



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform 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.

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

AVIS

La qualité de cette microforme 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.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-56425-3

ON BANDWIDTH EFFICIENT QAM TRANSMISSION SYSTEMS

By

Panayotis (Takis) Mathiopoulos

A THESIS
SUBMITTED TO THE SCHOOL OF
GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

OTTAWA-CARLETON INSTITUTE FOR ELECTRICAL ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF OTTAWA



UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

KIXΛH

Δαίμονος ἐπιπόνου καὶ τύχη χαλεπῆς
 ἔφημερον σπέρμα, τί μέ βιάζεσθε
 λέγειν, ἅ ὑμῖν ἄρειον μὴ γινῶναι.

Ὅστος οὕτως μ' ἔζωσαν
 καὶ μέ παρακινοῦν πυκνὸ κοπάδι
 τὰ φιλέταιρα σπουργίτια.

Η ΣΤΕΡΝΑ

Βρέθηκα στὴν ἀνάγκη νά βάλω τό
 νοσοκομεῖο τοῦ Δόν Χουάν Ταβέρα
 μέ τὴν μορφή μοντέλου,...

Δομήνικος Θεοτοκόπουλος

THRUSH

Emphemeral issue of a vicious daemon
 and a harsh fate, why do you force me
 to speak of things that it would
 be better for you not to know.

Still, I feel that I'm encircled and
 inspired by a dense flock of
 philataerus sparrows.

THE CISTERN

I found I had to put the hospital of
 Don Juan Tavera in the form of a model...

Domenikos Theotokopoulos

From the book **G. Seferis: Collected Poems** (Translated by E.K. and P.S.)

Ἦχῃστε, οἱ σάλπιγγες... Καμπάνες βροντερές,
 δυνῆστε σύγκορμη τὴ χώρα, πέρα ὡς πέρα...
 Βόγκα, Παιάνα! Οἱ σημαῖες οἱ φοβερές,
 στῆς Λευτεριᾶς ξεδιπλωθεῖτε τον ἄέρα!

Ἀγγελος Σικελιανός
 28-2-1943

“Ο ΗΡΩΑΣ ΠΟΥΛΟΥΣΕ ΦΡΟΥΤΑ ΜΕ ΤΟ ΚΑΡΟΤΣΙ”

“Ἐδῶ... Ἐδῶ... Σᾶς μιλά ὁ ...”

...τῇ δέ μύθοι οἴους παῖς
 ἔτ' ὧν ἁλίων ἢ αὐτουργῶν
 ἤκουσεν ἐν Σμύρνῃ
 ἢ που ἐν Κλαζομεναῖς...

...to tales heard in boyhood
 from fishermen or peasants
 at Smyrna or Clazomenae...

Abstract

In this thesis the performance of spectrally efficient Quadrature Amplitude Modulated (QAM) systems operating in distorted transmission channels is studied and methods to improve the performance of these systems are suggested. The methodology employed in this study includes analysis, computer simulation and extensive experimental research.

In order to improve the performance of M -ary QAM schemes operated in the presence of transmission System Caused Phase Jitter (SCPJ), we introduce and analyze a novel synchronization method based on two *out-of-band* pilot tones. The two pilot tones are symmetrically located near but outside the QAM signal spectrum and are referred to as a dual-pilot system. By means of these two pilot tones and by employing a wideband carrier extraction subsystem, instead of the commonly used minimal bandwidth one, we demonstrate that the proposed method can improve considerably the performance of an M -ary QAM scheme in a SCPJ environment. The purpose of this wideband carrier extraction subsystem is to extract the carrier and ideally all the unwanted SCPJ and therefore track, in order to cancel during demodulation its effects on the received digital data signal. We have analytically evaluated the performance of a 16-QAM which employs the dual-pilot system and is operated in a sinusoidal SCPJ environment. Our analysis takes into account the most important system imperfections, such as distortion and bandwidth of the pilot extraction filters. The results obtained indicate that performance gains of up to 8 dB (at a symbol error rate of 10^{-4}) over conventional non-pilot schemes are possible for an 11 degrees (zero-to-peak) sinusoidal SCPJ. It is also shown, that for the pilot extraction filters, group delay distortion is the prominent source of degradation as compared to amplitude distortion. However, both distor-

tions have to be minimized or compensated for, to achieve maximal SCPJ cancellation. Furthermore, we show that for the case of relatively high filter distortion, or equivalently if the compensation is incorrect, the unwanted SCPJ is enhanced rather than cancelled. In this case, the overall system produces a worse performance than a system which does not employ the SCPJ cancellation. To prove this concept and to gain additional insight into some of the problems involved, we designed and implemented in hardware an 800 kb/s, 16-QAM dual-pilot system. A number of different experimental set-ups and extensive measurements were used to verify some of the results obtained by analysis and computer simulation. The proposed dual-pilot system can also be employed in more spectrally efficient schemes, such as 64-, 256-, 512- and 1024-QAM.

In order to further improve the overall spectral efficiency of data transmission systems, we introduce a novel synchronization method for QAM systems, which employs two *in-band* pilot tones for simultaneous carrier and symbol timing recovery. The proposed method does not require an additional bandwidth to accommodate the pilot tones, as do all other previous pilot-based methods, and at the same time it employs a very simple, and therefore cost-effective receiver structure. It is shown that the only in-band pilot frequency which does not cause any interference in the digital modulated signal and at the same time provides carrier and symbol synchronization at the receiver is the Nyquist frequency. By analysis, computer simulation and measurements, we evaluated a 16-QAM scheme which employs the proposed in-band, at the Nyquist frequency pilot method in an additive white Gaussian noise (AWGN) channel. Our results indicate that, as long as the pilot extraction filter bandwidth is relatively small, the overall 16-QAM system performance has a negligible degradation. Like the out-of-band dual-pilot scheme, the proposed in-band, at the Nyquist frequency pilot insertion technique can also be used in more spectrally efficient systems, such as 64-, 256-, 512- and 1024-QAM.

Finally in this thesis we present a performance study for the 512- and 1024-QAM schemes which can achieve high spectral efficiencies of up to 9 and 10 b/s/Hz, respectively. Such high spectral efficiencies are required for a number of digital transmission applications, including the transmission of a CEPT-1 (2.048 Mb/s) into a 240 kHz supergroup band, transmission of a 19.2 kb/s rate signal into a single telephone channel (2.4 kHz) and trans-

mission of two 90 Mb/s rate signals in 20 MHz RF bandwidth. Due to these emerging requirements, and to satisfy our curiosity about the feasibility of such high spectral efficiencies, we analyzed and evaluated the performance of these QAM schemes in the presence of different system imperfections. Our studies include:

- i) A quantitative evaluation of the effects of linear channel distortions on the performance of 512- and 1024-QAM. The numerical results obtained could show modem designers and researchers how flat and how linear the filters' amplitude and phase responses should be.

- ii) Detailed analysis and examination of the effects of Co-Channel Interference (CCI), both of constant envelope and amplitude modulated type, on the performance of 512- and 1024-QAM systems.

- iii) The impact of Adjacent Channel Interference (ACI) on the performance of 512- and 1024-QAM.

Our study indicates that, under certain conditions both schemes are implementable, and we foresee that they will be employed in the future for applications requiring spectral efficiencies of up to 10 b/s/Hz.

In summary, this thesis presents a comprehensive study, at both the analytical and experimental levels, of two novel pilot-based synchronization schemes and their applicability to bandwidth efficient QAM systems. Furthermore, it provides timely information on the performance and the feasibility of the highly bandwidth efficient 512- and 1024-QAM schemes, which can be employed to achieve spectral efficiencies of up to 9 and 10 b/s/Hz, respectively.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Kamilo Feher, for his technical guidance, financial support and excellent suggestions during the course of my Ph.D. thesis research. Furthermore, I would like to thank him for pointing out to me how important it is *to be a practical guy*.

My affiliation with the Digital Communication Research Group at the University of Ottawa during the last four years proved to be a very rewarding experience. In this respect, I would like to thank D. Makrakis, H. Ohnishi, D. W. Prendergast, Dr. I. Sasase, Dr. J. S. Seo, K. Sreenath and Dr. K. T. Wu with whom I have had many valuable exchanges of ideas in the field of digital communications. Furthermore, I am indebted to Dr. M. El-Tanany of Carleton University, Dr. G. Jull, formerly with CRC, D. Makrakis, Dr. I. Sasase of Keio University, Japan, Dr. J. S. Seo of IDC and Dr. K. T. Wu of BNR for critically reading the manuscript and for making numerous suggestions to improve the presentation of this work. I assume responsibility, however, for the flaws and inadequacies that remain.

The excellent collaboration which I have had with my friend Dimitri Makraki, in research projects other than those reported in this thesis, is acknowledged with great pleasure. In addition to that, his help, support and encouragement during the course of my Ph.D. thesis research are greatly appreciated. Thanks are also due to Doug Prendergast for his technical advice and invaluable help during Dr. Feher's absence. I feel deeply indebted to Drs. W. Steenaart and N. D. Georganas for their support and words of encouragement. I wish to thank also Dr. P. Galko, Dr. W. McGee, Dr. R. G. Harrison of Carleton University, Dr. G. Costache, T. Tzakis, R. W. Scott, Z. Brun of Gandalf Data Ltd., J. Carre, C. Carruthers and especially

Dr. S. Stathopoulos for his advice and suggestions.

Mr. Mike Jervis, formerly with Raytheon Canada Limited (RCL), deserves credit for giving me, many years ago, the opportunity to obtain my first industrial experience with the design and development of a new generation of two air-navigational systems (DVOR and DME). Furthermore, by being a consultant to RCL over the last couple of years, I have had the opportunity to work with him on a number of challenging projects.

It is also with great pleasure that I acknowledge the efforts and help of Dr. George Jull in trying to identify *a beautiful research problem*. Dr. Jull's continuous interest in my studies and career is deeply appreciated.

I would like to thank the *anonymous referee*, who six years ago indirectly and *maturely* enough encouraged me to start my Ph.D. under the supervision of Dr. Feher and as a result to pursue a career in the field of digital communications.

Thanks must also go to all members of the jury of my thesis defence for the time they took to read this thesis. These are Dr. M. Lecours of Laval University, Dr. M. El-Tanany, Dr. E. Petriu and Dr. W. Steenaart.

In moments of doubt, my dearest friend, *Yogi*, showed constant understanding and was always extremely supportive. This mere *acknowledgement* by no means compensates the hardships she had to go through on account of this thesis.

I wish to thank my parents for encouraging and supporting my studies and career.

Last, but certainly not least, my wholehearted appreciation must go to Prof. Vasilios Makios of the University of Patras, Greece, my first teacher who guided me into the *unknown* of research. Prof. Makios' interest, help, constant involvement and encouragement throughout my studies were major factors in my being able to stay on course. Studying in Canada has been a privilege and I would like to thank him for giving me the opportunity to come here.

PREFACE

Based on the research reported in this thesis, the following publications have resulted:

REFEREED PAPERS IN JOURNALS

- R-3 P. Mathiopoulos, H. Ohnishi and K. Feher, "A study of 1024-QAM system in the presence of filtering imperfections," *IEE Proceedings, Part I*, vol. 136, pp. 175-179, April 1989.
- R-2 P. Mathiopoulos and K. Feher, "Pilot aided techniques for system caused phase jitter cancellation," *IEEE Transactions on Broadcasting*, vol. 34, pp. 356-366, September 1988.
- R-1 P. Mathiopoulos and K. Feher, "Performance evaluation of a 512-QAM system in distorted channels," *IEE Proceedings, Part F*, vol. 133, pp. 199-204, April 1986.

REFEREED PAPERS IN CONFERENCE PROCEEDINGS

- C-6 P. Mathiopoulos and K. Feher, "Error rate considerations for 256-QAM systems in the presence of adjacent channel interference," in the *Proceedings of ICC '88*, Philadelphia, U.S.A., pp. 1064-1068.
- C-5 P. Mathiopoulos and K. Feher, "In-band pilot insertion at the Nyquist frequency for transparent signalling," in the *Proceedings of GLOBECOM '87*, Tokyo, Japan, pp. 2063-2068.
- C-4 P. Mathiopoulos and K. Feher, "512- and 1024-QAM system performance in the presence of cochannel interference," in the *Proceedings of MIAMI TECHNICON '87*, Miami, U.S.A., pp. 74-77.
- C-3 P. Mathiopoulos and K. Feher, "Adjacent channel interference considerations for 1024- and 512-QAM systems," in the *Proceedings of MONTECH '87*, Montreal, Canada, pp. 47-50.
- C-2 P. Mathiopoulos and K. Feher, "A new technique for system caused phase jitter cancellation," in the *Proceedings of ICC '87*, Seattle, U.S.A., pp. 1451-1457.
- C-1 P. Mathiopoulos, H. Ohnishi and K. Feher, "A study of 1024-QAM modem performance in distorted linear channels and a method of cancelling sinusoidal phase jitter," in the *Proceedings of MONTECH '86*, Montreal, Canada, pp. 246-249.

Contents

Abstract	i
Acknowledgments	iv
Preface	v
Contents	vi
List of Figures	xi
List of Tables	xviii
List of Abbreviations	xx
1 INTRODUCTION	1
1.1 SPECTRALLY EFFICIENT DIGITAL MODULATION SYSTEMS	1
1.1.1 Summary of the Most Popular Spectrally Efficient Modulation Techniques	2
1.1.2 Comparison of the Spectral Efficiencies of QAM, PSK and QPR Schemes	4

1.1.3	QAM Systems Achieving Spectral Efficiencies of up to 10 b/s/Hz	7
1.1.4	Interference Encountered in Spectrally Efficient Digital QAM Transmission Systems	9
1.2	DEMODULATION METHODS FOR SPECTRALLY EFFICIENT QAM SYSTEMS	10
1.2.1	Transmission System Caused Phase Jitter (SCPJ)	11
1.2.2	Coherent Versus Noncoherent Demodulation of QAM Schemes	13
1.2.3	Coherent Digital QAM Systems	14
1.3	RESEARCH OBJECTIVES OF THIS THESIS	15
1.4	THESIS ORGANIZATION	17
2	OUT-OF-BAND PILOT-AIDED TECHNIQUES FOR SYSTEM CAUSED PHASE JITTER CANCELLATION	20
2.1	INTRODUCTION	20
2.2	A CHRONOLOGICAL NOTE	22
2.3	DPS-QAM RECEIVER ANALYSIS AND SCPJ MODELLING	24
2.4	EXACT SYMBOL ERROR RATE EVALUATION OF A 16-QAM SYSTEM IN THE PRESENCE OF SCPJ AND THERMAL NOISE	28
2.5	IMPACT OF THE IMPERFECT EXTRACTION FILTERS ON THE DPS 16-QAM SYSTEM PERFORMANCE	33

2.5.1	Wideband Systems	33
2.5.2	Narrowband Systems	38
2.6	DESIGN AND IMPLEMENTATION CONSIDERATIONS OF AN EXPERIMENTAL 16-QAM DPS	45
2.7	PERFORMANCE EVALUATION RESULTS OF THE EXPERIMENTAL 16-QAM DPS	49
2.7.1	Experimental Set-up Description	49
2.7.2	Experimental Measurements and Results	54
2.8	CONCLUSIONS	59
3	IN-BAND PILOT INSERTION FOR SIMULTANEOUS CARRIER AND SYMBOL TIMING RECOVERY IN AN AWGN CHANNEL	60
3.1	INTRODUCTION	60
3.2	IBDPS-QAM SYSTEM DESCRIPTION AND INTERFERENCE ANALYSIS	63
3.2.1	System Description	63
3.2.2	Pilot Interference Into the QAM Signal	64
3.2.2.1	Pilot Frequency Considerations	64
3.2.2.2	Pilot Level Considerations	67
3.2.3	Analysis of the QAM Signal Interference Into the Re- covered Carrier	67

3.3	COMPUTER SIMULATION SYSTEM DESCRIPTION AND PERFORMANCE EVALUATION RESULTS	73
3.3.1	Computer Simulation System Description	73
3.3.2	Performance Evaluation Results	79
3.4	CONCLUSIONS	88
4	512- AND 1024-QAM SYSTEM PERFORMANCE IN DISTORTED LINEAR CHANNELS	90
4.1	INTRODUCTION	90
4.2	DESCRIPTION OF THE COMPUTER SIMULATED SYSTEMS	91
4.3	PERFORMANCE EVALUATION WITH CHANNEL IMPERFECTIONS	100
4.3.1	Amplitude Distortions	100
4.3.2	Group Delay Distortions	106
4.4	PHYSICAL INTERPRETATION OF THE OBTAINED RESULTS	109
4.5	GENERALIZATION OF THE OBTAINED RESULTS TO SYSTEMS WITH DIFFERENT SIGNALLING RATES	117
4.6	CONCLUSIONS	119
5	THE PERFORMANCE OF 512- AND 1024-QAM SYSTEMS IN A CO-CHANNEL INTERFERENCE	

ENVIRONMENT	121
5.1 INTRODUCTION	121
5.2 CCI SYSTEM MODEL AND DESCRIPTION OF THE COMPUTER SIMULATIONS	122
5.3 COMPUTER SIMULATED RESULTS AND DISCUSSION	125
5.4 THEORETICAL EVALUATION OF THE MAXIMUM TOLERABLE C/I FOR ERROR-FREE TRANSMISSION WITH NO GAUSSIAN NOISE	130
5.5 CONCLUSIONS	133
6 ERROR RATE CONSIDERATIONS FOR 512- AND 1024-QAM SYSTEMS IN THE PRESENCE OF ADJACENT CHANNEL INTERFERENCE	134
6.1 INTRODUCTION	134
6.2 ADJACENT CHANNEL INTERFERENCE MODEL AND COMPUTER SIMULATION DESCRIPTION	135
6.3 PERFORMANCE EVALUATION RESULTS AND DISCUSSION	138
6.4 CONCLUSIONS	139
7 CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH	142

7.1	CONCLUSIONS	142
7.2	SUGGESTIONS FOR FUTURE RESEARCH	144
7.2.1	Analysis of the DPS in the Presence of Other Forms of SCPJ	144
7.2.2	Performance of 512- and 1024-QAM in the Presence of Multipath Fading	144
7.2.3	Cancellation Methods for Co-Channel and Adjacent Channel Interferences	144
7.2.4	Effects of Nonlinearities on the Performance of 512- and 1024-QAM Systems	145
7.2.5	Application of Trellis Coded Modulation to Digital Microwave Radio	145
	BIBLIOGRAPHY	146
	A 16-QAM MODEM DETAILS	156
	B COMPUTER SIMULATION PROGRAM LISTINGS	167

List of Figures

1.1	Constellations of M -ary QAM schemes with $M = 4, 16, 32, 64, \dots, 1024$. I: in-phase carrier (channel) ; Q: quadrature-phase carrier (channel).	3
1.2	Theoretical spectral efficiency η_{th} in b/s/Hz of RF bandwidth and C/N requirement at $P_S = 10^{-8}$ of various spectrally efficient modulation systems.	6
1.3	Measured state-space diagram of a 1.544 Mb/s rate 256-QAM system corrupted by System Caused Phase Jitter (SCPJ).	12
2.1	Dual Pilot System (DPS) configuration.	23
2.2	Receiver structure of a DPS-QAM system.	27
2.3	Signal constellation changes of a 16-QAM system, which is corrupted by SCPJ γ	30
2.4	Impact of the amplitude β (in rads) of the sinusoidal SCPJ on the $P_S = f(C/N)$ performance of the 16-QAM system.	32
2.5	Linear phase responses of the ideal $PBPF_{\pm}$	35
2.6	Rotating phasor representation of the two terms of Eq. 2.16.	36
2.7	C/N degradation of the DPS 16-QAM system for wideband extraction filters and as a function of the constant group delay τ	37
2.8	Spectrum of the sinusoidal SCPJ and the nonideal transfer function of the pilot extraction filters.	41

2.9	C/N degradation of a 16-QAM DPS with narrowband pilot extraction filters as a function of their amplitude variation ϵ	42
2.10	C/N degradation of a DPS 16-QAM system with narrowband pilot extraction filters as a function of their phase distortion ϕ	43
2.11	Block diagram of the used experimental 16-QAM DPS.	47
2.12	Measured spectrum of the transmitted 16-QAM DPS without any Gaussian noise or SCPJ.	50
2.13	Measured signal space diagram of the demodulated 16-QAM DPS.	51
2.14	Block diagram of the experimental set-up used to test the 16-QAM DPS.	53
2.15	Photograph of the complete experimental set-up in the Digital Communication Laboratory of the Department of Electrical Engineering at the University of Ottawa.	54
2.16	Demodulated eye-diagrams of both I- and Q-channels of the 16-QAM modem, showing the ISI caused mainly by imperfect filtering.	56
3.1	In-Band Dual Pilot System (IBDPS) configuration.	62
3.2	Arbitrary sequence of transmitted symbols each having duration of T . The sampling instants $\dots, t_{n-1}^*, t_n^*, t_{n+1}^*, \dots$ are assumed to be the optimum (i.e. result in minimum ISI).	66
3.3	Data into the pilots interference considerations of the IBDPS-QAM system employing $\sqrt{\alpha}$ raised-cosine filters with the pilot tones inserted at the Nyquist frequency.	71
3.4	Plot of the variance of the carrier data phase jitter, σ_ϕ^2 , as a function of the pilot extraction bandwidth ψ (normalized to the symbol rate) for different values of the excess bandwidth α	74

3.5	Block diagram of the employed computer simulation model of a 16-QAM IBDPS.	76
3.6	4-level ideal baseband eye-diagram of a 16-QAM system which employs raised-cosine filters with $\alpha = 1.0$	77
3.7	I-channel eye-diagrams of a 800 kb/s, 16-QAM employing $\alpha = 1.0$ raised-cosine filters with the pilot tone inserted at different in-band frequencies.	80
3.8	C/N degradation, D_{f_p} , of the 16-QAM IBDPS at a symbol error probability $P_S = 10^{-4}$ as a function of the inserted pilot frequency f_p , normalized to the symbol rate $1/T$ ($m = -10$ dB, $\alpha = 1.0$).	81
3.9	Measured, demodulated eye-diagrams of the I-channel of a 16-QAM ($\alpha = 0.2$) IBDPS with the pilots inserted at the Nyquist frequency.	83
3.10	Measured spectrum of the transmitted 16-QAM ($\alpha = 0.2$) IBDPS with the pilots inserted at the Nyquist frequency.	85
3.11	C/N degradation D_{IBDPS} of the 16-QAM IBDPS at $P_S = 10^{-4}$ as a function of ψ and with α as a variable.	86
4.1	512- and 1024-QAM system block diagram.	92
4.2	Encoder and decoder structures for the QAM schemes under consideration.	93
4.3	State-space diagram of a 512-QAM system.	94
4.4	Block diagram of the computer simulation model which was employed to simulate the impact of the linear channel distortions on the performance of 512- and 1024-QAM systems.	96
4.5	24-level ideal baseband eye-diagram of a 512-QAM system with $\alpha = 0.2$ raised-cosine filters.	97
4.6	Eye-diagram of a 1024-QAM system using raised-cosine filters with a roll-off factor of $\alpha = 0.2$	98

4.7	Eye-diagram of a 135 Mb/s, 512-QAM using $\alpha = 0.2$ raised-cosine filters and a channel having parabolic amplitude distortion of $P_A = 0.004 \text{ dB}/(\text{MHz})^2$	101
4.8	Eye-diagram of a 150 Mb/s, 1024-QAM using $\alpha = 0.2$ raised-cosine filters and a channel having parabolic amplitude distortion of $P_A = 0.0025 \text{ dB}/(\text{MHz})^2$	102
4.9	Impact of different linear amplitude distortion coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s (135 Mb/s) 512-QAM system.	103
4.10	Impact of different parabolic amplitude distortion coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s (135 Mb/s) 512-QAM system.	104
4.11	Impact of different sinusoidal amplitude distortion coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s (135 Mb/s) 512-QAM system.	105
4.12	Degradation of C/N as a function of maximal amplitude distortion A_m for 512- and 256-QAM for linear, parabolic and sinusoidal amplitude distortions.	107
4.13	Impact of linear, parabolic and sinusoidal amplitude distortions on the P_S performance degradation as a function of A_m of the 1024-QAM system which employs $\alpha = 0.2$ raised-cosine filters and a symbol rate of 15 MBaud/s.	108
4.14	Impact of different linear group delay distortions coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s 512-QAM system.	110
4.15	Impact of different parabolic group delay distortion coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s 512-QAM system.	111
4.16	Impact of different sinusoidal group delay distortion coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s 512-QAM system.	112

4.17	Degradation of C/N as a function of maximal group delay τ_m for 512- and 256-QAM for linear, parabolic and sinusoidal group delay distortions.	113
4.18	Impact of linear, parabolic and sinusoidal group delay distortions on the P_S performance degradation as a function of τ_m of a 1024-QAM system employing $\alpha = 0.2$ raised-cosine filters and having a symbol rate of 15 MBaud/s.	114
4.19	Graphical representation in the frequency domain of an M -ary QAM, $\alpha = 0.2$, signal with linear, parabolic and sinusoidal amplitude distortions which have the same peak distortion A_m	116
5.1	Block diagram of the employed simulation model of the CCI system.	123
5.2	C/N degradation at a symbol error rate of $P_S = 10^{-6}$ for a 512-QAM, $\alpha = 0.2$ system in the presence of amplitude and constant envelope interferers.	126
5.3	C/N degradation at a symbol error rate of $P_S = 10^{-6}$ for a 1024-QAM, $\alpha = 0.2$ system in the presence of amplitude and constant envelope interferers.	127
6.1	Block diagram of the simulation model of a multichannel, adjacent channel interference system, employing 512- and 1024-QAM modulation formats.	137
6.2	C/N degradation, at a symbol error rate $P_S = 10^{-6}$, of 512- and 1024-QAM systems in the presence of ACI.	140
6.3	C/N degradation, at a symbol rate $P_S = 10^{-6}$, of 512- and 1024-QAM systems in the presence of ACI.	141
A.1	Front panel of the 16-QAM modem.	157
A.2	Assembly drawing of the 2 x 4-PAM encoder.	160
A.3	Schematic diagram of the 2 x 4-PAM encoder.	161

A.4	Schematic diagram of the quadrature modulator and the pilot insertion.	162
A.5	Schematic diagram of the carrier and symbol timing recovery subsystem.	163
A.6	Schematic diagram of the quadrature demodulator.	164
A.7	Assembly drawing of the 2 x 4-PAM decoder.	165
A.8	Schematic diagram of the 2 x 4-PAM decoder.	166

List of Tables

1.1	Spectral efficiencies requirements for some low bit rate (≤ 2.048 Mb/s) applications.	8
1.2	Spectral efficiencies requirements for high capacity digital radio applications.	8
2.1	Equivalent C/N degradation D_p due to the power which is required to transmit the pilot tones. m is defined as the ratio of the power of the pilot tones divided by the power required to transmit the digitally modulated signal.	45
2.2	Measured characteristics of the employed extraction band-pass filter sets $PBPF_{\pm}$	52
2.3	Comparison between the analytical results and the measured performance of a 16-QAM DPS operated in the presence of AWGN and sinusoidal SCPJ.	57
4.1	Relationships between the different parameters of amplitude, group delay distortions and symbol rates for two digital transmission systems having the same modulation format and performance but different symbol rates. The definitions of the distortion parameters L_D, P_D, S_D and L_A, P_A, S_A are given in Eqs. 4.10 and 4.6, respectively.	118

5.1	$(C/I)_{min}$ values required for error-free transmission with no Gaussian noise and corresponding peak factors (PF) of 512- and 1024-QAM systems employing raised-cosine filters with $\alpha = 0.2$ operating in the presence of amplitude modulated and sinusoidal (constant envelope) CCI.	132
A.1	Description of the various signals at the different connectors of the 16-QAM modem.	158

List of Abbreviations

ACI:	Ajacent Channel Interference
AGC:	Automatic Gain Control
AWGN:	Additive White Gaussian Noise
BPSK:	Binary Phase Shift Keying
CCI:	Co-Channel Interference
C/I :	Carrier-to-Interference Ratio
C/N :	Carrier-to-Noise Ratio
CR:	Carrier Recovery
CRCPJ:	Carrier Recovery Caused Phase Jitter
CRS:	Carrier Recovery Subsystem
CS:	Carrier System
DIV:	Data-in-Voice
DPS:	Dual-Pilot System
FBC:	Feedback Balanced Code
FFT:	Fast Fourier Transform
GMSK:	Gaussian Minimum Shift Keying
IBDPS:	In-Band Dual Pilot System
ISI:	Inter-Symbol-Interference
LGDS:	Linear Group Delay System
LI:	Lower Interferer
LO:	Local Oscillator
MC:	Main Channel
MSK:	Minimum Shift Keying
PAM:	Pulse Amplitude Modulation
PF:	Peak Factor
PSC:	Parallel-to-Serial Converter
QAM:	Quadrature Amplitude Modulation
QPR:	Quadrature Partial Response
QPSK:	Quadrature Phase Shift Keying

RCSG:	Random Complex Sequence Generator
SAW:	Surface Acoustic Wave
SCPJ:	System Caused Phase Jitter
SCS:	Supressed Carrier System
SSB:	Single Sideband
SPC:	Serial-to-Parallel Converter
STR:	Symbol Timing Recovery
TCM:	Trellis Coded Modulation
UI:	Upper Interferer
VSF:	Vestigial Sideband

Chapter 1

INTRODUCTION

1.1 SPECTRALLY EFFICIENT DIGITAL MODULATION SYSTEMS

In this introductory section a brief review of three well-known generic modulation schemes, and their applicability in highly bandwidth efficient digital transmission systems is presented. These are Quadrature-Amplitude-Modulation (QAM), Quadrature-Partial-Response (QPR) and Phase-Shift-Keying (PSK). In addition, a brief comparison of the three schemes having their theoretical spectral efficiency η_{th} (in b/s/Hz) versus the required Carrier-to-Noise ratio (C/N), for a given error rate, as performance criterion will be made. Conclusions will be presented as to which of the three schemes should be employed for transmission applications requiring high spectral efficiency. Also discussed is the recently developed 256-QAM scheme which achieves currently the highest practical spectral efficiency of $\eta_{pr} \doteq 6.7$ b/s/Hz. However, as will be seen, for a number of important applications, a further increase in spectral efficiency is necessary. Thus, two highly spectral efficient QAM schemes, i.e. 512- and 1024-QAM, are studied, which can achieve spectral efficiencies of up to 9 and 10 b/s/Hz, respectively. This thesis describes studies carried out on these two QAM schemes and in particular will be concerned with the evaluation of their

performance in a complex interference environment.

1.1.1 Summary of the Most Popular Spectrally Efficient Modulation Techniques

Over the last decade or so, the increasing use of radio transmission systems has led to congestion in many regions of the radio spectrum. This situation, and the increased emphasis on digital transmission have resulted in a need for developing and employing improved spectrum utilization techniques. Efficient digital modulation techniques provide the most important means of achieving improved spectral efficiency at a reasonable cost. In this context, the three most cost-effective and widely accepted two-dimensional modulation techniques are QAM, PSK and QPR. The general identifying characteristics of these techniques are explained below:

i) ***M*-ary QAM:**¹ A technique in which the information symbols are transmitted using a combination of M amplitudes and phases of the transmitted carrier, in such a way that the transmitted signal constellation has a rectangular form as shown in Fig. 1.1 [1]. Equivalently, we can state that the transmitted M -ary QAM signal consists of two orthogonal carriers (hence the term quadrature) modulated by a set of $\sqrt{M}$ ² discrete amplitudes, which consist a $\sqrt{M}$ Pulse Amplitude Modulation (PAM) scheme. QAM belongs to the more general family of Amplitude and Phase Keying (APK) systems [8]. Although there are a number of more complex (and therefore more difficult to implement) signal constellations within the APK family, they are known to perform only marginally better than QAM [1]-[7]. Thus it is widely accepted today that QAM represents the best tradeoff between theoretical performance and implementation complexity [6],[8],[9],[11].

ii) ***M*-ary PSK:** A second technique in which the transmitted symbols are used to produce M distinct carrier phase changes, which are, traditionally, uniformly distributed over $[0, 2\pi)$ [8],[10],[11]. In this case the transmitted constellation can be represented by M signal sets which all lie

¹ $M = 2^n$ with $n=2,4,5,6,7,\dots$

²This is valid for even values of n . For n odd the number of the discrete amplitudes will depend upon the choice of the constellation diagram.

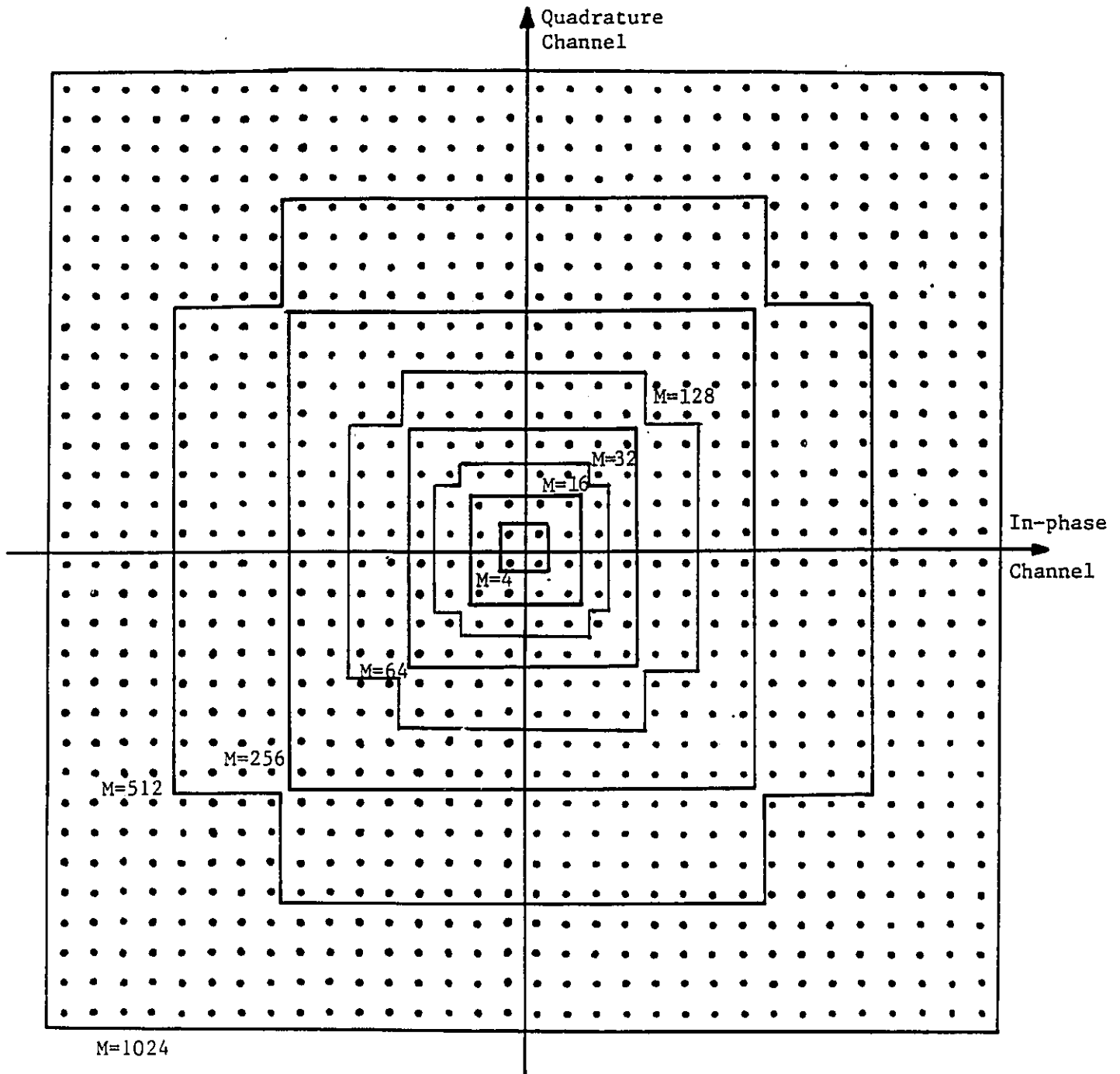


Figure 1.1: Constellations of M -ary QAM schemes with $M = 4, 16, 32, 64, \dots, 1024$.

I: in-phase carrier (channel) ; Q: quadrature-phase carrier (channel).

on a circle.³

iii) **M -ary QPR:** A third technique, in which two $\sqrt{M}$ -level partial response encoded signals are phase modulated in quadrature by two orthogonal carriers [12]. As is well known, the partial response encoding techniques deliberately introduce a controlled amount of Inter-Symbol-Interference (ISI) over a span of one, two or more symbols. As a result, certain beneficial effects are attained, such as convenient spectral shapes. Since the amount of the controlled ISI is known at the receiver, it can be subtracted from the sampled received signal. Thus, the transmitted data can still be detected correctly [13],[11]. One of the most popular and widely used partial response technique is the so-called class-I signalling, in which every symbol is added to the next symbol to be transmitted [13]. In this case the number of possible levels after passing a $\sqrt{M}$ -ary PAM signal through a partial response encoder (class-I) is $2\sqrt{M}-1$, and thus there are $(2\sqrt{M}-1)^2$ possible signal states for the QPR system.⁴

1.1.2 Comparison of the Spectral Efficiencies of QAM, PSK and QPR Schemes

Consider now the advantages and disadvantages of the above three transmission schemes, depending on the particular application in which they are to be employed.

One of the most important parameter of interest to many system design engineers is the theoretical spectral (or bandwidth) efficiency, η_{th} measured in b/s/Hz of RF bandwidth,⁵ which each of these modulation schemes can achieve. For purposes of this thesis, we define a communication system to be spectrally efficient if $\eta_{th} \geq 4$ b/s/Hz. Fig. 1.2 presents a summary of η_{th} versus the required C/N at a symbol error rate $P_S = 10^{-8}$ of the three spectrally efficient M -ary ($M \geq 16$) modulated systems under consideration.

³It can be argued that PSK systems do belong to the APK family and in fact for $M=4$ both 4-PSK and 4-QAM represent the same communication system. However it is instructive for our purposes here to treat them separately.

⁴To have a direct comparison with the previously described M -ary QAM schemes, we will assume again here that $M = 2^n$ with n even.

⁵We assume here that ideal minimum-bandwidth Nyquist filtering is used, i.e. raised-cosine filters with $\alpha = 0$.

Examination of Fig. 1.2 makes it clear that the QAM systems require the lowest C/N for any given η_{th} , thus making them the most spectral efficient schemes. For example, by using a 256-QAM system⁶ ($\eta_{th} = 8$ b/s/Hz) the obtained C/N gains are 2.4 dB and 15 dB as compared to the equivalent 961-QPRS and 256-PSK modulation schemes, respectively. These numbers indicate that the use of PSK is out of question, as far as bandwidth efficient applications are concerned. It should be mentioned, however, that although throughout this thesis we will consider systems operated in linear channels, M -PSK, due to its more constant envelope as compared to M -QAM, could be a good candidate for nonlinear channel applications.

QPR systems are also inferior to QAM schemes in terms of C/N ratio for the same theoretical bandwidth efficiency. Nevertheless, as well known, the shape of their spectrum allows filtering which does not require any more excess bandwidth than the Nyquist bandwidth. In this sense, the practical achievable bandwidth efficiency η_{pr} of all QPR schemes coincide with η_{th} . However, this is not true for QAM schemes, where their η_{pr} is $[\eta_{th}/(1 + \alpha)]$, with α being the excess bandwidth, which is required by raised-cosine filters.

Based on the above, arguments can be made to whether or not in terms of η_{pr} , a 225-QPR scheme can outperform the performance of a filtered 256-QAM scheme [15]. In fact a study comparing these two schemes as candidates for the transmission of a 1.6 Mb/s (T1 rate digital signal: 1.544 Mb/s plus overhead bits) data over a standard analog supergroup band of 240 kHz (T1/SG), concluded that for $\eta_{pr} > 6.3$ b/s/Hz, the 256-QAM modulation format was most appropriate for practical use [16]. 256-QAM was also chosen, independently of the author of [16], as the best candidate for applications requiring $\eta_{pr} > 6$ b/s/Hz by a number of researchers [15],[17]-[26]. As a result, over the last 2-3 years a number of 256-QAM digital transmission systems have been designed, implemented, field tested and are currently in operation both in North America and Japan [15],[18]-[26].

It is important to note here, that although in the past the design of raised-cosine filters with small α , i.e. $\alpha < 0.5$, was not considered to be straightforward, today previous realization problems with these filters have

⁶As it will become apparent later on in this section, the 256-QAM modulation technique represents the most spectrally efficient implemented digital communication system as of today.

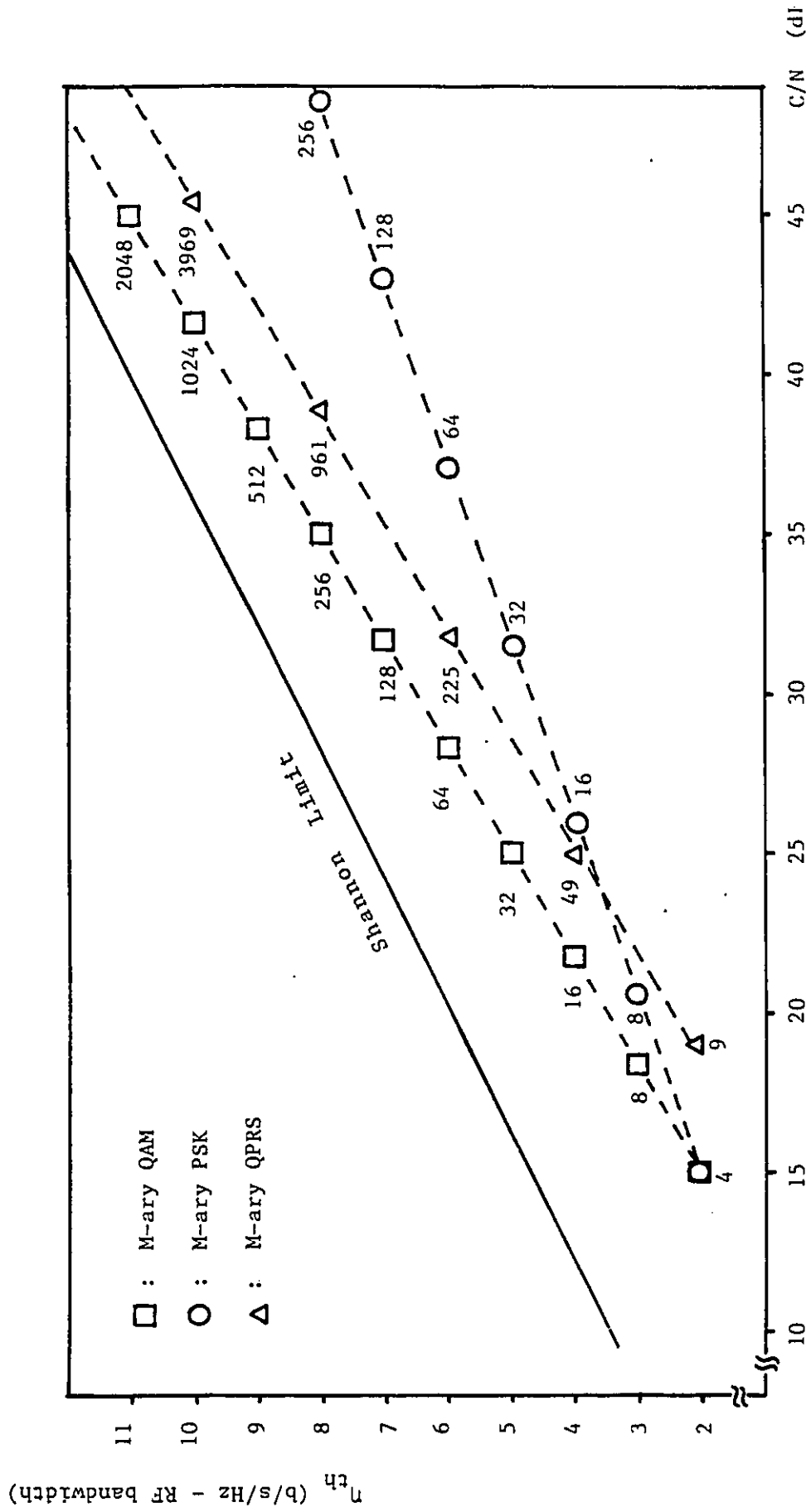


Figure 1.2: Theoretical spectral efficiency η_{th} in b/s/Hz of RF bandwidth and C/N requirement at $P_S = 10^{-8}$ of various spectrally efficient modulation systems. The average C/N is specified in the double-sided Nyquist bandwidth which equals the symbol rate. Since we refer to η_{th} ideal $\alpha = 0.0$ raised-cosine filters have been assumed. This figure is based on information provided in Refs. [10] and [14].

been considerably reduced. Examples of implemented raised-cosine filters with $\alpha \doteq 0.2$ can be found in a number of publications, including [15],[23] and [27]. A promising technology for the fabrication of filters with low roll-off factor seems to be the Surface Acoustic Waves (SAW). In [27] an implementation of raised-cosine filters with $\alpha \doteq 0.19$ using SAW modules is reported. In addition, other design results have been reported for the implementation of raised-cosine filters with $\alpha \doteq 0.08$ [28],[30]. Furthermore, adaptive equalizers employing more than 200 taps have been used to ensure the ISI caused by any residual filter distortion is essentially eliminated [29].

As a summary, it is fair to state that with today's advanced technology, raised-cosine filters with very small roll-off factors can be designed and implemented without major technical difficulties. In this sense, the excess bandwidth α which is required for the transmission of QAM schemes, can be minimized, thus eliminating their carrier bandwidth disadvantage compared with equivalent QPR schemes. Furthermore at present, the 256-QAM modulation technique represents the most spectrally efficient implemented means of transmitting digital information, achieving a practical bandwidth efficiency of $\eta_{pr} \doteq 6.7$ b/s/Hz.

1.1.3 QAM Systems Achieving Spectral Efficiencies of up to 10 b/s/Hz

From the previous discussion and further as suggested by a number of leading authorities in the digital transmission field, for further bandwidth efficiencies increases to, for example, $\eta_{th} > 8$ b/s/Hz, the QAM modulation format is the most suitable method to be employed [29],[31],[17]-[26],[32]. As it can be seen from Fig. 1.2, 512- and 1024-QAM achieve theoretical spectral efficiencies of 9 and 10 b/s/Hz, respectively. There are a number of applications where such high spectral efficiencies are needed. We present below a summary of some of the important applications for both low bit rate transmission (≤ 2.048 Mb/s), such as Data-in-Voice (DIV) and basic telephone systems, as well as emerging high capacity radio applications in Tables 1.1 and 1.2, respectively.

BIT RATE (kb/s)	ALLOWABLE BANDWIDTH (kHz)	REQUIRED BANDWIDTH EFFICIENCY (b/s/Hz)	APPLICATION
2048	240	8.5	DIV Transmission
19.2	2.4	8.0	Telephone channel

Table 1.1: Spectral efficiencies requirements for some low bit rate (≤ 2.048 Mb/s) applications.

AUTHORIZED FREQUENCY BAND (GHz)	ALLOWABLE BANDWIDTH (MHz)	BIT RATE (Mb/s)	REQUIRED BANDWIDTH EFFICIENCY b/s/Hz
3.700–4.200	20.0	2x90	9.0
5.925–6.425	30.0	2x90	6.0
		2x135	9.0
		2x140	9.3
10.700–11.700	40.0	4x90	9.0
		2x135	6.75
		2x140	7.0

Table 1.2: Spectral efficiencies requirements for high capacity digital radio applications.

1.1.4 Interference Encountered in Spectrally Efficient Digital QAM Transmission Systems

The practical application of any form of digital transmission modulation scheme demands that serious consideration be given to the interference characteristics of the equipment in the system. In order to achieve a successful (ideally optimum in some sense) engineering design, it is essential that the degradation in the performance be known when different inherent source of interference (distortions) are present as well as when a signal other than the desired one falls within the receiver passband. Quantitative analysis of interference depends critically upon the equipment characteristics and performance, actual channel bandwidth, spectral occupancy regulations and other operating frequency constraints set by the regulatory agencies.

The recent widescale employment of highly bandwidth efficient modulation formats has resulted in systems which are very sensitive to many forms of interference. Furthermore, the increase in congestion of frequency spectrum and use of satellites coupled with the development of various frequency reuse techniques and increased use of higher bands have emphasized the importance of various types of interference in defining performance of a transmission system for many applications.

A general treatment of interference would be necessarily very extensive. It is an almost impossible task to present, analyze and evaluate a model which includes all important forms of interference. Instead, research carried out in this thesis will focus on three of the most important sources of *data dependent* interference in the context of the highly spectral efficient QAM systems. In particular, we will evaluate both 512- and 1024-QAM in the following interference environments:

1. Inter-Symbol-Interference (ISI)
2. Adjacent-Channel-Interference (ACI)
3. Co-Channel-Interference (CCI).

A brief description of these types of interference is given below, with a more complete description and analysis to be given in Chapters 4-6:

- **ISI:** This term refers to interference resulting from distortion of one symbol by preceding and forthcoming symbols. The typical source of ISI is the deviation of filter characteristics from those of ideal amplitude and/or phase characteristics of Nyquist-type filters.
- **ACI:** This term refers to interference due to the presence of one or more radio carriers (modulated or otherwise) in frequency bands adjacent to that of the wanted modulated carrier.
- **CCI:** This type of interference arises from the presence of a signal (modulated or otherwise) centered at a frequency very close to that of the wanted carrier. In principle, this class of interference is a special case of ACI with very small frequency separation between the desired and undesired signals.

1.2 DEMODULATION METHODS FOR SPECTRALLY EFFICIENT QAM SYSTEMS

In this section we shall briefly summarize the most important demodulation methods employed in spectrally efficient QAM systems. It is important to clarify that although many of the methods, which we will describe here and in more detail later on in this thesis, can be applied to other less spectrally efficient modulation schemes as well, *our purpose in this thesis is to restrict our analysis and evaluation to QAM modulation formats only.*

We will begin this section by introducing the concept of transmission System Caused Phase Jitter (SCPJ) and the negative impact it can have on a system's performance. Then, the most important trade-offs between coherent and noncoherent demodulation methods of spectrally efficient QAM will be presented. As coherent demodulation is the most suitable method of demodulation for spectrally efficient QAM schemes, we present a comparison between suppressed carrier and carrier (i.e. pilot-aided) coherent QAM systems. We will end this section by concluding that in SCPJ environments, pilot based coherent QAM schemes appear to be the most appropriate to be employed in spectrally efficient system applications.

1.2.1 Transmission System Caused Phase Jitter (SCPJ)

It is well known that time-varying propagation and transmission system characteristics on long-haul terrestrial line-of-sight microwave links can distort the amplitude and phase of digitally modulated signal [10]. Fading and/or transmission SCPJ⁷ are factors of particular concern in coherent spectrally efficient schemes such as 16-QAM, 64-QAM up to 1024-QAM systems. Excessively high levels of additive white Gaussian noise (AWGN) combined with transmission SCPJ may lead to significant performance degradation and, under particularly severe transmission conditions, may also lead to even unacceptable error floors.⁸

In distinction to the Carrier Recovery Caused Phase Jitter (CRCPJ), [40],[37],[38], which is introduced in the carrier recovery subsystem, SCPJ is typically introduced by local oscillators, up- and down-converters, frequency multipliers and/or translators and even by Automatic Gain Control (AGC) amplifiers, i.e. components which are not part of the demodulator. All these units are located *within the communication system but prior to its carrier recovery subsystem*. The effects of SCPJ can be particularly severe in non-regenerative digital or hybrid (e.g. 1.544 Mb/s DIV) microwave systems in which up to 30 radio hops may be used prior to the regeneration [29],[32],[15]. Fig. 1.3 illustrates a measured state-space diagram of a 1.544 Mb/s rate 256-QAM system corrupted by SCPJ, where a pure sinusoid carrier signal is provided as the recovered carrier of the 256-QAM receiver. Notice that the outer dots, which are more smeared than the inner ones due to the phase rotation clearly indicate the phase jitter nature of the impairment. This field measurement result was obtained over a section of a long-haul commercial 6 GHz microwave system [15].

Most of the research efforts today have been devoted to analyze and reduce the effects of the phase jitter caused by the noise and/or Carrier Recovery Subsystem (CRS) (e.g. frequency instabilities, CRCPJ e.t.c.)

⁷Phase jitter is the equivalent time domain representation of phase noise, which is usually described in the frequency domain by a power spectral density function [33]-[35]. In this thesis, we will maintain the time domain representation since in our analysis it is more convenient to work with.

⁸An error floor occurs when the bit error rate performance does not improve even when the carrier power is increased.

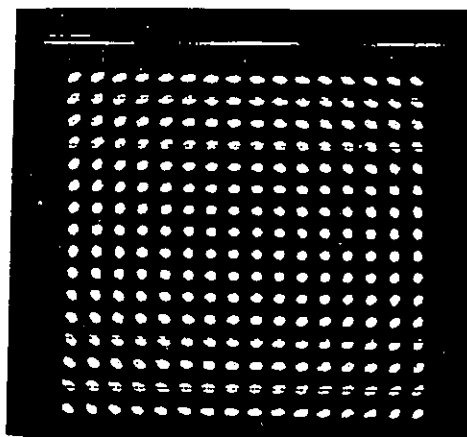


Figure 1.3: Measured state-space diagram of a 1.544 Mb/s rate 256-QAM system corrupted by System Caused Phase Jitter (SCPJ). This field measurement was performed over a section of a long-haul commercial 6 GHz microwave radio. Photograph courtesy of Karkar Electronics, Inc., (a unit of Signal General), San Francisco, CA 94103, USA.

[14],[33],[38],[39],[98]. The SCPJ, however, would remain in the demodulated signal even if a pure sinusoidal carrier signal is provided by the receiver's CRS and therefore distort the information carrying phase of the signal.

1.2.2 Coherent Versus Noncoherent Demodulation of QAM Schemes

Coherent demodulation of digital communications signals requires a receiver capable of determining or estimating within acceptable error both the frequency and the phase of the signal carrier. The performance of coherent systems is optimum in the presence of additive white Gaussian noise (AWGN). In general the receivers used in these systems employ some type of suppressed carrier loop for establishing a synchronous reference. However, in applications where SCPJ is considerable, most of the carrier recovery methods do not provide an accurate reference for demodulation purposes. Nevertheless, as we will explain in Section 1.2.3, their performance in such an environment can be improved considerably, by employing insertion/extraction of pilot tones for synchronization purposes.

For noncoherent demodulation, the requirement of carrier recovery is waived and thus instead of a CRS, traditionally a one-symbol delay element is used for comparison of two consecutive symbols (i.e. differential detection) [11]. The main advantages of noncoherent demodulation systems are ease of implementation, fast synchronization and good performance in channels with phase jitter. On the negative side, the bit error rate performance of noncoherent schemes in an AWGN channel are considerably inferior to that of equivalent coherent systems. For example, when compared to their equivalent coherent detection QPSK⁹ which is differentially detected suffers a 2.3 dB degradation, while the constant envelope Gaussian Minimum Shift Keying (GMSK)¹⁰ suffers about 7 dB degradation (when the premodulation filter time-bandwidth is 0.25) [36]. Further, the degradation of more spectrally efficient schemes, such as 8-PSK or 16-QAM,

⁹QPSK is the same modulation scheme as 4-QAM and 4-PSK of Section 1.1.1.

¹⁰As well known, GMSK is a QPSK-type scheme employing Minimum Shift Keying (MSK) modulation and in which the transmitted data are passed through a Gaussian premodulation lowpass filter.

which are employing noncoherent detection is expected to be even higher. Thus the use of noncoherent schemes is essentially out of consideration for schemes which have high spectral efficiency.

1.2.3 Coherent Digital QAM Systems

In general, coherent digital communication systems can be divided into the following two categories, as far as their synchronization methods are concerned [42],[43]:

1. **Suppressed Carrier Systems (SCS)**, in which no unmodulated carrier power is transmitted with the digitally transmitted signal. These systems require a nonlinearity at the receiver and some type of suppressed carrier tracking loop for establishing a synchronous reference.
2. **Carrier Systems (CS)**, in which the data bearing signal is modulated on an RF carrier or carriers (the so-called pilots) in such a way that residual spectral components of these pilots exist in the overall power spectrum. These spectral components can be tracked by a narrowband phase-locked loop (or an equivalent bandpass filters) to provide the desired reference signal.

The synchronization aspects of the SCS have been rather extensively analyzed and evaluated in the presence of additive white Gaussian noise in numerous original research papers. The state of knowledge of SCS has been summarized in a number of books, including Lindsey [37], Lindsey and Simon [38], Spilker [33] and Bhargava *et al.* [39]. These references have, for the most part, described two types of tracking loops, namely the Costas loop and the data-directed loop. Although these loops are on first examination, attractive for many applications they suffer from a number of drawbacks such as self-noise (or pattern-noise) [40], slow acquisition time (which results very frequently in the so-called 'hang-up' phenomenon [41]), and in many practical applications, complicated and therefore inherently costly system configurations. These drawbacks are especially important for systems employing spectrally efficient modulation formats, due to their

increased system sensitivity to any deterioration of the transmission conditions. Additionally, in a SCPJ environment, these loops can fail to provide an accurate estimate of the carrier phase. As a solution to these problems auxiliary low-level pilot tone signals are transmitted together with the digitally modulated signal, thus constituting a CS [42],[43]. This solution can be very effective in a SCPJ environment, if the frequencies of the pilot signals are very close to the bandwidth occupied by the digital transmitted signal. In this way, the effects of SCPJ on both the auxiliary pilots and digital signal can be assumed to be practically the same. Therefore, *by extracting the pilots, i.e. tracking the unwanted SCPJ, it can be cancelled out during the demodulation process.*

Recently, a number of papers have appeared describing such pilot-aided modulation techniques that seek to reduce the effects of multipath fading at the receiver of a mobile-satellite link. All of these methods employ pilot tones which are *out-of-band, that is, they are transmitted in a part of the spectrum which is not occupied by the digital transmitted signal.* We will present a brief description and comparison of these techniques at the beginning of Chapter 2.

As a summary of our findings we can immediately state that pilot based synchronization methods appear to be the best candidates for achieving high performance with spectrally efficient modulation schemes which are operated in the presence of SCPJ.

1.3 RESEARCH OBJECTIVES OF THIS THESIS

The foregoing discussion on pilot-aided, spectrally efficient QAM systems which are operated in the presence of transmission SCPJ and additive white Gaussian noise has established the background to the research tasks performed, and the results obtained, in this thesis. These are summarized below:

1. Introduce and analyze a method based on the use of two out-of-band pilot tones for cancelling the effects of the transmission SCPJ on

the performance of QAM systems. Our method employs wideband pilot extraction filters, instead of the frequently used narrowband ones. It is applied to a 16-QAM system which is operated in a combined additive white Gaussian noise and sinusoidal SCPJ environment. We present a number of analytical as well as computer simulated results for different system parameters, such as extraction filter bandwidth and particular distortions, as variables. These results indicate that performance gains of up to 8 dB over conventional non-pilot based schemes are possible for a 11.5 degrees (zero-to-peak) sinusoidal SCPJ. An important part of the work reported in this chapter is devoted to the design and implementation considerations of an experimental 16-QAM system. The scaled down (in bit rate and carrier frequency) hardware implemented 16-QAM, 800 kb/s modem includes transmission of the two out-of-band pilot tones for both carrier and symbol timing recovery purposes as well as cancelling the effects of the transmission SCPJ. A number of different experimental set-ups and extensive measurements were used to verify some of the results obtained by analysis and computer simulation. The proposed technique can also be employed in more highly spectral efficient schemes, such as 64-, 256-, 512- and 1024-QAM.

2. Introduce a novel synchronization method for QAM schemes, which employs two in-band pilot tones for simultaneous carrier and symbol timing recovery. The proposed method does not require an excess bandwidth to accommodate the pilot tones, as do all other previous pilot-based methods, and at the same time it employs a very simple, and therefore cost-effective receiver structure. It is shown that the only in-band pilot frequency which does not cause any interference into the digitally modulated signal and at the same time provides carrier and symbol synchronization at the receiver is the Nyquist frequency. By analysis, computer simulation and measurements we evaluated a 16-QAM scheme which employs the proposed in-band, at the Nyquist frequency pilot method in an AWGN channel. Our results indicate that, as long as the pilot extraction filter bandwidth is relatively small, the overall 16-QAM system performance has a negligible degradation. Like the out-of-band pilot insertion the proposed in-band, at the Nyquist pilot insertion technique can be also

used in more spectral efficient systems, such as 64-, 256-, 512- and 1024-QAM.

3. Present a performance study of the highly spectral efficient 512- and 1024-QAM schemes, which can achieve spectral efficiencies of up to 9 and 10 b/s/Hz, respectively. Such high spectral efficiencies are required for a number of digital transmission applications including the transmission of a CEPT-1 (2.048 Mb/s) in a 240 kHz supergroup band, transmission of a 19.2 kb/s rate signal into a single telephone channel (2.4 kHz), transmission of two 90 Mb/s rate signals into a 20 MHz RF bandwidth. Due to these emerging requirements, and to satisfy our curiosity in regards to the feasibility of such high spectral efficiencies, we analyzed and evaluated the performance of these two QAM schemes in the presence of three of the most important sources of data dependent system interferences, namely:

- Inter-Symbol-Interference (ISI)
- Adjacent-Channel-Interference (ACI)
- Co-Channel-Interference (CCI).

Our study indicates, that under certain conditions both schemes are implementable and we foresee that they will be employed in the future for applications requiring spectral efficiencies of up to 10 b/s/Hz.

1.4 THESIS ORGANIZATION

Including the introductory chapter, this thesis consists of seven chapters and two appendices, and is organized as follows.

Chapter 2 deals with the analysis and evaluation of a system termed as Dual-Pilot System (DPS), which employs a pilot tone pair to successfully cancel the unwanted effects of SCPJ on the performance of QAM modulated signals. The first two sections briefly introduce the most important previously employed pilot-based techniques and review their applicability and/or limitations in different communication system environments. Furthermore, the DPS is described and its advantages over the other pilot-aided schemes are discussed. In Section 2.3 we present a general analysis of a QAM system

which employs the DPS and we adopt a mathematical model for the SCPJ. Section 2.4 describes a method of analysis which will be used to evaluate the exact symbol error probability performance of a 16-QAM system in the presence of SCPJ and thermal noise. In Section 2.5 the effects of a number of system imperfections, such as pilot extraction filter bandwidth and distortions, are analytically derived. Hardware design considerations and implementation complexity issues of an experimental 16-QAM DPS are discussed in Section 2.6. Further, in Section 2.7 a number of experimentally measured results are compared with the analytical ones. The chapter ends with Section 2.7, where its conclusions are presented.

Chapter 3 introduces a novel in-band pilot insertion technique, namely the In-Band Dual Pilot System (IBDPS). After the introductory section, Section 3.2 presents a detailed description and analysis of the proposed IBDPS as it can be employed in conjunction with an QAM signalling scheme. Extensive computer simulated as well as measured performance evaluation results of an experimental 16-QAM IBDPS are presented in Section 3.3. Finally, Section 3.4 contains the conclusions of this chapter.

Chapter 4 deals with the evaluation of the effects of amplitude and group delay distortions on the performance of 512- and 1024-QAM systems. Section 4.2 contains the description of the employed simulation model. In Section 4.3 the obtained performance evaluation results are presented. In Section 4.4 the previously obtained computer simulation results are physically analyzed and interpreted. Further, we present an efficient technique of generalizing the obtained results in Section 4.5. The chapter is concluded with Section 4.6.

Chapter 5 presents a systematic evaluation, by means of computer simulation and some analysis, of the performance of both 512- and 1024-QAM systems in the presence of CCI. After a short introduction, Section 5.2 briefly describes the computer simulated CCI system model. In Section 5.3 the obtained computer simulated performance evaluation results are presented and analyzed. In order to verify some of these results a theoretical analysis is included in Section 5.4. The conclusions of the chapter are contained in Section 5.5.

Chapter 6 is devoted to the study of the performance of 512- and 1024-QAM systems in the presence of ACI. After a short introduction,

Section 6.2 describes the employed ACI system model. Section 6.3 presents and analyzes the obtained performance evaluation results. Finally, the conclusions of the chapter are given in Section 6.4.

Chapter 7 summarizes in Section 7.1 the contributions of this work. Furthermore, it presents in Section 7.2 a number of suggestions and ideas for future research tasks.

Detailed circuit diagrams of the 16-QAM DPS modem and a brief description of the signals involved are given in **Appendix A**.

Finally, the listings of the programs which were used to evaluate the error performance of the different QAM systems are presented in **Appendix B**.

Chapter 2

OUT-OF-BAND PILOT-AIDED TECHNIQUES FOR SYSTEM CAUSED PHASE JITTER CANCELLATION

2.1 INTRODUCTION

The use of auxiliary tones for the purpose of synchronization in digital communication systems goes back in the early 60's. For example Bennett and Davey report in [43] a method of transmitting reference carrier information by employing a sinusoidally modulated tone outside of the data signal bandwidth. The frequency of this auxiliary tone and also the frequency of its modulation are chosen to be harmonically related to the frequency of the data carrier. At the receiver they describe a CRS by which both the original carrier and any amount of frequency offset due to drifts can be obtained from the pilot signal. Lucky *et al.* describe in [42] a method of inserting a pilot tone at the carrier frequency for Vestigial-Sideband (VSB) and Single-Sideband (SSB) transmission applications. However, for VSB systems to ensure that the extracted pilot tone is free of data signal energy,

the baseband data signal is passed through a high-pass filter. This filtering guarantees that after modulation no data signal energy will exist around the carrier. Nevertheless, the loss of low-frequency energy has the effect of adding a great deal of ISI. This condition can be corrected at the receiver by employing dc restoration by which the demodulated baseband signal is processed by a feedback circuit which essentially reinserts the low frequency of the digital signal which were removed at the transmitter [44],[45].

A different approach to the same problem is to employ coding schemes, such as Manchester coding [38], delay modulation [38], Feedback Balanced Codes (FBC) [46] e.t.c., to suppress the spectral power of the digital signal near the carrier. In all cases this is achieved by increasing the required transmission bandwidth. Here we will briefly describe the FBC scheme, since it is the most bandwidth efficient spectrum suppressing coding method.

The basic idea behind this coding scheme is to ensure that the summation of all transmitted digital symbols is on the average zero, thus creating a null at the DC of the baseband spectrum or equivalently at the carrier frequency [46]. This is achieved by first dividing the transmitted symbols into frames (groups) of N symbols. In each of these groups one framing symbol, which does not carry digital information, is inserted. Summation of all the data in one frame is calculated and its sign is compared to that of the summation over all frames so far. If the summation of the frame has the same sign as the summation over the sum of all foregoing frames, all the data in the frame are inverted. The framing symbol carries the information whether the data are inverted or not. The disadvantages of this coding scheme is that it increases the transmission bandwidth requirement due to the insertion of the redundant framing bit and also that it increases considerably the system complexity both at the transmitter and receiver. Different versions of FBC codes having been applied to high capacity digital radio systems are given in the recent references [47]-[49].

During the last ten years or so, there has been a growing interest in employing pilot tones for digital mobile and digital line-of-sight microwave system applications. As a result a number of papers have appeared in the open technical literature describing techniques that seek to reduce the effects of multipath fading at the receiver of land mobile and mobile-satellite links through the use of pilot-based calibration processes [50]-[58].

Such pilot-aided modulation techniques have appeared under various titles such as Feed-Forward-Signal-Regeneration [53], Transparent-Tone-In-Band [54],[55], Dual-Pilot-Tone-Calibration-Technique which is also referred to as Dual-Pilot-System (DPS) [56] and Tone-Calibrated-Technique [57],[58].

The basic concept of all these techniques is to transmit pilot tones at convenient frequencies in the vicinity of the data signal and then extract the pilots and subsequently from them recover the carrier at the receiver with the channel impairments intact. From all these schemes the DPS seems to be the best compromise between bandwidth efficiency and system complexity. Furthermore, the performance evaluation results for a Binary PSK (BPSK) system, which are reported in [56] indicate that it outperforms all other equivalent pilot-aided modulation techniques. For mobile radio applications, having a relatively low bit rate (typically 4.8 kb/s) the DPS has some limitations, including error floors. However, for higher rate systems (typically above 1.544 Mb/s) DPS have been found experimentally to be most valuable in SCPJ cancellation [32],[15].

Fig. 2.1(a) illustrates the transmitter DPS signals, whereas Fig. 2.1(b) represents its carrier extraction subsystem. In [56] apart from the fact that the analysis was restricted to a BPSK scheme operated in a mobile-satellite environment, one of the assumptions used was that the characteristics of the pilot extraction filters $PBPF_+$ and $PBPF_-$ have ideal amplitude and phase characteristics. However it is well known that in practice filters always introduce distortion and therefore the matter has to be investigated further. Moreover, in applications where SCPJ is the main source of degradation (for example in microwave digital radio systems) the DPS performance should be analyzed and evaluated employing more spectrally efficient modulation schemes. Thus the objective of this chapter is to analyze the DPS system performance in a SCPJ environment by employing a 16-QAM signaling scheme. However, before we proceed with the system description, analysis and evaluation a chronological note is in order.

2.2 A CHRONOLOGICAL NOTE

Although the DPS configuration [56] was first published in a refereed journal (*IEEE Transactions on Vehicular Technology*) in the May 1986 issue,

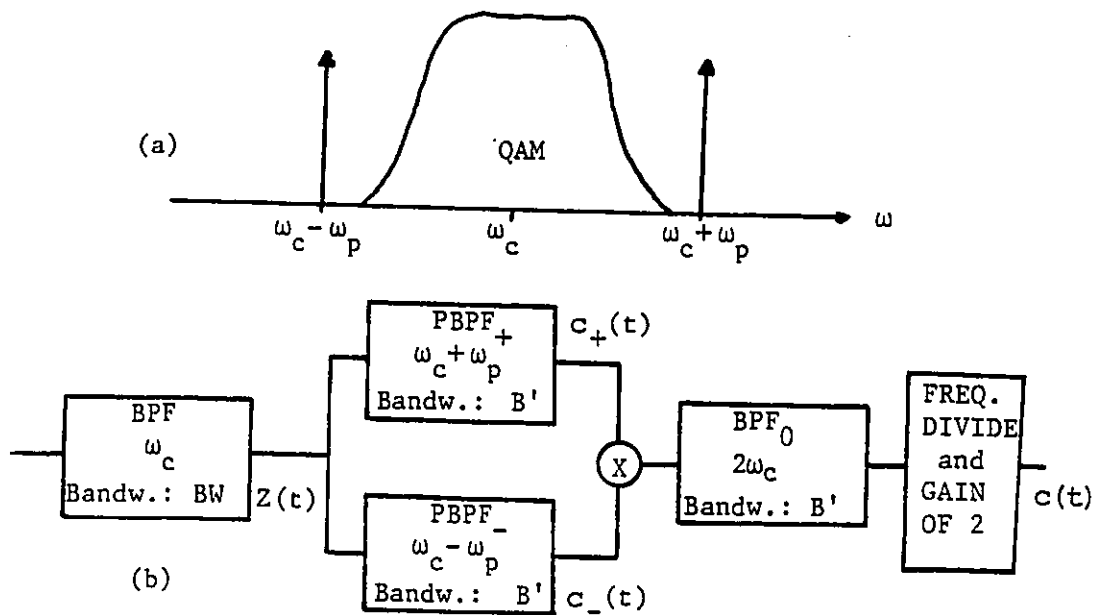


Figure 2.1: Dual Pilot System (DPS) configuration:

- (a) Transmitted signal power spectrum.
- (b) Carrier extraction subsystem used for QPSK as well as for more spectrally efficient systems, such as 16-QAM, 64-QAM,....

we became aware of this paper at the end of December 1986. We began independently to investigate the DPS scheme as it applies to SCPJ cancellation in microwave digital radio systems in March 1985. Some of the methodologies and results which will be presented in this chapter and also in Chapter 3 which employ the DPS configuration were included in the following publication of ours:

1. Ref. [59], which was completed in May 1986 and was presented to my Ph.D. advisory committee in June 1986.
2. Ref. [60], which was completed in August 1986.
3. Ref. [61], which was a preliminary version of Ref. [63].
4. Ref. [59] contains some preliminary results of [64], which employs the same DPS for in-band pilot insertion applications. A detailed description and analysis of the in-band pilot insertion scheme will be presented in Chapter 3.
5. Finally, most of the results of this chapter were published recently in the *IEEE Transaction on Broadcasting* [105].

2.3 DPS-QAM RECEIVER ANALYSIS AND SCPJ MODELLING

The received digitally modulated signal plus the two pilots are corrupted by SCPJ, denoted as $\phi(t)$, and additive Gaussian noise with a double-sided power spectral density of $N_0/2$. Mathematically, these signals can be represented as:

$$\begin{aligned}
 Z(t) = & A_p \cos[(\omega_c + \omega_p)t + \phi(t)] \\
 & + A_p \cos[(\omega_c - \omega_p)t + \phi(t)] \\
 & + S_I(t) \cos[\omega_c t + \phi(t)] + S_Q(t) \sin[\omega_c t + \phi(t)] \\
 & + N_i(t) \cos(\omega_c t) + N_q(t) \sin(\omega_c t).
 \end{aligned} \tag{2.1}$$

In Eq. 2.1 the first two terms represent the pilot tones of amplitude A_p and of radian frequencies $\omega_c \pm \omega_p$, whereas the last two terms represent the

bandpass Gaussian, i.e. thermal noise, with each of its components $N_i(t)$ and $N_q(t)$ having a variance of $N_0(BW)/2$, with BW being the equivalent noise bandwidth of the noise bandlimiting predetection filter BPF (Fig. 2.1). The M -ary QAM signal is represented by the third and fourth term of Eq. 2.1, where ω_c is the carrier radian frequency and $S_I(t)$ and $S_Q(t)$ are given by:

$$\begin{aligned} S_I(t) &= \sum_{n=-\infty}^{\infty} I_n g_T(t - nT) \\ S_Q(t) &= \sum_{n=-\infty}^{\infty} Q_n g_T(t - nT). \end{aligned} \quad (2.2)$$

I_n and Q_n are independent random variables representing the data information (symbols). They are taking equally likely values from the alphabet $\{\pm 1, \pm 3, \dots, \pm \sqrt{M} - 1\}$ for $M = 4, 16, 64, \dots$, T is the symbol duration and $g_T(t)$ is the impulse response of the baseband transmit filter.

In general, the extracted pilot signals $c_{\pm}(t)$ will depend on both thermal noise, which has a variance of $N_0 B'/2$, with B' the pilot extraction bandwidth, and on the transfer function of the extraction filters $PBPF_{\pm}$. However, it is shown in [56] that the degradation for a BPSK system, due to the thermal noise which corrupts the extracted pilots, is negligible if $B'T \leq 0.02$. Therefore, under this assumption the thermal noise effects on the extracted pilot signals can be neglected. It should be noted that this assumption would be valid for a more spectrally efficient modulation scheme, if the product $B'T$ was even smaller. As an example, we intuitively expect that for a 16-QAM system $B'T < 0.001$ should be enough. In any case, for the rest of this chapter we will assume that $B'T$ is small enough, so that the thermal noise effects on the extracted pilots can be neglected.

In view of this, the extracted pilot signals can be written as :

$$c_{\pm}(t) = A_p^{\pm} \cos[(\omega_c \pm \omega_p)t + \delta_{\pm}(t)] \quad (2.3)$$

with $A_p^{\pm}$ and $\delta_{\pm}(t)$ being the distorted amplitude and phase characteristics, respectively, due to the transfer function of the $PBPF_{\pm}$. As shown in Fig. 2.1(b), to obtain the recovered carrier $c(t)$, $c_+(t)$ and $c_-(t)$ are multiplied, afterwards the harmonic at $2\omega_c$ is extracted and finally the resulting signal is frequency divided and amplified by 2. Thus, the recovered carrier can be

written as:

$$c(t) = A_p^+ A_p^- \cos[\omega_c t + \frac{\delta_+(t) + \delta_-(t)}{2}]. \quad (2.4)$$

As well known, this frequency division by 2 results in a 180 degree ambiguity which is associated with the recovered carrier $c(t)$. This can be resolved by the usual process of differential encoding of the input data I_n and Q_n (see Eq. 2.2) at the expense of a slight increase in the error probability performance [38]. For any QAM system this increase will be not more than twice the error probability of that of an uncoded system or in equivalent C/N degradation not more than 0.3 dB at a symbol error probability of $P_s = 10^{-4}$ [68]. The receiver of the DPS-QAM system is shown in Fig. 2.2. The demodulated baseband I- and Q-channel signals are given by:

$$\begin{aligned} Z_I(t) &= A_p^+ A_p^- \{g_R(t) \otimes S_I(t) \cos[\gamma(t)] + g_R(t) \otimes S_Q(t) \sin[\gamma(t)] + n(t)\} \\ Z_Q(t) &= A_p^+ A_p^- \{g_R(t) \otimes S_Q(t) \cos[\gamma(t)] - g_R(t) \otimes S_I(t) \sin[\gamma(t)] + n(t)\} \\ \gamma(t) &= \phi(t) - \frac{\delta_+(t) + \delta_-(t)}{2} \end{aligned} \quad (2.5)$$

and $n(t)$ being a Gaussian noise process with power $N_0(BW)/2$, $g_R(t)$ the impulse response of the baseband receive filters and $\otimes$ denoting convolution. In deriving Eq. 2.5 we have assumed that the bandwidth of $g_R(t)$ is less than ω_p , so that the pilot signals are filtered out. As the pilots are transmitted outside of the bandwidth which is occupied by the digital signal, this is a practical assumption. Furthermore, we make the usual assumption that the cascade of $g_T(t)$ and $g_R(t)$ satisfy Nyquist's I criterion [14, Chapter 7], i.e.:

$$g_T(nT) \otimes g_R(nT) = 1 \quad \forall n. \quad (2.6)$$

Therefore, the sampled signals $Z_I(nT)$ and $Z_Q(nT)$ can be expressed as:

$$\begin{aligned} Z_I(nT) &= A_p^+ A_p^- \{I_n \cos[\gamma(nT)] + Q_n \sin[\gamma(nT)] + n(nT)\} \\ Z_Q(nT) &= A_p^+ A_p^- \{Q_n \cos[\gamma(nT)] - I_n \sin[\gamma(nT)] + n(nT)\}. \end{aligned} \quad (2.7)$$

It is clear from the above equation that a *complete* SCPJ cancellation occurs when $\gamma(nT) = 0$.

Before we proceed any further a discussion of the nature of the SCPJ, $\phi(t)$, is in order. Several models for $\phi(t)$ have been adopted in the technical literature, including Gaussian [21], Tikhonov [5] and sinusoidal [65]-[67],[78]-[80], depending on the application. In this paper we will adopt the model of sinusoidal phase jitter due to mainly two reasons.

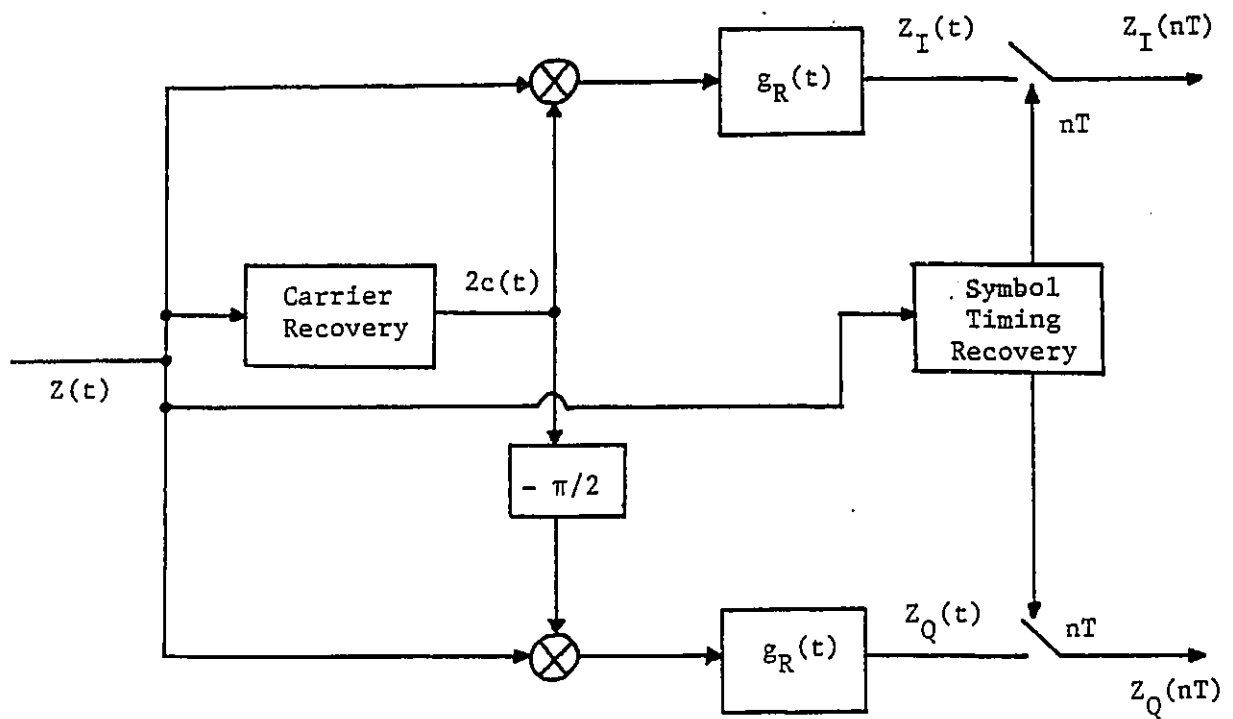


Figure 2.2: Receiver structure of a DPS-QAM system. The carrier recovery/phase cancellation (tracking) subsystem described in Fig. 2.1(b) is assumed.

1. It has been found experimentally that sinusoidal phase jitter is a good first order approximation of SCPJ, particularly phase jitter caused by power supply hum, ground loops and interference [35],[65],[66],[78]-[80].
2. It allows the subsequent mathematical analysis to be tractable and therefore in some cases closed form expressions of the system performance can be obtained.

However, it should be mentioned that the method of cancelling SCPJ applies to other SCPJ models. For example, SCPJ cancellation was successfully applied for a 1500 km non-regenerative microwave system, employing a 1.544 Mb/s 256-QAM modem [32]. In this terrestrial system the SCPJ is more complex than the assumed sinusoidal distribution.

In view of the above reasoning the SCPJ can be represented as:

$$\phi(t) = \beta \cos(\omega_m t) \quad (2.8)$$

with β its amplitude (in rads) and $\omega_m (=2\pi f_m)$ its radian frequency. Practical experience shows that $\beta \leq 0.2$ rads (11.5 degrees zero-to-peak), whereas f_m can take values up to a few hundred Hertz, depending on the application [65]-[67],[78]-[80]. In the next sections we will first derive the exact symbol error probability of a 16-QAM system in the presence of combined additive white Gaussian noise and system caused phase jitter $\gamma(t)$. Then, by using the derived expressions we will theoretically analyze and evaluate the impact of both the extraction bandwidth B' and the nonideal transfer function characteristics of the pilot extraction filters $PBPF_{\pm}$ on the 16-QAM DPS performance.

2.4 EXACT SYMBOL ERROR RATE EVALUATION OF A 16-QAM SYSTEM IN THE PRESENCE OF SCPJ AND THERMAL NOISE

Since 16-QAM has a quadrant symmetry we need to consider only the signal constellation shown in Fig. 2.3. Assuming that the transmitted symbols for

both I- and Q-channels take values from the alphabet $\{\pm 1, \pm 3\}$, then the square root of the ratio of the bit energy to the double-sided noise power spectral density is given by [11]:

$$\Gamma_b = \sqrt{\frac{39}{8N_0}}. \quad (2.9)$$

If we make the usual assumption, that the probability of error is dominated by symbol errors made between adjacent signal points, the conditioned on the SCPJ γ symbol error probability $P_{E/\gamma}$, can be written as:

$$P_{E/\gamma} = 1 - (1 - P_{EA/\gamma})(1 - P_{EB/\gamma})(1 - P_{EC/\gamma})(1 - P_{ED/\gamma}) \quad (2.10)$$

with $P_{Ei/\gamma}$ $i = \{A, B, C, D\}$ the error probabilities of the symbols A, B, C, D , conditioned on γ .

From Fig. 2.3 one can easily see that, the conditional probability of the symbol A to be in error is:

$$P_{EA/\gamma} = P_{EA1} + P_{EA2} - P_{EA1}P_{EA2} \quad (2.11)$$

with P_{EA1} and P_{EA2} given by:

$$\begin{aligned} P_{EA1} &= 0.5 \operatorname{erfc}(a_1 \Gamma_b) \\ P_{EA2} &= 0.5 \operatorname{erfc}(a_2 \Gamma_b). \end{aligned}$$

a_1, a_2 can be expressed as:

$$\begin{aligned} a_1 &= 3 (\cos \gamma - \sin \gamma) - 2 \\ a_2 &= 3 (\cos \gamma + \sin \gamma) - 2 \end{aligned}$$

and $\operatorname{erfc}(x)$ is the complementary error function which is given by:

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-t^2) dt. \quad (2.12)$$

Similarly, P_{EB} , which equals to P_{EC} because of the symmetry of the 16-QAM constellation, can be derived as:

$$P_{EB/\gamma} = 1 - (1 - P_{EB1})(1 - P_{EB2})(1 - P_{EB3}) \quad (2.13)$$

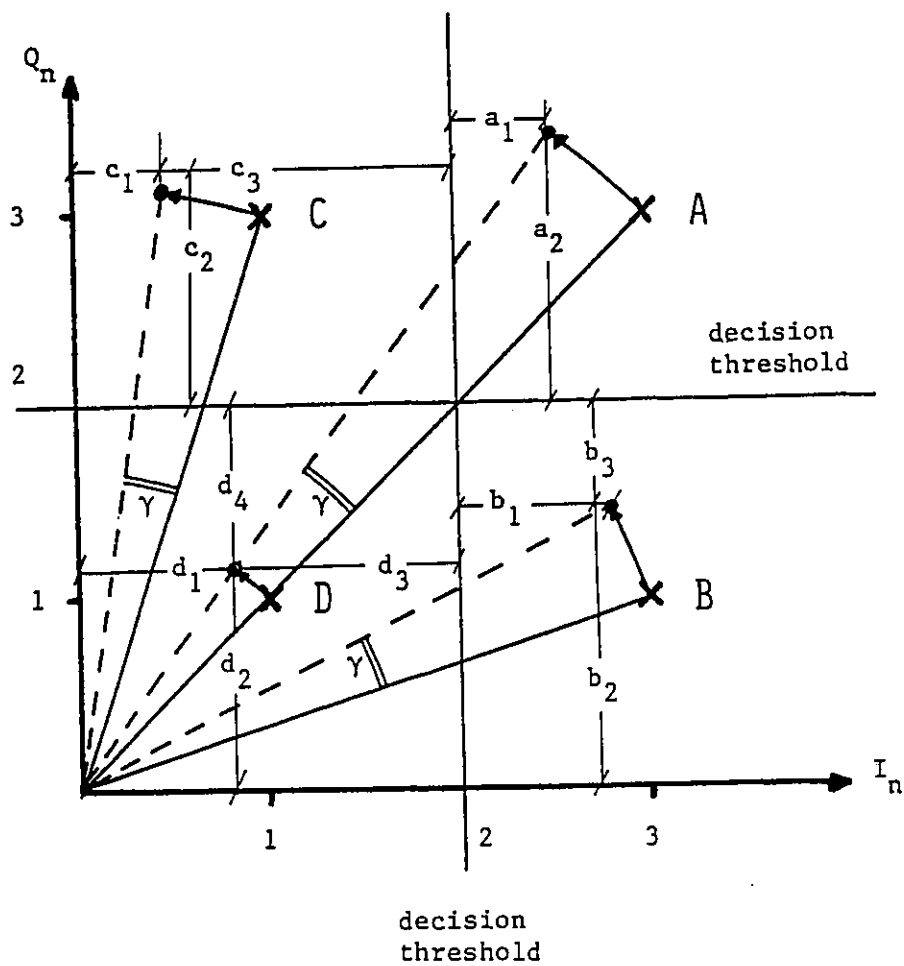


Figure 2.3: Signal constellation changes of a 16-QAM system, which is corrupted by SCPJ γ . Due to symmetry, only the points of the first quadrant are shown.

with

$$\begin{aligned} P_{EB1} &= 0.5 \operatorname{erfc}(b_1 \Gamma_b) \\ P_{EB2} &= 0.5 \operatorname{erfc}(b_2 \Gamma_b) \\ P_{EB3} &= 0.5 \operatorname{erfc}(b_3 \Gamma_b) \end{aligned}$$

and

$$\begin{aligned} b_1 &= \sqrt{10} [0.95 \cos \gamma - 0.31 \sin \gamma] - 2 \\ b_2 &= \sqrt{10} [0.31 \cos \gamma + 0.95 \sin \gamma] \\ b_3 &= 2 - b_2. \end{aligned}$$

Finally, P_{ED} is given by:

$$P_{ED/\gamma} = 1 - (1 - P_{ED1})(1 - P_{ED2})(1 - P_{ED3})(1 - P_{ED4}) \quad (2.14)$$

with

$$\begin{aligned} P_{ED1} &= 0.5 \operatorname{erfc}(d_1 \Gamma_b) \\ P_{ED2} &= 0.5 \operatorname{erfc}(d_2 \Gamma_b) \\ P_{ED3} &= 0.5 \operatorname{erfc}(d_3 \Gamma_b) \\ P_{ED4} &= 0.5 \operatorname{erfc}(d_4 \Gamma_b) \end{aligned}$$

and

$$\begin{aligned} d_1 &= \cos \gamma - \sin \gamma \\ d_2 &= \cos \gamma + \sin \gamma \\ d_3 &= 2 - d_1 \\ d_4 &= 2 - d_2. \end{aligned}$$

By averaging $P_{E/\gamma}$ over all possible values of γ the exact symbol error, P_E , can be evaluated. We have obtained results for the performance of the 16-QAM system operated in the presence of sinusoidal SCPJ and additive white Gaussian noise. These results are summarized in Fig. 2.4 and are presented for different values of β . From this figure it is clear that, for $\beta > 0.2$ rads the performance degradation of the 16-QAM system is unacceptably high for almost any practical application. However, as we

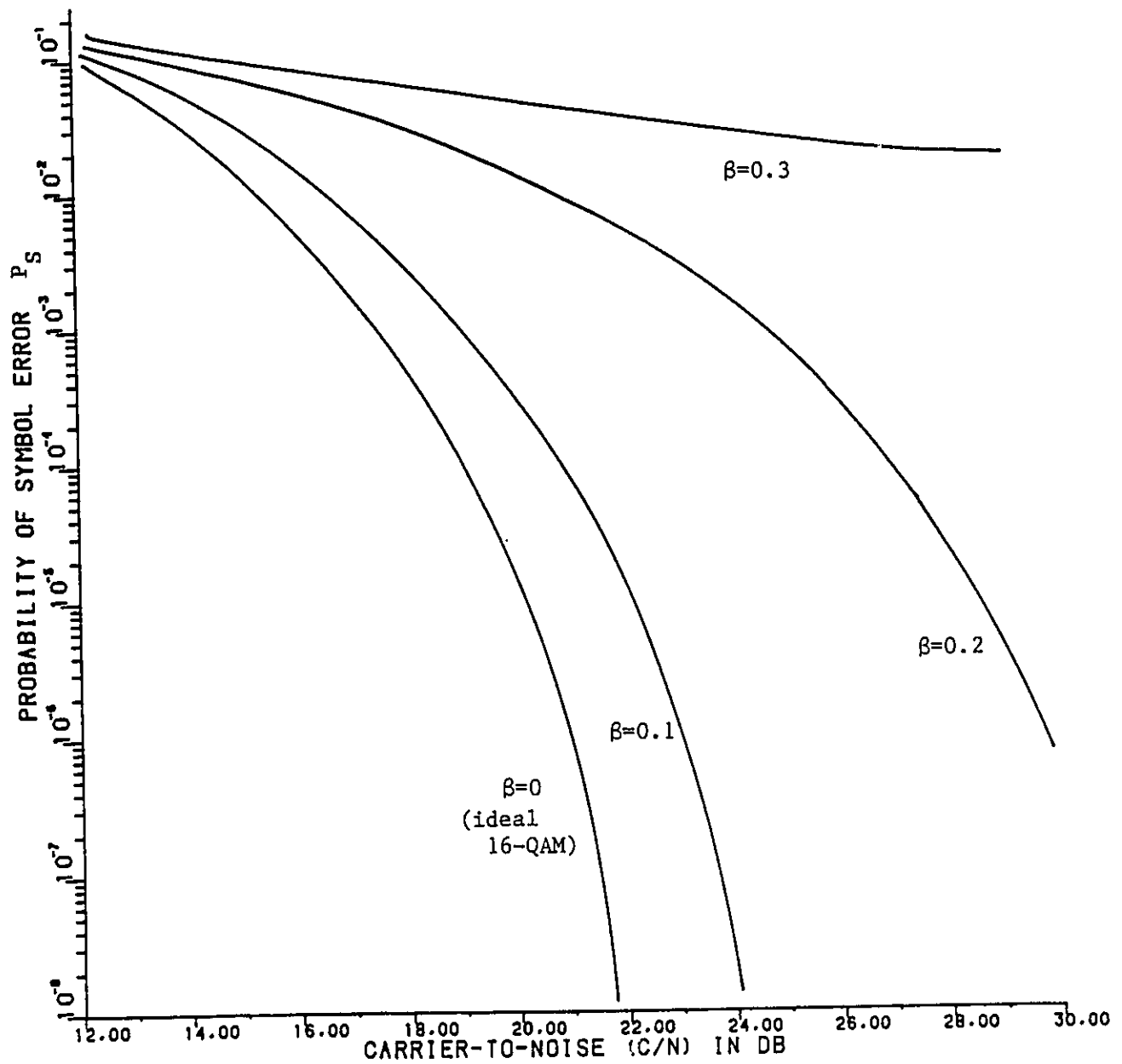


Figure 2.4: Impact of the amplitude β (in rads) of the sinusoidal SCPJ on the $P_S = f(C/N)$ performance of the 16-QAM system. The results are independent of the frequency of the sinusoidal interferer. For $\beta = 0$, the performance is the same as that of an ideal 16-QAM system, i.e. without any SCPJ.

will see later on, by using SCPJ cancellation (tracking) techniques, the performance of the 16-QAM systems in the presence of SCPJ could be considerably improved. One other interesting observation is that the obtained results are only a function of β and do not depend on the frequency f_m of the SCPJ. The reason for this is the previously described averaging of the individual symbol error probability of all 16 states over *one full cycle* of the sinusoidal SCPJ. We verified this result by evaluating the symbol error rate performance of the 16-QAM system for a number of different frequencies f_m .

