{for } 0 \le t \le T_s \\ 0 & \text{elsewhere.} \end{cases}$$

I_n and $Q_n = \pm 1, \pm 3, \ldots, \pm 15$ are the sampled values of the in-phase and quadrature symbols.

The simulation is performed entirely in the complex baseband form as shown in Fig.5.2 for a system with a bit rate of 120 Mb/s, i.e., with a symbol rate of 15 MBaud. The choice of this Baud rate is for an easy comparison with the performance of 90 Mb/s 64-QAM systems which have the same Baud rate and were used in [8]. The results can be applied to another bit rate by appropriate scaling. In the simulation a perfect carrier and symbol timing is assumed. For filtering we use FFT techniques to alternate between the frequency and time domains. The assumed pseudo-random-binary-sequence (PRBS) sequence length is 16,384 bits and the number of samples per symbol is 32.

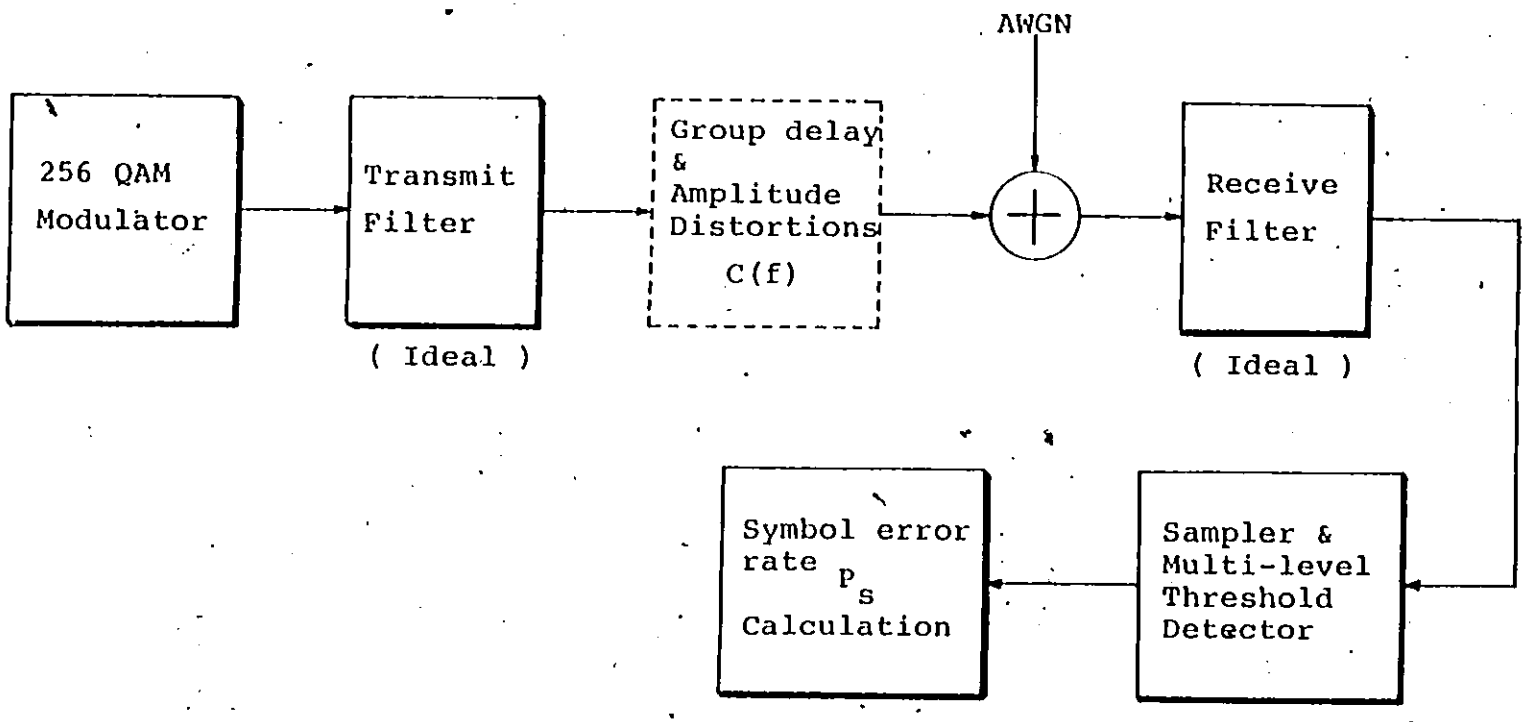


Fig.5.2 Computer simulation model.

The transmit and receive filters are assumed as ideal square root of raised-cosine filters with x/sin(x) equalization in the transmitter so that the whole system satisfies the Nyquist 1st criterion [14]. The resulting eye diagrams for α = 0.1 and 0.4 are shown in Fig.5.3(a) and (b), where α denotes the roll-off factor, that is the ratio of the excess bandwidth to the Nyquist band. The eye diagram of an α = 0.2 unequalized channel, having a sinusoidal group delay distortion, is illustrated in Fig.5.3(c). Illustrative measurement results are shown in Fig.5.4.

Note that for ideal filters there is no ISI at the optimum sampling instants. However, data transition jitter is very significant, particularly in the α = 0.1 case. This indicates a high-precision sampling clock would be required.

For a specified power of white Gaussian noise at the threshold detector input, the error probability of the i-th symbol with respect to the in-phase channel is calculated as follows:

$$p_{si}^{I} = \begin{cases} \frac{1}{2}\,\mathrm{erfc}\left(\dfrac{|\bar{S}_i - \mathrm{THR1}_i|}{\sqrt{2}\sigma}\right), & \text{for } I_i = \pm 15 \\ \frac{1}{2}\,\mathrm{erfc}\left(\dfrac{|\bar{S}_i - \mathrm{THR1}_i|}{\sqrt{2}\sigma}\right) + \frac{1}{2}\,\mathrm{erfc}\left(\dfrac{|\mathrm{THR2}_i - \bar{S}_i|}{\sqrt{2}\sigma}\right), & \text{for } I_i = \pm 1, \pm 3, \ldots, \pm 13. \end{cases} \tag{5.2}$$

where

I_i is the i-th transmitted symbol of the in-phase channel,

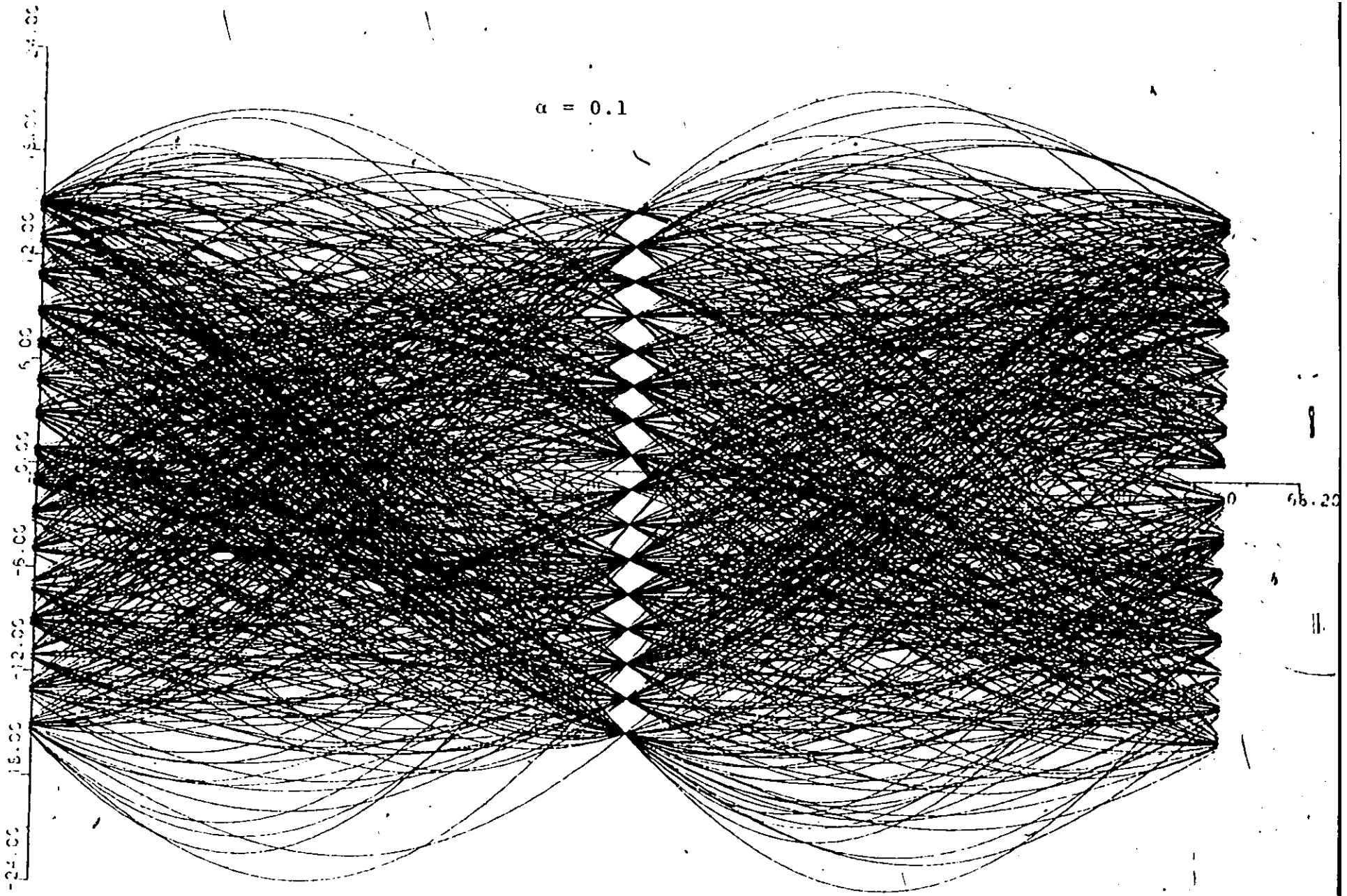


Fig.5.3(a)- Eye diagram of 256-QAM with α = 0.1.

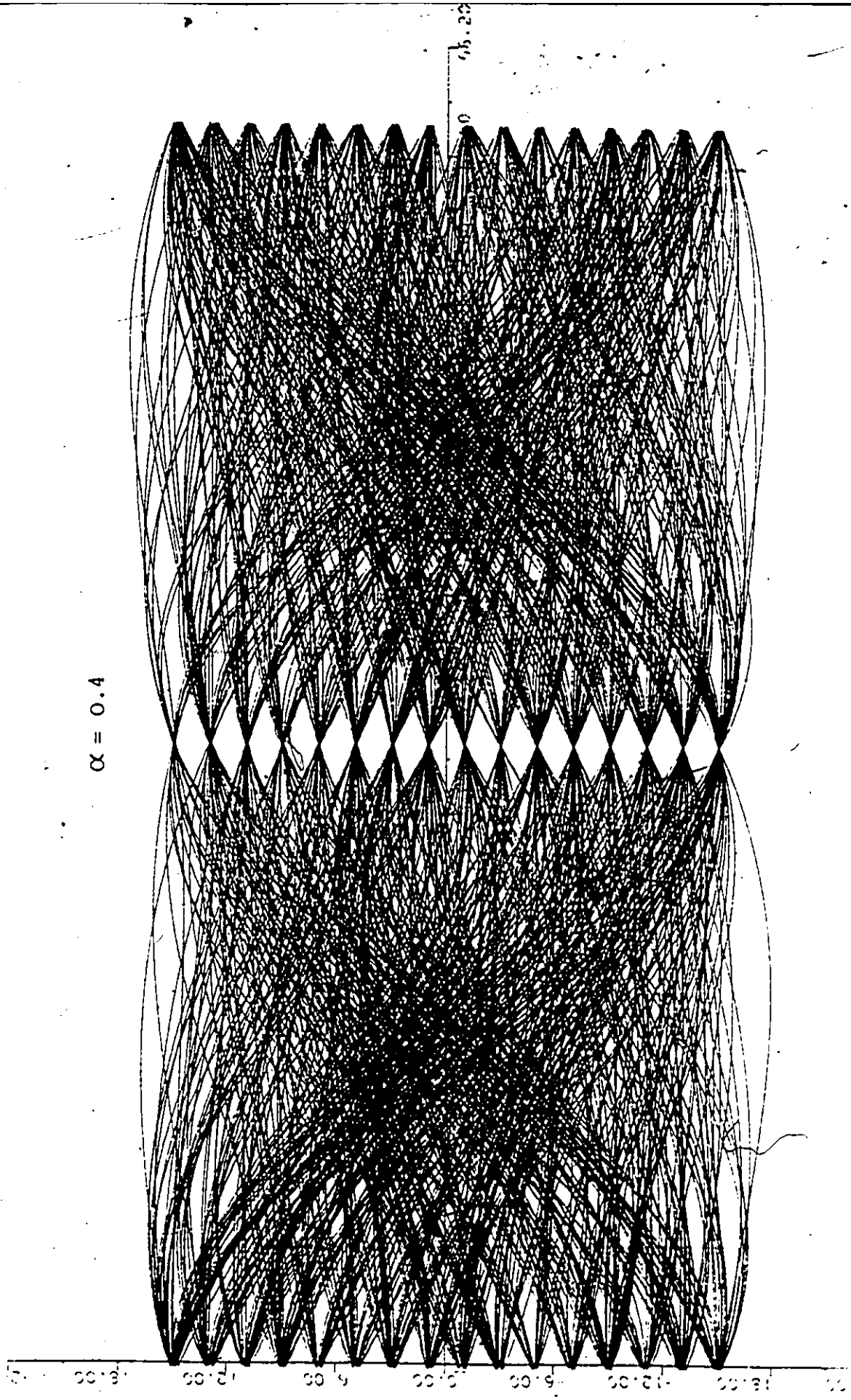


Fig.5.3(b) Eye diagram of 256-QAM with $\alpha = 0.4$.

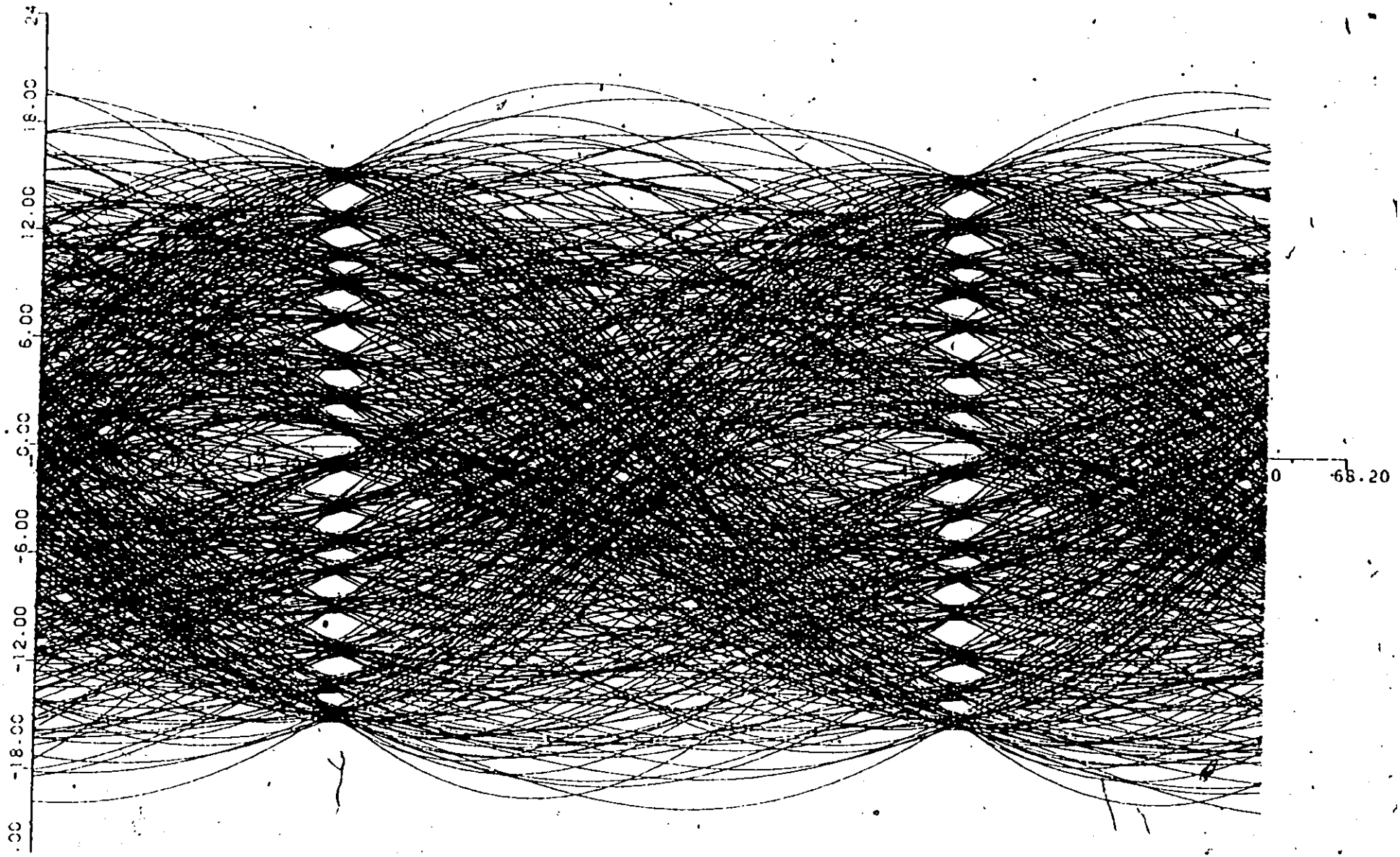
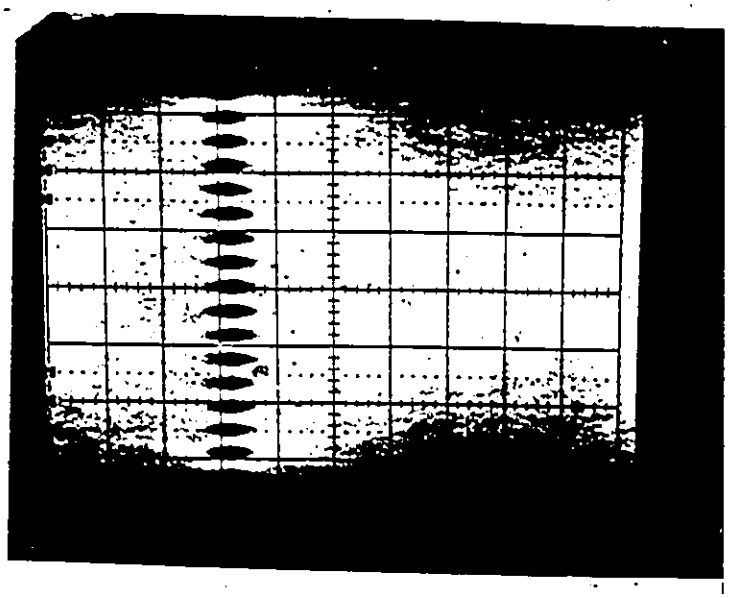
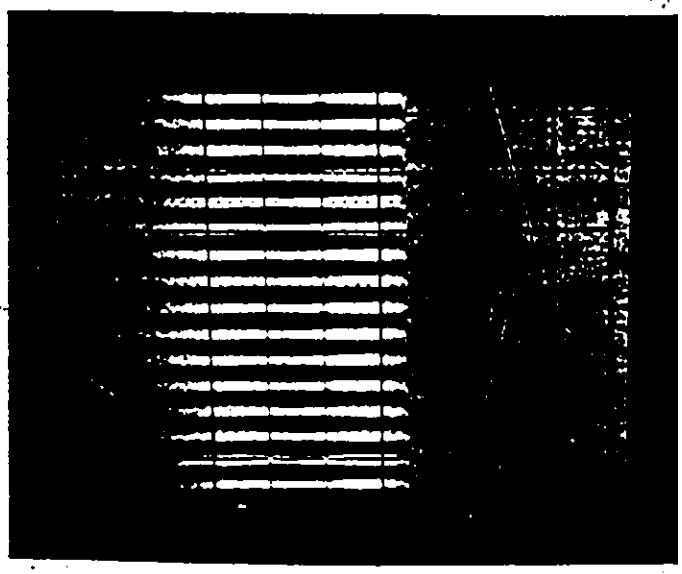


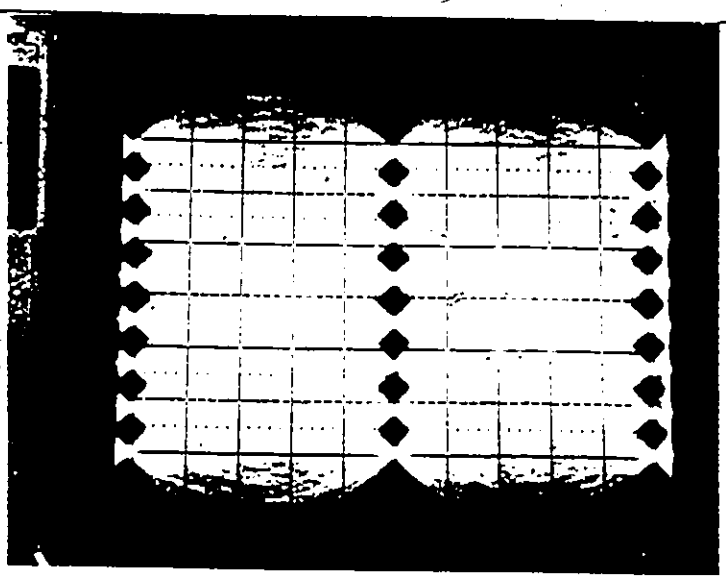
Fig.5.3(c) Eye diagram of 256-QAM (α = 0.2) with a sinusoidal group delay distortion, $D(f)=S_D \sin(2\pi Kf/2f_{BW})$, S_D = 12 ns and K = 4. Bit rate f_b = 120 Mb/s.



(a)
H: 500 ns/div
V: not calibrated.



(b)
not calibrated.



(c)
H: 1 us/div
V: not calibrated.

Fig.5.4 Measured 16-level and 8-level PAM eye diagrams, which are the equivalent baseband signals of 256-QAM and 64-QAM, respectively. The symbol rate is 200 kBaud, which corresponds to bit rates of 800 kb/s (16-level PAM) and 600 kb/s (8-level PAM). Raised-cosine Nyquist channel filters (f_N = 100 kHz) having a roll-off factor α = 0.2 and a 55 dB attenuation beyond 120 kHz, designed by Karkar Electronics, Inc., are used in this experiment. Photographs: (a) and (b) for 16-level PAM, (c) for 8-level PAM.

$\bar{S}_i$ is the magnitude of the i-th received sample of the in-phase channel,

$THR1_i = |I_i| - 1$; $THR2_i = |I_i| + 1$;

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt,$$

σ^2 is the received noise power at the threshold detector input.

The error probability P_{si}^{Q} with respect to the quadrature channel is obtained similarly from (5.2) and for a sequence of N symbols in each channel the average symbol error rate P_s is calculated as follows :

$$P_s = (1/N) \sum_{i=1}^{N} \tfrac{1}{2} (P_{si}^{I} + P_{si}^{Q}) \qquad -(5.3)$$

5.2 GROUP DELAY DISTORTIONS

To investigate the effects of selective fading and/or radio system hardware imperfections, we insert between the transmit and the receive filters a linear filter with the equivalent baseband transfer function C(f), as shown in Fig.5.2. We may express the frequency response C(f) as

$$C(f) = |C(f)| e^{j\theta(f)} \qquad (5.4)$$

and

$$A(f) = |C(f)| \qquad (5.5)$$

$$D(f) = (-1/2\pi) \frac{d\theta(f)}{df} \qquad (5.6)$$

To simulate the group delay distortion we assume there is no amplitude distortion, i.e., A(f) = 1. The group delay distortions are defined as follows :

$$D(f) = \begin{cases} L_D f & \text{for linear group delay,} \\ P_D f^2 & \text{for parabolic group delay,} \\ S_D \sin(2\pi K f/2f_{BW}) & \text{for sinusoidal group delay.} \end{cases} \quad (5.7)$$

where

$f_{BW} = (1+\alpha)f_N$ and f_N is the baseband Nyquist bandwidth.

This $(1+\alpha)f_N$ bandwidth definition is more appropriate than f_N, as it includes the critical effect of the group delay at the edge of the filter attenuation band. In the case of sinusoidal group delay, we present only the results for K = 4. The symbol error rate P_s versus average C/N is computed. Typical results are shown in Fig.5.5. We define a maximum group delay, Tm, in the filter bandwidth $(2f_{BW})$ as follows :

$$\tau_m = \begin{cases} L_D(2f_{BW}) & \text{ns for linear group delay,} \\ P_D(f_{BW})^2 & \text{ns for parabolic group delay,} \\ S_D & \text{ns for sinusoidal group delay.} \end{cases} \quad (5.8)$$

The degradation of C/N as a function of τ_m relative to the case with no distortion, for a P_s of 10^{-4} is shown in Fig.5.6. Note that here we simulate a 120 Mb/s system with α = 0.4. We also include the performance of a 90 Mb/s 64 QAM system with α = 0.4 in Fig.5.6, which is confirmed to be the same as in [8]. Again we notice that for a given

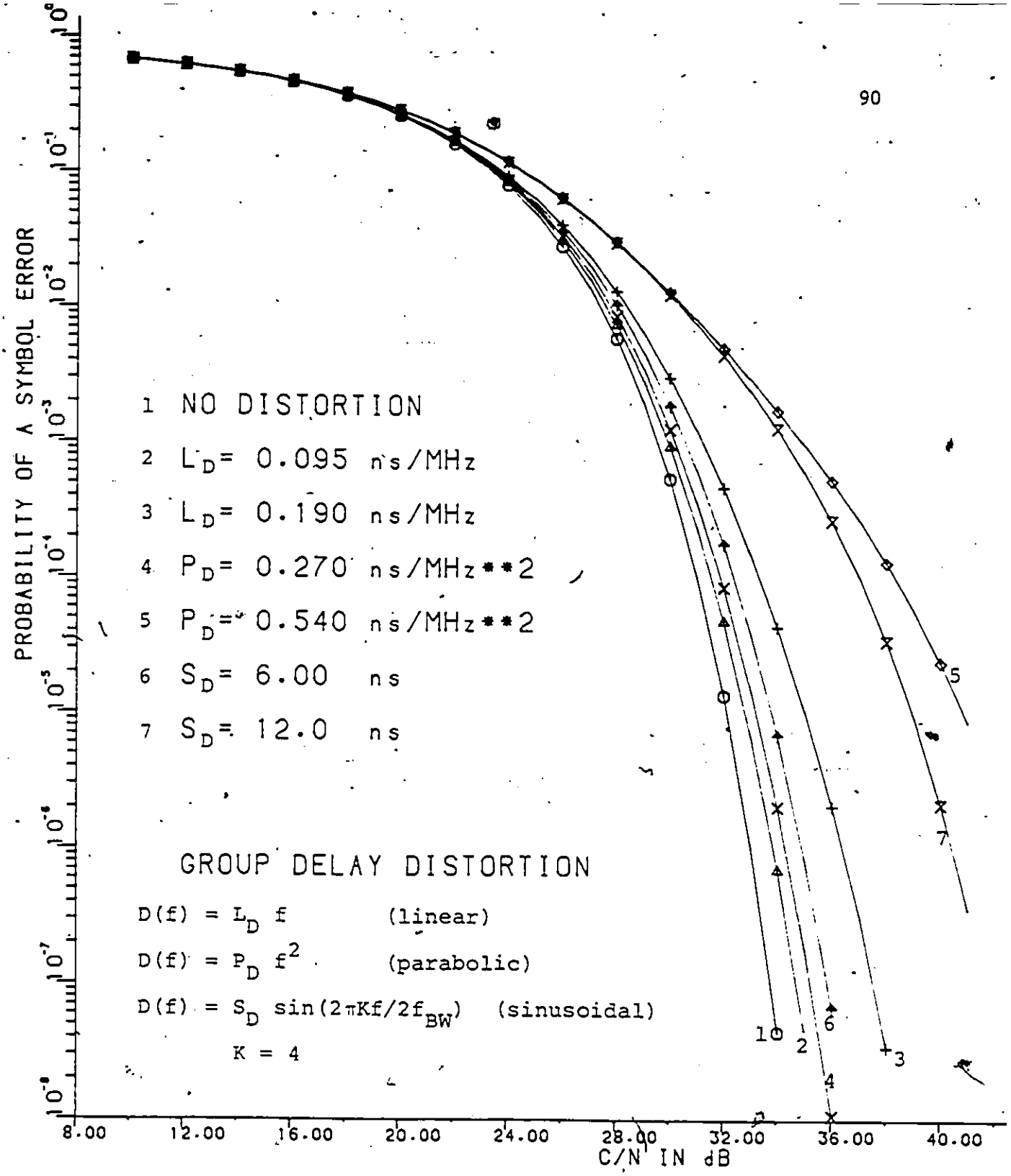


Fig.5.5 P_s versus C/N for linear, parabolic, and sinusoidal group delay distortions for 256-QAM with a bit rate of 120 Mb/s and $\alpha = 0.4$. Noise is defined in the double-sided Nyquist bandwidth, i.e., the equivalent noise bandwidth is 15 MHz.

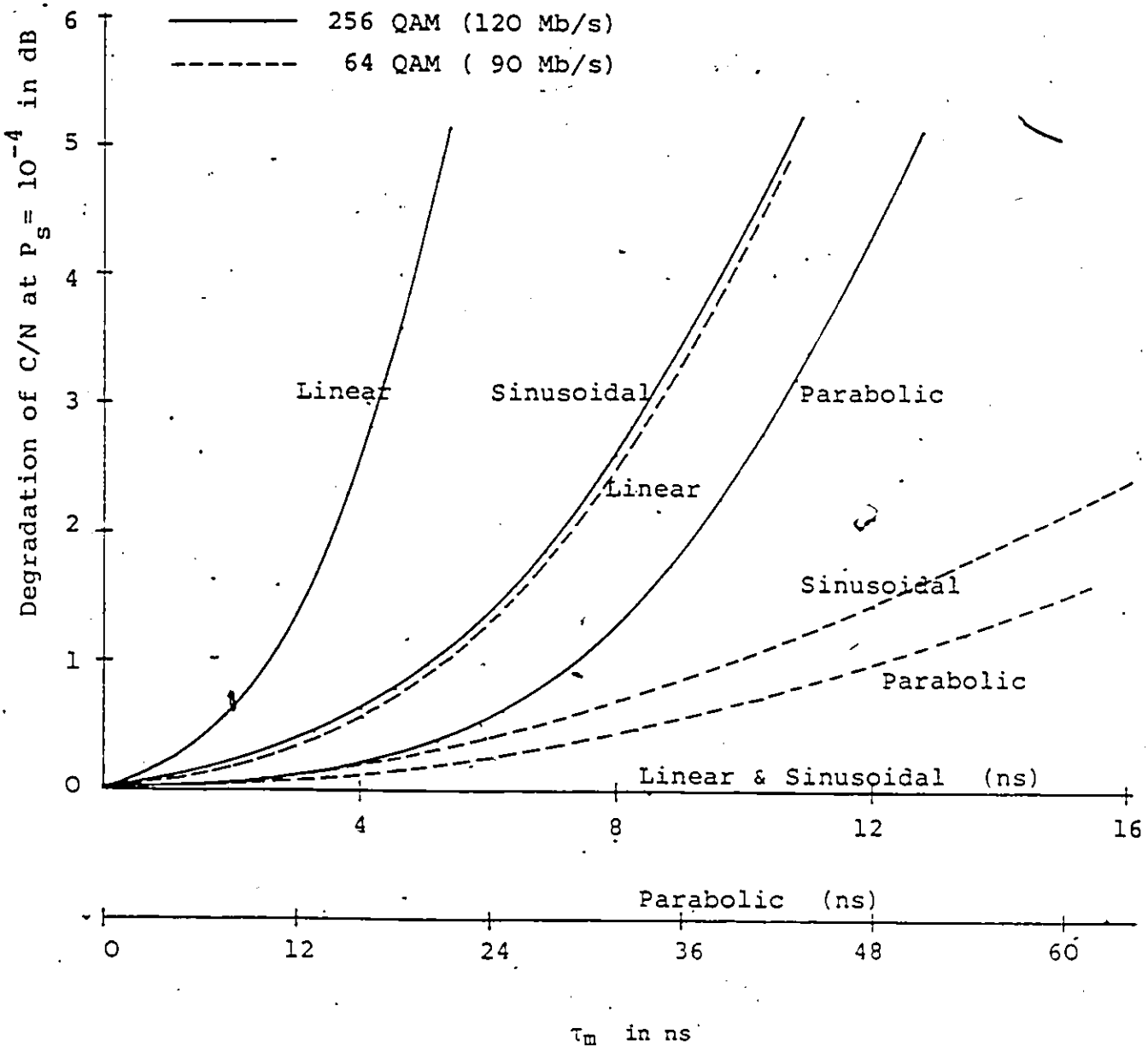


Fig.5.6 Degradation of C/N versus τ_m for linear, parabolic, and sinusoidal group delay distortions for 256-QAM and 64-QAM with a symbol rate of 15 MBaud and $\alpha = 0.4$ raised-cosine filters, where

$\tau_m = L_D(2f_{BW})$ ns (linear)

$= P_D(f_{BW})^2$ ns (parabolic)

$= S_D$ ns (sinusoidal)

$f_{BW} = (1+\alpha)f_N = 1.4f_N = 10.5$ MHz

value of maximum group delay τ_m in the filter bandwidth, linear group delay distortion causes the most severe degradation to the system's performance as compared to parabolic or sinusoidal group delay distortions [8].

5.3 AMPLITUDE DISTORTIONS

For the simulation of the amplitude distortion, we assume the group delay distortion is equalized, i.e., D(f) is equal to a constant. Three different characteristics for the amplitude distortion are defined as follows.

$$A(f) = \begin{cases} L_A f & \text{for linear amplitude distortion,} \\ P_A f^2 & \text{for parabolic amplitude distortion,} \\ S_A \sin(2\pi K f/2f_{BW}) & \text{for sinusoidal amplitude distortion.} \end{cases} \quad (5.9)$$

The symbol error rate Ps versus C/N is computed. A typical result is shown in Fig.5.7. Degradations of C/N for 256-QAM and 64-QAM with 15 MBaud and α = 0.4, versus maximum amplitude distortion, A_m, in the filter bandwidth, are plotted in Fig.5.8, where

$$A_m = \begin{cases} L_A(2f_{BW}) & \text{for linear amplitude distortion,} \\ P_A(f_{BW})^2 & \text{for parabolic amplitude distortion,} \\ S_A & \text{for sinusoidal amplitude distortion.} \end{cases} \quad (5.10)$$

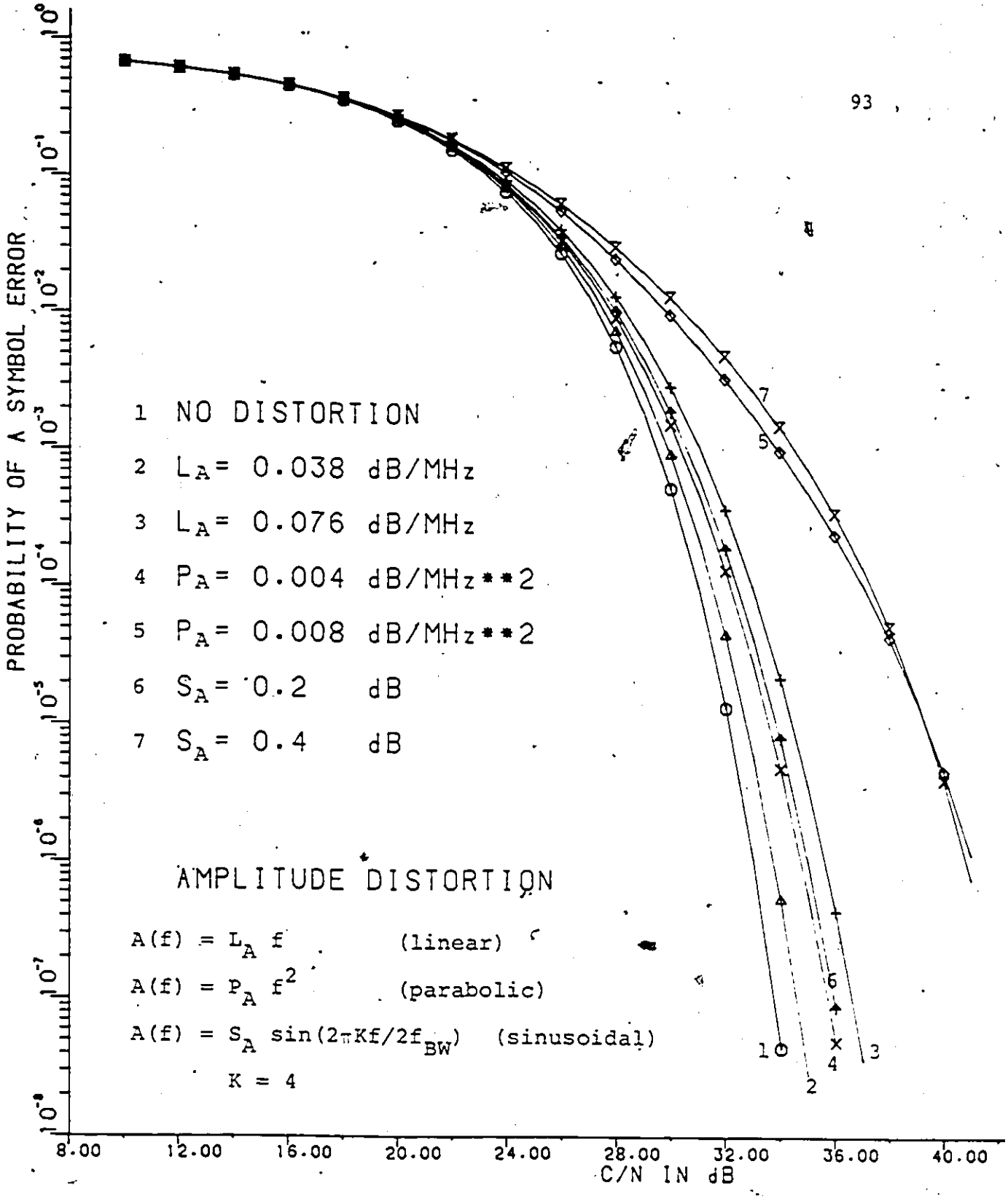


Fig.5.7 P_s versus C/N for linear, parabolic, and sinusoidal amplitude distortions for 256-QAM with a bit rate of 120 Mb/s, i.e., 15 MBaud and α = 0.4.

Noise is defined in the double-sided Nyquist bandwidth, i.e., the equivalent noise bandwidth is 15 MHz.

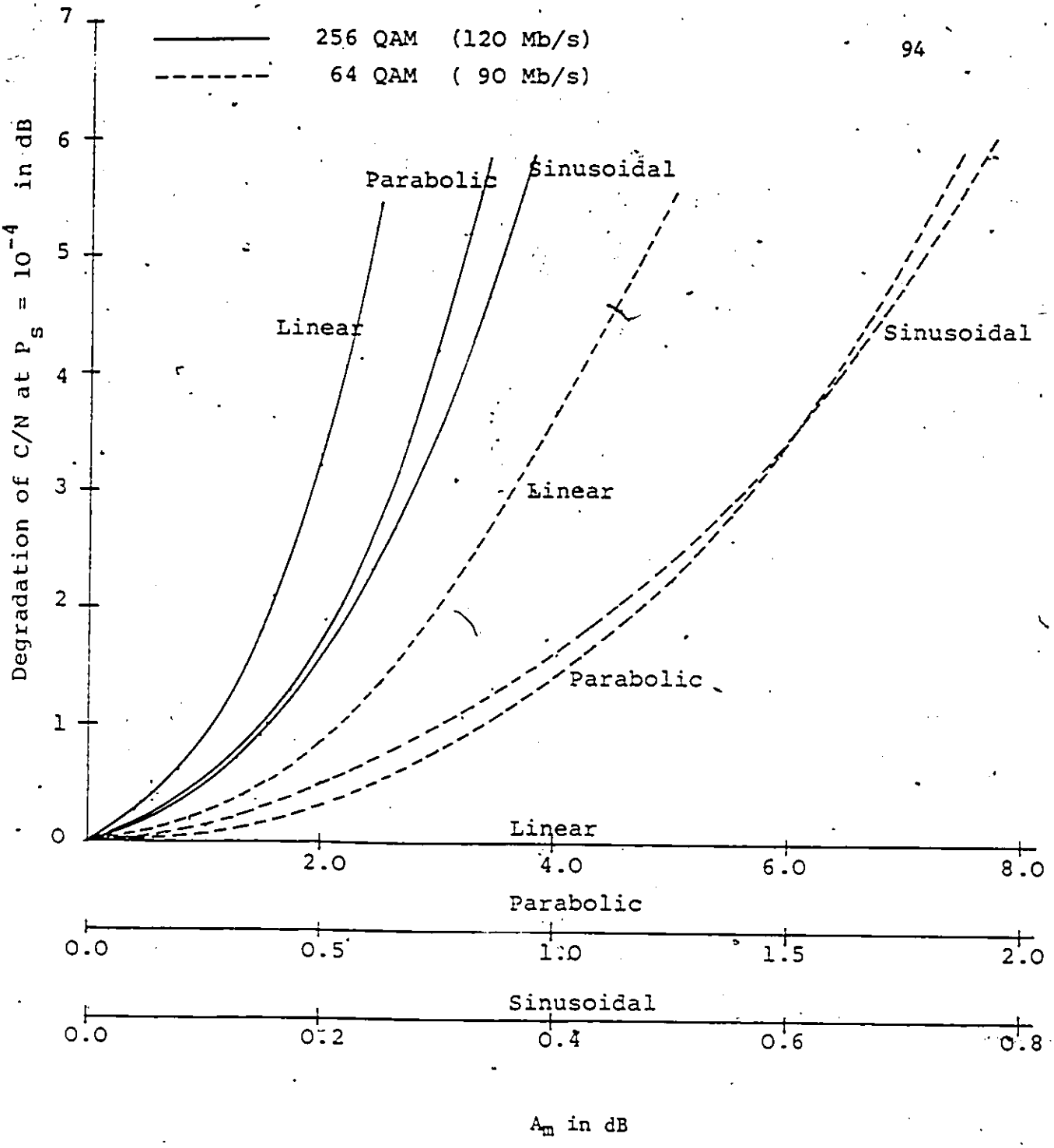


Fig.5.8 Degradation of C/N versus A_m for linear, parabolic, and sinusoidal amplitude distortions for 256-QAM and 64-QAM with a symbol rate of 15 MBaud and $\alpha = 0.4$ raised-cosine filters, where

$A_m = L_A(2f_{BW})$ dB (linear)

$= P_A(f_{BW})^2$ dB (parabolic)

$= S_A$ dB (sinusoidal)

$f_{BW} = (1+\alpha)f_N = 1.4f_N = 10.5$ MHz

We note that for a given value of maximum amplitude distortion, A_m, linear amplitude distortion causes the least degradation, followed in order of increasing degradation by parabolic and sinusoidal amplitude distortions.