2.5 IMPACT OF THE IMPERFECT EXTRACTION FILTERS ON THE DPS 16-QAM SYSTEM PERFORMANCE

2.5.1 Wideband Systems

For these systems the extraction filters bandwidth B' is wide enough so that all the unwanted SCPJ is extracted together with the pilot tones. In the case of sinusoidal SCPJ the bandwidth which it occupies is theoretically infinite. Thus, this wideband system is unrealistic in the sense that, since it requires an infinite bandwidth to extract all the unwanted SCPJ, the amount of thermal noise which gets also extracted will become prohibitively high. However, it is instructive to analyze this case and compare its results with that of the narrowband system, so that conclusions can be reached about the optimal value of B' . Furthermore, it should be mentioned that practical experience has shown that the extracted thermal noise does not cause any significant performance degradation as compared to the degradation caused by SCPJ [52]-[58].

With the above in mind, for a wideband system employing *ideal but non-identical* extraction filter characteristics as shown in Fig. 2.5, the distorted phases of the extracted pilots can be written as:

$$\delta_{\pm}(t) = \beta \cos[\omega_m(t - \tau_{\pm})] \quad (2.15)$$

with τ_+ and τ_- being the delay of $PBPF_+$ and $PBPF_-$ respectively. Moreover, it can be easily seen that for the DPS-QAM system the requirement for total SCPJ cancellation in this case is $\tau_+ = \tau_- = 0$ or in general $(\tau_{\pm})\omega_m = n(2\pi)$ with $n = 0, 1, 2, \dots$. Thus, although the extraction filter characteristics are ideal, i.e. linear phase and constant amplitude as functions of frequency, the DPS-QAM system will suffer degradation if the above mentioned conditions are not met. Clearly, in this case $\gamma(t)$ is given by:

$$\gamma(t) = \beta \{ \cos \omega_m t - \cos[\omega_m(\frac{\tau_+ - \tau_-}{2})] \cos[\omega_m(t - \frac{\tau_+ + \tau_-}{2})] \}. \quad (2.16)$$

A geometrical interpretation of Eq. 2.16 in terms of vectors is illustrated in Fig. 2.6, from which it can be easily seen that:

$$|\gamma(t)|_{\tau_+ = \tau_-} > |\gamma(t)|_{\tau_+ \neq \tau_-} \quad (2.17)$$

with $|\gamma(t)|$ representing the magnitude of $\gamma(t)$. Clearly the scenario $\tau_+ = \tau_- = \tau$ can be considered as a worst case performance for a given value of τ . In this case, Eq. 2.16 can be rewritten as:

$$\begin{aligned} \gamma(t)|_{\tau_+ = \tau_- = \tau} &= \beta \{ \cos \omega_m t - \cos[\omega_m(t - \tau)] \} \\ &= 2\beta \sin(\omega_m \frac{\tau}{2}) \sin[\omega_m(t - \frac{\tau}{2})]. \end{aligned}$$

In order for a system which employs the SCPJ cancellation to perform better than a 'conventional' (i.e. without any SCPJ cancellation) the following relation should be valid:

$$2\beta \sin(\omega_m \frac{\tau}{2}) \sin[\omega_m(t - \frac{\tau}{2})] < \beta.$$

It is easy to show, that the above condition can equivalently be rewritten as:

$$\tau < \frac{1}{6f_m}.$$

We have evaluated this wideband system by using the analysis which was described in Sections 2.3 and 2.4. The results obtained for the worst case $\tau_+ = \tau_-$ are presented in Fig. 2.7, in terms of Carrier-to-Noise Ratio

Phase responses
of $PBPF_{\pm}$

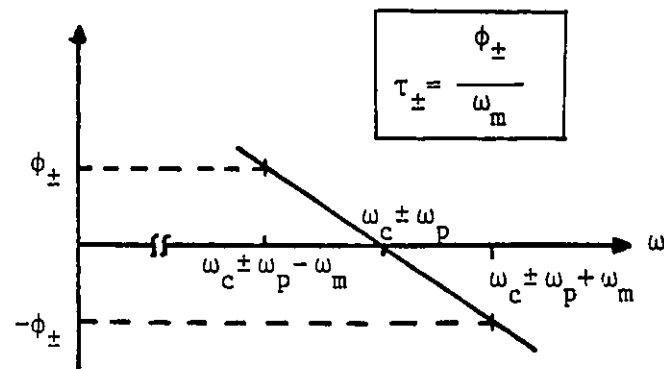


Figure 2.5: Linear phase responses of the ideal $PBPF_{\pm}$. Their amplitude responses are assumed to be 1 and constant over the bandwidth of interest.

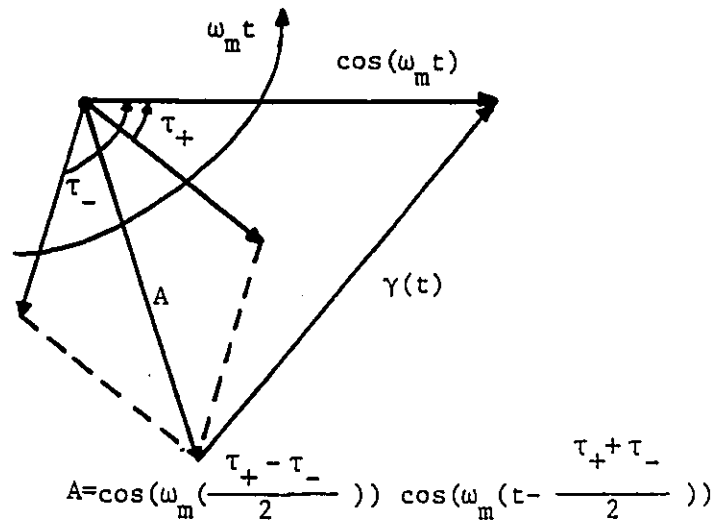


Figure 2.6: Rotating phasor representation of the two terms of Eq. 2.16. Both terms (i.e. A and $\cos\omega_m t$) are assumed to rotate with a radian frequency of ω_m .

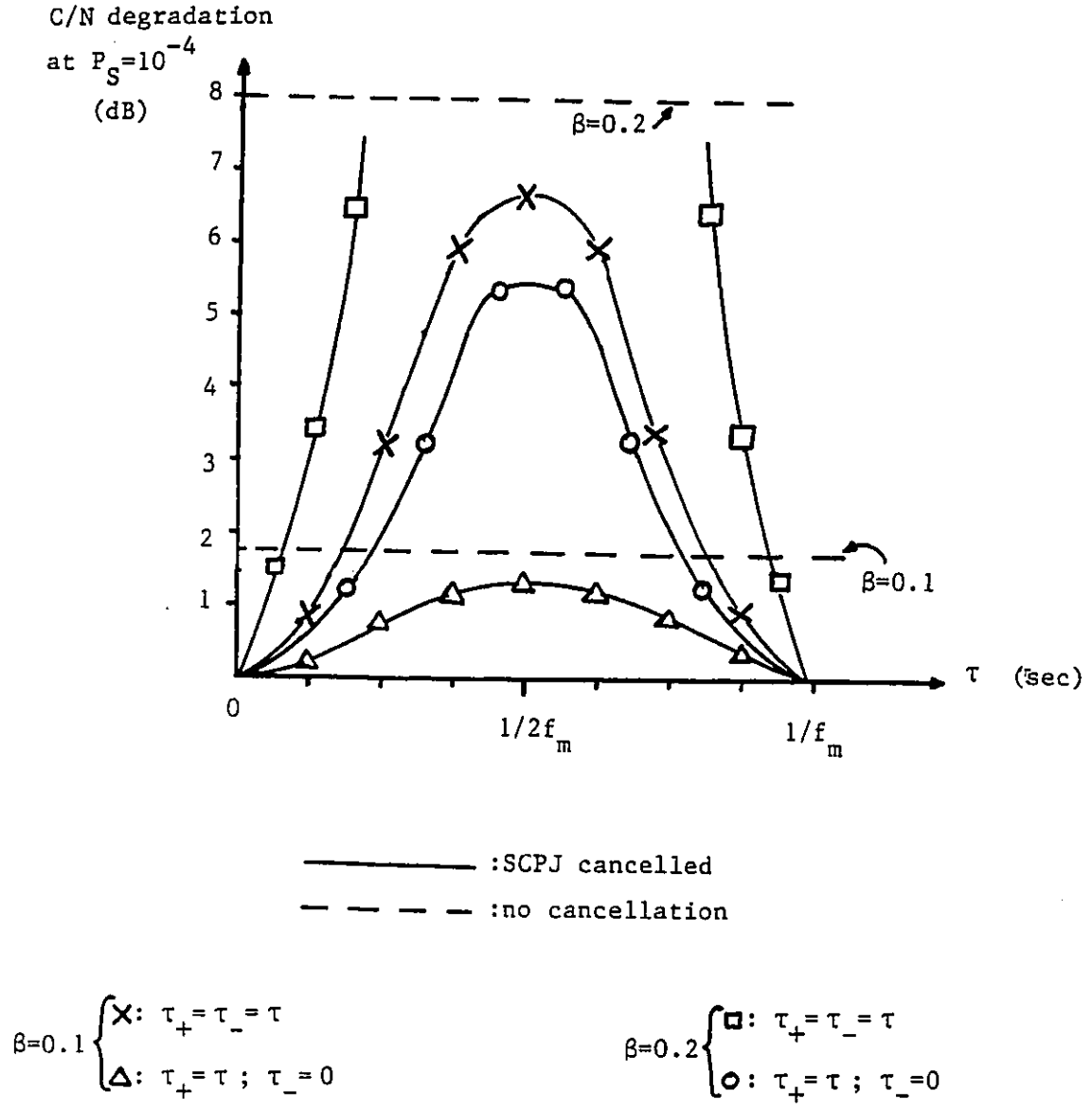


Figure 2.7: C/N degradation of the DPS 16-QAM system for wideband extraction filters and as a function of the constant group delay τ .

(C/N) degradation (at $P_S = 10^{-4}$) as compared to an ideal (i.e. without any SCPJ) 16-QAM system. Additionally, we have also included the previously (see Section 2.4) obtained performance degradation results of a 16-QAM operated in the presence of SCPJ. From this figure we can observe firstly the periodic nature of the system degradation. This suggests that by introducing the appropriate delay, i.e. $\tau_+ \doteq \tau_- \doteq 2n\pi/\omega_m$ with $n = 0, 1, 2, \dots$, the system degradation can be reduced considerably as compared to the narrowband system. Secondly it is clear that the requirement of having the appropriate delays becomes more severe as β increases. For comparison purposes we have included in the same figure similar results for the case where $\tau_- = 0$. It is interesting to observe that the system degradation is much less for the case where $\tau_+ = 2K, \tau_- = 0$ as compared to an equivalent case of $\tau_+ = \tau_- = K$. This suggests that it is more appropriate to ensure that one of the delays is minimized, rather than having both delays small but nonminimized, especially for applications with relatively high β . Finally, it should be mentioned that for all cases the maximum system degradation occurs for $\tau_+ = \tau_- = \pi/\omega_m$.

2.5.2 Narrowband Systems

For these systems, the extraction filters bandwidth B' is not wide enough, so that only part of the SCPJ is extracted. As a result we should expect some additional degradation, which should depend on the extraction bandwidth B' . To demonstrate this point, let us assume for convenience that $\tau_+ = \tau_- = 0$, and consider the extreme case where $B' < \omega_m/\pi$ in which only the frequencies $\omega_c \pm \omega_p$ are extracted. Thus, from Eqs. 2.3, 2.5 it follows that:

$$\gamma(t) = \beta \cos(\omega_m t). \quad (2.18)$$

Evidently, in this case we did not achieve any SCPJ cancellation. As mentioned before, these narrowband systems are more realistic and practical than their wideband counterparts, so therefore we will somewhat generalize our methodology to include both amplitude and group delay distortions for the extraction filters $PBPF_{\pm}$.

As shown in Fig. 2.8 these filters will extract frequency components of the sinusoidal SCPJ which are located at $\omega_0 \pm \omega_m$, where:

$$\omega_0 = \omega_c \pm \omega_p. \quad (2.19)$$

Then the phase jitter to be extracted, $\gamma_{tbe}(t)$, is given by:

$$\gamma_{tbe}(t) = J_0(\beta) \cos(\omega_0 t) - J_1(\beta) \{ \sin[(\omega_0 + \omega_m)t] + \sin[(\omega_0 - \omega_m)t] \} \quad (2.20)$$

From the above equation and by representing the transfer function of the extraction filters $PBPF_{\pm}$ as in Fig. 2.8(b), the extracted pilot signal $c_+(t)$ can be written as:

$$c_+(t) = J_0(\beta) \cos(\omega_1 t) - J_1(\beta) \{ E_1 \sin[(\omega_1 + \omega_m)t - D_{1+}] + E_2 \sin[(\omega_1 - \omega_m)t + D_{2+}] \} \quad (2.21)$$

with D_{1+} and D_{2+} given by:

$$\begin{aligned} D_{1+} &= \omega_m \tau_+ + \Delta\phi_+ \\ D_{2+} &= \omega_m \tau_+ + \Delta\delta_+. \end{aligned}$$

Also, without any loss of generality, we can assume that $E_2 = 1$ and $E_1 = 1 + e_+$, so that Eq. 2.21 can be rewritten as:

$$\begin{aligned} c_+(t) &= J_0(\beta) \cos(\omega_1 t) - J_1(\beta) \{ e_+ \sin[(\omega_1 + \omega_m)t - D_{2+}] \\ &\quad + 2 \left[\cos(\omega_m t - \frac{D_{1+} + D_{2+}}{2}) \sin(\omega_m t + \frac{D_{1+} - D_{2+}}{2}) \right] \}. \end{aligned} \quad (2.22)$$

Expanding the trigonometric terms of the above equation the following expression for $c_+(t)$ can be found:

$$c_+(t) = G_+ \cos(\omega_1 t) + H_+ \sin(\omega_1 t) \quad (2.23)$$

where G_+ and H_+ are given by:

$$\begin{aligned} G_+ &= J_0(\beta) + J_1(\beta) [e_+ \sin(\omega_m t - D_{2+}) + 2 \cos(\omega_m t - \frac{D_{1+} + D_{2+}}{2}) \sin(\frac{D_{1+} - D_{2+}}{2})] \\ H_+ &= J_1(\beta) [e_+ \sin(\omega_m t - D_{2+}) + 2 \cos(\omega_m t - \frac{D_{1+} + D_{2+}}{2}) \cos(\frac{D_{1+} - D_{2+}}{2})]. \end{aligned}$$

In a similar manner, the following expression for the other extracted pilot $c_-(t)$ can be obtained:

$$c_-(t) = G_- \cos(\omega_2 t) + H_- \sin(\omega_2 t) \quad (2.24)$$

where

$$\begin{aligned} G_- &= J_0(\beta) + J_1(\beta) [e_- \sin(\omega_m t - D_{2-}) + 2 \cos(\omega_m t - \frac{D_{1-} + D_{2-}}{2}) \sin(\frac{D_{1-} - D_{2-}}{2})] \\ H_- &= J_1(\beta) [e_- \sin(\omega_m t - D_{2-}) + 2 \cos(\omega_m t - \frac{D_{1-} + D_{2-}}{2}) \cos(\frac{D_{1-} - D_{2-}}{2})]. \end{aligned}$$

D_{1-} and D_{2-} of the previous equation are given by:

$$\begin{aligned} D_{1-} &= \omega_m \tau_- + \Delta\phi_- \\ D_{2-} &= \omega_m \tau_- + \Delta\delta_- \end{aligned}$$

In Eqs. 2.20 - 2.25 $J_0(\beta)$, $J_1(\beta)$ are Bessel functions coefficient of the first kind. $e_{\pm}$ denotes the amount of fluctuation of the transfer function of the extraction filters whereas $\Delta\phi_{\pm}$ and $\Delta\delta_{\pm}$ their phase distortions, all at $\omega_0 \pm \omega_m$ ($\omega_0 = \omega_c \pm \omega_p$), as shown in Fig. 2.8(b). By multiplying $c_+(t)$ with $c_-(t)$, extracting the frequency components at $2\omega_c$, frequency divide and amplifying the resulting signal by 2, the recovered carrier can be written as:

$$c(t) = A \cos[\omega_c t + \lambda(t)] \quad (2.25)$$

with

$$\begin{aligned} A^2 &= (G_+ G_- - H_+ H_-)^2 + (G_+ H_- + H_+ G_-)^2 \\ \lambda(t) &= -0.5 \tan^{-1} \left(\frac{G_+ H_- + H_+ G_-}{G_+ G_- - H_+ H_-} \right). \end{aligned}$$

Finally, by using the above expression for $\lambda(t)$ and Eq. 2.5 the following equation follows:

$$\gamma(t) = \beta \cos(\omega_m t) - 0.5 \tan^{-1} \left(\frac{G_+ H_- + G_- H_+}{G_+ G_- - H_+ H_-} \right). \quad (2.26)$$

Eqs. 2.10 and 2.26 are used to evaluate the C/N performance degradation as a function of the different filter imperfections, similarly to the wideband system case. The obtained results are presented in Figs. 2.9 and 2.10 having the amplitude, $e_{\pm}$, and phase distortions, $\Delta\phi_{\pm}$, $\Delta\delta_{\pm}$, respectively, as parameters. Fig. 2.9 presents the C/N degradation at $P_S = 10^{-4}$ as a function of the amplitude variation of the extraction filters, for $|e_+| = |e_-|$ and $e_- = 0$. From this figure we note that for higher β the system is more sensitive to amplitude fluctuations. However, since B' is relatively small, these fluctuations should be typically less than 0.12 (0.5 dB); therefore amplitude distortion should cause small C/N degradation, even for $\beta = 0.2$. Fig. 2.10 presents the system degradation as a function of the phase distortions $\Delta\phi_{\pm}$ and $\Delta\delta_{\pm}$. We considered the worst case in which all distortions are the same, i.e. $\Delta\phi_+ = \Delta\phi_- = \Delta\delta_+ = \Delta\delta_-$, and also the case where one

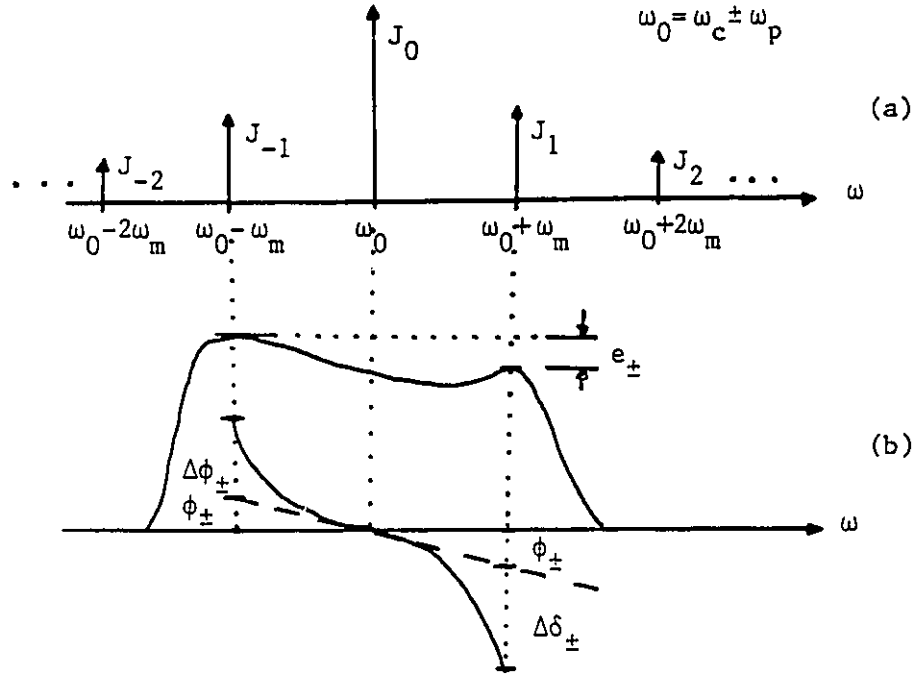
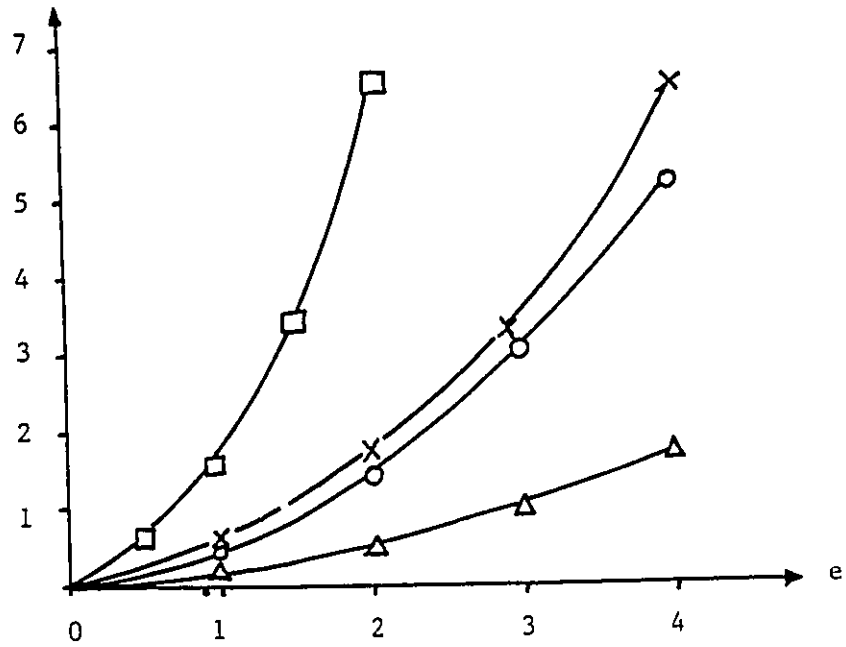


Figure 2.8: (a) Spectrum of the sinusoidal SCPJ with $\phi(t) = \beta \cos(\omega_m t)$.
(b) Nonideal transfer function of the extraction filters $P B P F_{\pm}$, having an extraction bandwidth of B' with $2f_m < B' < 4f_m$.

C/N Degradation
at $P_S = 10^{-4}$ (dB)

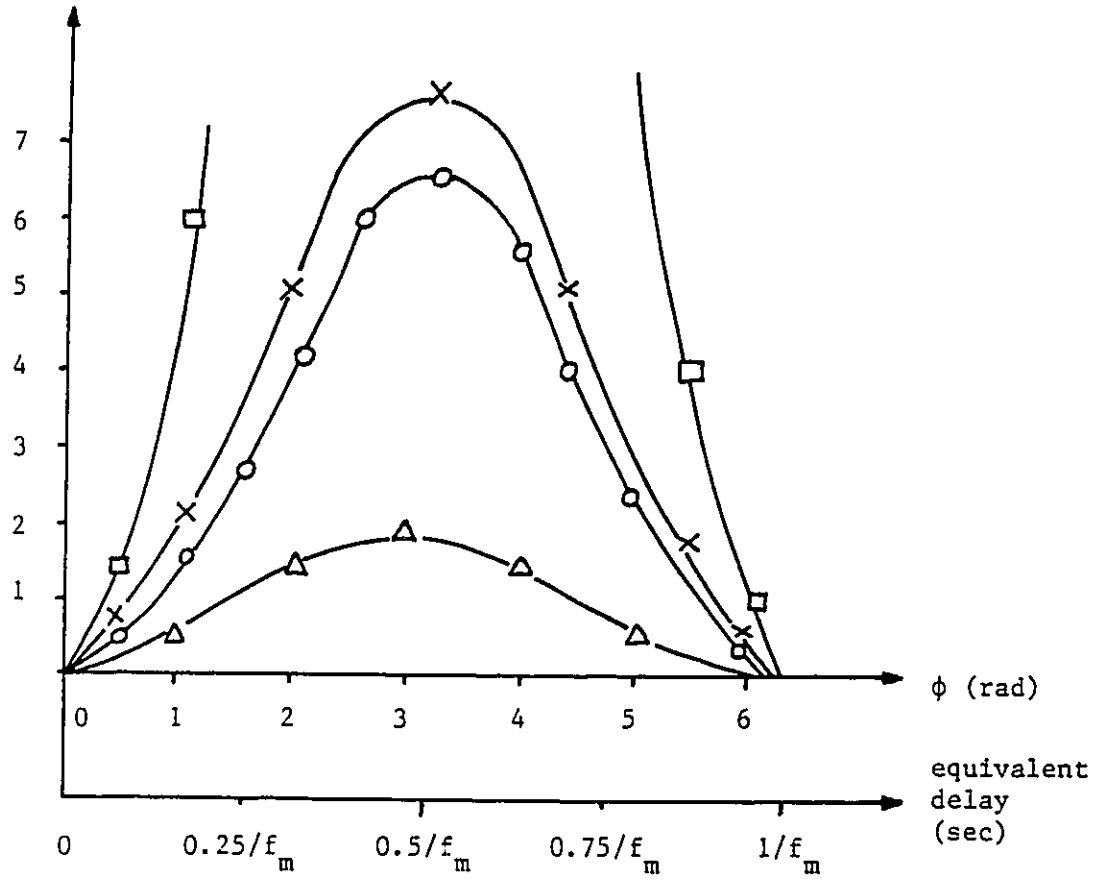


$$\beta=0.1 \begin{cases} \times : |e_+| = |e_-| = e \\ \Delta : |e_+| = e ; e_- = 0 \end{cases}$$

$$\beta=0.2 \begin{cases} \square : |e_+| = |e_-| = e \\ \circ : |e_+| = e ; e_- = 0 \end{cases}$$

Figure 2.9: C/N degradation of a 16-QAM DPS with narrowband pilot extraction filters as a function of their amplitude variation e .

C/N degradation
at $P_S = 10^{-4}$ (dB)



$$\beta=0.1 \begin{cases} \times : \Delta\phi_{\pm} = \Delta\delta_{\pm} = \phi \\ \Delta : \Delta\phi_{+} = \Delta\phi_{-} = \phi ; \Delta\delta_{\pm} = 0 \end{cases}$$

$$\beta=0.2 \begin{cases} \square : \Delta\phi_{\pm} = \Delta\delta_{\pm} = \phi \\ \circ : \Delta\phi_{+} = \Delta\phi_{-} = \phi ; \Delta\delta_{\pm} = 0 \end{cases}$$

Figure 2.10: C/N degradation of a DPS 16-QAM system with narrowband pilot extraction filters as a function of their phase distortion ϕ .

of the filter distortions is equalized, i.e. $\Delta\delta_+ = \Delta\delta_- = 0$. The trend of the results is similar to that of the linear phase case (Fig. 2.7) although the degradations are generally higher.

For all the results which were presented in this section we did not include the degradation which results from the additional power that is required to transmit the pilot tones. Towards this end, let us assume that a suppressed carrier QAM¹ system requires C power to operate at a certain symbol error rate P_S . If the inserted pilot tones require power of mC to be transmitted, then the total required power C' for the dual pilot QAM system to operate at the same symbol error rate P_S is given by:

$$C' = C + mC. \quad (2.27)$$

In other words, the additional power allocated to the pilots will cause a C/N degradation, D_p , of:

$$D_p \text{ (dB)} = 10 \log_{10}(1 + m). \quad (2.28)$$

Table 2.1 presents the values of D_p for different values of m . It is clear from this table that m has to be small, so that the degradation of pilot based QAM system is not significant. It is felt, for all practical purposes, that $m_{max} = -10$ dB (or 0.1) should be the maximum level of the pilots, so that D_p is not more than 0.4 dB. Thus, by including the additional degradation of 0.3 dB which results due to the differential encoding of the input data (see Section 2.3) the total degradation D_{DPS} which is present in our DPS-QAM system is 0.7 dB. Since this D_{DPS} is inherent to any transmission scheme which employs the DPS and further to concentrate our attention to the degradation resulting from previously described system imperfections, we did not include it in our analytical performance evaluation results.

Before we proceed to a comparison of the analytically derived results with a number of experimental results obtained by measurements, we will first describe in the next section the design and implementation aspects of a prototype 16-QAM DPS.

¹The discussion which follows applies also to any other suppressed carrier digital modulation system.

m	m (dB)	D_p (dB)
0.001	-30	0.0043
0.01	-20	0.043
0.025	-16	0.11
0.1	-10	0.41
0.25	-6	0.97
0.5	-3	1.76
1.0	0	3.0

Table 2.1: Equivalent C/N degradation D_p due to the power which is required to transmit the pilot tones. m is defined as the ratio of the power of the pilot tones divided by the power required to transmit the digitally modulated signal.

2.6 DESIGN AND IMPLEMENTATION CONSIDERATIONS OF AN EXPERIMENTAL 16-QAM DPS

In order to verify the validity of our theoretical results and gain a further physical insight of the previously obtained analytical results, we designed and implemented a prototype 16-QAM DPS.

At this point an important note is in order. It turned out that the successful realization of this experimental 16-QAM DPS was a challenging, rather difficult and generally speaking time consuming task. We estimated that we have spend the equivalent of 7-8 months of full time efforts in designing, implementing and of course *making the modem work*. However, since we consider that experimental research is of equal importance as theoretical and computer simulated work, it is felt that undertaking this project was more than a worthwhile task.

The block diagram of the 16-QAM DPS transmitter and receiver are shown in Figs. 2.11(a) and 2.11(b), respectively. The main purpose of this section is to briefly describe the design and implementation philosophy/methodology of the individual blocks of Fig. 2.11. Detailed circuit diagrams of this experimental 16-QAM DPS and a brief description of the

signals involved are given in Appendix A.

Let us first start with the transmit, $g_T(t)$, and receive, $g_R(t)$, filters since their frequency characteristics will determine the system's bit rate. Both of them are square root α ($\sqrt{\alpha}$) raised-cosine filters having an excess bandwidth of 20% and a Nyquist frequency of $f_N = 100$ kHz.² In the transmit filter an $(x)/\sin(x)$ aperture equalizer is included, so that the overall filtering strategy theoretically satisfies Nyquist's I criterion. These filters were designed and build by Karkar Electronics, Inc. (a unit of Signal General). As well known, the Nyquist rate of 100 kHz corresponds to a symbol rate of $f_S = 200$ kBaud/s which in turn implies that for a 16-QAM modem a bit rate $f_b = 800$ kb/s should be used.

In the transmitter, this 800 kb/s rate binary signal is commuted by means of a Serial-to-Parallel Converter (SPC) into two independent binary bit streams (I- and Q-channel), each having a rate of 400 kb/s. Both of these binary (2-level) bit streams are converted into 4-level digital signals, which have a symbol rate of 200 kBaud/s. In order to ensure that an error in an adjacent level of the received signal results in only one bit error the output of the 4-level converter is Gray coded. The implementation of the SCP as well as the 2-to-4 level converters was realized by employing standard TTL and CMOS devices, such as 74LS04, 74LS74, 74LS393, CD4098, CD4013 and others. Furthermore, two Digital-to-Analog Converters (DAC08CD) as well as a number of operational amplifiers (TL084 and LM318) and two line drivers (LH0002). To achieve a symmetric system configuration, which resulted in similarly behaving digital signals, it was decided to build the baseband part of both I- and Q-channels on one card (CARD #1; see also Fig. A.3).

For the same reason and furthermore to minimize the mutual interference (due to mainly ground loops) between the baseband and RF signals, the modulator for both I- and Q-channels was breadboarded on a separate card (CARD #2; see also Fig. A.4). The pilot tones are added by using a LM318 operational amplifier with the 4-level filtered baseband PAM signal of the I-channel just before the upconverting mixer. Both level (A_p) and frequency (f_p) of the inserted pilot signals can easily be varied. However,

²Their measured power transfer function indicated that they exhibit a 55 dB attenuation at 120 kHz.

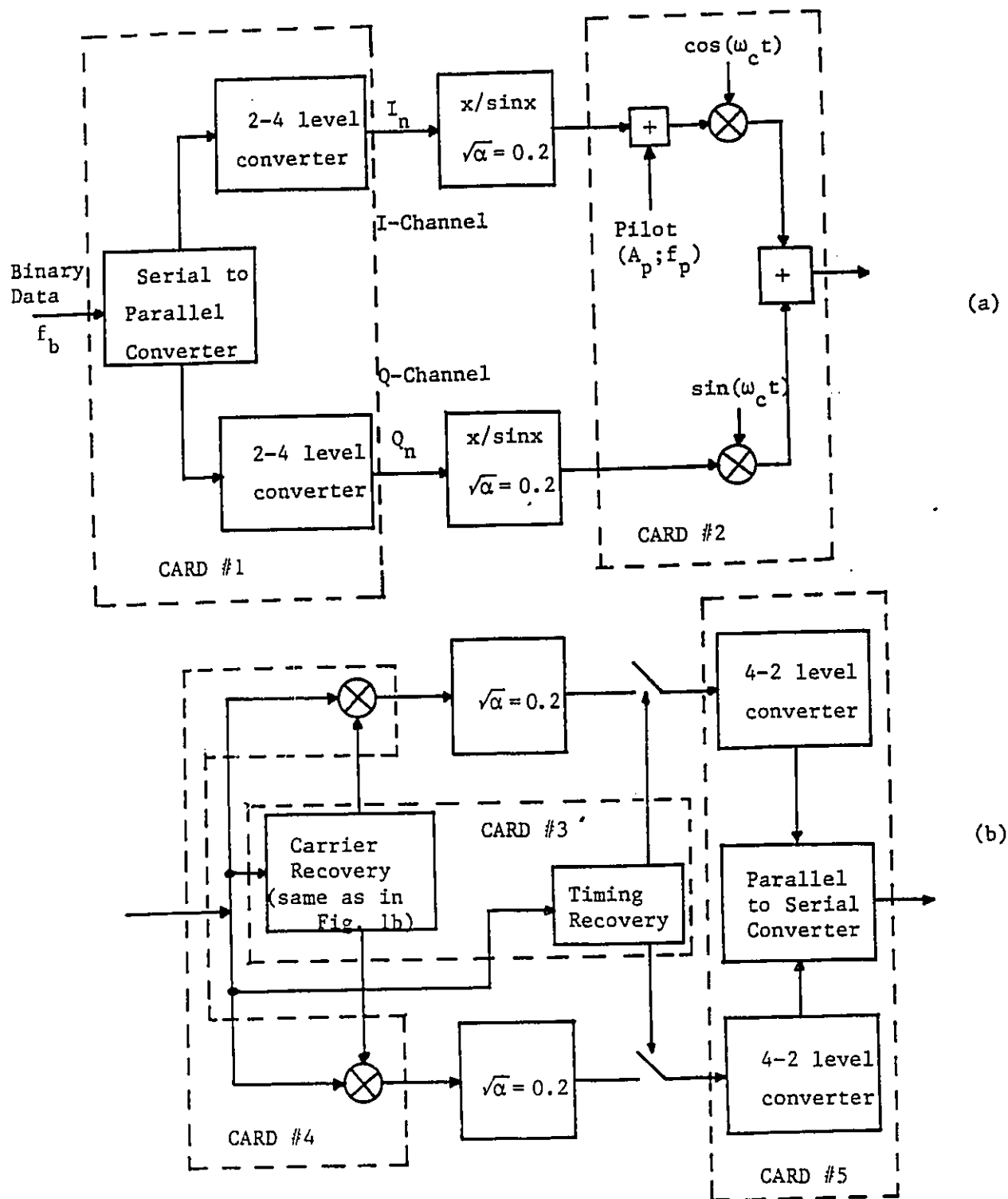


Figure 2.11: Block diagram of the used experimental 16-QAM DPS. The dashed lines indicate the plugboards (CARDS #1 - #5) on which the prototype hardware was build. Detailed circuit diagram are given in Appendix A.

- (a) Transmitter structure.
- (b) Receiver structure.

the experimental results which we will report later on are for $f_p = 200$ kHz and with each pilot having power 13 dB below the 16-QAM signal power, so that this additional energy allocation to the pilots does not degrade the overall system performance more than 0.4 dB (see Table 2.1). Furthermore, this particular pilot frequency was chosen since it is the same as the symbol rate, so that the timing signal can easily be obtained from the pilot signals. In the same card the two modulated signals are added and then transmitted to the channel. In designing the modulator we employed standard Mini-Circuit devices such as phase shifters (PCSQ-2-1.5), power dividers/combiners (MSC2-1) and double balanced mixers (SRA-1). Due to the availability of these components and also to ease some of the implementation requirements the carrier frequency was selected to be $f_c = 1.5$ MHz.

At the receiver, first the carrier and the symbol timing signal are recovered by using the configuration shown in Fig. 2.1(b) (CARD #3; see also Fig. A.5). Afterwards both signals are fed into the quadrature demodulator card (CARD #4; see also Fig. A.6). By employing similar to the modulator Mini-Circuit devices the RF signal is demodulated in quadrature and the two outputs are fed into the receive $\sqrt{\alpha}=0.2$ raised-cosine filters. The filtered outputs are inputs to the 16-QAM decoder card (CARD #5; see also Fig. A.7). This card, as shown in Fig. 2.11(b) employs 3 slicers (threshold detectors) for each of the I- and Q-channel signals. The employed slicers are realized by 6 operational amplifiers (LM 311) and their purpose is to quantize each of the received signals into one of the 4 transmitted levels $\{\pm 1, \pm 3\}$. In the absence of noise the quantized signals I_n^q and Q_n^q are identical to the transmitted ones I_n and Q_n . The threshold levels, V_{th1} , V_{th2} and V_{th3} are set properly in the horizontal middle (i.e. level) of each eye of the eye-diagrams of both I- and Q-channels. Furthermore, the sampling of each received symbol is adjusted so that it takes place in the vertical middle (i.e. time) of the eye-diagrams. Then, through a straight-forward logic two 4-to-2 level conversions take place and by means of a Parallel-to-Serial Converter (PSC) both I- and Q-channels are combined to generate a replica of the transmitted binary data. For the implementation of the 4-to-2 level converters as well as the PSC standard ICs have been employed (e.g. 74LS02, 74LS04, 74LS08, 74LS95, 74LS175, CD4013). Photographs of the measured spectrum of the transmitted 16-QAM DPS

as well as the demodulated signal constellation appear in Figs. 2.12 and 2.13, respectively.

To conclude this section, we would like to briefly point out the following three design suggestions, which we have discovered during the implementation and testing of the described 16-QAM DPS.

1. It is very important to ensure that the operating conditions and in particular the input signal levels of the employed RF components suggested by the manufacturer are met.³
2. It turned out that improper matching between the employed raised-cosine filters and the different modules had a major impact on the 16-QAM system performance. Thus proper care has to be taken in order to ensure that all input and output impedances of the different units are matched.
3. In order to ensure a time stable system operation we were forced to employ a number of voltage regulators such as 7915, 7815, 7905.

2.7 PERFORMANCE EVALUATION RESULTS OF THE EXPERIMENTAL 16-QAM DPS

2.7.1 Experimental Set-up Description

A block diagram of the experimental set-up used for testing the 16-QAM DPS is illustrated in Fig. 2.14. Further, a photograph of the complete experimental set-up in the Digital Communication Laboratory of the Department of Electrical Engineering at the University of Ottawa is shown in Fig. 2.15. During the course of the experiments

³Although this might be obvious to any experienced design engineer, we discovered by designing and testing this modem that input signal power levels were often forgotten to be adjusted to their proper value. This resulted in nonlinear operation of the used RF devices, which in turn increased the overall system degradation.

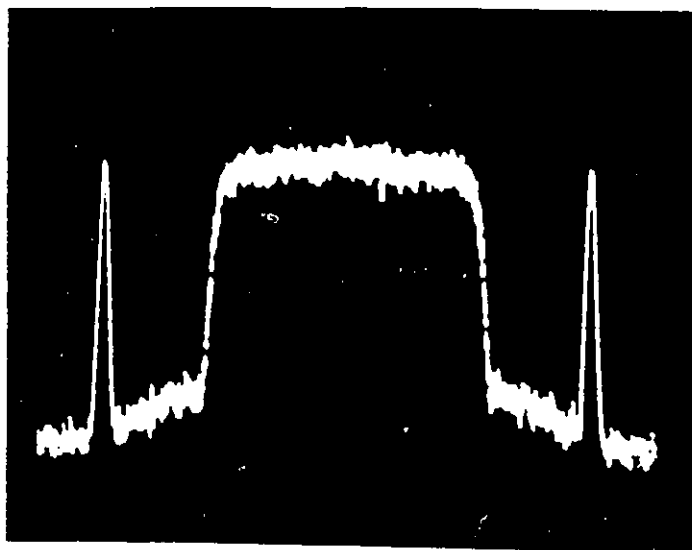


Figure 2.12: Measured spectrum of the transmitted 16-QAM DPS without any Gaussian noise or SCPJ. The power, which is allocated to each of the pilot tones is 17 dB below the data signal power.

Horizontal Axis: 50 kHz/div.

Vertical Axis: 10 dB/div.

Center Frequency: 1.5 MHz.

Resolution Bandwidth: 5 kHz.

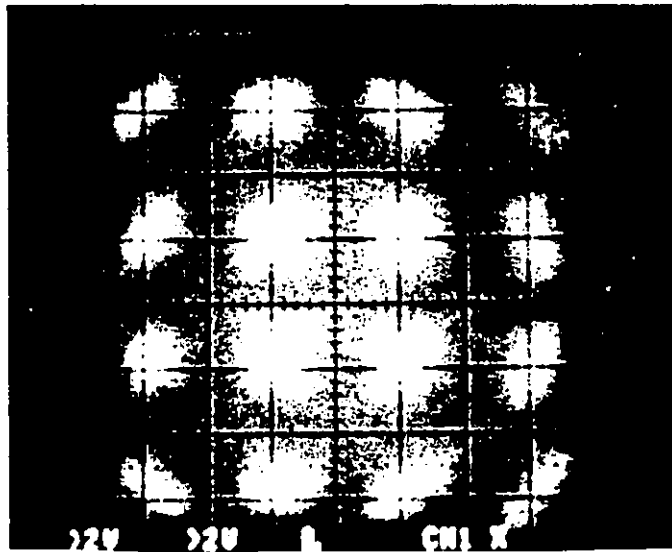


Figure 2.13: Measured signal space diagram of the demodulated 16-QAM DPS in the presence of sinusoidal SCPJ ($\beta = 0.1$ rads and $f_m = 9$ kHz) and additive white Gaussian noise ($C/N = 22$ dB).

it became apparent that the key element in the system performance are the characteristics of the pilot extraction filters $PBPF_{\pm}$. To simplify the experimental prototype design, especially the requirement for a relatively small extraction bandwidth B' for these filters, we selected the frequency of the sinusoidal interference to be $f_m = 9$ kHz. Although the chosen frequency is fairly high for a practical application, our scope here is to verify the previously derived analytical results. Furthermore, since the extraction bandwidth B' is directly related to f_m , our results are valid assuming the values of both of these parameters are linearly reduced. However, the drawback of this approach is that now the extracted thermal noise is not negligible any more. To bypass this problem in the experiments the pilots are extracted and further processed *before* the addition of the white Gaussian noise, as shown in Fig. 2.14.

We have used 2 sets of extraction filters, S_1 and S_2 . Each set consists of two bandpass filters with their center frequencies located at 1.3 and 1.7 MHz. Their measured transfer characteristics are summarized in Table 2.2. The bandwidth B' of S_1 will allow extraction of the frequency components at f_0 and $f_0 \pm f_m$, whereas S_2 will extract f_0 , $f_0 \pm f_m$, $f_0 \pm 2f_m$ and $f_0 \pm 3f_m$. It can be seen from Table 2.2 that the group delay variations of both filters of S_2 over the extraction bandwidth are much higher than that of S_1 .

	CENTER FREQUENCY f_0 (MHz)	EXTRACTION BANDWIDTH B' (kHz)	AMPLITUDE VARIATION ϵ	GROUP DELAY at f_0 τ' (μ sec)	GROUP DELAY VARIATION $\Delta\tau'$ (μ sec)
Filter Set 1 (S_1)	1.3	21	1.74	35	2
	1.7	22	1.78	35	1
Filter Set 2 (S_2)	1.3	61	1.95	30	20
	1.7	60	2.19	28	12

Table 2.2: Measured characteristics of the employed extraction bandpass filter sets $PBPF_{\pm}$.

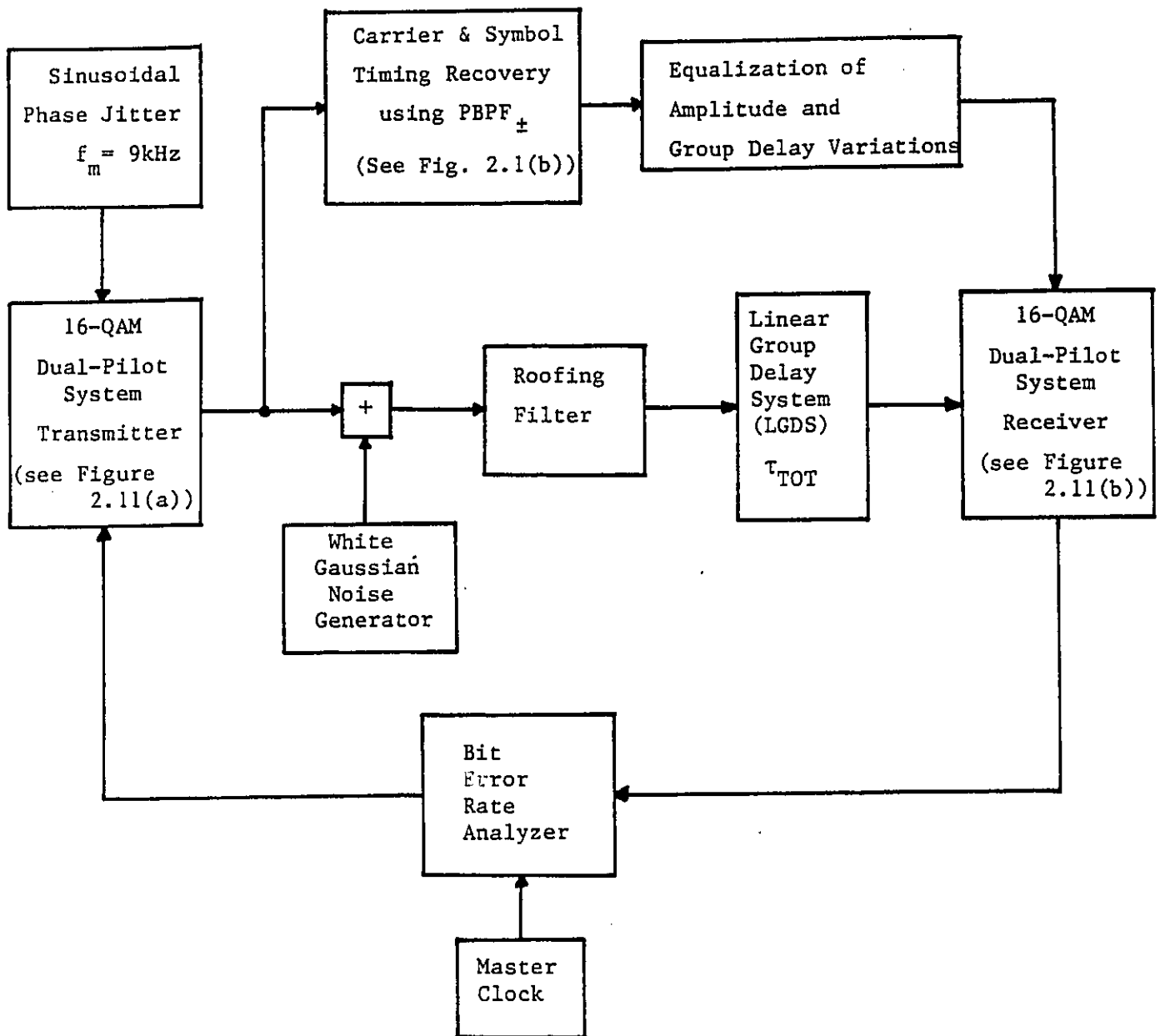


Figure 2.14: Block diagram of the experimental set-up used to test the 16-QAM DPS.

National Library
of Canada

Canadian Theses Service

Bibliothèque nationale
du Canada

Service des thèses canadiennes

NOTICE

THE QUALITY OF THIS MICROFICHE
IS HEAVILY DEPENDENT UPON THE
QUALITY OF THE THESIS SUBMITTED
FOR MICROFILMING.

UNFORTUNATELY THE COLOURED
ILLUSTRATIONS OF THIS THESIS
CAN ONLY YIELD DIFFERENT TONES
OF GREY.

AVIS

LA QUALITE DE CETTE MICROFICHE
DEPEND GRANDEMENT DE LA QUALITE DE LA
THESE SOUMISE AU MICROFILMAGE.

MALHEUREUSEMENT, LES DIFFERENTES
ILLUSTRATIONS EN COULEURS DE CETTE
THESE NE PEUVENT DONNER QUE DES
TEINTES DE GRIS.

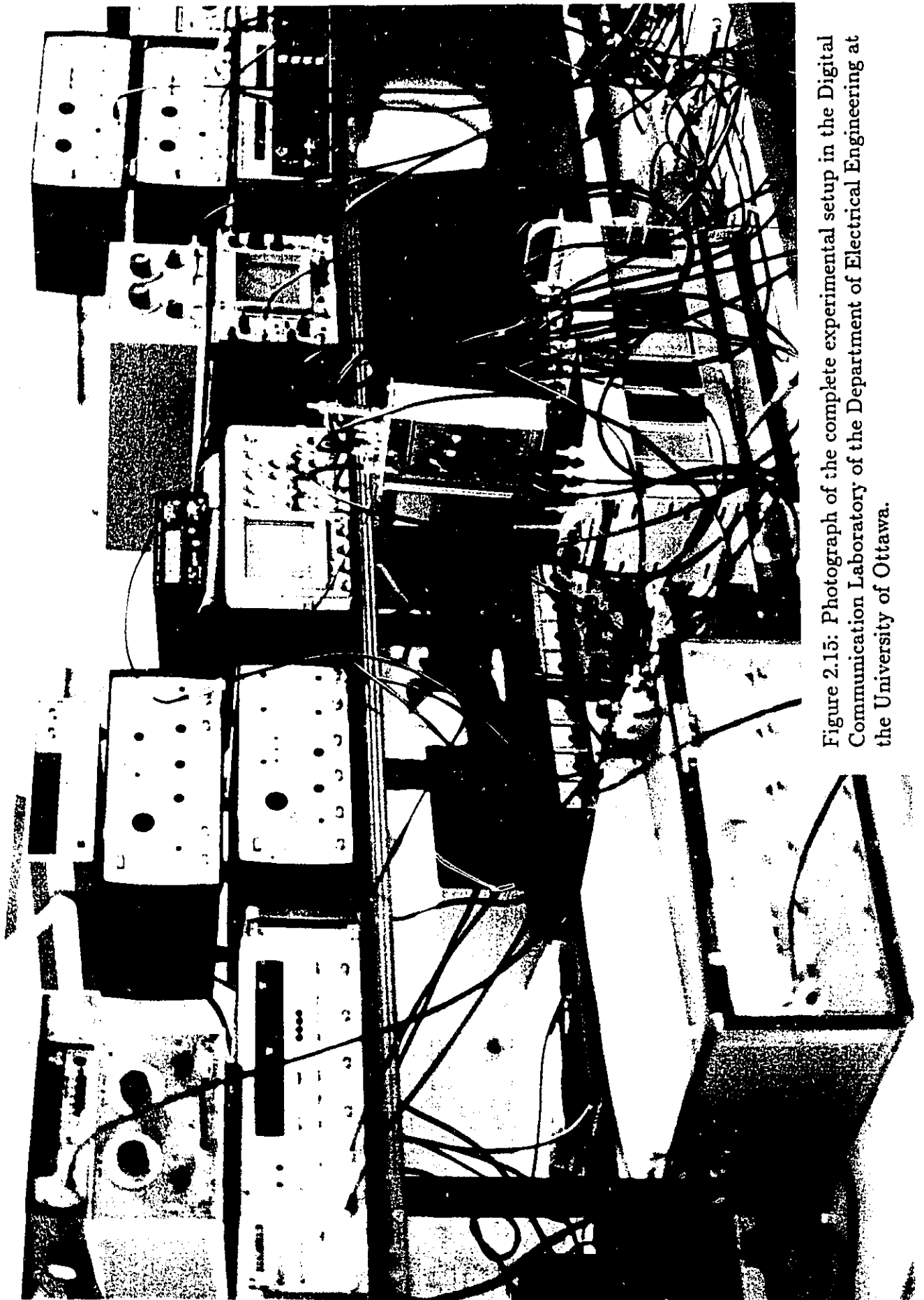


Figure 2.15: Photograph of the complete experimental setup in the Digital Communication Laboratory of the Department of Electrical Engineering at the University of Ottawa.

2.7.2 Experimental Measurements and Results

We first tested the 16-QAM modem back-to-back, i.e. with both the carrier and the symbol timing signals hardwired, without any SCPJ, i.e. only in an AWGN channel. The measured C/N degradation at an error probability of $P_S = 10^{-4}$ was approximately 1.0 dB. Since this represents degradation caused by the 16-QAM modem itself, it will not be included in the subsequent results. In this way, a more fair comparison between the measured and analytical results can be made. The main source of this degradation is imperfect filtering, as it can be seen from a photograph of the demodulated eye-diagrams of the both I- and Q-channels, which is illustrated in Fig. 2.16. Further, we tested the modem using the dual pilots to obtain both the carrier and the timing recovery signals, again in the presence of only AWGN. We employed the set of filters S_1 and by appropriately delaying the received signal by 35 μ sec, we were able to achieve performance almost identical to that of the hardwired modem. We used an amplitude equalizer to adjust for the differences in the amplitude characteristics of the extraction filters at 1.3 MHz and 1.7 MHz. As previously mentioned, it is important to recognize at this point, that by taking into account the power, which is allocated into the pilots (i.e. for our experiments 10 dB below the power of the 16-QAM signal) as well as the 0.3 dB degradation which will result due to the differential encoding of the input data, we end up having an additional degradation of approximately 0.7 dB. However, this degradation is present to any DPS (narrowband or wideband) and thus we will consider it of no significance, as far as the scope of this work is concerned.

Finally, the 16-QAM DPS was tested in the presence of both AWGN and SCPJ. The obtained results are summarized in Table 2.3 as C/N degradation in dB at $P_S = 10^{-4}$. The analytical results which appear in this Table were obtained by employing the same methodology as described in section 2.5 and using the appropriate filter characteristics. We first experimentally verified the fact that in the case of no SCPJ cancellation the C/N degradation is only a function of β and does not depend on f_m . As it can be seen from Table 2.3, the measured results for $\beta = 0.1$ are very close to the analytically derived ones. Similar results were obtained for values of $\beta < 0.1$. For $\beta > 0.15$ we observed

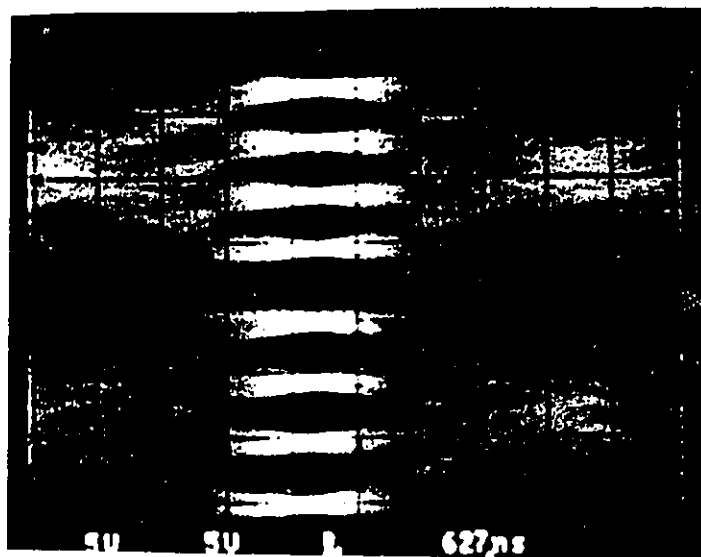


Figure 2.16: Demodulated eye-diagrams of both I- and Q-channels of the 16-QAM modem, showing the ISI caused mainly by imperfect filtering.

Carrier-to-Noise (C/N) Degradation in dB at $P_S=10^{-4}$				
	Analytical Results		Measured Performance [†]	
	$\beta=0.1$	$\beta=0.2$	$\beta=0.1$	$\beta=0.2$
No phase jitter cancellation ($B'=0$)	1.8	8.0	2.0 ± 0.2	∞ [§]
Partial phase jitter cancellation (Filter set S_1 ; $B'=2.3f_m$)	1.7	2.8	2.0 ± 0.2	4.0 ± 0.4
Practically total phase jitter cancellation (Filter set S_2 ; $B'=6.6f_m$)	0.2	0.5	0.5 ± 0.2	0.9 ± 0.2
	0.8	2.5	1.2 ± 0.2	3.5 ± 0.4
	≈ 0	≈ 0	≈ 0	0.2 ± 0.1

[§] Error floor at $P_S=10^{-3}$

[†] In order to have a direct comparison with the analytically derived results, these measured results do not include the 16-QAM modem back-to-back performance degradation, which was approximately 1.0dB. Furthermore, for the same reason, the additional carrier of 0.4dB, which is required to transmit the pilot tones is not included.

Table 2.3: Comparison between the analytical results and the measured performance of a 16-QAM DPS operated in the presence of AWGN and sinusoidal SCPJ.

increased error floors occurring at an error rate of more than 10^{-3} , which is for many practical applications unacceptably high.

By employing the filter set S_1 , it became apparent that both amplitude and group delay distortions caused by the bandpass extraction filters had to be at least partially compensated for, so that performance of less than 10^{-4} could be achieved. In fact, in this case the obtained error floors were higher than the equivalent (of same β) error floors observed with no phase jitter cancellation. Thus, we designed and implemented using standard operational amplifiers (LM 318) a number of amplitude and group delay variations equalizers. These equalizers were employed, as shown in Fig. 2.14, in cascade with the pilot extraction filters $PBPF_{\pm}$, so that they would cancel the introduced amplitude and group delay variation distortions. Furthermore, we similarly implemented a number of Linear Group Delay Systems (LGDS), which were used to delay the received 16-QAM signal by up to 40 μsec , as illustrated in Fig. 2.14.

In Table 2.3, we present a summary of some of the obtained results. With reference to the distortion compensation, they are divided into two categories. The first one, referred to as partial distortion compensation, presents results obtained for relatively small values of uncompensated distortions (ϵ'_r, τ'_r and $\Delta\tau'_r$). The second category summarizes results obtained by employing maximal practically achievable compensation for both amplitude and group delay distortions. Furthermore, ϵ'_r of S_2 was intentionally kept to a small value, since we will compare this set of filters with the wideband system case (Section 2.5.1) where we assume that there is no amplitude distortion. As it can be seen from this table, the measured results are in good agreement with those obtained analytically. For example for $\beta = 0.1$ the differences are not more than 2.0 dB whereas for $\beta = 0.1$ they are less than 0.5 dB. Furthermore, performance degradation within 0.2 dB from the theory was obtained by employing fairly wideband extraction filters ($B' \doteq 6.6f_m$). This residual degradation is believed to be due to nonideal distortion compensation. On the other hand, using narrowband extraction filters ($B' \doteq 2.3f_m$) the residual degradation was 0.9 dB. However, in this case the extracted thermal noise will be lower, as compared to the wideband case, and this tradeoff

should be considered in applications with low Signal-to-Noise ratio and/or low bit rate systems.

2.8 CONCLUSIONS

An out-of-band pilot based method of extracting and cancelling SCPJ was introduced and analyzed. It was demonstrated that by employing a wideband pilot extraction system instead of the frequently employed minimal bandwidth one, the system performance can be improved considerably. Analytical results for the C/N degradation of a 16-QAM system operated in the presence of sinusoidal, with frequency f_m , SCPJ were obtained as a function of the extraction filters bandwidth B' , as well as their amplitude and group delay distortions. It was shown that in order to achieve efficient SCPJ cancellation it is important to compensate for both of these distortions. However, as far as implementation issues are concerned, group delay distortion seems to be a more prominent source of degradation. Many of the analytical results were verified by employing a prototype 16-QAM DPS operated in the presence of sinusoidal SCPJ. All results indicate that with proper distortion compensation for a peak-to-peak phase jitter of 0.4 rads an extraction bandwidth of $B' \doteq 6.6f_m$ results in a negligible degradation, as compared to an ideal 16-QAM DPS.

Chapter 3

IN-BAND PILOT INSERTION FOR SIMULTANEOUS CARRIER AND SYMBOL TIMING RECOVERY IN AN AWGN CHANNEL

3.1 INTRODUCTION

As we have seen with great details in Chapter 2, the DPS is an effective means of performing two simultaneous functions at the receiver of an M -ary QAM system,¹ i.e.:

- (a) Combat the effects of SCPJ.
- (b) Provide a carrier and symbol timing reference for coherent demodulation and detection.

¹Again here we would like to point out that although the DPS is a generic scheme which can be applied to any digital modulation system, we will restrict our discussions only to QAM modulation formats.

Nevertheless there are a number of drawbacks associated with the DPS. Apart from the fact that this scheme requires an excess bandwidth to accommodate the pilots as compared to a SCS, one additional disadvantage is that the pilots are located at the ends of the channel spectrum where amplitude and phase characteristics are least stable. Furthermore, because the pilots are located near the edges of the band, they are susceptible to adjacent channel interference.

One way of eliminating the required excessive bandwidth and to avoid filter instabilities and adjacent channel interference, is the use of an In-Band Dual Pilot System (IBDPS). Fig. 3.1(a) illustrates the pilots and the data of such an approach, assuming the latter being a QAM signal filtered by a raised-cosine filter of excess bandwidth α . From Fig. 3.1(b), which illustrates the carrier recovery subsystem of the IBDPS, it is clear that it has the same structure as that of the one employed by the DPS (see Fig. 2.1). The only difference between the two schemes is the values of the pilot frequencies $\omega_c \pm \omega_p$ and the corresponding center frequencies of the pilot extraction bandpass filters $PBPF_{\pm}$. Despite the spectral efficiency of the IBDPS, as compared to the DPS, the disadvantage of this scheme is that there is mutual interference between the digital QAM signal and the pilots. As a result, both interferences have to be analyzed and their impact on the performance of the QAM signal has to be evaluated. Furthermore, due to the multiplicity of the interferences present in the IBDPS, it was decided that our research efforts will focus on system applications where the amount of SCPJ is negligible and therefore can be ignored. In other words, we will assume that the proposed IBDPS-QAM system is operated in an AWGN environment only. Thus the purpose of this IBDPS is to provide the QAM receiver, by means of the two in-band inserted pilot tones, with a simple method of recovering *both carrier and symbol timing reference signals* for coherent demodulation and synchronous detection.

It is of importance to notice that the IBDPS is a generic scheme which can be also employed with other than QAM modulation formats, such as for example Quadrature Partial Response (QPR) systems. Nevertheless, we will restrict our work in the analysis and evaluation of the IBDPS employing QAM signalling format.

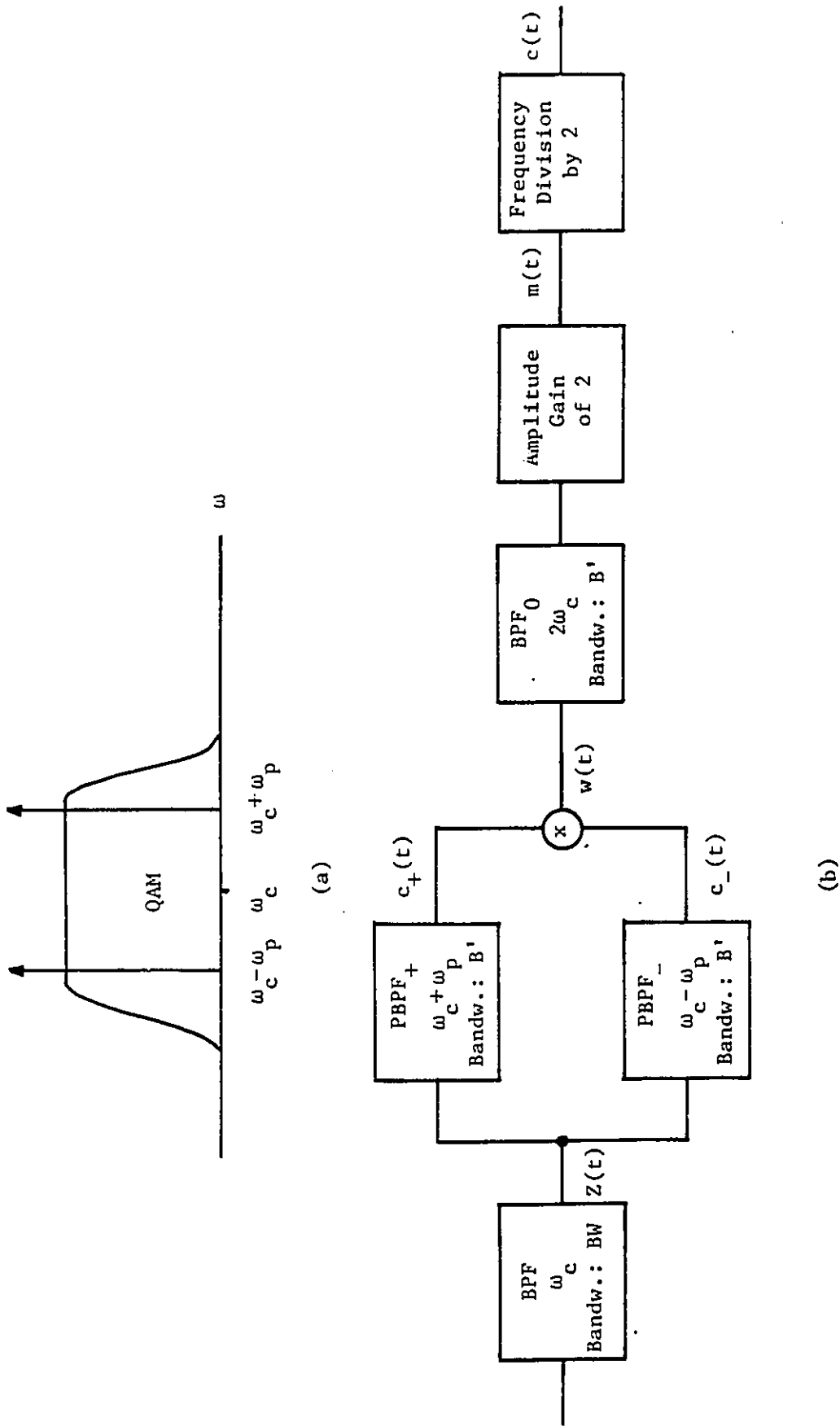


Figure 3.1: In-Band Dual Pilot System (IBDPS) configuration:
(a) Transmitted signal power spectrum.
(b) Carrier recovery subsystem. $PBPF_{\pm}$ represent the pilot extraction bandpass filters.

After this introductory section the rest of Chapter 3 is organized as follows. In Section 3.2 we present a detailed description of the IBDPS-QAM system. Furthermore, the interference of the inserted pilots into the QAM signal and vice-versa are modelled and their effect on the overall system performance is analytically evaluated. In Section 3.3 we present performance evaluation results of a 16-QAM IBDPS obtained by means of computer simulation as well as many experimental results of a prototype 16-QAM IBDPS system. Finally, Section 3.4 concludes this chapter.

3.2 IBDPS-QAM SYSTEM DESCRIPTION AND INTERFERENCE ANALYSIS

3.2.1 System Description

The received signal $Z(t)$, which is corrupted by additive white Gaussian noise with a two-sided spectral density of $N_0/2$ can be represented as:

$$\begin{aligned} Z(t) = & A_p \cos[(\omega_c + \omega_p)t] + A_p \cos[(\omega_c - \omega_p)t] \\ & + S_I \cos(\omega_c t) + S_Q(t) \sin(\omega_c t) \\ & + N_i(t) \cos(\omega_c t) + N_q(t) \sin(\omega_c t). \end{aligned} \quad (3.1)$$

In Eq. 3.1² the first two terms represent the pilot tones of amplitude A_p and of radian frequencies $\omega_c \pm \omega_p$, whereas the last two terms represent the bandpass thermal noise with each of its components $N_i(t)$ and $N_q(t)$ having a variance of $N_0(BW)/2$. BW is the bandwidth of the bandpass filter BPF (see Fig. 3.1). The M -ary QAM signal is represented by the third and fourth term of Eq. 3.1, where ω_c is the carrier radian frequency and $S_I(t)$ and $S_Q(t)$ are given by:

$$S_I(t) = \sum_{n=-\infty}^{\infty} I_n g(t - nT)$$

²The above equation is the same as Eq. 2.1 with $\phi(t) = 0$. However, for the reader's convenience we will repeat the description of the different signals which are involved.

$$S_Q(t) = \sum_{n=-\infty}^{\infty} Q_n g(t - nT). \quad (3.2)$$

I_n and Q_n are independent discrete random variables (impulses) taking equally likely values from the alphabet $\{\pm 1, \pm 3, \dots, \pm\sqrt{M} - 1\}$ for $M = 4, 16, 64, \dots$, T is the symbol duration and $g(t)$ is the baseband pulse shape. We will assume that $g(t)$ is the well-known raised-cosine pulse with an excess bandwidth of α and also that $g(t)$ includes both transmit and receive filters. The Fourier transform of $g(t)$ is given by $G(f)$ as follows:

$$G(f) = \begin{cases} 1, & \text{for } 0 < f < \frac{1-\alpha}{2T} \\ \cos^2[\frac{\pi T}{2\alpha}(f - \frac{1-\alpha}{2T})], & \text{for } \frac{1-\alpha}{2T} < f < \frac{1+\alpha}{2T} \\ 0, & \text{for } f > \frac{1+\alpha}{2T}. \end{cases} \quad (3.3)$$

It should be noted that for the pilots to be inserted within the bandwidth occupied by the QAM signal their radian frequency, ω_p , has to be:³

$$0 < \omega_p < (1 + \alpha)\frac{\pi}{T}. \quad (3.4)$$

As stated before the IBDPS-QAM system suffers from two sources of degradation. Firstly, there is interference from the pilots into the QAM signal, thus causing system degradation. Secondly, since eventually we want to extract the pilot signals in order to recover the carrier, as is shown in Fig. 3.1(b), there will be interference (data phase jitter) from the QAM signal into the pilots, resulting in a nonideal recovered reference signal. These two interferences will be analyzed individually in the next section.

3.2.2 Pilot Interference Into the QAM Signal

3.2.2.1 Pilot Frequency Considerations

It proves to be instructive to investigate the interference mechanism of the inserted pilot tone into the QAM system in an equivalent baseband system. In view of this let us assume, without any loss of generality, that $\omega_c = 0$. Let also $\{a_n\}$ be an arbitrary sequence with each

³ ω_p is the baseband radian frequency of the pilot tones. For a modulated signal, however, it represents the pilots frequency offset from the carrier.

symbol a_n having a duration of T and t_n^* its optimal sampling instant, as shown in Fig. 3.2. The necessary and sufficient conditions for not having any interference from an inserted pilot, $p(t)$, of frequency f_p into the sequence $\{a_n\}$ are:

$$p(t_n^*) = A_p \cos(2\pi f_p t_n^*) = 0 \quad (3.5)$$

and at the same time

$$p(t_{n+1}^*) = A_p \cos(2\pi f_p t_{n+1}^*) = 0. \quad (3.6)$$

Equivalently we can write:

$$2\pi f_p t_n^* = \begin{cases} l(\pi/2) \\ \text{or} \\ l(3\pi/2) \end{cases} \quad (3.7)$$

and

$$2\pi f_p t_{n+1}^* = \begin{cases} l(\pi/2) + k\pi \\ \text{or} \\ l(3\pi/2) + k\pi \end{cases} \quad (3.8)$$

with $k, l \in \{\dots, -1, 0, 1, \dots\}$.

From Eqs. 3.7, 3.8 and since $t_{n+1}^* - t_n^* = T$, this condition becomes:

$$f_p = k \frac{1}{2T}. \quad (3.9)$$

The above equation indicates that an infinite number of nonzero frequencies, namely $(1/2T)$, $(1/T)$, ..., do not interfere with the digital signal. However, *the only in-band nonzero pilot frequency, which does not cause any interference into a digital sequence of symbol rate T is the Nyquist frequency $f_N = 1/2T$* . Furthermore, as will be explained later on in this chapter, by inserting the pilots at the Nyquist frequency, we can at the receiver very simply recover *both the carrier and the symbol timing signal* using the transmitted pilot tones.

Nevertheless, it should be mentioned that for $f_p = 0$ there is also no interference from the pilot tones into the QAM signal. In this case, only one pilot tone is inserted at the carrier frequency (that is after modulation). This system configuration, however, does not provide a pilot based symbol timing signal at the receiver and therefore will not be investigated further.

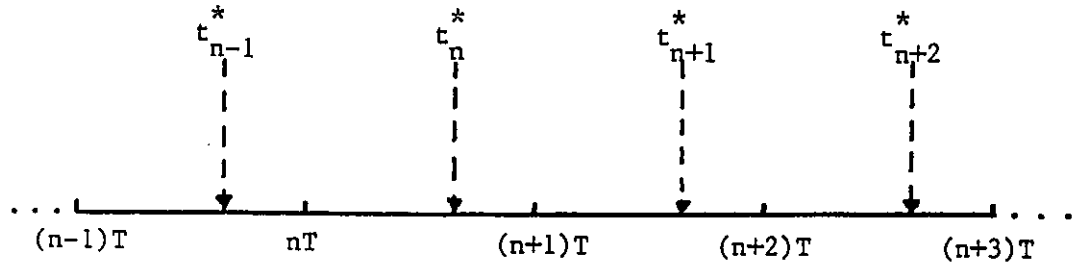


Figure 3.2: Arbitrary sequence of transmitted symbols each having duration of T . The sampling instants $\dots, t_{n-1}^*, t_n^*, t_{n+1}^*, \dots$ are assumed to be the optimum (i.e. result in minimum ISI).

3.2.2.2 Pilot Level Considerations

As explained in Chapter 2, where we considered the DPS, the power which is allocated for the transmission of the pilot tones results in an equivalent C/N degradation D_p . The same arguments can be made also for the IBDPS. In fact the results which were presented in the later part of Section 2.5.1 and in particular in Table 2.1 are also valid for the IBDPS. Here we will only summarize our previously obtained results, i.e. that if the ratio m of the pilot tones to the power which is required to transmit the digitally modulated signal is small (typically $m < 0.1$) then this degradation D_p is less than 0.4 dB.

3.2.3 Analysis of the QAM Signal Interference Into the Recovered Carrier

Let us assume that the pilots inserted at the Nyquist frequency are extracted by the extraction filters $PBPF_{\pm}$, which have an extraction bandwidth B' with $B' \ll 1/T$, i.e. $B' < 0.01/T$. The data interference into the extracted pilot tones can be represented by two signals, the in-phase to the pilots interference, $i_{I\pm}(t)$,⁴ and the quadrature interference, $i_{Q\pm}(t)$. Further, since single frequency components (i.e. the pilot tones) are to be extracted, we can assume that the delay occurring from the pilot bandpass extraction filters $PBPF_{\pm}$ can be easily equalized. In view of the above, the extracted pilot tones $c_{\pm}(t)$ can be written as:

$$c_{\pm}(t) = A_p \cos[(\omega_c \pm \omega_p)t] + i_{I\pm}(t) \cos[(\omega_c \pm \omega_p)t] + i_{Q\pm}(t) \sin[(\omega_c \pm \omega_p)t]. \quad (3.10)$$

In the above expression of the extracted pilots $c_{\pm}(t)$, we did not include terms which represent the AWGN. The reason for this is that in all spectrally efficient systems the carrier power (i.e. the data power, which in our case is the dominating source of interference into the extracted pilots) is much higher as compared to that of the Gaussian noise. For example, a 16-QAM scheme which is operated at a

⁴The + and - signs for both $i_{I\pm}$ and $i_{Q\pm}$ refer to the equivalent $PBPF_{\pm}$ filters.

$P_S = 10^{-4}$ requires a $C/N \doteq 19$ dB. This suggests that the Gaussian noise interference is 19 dB below the data interference for a 16-QAM scheme and thus can be considered to have a negligible effect on the extracted pilot tones.

As shown in Fig. 3.1 and similar to Chapter 2 (Section 2.3), in order to recover the carrier $c(t)$, $c_+(t)$ and $c_-(t)$ have first to be mixed. Afterwards, the frequency component at $2\omega_c$ is extracted by the bandpass filter BPF_0 and finally the resulting signal is amplified and frequency divide by two. Thus, as shown in Fig. 3.1, $m(t)$ can be expressed as:

$$m(t) = \{[A_p + i_{I+}(t)][A_p + i_{I-}(t)] - [i_{Q+}(t)i_{Q-}(t)]\} \cos(2\omega_c t) \\ + \{[A_p + i_{I+}(t)]i_{Q-}(t) + [A_p + i_{I-}(t)]i_{Q+}(t)\} \sin(2\omega_c t) \quad (3.11)$$

In order to simplify our analysis we will assume that the transfer function of the pilot extraction filters are identical,⁵ so that the interferences from both filters can be considered to be the same⁶ i.e.:

$$i_I(t) = i_{I+}(t) = i_{I-}(t) \quad (3.12)$$

and

$$i_Q(t) = i_{Q+} = i_{Q-}(t). \quad (3.13)$$

Furthermore, since it is expected that the interference signals $i_I(t)$ and $i_Q(t)$ will be much smaller in amplitude than the pilot signals,⁷ we can assume that:

$$[A_p + i_I(t)]^2 \gg i_Q^2(t). \quad (3.14)$$

In view of the above $m(t)$ can be rewritten as:

$$m(t) = \{[A_p + i_I(t)]^2\} \cos(2\omega_c t) + \{2[A_p + i_I(t)]i_Q(t)\} \sin(2\omega_c t) \quad (3.15)$$

⁵This is not a unreasonable assumption, since nowadays manufacturers can produce filters which have fairly identical transfer functions. This is particularly true for crystal filter implementations, such as the ones employed in our experiments, as we will see later on in the chapter.

⁶Strictly speaking, the two interferences have the same statistics. However, by making the assumption that they are equal, the subsequent analysis becomes mathematically more trackable.

⁷We will see later on in this section that this is true due to the fact that we have assumed that $B' \ll 1/T$.

or equivalently:

$$m(t) = \sqrt{B_I^2 + B_Q^2} \cos(2\omega_c t + \theta_{dpj}) \quad (3.16)$$

with B_I, B_Q and $\theta_{dpj}(t)$ given by:

$$\begin{aligned} B_I &= [A_p + i_I(t)]^2 \\ B_Q &= 2[A_p + i_I(t)]i_Q(t) \\ \theta_{dpj}(t) &= -\arctan(B_Q/B_I). \end{aligned} \quad (3.17)$$

Finally, after frequency dividing this signal by two, the recovered carrier $c(t)$ can be written as:

$$c(t) = B \cos[\omega_c t + \phi_{dpj}(t)] \quad (3.18)$$

with $B = \sqrt{B_I^2 + B_Q^2}$ the amplitude of the recovered carrier and $\phi_{dpj}(t) = \theta_{dpj}(t)/2$ being the carrier phase jitter due to the extracted QAM data interference.

As previously stated, due to the relatively small interference signals $i_I(t)$ and $i_Q(t)$ it is clear that the fluctuations of the amplitude B of the recovered carrier will be small. Furthermore, by passing the recovered carrier through an automatic gain control subsystem (or simply a hardlimiter) these amplitude fluctuations can be removed. Nevertheless, the data phase jitter $\phi_{dpj}(t)$, which results due to the terms B_I and B_Q of Eq. 3.17 has to and will be analyzed further.

$\phi_{dpj}(t)$ can be rewritten in the following form:

$$\phi_{dpj}(t) = -0.5 \arctan\left[\frac{2i_Q(t)}{A_p + i_I(t)}\right]. \quad (3.19)$$

Using Eq. 3.14, the above equation can be approximated by:

$$\phi_{dpj}(t) \doteq -\frac{i_Q(t)}{A_p + i_I(t)}.$$

Further, by assuming that $A_p \gg i_I(t)$, we finally end up with the following approximation:

$$\phi_{dpj}(t) \doteq -\frac{i_Q(t)}{A_p}.$$

Since it is more convenient to derive expressions for the equivalent power of the interference signals, the variance of $\phi_{dpj}(t)$, σ_ϕ^2 , is given by:

$$\sigma_\phi^2 \doteq \frac{P_{IQ}}{A_p^2} \quad (3.20)$$

where P_{IQ} is the power of the interference signal $i_Q(t)$.

Fig. 3.3(a) illustrates the transmitted QAM signal with the pilot signals inserted at the Nyquist frequency together and the QAM data interference shown with a dashed area. Furthermore, in Figs. 3.3(b) and (c), the low pass equivalent (in-phase and quadrature-phase) of the interference signals are shown. Clearly the power, P_{II} , of the in-phase interfering signal $i_I(t)$ is given by:

$$\begin{aligned} P_{II} &= \int_{-B'/2}^{B'/2} \left(\frac{\sqrt{2}}{2} \right)^2 df \\ &= 0.5B'. \end{aligned} \quad (3.21)$$

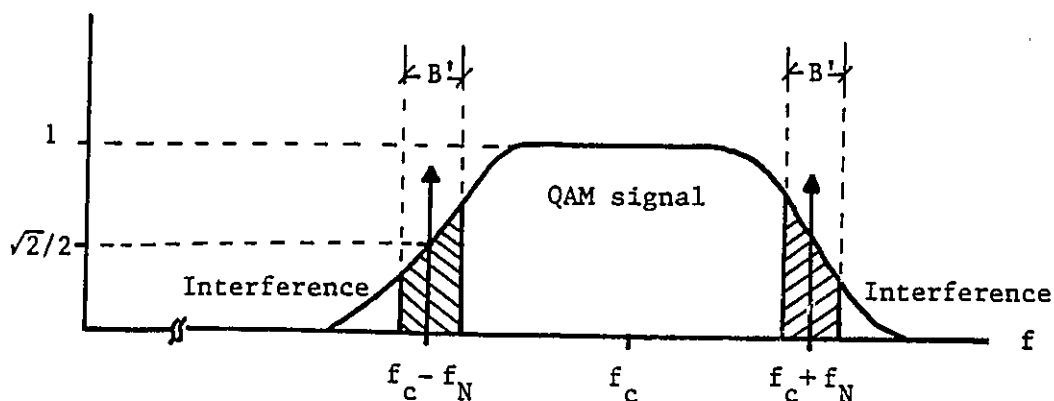
Similarly, the power for the quadrature interfering signal $i_Q(t)$ can be calculated as follows:

$$\begin{aligned} P_{IQ} &= \int_{-B'/2}^{B'/2} \left\{ \cos\left[\frac{\pi T}{2\alpha}\left(f - \frac{1-\alpha}{2T}\right)\right] - \frac{\sqrt{2}}{2} \right\}^2 df \\ &= \int_{-B'/2}^{B'/2} \cos^2\left[\frac{\pi T}{2\alpha}\left(f - \frac{1-\alpha}{2T}\right)\right] df \\ &\quad + \int_{-B'/2}^{B'/2} \left(\frac{\sqrt{2}}{2} \right)^2 df \\ &\quad - \sqrt{2} \int_{-B'/2}^{B'/2} \cos\left[\frac{\pi T}{2\alpha}\left(f - \frac{1-\alpha}{2T}\right)\right] df. \end{aligned} \quad (3.22)$$

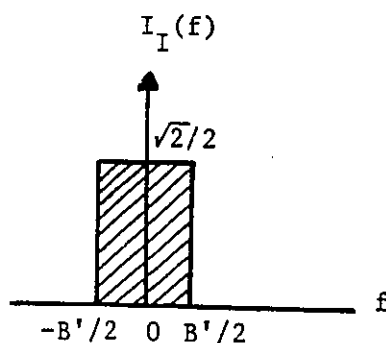
Straightforward evaluation of the above three integrals results in the following:

$$\begin{aligned} \int_{-B'/2}^{B'/2} \cos^2\left[\frac{\pi T}{2\alpha}\left(f - \frac{1-\alpha}{2T}\right)\right] df &= B'/2 \\ \int_{-B'/2}^{B'/2} \left(\frac{\sqrt{2}}{2} \right)^2 df &= B'/2 \\ \sqrt{2} \int_{-B'/2}^{B'/2} \cos\left[\frac{\pi T}{2\alpha}\left(f - \frac{1-\alpha}{2T}\right)\right] df &= \frac{4\alpha}{\pi T} \sin\left(\frac{\pi T B'}{4\alpha}\right). \end{aligned} \quad (3.23)$$

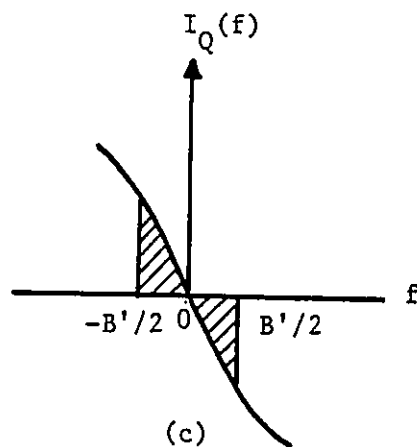
Modulated Signal
Spectrum



(a)



(b)



(c)

Figure 3.3: Data into the pilots interference considerations of the IB-DPS-QAM system employing $\sqrt{\alpha}$ raised-cosine filters with the pilot tones inserted at the Nyquist frequency. B' is the bandwidth of the pilot extraction bandpass filters $PBPF_{\pm}$.

(a) Transmitted (modulated) IBDPS-QAM signal spectrum. Due to the symmetry only the positive frequencies are shown. The shape of the QAM signal is given by $\sqrt{G(f)}$ of Eq. 3.3. The dashed areas represent the unwanted data interference into the extracted pilots.

(b) Baseband equivalent spectrum of the in-phase component of the extracted data interference.

(c) Baseband equivalent spectrum of the quadrature-phase component of the extracted data interference.