5.4 APPLYING THE SIMULATED RESULTS TO SYSTEMS WITH DIFFERENT BIT RATES

After presenting the simulated results for a 120 Mb/s system, we now want to show how these results can be applied to systems with different bit rates. This is important since in a number of applications where we might want to know the performance of a system in distorted channels which uses the same α = 0.4 256-QAM modulation format but has another bit rate. A simple scaling procedure to be described in this section will allow us to efficiently obtain the corresponding results without repeating similar computer simulation procedures, thereby saving a lot of computing time and efforts.

The key point lies in the fact that, in our computer simulation, all of the filtering processes are exactly related to the bit rate of a system and the modulation format. That is, assuming a fixed modulation format, once the bite is specified, the characteristics of filters are theoretically well defined. Consider two systems with bit rates f_{b1} and f_{b2}, suffering linear group delay distortions which are characterized by two parameters L_{D1} and L_{D2}, respectively. Let us define $\hat{\tau}_m$ to be the normalized maximum group delay with respect to symbol interval T_s, i.e.,

$$\hat{\tau}_m = \tau_m / T_s = \tau_m \cdot f_s \tag{5.11}$$

where

τ_m is the maximum group delay in the filter bandwidth, as defined in (5.8),

T_s is the symbol interval,

$T_s = 1/f_s = 8T_b = 8/f_b$ for 256-QAM.

It is obvious that, for systems having different bit rates, the group delay distortions which have the same characteristic and the same $\hat{\tau}_m$ will cause the same performance degradation. Thus for two linear group delay distortions causing the same degradation to the respective system a simple relation can easily be obtained from (5.11), i.e.,

$$\tau_{m1} f_{b1} = \tau_{m2} f_{b2} \tag{5.12}$$

or equivalently,

$$L_{D1} f_{b1}^2 = L_{D2} f_{b2}^2 \tag{5.13}$$

where (5.7) and (5.8) have been used. The following example will illustrate how (5.12) and (5.13) and the previous results for a 120 Mb/s system can be employed to predict the corresponding results for a 90 Mb/s system.

From Fig. 5.6, it is noted that in the case of linear group delay distortions, a τ_m of 2.5 ns causes a 1 dB degradation in C/N for a 120 Mb/s system. Substituting f_{b1} = 120 Mb/s, f_{b2} = 90 Mb/s, and τ_{m1} = 2.5 ns into (5.12),

$$\tau_{m2} = 2.5 \cdot \frac{120}{90} = 3.1 \text{ ns}$$

This indicates that a τ_m of 3.1 ns will cause the same 1 dB degradation in C/N for a 90 Mb/s system. Furthermore, substituting L_{D1} = 0.095 ns/MHz, f_{b1} = 120 Mb/s, f_{b2} = 90 Mb/s, we find

$$L_{D2} = 0.095 \cdot \left(\frac{120}{90}\right)^2 = 0.126 \text{ ns/MHz}$$

This means that the P_s curve corresponding to L_D = 0.095 ns/MHz for a 120 Mb/s system also represents the P_e performance corresponding to L_D = 0.126 ns/MHz for a 90 Mb/s system.

On the other hand, consider the amplitude distortions case. It is obvious that the amplitude distortions which have the same characteristic and the same maximum amplitude distortion in the filter bandwidth will cause the same performance degradation to systems with different bit rates. Thus, for two linear amplitude distortions with parameters L_{A1} and L_{A2} which cause the same degradation to two systems with bit rates f_{b1} and f_{b2}, a simple relation can be drawn from (5.10) and the above argument as follows.

$$L_{A1} f_{b1} = L_{A2} f_{b2} \tag{5.14}$$

As a further illustration, the P_s curve corresponding to L_A = 0.038 dB/MHz for a 120 Mb/s represents also the P_e performance corresponding to L_A = 0.038 x 120/90 = 0.0507 dB/MHz for a 90 Mb/s system. It is now clear that the results simulated for a 120 Mb/s system can easily be applied to systems with different bit rates. The equations which relate other distortion parameters to bit rates can be derived in the same way and are summarized in Table 5.1.

TABLE 5.1 RELATIONSHIP BETWEEN DISTORTION PARAMETERS AND BIT RATES FOR TWO SYSTEMS HAVING THE SAME MODULATION FORMAT AND PERFORMANCE BUT DIFFERENT BIT RATES.

	Group delay diatortions	Amplitude distortions
Linear	$L_{D1} f_{b1}^2 = L_{D2} f_{b2}^2$	$L_{A1} f_{b1} = L_{A2} f_{b2}$
Parabolic	$P_{D1} f_{b1}^3 = P_{D2} f_{b2}^3$	$P_{A1} f_{b1}^2 = P_{A2} f_{b2}^2$
Sinusoidal	$S_{D1} f_{b1} = S_{D2} f_{b2}$	$S_{A1} = S_{A2}$

Note: L_{D1} and L_{D2} are the parameters of linear group delay distortions which cause the same performance to systems with bit rates f_{b1} and f_{b2}, respectively. Other parameters and notations follow similarly.

5.5 CONCLUSIONS

The effects of group delay and amplitude distortions on the error performance of a 256-QAM modem have been studied. Computer simulation results indicate that linear group delay distortion is the most critical group delay parameter. The simulated results could be useful for modem researchers and designers since it could indicate the tolerable imperfection in filters implementation which would not degrade system performance to below a specified value. Also presented is a simple scaling procedure which enables us to apply the simulated results for a bit rate of 120 Mb/s to any other bit rates without going through similar computation procedures.

Chapter VI

COMPARISON OF 256-QAM AND IMPROVED EFFICIENCY 225-QPRS IN DISTORTED CHANNELS

In Chapter 5 and [36], the performance of a 256-QAM modem in distorted channels was evaluated. An ideal 225-QPRS achieves an efficiency of 6 b/s/Hz, which is equivalent to that of 256-QAM with a roll-off factor $\alpha = 0.33$. However, at $P_e = 10^{-9}$ a 4 dB increase in C/N immunity can be obtained by using 225-QPRS instead of 256-QAM to achieve 6 b/s/Hz. Furthermore, an improved efficiency class-I 225-QPRS using our new Model-B has been described in Chapter 2. Therefore, it is important to study the performance of improved efficiency 225-QPRS in distorted channels.

In this chapter, we shall compare the performances of 256-QAM and improved efficiency 225-QPRS in distorted channels [39]. The comparison will be based on the same spectral efficiency of 6.3 b/s/Hz since the required C/N (dB) at $P_e = 10^{-9}$ is practically the same for these two modulation schemes to achieve this efficiency. The effects of group delay and amplitude distortions on the performance of these two modulation schemes will be evaluated and compared. The results could indicate the sensitivity of the two modulation schemes to different imperfections.

6.1 BACKGROUND FOR THE COMPARISON OF 256-QAM AND 225-QPRS

Fig.6.1 compares some of the results of class-I 225-QPRS using our new Model-B in Chapter 2 with the performance of 256-QAM. It is seen that at $P_e=10^{-9}$ the required C/N is about 35 dB for both 225-QPRS with a 5% increase above the Nyquist rate and 256-QAM. The spectral efficiency of 225-QPRS with a 5% increase above the Nyquist rate is 6.3 b/s/Hz, which can be achieved by 256-QAM with $\alpha = 0.27$. As mentioned in Chapter 1, this high spectral efficiency is required in many applications such as T1/SG. Therefore it is of interest to compare the sensitivity of these two modulation schemes to channel distortions. The comparison will be based on the same transmission rate in the same bandwidth to achieve 6.3 b/s/Hz. In the simulation, a bit rate of 120 Mb/s is used. However, following the scaling procedure described in Chapter 5, the results can be applied for another bit rate.

6.2 COMPUTER SIMULATION DESCRIPTIONS

The simulation is carried out entirely in the complex baseband form for a 120 Mb/s system i.e., with a symbol rate of 15 MBaud for 256-QAM and 20 MBaud for 225-QPRS, respectively. For 256-QAM, the transmit filter consists of an analog x/sin(x) shaped amplitude equalizer and an ideal square-root of $\alpha = 0.27$ raised-cosine filter, i.e., with a spectral efficiency of 6.3 b/s/Hz. The receive filter is an ideal square root of $\alpha = 0.27$ raised-cosine filter.

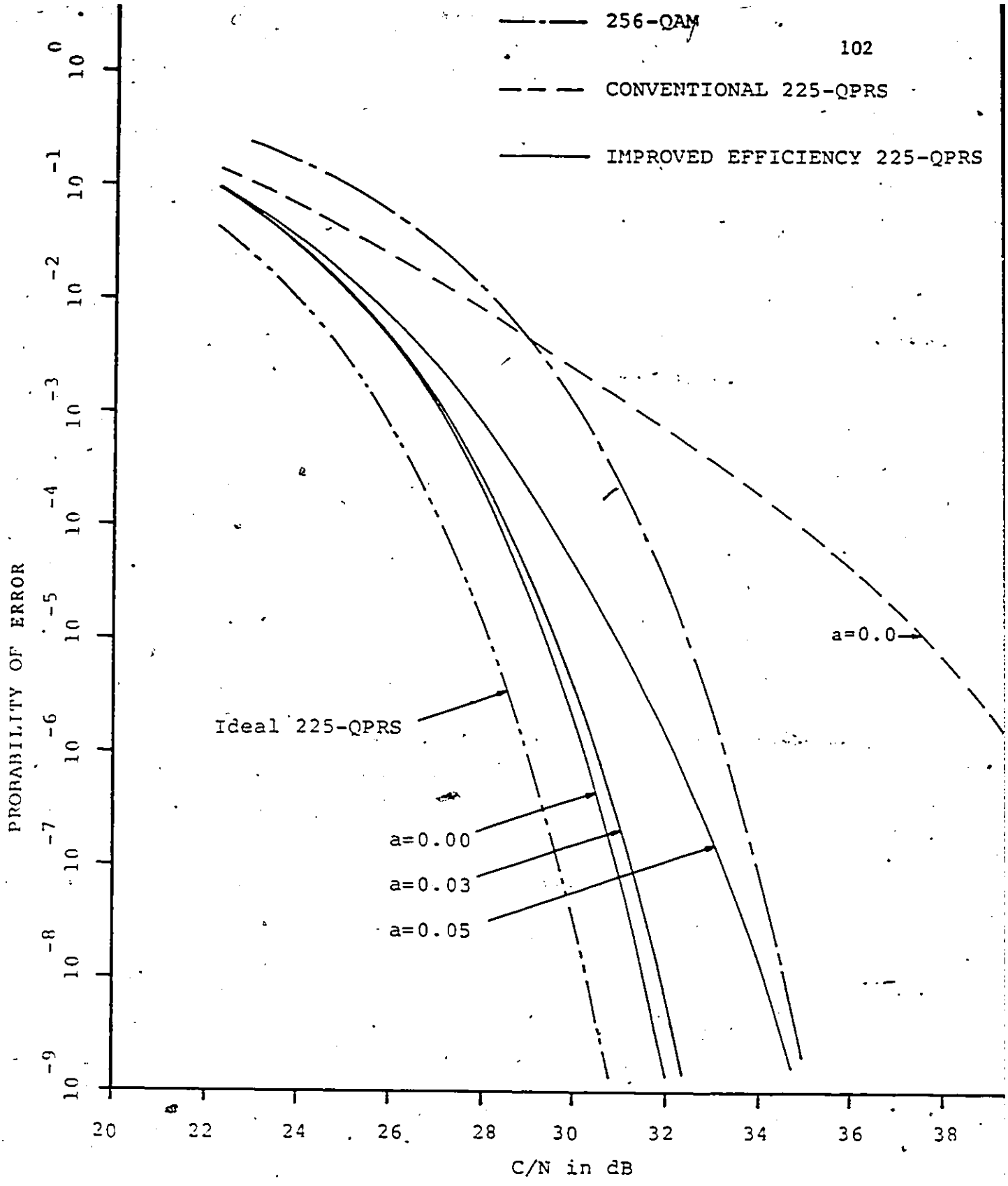


Fig.6.1 Probability of error of 256-QAM, conventional 225-QPRS, and improved efficiency 225-QPRS versus C/N.

For the improved efficiency class-I 225-QPRS, the transmitter consists of an analog x/sin(x) shaped amplitude equalizer, the duobinary filter and an ideal brick-wall filter with a cutoff frequency at $f_N/1.05$, where f_N = 10 MHz is the Nyquist frequency of the signal. Thus the spectral efficiency of this model is also 6.3 b/s/Hz. The receive filter simply bandlimits the channel noise.

We consider the effects of group delay and amplitude distortions with linear, parabolic, or sinusoidal characteristics on these two modulation techniques. The demodulated in-phase and quadrature signals are sampled and passed to the threshold comparators in the receiver. For a specified power of white Gaussian noise at the threshold detector input, the error probability of the i-th symbol with respect to the in-phase channel is calculated as follows:

$$P_{si}^{I} = \begin{cases} \tfrac{1}{2}\,\mathrm{erfc}(|\bar{S}_i - \mathrm{THR1}_i|/\sqrt{2}\sigma) & \text{for } I_i = \pm 15 \text{ (256-QAM)} \\ & \quad = \pm 14 \text{ (225-QPRS)} \\ \tfrac{1}{2}\,\mathrm{erfc}(|\bar{S}_i - \mathrm{THR1}_i|/\sqrt{2}\sigma) + \tfrac{1}{2}\,\mathrm{erfc}(|\mathrm{THR2}_i - \bar{S}_i|/\sqrt{2}\sigma) & \\ \qquad \text{for } I_i = \pm 1, \pm 3, \ldots, \pm 13 \text{ (256-QAM)} & \\ \qquad = 0, \pm 2, \ldots, \pm 12 \text{ (225-QPRS)} & \end{cases} \tag{6.1}$$

where

I_i, $\bar{S}_i$, $\mathrm{THR1}_i$, $\mathrm{THR2}_i$, $\mathrm{erfc}(x)$, and σ

are defined as in (5.2).

The error probability P_{si}^{Q} with respect to the quadrature channel is obtained similarly from (6.1), and for a sequence of N symbols in each channel, the average symbol error P_s is calculated as (5.3).

We assume that the number of samples per symbol is 32 and the number of symbols is 2048 (which corresponds to a PRBS sequence

length of 16,348 bits). We average the probability of error over many symbol patterns. To confirm that the simulated results are meaningful, we investigated the uncertainty (which might be caused by an insufficient length of symbols) by increasing the simulated symbol length significantly. We found that the performance practically converges to that of a 2048 symbols length when the simulated length is longer than 2048 symbols.

6.3 256-QAM AND IMPROVED EFFICIENCY 225-QPRS WITH GROUP DELAY AND AMPLITUDE DISTORTIONS

We insert between the transmit and receive filters a linear filter with the equivalent baseband transfer function C(f) expressed as (5.4)-(5.6). First, we assume that there is no amplitude distortion, i.e., A(f) = 1. The group delay distortions are defined as (5.7). The results are shown in Figs.6.2-6.4 for linear, parabolic and sinusoidal group delay distortions, respectively. The C/N degradation at $P_e = 10^{-4}$ of these modulation techniques for group delay distortions are compared in Table 6.1. It is found that 256-QAM is less sensitive to parabolic and sinusoidal group delay distortions than 225-QPRS. Next, we assume that the group delay distortion is equalized, i.e., D(f) = k (k is a constant). The amplitude distortions are defined as (5.9).

The results are illustrated in Figs.6.5-6.7 for linear, parabolic and sinusoidal amplitude distortions, respectively. The C/N degradation at $P_e = 10^{-4}$ of these modulation techniques for amplitude distortions are compared in Table 6.1. It is found that 225-QPRS is less sensitive to parabolic and sinusoidal amplitude distortions.

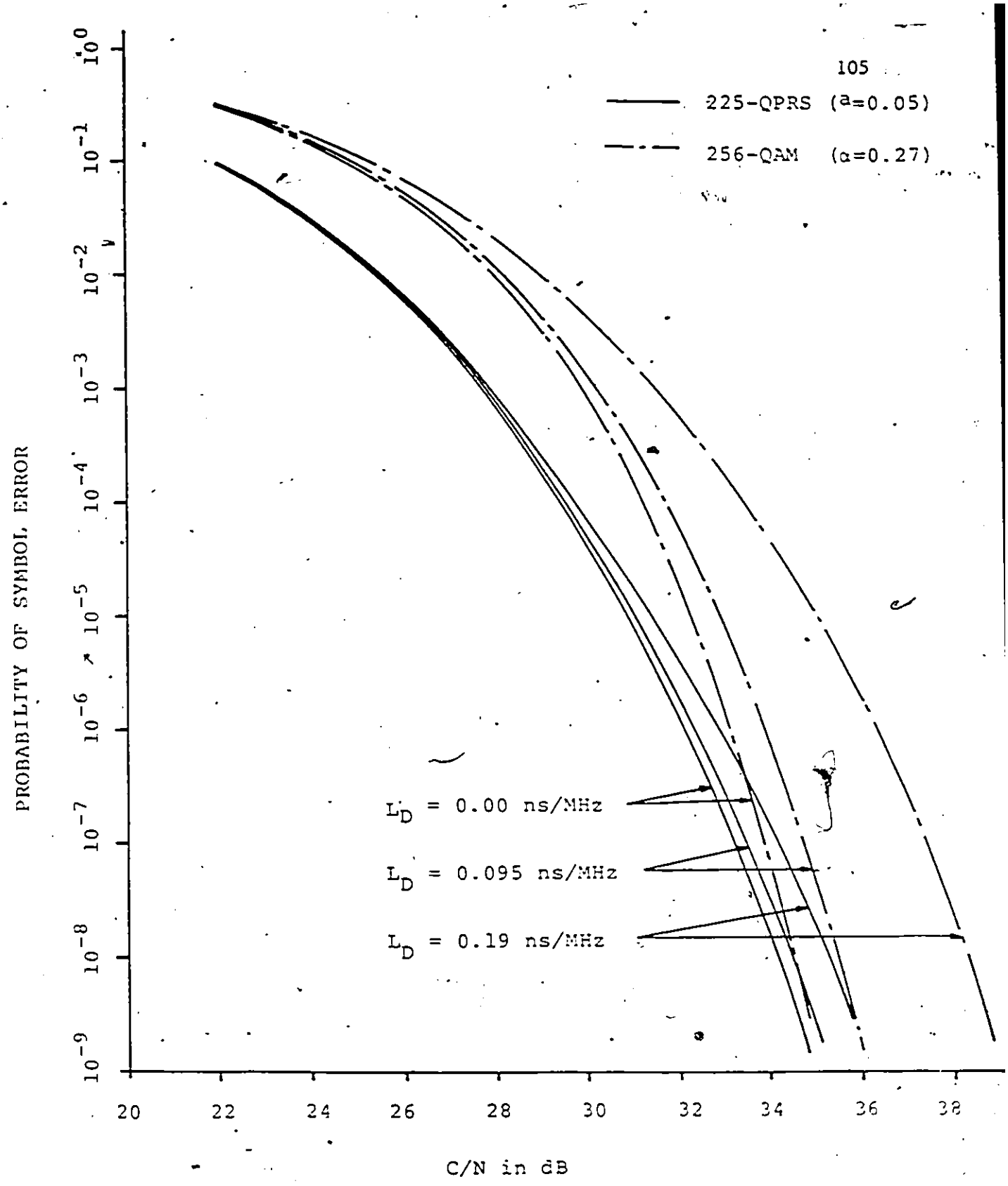


Fig.6.2 Probability of error of 225-QPRS above the Nyquist rate and 256-QAM for linear group delay distortions versus C/N.

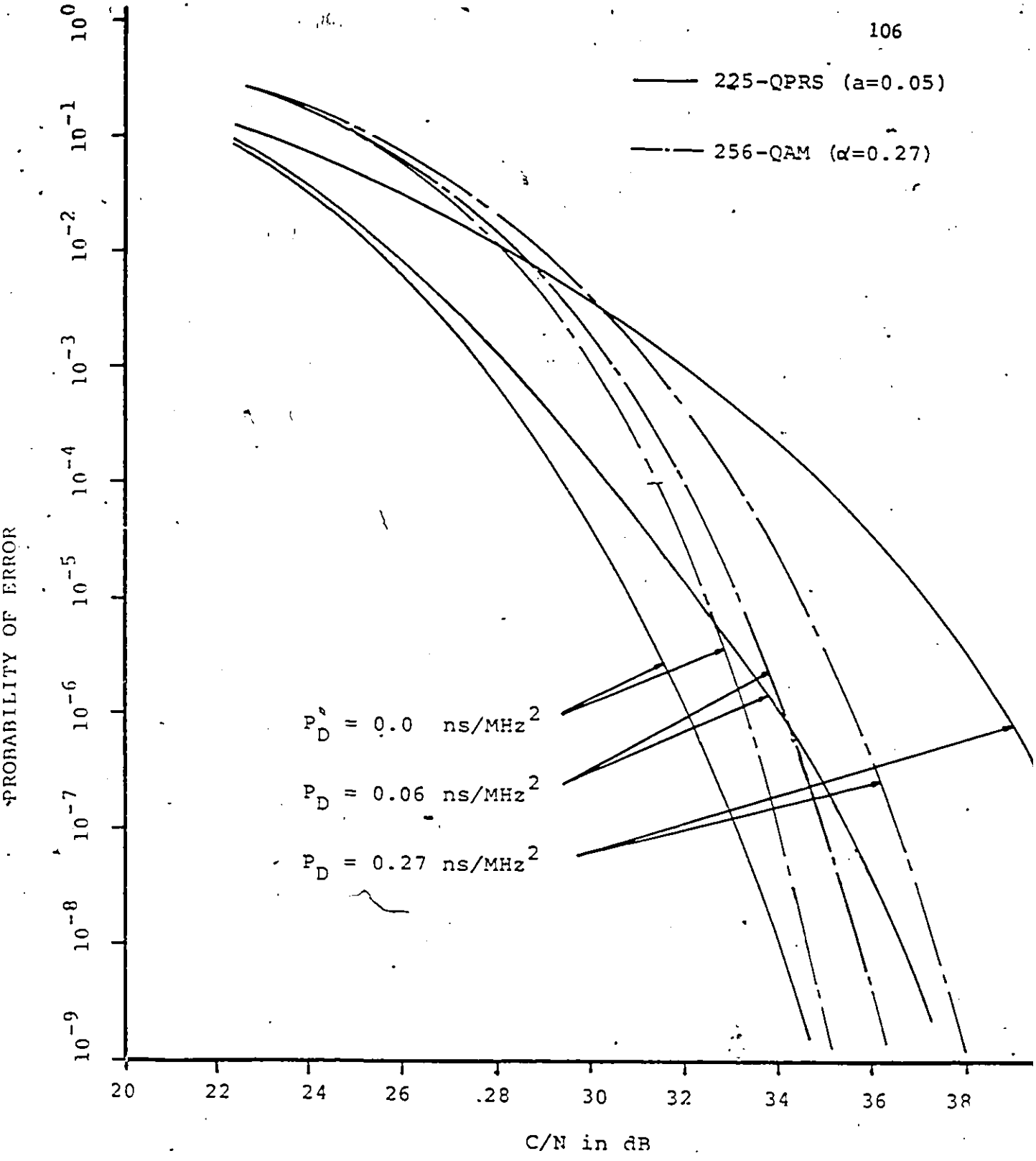


Fig.6.3 Probability of error of 225-QPRS above the Nyquist rate and 256-QAM for parabolic group delay distortions versus C/N.

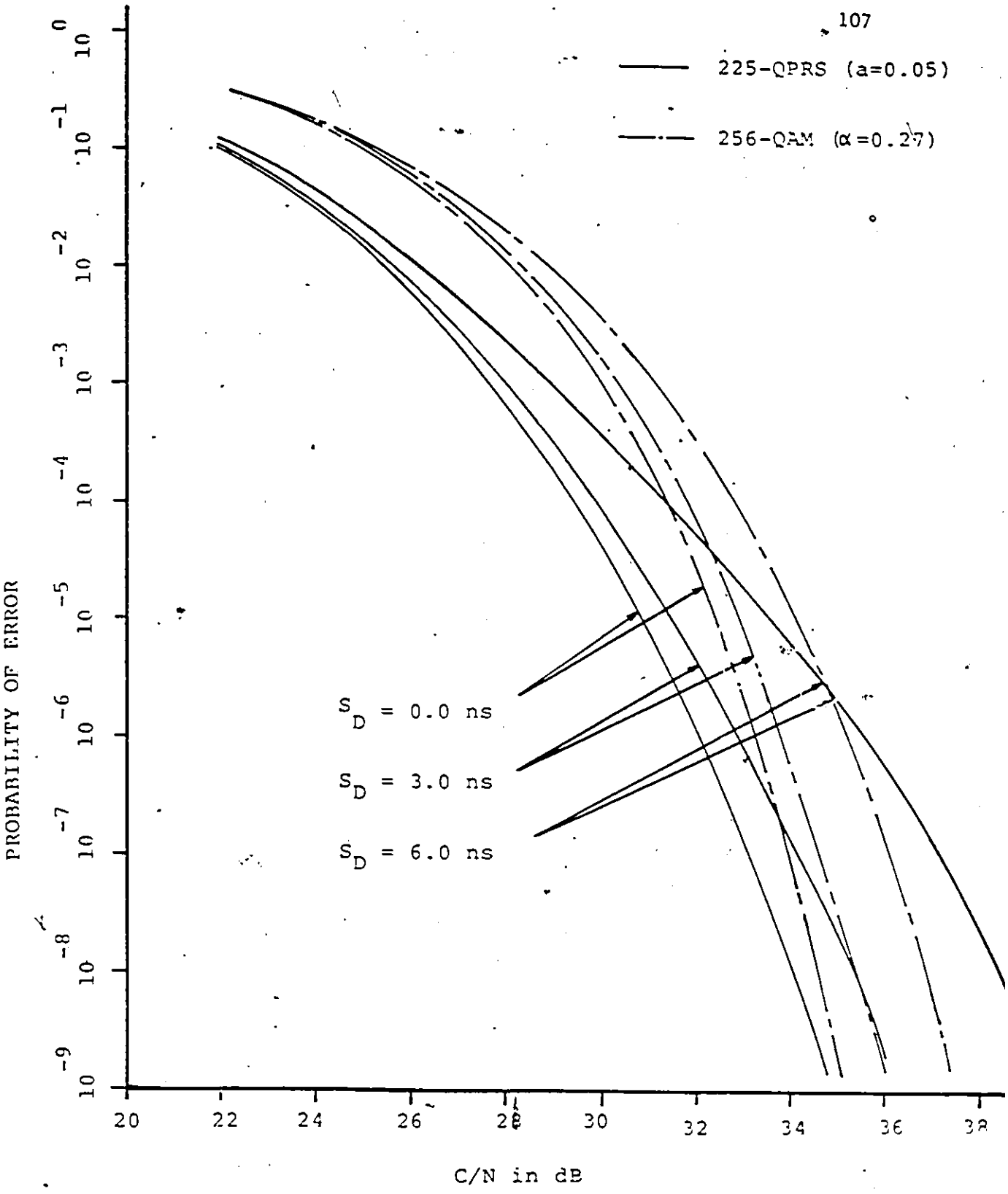


Fig.6.4 Probability of error of 225-QPRS above the Nyquist rate and 256-QAM for sinusoidal group delay distortions versus C/N.

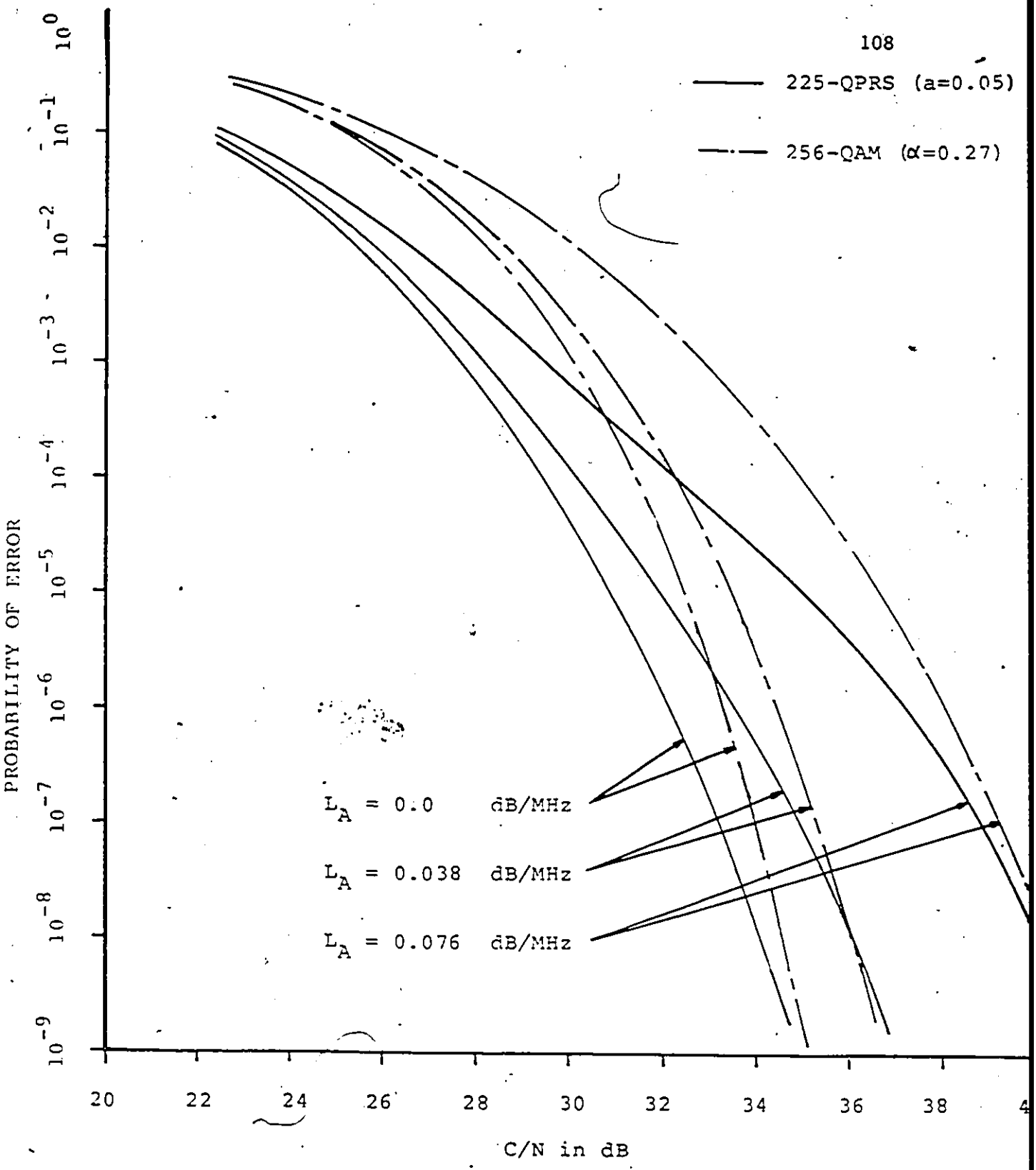


Fig.6.5 Probability of error of 225-QPRS above the Nyquist rate and 256-QAM for linear amplitude distortions versus C/N.

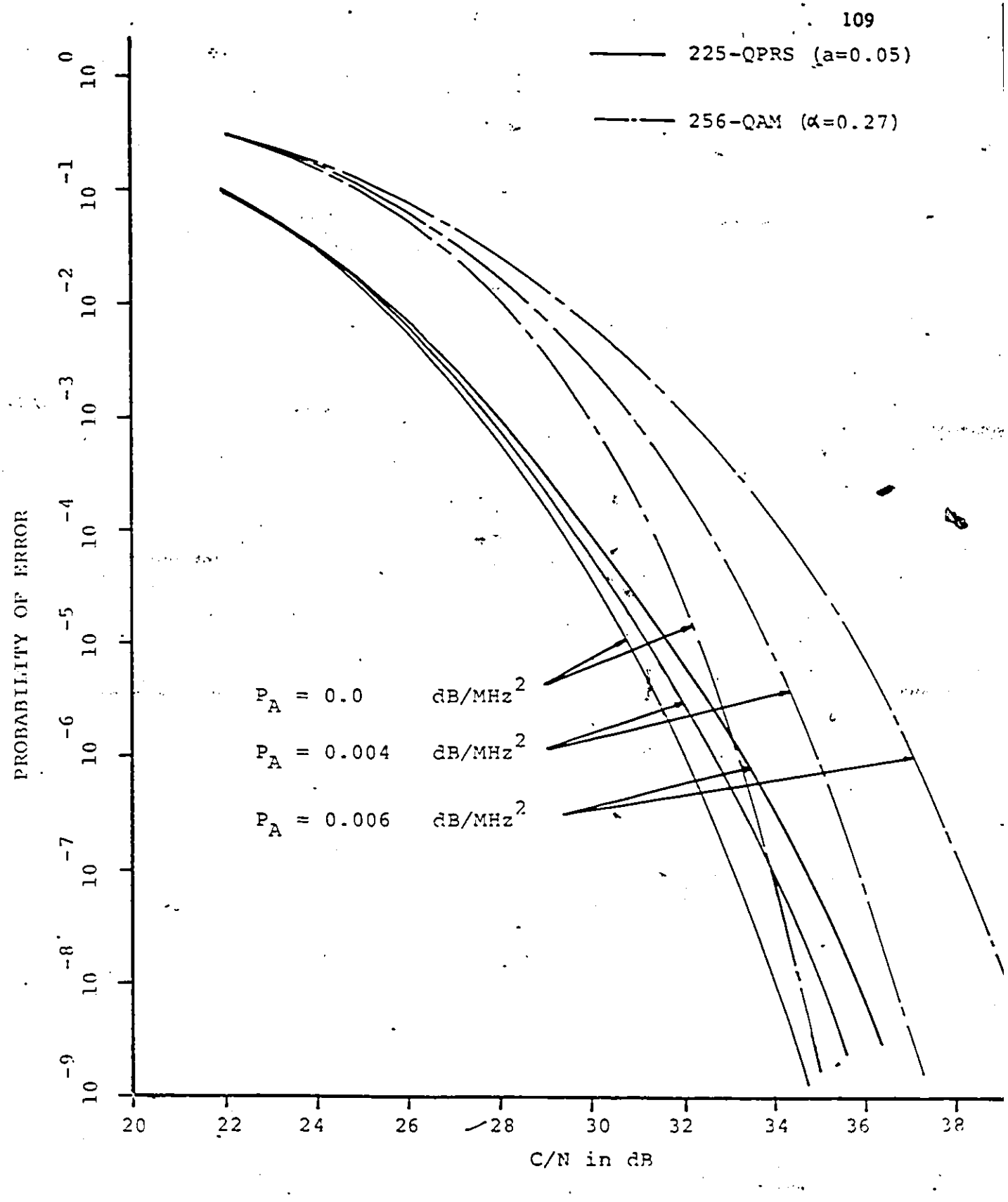


Fig.6.6 Probability of error of 225-QPRS above the Nyquist rate and 256-QAM for parabolic amplitude distortions versus C/N.

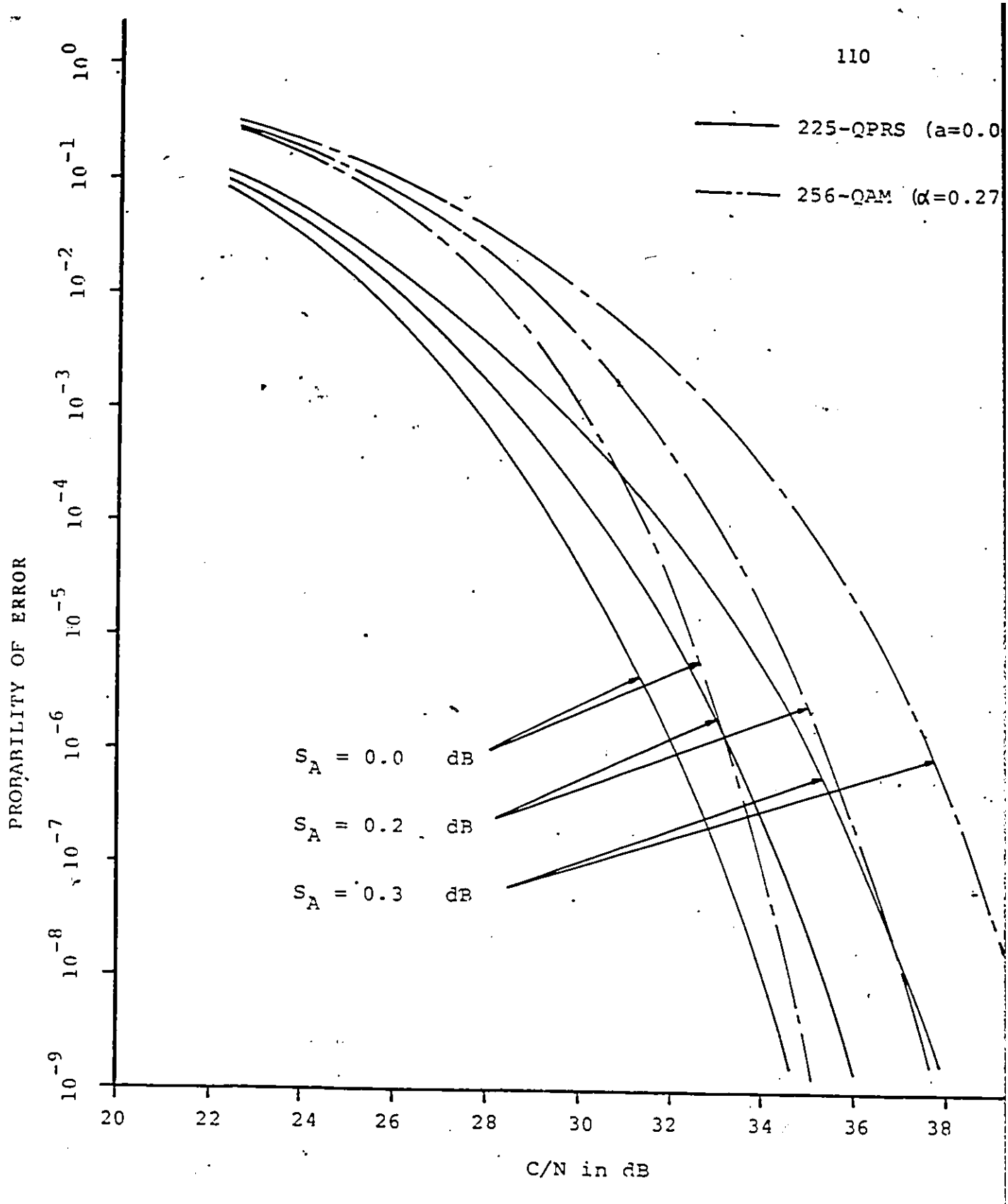


Fig.6.7 Probability of error of 225-QPRS above the Nyquist rate and 256-QAM for sinusoidal amplitude distortions versus C/N.

Group delay or amplitude distortions		C/N degradation at $P_e=10^{-4}$ (dB)	
		225-QPRS (a=0.05)	256-QAM (α=0.27)
linear group delay	L_D = 0.095 ns/MHz	0.1	0.6
	L_D = 0.19 ns/MHz	0.4	2.4
parabolic group delay	P_D = 0.06 ns/MHz^2	0.9	0.7
	P_D = 0.27 ns/MHz^2	5.5	1.8
sinusoidal group delay	S_D = 3.0 ns	0.5	0.5
	S_D = 6.0 ns	1.8	1.5
linear amplitude distortion	L_A = 0.038 dB/MHz	0.7	0.9
	L_A = 0.076 dB/MHz	2.8	3.6
parabolic amplitude distortion	P_A = 0.004 dB/MHz^2	0.3	1.3
	P_A = 0.006 dB/MHz^2	0.6	3.0
sinusoidal amplitude distortion	S_A = 0.2 dB	1.0	1.7
	S_A = 0.3 dB	2.3	3.6

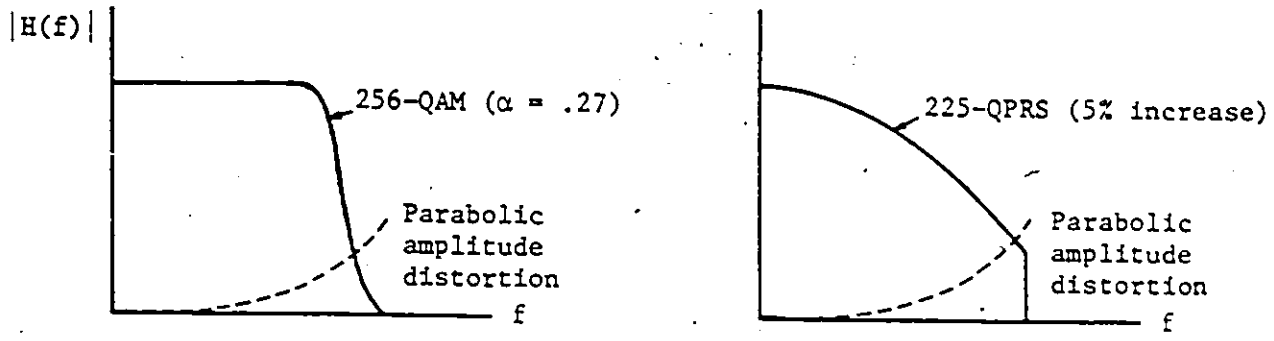
TABLE 6.1 C/N DEGRADATION OF 225-QPRS ABOVE THE NYQUIST RATE AND 256-QAM FOR GROUP DELAY AND AMPLITUDE DISTORTIONS AT $P_e = 10^{-4}$.

It is observed from the Figs.6.4-6.6 that, while 225-QPRS is worse than 256-QAM for low enough error probabilities, the situation is reverse for high P_e values. The interpretation is the following. When C/N is small, Gaussian noise dominates the performance and ISI is negligible. Thus, the performance of 225-QPRS would be better than that of 256-QAM. On the other hand, when C/N is higher, ISI is more dominant than Gaussian noise. In our simulation, the ISI is due only to the channel distortions in the 256-QAM case, whereas in 225-QPRS, the "undesired" ISI exists due to not only the channel distortions but also the higher transmission rate than the Nyquist rate. Thus when C/N is high, the performance of 225-QPRS might be worse than that of 256-QAM in distorted channels with some specific group delay and amplitude distortions.

6.4 PHYSICAL INTERPRETATION OF THE RESULTS

Having presented the various simulated results, it is of importance to seek the physical reason why one of the two modulation schemes is more sensitive to certain distortions than the other. To begin with, let us consider amplitude distortions.

As illustrated in Fig.6.8 for a parabolic amplitude distortion, we note that, in the case of 225-QPRS, most of the energy is concentrated around dc where the distortion is relatively small. On the other hand, for 256-QAM, a significant amount of energy is distributed in the region where the distortion is relatively large. This explains why 225-QPRS is less sensitive to parabolic amplitude distortions than 256-QAM, as already seen from the results in Fig.6.6.



* H(f) is the overall frequency response.

Fig.6.8 Frequency-domain comparison of 256-QAM (α = .27) and 225-QPRS (5% increase) with a parabolic amplitude distortion.

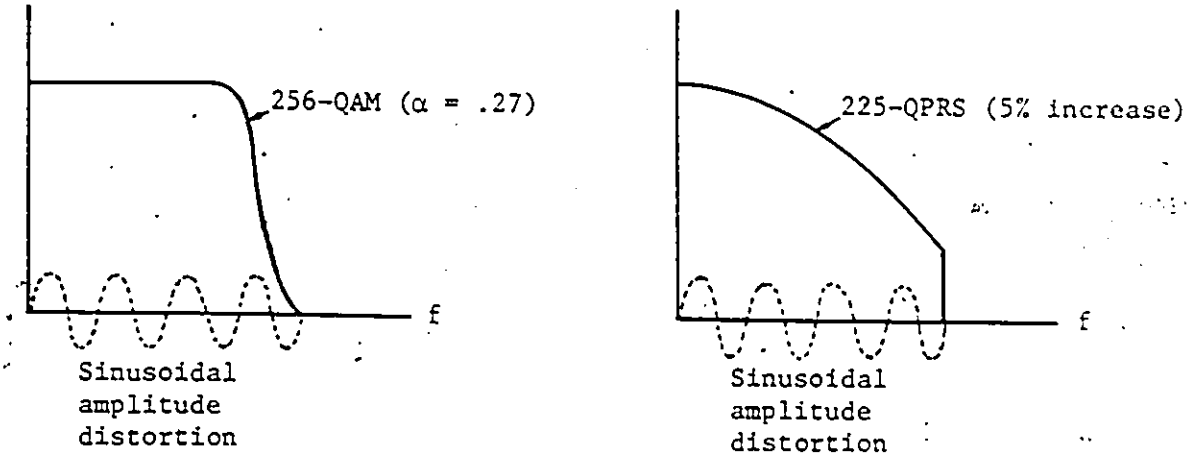


Fig.6.9 Frequency-domain comparison of 256-QAM (α = .27) and 225-QPRS (5% increase) with a sinusoidal amplitude distortion.

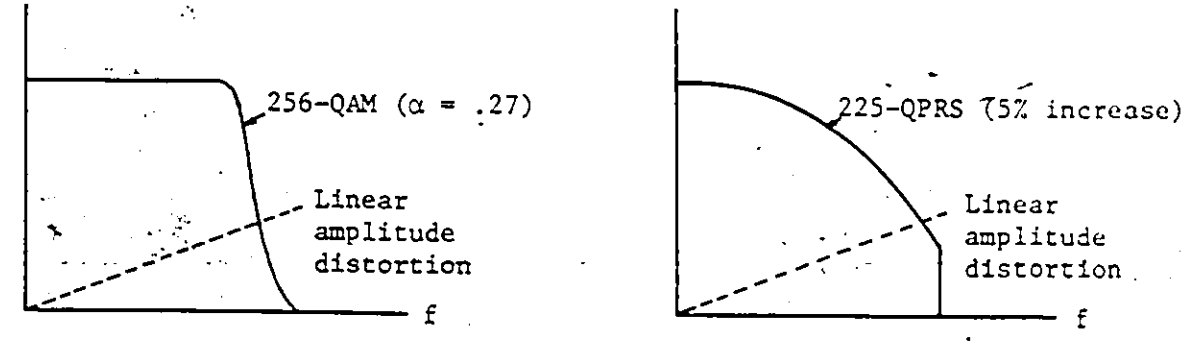


Fig.6.10 Frequency-domain comparison of 256-QAM (α = .27) and 225-QPRS (5% increase) with a linear amplitude distortion.

For sinusoidal amplitude distortions, the frequency-domain comparison is shown in Fig.6.9. In both cases, all the frequency components suffer distortion. However, in the case of 225-QPRS, due to its cosine-shaped energy distribution, the effect of distortion in the average sense will be somewhat less than that on 256-QAM. Thus, we conclude that 225-QPRS is also less sensitive to sinusoidal amplitude distortions than 256-QAM, but the difference may not be as significant as in the parabolic case, as observed from Figs.6.6 and 6.7.

As to linear amplitude distortions, looking at Fig.6.10 for comparison, it is difficult to judge which scheme is more sensitive than the other. In fact, the results presented in Fig.6.5 indicate that 256-QAM and 225-QPRS have more or less the same sensitivity to linear amplitude distortions.

The effects of group delay distortions are more difficult to analyze. However, with efforts, we still can get some insight into it. Ideal filters/channels should have a linear phase characteristic, i.e., a constant group delay. In the case of linear group delay distortions which correspond to having a parabolic phase characteristic, there will be crosstalks between the in-phase (I) and quadrature (Q) channels. 256-QAM being a 16-level signal would have higher introduced crosstalk than 225-QPRS, which is a 15-level signal. Thus 225-QPRS, is less sensitive to linear group delay distortions than 256-QAM, as confirmed by the illustrative results in Fig.6.2.

For parabolic group delay distortions, there is no crosstalk. Thus, it is important to note the critical effect of the group delay at the band

edge where the parabolic group delay increases significantly. And it is there where the energy is truncated in the 225-QPRS case. So 225-QPRS has more energy than 256-QAM at the band edge, with the result of suffering more degradations. This leads to the conclusion that 256-QAM is less sensitive to parabolic group delay distortions, as illustrated in Fig.6.3.

Finally, coming to the case of sinusoidal group delay distortions, it is not fully understood yet. However, the results in Fig.6.4 show that 256-QAM is slightly less sensitive than 225-QPRS.

6.5 CONCLUSIONS

The effects of group delay and amplitude distortions on the performance of 256-QAM and improved efficiency 225-QPRS have been compared, based on the same spectral efficiency of 6.3 b/s/Hz. It is found that 256-QAM is less sensitive to parabolic and sinusoidal group delay distortions than 225-QPRS. And 225-QPRS is less sensitive to linear group delay distortions and parabolic and sinusoidal amplitude distortions.

Chapter VII

ERROR FLOORS IN DIGITAL TRANSMISSION SYSTEMS

In the study of multi-level PRS/QPRS systems operated above the Nyquist rate [22], it was found that error floors exist in these systems. Error floors are known in industry as one of the most severe problems of transmission systems. It is often noted that a practically measured BER curve tends to tail off at high S/N. That is, in some cases due to transmission impairments, a system cannot have a BER lower than a certain limit. The mechanism of error floors is still not well understood.

In this chapter, we introduce a new method to predict error floors caused by operating PRS systems above the Nyquist rate. Our method is, however, more general and could be used also in the study of other transmission systems. As an illustration, the evaluation of error floors for PAM systems with severe distortions will be described.

7.1 ERROR FLOORS IN MULTI-LEVEL PRS/QPRS SYSTEMS OPERATED ABOVE THE NYQUIST RATE

As mentioned in Chapter 2, PRS systems operate at the Nyquist rate by introducing deliberately a "controlled " or "desired" amount of ISI. When operated above the Nyquist rate, in addition to the "desired" ISI, there is, in general, introduced "undesired" ISI no matter which scheme

(the conventional Model-A or our new Model-B) is used. The following discussion concentrates on a noise-free system.

Referring to Figs.2.2 and 2.3 and equations (2.1)-(2.5) for duobinary PRS, the parameter "a" indicates the system's spectral efficiency. For example, a = 0 means transmission at the Nyquist rate and a system with a = 0.05 would mean a 5% increase in spectral efficiency above the Nyquist rate. With a = 0 for the ideal duobinary PRS, the only two nonzero samples are at $t = -T_s/2$ and $T_s/2$, respectively. For m-ary inputs, i.e., the input symbols $\{x_n\}$ consist of the equally likely values of $\pm 1, \pm 3, \ldots, \pm(m-3), \pm(m-1)$, the output y(t) generally has 2m-1 levels if a=0, i.e., at the Nyquist rate.

As one intends to increase the spectral efficiency above the Nyquist rate using Model-A or Model-B, i.e., with a > 0, the nominally zero pulse samples at $t = \pm(2n+1)T_s/2$, n = 1, 2, 3, ..., are no longer zeros. These nonzero samples present "undesired" ISI to the system. The peak eye closure criterion [20],[40] is commonly used to determine the eye opening for a system in the presence of ISI. To apply this criterion, one must first evaluate the peak distortion D_p, which, for the present two models with m-ary inputs, can be expressed as follows:

$$D_p = 2(m-1) \sum_{n=1}^{\infty} \left| h\left(\frac{2n+1}{2} T_s\right) \right| \qquad (7.1)$$

where h(t) represents the overall impulse response of a system. The impulse responses of the conventional Model-A and our new Model-B are given in eqs.(2.3) and (2.5).

The peak distortion represents the undesired ISI generated by an infinite number of interfering symbols under the worst-case combination. Neglecting all the undesired ISI for the moment, it is easy to see the output levels at the optimum sampling instants would be 0, $\pm 2h(T_s/2)$, $\pm 4h(T_s/2)$,..., $\pm 2(m-2)h(T_s/2)$, $\pm 2(m-1)h(T_s/2)$ for m-ary inputs. For example, if $x_0 = -1$, $x_1 = 1$ then $y(T_s/2) = 0$. If $x_0 = x_1 = 1$ then $y(T_s/2) = 2h(T_s/2)$. The decision thresholds can be set at $\pm h(T_s/2)$, $\pm 3h(T_s/2)$,..., $\pm (2m-3)h(T_s/2)$, resulting in a decision distance of $h(T_s/2)$. According to the peak eye closure criterion if $D_p < h(T_s/2)$ the eye opening denoted by ρ, is

$$\rho = 2(h(T_s/2) - D_p) \tag{7.2}$$

And the eye is closed when the peak distortion is greater than the decision distance, i.e.,

$$D_p \geq h(T_s/2) \tag{7.3}$$

Our study [22] found that, for the worst case pulse sequence by using (7.1) and (7.2) to evaluate the eye opening, neither Model-A nor Model-B gives open eyes for m-ary inputs, if $m > 4$ and $a > 0.04$. This is because of inherently higher ISI in multi-level systems, as we note that a magnifying factor of m-1 appears in (7.1). However, this does not rule out the possibility of transmitting m-ary signals using Model-A or Model-B since the probability of occurrence of the worst sequence may be relatively low in some cases.

Instead of peak distortion, we introduce the concept of truncated peak distortion D_t defined by (7.4).

$$D_t = 2(m-1) \sum_{n=1}^{N} \left| h\left(\frac{2n+1}{2} T_s \right) \right| \tag{7.4}$$

The difference between (7.4) and (7.1) is that in (7.4) only a finite number of "undesired" ISI samples are considered.

The reason for limiting the number of interfering symbols to a finite value is that it leads to an error-floor concept which will be described in the following paragraphs. And it is true that the most significant ISI is practically due to finite symbols which are contiguous to the present symbol.

Now summing up the undesired ISI for N previous and N forthcoming symbols under the worst case, there might exist a minimum N, denoted by N_{min}, such that

$$D_t \geq h(T_s/2) \qquad \text{for } N \geq N_{min} \tag{7.5}$$

When (7.3) is met, the eye is closed. We define

$$L = 2(N_{min} + 1) \tag{7.6}$$

to be the shortest length of the worst data sequence which closes the eye. The probability of occurrence of this sequence is

$$E_f = 1/m^L \tag{7.7}$$

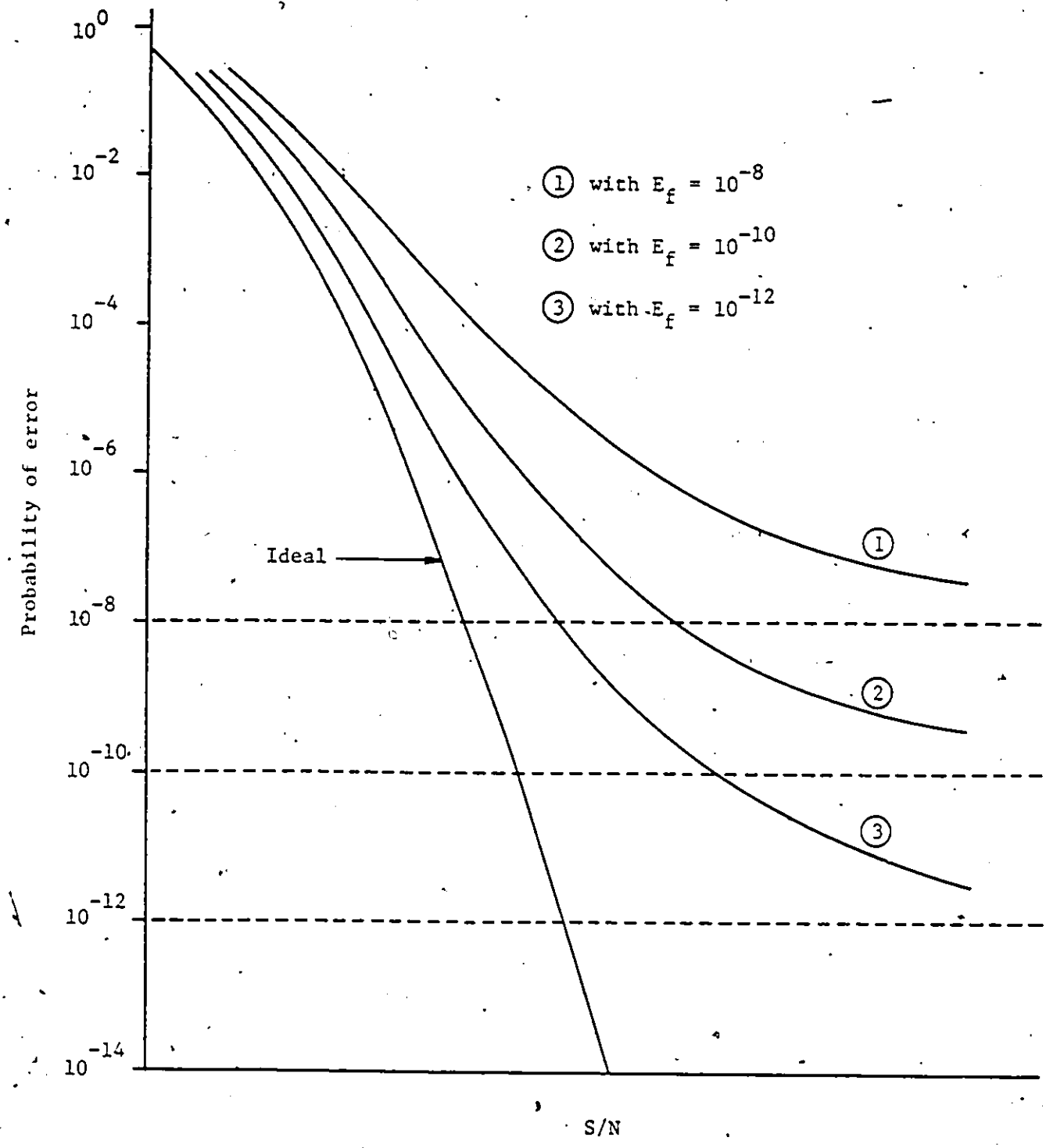


Fig.7.1 Illustrative P_e curves with error floors.

The probability E_f may be regarded as an error floor due to the shortest worst data sequence. This means, even without noise or practically with a very high S/N, the system cannot have a probability of error lower than E_f. To illustrate how error floors affect the P_e curves, a few examples are shown in Fig.7.1. The P_e curves tend to tail off at high S/N. E_f can serve as a means to predict how good a system would be. As a rule of thumb, a system with a lower error floor will be preferred.

The problem of finding L and the corresponding error floor, E_f, with 8-level inputs for Model-A and Model-B was programmed on a computer. The results, based on (7.4), (7.5) and (2.1)-(2.5), are summarized in Table 7.1.

From Table 7.1 we see that the error floors of our new Model-B are much lower than those of the conventional Model-A. For instance, for the same 5% increase above the Nyquist rate, the error floor caused by using our new Model-B is 8.2×10^{-84} whereas it is 6×10^{-8} by using the conventional Model-A. This indicates that our new scheme is better than the conventional method for a practical realization.

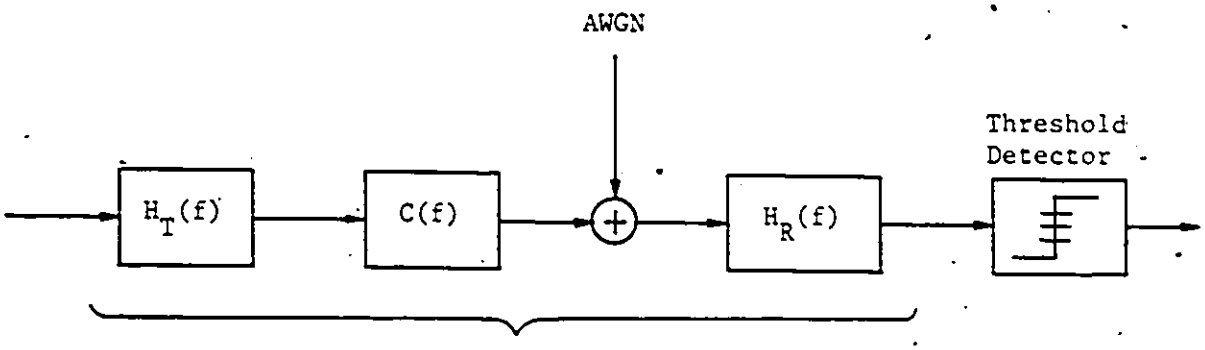
7.2 ERROR FLOORS IN PAM SYSTEMS UNDER SEVERE IMPAIRMENTS

Though the previous discussion was mainly to compare the error floors in PRS systems operated above the Nyquist rate using two different models, the concept and method could be extended to the study of other transmission systems. For instance, consider a PAM system under various impairments, such as channel distortions and hardware im-

TABLE 7.1 COMPARISON OF THE SHORTEST LENGTH OF THE WORST DATA SEQUENCE AND THE CORRESPONDING ERROR FLOORS FOR MODEL-A AND MODEL-B WITH 8-LEVEL INPUTS (15-LEVEL OUTPUTS)

%above Nyquist rate	Model-A (conventional model)		Model-B (new model)	
	L	$E_f = 1/8^L$	L	$E_f = 1/8^L$
3%	22	1.3×10^{-20}	686	$\simeq 0$
4%	12	1.4×10^{-11}	206	$\simeq 0$
5%	8	6×10^{-8}	92	8.2×10^{-84}
6%	6	3.8×10^{-6}	60	6.5×10^{-55}
7%	6	3.8×10^{-6}	34	1.9×10^{-31}
8%	6	3.9×10^{-6}	28	5.1×10^{-26}
9%	6	3.8×10^{-6}	22	1.3×10^{-20}

perfections, etc., which is modelled as Fig.7.2. We assume that the distortion is so severe that the system is regarded as failure by the peak eye closure criterion.



$$H(f) = H_T(f)C(f)H_R(f)$$

$$h(t) = F^{-1}\{H(f)\}$$

Fig.7.2 A PAM system model.

Let h(t) denote the overall impulse response of the system shown in Fig.7.2, which comprises the transmit and receive filters and other impairments modelled as C(f). Since h(t) may not be always symmetric, it is necessary to modify the previous approach a little bit as the following. The input symbols are assumed to be equiprobable numbers ±1, ±3,..., ±(M-1). Thus, in the absence of ISI and noise, the signal levels before the threshold detector are ± h(0), ±3h(0),..., ±(M-1)h(0), resulting in a decision distance of h(0). Under the worst case, the ISI components are

$$(M - 1)\,|h(nT_s)|, \quad n = -\infty \text{ to } \infty, \; n \neq 0 \tag{7.8}$$

Let us form a sequence $\{S_i\}_{i=1,2,\ldots}$ whose items are obtained by permuting all the ISI samples, shown in (7.8), in a non-decreasing order, i.e.,

$$S_i \geq S_{i+1} \quad \text{for } i = 1, 2, \ldots \tag{7.9}$$

Define a distortion series D_N as follows.

$$D_N = \sum_{i=1}^{N} S_i \tag{7.10}$$

Note that the sequence $\{D_N\}_{N=1,2,\ldots}$ is also non-decreasing and

$$\lim_{N\to\infty} D_N = D_p = (M-1) \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} |h(nT_s)| \tag{7.11}$$

In the event of severe distortions, there may exist a smallest integer number N_1 such that

$$D_{N_1} \geq h(0) \tag{7.12}$$

That is, N_1 is the smallest number of interfering symbols which may close the eye. The probability of occurrence of this particular sequence is M^{-N_1} which may be regarded as the error floor E_f of the system, i.e.,

$$E_f = M^{-N_1} \tag{7.13}$$

The above procedure has allowed us to evaluate error floors in PAM systems under sever distortions as long as we know the system's impulse response, which in general, can be measured practically.

7.3 CONCLUSIONS

A new method has been introduced to evaluate error floors in PRS systems operated above the Nyquist rate. The numerical results show that the error floors of our new Model-B are much lower than those of the conventional Model-A. This rigorously indicates that our new scheme is better than the conventional method for a practical realization.

The concept and method could be extended also to the study of other transmission systems, such as PAM systems with severe distortions. Stated briefly, a new concept has been introduced. That is, systems regarded as failure by the peak eye closure criterion are redefined as systems with error floors by the introduced truncated peak distortion. The error floors may be very small, depending on how severe the distortion is.

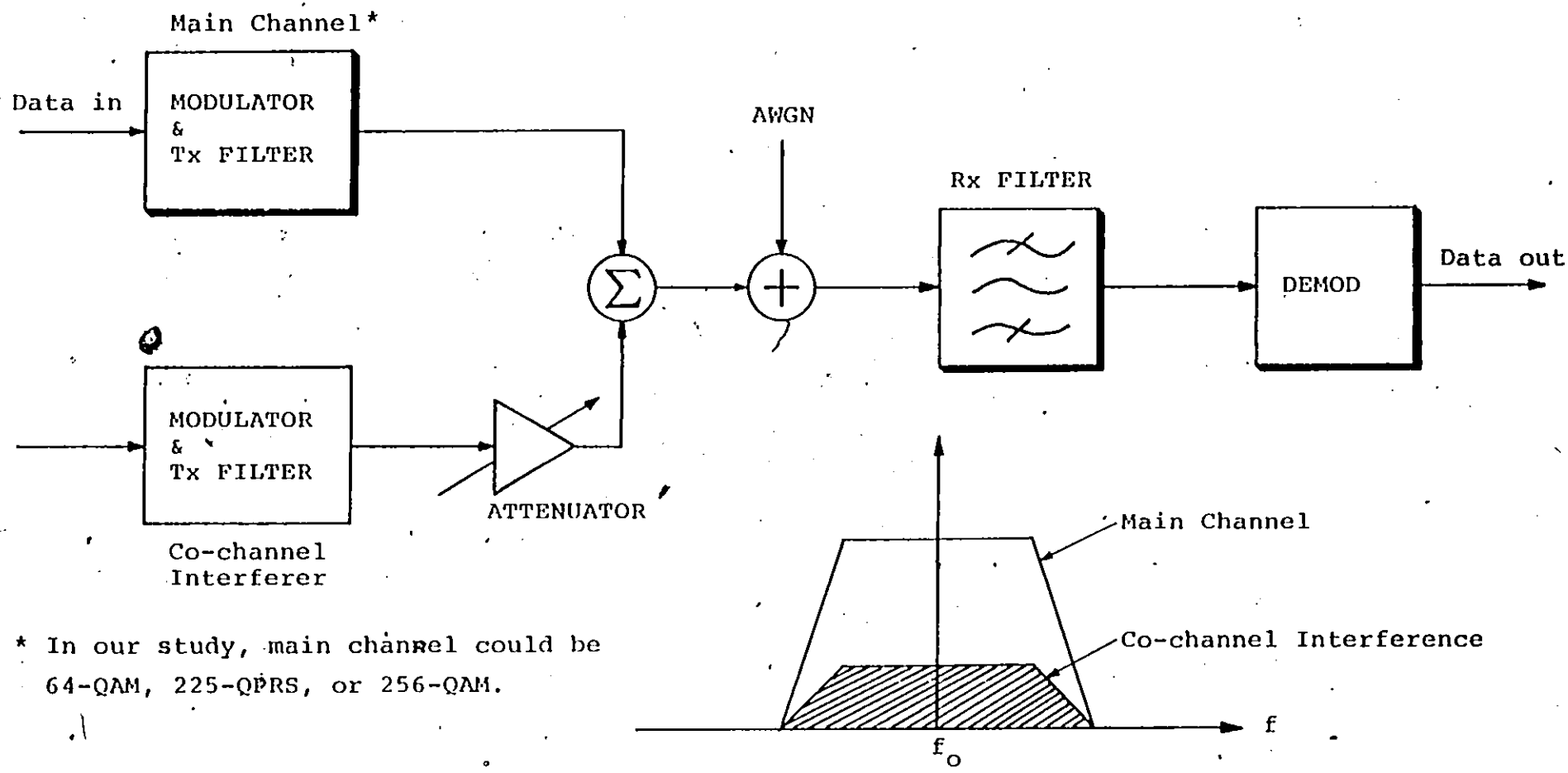
Chapter VIII

THE PERFORMANCE OF 64-QAM AND 225-QPRS AND 256-QAM IN A CO-CHANNEL INTERFERENCE ENVIRONMENT

One way to increase frequency utilization efficiency may be the use of dual polarization operation [42]. However, in a frequency-reuse communication system, co-channel interference (CCI) is one of the major sources of performance impairment [43]-[45]. In this chapter, we study the performance of 64-QAM, 225-QPRS, and 256-QAM in the presence of CCI so that their potential for radio systems could be assessed. Both constant-envelope (sinusoidal) and amplitude-modulated interferers are considered. In the simulation, coherent detection and perfect synchronization are assumed. Measured data for a 256-QAM modem in a CCI environment agree well with the simulated results.

8.1 SYSTEM PARAMETERS AND DESCRIPTION OF THE SIMULATION METHOD

A conceptual co-channel system block diagram is depicted in Fig.8.1 [46]. In this part of our study, we assume that only CCI and thermal noise cause degradation in the system. The main channel signal being considered may be 64-QAM, 225-QPRS, or 256-QAM in the presence of sinusoidal or amplitude-modulated interferers.



* In our study, main channel could be 64-QAM, 225-QPRS, or 256-QAM.

Illustrative frequency spectra of the desired and interfering signals

Fig.8.1 Block diagram of a conceptual co-channel system

For 64- and 256-QAM, the modulated signal without filtering can be represented by the general form:

$$s(t) = \mathrm{Re}\{ \sum_n (I_n + jQ_n)\, g(t-nT_s)\, e^{j2\pi f_o t} \} \tag{8.1}$$

where

f_o is the carrier frequency,

$1/T_s$ is the symbol rate,

$g(t)$ is a rectangular pulse defined by

$$g(t) = \begin{cases} 1 & \text{if } 0 \le t \le T_s \\ 0 & \text{elsewhere} \end{cases}$$

I and Q are independent and identically distributed random variables: I_n, $Q_n = \pm 1, \pm 3, \ldots, \pm(2N-1)$.

$N = 4$ for 64-QAM; $N = 8$ for 256-QAM.

The transmit and receive filters are assumed as ideal square root of $\alpha = 0.2$ raised-cosine filters with $x/\sin(x)$ equalization in the transmitter so that the whole system satisfies the Nyquist 1st criterion [14].

If the main channel consists of 225-QPRS, the transmitted signal can be written as

$$s(t) = \mathrm{Re}\{ \sum_n (I_n + jQ_n)\, h(t-nT_s)\, e^{j2\pi f_o t} \} \tag{8.2}$$

where

the duobinary PRS shape $h(t)$ [20] is given by

$$h(t) = \frac{\sin(\pi t/T_s)}{\pi t/T_s} \cdot \frac{1}{1 - t/T_s} \tag{8.3}$$

Note that the center of the duobinary impulse response is at $T_s/2$.

We assume that all the pulse shaping is done at the transmitter, whereas the receive filter merely bandlimits the channel noise. This would result in approximately a 1 dB degradation in C/N, as explained in Appendix A.

Two types of CCI are considered. If the CCI consists of a single sinusoid, the total received signal r(t) can be expressed as

$$r(t) = s_r(t) + K\cos(2\pi f_o t+\Phi) + n(t) \tag{8.4}$$

where

$K^2/2$ is the average power in the sinusoidal CCI,

Φ is uniformly distributed in $(-\pi, \pi]$,

$s_r(t)$ is the received desired signal,

$n(t)$ is the additive white Gaussian noise.

If the CCI is a single amplitude-modulated signal, we assume that the interferer and the desired signal are of the same format but statistically independent of each other. In other words, we consider 64-QAM with a 64-QAM CCI, 225-QPRS with a 225-QPRS CCI, and the like. For 64-, and 256-QAM interferers, the transmitted interfering signal without filtering can be modelled

$$i(t) = K\,\mathrm{Re}\left\{ \sum_n (\tilde{I}_n+j\tilde{Q}_n)\, g(t-nT_s-\tau)\, e^{j(2\pi f_o t+\Phi)} \right\} \tag{8.5}$$

In the case of 225-QPRS interferer, the transmitted signal is

$$i(t) = K\,\mathrm{Re}\left\{ \sum_n (\tilde{I}_n+j\tilde{Q}_n)\, h(t-nT_s-\tau)\, e^{j(2\pi f_o t+\Phi)} \right\} \tag{8.6}$$

In (8.5) and (8.6), $\tilde{I}_n$, $\tilde{Q}_n$, $g(t)$, $h(t)$, T_s, and f_o are defined as in (8.1) and (8.2). τ is the difference in symbol timing between the desired signal and interferer and is assumed to be uniformly distributed in $(0, T_s)$. Φ is a random phase angle uniformly distributed in $(-\pi, \pi]$. $1/K^2$ is the carrier-to-interference ratio, C/I.

For a fixed C/I, the probability of error is calculated by computer simulations using the similar steps in [8],[36] and Chapter 5 for randomly selected Φ and τ. To obtain a meaningful result, we average the probability of error for several sets of Φ, τ, and data sequences.

8.2 SIMULATED RESULTS AND DISCUSSIONS.

The probability of error performances for 64-QAM, 225-QPRS, and 256-QAM with a single sinusoidal CCI are shown in Figs.8.2-8.4, respectively. The parameter C/I is the ratio of the average power in the desired main channel signal to the average power in the CCI. The quantity C/N represents the ratio of the average power in the desired signal to the noise power. Since a system designer often wants to maintain a specified error rate even when there is interference, the increase in C/N required to maintain an error rate of 10^{-6} is plotted in Fig.8.5.

The error probability of 64-QAM ($\alpha = 0.2$) with a 64-QAM ($\alpha = 0.2$) CCI is plotted in Fig.8.6. For 225-QPRS with a 225-QPRS CCI, the error probability is shown in Fig.8.7. In Fig.8.8, we present the results for 256-QAM ($\alpha = 0.2$) with a 256-QAM ($\alpha = 0.2$) CCI. Fig.8.9 compares the increase in C/N required to maintain an error rate of 10^{-6} for the three modulation schemes. The results in Figs.8.5 and 8.9 are con-

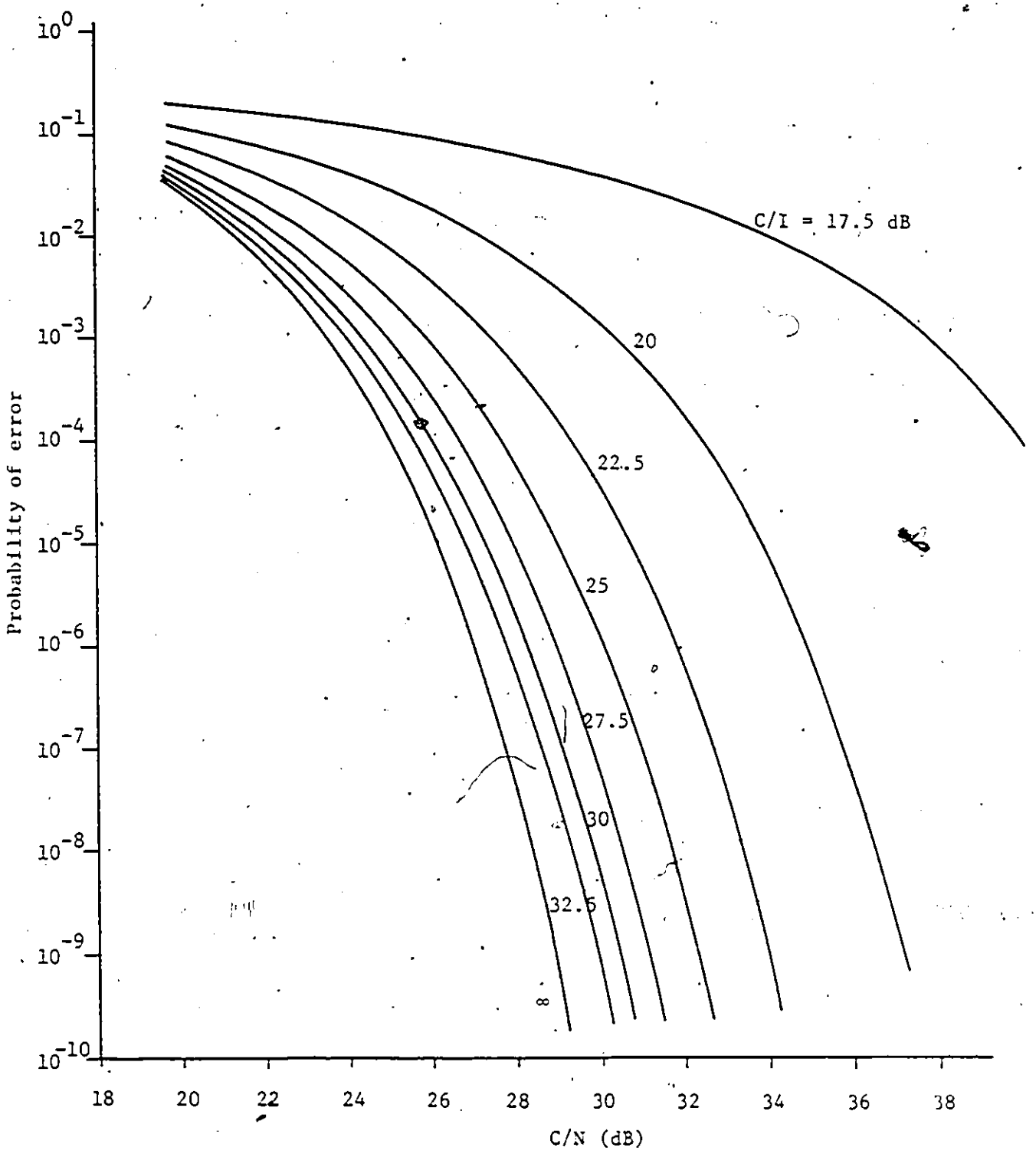


Fig. 8.2 Error probability of 64-QAM (α = 0.2) with a single sinusoidal CCI. The parameter C/I is the ratio of the average carrier power to the average power in the interference. The quantity C/N represents the ratio of the average carrier power to the noise power.

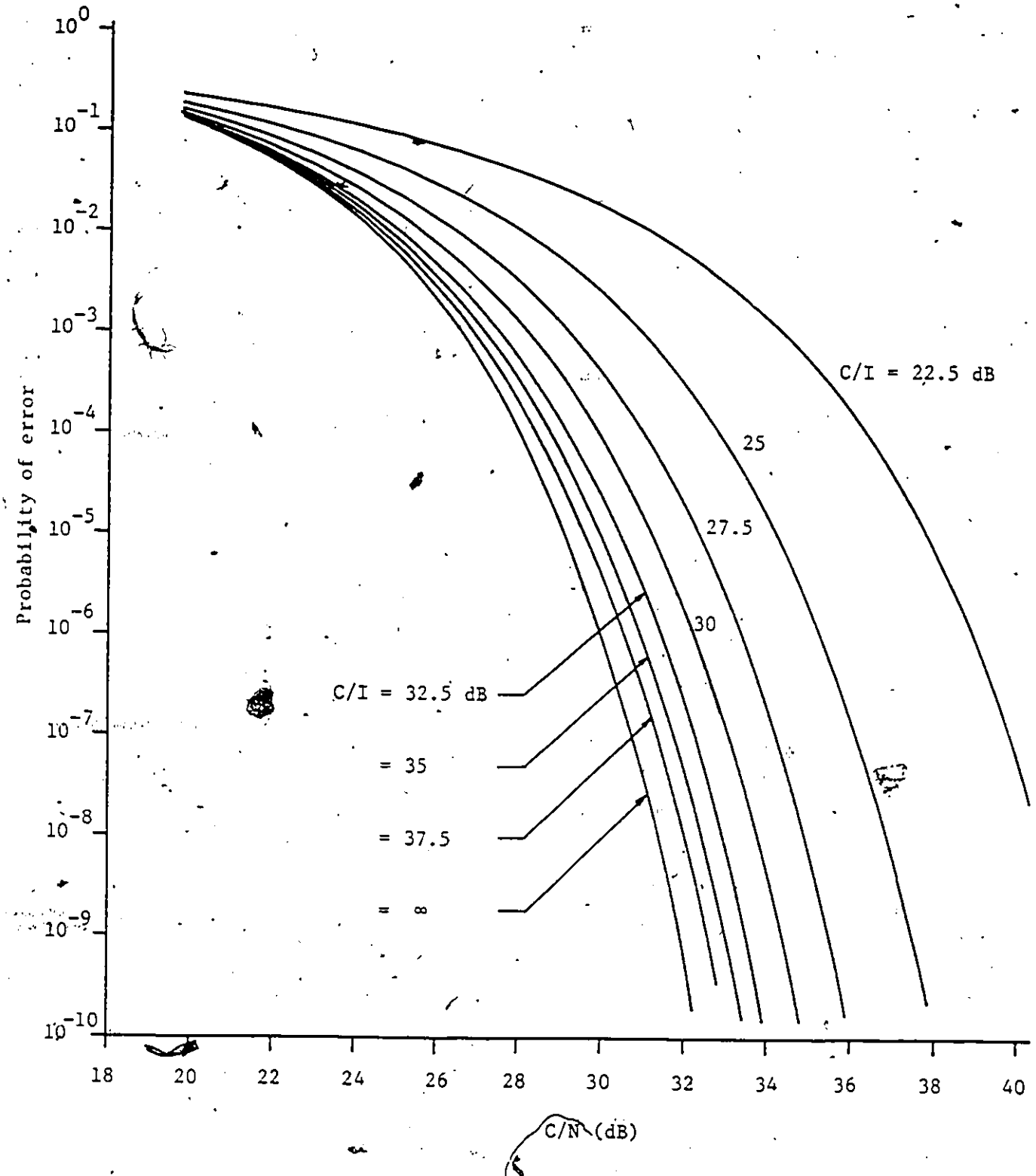


Fig.8.3 Error probability of 225-QPRS with a single sinusoidal CCI.