Therefore P_{IQ} is given by:

$$P_{IQ} = B' - \frac{4\alpha}{\pi T} \sin\left(\frac{\pi T B'}{4\alpha}\right). \quad (3.24)$$

It is convenient to express the bandpass pilot extraction bandwidth B' as well as the power which is allocated to the pilot tones $P_{pilots} = A_p^2$ as a function of the symbol rate $1/T$. Therefore let us represent B' as:

$$B' = \frac{\psi}{T} \quad (3.25)$$

with ψ being defined as the pilot extraction bandpass filters bandwidth B' multiplied by the symbol duration T . Furthermore, the normalized power which is allocated to the QAM signal is clearly given by the following equation:⁸

$$\begin{aligned} P_{QAM} &= \int_{-\frac{1+\alpha}{2T}}^{\frac{1+\alpha}{2T}} \cos^2\left[\frac{\pi T}{2\alpha}\left(f - \frac{1-\alpha}{2T}\right)\right] df \\ &= \frac{1}{T}. \end{aligned} \quad (3.26)$$

As explained in Section 3.2.2.2, P_{pilots} should be 10 dB below the power of the QAM signal. Thus we have:

$$P_{pilots} = 0.1 P_{QAM} \quad (3.27)$$

or equivalently:

$$A_p^2 = \frac{1}{10T}. \quad (3.28)$$

Finally, by substituting the results of Eqs. 3.24, 3.25 and 3.28 into Eq. 3.20, the carrier phase jitter ϕ_{dpj} can be approximated by:

$$\sigma_\phi^2 \doteq 10\left[\psi - \frac{4\alpha}{\pi} \cdot \sin\left(\frac{\pi\psi}{4\alpha}\right)\right]. \quad (3.29)$$

Plots of ϕ_{dpj} as a function of ψ and for several values of α are shown in Fig. 3.4. From this figure it is clear that for a given ψ the amount of data phase jitter ϕ_{dpj} depends on the excess bandwidth α of the

⁸Here we assume that the average energy of the transmitted symbols is unity.

employed raised-cosine filters. The reason for this is the direct dependence of ϕ_{dpj} on the quadrature component of the data interference signal. Clearly, as is illustrated in Fig. 3.3(c), the amount of data interference increases with α decreasing. However, for $\psi < 0.01$, the amount of data phase jitter becomes negligibly small.

3.3 COMPUTER SIMULATION SYSTEM DESCRIPTION AND PERFORMANCE EVALUATION RESULTS

In this section we will present performance evaluation results of a 16-QAM system employing the IBDPS scheme. The evaluation methodology which will be used is based on computer simulation as well as experiments.

First we will briefly describe our computer simulation method and discuss the system model employed. A number of simulated performance evaluation results for a 16-QAM IBDPS will then be presented. Finally, experimental results of a prototype 16-QAM IBDPS will be reported. This prototype is essentially the same system which was described and evaluated in Section 2.6 with the only difference being that here the pilot tones are inserted *within the bandwidth occupied by the 16-QAM signal*. Wherever possible comparison of both simulated and experimental results with the previously obtained analytical results will be made.

3.3.1 Computer Simulation System Description

In order to evaluate, by means of computer simulation, the performance of the proposed IBDPS 16-QAM scheme we have employed the quasi-analytical symbol error rate estimation technique [106]. The main feature of this technique is that the probability of error is calculated for each symbol using an analytical expression. Thus, the average symbol error rate is computed by averaging the calculated symbol

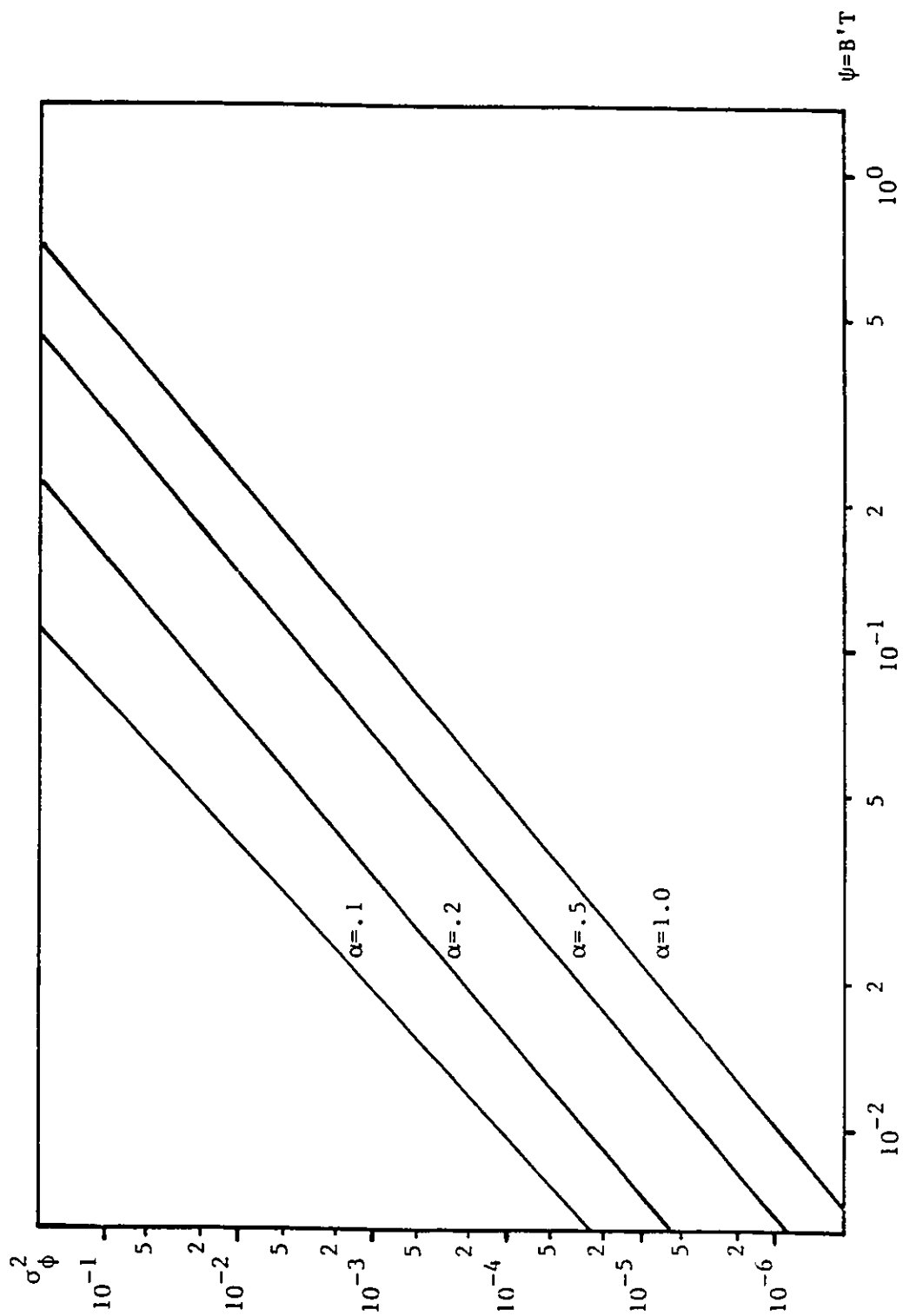


Figure 3.4: Plot of the variance of the carrier data phase jitter, σ_ϕ^2 , as a function of the pilot extraction bandwidth ψ (normalized to the symbol rate) for different values of the excess bandwidth α (see Eq. 3.29).

error rates for all received symbols. The quasi-analytical approach results in considerable savings in computer run-time by avoiding direct noise simulation and symbol error counting, as is the case for Monte Carlo simulation methods [106].

The block diagram of the employed computer simulation model of a 16-QAM IBDPS is illustrated in Fig. 3.5. In the simulations the binary non-return-to-zero data, which take values $+1$ or -1 , for both I- and Q-channel are generated by using a random number generator subroutine. Using blocks of 2 bits for each channel we obtain the four amplitude levels $\pm 1, \pm 3$ for the symbols I_n and Q_n , as shown in Fig. 3.5. The symbol rate f_s is chosen to be 200 kBaud/s, i.e. 800 kb/s for a 16-QAM system, since this bit rate is used in our experimental 16-QAM IBDPS modem,⁹ as we will see later on in the chapter. Evidently, our results apply in general to other transmission speeds, such as 1.544 Mb/s or 45 Mb/s.

We are transmitting 2^{14} symbols for each of the I- and Q-channels with each symbol represented by 32 samples. The transmit and receive filters are ideal $\sqrt{\alpha}$ raised-cosine filters, having a roll-off factor α . Since we are transmitting symbols each having a duration of $T = 1/f_s$, a $(x)/\sin(x)$ amplitude equalizer is added to the transmit $\sqrt{\alpha}$ raised-cosine filter, so that the overall filtering strategy satisfies the first Nyquist criterion [14, Chapter 8]. The eye-diagram of this ideal case with $\alpha = 1.0$ is shown in Fig. 3.6. We can observe that at the ideal sampling instants (i.e. middle of the eyes) there is no ISI. The filtering process is performed using Fast Fourier Transform (FFT) algorithms. For example, the complex signal which includes both I- and Q-channel signals, is transformed to the frequency domain, multiplied by the filter's transfer function and then transformed back to the time domain. The pilot signal is added to the 4-level signal of the I-channel after the transmit filter.

In order to modulate this composite signal a complex sinusoid $e^{j2\pi f_c t}$ having a carrier frequency of $f_c = 1/T$ is used. We have conveniently selected this carrier frequency in order to avoid the frequency division

⁹As previously mentioned, this experimental system is the same as the one described in Section 2.6 with the only difference that here the pilot tones are inserted *within the bandwidth occupied by the 16-QAM signal*.

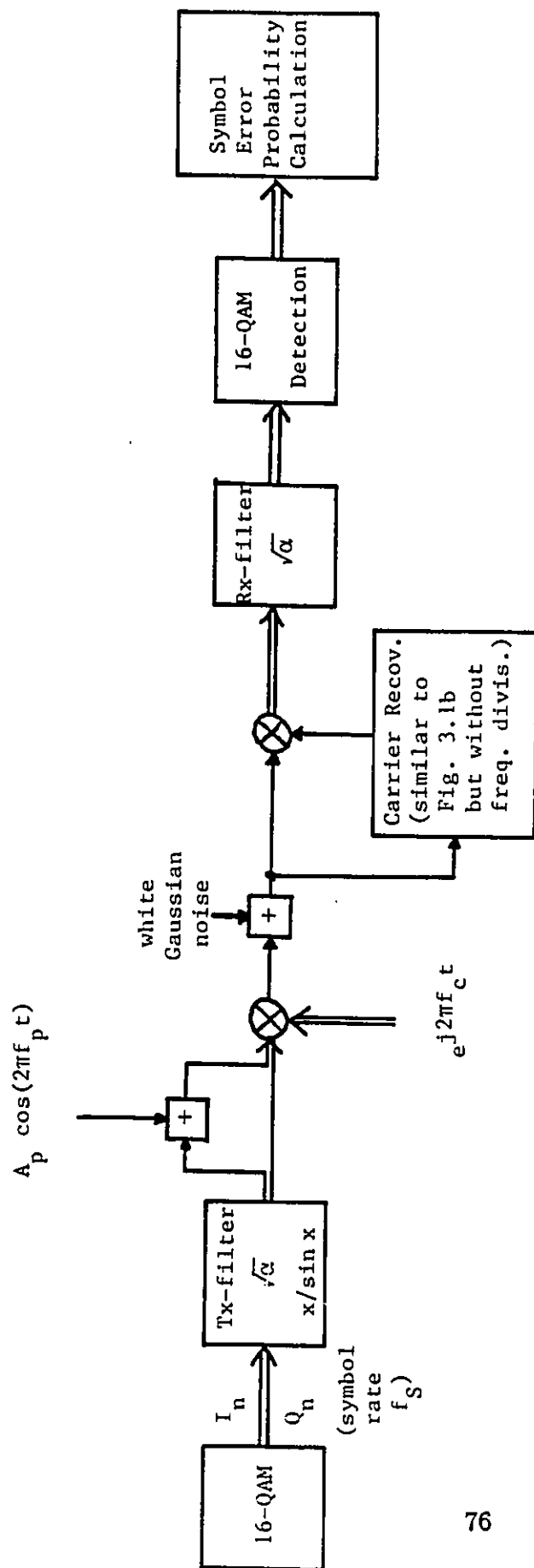


Figure 3.5: Block diagram of the employed computer simulation model of a 16-QAM IBDPS.
 $I_n, Q_n = \{\pm 1, \pm 3\}$
 $f_S = 200 \text{ kBaud/s}$.

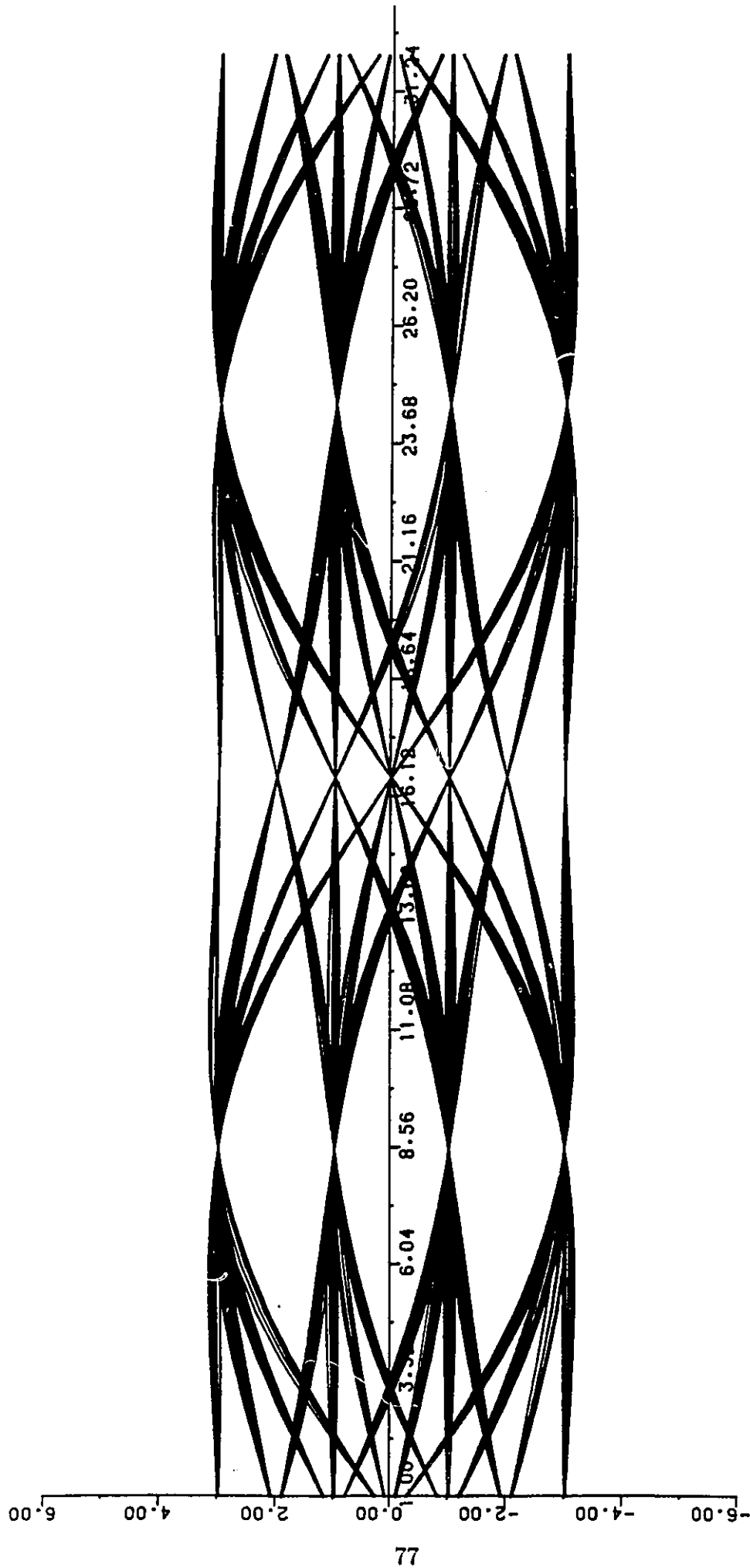


Figure 3.6: 4-level ideal baseband eye-diagram of a 16-QAM system which employs raised-cosine filters with $\alpha = 1.0$.

by two (see Eq. 3.18).¹⁰ In this case, if the pilot insertion frequency is the Nyquist frequency, i.e. $f_p = 1/2T$, then the upconverted pilot tones are located at the frequencies $1/2T$ and $3/2T$. Thus at the receiver, after extracting these two pilot tones and mixing them together the carrier frequency $f_c = 1/T$ results as the lower frequency component of the mixing process.¹¹

After quadrature demodulation the received baseband I- and Q-channel signals are sampled at their maximum eye opening and then a decision about the received symbol is made. By determining the distance from the ideal threshold levels for each of the received sampled symbols and by calculating the average signal and noise power at the input of the threshold detectors, the symbol error probability, P_S , can be computed as follows:

$$P_S = \frac{1}{2N} \sum_{n=1}^N (P_n^I + P_n^Q) \quad (3.30)$$

where $2N$ is the number of transmitted symbols and P_n^I, P_n^Q are the symbol error probabilities of the n -th received symbol for the I- and Q-channel, respectively. P_n^I is given by:

$$P_n^I = \begin{cases} \frac{1}{2} \text{erfc}\left(\frac{|I_n^S - \text{THR1}_n|}{\sqrt{2}\sigma}\right), & \text{for } I_n = \pm 3 \\ \frac{1}{2} \text{erfc}\left(\frac{|I_n^S - \text{THR1}_n|}{\sqrt{2}\sigma}\right) + \frac{1}{2} \text{erfc}\left(\frac{|I_n^S - \text{THR2}_n|}{\sqrt{2}\sigma}\right), & \text{for } I_n = \pm 1 \end{cases}$$

with I_n^S the sampled value of the n -th received symbol of the I-channel. σ^2 is the power of the additive white Gaussian noise at the input of the threshold detector and $\text{THR1}_n, \text{THR2}_n$ are defined as:

$$\begin{aligned} \text{THR1}_n &= |I_n| - 1 \\ \text{THR2}_n &= |I_n| + 1. \end{aligned} \quad (3.31)$$

$\text{erfc}(\cdot)$ is the complementary error function which was defined in Eq. 2.12. A similar formula, which was used in the calculations of the Q-channel, can be obtained by replacing I, I_n^S and I_n with Q, Q_n^S and Q_n , respectively.

¹⁰As previously mentioned (see Section 2.3), this frequency division results in an additional C/N degradation of about 0.3 dB.

¹¹In the simulations for extracting the pilot tones we employed ideal brickwall bandpass filters.

3.3.2 Performance Evaluation Results

Employing the above described computer simulated model the C/N degradation of the 16-QAM IDPS was evaluated for various pilot frequencies and levels. In order to investigate the impact of the choice of the pilot frequency f_p over the widest possible frequency range we assumed raised-cosine filters having $\alpha = 1.0$. Furthermore it was decided, as explained in Section 3.2.2.2, that the level of both pilots will be 10 dB (i.e. $m = 0.1$) below the power level required to transmit the 16-QAM signal. The estimated C/N degradation of 0.4 dB, due to this additional required power allocation to the pilots, was also verified using the computer simulated system. Additionally, the important result of Section 3.2.2.1, namely that the Nyquist frequency is the only in-band pilot frequency which does not cause any interference into the QAM signal was verified. This can be also observed from the eye-diagram of the I-channel with the pilot inserted at $f_p = f_N = 1/2T$ (Nyquist frequency) and $m = 0.1$, as presented in Fig. 3.7(a). We can observe from this figure that although the eye-diagram is distorted by the inserted pilot, at the ideal sampling instant the eye is *completely open, thus there is no ISI*.¹² However, if the pilot frequency is not *exactly the Nyquist frequency* then the degradation which results from the pilot insertion could be unacceptable. The resulting eye-diagrams for $f_p = 1/2.1T \doteq 0.476/T$ and $f_p = 1/1.9T \doteq 0.526/T$ are illustrated in Figs. 3.7(b) and (c), respectively. The obtained C/N degradation, D_{f_p} , at $P_S = 10^{-4}$ for $(0.4/T) < f_p < (1/T)$ is given in Fig. 3.8. Similar to the methodology followed in Chapter 2, for convenience in the interpretation of the obtained results and for comparison purposes, we have not included the DPS's inherent degradation of 0.7 dB. As previously mentioned, this inherent degradation D_{DPS} results due to the power which is allocated to the pilot tones (for $m = 0.1$ we get $D_{DPS} = 0.4$ dB) as well as due to the differential encoding of the input data sequence to combat the 180 degree carrier phase ambiguity ($D_{DPS} = 0.3$ dB). It is clear from Fig. 3.8 that the more inside into the QAM bandwidth

¹²We have to notice that in this case the phase of the inserted at the Nyquist frequency pilot is equally important, so that its amplitude becomes zero when the eye-diagram is at its maximum opening.

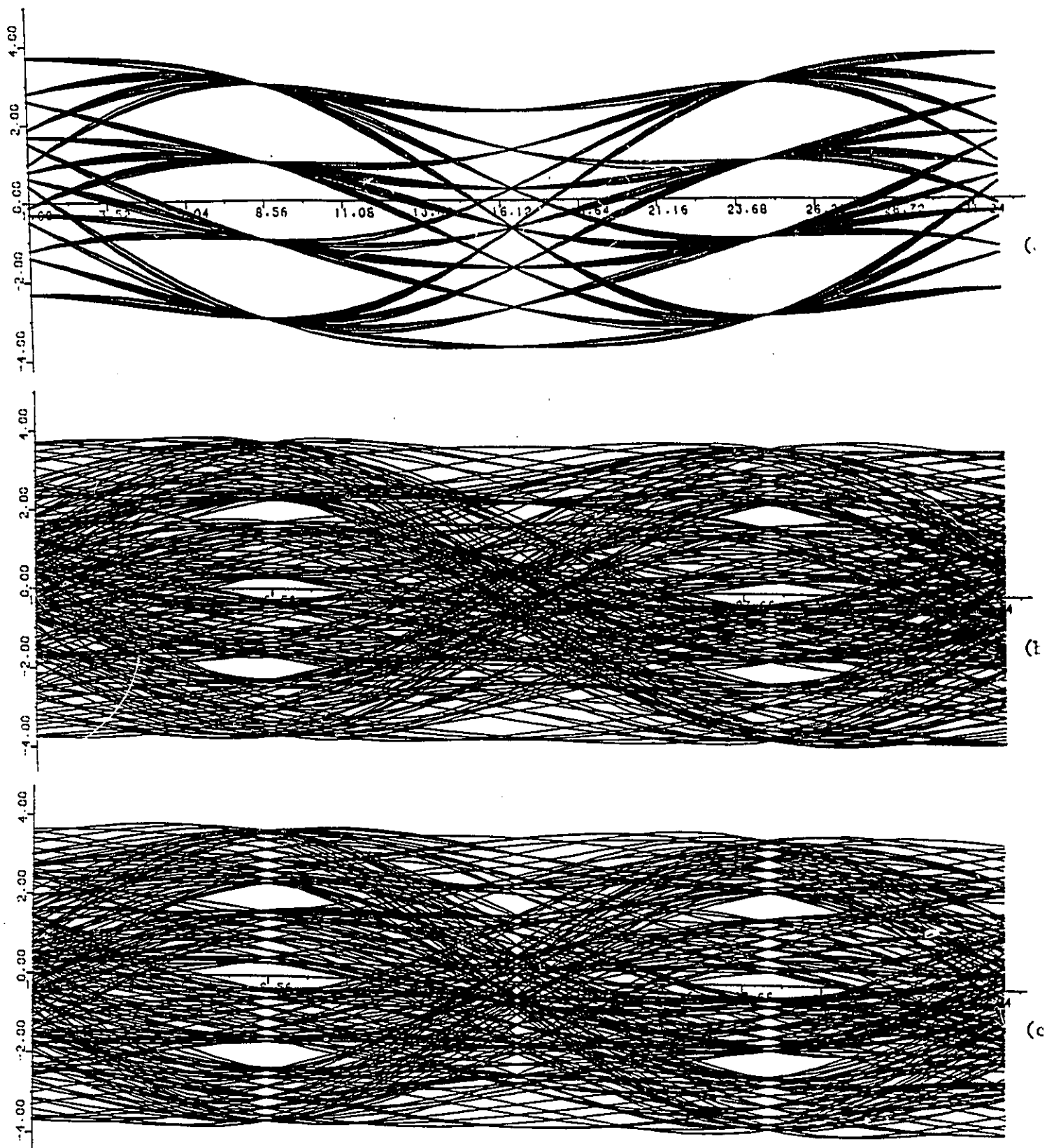


Figure 3.7: I-channel eye-diagrams of a 800 kb/s, 16-QAM employing $\alpha = 1.0$ raised-cosine filters with the pilot tone inserted at different in-band frequencies. In all cases the power of the pilot tone is 10 dB below that of the 16-QAM signal.

- (a) Pilot inserted at the Nyquist frequency ($f_p = 1/2T$).
- (b) Pilot inserted above the Nyquist frequency ($f_p = 1/1.9T$).
- (c) Pilot inserted below the Nyquist frequency ($f_p = 1/2.1T$).

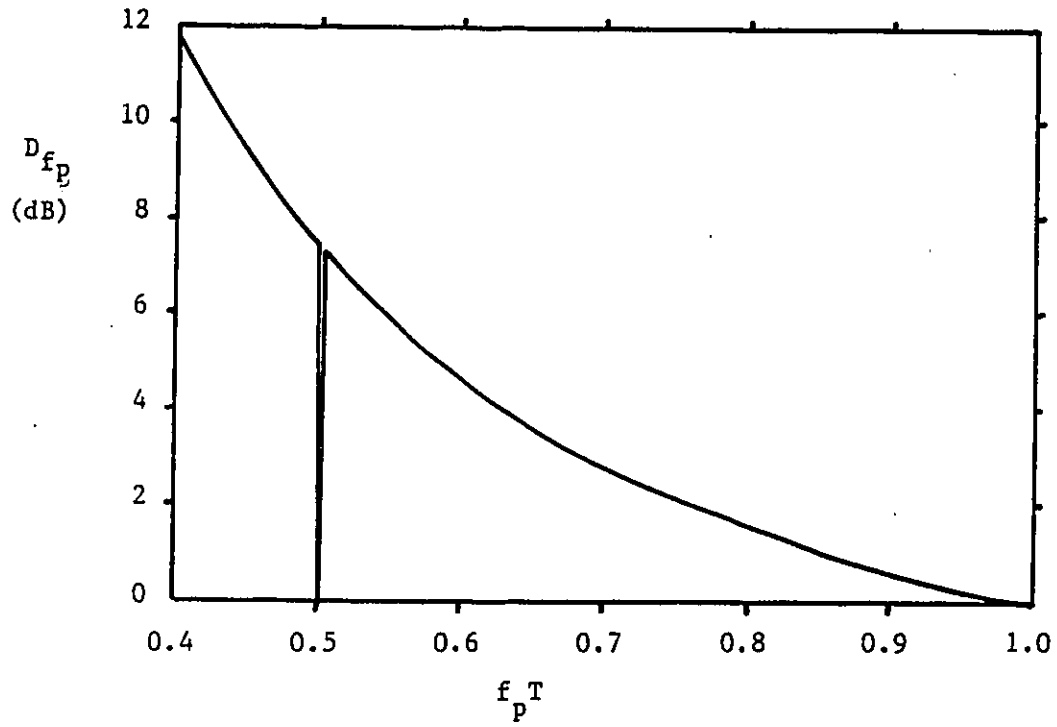


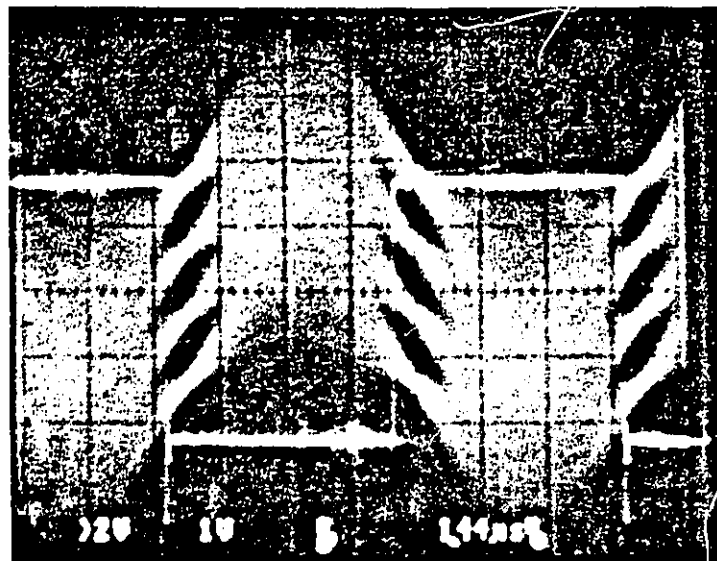
Figure 3.8: C/N degradation, D_{f_p} , of the 16-QAM IBDPS at a symbol error probability $P_S = 10^{-4}$ as a function of the inserted pilot frequency f_p , normalized to the symbol rate $1/T$ ($m = -10$ dB, $\alpha = 1.0$).

the pilot is inserted, the more D_{f_p} is increased. This can be also verified by comparing the maximal openings of the eye-diagrams in Figs. 3.7(b) and (c), where the eye opening of the former ($f_p \doteq 0.526/T$) is larger than the latter ($f_p \doteq 0.476/T$). Furthermore and as expected, there is no degradation for $f_p = 1/2T$ and $f_p = 1/T$. However, this last frequency is not an in-band frequency and thus should not be incorporated in our scheme.

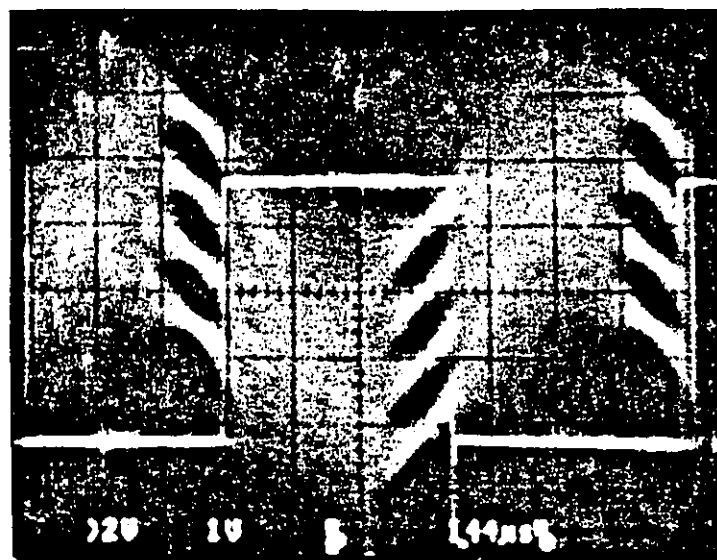
As previously mentioned (Section 3.2.2.1), one other major advantage of inserting the pilot at the Nyquist frequency is that it provides an easy means of recovering the symbol timing frequency at the receiver. This can be understood by observing that by inserting the pilots at the Nyquist frequency the signal $w(t)$ (see Fig. 3.1) contains in addition to the previously mentioned carrier information, a sinusoid with frequency the symbol rate $1/T$. Thus by simply employing a bandpass filter BPF_{f_s} which is centered around $f_s = 1/T$, we can recover the desired symbol timing signal. It should be noted that the bandwidth of this filter does not have to be narrow, since most of the data interference was rejected by the pilot extraction filters $PBPF_{\pm}$.

Measured, demodulated I-channel eye-diagrams of the prototype 16-QAM ($\alpha = 0.2$) IBDPS with the pilots inserted at the Nyquist frequency are illustrated in Fig. 3.9. It should be noted that the Nyquist frequency can be easily obtained by frequency dividing by two the symbol rate signal. Due to this frequency dependance, inserting the pilot at the Nyquist rate proved to be a robust technique to any symbol frequency changes. In part (a) of this figure, the pilot is inserted with the proper phase alignment and therefore, as it can be seen, all eyes are almost completely open.¹³ Furthermore, their level is constant over the two symbol interval. However, this is not the case in part (b) of the same figure, where although the pilot is inserted at the Nyquist frequency its incorrect phasing results in distorted (i.e. non-aligned) eye-diagrams. In both figures (a) and (b), the square wave signal is an arbitrary reference signal of frequency $2/T$. It should be mentioned that in order to demonstrate the implementation effec-

¹³As expected, there is a small amount of ISI in the middle of each of the eyes, which is mainly due to imperfect filtering of the 16-QAM system. The same amount of ISI was observed for the out-of-band pilot insertion, as is illustrated in Fig. 2.16.



(a)



(b)

Figure 3.9: Measured, demodulated eye-diagrams of the I-channel of a 16-QAM ($\alpha = 0.2$) IBDPS with the pilots inserted at the Nyquist frequency. The power allocated to the pilots is the same as that of the 16-QAM signal.
 (a) With proper phase alignment.
 (b) With incorrect phase alignment.

tiveness and accuracy of the pilot insertion at the Nyquist frequency, regardless of how strong the pilot signals are, for the eye-diagrams of Fig. 3.9 we have $m = 1.0$, i.e. both inserted pilots and the 16-QAM signal have the same power. The corresponding measured spectrum of the transmitted signals is illustrated in Fig. 3.10. It is important to mention that any other in-band pilot insertion frequency, that is even if it was very close to the Nyquist frequency, resulted in completely closed demodulated eyes.

The effects of the imperfect carrier, due to the data phase jitter, on the performance of the 16-QAM scheme were evaluated and the results are presented in Fig. 3.11.¹⁴ In this figure, the C/N degradation D_{IBDPS} (in dB) of the 16-QAM IBDPS at a symbol error probability $P_S = 10^{-4}$ as a function of ψ (see Eq. 3.25) for systems employing $\alpha = 0.2$ and $\alpha = 1.0$ raised-cosine filters are illustrated.

Clearly our results indicate that for $\psi \leq 10^{-3}$ the effects of the data interference into the recovered carrier and therefore on the 16-QAM system performance is negligible. Furthermore, we can see from the same figure that systems with large excess bandwidths (e.g. $\alpha = 1.0$) are less sensitive as compared to more bandwidth efficient systems (e.g. $\alpha = 0.2$) especially for applications where $\psi > 10^{-2}$. This can be easily understood by considering the analytical results derived for the variance of the carrier data phase jitter, σ_ϕ^2 , as they are illustrated in Fig. 3.4 and which indicate the sensitivity of systems with relatively low α . Finally our results suggest that for $\psi > 5 \cdot 10^{-2}$ the performance degradation of the 16-QAM scheme becomes unacceptably high.

In order to verify by measurements some of the results which were obtained by means of computer simulation, we employed the previously described prototype 16-QAM ($\alpha = 0.2$) IBDPS. As previously stated, the carrier frequency and the symbol rate of this prototype system are 1.5 MHz and 200 kBaud/s, respectively. Therefore, according to our computer simulated results shown in Fig. 3.11, in

¹⁴All the performance evaluation results which will be reported from now on assume that the pilots are inserted at the Nyquist frequency and their power level is 10 dB below the power allocated to the 16-QAM signal. Furthermore for comparison purposes, we will not include in these results the inherent to any DPS C/N degradation of 0.7 dB, as we did in Chapter 2.

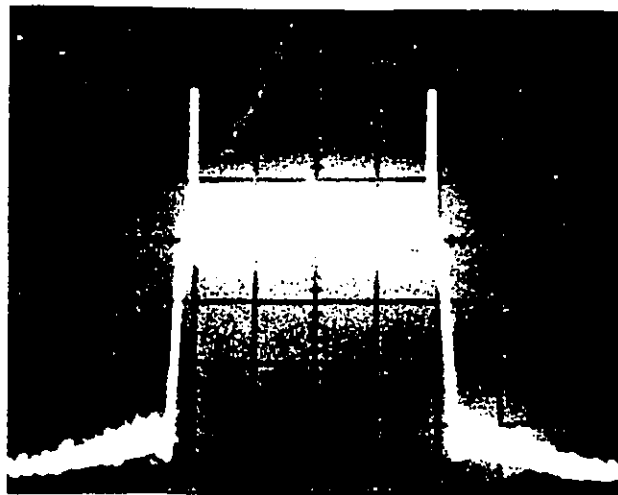


Figure 3.10: Measured spectrum of the transmitted 16-QAM ($\alpha = 0.2$) IBDPs with the pilots inserted at the Nyquist frequency. In this figure the power which is allocated to the pilots is the same as that of the 16-QAM signal, i.e. $m = 1.0$.

Horizontal Axis: 50 kHz/div.

Vertical Axis: 10 dB/div.

Center Frequency: 1.5 MHz.

Resolution Bandwidth: 5 kHz.

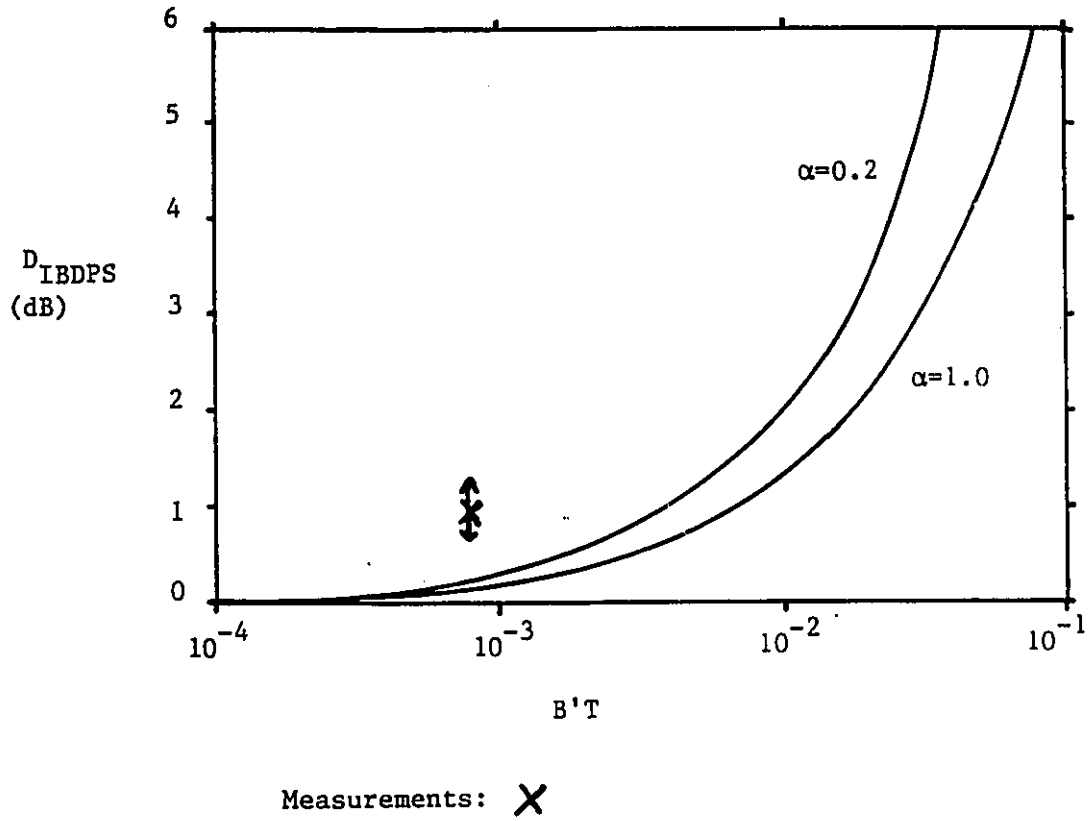


Figure 3.11: C/N degradation D_{IBDPS} of the 16-QAM IBDPS at $P_S = 10^{-4}$ as a function of ψ (product of the pilot extraction bandpass filter bandwidth B' multiplied by the symbol duration T) and for systems with $\alpha = 1.0$ and $\alpha = 0.2$.

order to achieve a reasonably good measured performance, we have to employ pilot extraction filters which have a $\psi < 10^{-2}$, or equivalently an extraction bandwidth B' of not more than 2 kHz. However, in our modem additional C/N degradation will be caused due to the fact that in practice the employed pilot extraction filters will not be ideal brickwall filters as assumed in the computer simulation model. Furthermore, these filters will not have identical bandpass frequency responses, as again was assumed in obtaining the results of Fig. 3.11. Therefore, in order to get meaningful performance measurements, we decided to use filters which have an extraction bandwidth B' of about 200 Hz.¹⁵ It is useful at this point to define Ψ as the ratio of B' over the bandpass filter center frequency. Clearly for the upper bandpass filter, $PBPF_+$, $\Psi_+ = 200 \text{ Hz}/1.6 \text{ MHz} = 1.25 \times 10^{-4}$ whereas for the lower one, $PBPF_-$, we have $\Psi_- = 200 \text{ Hz}/1.4 \text{ MHz} = 1.43 \times 10^{-4}$.

After considerable consultation with a number of filter manufacturing companies, we have decided that the best solution to achieve narrowband extraction bandwidths is to employ crystal filters. As well known, crystals because of their highly sensitive resonance frequency characteristics, provide a means of designing and implementing extremely narrowband filters without any major technical difficulties [69]-[74]. With today's technology crystal filters with very low Ψ can be implemented, over a wide variety of frequencies. Examples of designs include bandpass filter realizations with $\Psi = 10^{-6}$ with a center frequency within the 10 kHz - 200 MHz band [71]-[74].

We decided to order two crystal bandpass filters which were manufactured by K&L Quartztek [72], because of the fairly short delivery time (6 weeks) and their relatively low cost (\$ US 265.00). The obtained crystal filters had the following measured characteristics:

- **Center Frequency:** 1.4 MHz; **3 dB Bandwidth:** 238 Hz; **40 dB Bandwidth:** 400 Hz.
- **Center Frequency:** 1.6 MHz; **3 dB bandwidth:** 240 Hz; **40 dB bandwidth:** 450 Hz.

Thus, these extraction filters correspond to a ψ of approximately 10^{-3}

¹⁵It should be noted that B' represents for these practical filters their 3 dB bandwidth.

We measured the performance of our prototype 16-QAM IBDPS with the pilot tones inserted at the Nyquist rate and employing the crystal filters to extract the pilots and subsequently recover both the carrier and the symbol timing signal. Our measurements, which are included in Fig. 3.11, indicate that the modem degradation due to the data phase jitter at $P_S = 10^{-4}$ is approximately 1.0 ± 0.2 dB,¹⁶ as compared to the 0.3 dB degradation of the computer simulated system. We believe that this difference in performance is due to the nonideal and nonidentical transfer functions of the employed pilot extraction crystal bandpass filters.

3.4 CONCLUSIONS

In this chapter a novel synchronization scheme which employs an in-band at the Nyquist frequency pilot insertion method and enables simultaneous carrier and symbol timing recovery at the receiver, has been introduced and analyzed. The main advantage of the proposed scheme is that it does not require any excess bandwidth to accommodate the pilot tones as all other pilot based synchronization schemes do.

We have analyzed the effects of the mutual interference between the inserted pilots and a digitally transmitted QAM signal. Performance evaluation results in terms of C/N degradation having the pilot frequency and level as parameters were also presented. It was shown, that the only in-band pilot frequency, which does not cause any interference into the digital signal and at the same time provide both carrier and symbol timing information at the receiver by means of the pilot tones, is the Nyquist frequency.

As an example, a number of computer simulated as well as experimental results for 16-QAM systems employing the proposed in-band pilot insertion scheme were presented. They indicate that by employing pilot extraction bandpass filters which have bandwidth (normalized to the symbol rate) of 10^{-4} or less, the C/N degradation which results

¹⁶It should be noted that these measured results do not include the inherent to our modem degradation of 1 dB.

from the recovered carrier and symbol timing signals is practically negligible.

Chapter 4

512- AND 1024-QAM SYSTEM PERFORMANCE IN DISTORTED LINEAR CHANNELS

4.1 INTRODUCTION

In Chapter 1, after introducing the demand for highly bandwidth efficient modulation schemes, it was concluded that 512- and 1024-QAM are the most suitable candidates for applications requiring spectral efficiencies of up to 10 b/s/Hz. Essentially, 512- and 1024-QAM are equivalent to 24- and 32-level, respectively, PAM signals in baseband.

In general, the sensitivity of a system increases as the number of modulating states is increased [9]. For example, multilevel signals are more sensitive to various impairments because the same impairments (e.g. filter imperfections) yields higher ISI in multilevel systems than in a binary system [42],[43]. As a result, filter specifications which are suitable for a binary system do not necessarily guarantee a satisfactory performance of a 24- or 32-level system. It should be emphasized, that channel distortions can be caused also by residual errors of imperfect adaptive equalizers. Thus, it is important to investigate the

effects of channel distortions and/or hardware imperfections on both 512- and 1024-QAM systems [62],[107].

The objective of this chapter is to quantitatively evaluate the effects of amplitude and group delay distortions with linear, parabolic or sinusoidal characteristics on the performance of 512- and 1024-QAM systems. After this introduction, Section 4.2 will be devoted to a brief description of our simulation method. In Section 4.3 a number of simulated results for both of the QAM systems under consideration will be presented. Section 4.4 presents some heuristic explanations of the obtained performance evaluation results. In order to efficiently apply the obtained results to systems of different transmission rates, a simple scaling method for doing so will be described in Section 4.5. Finally, this chapter will be concluded in Section 4.6.

4.2 DESCRIPTION OF THE COMPUTER SIMULATED SYSTEMS

Figs. 4.1 and 4.2 illustrate a general block diagram of both 512- and 1024-QAM systems. For both schemes the modulated signal can be represented as:

$$s(t) = \text{Re}\left\{ \sum_{n=-\infty}^{\infty} (I_n + jQ_n)g_T(t - nT)e^{j2\pi f_c t} \right\} \quad (4.1)$$

where f_c is the carrier frequency, T is the symbol interval, $g_T(t)$ is the impulse response of the transmit filter $G_T(f)$, I_n and Q_n are independent and equiprobable discrete random variables which represent the transmitted data symbols.

For the 1024-QAM system I_n and Q_n take values from the alphabet $\{\pm 1, \pm 3, \dots, \pm 31\}$, which correspond to a square state space diagram (see Fig. 1.1). For the 512-QAM, among the different signal sets which can be used to generate it, the symmetric cross-like configuration shown in Fig. 4.3 was used (see also Fig. 1.1). This signal constellation consists of two 24-PAM signals, modulated in quadrature, with their 64 outermost stages deleted. The major reasons for

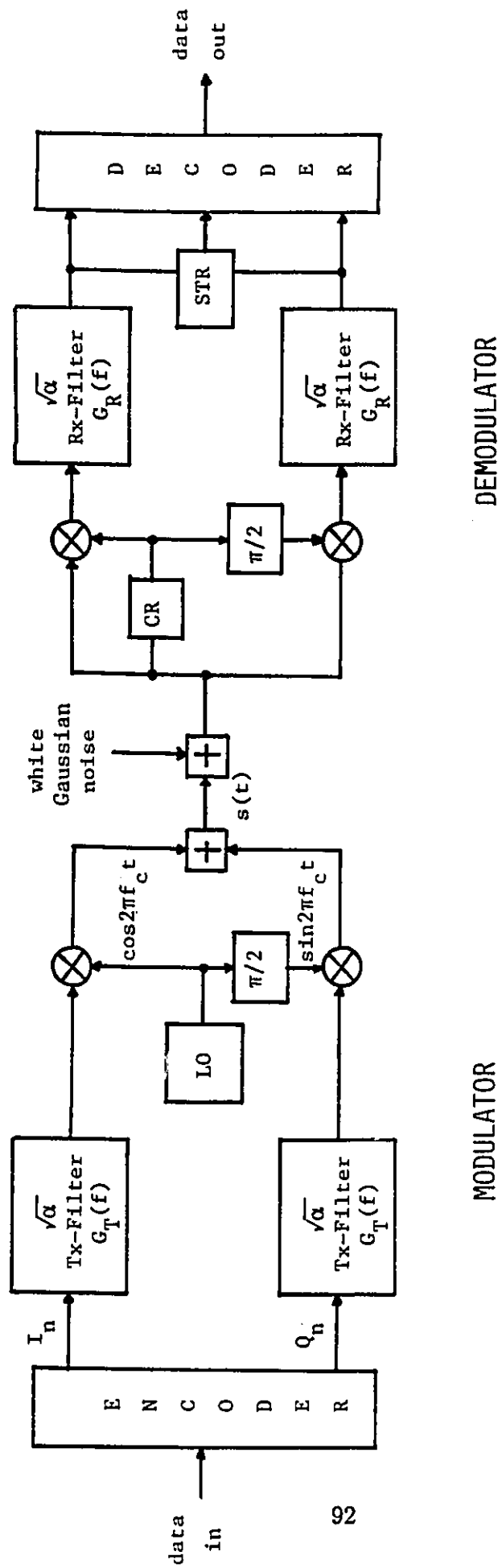
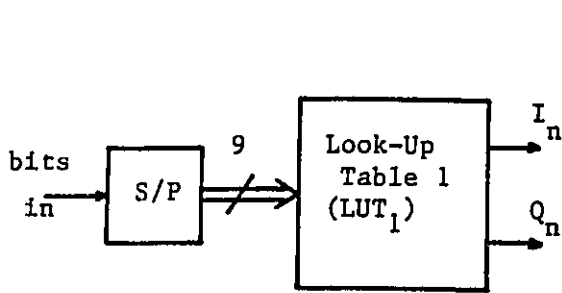
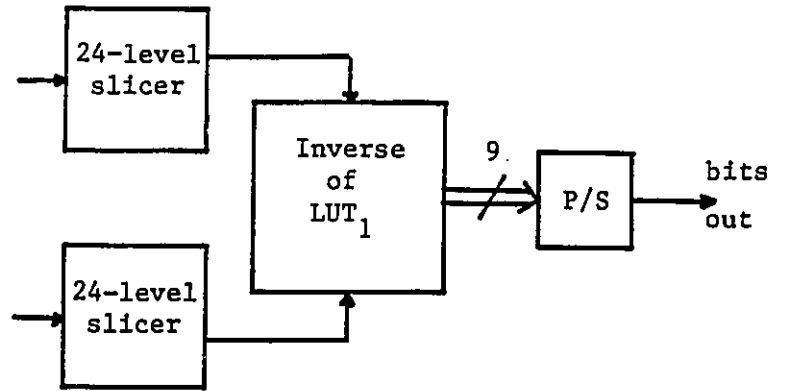


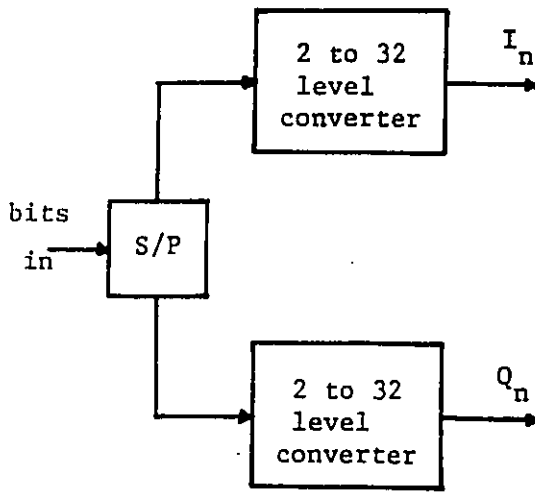
Figure 4.1: 512- and 1024-QAM system block diagram. Since the encoder and decoder for each of the two systems are somewhat different, they are illustrated separately in Fig. 4.2.
 (LO = Local Oscillator
 CR = Carrier Recovery
 STR = Symbol Timing Recovery)



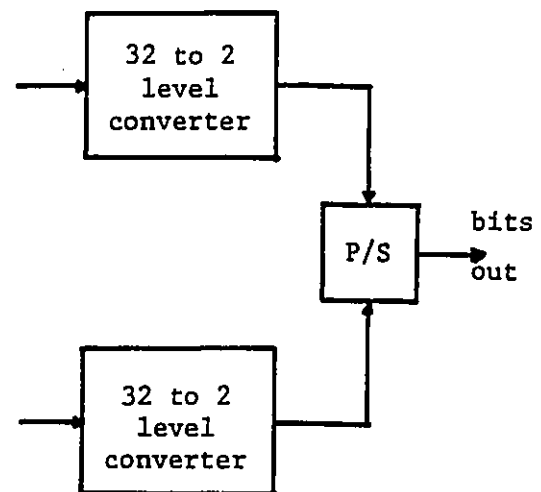
(a)



(b)



(c)



(d)

Figure 4.2: Encoder and decoder structures for the QAM schemes under consideration.

(a) 512-QAM Encoder. (b) 512-QAM Decoder.

(c) 1024-QAM Encoder. (d) 1024-QAM Decoder.

(S/P = Serial-to-Parallel Converter

P/S = Parallel-to-Serial Converter)

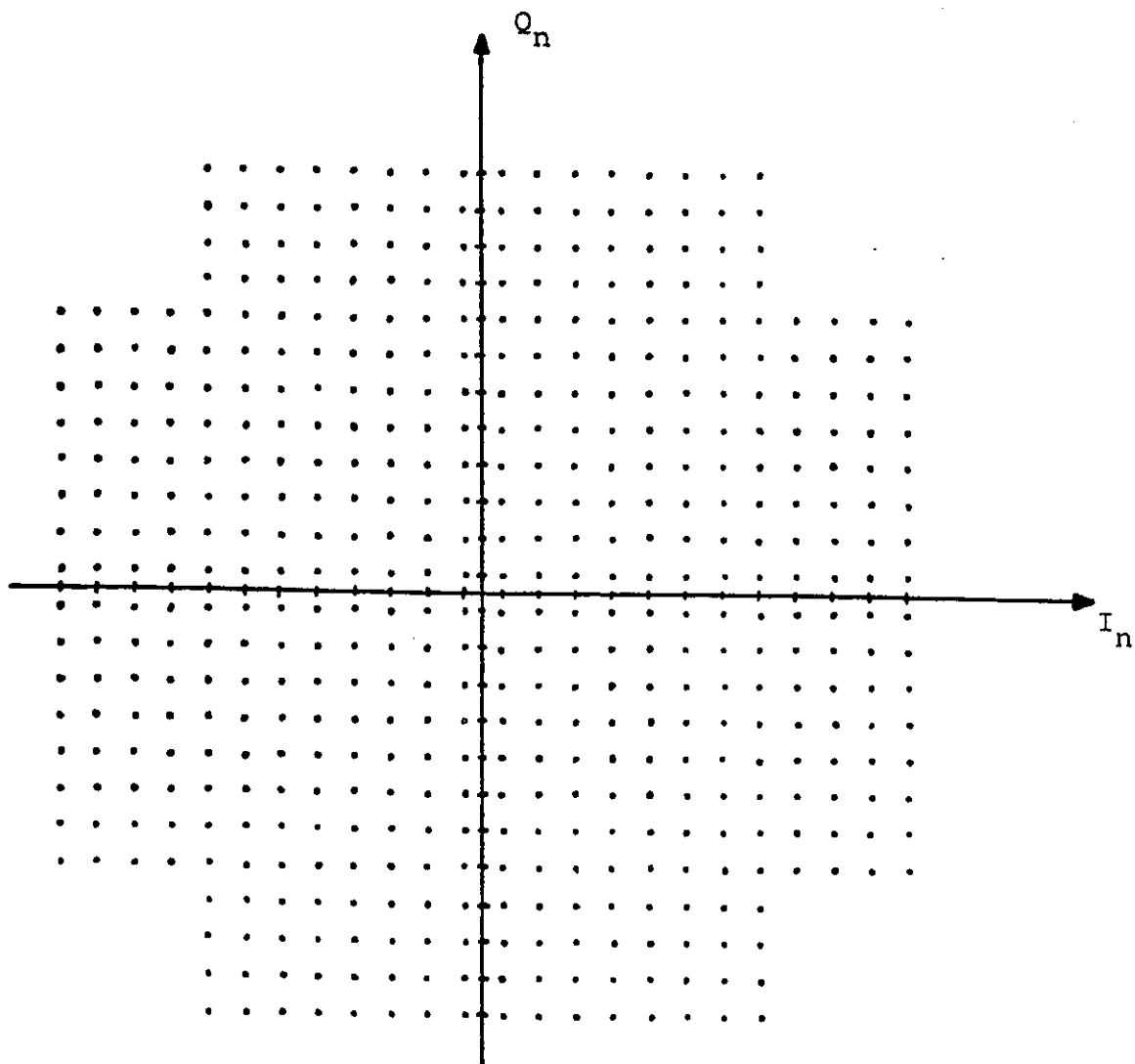


Figure 4.3: State-space diagram of a 512-QAM system.

choosing this configuration are near optimal performance in a linear, additive Gaussian noise channel, fairly easy hardware implementation and reduced peak amplitude compared to 24x24, i.e. 576-QAM [75]. For convenience, the levels of a 512-QAM signal can be defined as:

$$I_n = \pm 1, \pm 3, \dots, \pm 23$$

$$Q_n = \pm 1, \pm 3, \dots, \pm 23$$

where $I_n \neq Q_n$ for $\pm 17, \pm 19, \pm 21, \pm 23$. The simulation model used is illustrated in Fig. 4.4. All computations are carried out entirely in the complex baseband form. An ideal carrier recovery as well as symbol timing recovery is therefore assumed. It should be mentioned that, in principle, the same computer simulation method will be employed in evaluating both 512- and 1024-QAM systems, as the one described in Section 3.3.1. However, for the reader's convenience we will briefly describe it.

We are transmitting $(2^{19}-1)$ symbols for each of the I- and Q-channel with each symbol represented by 64 samples. The symbol rate $f_s = 1/T$ is chosen to be 15 MBaud/s, so that an easy comparison could be made with the 15MBaud/s 256-QAM system reported in [76]. Clearly, this Baud rate represents a 150 Mb/s rate for a 1024-QAM system whereas for a 512-QAM system it represents a 135 Mb/s rate signal. It is important to keep in mind that the obtained results can be applied to other bit rates by appropriate scaling as will be shown later on in this chapter. For filtering we employed FFT techniques to alternate between the frequency and the time domain. The transmit, $G_T(f)$, and receive, $G_R(f)$, filters are assumed to be ideal square root of α raised-cosine filters with an $(x)/\sin(x)$ amplitude equalizer in the transmitter, so that the combined filtering response satisfies Nyquist's I criterion [14, Chapter 7]. The eye-diagrams of this ideal case for systems employing $\alpha = 0.2$ filters are given in Figs. 4.5 (for 512-QAM) and 4.6 (for 1024-QAM). From both of these figures it can be observed that at the ideal sampling instants there is no ISI. However, data transition jitter is very significant particularly for the 1024-QAM system. This indicates that a high-precision sampling clock would be required. After the $G_R(f)$ filters, both I- and Q-channel signals are sampled and passed to the threshold comparators at the receiver. For

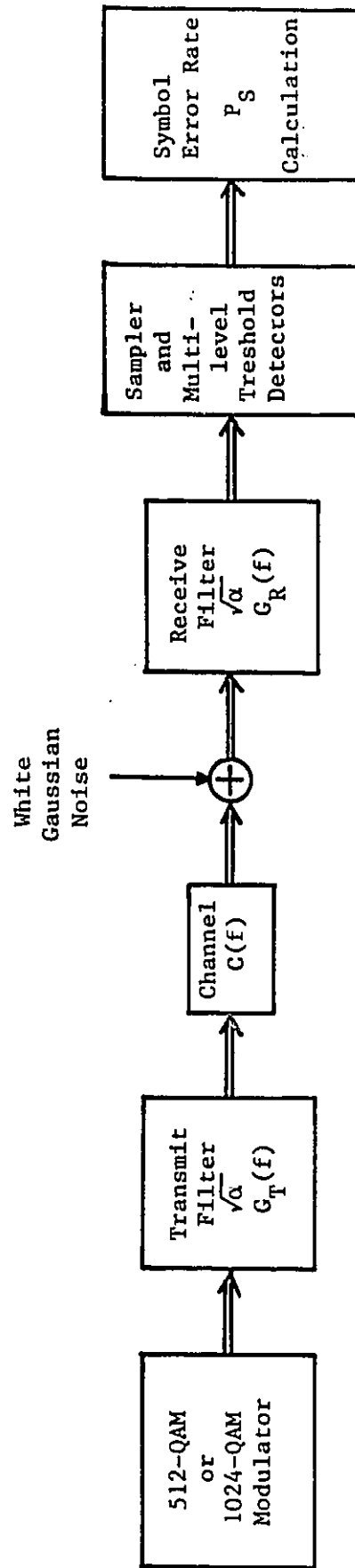


Figure 4.4: Block diagram of the computer simulation model which was employed to simulate the impact of the linear channel distortions on the performance of 512- and 1024-QAM systems.

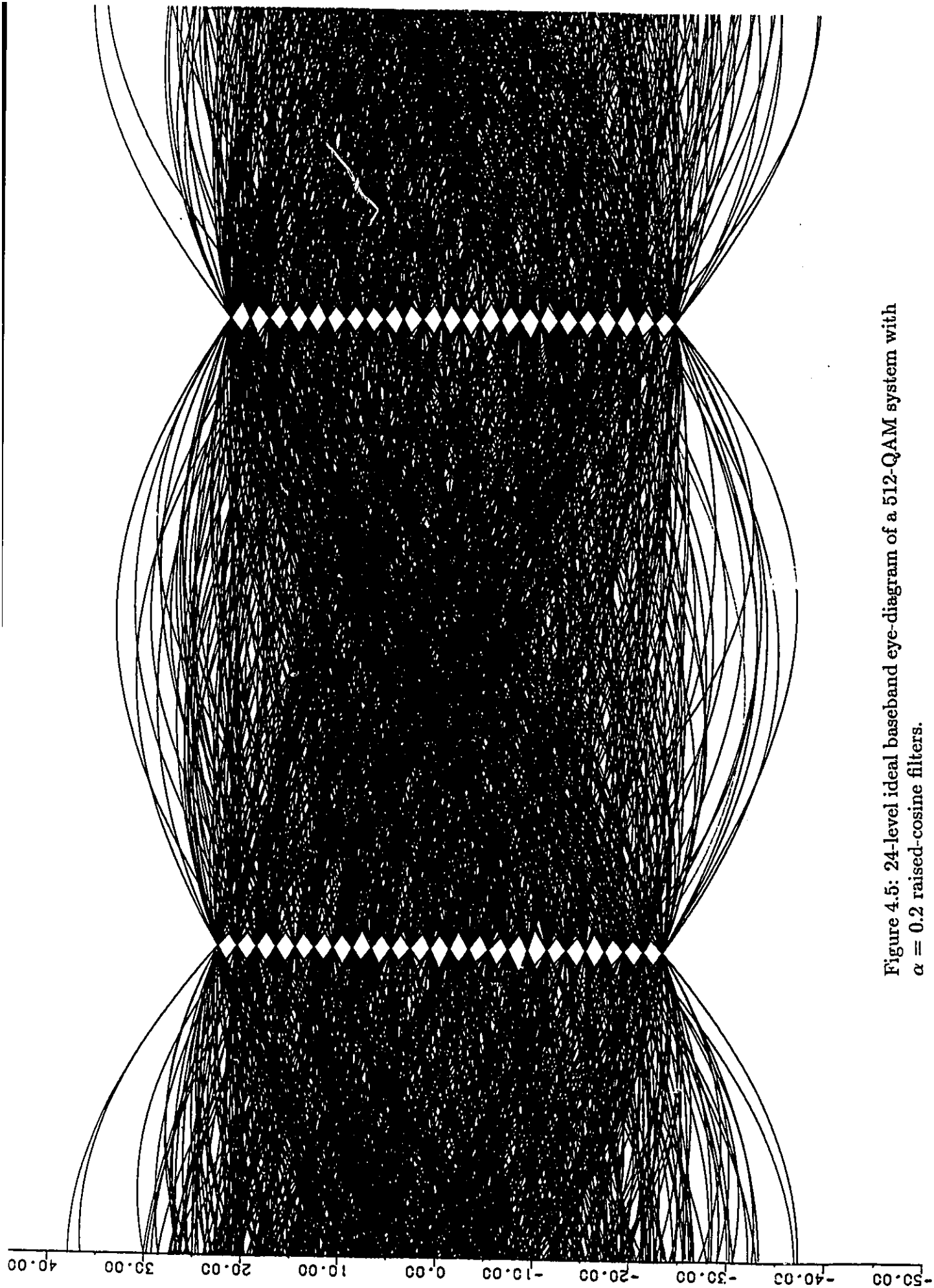


Figure 4.5: 24-level ideal baseband eye-diagram of a 512-QAM system with $\alpha = 0.2$ raised-cosine filters.

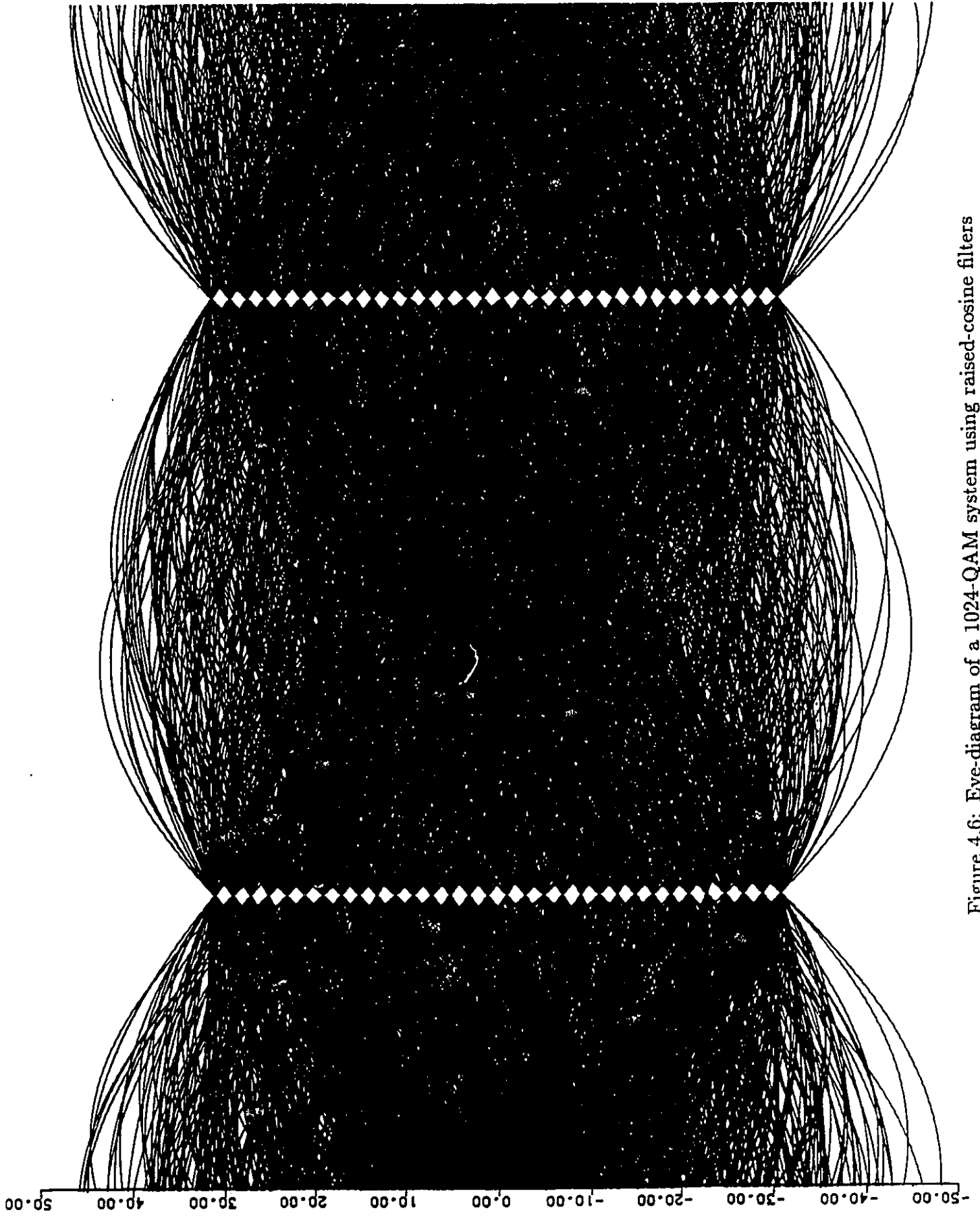


Figure 4.6: Eye-diagram of a 1024-QAM system using raised-cosine filters with a roll-off factor of $\alpha = 0.2$.

a specified power of white Gaussian noise at the threshold detector input, the error probability of the n -th symbol with reference to the I-channel is calculated as follows:

$$P_n^I = \begin{cases} \frac{1}{2} \text{erfc}\left(\frac{|I_n^s - \text{THR1}_n|}{\sqrt{2}\sigma}\right), & \text{for } I_n = I_n^1 \\ \frac{1}{2} \text{erfc}\left(\frac{|I_n^s - \text{THR1}_n|}{\sqrt{2}\sigma}\right) + \frac{1}{2} \text{erfc}\left(\frac{|I_n^s - \text{THR2}_n|}{\sqrt{2}\sigma}\right), & \text{for } I_n = I_n^2 \end{cases} \quad (4.2)$$

where I_n represents the n -th transmitted symbol at the I-channel, I_n^s its equivalent received sampled signal and THR1_n , THR2_n , $\text{erfc}(\cdot)$ and σ are defined as in Eq. 3.31. In addition to that I_n^1 and I_n^2 are given by:

$$I_n^1 = \begin{cases} \pm 31 & \text{for 1024-QAM} \\ \pm 23 & \text{for 512-QAM} \end{cases}$$

and

$$I_n^2 = \begin{cases} \pm 1, \pm 3, \dots, \pm 29 & \text{for 1024-QAM} \\ \pm 1, \pm 3, \dots, \pm 21 & \text{for 512-QAM.} \end{cases}$$

The error probability P_n^Q with reference to the quadrature channel is obtained similarly using an equivalent to the above expression. Finally, for a sequence of N symbols in each of the I- and Q-channel, the average symbol error probability is calculated as in Eq. 3.30.

In order to investigate the impact of illustrative different channel imperfections on the systems performances, we introduce amplitude and group delay distortions with linear, parabolic and sinusoidal characteristics in the channel transfer function $C(f)$ (see Fig. 4.4), which can be expressed as:

$$C(f) = |C(f)| e^{j\theta(f)}. \quad (4.3)$$

Clearly, the equivalent amplitude and group delay distortions can be represented by $A(f)$ and $D(f)$, where:

$$A(f) = |C(f)| \quad (4.4)$$

and

$$D(f) = -\frac{1}{2\pi} \frac{d\theta(f)}{df}. \quad (4.5)$$

In the next section, we will evaluate the effects of both of these distortions on the performance of the QAM systems under consideration.

4.3 PERFORMANCE EVALUATION WITH CHANNEL IMPERFECTIONS

4.3.1 Amplitude Distortions

To investigate the performance degradation due to amplitude distortions we assume that the phase distortions are equalized, i.e. $D(f)=B$, with B being a constant. The amplitude distortions are defined as follows:

$$A(f) = \begin{cases} L_A f & \text{for linear amplitude distortion} \\ P_A f^2 & \text{for parabolic amplitude distortion} \\ S_A \sin(2\pi K f / 2f_{BW}) & \text{for sinusoidal amplitude distortion} \end{cases} \quad (4.6)$$

where K , chosen to be equal to four, is the number of ripples of the sinusoidal distortion in the passband, f_{BW} is the system's baseband bandwidth and L_A , P_A and S_A are parameters characterizing the amount of amplitude distortion. Clearly:

$$f_{BW} = (1 + \alpha)f_N \quad (4.7)$$

with f_N being the Nyquist frequency which is given by:

$$f_N = f_s/2. \quad (4.8)$$

As illustrative examples we present in Figs. 4.7 and 4.8 eye-diagrams of both 512- and 1024-QAM systems operated in the presence of amplitude distortion. The computed symbol error probabilities for the 512-QAM system for different parameter values are given in Figs. 4.9-4.11. Its equivalent C/N degradation at $P_S = 10^{-4}$ for the three amplitude imperfection characteristics against the maximum amplitude distortion A_m , defined as:

$$A_m(dB) = \begin{cases} L_A(2f_{BW}) & \text{for linear amplitude distortion} \\ P_A(f_{BW}^2) & \text{for parabolic amplitude distortion} \\ S_A & \text{for sinusoidal amplitude distortion} \end{cases} \quad (4.9)$$

are given in Fig. 4.12. In the same figure, the degradations of an

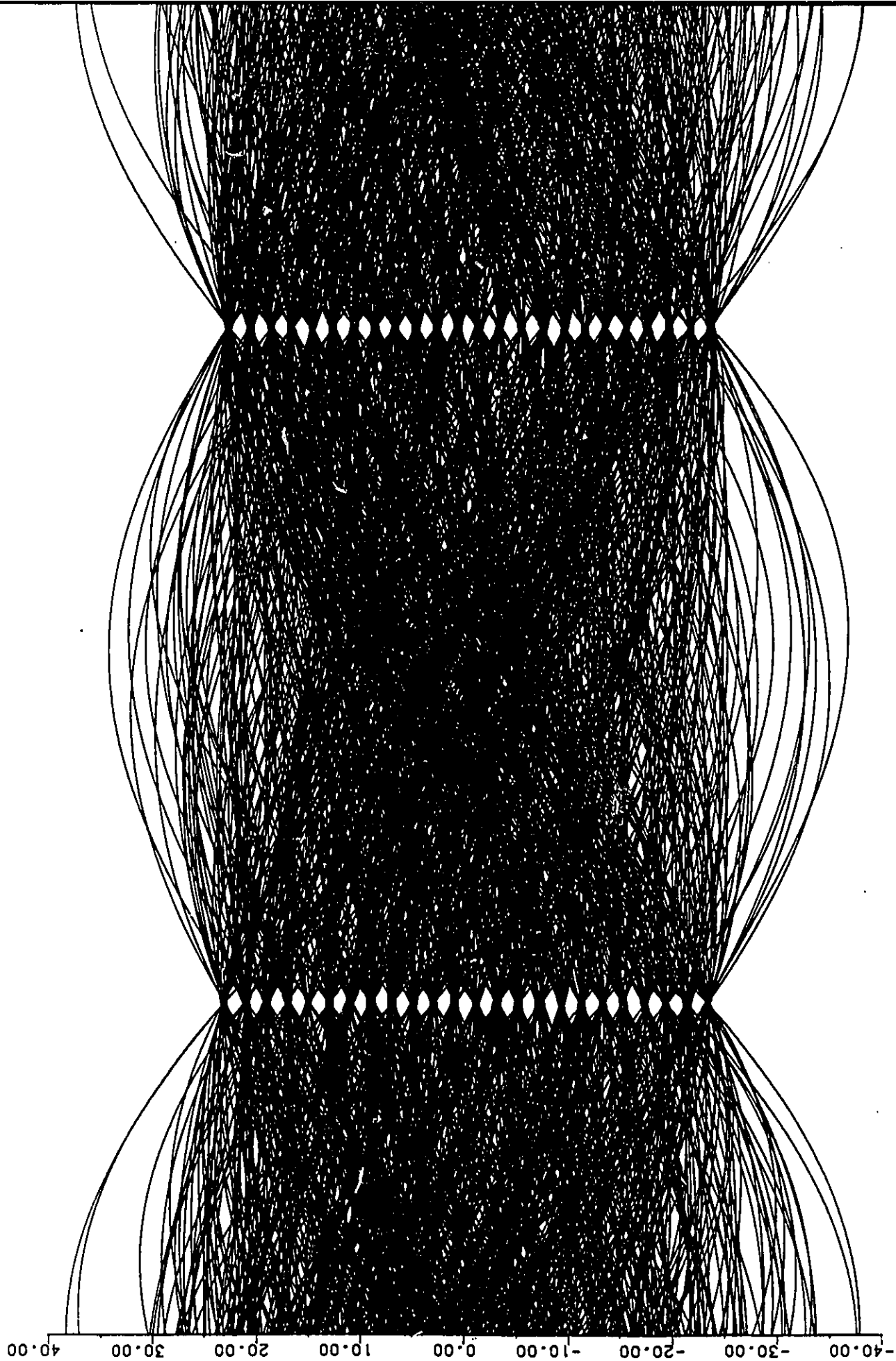


Figure 4.7: Eye-diagram of a 135 Mb/s, 512-QAM using $\alpha = 0.2$ raised-cosine filters and a channel having parabolic amplitude distortion of $P_A = 0.004 \text{ dB}/(\text{MHz})^2$.

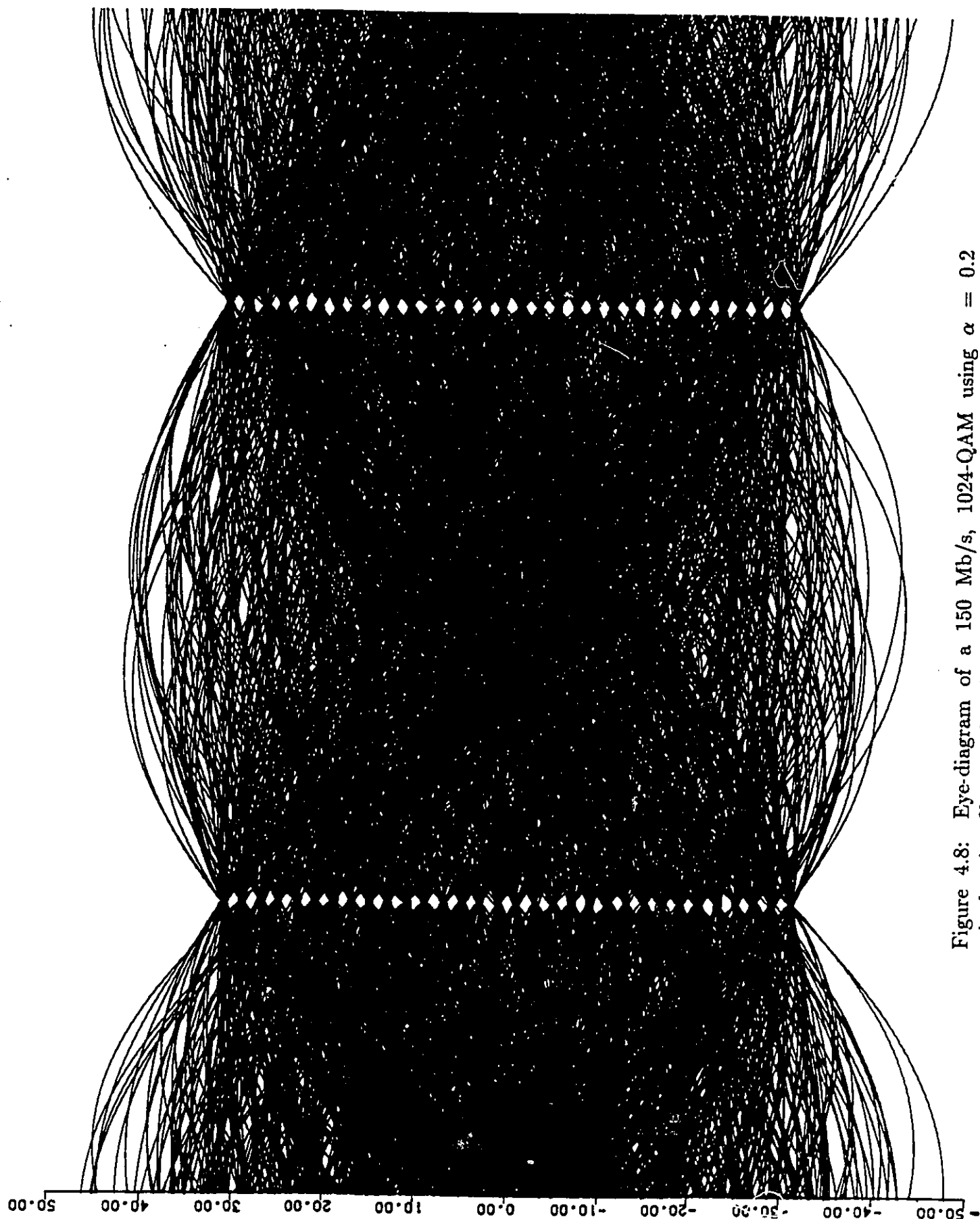


Figure 4.8: Eye-diagram of a 150 Mb/s, 1024-QAM using $\alpha = 0.2$ raised-cosine filters and a channel having parabolic amplitude distortion of $P_A = 0.0025 \text{ dB}/(\text{MHz})^2$.

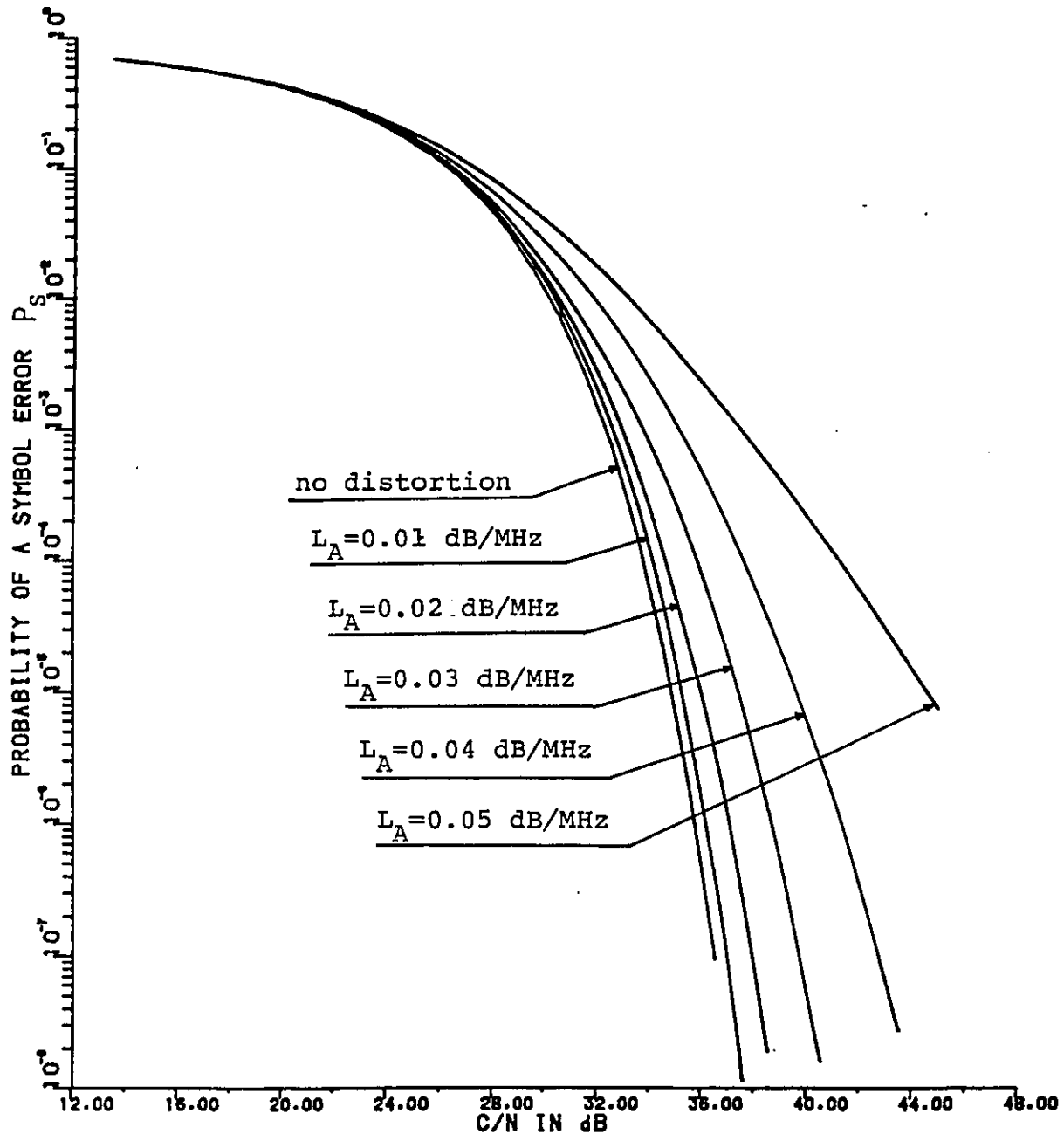


Figure 4.9: Impact of different linear amplitude distortion coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s (135 Mb/s) 512-QAM system. Filters with $\alpha = 0.2$ are used. The noise bandwidth is defined as the double-sided Nyquist bandwidth.

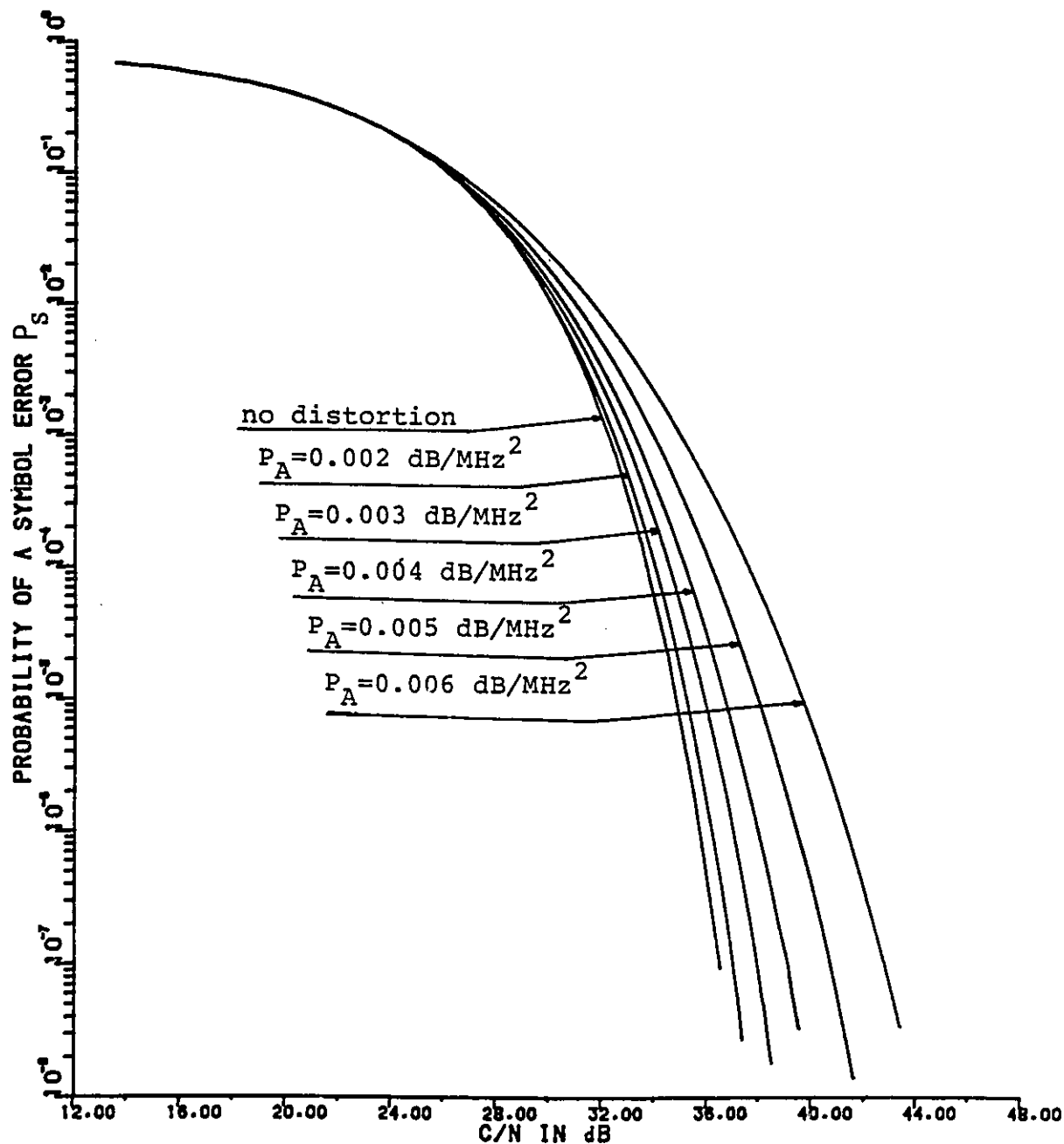


Figure 4.10: Impact of different parabolic amplitude distortion coefficients on the $P_s = f(C/N)$ performance of a 15 MBaud/s (135 Mb/s) 512-QAM system. Filters with $\alpha = 0.2$ are used. The noise bandwidth is defined as the double-sided Nyquist bandwidth.