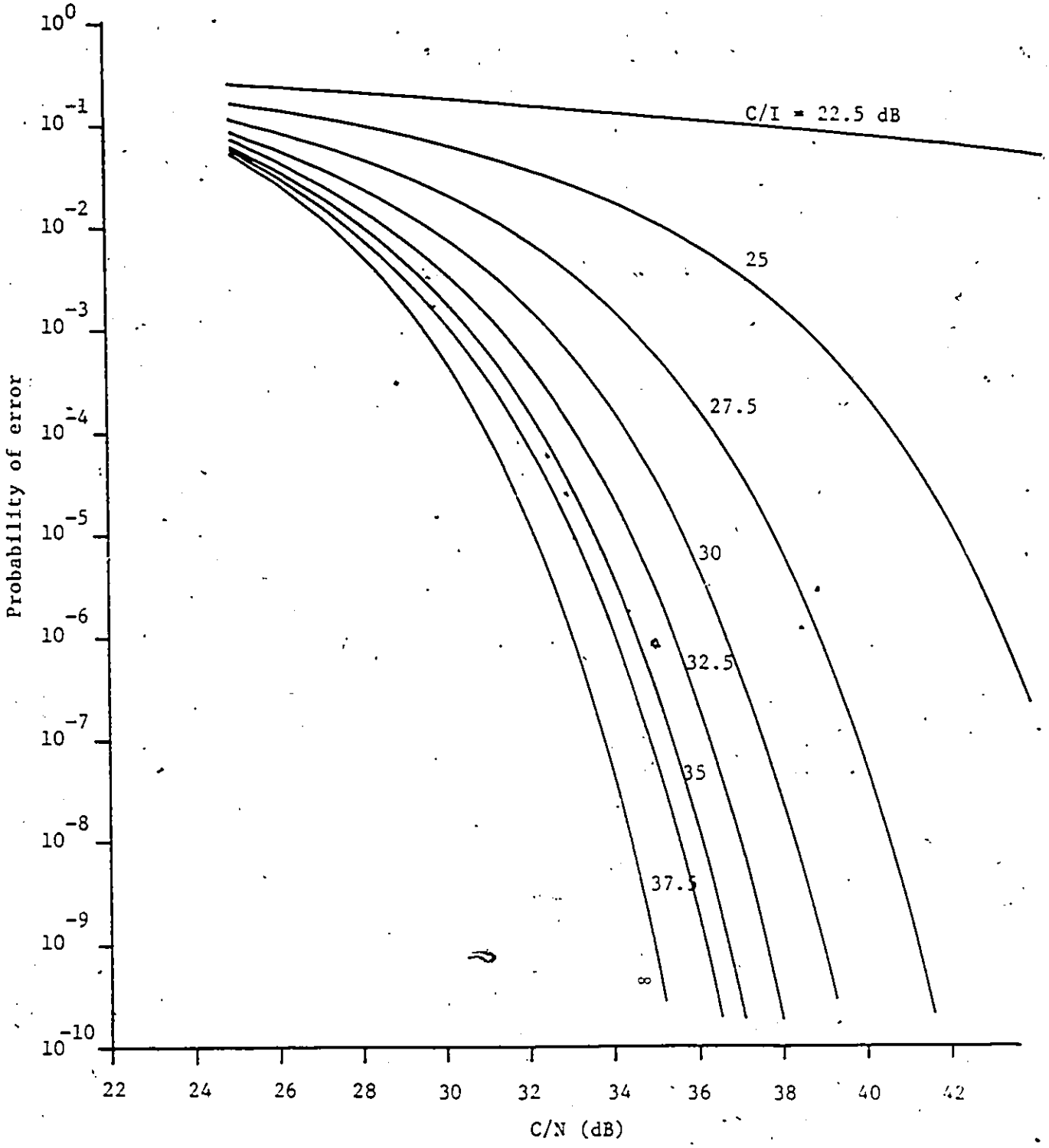


Fig.8.4 Error probability of 256-QAM (α = 0.2) with a single sinusoidal co-channel interference.

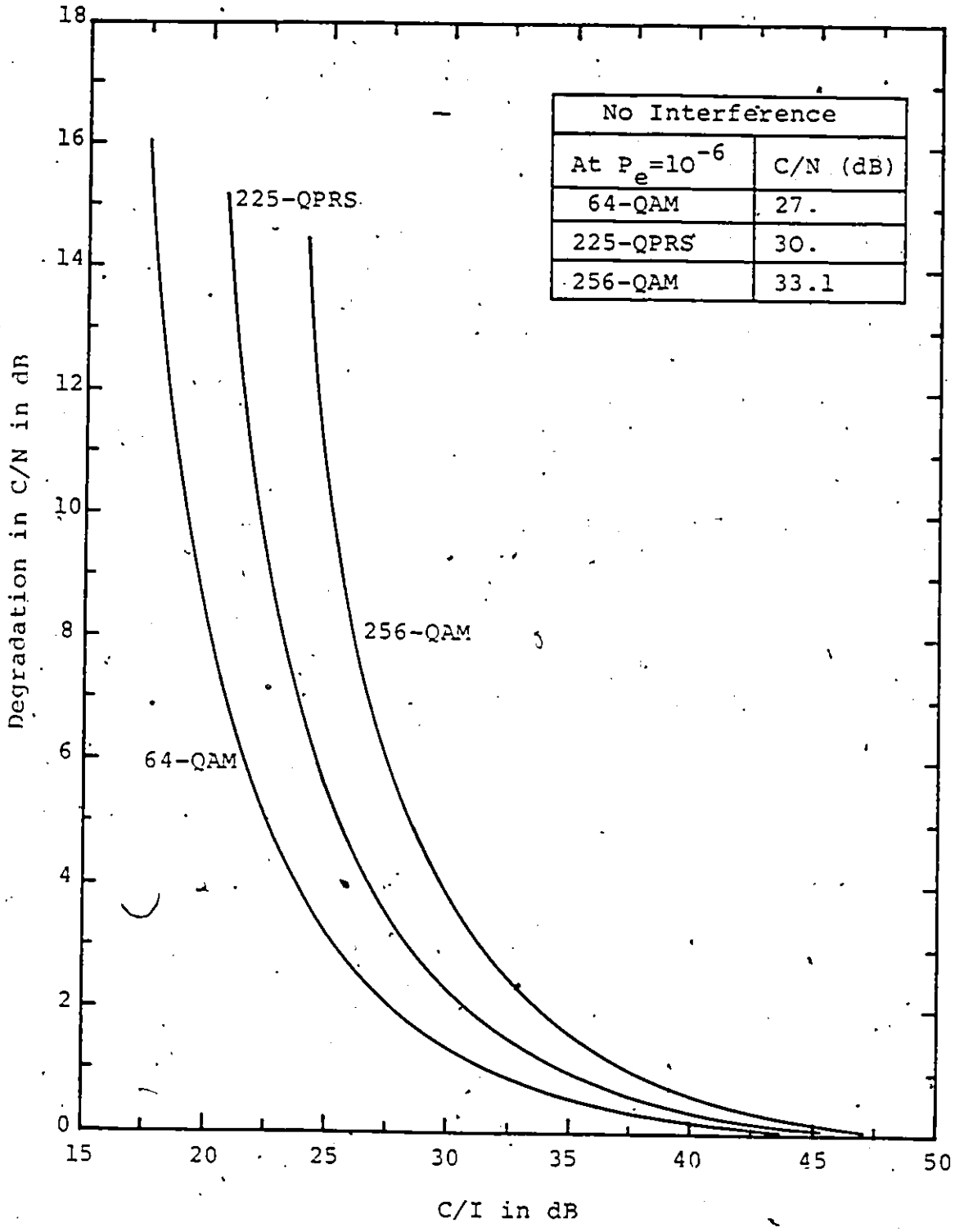


Fig.8.5 Degradation in C/N for 64-, 256-QAM (α = 0.2), and 225-QPRS with a single sinusoidal CCI.
The ordinate represents the increase in C/N to maintain an error rate of 10^{-6}.

firmed to agree with those of Prabhu's [43] and of Borgne's [13] for 64-QAM.

Comparing Figs.8.5 and 8.9, we observe that the amplitude-modulated CCI produces more degradation than the single sinusoidal CCI for the three modulation schemes considered. This can easily be perceived by comparing the peak value of the interference for the same power. For example, if $A^2/2$ is the average power in the interferer, the peak value for the single sinusoid is A ; for a QAM interferer (assume no filtering for simplicity) it is $A(2N-1)[3/(4N^2-1)]^{\frac{1}{2}}$ [44], where N = 4 for 64-QAM and N = 8 for 256-QAM. That is, the peak value for a 64-QAM CCI is $7/\sqrt{21}$ = 1.527 times that of a sinusoidal CCI for the same power ; and for 256-QAM CCI it is $15/\sqrt{85}$ = 1.627 times. The peak amplitude of the CCI reduces the minimum half-distance of any two signal levels. Thus, the degradation with an amplitude-modulated CCI is higher than with a sinusoidal CCI.

8.3 EXPERIMENTAL RESULTS

To confirm the simulated results, the performance of a 256-QAM modem in the presence of CCI was experimentally measured [46]. The 256-QAM modem was designed to transmit a 1.6 Mb/s data stream in a 240 kHz SG band. Illustrative results are shown in Fig.8.10. Note that the back-to-back modem through two SG filters has approximately a 1.3 dB degradation in C/N at $P_e = 10^{-6}$ with respect to the theoretical value when there is no CCI.

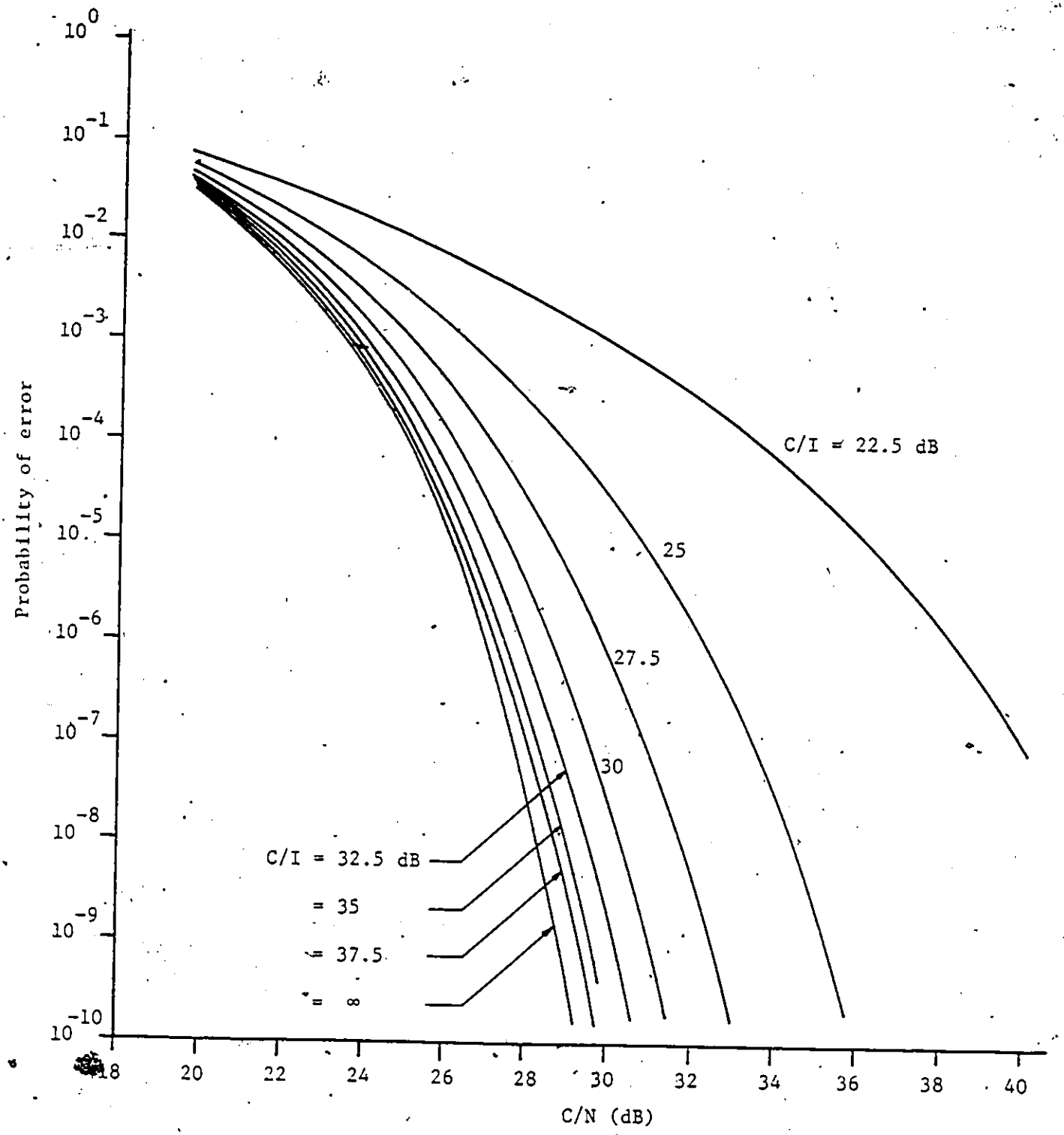


Fig.8.6 Error probability of 64-QAM (α = 0.2) with a 64-QAM (α = 0.2) CCI.

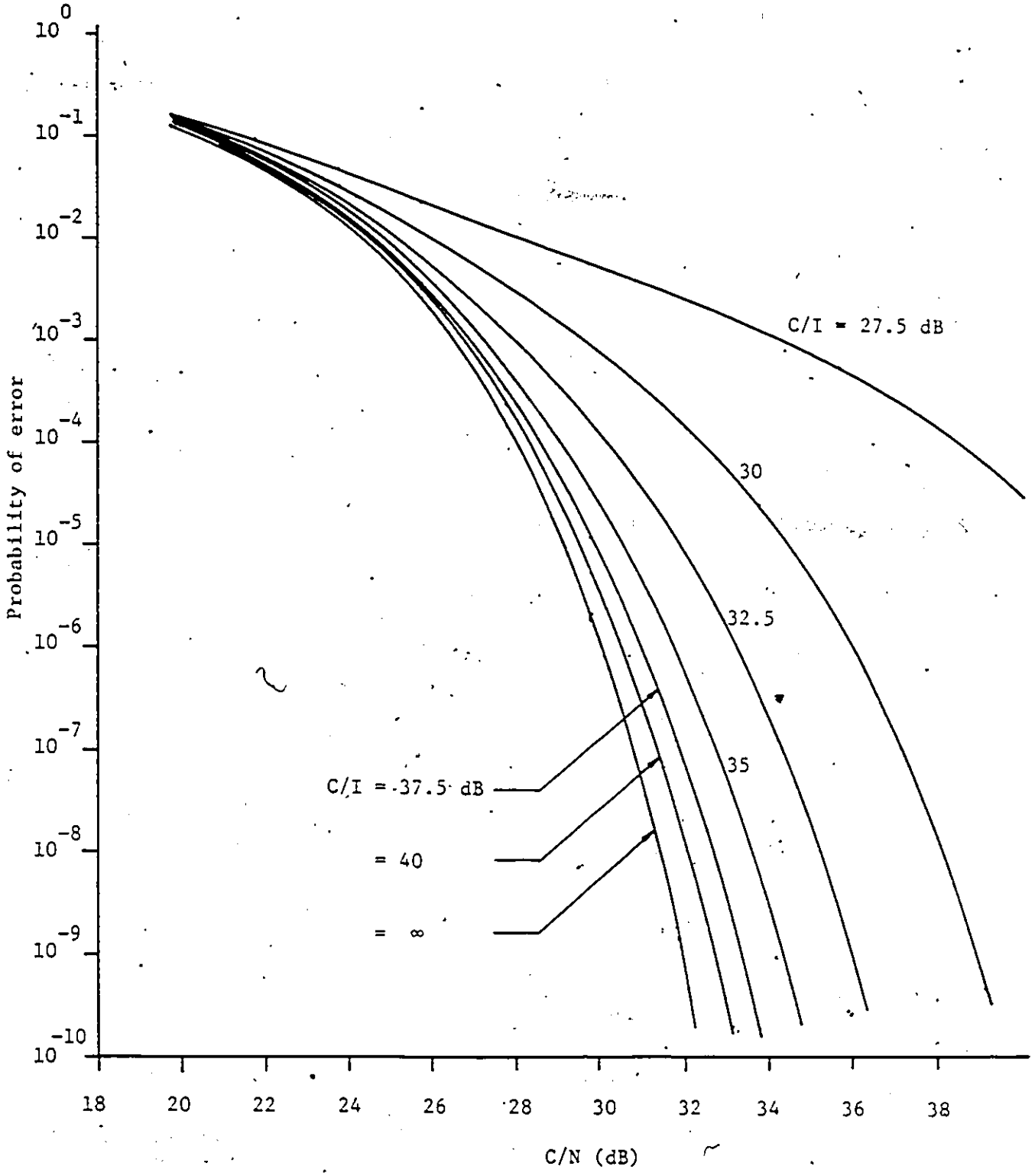


Fig.8.7 Error probability of 225-QPRS with a 225-QPRS CCI.

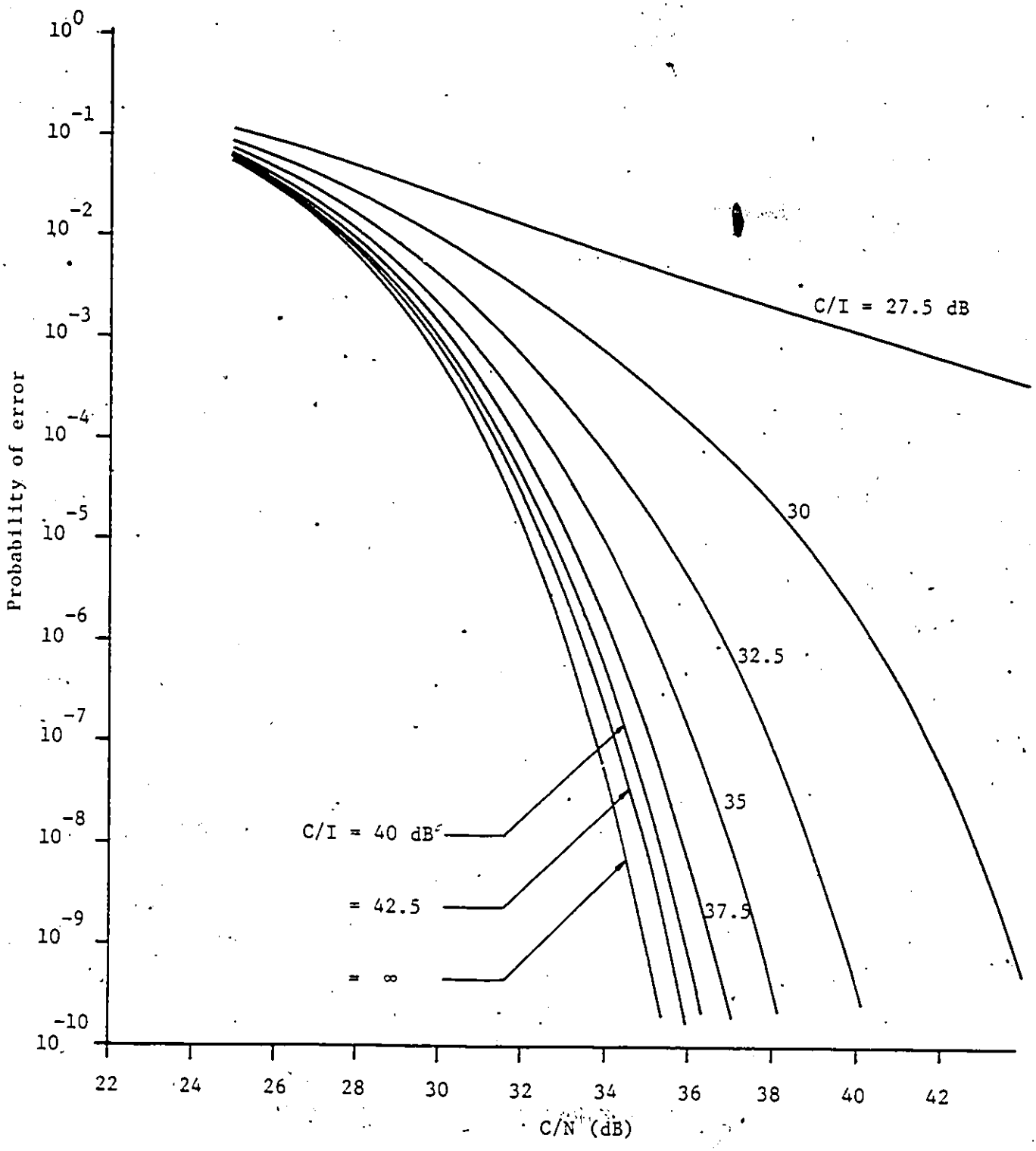


Fig.8.8 Error probability of 256-QAM (α = 0.2) with a 256-QAM (α = 0.2) CCI.

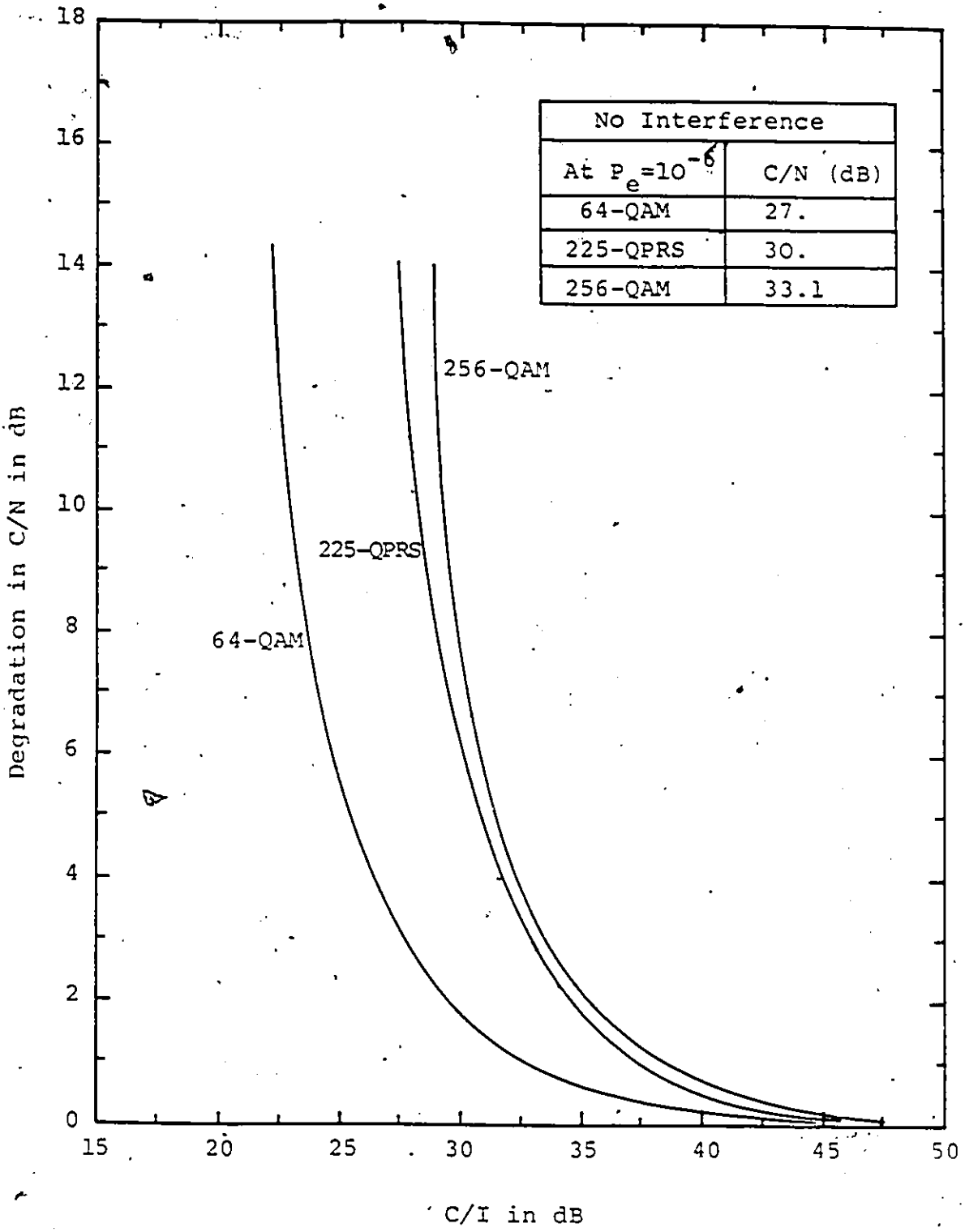


Fig.8.9 Degradation in C/N for 64-, 256-QAM (α = 0.2), and 225-QPRS with an amplitude-modulated co-channel interference which has the same format as the desired signal, e.g., 64-QAM with a 64-QAM CCI, etc. The ordinate represents the increase in C/N to maintain an error rate of 10^{-6}.

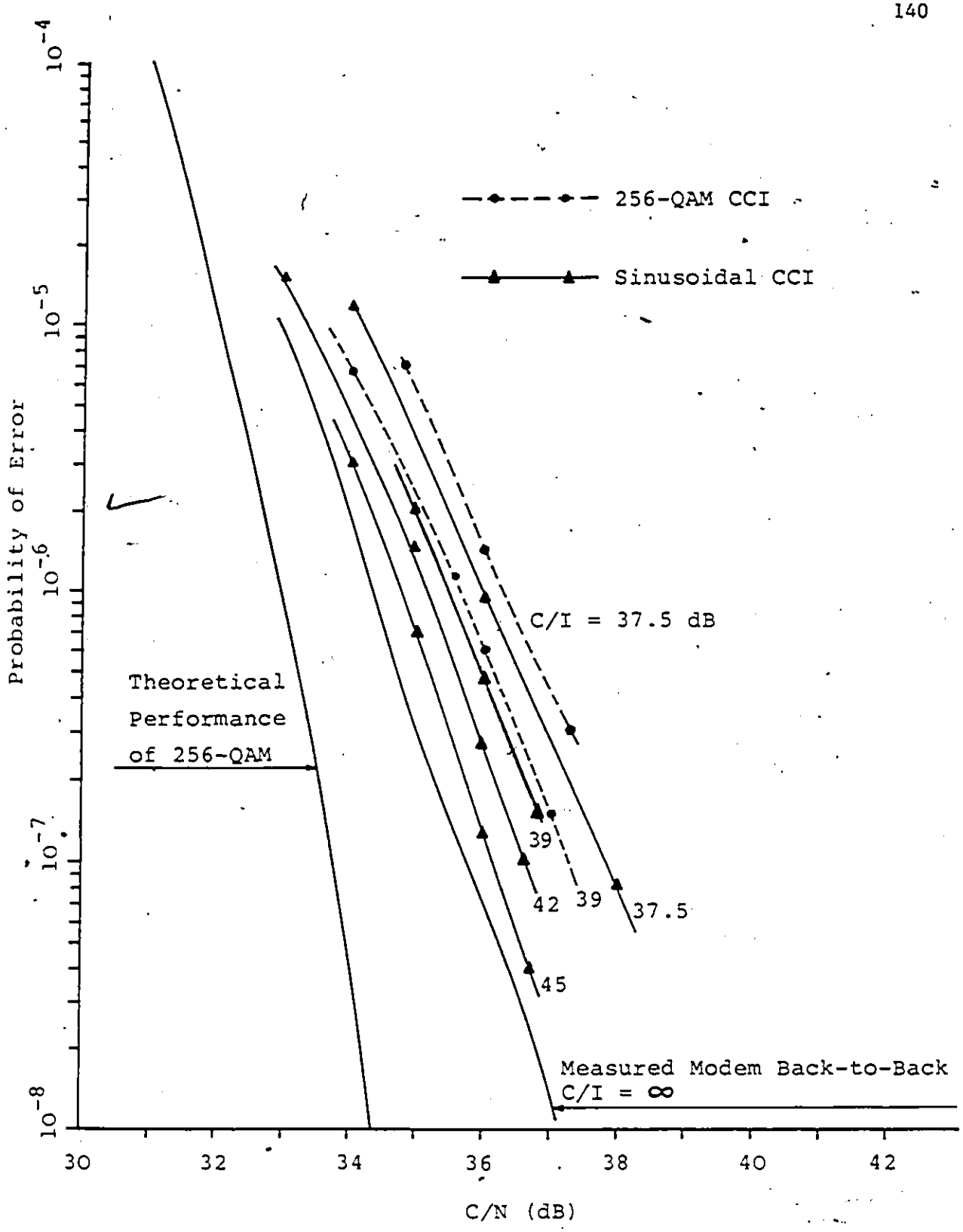


Fig.8.10 Measured performance of a 256-QAM modem in the presence of co-channel interference. The 256-QAM modem was designed to transmit 1.6 Mb/s data in a supergroup bandwidth of 240 kHz.

The measured data show more degradations than those in Figs.8.4 and 8.8 in which an ideal modem was assumed. This is because that, due to hardware and filters imperfections, the 256-QAM signal itself contains ISI which reduces the minimum distance between adjacent signal levels and in turn causes the signal to be more susceptible to degradation caused by CCI. For example, for a sinusoidal CCI, at $P_e = 10^{-6}$ and C/I = 37.5 dB the required C/N is 36 dB from the measured data (Fig.8.10), whereas it is 34 dB from Fig.8.4. The difference of 2 dB in C/N is caused by hardware and filters imperfections. Therefore, the measured data generally show a good agreement with the simulated results.

8:4 CONCLUSIONS

For an error rate of 10^{-6}, the required C/N for 64-QAM, 225-QPRS and 256-QAM are 27, 30, and 33.1 dB, respectively when there is no interference. When there is an amplitude-modulated CCI with C/I = 28.75 dB, the degradation in C/N for 64-QAM, 225-QPRS, and 256-QAM are 2.3, 8.5, and 13 dB, respectively. In particular, degradations in 225-QPRS and 256-QAM increase dramatically when C/I becomes less than 40 dB. Measured data for a 256-QAM modem in a CCI environment are in good agreement with the simulated results. We conclude that to make dual polarization operation feasible for high-level modulation schemes, such as 225-QPRS and 256-QAM, a cross polarization discrimination of better than 40 dB is required since these systems are also sensitive to other impairments.

Chapter IX

AN ADAPTIVE BASEBAND EQUALIZER FOR 256-QAM OVER DISTORTED CHANNELS

It is shown in Chapter 5 that 256-QAM is relatively sensitive to channel distortions. The design of filters must be made very carefully. However, in some cases, such as carrying digital data over existing analog cable and radio networks, the channel-protection filter cannot be removed [17]. Even if one could design a perfect modem, this channel filter will still cause performance degradation.

On the other hand, all digital radio systems, no matter what modulation method is used, suffer the variations of the propagation medium characteristics which occur during selective fading. Usually, in the design of the transmission channel, no ISI is the major aim. The desired amplitude and phase characteristics of the channel are obtained by means of filters in the transmitter and receiver, whose synthesis is made with reference to normal propagation conditions, i.e., only one ray from the transmitting antenna to the receiving one. In the presence of multiple paths, during selective fading, the behavior of the propagation medium shows notable variations which produce ISI in the signal. This ISI, in some propagation conditions, can cause system unavailability. Under these circumstances, adaptive equalizers [47]-[52] can be used to meet design objectives.

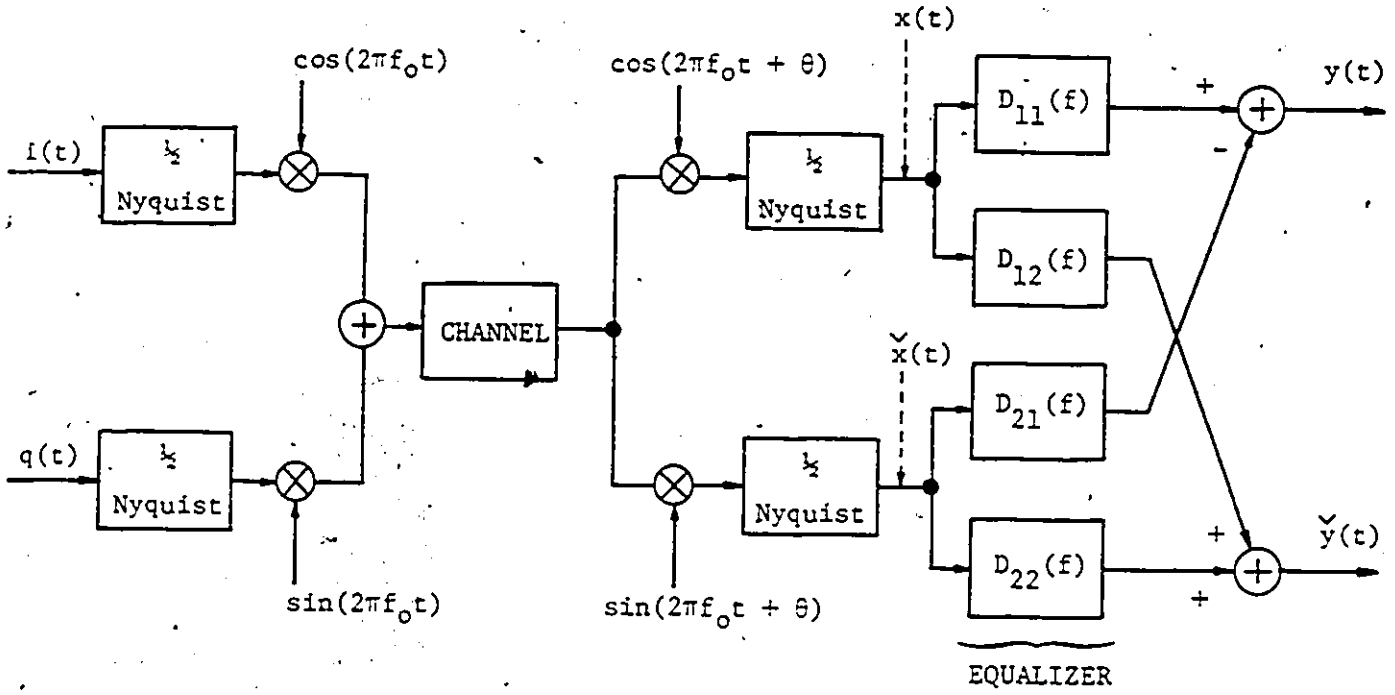
In this chapter, we shall study the possibility of using a baseband complex adaptive equalizer for 256-QAM to combat channel distortions. The function and structure of the equalizer will be described first. Then the computer simulated results for certain distortion conditions will be presented.

9.1 FUNCTIONAL DESCRIPTION OF THE EQUALIZER

The equalizer structure employed for 256-QAM is similar to that described in [47] for 16-QAM. Fig.9.1 depicts the block diagram of a 256-QAM system with $\alpha = 0.2$ raised-cosine filters equally split between the transmitter and receiver. First, it is important to see how the equalizer structure is constructed. It is convenient to represent the in-phase (I) and quadrature (Q) channel lowpass filter output signals in Fig.9.1 by $x(t)$ and $\check{x}(t)$ as the real and imaginary parts of a complex-value signal. (Note that the signals are real, but it will be convenient to use complex notation.) The baseband equalizer with a complex transfer function $D(f) = D_1(f) + j\, D_2(f)$ produces a complex equalized signal $y(t) + j\, \check{y}(t)$ as shown in Fig.9.2.

The function of the equalizer is the multiplication of $[X(f) + j\check{X}(f)]$ and $[D_1(f) + j\, D_2(f)]$, i.e.,

$$\begin{aligned}
&\{X(f) + j\check{X}(f)\}\{D_1(f) + jD_2(f)\} \\
&= X(f)D_1(f) - \check{X}(f)D_2(f) + j\{\check{X}(f)D_1(f) + X(f)D_2(f)\} \\
&= Y(f) + j\check{Y}(f) = \mathcal{F}\{y(t) + j\check{y}(t)\}
\end{aligned} \tag{9.1}$$



$$i(t) = \sum_n I_n \, \delta(t - nT_s)$$

$$q(t) = \sum_n Q_n \, \delta(t - nT_s)$$

I_n and $Q_n \in \{ \pm 1, \pm 3, \ldots, \pm 15 \}$

Fig.9.1 Block diagram of a 256-QAM system with a complex baseband equalizer.

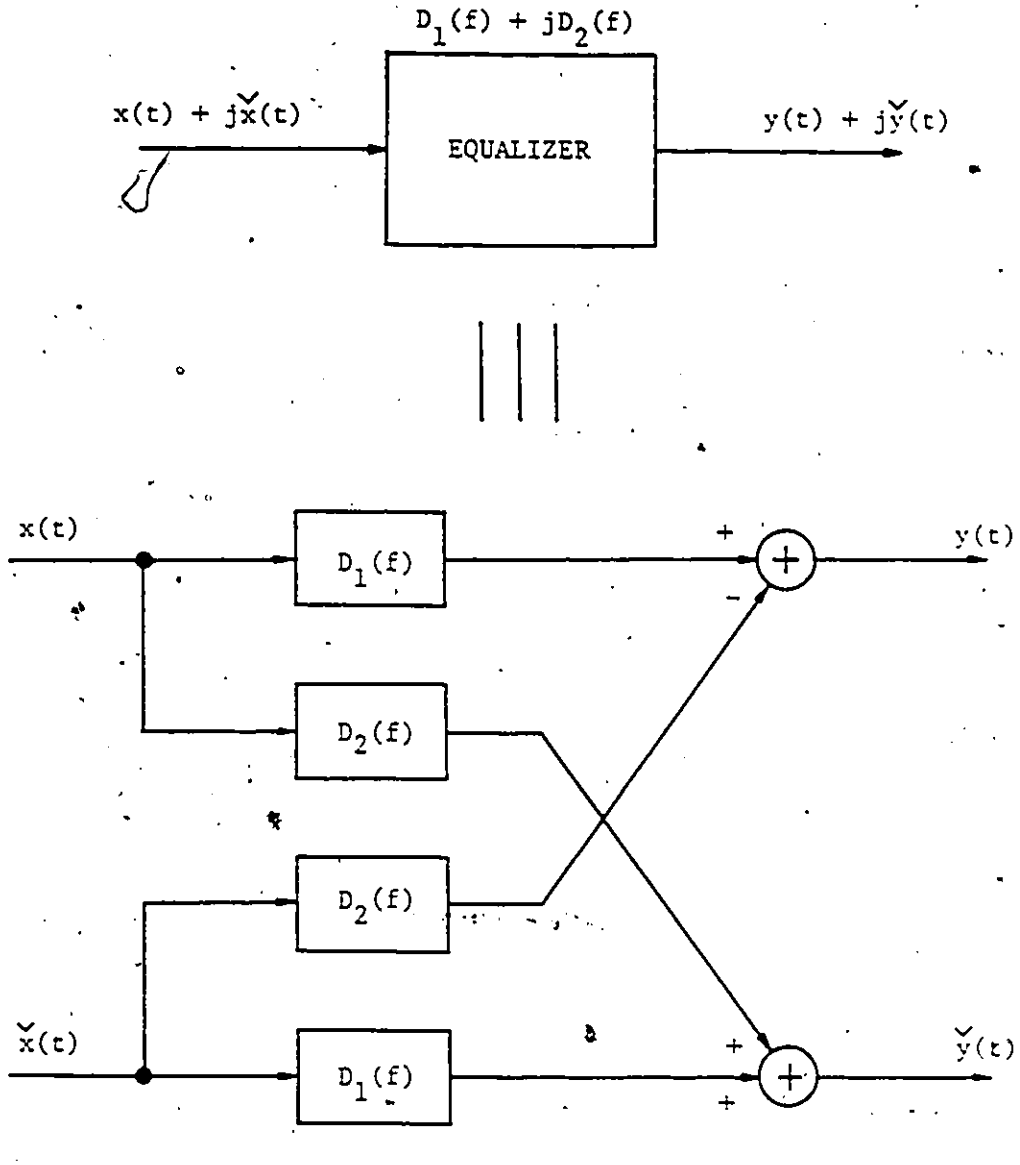


Fig.9.2 Complex transversal equalizer for QAM modems.

where

$y(t)$ and $\check{y}(t)$ are the equalized I and Q-channel signals.

Fig.9.2 and eq.(9.1) illustrate more concretely the concept of a complex equalizer as a set of four real baseband filters (with cross-coupling) for two inputs and two outputs. In practice, due to hardware imperfection, the combination of the receive filter and the equalizer in the I-channel may not be exactly identical to that in the Q-channel. Moreover, the channel distortion may not be always symmetric about the carrier frequency. In both cases, the introduced crosstalk from I-channel to Q-channel is not the same as that from Q-channel to I-channel. This suggests that it is better to have four different baseband equalizers in order to have a better control in reducing the ISI and crosstalk. This also explains the use of notations $D_{11}(f)$, $D_{12}(f)$, $D_{21}(f)$, and $D_{22}(f)$ in Fig.9.1. What is more, the implementation of the structure shown in Fig.9.1 is not more complicated than that of Fig.9.2. Because of hardware simplicity, the equalizer chosen is a linear tapped-delay-line transversal structure described by

$$D_{11}(f) = \sum_i c_{i,I}\, e^{-j2\pi nfT_s} \tag{9.2}$$

$$D_{12}(f) = \sum_i d_{i,I}\, e^{-j2\pi nfT_s} \tag{9.3}$$

$$D_{21}(f) = \sum_i d_{i,Q}\, e^{-j2\pi nfT_s} \tag{9.4}$$

$$D_{22}(f) = \sum_i c_{i,Q}\, e^{-j2\pi nfT_s} \tag{9.5}$$

A 5-tap adaptive transversal equalizer is depicted in Fig.9.3 for illustration. The number of taps can easily be changed in the simulation.

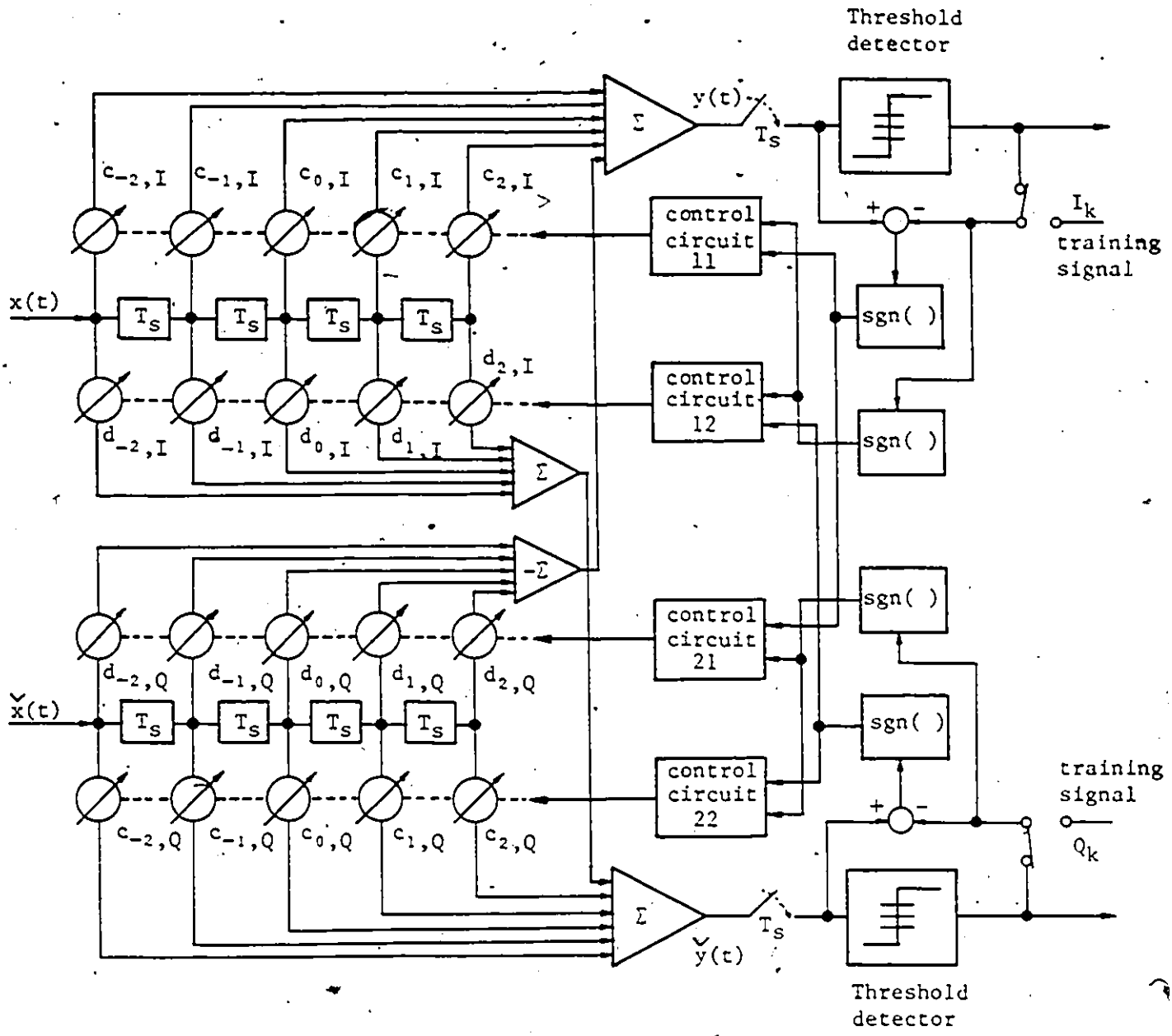


Fig.9.3 An illustrative structure of a 5-tap complex baseband adaptive equalizer for QAM systems.

In Figs.9.1 and 9.3, let y_k and $\check{y}_k$ be the samples of the output of the equalizer for the I- and Q-channel signals, respectively. The tap-weight adjustment algorithm is governed by the following equations (for the i-th tap-weight at the k-th symbol interval):

$$c_{i,I}^{(k+1)} = c_{i,I}^{(k)} - \Delta \operatorname{sgn}(e_k)\ \operatorname{sgn}(y_{k-i}) \tag{9.6}$$

$$c_{i,Q}^{(k+1)} = c_{i,Q}^{(k)} - \Delta \operatorname{sgn}(\check{e}_k)\ \operatorname{sgn}(\check{y}_{k-i}) \tag{9.7}$$

$$d_{i,I}^{(k+1)} = d_{i,I}^{(k)} - \Delta \operatorname{sgn}(\check{e}_k)\ \operatorname{sgn}(y_{k-i}) \tag{9.8}$$

$$d_{i,Q}^{(k+1)} = d_{i,Q}^{(k)} - \Delta \operatorname{sgn}(e_k)\ \operatorname{sgn}(\check{y}_{k-i}) \tag{9.9}$$

where $\{c_{i,I}, d_{i,I}\}$ are tap-weights associated with the I-channel tapped delay line and $\{c_{i,Q}, d_{i,Q}\}$ are with the Q-channel tapped delay line. Δ is the tap-weight incremental step size, and $e_k = y_k - I_k$, $\check{e}_k = \check{y}_k - Q_k$ represent the errors between the equalized and the transmitted samples for the I- and Q-channels, respectively. Implementation of (9.6)-(9.9) makes possible not only the minimization of the ISI but also the suppression of the crosstalks between the I and Q channels.

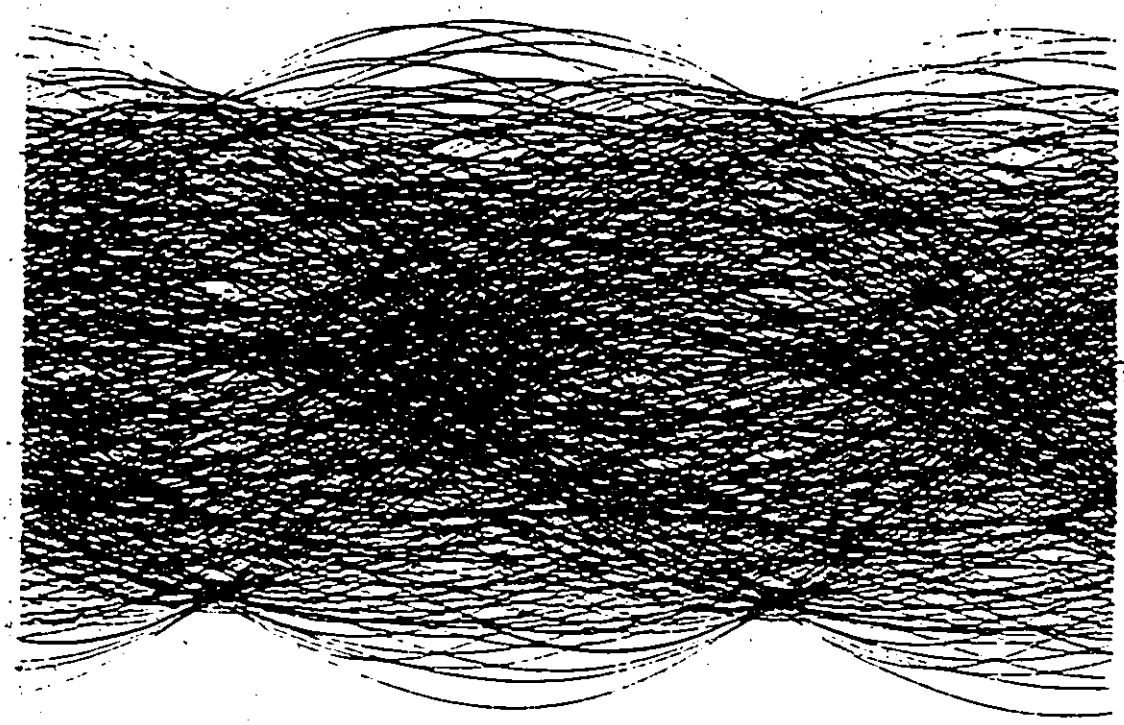
9.2 EQUALIZER PERFORMANCE

In this section, the equalizer performance is demonstrated by some typical results obtained by computer simulations. A bit rate of 120 Mb/s, which is equivalent to a symbol rate of 15 MBaud, is used in the simulation. The results can be applied to another bit rate by appropriate scaling, as previously described in Chapter 5. A training procedure

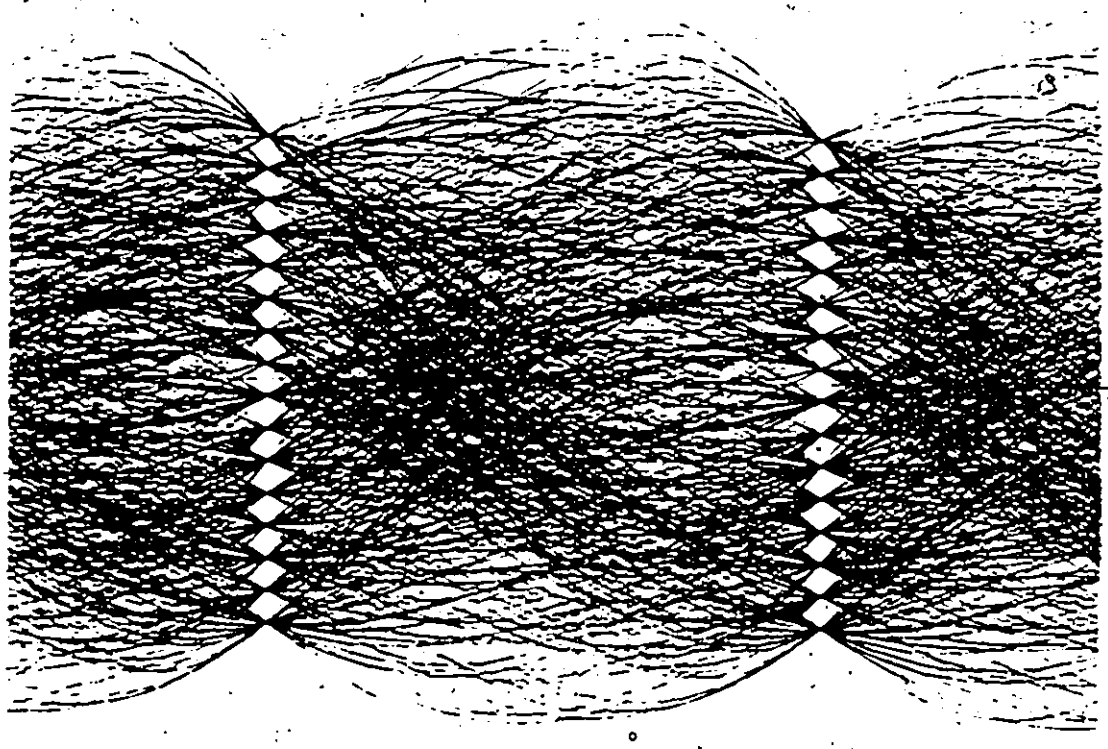
is used for fast convergence of the equalizer's tap coefficients and it is found that a training sequence of 2048 symbols is sufficient for ensuring convergence. A step size of 0.0001 was used in obtaining all the results presented in this section. This value was found to be most favorable after trying several different values for the step size.

First, we present a few eye diagrams which illustrate the powerfulness of the equalizer. For a channel distortion network which introduces a parabolic group delay distortion described by (5.7) with P_D=0.81 ns/MHz^2, the eye diagrams before and after the equalizer are shown in Fig.9.4(a) and (b), respectively. We see that the equalizer opens the multi-level eyes which were completely closed before the equalizer. Fig.9.5 compares the eye diagrams before and after the equalizer for 256-QAM (α= 0.2) with a sinusoidal group delay channel distortion described by (5:7) with S_D = 12 ns and K = 4.

We notice that significant ISI exists in the eye diagram before the equalizer and the equalizer effectively eliminates the ISI caused by the channel distortion. The error rate performance for a parabolic group delay distortion with P_D = 0.81 ns/MHz^2 is shown in Fig.9.6. Without the equalizer, the received signal cannot be decoded. The equalizer brings back the result (curve A) pretty close to the ideal curve. The small degradation is due to the finite taps of the equalizer which in turn yields a residual ISI. It is found that, in this case, a 31-tap adaptive equalizer is needed for 256-QAM to combat the channel distortion. Also shown in Fig.9.6 is curve B for a 15-tap equalizer in which the residual ISI is higher so that the performance is worse than curve

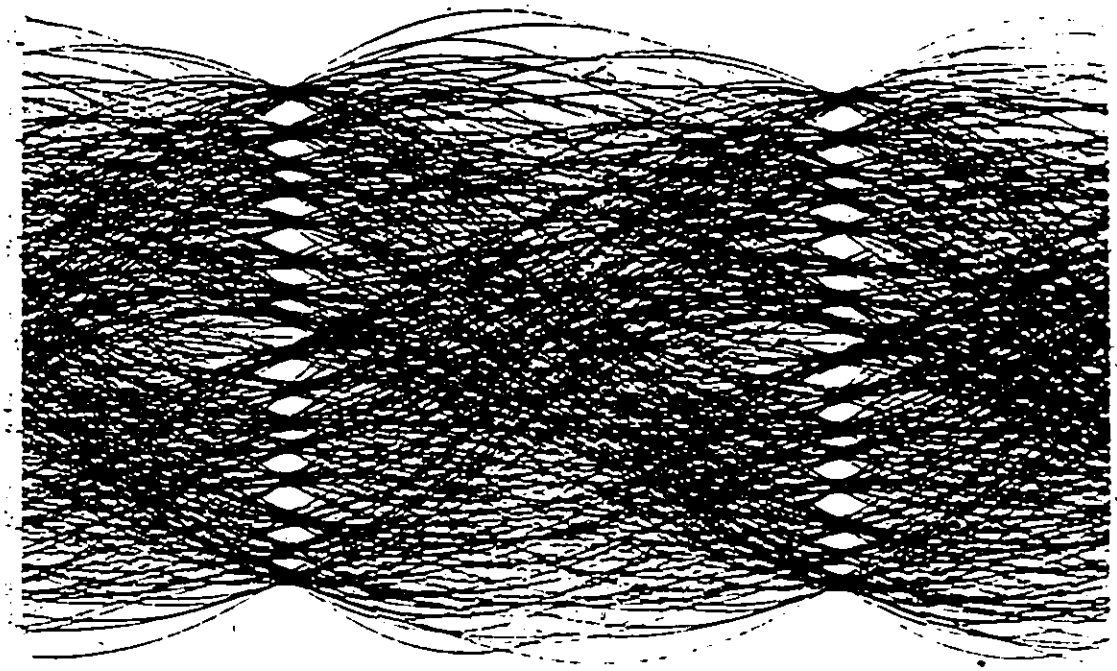


(a) Before equalization.

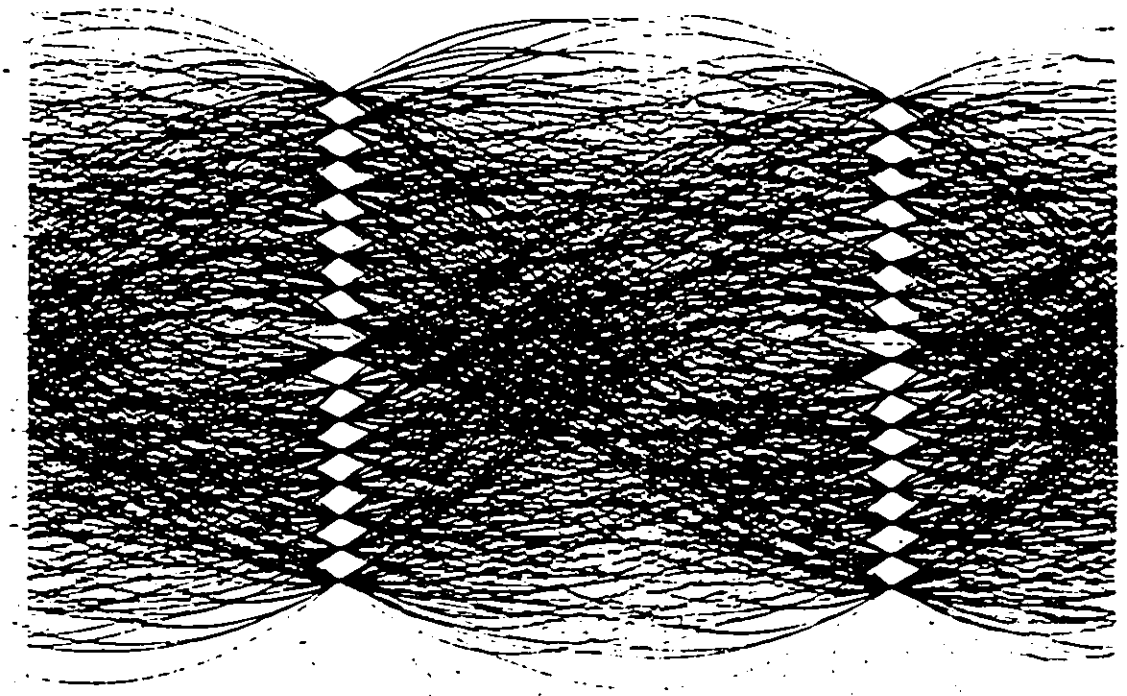


(b) After equalization.

Fig.9.4 Eye diagrams of 256-QAM (α = 0.2) with a parabolic group delay distortion, P_D = 0.81 ns/MHz^2. Bit rate = 120 Mb/s.
(a) Before equalization, (b) After equalization.



(a) Before equalization.



(b) After equalization.

Fig.9.5 Eye diagrams of 256-QAM (α = 0.2) with a sinusoidal group delay distortion $D(f) = S_D \sin(2\pi Kf/2f_{BW})$, S_D = 12 ns and K = 4. Bit rate = 120 b/s/Hz.

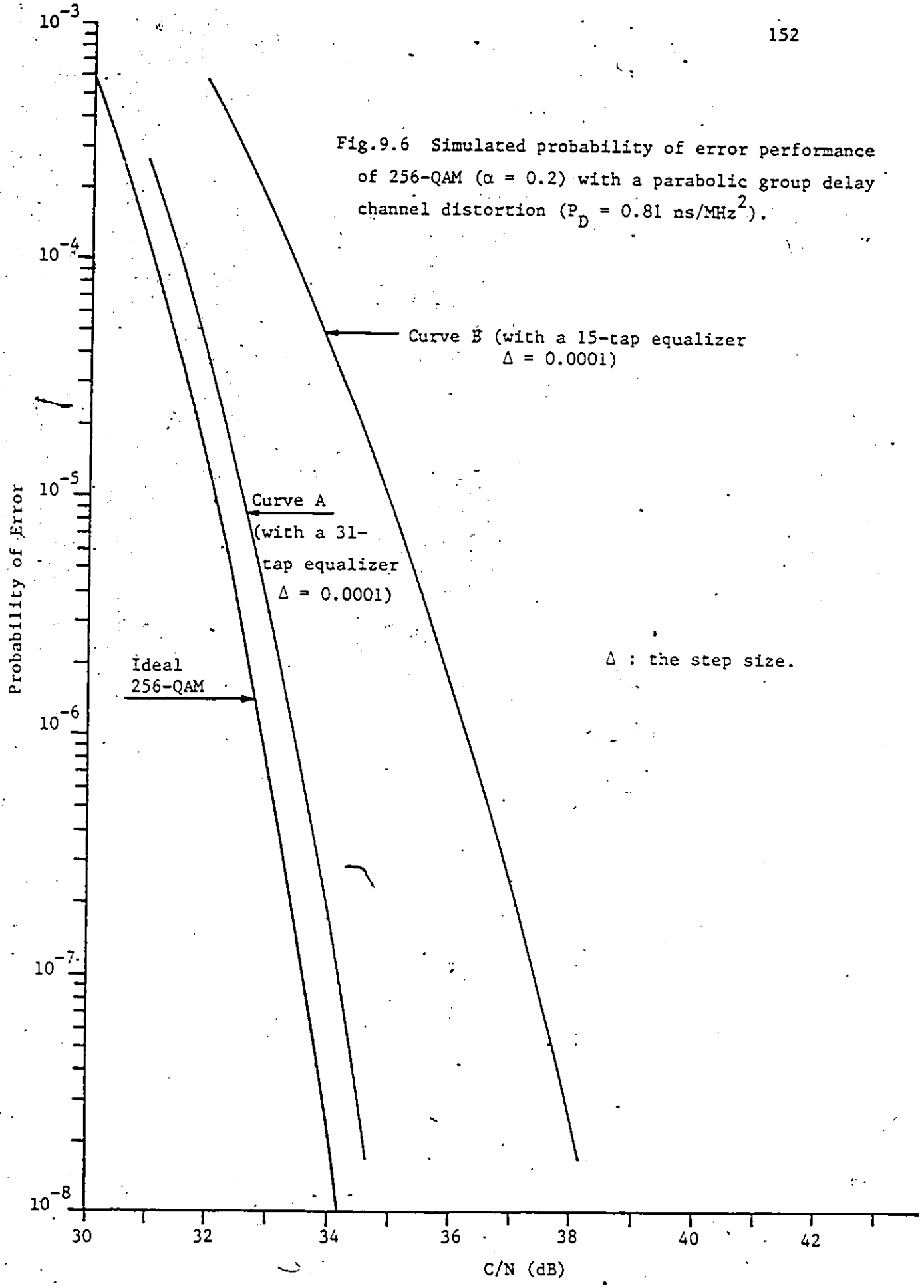


Fig.9.6 Simulated probability of error performance of 256-QAM ($\alpha = 0.2$) with a parabolic group delay channel distortion ($P_D = 0.81$ ns/MHz^2).

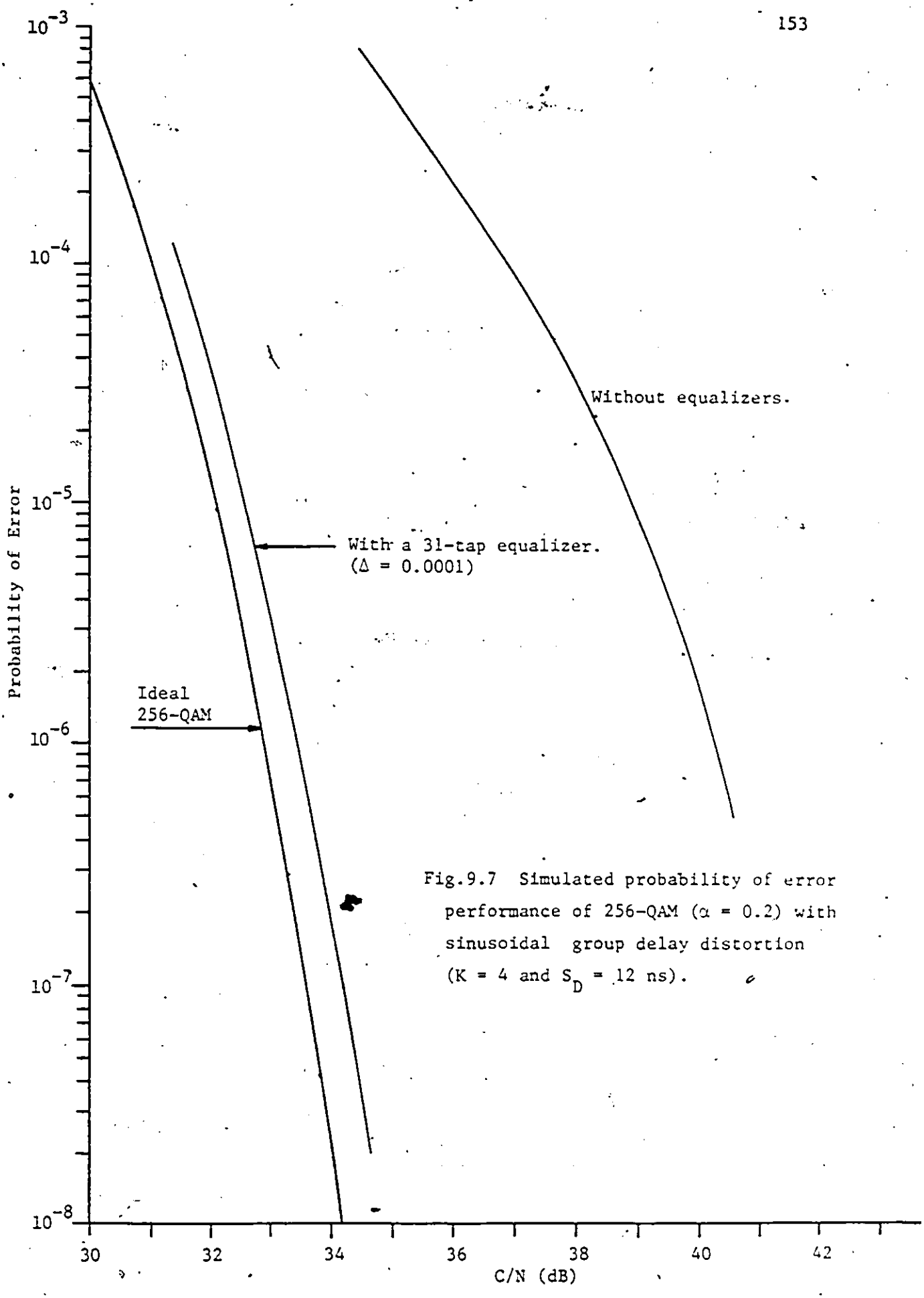


Fig.9.7 Simulated probability of error performance of 256-QAM (α = 0.2) with sinusoidal group delay distortion (K = 4 and S_D = 12 ns).

A. For the sinusoidal group delay distortion, the probability of error performance is given in Fig.9.7. Again we see that the equalizer works very well in significantly reducing the residual ISi to a negligible amount.

9.3 CONCLUSIONS

A baseband tapped delay line adaptive equalizer for a 256-QAM system over distorted channels has been described and simulated. The results show that the adaptive equalizer easily compensates for transmission impairments and in turn improves the error rate performance significantly. It has to be mentioned that, in collaboration with Karkar Electronics, Inc., San Francisco, we credited permission to use and modify their adaptive equalizer simulation programs for thesis. However, these programs are not to be listed in the thesis.

Chapter X

CONCLUSION AND RECOMMENDED FURTHER RESEARCH

10.1 SUMMARY

Motivated by the increasing need for efficient utilization of spectrum in the digital radio world and in hybrid analog and digital channels, such as the transmission of a T1 carrier in a standard analog 5G band, we have investigated two potential modulation schemes, i.e., 256-QAM and 225-QPRS. In order to achieve a spectral efficiency of more than 6 b/s/Hz by means of 225-QPRS, it is essential to operate it above the Nyquist rate, leading to the study of multi-level PRS/QPRS above the Nyquist rate. We found that the conventional method for multi-level PRS/QPRS above the Nyquist rate is inappropriate since it introduces too much "undesired" ISI, resulting in serious performance degradations. Therefore, we proposed an improved efficiency PRS/QPRS data filtering strategy ("chopping technique"), which enables a higher transmission rate than feasible with conventional PRS/QPRS systems. We demonstrated that, with the use of our new method, an increase of 5% above the Nyquist rate for class-I 225-QPRS, i.e., a spectral efficiency of 6.3 b/s/Hz can be achieved with a small degradation.

We also demonstrated that our new chopping technique can be applied to class-IV PRS systems which are useful in applications where reduced low-frequency components in the spectrum are required. The

advantages of combining class-IV PRS with our new technique are that a small bandwidth around dc can be carved out for accommodating a service channel and that a higher increase in spectral efficiency can be obtained in an FDM SSB system.

To confirm the concept of our new chopping technique for PRS above the Nyquist rate and the accuracy of computer simulations, an experimental 7-level class-IV PRS system was designed and implemented. System design principles and detailed description of circuit operations were presented. Measured BER performance was shown to agree well with the simulated results.

In order to enable one to select a suitable modulation scheme which is less sensitive to certain distortions, a quantitative evaluation of the effect of channel group delay and amplitude distortions with a linear, parabolic or sinusoidal characteristic on the performance of 256-QAM and 225-QPRS was performed with the aid of computer simulations. The numerical results could be useful for modem designers and researchers since it could indicate how flat and how linear their filter amplitude and phase responses should be.

In Chapter 7, we introduced a method to predict error floors in digital transmission systems. We compared the error floors caused by signalling above the Nyquist rate for duobinary PRS using the conventional method and our new scheme. The results show that the error floors caused by using our new scheme are much much lower than those by the conventional method. This rigorously verifies that our new scheme is better than the conventional method for multi-level PRS/QPRS above

the Nyquist rate for a practical realization. The method could be extended to the study of other transmission systems, as illustrated by an example which can evaluate error floors for PAM systems under severe distortions.

Since co-channel interference (CCI) is one of the major sources of performance impairments, we also evaluated the performance degradation caused by CCI for 64-QAM, 225-QPRS, and 256-QAM. Both sinusoidal and amplitude modulated CCI's were considered. The results indicate that, to make dual polarization operation feasible for high level modulation schemes, such as 225-QPRS and 256-QAM, a cross polarization discrimination of better than 40 dB is required since these systems are also sensitive to other impairments.

Finally, in order to improve the performance of a 256-QAM modem degraded by channel distortions as previously investigated, the use of an adaptive baseband transversal equalizer has been described and simulated. Illustrative eye diagrams and probability of error performance show that the adaptive equalizer easily compensates for transmission impairments and thus improve the error rate performance significantly.

10.2 SUGGESTIONS FOR FURTHER RESEARCH

In this final section, a few interesting research topics, which are related to our study, will be presented for further research. A brief description of these topics follows.

10.2.1 Carrier and Symbol Timing Recovery for 256-QAM and 225-QPRS

A perfect synchronization information was assumed to be available in the receiver in this thesis for the analysis and simulation of QAM and QPRS systems in an additive white Gaussian noise environment. High-level modulation schemes, such as 256-QAM and 225-QPRS require precise synchronization. Thus, it is important to study performance degradations caused by imperfect synchronization and methods to precisely recover the carrier and symbol timing.

10.2.2 Effects of Nonlinear Amplifiers on the Performance of 256-QAM and 225-QPRS

In this thesis, only linear distortions were considered. In a practical system, it is necessary to use power amplifiers, e.g., TWT or GaAs FET, at the output stage. These amplifiers exhibit a certain amount of nonlinearity [55], thus degrading the system's performance. The effect of nonlinearity on the performance of 256-QAM and 225-QPRS should be evaluated. Also the linearization of these power amplifiers could be another interesting and important topic [56].

10.2.3 The Performance of 256-QAM and 225-QPRS in an Adjacent Channel Interference (ACI) Environment

The performance of 256-QAM and 225-QPRS in a CCI environment has been evaluated in this thesis. In a multi-channel environment, ACI is also one of the major sources of performance impairment. Performance degradations caused by ACI on 256-QAM and 225-QPRS are thus worth studying.

10.2.4 Tailing Off BER Curves in QAM and QPRS Systems

It is often noted that a practically measured BER curve shows tailing off at high S/N. We have introduced in Chapter 7 a method to predict error floors for PRS systems operated above the Nyquist rate. A method very similar to ours has been presented in [41] at the same time to estimate the signature of QAM systems. This approach could be explored more and extended to investigate the tailing-off problems for QAM and QPRS systems caused by other impairments.

10.2.5 The Performance of 256-QAM and 225-QPRS in a Multipath Fading Environment

In the T1/SG DIV application example, as mentioned in Chapter 1, due to the narrow bandwidth, selective fading presents no problem. However, for high-speed digital radio applications, the effect of multipath (selective) fading should be considered [57]. Thus it is of interest to investigate the performance of 256-QAM and 225-QPRS in a multipath fading environment. This would in turn allow the outage of a system to be accurately estimated [58]. And it is of even more importance to study how equalization techniques can be employed to reduce the outage of these systems under selective fading [59], [60].

Appendix A

DERIVATION OF THE P_e PERFORMANCE OF CLASS-I AND CLASS-IV PRS SYSTEMS

The performance in terms of the probability of error of class-I and class-IV PRS systems over a white Gaussian noise channel will be evaluated. A symbol-by-symbol detection procedure is assumed. The model of the communications system is shown in Fig.A.1.

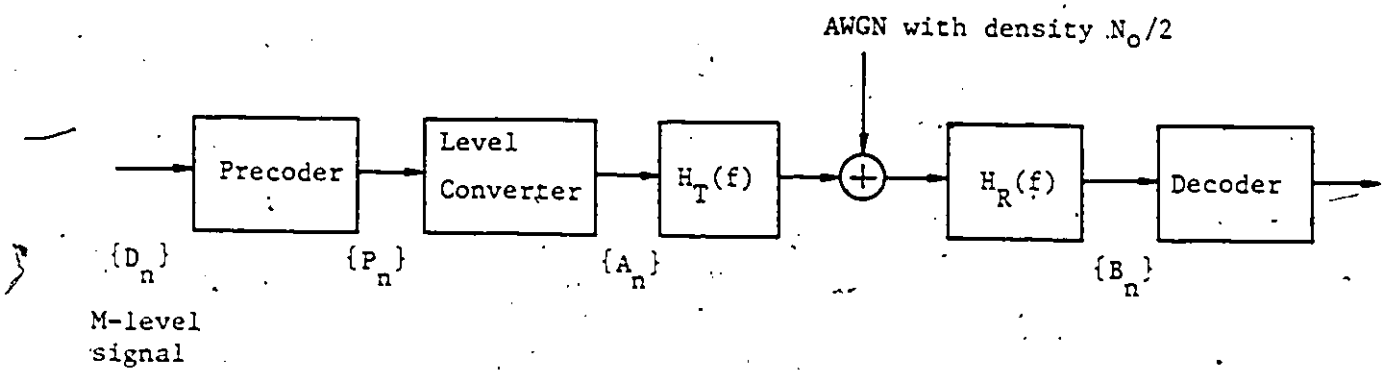


Fig.A.1 Block diagram of PRS systems.

Let $\{D_n\}$ denote the M-level input data sequence consisting of the equally likely numbers 0, 1, 2,..., M-1, and let $\{P_n\}$ denote the output of the precoder which is used to prevent error propagation. The relationship between $\{P_n\}$ and $\{D_n\}$ is

$$P_n = (D_n - P_{n-1}) \text{ Mod } M \quad \text{for class-I PRS} \tag{A.1}$$

and

$$P_n = (D_n + P_{n-2}) \text{ Mod } M \quad \text{for class-IV PRS} \tag{A.2}$$

where Mod M represents the modulo M arithmetic operation e.g., (-3) Mod 4 = 1, 7 Mod 4 = 3, etc.

The precoded symbol P_n is converted into the transmitted amplitude level A_n according to following relation.

$$A_n = 2P_n - (M-1) \tag{A.3}$$

The transmit filter with transfer function $H_T(f)$ is excited at the Nyquist rate by impulses having as areas the sequence of amplitudes $\{A_n\}$:

We shall first concentrate on class-I PRS. For class-I PRS, the overall transfer function X(f) is expressed as (A.4).

$$X(f) = H_T(f)H_R(f) = \begin{cases} 2T_s \cos(\pi f T_s) & \text{for } |f| \leq 1/2T_s, \\ 0 & \text{elsewhere.} \end{cases} \tag{A.4}$$

where T_s is the duration of a signaling interval. The Nyquist frequency is defined as

$$f_N = 1/2T_s \tag{A.5}$$

In the absence of noise, due to the class-I PRS operation, the received signal just before the decoder at the optimum sampling instants can be expressed as

$$B_n = A_n + A_{n-1} \tag{A.6}$$

The rule for converting the received sequence $\{B_n\}$ into the original data sequence $\{D_n\}$ is obtained as follows.

$$B_n = 2(P_n + P_{n-1} - (M-1)) \tag{A.7}$$

But
$$D_n = (P_n + P_{n-1}) \text{ Mod } M \tag{A.8}$$

Hence
$$D_n = (B_n/2 + (M-1)) \text{ Mod } M \tag{A.9}$$

Since the input symbols are assumed to be equally probable, $\{A_n\}$ thus has a uniform probability distribution, i.e.,

$$\Pr\{A_n=k\} = 1/M, \quad k= -(M-1), -(M-3),\ldots,-1, 1, \ldots, (M-3), (M-1). \tag{A.10}$$

From (A.6), we see that the probability density function (pdf) of B_n is the convolution of the two identical uniform pdf's shown as (A.10). Thus, it is easily shown that in the absence of noise, the output levels have a triangular probability distribution, i.e.,

$$\Pr(B_n = 2ld) = \frac{M-|l|}{M^2} \qquad l = -(M-1), -(M-2), \ldots, 0, 1, 2, \ldots, M-1. \tag{A.11}$$

where B_n denotes the received level in the n-th signaling interval and 2d is the distance between any two adjacent received levels.

The actual received sample (signal plus noise), call it y, is Gaussian with mean $E(y) = B_n = 2ld$ and variance σ^2,

where
$$\sigma^2 = \frac{N_0}{2}\int_{-\infty}^{\infty} |H_R(f)|^2 \, df \tag{A.12}$$

Thus a tight bound on the average probability of a symbol error, denoted by P_M, can be derived as follows.

$$P_M \le \sum_{l=-(M-2)}^{M-2} \Pr(|y-2ld|>d|B_n=2ld)\cdot\Pr(B_n=2ld)$$
$$+ 2\Pr[y+2(M-1)d>d|B_n=-2(M-1)d]\cdot\Pr[B_n=-2(M-1)d]$$
$$= \Pr(|y|>d|B_n=0)\left\{2\sum_{l=0}^{M-1}\Pr(B_n=2ld)-\Pr(B_n=0)\right.$$
$$\left.-\Pr[B_n=-2(M-1)d]\right\}$$
$$= \left(1-\frac{1}{M^2}\right)\Pr(|y|>d|B_n=0) \qquad (A.13)$$

But

$$\Pr(|y|>d|B_n=0)=\frac{2}{\sqrt{2\pi}\sigma}\int_d^\infty e^{-x^2/2\sigma^2}dx = 2Q(d/\sigma), \qquad (A.14)$$

where

$$Q(x)=\frac{1}{\sqrt{2\pi}\sigma}\int_x^\infty \exp(-t^2/2)\,dt$$

Hence, substituting (A.14) into (A.13), we obtain

$$P_M \le 2(1-1/M^2)\,Q(d/\sigma) \qquad (A.15)$$

That the sign "<" appears in the above equations is because of the "Mod M" decoding (see (A.9)). For example, let M = 4 and suppose due to noise a level of 3 has been driven to -1. Note that (-1) Mod 4 = 3, i.e., in this case no error is observed. However due to the nature of Gaussian noise, such cases occur at relatively low probability. Thus (A.14) is a tight bound.

It is necessary to express P_M in terms of the average S/N at the input of the receiver measured in the Nyquist bandwidth ($\frac{1}{2T_s}$). To do so, the distribution of filters is important and must be clarified.

A.1 OPTIMALLY SPLIT FILTERS

Optimum refers to splitting the overall class-I PRS shaping equally between the transmitter and the receiver, i.e.,

$$H_T(f) = H_R(f) = \begin{cases} (2T_s \cos \pi f T_s)^{\frac{1}{2}} & |f| \le 1/2T_s, \\ 0 & \text{elsewhere.} \end{cases} \tag{A.16}$$

In this case, the received noise power σ^2 is

$$\sigma^2 = \frac{N_o}{2} \int_{-\infty}^{\infty} |H_R(f)|^2 \, df$$

$$= N_o T_s \int_{-1/2T_s}^{1/2T_s} \cos(\pi f T_s) \, df = 2N_o/\pi \tag{A.17}$$

Denote by N the noise power at the input of the receiver measured in the Nyquist bandwidth $(\frac{1}{2T_s})$.

$$N = \int_{-1/2T_s}^{1/2T_s} \frac{N_o}{2} \, df = N_o/2T_s = \pi \sigma^2/(4T_s) \tag{A.18}$$

The transmitted power, denoted by S, is calculated as follows.

$$S = \frac{\overline{k^2}}{T_s} \int_{-1/2T_s}^{1/2T_s} |H_T(f)|^2 \, df = 4 \overline{k^2}/(\pi T_s) \tag{A.19}$$

where $\overline{k^2}$ is the mean-square value of the transmitted levels. For the case of equally likely transmitted levels with the adjacent levels separated by a distance of 2d,

$$\overline{k^2} = d^2 (M^2 - 1)/3 \tag{A.20}$$

Substituting (A.18) - (A.20) into (A.15), we obtain P_M in terms of S/N. That is

$$P_M \leq 2(1 - 1/M^2)Q\left(\sqrt{\frac{3}{M^2 - 1}\left(\frac{\pi}{4}\right)^2 \frac{S}{N}}\right) \qquad (A.21)$$

Instead of the "Q" function, it is often seen in the literature that some authors used the "erfc" function. Thus (A.21) can be expressed also as

$$P_M \leq (1 - 1/M^2)\ \mathrm{erfc}\left(\sqrt{\frac{3}{2(M^2 - 1)}\left(\frac{\pi}{4}\right)^2 \frac{S}{N}}\right) \qquad (A.22)$$

where the relation $\mathrm{erfc}(y) = 2Q(\sqrt{2}\,y)$ has been used.