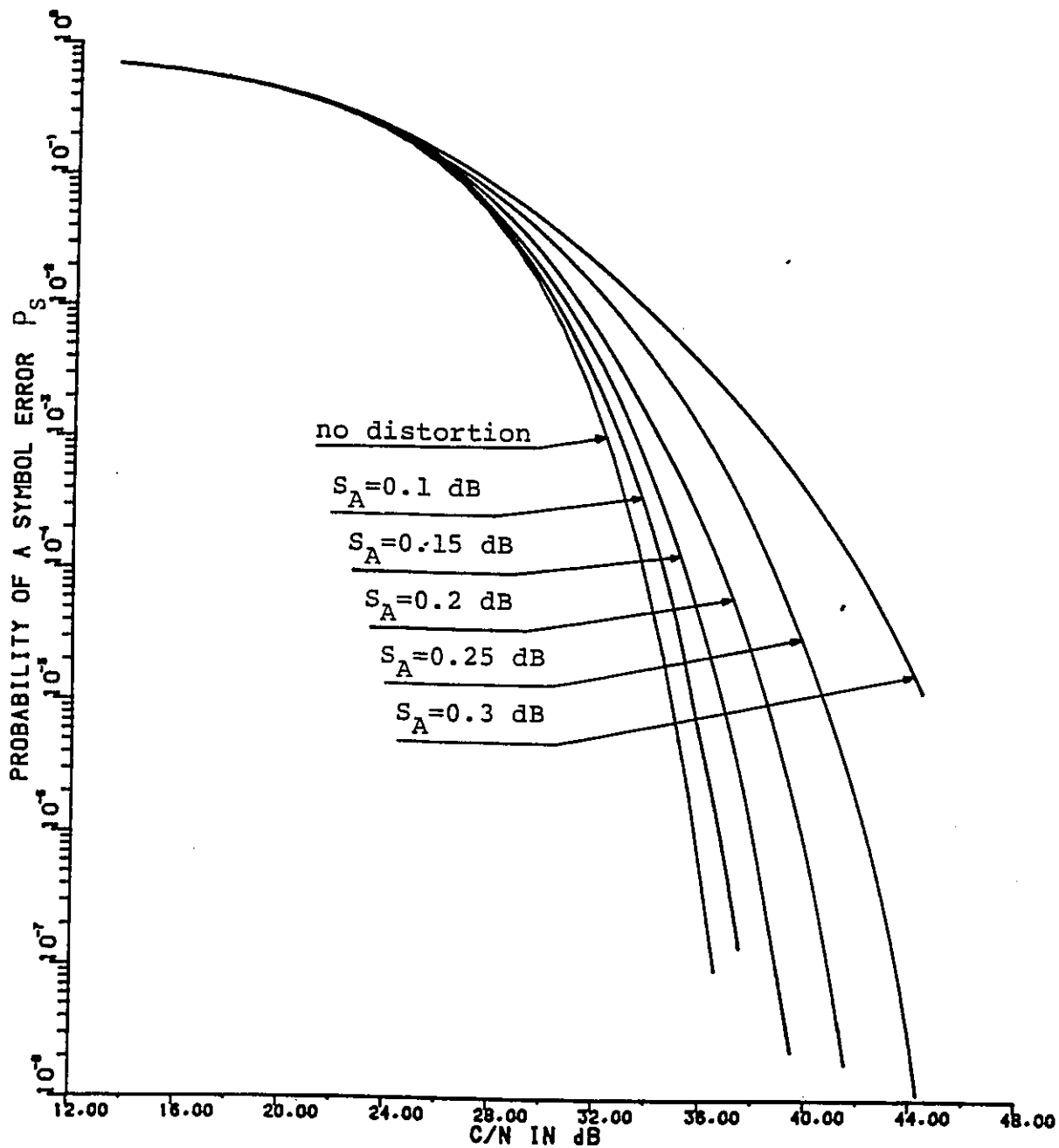


Figure 4.11: Impact of different sinusoidal amplitude distortion coefficients on the $P_s = f(C/N)$ performance of a 15 MBaud/s (135 Mb/s) 512-QAM system. Filters with $\alpha = 0.2$ are used. The noise bandwidth is defined as the double-sided Nyquist bandwidth.

equivalent (i.e. with the same symbol rate and roll-off factor α) 256-QAM system are included. The sensitivity of the 512-QAM system compared to the 256-QAM, especially for higher distortion values A_m , can be clearly seen. The comparison here is based on equal symbol rate and equal filtering strategy. Furthermore, similar to the 256-QAM modem study [76], for a given value of A_m , the sinusoidal amplitude distortion causes the worst degradation, whereas the linear distortion the least. Using exactly the same definitions for the amplitude distortions and by following the same procedure as for the 512-QAM, we obtained C/N degradation results for an equivalent (15 MBaud/s and $\alpha = 0.2$) 1024-QAM system. These results are illustrated in Fig. 4.13 where for comparison purposes we include also the previously obtained results for the 512-QAM. From this figure it is clear, as expected, that the 1024-QAM system is more sensitive as compared to the 512-QAM system. Further, we can also observe that for a given A_m and roll-off factor α , the sinusoidal amplitude distortion causes the most severe degradation of the system's performance followed in order of decreasing degradation by the parabolic and linear amplitude distortions. A physical interpretation of the obtained results is given in Section 4.4.

4.3.2 Group Delay Distortions

We now assume that the amplitude distortions are equalized, i.e. $A(f) = 1$. Again, three different characteristics for the group delay distortions are defined as follows:

$$D(f) = \begin{cases} L_D f & \text{for linear group delay distortion} \\ P_D f^2 & \text{for parabolic group delay distortion} \\ S_D \sin(2\pi K f / 2f_{BW}) & \text{for sinusoidal group delay distortion} \end{cases} \quad (4.10)$$

where L_D , P_D and S_D are parameters characterizing the amount of group delay, and the rest of the used symbols are the same as in Eq. 4.5.

We compute again first for the 512-QAM system the probability of a symbol error for the three characteristics and for different values of

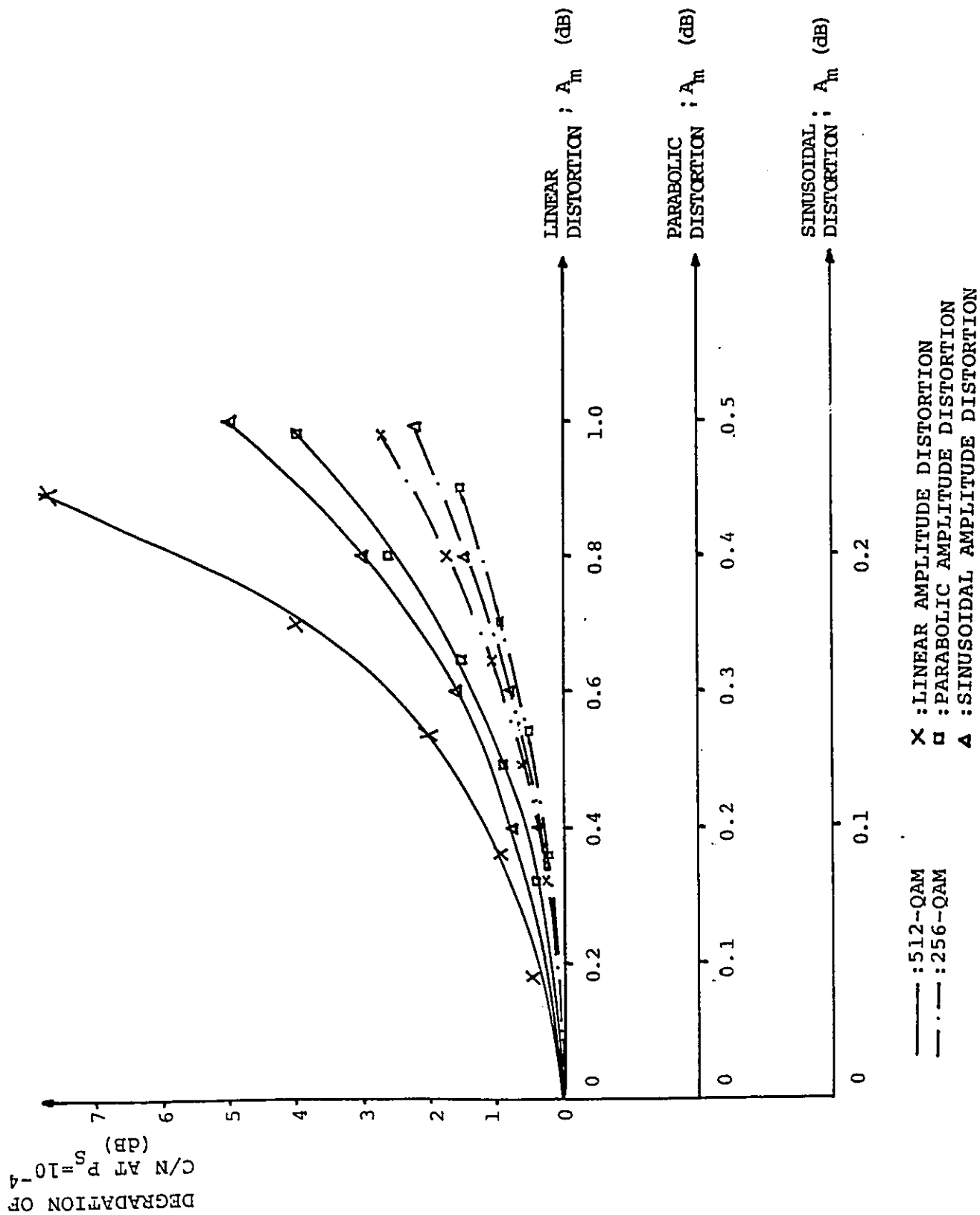


Figure 4.12: Degradation of C/N as a function of maximal amplitude distortion A_m for 512- and 256-QAM for linear, parabolic and sinusoidal amplitude distortions. Both systems have the same symbol rate of 15 MBaud/s and are raised-cosine filtered with roll-off factor of $\alpha = 0.2$.

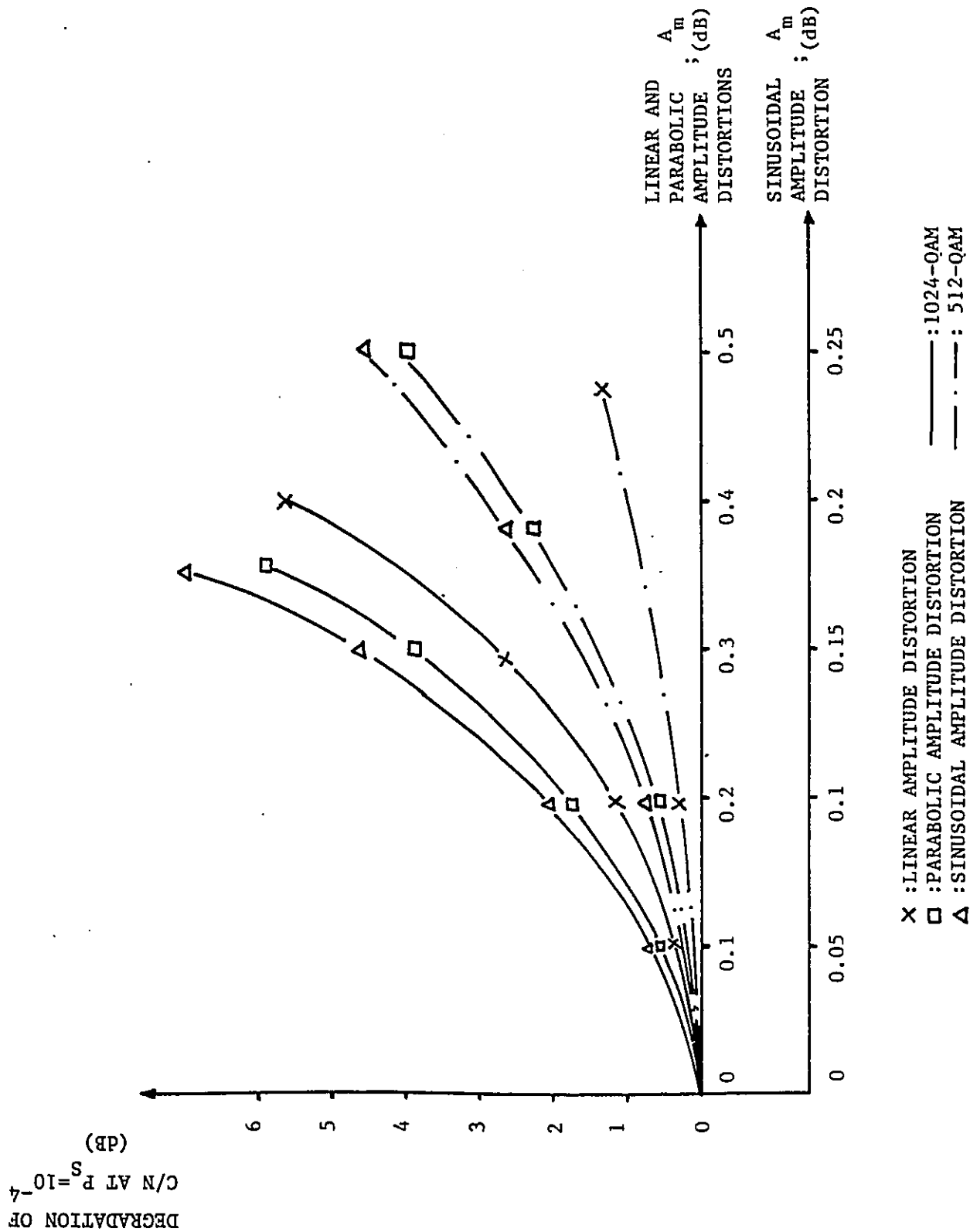


Figure 4.13: Impact of linear, parabolic and sinusoidal amplitude distortions on the P_s performance degradation as a function of A_m of the 1024-QAM system which employs $\alpha = 0.2$ raised-cosine filters and a symbol rate of 15 MBaud/s. For comparison purposes, we include the performance of an equivalent 512-QAM system (see Fig. 4.12).

the distortions. The results are given in Figs. 4.14-4.16. Defining the maximum group delay τ_m as:

$$\tau_m(\text{ns}) = \begin{cases} L_D(2f_{BW}) & \text{for linear group delay distortion} \\ P_D(f_{BW}^2) & \text{for parabolic group delay distortion} \\ S_D & \text{for sinusoidal group delay distortion} \end{cases} \quad (4.11)$$

we present the C/N degradation plotted against τ_m in Fig. 4.17. For comparison purposes, in the same figure the degradation of an equivalent 256-QAM is also illustrated. We note that for both systems, for a given value of τ_m , the linear group delay distortion causes the most severe degradation to the system performance, followed by sinusoidal and parabolic group delay distortions.

Similar to the amplitude distortion case, we obtained performance evaluation results for a 1024-QAM, 15 MBaud/s system which employs raised-cosine filters with $\alpha = 0.2$. The calculated C/N degradation at $P_S = 10^{-4}$ as a function of the maximal group delay τ_m for all three distortion types are presented in Fig. 4.18. In the same figure we have also included the previously obtained results for the equivalent 512-QAM system. Clearly, 1024-QAM is more sensitive to all types of distortion. As far as which distortion causes more degradation the trend of these results are the same as that of Fig. 4.17. Some heuristic explanations of the obtained results are given in the next section.

4.4 PHYSICAL INTERPRETATION OF THE OBTAINED RESULTS

Having obtained a great deal of performance evaluation results by computer simulation, it is of importance to seek the physical reasons of why certain distortion characteristics cause more degradation to the system performance. It should be mentioned that, due to the complexity of the problem, our scope here is not to obtain a rigorous method of analyzing the effects of the different distortions on the QAM schemes, but rather to present some heuristic explanations for

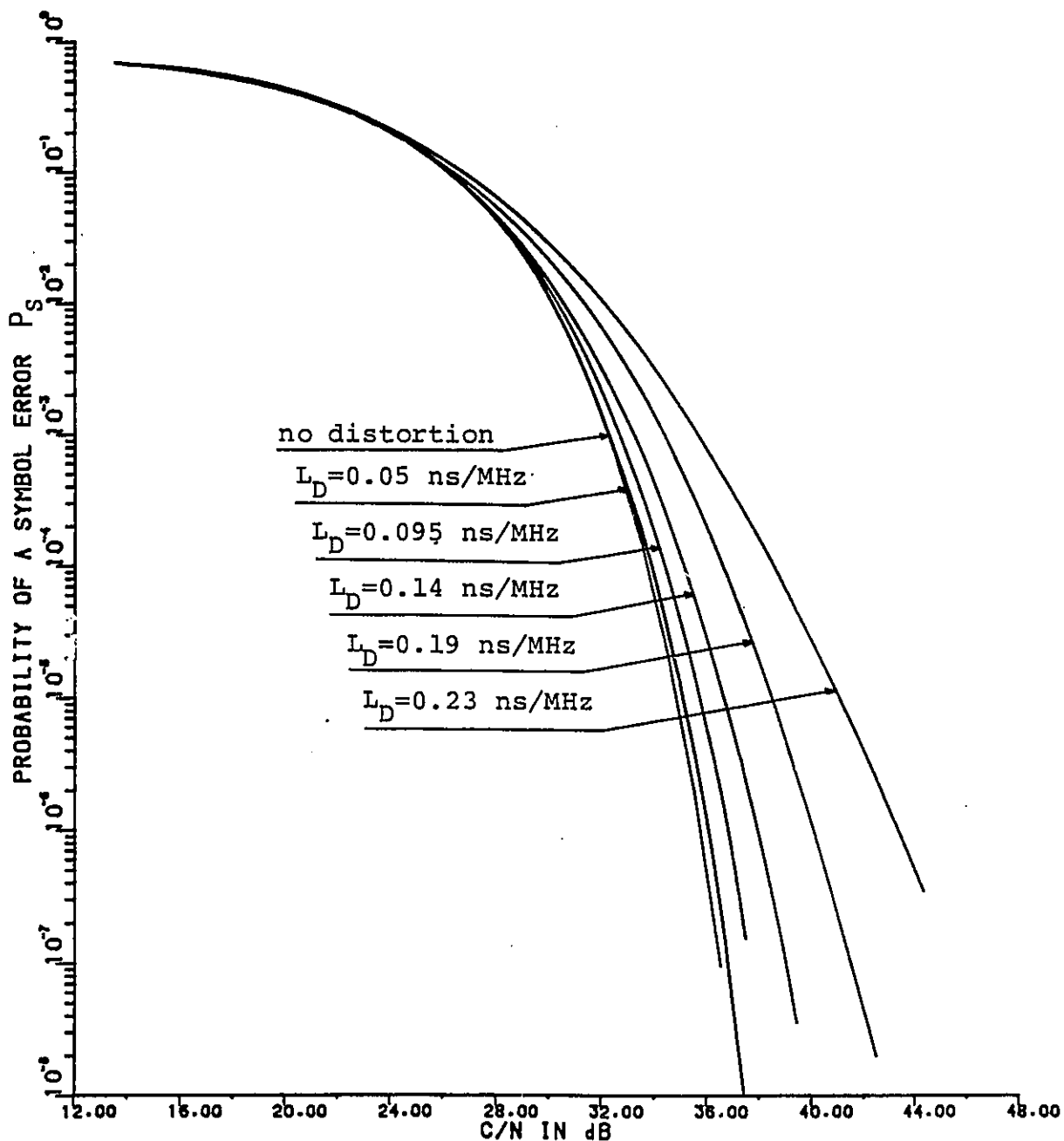


Figure 4.14: Impact of different linear group delay distortions coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s 512-QAM system. Filters with $\alpha = 0.2$ are used. The noise bandwidth is defined as the double-sided Nyquist bandwidth.

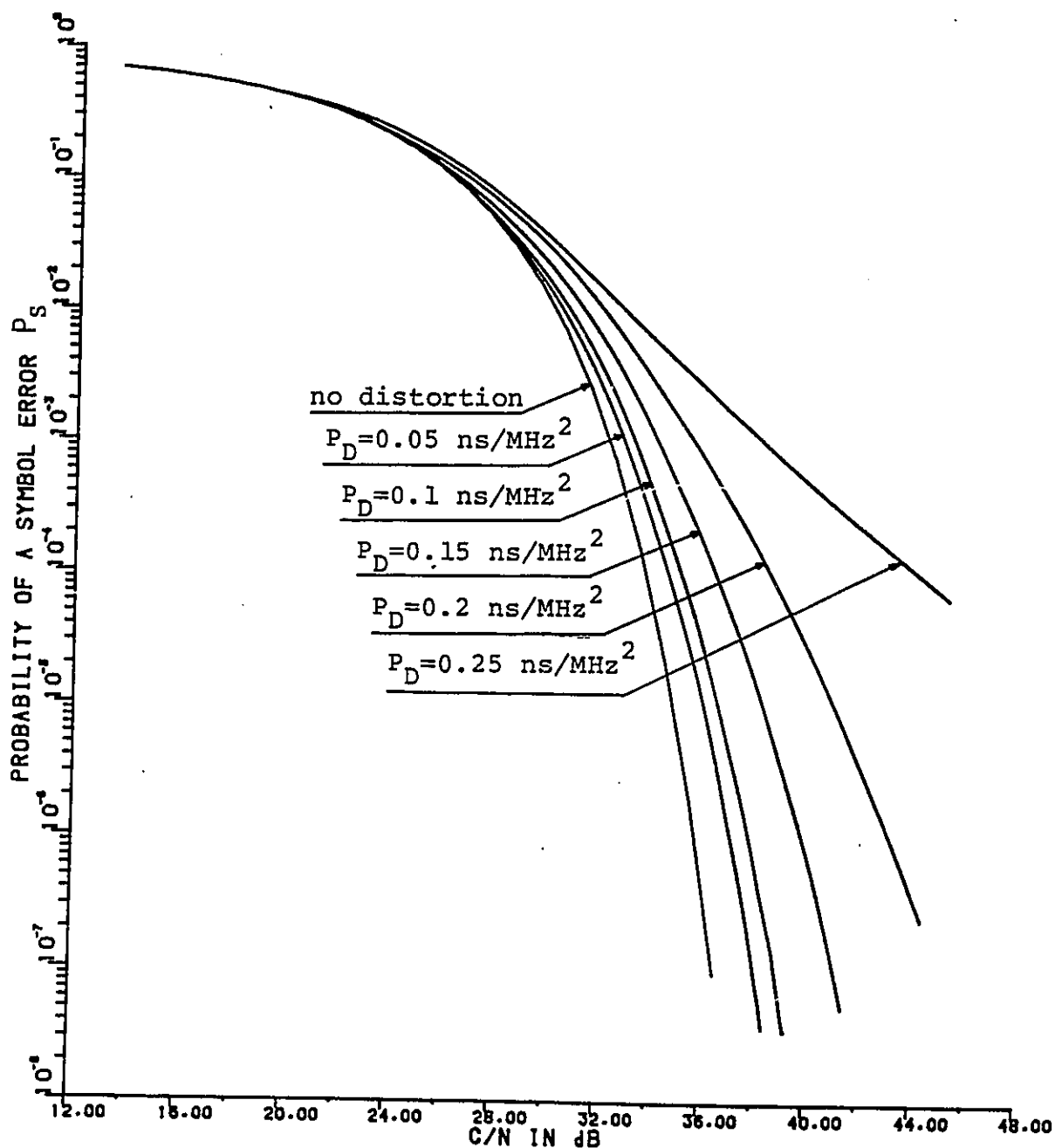


Figure 4.15: Impact of different parabolic group delay distortion coefficients on the $P_s = f(C/N)$ performance of a 15 MBaud/s 512-QAM system. Filters with $\alpha = 0.2$ are used. The noise bandwidth is defined as the double-sided Nyquist bandwidth.

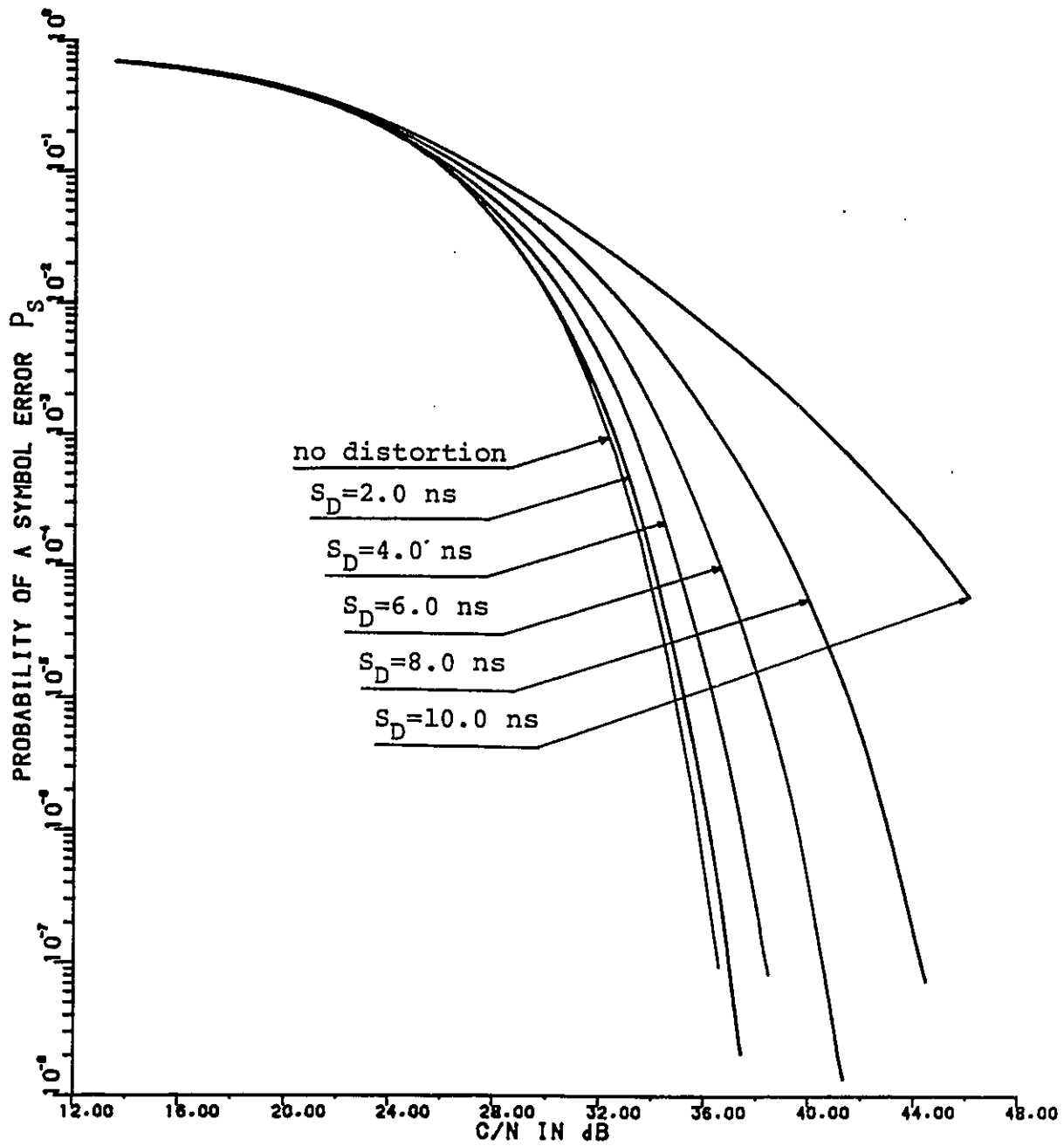


Figure 4.16: Impact of different sinusoidal group delay distortion coefficients on the $P_S = f(C/N)$ performance of a 15 MBaud/s 512-QAM system. Filters with $\alpha = 0.2$ are used. The noise bandwidth is defined as the double-sided Nyquist bandwidth.

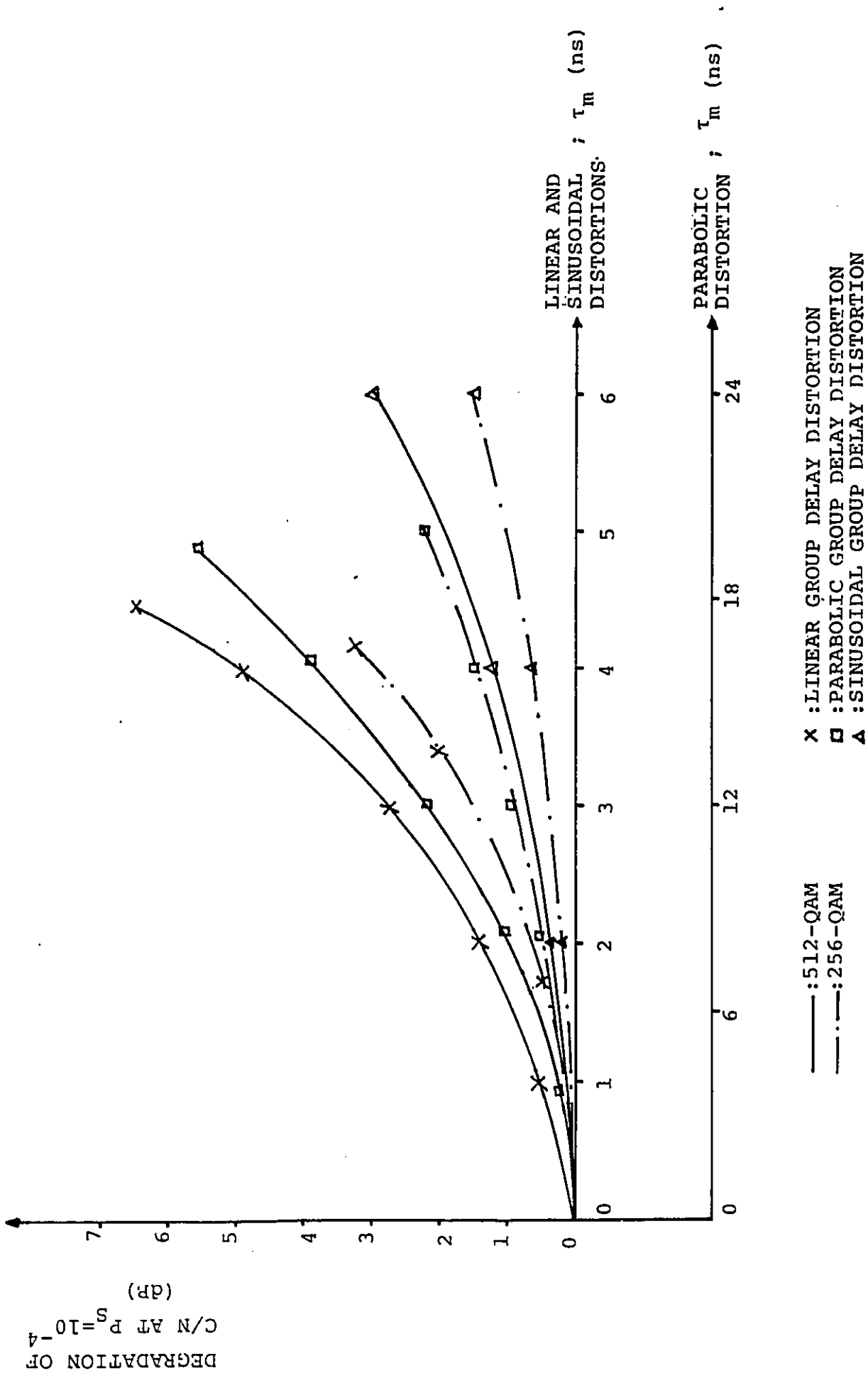


Figure 4.17: Degradation of C/N as a function of maximal group delay τ_m for 512- and 256-QAM for linear, parabolic and sinusoidal group delay distortions. Both systems have the same symbol rate of 15 MBaud/s and employ raised-cosine filters with roll-off factors of $\alpha = 0.2$.

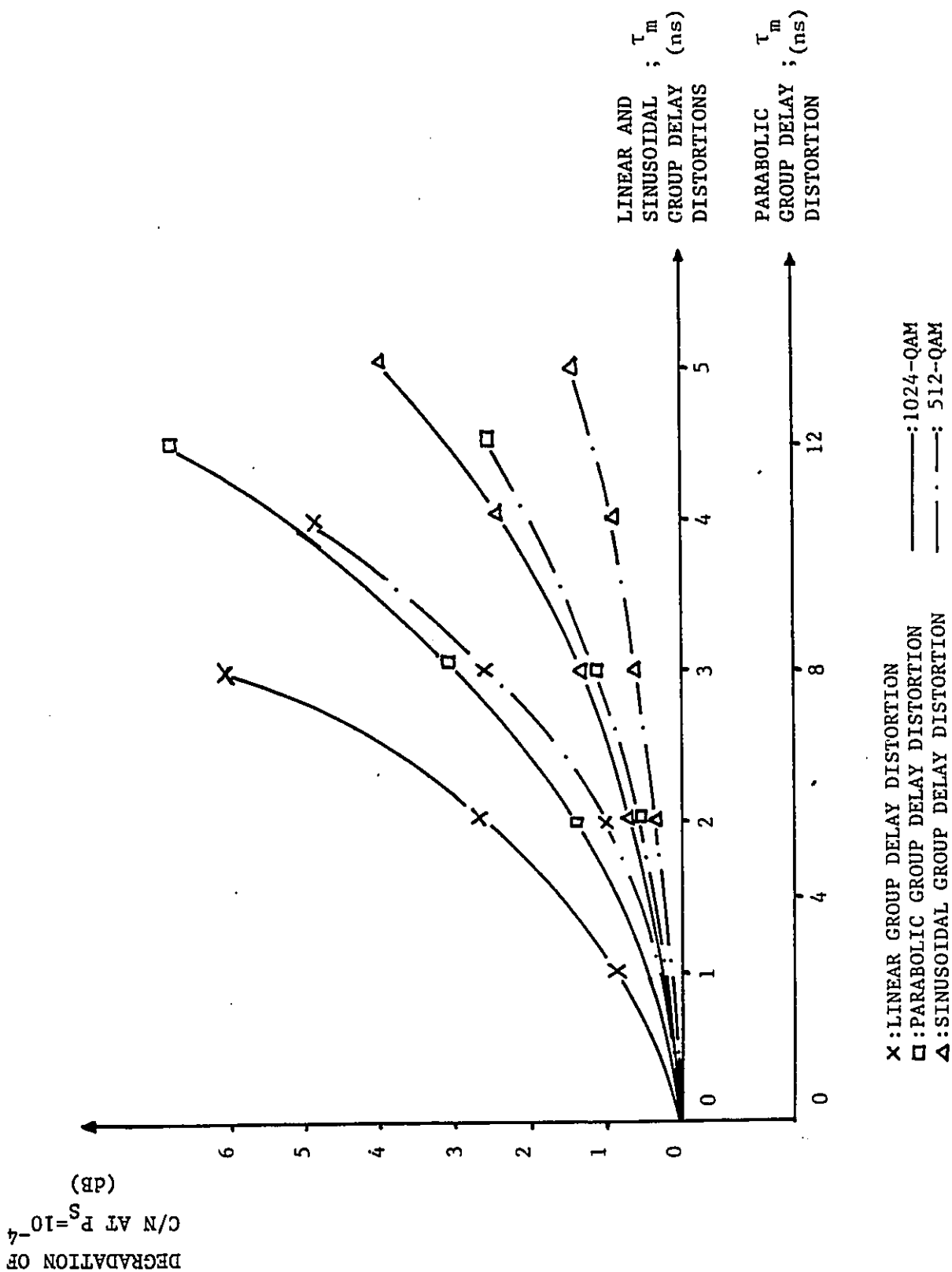


Figure 4.18: Impact of linear, parabolic and sinusoidal group delay distortions on the P_S performance degradation as a function of τ_m of a 1024-QAM system employing $\alpha = 0.2$ raised-cosine filters and having a symbol rate of 15 MBaud/s. For comparison purposes we have included the previously obtained results of an equivalent (same symbol rate and filtering strategy) 512-QAM system.

the obtained results. To begin with, let us consider first the amplitude distortions.

As illustrated in Fig. 4.19, the sinusoidal amplitude distortion with a peak amplitude of A_m seems to interfere with the QAM signal the most, whereas the linear amplitude distortion the least. Indeed, if we adopt the Square Area of each of the three Interferences (SAI), as measure of the overall system distortion, then the following expression for the SAI of each of the three distortions can easily be obtained from Fig. 4.19 as follows:

$$SAI_{lin.} = \frac{1}{12} A_m^2 f_{max}; \quad (4.12)$$

$$SAI_{parab.} = \frac{1}{5} A_m^2 f_{max}; \quad (4.13)$$

$$SAI_{sin.} = \frac{1}{4} A_m^2 f_{max}. \quad (4.14)$$

Clearly from the above equations, the “amount” of sinusoidal interference is the highest as compared to the equivalent linear and parabolic distortions, which suggests that the degradation due to sinusoidal amplitude distortion should be the highest. By the same token, one should expect that the linear amplitude distortion will cause the least degradation.

The effects of group delay distortion are more difficult to analyze. However, with some efforts, we still can get some insight into it. As well known, ideal filters should exhibit a constant group delay over their entire bandwidth. In terms of their phase characteristics this translates into a linear phase-frequency relationship. In the case of linear group delay distortion, which corresponds to parabolic phase-frequency characteristics, there will be crosstalk between the I- and Q-channels. On the other hand for parabolic group delay distortion there is no crosstalk. It turns out from a rigorous analysis by Sunde [77] that, in general, the crosstalk effect results in an increased degradation for the linear group delay distortion. Clearly our performance evaluation results of Figs. 4.17 and 4.18 are in agreement with Sunde’s

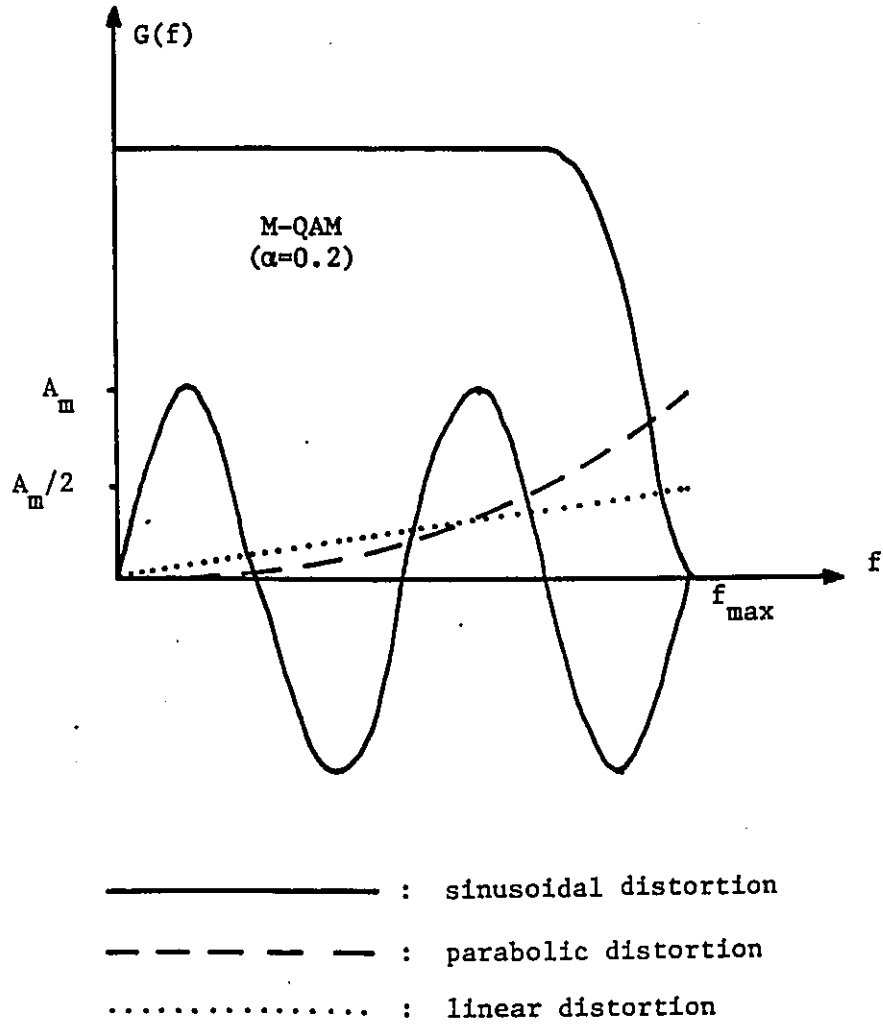


Figure 4.19: Graphical representation in the frequency domain of an M -ary QAM, $\alpha = 0.2$, signal with linear, parabolic and sinusoidal amplitude distortions which have the same peak distortion A_m . Notice that for a fair comparison the equivalent peak distortion of the linear amplitude distortion is $A_m/2$ (see Eq. 4.7).

observations. Finally, coming up to the case of sinusoidal group delay distortion, it is difficult to give a proper explanation. However, the results of Figs. 4.17 and 4.18 indicate that the degradation caused by sinusoidal group delay distortion is somewhere in between of that of linear and parabolic group delay distortions.

4.5 GENERALIZATION OF THE OBTAINED RESULTS TO SYSTEMS WITH DIFFERENT SIGNALLING RATES

The question which naturally arises at this point, is whether or not the obtained results can be applied to systems having the same modulation formats but *different symbol rates*. The answer to this question is fortunately yes, and thus the purpose of this section is to show how this can be achieved. In particular, a simple scaling procedure will be described, which will allow one to efficiently obtain corresponding results for equivalent QAM systems of different symbol rates, without repeating similar computer simulation procedures and thereby saving a considerable amount of computing time and efforts.

The central idea behind the proposed method is that the bandwidth of the employed $\sqrt{\alpha}$ raised-cosine filters is exactly related to the symbol rate, f_S , of the system under consideration. This means that once f_S is specified and assuming α is known, then the bandwidth as well as the shape of the employed raised-cosine filters is well defined. For example, let us consider two systems with Baud rates f_{S1} and f_{S2} , suffering linear group delay distortions, which are characterized by two different parameters L_{D1} and L_{D2} , respectively. It is useful to define the normalized to f_S maximum group delay distortion, $\bar{\tau}_m^{\infty}$, as:

$$\bar{\tau}_m^{\infty} = \tau_m f_S \quad (4.15)$$

where τ_m is the maximum group delay distortion as shown in Eq. 4.11. Then it is clear that, for systems having the same modulation format

but different f_s , the linear group delay distortions with the same τ_m will cause equal performance degradation. As a result, for two linear group delay distortions which are causing the same performance degradation the following relation can be obtained from Eq. 4.15:

$$\tau_{m1}f_{s1} = \tau_{m2}f_{s2} \quad (4.16)$$

or equivalently, by using Eq. 4.11, the above equation becomes:

$$L_{D1}f_{s1}^2 = L_{D2}f_{s2}^2. \quad (4.17)$$

Similarly, let us consider the case of having the same type amplitude distortions and the same normalized to the symbol rate maximum amplitude distortion, A_m . It is clear now that these distortions will result in equal degradation for systems with same modulation format but different symbol rates. Thus, for two linear amplitude distortions with parameters L_{A1} and L_{A2} , which will cause the same degradation to two systems with symbol rates f_{s1} and f_{s2} , respectively, the following relationship is valid:

$$L_{A1}f_{s1} = L_{A2}f_{s2}. \quad (4.18)$$

Using a similar procedure equivalent conditions for the remaining amplitude and group delay distortions can be derived. These results are summarized in Table 4.1.

	GROUP DELAY DISTORTIONS	AMPLITUDE DISTORTIONS
Linear	$L_{D1}f_{s1}^2 = L_{D2}f_{s2}^2$	$L_{A1}f_{s1} = L_{A2}f_{s2}$
Parabolic	$P_{D1}f_{s1}^3 = P_{D2}f_{s2}^3$	$P_{A1}f_{s1}^2 = P_{A2}f_{s2}^2$
Sinusoidal	$S_{D1}f_{s1} = S_{D2}f_{s2}$	$S_{A1} = S_{A2}$

Table 4.1: Relationships between the different parameters of amplitude, group delay distortions and symbol rates for two digital transmission systems having the same modulation format and performance but different symbol rates. The definitions of the distortion parameters L_D, P_D, S_D and L_A, P_A, S_A are given in Eqs. 4.10 and 4.6, respectively.

Finally, the following two examples will illustrate how our previous results for the 15 MBaud/s 1024-QAM (i.e. 150 Mb/s) system can be used to calculate the corresponding results of an equivalent 18 MBaud/s (180 Mb/s) system (see Table 1.1).

From Fig. 4.18, it is noted that in the case of linear group delay distortion, a $\tau_m = 2$ ns results in 2.5 dB degradation in C/N for a 15 MBaud/s, $\alpha = 0.2$, 1024-QAM system. It can be easily found, that this distortion corresponds to a coefficient $L_D = (1/9)$ ns/MHz. Thus, by using Eq. 4.16 and substituting for $\tau_{m1} = 2$ ns, $f_{S1} = 15$ MBaud/s and $f_{S2} = 18$ MBaud/s, we obtain $\tau_{m2} = 1.67$ ns. This means that a τ_m of 1.67 ns will result in the same 2.5 dB degradation in C/N for the 18 MBaud/s, 1024-QAM system. Furthermore, by using the relationship for linear group delay distortion of Table 4.1, we obtain the corresponding coefficient L_{D2} , as follows:

$$L_{D2} = \frac{1}{9} \frac{\text{ns}}{\text{MHz}} \cdot \left(\frac{18}{15}\right)^2 = 0.16 \frac{\text{ns}}{\text{MHz}}. \quad (4.19)$$

This means that the P_S curve which corresponds to $L_D = (1/9)$ ns/MHz for a 15 MBaud/s system also represents the performance of a 18 MBaud/s system with $L_D = 0.16$ ns/MHz. Following a similar procedure, it can be easily shown that for the linear amplitude distortion case, the performance of a 15 MBaud/s, 1024-QAM system with L_{A1} is the same as the performance of a 18 MBaud/s, 1024-QAM system with $L_{A2} = L_{A1} \cdot (18/15)$ dB/MHz.

Thus, as a conclusion, all the results presented in Section 4.3 and which were obtained for 15 MBaud/s, $\alpha = 0.2$ systems can be easily transformed, so that they can represent the performance of equivalent QAM systems with the same modulation formats but of different symbol rates.

4.6 CONCLUSIONS

In this chapter, the performance of 512- and 1024-QAM systems in the presence of amplitude and group delay distortions was studied. Computer simulated results indicate that sinusoidal amplitude and linear

group delay distortions cause the most severe $P_S = f(C/N)$ degradation. The presented results could be useful to modem designers since it could indicate the tolerance imperfections in filter implementation which would not degrade the system performance below a specific value. Additionally, we presented some heuristic interpretations of the obtained results. Finally, we introduced and analyzed a simple scaling procedure, which enables one to apply the obtained results to any other rate system without having to go through similar lengthy and sometimes tedious computational procedures.

Chapter 5

THE PERFORMANCE OF 512- AND 1024-QAM SYSTEMS IN A CO-CHANNEL INTERFERENCE ENVIRONMENT

5.1 INTRODUCTION

As mentioned in Chapter 1, multilevel QAM schemes are essential means of increasing the achievable spectral efficiency. One other equally attractive way of increasing digital channel capacity is the frequency reuse technique, where the data are transmitted at the same carrier frequency on both the horizontal and vertical polarization of the antenna. This technique is also known as dual polarization operation [82] and can theoretically increase the spectral efficiency by a factor of two. However, in this frequency reuse environment Co-Channel Interference (CCI) is one of the major sources of performance degradation [83]-[88].

As an extension to Chapter 4, in this chapter we evaluate the performance of both 512- and 1024-QAM systems in the presence of CCI by means of computer simulation and analysis. We will investigate the effects of amplitude modulated and constant envelope (sinusoidal) interferers on the performance of both QAM schemes, so that their potential use for radio system applications can be assessed [108]. The organization of Chapter 5 is as follows. After this introduction, in the next section we will present a brief description of the computer simulated CCI system model. In Section 5.3, the obtained performance evaluation results will be presented and analyzed. Section 5.4 presents a theoretical analysis which is used to verify some of the obtained computer simulation results. Finally, Section 5.5 contains the conclusions of this chapter.

5.2 CCI SYSTEM MODEL AND DESCRIPTION OF THE COMPUTER SIMULATIONS

Fig. 5.1 illustrates a block diagram of the system model which was used to simulate the effects of CCI on the QAM schemes. The computer generated 512- or 1024-QAM signals, $s(t)$, are mathematically represented as in Eq. 4.1. We will consider two types of CCI, namely:

- Amplitude Modulated CCI, and
- Constant Envelope (Sinusoidal) CCI.

For the case of amplitude modulated CCI, the interference which corrupts the transmitted QAM signal consists of the same format but statistically independent QAM signal, so that the interfering signal can be expressed as:

$$i(t) = \text{Re}\{R \sum_{n=-\infty}^{\infty} (\tilde{I}_n - j\tilde{Q}_n)g_T(t - nT - \tau)e^{j(2\pi f_c t + \theta)}\}. \quad (5.1)$$

In the above equation, $\text{Re}\{\cdot\}$ denotes real part of $\{\cdot\}$, $\tilde{I}_n$ and $\tilde{Q}_n$ are independent and equiprobable discrete random variables, taking

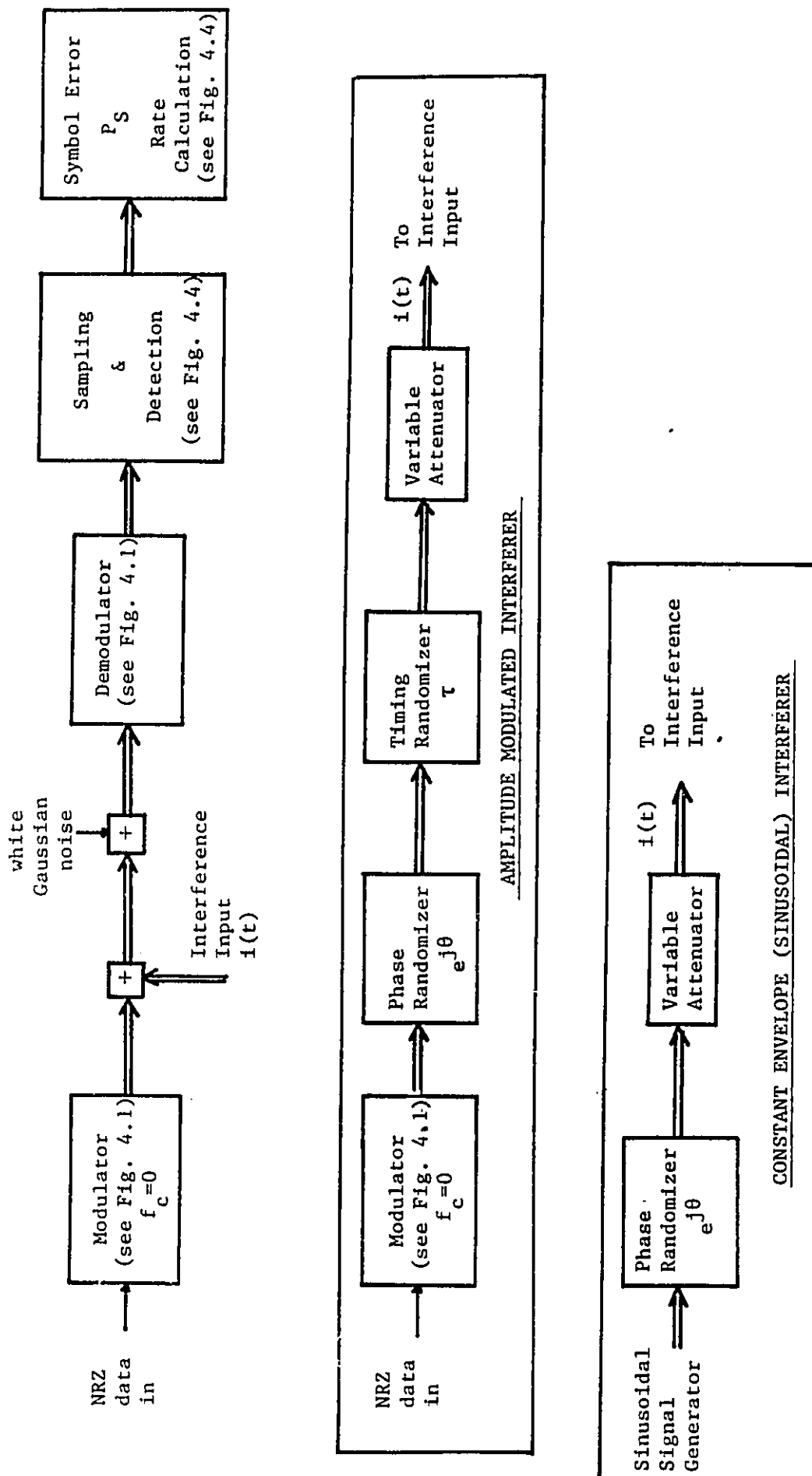


Figure 5.1: Block diagram of the employed simulation model of the CCI system. In this work the modulators generate 512- or 1024-QAM signals. Including in this figure are both types of CCI, i.e. amplitude modulated and constant envelope (sinusoidal) interferers.

values from the same alphabet as $\{I_n\}$ and $\{Q_n\}$ of Eq. 4.1, $1/R^2$ is the Carrier-to-Interference (C/I) ratio and f_c is the carrier frequency. τ and θ are the differences in symbol timing and carrier phase between the desired signal and the interferer. It is assumed, that both τ and θ are independent random variables uniformly distributed over the ranges $(0, T]$ and $(-\pi, \pi]$, respectively. All other symbols are the same as in Eq. 4.1.

For Constant Envelope CCI, the interference consists of one constant envelope sinusoidal interferer, which can be represented as:

$$i(t) = \sqrt{2}R \cos(2\pi f_c t + \theta) \quad (5.2)$$

with R^2 the power of the interferer and θ the difference in carrier phase angle between the desired signal and the interferer. We will again assume here that θ is a random variable uniformly distributed over the range $(-\pi, \pi]$.

Similarly to the simulations of Chapter 4, we employed the equivalent complex baseband form of the modulated system, thus it is assumed that the carrier frequency, f_c is known to the receiver. The transmit and receive filters are assumed to be ideal $\sqrt{\alpha}=0.2$ raised-cosine filters with a $x/\sin(x)$ aperture equalizer at the transmit filter, so that the cascade of these two filters satisfies Nyquist's I criterion. There are $(2^{20} - 1)$ complex symbols transmitted, with each of them represented by 64 samples. For filtering, Fast Fourier Transforms (FFT) techniques are employed to alternate between the frequency and time domains.

The lack of coherence between the main and the interfering channel is included in the phase randomizer, where the interfering signal is multiplied by $e^{i\theta}$. The lack of synchronization is simulated by the timing randomizer, which simply shifts the interfering data sequence randomly by τ samples. In our simulations, τ is assumed to be a discrete random variable uniformly distributed over the assigned number of samples per symbol, i.e. 64 samples. For both kind of interferences $1/R^2$ is the C/I ratio, which could easily be changed by means of a variable attenuator, as shown in Fig. 5.1.

For the computation of the average symbol error probability, P_S , sixteen values of θ as well as τ are taken for the interfering signal. Let

$P_S(\theta_i, \tau_i)$ be the conditional symbol error probability assuming $\theta = \theta_i$ and $\tau = \tau_i$. P_S can then be calculated as:

$$P_S = \frac{1}{16}[P_S(\theta_1, \tau_1) + P_S(\theta_2, \tau_2) + \dots + P_S(\theta_{16}, \tau_{16})]. \quad (5.3)$$

To compute each of the conditional symbol error probabilities as a function of C/N and having C/I as parameter, exactly the same procedure as the one described in Section 4.2 was followed. Clearly, the only conceptual difference here is that the system degradation is due to another interfering signal rather than filter imperfections as was the case described in Chapter 4. Finally, by averaging the results using Eq. 5.3, both 512- and 1024-QAM system performances were evaluated. The obtained results will be presented in the next section.

5.3 COMPUTER SIMULATED RESULTS AND DISCUSSION

Since a system design engineer often wants to maintain a specified symbol error rate even in the presence of interference, the increase in C/N required to maintain a constant symbol error rate of $P_S = 10^{-6}$ for 512- and 1024-QAM with both amplitude modulated and constant envelope CCI as functions of the C/I are plotted in Figs. 5.2 and 5.3, respectively. Before we interpret the obtained results a few explanations and findings about the simulation methodology are in order.

First of all, due to the symmetric constellation of the QAM signals only the range of $(0, \pi/2]$ is necessary to be considered as possible values for θ . To prove this point let us consider the following simplified expression for an amplitude modulated interferer: ¹

$$i_{\theta_1}(t) = \tilde{I}_n \cos(\theta_1) + \tilde{Q}_n \sin(\theta_2). \quad (5.4)$$

¹Clearly, Eq. 5.4 results from Eq. 5.1 by assuming, for convenience but without any loss of generality, that $R = 1$, $g_T(t) = 1$, $\tau = 0$ and $f_c = 0$.

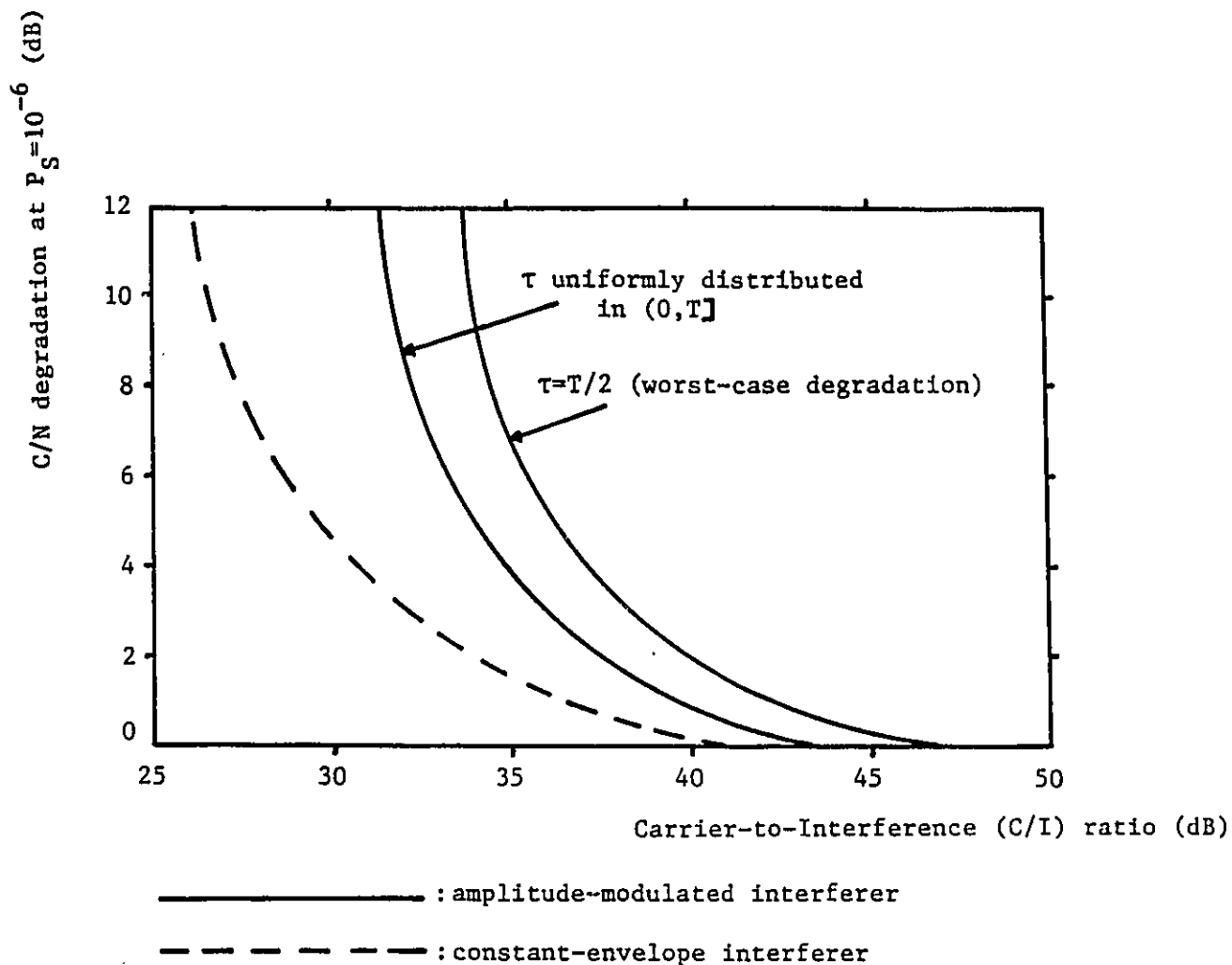


Figure 5.2: C/N degradation at a symbol error rate of $P_S = 10^{-6}$ for a 512-QAM, $\alpha = 0.2$ system in the presence of amplitude and constant envelope interferers. With no interference the required C/N for the same P_S is 36 dB.

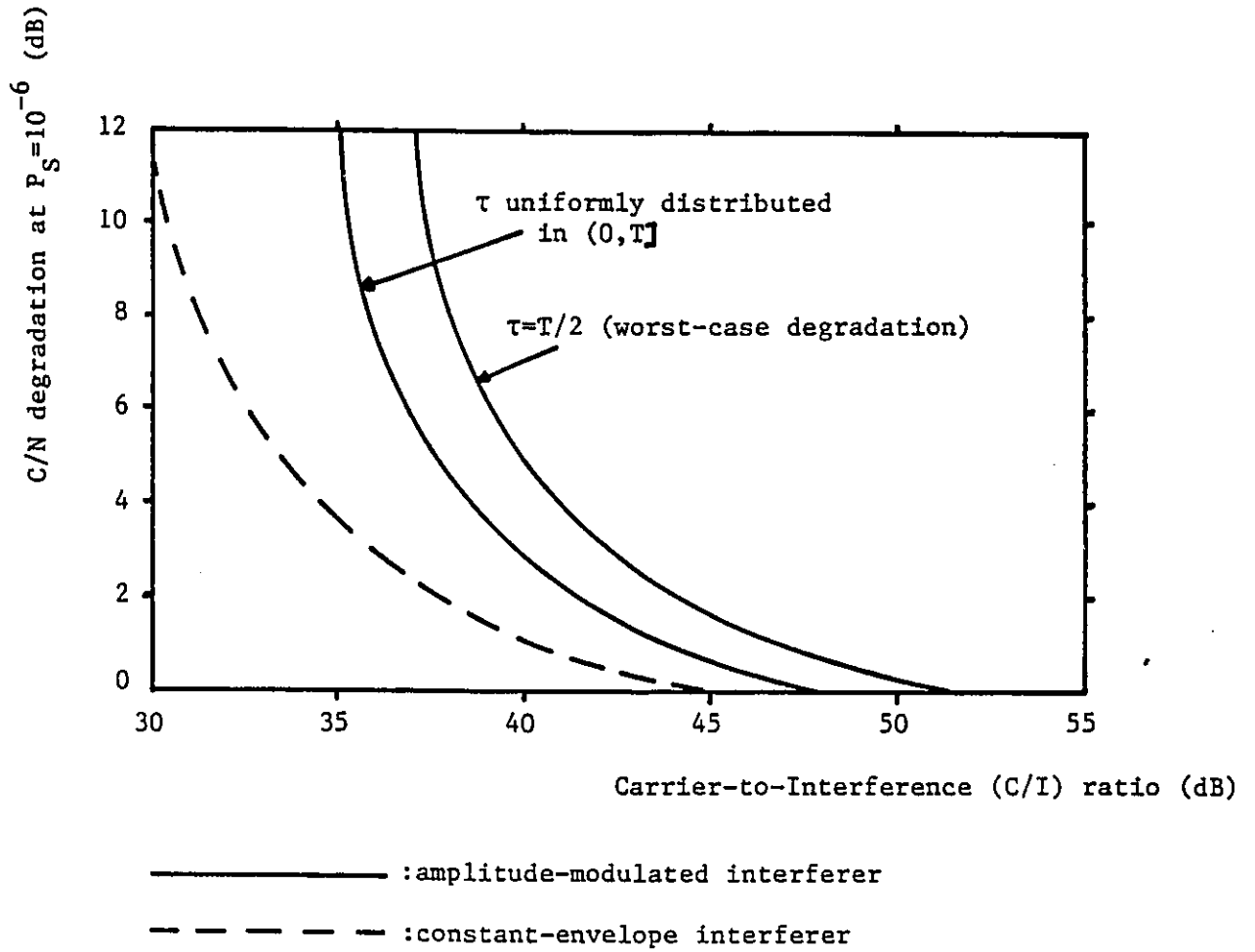


Figure 5.3: C/N degradation at a symbol error rate of $P_S = 10^{-6}$ for a 1024-QAM, $\alpha = 0.2$ system in the presence of amplitude and constant envelope interferers. With no interference the required C/N for the same P_S is 39 dB.

Assuming now that the interferer has a carrier phase angle of θ_2 , which is given by:

$$\theta_2 = \theta_1 + k(\pi/2) \ ; \ k = 0, 1, 2, 3 \quad (5.5)$$

then the corresponding interference signal can be written as:

$$i_{\theta_2}(t) = \tilde{I}_n \cos(\theta_2) + \tilde{Q}_n \sin(\theta_2) . \quad (5.6)$$

Equivalently, Eq. 5.6 can be rewritten as:

$$i_{\theta_2}(t) = \begin{cases} \tilde{I}_n \cos \theta_1 + \tilde{Q}_n \sin \theta_1, & \text{for } k=0; \\ -\tilde{I}_n \cos \theta_1 + \tilde{Q}_n \sin \theta_1, & \text{for } k=1; \\ \tilde{I}_n \cos \theta_1 - \tilde{Q}_n \sin \theta_1, & \text{for } k=2; \\ -\tilde{I}_n \cos \theta_1 - \tilde{Q}_n \sin \theta_1, & \text{for } k=3. \end{cases} \quad (5.7)$$

Eq. 5.7 indicates that for phase angles which are described by Eq. 5.5 do not alter the amplitude of the transmitted symbols but only their sign. As a result, values of θ in the range of only $(0, \pi/2]$ can be considered. This fact was also verified by extensive computer simulation runs.

One other finding was that the worst degradation, conditioned on τ_i , was obtained for $\tau_i = T/2$. Indeed, it is well-known that for a system employing raised-cosine filters with an excess bandwidth of α , the highest overshoot occurs for $\tau = t_{op} + T/2$, where t_{op} is the optimum sampling instant ² [89]. This phenomenon is more intensive in the QAM systems under consideration, since they are employing $\alpha = 0.2$ raised-cosine filtering strategies.

Finally, and in order to ensure the validity of our results, different patterns of the transmitted data sequences were used and then the obtained results were averaged. It was found that for each individual data sequence the difference in performance was less than ± 0.5 dB for a C/N degradation of 8 dB. Thus the estimated error of the obtained results is not more than 1 dB at a C/N degradation of 8 dB. However, it should be mentioned that for degradations of less than 8 dB this error is less.

² t_{op} is referred to as the sampling instant at which the eye has its maximal opening.

Comparing now Figs. 5.2 and 5.3, we observe that the amplitude modulated CCI produces more degradation in both QAM schemes than the constant envelope CCI of the same average power. This can be easily perceived by comparing the peak value of both types of interferers assuming they have the same average power. If for example, $R^2/2$ is the average power of the interfering signal, then the peak amplitude of the constant envelope CCI is R . However, for an amplitude modulated interferer³ since its average power is given by $R^2(L^2 - 1)/3$ its peak amplitude is $R(L - 1)\sqrt{3/(L^2 - 1)}$, where $L = 32$ for 1024-QAM and $L = 24$ for 512-QAM. Therefore, for a 1024-QAM system the peak amplitude for an amplitude modulated CCI is $31/\sqrt{341} \doteq 1.679$ times the peak amplitude for a constant envelope interferer of the same average power. For a 512-QAM system this ratio becomes $23/\sqrt{192} \doteq 1.660$. As a result, for both QAM systems, the peak amplitude of the amplitude modulated CCI reduces the minimum distance between any two signal points more than a constant envelope interferer of the same power. Thus, since the symbol error probability is a strong function of this minimum distance the expected degradation with the amplitude modulated CCI is higher than that with a constant envelope interferer.

Furthermore, in Figs. 5.2 and 5.3 two sets of results for the amplitude modulated CCI are presented. In particular, curves showing the C/N degradations at $P_S = 10^{-6}$ for the averaged τ (i.e. for uniformly distributed τ in the $(0, T]$ interval) as well as for the worst-case value of τ (i.e. $\tau = T/2$) are illustrated.

Because of the high-precision atomic standards used in many large telecommunication centers produce frequencies which differ only in parts per billion [90], the symbol timing clocks of two distinct signals (e.g. the desired signal and the CCI) may be almost identical, with a given value of τ persisting over many symbol intervals. If a near to worst-case value of τ persists over a large number of symbol intervals, a user may perceive a different and possibly worse performance than the expected with errors concentrated in these intervals with other values of τ . Thus, the results indicating the difference between worst-case τ and averaged τ would be useful in providing insight into the

³For simplicity it will be assumed here that there is no filtering in the system.

overall performance for a particular set of communication links.

5.4 THEORETICAL EVALUATION OF THE MAXIMUM TOLERABLE C/I FOR ERROR-FREE TRANSMISSION WITH NO GAUSSIAN NOISE

As it can be seen from Figs. 5.2 and 5.3, the C/N degradation curves clearly indicate a transition from a noise-limited performance at high C/I to an interference-limited performance at low C/I . We will define as $(C/I)_{min}$ of the interference-limited performance the asymptotical minimum value of C/I necessary to maintain error-free communication without any Gaussian noise, i.e. for $C/N \rightarrow \infty$.

$(C/I)_{min}$ can be read from Figs. 5.2 and 5.3 to be 26.5 dB and 30 dB for 512- and 1024-QAM, respectively, with a single sinusoidal interferer. From the same figures the values of $(C/I)_{min}$ are found to be 32.5 dB and 35.0 dB, respectively, with an amplitude-modulated CCI for an averaged τ . In what follows, we shall theoretically analyze and evaluate the $(C/I)_{min}$ which is required for error-free transmission in the absence of noise for both 512- and 1024-QAM in order to confirm the previously reported computer simulated results.

Under the assumption that the Gaussian noise can be considered negligible, error-free transmission can be achieved only if the peak value of the CCI is less than the decision distance, d , of the desired signal. As previously stated, the received levels in the I- and Q-channel of a QAM system are equally likely to take values from the alphabet $\{\pm d, \pm 3d, \pm 5d, \dots, \pm(L-1)d\}$. Clearly, the average received signal power C is given by:

$$C = \frac{2}{M} \sum_{i=1}^{L/2} d^2(2i-1) . \quad (5.8)$$

Evaluation of the above sum yields:

$$C = \frac{d^2}{3}(L^2 - 1) . \quad (5.9)$$

Similarly, if the CCI is of the same QAM format as the desired signal, its average interference power I can be written as:

$$I = \frac{d_I^2}{3}(L^2 - 1) \quad (5.10)$$

where $2d_I$ is the distance between adjacent signal levels of the CCI. Furthermore, its peak power I_p is given by:

$$I_p = [(L - 1)d_I]^2 . \quad (5.11)$$

As a result, the condition of error-free transmission now becomes:

$$I_p \leq d^2 . \quad (5.12)$$

Equivalently, by making use of Eqs. 5.8 – 5.11 the above condition can be rewritten as:

$$\left(\frac{C}{I}\right) \geq PF \frac{L^2 - 1}{3} \quad (5.13)$$

where PF is the well-known peak factor [91] and is defined as the ratio between the peak to the average power of the CCI, i.e.:

$$PF \stackrel{\text{def}}{=} \frac{I_p}{I} . \quad (5.14)$$

It is easy to understand that I_p depends on the modulation format as well as the employed filtering strategy. For example, in a QAM scheme employing $\alpha = 1.0$ there is practically no overshoot between different symbol transitions (see Fig. 3.6), whereas for $\alpha = 0.2$ filters the amount of overshoot is considerably high (see Fig. 4.7). In [91] it was shown that the power ratio of the maximum overshoot (due to filtering) to the nominal peak value of the transmitted signal (without any filtering) is 5.3 dB for QAM systems employing raised-cosine filters with $\alpha = 0.2$.

Thus, by using Eqs. 5.10, 5.11, 5.13 and 5.14, we can write the following expression for the $(C/I)_{\min}$ which is valid for an amplitude modulated CCI having a QAM signalling format and employing $\alpha =$

0.2 raised-cosine filters:

$$\begin{aligned}
\left(\frac{C}{I}\right)_{min}(\text{in dB}) &= PF + 10 \log_{10}\left(\frac{L^2 - 1}{3}\right) \\
&= 5.3\text{dB} + 10 \log_{10}\left[\frac{(L-1)^2 d^2}{\frac{d^2}{3}(L^2 - 1)}\right] + 10 \log_{10}\left(\frac{L^2 - 1}{3}\right) \\
&= 5.3\text{dB} + 10 \log_{10}\left[\frac{3(L-1)}{L+1}\right] \\
&\quad + 10 \log_{10}\left(\frac{L^2 - 1}{3}\right). \tag{5.15}
\end{aligned}$$

For sinusoidal CCI, since $PF = 3$ dB, the following formula is valid:

$$\left(\frac{C}{I}\right)_{min}(\text{in dB}) = 3 \text{ dB} + 10 \log_{10}\left(\frac{L^2 - 1}{3}\right). \tag{5.16}$$

For both Eqs. 5.15 and 5.16, we have $L = 32$ for 1024-QAM and $L = 24$ for 512-QAM.

The values for PF and $(C/I)_{min}$ for both 512- and 1024-QAM in the presence of CCI are summarized in Table 5.1. These theoretical results are in close agreement with the previously obtained results by computer simulation as shown in Figs. 5.2 and 5.3.

	AMPLITUDE MODULATED CCI		SINUSOIDAL CCI	
	PF (dB)	$(C/I)_{min}$ (dB)	PF (dB)	$(C/I)_{min}$ (dB)
512-QAM ($\alpha=0.2$)	9.7	32.2	3.0	25.5
1024-QAM ($\alpha=0.2$)	9.8	35.1	3.0	28.3

Table 5.1: $(C/I)_{min}$ values required for error-free transmission with no Gaussian noise and corresponding peak factors (PF) of 512- and 1024-QAM systems employing raised-cosine filters with $\alpha = 0.2$ operating in the presence of amplitude modulated and sinusoidal (constant envelope) CCI.

5.5 CONCLUSIONS

This chapter has dealt with the evaluation of the performance of 512- and 1024-QAM systems when they are corrupted by thermal noise and a set of constant envelope or amplitude modulated co-channel interferers.

For a symbol error rate of 10^{-6} and when there is an amplitude modulated CCI of 37.5 dB the degradations for 512- and 1024-QAM are 2 dB and 5 dB respectively. Similarly, for a constant envelope co-channel interferer of 37.5 dB the degradations are less, namely 0.5 dB and 2 dB for the 512- and 1024-QAM, respectively. It was also found that for a C/I of more than 45 dB the C/N degradation was less than 2 dB for the 1024-QAM, whereas it was negligible for the 512-QAM. Thus, if cross polarization discrimination of better than 45 dB can be attained, the design of 512- and 1024-QAM is feasible if the degradations due to other causes, such as filter imperfections, transmitter nonlinearities, nonideal carrier and symbol timing recovery, adjacent channel interference e.t.c. can be kept small.

Chapter 6

ERROR RATE CONSIDERATIONS FOR 512- AND 1024-QAM SYSTEMS IN THE PRESENCE OF ADJACENT CHANNEL INTERFERENCE

6.1 INTRODUCTION

In many high capacity digital radio system applications, multicarrier transmission strategies have been investigated as means of increasing the overall spectral efficiency. Examples of such transmission include the use of multiple carriers each carrying 16-QAM [92],[93], 64-QAM [26] or 256-QAM [20],[24],[26],[81],[94],[95] modulation formats. Additionally, for systems employing even more spectral efficient QAM schemes (i.e. 512- or 1024-QAM) potential applications include the transmission of four 90 Mb/s rate signals within a 40 MHz authorized

bandwidth, two 135 Mb/s rate signals within a 30 MHz bandwidth and other emerging applications (see Table 1.1). In these multicarrier systems, one of the main causes of performance degradation is the interference of the adjacent channels into the useful signal. Thus it is important to investigate the performance of 512- and 1024-QAM systems in an Adjacent Channel Interference (ACI) environment [109], which is also the subject of this chapter. Its organization is as follows. After this introductory section, in Section 6.2, the ACI model and its computer simulated realization will be described. Afterwards, Section 6.3 contains the obtained numerical results and their interpretation. The chapter is concluded with Section 6.4, where the conclusions are presented.

6.2 ADJACENT CHANNEL INTERFERENCE MODEL AND COMPUTER SIMULATION DESCRIPTION

Assuming that the m th interfering signal, $i_m(t)$, is of the same modulation format as the useful QAM signal, $s(t)$, then it can be represented as:

$$i_m(t) = \frac{1}{\sqrt{K}} \operatorname{Re} \left\{ \sum_n (I_{n,m} + jQ_{n,m}) g_T(t - nT - \tau_m) e^{j2\pi(f_c + f_m)t + \theta_m} \right\} \quad (6.1)$$

$I_{n,m}$, $Q_{n,m}$ are independent and equiprobable random variables taking values from the alphabet $\{\pm 1, \pm 3, \dots, \pm 23\}$ for 512-QAM, or $\{\pm 1, \pm 3, \dots, \pm 31\}$ for 1024-QAM. Further, f_c is the carrier frequency, τ_m is the difference in symbol timing between the desired signal and the m th interferer and θ_m denotes the lack of coherence again between $s(t)$ and $i_m(t)$. f_m is the difference in the frequency allocation of the two carriers and K is the Carrier-to-Interference (C/I) power ratio. These last two variables, namely f_m and K are the most important variables and will dominate the performance of the QAM systems under consideration. Finally, T is the symbol rate and $g_T(t)$ is the impulse response of the transmit filter $G_T(f)$.

Based on the above, a block diagram of the simulated multichannel system is shown in Fig. 6.1. It is an equivalent complex baseband model of the modulated system and thus all computations are carried out entirely in the complex baseband form. We consider here only two interfering channels, the Upper Interferer (*UI*) and the Lower Interferer (*LI*), i.e. $m = U$ and $m = L$. Their channel spacing from the Main Channel (*MC*) is f_U and f_L , respectively. The reason why we consider only two interferers is that in most practical applications the interference caused by immediate adjacent channels has the most significant impact on the performance degradation, whereas the impact of the more distant channels is negligible.

The binary data for all three channels are generated by using three different Random Complex Sequence Generators (RCSG). These complex binary data are converted into complex symbols with the in-phase components $I_n, I_{n,L}, I_{n,U}$ corresponding to the real parts of the complex symbols and the quadrature components $Q_n, Q_{n,L}, Q_{n,U}$ corresponding to their imaginary parts. Due to the high number of levels of the modulation schemes under consideration, we are transmitting 2^{20} symbols for each of the I- and Q-channel with every symbol represented by 64 samples. Although the bit rate was chosen to be 90 Mb/s, the results which will be presented in the next section are normalized to the Baud rate and thus are valid for any bit rate system.

The transmit, $G_T(f)$, and receive, $G_R(f)$, filters for all three channels considered are assumed to be ideal square root α raised-cosine filters with an $(x)/\sin(x)$ aperture equalizer at the transmit filter, so that their cascade satisfies Nyquist's I criterion. For filtering we employ fast Fourier transforms (FFT) techniques to alternate between the frequency and the time domain. The lack of coherence between the main and the interfering channels is included in the multiplications of the interfering signals by $e^{j\phi_U}$ and $e^{j\phi_L}$ with ϕ_U and ϕ_L continuous random variables uniformly distributed in $(0, 2\pi]$. Furthermore, the lack of synchronization is introduced by time shifting both interfering signals with respect to the useful signal by τ_U and τ_L . Both τ_U and τ_L are independent, discrete random variables taking equiprobable values from the alphabet $\{0, 1, \dots, T\}$ where T is the number of samples per symbol, which was 64 for our simulations. As already

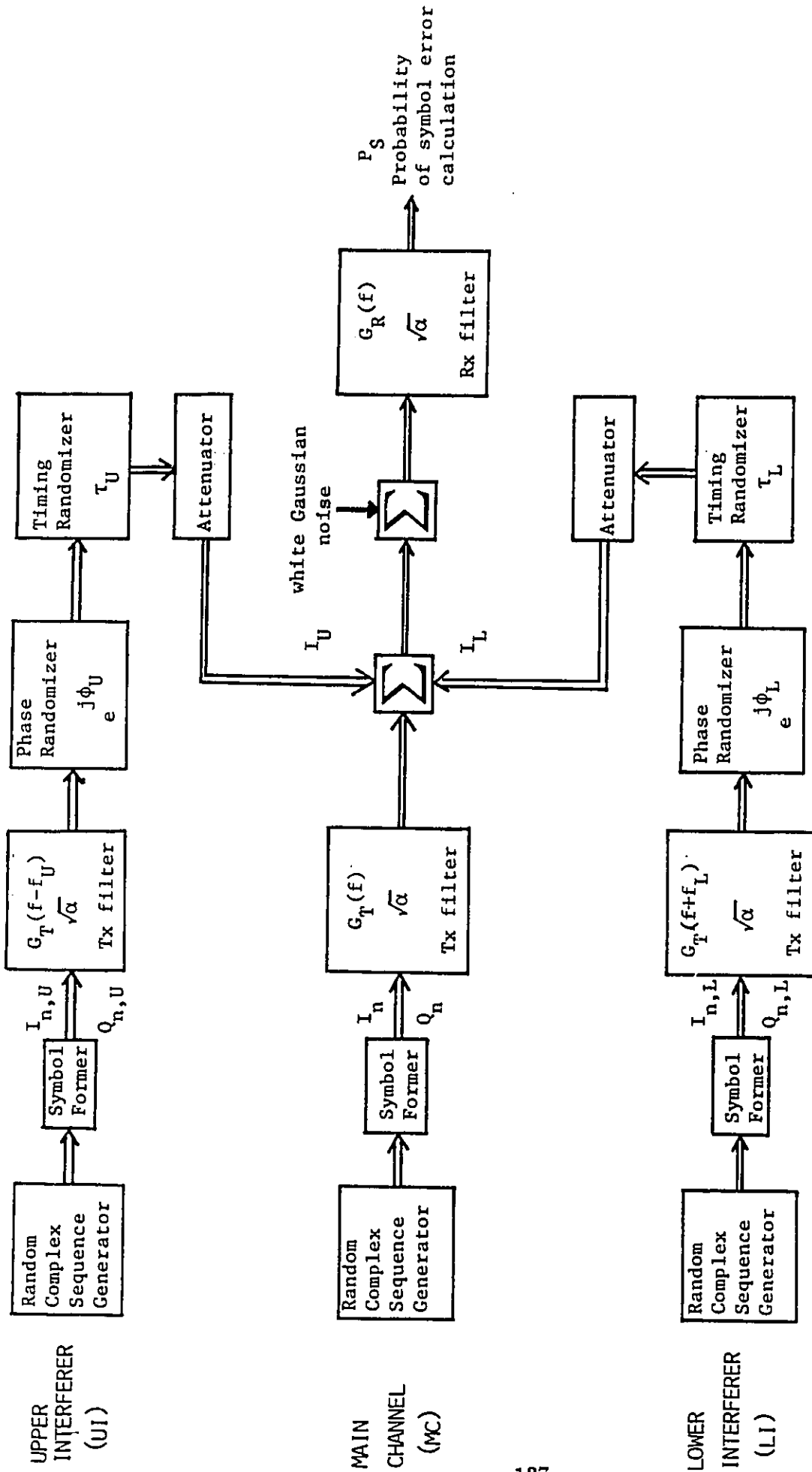


Figure 6.1: Block diagram of the simulation model of a multichannel, adjacent channel interference system, employing 512- and 1024-QAM modulation formats.

For 512-QAM: The transmitted symbols take values from the alphabet $\{\pm 1, \pm 3, \dots, \pm 23\}$.

For 1024-QAM: The transmitted symbols take values from the alphabet $\{\pm 1, \pm 3, \dots, \pm 31\}$.

mentioned in Chapter 5, due to the quadrant symmetry of the QAM scheme it was found that only values $0 < \theta \leq (\pi/2)$ needed to be examined. Furthermore, for the calculation of the average symbol error probability, P_S , we applied the same method as the one explained previously in Section 5.2 (see Eq. 5.3). The worst case interference, i.e. for $\tau = T/2$, was always included in the chosen values of τ_m . The power levels (I_L, I_U) of the interfering channels is controlled by attenuators as shown in Fig. 6.1, thus achieving different (C/I) power ratios. All three signals are added and then the composite signal is transmitted through an AWGN channel. At the receiver, after passing the signal through the receive square root α raised-cosine filter, both I- and Q- channels are sampled at their optimum sampling instants and a decision is made. By determining the distance from the ideal threshold levels for each of the received sampled symbols and by calculating the signal power at the input of the threshold detectors the average symbol error probability, P_S , can be computed using the same methodology as the one described in Section 4.2. The obtained computer simulated results will be presented and analyzed in the next section.

6.3 PERFORMANCE EVALUATION RESULTS AND DISCUSSION

We evaluated the P_S performance of 512- and 1024-QAM systems, which employ $\alpha = 0.5$ and $\alpha = 0.2$ raised-cosine filters in an ACI environment by having the normalized to the symbol rate channel spacing and C/I ratio as parameters. The results, which will be reported here, are obtained under the following assumptions for the interfering signals:

- i) Both have the same modulation format as the desired main channel.
- ii) Both have the same power, i.e. $I_L = I_U = I$ and are symmetrically placed around the main channel, i.e. $f_L = f_U = \Delta f$.

Fig. 6.2 presents the Carrier-to-Noise (C/N) degradation at $P_S = 10^{-6}$ for 512-QAM as well as 1024-QAM systems with $\alpha=0.5$ under

different ACI conditions. Similar results are illustrated in Fig. 6.3, where both 512- and 1024-QAM systems employing raised-cosine filters with $\alpha = 0.2$ were evaluated. By reducing Δf of both interferers essentially the overall bandwidth efficiency is increased. Thus we can define η_{SEI} as the Spectral Efficiency Increase (*SEI*) which for our system is given by:

$$\eta_{SEI} = 2 \cdot \left(1 - \frac{\Delta f \cdot T}{1 + \alpha}\right). \quad (6.2)$$

From the obtained results we can see that systems employing less severe filtering, i.e. higher α , are less susceptible to ACI. For example, for a 1024-QAM, $\alpha = 0.2$ system with $C/I = 0$ dB a *SEI* of 6% will result in a C/N degradation of about 2 dB, whereas an equivalent $\alpha = 0.5$ system results in a degradation of less than 0.5 dB. Furthermore, our results indicate that the spectral efficiency gains which are achieved by reducing the power of the interferers seem to increase considerably for $C/I < 10$ dB. Finally it is interesting to note that even for high interference power, i.e. $C/I = 0$ dB, relatively small spectral efficiencies increases result in practically negligible degradations. As it can be seen from Figs. 6.2 and 6.3, for a 1024-QAM system having $C/I = 0$ dB a *SEI* of about 3% (with $\alpha = 0.2$) and 5% (with $\alpha = 0.5$) is feasible with less than 0.5 dB C/N degradation. Clearly higher *SEI* are feasible for lower C/I ratios.

6.4 CONCLUSIONS

We have investigated, by means of computer simulation, the performance of both 512- and 1024-QAM systems in the presence of adjacent channel interference and additive white Gaussian noise. The obtained results are presented as equivalent C/N degradation with the channel spacing (Δf) and the Carrier-to-Interference (C/I) as parameters. We considered systems employing raised-cosine filters with $\alpha = 0.2$ or 0.5 . Some of our results indicate that for a 1024-QAM, $\alpha = 0.5$ system a spectral efficiency increase of 5% is feasible with negligible C/N degradation.

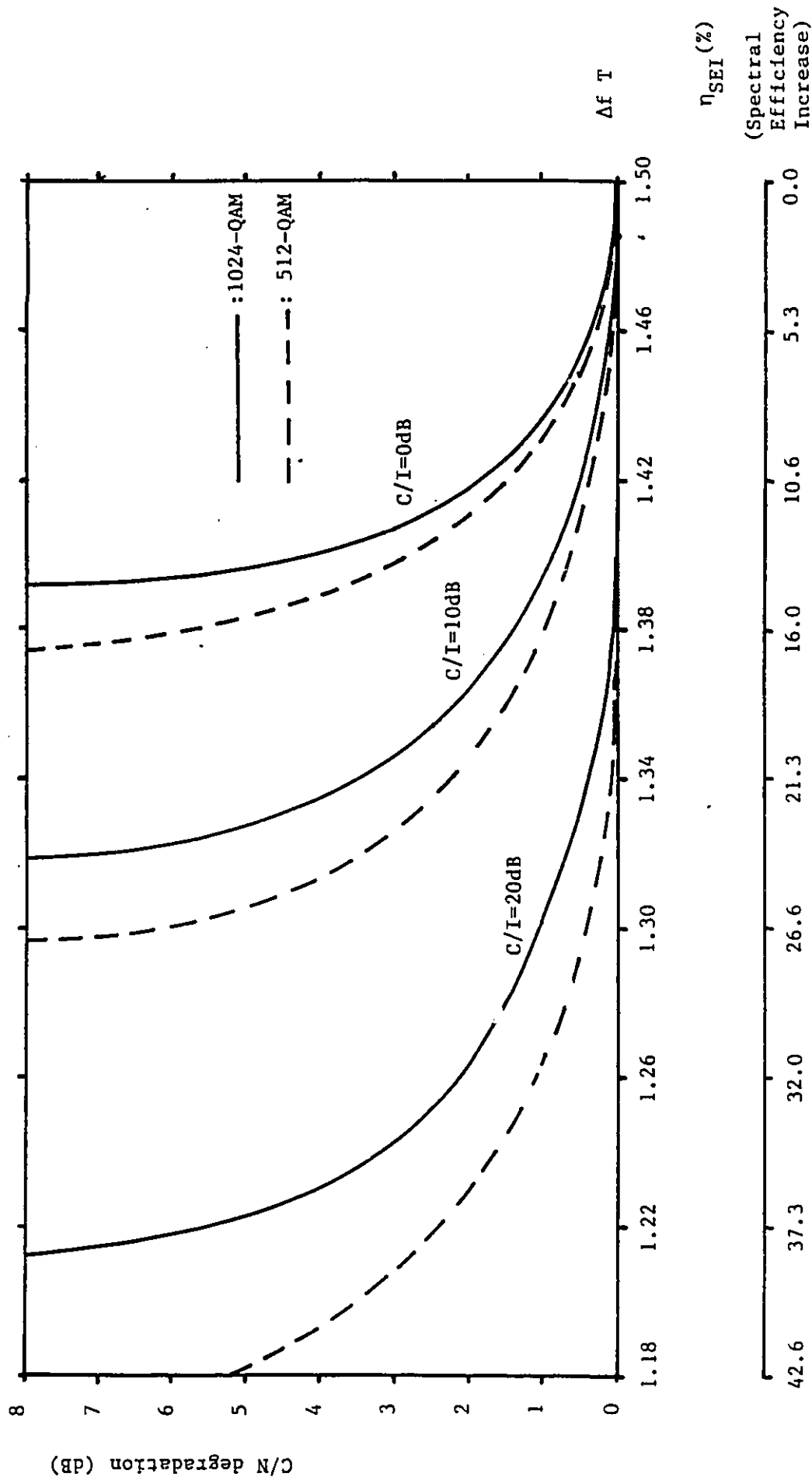


Figure 6.2: C/N degradation, at a symbol error rate $P_S = 10^{-6}$, of 512- and 1024-QAM systems in the presence of ACI. Both systems are employing raised-cosine filters with $\alpha = 0.5$ and are corrupted by two equal power I and equally spaced (Δf) adjacent channel interferers.

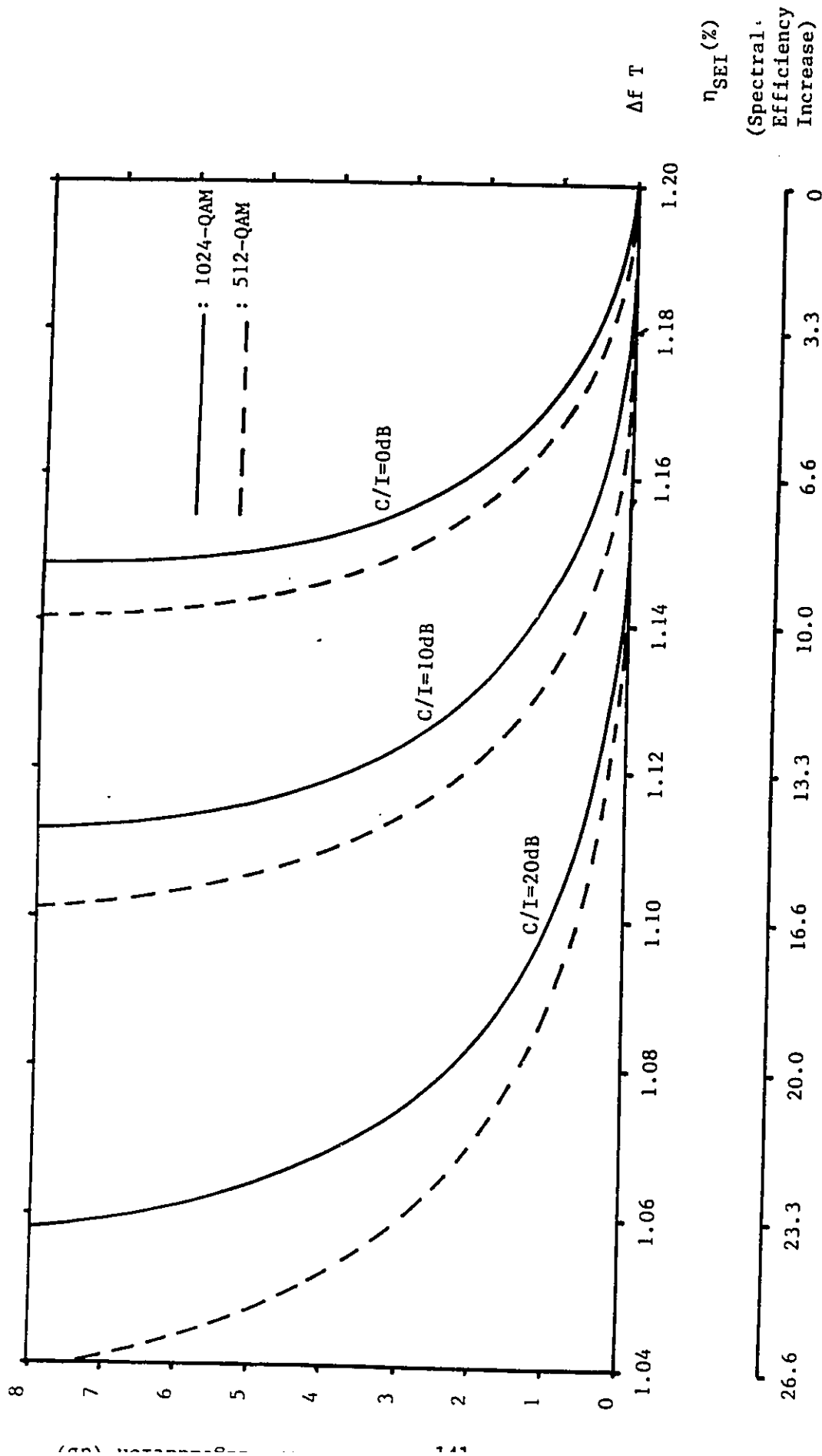


Figure 6.3: C/N degradation, at a symbol rate $P_S = 10^{-6}$, of 512- and 1024-QAM systems in the presence of ACI. Both systems are employing raised-cosine filters with $\alpha \approx 0.2$ and are corrupted by two equal power I and equally spaced (Δf) adjacent channel interferers.

Chapter 7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

7.1 CONCLUSIONS

In this thesis the performance of spectrally efficient QAM systems operating in distorted transmission channels has been studied and improved. The contributions of this thesis can be summarized as follows.

1. A novel technique based on the use of two out-of-band pilot tones for cancelling the effects of transmission SCPJ on the performance of bandwidth efficient QAM systems has been introduced and analyzed. The proposed technique has been applied to a 16-QAM system operated in a combined AWGN and sinusoidal SCPJ environment. Numerous analytical as well as experimental performance evaluation results have been presented having the most important system parameters (e.g. pilot extraction bandwidth and distortion of the pilot extraction filters) as variables. It has been shown that, performance gains of up to 8 dB over conventional, non-pilot based schemes are achievable.
2. A novel method, which employs two in-band pilot tones for si-

multaneous carrier and symbol timing recovery has been introduced and analyzed. Its application to bandwidth efficient QAM system has been investigated in great detail. The major advantage of the proposed technique is that it does not require any excess bandwidth as all other pilot based schemes do. It was shown that the only in-band pilot frequency which does not cause any interference into the QAM signal and at the same time provides simultaneous carrier and symbol timing synchronization at the receiver, is the Nyquist frequency. By analysis, computer simulation and measurements we have evaluated the performance of a 16-QAM scheme, which employs the proposed at the Nyquist frequency pilot insertion technique. The 16-QAM system was evaluated in an AWGN channel. The obtained results have indicated that, as long as the pilot extraction bandwidth is relatively small, the 16-QAM system performance has a negligible degradation (less than 0.2 dB).

3. Motivated by the increasing need for efficient utilization of the available spectrum in the digital radio world and in hybrid analog and digital channels, such as the transmission of a CEPT-1 digital signal in a standard analog SG band, we have investigated the feasibility of two QAM modulation schemes, i.e. 512- and 1024-QAM. These schemes can achieve spectral efficiencies of up to 9 and 10 b/s/Hz, respectively. In this context, we have analyzed and evaluated the performance of these two QAM modulation formats in the presence of the most important sources of system interference, namely:

- (a) Inter-Symbol Interference (ISI),
- (b) Adjacent-Channel Interference (ACI),
- (c) Co-Channel Interference (CCI).

Our results have indicated that, under certain constraints, spectral efficiencies of up to 10 b/s/Hz are feasible.

7.2 SUGGESTIONS FOR FUTURE RESEARCH

7.2.1 Analysis of the DPS in the Presence of Other Forms of SCPJ

The analysis which was carried out in Chapter 2 assumed that the SCPJ was sinusoidal. As previously mentioned, other models which can be adopted for the SCPJ include Gaussian and Tikhonov distributions. It would be thus interesting to analyze and evaluate the performance of the Dual-Pilot QAM system assuming that the SCPJ has Gaussian or Tikhonov distribution characteristics.

7.2.2 Performance of 512- and 1024-QAM in the Presence of Multipath Fading

For terrestrial microwave, high-capacity radio systems one other important cause of degradation is multipath (selective) fading [96]. Thus, it is of interest to investigate the performance of 512- and 1024-QAM in such an environment, so that the outage of the systems could accurately estimated [97]. Furthermore, the use of sophisticated receivers, e.g. employing adaptive equalization, should be studied in order to reduce the outage of these systems under selective fading [98].

7.2.3 Cancellation Methods for Co-Channel and Adjacent Channel Interferences

As discussed in Chapters 5 and 6, CCI and ACI could cause significant performance degradation, especially to bandwidth efficient digital communication systems. As a result, it would be very useful to investigate methods which can be employed to combat these interferences. For example, one can investigate the use of complex transversal equalizers with or without feedback in order to reduce the effects of

interference. Furthermore, coding techniques could be employed to decrease the overall system sensitivity to these interferences.

7.2.4 Effects of Nonlinearities on the Performance of 512- and 1024-QAM Systems

Although in this thesis the impact of only linear distortions on the performance of 512- and 1024-QAM systems was investigated, almost any practical digital communication system is necessary to employ some kind of power amplification, e.g. TWT or GaAs FET, at its output stage. Most of the power amplifiers exhibit a certain amount of nonlinear distortion [101],[102]. Thus, it is important to investigate the effects of nonlinearities on the performance of 512- and 1024-QAM schemes. Furthermore, it would be very useful to find efficient techniques for adaptively linearizing these power amplifiers [103],[104].

7.2.5 Application of Trellis Coded Modulation to Digital Microwave Radio

In recent years, Trellis Coded Modulation (TCM) has become an increasingly popular means of achieving substantial gains (of up to 6 dB in an AWGN channel) as compared to uncoded schemes of the same bit rate and bandwidth efficiency [99]. This coding technique has been successfully applied to voiceband modems [31] as well as to mobile-satellite communication system [100]. In all these applications the modulation technique PSK has been employed almost exclusively. Thus, it would be interesting to investigate how TCM can be applied to spectrally efficient QAM systems and what kind of performance improvements can be achieved over their equivalent, uncoded schemes.

Bibliography

- [1] C. N. Campopiano and B. G. Glazer, "A coherent digital amplitude and phase modulation scheme," *IRE Trans. Commun. Syst.*, vol. CS-10, pp. 90-95, March 1962.
- [2] C. R. Chan, "Combined digital phase and amplitude modulation communication systems," *IRE Trans. Commun. Syst.*, vol. CS-8, pp. 150-154, Sept. 1960.
- [3] J. C. Hancock and R. W. Lucky. "Performance of combined amplitude and phase-modulated communication systems," *IRE Trans. Commun. Syst.*, vol. CS-8, pp. 232-237, Dec. 1960.
- [4] R. W. Lucky and J. C. Hancock, "On the optimum performance of N-ary systems having two degrees of freedom," *IRE Trans. Commun. Syst.*, vol. CS-10, pp. 185-192, March 1962.
- [5] G. J. Foschini *et al.*, "On the selection of a two-dimensional signal constellation in the presence of phase jitter and Gaussian noise," *Bell Syst. Tech. J.*, vol. 52, pp. 927-965, July-Aug. 1973.
- [6] M. K. Simon and J. G. Smith, "Hexagonal multiple phase-and-amplitude-shifted-keyed signal sets," *IEEE Trans. Commun.*, vol. COM-21, pp. 1108-1115, Oct. 1973.
- [7] G. J. Foschini *et al.*, "Optimization of two-dimensional signal constellations in the presence of Gaussian noise," *IEEE Trans. Commun.*, vol. COM-22, pp. 28-38, Jan. 1974.
- [8] C. M. Thomas *et al.*, "Digital amplitude-phase keying with M-ary alphabets," *IEEE Trans. Commun.*, vol. COM-22, pp. 168-180, Feb. 1974.

- [9] M. Borge, "Comparison of high-level modulation schemes for high-capacity digital radio systems," *IEEE Trans. Commun.*, vol. COM-33, pp. 442-449, May 1985.
- [10] K. Feher, *Digital Communications: Microwave Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [11] J. G. Proakis, *Digital Communications*, McGraw-Hill, NY, 1983.
- [12] A. Lender, "Correlative (Partial response) Techniques and Applications to Digital Radio Systems," in K. Feher, *Digital Communications: Microwave Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [13] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. COM-23, pp. 921-934, Sept. 1975.
- [14] K. Feher, *Advanced Digital Communication Systems and Signal Processing Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [15] K. Feher *et al.*, "On 6.67 b/s/Hz 256-QAM and 225-QPRS modems for T1/SG data-in-voice (DIV) applications," in *Proc. ICC '85*, pp. 15.4.1-15.4.7.
- [16] K. T. Wu, "On spectrally efficient digital transmission systems", *Ph. D. Thesis*, Department of Electrical Engineering, University of Ottawa, April 1986.
- [17] H. Sinnreich and D. Stom, "System parameters and performance of high speed modems on analog radio," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 342-348, April 1987.
- [18] J. W. Chamberlain *et al.*, "Design and field test of the 256-QAM DIV modem," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 349-356, April 1987.
- [19] B. E. Collins *et al.*, "Application of coded modulation to 1.544 Mbit/s data-in-voice modems for FDM FM and SSB analog radio systems," *IEEE J. Select. Areas Commun.*, pp. 369-377, April 1987.
- [20] Y. Nakamura *et al.*, "256 QAM modem for multicarrier 400 Mbit/s digital radio," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 329-335, April 1987.

- [21] Y. Saito and Y. Nakamura, "256-QAM for high capacity digital radio systems," *IEEE Trans. Commun.*, vol. COM-34, pp. 799-805, Aug. 1986.
- [22] Y. Saito *et al.*, "Multilevel QAM modulation techniques for digital microwave radios," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 336-341, April 1987.
- [23] E. Fukuda and Y. Daido, "256-QAM digital radio system with a roll-off factor of 20 percent for attaining 6.75 bit/s/Hz," in the *Proceedings of ICC '87*, Seattle, U.S.A., pp. 1798-1802, June 1987.
- [24] H. Ichikawa *et al.*, "256-QAM multicarrier 400 Mb/s microwave radio system field tests," in the *Proceedings of GLOBECOM '87*, Tokyo, Japan, pp. 1803-1808, November 1987.
- [25] M. Tahara *et al.*, "6 GHz 140 Mbps digital radio system with 256-SSQAM modulation," in the *Proceedings of GLOBECOM '87*, Tokyo, Japan, pp. 1495-1500, November 1987.
- [26] B. Godfrey *et al.*, "RD-6B: On the world's wavelength," *Telesis*, vol. 14, no 2, 1987.
- [27] K. Vogel, "Orthogonal cochannel operation for high spectrum efficiency based on the example of 16-QAM - 140 Mbit/s using rolloff 0.19," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 321-328, April 1987.
- [28] K. T. Wu and K. Feher, "Multilevel PRS/QPRS above the Nyquist rate," *IEEE Trans. Commun.*, vol. COM-33, pp. 735-739, July 1985.
- [29] J. C. Y. Huang *et al.*, "Advances in digital communication by radio: Guest Editorial," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 317-320, April 1987.
- [30] *General Catalog*, 1987, Karkar Electronics, Inc., A unit of Signal General, San Francisco, CA 94103, USA. (See also U.S. Patent 3 271 705.)
- [31] G. D. Forney, Jr. *et al.*, "Efficient modulation for band-limited channels," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 632-647, Sept. 1984.

- [32] K. Feher, "1024-QAM and 256-QAM coded modems for microwave and cable systems applications," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 357-368, April 1987.
- [33] J. J. Spilker, *Digital Communication by Satellite*, Englewood Cliffs, NJ: Prentice-Hall, 1977.
- [34] W. P. Robins, *Phase Noise in Signal Sources*, Peter Peregrinus Ltd., London, UK, 1982.
- [35] C. M. Merigold, "Phase Noise Measurements," in K. Feher and Engineers of Hewlett-Packard, *Telecommunication Measurements, Analysis and Instrumentation*, Prentice Hall, 1987, Chapter 11.
- [36] M. K. Simon and C. C. Wang, "Differential detection of Gaussian MSK in a mobile radio environment," *IEEE Trans. Veh. Technol.*, vol. VT-33, pp. 307-320, Nov. 1984.
- [37] W. C. Lindsey, *Synchronization Systems in Communications*, Englewood Cliffs, NJ: Prentice-Hall, 1972.
- [38] W. C. Lindsey and M. K. Simon, *Telecommunication System Engineering*, Englewood Cliffs, NJ: Prentice-Hall, 1973.
- [39] V. K. Bhargava, D. Haccoun, R. Matyas and P. P. Nuspl, *Digital Communications by Satellite: Modulation, Multiple Access and Coding*, J. Wiley & Sons, Inc., 1981.
- [40] F. M. Gardner, "Self-Noise in Synchronizers," *IEEE Trans. Commun.*, vol. COM-28, pp. 1159-1163, August 1980.
- [41] F. M. Gardner, "Hang-up in phase-lock loops," *IEEE Trans. Commun.*, vol. COM-25, pp. 1210-1214, October 1977.
- [42] R. W. Lucky, J. Salz and E. J. Weldon, Jr., *Principles of Data Communication*, McGraw-Hill, 1965.
- [43] W. R. Bennett and J. R. Davey, *Data Transmission*, McGraw-Hill, 1965.
- [44] P. L. Zador, "Error probabilities in data system pulse regenerator with DC restoration," *Bell Syst. Techn. J.*, pp. 979-984, July-Aug. 1966.

- [45] M. A. Aaron and M. K. Simon, "Approximation of the error probability in a regenerative repeater with quantized feedback," *Bell Syst. Techn. J.*, pp. 1845-1847, Dec. 1966.
- [46] H. Kaneko and A. Sawai, "Feedback balanced code for multilevel PCM transmission," *IEEE Trans. Commun. Techn.*, vol. COM-17, pp. 554-563, Oct. 1969.
- [47] Y. Daido *et al.*, "Multilevel QAM modulation techniques for digital microwave radios," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 336-341, April 1987.
- [48] P. S.-K. Leung and K. Feher, "Block-Inversion-Coded QAM systems," *IEEE Trans. Commun.*, vol. COM-36, pp. 797-805, July 1988.
- [49] J. Orton and K. Feher, "A new channel coding algorithm for in-band spectral suppression in PAM and QAM systems," in the *Proceedings of ICC '88*, Philadelphia, U.S.A., pp. 657-661, June 1988.
- [50] B. B. Lusignan, "Single-sideband transmission for land mobile radio," *IEEE Spectrum*, pp. 33-37, July 1978.
- [51] J. P. McGeehan and D. F. Burrows, "Performance limits of feed-forward automatic gain control in mobile radio receivers," *IEE Proceedings, Part F*, pp. 385-392, June 1981.
- [52] K. W. Leland and N. R. Sollenberger, "Impairment mechanisms for SSB mobile communications at UHF with pilot-based Doppler/fading correction," *Bell System Techn. J.*, vol. 59, pp. 1923-1942.
- [53] J. P. McGeehan and A. J. Bateman, "Theoretical and experimental investigation of feed-forward signal regeneration as a means of combating multipath propagation effects in pilot-based SSB mobile radio systems," *IEEE Trans. Veh. Technol.*, pp. 106-120, Feb. 1983.
- [54] A. J. Bateman and J. P. McGeehan, "Phase-locked transparent tone-in-band (TTIB): A new spectrum configuration particularly suited to the transmission of data over SSB mobile radio networks," *IEEE Trans. Commun.*, vol. COM-32, pp. 81-87, Jan. 1984.

- [55] A. J. Bateman *et al.*, "Speech and data communications over 942 MHz TAB and TIIB single sideband mobile radio systems incorporating feed-forward signal regeneration," *IEEE Trans. Veh. Technol.*, vol. VT-34, pp.13-21, Feb. 1985.
- [56] M. K. Simon, "Dual-pilot tone calibration technique," *IEEE Trans. Veh. Technol.*, vol. VT-35, pp. 63-70, May 1986.
- [57] F. Davarian, "Mobile digital communications via tone calibration," *IEEE Trans. Veh. Technol.*, vol. VT-36, pp. 55-61, May 1987.
- [58] W. Rafferty *et al.*, "Laboratory measurements and a theoretical analysis of the TCT fading channel radio system," *IEEE Trans. Commun.*, vol. COM-35, pp. 172-180, February 1987.
- [59] P. Mathiopoulos, "On quadrature amplitude modulated (QAM) bandwidth efficient digital transmission systems," *Ph.D. Candidacy Paper*, University of Ottawa, Department of Electrical Engineering, Ottawa, June 1986.
- [60] P. Mathiopoulos, H. Ohnishi and K. Feher, "A study of 1024-QAM modem performance in distorted linear channels and a method of cancelling sinusoidal phase jitter," in the *Proceedings of MONTECH '86*, Montreal, Canada, pp. 246-249, October 1986.
- [61] P. Mathiopoulos and K. Feher, "A new technique for system caused phase jitter cancellation," Submitted to *ICC '87*, October 11, 1986.
- [62] P. Mathiopoulos, H. Ohnishi and K. Feher, "A study of 1024-QAM system performance in the presence of filtering imperfections," *IEE Proceedings, Part I*, vol. 136, pp. 175-179, April 1989.
- [63] P. Mathiopoulos and K. Feher, "A new technique for system caused phase jitter cancellation," in the *Proceedings of ICC '87*, Seattle, U.S.A., pp. 1451-1457, June 1987.
- [64] P. Mathiopoulos and K. Feher, "In-band pilot insertion at the Nyquist frequency for transparent signalling," in the *Proceedings of GLOBECOM '87*, Tokyo, Japan, pp. 2063-2068, November 1987.

- [65] D. D. Falconer, "Jointly Adaptive Equalization and Carrier Recovery in Two-Dimensional Digital Communication Systems," *Bell Syst. Tech. J.*, vol. 55, pp. 317-334, March 1976.
- [66] Members of the Technical Staff Bell Telephone Laboratories, *Transmission Systems for Communications*, Bell Telephone Laboratories, Technical Publications Winston-Salem, North Carolina, 1971.
- [67] K. Pahlavan and J. L. Holsinger, "Expanded TCM for channels with multiplicative noise," in the *Proceedings of ICC '87*, Seattle, U.S.A., pp. 404-408, June 1987.
- [68] W. J. Weber, III, "Differential encoding for multiple amplitude and phase shift keying systems," *IEEE Trans. Commun.*, vol. COM-26, pp. 385-391, March 1978.
- [69] A. J. Zverev, *Handbook of Filter Synthesis*, J. Wiley & Sons, Inc., New York, 1967.
- [70] H. Y.-F. Lam, *Analog and Digital Filters: Design and Realization*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1979.
- [71] *Crystal/LC Filters Catalog*, 1986, Microsonics Incorporated, 60 Winter St., Wegmouth, MA 02188, USA.
- [72] *Crystals, Crystal Filters, Crystal Oscillators and Crystal Discriminators Catalog*, 1987, K&L Quartztek, 20 South 48th Avenue, Phoenix, Arizona 85043, USA.
- [73] *Filters, SAW Delay Lines and Oscillators Catalog*, 1987, Damon-Electronics Division, 80 Wilson Way, Westwood, MA 02090, USA.
- [74] *LC & Crystal Filters Catalog*, 1987, PC Monitor Products, P.O. Box 1966, Osceanside, CA 92054, USA.
- [75] J. G. Smith, "Odd-bit quadrature amplitude-shift keying," *IEEE Trans. Commun.*, vol. COM-23, pp. 385-389, March 1975.
- [76] K. T. Wu and K. Feher, "256-QAM modem performance in distorted channels," *IEEE Trans. Commun.*, vol. COM-33, pp. 487-491, May 1985.
- [77] E. D. Sunde, *Communication Systems Engineering Theory*, J. Wiley & Sons, Inc., 1969.

- [78] J.-D. Wang and H. J. Trussel, "Adaptive harmonic noise cancellation with an application to distribution power line communication," *IEEE Trans. Commun.*, vol. COM-36, pp. 875-884, July 1988.
- [79] R. M. Vines *et al.*, "Noise on residential power distribution circuits," *IEEE Trans. Electromagn. Compat.*, vol. EMC-26, pp. 161-168, Nov. 1984.
- [80] H. J. Trussell and J.-D. Wang, "Cancellation of harmonic noise in distribution line communication," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 3338-3344, Dec. 1985.
- [81] Bell System Technique Reference, *Data Communications Using Telecommunication Network*, PUB 41005, May 1971.
- [82] Y. Saito *et al.*, "Feasibility considerations of high-level multi-carrier system," in the *Proceedings of ICC '84*, Amsterdam, The Netherlands, pp. 656-661, June 1984.
- [83] V. K. Prabhu, "The detection efficiency of 16-QAM," *Bell Syst. Techn. J.*, vol. 59, pp. 639-656, April 1980.
- [84] V. K. Prabhu, "Co-channel interference immunity of high capacity QAM," in the *Proceedings of ICC '81*, Denver, USA, pp. 68.8.1-68.8.4, June 1981.
- [85] K. T. Wu and K. Feher, "Performance of 64-QAM, 225-QPRS and 256-QAM in a cochannel interference environment," *IEE Proceedings*, vol. 133, Pt. F, pp. 576-581, October 1986.
- [86] M. Borge, "A new class of adaptive cross-polarization interference cancellers for digital radio systems," *IEEE J. Select. Areas Commun.*, vol. SAC-5, April 1987, pp. 484-492.
- [87] H. Matsue *et al.*, "Digitalized cross-polarization interference canceller for multilevel digital radio," *IEEE J. Select. Areas Commun.*, vol. SAC-5, April 1987, pp. 493-501.
- [88] J. W. Carlin *et al.*, "An IF cross-polarization canceller for microwave radio systems," *IEEE J. Select. Areas Commun.*, vol. SAC-5, April 1987, pp. 502-514.
- [89] K. Feher, *Digital Communications: Satellite/Earth Station Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

- [90] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.
- [91] B. T. Bynum and E. W. Allen, "135 Mb/s - 6 GHz transmission system design considerations," in the *Proceedings of ICC '83*, Boston, U.S.A., pp. 1478-1483, June 1983.
- [92] K. Otremba *et al.*, "High capacity digital radio system for the 18 GHz band," in the *Proceedings of GLOBECOM '87*, Tokyo, Japan, pp. 1485-1489, November 1987.
- [93] S. Nakatani *et al.*, "Performance of a long overwater span digital microwave radio system," in the *Proceedings of GLOBECOM '87*, Tokyo, Japan, pp. 1510-1514, November 1987.
- [94] H. Matsue *et al.*, "256-QAM 400 Mb/s microwave radio system with DPS countermeasures," in the *Proceedings of ICC '88*, Philadelphia, USA, pp. 1362-1367, June 1988.
- [95] P. Mathiopoulos and K. Feher, "Error rate considerations for 256-QAM systems in the presence of adjacent channel interference," in the *Proceedings of ICC '88*, Philadelphia, USA, pp. 1064-1068, June 1988.
- [96] W. D. Rummier *et al.*, "Multipath fading channel models for microwave digital radio," *IEEE Communications Mag.*, vol. 24, pp. 30-42, November 1986.
- [97] L. J. Greenstein and M. Shafi, "Outage calculation methods for microwave digital radio," *IEEE Communications Mag.*, vol. 25, pp. 30-39, Feb. 1987.
- [98] J. K. Chamberlain *et al.*, "Receiver techniques for microwave digital radio," *IEEE Communications Mag.*, vol. 24, pp. 43-54, Nov. 1986.
- [99] G. Ungerboeck, "Channel coding with multilevel/multiphase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 56-67, Jan. 1982.
- [100] D. Divsalar and M. K. Simon, "Trellis coded modulation for 4800-9600 bits/s transmission over a fading mobile satellite channel," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 162-175, Feb. 1987.

- [101] H. Balshem *et al.*, "Advanced RF power amplifiers for high capacity digital radio systems," in the *Proceedings of ICC '87*, Seattle, U.S.A., pp. 28-33, June 1987.
- [102] S. Pupolin and L. J. Greenstein, "Performance analysis of digital radio links with nonlinear transmit amplifiers," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 534-546, April 1987.
- [103] J. Namiki *et al.*, "An automatically controlled predistorter for multilevel quadrature amplitude modulation," *IEEE Trans. Commun.*, vol. COM-31, pp. 707-712, May 1983.
- [104] A. A. M. Saleh and J. Salz, "Adaptive linearization of power amplifiers in digital radio systems," *Bell Syst. Techn. J.*, vol. 62, pp. 1019-1033, April 1983.
- [105] P. Mathiopoulos and K. Feher, "Pilot aided techniques for system caused phase jitter cancellation," *IEEE Trans. on Broadcasting*, vol. 34, pp. 356-366, Sept. 1988.
- [106] M. J. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 153-170, Jan. 1984.
- [107] P. Mathiopoulos and K. Feher, "Performance evaluation of a 512-QAM system in distorted channels," *IEE Proceedings, Part F*, vol. 133, pp. 199-204, April 1986.
- [108] P. Mathiopoulos and K. Feher, "512- and 1024-QAM system performance in the presence of co-channel interference," in the *Proceedings of MIAMI TECHNICON '87*, Miami, USA, pp. 47-77, October 1987.
- [109] P. Mathiopoulos and K. Feher, "Adjacent channel interference considerations for 1024- and 512-QAM systems," in the *Proceedings of MONTECH '87*, Montreal, Canada, pp. 47-50, November 1987.

Appendix A

16-QAM MODEM DETAILS

In this appendix the implemented prototype 16-QAM modem is described and its detailed circuit diagrams are documented.

The experimental 16-QAM modem consists of the following five cards:

- (a) CARD #1: *2 x 4-PAM Encoder*
- (b) CARD #2: *Quadrature Modulator and Pilot Insertion*
- (c) CARD #3: *Carrier and Symbol Timing Recovery*
- (d) CARD #4: *Quadrature Demodulator*
- (e) CARD #5: *2 x 4-PAM Decoder*

All cards consist of a Vector plugboard 4610 construction and are plugged into a module cage (Vector-Pak) which can accommodate up to eight cards. A drawing of the front panel of the modem is shown in Fig. A.1. All external connections are made from the front panel using BNC connectors. For the reader's convenience the used connectors are numbered and their function is briefly described in Table A.1. Finally, the detailed schematic (circuit) diagrams of each card are shown in the following figures:

CARD #1: Figures A.2 and A.3,
CARD #2: Figure A.4,

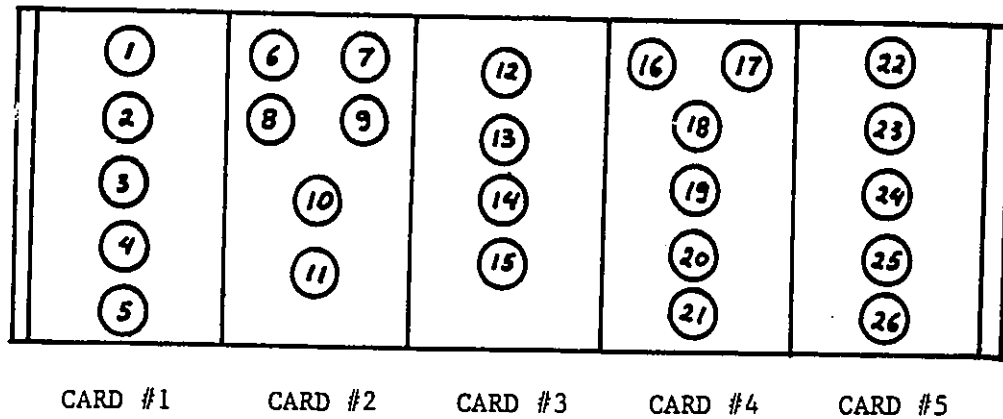


Figure A.1: Front panel of the 16-QAM modem. It consists of the following five cards:

- (1) CARD #1: 2 x 4-PAM Encoder
- (2) CARD #2: Quadrature Modulator and Pilot Insertion
- (3) CARD #3: Carrier and Symbol Timing Recovery
- (4) CARD #4: Quadrature Demodulator
- (5) CARD #5: 2 x 4-PAM Decoder.

A brief description of the shown numbered connectors is given in Table A.1. For a block diagram of the modem is given in Fig. 2.11.

CONNECTOR #	SIGNAL DESCRIPTION	INPUT/OUTPUT
1	Binary Data	Input
2	Master Clock (3.2 MHz)	Input
3	Bit Clock (800 kHz)	Output
4	4-PAM (Q-Channel)	Output
5	4-PAM (I-Channel)	Output
6	Pilot Signal	Output
7	Pilot Signal	Input
8	Filtered 4-PAM (I-Channel)	Input
9	Filtered 4-PAM (Q-Channel)	Input
10	Carrier (1.5 MHz)	Input
11	16-QAM Modulated Signal	Output
12	Extracted Pilot ($f_c + f_p$)	Input
13	Extracted Pilot ($f_c - f_p$)	Input
14	Recovered Carrier (f_c)	Output
15	Recovered Symbol Timing Signal	Output
16	Demodulated 4-PAM (I-Channel)	Output
17	Demodulated 4-PAM (Q-Channel)	Output
18	Recovered Carrier	Input
19	16-QAM Modulated Signal	Input
20	To Card #3	Output
21	Monitor of the Spectrum	Output
22	Demodulated, Filtered 4-PAM (I-Channel)	Input
23	Detected, Binary Data	Output
24	Demodulated, Filtered 4-PAM (Q-Channel)	Input
25	Monitor of the eye-diagram (I-Channel)	Output
26	Monitor of the eye-diagram (Q-Channel)	Output

Table A.1: Description of the various signals at the different connectors of the 16-QAM modem.

CARD #3: Figure A.5,
CARD #4: Figure A.6,
CARD #5: Figures A.7 and A.8.

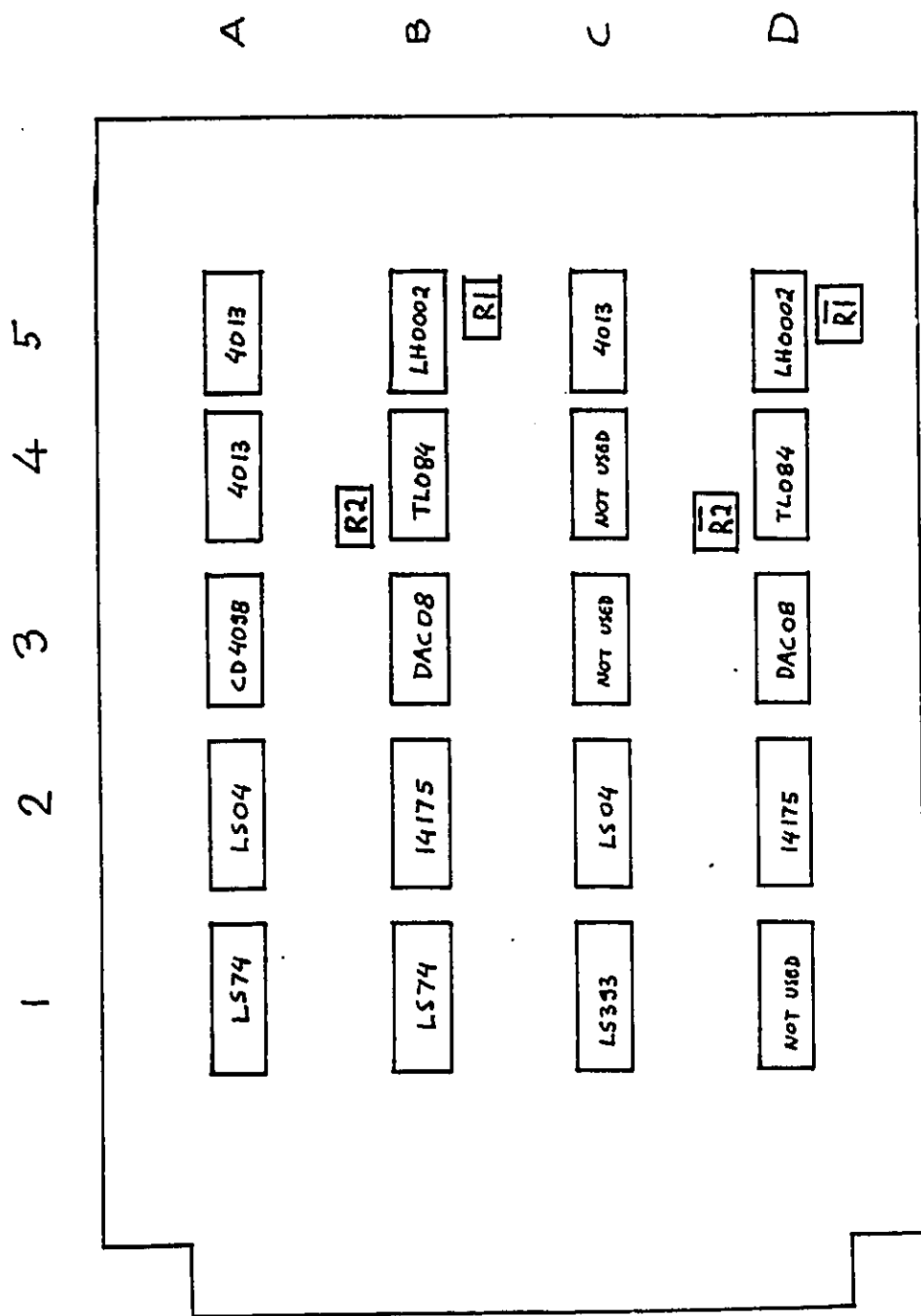


Figure A.2: Assembly drawing of the 2 x 4-PAM encoder.



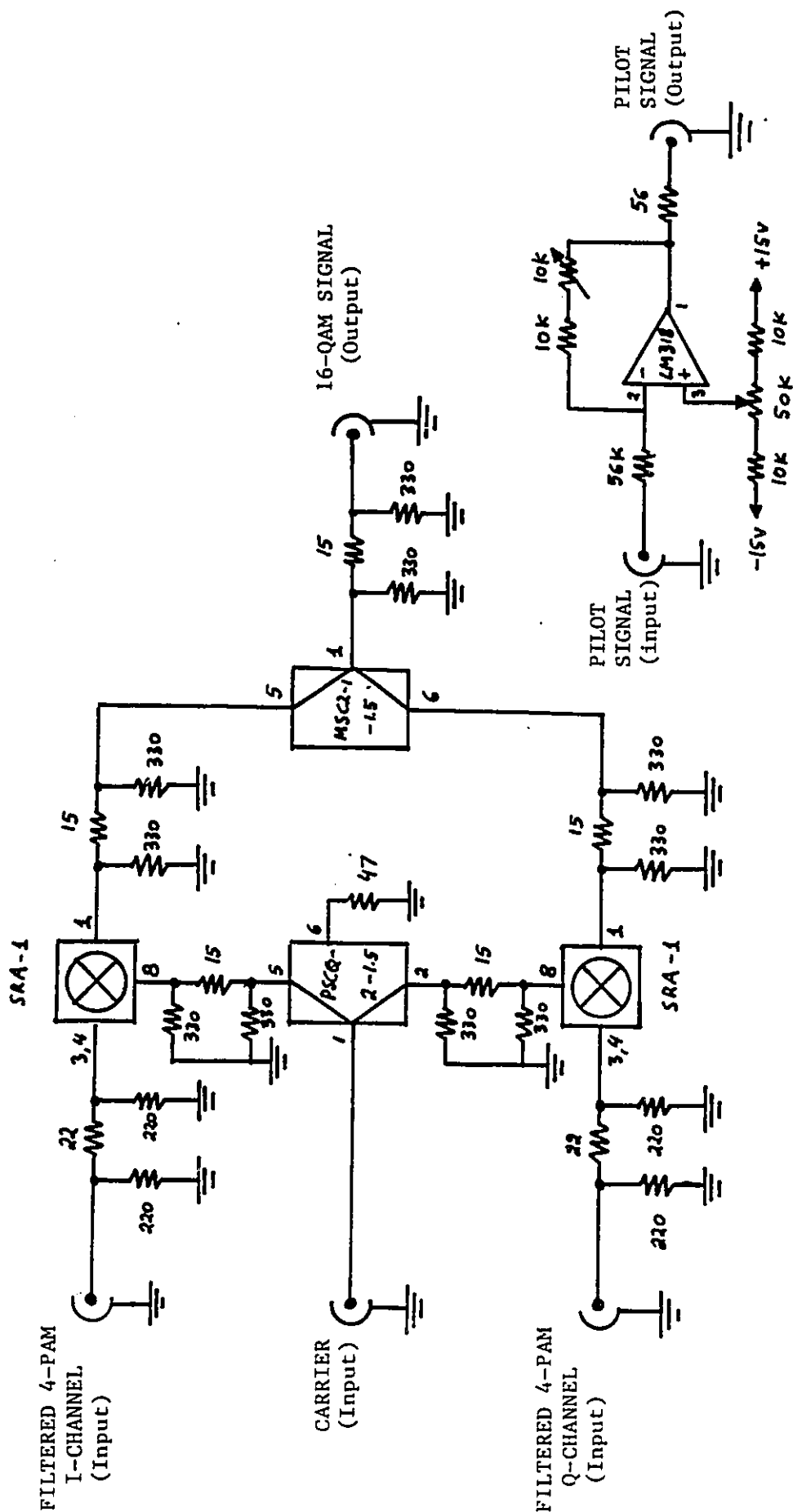
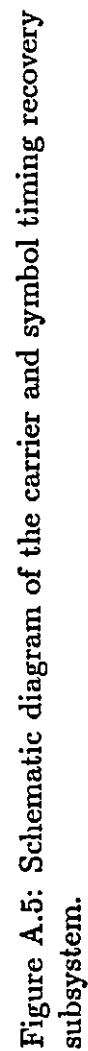
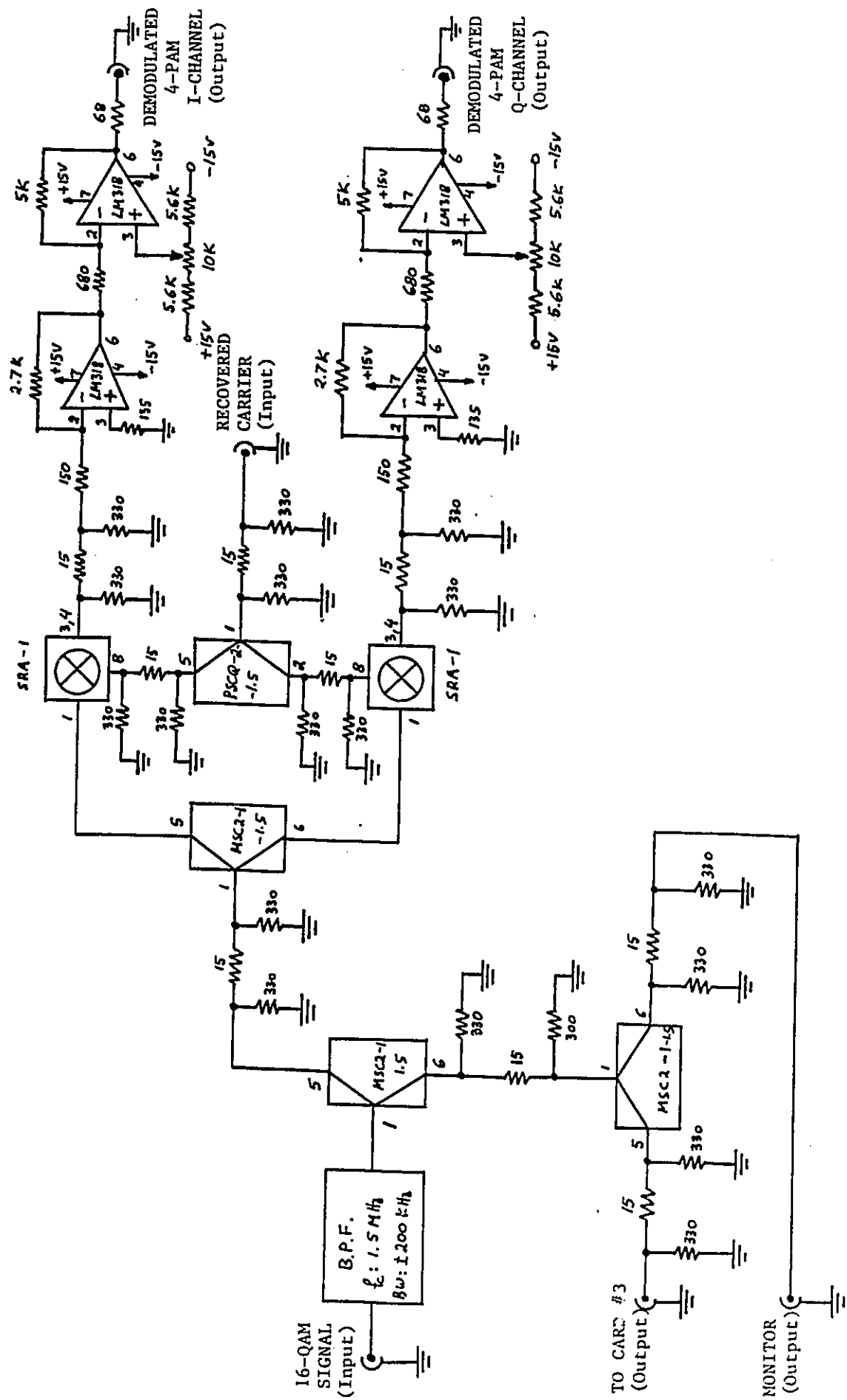


Figure A.4: Schematic diagram of the quadrature modulator and the pilot insertion.





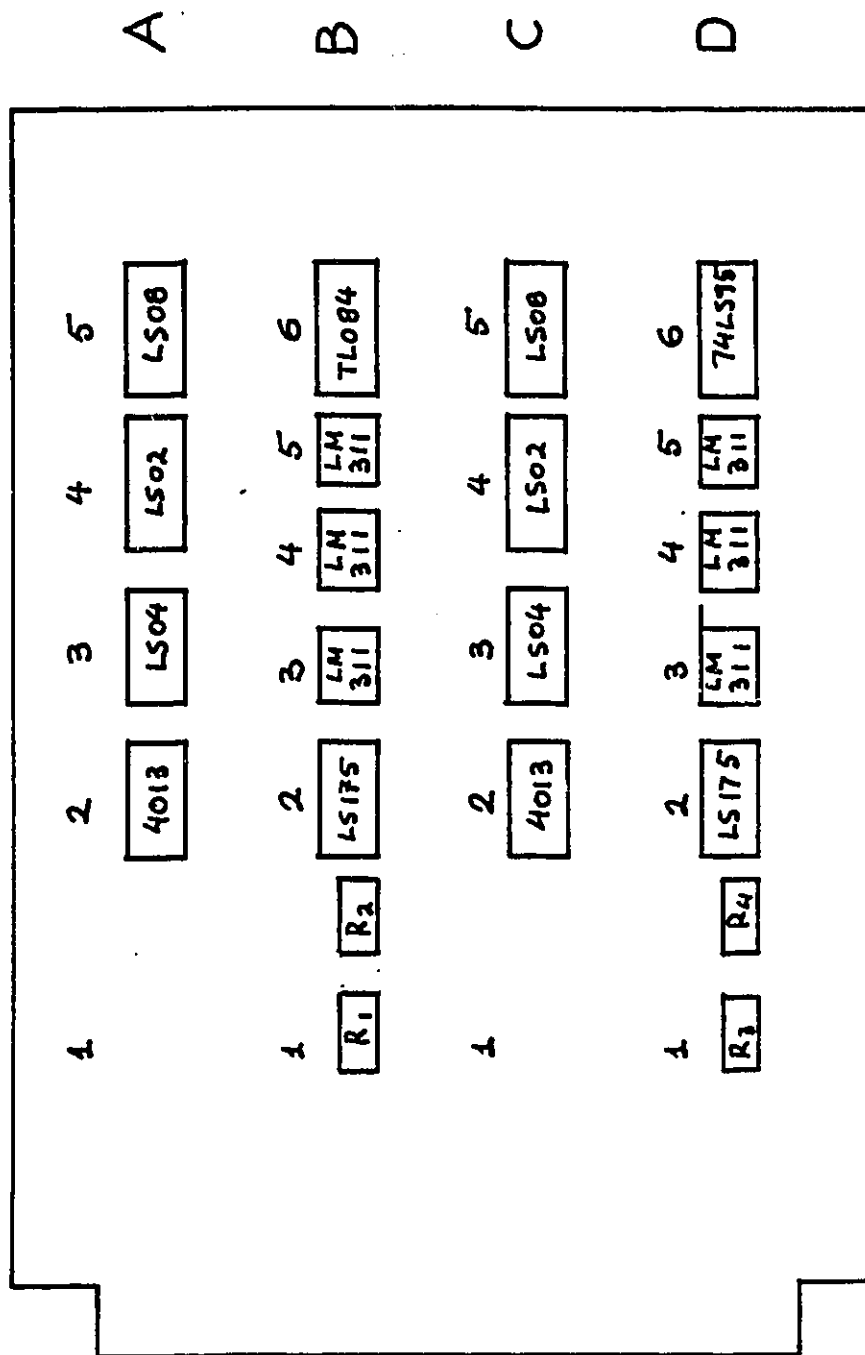


Figure A.7: Assembly drawing of the 2 x 4-PAM decoder.

Appendix B

COMPUTER SIMULATION PROGRAM LISTINGS

The computer simulation programs listed in the appendix were used to evaluate the performances of various pilot based and baseband QAM system, operated in the presence of sinusoidal SCPJ, distorted linear channels, CCI and ACI.

The author of this thesis used, modified and extended numerous sub-routines and programs which were developed by the Digital Communication Research Group of the Department of Electrical Engineering at the University of Ottawa.

```
C***** 51S00010
C*      * 51S00020
C*      * 51S00030
C*      * 51S00040
C*      * 51S00050
C*      * 51S00060
C*      * 51S00070
C*      * 51S00080
C*      * 51S00090
C*      * 51S00100
C***** 51S00110
C      * 51S00120
C      * 51S00130
C      * 51S00140
C      * 51S00150
C      * 51S00160
C      * 51S00170
C      * 51S00180
C      * 51S00190
C      * 51S00200
C      * 51S00210
C      * 51S00220
C      * 51S00230
C      * 51S00240
C      * 51S00250
C      * 51S00260
C      * 51S00270
C      * 51S00280
C      * 51S00290
C      * 51S00300
C      * 51S00310
C      * 51S00320
C      * 51S00330
C      * 51S00340
C      * 51S00350
C      * 51S00360
C      * 51S00370
C      * 51S00380
C      * 51S00390
C      * 51S00400
C      * 51S00410
C      * 51S00420
C      * 51S00430
C      * 51S00440
C      * 51S00450
C      * 51S00460
C      * 51S00470
C      * 51S00480
C      * 51S00490
C      * 51S00500
C      * 51S00510
C      * 51S00520
C      * 51S00530
C      * 51S00540
C      * 51S00550
C      * 51S00560
C      * 51S00570
C      * 51S00580
C      * 51S00590
C      * 51S00600
C      * 51S00610
C      * 51S00620
C      * 51S00630
C      * 51S00640
C      * 51S00650
C      * 51S00660
C      * 51S00670
C      * 51S00680
C      * 51S00690
C      * 51S00700
C      * 51S00710
C      * 51S00720

C      FILE NAME: 51SI

C      THIS PROGRAM EVALUATES THE PERFORMANCE OF 512-QAM
C      SYSTEMS IN THE PRESENCE OF A CONSTANT ENVELOPE
C      CO-CHANNEL INTERFERER (SINUSOIDAL INTERFERER)

C      COMPLEX DATA(524288),CCI(524288),TF(524288)
C      COMPLEX XSIGNL(524288),BUFF(524288)
C      DIMENSION PEI(50),EBNO(50)
C      DIMENSION NI(8192),NQ(8192),NNI(8192),NNQ(8192)
C      DIMENSION XARRAY(150),YARRAY(150)
C      INTEGER IR(8192)
C      REAL IBAUD,IMAX,QMAX

C      INITIALIZE PROGRAM.

C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
C      COMMON IMAX,QMAX,EBNO1,RK,CN1

C      FOR 512-QAM, NBPSYM= 9, EBNO1=18., CN1=24., CDIM=40.0

C      NBPSYM= 9
C      EBNO1=18.
C      CN1=24.
C      CDIM=45.0
C      ATM=45.0
C      NLEVL=24
C      IMAX=FLOAT(NLEVL)-1.
C      QMAX=IMAX
C      RK=FLOAT(NBPSYM)
C      KQAM=512

C      ALPHA=0.2

C      IND=1 FOR PLOTTING THE EYE-DIAGRAM
C      IND=2 FOR PLOTTING THE P(E)

C      IND=2
C      NRUNS=1
C      IOFF=0
C      LSAMPL=64
C      NSNR=45
C      NSYMB=8192
C      LDIM=LSAMPL*NSYMB

C      THE BAUD RATE BUAD IS GIVEN IN KBAUD/S

C      BAUD=177.8
C      SBANDW=BAUD*LSAMPL
C      NO1=LDIM/2+1
C      NO2=NO1+1
C      FBW=BAUD/2.
C      PI=3.14159

C      END OF INITIALIZATION.

C      START OF COMPUTATIONS.

C      THETA = 0.240*PI
C      WRITE(6,14) THETA
14  FORMAT(1X,'THETA=',F10.4,2X,'RADS'////)
92  WRITE(6,245) KQAM
245 FORMAT(5X,15,'-QAM',/)
    WRITE(6,45) ALPHA
45  FORMAT(////,2X,'*****ALPHA=',F6.2)
```

```

DSEED = 135791.DO
CALL RAND (NI, NSYMB, DSEED)
DSEED = 780958.DO
CALL RAND (NQ, NSYMB, DSEED)
CALL SIGNAL (XSIGNL, NI, NQ)
C
C CALL RCOSTX(TF)
C
C CALL FILTER (XSIGNL, TF)
C
C-----
C
C 70 DO 101 MM=1, 1
C
C   WRITE(6,102) THETA
102 FORMAT(1X,'THETA=',F11.4,2X,'RADS'///)
C
C   DO 100 NR=1,NRUNS
C
C-----MAIN CHANNEL-----
C
C   DO 40 I=1,LDIM
40 DATA(I)=XSIGNL(I)
CALL ENERGY(DATA,C,EB)
IF(NR.EQ.1) GO TO 88
CDI =CDIM-(NR-2)*2.5
C-----CO-CHANNEL-----
AM=SQRT(2.*C*10**(-0.1*CDI))
DO 177 K=1,LDIM
177 DATA(K)=DATA(K)+AM*CMPLX(COS(THETA),-SIN(THETA))
WRITE(6,199) CDI
199 FORMAT(5X,'C/I=',F5.1,'DB')
C
C-----
C
C   END OF THE TRANSMITTER
C-----
C
C 88 CONTINUE
82 CALL RCOSRX(TF)
83 CALL HHGG(TF,PNOISE)
CALL FILTER(DATA,TF)
C
C   GO TO (26,27), IND
27 CALL SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
CALL DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PEI,EB)
CCC IF((NERROR.NE.0).AND.(NR.EQ.NRUNS)) CALL PLOT(0.0,0.0,999)
IF (NERROR.NE.0) GO TO 100
DO 99 I=1,NSNR
XARRAY(I)=EBNO(I)
99 YARRAY(I)=PEI(I)
A=1.
CALL DRAWCN(XARRAY,YARRAY,NR,NRUNS,A)
C CALL DRAWEB(XARRAY,YARRAY,JCURV,JLAST)
GO TO 100
26 CALL EYEQ(DATA)
100 CONTINUE
THETA = THETA + 0.02*PI
101 CONTINUE
STOP
END
C*****
C   SUBROUTINE:  SIGNAL
C
C   THIS SUBROUTINE FORMS A COMPLEX BASEBAND SIGNAL
C*****
C   SUBROUTINE SIGNAL(DATA,NI,NQ)
DIMENSION NI(1),NQ(1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
DO 1 I=1, NSYMB
IF ((NI(I) .EQ. NQ(I)) .AND. (NI(I) .EQ. 23)) NI(I)=15
IF ((NI(I) .EQ. NQ(I)) .AND. (NI(I) .EQ. 21)) NI(I)=13

```

```

51S00730
51S00740
51S00750
51S00760
51S00770
51S00780
51S00790
51S00800
51S00810
51S00820
51S00830
51S00840
51S00850
51S00860
51S00870
51S00880
51S00890
51S00900
51S00910
51S00920
51S00930
51S00940
51S00950
51S00960
51S00970
51S00980
51S00990
51S01000
51S01010
51S01020
51S01030
51S01040
51S01050
51S01060
51S01070
51S01080
51S01090
51S01100
51S01110
51S01120
51S01130
51S01140
51S01150
51S01160
51S01170
51S01180
51S01190
51S01200
51S01210
51S01220
51S01230
51S01240
51S01250
51S01260
51S01270
51S01280
51S01290
51S01300
51S01310
51S01320
51S01330
51S01340
51S01350
51S01360
51S01370
51S01380
51S01390
51S01400
51S01410
51S01420
51S01430
51S01440

```

```

1 CONTINUE
DO 2 I=1,NSYMB
C   LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL
J1=(I-1)*LSAMPL+1
J2=I*LSAMPL
DO 3 J3=J1,J2
3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))
2 CONTINUE
RETURN
END
C*****
C   SUBROUTINE: RCOSTX
C
C   THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C   RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A
C   X/SINX APERTURE EQUALIZER
C*****
SUBROUTINE RCOSTX(TF)
  COMPLEX TF( 1)
  COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
  COMMON /NUMB2/ NSNR,NSYMB,BAUD
  SBANDW=BAUD*LSAMPL
  NO=LDIM/2
  NO1=NO+1
  IF (ALPHA.EQ.0) ALPHA=0.0001
  FN=LDIM*(FBW/SBANDW)
  F1=(1.-ALPHA)*FN
  F2=(1.+ALPHA)*FN
  IFN=IFIX(FN)
  IF1=IFIX(F1)+1
  IF2=IFIX(F2)+1
C
C   THE AMPLITUDE CHARACTERISTICS
C
  A1=3.1415926/(2.*FLOAT(IFN))
  DO 8 I=2,IF1
  TF(I)=CMPLX(1.0,0.0)
  J=I-1
C
C   X/SIN(X) EQUALIZATION
C
  A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))
  TF(I)=CMPLX(A2,0.0)
8 CONTINUE
  JK=IF1+1
  DO 9 J=JK,IF2
  I=J-1
C
C   ROOT OF RAISED COSINE
C
  A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))
  A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
  TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
  TF(J)=TF(J)*CMPLX(A3,0.0)
9 CONTINUE
  JH=IF2+1
  DO 10 I=JH,NO1
  TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
  NO2=NO1+1
  DO 5 I=NO2,LDIM
  TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
RETURN
END
C*****
C   SUBROUTINE: RCOSRX
C
C   THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C   RAISED COSINE FILTER WITH ARBITRARY ALPHA
C*****
SUBROUTINE RCOSRX(TF)

```

```

51S01450
51S01460
51S01470
51S01480
51S01490
51S01500
51S01510
51S01520
51S01530
51S01540
51S01550
51S01560
51S01570
51S01580
51S01590
51S01600
51S01610
51S01620
51S01630
51S01640
51S01650
51S01660
51S01670
51S01680
51S01690
51S01700
51S01710
51S01720
51S01730
51S01740
51S01750
51S01760
51S01770
51S01780
51S01790
51S01800
51S01810
51S01820
51S01830
51S01840
51S01850
51S01860
51S01870
51S01880
51S01890
51S01900
51S01910
51S01920
51S01930
51S01940
51S01950
51S01960
51S01970
51S01980
51S01990
51S02000
51S02010
51S02020
51S02030
51S02040
51S02050
51S02060
51S02070
51S02080
51S02090
51S02100
51S02110
51S02120
51S02130
51S02140
51S02150
51S02160

```

```

COMPLEX TF( 1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
SBANDW=BAUD*LSAMPL
NO=LDIM/2
NO1=NO+1
IF (ALPHA.EQ.0) ALPHA=0.0001
FN=LDIM*(FBW/SBANDW)
F1=(1.-ALPHA)*FN
F2=(1.+ALPHA)*FN
IFN=IFIX(FN)
IF1=IFIX(F1)+1
IF2=IFIX(F2)+1
C
C      AMPLITUDE CHARACTERISTIC
C
      A1=3.141592/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
      TF(I)=CMPLX(1.0,0.0)
      J=I-1
      TF(I)=CMPLX(1.,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
      I=J-1
C
C      ROOT OF RAISED COSINE
C
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
      TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
      TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: FILTER
C
C      THIS SUBROUTINE PERFORMS THE FILTERING
C      PROCESS ON A DATA SEQUENCE
C*****
      SUBROUTINE FILTER(DATA,TF)
      COMPLEX DATA(1),TF(1)
      DIMENSION IWK(20)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      CALL FFT2C(DATA,19,IWK)
      DO 1 I=1,LDIM
      1 DATA(I)=CONJG(DATA(I)*TF(I))
      CALL FFT2C(DATA,19,IWK)
      DO 2 I=1,LDIM
      2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END
C*****
C      SUBROUTINE: HHGG
C
C      THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER
C      AT THE OUTPUT OF THE RECEIVE FILTER
C*****
      SUBROUTINE HHGG(TF,PNOISE)
      COMPLEX TF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SUM=0.0
      SBANDW=BAUD*LSAMPL
      DO 1 L=1,LDIM

```

```

51S02170
51S02180
51S02190
51S02200
51S02210
51S02220
51S02230
51S02240
51S02250
51S02260
51S02270
51S02280
51S02290
51S02300
51S02310
51S02320
51S02330
51S02340
51S02350
51S02360
51S02370
51S02380
51S02390
51S02400
51S02410
51S02420
51S02430
51S02440
51S02450
51S02460
51S02470
51S02480
51S02490
51S02500
51S02510
51S02520
51S02530
51S02540
51S02550
51S02560
51S02570
51S02580
51S02590
51S02600
51S02610
51S02620
51S02630
51S02640
51S02650
51S02660
51S02670
51S02680
51S02690
51S02700
51S02710
51S02720
51S02730
51S02740
51S02750
51S02760
51S02770
51S02780
51S02790
51S02800
51S02810
51S02820
51S02830
51S02840
51S02850
51S02860
51S02870
51S02880

```

```

      HH=(CABS(TF(L)))**2
1     SUM=SUM+HH
      PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.
      WRITE (6,2) PNOISE
2     FORMAT(5X,'PNOISE=',F7.3,/)
      RETURN
      END
C*****
C     SUBROUTINE: ENERGY
C
C     THIS SUBROUTINE COMPUTES THE CARRIER POWER
C     AND THE AVERAGE BIT ENERGY
C*****
      SUBROUTINE ENERGY(DATA,C,EB)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      COMPLEX DATA(1)
      WATTS=0.
      DO 1 I=1,LDIM
        WATTS=WATTS+CABS(DATA(I))**2
1     CONTINUE
      C=WATTS/FLOAT(LDIM)/2.
      RRK=RK/2.
      EB=C/(RRK*BAUD)
      WRITE (6,40) C,EB
40    FORMAT(10X,'C=',F15.6,5X,'EB=',E15.8)
      RETURN
      END
      END
C*****
C     SUBROUTINE: SYNCRO
C
C     THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA
C*****
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      INTEGER QMAXF
      COMPLEX DATA(1),AMP
      DIMENSION NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      NERROR=0
      IF (IOFF.EQ.0) GO TO 111
      DO 6 K=1,IOFF
        XX=AIMAG(DATA(LDIM))
        DO 7 KK=2,LDIM
          DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))
6       DATA(1)=CMPLX(REAL(DATA(1)),XX)
      C
      C     SYNCHRONIZE THE RECEIVED DATA.
      C
111   NOLD=0
      NOF=0
      K=1
300   CONTINUE
      NEW=0
      DO 200 J=1,NSYMB
        THR1I=FLOAT(NI(J))-1.
        THR2I=FLOAT(NI(J))+1.
        THR1Q=FLOAT(NQ(J))-1.
        THR2Q=FLOAT(NQ(J))+1.
        J1=K+(J-1)*LSAMPL
        IF (J1.GT.LDIM) J1=J1-LDIM
        AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
        AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
        IF ((AXBAR.GT.THR1I).AND.(AXBAR.LT.THR2I)) NEW=NEW+1
        IF ((AYBAR.GT.THR1Q).AND.(AYBAR.LT.THR2Q)) NEW=NEW+1
200   CONTINUE
      IF (NOLD.GE.NEW) GO TO 399
      NOLD=NEW
      NOF=K

```

51S02890
 51S02900
 51S02910
 51S02920
 51S02930
 51S02940
 51S02950
 51S02960
 51S02970
 51S02980
 51S02990
 51S03000
 51S03010
 51S03020
 51S03030
 51S03040
 51S03050
 51S03060
 51S03070
 51S03080
 51S03090
 51S03100
 51S03110
 51S03120
 51S03130
 51S03140
 51S03150
 51S03160
 51S03170
 51S03180
 51S03190
 51S03200
 51S03210
 51S03220
 51S03230
 51S03240
 51S03250
 51S03260
 51S03270
 51S03280
 51S03290
 51S03300
 51S03310
 51S03320
 51S03330
 51S03340
 51S03350
 51S03360
 51S03370
 51S03380
 51S03390
 51S03400
 51S03410
 51S03420
 51S03430
 51S03440
 51S03450
 51S03460
 51S03470
 51S03480
 51S03490
 51S03500
 51S03510
 51S03520
 51S03530
 51S03540
 51S03550
 51S03560
 51S03570
 51S03580
 51S03590
 51S03600


```

399 K=K+1
LFU=2*NSYMB
C
C SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP
C
NSFT=LSAMPL*6
IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300
NOF=NOF-1
KT=K-2
WRITE(6,56) NOLD,NOF,KT
56 FORMAT(1X,'NO OF MAX. CORRECT SYMBOLS = ',I6,/,1X,
*'NO OF SHIFT IN SAMPLES FOR MAX. CORRECT SYMBOLS= ',I4,/,
*1X,'TOTAL NO OF SHIFT FOR SEARCHING = ',I4)
C
IF (NOF.EQ.0) GO TO 230
LO=LDIM-1
DO 250 I=1,NOF
AMP=DATA(I)
DO 240 J=1,LO
DATA(J)=DATA(J+1)
240 CONTINUE
DATA(LDIM)=AMP
250 CONTINUE
C
C OPTIMIZE THE SAMPLING INSTANT
C
230 MI=1
MQ=1
EOI=FLOAT(NSYMB) + 1.
EQ=FLOAT(NSYMB) + 1.
M=22
VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
SIGMA=SQRT(VARIAN)
MOS=LSAMPL/4
DO 80 J=1,MOS
EI=0.
EQ=0.
IERX=0
IERY=0
DO 70 K=1,NSYMB
C
C VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
J1=(K-1)*LSAMPL+J
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
INDEXI=0
AMPX=ABS(AXBAR)
AMPI=ABS(FLOAT(NQ(K)))
THR1I=AMPI - 1.0
THR2I=AMPI + 1.0
IMAXF=0
IF (AMPI.EQ.IMAX) GO TO 30
IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
GO TO 40
30 IF (AMPX.LE.THR1I) INDEXI=1
IMAXF=1
40 CONTINUE
INDEXQ=0
AMPY=ABS(AYBAR)
AMPQ=ABS(FLOAT(NQ(K)))
THR1Q=AMPQ - 1.0
THR2Q=AMPQ + 1.0
QMAXF=0
IF (AMPQ.EQ.QMAX) GO TO 50
IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
GO TO 60
50 IF (AMPY.LE.THR1Q) INDEXQ=1
QMAXF=1
60 CONTINUE
IF (INDEXI.EQ.1) EI=EI + 1.
IF (INDEXQ.EQ.1) EQ=EQ + 1.
C
C COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB

```

```

51S03610
51S03620
51S03630
51S03640
51S03650
51S03660
51S03670
51S03680
51S03690
51S03700
51S03710
51S03720
51S03730
51S03740
51S03750
51S03760
51S03770
51S03780
51S03790
51S03800
51S03810
51S03820
51S03830
51S03840
51S03850
51S03860
51S03870
51S03880
51S03890
51S03900
51S03910
51S03920
51S03930
51S03940
51S03950
51S03960
51S03970
51S03980
51S03990
51S04000
51S04010
51S04020
51S04030
51S04040
51S04050
51S04060
51S04070
51S04080
51S04090
51S04100
51S04110
51S04120
51S04130
51S04140
51S04150
51S04160
51S04170
51S04180
51S04190
51S04200
51S04210
51S04220
51S04230
51S04240
51S04250
51S04260
51S04270
51S04280
51S04290
51S04300
51S04310
51S04320

```

```

C          D1=ABS(AMPX-THR1I)
ARG=D1/(SIGMA*SQRT(2.))
C          CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C          IF (ARG.GT.12.) ARG=12.
EI=EI + ERFC(ARG)/2.
IF (IMAXF.EQ. 1) GO TO 65
D2=ABS(THR2I-AMPX)
ARG=D2/(SIGMA*SQRT(2.))
IF (ARG.GT.12.) ARG=12.
EI=EI + ERFC(ARG)/2.
65 D1=ABS(AMPY-THR1Q)
ARG=D1/(SIGMA*SQRT(2.))
IF (ARG.GT.12.) ARG=12.
EQ=EQ + ERFC(ARG)/2.
IF (QMAXF.EQ. 1) GO TO 70
D2=ABS(THR2Q-AMPY)
ARG=D2/(SIGMA*SQRT(2.))
IF (ARG.GT.12.) ARG=12.
EQ=EQ + ERFC(ARG)/2.
IERX=IERX+INDEXI
IERY=IERY+INDEXQ
70 CONTINUE
WRITE(6,81) IERX,IERY
81 FORMAT(5X,'IERX=',I5,5X,'IERY=',I5)
IF(EOI.LE.EI) GO TO 75
EOI=EI
MI=J
75 CONTINUE
IF (EQQ.LE.EQ) GO TO 80
EQQ=EQ
MQ=J
80 MOFF=IABS(MI-MQ)
WRITE(6,91) MI,MQ
91 FORMAT(5X,'MI=',I3,5X,'MQ=',I3)
IF(MOFF.NE.0) WRITE(6,90) MOFF
OFF=(FLOAT(MI) + FLOAT(MQ))/FLOAT(LSAMPL)
90 FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
#12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
WRITE(6,95) OFF
95 FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
RETURN
END
C*****
C          SUBROUTINE: DECODE
C
C          THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA
C*****
SUBROUTINE DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)
INTEGER QMAXF
COMPLEX DATA(1),AMP
DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
NERROR=0
DO 1 I=1,NSNR
1 PE(I)=0.
COMMON IMAX,QMAX,EBNO1,RK,CN1
C          CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE
C          ERRORS IN ADJACENT LEVELS ARE CONSIDERED
3 DO 2 K=1,NSYMB
J1=(K-1)*LSAMPL+MI
J2=(K-1)*LSAMPL + MQ
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.
INDEXI=0
AMPX=ABS(AXBAR)
AMPI=ABS(FLOAT(NI(K)))
THR1I=AMPI - 1.0
THR2I=AMPI + 1.0
IMAXF=0

```

51S04330
51S04340
51S04350
51S04360
51S04370
51S04380
51S04390
51S04400
51S04410
51S04420
51S04430
51S04440
51S04450
51S04460
51S04470
51S04480
51S04490
51S04500
51S04510
51S04520
51S04530
51S04540
51S04550
51S04560
51S04570
51S04580
51S04590
51S04600
51S04610
51S04620
51S04630
51S04640
51S04650
51S04660
51S04670
51S04680
51S04690
51S04700
51S04710
51S04720
51S04730
51S04740
51S04750
51S04760
51S04770
51S04780
51S04790
51S04800
51S04810
51S04820
51S04830
51S04840
51S04850
51S04860
51S04870
51S04880
51S04890
51S04900
51S04910
51S04920
51S04930
51S04940
51S04950
51S04960
51S04970
51S04980
51S04990
51S05000
51S05010
51S05020
51S05030
51S05040

```

      IF (AMPI.EQ.IMAX) GO TO 30
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30  IF(AMPX.LE.THR1I) INDEXI=1
      IMAXF=1
40  CONTINUE
      INDEXQ=0
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      QMAXF=0
      IF (AMPQ.EQ.QMAX) GO TO 50
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50  IF(AMPY.LE.THR1Q) INDEXQ=1
      QMAXF=1
60  CONTINUE
      IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1

C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.
C      THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS
C      1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE.
C      FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQSRT(2)))
C      AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +
C      1/2*ERFC(D2/(SIGMA*SQRT(2))).
      DO 4 M=1,NSNR
      EBNO(M)=EBNO1+1.0*FLOAT(M-1)
      VARIAN=PNOISE*EB*(10.**(-0.1*EBNO(M)))
      SIGMA=SQRT(VARIAN)
      D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C      CHECK IF ARG IS LARGE IN WHICH CASE PE IS INSIGNIFICANT
C      CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN
      NERI=K
      IF (INDEXI.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEI=ERFC(ARG)/2.
      IF (IMAXF.EQ.1) GO TO 100
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEI=PEI + ERFC(ARG)/2.
100  D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (INDEXQ.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEQ=ERFC(ARG)/2.
      IF (QMAXF.EQ.1) GO TO 110
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEQ=PEQ + ERFC(ARG)/2.
110  IF(PEI.LT.1.E-15) PEI=0.
      IF(PEQ.LT.1.E-15) PEQ=0.
      4  PE(M)=PE(M)+(PEI+PEQ)/2.
      2  CONTINUE
      DO 5 I=1,NSNR
      5  PE(I)=PE(I)/FLOAT(NSYMB)
C      PRINT 150
C150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)
C      WRITE (8,151) (EBNO(I),PE(I),I=1,NSNR)
C151  FORMAT(5X,F5.1,10X,E10.2)
      WRITE (8,152) NERROR
152  FORMAT(/5X,'ERRORS=',I5,/)
      GO TO 155
153  PRINT 154
154  FORMAT (10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)
      WRITE(8,157) NERI
157  FORMAT(5X,'NERI=',I5)

```

```

51S05050
51S05060
51S05070
51S05080
51S05090
51S05100
51S05110
51S05120
51S05130
51S05140
51S05150
51S05160
51S05170
51S05180
51S05190
51S05200
51S05210
51S05220
51S05230
51S05240
51S05250
51S05260
51S05270
51S05280
51S05290
51S05300
51S05310
51S05320
51S05330
51S05340
51S05350
51S05360
51S05370
51S05380
51S05390
51S05400
51S05410
51S05420
51S05430
51S05440
51S05450
51S05460
51S05470
51S05480
51S05490
51S05500
51S05510
51S05520
51S05530
51S05540
51S05550
51S05560
51S05570
51S05580
51S05590
51S05600
51S05610
51S05620
51S05630
51S05640
51S05650
51S05660
51S05670
51S05680
51S05690
51S05700
51S05710
51S05720
51S05730
51S05740
51S05750
51S05760

```

```

155 CONTINUE
RETURN
END
C*****
C   SUBROUTINE: DRAWCN
C
C   THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C   P(E) AS LOW AS 1.0E-10 AND A C/N RATIO AS HIGH AS 42 DB
C*****
SUBROUTINE DRAWCN(XARRAY,YARRAY,JCURV,JLAST,A)
DIMENSION XARRAY(1),YARRAY(1),XA(150),XX(150),YY(150)
COMMON /NUMB2/ NSNR,NSYMB,BAUD
COMMON IMAX,QMAX,EBNO1,RK,CN1
M=NSNR
BR=RK*BAUD
NBW=BAUD/A
CF=BR/NBW
DO 1 K = 1 , NSNR
XA(K)=XARRAY( K ) + 10*(ALOG10(CF))
IF(YARRAY( K).LE.1.0E-10) GOTO 3
GOTO 1
3 M= K-1
GOTO 4
1 CONTINUE
4 CONTINUE
I=M+1
II=I+1
XA(I)=9.0
XA(II)=1.0
YARRAY(I)=1.E-10
YARRAY(II)=0.4
CC GO TO 2
IF(JCURV.GT.1) GOTO 2
C
C ESTABLISH THE SURFACE AREA.
C
CALL PLOTS(57.0,27.5)
CALL FACTOR(0.85)
C
C ESTABLISH THE ORIGIN.
C
CALL PLOT(3.0,1.5,-3)
C
C DRAW THE LOGARITHMIC Y-AXIS.
C
CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,25.,
+90.,1.0E-10,.4)
C
C DRAW THE LINEAR X-AXIS.
C
CALL AXIS(0.0,0.0,9HC/N IN dB,-9,
+48.0,0.0,9.0,1.0)
C
C PLOT THE DATA IN LOG-LINEAR MODE.
C
2 CONTINUE
PRINT 150
150 FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',7X,'C/N',/)
WRITE(6,151) (XARRAY(I),YARRAY(I),XA(I),I=1,M)
151 FORMAT(5X,F5.0,10X,E10.2,10X,F5.1)
CC GO TO 45
CALL LGLIN(XA,YARRAY,M,1,4,JCURV,1)
IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
45 RETURN
END
C*****
C   SUBROUTINE: EYEQ
C
C   THIS SUBROUTINE DRAWS THE EYE DIAGRAM
C   FOR A TWO SYMBOL DURATION
C*****
SUBROUTINE EYEQ(DATA)

```

```

51S05770
51S05780
51S05790
51S05800
51S05810
51S05820
51S05830
51S05840
51S05850
51S05860
51S05870
51S05880
51S05890
51S05900
51S05910
51S05920
51S05930
51S05940
51S05950
51S05960
51S05970
51S05980
51S05990
51S06000
51S06010
51S06020
51S06030
51S06040
51S06050
51S06060
51S06070
51S06080
51S06090
51S06100
51S06110
51S06120
51S06130
51S06140
51S06150
51S06160
51S06170
51S06180
51S06190
51S06200
51S06210
51S06220
51S06230
51S06240
51S06250
51S06260
51S06270
51S06280
51S06290
51S06300
51S06310
51S06320
51S06330
51S06340
51S06350
51S06360
51S06370
51S06380
51S06390
51S06400
51S06410
51S06420
51S06430
51S06440
51S06450
51S06460
51S06470
51S06480

```

```

      DIMENSION DATA2(66),DATA3(66)
      COMPLEX DATA(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
C     ESTABLISH THE SURFACE AREA.
      CALL PLOTS(37.0,27.5)
C     ESTABLISH THE ORIGIN.
      CALL PLOT(3.0,13.0,-3)
C     WRITE THE TITLE OF THE GRAPH.
      CALL SYMBOL(10.0,12.0,0.49,16HQPSK EYE DIAGRAM,0.0,16)
C     DRAW THE TIME AXIS.
      CALL AXIS(0.0,0.0,1H,-1,24.,0.0,1.0,2.8)
C     DRAW THE AMPLITUDE AXIS.
      CALL AXIS(0.0,-8.00,1H,1,16.0,90.0,-24.,3.0)
C     PLOT THE DATA.
      JJ=2*LSAMPL
      DATA2(JJ+1)=0.0
      DATA2(JJ+2)=3.0
      DATA3(JJ+1)=1.0
      DATA3(JJ+2)=2.8
      M2=NSYMB/8
      M3=3*M2+1
      M4=4*M2-1
      DO 4 KK=1,M2
      DO 2 I=1,JJ
      II=(KK-1)*LSAMPL+I
      IF (II.GT.LDIM) II=II-(II/LDIM)*LDIM
      DATA2(I)=REAL(DATA(II))
2     DATA3(I)=FLOAT(I)
      CALL LINE(DATA3,DATA2,JJ,1,0,0)
4     CONTINUE
      CALL PLOT(0.0,0.0,999)
      RETURN
      END
C*****
C     SUBROUTINE: ATT
C
C     THIS SUBROUTINE PERFORMS THE ATTENUATION OF A COMPLEX
C     SIGNAL (AT IS THE ATTENUATION FACTOR IN DB)
C*****
      SUBROUTINE ATT(DATA,AT)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMPLEX DATA(1)
      ATTT=10.**(-AT/20.)
      DO 10 I=1,LDIM
10     DATA(I)=DATA(I)*ATTT
      RETURN
      END
C*****
C     SUBROUTINE: OFFSET
C
C     THIS SUBROUTINE SHIFTS THE DATA SEQUENCE BY IOF
C*****
      SUBROUTINE OFFSET(DATA,IOF)
      COMPLEX DATA(1),BUFF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
      J=LDIM-IABS(IOF)
      K=J+1
      IF (IOF.EQ.0) GOTO 9
      IF (IOF.LT.0) GOTO 4
      DO 1 I=1,J
1     BUFF(I)=DATA(I)
      DO 2 I=K,LDIM
2     DATA(I-J)=DATA(I)
      DO 3 I=1,J
3     DATA(I+IOF)=BUFF(I)
      GOTO 9
4     IOF=-IOF
      L=IOF+1
      DO 5 I=L,LDIM
5     BUFF(I-IOF)=DATA(I)

```

```
      DO 6 I=1,IOF                                51S07210
6     DATA(I+J)=DATA(I)                          51S07220
      DO 7 I=1,J                                  51S07230
7     DATA(I)=BUFF(I)                            51S07240
9     RETURN                                       51S07250
      END                                         51S07260
C*****                                           51S07270
C     SUBROUTINE:  PHASE                          * 51S07280
C                                           * 51S07290
C     THIS SUBROUTINE PERFORMS THE PHASE SHIFT    * 51S07300
C     OF A COMPLEX DATA SEQUENCE                * 51S07310
C*****                                           51S07320
      SUBROUTINE PHASE(DATA,PHI)                  51S07330
      COMPLEX EPS,DATA(1)                        51S07340
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 51S07350
      EPS=CMPLX(COS(PHI),-SIN(PHI))             51S07360
      DO 10 I=1,LDIM                             51S07370
10    DATA(I)=DATA(I)*EPS                       51S07380
      RETURN                                       51S07390
      END                                         51S07400
C*****                                           51S07410
C     SUBROUTINE:  RAND                          * 51S07420
C                                           * 51S07430
C     THIS SUBROUTINE GENERATES RANDOM SYMBOLS    * 51S07440
C*****                                           51S07450
      SUBROUTINE RAND(IR,N,DSEED)                 51S07460
      INTEGER IR(1)                             51S07470
      DOUBLE PRECISION DSEED                     51S07480
      DO 3000 I=1,N                             51S07490
      R=GGUBFS(DSEED)                           51S07500
      IR(I)=IFIX(R)                             51S07510
      IF (R.LE.0.041) IR(I)=23                   51S07520
      IF (R.GT.0.041) .AND. (R.LE.0.082) IR(I)=21 51S07530
      IF (R.GT.0.082) .AND. (R.LE.0.123) IR(I)=19 51S07540
      IF (R.GT.0.123) .AND. (R.LE.0.164) IR(I)=17 51S07550
      IF (R.GT.0.164) .AND. (R.LE.0.205) IR(I)=15 51S07560
      IF (R.GT.0.205) .AND. (R.LE.0.246) IR(I)=13 51S07570
      IF (R.GT.0.246) .AND. (R.LE.0.287) IR(I)=11 51S07580
      IF (R.GT.0.287) .AND. (R.LE.0.328) IR(I)=9   51S07590
      IF (R.GT.0.328) .AND. (R.LE.0.369) IR(I)=7   51S07600
      IF (R.GT.0.369) .AND. (R.LE.0.41)  IR(I)=5   51S07610
      IF (R.GT.0.41)  .AND. (R.LE.0.451) IR(I)=3   51S07620
      IF (R.GT.0.451) .AND. (R.LE.0.492) IR(I)=1   51S07630
      IF (R.GT.0.492) .AND. (R.LE.0.533) IR(I)=1   51S07640
      IF (R.GT.0.533) .AND. (R.LE.0.574) IR(I)=3   51S07650
      IF (R.GT.0.574) .AND. (R.LE.0.615) IR(I)=5   51S07660
      IF (R.GT.0.615) .AND. (R.LE.0.656) IR(I)=7   51S07670
      IF (R.GT.0.656) .AND. (R.LE.0.697) IR(I)=9   51S07680
      IF (R.GT.0.697) .AND. (R.LE.0.738) IR(I)=11  51S07690
      IF (R.GT.0.738) .AND. (R.LE.0.779) IR(I)=13  51S07700
      IF (R.GT.0.779) .AND. (R.LE.0.82)  IR(I)=15  51S07710
      IF (R.GT.0.82)  .AND. (R.LE.0.861) IR(I)=17  51S07720
      IF (R.GT.0.861) .AND. (R.LE.0.902) IR(I)=19  51S07730
      IF (R.GT.0.902) .AND. (R.LE.0.943) IR(I)=21  51S07740
      IF (R.GT.0.943) .AND. (R.LE.0.984) IR(I)=23  51S07750
      IF (R.GT.0.984) IR(I)=23                   51S07760
3000  CONTINUE                                  51S07770
      RETURN                                       51S07780
      END                                         51S07790
```

```

C***** 51C00010
C*      * 51C00020
C*      FILE NAME: 51CCI      * 51C00030
C*      * 51C00040
C*      * 51C00050
C*      * 51C00060
C*      THIS PROGRAM EVALUATES THE PERFORMANCE OF 512-QAM      * 51C00070
C*      SYSTEMS IN THE PRESENCE OF MODULATED CO-CHANNEL      * 51C00080
C*      OF THE SAME TYPE, I.E. 512-QAM      * 51C00090
C*      * 51C00100
C***** 51C00110
C      COMPLEX DATA(524288),CCI(524288),TF(524288)      51C00120
C      COMPLEX XSIGNL(524288), XCCI(524288), BUFF(524288)      51C00130
C      DIMENSION PEI(50),EBNO(50),NI(8192),NQ(8192),NNI(8192),NNQ(8192)      51C00140
C      DIMENSION XARRAY(150),YARRAY(150),NIC(8192),NQC(8192)      51C00150
C      INTEGER IR(8192)      51C00160
C      REAL IBAUD,IMAX,QMAX      51C00170
C      51C00180
C      INITIALIZE PROGRAM.      51C00190
C      51C00200
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW      51C00210
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2      51C00220
C      COMMON IMAX,QMAX,EBNO1,RK,CN1      51C00230
C      51C00240
C      FOR 512-QAM, NBPSYM= 9, EBNO1=18., CN1=24., CDIM=40.0      51C00250
C      51C00260
C      NBPSYM= 9      51C00270
C      EBNO1=18.      51C00280
C      CN1=24.      51C00290
C      CDIM=40.0      51C00300
C      ATM=45.0      51C00310
C      NLEVL=24      51C00320
C      IMAX=FLOAT(NLEVL)-1.      51C00330
C      QMAX=IMAX      51C00340
C      RK=FLOAT(NBPSYM)      51C00350
C      KQAM=512      51C00360
C      51C00370
C      ALPHA=0.2      51C00380
C      51C00390
C      IND=1 FOR PLOTTING THE EYE-DIAGRAM      51C00400
C      IND=2 FOR PLOTTING THE P(E)      51C00410
C      51C00420
C      IND=2      51C00430
C      NRUNS=2      51C00440
C      IOFF=0      51C00450
C      LSAMPL=64      51C00460
C      NSNR=45      51C00470
C      NSYMB=8192      51C00480
C      LDIM=LSAMPL*NSYMB      51C00490
C      51C00500
C      THE BAUD RATE BAUD IS GIVEN IN KBAUD/S      51C00510
C      51C00520
C      BAUD=177.8      51C00530
C      SBANDW=BAUD*LSAMPL      51C00540
C      NO1=LDIM/2+1      51C00550
C      NO2=NO1+1      51C00560
C      FBW=BAUD/2.      51C00570
C      PI=3.14159      51C00580
C      51C00590
C      END OF INITIALIZATION.      51C00600
C      51C00610
C      START OF COMPUTATIONS.      51C00620
C      51C00630
C      PHI= 0.20*PI      51C00640
C      IOF= 32      51C00650
C      WRITE(6,14) PHI, IOF      51C00660
C      14 FORMAT(1X,'PHI=',F10.4,2X,'RADS',5X,'IOF=',I8///)      51C00670
C      92 WRITE(6,245) KQAM      51C00680
C      245 FORMAT(5X,I5,'-QAM',/)      51C00690
C      WRITE(6,45) ALPHA      51C00700
C      45 FORMAT(////,2X,'*****ALPHA=',F6.2)      51C00710
C      51C00720

```

```

      DSEED = 135791.D0
      CALL RAND (NI, NSYMB, DSEED)
      DSEED = 780958.D0
      CALL RAND (NQ, NSYMB, DSEED)
      CALL SIGNAL (XSIGNL, NI, NQ)
C
      DSEED = 148357.D0
      CALL RAND (NIC, NSYMB, DSEED)
      DSEED = 123338.D0
      CALL RAND (NQC, NSYMB, DSEED)
      CALL SIGNAL (XCCI, NIC, NQC)
      CALL RCOSTX(TF)
C
      CALL FILTER (XSIGNL, TF)
      CALL FILTER (XCCI, TF)
C-----
      70 DO 101 MM=1, 1
C
      WRITE(6,102) IOF, PHI
      102 FORMAT(1X, 'IOF=', I6, 5X, 'PHI=', F10.4///)
C
      DO 100 NR=1, NRUNS
C-----
C-----MAIN CHANNEL-----
C
      DO 40 I=1, LDIM
      40 DATA(I)=XSIGNL(I)
      CALL ENERGY(DATA, C, EB)
      IF(NR.EQ.1) GO TO 88
      AT=ATM-(NR-2)*2.5
C-----CO-CHANNEL-----
      DO 140 K=1, LDIM
      140 CCI(K)=XCCI(K)
      CALL ATT(CCI, AT)
      CALL PHASE(CCI, PHI)
      CALL OFFSET(CCI, IOF)
      CALL ENERGY(CCI, CI, EBC)
      CDI=10.*ALOG10(C/CI)
      WRITE(6,199) CDI
      199 FORMAT(5X, 'C/I=', F5.1, 'DB')
      DO 77 I=1, LDIM
      77 DATA(I)=DATA(I)+CCI(I)
C-----
C-----
C-----END OF THE TRANSMITTER-----
C-----
      88 CONTINUE
      82 CALL RCOSRX(TF)
      83 CALL HHGG(TF, PNOISE)
      CALL FILTER(DATA, TF)
C
      GO TO (26, 27), IND
      27 CALL SYNCRO(DATA, PNOISE, NI, NQ, MI, MQ, EB)
      CALL DECODE(NERROR, DATA, PNOISE, MI, MQ, NI, NQ, EBNO, PEI, EB)
CCC IF((NERROR.NE.0).AND.(NR.EQ.NRUNS)) CALL PLOT(0.0, 0.0, 999)
      IF(NERROR.NE.0) GO TO 100
      DO 99 I=1, NSNR
      XARRAY(I)=EBNO(I)
      99 YARRAY(I)=PEI(I)
      A=1.
      CALL DRAWCN(XARRAY, YARRAY, NR, NRUNS, A)
C
      CALL DRAWEB(XARRAY, YARRAY, JCURV, JLAST)
      GO TO 100
      26 CALL EYEQ(DATA)
      100 CONTINUE
      PHI=PHI+0.01*PI
      101 CONTINUE
      STOP
      END
C*****
C SUBROUTINE: SIGNAL *
```

```

51C00730
51C00740
51C00750
51C00760
51C00770
51C00780
51C00790
51C00800
51C00810
51C00820
51C00830
51C00840
51C00850
51C00860
51C00870
51C00880
51C00890
51C00900
51C00910
51C00920
51C00930
51C00940
51C00950
51C00960
51C00970
51C00980
51C00990
51C01000
51C01010
51C01020
51C01030
51C01040
51C01050
51C01060
51C01070
51C01080
51C01090
51C01100
51C01110
51C01120
51C01130
51C01140
51C01150
51C01160
51C01170
51C01180
51C01190
51C01200
51C01210
51C01220
51C01230
51C01240
51C01250
51C01260
51C01270
51C01280
51C01290
51C01300
51C01310
51C01320
51C01330
51C01340
51C01350
51C01360
51C01370
51C01380
51C01390
51C01400
51C01410
51C01420
51C01430
51C01440
```



```

C * 51C01450
C THIS SUBROUTINE FORMS A COMPLEX BASEBAND SIGNAL * 51C01460
C***** 51C01470
SUBROUTINE SIGNAL(DATA,NI,NQ) 51C01480
COMPLEX DATA(1) 51C01490
DIMENSION NI(1),NQ(1) 51C01500
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 51C01510
COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2 51C01520
DO 1 I=1, NSYMB 51C01530
IF ((NI(I) .EQ. NQ(I)) .AND. (NI(I) .EQ. 23)) NI(I)=15 51C01540
IF ((NI(I) .EQ. NQ(I)) .AND. (NI(I) .EQ. 21)) NI(I)=13 51C01550
1 CONTINUE 51C01560
DO 2 I=1,NSYMB 51C01570
C LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL 51C01580
J1=(I-1)*LSAMPL+1 51C01590
J2=I*LSAMPL 51C01600
DO 3 J3=J1,J2 51C01610
3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I))) 51C01620
2 CONTINUE 51C01630
RETURN 51C01640
END 51C01650
C***** 51C01660
C SUBROUTINE: RCOSTX * 51C01670
C * 51C01680
C THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A * 51C01690
C RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A * 51C01700
C X/SINX APERTURE EQUALIZER * 51C01710
C***** 51C01720
SUBROUTINE RCOSTX(TF) 51C01730
COMPLEX TF( 1) 51C01740
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL 51C01750
COMMON /NUMB2/ NSNR,NSYMB,BAUD 51C01760
SBANDW=BAUD*LSAMPL 51C01770
NO=LDIM/2 51C01780
NO1=NO+1 51C01790
IF (ALPHA.EQ.0) ALPHA=0.0001 51C01800
FN=LDIM*(FBW/SBANDW) 51C01810
F1=(1.-ALPHA)*FN 51C01820
F2=(1.+ALPHA)*FN 51C01830
IFN=IFIX(FN) 51C01840
IF1=IFIX(F1)+1 51C01850
IF2=IFIX(F2)+1 51C01860
C 51C01870
C THE AMPLITUDE CHARACTERISTICS 51C01880
C 51C01890
A1=3.1415926/(2.*FLOAT(IFN)) 51C01900
DO 8 I=2,IF1 51C01910
TF(I)=CMPLX(1.0,0.0) 51C01920
J=I-1 51C01930
C 51C01940
C X/SIN(X) EQUALIZATION 51C01950
C 51C01960
A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1)) 51C01970
TF(I)=CMPLX(A2,0.0) 51C01980
8 CONTINUE 51C01990
JK=IF1+1 51C02000
DO 9 J=JK,IF2 51C02010
I=J-1 51C02020
C 51C02030
C ROOT OF RAISED COSINE 51C02040
C 51C02050
A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1)) 51C02060
A=(3.1415926/(2.0*ALPHA))*(((FLOAT(I)/FLOAT(IFN))-1.) 51C02070
TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0) 51C02080
TF(J)=TF(J)*CMPLX(A3,0.0) 51C02090
9 CONTINUE 51C02100
JH=IF2+1 51C02110
DO 10 I=JH,NO1 51C02120
TF(I)=CMPLX(0.0,0.0) 51C02130
10 CONTINUE 51C02140
NO2=NO1+1 51C02150
DO 5 I=NO2,LDIM 51C02160