A.2 . SUBOPTIMUM DISTRIBUTION OF FILTERS

By suboptimum, we mean that the class-I PRS shaping is fully at the transmitter, whereas the receiver simply bandlimits the noise. That is

$$H_T(f) = \begin{cases} 2\,T_s \cos(\pi f T_s) & |f| \leq 1/2T_s \\ 0 & \text{elsewhere.} \end{cases}$$

and (A.23)

$$H_R(f) = \begin{cases} 1 & |f| \leq 1/2T_s \\ 0 & \text{elsewhere.} \end{cases}$$

The transmitted power S is

$$S = \frac{k^2}{T_s}\int_{-\infty}^{\infty} |H_T(f)|^2\, df$$

$$S = 8T_s \overline{k^2} \int_0^{1/2T_s} \cos(\pi f T_s)^2 \, df$$

$$= 2\,\overline{k^2} = 2\,d^2\,(M^2 - 1)/3 \qquad (A.24)$$

In this case, the noise power at the input of the receiver measured in the Nyquist bandwidth is the same as that of the output of the receive filter, i.e.,

$$N = \sigma^2 = \int_{-1/2T_s}^{1/2T_s} \frac{N_o}{2} \, df = N_o/2T_s \qquad (A.25)$$

Substituting (A.24) and (A.25) into (A.15), the following expression can be derived:

$$P_M \leq 2\,(1 - 1/M^2)\, Q\left(\sqrt{\frac{3}{M^2 - 1}\,\frac{1}{2}\,\frac{S}{N}}\right) \qquad (A.26)$$

or equivalently,

$$P_M \leq (1 - 1/M^2)\, \mathrm{erfc}\left(\sqrt{\frac{3}{M^2 - 1}\,\frac{1}{4}\,\frac{S}{N}}\right) \qquad (A.27)$$

Comparing the arguments of the "Q" functions in (A.26) and (A.21), we note that the suboptimum distribution of filters results in a loss of approximately

$$10 \log_{10} \frac{(\pi/4)^2}{1/2} = 0.91 \text{ dB} \qquad (A.28)$$

in S/N relative to the case of optimally split filters.

It has to be noted that if the S/N is measured at the output of the receive filter, i.e., just before the decoder, the two filters distribution strategies (optimum and suboptimum) result in the same P_M expression as (A.26).

P_M for class-IV PRS can be derived similarly. For the class-IV PRS shaping,

$$X(f) = H_T(f)H_R(f) = \begin{cases} j2T_s \sin(2\pi f T_s), & |f| \leq 1/2T_s \\ 0 & \text{elsewhere.} \end{cases} \tag{A.29}$$

In the absence of noise, the received signal at the optimum sampling instants can be represented as

$$B_n = A_n - A_{n-2} \tag{A.30}$$

To obtain the rule for converting the received sequence $\{B_n\}$ into the data sequence $\{D_n\}$ we substitute for A_n.

Thus, $$B_n = 2\,(P_n - P_{n-2}) \tag{A.31}$$

But $$D_n = (P_n - P_{n-2}) \text{ Mod } M \tag{A.32}$$

Hence, $$D_n = (B_n/2) \text{ Mod } M \tag{A.33}$$

Following the above procedures, it is very straightforward to obtain the P_M for class-IV PRS and the results are found to be exactly identical to (A.21) and (A.26). This can be understood by noting that the areas under class-I and class-IV PRS shapes are the same. That is

$$\int_{-1/2T_s}^{1/2T_s} |2T_s \cos(\pi f T_s)|^2 \, df \qquad \text{(Area under class-I PRS shape)}$$

$$= \int_{-1/2T_s}^{1/2T_s} |j2T_s \sin(2\pi f T_s)|^2 \, df \qquad \text{(... class-IV PRS ...)}$$

$$= 4/\pi. \qquad \text{(A.34)}$$

Appendix B

SIMULATION PROGRAMS FOR QAM AND QPRS SYSTEMS

The simulation programs listed in this appendix were used to simulate various performances of QAM and QPRS systems, such as the performance of QPRS above the Nyquist rate, QAM and QPRS systems' performances in distorted channels, the impact of CCI on QAM and QPRS systems, etc. The author of this thesis used, modified, and extended numerous subroutines and programs which were developed by the Digital Communications Research Group, Department of Electrical Engineering, University of Ottawa.

FILE: DIGIMP FORTRAN A UNIV D'/OF OTTAWA

```
C*****************************************************************      DIG00010
C       FILE NAME : DIGIMP FORTRAN                                      DIG00020
C       DRAWS THE DISCRETE IMPULSE RESPONSE OF PRS ABOVE THE            DIG00030
C       NYQUIST RATE USING MODEL-A OR B.                                DIG00040
C       ONLY THE ABSOLUTE VALUES OF IMPULSE RESPONSE SAMPLES            DIG00050
C       ARE PLOTTED.                                                    DIG00060
C                                                                       DIG00070
C       PARAMETERS: MODEL=1 FOR CONVENTIONAL MODEL-A.                   DIG00080
C                   MODEL=2 FOR OUR NEW MODEL-B.                        DIG00090
C                   B IS THE PERCENTAGE INCREASE.                       DIG00100
C                   FOR EXAMPLE, B=.05 MEANS A 5% INCREASE.             DIG00110
C*****************************************************************      DIG00120
      DIMENSION XA(153),YA(153)                                         DIG00130
      MODEL=2                                                           DIG00140
      B=0.05                                                            DIG00150
      A=1./(1.+B)                                                       DIG00160
      X=-50.5                                                           DIG00170
      PI=3.1415926                                                      DIG00180
      GO TO (1,2), MODEL                                                DIG00190
    1 DO 100 I=1,101                                                    DIG00200
      Y=PI*(A*X+0.5)                                                    DIG00210
      Z=PI*(A*X-0.5)                                                    DIG00220
      IF((Z*Y).EQ.0.) GO TO 33                                          DIG00230
      HX=SIN(Y)/Y+SIN(Z)/Z                                              DIG00240
      GO TO 55                                                          DIG00250
   33 HX=1.                                                             DIG00260
   55 YA(I)=HX                                                          DIG00270
      XA(I)=X                                                           DIG00280
      X=X+1.0                                                           DIG00290
  100 CONTINUE                                                          DIG00300
      GO TO 300                                                         DIG00310
    2 DO 200 I=1,101                                                    DIG00320
      Y=PI*A*(X+0.5)                                                    DIG00330
      Z=PI*A*(X-0.5)                                                    DIG00340
      IF(Y.EQ.0.) HX=A+A*SIN(Z)/Z                                       DIG00350
      IF(Z.EQ.0.) HX=A*SIN(Y)/Y+A                                       DIG00360
      IF((Y*Z).EQ.0.) GO TO 45                                          DIG00370
      HX=A*SIN(Y)/Y+A*SIN(Z)/Z                                          DIG00380
   45 YA(I)=HX                                                          DIG00390
      XA(I)=X                                                           DIG00400
      X=X+1.0                                                           DIG00410
  200 CONTINUE                                                          DIG00420
  300 WRITE(6,80) MODEL,A                                               DIG00430
   80 FORMAT(10X,'MODEL',I2,'  A=',F8.4,//)                             DIG00440
      WRITE(6,85)                                                       DIG00450
   85 FORMAT(13X,' I ',13X,' X ',15X,'IMPULSE SAMPLES H(X)',/)          DIG00460
      WRITE(6,90) (I,XA(I),YA(I),I=1,101)                               DIG00470
   90 FORMAT(10X,I5,10X,F10.5,10X,F10.5)                                DIG00480
      CALL DRAW(XA,YA)                                                  DIG00490
      STOP                                                              DIG00500
      END                                                               DIG00510
      SUBROUTINE DRAW(XA,YA)                                            DIG00520
      DIMENSION X(4),Y(4),YA(1),XA(1)                                   DIG00530
C     ESTABLISH THE SURFACE AREA.                                       DIG00540
      CALL PLOTS(30.0,27.5)                                             DIG00550
```

FILE: DIGIMP FORTRAN A UNIV D'/OF OTTAWA

```
C     CALL FACTOR(0.75)                                                  DIG00560
C     ESTABLISH THE ORIGIN.                                              DIG00570
      CALL PLOT(3.0,6.0,-3)                                              DIG00580
C     WRITE THE TITLE OF THE GRAPH.                                      DIG00590
C     CALL SYMBOL(10.0,10.0,0.49,16HQPRS EYE DIAGRAM,0.0,16)             DIG00600
C     DRAW THE TIME AXIS.                                                DIG00610
      CALL AXIS(0.0,0.0,1H ,-1,21.,0.0,-50.5,5.0)                        DIG00620
C     DRAW THE AMPLITUDE AXIS.                                           DIG00630
      DIV=0.05/15.                                                       DIG00640
C     DIV=0.1/15.  FOR B=0.10                                            DIG00650
      CALL AXIS(0.0,0.0,1H ,1,16.,90.0,0.0,DIV)                          DIG00660
C     PLOT THE DATA.                                                     DIG00670
      JJ=2                                                               DIG00680
      Y(JJ+1)=0.0                                                        DIG00690
      Y(JJ+2)=DIV                                                        DIG00700
      X(JJ+1)=-50.5                                                      DIG00710
      X(JJ+2)=5.0                                                        DIG00720
      DO 100 I=1,101                                                     DIG00730
      X(1)=XA(I)                                                         DIG00740
      X(2)=X(1)                                                          DIG00750
      Y(1)=0.                                                            DIG00760
      Y(2)=YA(I)/YA(51)                                                  DIG00770
      Y(2)=ABS(Y(2))                                                     DIG00780
C     IF(Y(2).GT.0.1) Y(2)=0.1 FOR B=0.10                                DIG00790
      IF(Y(2).GT.0.05) Y(2)=0.05                                         DIG00800
      CALL LINE(X,Y,JJ,1,0,0)                                            DIG00810
  100 CONTINUE                                                           DIG00820
      CALL PLOT(0.0,0.0,999)                                             DIG00830
      RETURN                                                             DIG00840
      END                                                                DIG00850
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
C*****************************************************************      25600010
C     PROGRAM FOR SIMULATING 256-QAM IN LINEAR CHANNEL WITH LINEAR,      25600020
C     PARABOLIC, AND SIN/COSINE RIPPLE FILTER DISTORTIONS.               25600030
C     FILE NAME: 256GD                                                   25600040
C     FOR AMPLITUDE DISTORTIONS, SET ID=1 AND USE SUB. AMPD.             25600050
C     FOR GD DISTORTIONS,        SET ID=2 AND USE SUB. DLYD.             25600060
C     SOME PARAMETERS IN SUB. DRAWCN MUST BE MODFIED IN                  25600070
C     ACCORDANCE WITH THOSE IN SUB. AMPD OR DLYD.                        25600080
C     THE NOMINAL BIT RATE IS 120 MB/S, BUT IT CAN BE                    25600090
C     CHANGED TO ANOTHER BIT RATE BY CHANGING THE PARAMETER BR.          25600100
C*****************************************************************      25600110
      COMPLEX DATA(65536),TF(65536)                                      25600120
      DIMENSION PEI(32),EBNO(32),NI(2048),NQ(2048)                       25600130
      DIMENSION XARRAY(32),YARRAY(32)                                    25600140
C     INITIALIZATION                                                     25600150
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                          25600160
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                     25600170
      ALPHA=0.4                                                          25600180
      LDIM=65536                                                         25600190
      IOFF=0                                                             25600200
      LSAMPL=32                                                          25600210
      NSNR=32                                                            25600220
      NSYMB=2047                                                         25600230
      BR=120.                                                            25600240
      WRITE(6,19) BR                                                     25600250
   19 FORMAT(10X,'BIT RATE =',F8.3,' MB/S')                              25600260
      BAUD=15.*(BR/120.)                                                 25600270
      FBW=BAUD/2.                                                        25600280
      ISIN=1                                                             25600290
      NFILTR=2                                                           25600300
      NRUNS=7                                                            25600310
      WRITE(6,666) ALPHA                                                 25600320
  666 FORMAT(10X,'ALPHA=',F8.4)                                          25600330
C=================================                                     25600340
C ID=1 FOR AMPLITUDE DISTORTIONS                                         25600350
C ID=2 FOR GD DISTORTIONS                                                25600360
C ID WILL BE USED IN SUB. DRAWCN                                         25600370
C---------------------------------                                     25600380
      ID=1                                                               25600390
C     END OF INITIALIZATION.                                             25600400
                                                                         25600410
C     START OF COMPUTATIONS.                                             25600420
      DO 100 NR=1,NRUNS                                                  25600430
C=================================                                     25600440
      CALL LD256(DATA,NI,NQ)                                             25600450
C     CALL LOAD64(DATA,NI,NQ)                                            25600460
C     DO 43 I=1,LDIM                                                     25600470
C  43 DATA(I)=2.*DATA(I)                                                 25600480
C---------------------------------                                     25600490
      CALL RCOSTX(TF)                                                    25600500
C=================================                                     25600510
      GO TO (31,32),ID                                                   25600520
   31 CALL AMPD(NR,TF,ISIN)                                              25600530
      GO TO 33                                                           25600540
   32 CALL DLYD(NR,TF,ISIN)                                              25600550
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
C--------------------------------                                    25600560
   33 CALL FILTER(DATA,TF)                                          25600570
      CALL ENERGY(DATA,EB)                                          25600580
C                                                                   25600590
C  .   END OF TRANSMITTER                                           25600600
C                                                                   25600610
      CALL RCOSRX(TF)                                               25600620
      CALL HHGG(TF,PNOISE)                                          25600630
      CALL FILTER(DATA,TF)                                          25600640
      CALL SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)                       25600650
      CALL DECODE(DATA,PNOISE,MI,MQ,NI,NQ,                          25600660
     #EBNO,PEI,EB)                                                  25600670
C                                                                   25600680
C     CALL THE DRAWING ROUTINE                                      25600690
C                                                                   25600700
      DO 99 I=1,NSNR                                                25600710
      XARRAY(I)=EBNO(I)                                             25600720
   99 YARRAY(I)=PEI(I)                                              25600730
      CALL DRAWCN(ID,XARRAY,YARRAY,NR,NRUNS)                        25600740
C                                                                   25600750
C     CALL THE EYE DIAGRAM ROUTINE                                  25600760
C                                                                   25600770
C     CALL EYEQ(DATA)                                               25600780
  100 CONTINUE                                                      25600790
      STOP                                                          25600800
      END                                                           25600810
C*************************************************************      25600820
C     THIS SUBROUTINE IS TO SIMULATE GROUP DELAY DISTORTIONS        25600830
C     I/P : NR                                                      25600840
C           ISIN                                                    25600850
C           TF                                                      25600860
C     O/P : TF                                                      25600870
C*************************************************************      25600880
      SUBROUTINE DLYD(NR,TF,ISIN)                                   25600890
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                25600900
      COMPLEX TF(1)                                                 25600910
      GO TO (84,2,3,4,5,6,7),NR                                     25600920
    2 B=.095*(15./BAUD)**2                                          25600930
      GO TO 10                                                      25600940
    3 B=.19*(15./BAUD)**2                                           25600950
   10 WRITE (6,11) B                                                25600960
   11 FORMAT.(10X,'LINEAR GROUP DELAY IN NS/MHZ=',F8.5)             25600970
      CALL LINGD(B,TF)                                              25600980
      GO TO 84                                                      25600990
    4 S=0.27*(15./BAUD)**3                                          25601000
      GO TO 20                                                      25601010
    5 S=0.54*(15./BAUD)**3                                          25601020
   20 WRITE (6,12) S                                                25601030
   12 FORMAT (10X,'PARABOLIC GROUP DELAY IN NS/MHZ**2=',F8.5)       25601040
      CALL PARGD(S,TF)                                              25601050
      GO TO 84                                                      25601060
    6 C=6.00*(15./BAUD)                                             25601070
      GO TO 30                                                      25601080
    7 C=12.0*(15./BAUD)                                             25601090
   30 K=4                                                           25601100
```

FILE: 256GD .FORTRAN A UNIV D'/OF OTTAWA

```
      WRITE  (6,17) K,C                                                  25601110
   17 FORMAT (5X,'# OF RIPPLES=',I3,'SINE AMPLITUDE GROUP DELAY=',       25601120
     #F8.5,/)                                                            25601130
      CALL SINCGD(ISIN,C,K,TF)                                           25601140
   84 RETURN                                                             25601150
      END                                                                25601160
C*******************************************************************    25601170
C     THIS SUBROUTINE IS TO SIMULATE AMPLITUDE DISTORTIONS              25601180
C     I/P : NR                                                          25601190
C           ISIN                                                        25601200
C           TF                                                          25601210
C     O/P : TF                                                          25601220
C*******************************************************************    25601230
      SUBROUTINE AMPD(NR,TF,ISIN)                                        25601240
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                     25601250
      COMPLEX TF(1)                                                      25601260
      GO TO (74,2,3,4,5,6,7),NR                                          25601270
    2 Z=.038*(15./BAUD)                                                  25601280
      GO TO 10                                                           25601290
    3 Z=0.076*(15./BAUD)                                                 25601300
   10 WRITE (6,8) Z                                                      25601310
    8 FORMAT (10X,'LINEAR AMPLITUDE SLOPE IN DB/MGZ=',F8.5,/)            25601320
      CALL LINAMP(Z,TF)                                                  25601330
      GO TO 74                                                           25601340
    4 P=.00400*(15./BAUD)**2                                             25601350
      GO TO 20                                                           25601360
    5 P=.00800*(15./BAUD)**2                                             25601370
   20 WRITE(6,9) P                                                       25601380
    9 FORMAT(10X,'PARABOLIC AMPLITUDE DIST. IN DB/MHX**2=',F8.5./)       25601390
      CALL PARAMP(P,TF)                                                  25601400
      GO TO 74                                                           25601410
    6 D=0.2                                                              25601420
      GO TO 30                                                           25601430
    7 D=.4                                                               25601440
   30 K=4                                                                25601450
      WRITE(6,77) K,D                                                    25601460
   77 FORMAT(5X,'# OF RIPPLES=',I3,'SINE AMPLITUDE DIST. IN DB=',        25601470
     #F8.5,/)                                                            25601480
      CALL SINAMP(ISIN,D,K,TF)                                           25601490
   74 RETURN                                                             25601500
      END                                                                25601510
C*******************************************************************    25601520
C    RAISED COSINE FILTER WITH ARBITRARY ALPHA AND X/SINX               25601530
C*******************************************************************    25601540
      SUBROUTINE RCOSTX(TF)                                              25601550
      COMPLEX TF(65536)                                                  25601560
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                          25601570
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                     25601580
      SBANDW=BAUD*LSAMPL                                                 25601590
      NO=LDIM/2                                                          25601600
      NO1=NO+1                                                           25601610
      IF (ALPHA.EQ.0) ALPHA=0.0001                                       25601620
      FN=LDIM*(FBW/SBANDW)                                               25601630
      F1=(1.-ALPHA)*FN                                                   25601640
      F2=(1.+ALPHA)*FN                                                   25601650
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
      IFN=IFIX(FN)                                                      25601660
      IF1=IFIX(F1)+1                                                    25601670
      IF2=IFIX(F2)+1                                                    25601680
C                                                                       25601690
C     THE AMPLITUDE CHARACTERISTICS                                     25601700
C                                                                       25601710
       A1=3.1415926/(2.*FLOAT(IFN))                                     25601720
       DO 8 I=2,IF1                                                     25601730
      TF(1)=CMPLX(1.0,0.0)                                              25601740
       J=I-1                                                            25601750
C                                                                       25601760
C     X/SIN(X) EQUALIZATION                                             25601770
C                                                                       25601780
       A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))                              25601790
      TF(I)=CMPLX(A2,0.0)                                               25601800
    8 CONTINUE                                                          25601810
      JK=IF1+1                                                          25601820
      DO 9 J=JK,IF2                                                     25601830
       I=J-1                                                            25601840
C                                                                       25601850
C     ROOT OF RAISED COSINE                                             25601860
C                                                                       25601870
      A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))                               25601880
      A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)              25601890
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)                           25601900
       TF(J)=TF(J)*CMPLX(A3,0.0)                                        25601910
    9 CONTINUE                                                          25601920
      JH=IF2+1                                                          25601930
      DO 10 I=JH,NO1                                                    25601940
      TF(I)=CMPLX(0.0,0.0)                                              25601950
   10 CONTINUE                                                          25601960
        NO2=NO1+1                                                       25601970
      DO 5 I=NO2,LDIM                                                   25601980
      TF(I)=CONJG(TF(LDIM+2-I))                                         25601990
    5 CONTINUE                                                          25602000
      RETURN                                                            25602010
      END                                                               25602020
C*****************************************************************     25602030
C     RAISED COSINE FILTER WITH ARBITRARY ALPHA                         25602040
C*****************************************************************     25602050
      SUBROUTINE RCOSRX(TF)                                             25602060
      COMPLEX TF(65536)                                                 25602070
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25602080
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    25602090
      SBANDW=BAUD*LSAMPL                                                25602100
      NO=LDIM/2                                                         25602110
      NO1=NO+1                                                          25602120
      IF (ALPHA.EQ.0) ALPHA=0.0001                                      25602130
      FN=LDIM*(FBW/SBANDW)                                              25602140
      F1=(1.-ALPHA)*FN                                                  25602150
      F2=(1.+ALPHA)*FN                                                  25602160
      IFN=IFIX(FN)                                                      25602170
      IF1=IFIX(F1)+1                                                    25602180
      IF2=IFIX(F2)+1                                                    25602190
C                                                                       25602200
```

FILE: 256GD FORTRAN A UNIV D'YOF OTTAWA

```
C     AMPLITUDE CHARACTERISTIC                                          25602210
C                                                                       25602220
      A1=3.141592/(2.*FLOAT(IFN))                                       25602230
      DO 8 I=2,IF1                                                      25602240
      TF(1)=CMPLX(1.0,0.0)                                              25602250
      J=I-1                                                             25602260
      TF(I)=CMPLX(1.,0.0)                                               25602270
    8 CONTINUE                                                          25602280
      JK=IF1+1                                                          25602290
      DO 9 J=JK,IF2                                                     25602300
      I=J-1                                                             25602310
C                                                                       25602320
C     ROOT OF RAISED COSINE                                             25602330
C                                                                       25602340
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)               25602350
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)                           25602360
    9 CONTINUE                                                          25602370
      JH=IF2+1                                                          25602380
      DO 10 I=JH,NO1                                                    25602390
      TF(I)=CMPLX(0.0,0.0)                                              25602400
   10 CONTINUE                                                          25602410
      NO2=NO1+1                                                         25602420
      DO 5 I=NO2,LDIM                                                   25602430
      TF(I)=CONJG(TF(LDIM+2-I))                                         25602440
    5 CONTINUE                                                          25602450
      RETURN                                                            25602460
      END                                                               25602470
C*****************************************************************     25602480
C     THE FOLLOWING SUBROUTINE PERFORMS THE FILTERING                   25602490
C     PROCESS ON THE DATA SEQUENCE.                                     25602500
C*****************************************************************     25602510
      SUBROUTINE FILTER(DATA,TF)                                        25602520
      COMPLEX DATA(1),TF(1)                                             25602530
      DIMENSION IWK(17)                                                 25602540
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25602550
      CALL FFT2C(DATA,16,IWK)                                           25602560
      DO 1 I=1,LDIM                                                     25602570
    1 DATA(I)=CONJG(DATA(I)*TF(I))                                      25602580
      CALL FFT2C(DATA,16,IWK)                                           25602590
      DO 2 I=1,LDIM                                                     25602600
    2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)                                25602610
      RETURN                                                            25602620
      END                                                               25602630
C*****************************************************************     25602640
C     THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE (PNOISE)             25602650
C     AT THE OUTPUT OF THE RECEIVE FILTER                               25602660
C*****************************************************************     25602670
      SUBROUTINE HHGG(TF,PNOISE)                                        25602680
      COMPLEX TF(1)                                                     25602690
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25602700
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    25602710
      SUM=0.0                                                           25602720
      SBANDW=BAUD*LSAMPL                                                25602730
      DO 1 L=1,LDIM                                                     25602740
      HH=(CABS(TF(L)))**2                                               25602750
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
    1  SUM=SUM+HH                                                    25602760
       PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.                              25602770
       WRITE (6,2) PNOISE                                            25602780
    2  FORMAT(5X,'PNOISE=',F7.3,/)                                   25602790
       RETURN                                                        25602800
       END                                                           25602810
C*****************************************************************  25602820
C CALCULATE EB                                                       25602830
C*****************************************************************  25602840
       SUBROUTINE ENERGY(DATA,EB)                                    25602850
       COMPLEX DATA(1)                                               25602860
       WATTS=0.                                                      25602870
       COMMON /NUMB2/ NSNR,NSYMB,BAUD                                25602880
       COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                     25602890
       DO 1 I=1,LDIM                                                 25602900
       WATTS=WATTS+((CABS(DATA(I)))**2.)                             25602910
     1 CONTINUE                                                      25602920
       WATTS=WATTS/(FLOAT(LDIM))                                     25602930
       EB=WATTS/(8.*BAUD)                                            25602940
       WRITE (6,40) WATTS,EB                                         25602950
    40 FORMAT(10X,F15.6,5X,E15.8)                                    25602960
       RETURN                                                        25602970
       END                                                           25602980
C*****************************************************************  25602990
C     THIS SUBROUTINE GENERATES A BASEBAND 256-QAM SIGNAL.           25603000
C*****************************************************************  25603010
       SUBROUTINE LD256(DATA,NI,NQ)                                  25603020
       COMPLEX DATA(1)                                               25603030
       DIMENSION NY(11),NI(1),NQ(1),NX(2048)                         25603040
       COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                     25603050
C      DATA NX(1),NX(2),NX(3),NX(4),NX(5),NX(6),NX(7),NX(8),NX(9),   25603060
C     #NX(10),NX(11),NX(12)/-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1/     25603070
       DATA NY(1),NY(2),NY(3),NY(4),NY(5),NY(6),NY(7),NY(8),NY(9),   25603080
      #NY(10),NY(11)/-1,-1,1,1,1,1,-1,-1,-1,-1,-1,-1/                25603090
       DATA JLAST,JTAP/11,1/                                         25603100
       KKK=2**JLAST                                                  25603110
       KK=KKK-(1+JLAST)                                              25603120
       DO 6 I=1,JLAST                                                25603130
       NX(I)=NY(I)                                                   25603140
    6  CONTINUE                                                      25603150
C                                                                    25603160
C      GENERATE 2**JLAST -1 LENGTH SEQUENCE USING GIVEN DATA AND     25603170
C      GENERATOR POLYNOMIAL.                                         25603180
C                                                                    25603190
       DO 1 J=1,KK                                                   25603200
       I=J+JLAST                                                     25603210
       NX(I)=NX(J)*NX(J+2)                                           25603220
       NX(I)=0-NX(I)                                                 25603230
    1  CONTINUE                                                      25603240
C                                                                    25603250
C      GENERATE 8 CYCLICALLY SHIFTED VERSIONS OF ORIGINAL            25603260
C      SEQUENCE OF LENGTH 2**JLAST.                                  25603270
C                                                                    25603280
       K1=1                                                          25603290
       K2=2                                                          25603300
```

FILE: 256CD FORTRAN A UNIV D'/OF OTTAWA

```
      K3=3                                                              25603310
      K4=4                                                              25603320
      K5=5                                                              25603330
      K6=6                                                              25603340
      K7=7                                                              25603350
      K8=8                                                              25603360
      DO 2 I=1,KKK                                                      25603370
      IF (K1.EQ.KKK) K1=1                                               25603380
      IF (K2.EQ.KKK) K2=1                                               25603390
      IF (K3.EQ.KKK) K3=1                                               25603400
      IF (K4.EQ.KKK) K4=1                                               25603410
      IF (K5.EQ.KKK) K5=1                                               25603420
      IF (K6.EQ.KKK) K6=1                                               25603430
      IF (K7.EQ.KKK) K7=1                                               25603440
      IF (K8.EQ.KKK) K8=1                                               25603450
      NI(I)=NX(K1)+NX(K3)*2+NX(K5)*4+NX(K7)*8                           25603460
      NQ(I)=NX(K2)+NX(K4)*2+NX(K6)*4+NX(K8)*8                           25603470
C                                                                       25603480
C     LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL                     25603490
      J1=(I-1)*LSAMPL+1                                                 25603500
      J2=I*LSAMPL                                                       25603510
      DO 3 J3=J1,J2                                                     25603520
    3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))                         25603530
      K1=K1+1                                                           25603540
      K2=K2+1                                                           25603550
      K3=K3+1                                                           25603560
      K4=K4+1                                                           25603570
      K5=K5+1                                                           25603580
      K6=K6+1                                                           25603590
      K7=K7+1                                                           25603600
      K8=K8+1                                                           25603610
    2 CONTINUE                                                          25603620
      IF(IOFF.EQ.0) RETURN                                              25603630
      LL=LDIM-1                                                         25603640
      DO 4 I=1,IOFF                                                     25603650
      XX=AIMAG(DATA(1))                                                 25603660
      DO 5 K=1,LL                                                       25603670
5     DATA(K)=CMPLX(REAL(DATA(K)),AIMAG(DATA(K+1)))                     25603680
4     DATA(LDIM)=CMPLX(REAL(DATA(LDIM)),XX)                             25603690
      RETURN                                                            25603700
      END                                                               25603710
C*****************************************************************     25603720
C     THE FOLLOWING SUBROUTINE SYNCHRONIZES THE RECEIVED DATA.          25603730
C*****************************************************************     25603740
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)                     25603750
      INTEGER Q7FLAG                                                    25603760
      COMPLEX DATA(1),AMP                                               25603770
      DIMENSION NI(1),NQ(1)                                             25603780
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25603790
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    25603800
      NERROR=0                                                          25603810
      IF (IOFF.EQ.0) GO TO 111                                          25603820
      DO 6 K=1,IOFF                                                     25603830
      XX=AIMAG(DATA(LDIM))                                              25603840
      DO 7 KK=2,LDIM                                                    25603850
```

```
FILE: 256GD    FORTRAN  A   UNIV D'/OF OTTAWA

 7     DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))                  25603860
 6     DATA(1)=CMPLX(REAL(DATA(1)),XX)                                   25603870
C                                                                        25603880
C      SYNCHRONIZE THE RECEIVED DATA                                     25603890
C                                                                        25603900
111    NOLD=0                                                            25603910
       NOF=0                                                             25603920
       K=1                                                               25603930
 300   CONTINUE                                                          25603940
       NEW=0                                                             25603950
       DO 200 J=1,NSYMB                                                  25603960
       J1=K+(J-1)*LSAMPL                                                 25603970
       IF(J1.GT.LDIM) J1=J1-LDIM                                         25603980
       AXBAR=REAL(DATA(J1))                                              25603990
       AYBAR=AIMAG(DATA(J1))                                             25604000
       SS=AXBAR*NI(J)                                                    25604010
       IF (SS .GT. 0.) NEW=NEW+1                                         25604020
       SS=AYBAR*NQ(J)                                                    25604030
       IF (SS .GT. 0.) NEW=NEW+1                                         25604040
 200   CONTINUE                                                          25604050
       IF(NOLD.GE.NEW) GO TO 399                                         25604060
       NOLD=NEW                                                          25604070
       NOF=K                                                             25604080
 399   K=K+1                                                             25604090
       LFU=2*NSYMB                                                       25604100
C                                                                        25604110
C      SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP           25604120
C                                                                        25604130
C========================================================              25604140
       NSFT=LSAMPL*3                                                     25604150
       IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300                          25604160
       NOF=NOF-1                                                         25604170
       WRITE(6,56) NOLD,K                                                25604180
   56  FORMAT(1X,'NO OF CORRECT SYMBOLS = ',I6,5X,'NO OF                 25604190
      *SHIFT IN SAMPLES = ',I6,/)                                        25604200
       IF (NOF.EQ.0) GO TO 230                                           25604210
       LO=LDIM-1                                                         25604220
       DO 250 I=1,NOF                                                    25604230
       AMP=DATA(1)                                                       25604240
       DO 240 J=1,LO                                                     25604250
       DATA(J)=DATA(J+1)                                                 25604260
 240   CONTINUE                                                          25604270
       DATA(LDIM)=AMP                                                    25604280
 250   CONTINUE                                                          25604290
C                                                                        25604300
C       OPTIMIZE THE SAMPLING INSTANT                                    25604310
C                                                                        25604320
 230   MI=1                                                              25604330
       MQ=1                                                              25604340
       EOI=FLOAT(NSYMB) + 1.                                             25604350
       EOQ=FLOAT(NSYMB) + 1.                                             25604360
       M=22                                                              25604370
       VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))                           25604380
       SIGMA=SQRT(VARIAN)                                                25604390
       DO 80 J=1,LSAMPL                                                  25604400
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
      EI=0.                                                              25604410
      EQ=0.                                                              25604420
      DO 70 K=1,NSYMB.                                                   25604430
C     VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR                          25604440
      J1=(K-1)*LSAMPL+J                                                  25604450
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.                         25604460
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.                       25604470
      INDEXI=0                                                           25604480
      SS=FLOAT(NI(K))*AXBAR                                              25604490
      IF (SS .LT. 0.) INDEXI=1                                           25604500
      AMPX=ABS(AXBAR)                                                    25604510
      AMPI=ABS(FLOAT(NI(K)))                                             25604520
      THR1I=AMPI - 1.0                                                   25604530
      THR2I=AMPI + 1.0                                                   25604540
      I7FLAG=0                                                           25604550
      IF (AMPI.EQ. 15.) GO TO 30                                         25604560
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1                  25604570
      GO TO 40                                                           25604580
   30 IF(AMPX.LE.THR1I) INDEXI=1                                         25604590
      I7FLAG=1                                                           25604600
   40 CONTINUE                                                           25604610
      INDEXQ=0                                                           25604620
      SS=FLOAT(NQ(K))*AYBAR                                              25604630
      IF(SS .LT. 0.) INDEXQ=1                                            25604640
      AMPY=ABS(AYBAR)                                                    25604650
      AMPQ=ABS(FLOAT(NQ(K)))                                             25604660
      THR1Q=AMPQ - 1.0                                                   25604670
      THR2Q=AMPQ + 1.0                                                   25604680
      Q7FLAG=0                                                           25604690
      IF (AMPQ.EQ. 15.) GO TO 50                                         25604700
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1                  25604710
      GO TO 60                                                           25604720
   50 IF(AMPY.LE.THR1Q) INDEXQ=1                                         25604730
      Q7FLAG=1                                                           25604740
   60 CONTINUE                                                           25604750
      IF(INDEXI.EQ.1) EI=EI + 1.                                         25604760
      IF(INDEXQ.EQ.1) EQ= EQ + 1.                                        25604770
C                                                                        25604780
C     COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB     25604790
C                                                                        25604800
      D1=ABS(AMPX-THR1I)                                                 25604810
      ARG=D1/(SIGMA*SQRT(2.))                                            25604820
C     CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT             25604830
C                                                                        25604840
      IF (ARG.GT.12.) ARG=12.                                            25604850
      EI=EI + ERFC(ARG)/2.                                               25604860
      IF (I7FLAG .EQ. 1) GO TO 65                                        25604870
      D2=ABS(THR2I-AMPX)                                                 25604880
      ARG=D2/(SIGMA*SQRT(2.))                                            25604890
      IF (ARG.GT.12.) ARG=12.                                            25604900
      EI=EI + ERFC(ARG)/2.                                               25604910
   65 D1=ABS(AMPY-THR1Q)                                                 25604920
      ARG=D1/(SIGMA*SQRT(2.))                                            25604930
      IF (ARG.GT.12.) ARG=12.                                            25604940
      EQ=EQ + ERFC(ARG)/2.                                               25604950
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
      IF (Q7FLAG .EQ. 1) GO TO 70                                          25604960
      D2=ABS(THR2Q-AMPY)                                                   25604970
      ARG=D2/(SIGMA*SQRT(2.))                                              25604980
      IF (ARG.GT.12.) ARG=12.                                              25604990
      EQ=EQ + ERFC(ARG)/2.                                                 25605000
   70 CONTINUE                                                             25605010
      IF(EOI.LE.EI) GO TO 75                                               25605020
      EOI=EI                                                               25605030
      MI=J                                                                 25605040
   75 CONTINUE                                                             25605050
      IF (EOQ.LE.EQ) GO TO 80                                              25605060
      EOQ=EQ                                                               25605070
      MQ=J                                                                 25605080
   80 MOFF=IABS(MI-MQ)                                                     25605090
      IF(MOFF.NE.0) WRITE(6,90) MOFF                                       25605100
      OFF=(FLOAT(MI) + FLOAT(NOF))/FLOAT(LSAMPL)                           25605110
   90 FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',          25605120
     #I2,'  SIXTEENTHS OF THE SYMBOL INTERVAL',/)                          25605130
      WRITE(6,95) OFF                                                      25605140
   95 FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)          25605150
      RETURN                                                               25605160
      END                                                                  25605170
C*********************************************************************   25605180
C     THE FOLLOWING SUBROUTINE DECODES THE RECEIVED DATA.                  25605190
C*********************************************************************   25605200
      SUBROUTINE DECODE(DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)               25605210
      INTEGER Q7FLAG                                                       25605220
      COMPLEX DATA(1),AMP                                                  25605230
      DIMENSION EBNO(1),PE(1),NI(1),NQ(1)                                  25605240
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                            25605250
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                       25605260
      NERROR=0                                                             25605270
      DO 1 I=1,NSNR                                                        25605280
    1 PE(I)=0.                                                             25605290
C     CHECK FOR AN ERROR AT THE SAMPLING INSTANT, OSLY THOSE               25605300
C     ERRORS IN ADJACENT LEVELS ARE CONSIDERED                             25605310
    3 DO 2 K=1,NSYMB                                                       25605320
      J1=(K-1)*LSAMPL+MI                                                   25605330
      J2=(K-1)*LSAMPL + MQ                                                 25605340
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.                           25605350
      AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.                         25605360
      INDEXI=0                                                             25605370
      SS=FLOAT(NI(K))*AXBAR                                                25605380
      IF (SS .LT. 0.) INDEXI=1                                             25605390
      AMPX=ABS(AXBAR)                                                      25605400
      AMPI=ABS(FLOAT(NI(K)))                                               25605410
      THRTI=AMPI - 1.0                                                     25605420
      THR2I=AMPI + 1.0                                                     25605430
      I7FLAG=0                                                             25605440
      IF (AMPI.EQ. 15.) GO TO 30                                           25605450
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1                    25605460
      GO TO 40                                                             25605470
   30 IF(AMPX.LE.THR1I) INDEXI=1                                           25605480
      I7FLAG=1                                                             25605490
   40 CONTINUE                                                             25605500
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
      INDEXQ=0                                                          25605510
      SS=FLOAT(NQ(K))*AYBAR                                             25605520
      IF(SS .LT. 0.) INDEXQ=1                                           25605530
      AMPY=ABS(AYBAR)                                                   25605540
      AMPQ=ABS(FLOAT(NQ(K)))                                            25605550
      THR1Q=AMPQ - 1.0                                                  25605560
      THR2Q=AMPQ + 1.0                                                  25605570
      Q7FLAG=0                                                          25605580
      IF (AMPQ.EQ. 15.) GO TO 50                                        25605590
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1                 25605600
      GO TO 60                                                          25605610
   50 IF(AMPY.LE.THR1Q) INDEXQ=1                                        25605620
      Q7FLAG=1                                                          25605630
   60 CONTINUE                                                          25605640
      IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1                25605650
C                                                                       25605660
C     COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.                 25605670
C     THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS         25605680
C     1/2*INTERGAL ( H(F)**2 DF); AND VARIAN = NO*PNOISE.               25605690
C     FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQSRT(2))) 25605700
C     AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +    25605710
C     1/2*ERFC(D2/(SIGMA*SQRT(2))).                                     25605720
C                                                                       25605730
      DO 4 M=1,NSNR                                                     25605740
      VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))                           25605750
      SIGMA=SQRT(VARIAN)                                                25605760
      D1=ABS(AMPX-THR1I)                                                25605770
      ARG=D1/(SIGMA*SQRT(2.))                                           25605780
C     CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT            25605790
C     CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN               25605800
C                                                                       25605810
      IF (INDEXI.EQ.1) GO TO 153                                        25605820
      IF (ARG.GT.12.) ARG=12.                                           25605830
      PEI=ERFC(ARG)/2.                                                  25605840
      IF (I7FLAG .EQ. 1) GO TO 100                                      25605850
      D2=ABS(THR2I-AMPX)                                                25605860
      ARG=D2/(SIGMA*SQRT(2.))                                           25605870
      IF (ARG.GT.12.) ARG=12.                                           25605880
      PEI=PEI + ERFC(ARG)/2.                                            25605890
  100 D1=ABS(AMPY-THR1Q)                                                25605900
      ARG=D1/(SIGMA*SQRT(2.))                                           25605910
      IF (INDEXQ.EQ.1) GO TO 153                                        25605920
      IF (ARG.GT.12.) ARG=12.                                           25605930
      PEQ=ERFC(ARG)/2.                                                  25605940
      IF (Q7FLAG .EQ. 1) GO TO 110                                      25605950
      D2=ABS(THR2Q-AMPY)                                                25605960
      ARG=D2/(SIGMA*SQRT(2.))                                           25605970
      IF (ARG.GT.12.) ARG=12.                                           25605980
      PEQ=PEQ + ERFC(ARG)/2.                                            25605990
  110 IF(PEI.LT.1.E-15) PEI=0.                                          25606000
      IF(PEQ.LT.1.E-15) PEQ=0.                                          25606010
    4 PE(M)=PE(M)+(PEI+PEQ)/2.                                          25606020
    2 CONTINUE                                                          25606030
      DO 5 I=1,NSNR                                                     25606040
      PE(I)=PE(I)/FLOAT(NSYMB)                                          25606050
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
    5  EBNO(I)=FLOAT(I)                                                    25606060
       PRINT 150                                                           25606070
  150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)                           25606080
       WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)                              25606090
  151  FORMAT(5X,F5.1,10X,E13.6)                                           25606100
       WRITE (6,152) NERROR                                                25606110
  152  FORMAT(/5X,'ERRORS=',I5,/)                                          25606120
       GO TO 155                                                           25606130
  153  PRINT 154                                                           25606140
  154  FORMAT (10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)            25606150
  155  CONTINUE                                                            25606160
       RETURN                                                              25606170
       END                                                                 25606180
C*********************************************************************    25606190
C      THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR              25606200
C      P(E) AS LOW AS 1.0E-8 AND A C/N RATIO AS HIGH AS 35 dB.            25606210
C      I/P : ID=1 FOR AMPLITUDE DISTORTIONS                               25606220
C            ID=2 FOR GD DISTORTIONS                                      25606230
C*********************************************************************    25606240
       SUBROUTINE DRAWCN(ID,XARRAY,YARRAY,JCURV,JLAST)                     25606250
       DIMENSION XARRAY(34),YARRAY(34),X(29),Y(29)                         25606260
       COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                          25606270
       COMMON /NUMB2/ NSNR,NSYMB,BAUD                                      25606280
       M=NSNR                                                              25606290
       CF=8.                                                               25606300
       DO 1 K=1,NSNR                                                       25606310
       XARRAY(K)=XARRAY(K) + 10*(ALOG10(CF))                               25606320
       IF(YARRAY(K).GE.1.0) YARRAY(K)=1.0                                  25606330
       IF(YARRAY(K).LE.1.0E-8) GOTO 3                                      25606340
       GOTO 1                                                              25606350
  3    M=K-1                                                               25606360
       GOTO 4                                                              25606370
  1    CONTINUE                                                            25606380
  4    CONTINUE                                                            25606390
       I=M+1                                                               25606400
       II=I+1                                                              25606410
       XARRAY(I)=8.0                                                       25606420
       XARRAY(II)=2.0                                                      25606430
       YARRAY(I)=1.E-8                                                     25606440
       YARRAY(II)=0.4                                                      25606450
       IF(JCURV.GT.1) GOTO 2                                               25606460
C                                                                          25606470
C     ESTABLISH THE SURFACE AREA.                                          25606480
C                                                                          25606490
       CALL PLOTS(30.0,27.5)                                               25606500
C      CALL FACTOR(.8)                                                     25606510
C                                                                          25606520
C     ESTABLISH THE ORIGIN.                                                25606530
C                                                                          25606540
       CALL PLOT(3.0,1.5,-3)                                               25606550
C                                                                          25606560
C     DRAW THE LOGARITHMIC Y-AXIS.                                         25606570
C                                                                          25606580
      CALL LGAXS(0.0,0.0,29HPROBABILITY OF A SYMBOL ERROR,29,25.,          25606590
     +90.,1.0E-8,.4)                                                       25606600
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
C
C     DRAW THE LINEAR X-AXIS.                                                  25606610
C                                                                              25606620
      CALL AXIS(0.0,0.0,9HC/N IN dB,-9,                                        25606630
     +22.0,0.0,8.0,2.0)                                                        25606640
C                                                                              25606650
C     WRITE THE TITLE OF THE GRAPH.                                            25606660
C                                                                              25606670
C      CALL SYMBOL(6.,23.,.35,'PARABOLIC AMPLITUDE DISTORTION',0.,30)          25606680
C                                                                              25606690
C     PLOT LEGEND                                                              25606700
C*****************************************                                     25606710
      GO TO (11,22), ID                                                        25606720
   22 CALL SYMBOL(2.,13.,.35,'NO DISTORTION',0.,13)                            25606730
      CALL SYMBOL(2.,12.,.35,'L = 0.095 ns/MHz',0.,16)                         25606740
      CALL SYMBOL(2.,11.,.35,'L = 0.190 ns/MHz',0.,16)                         25606750
      CALL SYMBOL(2.,10.,.35,'P = 0.270 ns/MHz**2',0.,19)                      25606760
      CALL SYMBOL(2.,9.,.35,'P = 0.540 ns/MHz**2',0.,19)                       25606770
      CALL SYMBOL(2.,8.,.35,'S = 6.00  ns',0.,12)                              25606780
      CALL SYMBOL(2.,7.,.35,'S = 12.0  ns',0.,12)                              25606790
      CALL SYMBOL(2.,4.5,.35,22HGROUP DELAY DISTORTION,0.,22)                  25606800
      GO TO 2                                                                  25606810
C*****************************************                                     25606820
   11 CALL SYMBOL(2.,13.,.35,'NO DISTORTION',0.,13)                            25606830
      CALL SYMBOL(2.,12.,.35,'L = 0.038 dB/MHz',0.,16)                         25606840
      CALL SYMBOL(2.,11.,.35,'L = 0.076 dB/MHz',0.,16)                         25606850
      CALL SYMBOL(2.,10.,.35,'P = 0.004 dB/MHz**2',0.,19)                      25606860
      CALL SYMBOL(2.,9.,.35,'P = 0.008 dB/MHz**2',0.,19)                       25606870
      CALL SYMBOL(2.,8.,.35,'S = 0.2   dB',0.,12)                              25606880
      CALL SYMBOL(2.,7.,.35,'S = 0.4   dB',0.,12)                              25606890
      CALL SYMBOL(2.,4.5,.35,20HAMPLITUDE DISTORTION,0.,20)                    25606900
    2 CONTINUE                                                                 25606910
C      YPAGE=6.5 - (JCURV-1)                                                   25606920
C      YPGE=YPAGE-.1                                                           25606930
C      CALL SYMBOL(2.,YPAGE,.2,JCURV,0.,-1)                                    25606940
C      CALL SYMBOL(2.5,YPGE,.28,2Hz=,0.,2)                                     25606950
C      CALL SYMBOL (5.0,YPAGE,.28,'IN dB/MHz**2',0.,12)                        25606960
C                                                                              25606970
C     PLOT THE DATA IN LOG-LINEAR MODE.                                        25606980
C                                                                              25606990
      CALL LGLIN(XARRAY,YARRAY,M,1,2,JCURV,1)                                  25607000
      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)                                25607010
      RETURN                                                                   25607020
      END                                                                      25607030
C*****************************************************************           25607040
C     THIS SUBROUTINE DRAWS THE EYE DIAGRAM FOR A                              25607050
C     SYMBOL DURATION.                                                         25607060
C*****************************************************************           25607070
      SUBROUTINE EYEQ(DATA)                                                    25607080
      DIMENSION DATA2(64),DATA3(64)                                            25607090
      COMPLEX DATA(1)                                                          25607100
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                                25607110
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                           25607120
C     ESTABLISH THE SURFACE AREA.                                              25607130
      CALL PLOTS(35.0,27.5)                                                    25607140
                                                                               25607150
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
C       ESTABLISH THE ORIGIN.                                                   25607160
        CALL PLOT(3.0,13.0,-3)                                                  25607170
C       WRITE THE TITLE OF THE GRAPH.                                           25607180
        CALL SYMBOL(10.0,12.0,0.49,16HQPSK EYE DIAGRAM,0.0,16)                  25607190
C       DRAW THE TIME AXIS.                                                     25607200
        CALL AXIS(0.0,0.0,1H ,-1,31.,0.0,1.0,1.0)                               25607210
C       DRAW THE AMPLITUDE AXIS.                                                25607220
        CALL AXIS(0.0,-12.0,1H ,1,26.0,90.0,-3.0,0.25)                          25607230
C       PLOT THE DATA.                                                          25607240
        JJ=2*LSAMPL                                                             25607250
        DATA2(JJ+1)=0.0                                                         25607260
        DATA2(JJ+2)=0.25                                                        25607270
        DATA3(JJ+1)=1.0                                                         25607280
        DATA3(JJ+2)=1.0                                                         25607290
        M2=NSYMB/2                                                              25607300
        DO 4 KK=1,M2                                                            25607310
        DO 2 I=1,JJ                                                             25607320
        II=(KK-1)*2*LSAMPL+I                                                    25607330
        DATA2(I)=REAL(DATA(II))                                                 25607340
    2   DATA3(I)=FLOAT(I)                                                       25607350
        CALL LINE(DATA3,DATA2,JJ,1,0,0)                                         25607360
    4   CONTINUE                                                                25607370
        CALL PLOT(0.0,0.0,999)                                                  25607380
        RETURN                                                                  25607390
        END                                                                     25607400
C*****************************************************************          25607410
C       THIS SUBROUTINE COMPUTES THE POWER SPECTRUM                             25607420
C*****************************************************************          25607430
        SUBROUTINE SPECT(SIGNAL,PO)                                             25607440
        COMPLEX SIGNAL(1)                                                       25607450
        COMPLEX CWK(130)                                                        25607460
        DIMENSION X(4096),PO(1),IWK(7),WK(64),TPO(65)                           25607470
        COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                               25607480
        S=0.                                                                    25607490
        DO 10 I=1,LDIM                                                          25607500
        L=I-IOFF                                                                25607510
        IF(L.LE.0) L=L+LDIM                                                     25607520
        X(I)=REAL(SIGNAL(I))                                                    25607530
        X(I+LDIM)=AIMAG(SIGNAL(L))                                              25607540
10      S=S+X(I)+X(I+LDIM)                                                      25607550
        S=S/FLOAT(4096)                                                         25607560
        DO 20 I=1,4096                                                          25607570
20      X(I)=X(I)-S                                                             25607580
        CALL FTFPS(X,Y,4096,4096,0,PO,PSY,XPS,IWK,WK,CWK,IER)                   25607590
        PMAX=PO(1)                                                              25607600
        DO  30 I=1,65                                                           25607610
30      TPO(I)=-10.*ALOG10(PO(I)/PMAX)                                          25607620
        WRITE(6,1) (TPO(I),I=1,65)                                              25607630
        TPO(65)=PO(65)                                                          25607640
        DO 40 I=1,63                                                            25607650
40      TPO(65-I)=TPO(66-I)+2.*PO(65-I)                                         25607660
        TPO(1)=TPO(2)+PO(1)                                                     25607670
        TPMAX=TPO(1)                                                            25607680
        DO 50 I=1,65                                                            25607690
50      PO(I)=-10.*ALOG10(PO(I)/TPMAX)                                          25607700
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
      WRITE(6,2) (PO(I),I=1,65)                                          25607710
2     FORMAT(2X,'NORM. POWER SPECTRUM WITH TOTAL POWER=1'/               25607720
     &,10(2X,F6.1))                                                      25607730
1     FORMAT(2X,'POWER SPECTRUM'/,10(2X,F6.1))                           25607740
      RETURN                                                             25607750
      END                                                                25607760
C*****************************************************************     25607770
C THIS SUBROUTINE SIMULATES A HARD LIMITER                              25607780
C*****************************************************************     25607790
      SUBROUTINE HLIM(DATA)                                              25607800
      COMPLEX DATA(1)                                                    25607810
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                          25607820
      DO 10 I=1,LDIM                                                     25607830
10    DATA(I)=DATA(I)/CABS(DATA(I))                                      25607840
      RETURN                                                             25607850
      END                                                                25607860
C*****************************************************************     25607870
C                PARABOLIC GROUP DELAY                                  25607880
C     GROUP DELAY IN NANOSECONDS = S*F*F                                25607890
C*****************************************************************     25607900
      SUBROUTINE PARGD(S,TF)                                             25607910
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                          25607920
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                     25607930
      COMPLEX TF(1)                                                      25607940
      SBANDW=FLOAT(LSAMPL)*BAUD                                          25607950
      NO1=LDIM/2 + 1                                                     25607960
      NO2=NO1 + 1                                                        25607970
      DO 20 I=1,NO1                                                      25607980
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                   25607990
      PHI=2.*3.141592/3.*(S*(AI**3.))/(10.**3.))                         25608000
      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                              25608010
20    CONTINUE                                                           25608020
      DO 21 I=NO2,LDIM                                                   25608030
      TF(I)=CONJG(TF(LDIM+2-I))                                          25608040
21    CONTINUE                                                           25608050
      RETURN                                                             25608060
      END                                                                25608070
C*****************************************************************     25608080
C     SINE OR COSINE GROUP DELAY WITH ARBITRARY AMPLITUDE               25608090
C     AND NUMBER OF RIPPLES                                             25608100
C     GROUP DELAY IN NANOSECONDS = C*COS(2*PIE*K*F/FNYQ +ISIN*PIE/2)    25608110
C*****************************************************************     25608120
      SUBROUTINE SINCGD(ISIN,C,K,TF)                                     25608130
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                          25608140
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                     25608150
      COMPLEX TF(1)                                                      25608160
      SBANDW=FLOAT(LSAMPL)*BAUD                                          25608170
      NO1=LDIM/2 + 1                                                     25608180
      NO2=NO1 + 1                                                        25608190
      C1=(1.+ALPHA)*BAUD/K                                               25608200
      IF (ISIN.EQ.1) GO TO 50                                            25608210
      DO 10 I=1,NO1                                                      25608220
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                   25608230
      PHI=(C*C1/(10.**3.))*SIN(2.*3.141592*AI/C1)                        25608240
      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                              25608250
```