```

```

      TF(1)=CONJG(TF(LDIM+2-I))
5  CONTINUE
      RETURN
      END
C*****
C  SUBROUTINE: RCOSRX
C
C  THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C  RAISED COSINE FILTER WITH ARBITRARY ALPHA
C*****
      SUBROUTINE RCOSRX(TF)
      COMPLEX TF( 1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SBANDW=BAUD*LSAMPL
      NO=LDIM/2
      NO1=NO+1
      IF (ALPHA.EQ.0) ALPHA=0.0001
      FN=LDIM*(FBW/SBANDW)
      F1=(1.-ALPHA)*FN
      F2=(1.+ALPHA)*FN
      IFN=IFIX(FN)
      IF1=IFIX(F1)+1
      IF2=IFIX(F2)+1
C
C  AMPLITUDE CHARACTERISTIC
C
      A1=3.141592/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
      TF(1)=CMPLX(1.0,0.0)
      J=I-1
      TF(I)=CMPLX(1.,0.0)
8  CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
      I=J-1
C
C  ROOT OF RAISED COSINE
C
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9  CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
      TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
      TF(I)=CONJG(TF(LDIM+2-I))
5  CONTINUE
      RETURN
      END
C*****
C  SUBROUTINE: FILTER
C
C  THIS SUBROUTINE PERFORMS THE FILTERING
C  PROCESS ON A DATA SEQUENCE
C*****
      SUBROUTINE FILTER(DATA,TF)
      COMPLEX DATA(1),TF(1)
      DIMENSION IWK(20)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      CALL FFT2C(DATA,19,IWK)
      DO 1 I=1,LDIM
1  DATA(I)=CONJG(DATA(I)*TF(I))
      CALL FFT2C(DATA,19,IWK)
      DO 2 I=1,LDIM
2  DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END
C*****
C  SUBROUTINE: HMGG

```

```

51C02170
51C02180
51C02190
51C02200
51C02210
51C02220
51C02230
51C02240
51C02250
51C02260
51C02270
51C02280
51C02290
51C02300
51C02310
51C02320
51C02330
51C02340
51C02350
51C02360
51C02370
51C02380
51C02390
51C02400
51C02410
51C02420
51C02430
51C02440
51C02450
51C02460
51C02470
51C02480
51C02490
51C02500
51C02510
51C02520
51C02530
51C02540
51C02550
51C02560
51C02570
51C02580
51C02590
51C02600
51C02610
51C02620
51C02630
51C02640
51C02650
51C02660
51C02670
51C02680
51C02690
51C02700
51C02710
51C02720
51C02730
51C02740
51C02750
51C02760
51C02770
51C02780
51C02790
51C02800
51C02810
51C02820
51C02830
51C02840
51C02850
51C02860
51C02870
51C02880

```

```

C                                     * 51C02890
C THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER * 51C02900
C AT THE OUTPUT OF THE RECEIVE FILTER * 51C02910
C***** * 51C02920
SUBROUTINE MHGG(TF,PNOISE) * 51C02930
COMPLEX TF(1) * 51C02940
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL * 51C02950
COMMON /NUMB2/ NSNR,NSYMB,BAUD * 51C02960
SUM=0.0 * 51C02970
SBANDW=BAUD*LSAMPL * 51C02980
DO 1 L=1,LDIM * 51C02990
HH=(CABS(TF(L)))**2 * 51C03000
1 SUM=SUM+HH * 51C03010
PNOISE=SUM*SBANDW/FLOAT(LDIM)/2. * 51C03020
WRITE (6,2) PNOISE * 51C03030
2 FORMAT(5X,'PNOISE=',F7.3,/) * 51C03040
RETURN * 51C03050
END * 51C03060
C***** * 51C03070
C SUBROUTINE: ENERGY * 51C03080
C * 51C03090
C THIS SUBROUTINE COMPUTES THE CARRIER POWER * 51C03100
C AND THE AVERAGE BIT ENERGY * 51C03110
C***** * 51C03120
SUBROUTINE ENERGY(DATA,C,EB) * 51C03130
COMMON /NUMB2/ NSNR,NSYMB,BAUD * 51C03140
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL * 51C03150
COMMON IMAX,QMAX,EBNO1,RK,CN1 * 51C03160
COMPLEX DATA(1) * 51C03170
WATTS=0. * 51C03180
DO 1 I=1,LDIM * 51C03190
WATTS=WATTS+CABS(DATA(I))**2 * 51C03200
1 CONTINUE * 51C03210
C=WATTS/FLOAT(LDIM)/2. * 51C03220
RRK=RK/2. * 51C03230
EB=C/(RRK*BAUD) * 51C03240
WRITE (6,40) C,EB * 51C03250
40 FORMAT(10X,'C=',F15.6,5X,'EB=',E15.8) * 51C03260
RETURN * 51C03270
END * 51C03280
C***** * 51C03290
C SUBROUTINE: SYNCRO * 51C03300
C * 51C03310
C THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA * 51C03320
C***** * 51C03330
SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB) * 51C03340
INTEGER QMAXF * 51C03350
COMPLEX DATA(1),AMP * 51C03360
DIMENSION NI(1),NQ(1) * 51C03370
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL * 51C03380
COMMON /NUMB2/ NSNR,NSYMB,BAUD * 51C03390
COMMON IMAX,QMAX,EBNO1,RK,CN1 * 51C03400
NERROR=0 * 51C03410
IF (IOFF.EQ.0) GO TO 111 * 51C03420
DO 6 K=1,IOFF * 51C03430
XX=AIMAG(DATA(LDIM)) * 51C03440
DO 7 KK=2,LDIM * 51C03450
7 DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1))) * 51C03460
6 DATA(1)=CMPLX(REAL(DATA(1)),XX) * 51C03470
C * 51C03480
C SYNCHRONIZE THE RECEIVED DATA * 51C03490
C * 51C03500
111 NOLD=0 * 51C03510
NOF=0 * 51C03520
K=1 * 51C03530
300 CONTINUE * 51C03540
NEW=0 * 51C03550
DO 200 J=1,NSYMB * 51C03560
THR1I=FLOAT(NI(J))-1. * 51C03570
THR2I=FLOAT(NI(J))+1. * 51C03580
THR1Q=FLOAT(NQ(J))-1. * 51C03590
THR2Q=FLOAT(NQ(J))+1. * 51C03600

```

```

      J1=K+(J-1)*LSAMPL
      IF(J1.GT.LDIM) J1=J1-LDIM
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      IF((AXBAR.GT.THR1I).AND.(AXBAR.LT.THR2I)) NEW=NEW+1
      IF((AYBAR.GT.THR1Q).AND.(AYBAR.LT.THR2Q)) NEW=NEW+1
200  CONTINUE
      IF(NOLD.GE.NEW) GO TO 399
      NOLD=NEW
      NOF=K
399  K=K+1
      LFU=2*NSYMB
C
C      SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP
C
C      NSFT=LSAMPL*8
      IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300
      NOF=NOF-1
      KT=K-2
      WRITE(6,56) NOLD,NOF,KT
56  FORMAT(1X,'NO OF MAX. CORRECT SYMBOLS = ',I4,/,1X,
* 'NO OF SHIFT IN SAMPLES FOR MAX. CORRECT SYMBOLS= ',I4,/,
* '1X,'TOTAL NO OF SHIFT FOR SEARCHING = ',I4)
C
      IF (NOF.EQ.0) GO TO 230
      LO=LDIM-1
      DO 250 I=1,NOF
      AMP=DATA(1)
      DO 240 J=1,LO
      DATA(J)=DATA(J+1)
240  CONTINUE
      DATA(LDIM)=AMP
250  CONTINUE
C
C      OPTIMIZE THE SAMPLING INSTANT
C
230  MI=1
      MQ=1
      EOI=FLOAT(NSYMB) + 1.
      EOQ=FLOAT(NSYMB) + 1.
      M=22
      VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
      SIGMA=SQRT(VARIAN)
      MOS=LSAMPL/4
      DO 80 J=1,MOS
      EI=0.
      EQ=0.
      IERX=0
      IERY=0
      DO 70 K=1,NSYMB
C      VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
      J1=(K-1)*LSAMPL+J
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      INDEXI=0
      AMPX=ABS(AXBAR)
      AMPI=ABS(FLOAT(NI(K)))
      THR1I=AMPI - 1.0
      THR2I=AMPI + 1.0
      IMAXF=0
      IF (AMPI.EQ.IMAX) GO TO 30
      IF ((AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30  IF(AMPX.LE.THR1I) INDEXI=1
      IMAXF=1
40  CONTINUE
      INDEXQ=0
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0

```

51C03810
51C03820
51C03830
51C03840
51C03850
51C03860
51C03870
51C03880
51C03890
51C03900
51C03910
51C03920
51C03930
51C03940
51C03950
51C03960
51C03970
51C03980
51C03990
51C04000
51C04010
51C04020
51C04030
51C04040
51C04050
51C04060
51C04070
51C04080
51C04090
51C04100
51C04110
51C04120
51C04130
51C04140
51C04150
51C04160
51C04170
51C04180
51C04190
51C04200
51C04210
51C04220
51C04230
51C04240
51C04250
51C04260
51C04270
51C04280
51C04290
51C04300
51C04310
51C04320

```

      QMAXF=0
      IF (AMPQ.EQ.QMAX) GO TO 50
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50  IF (AMPY.LE.THR1Q) INDEXQ=1
      QMAXF=1
60  CONTINUE
      IF (INDEXI.EQ.1) EI=EI + 1.
      IF (INDEXQ.EQ.1) EQ=EQ + 1.
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB
C
      D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C      CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
      IF (IMAXF .EQ. 1) GO TO 65
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
65  D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IF (QMAXF .EQ. 1) GO TO 70
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IERX=IERX+INDEXI
      IERY=IERY+INDEXQ
70  CONTINUE
      WRITE(6,81) IERX,IERY
81  FORMAT(5X,'IERX=',I5,5X,'IERY=',I5)
      IF (EOI.LE.EI) GO TO 75
      EOI=EI
      MI=J
75  CONTINUE
      IF (EOQ.LE.EQ) GO TO 80
      EOQ=EQ
      MQ=J
80  MOFF=IABS(MI-MQ)
      WRITE(6,91) MI,MQ
91  FORMAT(5X,'MI=',I3,5X,'MQ=',I3)
      IF (MOFF.NE.0) WRITE(6,90) MOFF
      OFF=(FLOAT(MI) + FLOAT(MQ))/FLOAT(LSAMPL)
90  FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
      #12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
      WRITE(6,95) OFF
95  FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
      RETURN
      END
C*****
C      SUBROUTINE: DECODES
C
C      THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA
C*****
      SUBROUTINE DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)
      INTEGER QMAXF
      COMPLEX DATA(1),AMP
      DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      NERROR=0
      DO 1 I=1,NSNR
1    PE(I)=0.
      COMMON IMAX,QMAX,EBNO1,RK,CN1
C      CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE
C      ERRORS IN ADJACENT LEVELS ARE CONSIDERED

```

51C04330
 51C04340
 51C04350
 51C04360
 51C04370
 51C04380
 51C04390
 51C04400
 51C04410
 51C04420
 51C04430
 51C04440
 51C04450
 51C04460
 51C04470
 51C04480
 51C04490
 51C04500
 51C04510
 51C04520
 51C04530
 51C04540
 51C04550
 51C04560
 51C04570
 51C04580
 51C04590
 51C04600
 51C04610
 51C04620
 51C04630
 51C04640
 51C04650
 51C04660
 51C04670
 51C04680
 51C04690
 51C04700
 51C04710
 51C04720
 51C04730
 51C04740
 51C04750
 51C04760
 51C04770
 51C04780
 51C04790
 51C04800
 51C04810
 51C04820
 51C04830
 51C04840
 51C04850
 51C04860
 51C04870
 51C04880
 51C04890
 51C04900
 51C04910
 51C04920
 51C04930
 51C04940
 51C04950
 51C04960
 51C04970
 51C04980
 51C04990
 51C05000
 51C05010
 51C05020
 51C05030
 51C05040

```

3  DO 2 K=1,NSYMB                                51C05050
    J1=(K-1)*LSAMPL+MI                            51C05060
    J2=(K-1)*LSAMPL + MQ                          51C05070
    AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.    51C05080
    AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.  51C05090
    INDEXI=0                                       51C05100
    AMPX=ABS(AXBAR)                               51C05110
    AMPI=ABS(FLOAT(NI(K)))                        51C05120
    THR1I=AMPI - 1.0                             51C05130
    THR2I=AMPI + 1.0                             51C05140
    IMAXF=0                                       51C05150
    IF (AMPI.EQ.IMAX) GO TO 30                    51C05160
    IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1 51C05170
    GO TO 40                                       51C05180
30  IF(AMPX.LE.THR1I) INDEXI=1                    51C05190
    IMAXF=1                                       51C05200
40  CONTINUE                                       51C05210
    INDEXQ=0                                       51C05220
    AMPY=ABS(AYBAR)                               51C05230
    AMPQ=ABS(FLOAT(NQ(K)))                        51C05240
    THR1Q=AMPQ - 1.0                             51C05250
    THR2Q=AMPQ + 1.0                             51C05260
    QMAXF=0                                       51C05270
    IF (AMPQ.EQ.QMAX) GO TO 50                    51C05280
    IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1 51C05290
    GO TO 60                                       51C05300
50  IF(AMPY.LE.THR1Q) INDEXQ=1                    51C05310
    QMAXF=1                                       51C05320
60  CONTINUE                                       51C05330
    IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1 51C05340
C                                                     51C05350
C  COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL. 51C05360
C  THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS 51C05370
C  1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE. 51C05380
C  FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQR(2))) 51C05390
C  AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQR(2))) + 51C05400
C  1/2*ERFC(D2/(SIGMA*SQR(2))).                  51C05410
C                                                     51C05420
C  DO 4 M=1,NSNR                                  51C05430
    EBNO(M)=EBNO1+1.0*FLOAT(M-1)                 51C05440
    VARIAN=PNOISE*EB*(10.**(-0.1*EBNO(M)))         51C05450
    SIGMA=SQRT(VARIAN)                            51C05460
    D1=ABS(AMPX-THR1I)                             51C05470
    ARG=D1/(SIGMA*SQR(2.))                         51C05480
C  CHECK IF ARG IS LARGE IN WHICH CASE PE IS INSIGNIFICANT 51C05490
C  CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN 51C05500
C                                                     51C05510
    NERI=K                                          51C05520
    IF (INDEXI.EQ.1) GO TO 153                     51C05530
    IF (ARG.GT.12.) ARG=12.                        51C05540
    PEI=ERFC(ARG)/2.                               51C05550
    IF (IMAXF .EQ. 1) GO TO 100                    51C05560
    D2=ABS(THR2I-AMPX)                             51C05570
    ARG=D2/(SIGMA*SQR(2.))                         51C05580
    IF (ARG.GT.12.) ARG=12.                        51C05590
    PEI=PEI + ERFC(ARG)/2.                         51C05600
100 D1=ABS(AMPY-THR1Q)                             51C05610
    ARG=D1/(SIGMA*SQR(2.))                         51C05620
    IF (INDEXQ.EQ.1) GO TO 153                     51C05630
    IF (ARG.GT.12.) ARG=12.                        51C05640
    PEQ=ERFC(ARG)/2.                               51C05650
    IF (QMAXF .EQ. 1) GO TO 110                    51C05660
    D2=ABS(THR2Q-AMPY)                             51C05670
    ARG=D2/(SIGMA*SQR(2.))                         51C05680
    IF (ARG.GT.12.) ARG=12.                        51C05690
    PEQ=PEQ + ERFC(ARG)/2.                         51C05700
110 IF(PEI.LT.1.E-15) PEI=0.                     51C05710
    IF(PEQ.LT.1.E-15) PEQ=0.                     51C05720
    4  PE(M)=PE(M)+(PEI+PEQ)/2.                   51C05730
    2  CONTINUE                                     51C05740
    DO 5 I=1,NSNR                                  51C05750
    5  PE(I)=PE(I)/FLOAT(NSYMB)                   51C05760

```

```

C      PRINT 150
C150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)
C      WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)
C151  FORMAT(5X,F5.1,10X,E10.2)
      WRITE (6,152) NERROR
152   FORMAT(/5X,'ERRORS=',I5,/)
      GO TO 155
153  PRINT 154
154  FORMAT (10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)
      WRITE(6,157) NERI
157  FORMAT(5X,'NERI=',I5)
155  CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: DRAWCN
C
C      THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C      P(E) AS LOW AS 1.0E-10 AND A C/N RATIO AS HIGH AS 42 DB
C*****
      SUBROUTINE DRAWCN(XARRAY,YARRAY,JCURV,JLAST,A)
      DIMENSION XARRAY(1),YARRAY(1),XA(150),XX(150),YY(150)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      M=NSNR
      BR=RK*BAUD
      NBW=BAUD/A
      CF=BR/NBW
      DO 1 K = 1 , NSNR
      XA(K)=XARRAY( K ) + 10*(ALOG10(CF))
      IF(YARRAY( K ).LE.1.0E-10) GOTO 3
      GOTO 1
3     M= K-1
      GOTO 4
1     CONTINUE
4     CONTINUE
      I=M+1
      II=I+1
      XA(I)=9.0
      XA(II)=1.0
      YARRAY(I)=1.E-10
      YARRAY(II)=0.4
CC     GO TO 2
      IF(JCURV.GT.1) GOTO 2
C
C      ESTABLISH THE SURFACE AREA.
C
      CALL PLOTS(57.0,27.5)
      CALL FACTOR(0.85)
C
C      ESTABLISH THE ORIGIN.
C
      CALL PLOT(3.0,1.5,-3)
C
C      DRAW THE LOGARITHMIC Y-AXIS.
C
      CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,25.,
+90.,1.0E-10,.4)
C
C      DRAW THE LINEAR X-AXIS.
C
      CALL AXIS(0.0,0.0,9HC/N IN dB,-9,
+48.0,0.0,9.0,1.0)
C
C      PLOT THE DATA IN LOG-LINEAR MODE.
C
2     CONTINUE
      PRINT 150
150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',7X,'C/N',/)
      WRITE(6,151) (XARRAY(I),YARRAY(I),XA(I),I=1,M)
151  FORMAT(5X,F5.0,10X,E10.2,10X,F5.1)
CC     GO TO 45

```

51C05770
51C05780
51C05790
51C05800
51C05810
51C05820
51C05830
51C05840
51C05850
51C05860
51C05870
51C05880
51C05890
51C05900
51C05910
51C05920
51C05930
51C05940
51C05950
51C05960
51C05970
51C05980
51C05990
51C06000
51C06010
51C06020
51C06030
51C06040
51C06050
51C06060
51C06070
51C06080
51C06090
51C06100
51C06110
51C06120
51C06130
51C06140
51C06150
51C06160
51C06170
51C06180
51C06190
51C06200
51C06210
51C06220
51C06230
51C06240
51C06250
51C06260
51C06270
51C06280
51C06290
51C06300
51C06310
51C06320
51C06330
51C06340
51C06350
51C06360
51C06370
51C06380
51C06390
51C06400
51C06410
51C06420
51C06430
51C06440
51C06450
51C06460
51C06470
51C06480

```

      CALL LGLIN(XA,YARRAY,M,1,4,JCURV,1)
      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
45 RETURN
      END
C*****
C      SUBROUTINE: EYEQ
C
C      THIS SUBROUTINE DRAWS THE EYE DIAGRAM
C      FOR A TWO SYMBOL DURATION
C*****
      SUBROUTINE EYEQ(DATA)
      DIMENSION DATA2(86),DATA3(86)
      COMPLEX DATA(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
C      ESTABLISH THE SURFACE AREA.
      CALL PLOTS(37.0,27.5)
C      ESTABLISH THE ORIGIN.
      CALL PLOT(3.0,13.0,-3)
C      WRITE THE TITLE OF THE GRAPH.
C      CALL SYMBOL(10.0,12.0,0.49,16HQPSK EYE DIAGRAM,0.0,16)
C      DRAW THE TIME AXIS.
      CALL AXIS(0.0,0.0,1H,-1,24.,0.0,1.0,2.8)
C      DRAW THE AMPLITUDE AXIS.
      CALL AXIS(0.0,-8.00,1H,1,16.0,90.0,-24.,3.0)
C      PLOT THE DATA.
      JJ=2*LSAMPL
      DATA2(JJ+1)=0.0
      DATA2(JJ+2)=3.0
      DATA3(JJ+1)=1.0
      DATA3(JJ+2)=2.8
      M2=NSYMB/8
      M3=3*M2+1
      M4=4*M2-1
      DO 4 KK=1,M2
      DO 2 I=1,JJ
      II=(KK-1)*LSAMPL+I
      IF(II.GT.LDIM) II=II-(II/LDIM)*LDIM
      DATA2(I)=REAL(DATA(II))
2 DATA3(I)=FLOAT(I)
      CALL LINE(DATA3,DATA2,JJ,1,0,0)
4 CONTINUE
      CALL PLOT(0.0,0.0,999)
      RETURN
      END
C*****
C      SUBROUTINE: ATT
C
C      THIS SUBROUTINE PERFORMS THE ATTENUATION OF A COMPLEX
C      SIGNAL (AT IS THE ATTENUATION FACTOR IN DB)
C*****
      SUBROUTINE ATT(DATA,AT)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMPLEX DATA(1)
      ATTT=10.**(-AT/20.)
      DO 10 I=1,LDIM
10 DATA(I)=DATA(I)*ATTT
      RETURN
      END
C*****
C      SUBROUTINE: OFFSET
C
C      THIS SUBROUTINE SHIFTS THE DATA SEQUENCE BY IOF
C*****
      SUBROUTINE OFFSET(DATA,IOF)
      COMPLEX DATA(1),BUFF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
      J=LDIM-IABS(IOF)
      K=J+1
      IF (IOF.EQ.0) GOTO 9
      IF (IOF.LT.0) GOTO 4

```

```

51C06490
51C06500
51C06510
51C06520
51C06530
51C06540
51C06550
51C06560
51C06570
51C06580
51C06590
51C06600
51C06610
51C06620
51C06630
51C06640
51C06650
51C06660
51C06670
51C06680
51C06690
51C06700
51C06710
51C06720
51C06730
51C06740
51C06750
51C06760
51C06770
51C06780
51C06790
51C06800
51C06810
51C06820
51C06830
51C06840
51C06850
51C06860
51C06870
51C06880
51C06890
51C06900
51C06910
51C06920
51C06930
51C06940
51C06950
51C06960
51C06970
51C06980
51C06990
51C07000
51C07010
51C07020
51C07030
51C07040
51C07050
51C07060
51C07070
51C07080
51C07090
51C07100
51C07110
51C07120
51C07130
51C07140
51C07150
51C07160
51C07170
51C07180
51C07190
51C07200

```



```

DO 1 I=1,J                                51C07210
1  BUFF(I)=DATA(I)                        51C07220
DO 2 I=K,LDIM                              51C07230
2  DATA(I-J)=DATA(I)                    51C07240
DO 3 I=1,J                                51C07250
3  DATA(I+IOF)=BUFF(I)                  51C07260
GOTO 9                                    51C07270
4  IOF=-IOF                              51C07280
L=IOF+1                                  51C07290
DO 5 I=L,LDIM                              51C07300
5  BUFF(I-IOF)=DATA(I)                  51C07310
DO 6 I=1,IOF                              51C07320
6  DATA(I+J)=DATA(I)                  51C07330
DO 7 I=1,J                                51C07340
7  DATA(I)=BUFF(I)                    51C07350
9  RETURN                                51C07360
END                                        51C07370
C*****                                51C07380
C  SUBROUTINE: PHASE                      * 51C07390
C                                          * 51C07400
C  THIS SUBROUTINE PERFORMS THE PHASE SHIFT  * 51C07410
C  OF A COMPLEX DATA SEQUENCE              * 51C07420
C*****                                51C07430
C  SUBROUTINE PHASE(DATA,PHI)              51C07440
C  COMPLEX EPS,DATA(1)                    51C07450
C  COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 51C07460
C  EPS=CMPLX(COS(PHI),-SIN(PHI))          51C07470
C  DO 10 I=1,LDIM                         51C07480
10 DATA(I)=DATA(I)*EPS                  51C07490
RETURN                                51C07500
END                                        51C07510
C*****                                51C07520
C  SUBROUTINE: RAND                      * 51C07530
C                                          * 51C07540
C  THIS SUBROUTINE GENERATES RANDOM SYMBOLS * 51C07550
C*****                                51C07560
C  SUBROUTINE RAND(IR,N,DSEED)              51C07570
C  INTEGER IR(1)                          51C07580
C  DOUBLE PRECISION DSEED                  51C07590
C  DO 3000 I=1,N                          51C07600
C  R=GGUBFS(DSEED)                        51C07610
C  IR(I)=IFIX(R)                          51C07620
C  IF (R.LE.0.041) IR(I)=-23              51C07630
C  IF (R.GT.0.041) .AND. (R.LE.0.082) IR(I)=-21 51C07640
C  IF (R.GT.0.082) .AND. (R.LE.0.123) IR(I)=-19 51C07650
C  IF (R.GT.0.123) .AND. (R.LE.0.164) IR(I)=-17 51C07660
C  IF (R.GT.0.164) .AND. (R.LE.0.205) IR(I)=-15 51C07670
C  IF (R.GT.0.205) .AND. (R.LE.0.246) IR(I)=-13 51C07680
C  IF (R.GT.0.246) .AND. (R.LE.0.287) IR(I)=-11 51C07690
C  IF (R.GT.0.287) .AND. (R.LE.0.328) IR(I)=-9   51C07700
C  IF (R.GT.0.328) .AND. (R.LE.0.369) IR(I)=-7   51C07710
C  IF (R.GT.0.369) .AND. (R.LE.0.41) IR(I)=-5    51C07720
C  IF (R.GT.0.41) .AND. (R.LE.0.451) IR(I)=-3    51C07730
C  IF (R.GT.0.451) .AND. (R.LE.0.492) IR(I)=-1    51C07740
C  IF (R.GT.0.492) .AND. (R.LE.0.533) IR(I)=1     51C07750
C  IF (R.GT.0.533) .AND. (R.LE.0.574) IR(I)=3     51C07760
C  IF (R.GT.0.574) .AND. (R.LE.0.615) IR(I)=5     51C07770
C  IF (R.GT.0.615) .AND. (R.LE.0.656) IR(I)=7     51C07780
C  IF (R.GT.0.656) .AND. (R.LE.0.697) IR(I)=9     51C07790
C  IF (R.GT.0.697) .AND. (R.LE.0.738) IR(I)=11    51C07800
C  IF (R.GT.0.738) .AND. (R.LE.0.779) IR(I)=13    51C07810
C  IF (R.GT.0.779) .AND. (R.LE.0.82) IR(I)=15     51C07820
C  IF (R.GT.0.82) .AND. (R.LE.0.861) IR(I)=17     51C07830
C  IF (R.GT.0.861) .AND. (R.LE.0.902) IR(I)=19     51C07840
C  IF (R.GT.0.902) .AND. (R.LE.0.943) IR(I)=21    51C07850
C  IF (R.GT.0.943) .AND. (R.LE.0.984) IR(I)=23    51C07860
C  IF (R.GT.0.984) IR(I)=23                51C07870
3000 CONTINUE                            51C07880
RETURN                                51C07890
END                                        51C07900

```

```

C***** 10S00010
C*      * 10S00020
C*      * 10S00030
C*      * 10S00040
C*      * 10S00050
C*      * 10S00060
C*      * 10S00070
C*      * 10S00080
C*      * 10S00090
C*      * 10S00100
C***** 10S00110
C      COMPLEX DATA(524288),CCI(524288),TF(524288) 10S00120
C      COMPLEX XSIGNL(524288),BUFF(524288) 10S00130
C      DIMENSION PEI(50),EBNO(50),NI(8192),NQ(8192),NNI(8192),NNQ(8192) 10S00140
C      DIMENSION XARRAY(150),YARRAY(150) 10S00150
C      INTEGER IR(8192) 10S00160
C      REAL IBAUD,IMAX,QMAX 10S00170
C 10S00180
C      INITIALIZE PROGRAM. 10S00190
C 10S00200
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 10S00210
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2 10S00220
C      COMMON IMAX,QMAX,EBNO1,RK,CN1 10S00230
C 10S00240
C      FOR 1024QAM, NBPSYM= 10 EBNO1=18., CN1=24., CDIM=40.0 10S00250
C 10S00260
C      NBPSYM=10 10S00270
C      EBNO1=18. 10S00280
C      CN1=24. 10S00290
C      CDIM=47.5 10S00300
C      NLEVL=2** (NBPSYM/2) 10S00310
C      IMAX=FLOAT(NLEVL)-1. 10S00320
C      QMAX=IMAX 10S00330
C      RK=FLOAT(NBPSYM) 10S00340
C      KQAM=NLEVL**2 10S00350
C 10S00360
C      ALPHA=0.2 10S00370
C 10S00380
C      IND=1 FOR PLOTTING THE EYE-DIAGRAM 10S00390
C      IND=2 FOR PLOTTING THE P(E) 10S00400
C 10S00410
C      IND=2 10S00420
C      NRUNS=1 10S00430
C      IOFF=0 10S00440
C      LSAMPL=64 10S00450
C      NSNR=45 10S00460
C      NSYMB=8192 10S00470
C      LDIM=LSAMPL*NSYMB 10S00480
C 10S00490
C      THE BAUD RATE BAUD IS GIVEN IN KBAUD/S 10S00500
C 10S00510
C      BAUD=160.0 10S00520
C      SBANDW=BAUD*LSAMPL 10S00530
C      NO1=LDIM/2+1 10S00540
C      NO2=NO1+1 10S00550
C      FBW=BAUD/2. 10S00560
C      PI=3.14159 10S00570
C 10S00580
C      END OF INITIALIZATION. 10S00590
C 10S00600
C      START OF COMPUTATIONS. 10S00610
C 10S00620
C      THETA = 0.150*PI 10S00630
C      WRITE(6,14) THETA 10S00640
C      14 FORMAT(1X,'THETA=',F10.4,2X,'RADS'////) 10S00650
C      92 WRITE(6,245) KQAM 10S00660
C      245 FORMAT(5X,I5,'-QAM',/) 10S00670
C      WRITE(6,45) ALPHA 10S00680
C      45 FORMAT(////,2X,'*****ALPHA=',F6.2) 10S00690
C 10S00700
C      DSEED = 135791.D0 10S00710
C      CALL RAND (NI, NSYMB, DSEED) 10S00720

```

```

      DSEED = 780956.D0
      CALL RAND (NQ, NSYMB, DSEED)
      CALL SIGNAL (XSIGNL, NI, NQ)
C
      CALL RCOSTX(TF)
C
      CALL FILTER (XSIGNL, TF)
C-----
C
      70 DO 101 MM=1, 1
C
      WRITE(6,102) THETA
      102 FORMAT(1X,'THETA=',F11.4,2X,'RADS'////)
C
      DO 100 NR=1,NRUNS
C-----
C      MAIN CHANNEL
C-----
      DO 40 I=1,LDIM
      40 DATA(I)=XSIGNL(I)
      CALL ENERGY(DATA,C,EB)
      IF(NR.EQ.1) GO TO 88
      CDI =CDIM-(NR-2)*2.5
C-----
C      CO-CHANNEL
C-----
      AM=SQRT(2.*C*10**(-0.1*CDI))
      DO 177 K=1,LDIM
      177 DATA(K)=DATA(K)+AM*CMPLX(COS(THETA),-SIN(THETA))
      WRITE(6,199) CDI
      199 FORMAT(5X,'C/I=',F5.1,'DB')
C
C-----
C      END OF THE TRANSMITTER
C-----
C
      88 CONTINUE
      82 CALL RCOSRX(TF)
      83 CALL HHGG(TF,PNOISE)
      CALL FILTER(DATA,TF)
C
      GO TO (26,27), IND
      27 CALL SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      CALL DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PEI,EB)
CCC  IF((NERROR.NE.0).AND.(NR.EQ.NRUNS)) CALL PLOT(0.0,0.0,999)
      IF (NERROR.NE.0) GO TO 100
      DO 99 I=1,NSNR
      XARRAY(I)=EBNO(I)
      99 YARRAY(I)=PEI(I)
      A=1.
      CALL DRAWCH(XARRAY,YARRAY,NR,NRUNS,A)
C
      CALL DRAWEB(XARRAY,YARRAY,JCURV,JLAST)
      GO TO 100
      26 CALL EYEQ(DATA)
      100 CONTINUE
      THETA = THETA + 0.02*PI
      101 CONTINUE
      STOP
      END
C*****
C      SUBROUTINE:  SIGNAL
C
C      THIS SUBROUTINE FORMS A COMPLEX BASEBAND SIGNAL
C*****
C      SUBROUTINE SIGNAL(DATA,NI,NQ)
C      COMPLEX DATA(1)
C      DIMENSION NI(1),NQ(1)
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
C      DO 1 I=1, NSYMB
C      IF ((NI(I) .EQ. NQ(I)) .AND. (NI(I) .EQ. 23)) NI(I)=15
C      IF ((NI(I) .EQ. NQ(I)) .AND. (NI(I) .EQ. 21)) NI(I)=13
      1 CONTINUE
      DO 2 I=1,NSYMB

```

```

10S00730
10S00740
10S00750
10S00760
10S00770
10S00780
10S00790
10S00800
10S00810
10S00820
10S00830
10S00840
10S00850
10S00860
10S00870
10S00880
10S00890
10S00900
10S00910
10S00920
10S00930
10S00940
10S00950
10S00960
10S00970
10S00980
10S00990
10S01000
10S01010
10S01020
10S01030
10S01040
10S01050
10S01060
10S01070
10S01080
10S01090
10S01100
10S01110
10S01120
10S01130
10S01140
10S01150
10S01160
10S01170
10S01180
10S01190
10S01200
10S01210
10S01220
10S01230
10S01240
10S01250
10S01260
10S01270
10S01280
10S01290
10S01300
10S01310
10S01320
10S01330
10S01340
10S01350
10S01360
10S01370
10S01380
10S01390
10S01400
10S01410
10S01420
10S01430
10S01440

```

```
C      LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL          10S01450
      J1=(I-1)*LSAMPL+1          10S01460
      J2=I*LSAMPL                10S01470
      DO 3 J3=J1,J2              10S01480
3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))          10S01490
2 CONTINUE                    10S01500
  RETURN                      10S01510
  END                        10S01520
C*****                  10S01530
C      SUBROUTINE: RCOSTX          * 10S01540
C                                * 10S01550
C      THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A    * 10S01560
C      RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A        * 10S01570
C      X/SINX APERTURE EQUALIZER          * 10S01580
C*****                  10S01590
      SUBROUTINE RCOSTX(TF)          10S01600
      COMPLEX TF( 1)                10S01610
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL          10S01620
      COMMON /NUMB2/ NSNR,NSYMB,BAUD          10S01630
      SBANDW=BAUD*LSAMPL          10S01640
      NO=LDIM/2                    10S01650
      NO1=NO+1                    10S01660
      IF (ALPHA.EQ.0) ALPHA=0.0001          10S01670
      FN=LDIM*(FBW/SBANDW)          10S01680
      F1=(1.-ALPHA)*FN          10S01690
      F2=(1.+ALPHA)*FN          10S01700
      IFN=IFIX(FN)                10S01710
      IF1=IFIX(F1)+1              10S01720
      IF2=IFIX(F2)+1              10S01730
C                                10S01740
C      THE AMPLITUDE CHARACTERISTICS          10S01750
C                                10S01760
      A1=3.1415926/(2.*FLOAT(IFN))          10S01770
      DO 8 I=2,IF1                10S01780
      TF(I)=CMPLX(1.0,0.0)          10S01790
      J=I-1                      10S01800
C                                10S01810
C      X/SIN(X) EQUALIZATION          10S01820
C                                10S01830
      A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))          10S01840
      TF(I)=CMPLX(A2,0.0)          10S01850
8 CONTINUE                    10S01860
      JK=IF1+1                    10S01870
      DO 9 J=JK,IF2                10S01880
      I=J-1                      10S01890
C                                10S01900
C      ROOT OF RAISED COSINE          10S01910
C                                10S01920
      A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))          10S01930
      A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)          10S01940
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)          10S01950
      TF(J)=TF(J)*CMPLX(A3,0.0)          10S01960
9 CONTINUE                    10S01970
      JH=IF2+1                    10S01980
      DO 10 I=JH,NO1                10S01990
      TF(I)=CMPLX(0.0,0.0)          10S02000
10 CONTINUE                    10S02010
      NO2=NO1+1                    10S02020
      DO 5 I=NO2,LDIM                10S02030
      TF(I)=CONJG(TF(LDIM+2-I))          10S02040
5 CONTINUE                    10S02050
  RETURN                      10S02060
  END                        10S02070
C*****                  10S02080
C      SUBROUTINE: RCOSRX          * 10S02090
C                                * 10S02100
C      THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A    * 10S02110
C      RAISED COSINE FILTER WITH ARBITRARY ALPHA          * 10S02120
C*****                  10S02130
      SUBROUTINE RCOSRX(TF)          10S02140
      COMPLEX TF( 1)                10S02150
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL          10S02160
```

```

COMMON /NUMB2/ NSNR, NSYMB, BAUD
SBANDW=BAUD*LSAMPL
NO=LDIM/2
NO1=NO+1
IF (ALPHA.EQ.0) ALPHA=0.0001
FN=LDIM*(FBW/SBANDW)
F1=(1.-ALPHA)*FN
F2=(1.+ALPHA)*FN
IFN=IFIX(FN)
IF1=IFIX(F1)+1
IF2=IFIX(F2)+1
C
C AMPLITUDE CHARACTERISTIC
C
A1=3.141592/(2.*FLOAT(IFN))
DO 8 I=2, IF1
TF(I)=CMPLX(1.0,0.0)
J=I-1
TF(I)=CMPLX(1.,0.0)
8 CONTINUE
JK=IF1+1
DO 9 J=JK, IF2
I=J-1
C
C ROOT OF RAISED COSINE
C
A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9 CONTINUE
JH=IF2+1
DO 10 I=JH, NO1
TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
NO2=NO1+1
DO 5 I=NO2, LDIM
TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
RETURN
END
C*****
C SUBROUTINE: FILTER *
C *
C THIS SUBROUTINE PERFORMS THE FILTERING *
C *
C PROCESS ON A DATA SEQUENCE *
C*****
SUBROUTINE FILTER(DATA, TF)
COMPLEX DATA(1), TF(1)
DIMENSION IWK(20)
COMMON /NUMB1/ FBW, ALPHA, LDIM, IOFF, LSAMPL
CALL FFT2C(DATA, 19, IWK)
DO 1 I=1, LDIM
1 DATA(I)=CONJG(DATA(I)*TF(I))
CALL FFT2C(DATA, 19, IWK)
DO 2 I=1, LDIM
2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
RETURN
END
C*****
C SUBROUTINE: HHGG *
C *
C THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER *
C *
C AT THE OUTPUT OF THE RECEIVE FILTER *
C*****
SUBROUTINE HHGG(TF, PNOISE)
COMPLEX TF(1)
COMMON /NUMB1/ FBW, ALPHA, LDIM, IOFF, LSAMPL
COMMON /NUMB2/ NSNR, NSYMB, BAUD
SUM=0.0
SBANDW=BAUD*LSAMPL
DO 1 L=1, LDIM
HH=(CABS(TF(L)))**2
1 SUM=SUM+HH

```

```

10S02170
10S02180
10S02190
10S02200
10S02210
10S02220
10S02230
10S02240
10S02250
10S02260
10S02270
10S02280
10S02290
10S02300
10S02310
10S02320
10S02330
10S02340
10S02350
10S02360
10S02370
10S02380
10S02390
10S02400
10S02410
10S02420
10S02430
10S02440
10S02450
10S02460
10S02470
10S02480
10S02490
10S02500
10S02510
10S02520
10S02530
10S02540
10S02550
10S02560
10S02570
10S02580
10S02590
10S02600
10S02610
10S02620
10S02630
10S02640
10S02650
10S02660
10S02670
10S02680
10S02690
10S02700
10S02710
10S02720
10S02730
10S02740
10S02750
10S02760
10S02770
10S02780
10S02790
10S02800
10S02810
10S02820
10S02830
10S02840
10S02850
10S02860
10S02870
10S02880

```

```

      PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.
      WRITE (6,2) PNOISE
2     FORMAT(5X,'PNOISE=',F7.3,/)
      RETURN
      END
C*****
C     SUBROUTINE: ENERGY
C
C     THIS SUBROUTINE COMPUTES THE CARRIER POWER
C     AND THE AVERAGE BIT ENERGY
C*****
      SUBROUTINE ENERGY(DATA,C,EB)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      COMPLEX DATA(1)
      WATTS=0.
      DO 1 I=1,LDIM
      WATTS=WATTS+CABS(DATA(I))**2
1     CONTINUE
      C=WATTS/FLOAT(LDIM)/2.
      RRK=RK/2.
      EB=C/(RRK*BAUD)
      WRITE (6,40) C,EB
40    FORMAT(10X,'C=',F15.6,5X,'EB=',E15.8)
      RETURN
      END
C*****
C     SUBROUTINE: SYNCRO
C
C     THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA
C*****
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      INTEGER QMAXF
      COMPLEX DATA(1),AMP
      DIMENSION NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      NERROR=0
      IF (IOFF.EQ.0) GO TO 111
      DO 6 K=1,IOFF
      XX=AIMAG(DATA(LDIM))
      DO 7 KK=2,LDIM
7     DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))
8     DATA(1)=CMPLX(REAL(DATA(1)),XX)
C
C     SYNCHRONIZE THE RECEIVED DATA
C
111  NOLD=0
      NOF=0
      K=1
300  CONTINUE
      NEW=0
      DO 200 J=1,NSYMB
      THR1I=FLOAT(NI(J))-1.
      THR2I=FLOAT(NI(J))+1.
      THR1Q=FLOAT(NQ(J))-1.
      THR2Q=FLOAT(NQ(J))+1.
      J1=K+(J-1)*LSAMPL
      IF (J1.GT.LDIM) J1=J1-LDIM
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      IF ((AXBAR.GT.THR1I).AND.(AXBAR.LT.THR2I)) NEW=NEW+1
      IF ((AYBAR.GT.THR1Q).AND.(AYBAR.LT.THR2Q)) NEW=NEW+1
200  CONTINUE
      IF (NOLD.GE.NEW) GO TO 399
      NOLD=NEW
      NOF=K
399  K=K+1
      LFU=2*NSYMB

```

10S02890
 10S02900
 10S02910
 10S02920
 10S02930
 10S02940
 * 10S02950
 * 10S02960
 * 10S02970
 * 10S02980
 10S02990
 10S03000
 10S03010
 10S03020
 10S03030
 10S03040
 10S03050
 10S03060
 10S03070
 10S03080
 10S03090
 10S03100
 10S03110
 10S03120
 10S03130
 10S03140
 10S03150
 10S03160
 10S03170
 * 10S03180
 * 10S03190
 * 10S03200
 10S03210
 10S03220
 10S03230
 10S03240
 10S03250
 10S03260
 10S03270
 10S03280
 10S03290
 10S03300
 10S03310
 10S03320
 10S03330
 10S03340
 10S03350
 10S03360
 10S03370
 10S03380
 10S03390
 10S03400
 10S03410
 10S03420
 10S03430
 10S03440
 10S03450
 10S03460
 10S03470
 10S03480
 10S03490
 10S03500
 10S03510
 10S03520
 10S03530
 10S03540
 10S03550
 10S03560
 10S03570
 10S03580
 10S03590
 10S03600

```

C      SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP
C
C      _____
      NSFT=LSAMPL*6
      IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300
      NOF=NOF-1
      KT=K-2
      WRITE(6,56) NOLD,NOF,KT
56  FORMAT(1X,'NO OF MAX. CORRECT SYMBOLS = ',I6, '//,1X,
      *'NO OF SHIFT IN SAMPLES FOR MAX. CORRECT SYMBOLS= ',I4, '//,
      *'1X,'TOTAL NO OF SHIFT FOR SEARCHING = ',I4)
C      _____
      IF (NOF.EQ.0) GO TO 230
      LO=LDIM-1
      DO 250 I=1,NOF
      AMP=DATA(1)
      DO 240 J=1,LO
      DATA(J)=DATA(J+1)
240  CONTINUE
      DATA(LDIM)=AMP
250  CONTINUE
C
C      OPTIMIZE THE SAMPLING INSTANT
C
230  MI=1
      MQ=1
      EOI=FLOAT(NSYMB) + 1.
      EQQ=FLOAT(NSYMB) + 1.
      M=22
      VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
      SIGMA=SQRT(VARIAN)
      MOS=LSAMPL/4
      DO 80 J=1,MOS
      EI=0.
      EQ=0.
      IERX=0
      IERY=0
      DO 70 K=1,NSYMB
      VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
      J1=(K-1)*LSAMPL+J
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      INDEXI=0
      AMPX=ABS(AXBAR)
      AMPI=ABS(FLOAT(NI(K)))
      THR1I=AMPI - 1.0
      THR2I=AMPI + 1.0
      IMAXF=0
      IF (AMPI.EQ.IMAX) GO TO 30
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30  IF (AMPX.LE.THR1I) INDEXI=1
      IMAXF=1
40  CONTINUE
      INDEXQ=0
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      QMAXF=0
      IF (AMPQ.EQ.QMAX) GO TO 50
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50  IF (AMPY.LE.THR1Q) INDEXQ=1
      QMAXF=1
60  CONTINUE
      IF (INDEXI.EQ.1) EI=EI + 1.
      IF (INDEXQ.EQ.1) EQ=EQ + 1.
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB
C

```

```

10S03610
10S03620
10S03630
10S03640
10S03650
10S03660
10S03670
10S03680
10S03690
10S03700
10S03710
10S03720
10S03730
10S03740
10S03750
10S03760
10S03770
10S03780
10S03790
10S03800
10S03810
10S03820
10S03830
10S03840
10S03850
10S03860
10S03870
10S03880
10S03890
10S03900
10S03910
10S03920
10S03930
10S03940
10S03950
10S03960
10S03970
10S03980
10S03990
10S04000
10S04010
10S04020
10S04030
10S04040
10S04050
10S04060
10S04070
10S04080
10S04090
10S04100
10S04110
10S04120
10S04130
10S04140
10S04150
10S04160
10S04170
10S04180
10S04190
10S04200
10S04210
10S04220
10S04230
10S04240
10S04250
10S04260
10S04270
10S04280
10S04290
10S04300
10S04310
10S04320

```

```

      D1=ABS(AMPX-THR11)
      ARG=D1/(SIGMA*SQRT(2.))
C     CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
      IF (IMAXF.EQ. 1) GO TO 65
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
65   D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IF (QMAXF.EQ. 1) GO TO 70
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IERX=IERX+INDEXI
      IERY=IERY+INDEXQ
70   CONTINUE
      WRITE(6,81) IERX,IERY
81   FORMAT(5X,'IERX=',I5,5X,'IERY=',I5)
      IF(EOI.LE.EI) GO TO 75
      EOI=EI
      MI=J
75   CONTINUE
      IF (EOQ.LE.EQ) GO TO 80
      EQQ=EQ
      MQ=J
80   MOFF=IABS(MI-MQ)
      WRITE(6,91) MI,MQ
91   FORMAT(5X,'MI=',I3,5X,'MQ=',I3)
      IF(MOFF.NE.0) WRITE(6,90) MOFF
      OFF=(FLOAT(MI) + FLOAT(MQ))/FLOAT(LSAMPL)
90   FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
      #12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
      WRITE(6,95) OFF
95   FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
      RETURN
      END
C*****
C     SUBROUTINE: DECODE
C
C     THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA
C*****
      SUBROUTINE DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)
      INTEGER QMAXF
      COMPLEX DATA(1),AMP
      DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      NERROR=0
      DO 1 I=1,NSNR
1     PE(I)=0.
      COMMON IMAX,QMAX,EBNO1,RK,CN1
C     CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE
C     ERRORS IN ADJACENT LEVELS ARE CONSIDERED
3     DO 2 K=1,NSYMB
          J1=(K-1)*LSAMPL+MI
          J2=(K-1)*LSAMPL + MQ
          AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
          AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.
          INDEXI=0
          AMPX=ABS(AXBAR)
          AMPI=ABS(FLOAT(NI(K)))
          THR1I=AMPI - 1.0
          THR2I=AMPI + 1.0
          IMAXF=0
          IF (AMPI.EQ.IMAX) GO TO 30

```

10S04330
 10S04340
 10S04350
 10S04360
 10S04370
 10S04380
 10S04390
 10S04400
 10S04410
 10S04420
 10S04430
 10S04440
 10S04450
 10S04460
 10S04470
 10S04480
 10S04490
 10S04500
 10S04510
 10S04520
 10S04530
 10S04540
 10S04550
 10S04560
 10S04570
 10S04580
 10S04590
 10S04600
 10S04610
 10S04620
 10S04630
 10S04640
 10S04650
 10S04660
 10S04670
 10S04680
 10S04690
 10S04700
 10S04710
 10S04720
 10S04730
 10S04740
 10S04750
 10S04760
 10S04770
 10S04780
 10S04790
 10S04800
 10S04810
 10S04820
 10S04830
 10S04840
 10S04850
 10S04860
 10S04870
 10S04880
 10S04890
 10S04900
 10S04910
 10S04920
 10S04930
 10S04940
 10S04950
 10S04960
 10S04970
 10S04980
 10S04990
 10S05000
 10S05010
 10S05020
 10S05030
 10S05040


```

      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30  IF(AMPX.LE.THR1I) INDEXI=1
      IMAXF=1
40  CONTINUE
      INDEXQ=0
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      QMAXF=0
      IF (AMPQ.EQ.QMAX) GO TO 50
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50  IF(AMPY.LE.THR1Q) INDEXQ=1
      QMAXF=1
60  CONTINUE
      IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.
C      THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS
C      1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE.
C      FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQSRT(2)))
C      AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +
C      1/2*ERFC(D2/(SIGMA*SQRT(2))).
C
      DO 4 M=1,NSNR
      EBNO(M)=EBNO1+1.0*FLOAT(M-1)
      VARIAN=PNOISE*EB*(10.**(-0.1*EBNO(M)))
      SIGMA=SQRT(VARIAN)
      D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C      CHECK IF ARG IS LARGE IN WHICH CASE PE IS INSIGNIFICANT
C      CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN
C
      NERI=K
      IF (INDEXI.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEI=ERFC(ARG)/2.
      IF (IMAXF .EQ. 1) GO TO 100
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEI=PEI + ERFC(ARG)/2.
100  D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (INDEXQ.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEQ=ERFC(ARG)/2.
      IF (QMAXF .EQ. 1) GO TO 110
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEQ=PEQ + ERFC(ARG)/2.
110  IF(PEI.LT.1.E-15) PEI=0.
      IF(PEQ.LT.1.E-15) PEQ=0.
      4  PE(M)=PE(M)+(PEI+PEQ)/2.
      2  CONTINUE
      DO 5 I=1,NSNR
      5  PE(I)=PE(I)/FLOAT(NSYMB)
C      PRINT 150
C150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)
C      WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)
C151  FORMAT(5X,F5.1,10X,E10.2)
      WRITE (6,152) NERROR
152  FORMAT(/5X,'ERRORS=',I5,/)
      GO TO 155
153  PRINT 154
154  FORMAT (10X,'SYMBOL WAS IN ERROR. RUN WAS TERMINATED',/)
      WRITE(6,157) NERI
157  FORMAT(5X,'NERI=',I5)
155  CONTINUE

```

```

10S05050
10S05060
10S05070
10S05080
10S05090
10S05100
10S05110
10S05120
10S05130
10S05140
10S05150
10S05160
10S05170
10S05180
10S05190
10S05200
10S05210
10S05220
10S05230
10S05240
10S05250
10S05260
10S05270
10S05280
10S05290
10S05300
10S05310
10S05320
10S05330
10S05340
10S05350
10S05360
10S05370
10S05380
10S05390
10S05400
10S05410
10S05420
10S05430
10S05440
10S05450
10S05460
10S05470
10S05480
10S05490
10S05500
10S05510
10S05520
10S05530
10S05540
10S05550
10S05560
10S05570
10S05580
10S05590
10S05600
10S05610
10S05620
10S05630
10S05640
10S05650
10S05660
10S05670
10S05680
10S05690
10S05700
10S05710
10S05720
10S05730
10S05740
10S05750
10S05760

```

```

      RETURN
      END
C*****
C      SUBROUTINE: DRAWCN
C
C      THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C      P(E) AS LOW AS 1.0E-10 AND A C/N RATIO AS HIGH AS 42 DB
C*****
      SUBROUTINE DRAWCN(XARRAY,YARRAY,JCURV,JLAST,A)
      DIMENSION XARRAY(1),YARRAY(1),XA(150),XX(150),YY(150)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      M=NSNR
      BR=RK*BAUD
      NBW=BAUD/A
      CF=BR/NBW
      DO 1 K = 1, NSNR
      XA(K)=XARRAY( K ) + 10*(ALOG10(CF))
      IF(YARRAY( K ).LE.1.0E-10) GOTO 3
      GOTO 1
3     M= K-1
      GOTO 4
1     CONTINUE
4     CONTINUE
      I=M+1
      II=I+1
      XA(I)=9.0
      XA(II)=1.0
      YARRAY(I)=1.E-10
      YARRAY(II)=0.4
CC     GO TO 2
      IF(JCURV.GT.1) GOTO 2
C
C     ESTABLISH THE SURFACE AREA.
C
      CALL PLOTS(57.0,27.5)
      CALL FACTOR(0.85)
C
C     ESTABLISH THE ORIGIN.
C
      CALL PLOT(3.0,1.5,-3)
C
C     DRAW THE LOGARITHMIC Y-AXIS.
C
      CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,25.,
+90.,1.0E-10,.4)
C
C     DRAW THE LINEAR X-AXIS.
C
      CALL AXIS(0.0,0.0,9HC/N IN dB,-9,
+48.0,0.0,9.0,1.0)
C
C     PLOT THE DATA IN LOG-LINEAR MODE.
C
2     CONTINUE
      PRINT 150
150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',7X,'C/N',/)
      WRITE(6,151) (XARRAY(I),YARRAY(I),XA(I),I=1,M)
151  FORMAT(5X,F5.0,10X,E10.2,10X,F5.1)
CC     GO TO 45
      CALL LGLIN(XA,YARRAY,M,1,4,JCURV,1)
      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
45  RETURN
      END
C*****
C      SUBROUTINE: EYEQ
C
C      THIS SUBROUTINE DRAWS THE EYE DIAGRAM
C      FOR A TWO SYMBOL DURATION
C*****
      SUBROUTINE EYEQ(DATA)
      DIMENSION DATA2(66),DATA3(66)

```

```

10S05770
10S05780
10S05790
10S05800
10S05810
10S05820
10S05830
10S05840
10S05850
10S05860
10S05870
10S05880
10S05890
10S05900
10S05910
10S05920
10S05930
10S05940
10S05950
10S05960
10S05970
10S05980
10S05990
10S06000
10S06010
10S06020
10S06030
10S06040
10S06050
10S06060
10S06070
10S06080
10S06090
10S06100
10S06110
10S06120
10S06130
10S06140
10S06150
10S06160
10S06170
10S06180
10S06190
10S06200
10S06210
10S06220
10S06230
10S06240
10S06250
10S06260
10S06270
10S06280
10S06290
10S06300
10S06310
10S06320
10S06330
10S06340
10S06350
10S06360
10S06370
10S06380
10S06390
10S06400
10S06410
10S06420
10S06430
10S06440
10S06450
10S06460
10S06470
10S06480

```

```

      COMPLEX DATA(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
C     ESTABLISH THE SURFACE AREA.
      CALL PLOTS(37.0,27.5)
C     ESTABLISH THE ORIGIN.
      CALL PLOT(3.0,13.0,-3)
C     WRITE THE TITLE OF THE GRAPH.
      CALL SYMBOL(10.0,12.0,0.49,16HQPSK EYE DIAGRAM,0.0,16)
C     DRAW THE TIME AXIS.
      CALL AXIS(0.0,0.0,1H,-1,24.,0.0,1.0,2.8)
C     DRAW THE AMPLITUDE AXIS.
      CALL AXIS(0.0,-8.00,1H,1,16.0,90.0,-24.,3.0)
C     PLOT THE DATA.
      JJ=2*LSAMPL
      DATA2(JJ+1)=0.0
      DATA2(JJ+2)=3.0
      DATA3(JJ+1)=1.0
      DATA3(JJ+2)=2.8
      M2=NSYMB/8
      M3=3*M2+1
      M4=4*M2-1
      DO 4 KK=1,M2
      DO 2 I=1,JJ
      II=(KK-1)*LSAMPL+I
      IF(II.GT.LDIM) II=II-(II/LDIM)*LDIM
      DATA2(I)=REAL(DATA(II))
      DATA3(I)=FLOAT(I)
      CALL LINE(DATA3,DATA2,JJ,1,0,0)
4     CONTINUE
      CALL PLOT(0.0,0.0,999)
      RETURN
      END
C*****
C     SUBROUTINE: ATT
C
C     THIS SUBROUTINE PERFORMS THE ATTENUATION OF A COMPLEX
C     SIGNAL (AT IS THE ATTENUATION FACTOR IN DB)
C*****
      SUBROUTINE ATT(DATA,AT)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMPLEX DATA(1)
      ATTT=10.**(-AT/20.)
      DO 10 I=1,LDIM
10    DATA(I)=DATA(I)*ATTT
      RETURN
      END
C*****
C     SUBROUTINE: OFFSET
C
C     THIS SUBROUTINE SHIFTS THE DATA SEQUENCE BY IOF
C*****
      SUBROUTINE OFFSET(DATA,IOF)
      COMPLEX DATA(1),BUFF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
      J=LDIM-IABS(IOF)
      K=J+1
      IF (IOF.EQ.0) GOTO 9
      IF (IOF.LT.0) GOTO 4
      DO 1 I=1,J
1    BUFF(I)=DATA(I)
      DO 2 I=K,LDIM
2    DATA(I-J)=DATA(I)
      DO 3 I=1,J
3    DATA(I+IOF)=BUFF(I)
      GOTO 9
4    IOF=-IOF
      L=IOF+1
      DO 5 I=L,LDIM
5    BUFF(I-IOF)=DATA(I)
      DO 6 I=1,IOF

```

```

10S06490
10S06500
10S06510
10S06520
10S06530
10S06540
10S06550
10S06560
10S06570
10S06580
10S06590
10S06600
10S06610
10S06620
10S06630
10S06640
10S06650
10S06660
10S06670
10S06680
10S06690
10S06700
10S06710
10S06720
10S06730
10S06740
10S06750
10S06760
10S06770
10S06780
10S06790
10S06800
10S06810
10S06820
10S06830
10S06840
10S06850
10S06860
10S06870
10S06880
10S06890
10S06900
10S06910
10S06920
10S06930
10S06940
10S06950
10S06960
10S06970
10S06980
10S06990
10S07000
10S07010
10S07020
10S07030
10S07040
10S07050
10S07060
10S07070
10S07080
10S07090
10S07100
10S07110
10S07120
10S07130
10S07140
10S07150
10S07160
10S07170
10S07180
10S07190
10S07200

```

```

6 DATA(I+J)=DATA(I) 10S07210
DO 7 I=1,J 10S07220
7 DATA(I)=BUFF(I) 10S07230
9 RETURN 10S07240
END 10S07250
C***** 10S07260
C SUBROUTINE: PHASE * 10S07270
C * 10S07280
C THIS SUBROUTINE PERFORMS THE PHASE SHIFT * 10S07290
C OF A COMPLEX DATA SEQUENCE * 10S07300
C***** 10S07310
SUBROUTINE PHASE(DATA,PHI) 10S07320
COMPLEX EPS,DATA(1) 10S07330
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 10S07340
EPS=CMPLX(COS(PHI),-SIN(PHI)) 10S07350
DO 10 I=1,LDIM 10S07360
10 DATA(I)=DATA(I)*EPS 10S07370
RETURN 10S07380
END 10S07390
C***** 10S07400
C SUBROUTINE: RAND * 10S07410
C * 10S07420
C THIS SUBROUTINE GENERATES RANDOM SYMBOLS * 10S07430
C***** 10S07440
SUBROUTINE RAND(IR,N,DSEED) 10S07450
INTEGER IR(1) 10S07460
DOUBLE PRECISION DSEED 10S07470
DO 3000 I=1,N 10S07480
R=GGUBFS(DSEED) 10S07490
IR(I)=IFIX(R) 10S07500
IF (R.LE.0.031) IR(I)=31 10S07510
IF ((R.GT.0.031).AND. (R.LE.0.082)) IR(I)=29 10S07520
IF ((R.GT.0.082).AND. (R.LE.0.093)) IR(I)=27 10S07530
IF ((R.GT.0.093).AND. (R.LE.0.124)) IR(I)=25 10S07540
IF ((R.GT.0.124).AND. (R.LE.0.155)) IR(I)=23 10S07550
IF ((R.GT.0.155).AND. (R.LE.0.186)) IR(I)=21 10S07560
IF ((R.GT.0.186).AND. (R.LE.0.217)) IR(I)=19 10S07570
IF ((R.GT.0.217).AND. (R.LE.0.248)) IR(I)=17 10S07580
IF ((R.GT.0.248).AND. (R.LE.0.279)) IR(I)=15 10S07590
IF ((R.GT.0.279).AND. (R.LE.0.31)) IR(I)=13 10S07600
IF ((R.GT.0.31).AND. (R.LE.0.341)) IR(I)=11 10S07610
IF ((R.GT.0.341).AND. (R.LE.0.372)) IR(I)=9 10S07620
IF ((R.GT.0.372).AND. (R.LE.0.403)) IR(I)=7 10S07630
IF ((R.GT.0.403).AND. (R.LE.0.434)) IR(I)=5 10S07640
IF ((R.GT.0.434).AND. (R.LE.0.465)) IR(I)=3 10S07650
IF ((R.GT.0.465).AND. (R.LE.0.496)) IR(I)=1 10S07660
IF ((R.GT.0.496).AND. (R.LE.0.527)) IR(I)=1 10S07670
IF ((R.GT.0.527).AND. (R.LE.0.558)) IR(I)=3 10S07680
IF ((R.GT.0.558).AND. (R.LE.0.589)) IR(I)=5 10S07690
IF ((R.GT.0.589).AND. (R.LE.0.62)) IR(I)=7 10S07700
IF ((R.GT.0.62).AND. (R.LE.0.651)) IR(I)=9 10S07710
IF ((R.GT.0.651).AND. (R.LE.0.682)) IR(I)=11 10S07720
IF ((R.GT.0.682).AND. (R.LE.0.713)) IR(I)=13 10S07730
IF ((R.GT.0.713).AND. (R.LE.0.744)) IR(I)=15 10S07740
IF ((R.GT.0.744).AND. (R.LE.0.775)) IR(I)=17 10S07750
IF ((R.GT.0.775).AND. (R.LE.0.806)) IR(I)=19 10S07760
IF ((R.GT.0.806).AND. (R.LE.0.837)) IR(I)=21 10S07770
IF ((R.GT.0.837).AND. (R.LE.0.868)) IR(I)=23 10S07780
IF ((R.GT.0.868).AND. (R.LE.0.899)) IR(I)=25 10S07790
IF ((R.GT.0.899).AND. (R.LE.0.930)) IR(I)=27 10S07800
IF ((R.GT.0.930).AND. (R.LE.0.961)) IR(I)=29 10S07810
IF ((R.GT.0.961).AND. (R.LE.0.996)) IR(I)=31 10S07820
IF (R.GT.0.996) IR(I)=31 10S07830
3000 CONTINUE 10S07840
RETURN 10S07850
END 10S07860

```

```

C***** 10C00010
C* 10C00020
C* FILE NAME: 51CCI 10C00030
C* 10C00040
C* 10C00050
C* 10C00060
C* THIS PROGRAM EVALUATES THE PERFORMANCE OF 1024-QAM 10C00070
C* SYSTEMS IN THE PRESENCE OF MODULATED CO-CHANNEL 10C00080
C* OF THE SAME TYPE, I.E. 1024-QAM 10C00090
C* 10C00100
C***** 10C00110
C* COMPLEX DATA(524288),CCI(524288),TF(524288) 10C00120
C* COMPLEX XSIGNL(524288),XCCI(524288) 10C00130
C* DIMENSION PEI(50),EBNO(50),NI(8192),NQ(8192),NNI(8192),NNQ(8192) 10C00140
C* DIMENSION XARRAY(150),YARRAY(150),NIC(8192),NQC(8192) 10C00150
C* INTEGER IR(8192) 10C00160
C* REAL IBAUD,IMAX,QMAX 10C00170
C 10C00180
C INITIALIZE PROGRAM. 10C00190
C 10C00200
C COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 10C00210
C COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2 10C00220
C COMMON IMAX,QMAX,EBNO1,RK,CN1 10C00230
C 10C00240
C FOR 1024-QAM, NBPSYM=10, EBNO1=20., CN1=26., CDIM=42.5 10C00250
C 10C00260
C NBPSYM=10 10C00270
C EBNO1=20. 10C00280
C CN1=26. 10C00290
C CDIM=42.5 10C00300
C ATM=47.5 10C00310
C NLEVL=2** (NBPSYM/2) 10C00320
C IMAX=FLOAT(NLEVL)-1. 10C00330
C QMAX=IMAX 10C00340
C RK=FLOAT(NBPSYM) 10C00350
C KQAM=NLEVL**2 10C00360
C 10C00370
C ALPHA=0.2 10C00380
C 10C00390
C IND=1 FOR PLOTTING THE EYE-DIAGRAM 10C00400
C IND=2 FOR PLOTTING THE P(E) 10C00410
C 10C00420
C IND=2 10C00430
C NRUNS=2 10C00440
C IOFF=0 10C00450
C LSAMPL=64 10C00460
C NSNR=45 10C00470
C NSYMB=8192 10C00480
C LDIM=LSAMPL*NSYMB 10C00490
C 10C00500
C THE BAUD RATE BAUD IS GIVEN IN KBAUD/S 10C00510
C 10C00520
C BAUD=160.0 10C00530
C SBANDW=BAUD*LSAMPL 10C00540
C NO1=LDIM/2+1 10C00550
C NO2=NO1+1 10C00560
C FBW=BAUD/2. 10C00570
C PI=3.14159 10C00580
C 10C00590
C END OF INITIALIZATION. 10C00600
C 10C00610
C START OF COMPUTATIONS. 10C00620
C 10C00630
C PHI= 0.20*PI 10C00640
C IOF= 32 10C00650
C WRITE(6,14) PHI, IOF 10C00660
C 14 FORMAT(1X,'PHI=',F10.4,2X,'RADS',5X,'IOF=',I8///) 10C00670
C 92 WRITE(6,245) KQAM 10C00680
C 245 FORMAT(5X,I5,'-QAM',/) 10C00690
C WRITE(6,45) ALPHA 10C00700
C 45 FORMAT(////,2X,'*****ALPHA=',F6.2) 10C00710
C 10C00720

```

```

DSEED = 135791.DO
CALL RAND (NI, NSYMB, DSEED)
DSEED = 780956.DO
CALL RAND (NQ, NSYMB, DSEED)
CALL SIGNAL (XSIGNL, NI, NQ)
C
DSEED = 148357.DO
CALL RAND (NIC, NSYMB, DSEED)
DSEED = 123338.DO
CALL RAND (NGC, NSYMB, DSEED)
CALL SIGNAL (XCCI, NIC, NGC)
CALL RCOSTX(TF)
C
CALL FILTER (XSIGNL, TF)
CALL FILTER (XCCI, TF)
C
C
C
70 DO 101 MM=1, 1
C
WRITE(6,102) IOF, PHI
102 FORMAT(1X,'IOF=',I6,5X,'PHI=',F10.4///)
C
DO 100 NR=1,NRUNS
C
C-----MAIN CHANNEL-----
C
DO 40 I=1,LDIM
40 DATA(I)=XSIGNL(I)
CALL ENERGY(DATA,C,EB)
IF(NR.EQ.1) GO TO 88
AT=ATM-(NR-2)*2.5
C-----CO-CHANNEL-----
DO 140 K=1,LDIM
140 CCI(K)=XCCI(K)
CALL ATT(CCI,AT)
CALL PHASE(CCI,PHI)
CALL OFFSET(CCI,IOF)
CALL ENERGY(CCI,CI,EBC)
CDI=10.*ALOG10(C/CI)
WRITE(6,199) CDI
199 FORMAT(5X,'C/I=',F5.1,'DB')
DO 77 I=1,LDIM
77 DATA(I)=DATA(I)+CCI(I)
C
C-----
C
END OF THE TRANSMITTER
C
C-----
C
88 CONTINUE
82 CALL RCOSRX(TF)
83 CALL HHGG(TF,PNOISE)
CALL FILTER(DATA,TF)
C
GO TO (26,27), IND
27 CALL SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
CALL DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PEI,EB)
CCC IF((NERROR.NE.0).AND.(NR.EQ.NRUNS)) CALL PLOT(0.0,0.0,999)
CC IF(NERROR.NE.0) GO TO 100
DO 99 I=1,NSNR
XARRAY(I)=EBNO(I)
YARRAY(I)=PEI(I)
99 A=1.
CALL DRAWCN(XARRAY,YARRAY,NR,NRUNS,A)
C
CALL DRAWEB(XARRAY,YARRAY,JCURV,JLAST)
GO TO 100
26 CALL EYEQ(DATA)
100 CONTINUE
PHI=PHI+0.1 *PI
101 CONTINUE
STOP
END

```

```

10C00730
10C00740
10C00750
10C00760
10C00770
10C00780
10C00790
10C00800
10C00810
10C00820
10C00830
10C00840
10C00850
10C00860
10C00870
10C00880
10C00890
10C00900
10C00910
10C00920
10C00930
10C00940
10C00950
10C00960
10C00970
10C00980
10C00990
10C01000
10C01010
10C01020
10C01030
10C01040
10C01050
10C01060
10C01070
10C01080
10C01090
10C01100
10C01110
10C01120
10C01130
10C01140
10C01150
10C01160
10C01170
10C01180
10C01190
10C01200
10C01210
10C01220
10C01230
10C01240
10C01250
10C01260
10C01270
10C01280
10C01290
10C01300
10C01310
10C01320
10C01330
10C01340
10C01350
10C01360
10C01370
10C01380
10C01390
10C01400
10C01410
10C01420
10C01430
10C01440

```

```

C***** 10C01450
C  SUBROUTINE:  SIGNAL * 10C01460
C * 10C01470
C  THIS SUBROUTINE FORMS A COMPLEX BASEBAND SIGNAL * 10C01480
C***** 10C01490
      SUBROUTINE SIGNAL(DATA,NI,NQ) 10C01500
      COMPLEX DATA(1) 10C01510
      DIMENSION NI(1),NQ(1) 10C01520
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 10C01530
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2 10C01540
      DO 2 I=1,NSYMB 10C01550
C  LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL 10C01560
      J1=(I-1)*LSAMPL+1 10C01570
      J2=I*LSAMPL 10C01580
      DO 3 J3=J1,J2 10C01590
      3 DATA(J3)=CMPLX(FLOAT(NI(1)),FLOAT(NQ(1))) 10C01600
      2 CONTINUE 10C01610
      RETURN 10C01620
      END 10C01630
C***** 10C01640
C  SUBROUTINE:  RCOSTX * 10C01650
C * 10C01660
C  THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A * 10C01670
C  RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A * 10C01680
C  X/SINX APERTURE EQUALIZER * 10C01690
C***** 10C01700
      SUBROUTINE RCOSTX(TF) 10C01710
      COMPLEX TF( 1) 10C01720
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL 10C01730
      COMMON /NUMB2/ NSNR,NSYMB,BAUD 10C01740
      SEANDW=BAUD*LSAMPL 10C01750
      NO=LDIM/2 10C01760
      NO1=NO+1 10C01770
      IF (ALPHA.EQ.0) ALPHA=0.0001 10C01780
      FN=LDIM*(FBW/SBANDW) 10C01790
      F1=(1.-ALPHA)*FN 10C01800
      F2=(1.+ALPHA)*FN 10C01810
      IFN=IFIX(FN) 10C01820
      IF1=IFIX(F1)+1 10C01830
      IF2=IFIX(F2)+1 10C01840
C  THE AMPLITUDE CHARACTERISTICS 10C01850
C  10C01860
C  A1=3.1415926/(2.*FLOAT(IFN)) 10C01870
      DO 8 I=2,IF1 10C01880
      TF(I)=CMPLX(1.0,0.0) 10C01890
      J=I-1 10C01900
C  X/SIN(X) EQUALIZATION 10C01910
C  10C01920
C  A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1)) 10C01940
      TF(I)=CMPLX(A2,0.0) 10C01950
      8 CONTINUE 10C01960
      JK=IF1+1 10C01970
      DO 9 J=JK,IF2 10C01980
      I=J-1 10C01990
C  ROOT OF RAISED COSINE 10C02000
C  10C02010
C  A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1)) 10C02020
      A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.) 10C02030
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0) 10C02040
      TF(J)=TF(J)*CMPLX(A3,0.0) 10C02050
      9 CONTINUE 10C02060
      JH=IF2+1 10C02070
      DO 10 I=JH,NO1 10C02080
      TF(I)=CMPLX(0.0,0.0) 10C02090
      10 CONTINUE 10C02100
      NO2=NO1+1 10C02110
      DO 5 I=NO2,LDIM 10C02120
      TF(I)=CONJG(TF(LDIM+2-I)) 10C02130
      5 CONTINUE 10C02140
      10C02150
      10C02160

```

```

      RETURN
      END
C*****
C      SUBROUTINE: RCOSRX
C
C      THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C      RAISED COSINE FILTER WITH ARBITRARY ALPHA
C*****
      SUBROUTINE RCOSRX(TF)
      COMPLEX TF( 1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SBANDW=BAUD*LSAMPL
      NO=LDIM/2
      NO1=NO+1
      IF (ALPHA.EQ.0) ALPHA=0.0001
      FN=LDIM*(FBW/SBANDW)
      F1=(1.-ALPHA)*FN
      F2=(1.+ALPHA)*FN
      IFN=IFIX(FN)
      IF1=IFIX(F1)+1
      IF2=IFIX(F2)+1
C
C      AMPLITUDE CHARACTERISTIC
C
      A1=3.141592/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
      TF(I)=CMPLX(1.0,0.0)
      J=I-1
      TF(I)=CMPLX(1.,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
      I=J-1
C
C      ROOT OF RAISED COSINE
C
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
      TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
      TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: FILTER
C
C      THIS SUBROUTINE PERFORMS THE FILTERING
C      PROCESS ON A DATA SEQUENCE
C*****
      SUBROUTINE FILTER(DATA,TF)
      COMPLEX DATA(1),TF(1)
      DIMENSION IWK(20)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      CALL FFT2C(DATA,19,IWK)
      DO 1 I=1,LDIM
1 DATA(I)=CONJG(DATA(I)*TF(I))
      CALL FFT2C(DATA,19,IWK)
      DO 2 I=1,LDIM
2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END
C*****
C      SUBROUTINE: HHGG
C
C      THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER

```

```

10C02170
10C02180
10C02190
10C02200
10C02210
10C02220
10C02230
10C02240
10C02250
10C02260
10C02270
10C02280
10C02290
10C02300
10C02310
10C02320
10C02330
10C02340
10C02350
10C02360
10C02370
10C02380
10C02390
10C02400
10C02410
10C02420
10C02430
10C02440
10C02450
10C02460
10C02470
10C02480
10C02490
10C02500
10C02510
10C02520
10C02530
10C02540
10C02550
10C02560
10C02570
10C02580
10C02590
10C02600
10C02610
10C02620
10C02630
10C02640
10C02650
10C02660
10C02670
10C02680
10C02690
10C02700
10C02710
10C02720
10C02730
10C02740
10C02750
10C02760
10C02770
10C02780
10C02790
10C02800
10C02810
10C02820
10C02830
10C02840
10C02850
10C02860
10C02870
10C02880

```



```

C      AT THE OUTPUT OF THE RECEIVE FILTER * 10C02890
C***** 10C02900
SUBROUTINE HHGG(TF,PNOISE) 10C02910
COMPLEX TF(1) 10C02920
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL 10C02930
COMMON /NUMB2/ NSNR,NSYMB,BAUD 10C02940
SUM=0.0 10C02950
SBANDW=BAUD*LSAMPL 10C02960
DO 1 L=1,LDIM 10C02970
HH=(CABS(TF(L)))**2 10C02980
1 SUM=SUM+HH 10C02990
PNOISE=SUM*SBANDW/FLOAT(LDIM)/2. 10C03000
WRITE (6,2) PNOISE 10C03010
2 FORMAT(5X,'PNOISE=',F7.3,/) 10C03020
RETURN 10C03030
END 10C03040
C***** 10C03050
C      SUBROUTINE: ENERGY * 10C03060
C * 10C03070
C      THIS SUBROUTINE COMPUTES THE CARRIER POWER * 10C03080
C      AND THE AVERAGE BIT ENERGY * 10C03090
C***** 10C03100
SUBROUTINE ENERGY(DATA,C,EB) 10C03110
COMMON /NUMB2/ NSNR,NSYMB,BAUD 10C03120
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL 10C03130
COMMON IMAX,QMAX,EBNO1,RK,CN1 10C03140
COMPLEX DATA(1) 10C03150
WATTS=0. 10C03160
DO 1 I=1,LDIM 10C03170
WATTS=WATTS+CABS(DATA(I))**2 10C03180
1 CONTINUE 10C03190
C=WATTS/FLOAT(LDIM)/2. 10C03200
RRK=RK/2. 10C03210
EB=C/(RRK*BAUD) 10C03220
WRITE (6,40) C,EB 10C03230
40 FORMAT(10X,'C=',F15.6,5X,'EB=',E15.8) 10C03240
RETURN 10C03250
END 10C03260
C***** 10C03270
C      SUBROUTINE: SYNCRO * 10C03280
C * 10C03290
C      THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA * 10C03300
C***** 10C03310
SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB) 10C03320
INTEGER QMAXF 10C03330
COMPLEX DATA(1),AMP 10C03340
DIMENSION NI(1),NQ(1) 10C03350
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL 10C03360
COMMON /NUMB2/ NSNR,NSYMB,BAUD 10C03370
COMMON IMAX,QMAX,EBNO1,RK,CN1 10C03380
NERROR=0 10C03390
IF (IOFF.EQ.0) GO TO 111 10C03400
DO 6 K=1,IOFF 10C03410
XX=AIMAG(DATA(LDIM)) 10C03420
DO 7 KK=2,LDIM 10C03430
7 DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1))) 10C03440
6 DATA(1)=CMPLX(REAL(DATA(1)),XX) 10C03450
C 10C03460
C      SYNCHRONIZE THE RECEIVED DATA 10C03470
C 10C03480
111 NOLD=0 10C03490
NOF=0 10C03500
K=1 10C03510
300 CONTINUE 10C03520
NEW=0 10C03530
DO 200 J=1,NSYMB 10C03540
THR1I=FLOAT(NI(J))-1. 10C03550
THR2I=FLOAT(NI(J))+1. 10C03560
THR1Q=FLOAT(NQ(J))-1. 10C03570
THR2Q=FLOAT(NQ(J))+1. 10C03580
J1=K+(J-1)*LSAMPL 10C03590
10C03600

```

```

      IF(J1.GT.LDIM) J1=J1-LDIM
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      IF((AXBAR.GT.THR1I).AND.(AXBAR.LT.THR2I)) NEW=NEW+1
      IF((AYBAR.GT.THR1Q).AND.(AYBAR.LT.THR2Q)) NEW=NEW+1
200  CONTINUE
      IF(NOLD.GE.NEW) GO TO 399
      NOLD=NEW
      NOF=K
399  K=K+1
      LFU=2*NSYMB
C
C      SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP
C
C      _____
      NSFT=LSAMPL*6
      IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300
      NOF=NOF-1
      KT=K-2
      WRITE(6,56) NOLD,NOF,KT
56  FORMAT(1X,'NO OF MAX. CORRECT SYMBOLS = ',I4,'//,1X,
      *'NO OF SHIFT IN SAMPLES FOR MAX. CORRECT SYMBOLS= ',I4,'//,
      *'1X,'TOTAL NO OF SHIFT FOR SEARCHING = ',I4)
C
C      _____
      IF (NOF.EQ.0) GO TO 230
      LO=LDIM-1
      DO 250 I=1,NOF
      AMP=DATA(1)
      DO 240 J=1,LO
      DATA(J)=DATA(J+1)
240  CONTINUE
      DATA(LDIM)=AMP
250  CONTINUE
C
C      OPTIMIZE THE SAMPLING INSTANT
C
C      _____
230  MI=1
      MQ=1
      EOI=FLOAT(NSYMB) + 1.
      EOQ=FLOAT(NSYMB) + 1.
      M=22
      VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
      SIGMA=SQRT(VARIAN)
      MOS=LSAMPL/4
      DO 80 J=1,MOS
      EI=0.
      EQ=0.
      IERX=0
      IERY=0
      DO 70 K=1,NSYMB
C      VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
      J1=(K-1)*LSAMPL+J
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      INDEXI=0
      AMPX=ABS(AXBAR)
      AMPI=ABS(FLOAT(NI(K)))
      THR1I=AMPI - 1.0
      THR2I=AMPI + 1.0
      IMAXF=0
      IF (AMPI.EQ.IMAX) GO TO 30
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30  IF(AMPX.LE.THR1I) INDEXI=1
      IMAXF=1
40  CONTINUE
      INDEXQ=0
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      QMAXF=0

```

```

10C03610
10C03620
10C03630
10C03640
10C03650
10C03660
10C03670
10C03680
10C03690
10C03700
10C03710
10C03720
10C03730
10C03740
10C03750
10C03760
10C03770
10C03780
10C03790
10C03800
10C03810
10C03820
10C03830
10C03840
10C03850
10C03860
10C03870
10C03880
10C03890
10C03900
10C03910
10C03920
10C03930
10C03940
10C03950
10C03960
10C03970
10C03980
10C03990
10C04000
10C04010
10C04020
10C04030
10C04040
10C04050
10C04060
10C04070
10C04080
10C04090
10C04100
10C04110
10C04120
10C04130
10C04140
10C04150
10C04160
10C04170
10C04180
10C04190
10C04200
10C04210
10C04220
10C04230
10C04240
10C04250
10C04260
10C04270
10C04280
10C04290
10C04300
10C04310
10C04320

```

```

      IF (AMPQ.EQ.QMAX) GO TO 50
      IF ((AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50  IF (AMPY.LE.THR1Q) INDEXQ=1
      QMAXF=1
60  CONTINUE
      IF (INDEXI.EQ.1) EI=EI + 1.
      IF (INDEXQ.EQ.1) EQ=EQ + 1.
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB
C
      D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C      CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
      IF (IMAXF.EQ.1) GO TO 65
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
65  D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IF (QMAXF.EQ.1) GO TO 70
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IERY=IERX+INDEXI
      IEFY=IERY+INDEXQ
70  CONTINUE
      WRITE(6,81) IERX,IERY
81  FORMAT(5X,'IERX=',I5,5X,'IERY=',I5)
      IF (EOI.LE.EI) GO TO 75
      EOI=EI
      MI=J
75  CONTINUE
      IF (EQQ.LE.EQ) GO TO 80
      EQQ=EQ
      MQ=J
80  MOFF=IABS(MI-MQ)
      WRITE(6,91) MI,MQ
91  FORMAT(5X,'MI=',I3,5X,'MQ=',I3)
      IF (MOFF.NE.0) WRITE(6,90) MOFF
      OFF=(FLOAT(MI) + FLOAT(MQ))/FLOAT(LSAMPL)
90  FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
      #12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
      WRITE(6,95) OFF
95  FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
      RETURN
      END
C*****
C      SUBROUTINE: DECODES
C
C      THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA
C*****
      SUBROUTINE DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)
      INTEGER QMAXF
      COMPLEX DATA(1),AMP
      DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      NERROR=0
      DO 1 I=1,NSNR
1    PE(I)=0.
      COMMON IMAX,QMAX,EBNO1,RK,CN1
C      CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE
C      ERRORS IN ADJACENT LEVELS ARE CONSIDERED
3    DO 2 K=1,NSYMB

```

```

10C04330
10C04340
10C04350
10C04360
10C04370
10C04380
10C04390
10C04400
10C04410
10C04420
10C04430
10C04440
10C04450
10C04460
10C04470
10C04480
10C04490
10C04500
10C04510
10C04520
10C04530
10C04540
10C04550
10C04560
10C04570
10C04580
10C04590
10C04600
10C04610
10C04620
10C04630
10C04640
10C04650
10C04660
10C04670
10C04680
10C04690
10C04700
10C04710
10C04720
10C04730
10C04740
10C04750
10C04760
10C04770
10C04780
10C04790
10C04800
10C04810
10C04820
10C04830
10C04840
10C04850
10C04860
10C04870
10C04880
10C04890
10C04900
10C04910
10C04920
10C04930
10C04940
10C04950
10C04960
10C04970
10C04980
10C04990
10C05000
10C05010
10C05020
10C05030
10C05040

```

```

J1=(K-1)*LSAMPL+MI
J2=(K-1)*LSAMPL + M0
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.
INDEXI=0
AMPX=ABS(AXBAR)
AMPI=ABS(FLOAT(NI(K)))
THR1I=AMPI - 1.0
THR2I=AMPI + 1.0
IMAXF=0
IF (AMPI.EQ.IMAX) GO TO 30
IF ((AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
GO TO 40
30 IF(AMPX.LE.THR1I) INDEXI=1
IMAXF=1
40 CONTINUE
INDEXQ=0
AMPY=ABS(AYBAR)
AMPQ=ABS(FLOAT(NQ(K)))
THR1Q=AMPQ - 1.0
THR2Q=AMPQ + 1.0
QMAXF=0
IF (AMPQ.EQ.QMAX) GO TO 50
IF ((AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
GO TO 60
50 IF(AMPY.LE.THR1Q) INDEXQ=1
QMAXF=1
60 CONTINUE
IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1
C
C COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.
C THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS
C 1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE.
C FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2)))
C AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +
C 1/2*ERFC(D2/(SIGMA*SQRT(2))).
C
DO 4 M=1,NSNR
EBNO(M)=EBNO1+1.0*FLOAT(M-1)
VARIAN=PNOISE*EB*(10.**(-0.1*EBNO(M)))
SIGMA=SQRT(VARIAN)
D1=ABS(AMPX-THR1I)
ARG=D1/(SIGMA*SQRT(2.))
C CHECK IF ARG IS LARGE IN WHICH CASE PE IS INSIGNIFICANT
C CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN
C
NERI=K
IF (INDEXI.EQ.1) GO TO 153
IF (ARG.GT.12.) ARG=12.
PEI=ERFC(ARG)/2.
IF (IMAXF.EQ.1) GO TO 100
D2=ABS(THR2I-AMPX)
ARG=D2/(SIGMA*SQRT(2.))
IF (ARG.GT.12.) ARG=12.
PEI=PEI + ERFC(ARG)/2.
100 D1=ABS(AMPY-THR1Q)
ARG=D1/(SIGMA*SQRT(2.))
IF (INDEXQ.EQ.1) GO TO 153
IF (ARG.GT.12.) ARG=12.
PEQ=ERFC(ARG)/2.
IF (QMAXF.EQ.1) GO TO 110
D2=ABS(THR2Q-AMPY)
ARG=D2/(SIGMA*SQRT(2.))
IF (ARG.GT.12.) ARG=12.
PEQ=PEQ + ERFC(ARG)/2.
110 IF(PEI.LT.1.E-15) PEI=0.
IF(PEQ.LT.1.E-15) PEQ=0.
4 PE(M)=PE(M)+(PEI+PEQ)/2.
2 CONTINUE
DO 5 I=1,NSNR
5 PE(I)=PE(I)/FLOAT(NSYMB)
C PRINT 150

```

```

10C05050
10C05060
10C05070
10C05080
10C05090
10C05100
10C05110
10C05120
10C05130
10C05140
10C05150
10C05160
10C05170
10C05180
10C05190
10C05200
10C05210
10C05220
10C05230
10C05240
10C05250
10C05260
10C05270
10C05280
10C05290
10C05300
10C05310
10C05320
10C05330
10C05340
10C05350
10C05360
10C05370
10C05380
10C05390
10C05400
10C05410
10C05420
10C05430
10C05440
10C05450
10C05460
10C05470
10C05480
10C05490
10C05500
10C05510
10C05520
10C05530
10C05540
10C05550
10C05560
10C05570
10C05580
10C05590
10C05600
10C05610
10C05620
10C05630
10C05640
10C05650
10C05660
10C05670
10C05680
10C05690
10C05700
10C05710
10C05720
10C05730
10C05740
10C05750
10C05760

```

```

C150 FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)
C WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)
C151 FORMAT(5X,F5.1,10X,E10.2)
WRITE (6,152) NERROR
152 FORMAT(/5X,'ERRORS=',I5,/)
GO TO 155
153 PRINT 154
154 FORMAT (10X,'SYMBOL WAS IN ERROR. RUN WAS TERMINATED',/)
WRITE(6,157) NERI
157 FORMAT(5X,'NERI=',I5)
155 CONTINUE
RETURN
END
C*****
C SUBROUTINE: DRAWCN *
C *
C THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR *
C P(E) AS LOW AS 1.0E-10 AND A C/N RATIO AS HIGH AS 42 DB *
C*****
SUBROUTINE DRAWCN(XARRAY,YARRAY,JCURV,JLAST,A)
DIMENSION XARRAY(1),YARRAY(1),XA(150),XX(150),YY(150)
COMMON /NUMB2/ NSNR,NSYMB,BAUD
COMMON IMAX,QMAX,EBNO1,RK,CN1
M=NSNR
BR=RK*BAUD
NBW=BAUD/A
CF=BR/NBW
DO 1 K = 1, NSNR
XA(K)=XARRAY( K ) + 10*(ALOG10(CF))
IF(YARRAY( K ).LE.1.0E-10) GOTO 3
GOTO 1
3 M= K-1
GOTO 4
1 CONTINUE
4 CONTINUE
I=M+1
II=I+1
XA(I)=9.0
XA(II)=1.0
YARRAY(I)=1.E-10
YARRAY(II)=0.4
CC GO TO 2
IF(JCURV.GT.1) GOTO 2
C
C ESTABLISH THE SURFACE AREA.
C
CALL PLOTS(55.0,27.5)
CALL FACTOR(0.85)
C
C ESTABLISH THE ORIGIN.
C
CALL PLOT(3.0,1.5,-3)
C
C DRAW THE LOGARITHMIC Y-AXIS.
C
CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,25.,
+90.,1.0E-10,.4)
C
C DRAW THE LINEAR X-AXIS.
C
CALL AXIS(0.0,0.0,9HC/N IN dB,-9,
+48.0,0.0,9.0,1.0)
C
C PLOT THE DATA IN LOG-LINEAR MODE.
C
2 CONTINUE
PRINT 150
150 FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',7X,'C/N',/)
WRITE(6,151) (XARRAY(I),YARRAY(I),XA(I),I=1,M)
151 FORMAT(5X,F5.0,10X,E10.2,10X,F5.1)
CC GO TO 45
CALL LGLIN(XA,YARRAY,M,1,4,JCURV,1)

```

10C05770
10C05780
10C05790
10C05800
10C05810
10C05820
10C05830
10C05840
10C05850
10C05860
10C05870
10C05880
10C05890
10C05900
10C05910
10C05920
10C05930
10C05940
10C05950
10C05960
10C05970
10C05980
10C05990
10C06000
10C06010
10C06020
10C06030
10C06040
10C06050
10C06060
10C06070
10C06080
10C06090
10C06100
10C06110
10C06120
10C06130
10C06140
10C06150
10C06160
10C06170
10C06180
10C06190
10C06200
10C06210
10C06220
10C06230
10C06240
10C06250
10C06260
10C06270
10C06280
10C06290
10C06300
10C06310
10C06320
10C06330
10C06340
10C06350
10C06360
10C06370
10C06380
10C06390
10C06400
10C06410
10C06420
10C06430
10C06440
10C06450
10C06460
10C06470
10C06480

```

      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
45 RETURN
      END
C*****
C      SUBROUTINE: EYEQ
C
C      THIS SUBROUTINE DRAWS THE EYE DIAGRAM
C      FOR A TWO SYMBOL DURATION
C*****
      SUBROUTINE EYEQ(DATA)
      DIMENSION DATA2(66),DATA3(66)
      COMPLEX DATA(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
C      ESTABLISH THE SURFACE AREA.
      CALL PLOTS(37.0,27.5)
C      ESTABLISH THE ORIGIN.
      CALL PLOT(3.0,13.0,-3)
C      WRITE THE TITLE OF THE GRAPH.
C      CALL SYMBOL(10.0,12.0,0.49,16HQPSK EYE DIAGRAM,0.0,16)
C      DRAW THE TIME AXIS.
      CALL AXIS(0.0,0.0,1H,-1,24.,0.0,1.0,2.8)
C      DRAW THE AMPLITUDE AXIS.
      CALL AXIS(0.0,-8.00,1H,1,16.0,90.0,-24.,3.0)
C      PLOT THE DATA.
      JJ=2*LSAMPL
      DATA2(JJ+1)=0.0
      DATA2(JJ+2)=3.0
      DATA3(JJ+1)=1.0
      DATA3(JJ+2)=2.8
      M2=NSYMB/8
      M3=3*M2+1
      M4=4*M2-1
      DO 4 KK=1,M2
      DO 2 I=1,JJ
      II=(KK-1)*LSAMPL+I
      IF(II.GT.LDIM) II=II-(II/LDIM)*LDIM
      DATA2(I)=REAL(DATA(II))
2      DATA3(I)=FLOAT(I)
      CALL LINE(DATA3,DATA2,JJ,1,0,0)
4      CONTINUE
      CALL PLOT(0.0,0.0,999)
      RETURN
      END
C*****
C      SUBROUTINE: ATT
C
C      THIS SUBROUTINE PERFORMS THE ATTENUATION OF A COMPLEX
C      SIGNAL (AT IS THE ATTENUATION FACTOR IN DB)
C*****
      SUBROUTINE ATT(DATA,AT)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMPLEX DATA(1)
      ATTT=10.**(-AT/20.)
      DO 10 I=1,LDIM
10 DATA(I)=DATA(I)*ATTT
      RETURN
      END
C*****
C      SUBROUTINE: OFFSET
C
C      THIS SUBROUTINE SHIFTS THE DATA SEQUENCE BY IOF
C*****
      SUBROUTINE OFFSET(DATA,IOF)
      COMPLEX DATA(1),BUFF(262144)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
      J=LDIM-IABS(IOF)
      K=J+1
      IF (IOF.EQ.0) GOTO 9
      IF (IOF.LT.0) GOTO 4
      DO 1 I=1,J

```

```

10C06490
10C06500
10C06510
10C06520
10C06530
10C06540
10C06550
10C06560
10C06570
10C06580
10C06590
10C06600
10C06610
10C06620
10C06630
10C06640
10C06650
10C06660
10C06670
10C06680
10C06690
10C06700
10C06710
10C06720
10C06730
10C06740
10C06750
10C06760
10C06770
10C06780
10C06790
10C06800
10C06810
10C06820
10C06830
10C06840
10C06850
10C06860
10C06870
10C06880
10C06890
10C06900
10C06910
10C06920
10C06930
10C06940
10C06950
10C06960
10C06970
10C06980
10C06990
10C07000
10C07010
10C07020
10C07030
10C07040
10C07050
10C07060
10C07070
10C07080
10C07090
10C07100
10C07110
10C07120
10C07130
10C07140
10C07150
10C07160
10C07170
10C07180
10C07190
10C07200

```

```

1  BUFF(I)=DATA(I)                                10C07210
DO 2 I=K,LDIM                                     10C07220
2  DATA(I-J)=DATA(I)                              10C07230
DO 3 I=1,J                                         10C07240
3  DATA(I+IOF)=BUFF(I)                            10C07250
GOTO 9                                             10C07260
4  IOF=-IOF                                         10C07270
L=IOF+1                                           10C07280
DO 5 I=L,LDIM                                     10C07290
5  BUFF(I-IOF)=DATA(I)                            10C07300
DO 6 I=1,IOF                                       10C07310
6  DATA(I+J)=DATA(I)                              10C07320
DO 7 I=1,J                                         10C07330
7  DATA(I)=BUFF(I)                                10C07340
9  RETURN                                           10C07350
END                                                10C07360
C*****                                           10C07370
C  SUBROUTINE: PHASE                               * 10C07380
C                                                    * 10C07390
C  THIS SUBROUTINE PERFORMS THE PHASE SHIFT        * 10C07400
C  OF A COMPLEX DATA SEQUENCE                     * 10C07410
C*****                                           10C07420
SUBROUTINE PHASE(DATA,PHI)                        10C07430
COMPLEX EPS,DATA(1)                              10C07440
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW  10C07450
EPS=CMPLX(COS(PHI),-SIN(PHI))                   10C07460
DO 10 I=1,LDIM                                    10C07470
10 DATA(I)=DATA(I)*EPS                           10C07480
RETURN                                             10C07490
END                                                10C07500
C*****                                           10C07510
C  SUBROUTINE: RAND                               * 10C07520
C                                                    * 10C07530
C  THIS SUBROUTINE GENERATES RANDOM SYMBOLS        * 10C07540
C*****                                           10C07550
SUBROUTINE RAND(IR,N,DSEED)                       10C07560
INTEGER IR(1)                                     10C07570
DOUBLE PRECISION DSEED                           10C07580
DO 3000 I=1,N                                     10C07590
R=GGUBFS(DSEED)                                  10C07600
IR(I)=IFIX(R)                                     10C07610
IF (R.LE.0.031) IR(I)=-31                         10C07620
IF ((R.GT.0.031).AND. (R.LE.0.062)) IR(I)=-29    10C07630
IF ((R.GT.0.062).AND. (R.LE.0.093)) IR(I)=-27    10C07640
IF ((R.GT.0.093).AND. (R.LE.0.124)) IR(I)=-25    10C07650
IF ((R.GT.0.124).AND. (R.LE.0.155)) IR(I)=-23    10C07660
IF ((R.GT.0.155).AND. (R.LE.0.186)) IR(I)=-21    10C07670
IF ((R.GT.0.186).AND. (R.LE.0.217)) IR(I)=-19    10C07680
IF ((R.GT.0.217).AND. (R.LE.0.248)) IR(I)=-17    10C07690
IF ((R.GT.0.248).AND. (R.LE.0.279)) IR(I)=-15    10C07700
IF ((R.GT.0.279).AND. (R.LE.0.31)) IR(I)=-13     10C07710
IF ((R.GT.0.31).AND. (R.LE.0.341)) IR(I)=-11     10C07720
IF ((R.GT.0.341).AND. (R.LE.0.372)) IR(I)=-9      10C07730
IF ((R.GT.0.372).AND. (R.LE.0.403)) IR(I)=-7      10C07740
IF ((R.GT.0.403).AND. (R.LE.0.434)) IR(I)=-5      10C07750
IF ((R.GT.0.434).AND. (R.LE.0.465)) IR(I)=-3      10C07760
IF ((R.GT.0.465).AND. (R.LE.0.496)) IR(I)=-1      10C07770
IF ((R.GT.0.496).AND. (R.LE.0.527)) IR(I)=1       10C07780
IF ((R.GT.0.527).AND. (R.LE.0.558)) IR(I)=3       10C07790
IF ((R.GT.0.558).AND. (R.LE.0.589)) IR(I)=5       10C07800
IF ((R.GT.0.589).AND. (R.LE.0.62)) IR(I)=7        10C07810
IF ((R.GT.0.62).AND. (R.LE.0.651)) IR(I)=9        10C07820
IF ((R.GT.0.651).AND. (R.LE.0.682)) IR(I)=11      10C07830
IF ((R.GT.0.682).AND. (R.LE.0.713)) IR(I)=13      10C07840
IF ((R.GT.0.713).AND. (R.LE.0.744)) IR(I)=15      10C07850
IF ((R.GT.0.744).AND. (R.LE.0.775)) IR(I)=17      10C07860
IF ((R.GT.0.775).AND. (R.LE.0.806)) IR(I)=19      10C07870
IF ((R.GT.0.806).AND. (R.LE.0.837)) IR(I)=21      10C07880
IF ((R.GT.0.837).AND. (R.LE.0.868)) IR(I)=23      10C07890
IF ((R.GT.0.868).AND. (R.LE.0.899)) IR(I)=25      10C07900
IF ((R.GT.0.899).AND. (R.LE.0.930)) IR(I)=27      10C07910
IF ((R.GT.0.930).AND. (R.LE.0.961)) IR(I)=29      10C07920

```

```
      IF((R.GT.0.961).AND . (R.LE.0.996))IR(I)=31
      IF (R.GT.0.996) IR(I)=31
3000  CONTINUE
      RETURN
      END
```

```
10C07930
10C07940
10C07950
10C07960
10C07970
```



```

C***** 10A00010
C*      * 10A00020
C*      * 10A00030
C*      * 10A00040
C*      * 10A00050
C*      * 10A00060
C*      * 10A00070
C*      * 10A00080
C*      * 10A00090
C*      * 10A00100
C***** 10A00110
C      * 10A00120
C      * 10A00130
C      * 10A00140
C      * 10A00150
C      * 10A00160
C      * 10A00170
C      * 10A00180
C      * 10A00190
C      * 10A00200
C      * 10A00210
C      * 10A00220
C      * 10A00230
C      * 10A00240
C      * 10A00250
C      * 10A00260
C      * 10A00270
C      * 10A00280
C      * 10A00290
C      * 10A00300
C      * 10A00310
C      * 10A00320
C      * 10A00330
C      * 10A00340
C      * 10A00350
C      * 10A00360
C      * 10A00370
C      * 10A00380
C      * 10A00390
C      * 10A00400
C      * 10A00410
C      * 10A00420
C      * 10A00430
C      * 10A00440
C      * 10A00450
C      * 10A00460
C      * 10A00470
C      * 10A00480
C      * 10A00490
C      * 10A00500
C      * 10A00510
C      * 10A00520
C      * 10A00530
C      * 10A00540
C      * 10A00550
C      * 10A00560
C      * 10A00570
C      * 10A00580
C      * 10A00590
C      * 10A00600
C      * 10A00610
C      * 10A00620
C      * 10A00630
C      * 10A00640
C      * 10A00650
C      * 10A00660
C      * 10A00670
C      * 10A00680
C      * 10A00690
C      * 10A00700
C      * 10A00710
C      * 10A00720

C      FILE NAME: 10ACI

C      THIS PROGRAM EVALUATES THE PERFORMANCE OF 1024-QAM
C      SYSTEMS IN THE PRESENCE OF TWO ADJACENT CHANNEL
C      INTERFERERS OF THE SAME MODULATION FORMAT (1024-QAM)

C*****
C      COMPLEX DATA(524288),CCI(524288),TF(524288)
C      COMPLEX XSIGNL(524288)
C      COMPLEX ACI1(524288), ACI2(524288)
C      DIMENSION NIA1(8192), NQA1(8192), NIA2(8192), NQA2(8192)
C      DIMENSION PEI(50),EBNO(50),NI(8192),NQ(8192),NNI(8192),NNQ(8192)
C      DIMENSION XARRAY(150),YARRAY(150),NIC(4096),NQC(8192)
C      INTEGER IR(8192)
C      REAL IBAUD,IMAX,QMAX

C      INITIALIZE PROGRAM.

C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
C      COMMON IMAX,QMAX,EBNO1,RK,CN1

C      FOR 1024-QAM, NBPSYM=10, EBNO1=20., CN1=26., CDIM=42.5

C      NBPSYM=10
C      EBNO1=20.
C      CN1=26.
C      CDIM=42.5
C      ATM=42.5
C      NLEVL=2**(NBPSYM/2)
C      IMAX=FLOAT(NLEVL)-1.
C      QMAX=IMAX
C      RK=FLOAT(NBPSYM)
C      KGAM=NLEVL**2

C      ALPHA=0.5

C      IND=1 FOR PLOTTING THE EYE-DIAGRAM
C      IND=2 FOR PLOTTING THE P(E)

C      IND=2
C      NRUNS=2
C      IOFF=0
C      LSAMPL=64
C      NSNR=35
C      NSYMB=8192
C      LDIM=LSAMPL*NSYMB

C      THE BAUD RATE BAUD IS GIVEN IN KBAUD/S

C      BAUD=100.0
C      SBANDW=BAUD*LSAMPL
C      NO1=LDIM/2+1
C      NO2=NO1+1
C      FBW=BAUD/2.
C      PI=3.14159

C      INITIALIZE THE DATA FOR THE TWO INTERFERERS

C      DATA AT1, AT2 / 20.0, 20.0/
C      FOF = 0.
C      FOF1 = 134.
C      FOF2 = - 134.
C      DF = 2.

C      START THE COMPUTATIONS

C      DO 101 IKKK =1, 7

```

```

      WRITE(6,83) FOF, FOF1, FOF2
83  FORMAT(1X,'FOF=',E11.4,3X,'FOF1=',E11.4,3X,'FOF2=',E11.4///)
      WRITE(6,84) BAUD
84  FORMAT(1X,'BAUDRATE=',E14.5///)
      PHI= 0.25*PI
      IOF= 32
      WRITE(6,14) PHI, IOF
14  FORMAT(1X,'PHI=',F10.4,2X,'RADS',5X,'IOF=',I8///)
92  WRITE(6,245) KQAM
245 FORMAT(5X,I5,'-QAM',/)
      WRITE(6,45) ALPHA
45  FORMAT(////,2X,'*****ALPHA=',F6.2)
C
C   MAIN CHANNEL
C
      DSEED = 135791.D0
      CALL RAND (NI, NSYMB, DSEED)
      DSEED = 780956.D0
      CALL RAND (NQ, NSYMB, DSEED)
      CALL SIGNAL (XSIGNL, NI, NQ)
      CALL RCOSTX (TF)
      CALL FILTER (XSIGNL, TF, FOF)
      CALL ENERGY (XSIGNL,CC, EB)
C
C   ADJACENT CHANNEL 1
C
      DSEED = 958439.D0
      CALL RAND (NIA1, NSYMB, DSEED)
      DSEED = 543895.D0
      CALL RAND (NQA1, NSYMB, DSEED)
      CALL SIGNAL (ACI1, NIA1, NQA1)
      CALL ATT (ACI1, AT1)
      CALL PHASE (ACI1, PHI)
      CALL OFFSET (ACI1, IOF)
      CALL ENERGY (ACI1, C1, EB1)
      CALL FILTER (ACI1, TF, FOF)
      CALL FRESHI (ACI1, FOF1)
      DO 10 I=1, LDIM
10  XSIGNL(I) = XSIGNL(I) + ACI1(I)
C
C   ADJACENT CHANNEL 2
C
      DSEED = 145871.D0
      CALL RAND (NIA2, NSYMB, DSEED)
      DSEED = 431554.D0
      CALL RAND (NQA2, NSYMB, DSEED)
      CALL SIGNAL (ACI2, NIA2, NQA2)
      CALL ATT (ACI2, AT2)
      CALL PHASE (ACI2, PHI)
      CALL OFFSET (ACI2, IOF)
      CALL ENERGY (ACI2, C2, EB2)
      CALL FILTER (ACI2, TF, FOF)
      CALL FRESHI (ACI2, FOF2)
      DO 11 I =1, LDIM
11  XSIGNL(I) = XSIGNL(I) + ACI2(I)
C
C   CALCULATION OF THE CARRIER-TO-INTERFERENCE RATIO
C
      A1 = 10.*ALOG10(CC/C1)
      A2 = 10.*ALOG10(CC/C2)
      WRITE(6,12) A1, A2
12  FORMAT(1X,'C/I1=',F10.3,3X,'C/I2=',F10.3///)
C
C   WRITE THE PHASE SHIFT AND THE OFFSET OF THE INTERFERERS
C
      WRITE(6,102) IOF, PHI
102 FORMAT(1X,'IOF=',I6,5X,'PHI=',F10.4///)
C
C   END OF THE TRANSMITTER
C
      DO 19 I=1, LDIM
19  DATA(I) = XSIGNL (I)

```

```

10A00730
10A00740
10A00750
10A00760
10A00770
10A00780
10A00790
10A00800
10A00810
10A00820
10A00830
10A00840
10A00850
10A00860
10A00870
10A00880
10A00890
10A00900
10A00910
10A00920
10A00930
10A00940
10A00950
10A00960
10A00970
10A00980
10A00990
10A01000
10A01010
10A01020
10A01030
10A01040
10A01050
10A01060
10A01070
10A01080
10A01090
10A01100
10A01110
10A01120
10A01130
10A01140
10A01150
10A01160
10A01170
10A01180
10A01190
10A01200
10A01210
10A01220
10A01230
10A01240
10A01250
10A01260
10A01270
10A01280
10A01290
10A01300
10A01310
10A01320
10A01330
10A01340
10A01350
10A01360
10A01370
10A01380
10A01390
10A01400
10A01410
10A01420
10A01430
10A01440

```

```

      CALL RCOSRX(TF)
      CALL MHGG(TF,PNOISE)
      CALL FILTER(DATA,TF, FOF)
C
      GO TO (26,27), IND
      27 CALL SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      CALL DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PEI,EB)
CCC  IF((NERROR.NE.0).AND.(NR.EQ.NRUNS)) CALL PLOT(0.0,0.0,999)
      IF(NERROR.NE.0) GO TO 100
      DO 99 I=1,NSNR
      XARRAY(I)=EBNO(I)
      99 YARRAY(I)=PEI(I)
      A=1.
      CALL DRAWCN(XARRAY,YARRAY,NR,NRUNS,A)
C      CALL DRAWEB(XARRAY,YARRAY,JCURV,JLAST)
      GO TO 100
      26 CALL EYEQ(DATA)
      100 CONTINUE
      FOF1 = FOF1 - DF
      FOF2 = FOF2 + DF
      101 CONTINUE
      STOP
      END
C*****
C      SUBROUTINE:  SIGNAL
C
C      THIS SUBROUTINE FORMS A COMPLEX BASEBAND SIGNAL
C*****
      SUBROUTINE SIGNAL(DATA,NI,NQ)
      COMPLEX DATA(1)
      DIMENSION NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
      DO 2 I=1,NSYMB
C      LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL
      J1=(I-1)*LSAMPL+1
      J2=I*LSAMPL
      DO 3 J3=J1,J2
      3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))
      2 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE:  RCOSTX
C
C      THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C      RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A
C      X/SINX APERTURE EQUALIZER
C*****
      SUBROUTINE RCOSTX(TF)
      COMPLEX TF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SBANDW=BAUD*LSAMPL
      NO=LDIM/2
      NO1=NO+1
      IF (ALPHA.EQ.0) ALPHA=0.0001
      FN=LDIM*(FBW/SBANDW)
      F1=(1.-ALPHA)*FN
      F2=(1.+ALPHA)*FN
      IFN=IFIX(FN)
      IF1=IFIX(F1)+1
      IF2=IFIX(F2)+1
C
C      THE AMPLITUDE CHARACTERISTICS
C
      A1=3.1415926/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
      TF(I)=CMPLX(1.0,0.0)
      J=I-1
C
C      X/SIN(X) EQUALIZATION

```

```

10A01450
10A01460
10A01470
10A01480
10A01490
10A01500
10A01510
10A01520
10A01530
10A01540
10A01550
10A01560
10A01570
10A01580
10A01590
10A01600
10A01610
10A01620
10A01630
10A01640
10A01650
10A01660
10A01670
10A01680
10A01690
10A01700
10A01710
10A01720
10A01730
10A01740
10A01750
10A01760
10A01770
10A01780
10A01790
10A01800
10A01810
10A01820
10A01830
10A01840
10A01850
10A01860
10A01870
10A01880
10A01890
10A01900
10A01910
10A01920
10A01930
10A01940
10A01950
10A01960
10A01970
10A01980
10A01990
10A02000
10A02010
10A02020
10A02030
10A02040
10A02050
10A02060
10A02070
10A02080
10A02090
10A02100
10A02110
10A02120
10A02130
10A02140
10A02150
10A02160

```

```

C      A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))
      TF(I)=CMPLX(A2,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
        I=J-1
C
C      ROOT OF RAISED COSINE
C
      A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))
      A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
      TF(J)=TF(J)*CMPLX(A3,0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
        TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
        TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: RCOSRX
C
C      THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C      RAISED COSINE FILTER WITH ARBITRARY ALPHA
C*****
      SUBROUTINE RCOSRX(TF)
      COMPLEX TF( 1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SBANDW=BAUD*LSAMPL
      NO=LDIM/2
      NO1=NO+1
      IF (ALPHA.EQ.0) ALPHA=0.0001
      FN=LDIM*(FBW/SBANDW)
      F1=(1.-ALPHA)*FN
      F2=(1.+ALPHA)*FN
      IFN=IFIX(FN)
      IF1=IFIX(F1)+1
      IF2=IFIX(F2)+1
C
C      AMPLITUDE CHARACTERISTIC
C
      A1=3.141592/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
        TF(I)=CMPLX(1.0,0.0)
        J=I-1
        TF(I)=CMPLX(1.,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
        I=J-1
C
C      ROOT OF RAISED COSINE
C
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
        TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
        TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN

```

10A02170
 10A02180
 10A02190
 10A02200
 10A02210
 10A02220
 10A02230
 10A02240
 10A02250
 10A02260
 10A02270
 10A02280
 10A02290
 10A02300
 10A02310
 10A02320
 10A02330
 10A02340
 10A02350
 10A02360
 10A02370
 10A02380
 10A02390
 10A02400
 10A02410
 10A02420
 10A02430
 10A02440
 10A02450
 10A02460
 10A02470
 10A02480
 10A02490
 10A02500
 10A02510
 10A02520
 10A02530
 10A02540
 10A02550
 10A02560
 10A02570
 10A02580
 10A02590
 10A02600
 10A02610
 10A02620
 10A02630
 10A02640
 10A02650
 10A02660
 10A02670
 10A02680
 10A02690
 10A02700
 10A02710
 10A02720
 10A02730
 10A02740
 10A02750
 10A02760
 10A02770
 10A02780
 10A02790
 10A02800
 10A02810
 10A02820
 10A02830
 10A02840
 10A02850
 10A02860
 10A02870
 10A02880

```

      END
C*****
C      SUBROUTINE: FILTER
C
C      THIS SUBROUTINE PERFORMS THE FILTERING
C      PROCESS ON A DATA SEQUENCE
C*****
      SUBROUTINE FILTER(DATA,TF,FFFF)
      COMPLEX DATA(1),TF(1)
      DIMENSION IWK(20)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      CALL FFT2C(DATA,19,IWK)
      IFOF=IFIX(FFFF*FLOAT(LDIM)/SBANDW)
      CALL OFFSET(DATA,IFOF)
      DO 1 I=1,LDIM
1      DATA(I)=CONJG(DATA(I)*TF(I))
      CALL FFT2C(DATA,19,IWK)
      DO 2 I=1,LDIM
2      DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END
C*****
C      SUBROUTINE: HHGG
C
C      THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER
C      AT THE OUTPUT OF THE RECEIVE FILTER
C*****
      SUBROUTINE HHGG(TF,PNOISE)
      COMPLEX TF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SUM=0.0
      SBANDW=BAUD*LSAMPL
      DO 1 L=1,LDIM
      HH=(CABS(TF(L)))**2
1      SUM=SUM+HH
      PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.
      WRITE (6,2) PNOISE
2      FORMAT(5X,'PNOISE=',F7.3,/)
      RETURN
      END
C*****
C      SUBROUTINE: ENERGY
C
C      THIS SUBROUTINE COMPUTES THE CARRIER POWER
C      AND THE AVERAGE BIT ENERGY
C*****
      SUBROUTINE ENERGY(DATA,C,EB)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      COMPLEX DATA(1)
      WATTS=0.
      DO 1 I=1,LDIM
      WATTS=WATTS+CABS(DATA(I))**2
1      CONTINUE
      C=WATTS/FLOAT(LDIM)/2.
      RRK=RK/2.
      EB=C/(RRK*BAUD)
      WRITE (6,40) C,EB
40      FORMAT(10X,'C=',F15.6,5X,'EB=',E15.8)
      RETURN
      END
C*****
C      SUBROUTINE: SYNCRO
C
C      THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA
C*****
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      INTEGER QMAXF
      COMPLEX DATA(1),AMP
      DIMENSION NI(1),NQ(1)

```

```

10A02890
10A02900
* 10A02910
* 10A02920
* 10A02930
* 10A02940
10A02950
10A02960
10A02970
10A02980
10A02990
10A03000
10A03010
10A03020
10A03030
10A03040
10A03050
10A03060
10A03070
10A03080
10A03090
10A03100
* 10A03110
* 10A03120
* 10A03130
* 10A03140
10A03150
10A03160
10A03170
10A03180
10A03190
10A03200
10A03210
10A03220
10A03230
10A03240
10A03250
10A03260
10A03270
10A03280
10A03290
10A03300
* 10A03310
* 10A03320
* 10A03330
* 10A03340
10A03350
10A03360
10A03370
10A03380
10A03390
10A03400
10A03410
10A03420
10A03430
10A03440
10A03450
10A03460
10A03470
10A03480
10A03490
10A03500
10A03510
10A03520
* 10A03530
* 10A03540
* 10A03550
10A03560
10A03570
10A03580
10A03590
10A03600

```

```

COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
COMMON IMAX,QMAX,EBNO1,RK,CN1
NERROR=0
IF (IOFF.EQ.0) GO TO 111
DO 6 K=1,IOFF
XX=AIMAG(DATA(LDIM))
DO 7 KK=2,LDIM
7 DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))
8 DATA(1)=CMPLX(REAL(DATA(1)),XX)
C
C SYNCHRONIZE THE RECEIVED DATA
C
111 NOLD=0
NOF=0
K=1
300 CONTINUE
NEW=0
DO 200 J=1,NSYMB
THR1I=FLOAT(NI(J))-1.
THR2I=FLOAT(NI(J))+1.
THR1Q=FLOAT(NQ(J))-1.
THR2Q=FLOAT(NQ(J))+1.
J1=K+(J-1)*LSAMPL
IF(J1.GT.LDIM) J1=J1-LDIM
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
IF((AXBAR.GT.THR1I).AND.(AXBAR.LT.THR2I)) NEW=NEW+1
IF((AYBAR.GT.THR1Q).AND.(AYBAR.LT.THR2Q)) NEW=NEW+1
200 CONTINUE
IF(NOLD.GE.NEW) GO TO 399
NOLD=NEW
NOF=K
399 K=K+1
LFU=2*NSYMB
C
C SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP
C
C
NSFT=LSAMPL*6
IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300
NOF=NOF-1
KT=K-2
WRITE(6,56) NOLD,NOF,KT
56 FORMAT(1X,'NO OF MAX. CORRECT SYMBOLS = ',I4,/,1X,
* 'NO OF SHIFT IN SAMPLES FOR MAX. CORRECT SYMBOLS= ',I4,/,
* 1X,'TOTAL NO OF SHIFT FOR SEARCHING = ',I4)
C
IF (NOF.EQ.0) GO TO 230
LO=LDIM-1
DO 250 I=1,NOF
AMP=DATA(1)
DO 240 J=1,LO
DATA(J)=DATA(J+1)
240 CONTINUE
DATA(LDIM)=AMP
250 CONTINUE
C
C OPTIMIZE THE SAMPLING INSTANT
C
230 MI=1
MQ=1
EOI=FLOAT(NSYMB) + 1.
EQQ=FLOAT(NSYMB) + 1.
M=22
VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
SIGMA=SQRT(VARIAN)
MOS=LSAMPL/4
DO 80 J=1,MOS
EI=0.
EQ=0.
IERX=0

```

```

10A03610
10A03620
10A03630
10A03640
10A03650
10A03660
10A03670
10A03680
10A03690
10A03700
10A03710
10A03720
10A03730
10A03740
10A03750
10A03760
10A03770
10A03780
10A03790
10A03800
10A03810
10A03820
10A03830
10A03840
10A03850
10A03860
10A03870
10A03880
10A03890
10A03900
10A03910
10A03920
10A03930
10A03940
10A03950
10A03960
10A03970
10A03980
10A03990
10A04000
10A04010
10A04020
10A04030
10A04040
10A04050
10A04060
10A04070
10A04080
10A04090
10A04100
10A04110
10A04120
10A04130
10A04140
10A04150
10A04160
10A04170
10A04180
10A04190
10A04200
10A04210
10A04220
10A04230
10A04240
10A04250
10A04260
10A04270
10A04280
10A04290
10A04300
10A04310
10A04320

```

```

      IERY=0
      DO 70 K=1,NSYMB
C      VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
      J1=(K-1)*LSAMPL+J
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      INDEXI=0
      AMPX=ABS(AXBAR)
      AMPI=ABS(FLOAT(NI(K)))
      THR1I=AMPI - 1.0
      THR2I=AMPI + 1.0
      IMAXF=0
      IF (AMPI.EQ.IMAX) GO TO 30
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30  IF(AMPX.LE.THR1I) INDEXI=1
      IMAXF=1
40  CONTINUE
      INDEXQ=0
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      QMAXF=0
      IF (AMPQ.EQ.QMAX) GO TO 50
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50  IF(AMPY.LE.THR1Q) INDEXQ=1
      QMAXF=1
60  CONTINUE
      IF(INDEXI.EQ.1) EI=EI + 1.
      IF(INDEXQ.EQ.1) EQ=EQ + 1.
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB
C
      D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C      CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
      IF (IMAXF .EQ. 1) GO TO 85
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
65  D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IF (QMAXF .EQ. 1) GO TO 70
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IERX=IERX+INDEXI
      IERY=IERY+INDEXQ
70  CONTINUE
      WRITE(6,81) IERX,IERY
81  FORMAT(5X,'IERX=',I5,5X,'IERY=',I5)
      IF(EOI.LE.EI) GO TO 75
      EOI=EI
      MI=J
75  CONTINUE
      IF (EQQ.LE.EQ) GO TO 80
      EQQ=EQ
      MQ=J
80  MOFF=IABS(MI-MQ)
      WRITE(6,91) MI,MQ
91  FORMAT(5X,'MI=',I3,5X,'MQ=',I3)
      IF(MOFF.NE.0) WRITE(6,90) MOFF
      OFF=(FLOAT(MI) + FLOAT(MQ))/FLOAT(LSAMPL)

```

```

10A04330
10A04340
10A04350
10A04360
10A04370
10A04380
10A04390
10A04400
10A04410
10A04420
10A04430
10A04440
10A04450
10A04460
10A04470
10A04480
10A04490
10A04500
10A04510
10A04520
10A04530
10A04540
10A04550
10A04560
10A04570
10A04580
10A04590
10A04600
10A04610
10A04620
10A04630
10A04640
10A04650
10A04660
10A04670
10A04680
10A04690
10A04700
10A04710
10A04720
10A04730
10A04740
10A04750
10A04760
10A04770
10A04780
10A04790
10A04800
10A04810
10A04820
10A04830
10A04840
10A04850
10A04860
10A04870
10A04880
10A04890
10A04900
10A04910
10A04920
10A04930
10A04940
10A04950
10A04960
10A04970
10A04980
10A04990
10A05000
10A05010
10A05020
10A05030
10A05040

```

```

90 FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
#12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
WRITE(6,95) OFF
95 FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
RETURN
END
C*****
C SUBROUTINE: DECODE *
C *
C THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA *
C*****
SUBROUTINE DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)
INTEGER QMAXF
COMPLEX DATA(1),AMP
DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
NERROR=0
DO 1 I=1,NSNR
1 PE(I)=0.
COMMON IMAX,QMAX,EBNO1,RK,CN1
C CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE
C ERRORS IN ADJACENT LEVELS ARE CONSIDERED
3 DO 2 K=1,NSYMB
J1=(K-1)*LSAMPL+MI
J2=(K-1)*LSAMPL + MQ
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.
INDEXI=0
AMPX=ABS(AXBAR)
AMPI=ABS(FLOAT(NI(K)))
THR1I=AMPI - 1.0
THR2I=AMPI + 1.0
IMAXF=0
IF (AMPI.EQ.IMAX) GO TO 30
IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
GO TO 40
30 IF(AMPX.LE.THR1I) INDEXI=1
IMAXF=1
40 CONTINUE
INDEXQ=0
AMPY=ABS(AYBAR)
AMPQ=ABS(FLOAT(NQ(K)))
THR1Q=AMPQ - 1.0
THR2Q=AMPQ + 1.0
QMAXF=0
IF (AMPQ.EQ.QMAX) GO TO 50
IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
GO TO 60
50 IF(AMPY.LE.THR1Q) INDEXQ=1
QMAXF=1
60 CONTINUE
IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1
C
C COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.
C THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS
C 1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE.
C FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2)))
C AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +
C 1/2*ERFC(D2/(SIGMA*SQRT(2))).
C
DO 4 M=1,NSNR
EBNO(M)=EBNO1+1.0*FLOAT(M-1)
VARIAN=PNOISE*EB*(10.**(-0.1*EBNO(M)))
SIGMA=SQRT(VARIAN)
D1=ABS(AMPX-THR1I)
ARG=D1/(SIGMA*SQRT(2.))
C CHECK IF ARG IS LARGE IN WHICH CASE PE IS INSIGNIFICANT
C CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN
C
NERI=M
IF (INDEXI.EQ.1) GO TO 153

```

10A05050
10A05060
10A05070
10A05080
10A05090
10A05100
10A05110
10A05120
10A05130
10A05140
10A05150
10A05160
10A05170
10A05180
10A05190
10A05200
10A05210
10A05220
10A05230
10A05240
10A05250
10A05260
10A05270
10A05280
10A05290
10A05300
10A05310
10A05320
10A05330
10A05340
10A05350
10A05360
10A05370
10A05380
10A05390
10A05400
10A05410
10A05420
10A05430
10A05440
10A05450
10A05460
10A05470
10A05480
10A05490
10A05500
10A05510
10A05520
10A05530
10A05540
10A05550
10A05560
10A05570
10A05580
10A05590
10A05600
10A05610
10A05620
10A05630
10A05640
10A05650
10A05660
10A05670
10A05680
10A05690
10A05700
10A05710
10A05720
10A05730
10A05740
10A05750
10A05760


```

      IF (ARG.GT.12.) ARG=12.
      PEI=ERFC(ARG)/2.
      IF (IMAXF.EQ.1) GO TO 100
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEI=PEI + ERFC(ARG)/2.
100  D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (INDEXQ.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEQ=ERFC(ARG)/2.
      IF (QMAXF.EQ.1) GO TO 110
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEQ=PEQ + ERFC(ARG)/2.
110  IF (PEI.LT.1.E-15) PEI=0.
      IF (PEQ.LT.1.E-15) PEQ=0.
      4  PE(M)=PE(M)+(PEI+PEQ)/2.
      2  CONTINUE
      DO 5 I=1,NSNR
      5  PE(I)=PE(I)/FLOAT(NSYMB)
C      PRINT 150
C150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)
C      WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)
C151  FORMAT(5X,F5.1,10X,E10.2)
      WRITE (6,152) NERROR
152  FORMAT(/5X,'ERRORS=',I5,/)
      GO TO 155
153  PRINT 154
154  FORMAT(10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)
      WRITE(6,157) NERI
157  FORMAT(5X,'NERI=',I5)
155  CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: DRAWCN
C
C      THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C      P(E) AS LOW AS 1.0E-10 AND A C/N RATIO AS HIGH AS 42 DB
C*****
      SUBROUTINE DRAWCN(XARRAY,YARRAY,JCURV,JLAST,A)
      DIMENSION XARRAY(1),YARRAY(1),XA(150),XX(150),YY(150)
      COMMON /NUMB/ NSNR,NSYMB,BAUD
      COMMON IMAX,EBNO1,RK,CN1
      M=NSNR
      BR=RK*BAUD
      NBW=BAUD/A
      CF=BR/NBW
      DO 1 K=1, NSNR
      XA(K)=XARRAY(K) * 10*(ALOG10(CF))
      IF (YARRAY(K).LE.1.0E-10) GOTO 3
      GOTO 1
      3  M=K-1
      GOTO 4
      1  CONTINUE
      4  CONTINUE
      I=M+1
      II=I+1
      XA(I)=9.0
      XA(II)=1.0
      YARRAY(I)=1.E-10
      YARRAY(II)=0.4
      GO TO 2
      IF (JCVRV.GT.1) GOTO 2
C
C      ESTABLISH THE SURFACE AREA.
C
      CALL PLOTS(30.0,37.5)
      CALL FACTOR(0.85)

```

10A05770
 10A05780
 10A05790
 10A05800
 10A05810
 10A05820
 10A05830
 10A05840
 10A05850
 10A05860
 10A05870
 10A05880
 10A05890
 10A05900
 10A05910
 10A05920
 10A05930
 10A05940
 10A05950
 10A05960
 10A05970
 10A05980
 10A05990
 10A06000
 10A06010
 10A06020
 10A06030
 10A06040
 10A06050
 10A06060
 10A06070
 10A06080
 10A06090
 10A06100
 10A06110
 10A06120
 10A06130
 10A06140
 10A06150
 10A06160
 10A06170
 10A06180
 10A06190
 10A06200
 10A06210
 10A06220
 10A06230
 10A06240
 10A06250
 10A06260
 10A06270
 10A06280
 10A06290
 10A06300
 10A06310
 10A06320
 10A06330
 10A06340
 10A06350
 10A06360
 10A06370
 10A06380
 10A06390
 10A06400
 10A06410
 10A06420
 10A06430
 10A06440
 10A06450
 10A06460
 10A06470
 10A06480

```

C      ESTABLISH THE ORIGIN.
C      CALL PLOT(3.0,1.5,-3)
C      DRAW THE LOGARITHMIC Y-AXIS.
C      CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,25.,
+90.,1.0E-10,.4)
C      DRAW THE LINEAR X-AXIS.
C      CALL AXIS(0.0,0.0,9HC/N IN dB,-9,
+32.0,0.0,9.0,1.0)
C      PLOT THE DATA IN LOG-LINEAR MODE.
C
2 CONTINUE
  PRINT 150
150 FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',7X,'C/N',/)
  WRITE(6,151) (XARRAY(I),YARRAY(I),XA(I),I=1,M)
151 FORMAT(5X,F5.0,10X,E10.2,10X,F5.1)
  GO TO 45
  CALL LGLIN(XA,YARRAY,M,1,4,JCURV,1)
  IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
45 RETURN
END
C*****
C      SUBROUTINE: EYEQ
C
C      THIS SUBROUTINE DRAWS THE EYE DIAGRAM
C      FOR A TWO SYMBOL DURATION
C*****
C      SUBROUTINE EYEQ(DATA)
C      DIMENSION DATA2(66),DATA3(66)
C      COMPLEX DATA(1)
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD
C      ESTABLISH THE SURFACE AREA.
C      CALL PLOTS(37.0,27.5)
C      ESTABLISH THE ORIGIN.
C      CALL PLOT(3.0,13.0,-3)
C      DRAW THE TIME AXIS.
C      CALL AXIS(0.0,0.0,1H,-1,24.,0.0,1.0,2.8)
C      DRAW THE AMPLITUDE AXIS.
C      CALL AXIS(0.0,-8.00,1H,1,16.0,90.0,-24.,3.0)
C      PLOT THE DATA.
C      JJ=2*LSAMPL
C      DATA2(JJ+1)=0.0
C      DATA2(JJ+2)=3.0
C      DATA3(JJ+1)=1.0
C      DATA3(JJ+2)=2.8
C      M2=NSYMB/8
C      M3=3*M2+1
C      M4=4*M2-1
C      DO 4 KK=1,M2
C      DO 2 I=1,JJ
C      II=(KK-1)*LSAMPL+I
C      IF(II.GT.LDIM) II=II-(II/LDIM)*LDIM
C      DATA2(I)=REAL(DATA(II))
C      DATA3(I)=FLOAT(I)
C      CALL LINE(DATA3,DATA2,JJ,1,0,0)
2 CONTINUE
  CALL PLOT(0.0,0.0,999)
  RETURN
END
C*****
C      SUBROUTINE: ATT
C
C      THIS SUBROUTINE PERFORMS THE ATTENUATION OF A COMPLEX
C      SIGNAL (AT IS THE ATTENUATION FACTOR IN DB)
C*****

```

```

10A06490
10A06500
10A06510
10A06520
10A06530
10A06540
10A06550
10A06560
10A06570
10A06580
10A06590
10A06600
10A06610
10A06620
10A06630
10A06640
10A06650
10A06660
10A06670
10A06680
10A06690
10A06700
10A06710
10A06720
10A06730
10A06740
10A06750
10A06760
10A06770
10A06780
10A06790
10A06800
10A06810
10A06820
10A06830
10A06840
10A06850
10A06860
10A06870
10A06880
10A06890
10A06900
10A06910
10A06920
10A06930
10A06940
10A06950
10A06960
10A06970
10A06980
10A06990
10A07000
10A07010
10A07020
10A07030
10A07040
10A07050
10A07060
10A07070
10A07080
10A07090
10A07100
10A07110
10A07120
10A07130
10A07140
10A07150
10A07160
10A07170
10A07180
10A07190
10A07200

```

```

SUBROUTINE ATT(DATA,AT)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
COMPLEX DATA(1)
ATTT=10.**(-AT/20.)
DO 10 I=1,LDIM
10 DATA(I)=DATA(I)*ATTT
RETURN
END
C*****
C SUBROUTINE: OFFSET *
C *
C THIS SUBROUTINE SHIFTS THE DATA SEQUENCE BY IOF *
C*****
SUBROUTINE OFFSET(DATA,IOF)
COMPLEX DATA(1),BUFF(262144)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
J=LDIM-IABS(IOF)
K=J+1
IF (IOF.EQ.0) GOTO 9
IF (IOF.LT.0) GOTO 4
DO 1 I=1,J
1 BUFF(I)=DATA(I)
DO 2 I=K,LDIM
2 DATA(I-J)=DATA(I)
DO 3 I=1,J
3 DATA(I+IOF)=BUFF(I)
GOTO 9
4 IOF=-IOF
L=IOF+1
DO 5 I=L,LDIM
5 BUFF(I-IOF)=DATA(I)
DO 6 I=1,IOF
6 DATA(I+J)=DATA(I)
DO 7 I=1,J
7 DATA(I)=BUFF(I)
9 RETURN
END
C*****
C SUBROUTINE: PHASE *
C *
C THIS SUBROUTINE PERFORMS THE PHASE SHIFT *
C OF A COMPLEX DATA SEQUENCE *
C*****
SUBROUTINE PHASE(DATA,PHI)
COMPLEX EPS,DATA(1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
EPS=CMPLX(COS(PHI),-SIN(PHI))
DO 10 I=1,LDIM
10 DATA(I)=DATA(I)*EPS
RETURN
END
C*****
C SUBROUTINE: RAND *
C *
C THIS SUBROUTINE GENERATES RANDOM SYMBOLS *
C*****
SUBROUTINE RAND(IR,N,DSEED)
INTEGER IR(1)
DOUBLE PRECISION DSEED
DO 3000 I=1,N
R=GGUBFS(DSEED)
IR(I)=IFIX(R)
IF (R.LE.0.031) IR(I)=-31
IF (R.GT.0.031) .AND. (R.LE.0.062) IR(I)=-29
IF (R.GT.0.062) .AND. (R.LE.0.093) IR(I)=-27
IF (R.GT.0.093) .AND. (R.LE.0.124) IR(I)=-25
IF (R.GT.0.124) .AND. (R.LE.0.155) IR(I)=-23
IF (R.GT.0.155) .AND. (R.LE.0.186) IR(I)=-21
IF (R.GT.0.186) .AND. (R.LE.0.217) IR(I)=-19
IF (R.GT.0.217) .AND. (R.LE.0.248) IR(I)=-17
IF (R.GT.0.248) .AND. (R.LE.0.279) IR(I)=-15

```

10A07210
10A07220
10A07230
10A07240
10A07250
10A07260
10A07270
10A07280
10A07290
10A07300
10A07310
10A07320
10A07330
10A07340
10A07350
10A07360
10A07370
10A07380
10A07390
10A07400
10A07410
10A07420
10A07430
10A07440
10A07450
10A07460
10A07470
10A07480
10A07490
10A07500
10A07510
10A07520
10A07530
10A07540
10A07550
10A07560
10A07570
10A07580
10A07590
10A07600
10A07610
10A07620
10A07630
10A07640
10A07650
10A07660
10A07670
10A07680
10A07690
10A07700
10A07710
10A07720
10A07730
10A07740
10A07750
10A07760
10A07770
10A07780
10A07790
10A07800
10A07810
10A07820
10A07830
10A07840
10A07850
10A07860
10A07870
10A07880
10A07890
10A07900
10A07910
10A07920

```
IF((R.GT.0.279).AND.(R.LE.0.31)) IR(I)=13      10A07930
IF((R.GT.0.31).AND.(R.LE.0.341)) IR(I)=11      10A07940
IF((R.GT.0.341).AND.(R.LE.0.372)) IR(I)=9       10A07950
IF((R.GT.0.372).AND.(R.LE.0.403)) IR(I)=7       10A07960
IF((R.GT.0.403).AND.(R.LE.0.434)) IR(I)=5       10A07970
IF((R.GT.0.434).AND.(R.LE.0.465)) IR(I)=3       10A07980
IF((R.GT.0.465).AND.(R.LE.0.496)) IR(I)=1       10A07990
IF((R.GT.0.496).AND.(R.LE.0.527)) IR(I)=1       10A08000
IF((R.GT.0.527).AND.(R.LE.0.558)) IR(I)=3       10A08010
IF((R.GT.0.558).AND.(R.LE.0.589)) IR(I)=5       10A08020
IF((R.GT.0.589).AND.(R.LE.0.62)) IR(I)=7        10A08030
IF((R.GT.0.62).AND.(R.LE.0.651)) IR(I)=9        10A08040
IF((R.GT.0.651).AND.(R.LE.0.682)) IR(I)=11      10A08050
IF((R.GT.0.682).AND.(R.LE.0.713)) IR(I)=13      10A08060
IF((R.GT.0.713).AND.(R.LE.0.744)) IR(I)=15      10A08070
IF((R.GT.0.744).AND.(R.LE.0.775)) IR(I)=17      10A08080
IF((R.GT.0.775).AND.(R.LE.0.806)) IR(I)=19      10A08090
IF((R.GT.0.806).AND.(R.LE.0.837)) IR(I)=21      10A08100
IF((R.GT.0.837).AND.(R.LE.0.868)) IR(I)=23      10A08110
IF((R.GT.0.868).AND.(R.LE.0.899)) IR(I)=25      10A08120
IF((R.GT.0.899).AND.(R.LE.0.930)) IR(I)=27      10A08130
IF((R.GT.0.930).AND.(R.LE.0.961)) IR(I)=29      10A08140
IF((R.GT.0.961).AND.(R.LE.0.996)) IR(I)=31      10A08150
IF(R.GT.0.996) IR(I)=31                        10A08160
3000 CONTINUE                                10A08170
      RETURN                                10A08180
      END                                10A08190
C*****                                10A08200
C  SUBROUTINE: FRESHI                      * 10A08210
C                                           * 10A08220
C  THIS SUBROUTINE SHIFTS A COMPLEX SIGNAL * 10A08230
C  IN THE FREQUENCY DOMAIN                * 10A08240
C*****                                10A08250
      SUBROUTINE FRESHI(DATA,FFFF)          10A08260
      COMPLEX DATA(1)                     10A08270
      DIMENSION IWK(20)                    10A08280
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW 10A08290
      CALL FFT2C(DATA,19,IWK)              10A08300
      IFOF=IFIX(FFFF*FLOAT(LDIM)/SBANDW)   10A08310
      CALL OFFSET(DATA,IFOF)               10A08320
      DO 1 I=1,LDIM                        10A08330
1 DATA(I)=CONJG(DATA(I))                  10A08340
      CALL FFT2C(DATA,19,IWK)              10A08350
      DO 2 I=1,LDIM                        10A08360
2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)      10A08370
      RETURN                                10A08380
      END                                10A08390
```

```

C***** 51A00010
C*      * 51A00020
C*      FILE NAME: 51ACI      * 51A00030
C*      * 51A00040
C*      * 51A00050
C*      * 51A00060
C*      THIS PROGRAM EVALUATES THE PERFORMANCE OF 512-QAM      * 51A00070
C*      SYSTEMS IN THE PRESENCE OF TWO ADJACENT CHANNEL      * 51A00080
C*      INTERFERERS OF THE SAME MODULATION FORMAT (512-QAM)      * 51A00090
C*      * 51A00100
C***** 51A00110
C      COMPLEX DATA(524288),CCI(524288),TF(524288)      51A00120
C      COMPLEX XSIGNL(524288),BUFF(524288)      51A00130
C      COMPLEX ACI1(524288), ACI2(524288)      51A00140
C      DIMENSION NIA1(8192), NQA1(8192), NIA2(8192), NQA2(8192)      51A00150
C      DIMENSION PEI(50),EBNO(50),NI(8192),NQ(8192),NNI(8192),NNQ(8192)      51A00160
C      DIMENSION XARRAY(150),YARRAY(150),NIC(8192),NGC(8192)      51A00170
C      INTEGER IR(8192)      51A00180
C      REAL IBAUD,IMAX,QMAX      51A00190
C      51A00200
C      INITIALIZE PROGRAM.      51A00210
C      51A00220
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW      51A00230
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2      51A00240
C      COMMON IMAX,QMAX,EBNO1,RK,CN1      51A00250
C      51A00260
C      FOR 512-QAM, NBPSYM= 9, EBNO1=18., CN1=24., CDIM=40.0      51A00270
C      51A00280
C      NBPSYM= 9      51A00290
C      EBNO1=18.      51A00300
C      CN1=24.      51A00310
C      CDIM=40.0      51A00320
C      ATM=45.0      51A00330
C      NLEVEL=24      51A00340
C      IMAX=FLOAT(NLEVEL)-1.      51A00350
C      QMAX=IMAX      51A00360
C      RK=FLOAT(NBPSYM)      51A00370
C      KQAM=512      51A00380
C      51A00390
C      ALPHA=0.5      51A00400
C      51A00410
C      IND=1 FOR PLOTTING THE EYE-DIAGRAM      51A00420
C      IND=2 FOR PLOTTING THE P(E)      51A00430
C      51A00440
C      IND=2      51A00450
C      NRUNS=8      51A00460
C      IOFF=0      51A00470
C      LSAMPL=64      51A00480
C      NSNR=35      51A00490
C      NSYMB=8192      51A00500
C      LDIM=LSAMPL*NSYMB      51A00510
C      51A00520
C      THE BAUD RATE BAUD IS GIVEN IN KBAUD/S      51A00530
C      51A00540
C      BAUD=100.0      51A00550
C      SBANDW=BAUD*LSAMPL      51A00560
C      NO1=LDIM/2+1      51A00570
C      NO2=NO1+1      51A00580
C      FBW=BAUD/2.      51A00590
C      PI=3.14159      51A00600
C      51A00610
C      INITIALIZE THE DATA FOR THE TWO INTERFERERS      51A00620
C      51A00630
C      DATA AT1, AT2 /20.0, 20.0/      51A00640
C      FOF = 0.      51A00650
C      FOF1 = 122.0      51A00660
C      FOF2 =-122.0      51A00670
C      DF = 2.      51A00680
C      51A00690
C      START THE COMPUTATIONS      51A00700
C      51A00710
C      DO 101 IKKK =1, 2      51A00720

```

```

      WRITE(6,83) FOF, FOF1, FOF2
83  FORMAT(1X,'FOF=',E11.4,3X,'FOF1=',E11.4,3X,'FOF2=',E11.4/)
      WRITE(6,84) BAUD
84  FORMAT(1X,'BAUDRATE=',E14.5///)
      PHI= 0.17*PI
      IOF= 32
      WRITE(6,14) PHI, IOF
14  FORMAT(1X,'PHI=',F10.4,2X,'RADS',5X,'IOF=',I8///)
82  WRITE(6,245) KQAM
245 FORMAT(5X,I5,'-QAM',/)
      WRITE(6,45) ALPHA
45  FORMAT(////,2X,'*****ALPHA=',F6.2)
C
C      MAIN CHANNEL
C
      DSEED = 135791.D0
      CALL RAND (NI, NSYMB, DSEED)
      DSEED = 780956.D0
      CALL RAND (NQ, NSYMB, DSEED)
      CALL SIGNAL (XSIGNL, NI, NQ)
      CALL RCOSTX (TF)
      CALL FILTER (XSIGNL, TF, FOF)
      CALL ENERGY (XSIGNL, CC, EB)
C
C      ADJACENT CHANNEL 1
C
      DSEED = 958439.D0
      CALL RAND (NIA1, NSYMB, DSEED)
      DSEED = 543895.D0
      CALL RAND (NQA1, NSYMB, DSEED)
      CALL SIGNAL (ACI1, NIA1, NQA1)
      CALL ATT (ACI1, AT1)
      CALL PHASE (ACI1, PHI)
      CALL OFFSET (ACI1, IOF)
      CALL ENERGY (ACI1, C1, EB1)
      CALL FILTER (ACI1, TF, FOF)
      CALL FRESHI (ACI1, FOF1)
      DO 10 I=1, LDIM
10  XSIGNL(I) = XSIGNL(I) + ACI1(I)
C
C      ADJACENT CHANNEL 2
C
      DSEED = 145871.D0
      CALL RAND (NIA2, NSYMB, DSEED)
      DSEED = 431554.D0
      CALL RAND (NQA2, NSYMB, DSEED)
      CALL SIGNAL (ACI2, NIA2, NQA2)
      CALL ATT (ACI2, AT2)
      CALL PHASE (ACI2, PHI)
      CALL OFFSET (ACI2, IOF)
      CALL ENERGY (ACI2, C2, EB2)
      CALL FILTER (ACI2, TF, FOF)
      CALL FRESHI (ACI2, FOF2)
      DO 11 I=1, LDIM
11  XSIGNL(I) = XSIGNL(I) + ACI2(I)
C
C      CALCULATION OF THE CARRIER-TO-INTERFERENCE RATIO
C
      A1 = 10.*ALOG10(CC/C1)
      A2 = 10.*ALOG10(CC/C2)
      WRITE(6,12) A1, A2
12  FORMAT(1X,'C/I1=',F10.3,3X,'C/I2=',F10.3///)
C
C      WRITE THE PHASE SHIFT AND THE OFFSET OF THE INTERFERERS
C
      WRITE(6,102) IOF, PHI
102 FORMAT(1X,'IOF=',I6,5X,'PHI=',F10.4///)
C
C      END OF THE TRANSMITTER
C
      DO 19 I=1, LDIM
19  DATA(I) = XSIGNL (I)

```

51A00730
 51A00740
 51A00750
 51A00760
 51A00770
 51A00780
 51A00790
 51A00800
 51A00810
 51A00820
 51A00830
 51A00840
 51A00850
 51A00860
 51A00870
 51A00880
 51A00890
 51A00900
 51A00910
 51A00920
 51A00930
 51A00940
 51A00950
 51A00960
 51A00970
 51A00980
 51A00990
 51A01000
 51A01010
 51A01020
 51A01030
 51A01040
 51A01050
 51A01060
 51A01070
 51A01080
 51A01090
 51A01100
 51A01110
 51A01120
 51A01130
 51A01140
 51A01150
 51A01160
 51A01170
 51A01180
 51A01190
 51A01200
 51A01210
 51A01220
 51A01230
 51A01240
 51A01250
 51A01260
 51A01270
 51A01280
 51A01290
 51A01300
 51A01310
 51A01320
 51A01330
 51A01340
 51A01350
 51A01360
 51A01370
 51A01380
 51A01390
 51A01400
 51A01410
 51A01420
 51A01430
 51A01440

```

      CALL RCOSRX(TF)
      CALL HHGG(TF,PNOISE)
      CALL FILTER(DATA,TF, FOF)
C
      GO TO (26,27), IND
27 CALL SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      CALL DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PEI,EB)
CCC IF((NERROR.NE.0).AND.(NR.EQ.NRUNS)) CALL PLOT(0.0,0.0,999)
      IF(NERROR.NE.0) GO TO 100
      DO 99 I=1,NSNR
      XARRAY(I)=EBNO(I)
99 YARRAY(I)=PEI(I)
      A=1.
      CALL DRAWCH(XARRAY,YARRAY,NR,NRUNS,A)
C      CALL DRAWEB(XARRAY,YARRAY,JCURV,JLAST)
      GO TO 100
26 CALL EYEQ(DATA)
100 CONTINUE
      FOF1 = FOF1 - DF
      FOF2 = FOF2 + DF
101 CONTINUE
      STOP
      END
C*****
C      SUBROUTINE: SIGNAL
C
C      THIS SUBROUTINE FORMS A COMPLEX BASEBAND SIGNAL
C*****
      SUBROUTINE SIGNAL(DATA,NI,NQ)
      COMPLEX DATA(1)
      DIMENSION NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
      DO 2 I=1,NSYMB
      LOAD INTO SAMPLE ARRAY, 16 SAMPLES PER SYMBOL
      J1=(I-1)*LSAMPL+1
      J2=I*LSAMPL
      DO 3 J3=J1,J2
      DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))
      3 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: RCOSTX
C
C      THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C      RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A
C      X/SINX APERTURE EQUALIZER
C*****
      SUBROUTINE RCOSTX(TF)
      COMPLEX TF( 1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SBANDW=BAUD*LSAMPL
      NO=LDIM/2
      NO1=NO+1
      IF (ALPHA.EQ.0) ALPHA=0.0001
      FN=LDIM*(FBW/SBANDW)
      F1=(1.-ALPHA)*FN
      F2=(1.+ALPHA)*FN
      IFN=IFIX(FN)
      IF1=IFIX(F1)+1
      IF2=IFIX(F2)+1
C
C      THE AMPLITUDE CHARACTERISTICS
C
      A1=3.1415926/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
      TF(1)=CMPLX(1.0,0.0)
      J=I-1
C
C      X/SIN(X) EQUALIZATION

```

```

51A01450
51A01460
51A01470
51A01480
51A01490
51A01500
51A01510
51A01520
51A01530
51A01540
51A01550
51A01560
51A01570
51A01580
51A01590
51A01600
51A01610
51A01620
51A01630
51A01640
51A01650
51A01660
51A01670
51A01680
51A01690
51A01700
51A01710
51A01720
51A01730
51A01740
51A01750
51A01760
51A01770
51A01780
51A01790
51A01800
51A01810
51A01820
51A01830
51A01840
51A01850
51A01860
51A01870
51A01880
51A01890
51A01900
51A01910
51A01920
51A01930
51A01940
51A01950
51A01960
51A01970
51A01980
51A01990
51A02000
51A02010
51A02020
51A02030
51A02040
51A02050
51A02060
51A02070
51A02080
51A02090
51A02100
51A02110
51A02120
51A02130
51A02140
51A02150
51A02160

```

```

C      A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))
      TF(I)=CMPLX(A2,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
        I=J-1
C
C      ROOT OF RAISED COSINE
C
      A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))
      A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
      TF(J)=TF(J)*CMPLX(A3,0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
        TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
        TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: RCOSRX
C
C      THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C      RAISED COSINE FILTER WITH ARBITRARY ALPHA
C*****
      SUBROUTINE RCOSRX(TF)
      COMPLEX TF( 1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SBANDW=BAUD*LSAMPL
      NO=LDIM/2
      NO1=NO+1
      IF (ALPHA.EQ.0) ALPHA=0.0001
      FN=LDIM*(FBW/SBANDW)
      F1=(1.-ALPHA)*FN
      F2=(1.+ALPHA)*FN
      IFN=IFIX(FN)
      IF1=IFIX(F1)+1
      IF2=IFIX(F2)+1
C
C      AMPLITUDE CHARACTERISTIC
C
      A1=3.141592/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
        TF(I)=CMPLX(1.0,0.0)
        J=I-1
        TF(I)=CMPLX(1.,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
        I=J-1
C
C      ROOT OF RAISED COSINE
C
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
        TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
        TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN

```

51A02170
51A02180
51A02190
51A02200
51A02210
51A02220
51A02230
51A02240
51A02250
51A02260
51A02270
51A02280
51A02290
51A02300
51A02310
51A02320
51A02330
51A02340
51A02350
51A02360
51A02370
51A02380
51A02390
51A02400
51A02410
51A02420
51A02430
51A02440
51A02450
51A02460
51A02470
51A02480
51A02490
51A02500
51A02510
51A02520
51A02530
51A02540
51A02550
51A02560
51A02570
51A02580
51A02590
51A02600
51A02610
51A02620
51A02630
51A02640
51A02650
51A02660
51A02670
51A02680
51A02690
51A02700
51A02710
51A02720
51A02730
51A02740
51A02750
51A02760
51A02770
51A02780
51A02790
51A02800
51A02810
51A02820
51A02830
51A02840
51A02850
51A02860
51A02870
51A02880


```

      END
C*****
C      SUBROUTINE: FILTER
C
C      THIS SUBROUTINE PERFORMS THE FILTERING
C      PROCESS ON A DATA SEQUENCE
C*****
      SUBROUTINE FILTER(DATA,TF,FFFF)
      COMPLEX DATA(1),TF(1)
      DIMENSION IWK(20)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      CALL FFT2C(DATA,19,IWK)
      IFOF=IFIX(FFFF*FLOAT(LDIM)/SBANDW)
      CALL OFFSET(DATA,IFOF)
      DO 1 I=1,LDIM
1 DATA(I)=CONJG(DATA(I)*TF(I))
      CALL FFT2C(DATA,19,IWK)
      DO 2 I=1,LDIM
2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END
C*****
C      SUBROUTINE: HHGG
C
C      THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER
C      AT THE OUTPUT OF THE RECEIVE FILTER
C*****
      SUBROUTINE HHGG(TF,PNOISE)
      COMPLEX TF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SUM=0.0
      SBANDW=BAUD*LSAMPL
      DO 1 L=1,LDIM
      HH=(CABS(TF(L)))**2
1 SUM=SUM+HH
      PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.
      WRITE (6,2) PNOISE
2 FORMAT(5X,'PNOISE=',F7.3,/)
      RETURN
      END
C*****
C      SUBROUTINE: ENERGY
C
C      THIS SUBROUTINE COMPUTES THE CARRIER POWER
C      AND THE AVERAGE BIT ENERGY
C*****
      SUBROUTINE ENERGY(DATA,C,EB)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON IMAX,QMAX,EBNO1,RK,CN1
      COMPLEX DATA(1)
      WATTS=0.
      DO 1 I=1,LDIM
      WATTS=WATTS+CABS(DATA(I))**2
1 CONTINUE
      C=WATTS/FLOAT(LDIM)/2.
      RRK=RK/2.
      EB=C/(RRK*BAUD)
      WRITE (6,40) C,EB
40 FORMAT(10X,'C=',F15.6,5X,'EB=',E15.8)
      RETURN
      END
C*****
C      SUBROUTINE: SYNCRO
C
C      THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA
C*****
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      INTEGER QMAXF
      COMPLEX DATA(1),AMP
      DIMENSION NI(1),NQ(1)

```

```

51A02890
51A02900
51A02910
51A02920
51A02930
51A02940
51A02950
51A02960
51A02970
51A02980
51A02990
51A03000
51A03010
51A03020
51A03030
51A03040
51A03050
51A03060
51A03070
51A03080
51A03090
51A03100
51A03110
51A03120
51A03130
51A03140
51A03150
51A03160
51A03170
51A03180
51A03190
51A03200
51A03210
51A03220
51A03230
51A03240
51A03250
51A03260
51A03270
51A03280
51A03290
51A03300
51A03310
51A03320
51A03330
51A03340
51A03350
51A03360
51A03370
51A03380
51A03390
51A03400
51A03410
51A03420
51A03430
51A03440
51A03450
51A03460
51A03470
51A03480
51A03490
51A03500
51A03510
51A03520
51A03530
51A03540
51A03550
51A03560
51A03570
51A03580
51A03590
51A03600

```

```

COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
COMMON IMAX,QMAX,EBNO1,RK,CN1
NERROR=0
IF (IOFF.EQ.0) GO TO 111
DO 6 K=1,IOFF
XX=AIMAG(DATA(LDIM))
DO 7 KK=2,LDIM
7 DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))
6 DATA(1)=CMPLX(REAL(DATA(1)),XX)
C
C SYNCHRONIZE THE RECEIVED DATA
C
111 NOLD=0
NOF=0
K=1
300 CONTINUE
NEW=0
DO 200 J=1,NSYMB
THR1I=FLOAT(NI(J))-1.
THR2I=FLOAT(NI(J))+1.
THR1Q=FLOAT(NQ(J))-1.
THR2Q=FLOAT(NQ(J))+1.
J1=K+(J-1)*LSAMPL
IF (J1.GT.LDIM) J1=J1-LDIM
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
IF ((AXBAR.GT.THR1I).AND.(AXBAR.LT.THR2I)) NEW=NEW+1
IF ((AYBAR.GT.THR1Q).AND.(AYBAR.LT.THR2Q)) NEW=NEW+1
200 CONTINUE
IF (NOLD.GE.NEW) GO TO 399
NOLD=NEW
NOF=K
399 K=K+1
LFU=2*NSYMB
C
C SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP
C
C
NSFT=LSAMPL*6
IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300
NOF=NOF-1
KT=K-2
WRITE(6,56) NOLD,NOF,KT
56 FORMAT(1X,'NO OF MAX. CORRECT SYMBOLS = ',I4,'/',1X,
*'NO OF SHIFT IN SAMPLES FOR MAX. CORRECT SYMBOLS= ',I4,'/',
*'1X,'TOTAL NO OF SHIFT FOR SEARCHING = ',I4)
C
IF (NOF.EQ.0) GO TO 230
LO=LDIM-1
DO 250 I=1,NOF
AMP=DATA(1)
DO 240 J=1,LO
240 DATA(J)=DATA(J+1)
CONTINUE
DATA(LDIM)=AMP
250 CONTINUE
C
C OPTIMIZE THE SAMPLING INSTANT
C
230 MI=1
MQ=1
EOI=FLOAT(NSYMB) + 1.
EQO=FLOAT(NSYMB) + 1.
M=22
VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
SIGMA=SQRT(VARIAN)
MOS=LSAMPL/4
DO 80 J=1,MOS
80 EI=0.
EQ=0.
IERX=0

```

51A03610
51A03620
51A03630
51A03640
51A03650
51A03660
51A03670
51A03680
51A03690
51A03700
51A03710
51A03720
51A03730
51A03740
51A03750
51A03760
51A03770
51A03780
51A03790
51A03800
51A03810
51A03820
51A03830
51A03840
51A03850
51A03860
51A03870
51A03880
51A03890
51A03900
51A03910
51A03920
51A03930
51A03940
51A03950
51A03960
51A03970
51A03980
51A03990
51A04000
51A04010
51A04020
51A04030
51A04040
51A04050
51A04060
51A04070
51A04080
51A04090
51A04100
51A04110
51A04120
51A04130
51A04140
51A04150
51A04160
51A04170
51A04180
51A04190
51A04200
51A04210
51A04220
51A04230
51A04240
51A04250
51A04260
51A04270
51A04280
51A04290
51A04300
51A04310
51A04320

```

      IERY=0
      DO 70 K=1,NSYMB
C     VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
      J1=(K-1)*LSAMPL+J
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      INDEXI=0
      AMPX=ABS(AXBAR)
      AMPI=ABS(FLOAT(NI(K)))
      THR1I=AMPI - 1.0
      THR2I=AMPI + 1.0
      IMAXF=0
      IF (AMPI.EQ.IMAX) GO TO 30
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30    IF(AMPX.LE.THR1I) INDEXI=1
      IMAXF=1
40    CONTINUE
      INDEXQ=0
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      QMAXF=0
      IF (AMPQ.EQ.QMAX) GO TO 50
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50    IF(AMPY.LE.THR1Q) INDEXQ=1
      QMAXF=1
60    CONTINUE
      IF(INDEXI.EQ.1) EI=EI + 1.
      IF(INDEXQ.EQ.1) EQ=EQ + 1.
C
C     COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB
C
      D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C     CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
      IF (IMAXF .EQ. 1) GO TO 65
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
65    D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IF (QMAXF .EQ. 1) GO TO 70
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IERX=IERX+INDEXI
      IERY=IERY+INDEXQ
70    CONTINUE
      WRITE(6,81) IERX,IERY
81    FORMAT(5X,'IERX=',I5,5X,'IERY=',I5)
      IF(EOI.LE.EI) GO TO 75
      EOI=EI
      MI=J
75    CONTINUE
      IF (EOQ.LE.EQ) GO TO 80
      EQ=EQ
      MQ=J
80    MOFF=IABS(MI-MQ)
      WRITE(6,91) MI,MQ
91    FORMAT(5X,'MI=',I3,5X,'MQ=',I3)
      IF(MOFF.NE.0) WRITE(6,90) MOFF
      OFF=(FLOAT(MI) + FLOAT(MQ))/FLOAT(LSAMPL)

```

```

51A04330
51A04340
51A04350
51A04360
51A04370
51A04380
51A04390
51A04400
51A04410
51A04420
51A04430
51A04440
51A04450
51A04460
51A04470
51A04480
51A04490
51A04500
51A04510
51A04520
51A04530
51A04540
51A04550
51A04560
51A04570
51A04580
51A04590
51A04600
51A04610
51A04620
51A04630
51A04640
51A04650
51A04660
51A04670
51A04680
51A04690
51A04700
51A04710
51A04720
51A04730
51A04740
51A04750
51A04760
51A04770
51A04780
51A04790
51A04800
51A04810
51A04820
51A04830
51A04840
51A04850
51A04860
51A04870
51A04880
51A04890
51A04900
51A04910
51A04920
51A04930
51A04940
51A04950
51A04960
51A04970
51A04980
51A04990
51A05000
51A05010
51A05020
51A05030
51A05040

```

```

90 FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
#12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
WRITE(6,95) OFF
95 FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
RETURN
END
C*****
C SUBROUTINE: DECODE *
C *
C THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA *
C*****
SUBROUTINE DECODE(NERROR,DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)
INTEGER QMAXF
COMPLEX DATA(1),AMP
DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
NERROR=0
DO 1 I=1,NSNR
1 PE(I)=0.
COMMON IMAX,QMAX,EBNO1,RK,CN1
C CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE
C ERRORS IN ADJACENT LEVELS ARE CONSIDERED
3 DO 2 K=1,NSYMB
J1=(K-1)*LSAMPL+MI
J2=(K-1)*LSAMPL + MQ
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.
INDEXI=0
AMPX=ABS(AXBAR)
AMPI=ABS(FLOAT(NI(K)))
THR1I=AMPI - 1.0
THR2I=AMPI + 1.0
IMAXF=0
IF (AMPI.EQ.IMAX) GO TO 30
IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
GO TO 40
30 IF(AMPX.LE.THR1I) INDEXI=1
IMAXF=1
40 CONTINUE
INDEXQ=0
AMPY=ABS(AYBAR)
AMPQ=ABS(FLOAT(NQ(K)))
THR1Q=AMPQ - 1.0
THR2Q=AMPQ + 1.0
QMAXF=0
IF (AMPQ.EQ.QMAX) GO TO 50
IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
GO TO 60
50 IF(AMPY.LE.THR1Q) INDEXQ=1
QMAXF=1
60 CONTINUE
IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1
C
C COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.
C THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS
C 1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE.
C FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2)))
C AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +
C 1/2*ERFC(D2/(SIGMA*SQRT(2))).
C
DO 4 M=1,NSNR
EBNO(M)=EBNO1+1.0*FLOAT(M-1)
VARIAN=PNOISE*EB*(10.**(-0.1*EBNO(M)))
SIGMA=SQRT(VARIAN)
D1=ABS(AMPX-THR1I)
ARG=D1/(SIGMA*SQRT(2.))
C CHECK IF ARG IS LARGE IN WHICH CASE PE IS INSIGNIFICANT
C CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN
C
NERI=M
IF (INDEXI.EQ.1) GO TO 153

```

51A05050
51A05060
51A05070
51A05080
51A05090
51A05100
51A05110
51A05120
51A05130
51A05140
51A05150
51A05160
51A05170
51A05180
51A05190
51A05200
51A05210
51A05220
51A05230
51A05240
51A05250
51A05260
51A05270
51A05280
51A05290
51A05300
51A05310
51A05320
51A05330
51A05340
51A05350
51A05360
51A05370
51A05380
51A05390
51A05400
51A05410
51A05420
51A05430
51A05440
51A05450
51A05460
51A05470
51A05480
51A05490
51A05500
51A05510
51A05520
51A05530
51A05540
51A05550
51A05560
51A05570
51A05580
51A05590
51A05600
51A05610
51A05620
51A05630
51A05640
51A05650
51A05660
51A05670
51A05680
51A05690
51A05700
51A05710
51A05720
51A05730
51A05740
51A05750
51A05760

```

      IF (ARG.GT.12.) ARG=12.
      PEI=ERFC(ARG)/2.
      IF (IMAXF.EQ.1) GO TO 100
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEI=PEI + ERFC(ARG)/2.
100  D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (INDEXQ.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEQ=ERFC(ARG)/2.
      IF (QMAXF.EQ.1) GO TO 110
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEQ=PEQ + ERFC(ARG)/2.
110  IF(PEI.LT.1.E-15) PEI=0.
      IF(PEQ.LT.1.E-15) PEQ=0.
      4  PE(M)=PE(M)+(PEI+PEQ)/2.
      2  CONTINUE
      DO 5 I=1,NSNR
      5  PE(I)=PE(I)/FLOAT(NSYMB)
C      PRINT 150
C150  FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)
C      WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)
C151  FORMAT(5X,F5.1,10X,E10.2)
      WRITE (6,152) NERROR
152  FORMAT(/5X,'ERRORS=',I5,/)
      GO TO 155
153  PRINT 154
154  FORMAT (10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)
      WRITE(6,157) NERI
157  FORMAT(5X,'NERI=',I5)
155  CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: DRAWCN
C
C      THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C      P(E) AS LOW AS 1.0E-10 AND A C/N RATIO AS HIGH AS 42 DB
C*****
C      SUBROUTINE DRAWCN(XARRAY,YARRAY,JCURV,JLAST,A)
C      DIMENSION XARRAY(1),YARRAY(1),XA(150),XX(150),YY(150)
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD
C      COMMON IMAX,QMAX,EBNO1,RK,CN1
C      M=NSNR
C      BR=RK*BAUD
C      NBW=BAUD/A
C      CF=BR/NBW
C      DO 1 K = 1 , NSNR
C      XA(K)=XARRAY( K ) + 10*(ALOG10(CF))
C      IF(YARRAY( K ).LE.1.0E-10) GOTO 3
C      GOTO 1
C      3  M= K-1
C      GOTO 4
C      1  CONTINUE
C      4  CONTINUE
C      I=M+1
C      II=I+1
C      XA(I)=9.0
C      XA(II)=1.0
C      YARRAY(I)=1.E-10
C      YARRAY(II)=0.4
C      GO TO 2
C      IF(JCURV.GT.1) GOTO 2
C
C      ESTABLISH THE SURFACE AREA.
C
C      CALL PLOTS(30.0,37.5)
C      CALL FACTOR(0.85)

```

51A05770
 51A05780
 51A05790
 51A05800
 51A05810
 51A05820
 51A05830
 51A05840
 51A05850
 51A05860
 51A05870
 51A05880
 51A05890
 51A05900
 51A05910
 51A05920
 51A05930
 51A05940
 51A05950
 51A05960
 51A05970
 51A05980
 51A05990
 51A06000
 51A06010
 51A06020
 51A06030
 51A06040
 51A06050
 51A06060
 51A06070
 51A06080
 51A06090
 51A06100
 51A06110
 51A06120
 51A06130
 51A06140
 51A06150
 51A06160
 51A06170
 51A06180
 51A06190
 51A06200
 51A06210
 51A06220
 51A06230
 51A06240
 51A06250
 51A06260
 51A06270
 51A06280
 51A06290
 51A06300
 51A06310
 51A06320
 51A06330
 51A06340
 51A06350
 51A06360
 51A06370
 51A06380
 51A06390
 51A06400
 51A06410
 51A06420
 51A06430
 51A06440
 51A06450
 51A06460
 51A06470
 51A06480

```

C          ESTABLISH THE ORIGIN.
C          CALL PLOT(3.0,1.5,-3)
C          DRAW THE LOGARITHMIC Y-AXIS.
C          CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,25.,
+90.,1.0E-10,.4)
C          DRAW THE LINEAR X-AXIS.
C          CALL AXIS(0.0,0.0,9HC/N IN dB,-9,
+32.0,0.0,9.0,1.0)
C          PLOT THE DATA IN LOG-LINEAR MODE.
C
2 CONTINUE
  PRINT 150
150 FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',7X,'C/N',/)
  WRITE(6,151) (XARRAY(I),YARRAY(I),XA(I),I=1,M)
151 FORMAT(5X,F5.0,10X,E10.2,10X,F5.1)
  GO TO 45
  CALL LGLIN(XA,YARRAY,M,1,4,JCURV,1)
  IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
45 RETURN
  END
C*****
C  SUBROUTINE: EYEQ
C
C  THIS SUBROUTINE DRAWS THE EYE DIAGRAM
C  FOR A TWO SYMBOL DURATION
C*****
C  SUBROUTINE EYEQ(DATA)
C  DIMENSION DATA2(66),DATA3(66)
C  COMPLEX DATA(1)
C  COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
C  COMMON /NUMB2/ NSNR,NSYMB,BAUD
C
C  ESTABLISH THE SURFACE AREA.
C
C  CALL PLOTS(37.0,27.5)
C
C  ESTABLISH THE ORIGIN.
C
C  CALL PLOT(3.0,13.0,-3)
C
C  DRAW THE TIME AXIS.
C
C  CALL AXIS(0.0,0.0,1H,-1,24.,0.0,1.0,2.8)
C
C  DRAW THE AMPLITUDE AXIS.
C
C  CALL AXIS(0.0,-8.00,1H,1,16.0,90.0,-24.,3.0)
C
C  PLOT THE DATA.
C
  JJ=2*LSAMPL
  DATA2(JJ+1)=0.0
  DATA2(JJ+2)=3.0
  DATA3(JJ+1)=1.0
  DATA3(JJ+2)=2.8
  M2=NSYMB/8
  M3=3*M2+1
  M4=4*M2-1
  DO 4 KK=1,M2
  DO 2 I=1,JJ
    II=(KK-1)*LSAMPL+I
    IF(II.GT.LDIM) II=II-(II/LDIM)*LDIM
    DATA2(I)=REAL(DATA(II))
2 DATA3(I)=FLOAT(I)
  CALL LINE(DATA3,DATA2,JJ,1,0,0)

```

51A06490
51A06500
51A06510
51A06520
51A06530
51A06540
51A06550
51A06560
51A06570
51A06580
51A06590
51A06600
51A06610
51A06620
51A06630
51A06640
51A06650
51A06660
51A06670
51A06680
51A06690
51A06700
51A06710
51A06720
51A06730
51A06740
51A06750
51A06760
51A06770
51A06780
51A06790
51A06800
51A06810
51A06820
51A06830
51A06840
51A06850
51A06860
51A06870
51A06880
51A06890
51A06900
51A06910
51A06920
51A06930
51A06940
51A06950
51A06960
51A06970
51A06980
51A06990
51A07000
51A07010
51A07020
51A07030
51A07040
51A07050
51A07060
51A07070
51A07080
51A07090
51A07100
51A07110
51A07120
51A07130
51A07140
51A07150
51A07160
51A07170
51A07180
51A07190
51A07200

```

4  CONTINUE
   CALL PLOT(0.0,0.0,999)
   RETURN
   END
C*****
C  SUBROUTINE: ATT
C
C  THIS SUBROUTINE PERFORMS THE ATTENUATION OF A COMPLEX
C  SIGNAL (AT IS THE ATTENUATION FACTOR IN DB)
C*****
   SUBROUTINE ATT(DATA,AT)
   COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
   COMPLEX DATA(1)
   ATTT=10.**(-AT/20.)
   DO 10 I=1,LDIM
10  DATA(I)=DATA(I)*ATTT
   RETURN
   END
C*****
C  SUBROUTINE: OFFSET
C
C  THIS SUBROUTINE SHIFTS THE DATA SEQUENCE BY IOF
C*****
   SUBROUTINE OFFSET(DATA,IOF)
   COMPLEX DATA(1),BUFF(1)
   COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
   COMMON /NUMB2/ NSNR,NSYMB,BAUD,NO1,NO2
   J=LDIM-IABS(IOF)
   K=J+1
   IF (IOF.EQ.0) GOTO 9
   IF (IOF.LT.0) GOTO 4
   DO 1 I=1,J
1  BUFF(I)=DATA(I)
   DO 2 I=K,LDIM
2  DATA(I-K)=DATA(I)
   DO 3 I=1,J
3  DATA(I+IOF)=BUFF(I)
   GOTO 9
4  IOF=-IOF
   L=IOF+1
   DO 5 I=L,LDIM
5  BUFF(I-IOF)=DATA(I)
   DO 6 I=1,IOF
6  DATA(I+J)=DATA(I)
   DO 7 I=1,J
7  DATA(I)=BUFF(I)
9  RETURN
   END
C*****
C  SUBROUTINE: PHASE
C
C  THIS SUBROUTINE PERFORMS THE PHASE SHIFT
C  OF A COMPLEX DATA SEQUENCE
C*****
   SUBROUTINE PHASE(DATA,PHI)
   COMPLEX EPS,DATA(1)
   COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
   EPS=CMPLX(COS(PHI),-SIN(PHI))
   DO 10 I=1,LDIM
10  DATA(I)=DATA(I)*EPS
   RETURN
   END
C*****
C  SUBROUTINE: RAND
C
C  THIS SUBROUTINE GENERATES RANDOM SYMBOLS
C*****
   SUBROUTINE RAND(IR,N,DSEED)
   INTEGER IR(1)
   DOUBLE PRECISION DSEED
   DO 3000 I=1,N
   R=GGUBFS(DSEED)

```

```

51A07210
51A07220
51A07230
51A07240
51A07250
51A07260
51A07270
51A07280
51A07290
51A07300
51A07310
51A07320
51A07330
51A07340
51A07350
51A07360
51A07370
51A07380
51A07390
51A07400
51A07410
51A07420
51A07430
51A07440
51A07450
51A07460
51A07470
51A07480
51A07490
51A07500
51A07510
51A07520
51A07530
51A07540
51A07550
51A07560
51A07570
51A07580
51A07590
51A07600
51A07610
51A07620
51A07630
51A07640
51A07650
51A07660
51A07670
51A07680
51A07690
51A07700
51A07710
51A07720
51A07730
51A07740
51A07750
51A07760
51A07770
51A07780
51A07790
51A07800
51A07810
51A07820
51A07830
51A07840
51A07850
51A07860
51A07870
51A07880
51A07890
51A07900
51A07910
51A07920

```

```

IR(I)=IFIX(R)
IF (R.LE.0.041) IR(I)=23
IF (R.GT.0.041) .AND. (R.LE.0.082) IR(I)=21
IF (R.GT.0.082) .AND. (R.LE.0.123) IR(I)=19
IF (R.GT.0.123) .AND. (R.LE.0.164) IR(I)=17
IF (R.GT.0.164) .AND. (R.LE.0.205) IR(I)=15
IF (R.GT.0.205) .AND. (R.LE.0.246) IR(I)=13
IF (R.GT.0.246) .AND. (R.LE.0.287) IR(I)=11
IF (R.GT.0.287) .AND. (R.LE.0.328) IR(I)=9
IF (R.GT.0.328) .AND. (R.LE.0.369) IR(I)=7
IF (R.GT.0.369) .AND. (R.LE.0.41) IR(I)=5
IF (R.GT.0.41) .AND. (R.LE.0.451) IR(I)=3
IF (R.GT.0.451) .AND. (R.LE.0.492) IR(I)=1
IF (R.GT.0.492) .AND. (R.LE.0.533) IR(I)=1
IF (R.GT.0.533) .AND. (R.LE.0.574) IR(I)=3
IF (R.GT.0.574) .AND. (R.LE.0.615) IR(I)=5
IF (R.GT.0.615) .AND. (R.LE.0.656) IR(I)=7
IF (R.GT.0.656) .AND. (R.LE.0.697) IR(I)=9
IF (R.GT.0.697) .AND. (R.LE.0.738) IR(I)=11
IF (R.GT.0.738) .AND. (R.LE.0.779) IR(I)=13
IF (R.GT.0.779) .AND. (R.LE.0.82) IR(I)=15
IF (R.GT.0.82) .AND. (R.LE.0.861) IR(I)=17
IF (R.GT.0.861) .AND. (R.LE.0.902) IR(I)=19
IF (R.GT.0.902) .AND. (R.LE.0.943) IR(I)=21
IF (R.GT.0.943) .AND. (R.LE.0.984) IR(I)=23
IF (R.GT.0.984) IR(I)=23
3000 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: FRESHI
C
C      THIS SUBROUTINE SHIFTS A COMPLEX SIGNAL
C      IN THE FREQUENCY DOMAIN
C*****
      SUBROUTINE FRESHI(DATA,FFFF)
      COMPLEX DATA(1)
      DIMENSION IWK(20)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL,SBANDW
      CALL FFT2C(DATA,19,IWK)
      IFOF=IFIX(FFFF*FLOAT(LDIM)/SBANDW)
      CALL OFFSET(DATA,IFOF)
      DO 1 I=1,LDIM
1 DATA(I)=CONJG(DATA(I))
      CALL FFT2C(DATA,19,IWK)
      DO 2 I=1,LDIM
2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END