FILE: 256GD FORTRAN. A UNIV D'/OF OTTAWA

```
10    CONTINUE                                                          25608260
      DO 11 I=NO2,LDIM                                                  25608270
      TF(I)=CONJG(TF(LDIM+2-I))                                         25608280
11    CONTINUE                                                          25608290
      GO TO 60                                                          25608300
50    DO 12 I=1,NO1                                                     25608310
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                  25608320
      PHI=(C*C1/(10.**3.))*COS(2.*3.141592*AI/C1)                       25608330
      TF(I)=TF(I)*CMPLX(COS(PHI),SIN(PHI))                              25608340
12    CONTINUE                                                          25608350
      DO 13 I=NO2,LDIM                                                  25608360
      TF(I)=TF(LDIM+2-I)                                                25608370
13    CONTINUE                                                          25608380
60    RETURN                                                            25608390
      END                                                               25608400
C*****************************************************************     25608410
C          LINEAR GROUP DELAY                                           25608420
C     GROUP DELAY IN NANOSECONDS = B*F                                  25608430
C*****************************************************************     25608440
      SUBROUTINE LINGD(B,TF)                                            25608450
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25608460
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    25608470
      COMPLEX TF(1)                                                     25608480
      SBANDW=FLOAT(LSAMPL)*BAUD                                         25608490
      NO1=LDIM/2 + 1                                                    25608500
      NO2=NO1 + 1                                                       25608510
      DO 10 I=1,NO1                                                     25608520
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                  25608530
      PHI=3.141592*(B/(10.**3.))*(AI**2.)                               25608540
      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                             25608550
10    CONTINUE                                                          25608560
      DO 20 I=NO2,LDIM                                                  25608570
      TF(I)=TF(LDIM+2-I)                                                25608580
20    CONTINUE                                                          25608590
      RETURN                                                            25608600
      END                                                               25608610
C*****************************************************************     25608620
C                LINEAR AMPLITUDE DISTORTION                          25608630
C*****************************************************************     25608640
      SUBROUTINE LINAMP(Z,TF)                                           25608650
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25608660
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    25608670
      COMPLEX TF(1)                                                     25608680
      SBANDW=FLOAT(LSAMPL)*BAUD                                         25608690
      NO1=LDIM/2 + 1                                                    25608700
      NO2=NO1 + 1                                                       25608710
      AI=Z*SBANDW/FLOAT(LDIM)                                           25608720
      DO 10 I=1,NO1                                                     25608730
      AJ=(I-1)*AI/20.                                                   25608740
      TF(I)=TF(I)*(10.**AJ)                                             25608750
10    CONTINUE                                                          25608760
      DO 20 I=NO2,LDIM                                                  25608770
      AZ=(LDIM+2-I)*AI/20.                                              25608780
      TF(I)=TF(I)/(10.**AZ)                                             25608790
20    CONTINUE                                                          25608800
```

FILE: 256GD FORTRAN A UNIV D'/OF OTTAWA

```
      RETURN                                                            25608810
      END                                                               25608820
C*****************************************************************     25608830
C       PARABOLIC AMPLITUDE DISTORTION                                  25608840
C*****************************************************************     25608850
      SUBROUTINE PARAMP(P,TF)                                           25608860
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25608870
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    25608880
      COMPLEX TF(1)                                                     25608890
      SBANDW=FLOAT(LSAMPL)*BAUD                                         25608900
      NO1=LDIM/2 + 1                                                    25608910
      NO2=NO1 + 1                                                       25608920
      AI=P*((SBANDW/FLOAT(LDIM))**2.)                                   25608930
      DO 10 I=1,NO1                                                     25608940
      AJ=((I-1)**2.)*AI/20.                                             25608950
      TF(I)=TF(I)*(10.**AJ)                                             25608960
10    CONTINUE                                                          25608970
      DO 11 I=NO2,LDIM                                                  25608980
      TF(I)=TF(LDIM+2-I)                                                25608990
11    CONTINUE                                                          25609000
      RETURN                                                            25609010
      END                                                               25609020
C*****************************************************************     25609030
C     SINE OR COSINE RIPPLE FOR AMPLITUDE DISTORTION                    25609040
C*****************************************************************     25609050
      SUBROUTINE SINAMP(ISIN,D,K,TF)                                    25609060
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         25609070
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    25609080
      COMPLEX TF(1)                                                     25609090
      SBANDW=FLOAT(LSAMPL)*BAUD                                         25609100
      NO1=LDIM/2 + 1                                                    25609110
      NO2=NO1 + 1                                                       25609120
      AJ=SBANDW/FLOAT(LDIM)                                             25609130
      C1=(1.+ALPHA)*BAUD/K                                              25609140
      IF(ISIN.EQ.1) GO TO 50                                            25609150
      DO 10 I=1,NO1                                                     25609160
      AZ=D*COS(2.*3.141592*AJ*(FLOAT(I)-1.)/C1)/20.                     25609170
      TF(I)=TF(I)*(10.**AZ)                                             25609180
10    CONTINUE                                                          25609190
      DO 20 I=NO2,LDIM                                                  25609200
      TF(I)=TF(LDIM+2-I)                                                25609210
20    CONTINUE                                                          25609220
      RETURN                                                            25609230
50    DO 30 I=1,NO1                                                     25609240
      AZ=D*SIN(2.*3.141592*AJ*(FLOAT(I)-1.)/C1)/20.                     25609250
      TF(I)=TF(I)*(10.**AZ)                                             25609260
30    CONTINUE                                                          25609270
      DO 40 I=NO2,LDIM                                                  25609280
      AZ=D*SIN(2.*3.141592*AJ*(FLOAT(LDIM+2-I)-1.)/C1)/20.              25609290
      TF(I)=TF(I)/(10.**AZ)                                             25609300
40    CONTINUE                                                          25609310
      RETURN                                                            25609320
      END                                                               25609330
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
C*****************************************************************      22500010
C     FILENAME : 225Q FORTRAN                                          22500020
C     CONVENTIONAL MODEL (BY INCREASING THE TRANSMISSION RATE)         22500030
C     225-QPR FORT1/SG IN LINEAR CHANNEL WITH LINEAR,                  22500040
C     PARABOLIC,OR SIN/COSINE RIPPLE FILTER DISTORTIONS.               22500050
C     BAUD RATE IS IN KBAUD, SBANDW IS IN KHZ.                         22500060
C     P(E) FOR MODEL-1 (FILTERS ARE EQUALLY APPORTIONED)               22500070
C     THIS PROGRAM CAN EASILY BE MODIFIED FOR SIMULATING MODEL-B.      22500080
C*****************************************************************      22500090
      COMPLEX DATA(65536),TF(65536),TF1(65536)                         22500100
      DIMENSION PE1(50),EBNO(50),NI(2048),NQ(2048),BB(10)              22500110
      DIMENSION XARRAY(52),YARRAY(52),NIP(2048),NQP(2048)              22500120
      REAL IBAUD                                                       22500130
C     INITIALIZATION.                                                  22500140
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW                 22500150
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2                           22500160
      DATA BB/1.,1.03,1.04,1.05,1.06,1.07,1.09/                        22500170
      LDIM=65536                                                       22500180
      IOFF=0                                                           22500190
      LSAMPL=32                                                        22500200
      NSNR=45                                                          22500210
      NSYMB=2048                                                       22500220
      BAUD=266.7                                                       22500230
      SBANDW=BAUD*LSAMPL                                               22500240
      NRUNS=1                                                          22500250
      NO1=LDIM/2+1                                                     22500260
      NO2=NO1+1                                                        22500270
C     END OF INITIALIZATION.                                           22500280
C     START OF COMPUTATIONS.                                           22500290
      DO 100 NR=1,NRUNS                                                22500300
      A=BB(NR)                                                         22500310
      XA=A-1.                                                          22500320
      WRITE(6,200) XA                                                  22500330
  200 FORMAT('*******',5X,'BW EFFICIENCY INC BY',F8.3)                 22500340
      CALL LOAD64(DATA,NI,NQ)                                          22500350
C     THE FOLLOWING SCALING IS FOR EYE DIAGRAM                         22500360
      DO 43 I=1,LDIM                                                   22500370
   43 DATA(I)=2.*DATA(I)                                               22500380
      FBX=BAUD/2.                                                      22500390
      ALPHA=0.0                                                        22500400
      CALL XSINX(TF,FBX)                                               22500410
      CALL FILTER(DATA,TF)                                             22500420
      IS=1                                                             22500430
      IBAUD=BAUD/A                                                     22500440
      CALL PR1(TF,IBAUD,IS)                                            22500450
      FB=BAUD/2./A                                                     22500460
      CALL RECTF(TF1,FB)                                               22500470
      DO 17 I=1,LDIM                                                   22500480
   17 TF(I)=TF(I)*TF1(I)                                               22500490
      CALL FILTER(DATA,TF)                                             22500500
      GO TO 111                                                        22500510
C================================                                      22500520
C     END OF THE TRANSMITTER   =                                       22500530
C================================                                      22500540
 45   CALL ENERGY(DATA,EB)                                             22500550
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      IS=2                                                                    22500560
      IBAUD=BAUD/A                                                            22500570
      CALL PR1(TF,IBAUD,IS)                                                   22500580
      CALL HHGG(TF,PNOISE)                                                    22500590
      CALL FILTER(DATA,TF)                                                    22500600
C                                                                             22500610
      CALL PRCODE(NI,NIP,NSYMB)                                               22500620
      CALL PRCODE(NQ,NQP,NSYMB)                                               22500630
      CALL SYNCRO(DATA,PNOISE,NIP,NQP,MI,MQ,EB)                               22500640
      CALL DECODE(NERROR,DATA,PNOISE,MI,MQ,NIP,NQP,EBNO,PEI,EB)               22500650
      IF((NERROR.NE.0).AND.(NR.EQ.NRUNS)) CALL PLOT(0.0,0.0,999)              22500660
      IF(NERROR.NE.0) GO TO 100                                               22500670
      DO 99 I=1,NSNR                                                          22500680
      XARRAY(I)=EBNO(I)                                                       22500690
   99 YARRAY(I)=PEI(I)                                                        22500700
      A=1.                                                                    22500710
      CALL DRAWCN(XARRAY,YARRAY,NR,NRUNS,A)                                   22500720
  111 CALL EYEQ(DATA)                                                         22500730
  100 CONTINUE                                                                22500740
      STOP                                                                    22500750
      END                                                                     22500760
C*****************************************                                    22500770
C     PARTIAL RESPONSE ENCODER (1+D)                                          22500780
C     B: I/P SYMBOL SEQUENCE                                                  22500790
C     C: O/P SYMBOL SEQUENCE                                                  22500800
C     C(I)=B(I)+B(I-1)                                                        22500810
C*****************************************                                    22500820
      SUBROUTINE PRCODE(B,C,NSYMB)                                            22500830
      INTEGER B(1),C(1)                                                       22500840
      C(1)=B(1)+B(NSYMB)                                                      22500850
      DO 10 I=2,NSYMB                                                         22500860
      C(I)=B(I)+B(I-1)                                                        22500870
   10 CONTINUE                                                                22500880
      RETURN                                                                  22500890
      END                                                                     22500900
C*************************************************                           22500910
C     FORMS A COMPLEX BASEBAND SIGNAL                                         22500920
C     NI : I-CHANNEL SYMBOL SEQ                                               22500930
C     NQ : Q-CHANNEL SYMBOL SEQ                                               22500940
C     DATA : O/P SIGNAL                                                       22500950
C*************************************************                           22500960
      SUBROUTINE SIGNAL(DATA,NI,NQ)                                           22500970
      COMPLEX DATA(1)                                                         22500980
      DIMENSION NI(1),NQ(1)                                                   22500990
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW                        22501000
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2                                  22501010
      DO 2 I=1,NSYMB                                                          22501020
C     LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL                           22501030
      J1=(I-1)*LSAMPL+1                                                       22501040
      J2=I*LSAMPL                                                             22501050
      DO 3 J3=J1,J2                                                           22501060
    3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))                               22501070
    2 CONTINUE                                                                22501080
      RETURN                                                                  22501090
      END                                                                     22501100
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
C*****************************************************
C     IDEAL BRICKWALL FILTER                                    22501110
C     I/P: FB; CUTOFF FREQ                                      22501120
C     O/P: TF; FILTER RESPONSE                                  22501130
C*****************************************************        22501140
      SUBROUTINE RECTF(TF,FB)                                   22501150
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW          22501160
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2                    22501170
      COMPLEX TF(1)                                             22501180
      FBNOR=FB/SBANDW*FLOAT(LDIM)                               22501190
      NFB=IFIX(FBNOR)+1                                         22501200
      DO 10 I=1,NFB                                             22501210
   10 TF(I)=CMPLX(1.,0.)                                        22501220
      NFB=NFB+1                                                 22501230
      DO 30 I=NFB,NO1                                           22501240
   30 TF(I)=CMPLX(0.,0.)                                        22501250
      DO 20 I=NO2,LDIM                                          22501260
   20 TF(I)=CONJG(TF(LDIM+2-I))                                 22501270
      RETURN                                                    22501280
      END                                                       22501290
C*****************************************************        22501300
C     DUOBINARY FILTER                                          22501310
C     IS=1 FOR FULL CLASS-1                                     22501320
C     IS=2 FOR SQUARE ROOT OF CLASS-1                           22501330
C     IBAUD IS THE IDEAL SYMBOL RATE                            22501340
C     BAUD IS THE ACTUAL SYMBOL RATE                            22501350
C*****************************************************        22501360
      SUBROUTINE PR1(TF,IBAUD,IS)                               22501370
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                 22501380
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2                    22501390
      COMPLEX TF(1)                                             22501400
      REAL IBAUD                                                22501410
      SBANDW=BAUD*LSAMPL                                        22501420
      DELF=SBANDW/LDIM                                          22501430
      PI=3.1415926                                              22501440
      FN=0.5*IBAUD/DELF                                         22501450
      IFN=IFIX(FN)+1                                            22501460
      DO 100 I=1,IFN                                            22501470
      J=I-1                                                     22501480
      A=COS(PI*J*DELF/IBAUD)                                    22501490
      GO TO (1,2),IS                                            22501500
    1 TF(I)=CMPLX(A,0.0)                                        22501510
      GO TO 100                                                 22501520
    2 TF(I)=CMPLX(SQRT(A),0.0)                                  22501530
  100 CONTINUE                                                  22501540
      IF1=IFN+1                                                 22501550
      DO 10 I=IF1,NO1                                           22501560
   10 TF(I)=CMPLX(0.0,0.0)                                      22501570
      DO 20 I=NO2,LDIM                                          22501580
   20 TF(I)=CONJG(TF(LDIM+2-I))                                 22501590
      RETURN                                                    22501600
      END                                                       22501610
C*****************************************************        22501620
C     THIS SUBROUTINE GENERATES A BASEBAND 64-QAM SIGNAL.       22501630
C*****************************************************        22501640
                                                                22501650
```

FILE: 225Q FORTRAN A , UNIV D'/OF OTTAWA

```
      SUBROUTINE LOAD64(DATA,NI,NQ)                                          22501660
      COMPLEX DATA(1)                                                        22501670
      DIMENSION NY(11),NI(1),NQ(1),NX(2048)                                  22501680
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                              22501690
C     DATA NX(1),NX(2),NX(3),NX(4),NX(5),NX(6),NX(7),NX(8),NX(9),            22501700
C     #NX(10),NX(11),NX(12)/-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1/            22501710
      DATA NY(1),NY(2),NY(3),NY(4),NY(5),NY(6),NY(7),NY(8),NY(9),            22501720
     #NY(10),NY(11)/-1,-1,1,1,1,-1,-1,-1,1,-1,-1/                            22501730
      DATA JLAST,JTAP/11,1/                                                  22501740
      KKK=2**JLAST                                                           22501750
      KK=KKK-(1+JLAST)                                                       22501760
      DO 6 I=1,JLAST                                                         22501770
      NX(I)=NY(I)                                                            22501780
    6 CONTINUE                                                               22501790
C     GENERATE 2**JLAST -1 LENGTH SEQUENCE USING GIVEN DATA AND              22501800
C     GENERATOR POLYNOMIAL.                                                  22501810
      DO 1 J=1,KK                                                            22501820
      I=J+JLAST                                                              22501830
      NX(I)=NX(J)*NX(J+2)                                                    22501840
      NX(I)=0-NX(I)                                                          22501850
    1 CONTINUE                                                               22501860
C                                                                            22501870
C     GENERATE 6 CYCLICALLY SHIFTED VERSIONS OF ORIGINAL                     22501880
C     SEQUENCE OF LENGTH 2**JLAST.                                           22501890
C                                                                            22501900
      K1=1                                                                   22501910
      K2=2                                                                   22501920
      K3=3                                                                   22501930
      K4=4                                                                   22501940
      K5=5                                                                   22501950
      K6=6                                                                   22501960
      DO 2 I=1,KKK                                                           22501970
      IF (K1.EQ.KKK) K1=1                                                    22501980
      IF (K2.EQ.KKK) K2=1                                                    22501990
      IF (K3.EQ.KKK) K3=1                                                    22502000
      IF (K4.EQ.KKK) K4=1                                                    22502010
      IF (K5.EQ.KKK) K5=1                                                    22502020
      IF (K6.EQ.KKK) K6=1                                                    22502030
      NI(I)=(NX(K1)*4) + (NX(K3)*2) + NX(K5)                                 22502040
      NQ(I)=(NX(K2)*4) + (NX(K4)*2) + NX(K6)                                 22502050
C                                                                            22502060
C     LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL                          22502070
      J1=(I-1)*LSAMPL+1                                                      22502080
      J2=I*LSAMPL                                                            22502090
      DO 3 J3=J1,J2                                                          22502100
    3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))                              22502110
      K1=K1+1                                                                22502120
      K2=K2+1                                                                22502130
      K3=K3+1                                                                22502140
      K4=K4+1                                                                22502150
      K5=K5+1                                                                22502160
      K6=K6+1                                                                22502170
    2 CONTINUE                                                               22502180
      IF(IOFF.EQ.0) RETURN                                                   22502190
      LL=LDIM-1                                                              22502200
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      DO 4 I=1,IOFF                                                     22502210
      XX=AIMAG(DATA(1))                                                 22502220
      DO 5 K=1,LL                                                       22502230
5     DATA(K)=CMPLX(REAL(DATA(K)),AIMAG(DATA(K+1)))                     22502240
4     DATA(LDIM)=CMPLX(REAL(DATA(LDIM)),XX)                             22502250
      RETURN                                                            22502260
      END                                                               22502270
C*****************************************************************     22502280
C     RAISED COSINE FILTER WITH ARBITRARY ALPHA AND X/SINX              22502290
C*****************************************************************     22502300
      SUBROUTINE RCOSTX(TF)                                             22502310
       COMPLEX TF(65536)                                                22502320
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         22502330
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    22502340
      SBANDW=BAUD*LSAMPL                                                22502350
      NO=LDIM/2                                                         22502360
      NO1=NO+1                                                          22502370
       IF (ALPHA.EQ.0) ALPHA=0.0001                                     22502380
      FN=LDIM*(FBW/SBANDW)                                              22502390
      F1=(1.-ALPHA)*FN                                                  22502400
      F2=(1.+ALPHA)*FN                                                  22502410
      IFN=IFIX(FN)                                                      22502420
      IF1=IFIX(F1)+1                                                    22502430
      IF2=IFIX(F2)+1                                                    22502440
C                                                                       22502450
C     THE AMPLITUDE CHARACTERISTICS                                     22502460
C                                                                       22502470
       A1=3.1415926/(2.*FLOAT(IFN))                                     22502480
       DO 8 I=2,IF1                                                     22502490
      TF(1)=CMPLX(1.0,0.0)                                              22502500
       J=I-1                                                            22502510
C                                                                       22502520
C     X/SIN(X) EQUALIZATION                                             22502530
C                                                                       22502540
       A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))                              22502550
      TF(I)=CMPLX(A2,0.0)                                               22502560
    8 CONTINUE                                                          22502570
      JK=IF1+1                                                          22502580
      DO 9 J=JK,IF2                                                     22502590
       I=J-1                                                            22502600
C                                                                       22502610
C     ROOT OF RAISED COSINE                                             22502620
C                                                                       22502630
      A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))                               22502640
      A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)              22502650
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)                           22502660
       TF(J)=TF(J)*CMPLX(A3,0.0)                                        22502670
    9 CONTINUE                                                          22502680
      JH=IF2+1                                                          22502690
      DO 10 I=JH,NO1                                                    22502700
      TF(I)=CMPLX(0.0,0.0)                                              22502710
   10 CONTINUE                                                          22502720
        NO2=NO1+1                                                       22502730
      DO 5 I=NO2,LDIM                                                   22502740
      TF(I)=CONJG(TF(LDIM+2-I))                                         22502750
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
    5 CONTINUE                                                          22502760
      RETURN                                                            22502770
      END                                                               22502780
C*****************************************************************    22502790
C     RAISED COSINE FILTER WITH ARBITRARY ALPHA                         22502800
C*****************************************************************    22502810
      SUBROUTINE RCOSRX(TF)                                             22502820
       COMPLEX TF(65536)                                                22502830
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         22502840
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    22502850
      SBANDW=BAUD*LSAMPL                                                22502860
      NO=LDIM/2                                                         22502870
      NO1=NO+1                                                          22502880
       IF (ALPHA.EQ.0) ALPHA=0.0001                                     22502890
      FN=LDIM*(FBW/SBANDW)                                              22502900
      F1=(1.-ALPHA)*FN                                                  22502910
      F2=(1.+ALPHA)*FN                                                  22502920
      IFN=IFIX(FN)                                                      22502930
      IF1=IFIX(F1)+1                                                    22502940
      IF2=IFIX(F2)+1.                                                   22502950
C                                                                       22502960
C     AMPLITUBE CHARACTERISTIC                                          22502970
C                                                                       22502980
       A1=3.141592/(2.*FLOAT(IFN))                                      22502990
       DO 8 I=2,IF1                                                     22503000
      TF(1)=CMPLX(1.0,0.0)                                              22503010
       J=I-1                                                            22503020
      TF(I)=CMPLX(1.,0.0)                                               22503030
    8 CONTINUE                                                          22503040
      JK=IF1+1                                                          22503050
      DO 9 J=JK,IF2                                                     22503060
       I=J-1                                                            22503070
C                                                                       22503080
C     ROOT OF RAISED COSINE                                             22503090
C                                                                       22503100
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)               22503110
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)                           22503120
    9 CONTINUE                                                          22503130
      JH=IF2+1                                                          22503140
      DO 10 I=JH,NO1                                                    22503150
      TF(I)=CMPLX(0.0,0.0)                                              22503160
   10 CONTINUE                                                          22503170
         NO2=NO1+1                                                      22503180
      DO 5 I=NO2,LDIM                                                   22503190
      TF(I)=CONJG(TF(LDIM+2-I))                                         22503200
    5 CONTINUE                                                          22503210
      RETURN                                                            22503220
      END                                                               22503230
C*****************************************************************    22503240
C     X/SIN(X) EQUALIZER                                                22503250
C     FBX : NYQUIST FREQ FOR THIS EQUALIZER                             22503260
C*****************************************************************    22503270
      SUBROUTINE XSINX(TF,FBX)                                          22503280
       COMPLEX TF(65536)                                                22503290
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         22503300
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    22503310
      SBANDW=BAUD*LSAMPL                                                22503320
      NO=LDIM/2                                                         22503330
      NO1=NO+1                                                          22503340
       IF (ALPHA.EQ.0) ALPHA=0.0001                                     22503350
      FN=LDIM*(FBX/SBANDW)                                              22503360
      F1=(1.-ALPHA)*FN                                                  22503370
      F2=(1.+ALPHA)*FN                                                  22503380
      IFN=IFIX(FN)                                                      22503390
      IF1=IFIX(F1)+1                                                    22503400
      IF2=IFIX(F2)+1                                                    22503410
C                                                                       22503420
C     THE AMPLITUDE CHARACTERISTICS                                     22503430
C                                                                       22503440
       A1=3.1415926/(2.*FLOAT(IFN))                                     22503450
      TF(1)=CMPLX(1.0,0.0)                                              22503460
       DO 8 I=2,IF1                                                     22503470
       J=I-1                                                            22503480
C                                                                       22503490
C     X/SIN(X) EQUALIZATION                                             22503500
C                                                                       22503510
       A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))                              22503520
      TF(I)=CMPLX(A2,0.0)                                               22503530
    8 CONTINUE                                                          22503540
      JK=IF1+1                                                          22503550
      DO 9 J=JK,IF2                                                     22503560
       I=J-1                                                            22503570
C                                                                       22503580
C     ROOT OF RAISED COSINE                                             22503590
C                                                                       22503600
      A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))                               22503610
       TF(J)=CMPLX(A3,0.0)                                              22503620
    9 CONTINUE                                                          22503630
      JH=IF2+1                                                          22503640
      DO 10 I=JH,NO1                                                    22503650
      TF(I)=CMPLX(0.0,0.0)                                              22503660
   10 CONTINUE                                                          22503670
       NO2=NO1+1                                                        22503680
      DO 5 I=NO2,LDIM                                                   22503690
      TF(I)=CONJG(TF(LDIM+2-I))                                         22503700
    5 CONTINUE                                                          22503710
      RETURN                                                            22503720
      END                                                               22503730
C*****************************************************************    22503740
C     THE FOLLOWING SUBROUTINE PERFORMS THE FILTERING                   22503750
C     PROCESS ON THE DATA SEQUENCE.                                     22503760
C*****************************************************************    22503770
      SUBROUTINE FILTER(DATA,TF)                                        22503780
      COMPLEX DATA(1),TF(1)                                             22503790
      DIMENSION IWK(17)                                                 22503800
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         22503810
      CALL FFT2C(DATA,16,IWK)                                           22503820
      DO 1 I=1,LDIM                                                     22503830
    1 DATA(I)=CONJG(DATA(I)*TF(I))                                      22503840
      CALL FFT2C(DATA,16,IWK)                                           22503850
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      DO 2 I=1,LDIM                                                    22503860.
    2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)                               22503870
      RETURN                                                           22503880
      END                                                              22503890
C*****************************************************************   22503900
C     THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE (PNOISE)            22503910
C     AT THE OUTPUT OF THE RECEIVE FILTER                              22503920
C*****************************************************************   22503930
      SUBROUTINE HHGG(TF,PNOISE)                                       22503940
      COMPLEX TF(1)                                                    22503950
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                        22503960
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                   22503970
      SUM=0.0                                                          22503980
      SBANDW=BAUD*LSAMPL                                               22503990
      DO 1 L=1,LDIM                                                    22504000
      HH=(CABS(TF(L)))**2                                              22504010
    1 SUM=SUM+HH                                                       22504020
      PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.                                 22504030
      WRITE (6,2) PNOISE                                               22504040
    2 FORMAT(5X,'PNOISE=',F7.3,/)                                      22504050
      RETURN                                                           22504060
      END                                                              22504070
C*****************************************************************   22504080
C CALCULATE EB                                                         22504090
C*****************************************************************   22504100
          SUBROUTINE ENERGY(DATA,EB)                                   22504110
          COMPLEX DATA(1)                                              22504120
          WATTS=0.                                                     22504130
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                   22504140
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                        22504150
          DO 1 I=1,LDIM                                                22504160
          WATTS=WATTS+((CABS(DATA(I)))**2.)                            22504170
    1     CONTINUE                                                     22504180
          WATTS=WATTS/(FLOAT(LDIM))                                    22504190
          EB=WATTS/(6.*BAUD)                                           22504200
          WRITE (6,40) WATTS,EB                                        22504210
   40     FORMAT(10X,'C=',F15.6,5X,'EB=',E15.8)                        22504220
          RETURN                                                       22504230
          END                                                          22504240
C*****************************************************************   22504250
C     THE FOLLOWING SUBROUTINE SYNCHRONIZES THE RECEIVED DATA.         22504260
C*****************************************************************   22504270
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)                    22504280
      INTEGER QL14                                                     22504290
      COMPLEX DATA(1),AMP                                              22504300
      DIMENSION NI(1),NQ(1)                                            22504310
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                        22504320
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                   22504330
      NERROR=0                                                         22504340
      IF (IOFF.EQ.0) GO TO 111                                         22504350
      DO 6 K=1,IOFF                                                    22504360
      XX=AIMAG(DATA(LDIM))                                             22504370
      DO 7 KK=2,LDIM                                                   22504380
    7 DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))                 22504390
    6 DATA(1)=CMPLX(REAL(DATA(1)),XX)                                  22504400
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
C                                                                       22504410
C     SYNCHRONIZE THE RECEIVED DATA                                     22504420
C                                                                       22504430
C     THE FOLLOWING 9 STMTS SHIFT DATA RIGHT BY 8 SAMPLES               22504440
  111 NRSFT=LSAMPL/4                                                    22504450
      DO 190 I=1,NRSFT                                                  22504460
      AMP=DATA(LDIM)                                                    22504470
      LO=LDIM-1                                                         22504480
      DO 191 J=1,LO                                                     22504490
      K=LDIM-J                                                          22504500
  191 DATA(K+1)=DATA(K)                                                 22504510
      DATA(1)=AMP                                                       22504520
  190 CONTINUE                                                          22504530
      NOLD=0                                                            22504540
      NOF=0                                                             22504550
      K=1                                                               22504560
  300 CONTINUE                                                          22504570
      NEW=0                                                             22504580
      DO 200 J=1,NSYMB                                                  22504590
      THR1I=FLOAT(NI(J))-1.                                             22504600
      THR2I=FLOAT(NI(J))+1.                                             22504610
      THR1Q=FLOAT(NQ(J))-1.                                             22504620
      THR2Q=FLOAT(NQ(J))+1.                                             22504630
      J1=K+(J-1)*LSAMPL                                                 22504640
      IF(J1.GT.LDIM) J1=J1-LDIM                                         22504650
      AXBAR=REAL(DATA(J1))                                              22504660
      AYBAR=AIMAG(DATA(J1))                                             22504670
      IF((AXBAR.GT.THR1I).AND.(AXBAR.LT.THR2I)) NEW=NEW+1               22504680
      IF((AYBAR.GT.THR1Q).AND.(AYBAR.LT.THR2Q)) NEW=NEW+1               22504690
  200 CONTINUE                                                          22504700
      IF(NOLD.GE.NEW) GO TO 399                                         22504710
      NOLD=NEW                                                          22504720
      NOF=K                                                             22504730
  399 K=K+1                                                             22504740
      LFU=2*NSYMB                                                       22504750
C                                                                       22504760
C     SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP           22504770
C                                                                       22504780
C=================================================                      22504790
      NSFT=LSAMPL*5                                                     22504800
      IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300                          22504810
      NOF=NOF-1                                                         22504820
      KT=K-2                                                            22504830
      WRITE(6,56) NOLD,NOF,KT                                           22504840
   56 FORMAT(1X,'NO OF MAX. CORRECT SYMBOLS = ',I4,//,1X,               22504850
     *'NO OF SHIFT IN SAMPLES FOR MAX. CORRECT SYMBOLS= ',I4,//,        22504860
     *1X,'TOTAL NO OF SHIFT FOR SEARCHING = ',I4)                       22504870
C-------------------------------------------------                      22504880
      IF (NOF.EQ.0) GO TO 230                                           22504890
      LO=LDIM-1                                                         22504900
      DO 250 I=1,NOF                                                    22504910
      AMP=DATA(1)                                                       22504920
      DO 240 J=1,LO                                                     22504930
      DATA(J)=DATA(J+1)                                                 22504940
  240 CONTINUE                                                          22504950
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      DATA(LDIM)=AMP                                                    22504960
  250 CONTINUE                                                          22504970
C                                                                       22504980
C      OPTIMIZE THE SAMPLING INSTANT                                    22504990
C                                                                       22505000
  230 MI=1                                                              22505010
      MQ=1                                                              22505020
      EOI=FLOAT(NSYMB) + 1.                                             22505030
      EOQ=FLOAT(NSYMB) + 1.                                             22505040
      M=22                                                              22505050
      VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))                           22505060
      SIGMA=SQRT(VARIAN)                                                22505070
      MOS=LSAMPL/4                                                      22505080
      DO 80 J=1,MOS                                                     22505090
      EI=0.                                                             22505100
      EQ=0.                                                             22505110
      IERX=0                                                            22505120
      IERY=0                                                            22505130
      DO 70 K=1,NSYMB                                                   22505140
C     VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR                         22505150
      J1=(K-1)*LSAMPL+J                                                 22505160
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.                        22505170
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.                      22505180
      INDEXI=0                                                          22505190
      AMPX=ABS(AXBAR)                                                   22505200
      AMPI=ABS(FLOAT(NI(K)))                                            22505210
      THR1I=AMPI - 1.0                                                  22505220
      THR2I=AMPI + 1.0                                                  22505230
      IL14=0                                                            22505240
      IF (AMPI.EQ. 14.) GO TO 30                                        22505250
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1                 22505260
      GO TO 40                                                          22505270
   30 IF(AMPX.LE.THR1I) INDEXI=1                                        22505280
      IL14=1                                                            22505290
   40 CONTINUE                                                          22505300
      INDEXQ=0                                                          22505310
      AMPY=ABS(AYBAR)                                                   22505320
      AMPQ=ABS(FLOAT(NQ(K)))                                            22505330
      THR1Q=AMPQ - 1.0                                                  22505340
      THR2Q=AMPQ + 1.0                                                  22505350
      QL14=0                                                            22505360
      IF (AMPQ.EQ. 14.) GO TO 50                                        22505370
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1                 22505380
      GO TO 60                                                          22505390
   50 IF(AMPY.LE.THR1Q) INDEXQ=1                                        22505400
      QL14=1                                                            22505410
   60 CONTINUE                                                          22505420
      IF(INDEXI.EQ.1) EI=EI + 1.                                        22505430
      IF(INDEXQ.EQ.1) EQ= EQ + 1.                                       22505440
C                                                                       22505450
C     COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB    22505460
C                                                                       22505470
      D1=ABS(AMPX-THR1I)                                                22505480
      ARG=D1/(SIGMA*SQRT(2.))                                           22505490
C     CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT            22505500
```

FILE: 225Q .FORTRAN A UNIV D'/OF OTTAWA

```
C
      IF (ARG.GT.12.) ARG=12.                                          22505510
      EI=EI + ERFC(ARG)/2.                                             22505520
      IF (IL14 .EQ. 1) GO TO 65                                        22505530
      D2=ABS(THR2I-AMPX)                                               22505540
      ARG=D2/(SIGMA*SQRT(2.))                                          22505550
      IF (ARG.GT.12.) ARG=12.                                          22505560
      EI=EI + ERFC(ARG)/2.                                             22505570
   65 D1=ABS(AMPY-THR1Q)                                               22505580
      ARG=D1/(SIGMA*SQRT(2.))                                          22505590
      IF (ARG.GT.12.) ARG=12.                                          22505600
      EQ=EQ + ERFC(ARG)/2.                                             22505610
      IF (QL14 .EQ. 1) GO TO 70                                        22505620
      D2=ABS(THR2Q-AMPY)                                               22505630
      ARG=D2/(SIGMA*SQRT(2.))                                          22505640
      IF (ARG.GT.12.) ARG=12.                                          22505650
      EQ=EQ + ERFC(ARG)/2.                                             22505660
      IERX=IERX+INDEXI                                                 22505670
      IERY=IERY+INDEXQ                                                 22505680
   70 CONTINUE                                                         22505690
      WRITE(6,81) IERX,IERY                                            22505700
   81 FORMAT(5X,'IERX=',I5,5X,'IERY=',I5)                              22505710
      IF(EOI.LE.EI) GO TO 75                                           22505720
      EOI=EI                                                           22505730
      MI=J                                                             22505740
   75 CONTINUE                                                         22505750
      IF (EOQ.LE.EQ) GO TO 80                                          22505760
      EOQ=EQ                                                           22505770
      MQ=J                                                             22505780
   80 MOFF=IABS(MI-MQ)                                                 22505790
      WRITE(6,91) MI,MQ                                                22505800
   91 FORMAT(5X,'MI= ',I3,5X,'MQ= ',I3)                                22505810
      IF(MOFF.NE.0) WRITE(6,90) MOFF                                   22505820
      OFF=(FLOAT(MI) + FLOAT(NOF))/FLOAT(LSAMPL)                       22505830
   90 FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',      22505840
     #I2,'  SIXTEENTHS OF THE SYMBOL INTERVAL',/)                      22505850
      WRITE(6,95) OFF                                                  22505860
   95 FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)      22505870
      RETURN                                                           22505880
      END                                                              22505890
C*******************************************************************  22505900
C     THE FOLLOWING SUBROUTINE DECODES THE RECEIVED DATA.              22505910
C*******************************************************************  22505920
      SUBROUTINE DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)     22505930
      INTEGER QL14                                                     22505940
      COMPLEX DATA(1),AMP                                              22505950
      DIMENSION EBNO(1),PE(1),NI(1),NQ(1)                              22505960
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                        22505970
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                   22505980
      NERROR=0                                                         22505990
      DO 1 I=1,NSNR                                                    22506000
    1 PE(I)=0.                                                         22506010
C     CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE           22506020
C     ERRORS IN ADJACENT LEVELS ARE CONSIDERED                         22506030
    3 DO 2 K=1,NSYMB                                                   22506040
                                                                       22506050
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      J1=(K-1)*LSAMPL+M1                                                    22506060
      J2=(K-1)*LSAMPL + MQ                                                  22506070
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.                            22506080
      AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.                          22506090
      INDEXI=0                                                              22506100
      AMPX=ABS(AXBAR)                                                       22506110
      AMPI=ABS(FLOAT(NI(K)))                                                22506120
      THR1I=AMPI - 1.0                                                      22506130
      THR2I=AMPI + 1.0                                                      22506140
      IL14=0                                                                22506150
      IF (AMPI.EQ. 14.) GO TO 30                                            22506160
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1                     22506170
      GO TO 40                                                              22506180
   30 IF(AMPX.LE.THR1I) INDEXI=1                                            22506190
      IL14=1                                                                22506200
   40 CONTINUE                                                              22506210
      INDEXQ=0                                                              22506220
      AMPY=ABS(AYBAR)                                                       22506230
      AMPQ=ABS(FLOAT(NQ(K)))                                                22506240
      THR1Q=AMPQ - 1.0                                                      22506250
      THR2Q=AMPQ + 1.0                                                      22506260
      QL14=0                                                                22506270
      IF (AMPQ.EQ. 14.) GO TO 50                                            22506280
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1                     22506290
      GO TO 60                                                              22506300
   50 IF(AMPY.LE.THR1Q) INDEXQ=1                                            22506310
      QL14=1                                                                22506320
   60 CONTINUE                                                              22506330
      IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1                    22506340
C                                                                           22506350
C     COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.                     22506360
C     THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS             22506370
C     1/2*INTERGAL ( H(F)**2 DF); AND VARIAN = NO*PNOISE.                   22506380
C     FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQSRT(2)))     22506390
C     AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +        22506400
C     1/2*ERFC(D2/(SIGMA*SQRT(2))).                                         22506410
C                                                                           22506420
      DO 4 M=1,NSNR                                                         22506430
      EBNO(M)=14.5+0.5*FLOAT(M-1)                                           22506440
      VARIAN=PNOISE*EB*(10.**(-0.1*EBNO(M)))                                22506450
      SIGMA=SQRT(VARIAN)                                                    22506460
      D1=ABS(AMPX-THR1I)                                                    22506470
      ARG=D1/(SIGMA*SQRT(2.))                                               22506480
C     CHECK IF ARG IS LARGE IN WHICH CASE PE IS INSIGNIFICANT               22506490
C     CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN                   22506500
C                                                                           22506510
      NERI=K                                                                22506520
      IF (INDEXI.EQ.1) GO TO 153                                            22506530
      IF (ARG.GT.12.) ARG=12.                                               22506540
      PEI=ERFC(ARG)/2.                                                      22506550
      IF (IL14 .EQ. 1) GO TO 100                                            22506560
      D2=ABS(THR2I-AMPX)                                                    22506570
      ARG=D2/(SIGMA*SQRT(2.))                                               22506580
      IF (ARG.GT.12.) ARG=12.                                               22506590
      PEI=PEI + ERFC(ARG)/2.                                                22506600
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
  100 D1=ABS(AMPY-THR1Q)                                              22506610
      ARG=D1/(SIGMA*SQRT(2.))                                         22506620
      IF (INDEXQ.EQ.1) GO TO 153                                      22506630
      IF (ARG.GT.12.) ARG=12.                                         22506640
      PEQ=ERFC(ARG)/2.                                                22506650
      IF (QL14 .EQ. 1) GO TO 110                                      22506660
      D2=ABS(THR2Q-AMPY)                                              22506670
      ARG=D2/(SIGMA*SQRT(2.))                                         22506680
      IF (ARG.GT.12.) ARG=12.                                         22506690
      PEQ=PEQ + ERFC(ARG)/2.                                          22506700
  110 IF(PEI.LT.1.E-15) PEI=0.                                        22506710
      IF(PEQ.LT.1.E-15) PEQ=0.                                        22506720
    4 PE(M)=PE(M)+(PEI+PEQ)/2.                                        22506730
    2 CONTINUE                                                        22506740
      DO 5 I=1,NSNR                                                   22506750
    5 PE(I)=PE(I)/FLOAT(NSYMB)                                        22506760
      PRINT 150                                                       22506770
  150 FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)                       22506780
      WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)                          22506790
  151 FORMAT(5X,F5.1,10X,E10.2)                                       22506800
      WRITE (6,152) NERROR                                            22506810
  152 FORMAT(/5X,'ERRORS=',I5,/)                                      22506820
      GO TO 155                                                       22506830
  153 PRINT 154                                                       22506840
  154 FORMAT (10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)        22506850
      WRITE(6,157) NERI                                               22506860
  157 FORMAT(5X,'NERI=',I5)                                           22506870
  155 CONTINUE                                                        22506880
      RETURN                                                          22506890
      END                                                             22506900
C*****************************************************************  22506910
C     THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR         22506920
C     P(E) AS LOW AS 1.0E-10 AND A C/N RATIO AS HIGH AS 42 dB.      22506930
C*****************************************************************  22506940
      SUBROUTINE DRAWCN(XARRAY,YARRAY,JCURV,JLAST,A)                  22506950
      DIMENSION XARRAY(1),YARRAY(1)                                   22506960
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                  22506970
      M=NSNR                                                          22506980
      BR=2.*3.*BAUD                                                   22506990
      NBW=BAUD/A                                                      22507000
      CF=BR/NBW                                                       22507010
      DO 1 K=1,NSNR                                                   22507020
      XARRAY(K)=XARRAY(K) + 10*(ALOG10(CF))                           22507030
      IF(YARRAY(K).LE.1.0E-10) GOTO 3                                 22507040
      GOTO 1                                                          22507050
    3 M=K-1                                                           22507060
      GOTO 4                                                          22507070
    1 CONTINUE                                                        22507080
    4 CONTINUE                                                        22507090
      I=M+1                                                           22507100
      II=I+1                                                          22507110
      XARRAY(I)=22.                                                   22507120
      XARRAY(II)=1.0                                                  22507130
      YARRAY(I)=1.E-10                                                22507140
      YARRAY(II)=0.4                                                  22507150
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      IF(JCURV.GT.1) GOTO 2                                              22507160
C                                                                        22507170
C     ESTABLISH THE SURFACE AREA.                                        22507180
C                                                                        22507190
      CALL PLOTS(30.0,27.5)                                              22507200
      CALL FACTOR(0.85)                                                  22507210
C                                                                        22507220
C     ESTABLISH THE ORIGIN.                                              22507230
C                                                                        22507240
      CALL PLOT(3.0,1.5,-3)                                              22507250
C                                                                        22507260
C     DRAW THE LOGARITHMIC Y-AXIS.                                       22507270
C                                                                        22507280
      CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,25.,                 22507290
     +90.,1.0E-10,.4)                                                    22507300
C                                                                        22507310
C     DRAW THE LINEAR X-AXIS.                                            22507320
C                                                                        22507330
      CALL AXIS(0.0,0.0,9HC/N IN dB,-9,                                  22507340
     +22.0,0.0,22.,1.0)                                                  22507350
C                                                                        22507360
C     WRITE THE TITLE OF THE GRAPH.                                      22507370
C                                                                        22507380
C      CALL SYMBOL(6.,23.,.35,'PARABOLIC AMPLITUDE DISTORTION',0.,30)    22507390
C                                                                        22507400
C     PLOT LEGEND                                                        22507410
C*****************************************                               22507420
      GO TO 2                                                            22507430
   41 CALL SYMBOL(2.,13.,.35,'NO DISTORTION',0.,13)                      22507440
      CALL SYMBOL(2.,12.,.35,'B=0.095 ns/MHz',0.,14)                     22507450
      CALL SYMBOL(2.,11.,.35,'B=0.190 ns/MHz',0.,14)                     22507460
      CALL SYMBOL(2.,10.,.35,'S=0.270 ns/MHz**2',0.,17)                  22507470
      CALL SYMBOL(2.,9.,.35,'S=0.540 ns/MHz**2',0.,17)                   22507480
      CALL SYMBOL(2.,8.,.35,'C=12.0 ns',0.,9)                            22507490
      CALL SYMBOL(2.,7.,.35,'C=24.0 ns',0.,9)                            22507500
      CALL SYMBOL(2.,4.5,.35,22HGROUP DELAY DISTORTION,0.,22)            22507510
       CALL SYMBOL(2.0,3.5,.35,'T(f)=B*f',0.,8)                          22507520
      CALL SYMBOL(2.0,2.5,.35,'T(f)=S*f*f',0.,11)                        22507530
      CALL SYMBOL(2.0,1.5,.35,'T(f)=C*sin(4 f/f )',0.,18)                22507540
      CALL SYMBOL(2.0,1.3,.35,'                  f ',0.,18)              22507550
C*****************************************                               22507560
C      CALL SYMBOL(2.,13.,.35,'NO DISTORTION',0.,13)                     22507570
C      CALL SYMBOL(2.,12.,.35,'Z=0.076 dB/MHz',0.,14)                    22507580
C      CALL SYMBOL(2.,11.,.35,'Z=0.152 dB/MHz',0.,14)                    22507590
C      CALL SYMBOL(2.,10.,.35,'P=0.00725 dB/MHz**2',0.,19)               22507600
C      CALL SYMBOL(2.,9.,.35,'P=0.0145 dB/MHz**2',0.,18)                 22507610
C      CALL SYMBOL(2.,8.,.35,'D=0.4 dB',0.,8)                            22507620
C      CALL SYMBOL(2.,7.,.35,'D=0.8 dB',0.,8)                            22507630
C      CALL SYMBOL(2.,4.5,.35,20HAMPLITUDE DISTORTION,0.,20)             22507640
C       CALL SYMBOL(2.0,3.5,.35,'A(f)=Z*f',0.,8)                         22507650
C      CALL SYMBOL(2.0,2.5,.35,'A(f)=P*f*f',0.,11)                       22507660
C      CALL SYMBOL(2.0,1.5,.35,'A(f)=D*sin(4 f/f )',0.,18)               22507670
C      CALL SYMBOL(2.0,1.3,.35,'                  f ',0.,18)             22507680
   2  CONTINUE                                                           22507690
C      YPAGE=6.5 - (JCURV-1)                                             22507700
```

FILE: 225Q FORTRAN A -UNIV D'/OF OTTAWA

```
C      YPOS=YPAGE-.1                                                    22507710
C      CALL SYMBOL(2.,YPAGE,.2,JCURV,0.,-1)                             22507720
C      CALL SYMBOL(2.5,YPOS,.28,2HZ=,0.,2)                              22507730
C      CALL SYMBOL (5:0,YPAGE,.28,'IN dB/MHz**2',0.,12)                 22507740
C                                                                       22507750
C      PLOT THE DATA IN LOG-LINEAR MODE.                                22507760
C                                                                       22507770
       CALL LGLIN(XARRAY,YARRAY,M,1;4,JCURV,1)                          22507780
       IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)                        22507790
       RETURN                                                           22507800
       END                                                              22507810
C*****************************************************************    22507820
C      THIS SUBROUTINE DRAWS THE EYE DIAGRAM FOR A                      22507830
C      SYMBOL DURATION.                                                 22507840
C*****************************************************************    22507850
       SUBROUTINE EYEQ(DATA)                                            22507860
       DIMENSION DATA2(66),DATA3(66)                                    22507870
       COMPLEX DATA(1)                                                  22507880
       COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                        22507890
       COMMON /NUMB2/ NSNR,NSYMB,BAUD                                   22507900
C      ESTABLISH THE SURFACE AREA.                                      22507910
       CALL PLOTS(37.0,27.5)                                            22507920
C      ESTABLISH THE ORIGIN.                                            22507930
       CALL PLOT(3.0,13.0,-3)                                           22507940
C      WRITE THE TITLE OF THE GRAPH.                                    22507950
C      CALL SYMBOL(10.0,12.0,0.49,16HQPSK EYE DIAGRAM,0.0,16)           22507960
C      DRAW THE TIME AXIS.                                              22507970
       CALL AXIS(0.0,0.0,1H ,-1,24.,0.0,1.0,2.8)                        22507980
C      DRAW THE AMPLITUDE AXIS.                                         22507990
       CALL AXIS(0.0,-8.00,1H ,1,16.0,90.0,-24.,3.0)                    22508000
C      PLOT THE DATA.                                                   22508010
       JJ=2*LSAMPL                                                      22508020
       DATA2(JJ+1)=0.0                                                  22508030
       DATA2(JJ+2)=3.0                                                  22508040
       DATA3(JJ+1)=1.0                                                  22508050
       DATA3(JJ+2)=2.8                                                  22508060
       M2=NSYMB/4                                                       22508070
       M3=3*M2+1                                                        22508080
       M4=4*M2-1                                                        22508090
       DO 4 KK=1,M2                                                     22508100
       DO 2 I=1,JJ                                                      22508110
       II=(KK-1)*LSAMPL+I                                               22508120
       IF(II.GT.LDIM) II=II-(II/LDIM)*LDIM                              22508130
       DATA2(I)=REAL(DATA(II))                                          22508140
   2   DATA3(I)=FLOAT(I)                                                22508150
       CALL LINE(DATA3,DATA2,JJ,1,0,0)                                  22508160
   4   CONTINUE                                                         22508170
       CALL PLOT(0.0,0.0,999)                                           22508180
       RETURN                                                           22508190
       END                                                              22508200
C*****************************************************************    22508210
C      PARABOLIC GROUP DELAY                                            22508220
C      GROUP DELAY IN NANOSECONDS = S*F*F                               22508230
C      F IN KHZ                                                         22508240
C*****************************************************************    22508250
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      SUBROUTINE PARGD(S,TF)                                                22508260
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                             22508270
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                        22508280
      COMPLEX TF(1)                                                         22508290
      SBANDW=FLOAT(LSAMPL)*BAUD                                             22508300
      NO1=LDIM/2 + 1                                                        22508310
      NO2=NO1 + 1                                                           22508320
      DO 20 I=1,NO1                                                         22508330
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                      22508340
      PHI=2.*3.141592/3.*((S*(AI**3.))/(10.**6.))                           22508350
      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                                 22508360
20    CONTINUE                                                              22508370
      DO 21 I=NO2,LDIM                                                      22508380
      TF(I)=CONJG(TF(LDIM+2-I))                                             22508390
21    CONTINUE                                                              22508400
      RETURN                                                                22508410
      END                                                                   22508420
C*************************************************************            22508430
C  SINE OR COSINE GROUP DELAY WITH ARBITRARY AMPLITUDE                      22508440
C    AND NUMBER OF RIPPLES                                                  22508450
C     GROUP DELAY IN NANOSECONDS = C*COS(2*PIE*K*F/FNYQ +ISIN*PIE/2)        22508460
C     F IN KHZ                                                            22508470
C*************************************************************            22508480
      SUBROUTINE SINCGD(ISIN,C,K,TF)                                        22508490
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                             22508500
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                        22508510
      COMPLEX TF(1)                                                         22508520
      SBANDW=FLOAT(LSAMPL)*BAUD                                             22508530
      NO1=LDIM/2 + 1                                                        22508540
      NO2=NO1 + 1                                                           22508550
      C1=2.*(1.+ALPHA)*FBW/K                                                22508560
      IF (ISIN.EQ.1) GO TO 50                                               22508570
      DO 10 I=1,NO1                                                         22508580
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                      22508590
      PHI=(C*C1/(10.**6.))*SIN(2.*3.141592*AI/C1)                           22508600
      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                                 22508610
10    CONTINUE                                                              22508620
      DO 11 I=NO2,LDIM                                                      22508630
      TF(I)=CONJG(TF(LDIM+2-I))                                             22508640
11    CONTINUE                                                              22508650
      GO TO 60                                                              22508660
50    DO 12 I=1,NO1                                                         22508670
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                      22508680
      PHI=(C*C1/(10.**6.))*COS(2.*3.141592*AI/C1)                           22508690
      TF(I)=TF(I)*CMPLX(COS(PHI),SIN(PHI))                                  22508700
12    CONTINUE                                                              22508710
      DO 13 I=NO2,LDIM                                                      22508720
      TF(I)=TF(LDIM+2-I)                                                    22508730
13    CONTINUE                                                              22508740
60    RETURN                                                                22508750
      END                                                                   22508760
C*************************************************************            22508770
C           LINEAR GROUP DELAY                                              22508780
C     GROUP DELAY IN NANOSECONDS = B*F                                      22508790
C     B : ns/KHz                                                            22508800
```

FILE: 225Q FORTRAN, A UNIV D'/OF OTTAWA

```
C     DISCRETE FREQ SPACING =SBANDW/LDIM KHZ                          22508810
C*****************************************************************   22508820
      SUBROUTINE LINGD(B,TF)                                          22508830
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                       22508840
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                  22508850
      COMPLEX TF(1)                                                   22508860
      SBANDW=FLOAT(LSAMPL)*BAUD                                       22508870
      NO1=LDIM/2 + 1                                                  22508880
      NO2=NO1 + 1                                                     22508890
      DO 10 I=1,NO1                                                   22508900
      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                                22508910
      PHI=3.141592*(B/(10.**6.))*(AI**2.)                             22508920
      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                           22508930
10    CONTINUE                                                        22508940
      DO 20 I=NO2,LDIM                                                22508950
      TF(I)=TF(LDIM+2-I)                                              22508960
20    CONTINUE                                                        22508970
      RETURN                                                          22508980
      END                                                             22508990
C*****************************************************************   22509000
C     LINEAR AMPLITUDE DISTORTION                                     22509010
C*****************************************************************   22509020
      SUBROUTINE LINAMP(Z,TF)                                         22509030
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                       22509040
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                  22509050
      COMPLEX TF(1)                                                   22509060
      SBANDW=FLOAT(LSAMPL)*BAUD                                       22509070
      NO1=LDIM/2 + 1                                                  22509080
      NO2=NO1 + 1                                                     22509090
      AI=Z*SBANDW/FLOAT(LDIM)                                         22509100
      DO 10 I=1,NO1                                                   22509110
      AJ=(I-1)*AI/20.                                                 22509120
      TF(I)=TF(I)*(10.**AJ)                                           22509130
10    CONTINUE                                                        22509140
      DO 20 I=NO2,LDIM                                                22509150
      AZ=(LDIM+2-I)*AI/20.                                            22509160
      TF(I)=TF(I)/(10.**AZ)                                           22509170
20    CONTINUE                                                        22509180
      RETURN                                                          22509190
      END                                                             22509200
C*****************************************************************   22509210
C     PARABOLIC AMPLITUDE DISTORTION                                  22509220
C*****************************************************************   22509230
      SUBROUTINE PARAMP(P,TF)                                         22509240
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                       22509250
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                  22509260
      COMPLEX TF(1)                                                   22509270
      SBANDW=FLOAT(LSAMPL)*BAUD                                       22509280
      NO1=LDIM/2 + 1                                                  22509290
      NO2=NO1 + 1                                                     22509300
      AI=P*((SBANDW/FLOAT(LDIM))**2.)                                 22509310
      DO 10 I=1,NO1                                                   22509320
      AJ=((I-1)**2.)*AI/20.                                           22509330
      TF(I)=TF(I)*(10.**AJ)                                           22509340
10    CONTINUE                                                        22509350
```

FILE: 225Q FORTRAN A UNIV D'/OF OTTAWA

```
      DO 11 I=NO2,LDIM                                                  22509360
      TF(I)=TF(LDIM+2-I)                                                22509370
11    CONTINUE                                                          22509380
      RETURN                                                            22509390
      END                                                               22509400
C*****************************************************************     22509410
C     SINE OR COSINE RIPPLE FOR AMPLITUDE DISTORTION                    22509420
C*****************************************************************     22509430
      SUBROUTINE SINAMP(ISIN,D,K,TF)                                    22509440
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL                         22509450
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                                    22509460
      COMPLEX TF(1)                                                     22509470
      SBANDW=FLOAT(LSAMPL)*BAUD                                         22509480
      NO1=LDIM/2 + 1                                                    22509490
      NO2=NO1 + 1                                                       22509500
      AJ=SBANDW/FLOAT(LDIM)                                             22509510
      C1=BAUD/K                                                         22509520
      IF(ISIN.EQ.1) GO TO 50                                            22509530
      DO 10 I=1,NO1                                                     22509540
      AZ=D*COS(2.*3.141592*AJ*(FLOAT(I)-1.)/C1)/20.                     22509550
      TF(I)=TF(I)*(10.**AZ)                                             22509560
10    CONTINUE                                                          22509570
      DO 20 I=NO2,LDIM                                                  22509580
      TF(I)=TF(LDIM+2-I)                                                22509590
20    CONTINUE                                                          22509600
      RETURN                                                            22509610
50    DO 30 I=1,NO1                                                     22509620
      AZ=D*SIN(2.*3.141592*AJ*(FLOAT(I)-1.)/C1)/20.                     22509630
      TF(I)=TF(I)*(10.**AZ)                                             22509640
30    CONTINUE                                                          22509650
      DO 40 I=NO2,LDIM                                                  22509660
      AZ=D*SIN(2.*3.141592*AJ*(FLOAT(LDIM+2-I)-1.)/C1)/20.              22509670
      TF(I)=TF(I)/(10.**AZ)                                             22509680
40    CONTINUE                                                          22509690
      RETURN                                                            22509700
      END                                                               22509710
C*****************************************************************     22509720
C     RANDOM NUMBER GENERATOR                                           22509730
C*****************************************************************     22509740
      SUBROUTINE RAND(IR)                                               22509750
      INTEGER IR(256)                                                   22509760
      REAL R(256)                                                       22509770
      DOUBLE PRECISION DSEED                                            22509780
      DSEED=123457.D0                                                   22509790
      N=256                                                             22509800
      DO 300 I=1,N                                                      22509810
      R(I)=GGUBFS(DSEED)                                                22509820
      IR(I)=IFIX(R(I)*2040.)                                            22509830
300   CONTINUE                                                          22509840
      WRITE(6,10) IR                                                    22509850
10    FORMAT(10X,I5)                                                    22509860
      RETURN                                                            22509870
      END                                                               22509880
```

REFERENCES

1. Y. Tan, S. Shinmyo, M. Murakami, and M. Tsunokuni, "2 GHz band digital radio equipment employing 8-level PSK with cosine roll-off spectrum shaping," Proc. of the IEEE-ICC'78, pp. 33.3.1-33.3.5.

2. S.G. Barber and C.W. Anderson, "Modulation considerations for the DRS-8 91 Mb/s digital radio," Proc. of the IEEE-ICC'77, pp. 5.6-111-5.6-115.

3. G.R. McMillen, M. Shafi, and D.P. Taylor, "Simultaneous adaptive estimation of carrier phase, symbol timing and data for a 49-QPRS DFE radio receiver," IEEE Trans. Commun., Vol. COM-32, pp. 429-443, April 1984.

4. H. Yamamoto, "Advanced 16-QAM techniques for digital microwave radio," IEEE Trans. Commun., COM-19, pp.36-45,1981.

5. I. Horikawa, T. Murase, and Y. Saito, "Design and performance of a 200 Mb/s 16-QAM digital radio system," IEEE Trans. Commun., Vol. COM-2, pp. 1953-1958, 1979.

6. Y. Saito, K. Morita, and H. Yamamoto, "5L-D1 digital radio system," Proc. of the IEEE-ICC'82, pp. 2B.1.1-2B.1.7.

7. P. Dupuis, M. Joindot, and D. Soufflet, "16-QAM modulation for high capacity digital radio system," IEEE Trans. Commun., Vol. COM-27, pp. 1771-1782, 1979.

8. T. Hill and K. Feher, "A performance study of NLA 64-state QAM," IEEE Trans. Commun., Vol. COM-31, pp. 821-826, June 1983.

9. B.T. Bynum and E.W. Allen, "135 Mb/s-6 GHz transmission system design considerations," Proc. of the IEEE-ICC'83, pp. F2.5.1-F2.5.6.

10. T. Noguchi, et al., "6 GHz 135 MBPS digital radio system with 64-QAM modulation," Proc. of the IEEE-ICC'83, pp. F2.4:1-F2.4.6.

11. P.R. Hartmann and J.A. Crossett, "135 MBS-6GHz Transmission system using 64-QAM modulation," Proc. of the IEEE-ICC'83, pp. F2.6.1-F2.6.7.

12. J.D. McNicol, S.G. Barber, and F. Rivest, "Design and application of the RD-4A and RD-6A 64-QAM digital radio systems," Proc. of the IEEE-ICC'84, pp. 646-652.

13. M. Borgne, "Comparison of 16, 32, 64 and 128 QAM modulation schemes for digital radio systems," Proc. of the IEEE-GLOBECOM'83, pp.1.7.1-1.7.5.

14. K. Feher, "Digital Communications : Satellite/Earth Station Engineering," Englewood cliffs, NJ : Prentice-Hall, 1983, Chapter 5.

15. K. Feher, "Digital Communications : Microwave Applications," Englewood Cliffs, NJ : Prentice-Hall, 1981, Chapter 8.

16. R.L. Freeman, "Telecommunication System Engineering : Analog and Digital Design," John Wiley & Sons, New York, 1980, Chapter 4.

17. K. Feher, E.M. Karkar, and J.C.Y. Huang, "On 6.67 b/s/Hz - 256-QAM and 225-QPRS modems for T1/SG data-in-voice (DIV) applications," Proc. of the IEEE-ICC'85, pp. 453-459, June 1985.

18. W.R. Bennett and J.R. Davey, "Data Transmission," New York : McGraw-Hill, 1965, Chapter 7.

19. E.R. Kretzmer, "Generalization of a technique for binary data communication," IEEE Trans. Commun., Vol. COM-14, pp.67-68 Feb. 1966.

20. P. Kabal and S. Pasupathy, "Partial-response signalling," IEEE Trans. Commun., Vol. COM-23, pp. 921-934, Sept. 1975.

21. A. Lender, "Correlative (Partial response) techniques and applications to digital radio systems," Published as Chapter 7 in K. Feher, "Digital Communications : Microwave Applications," Englewood Cliffs, NJ : Prentice-Hall, 1981.

22. K.T. Wu and K. Feher, "Multi-level PRS/QPRS above the Nyquist rate," IEEE Trans. Commun., Vol. COM-33, pp. 735-739., July 1985.

23. J.E. Mazo, "Faster than Nyquist signaling," Bell Syst. Tech. J., Vol. 54, pp. 1451-1462, Oct. 1976.

24. G.J. Foschini, "Contrasting performance of faster binary signaling with QAM," Bell Syst. Tech. J., Vol. 63, pp. 1419-1445, Oct. 1984.

25. A. Fihel and H. Sari, "Reduced-bandwidth 16-QAM : A power and bandwidth efficient modulation technique for digital microwave radio systems," Proc. of the IEEE ICC'85, Chicago, pp.15.6.1-15.6.5, June 1985.

26. M.D. Eggers and J.H. Painter, "Optimal symbol by symbol detection for duobinary signaling," IEEE Trans. Commun. Vol. COM-31, pp. 1077-1085 Sept. 1983.

27. G.D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," IEEE Trans. Inform. Theory, Vol. IT-18, pp. 363-378, May 1972.

28. C.A. Belfiore and J.H. Park, Jr., "Decision feedback equalization," Proc. IEEE, Aug. 1979, pp. 1143-1156.

29. D.D. Falconer and F.R. Magee, Jr., "Evaluation of decision feedback equalization and Viterbi detection for voiceband data transmission-part I," IEEE Trans. Commun. Vol. COM-24, pp. 1130-1139, Oct. 1976.

30. D.D. Falconer and F.R. Magee, Jr., "Evaluation of decision feedback and viterbi algorithm detection for voiceband data transmission-part II," IEEE Trans. Commun., Vol. COM-24, pp. 1238-1245, Nov. 1976.

31. D.D. Falconer, "Application of passband decision feedback equalization in two-dimension data communication systems," IEEE Trans. Commun., Vol. COM-24, pp. 1159-1166, Oct. 1976.

32. D.D. Falconer and G.J. Foschini, "Theory of minimum mean-square-error QAM systems employing decision feedback equalization," Bell Syst. Tech. J., Vol. 52, pp. 1821-1849, Dec. 1973.

33. K. Feher, K.T. Wu, J.C.Y. Huang, and D.E. MacNally, "Improved efficiency data transmission technique," U.S. and Canadian patent disclosure, University of Ottawa and Pascal Associate, No. 256-8148-1, Oct 1, 1984.

34. A. Lender, R. Rogers, and H. Olszanski, "4 Bits/Hz correlative single-sideband digital radio at 2 GHz," Proc. of the IEEE-ICC'79 pp.5.2.1-5.2.5.

35. K.T. Wu, I. Sasase, and K. Feher, "Improved efficiency 15-level modified duobinary PRS above the Nyquist rate," Proc. of the IEEE-ICC'85, pp. 31.3.1-31.3.5,June 1985.

36. K.T. Wu and K. Feher, "256-QAM modem performance in distorted channels," IEEE Trans. Commun. Vol. COM-33, pp. 487-491, May 1985.

37. M. Subramanian, K.C. O'Brien, and P.J. Puglis, "Phase dispersion characteristics during fade in a microwave line-of -sight radio channel," Bell Syst. Tech. J., Vol. 52, Dec. 1973.

38. G.M. Babler, "Selectively faded nondiversity and space diversity narrowband microwave radio channels," Bell Syst. Tech. J., Vol. 52, Feb. 1973.

39. I. Sasase, K.T. Wu, and K. Feher, "Comparison of improved efficiency 225-QPRS and 256-QAM in distorted channels," Proc. of the IEEE-ICC'85, pp. 15.3.1-15.3.5, June 1985.

40. R.W. Lucky, J. Salz, and E.J. Weldon, Jr., "Principles of Data Communication," New York : McGraw-Hill, 1968, Chapter 4.

41. G. Bianconi and L. Calandrino, "Adaptive baseband equalization in digital M-QAM radio systems in presence of multipath propagation," Proc. of the IEEE-ICC'85, Chicago, pp. 15.7.1-15.7.8, June 1985.

42. Y. Saito, S. Komaki, and M. Murotani, "Feasibility considerations of high-level multi-carrier system," Proc. of the IEEE-ICC'84, pp. 656-671.

43. V.K. Prabhu, "Co-channel interference immunity of high capacity QAM," Proc. of the IEEE-ICC'81, pp. 68.8.1-68.8.4.

44. V.K. Prabhu, "The detection efficiency of 16-ray QAM," Bell Syst. Tech. J., Vol. 59, pp. 639-656, April 1980.

45. J.J. Spilker, Jr., "Digital Communications by Satellite," Englewood Cliffs, NJ : Prentice-Hall, 1977, Chapter 11.

46. K.T. Wu and K. Feher, "Error rate considerations for 64-,256-QAM and 225-QPRS with co-channel interference," Proc. of the IEEE ELECTRONICOM'85, pp.662-665, Toronto, Oct. 1985.

47. Y.L. Kuo and T.J. Aprille, "A baseband equalizer for a 16-state QAM digital system over mastergroup band analog networks," Proc. of the IEEE-GLOBECOM'82, pp. F3.6.1-F3.6.5.

48. D.E. MacNally and J.C.Y. Huang, "Propagation protection techniques for high capacity digital radio systems," Proc. of the IEEE-ICC'84, pp. 1007-1010.

49. K. Aoki, et al., "The adaptive transversal equalizer for 90 MBPS 64-QAM radio relay system," Proc. of the IEEE-ICC'84, pp.1003-1006.

50. R.W. Lucky, "Survey of communication theory literature : 1968-1973," IEEE Trans. Inform. Theory, pp.725-739, 1973.

51. R.W. Lucky, "Techniques for adaptive equalization of digital communication systems," Bell Syst. Tech. J., Vol. 45, No.2, pp. 255-286, Feb. 1966.

52. D. Hirsch and W.J. Wolf, "A simple adaptive equalizer for efficient data transmission," IEEE Trans. Commun. Tech. Vol. COM-18, pp. 5-11, Feb. 1970.

53. J. Proakis, "Digital Communications," New York : McGraw-Hill, 1983, Chapter 6.

54. G.L. Fenderson, et al., "Adaptive transversal equalization of multipath propagation for 16-QAM, 90 Mb/s digital radio," Bell Syst. Tech. J., Vol. 63, pp. 1447-1463 Oct. 1984.

55. A.A.M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," IEEE Trans. Commun., Vol. COM-29, pp. 1715-1720, Nov. 1981.

56. A.A.M. Saleh and J. Salz, "Adaptive linearization of power amplifiers in digital radio systems," Proc. of the IEEE-ICC'83, pp. C8.4.1-C8.4.5, 1983.

57. W.D. Rummler, "A new selective fading model : application to propagation data," Bell Syst. Tech. J., Vol. 58, pp. 1037-1071, May-June 1979.

58. W.D. Rummler, "Digital radio outage due to selective fading-observation vs prediction from laboratory simulation," Bell Syst. Tech. J., Vol. 58, pp. 1073-1110, May-June, 1979.

59. G.J. Foschini and J. Salz, "Digital communications over fading radio channels," Proc. of the IEEE-ICC'83, pp. C8.1.1-C8.1.7.

60. N. Amitay and L.J. Greenstein, "Multipath outage performance of digital radio receivers using finite-tap adaptive equalizers," IEEE Trans. Commun., Vol. COM-32, pp. 597-608, May 1984.