```

51A07930
51A07940
51A07950
51A07960
51A07970
51A07980
51A07990
51A08000
51A08010
51A08020
51A08030
51A08040
51A08050
51A08060
51A08070
51A08080
51A08090
51A08100
51A08110
51A08120
51A08130
51A08140
51A08150
51A08160
51A08170
51A08180
51A08190
51A08200
51A08210
51A08220
* 51A08230
* 51A08240
* 51A08250
* 51A08260
51A08270
51A08280
51A08290
51A08300
51A08310
51A08320
51A08330
51A08340
51A08350
51A08360
51A08370
51A08380
51A08390
51A08400
51A08410


```

C***** PJ 00010
C      * PJ 00020
C      THIS PROGRAM CALCULATES THE DEGRADATION OF AN * PJ 00030
C      16QAM SYSTEM * PJ 00040
C      IN THE PRESENCE OF SINUSOIDAL SYSTEM CAUSED PHASE JITTER * PJ 00050
C      * PJ 00060
C      PARAMETERS: BETA = THE AMPLITUDE OF S.J. IN RADS * PJ 00070
C      FM = THE FREQUENCY OF THE S.J. IN HZ * PJ 00080
C      FB = THE BIT RATE IN BITS/SEC * PJ 00090
C      NSNR = EB/NO IN DB * PJ 00100
C***** PJ 00110
C      DIMENSION THETA(10000 ), PE(10000 ), PROB(50) PJ 00120
C      DIMENSION PEA(10000 ), PEB(10000 ), PEC(10000 ), PED(10000 ) PJ 00130
C      DIMENSION PEPAM(50), PEQAM(50) PJ 00140
C      DIMENSION XARRAY(50), YARRAY(50) PJ 00150
C      NSNR = 23 PJ 00160
C      BETA = 0.000 PJ 00170
C      FM = 120.0 PJ 00180
C      FB = 200000.0 PJ 00190
C PJ 00200
C PJ 00210
C PJ 00220
C      NRUNS = 1 PJ 00230
C PJ 00240
C PJ 00250
C      CALCULATION OF THE THEORETICAL SYMBOL ERROR PROBABILITY FOR PJ 00260
C      A-QAM PJ 00270
C      WITH C BITS/SYMBOL PJ 00280
C PJ 00290
C PJ 00300
C      INITIALIZATION PJ 00310
C      A = 16. PJ 00320
C      C = 4. PJ 00330
C PJ 00340
C      START THE CALCULATIONS PJ 00350
C PJ 00360
C      DO 124 I= 1 , NSNR PJ 00370
C      BB=10** (FLOAT(I)/10.) PJ 00380
C      A1 = SQRT (A) PJ 00390
C      B = SQRT (3.*C* BB / (2.*(A-1.))) PJ 00400
C      IF (B.GT.12.) B=12. PJ 00410
C      PEPAM (I) =(1.-(1./A1))*ERFC (B) PJ 00420
C      IF (PEPAM(I) .LT.1.E-35) PEPAM(I)=0.0 PJ 00430
C      PEQAM (I) = 2.*PEPAM(I)-(PEPAM(I)**2) PJ 00440
124 CONTINUE PJ 00450
C      WRITE (6,121) A PJ 00460
121 FORMAT (10X,'THEORETICAL SYMBOL ERROR PROBABILITY OF',F6.0,2X, PJ 00470
C      '$QAM') PJ 00480
C      WRITE (6,125) C PJ 00490
125 FORMAT (1X,'WITH THEORETICAL EFFICIENCY OF',F5.0,2X,'BITS/S/HZ'//) PJ 00500
C      WRITE (6,122) PJ 00510
122 FORMAT(17X,'EB/NO',10X,'THEORETICAL PROB. OF SYMBOL ERROR') PJ 00520
C      WRITE (6,123) (I, PEQAM(I), I=1,NSNR) PJ 00530
123 FORMAT (15X, I5 ,18X,E13.6) PJ 00540
C      WRITE(6,56) PJ 00550
56 FORMAT(1X,'*****'//) PJ 00560
C PJ 00570
C PJ 00580
C PJ 00590
C      MALL = IFIX(FB/FM) PJ 00600
C PJ 00610
C      DO 100 NR = 1, NRUNS PJ 00620
C      WRITE(6,12) FM, FB PJ 00630
12 FORMAT(1X,'SIN. PHASE JITTER FREQ.=' ,F6.2,1X,'HZ',3X,'BIT RATE=' , PJ 00640
C      &E10.4,1X,'BIT/SEC'//) PJ 00650
C PJ 00660
C      DO 100 NR= 1, NRUNS PJ 00670
C      PEAK-TO-PEAK PHASE JITTER DP (IN DEGREES) PJ 00680
C      DP = 2*BETA*(360./(2.*3.14159)) PJ 00690
C      WRITE(6,44) BETA, DP PJ 00700
44 FORMAT(1X,'BETA=' ,F10.3,1X,'RADS',4X,'EQUIV. PEAK-TO-PEAK PHASE PJ 00710
C      &JITTER=' , F10.4,1X,'IN DEGREES'//) PJ 00720

```

```

C      DO 20 K = 1, NSNR
      VARIAN = 10.**( 0.1*FLOAT(K))
      SIGMA = SQRT(VARIAN)
      TOT = 0.0
C      WRITE(6,31) SIGMA
31     FORMAT(1X,'SIGMA=',E10.5)
      DO 21 I = 1, MALL
      THETA(I) = BETA*COS(2.*3.14159*FM*I*(1./FB))
C      WRITE(6,87) I, THETA(I)
87     FORMAT(1X,'I=',15,3X,'THETA(I)=' ,E14.6)
C      WRITE(6,30) SIGMA
30     FORMAT(1X,'SIGMA=',E10.5)
C
C
C      GG = SIGMA/SQRT(39./16.)
C
C      DA1 = 3.*(COS(THETA(I)) - SIN(THETA(I))) - 2.
      DA2 = 3.*(COS(THETA(I)) + SIN(THETA(I))) - 2.
C
C      FF1 = GG*DA1
      FF2 = GG*DA2
      IF (FF1 .GT. 12.) FF1=12.
      IF (FF2 .GT. 12.) FF2=12.
      PEA1 = 0.5 * ERFC( FF1 )
      PEA2 = 0.5 * ERFC( FF2 )
C
C      PEA(I) = PEA1 + PEA2
      IF (PEA(I) .LT. 1.E-35) PEA(I)=0.0
C11     PEA(I) = PEA1 + PEA2 - PEA1*PEA2
CC      WRITE(6,200) PEA(I)
200     FORMAT(1X,'HI 1',11X,'PEA(I)=' ,E14.5)
C
C      DB1 = SQRT(10.) * (0.95*COS(THETA(I)) - 0.31*SIN(THETA(I))) - 2.
      DB2 = SQRT(10.) * (0.31*COS(THETA(I)) + 0.95*SIN(THETA(I)))
      DB3 = 2. - DB1
C
C      FF3 = GG*DB1
      FF4 = GG*DB2
      FF5 = GG*DB3
      IF (FF3 .GT. 12.) FF3=12.
      IF (FF4 .GT. 12.) FF4=12.
      IF (FF5 .GT. 12.) FF5=12.
      PEB1 = 0.5 * ERFC( FF3 )
      PEB2 = 0.5 * ERFC( FF4 )
      PEB3 = 0.5 * ERFC( FF5 )
C
C      PEB(I) = PEB1+PEB2+PEB3
      IF (PEB(I) .LT. 1.E-35) PEB(I)=0.0
C11     PEB(I) = PEB1+PEB2+PEB3 - PEB1*PEB2-PEB1*PEB3-PEB2*PEB3
C11     & +PEB1*PEB2*PEB3
CC      WRITE(6,201) PEB(I)
201     FORMAT(1X,'HI 2',6X,'PEB(I)=' ,E14.5)
C
C      DC1 = SQRT(10.) * (0.32*COS(THETA(I)) - 0.95*SIN(THETA(I)))
      DC2 = SQRT(10.) * (0.95*COS(THETA(I)) + 0.32*SIN(THETA(I))) - 2.
      DC3 = 2. - DC1
CC      WRITE(6,203) DC1, DC2, DC3
203     FORMAT(1X,'HI 3',2X,'DC1,DC2,DC3',3E15.5)
C
C      FF6 = GG*DC1
      FF7 = GG*DC2
      FF8 = GG*DC3
      IF (FF6 .GT. 12.) FF6=12.
      IF (FF7 .GT. 12.) FF7=12.
      IF (FF8 .GT. 12.) FF8=12.
      PEC1 = 0.5 * ERFC( FF6 )
      PEC2 = 0.5 * ERFC( FF7 )
      PEC3 = 0.5 * ERFC( FF8 )
CC      WRITE(6,204) PEC1, PEC2, PEC3
204     FORMAT(1X,'HI 4',2X,'PEC1,PEC2,PEC3',3E15.5)

```

PJ 00730
 PJ 00740
 PJ 00750
 PJ 00760
 PJ 00770
 PJ 00780
 PJ 00790
 PJ 00800
 PJ 00810
 PJ 00820
 PJ 00830
 PJ 00840
 PJ 00850
 PJ 00860
 PJ 00870
 PJ 00880
 PJ 00890
 PJ 00900
 PJ 00910
 PJ 00920
 PJ 00930
 PJ 00940
 PJ 00950
 PJ 00960
 PJ 00970
 PJ 00980
 PJ 00990
 PJ 01000
 PJ 01010
 PJ 01020
 PJ 01030
 PJ 01040
 PJ 01050
 PJ 01060
 PJ 01070
 PJ 01080
 PJ 01090
 PJ 01100
 PJ 01110
 PJ 01120
 PJ 01130
 PJ 01140
 PJ 01150
 PJ 01160
 PJ 01170
 PJ 01180
 PJ 01190
 PJ 01200
 PJ 01210
 PJ 01220
 PJ 01230
 PJ 01240
 PJ 01250
 PJ 01260
 PJ 01270
 PJ 01280
 PJ 01290
 PJ 01300
 PJ 01310
 PJ 01320
 PJ 01330
 PJ 01340
 PJ 01350
 PJ 01360
 PJ 01370
 PJ 01380
 PJ 01390
 PJ 01400
 PJ 01410
 PJ 01420
 PJ 01430
 PJ 01440

```

C
  PEC(I) = PEC1+PEC2+PEC3
  IF (PEC(I) .LT. 1.E-35) PEC(I)=0.0
C11 PEC(I) = PEC1+PEC2+PEC3 - PEC1*PEC2-PEC1*PEC3-PEC2*PEC3
C11 & +PEC1*PEC2*PEC3
CC WRITE(6,202) PEC(I)
202 FORMAT(1X,'HI 5',14X,'PEC(I)=',E15.5)
C
  DD1 = COS(THETA(I)) - SIN(THETA(I))
  DD2 = COS(THETA(I)) + SIN(THETA(I))
  DD3 = 2. - DD1
  DD4 = 2. - DD2
  FF9 = GG*DD1
  FF10= GG*DD2
  FF11= GG*DD3
  FF12= GG*DD4
  IF (FF9 .GT. 12.) FF9=12.
  IF (FF10.GT. 12.) FF10=12.
  IF (FF11.GT. 12.) FF11=12.
  IF (FF12.GT. 12.) FF12=12.
C
  PED1 = 0.5 * ERFC ( FF9 )
  PED2 = 0.5 * ERFC ( FF10 )
  PED3 = 0.5 * ERFC ( FF11 )
  PED4 = 0.5 * ERFC ( FF12 )
C
  PED(I) = PED1+PED2+PED3+PED4
  IF (PED(I) .LT. 1.E-35) PED(I)=0.0
C11 PED(I) = PED1+PED2+PED3+PED4
C11 & -PED1*PED2-PED1*PED3-PED2*PED3-PED1*PED4-PED2*PED4-PED3*PED4
C11 & +PED1*PED2*PED3+PED1*PED2*PED4+PED1*PED3*PED4+PED2*PED3*PED4
C11 & -PED1*PED2*PED3*PED4
C
  TOTAL SYMBOL ERROR PROBABILITY PE
  FOR THE PARTICULAR PHASE ERROR I
C
  PE(I) = PEA(I)+PEB(I)+PEC(I)+PED(I)
CC WRITE(6,77) I, PEA(I), PEB(I), PEC(I), PED(I)
77 FORMAT(1X,'I=',I4,3X,'PEA(I)=',E9.3,3X,'PEB=',E9.3,3X,'PEC=',
&E9.3,3X,'PED=',E9.3)
C
21 CONTINUE
  TT = 0.0
  DO 1 MM=1, MALL
    PPP = PE(MM)
    1 TT = TT + PPP
    TOT = TT/FLOAT(4*MALL)
    XARRAY(K) = FLOAT(K)
    YARRAY(K) = TOT
    WRITE (6,2) XARRAY(K), YARRAY(K)
    2 FORMAT(1X,'EB/NO=',F6.2,3X,'SYMBOL ERROR PROB.=',E18.9)
  20 CONTINUE
C
  CALL DRAWEB(XARRAY,YARRAY,NSNR,NR,NRUNS)
  BETA = BETA + 0.10
CCC FM = FM + 120.
C
  100 CONTINUE
  STOP
  END
C*****
C THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR *
C P(E) AS LOW AS 1.0E-8 AND AN EB/NO RATIO AS HIGH AS 16 DB. *
C*****
SUBROUTINE DRAWEB(XARRAY,YARRAY,NSNR,JCURV,JLAST)
  DIMENSION XARRAY( 1),YARRAY( 1)
  M=NSNR
  DO 1 K=1,NSNR
    IF(YARRAY(K).LE.1.0E-8) GOTO 3
    GOTO 1
  1 M=K-1
  3 GOTO 4
  4

```

PJ 01450
 PJ 01460
 PJ 01470
 PJ 01480
 PJ 01490
 PJ 01500
 PJ 01510
 PJ 01520
 PJ 01530
 PJ 01540
 PJ 01550
 PJ 01560
 PJ 01570
 PJ 01580
 PJ 01590
 PJ 01600
 PJ 01610
 PJ 01620
 PJ 01630
 PJ 01640
 PJ 01650
 PJ 01660
 PJ 01670
 PJ 01680
 PJ 01690
 PJ 01700
 PJ 01710
 PJ 01720
 PJ 01730
 PJ 01740
 PJ 01750
 PJ 01760
 PJ 01770
 PJ 01780
 PJ 01790
 PJ 01800
 PJ 01810
 PJ 01820
 PJ 01830
 PJ 01840
 PJ 01850
 PJ 01860
 PJ 01870
 PJ 01880
 PJ 01890
 PJ 01900
 PJ 01910
 PJ 01920
 PJ 01930
 PJ 01940
 PJ 01950
 PJ 01960
 PJ 01970
 PJ 01980
 PJ 01990
 PJ 02000
 PJ 02010
 PJ 02020
 PJ 02030
 PJ 02040
 PJ 02050
 PJ 02060
 PJ 02070
 PJ 02080
 PJ 02090
 PJ 02100
 PJ 02110
 PJ 02120
 PJ 02130
 PJ 02140
 PJ 02150
 PJ 02160

1	CONTINUE	PJ 02170
4	CONTINUE	PJ 02180
	I=M+1	PJ 02190
	II=I+1	PJ 02200
	XARRAY(I)=0.0	PJ 02210
	XARRAY(II)=1.0	PJ 02220
	YARRAY(I)=1.E-8	PJ 02230
	YARRAY(II)=0.4	PJ 02240
	IF(JCURV.GT.1) GOTO 2	PJ 02250
C		PJ 02260
C	ESTABLISH THE SURFACE AREA.	PJ 02270
C		PJ 02280
	CALL PLOTS(31.0,27.5)	PJ 02290
	CALL FACTOR(.85)	PJ 02300
C		PJ 02310
C	ESTABLISH THE ORIGIN.	PJ 02320
C		PJ 02330
	CALL PLOT(2.0,1.5,-3)	PJ 02340
C		PJ 02350
C		PJ 02360
	CALL SYMBOL(6.,23.,0.35,	PJ 02370
	&49H16-QAM IN THE PRESENCE OF SINUSOIDAL PHASE JITTER,0.,49)	PJ 02380
C		PJ 02390
C		PJ 02400
C	DRAW THE LOGARITHMIC Y-AXIS.	PJ 02410
C		PJ 02420
	CALL LGAXS(0.0,0.0,27HPROBABILITY OF SYMBOL ERROR,27,20.5,	PJ 02430
	+90.,1.0E-8,.4)	PJ 02440
C		PJ 02450
C	DRAW THE LINEAR X-AXIS.	PJ 02460
C		PJ 02470
	CALL AXIS(0.0,0.0,11HEB/NO IN dB,-11,	PJ 02480
	+20.0,0.0,0.0,1.0)	PJ 02490
2	CONTINUE	PJ 02500
C		PJ 02510
C	PLOT THE DATA IN LOG-LINEAR MODE.	PJ 02520
C		PJ 02530
	CALL LGLIN(XARRAY,YARRAY,M,1,2,JCURV,1)	PJ 02540
	IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)	PJ 02550
	RETURN	PJ 02560
	END	PJ 02570

```

C*****
C      THIS PROGRAM CALCULATES THE DEGRADATION OF AN
C      1SQAM SYSTEM
C      IN THE PRESENCE OF SINUSOIDAL SYSTEM CAUSED PHASE JITTER
C*****
C      THREE FREQUENCIES ARE USED ( J0, J1, J(-1) )
C      WITH CONSTANT GROUP DELAYS OF TAUPLS, TAUMIN.
C      ADDITIONALLY AMPLITUDE IMPERFECTIONS OF ERRPLS, ERRMIN ARE USED.
C*****
C      PARAMETERS:  BETA = THE AMPLITUDE OF S.J. IN RADS
C                   FM  = THE FREQUENCY OF THE S.J. IN HZ
C                   FB  = THE BIT RATE IN BITS/SEC
C                   NSNR = EB/NO IN DB
C*****
C      DELAY "DISTORTIONS":
C
C      (TAUPLS/FB)= TPLS = THE CONST. GROUP DELAY (+)
C      (TAUMIN/FB)= TMIN = THE CONST. GROUP DELAY (-)
C*****
C      AMPLITUDE DISTORTIONS:
C
C      ERRPLS = FLUCTUATIONS OF AMPLITUDE OF PBPF(+)
C      ERRMIN = FLUCTUATIONS OF AMPLITUDE OF PBPF(-)
C      NOTICE THAT BOTH ERRPLS, ERRMIN SHOULD BE > THAN 0.0111
C*****
C      PHASE DISTORTIONS:
C
C      DP1PLS = PHASE DISTORTION OF PBPF(+) AT F1
C              (IN RADS)
C      DP2PLS = PHASE DISTORTION OF PBPF(+) AT F(-1)
C              (IN RADS)
C      DP1MIN = PHASE DISTORTION OF PBPF(-) AT F1
C              (IN RADS)
C      DP2MIN = PHASE DISTORTION OF PBPF(-) AT F(-1)
C              (IN RADS)
C*****
C      PARAMETER (NP=10000,NS=50)
C      DIMENSION THETA(NP), PE(NP), SPJ(NP), SPJPLS(NP), SPJMIN(NP)
C      DIMENSION PEA(NP), PEB(NP), PEC(NP), PED(NP)
C      DIMENSION PEPAM(NS), PEQAM(NS)
C      DIMENSION XARRAY(NS), YARRAY(NS)
C      NSNR = 23
C      FM = 120.0
C      FB = 200000.0
C      NRUNS = 1
C*****
C      INITIALIZATION OF GROUP DELAY PARAMETERS:
C
C      TAUPLS = 000.0000
C      TAUMIN = 000.0000
C*****
C      INITIALIZATION OF AMPLITUDE FLUCTUATION PARAMETERS:
C
C      ERRPLS= 0.00000
C      ERRMIN= 0.00000
C*****
C      WRITE(6,12) FM, FB
C      12 FORMAT(1X,'SIN. PHASE JITTER FREQ.=',F6.2,1X,'HZ',3X,'BIT RATE=',
C      &E10.4,1X,'BIT/SEC'/)
C*****
C      TPLS = TAUPLS/FB
C      TMIN = TAUMIN/FB
C      WRITE(6,13) TPLS, TMIN
C      13 FORMAT(1X,'TPLS=',E 9.4,2X,'IN SEC',5X,'TMIN=',E12.5,
C      &2X,'IN SEC'/)
C*****
C      WRITE (6,14) ERRPLS, ERRMIN
C      14 FORMAT(1X,'ERRPLS=',E12.6,3X,'ERRMIN=',E12.6/)
C*****

```

PJ400010
 PJ400020
 PJ400030
 PJ400040
 PJ400050
 PJ400060
 PJ400070
 PJ400080
 PJ400090
 PJ400100
 PJ400110
 PJ400120
 PJ400130
 PJ400140
 PJ400150
 PJ400160
 PJ400170
 PJ400180
 PJ400190
 PJ400200
 PJ400210
 PJ400220
 PJ400230
 PJ400240
 PJ400250
 PJ400260
 PJ400270
 PJ400280
 PJ400290
 PJ400300
 PJ400310
 PJ400320
 PJ400330
 PJ400340
 PJ400350
 PJ400360
 PJ400370
 PJ400380
 PJ400390
 PJ400400
 PJ400410
 PJ400420
 PJ400430
 PJ400440
 PJ400450
 PJ400460
 PJ400470
 PJ400480
 PJ400490
 PJ400500
 PJ400510
 PJ400520
 PJ400530
 PJ400540
 PJ400550
 PJ400560
 PJ400570
 PJ400580
 PJ400590
 PJ400600
 PJ400610
 PJ400620
 PJ400630
 PJ400640
 PJ400650
 PJ400660
 PJ400670
 PJ400680
 PJ400690
 PJ400700
 PJ400710
 PJ400720

```

C      CALCULATION OF THE THEORETICAL SYMBOL ERROR PROBABILITY FOR
C      A-QAM
C      WITH      C BITS/SYMBOL
C
C      INITIALIZATION
C
C      A = 16.
C      C = 4.
C
C      START THE CALCULATIONS
C
C      DO 124 I=1,NSNR
C          BB = 10**((FLOAT(I)/10.))
C          AA1 = SQRT (A)
C          B = SQRT (3.*C*      BB      / (2.*(A-1.)))
C          IF (B.GT.12.) B=12.
C          PEPAM (I) =(1.-(1./AA1))*ERFC (B)
C          IF (PEPAM(I) .LT.1.E-35) PEPAM(I)=0.0
C          PEQAM (I) = 2.*PEPAM(I)-(PEPAM(I)**2)
124 CONTINUE
C      WRITE (6,121) A
121 FORMAT (10X,'THEORETICAL SYMBOL ERROR PROBABILITY OF',F8.0,2X,
C      '$QAM')
C      WRITE (6,125) C
125 FORMAT (1X,'WITH THEORETICAL EFFICIENCY OF',F5.0,2X,'BITS/S/HZ'//)
C      WRITE (6,122)
122 FORMAT (17X,'EB/NO',10X,'THEORETICAL PROB. OF SYMBOL ERROR')
C      WRITE (6,123) (I, PEQAM(I),      I=1,NSNR)
123 FORMAT (15X, 15 ,18X,E13.6)
C      WRITE(6,56)
56 FORMAT(1X,'*****'////)
C
C      INITIALIZATION OF PHASE DISTORTION PARAMETERS
C
C      DP1PLS = 4.00000
C      DP2PLS = 4.00000
C      DP1MIN = 4.00000
C      DP2MIN = 4.00000
C
C      DO 999 MAM=1, 10
C
C      WRITE(6,155) DP1PLS, DP2PLS, DP1MIN, DP2MIN
155 FORMAT(1X,'DP1PLS=',E10.4,1X,'DP2PLS=',E10.4,1X,'DP1MIN=',
C      &E10.4,1X,'DP2MIN=',E10.4,4X,'ALL IN RADS' ////)
C
C      DO 100 NR= 1, NRUNS
C      GO TO (1, 2, 3, 4, 5, 6, 7, 8) NR
1  RJ0 = 0.9975015620D0
   RJ1 = 0.049937526D0
   RJ2 = 0.0012489587D0
   BETA = 0.1
   GO TO 99
C
2  RJ0 = 0.9900249722D0
   RJ1 = 0.0995008326D0
   RJ2 = 0.0049833542D0
   BETA = 0.2
   GO TO 99
C
3  RJ0 = 0.9776262465D0
   RJ1 = 0.1483188163D0
   RJ2 = 0.0111658619D0
   BETA = 0.3
   GO TO 99
C
4  RJ0 = 0.9603982266D0

```

PJ400730
 PJ400740
 PJ400750
 PJ400760
 PJ400770
 PJ400780
 PJ400790
 PJ400800
 PJ400810
 PJ400820
 PJ400830
 PJ400840
 PJ400850
 PJ400860
 PJ400870
 PJ400880
 PJ400890
 PJ400900
 PJ400910
 PJ400920
 PJ400930
 PJ400940
 PJ400950
 PJ400960
 PJ400970
 PJ400980
 PJ400990
 PJ401000
 PJ401010
 PJ401020
 PJ401030
 PJ401040
 PJ401050
 PJ401060
 PJ401070
 PJ401080
 PJ401090
 PJ401100
 PJ401110
 PJ401120
 PJ401130
 PJ401140
 PJ401150
 PJ401160
 PJ401170
 PJ401180
 PJ401190
 PJ401200
 PJ401210
 PJ401220
 PJ401230
 PJ401240
 PJ401250
 PJ401260
 PJ401270
 PJ401280
 PJ401290
 PJ401300
 PJ401310
 PJ401320
 PJ401330
 PJ401340
 PJ401350
 PJ401360
 PJ401370
 PJ401380
 PJ401390
 PJ401400
 PJ401410
 PJ401420
 PJ401430
 PJ401440

```

RJ1 = 0.1960265780D0
RJ2 = 0.0197346631D0
BETA = 0.4
GO TO 99

5 RJ0 = 0.9384698072D0
RJ1 = 0.2422684577D0
RJ2 = 0.0306040235D0
BETA = 0.5
GO TO 99

6 RJ0 = 0.9120048634D0
RJ1 = 0.2867009881D0
RJ2 = 0.0436650967D0
BETA = 0.6
GO TO 99

7 RJ0 = 0.8812008886D0
RJ1 = 0.3289957415D0
RJ2 = 0.0587869444D0
BETA = 0.7
GO TO 99

8 RJ0 = 0.8462873527D0
RJ1 = 0.3688420461D0
RJ2 = 0.0758177625D0
BETA = 0.8

99 CONTINUE
AAA = RJ0
BBB = RJ1
CCC = RJ2
WRITE (6,19) AAA, BBB, CCC, BETA
19 FORMAT ('J0=',E14.8,2X,'J1=',E14.8,2X,'J2=',E14.8,2X,'BETA=',
&E 9.4/)

MALL = IFIX(FB/FM)

PEAK-TO-PEAK PHASE JITTER DP (IN DEGREES)

DP = 2*BETA*(360./((2.*3.14159)))
WRITE(6,44) BETA, DP
44 FORMAT(1X,'BETA=',F 5.2,1X,'RADS',3X,'EQUIV. PEAK-TO-PEAK PHASE
&JITTER=', F 8.4,1X,'IN DEGREES'/ )

DO 20 K = 1, NSNR
VARIAN = 10.** ( 0.1*FLOAT(K))
SIGMA = SQRT(VARIAN)
TOT = 0.0
WRITE(6,31) SIGMA
31 FORMAT(1X,'SIGMA=',E10.5)
DO 21 I = 1, MALL

HERE WE DEFINE THE PHASE FUNCTION DENOTED AS
THETA(I), WHICH IS EQUIVALENT TO GAMMA(T) OF THE THEORY

SPJ (I) = BETA*COS(2.*3.14159*FM*I*(1./FB))
SPJPLS(I) = BETA * COS(2.*3.14159*FM*(FLOAT(I)-TAUPLS)*(1./FB))
SPJMIN(I) = BETA * COS(2.*3.14159*FM*(FLOAT(I)-TAUMIN)*(1./FB))
PP = 2.*3.14159*FM*FLOAT(I)/FB
D1 = 2.*3.14159*FM*TAUPLS/FB + DP1PLS
D2 = 2.*3.14159*FM*TAUPLS/FB + DP2PLS

E1 = 2.*3.14159*FM*TAUMIN/FB + DP1MIN
E2 = 2.*3.14159*FM*TAUMIN/FB + DP2MIN

```

```

      G1 = RJ0 + RJ1*(ERRPLS*SIN(PP-D2) + 2.*COS(PP-(D1+D2)/2.)*
&      SIN((D1-D2)/2.))
      H1 = RJ1 *(ERRPLS*COS(PP-D2) + 2.*COS(PP-(D1+D2)/2.)*
&      COS((D1-D2)/2.))
C
      G2 = RJ0 +RJ1*(ERRMIN*SIN(PP-E2) + 2.*COS(PP-(E1+E2)/2.)*
&      SIN((E1-E2)/2.))
      H2 = RJ1 *(ERRMIN*COS(PP-E2) + 2.*COS(PP-(E1+E2)/2.)*
&      COS((E1-E2)/2.))
C
C-----
      WRITE (6,18) PP, D1, D2, E1, E2
      18 FORMAT('PP=',E9.3,1X,'D1=',E9.3,1X,'D2=',E9.3,1X,'E1=',E9.3,
& 'E2=',E9.3)
C
      WRITE (6,119)G1, H1, G2, H2
      119 FORMAT('G1=',E11.4,1X,'H1=',E11.4,1X,'G2=',E11.4,1X,'H2=',E11.4)
      THETA(I) = SPJ(I) - 0.5*ATAN((G1*H2+H1*G2) / (G1*G2-H1*H2))
CC
      WRITE (6,120) I, THETA(I)
      120 FORMAT('I=',I5,3X,'THETA(I)=', E11.4)
C+++++
CC
      A1 = RJ0 + RJ1*ERRPLS*SIN(2.*3.14159*FM*(FLOAT(I)-TAUPLS)/FB)
CC
      B1 = RJ1 *(ERRPLS+2.) * SPJPLS(I)
C-----
CC
      A2 = RJ0 + RJ1*ERRMIN*SIN(2.*3.14159*FM*(FLOAT(I)-TAUMIN)/FB)
CC
      B2 = RJ1 *(ERRMIN +2.) * SPJMIN(I)
C-----
CC
      THETA(I) = SPJ(I) - 0.5*ATAN((A1*B2+B1*A2)/(A1*A2-B1*B2))
C-----
CCC
      WRITE(6,87) I, THETA(I)
      87 FORMAT(1X,'I=',I5,3X,'THETA(I)=',E14.6)
C
      WRITE (6,30) SIGMA
      30 FORMAT(1X,'SIGMA=',E10.5)
C
C
C
      GG = SIGMA/SQRT(39./16.)
C
      DA1 = 3.*(COS(THETA(I)) - SIN(THETA(I))) - 2.
      DA2 = 3.*(COS(THETA(I)) + SIN(THETA(I))) - 2.
C
      FF1 = GG*DA1
      FF2 = GG*DA2
      IF (FF1 .GT. 12.) FF1=12.
      IF (FF2 .GT. 12.) FF2=12.
      PEA1 = 0.5 * ERFC( FF1 )
      PEA2 = 0.5 * ERFC( FF2 )
C
      PEA(I) = PEA1 + PEA2
C||
      PEA(I) = PEA1 + PEA2 - PEA1*PEA2
C-----
      DB1 = SQRT(10.) * (0.95*COS(THETA(I)) - 0.31*SIN(THETA(I))) - 2.
      DB2 = SQRT(10.) * (0.31*COS(THETA(I)) + 0.95*SIN(THETA(I)))
      DB3 = 2. - DB1
C
      FF3 = GG*DB1
      FF4 = GG*DB2
      FF5 = GG*DB3
      IF (FF3 .GT. 12.) FF3=12.
      IF (FF4 .GT. 12.) FF4=12.
      IF (FF5 .GT. 12.) FF5=12.
      PEB1 = 0.5 * ERFC( FF3 )
      PEB2 = 0.5 * ERFC( FF4 )
      PEB3 = 0.5 * ERFC( FF5 )
C
      PEB(I) = PEB1+PEB2+PEB3
C||
      PEB(I) = PEB1+PEB2+PEB3 - PEB1*PEB2-PEB1*PEB3-PEB2*PEB3
C||
      &      +PEB1*PEB2*PEB3
C-----
      DC1 = SQRT(10.) * (0.32*COS(THETA(I)) - 0.95*SIN(THETA(I)))
      DC2 = SQRT(10.) * (0.95*COS(THETA(I)) + 0.32*SIN(THETA(I))) - 2.
      DC3 = 2. - DC1
C

```

PJ402170
 PJ402180
 PJ402190
 PJ402200
 PJ402210
 PJ402220
 PJ402230
 PJ402240
 PJ402250
 PJ402260
 PJ402270
 PJ402280
 PJ402290
 PJ402300
 PJ402310
 PJ402320
 PJ402330
 PJ402340
 PJ402350
 PJ402360
 PJ402370
 PJ402380
 PJ402390
 PJ402400
 PJ402410
 PJ402420
 PJ402430
 PJ402440
 PJ402450
 PJ402460
 PJ402470
 PJ402480
 PJ402490
 PJ402500
 PJ402510
 PJ402520
 PJ402530
 PJ402540
 PJ402550
 PJ402560
 PJ402570
 PJ402580
 PJ402590
 PJ402600
 PJ402610
 PJ402620
 PJ402630
 PJ402640
 PJ402650
 PJ402660
 PJ402670
 PJ402680
 PJ402690
 PJ402700
 PJ402710
 PJ402720
 PJ402730
 PJ402740
 PJ402750
 PJ402760
 PJ402770
 PJ402780
 PJ402790
 PJ402800
 PJ402810
 PJ402820
 PJ402830
 PJ402840
 PJ402850
 PJ402860
 PJ402870
 PJ402880


```

      FF6 = GG*DC1
      FF7 = GG*DC2
      FF8 = GG*DC3
      IF (FF6 .GT. 12.) FF6=12.
      IF (FF7 .GT. 12.) FF7=12.
      IF (FF8 .GT. 12.) FF8=12.
      PEC1 = 0.5 *ERFC( FF6 )
      PEC2 = 0.5 * ERFC( FF7 )
      PEC3 = 0.5 * ERFC( FF8 )
C
      PEC(I) = PEC1+PEC2+PEC3
C11  PEC(I) = PEC1+PEC2+PEC3 - PEC1*PEC2-PEC1*PEC3-PEC2*PEC3
C11  &      +PEC1*PEC2*PEC3
C-----
      DD1 = COS(THETA(I)) - SIN(THETA(I))
      DD2 = COS(THETA(I)) + SIN(THETA(I))
      DD3 = 2. - DD1
      DD4 = 2. - DD2
C
      FF9 = GG*DD1
      FF10= GG*DD2
      FF11= GG*DD3
      FF12= GG*DD4
      IF (FF9 .GT. 12.) FF9=12.
      IF (FF10 .GT. 12.) FF10=12.
      IF (FF11 .GT. 12.) FF11=12.
      IF (FF12 .GT. 12.) FF12=12.
      PED1 = 0.5 * ERFC ( FF9 )
      PED2 = 0.5 * ERFC ( FF10 )
      PED3 = 0.5 * ERFC ( FF11 )
      PED4 = 0.5 * ERFC ( FF12 )
C
      PED(I) = PED1+PED2+PED3+PED4
C11  PED(I) = PED1+PED2+PED3+PED4
C11  &      -PED1*PED2-PED1*PED3-PED2*PED3-PED1*PED4-PED2*PED4-PED3*PED4
C11  &      +PED1*PED2*PED3+PED1*PED2*PED4+PED1*PED3*PED4+PED2*PED3*PED4
C11  &      -PED1*PED2*PED3*PED4
C
      TOTAL SYMBOL ERROR PROBABILITY PE
      FOR THE PARTICULAR PHASE ERROR I
C
      PE(I) = PEA(I)+PEB(I)+PEC(I)+PED(I)
CCC  WRITE(6,77) I, PEA(I), PEB(I), PEC(I), PED(I)
      77 FORMAT(1X,'I=',I4,3X,'PEA(I)=' ,E9.3,3X,'PEB=' ,E9.3,3X,'PEC=' ,
      &E9.3,3X,'PED=' ,E9.3)
C-----
21  CONTINUE
      TT = 0.0
      DO 23 MM=1, MALL
      PPP = PE(MM)
23  TT = TT + PPP
      TOT = TT/FLOAT(4*MALL)
      XARRAY(K) = FLOAT(K)
      YARRAY(K) = TOT
      WRITE (6,25)XARRAY(K), YARRAY(K)
25  FORMAT(1X,'EB/NO=' ,F6.2,3X,'SYMBOL ERROR PROB.=' ,E14.5)
20  CONTINUE
C
C      CALL DRAWEB(XARRAY,YARRAY,NSNR,NR,NRUNS)
100 CONTINUE
      DP1PLS = DP1PLS + 0.50
      DP2PLS = DP2PLS + 0.50
      DP1MIN = DP1MIN + 0.50
      DP2MIN = DP2MIN + 0.50
C
999 CONTINUE
      STOP
      END
C*****
C      THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C      P(E) AS LOW AS 1.0E-8 AND AN EB/NO RATIO AS HIGH AS 16 DB.
C*****

```

PJ402890
 PJ402900
 PJ402910
 PJ402920
 PJ402930
 PJ402940
 PJ402950
 PJ402960
 PJ402970
 PJ402980
 PJ402990
 PJ403000
 PJ403010
 PJ403020
 PJ403030
 PJ403040
 PJ403050
 PJ403060
 PJ403070
 PJ403080
 PJ403090
 PJ403100
 PJ403110
 PJ403120
 PJ403130
 PJ403140
 PJ403150
 PJ403160
 PJ403170
 PJ403180
 PJ403190
 PJ403200
 PJ403210
 PJ403220
 PJ403230
 PJ403240
 PJ403250
 PJ403260
 PJ403270
 PJ403280
 PJ403290
 PJ403300
 PJ403310
 PJ403320
 PJ403330
 PJ403340
 PJ403350
 PJ403360
 PJ403370
 PJ403380
 PJ403390
 PJ403400
 PJ403410
 PJ403420
 PJ403430
 PJ403440
 PJ403450
 PJ403460
 PJ403470
 PJ403480
 PJ403490
 PJ403500
 PJ403510
 PJ403520
 PJ403530
 PJ403540
 PJ403550
 PJ403560
 PJ403570
 PJ403580
 PJ403590
 PJ403600

```

SUBROUTINE DRAWB(XARRAY,YARRAY,NSNR,JCURV,JLAST)
DIMENSION XARRAY( 1),YARRAY( 1)
M=NSNR
DO 1 K=1,NSNR
IF(YARRAY(K).LE.1.0E-8) GOTO 3
GOTO 1
3 M=M-1
GOTO 4
1 CONTINUE
4 CONTINUE
I=M+1
II=I+1
XARRAY(I)=0.0
XARRAY(II)=1.0
YARRAY(I)=1.E-8
YARRAY(II)=0.4
IF(JCURV.GT.1) GOTO 2

C
C ESTABLISH THE SURFACE AREA.
C
C CALL PLOTS(31.0,27.5)
C CALL FACTOR(.7)
C
C ESTABLISH THE ORIGIN.
C
C CALL PLOT(2.0,1.5,-3)
C
C
C
C CALL SYMBOL(6.,23.,0.35,
&49H16-QAM IN THE PRESENCE OF SINUSOIDAL PHASE JITTER,0.,49)
C
C
C
C DRAW THE LOGARITHMIC Y-AXIS.
C
C CALL LGAXS(0.0,0.0,27HPROBABILITY OF SYMBOL ERROR,27,20.5,
+90.,1.0E-8,.4)
C
C DRAW THE LINEAR X-AXIS.
C
C CALL AXIS(0.0,0.0,11HEB/NO IN dB,-11,
+20.0,0.0,0.0,1.0)
2 CONTINUE
C
C PLOT THE DATA IN LOG-LINEAR MODE.
C
CALL LGLIN(XARRAY,YARRAY,M,1,2,JCURV,1)
IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
RETURN
END
```

PJ403610
PJ403620
PJ403630
PJ403640
PJ403650
PJ403660
PJ403670
PJ403680
PJ403690
PJ403700
PJ403710
PJ403720
PJ403730
PJ403740
PJ403750
PJ403760
PJ403770
PJ403780
PJ403790
PJ403800
PJ403810
PJ403820
PJ403830
PJ403840
PJ403850
PJ403860
PJ403870
PJ403880
PJ403890
PJ403900
PJ403910
PJ403920
PJ403930
PJ403940
PJ403950
PJ403960
PJ403970
PJ403980
PJ403990
PJ404000
PJ404010
PJ404020
PJ404030
PJ404040
PJ404050
PJ404060
PJ404070
PJ404080
PJ404090
PJ404100
PJ404110

```

C***** P1100010
C* P1100020
C* FILE NAME: P11 P1100030
C* P1100040
C*----- P1100050
C* P1100060
C* THIS PROGRAM EVALUATES THE PERFORMANCE OF P1100070
C* 16-QAM SYSTEM P1100080
C* WHICH EMPLOYS AN IN-BAND PILOT INSERTION SCHEME P1100090
C* P1100100
C***** P1100110
C* COMPLEX DATA(65536), TF(65536), TONE(65536) P1100120
C* DIMENSION NI(4096), NQ(4096) P1100130
C* INTEGER IR(4096) P1100140
C* DIMENSION PE(36), PO(32768), OUTPOW(32768) P1100150
C* DIMENSION XARRAY(36), YARRAY(36) P1100160
C* DIMENSION OUT1(513),EBNO(36) P1100170
C* COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL P1100180
C* COMMON /NUMB2/ NSNR,NSYMB,BAUD P1100190
C P1100200
C INITIALIZE PROGRAM P1100210
C P1100220
C ALPHA =1.0 P1100230
C NSYMB = 2048 P1100240
C LSAMPL = 16 P1100250
C LDIM = LSAMPL*NSYMB P1100260
C BAUD = 15. P1100270
C FBW = BAUD /2. P1100280
C IOFF = 0 P1100290
C SIMBW =BAUD*FLOAT(LSAMPL) P1100300
C NSNR = 32 P1100310
C ISIN = 1 P1100320
C NRUNS= 1 P1100330
C P1100340
C P1100350
C END OF INITIALIZATIONS P1100360
C P1100370
C WRITE (6,666) ALPHA P1100380
C 666 FORMAT (10X, 'ALPHA=', F8.4//) P1100390
C WRITE (6,667) NSYMB, LSAMPL, BAUD P1100400
C 667 FORMAT (1X, 'NSYMB=', I8,3X, 'LSAMPL=', I3,3X, 'BAUD=', F8.3///) P1100410
C WRITE (6,669) SIMBW P1100420
C 669 FORMAT(1X, 'THE SIMULATION BANDWIDTH IS = ', F9.3///) P1100430
C P1100440
C P1100450
C START OF COMPUTATIONS P1100460
C P1100470
C DO 100 NR=1,NRUNS P1100480
C DSEED=135791.D0 P1100490
C CALL RAND (NI, NSYMB, DSEED) P1100500
C DSEED=780956.D0 P1100510
C CALL RAND (NQ, NSYMB, DSEED) P1100520
C CALL SIGNAL (DATA, NI, NQ) P1100530
C CALL RCOSTX (TF) P1100540
C 33 CALL FILTER (DATA,TF) P1100550
C CALL INSERT (NR, DATA, TONE) P1100560
C CALL ENERGY (DATA,EB) P1100570
C P1100580
C P1100590
C END OF THE TRANSMITTER AND START OF THE RECEIVER P1100600
C P1100610
C CALL RCOSRX(TF) P1100620
C CALL HHGG (TF,PNOISE) P1100630
C CALL FILTER (DATA,TF) P1100640
C CALL EYEQ (DATA) P1100650
C CALL SYNCRO (DATA,PNOISE,NI,NQ,MI,MQ,EB) P1100660
C CALL DECODE (DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE ,EB,NR) P1100670
C P1100680
C P1100690
C CALL THE DRAWING SUBROUTINE P1100700
C P1100710
C DO 99 I=1,NSNR P1100720
C XARRAY(I)=EBNO(I) P1100730
C 99 YARRAY(I)=PE(I) P1100740
C P1100750
C CALL DRAWCN (ID,XARRAY,YARRAY,NR,NRUNS) P1100760

```

C	CALL DRAWED (XARRAY,YARRAY,NR,NRUNS)	P1100730
C	CALL THE EYE DIAGRAM SUBROUTINE	P1100740
C		P1100750
C	CALL EYEQ (DATA)	P1100760
C	100 CONTINUE	P1100770
C	STOP	P1100780
C	END	P1100790
C	*****	P1100800
C	SUBROUTINE: PILOT	* P1100810
C		* P1100820
C	THIS SUBROUTINE GENERATES AND INSERTS A PILOT SIGNAL	* P1100830
C	INTO THE 16-QAM SIGNAL	* P1100840
C		* P1100850
C	NOTE: THE FREQUENCY OF THE PILOT IS GIVEN BY:	* P1100860
C	(1/C) * (SYMBOL RATE) ; C IS THE VARIABLE	* P1100870
C		* P1100880
C	*****	P1100890
C	SUBROUTINE PILOT (DATA,TONE,C,AA)	P1100900
C	COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL	P1100910
C	COMPLEX DATA(1),TONE(1)	P1100920
C	GO TO 97	P1100930
C	WRITE (6,8)	P1100940
C	8 FORMAT('HI')	P1100950
C	DO 17 I=1,LDIM	P1100960
C	IF (I.LT.300) WRITE (6,8)	P1100970
C	17 TONE(I)=(0.0,0.0)	P1100980
C	D=1./C	P1100990
C	WRITE (6,3) D , C	P1101000
C	3 FORMAT(1X,'THE PILOT IS INESERTED AT ',F8.4,2X,'TIMES THE SYMBOL	RP1101010
C	RATE',3X,'(OR C=',F5.2,')'//)	P1101020
C	WRITE (6,31) AA	P1101030
C	31 FORMAT(1X,'AMPLITUDE OF THE INSERTED PILOT IS =',F7.2,3X,'ZERO -	P1101040
C	\$ PEAK'//)	P1101050
C	WRITE(6,26) (I,DATA(I),I=1,100)	P1101060
C	26 FORMAT(1X,'I=',I7,5X,'DATA(I)=' ,2E10.3)	P1101070
C	WRITE (6,19) (I,TONE(I),I=1,100)	P1101080
C	19 FORMAT(1X,'I=',I5,5X,'TONE(I) BEFORE DO-LOOP=' ,2E13.6)	P1101090
C	WRITE(6,15) (I,TONE(I),I=1,100)	P1101100
C	15 FORMAT(1X,'I=',I6,5X,'TONE(I)=' ,2E12.3)	P1101110
C	97 C1=FLOAT(LSAMPL)*C	P1101120
C	M1 = IFIX(C1)	P1101130
C	DO 10 K=1, LDIM , M1	P1101140
C	WRITE (6,45) K	P1101150
C	45 FORMAT(1X,'K=',I8)	P1101160
C	DO 10 I=1, M1	P1101170
C		P1101180
C	NOTICE THAT THE BEST SAMPLING INSTANT IS AT: *-.5*	P1101190
C		P1101200
C	A1= AA * COS((FLOAT(I) -.5)*2.0*3.141592 / FLOAT(M1))	P1101210
C	MM1 = K+I-1	P1101220
C	IF (MM1.GT.LDIM) GO TO 33	P1101230
C	10 TONE(K+I-1) = CMPLX(A1,0.0)	P1101240
C	98 DO 22 I= 1,LDIM	P1101250
C	22 TONE(I)=(0.0,0.0)	P1101260
C	33 M2=2*M1	P1101270
C	M3 = LDIM - 100	P1101280
C	M4 = LDIM	P1101290
C	WRITE(6,23) (I,TONE(I),I= 1,100)	P1101300
C	23 FORMAT(4X,'I=',I6,5X,'TONE(I) AT PILOT SUBROUTINE =' ,2E12.3)	P1101310
C	DO 11 I = 1, LDIM	P1101320
C	11 DATA(I)=DATA(I)+TONE(I)	P1101330
C	99 RETURN	P1101340
C	END	P1101350
C	*****	P1101360
C	SUBROUTINE: RCOSTX	* P1101370
C		* P1101380
C	THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A	* P1101390
C	RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A	* P1101400
C	X/SINX APERTURE EQUALIZER	* P1101410
C	*****	P1101420
C	SUBROUTINE RCOSTX(TF)	P1101430
C	COMPLEX TF(1)	P1101440

```

COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
SBANDW=BAUD*LSAMPL
NO=LDIM/2
NO1=NO+1
IF (ALPHA.EQ.0) ALPHA=0.0001
FN=LDIM*(FBW/SBANDW)
F1=(1.-ALPHA)*FN
F2=(1.+ALPHA)*FN
IFN=IFIX(FN)
IF1=IFIX(F1)+1
IF2=IFIX(F2)+1
C
C THE AMPLITUDE CHARACTERISTICS
C
  A1=3.1415926/(2.*FLOAT(IFN))
  DO 8 I=2, IF1
    TF(I)=CMPLX(1.0,0.0)
    J=I-1
C
C X/SIN(X) EQUALIZATION
C
  A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))
  TF(I)=CMPLX(A2,0.0)
8 CONTINUE
  JK=IF1+1
  DO 9 J=JK, IF2
    I=J-1
C
C ROOT OF RAISED COSINE
C
  A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))
  A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
  TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
  TF(J)=TF(J)*CMPLX(A3,0.0)
9 CONTINUE
  JH=IF2+1
  DO 10 I=JH, NO1
    TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
    NO2=NO1+1
    DO 5 I=NO2, LDIM
      TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
  RETURN
  END
C*****
C SUBROUTINE: RCOSRX *
C *
C THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A *
C RAISED COSINE FILTER WITH ARBITRARY ALPHA *
C*****
SUBROUTINE RCOSRX(TF)
  COMPLEX TF(1)
  COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
  COMMON /NUMB2/ NSNR,NSYMB,BAUD
  SBANDW=BAUD*LSAMPL
  NO=LDIM/2
  NO1=NO+1
  IF (ALPHA.EQ.0) ALPHA=0.0001
  FN=LDIM*(FBW/SBANDW)
  F1=(1.-ALPHA)*FN
  F2=(1.+ALPHA)*FN
  IFN=IFIX(FN)
  IF1=IFIX(F1)+1
  IF2=IFIX(F2)+1
C
C AMPLITUDE CHARACTERISTIC
C
  A1=3.141592/(2.*FLOAT(IFN))
  DO 8 I=2, IF1
    TF(I)=CMPLX(1.0,0.0)

```

P1101450
 P1101460
 P1101470
 P1101480
 P1101490
 P1101500
 P1101510
 P1101520
 P1101530
 P1101540
 P1101550
 P1101560
 P1101570
 P1101580
 P1101590
 P1101600
 P1101610
 P1101620
 P1101630
 P1101640
 P1101650
 P1101660
 P1101670
 P1101680
 P1101690
 P1101700
 P1101710
 P1101720
 P1101730
 P1101740
 P1101750
 P1101760
 P1101770
 P1101780
 P1101790
 P1101800
 P1101810
 P1101820
 P1101830
 P1101840
 P1101850
 P1101860
 P1101870
 P1101880
 P1101890
 P1101900
 P1101910
 P1101920
 P1101930
 P1101940
 P1101950
 P1101960
 P1101970
 P1101980
 P1101990
 P1102000
 P1102010
 P1102020
 P1102030
 P1102040
 P1102050
 P1102060
 P1102070
 P1102080
 P1102090
 P1102100
 P1102110
 P1102120
 P1102130
 P1102140
 P1102150
 P1102160

```

      J=I-1
      TF(I)=CMPLX(1.,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
        I=J-1
C
C      ROOT OF RAISED COSINE
C
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
        TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
        NO2=NO1+1
        DO 5 I=NO2,LDIM
          TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: FILTER
C
C      THIS SUBROUTINE PERFORMS THE FILTERING
C      PROCESS ON A DATA SEQUENCE
C*****
      SUBROUTINE FILTER(DATA,TF)
      COMPLEX DATA(1),TF(1)
      DIMENSION IWK(17)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      CALL FFT2C(DATA,16,IWK)
      DO 1 I=1,LDIM
1 DATA(I)=CONJG(DATA(I)*TF(I))
      CALL FFT2C(DATA,16,IWK)
      DO 2 I=1,LDIM
2 DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END
C*****
C      SUBROUTINE: HHGG
C
C      THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER
C      AT THE OUTPUT OF THE RECEIVE FILTER
C*****
      SUBROUTINE HHGG(TF,PNOISE)
      COMPLEX TF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SUM=0.0
      SBANDW=BAUD*LSAMPL
      DO 1 L=1,LDIM
        HH=(CABS(TF(L)))**2
1 SUM=SUM+HH
      PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.
      WRITE (6,2) PNOISE
2 FORMAT(5X,'PNOISE=',F7.3,/)
      RETURN
      END
C*****
C      SUBROUTINE: ENERGY
C
C      THIS SUBROUTINE COMPUTES THE CARRIER POWER
C      AND THE AVERAGE BIT ENERGY
C*****
      SUBROUTINE ENERGY(DATA,EB)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMPLEX DATA(1)
      WATTS=0.
      DO 1 I=1,LDIM

```

P1102170
 P1102180
 P1102190
 P1102200
 P1102210
 P1102220
 P1102230
 P1102240
 P1102250
 P1102260
 P1102270
 P1102280
 P1102290
 P1102300
 P1102310
 P1102320
 P1102330
 P1102340
 P1102350
 P1102360
 P1102370
 P1102380
 P1102390
 * P1102400
 * P1102410
 * P1102420
 * P1102430
 P1102440
 P1102450
 P1102460
 P1102470
 P1102480
 P1102490
 P1102500
 P1102510
 P1102520
 P1102530
 P1102540
 P1102550
 P1102560
 P1102570
 * P1102580
 * P1102590
 * P1102600
 * P1102610
 P1102620
 P1102630
 P1102640
 P1102650
 P1102660
 P1102670
 P1102680
 P1102690
 P1102700
 P1102710
 P1102720
 P1102730
 P1102740
 P1102750
 P1102760
 P1102770
 * P1102780
 * P1102790
 * P1102800
 * P1102810
 P1102820
 P1102830
 P1102840
 P1102850
 P1102860
 P1102870
 P1102880

```

      WATTS=WATTS+((CABS(DATA(I)))**2.)
1      CONTINUE
      WATTS=WATTS/(FLOAT(LDIM))
      EB=WATTS/(4.*BAUD)
      WRITE (6,40) WATTS,EB
40     FORMAT(5X,'POWER=',F15.6,3X,'BIT ENERGY=',E15.8)
      RETURN
      END
C*****
C      SUBROUTINE: SYNCRO
C
C      THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA
C*****
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,MI,MQ,EB)
      INTEGER Q7FLAG
      COMPLEX DATA(1),AMP
      DIMENSION NI(1),NQ(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      NERROR=0
      IF (IOFF.EQ.0) GO TO 111
      DO 6 K=1,IOFF
      XX=AIMAG(DATA(LDIM))
      DO 7 KK=2,LDIM
7      DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))
6      DATA(1)=CMPLX(REAL(DATA(1)),XX)
C
C      SYNCHRONIZE THE RECEIVED DATA
C
111     NOLD=0
      NOF=0
      K=1
300     CONTINUE
      NEW=0
      DO 200 J=1,NSYMB
      J1=K+(J-1)*LSAMPL
      IF(J1.GT.LDIM) J1=J1-LDIM
      AXBAR=REAL(DATA(J1))
      AYBAR=AIMAG(DATA(J1))
      SS=AXBAR*NI(J)
      IF (SS .GT. 0.) NEW=NEW+1
      SS=AYBAR*NQ(J)
      IF (SS .GT. 0.) NEW=NEW+1
200     CONTINUE
      IF(NOLD.GE.NEW) GO TO 399
      NOLD=NEW
      NOF=K
399     K=K+1
      LFU=2*NSYMB
C
C      SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP
C
      NSFT=LSAMPL*3
      IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300
      NOF=NOF-1
      WRITE(6,56) NOLD,K
56     FORMAT(1X,'NO OF CORRECT SYMBOLS = ',I6,5X,'NO OF SHIFT IN SAMPLES',
      *',I6,/)
      IF (NOF.EQ.0) GO TO 230
      LO=LDIM-1
      DO 250 I=1,NOF
      AMP=DATA(1)
      DO 240 J=1,LO
      DATA(J)=DATA(J+1)
240     CONTINUE
      DATA(LDIM)=AMP
250     CONTINUE
C
C      OPTIMIZE THE SAMPLING INSTANT
C
230     MI=1
      MQ=1

```

P1102890
 P1102900
 P1102910
 P1102920
 P1102930
 P1102940
 P1102950
 P1102960
 P1102970
 * P1102980
 * P1102990
 * P1103000
 P1103010
 P1103020
 P1103030
 P1103040
 P1103050
 P1103060
 P1103070
 P1103080
 P1103090
 P1103100
 P1103110
 P1103120
 P1103130
 P1103140
 P1103150
 P1103160
 P1103170
 P1103180
 P1103190
 P1103200
 P1103210
 P1103220
 P1103230
 P1103240
 P1103250
 P1103260
 P1103270
 P1103280
 P1103290
 P1103300
 P1103310
 P1103320
 P1103330
 P1103340
 P1103350
 P1103360
 P1103370
 P1103380
 P1103390
 P1103400
 P1103410
 P1103420
 P1103430
 P1103440
 P1103450
 P1103460
 P1103470
 P1103480
 P1103490
 P1103500
 P1103510
 P1103520
 P1103530
 P1103540
 P1103550
 P1103560
 P1103570
 P1103580
 P1103590
 P1103600

```

      EO1=FLOAT(NSYMB) + 1.
      EQ=FLOAT(NSYMB) + 1.
      M=15
      VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
      SIGMA=SQRT(VARIAN)
      DO 80 J=1,LSAMPL
      EI=0.
      EQ=0.
      DO 70 K=1,NSYMB
C
C      VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
C
      J1=(K-1)*LSAMPL+J
      AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
      AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
      INDEXI=0
      SS=FLOAT(NI(K))*AXBAR
      IF (SS .LT. 0.) INDEXI=1
      AMPX=ABS(AXBAR)
      AMPI=ABS(FLOAT(NI(K)))
      THR1I=AMPI - 1.0
      THR2I=AMPI + 1.0
      I7FLAG=0
      IF (AMPI.EQ. 3.) GO TO 30
      IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30 IF(AMPX.LE.THR1I) INDEXI=1
      I7FLAG=1
40 CONTINUE
      INDEXQ=0
      SS=FLOAT(NQ(K))*AYBAR
      IF(SS .LT. 0.) INDEXQ=1
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      Q7FLAG=0
      IF (AMPQ.EQ. 3.) GO TO 50
      IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50 IF(AMPY.LE.THR1Q) INDEXQ=1
      Q7FLAG=1
60 CONTINUE
      IF(INDEXI.EQ.1) EI=EI + 1.
      IF(INDEXQ.EQ.1) EQ=EQ + 1.
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB
C
      D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C
C      CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C
      IF (ARG.GT.12.) ARG=12.
      EI=EI +ERFC(ARG)/2.
      IF (I7FLAG .EQ. 1) GO TO 65
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EI=EI + ERFC(ARG)/2.
65 D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
      IF (Q7FLAG .EQ. 1) GO TO 70
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      EQ=EQ + ERFC(ARG)/2.
70 CONTINUE
      IF(EO1.LE.EI) GO TO 75
      EO1=EI

```

P1103610
 P1103620
 P1103630
 P1103640
 P1103650
 P1103660
 P1103670
 P1103680
 P1103690
 P1103700
 P1103710
 P1103720
 P1103730
 P1103740
 P1103750
 P1103760
 P1103770
 P1103780
 P1103790
 P1103800
 P1103810
 P1103820
 P1103830
 P1103840
 P1103850
 P1103860
 P1103870
 P1103880
 P1103890
 P1103900
 P1103910
 P1103920
 P1103930
 P1103940
 P1103950
 P1103960
 P1103970
 P1103980
 P1103990
 P1104000
 P1104010
 P1104020
 P1104030
 P1104040
 P1104050
 P1104060
 P1104070
 P1104080
 P1104090
 P1104100
 P1104110
 P1104120
 P1104130
 P1104140
 P1104150
 P1104160
 P1104170
 P1104180
 P1104190
 P1104200
 P1104210
 P1104220
 P1104230
 P1104240
 P1104250
 P1104260
 P1104270
 P1104280
 P1104290
 P1104300
 P1104310
 P1104320


```

MI=J
75 CONTINUE
IF (EQQ.LE.EQ) GO TO 80
EQQ=EQ
MQ=J
80 MOFF=IABS(MI-MQ)
IF(MOFF.NE.0) WRITE(6,90) MOFF
OFF=(FLOAT(MI) + FLOAT(MQ))/FLOAT(LSAMPL)
90 FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
#12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
WRITE(6,95) OFF
95 FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
RETURN
END
C*****
C SUBROUTINE: DECODE *
C *
C THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA *
C*****
SUBROUTINE DECODE(DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB,NR)
INTEGER Q7FLAG
COMPLEX DATA(1),AMP
DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
DIMENSION PEPAM(32),PEQAM(32),DIFF(32),PPEQAM(32)
IF (NR.NE.1 ) GO TO 51

C CALCULATION OF THE THEORETICAL SYMBOL ERROR PROBABILITY FOR
C A-QAM
C WITH C BITS/SYMBOL

C INITIALIZATION
A =16.
C = 4.

C START THE CALCULATIONS

DO 124 I=1,NSNR
BB=10**((FLOAT(I)/10.))
A1 = SQRT (A)
B = SQRT (3.*C* BB / (2.*(A-1.)))
IF (B.GT.12.) B=12.
PEPAM (I) =(1.-(1./A1))*ERFC (B)
IF (PEPAM(I) .LT.1.E-35) PEPAM(I)=0.0
PEQAM (I) = 2.*PEPAM(I)-(PEPAM(I)**2)
124 CONTINUE
WRITE (6,121) A
121 FORMAT (10X,'THEORETICAL SYMBOL ERROR PROBABILITY OF',F6.0,2X,
$'QAM')
WRITE (6,125) C
125 FORMAT (1X,'WITH THEORETICAL EFFICIENCY OF',F5.0,2X,'BITS/S/HZ'//)
WRITE (6,122)
122 FORMAT(17X,'EB/NO',10X,'THEORETICAL PROB. OF SYMBOL ERROR')
WRITE (6,123) (I, PEQAM(I), I=1,NSNR)
123 FORMAT (15X, I5 ,18X,E13.6)
C-----
51 CONTINUE
NERROR=0
DO 1 I=1, NSNR
1 PE(I)=0.

C CHECK FOR AN ERROR AT THE SAMPLING INSTANT, OSLY THOSE
C ERRORS IN ADJACENT LEVELS ARE CONSIDERED
C
DO 2 K=1,NSYMB
J1=(K-1)*LSAMPL +MI
J2=(K-1)*LSAMPL +MQ
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.
INDEXI=0

```

P1104330
P1104340
P1104350
P1104360
P1104370
P1104380
P1104390
P1104400
P1104410
P1104420
P1104430
P1104440
P1104450
P1104460
P1104470
P1104480
P1104490
P1104500
P1104510
P1104520
P1104530
P1104540
P1104550
P1104560
P1104570
P1104580
P1104590
P1104600
P1104610
P1104620
P1104630
P1104640
P1104650
P1104660
P1104670
P1104680
P1104690
P1104700
P1104710
P1104720
P1104730
P1104740
P1104750
P1104760
P1104770
P1104780
P1104790
P1104800
P1104810
P1104820
P1104830
P1104840
P1104850
P1104860
P1104870
P1104880
P1104890
P1104900
P1104910
P1104920
P1104930
P1104940
P1104950
P1104960
P1104970
P1104980
P1104990
P1105000
P1105010
P1105020
P1105030
P1105040

```

      SS=FLOAT(NI(K))*AXBAR
      IF (SS .LT. 0.) INDEXI=1
      AMPX=ABS(AXBAR)
      AMPI=ABS(FLOAT(NI(K)))
      THR1I=AMPI - 1.0
      THR2I=AMPI + 1.0
      I7FLAG=0
      IF (AMPI.EQ. 3.) GO TO 30
      IF ((AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
      GO TO 40
30  IF (AMPX.LE.THR1I) INDEXI=1
      I7FLAG=1
40  CONTINUE
      INDEXQ=0
      SS=FLOAT(NQ(K))*AYBAR
      IF (SS .LT. 0.) INDEXQ=1
      AMPY=ABS(AYBAR)
      AMPQ=ABS(FLOAT(NQ(K)))
      THR1Q=AMPQ - 1.0
      THR2Q=AMPQ + 1.0
      Q7FLAG=0
      IF (AMPQ.EQ. 3.) GO TO 50
      IF ((AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
      GO TO 60
50  IF (AMPY.LE.THR1Q) INDEXQ=1
      Q7FLAG=1
60  CONTINUE
      IF ((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.
C      THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS
C      1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE.
C      FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2)))
C      AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +
C      1/2*ERFC(D2/(SIGMA*SQRT(2))).
C
      DO 4 M=1, NSNR
      VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
      SIGMA=SQRT(VARIAN)
57  D1=ABS(AMPX-THR1I)
      ARG=D1/(SIGMA*SQRT(2.))
C
C      CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C      CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN
C
      IF (INDEXI.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEI=ERFC(ARG)/2.
      IF (I7FLAG .EQ. 1) GO TO 100
      D2=ABS(THR2I-AMPX)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEI=PEI + ERFC(ARG)/2.
100 D1=ABS(AMPY-THR1Q)
      ARG=D1/(SIGMA*SQRT(2.))
      IF (INDEXQ.EQ.1) GO TO 153
      IF (ARG.GT.12.) ARG=12.
      PEQ=ERFC(ARG)/2.
      IF (Q7FLAG .EQ. 1) GO TO 110
      D2=ABS(THR2Q-AMPY)
      ARG=D2/(SIGMA*SQRT(2.))
      IF (ARG.GT.12.) ARG=12.
      PEQ=PEQ + ERFC(ARG)/2.
110 IF (PEI.LT.1.E-15) PEI=0.
      IF (PEQ.LT.1.E-15) PEQ=0.
      4  PE(M)=PE(M)+(PEI+PEQ)/1. -(PEI*PEQ)
      2  CONTINUE
      DO 5 I=1, NSNR
      PE(I)=PE(I)/FLOAT(NSYMB)
      5  EBNO(I)=FLOAT(I)
      PRINT 150
150  FORMAT(5X, 'EB/NO', 10X, 'PROB. OF ERROR', /)

```

P1105050
 P1105060
 P1105070
 P1105080
 P1105090
 P1105100
 P1105110
 P1105120
 P1105130
 P1105140
 P1105150
 P1105160
 P1105170
 P1105180
 P1105190
 P1105200
 P1105210
 P1105220
 P1105230
 P1105240
 P1105250
 P1105260
 P1105270
 P1105280
 P1105290
 P1105300
 P1105310
 P1105320
 P1105330
 P1105340
 P1105350
 P1105360
 P1105370
 P1105380
 P1105390
 P1105400
 P1105410
 P1105420
 P1105430
 P1105440
 P1105450
 P1105460
 P1105470
 P1105480
 P1105490
 P1105500
 P1105510
 P1105520
 P1105530
 P1105540
 P1105550
 P1105560
 P1105570
 P1105580
 P1105590
 P1105600
 P1105610
 P1105620
 P1105630
 P1105640
 P1105650
 P1105660
 P1105670
 P1105680
 P1105690
 P1105700
 P1105710
 P1105720
 P1105730
 P1105740
 P1105750
 P1105760

```

      WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)
151  FORMAT(5X,F5.1,10X,E13.6)
      WRITE (6,152) NERROR
152  FORMAT(/5X,'ERRORS=',I5,/)
C    IF (NR.NE.1) GO TO 155
      DO 129 I=1,NSNR
      MM1 = I
      IF (PEQAM(I).EQ.0.0) GO TO 160
129  DIFF(I)= ABS((PEQAM(I)-PE(I))/PEQAM(I))
160  WRITE (6,128) (I,DIFF(I), I=1,MM1 )
128  FORMAT(5X,'EB/NO=',I5,5X,'THEOR. MINUS SIMUL. VALUES =',E13.8,
      $ 'NORMALIZED')
      GO TO 155
153  PRINT 154
154  FORMAT (10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)
155  CONTINUE
      RETURN
      END
C*****
C    SUBROUTINE: DRAWEB
C
C    THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C    P(E) AS LOW AS 1.0E-8 AND A EB/NO RATIO AS HIGH AS 33 DB
C*****
      SUBROUTINE DRAWEB(XARRAY,YARRAY,JCURV,JLAST)
      DIMENSION XARRAY( 1),YARRAY( 1),X(22),Y(22)
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      M=NSNR
      DO 1 K=1,NSNR
      IF(YARRAY(K).LE.1.0E-8) GOTO 3
      GOTO 1
3     M=K-1
      GOTO 4
1     CONTINUE
4     CONTINUE
      I=M+1
      II=I+1
      XARRAY(I)=0.0
      XARRAY(II)=2.0
      YARRAY(I)=1.E-8
      YARRAY(II)=0.4
      IF(JCURV.GT.1) GOTO 2
C
C    ESTABLISH THE SURFACE AREA.
C
C    CALL PLOTS(31.0,27.5)
C    CALL FACTOR(.7)
C
C    ESTABLISH THE ORIGIN.
C
C    CALL PLOT(2.0,1.5,-3)
C
C    DRAW THE LOGARITHMIC Y-AXIS.
C
C    CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,20.5,
+90.,1.0E-8,.4)
C
C    DRAW THE LINEAR X-AXIS.
C
C    CALL AXIS(0.0,0.0,11HEB/NO IN dB,-11,
+20.0,0.0,0.0,2.0)
C
2    CONTINUE
C
C    PLOT THE DATA IN LOG-LINEAR MODE.
C
C    CALL LGLIN(XARRAY,YARRAY,M,1,2,JCURV,1)
      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,903)
      RETURN
      END
C*****

```

P1105770
 P1105780
 P1105790
 P1105800
 P1105810
 P1105820
 P1105830
 P1105840
 P1105850
 P1105860
 P1105870
 P1105880
 P1105890
 P1105900
 P1105910
 P1105920
 P1105930
 P1105940
 P1105950
 P1105960
 P1105970
 P1105980
 P1105990
 P1106000
 P1106010
 P1106020
 P1106030
 P1106040
 P1106050
 P1106060
 P1106070
 P1106080
 P1106090
 P1106100
 P1106110
 P1106120
 P1106130
 P1106140
 P1106150
 P1106160
 P1106170
 P1106180
 P1106190
 P1106200
 P1106210
 P1106220
 P1106230
 P1106240
 P1106250
 P1106260
 P1106270
 P1106280
 P1106290
 P1106300
 P1106310
 P1106320
 P1106330
 P1106340
 P1106350
 P1106360
 P1106370
 P1106380
 P1106390
 P1106400
 P1106410
 P1106420
 P1106430
 P1106440
 P1106450
 P1106460
 P1106470
 P1106480

```

C      SUBROUTINE: DRAWCN                                * P1106490
C                                                         * P1106500
C      THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR * P1106510
C      P(E) AS LOW AS 1.0E-8 AND A C/N RATIO AS HIGH AS 35 DB * P1106520
C***** P1106530
C      SUBROUTINE DRAWCN(ID,XARRAY,YARRAY,JCURV,JLAST)    P1106540
C      DIMENSION XARRAY(34),YARRAY(34),X(29),Y(29)      P1106550
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL         P1106560
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD                   P1106570
C      M=NSNR                                             P1106580
C      CF=4.                                              P1106590
C      DO 1 K=1,NSNR                                     P1106600
C      XARRAY(K)=XARRAY(K) + 10*(ALOG10(CF))             P1106610
C      IF(YARRAY(K).GE.1.0) YARRAY(K)=1.0               P1106620
C      IF(YARRAY(K).LE.1.0E-11)GOTO 3                    P1106630
C      GOTO 1                                              P1106640
3     M=M-1                                              P1106650
C      GOTO 4                                              P1106660
1     CONTINUE                                           P1106670
4     CONTINUE                                           P1106680
C      I=M+1                                              P1106690
C      II=I+1                                             P1106700
C      XARRAY(I)= 1.0                                    P1106710
C      XARRAY(II)=2.0                                    P1106720
C      YARRAY(I)=1.E-11                                  P1106730
C      YARRAY(II)=0.5                                    P1106740
C      IF(JCURV.GT.1) GOTO 2                             P1106750
C                                                         P1106760
C      ESTABLISH THE SURFACE AREA.                       P1106770
C                                                         P1106780
C      CALL PLOTS(30.0,27.5)                             P1106790
C      CALL FACTOR(.8)                                    P1106800
C                                                         P1106810
C      ESTABLISH THE ORIGIN.                             P1106820
C                                                         P1106830
C      CALL PLOT(3.0,1.5,-3)                              P1106840
C                                                         P1106850
C      WRITE THE FOLLOWING SYMBOL:                        P1106860
C                                                         P1106870
C      CALL SYMBOL(8., 23.,0.35,28HSYMBOL ERROR PROB. FOR 16QAM,0.,32) P1106880
C                                                         P1106890
C      DRAW THE LOGARITHMIC Y-AXIS.                      P1106900
C                                                         P1106910
C      CALL LGAXS(0.0,0.0,29HPROBABILITY OF A SYMBOL ERROR,29,22., P1106920
C      +90.,1.0E-11,0.5)                                P1106930
C                                                         P1106940
C      DRAW THE LINEAR X-AXIS.                           P1106950
C                                                         P1106960
C      CALL AXIS(0.0,0.0,9HC/N IN dB,-9,                P1106970
C      +18.0,0.0, 1.0,2.0)                              P1106980
C                                                         P1106990
C      PLOT THE DATA IN LOG-LINEAR MODE.                P1107000
C                                                         P1107010
C      CALL LGLIN(XARRAY,YARRAY,M,1,1,JCURV,1)           P1107020
C      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)         P1107030
C      RETURN                                             P1107040
C      END                                                P1107050
C***** P1107060
C      SUBROUTINE: EYEQ                                    * P1107070
C                                                         * P1107080
C      THIS SUBROUTINE DRAWS THE EYE DIAGRAM             * P1107090
C      FOR A TWO SYMBOL DURATION                         * P1107100
C***** P1107110
C      SUBROUTINE EYEQ(DATA)                              P1107120
C      DIMENSION DATA2(34),DATA3(34)                   P1107130
C      COMPLEX DATA(1)                                  P1107140
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL         P1107150
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD                   P1107160
C      WRITE (6,33)                                       P1107170
33  FORMAT (1X,'THE SUBROUTINE    EYEQ    IS CALLED'///) P1107180
C      ESTABLISH THE SURFACE AREA.                      P1107190
C      CALL PLOTS(35.0,27.5)                             P1107200

```

```

C      ESTABLISH THE ORIGIN.                                P1107210
C      CALL PLOT(3.0,12.0,-3)                                P1107220
C      CALL AXIS(0.0,0.0,1H,-1, 25.5,0.0,1.0, 1.26)         P1107230
C      DRAW THE AMPLITUDE AXIS.                              P1107240
C      CALL AXIS(0.0,- 6.0,1H,1,12.0,90.0,-6.00,1.00)       P1107250
C      PLOT THE DATA.                                       P1107260
C      JJ=2*LSAMPL                                           P1107270
C      DATA2(JJ+1)=0.0                                       P1107280
C      DATA2(JJ+2)=1.00                                       P1107290
C      DATA3(JJ+1)=1.0                                       P1107300
C      DATA3(JJ+2)=1.26                                       P1107310
C      M2=NSYMB/8                                             P1107320
C      DO 4 KK=10, M2                                         P1107330
C      DO 2 I=1,JJ                                           P1107340
C      II=(KK-1)*2*LSAMPL+I                                   P1107350
C      DATA2(I)=REAL(DATA(II))                               P1107360
2     DATA3(I)=FLOAT(I)                                       P1107370
C      CALL LINE(DATA3,DATA2,JJ,1,0,0)                       P1107380
4     CONTINUE                                                P1107390
C      CALL PLOT(0.0,0.0,999)                                P1107400
C      RETURN                                                P1107410
C      END                                                    P1107420
C      END                                                    P1107430
C*****
C      SUBROUTINE: INSERT. *                                P1107440
C *                                                            P1107450
C *                                                            P1107460
C      THIS SUBROUTINE INSERTS A PILOT TONE INTO THE 16-QAM SIGNAL * P1107470
C*****
C      SUBROUTINE INSERT (NR,DATA,TONE)                      P1107480
C      COMPLEX DATA(1), TONE(1)                             P1107490
C      GO TO ( 1,1,2,3,4, 5, 6, 7, 8, 9, 10,11,12,13), NR   P1107500
1     C=2.5                                                    P1107510
C      AA = 2.0                                               P1107520
C      GO TO 22                                               P1107530
2     C=1.1                                                    P1107540
C      AA = 1.0                                               P1107550
C      GO TO 22                                               P1107560
3     C=1.2                                                    P1107570
C      AA = 1.0                                               P1107580
C      GO TO 22                                               P1107590
4     C=1.3                                                    P1107600
C      AA = 1.0                                               P1107610
C      GO TO 22                                               P1107620
5     C=1.4                                                    P1107630
C      AA = 1.0                                               P1107640
C      GO TO 22                                               P1107650
6     C=1.5                                                    P1107660
C      AA = 1.0                                               P1107670
C      GO TO 22                                               P1107680
7     AA = 1.0                                                P1107690
C      C=1.6                                                  P1107700
C      GO TO 22                                               P1107710
8     AA = 1.0                                                P1107720
C      C=1.7                                                  P1107730
C      GO TO 22                                               P1107740
9     AA = 1.0                                                P1107750
C      C=1.8                                                  P1107760
C      GO TO 22                                               P1107770
10    AA = 1.0                                                P1107780
C      C=1.9                                                  P1107790
C      GO TO 22                                               P1107800
11    AA = 1.0                                                P1107810
C      C=2.0                                                  P1107820
C      GO TO 22                                               P1107830
12    AA = 1.0                                                P1107840
C      C=2.1                                                  P1107850
C      GO TO 22                                               P1107860
13    AA = 1.0                                                P1107870
C      C = 2.2                                                P1107880
22    WRITE(6,19)                                             P1107890
19    FORMAT(1X,'SUBROUTINE PILOT IS CALLED'////)           P1107900
C      CALL PILOT (DATA,TONE,C,AA)                           P1107910
C      CALL PILOT (DATA,TONE,C,AA)                           P1107920

```

```

74 RETURN P1107930
END P1107940
C***** P1107950
C SUBROUTINE: RAND * P1107960
C * P1107970
C THIS SUBROUTINE GENERATES RANDOM SYMBOLS * P1107980
C***** P1107990
SUBROUTINE RAND(IR,N,DSEED) P1108000
INTEGER IR(1) P1108010
DOUBLE PRECISION DSEED P1108020
DO 3000 I=1,N P1108030
R=GGUBFS(DSEED) P1108040
IR(I)=IFIX(R) P1108050
IF (R.LE.0.031) IR(I)= 3 P1108060
IF ((R.GT.0.031).AND. (R.LE.0.062)) IR(I)= 3 P1108070
IF ((R.GT.0.062).AND. (R.LE.0.093)) IR(I)= 3 P1108080
IF ((R.GT.0.093).AND. (R.LE.0.124)) IR(I)= 3 P1108090
IF ((R.GT.0.124).AND. (R.LE.0.155)) IR(I)= 3 P1108100
IF ((R.GT.0.155).AND. (R.LE.0.186)) IR(I)= 3 P1108110
IF ((R.GT.0.186).AND. (R.LE.0.217)) IR(I)= 3 P1108120
IF ((R.GT.0.217).AND. (R.LE.0.248)) IR(I)= 3 P1108130
IF ((R.GT.0.248).AND. (R.LE.0.279)) IR(I)= 1 P1108140
IF ((R.GT.0.279).AND. (R.LE.0.31)) IR(I)= 1 P1108150
IF ((R.GT.0.31).AND. (R.LE.0.341)) IR(I)= 1 P1108160
IF ((R.GT.0.341).AND. (R.LE.0.372)) IR(I)= 1 P1108170
IF ((R.GT.0.372).AND. (R.LE.0.403)) IR(I)= 1 P1108180
IF ((R.GT.0.403).AND. (R.LE.0.434)) IR(I)= 1 P1108190
IF ((R.GT.0.434).AND. (R.LE.0.465)) IR(I)= 1 P1108200
IF ((R.GT.0.465).AND. (R.LE.0.496)) IR(I)= 1 P1108210
IF ((R.GT.0.496).AND. (R.LE.0.527)) IR(I)= 1 P1108220
IF ((R.GT.0.527).AND. (R.LE.0.558)) IR(I)= 1 P1108230
IF ((R.GT.0.558).AND. (R.LE.0.589)) IR(I)= 1 P1108240
IF ((R.GT.0.589).AND. (R.LE.0.62)) IR(I)= 1 P1108250
IF ((R.GT.0.62).AND. (R.LE.0.651)) IR(I)= 1 P1108260
IF ((R.GT.0.651).AND. (R.LE.0.682)) IR(I)= 1 P1108270
IF ((R.GT.0.682).AND. (R.LE.0.713)) IR(I)= 1 P1108280
IF ((R.GT.0.713).AND. (R.LE.0.744)) IR(I)= 1 P1108290
IF ((R.GT.0.744).AND. (R.LE.0.775)) IR(I)= 3 P1108300
IF ((R.GT.0.775).AND. (R.LE.0.806)) IR(I)= 3 P1108310
IF ((R.GT.0.806).AND. (R.LE.0.837)) IR(I)= 3 P1108320
IF ((R.GT.0.837).AND. (R.LE.0.868)) IR(I)= 3 P1108330
IF ((R.GT.0.868).AND. (R.LE.0.899)) IR(I)= 3 P1108340
IF ((R.GT.0.899).AND. (R.LE.0.930)) IR(I)= 3 P1108350
IF ((R.GT.0.930).AND. (R.LE.0.961)) IR(I)= 3 P1108360
IF ((R.GT.0.961).AND. (R.LE.0.996)) IR(I)= 3 P1108370
IF (R.GT.0.996) IR(I)= 3 P1108380
3000 CONTINUE P1108390
RETURN P1108400
END P1108410
C***** P1108420
C SUBROUTINE: SIGNAL * P1108430
C * P1108440
C THIS SUBROUTINE FORMS A COMPLEX BASEBAND SIGNAL * P1108450
C***** P1108460
SUBROUTINE SIGNAL (DATA,NI,NQ) P1108470
COMPLEX DATA(1) P1108480
DIMENSION NI(1), NQ(1) P1108490
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL P1108500
COMMON /NUMB2/ NSNR,NSYMB,BAUD P1108510
DO 2 I=1,NSYMB P1108520
C LOAD INTO SAMPLE ARRAY , LSAMPLES PER SYMBOL P1108530
J1=(I-1)*LSAMPL+1 P1108540
J2=I*LSAMPL P1108550
DO 3 J3=J1,J2 P1108560
3 DATA(J3)=CMPLX(FLOAT(NI(I)),FLOAT(NQ(I))) P1108570
2 CONTINUE P1108580
RETURN P1108590
END P1108600

```

```

C***** 51200010
C*      * 51200020
C*      FILE NAME: 512CD      * 51200030
C*      * 51200040
C*      * 51200050
C*      * 51200060
C*      THIS PROGRAM EVALUATES THE PERFORMANCE OF 512-QAM      * 51200070
C*      SYSTEMS IN THE PRESENCE OF LINEAR CHANNEL DISTORTIONS      * 51200080
C*      * 51200090
C***** 51200100
C      COMPLEX DATA(524288),TF(524288)      51200110
C      DIMENSION NI(8192),NQ(8192),PE(36),EBNO(36)      51200120
C      DIMENSION XARRAY(36),YARRAY(36),AEB(36),APE(36)      51200130
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL      51200140
C      COMMON/NUMB2/ NSNR,NSYMB,BAUD      51200150
C      INITIALIZE PROGRAM      51200160
C      ALPHA=0.2      51200170
C      FBW=7.5      51200180
C      NSYMB=2048      51200190
C      LSAMPL=64      51200200
C      LDIM=LSAMPL*NSYMB      51200210
C      BAUD=15.      51200220
C      IOFF=0      51200230
C      NSNR=32      51200240
C      ISIN=1      51200250
C      NFILTR=2      51200260
C      NRUNS=1      51200270
C      ID=1 FOR AMPLITUDE DISTORTIONS      51200280
C      ID=2 FOR GD DISTORTIONS      51200290
C      ID=2      51200300
C      WRITE(6,666) ALPHA      51200310
C      666 FORMAT(10X,'ALPHA=',F8.4)      51200320
C      END OF INITIALIZATION      51200330
C      START OF COMPUTATIONS      51200340
C      DO 100 NR=1,NRUNS      51200350
C      CALL LD512 (DATA,NI,NQ)      51200360
C      CALL RCOSTX (TF)      51200370
C      CALL LAD(NR,TF)      51200380
C      CALL PAD(NR,TF)      51200390
C      CALL SAD(NR,TF,ISIN)      51200400
C      CALL LGD(NR,TF)      51200410
C      CALL PGD(NR,TF)      51200420
C      CALL SAGD(NR,TF,ISIN)      51200430
C      CALL LAD(NR,TF)      51200440
C      CALL PAD(NR,TF)      51200450
C      CALL SAD(NR,TF,ISIN)      51200460
C      CALL LGD(NR,TF)      51200470
C      CALL PGD(NR,TF)      51200480
C      CALL SAGD(NR,TF,ISIN)      51200490
C      CALL LAD(NR,TF)      51200500
C      CALL PAD(NR,TF)      51200510
C      CALL SAD(NR,TF,ISIN)      51200520
C      CALL LGD(NR,TF)      51200530
C      CALL PGD(NR,TF)      51200540
C      CALL SAGD(NR,TF,ISIN)      51200550
C      33 CALL FILTER (DATA,TF)      51200560
C      CALL ENERGY (DATA,EB)      51200570
C      END OF THE TRANSMITTER AND START OF THE RECEIVER      51200580
C      CALL RCOSRX(TF)      51200590
C      CALL HHGG (TF,PNOISE)      51200600
C      CALL FILTER (DATA,TF)      51200610
C      CALL SYNCRO (DATA,PNOISE,NI,NQ,MI,MQ,EB)      51200620
C      CALL DECODE (DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE ,EB)      51200630
C      CALL THE DRAWING SUBROUTINE      51200640
C      DO 99 I=1,NSNR      51200650
C      XARRAY(I)=EBNO(I)      51200660
C      99 YARRAY(I)=PE(I)      51200670
C      CALL DRAWCN (ID,XARRAY,YARRAY,NR,NRUNS)      51200680
C      CALL DRAWEB (XARRAY,YARRAY,NR,NRUNS)      51200690
C      CALL DRAWCN (ID,XARRAY,YARRAY,NR,NRUNS)      51200700
C      CALL DRAWEB (XARRAY,YARRAY,NR,NRUNS)      51200710
C      CALL DRAWCN (ID,XARRAY,YARRAY,NR,NRUNS)      51200720

```

```

C                                     51200730
C                                     51200740
C      CALL THE EYE DIAGRAM SUBROUTINE 51200750
C                                     51200760
C      CALL EYEQ (DATA)                51200770
100  CONTINUE                          51200780
      STOP                             51200790
      END                              51200800
C*****                              51200810
C      SUBROUTINE:  DLYD                * 51200820
C                                     * 51200830
C      THIS SUBROUTINE SIMULATES GROUP DELAY DISTORTIONS * 51200840
C*****                              51200850
      SUBROUTINE DLYD(NR,TF,ISIN)
      COMPLEX TF(1)
      GO TO (84,2,3,4,5,6,7),NR
      2 B=.095
      GO TO 10
      3 B=.19
      10 WRITE (6,11) B
      11 FORMAT (10X,'LINEAR GROUP DELAY IN NS/MHZ=',F8.5)
      CALL LINGD(B,TF)
      GO TO 84
      4 S=.27
      GO TO 20
      5 S=.30
      20 WRITE (6,12) S
      12 FORMAT (10X,'PARABOLIC GROUP DELAY IN NS/MHZ**2=',F8.5)
      CALL PARGD(S,TF)
      GO TO 84
      6 C= 6.0
      GO TO 30
      7 C=10.0
      30 K=4
      WRITE (6,17) K,C
      17 FORMAT (5X,'# OF RIPPLES=',I3,'SINE AMPLITUDE GROUP DELAY=',
      #F8.5,/)
      CALL SINCGD(ISIN,C,K,TF)
      84 RETURN
      END
C*****                              51201120
C      SUBROUTINE:  AMPD                * 51201130
C                                     * 51201140
C      THIS SUBROUTINE SIMULATES AMPLITUDE DISTORTIONS * 51201150
C*****                              51201160
      SUBROUTINE AMPD(NR,TF,ISIN)
      COMPLEX TF(1)
      GO TO (74,2,3,4,5,6,7),NR
      2 Z=.038
      GO TO 10
      3 Z=.055
      10 WRITE (6,8) Z
      8 FORMAT (10X,'LINEAR AMPLITUDE SLOPE IN DB/MGZ=',F8.5,/)
      CALL LINAMP(Z,TF)
      GO TO 74
      4 P=.00400
      GO TO 20
      5 P=.0063
      20 WRITE(6,9) P
      9 FORMAT(10X,'PARABOLIC AMPLITUDE DIST. IN DB/MHZ**2=',F8.5,/)
      CALL PARAMP(P,TF)
      GO TO 74
      6 D=.20
      GO TO 30
      7 D=.32
      30 K=4
      WRITE(6,77) K,D
      77 FORMAT(5X,'# OF RIPPLES=',I3,'SINE AMPLITUDE DIST. IN DB=',
      #F8.5,/)
      CALL SINAMP(ISIN,D,K,TF)
      74 RETURN
      END
      51201430
      51201440

```



```

C*****
C   SUBROUTINE: RCOSTX
C
C   THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C   RAISED COSINE FILTER WITH ARBITRARY ALPHA WITH A
C   X/SINX APERTURE EQUALIZER
C*****
C   SUBROUTINE RCOSTX(TF)
C     COMPLEX TF(1)
C     COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
C     COMMON /NUMB2/ NSNR,NSYMB,BAUD
C     SBANDW=BAUD*LSAMPL
C     NO=LDIM/2
C     NO1=NO+1
C     IF (ALPHA.EQ.0) ALPHA=0.0001
C     FN=LDIM*(FBW/SBANDW)
C     F1=(1.-ALPHA)*FN
C     F2=(1.+ALPHA)*FN
C     IFN=IFIX(FN)
C     IF1=IFIX(F1)+1
C     IF2=IFIX(F2)+1
C
C   THE AMPLITUDE CHARACTERISTICS
C
C     A1=3.1415926/(2.*FLOAT(IFN))
C     DO 8 I=2, IF1
C       TF(I)=CMPLX(1.0,0.0)
C       J=I-1
C
C   X/SIN(X) EQUALIZATION
C
C     A2=(FLOAT(J)*A1)/(SIN(FLOAT(J)*A1))
C     TF(I)=CMPLX(A2,0.0)
C   8 CONTINUE
C     JK=IF1+1
C     DO 9 J=JK,IF2
C       I=J-1
C
C   ROOT OF RAISED COSINE
C
C     A3=(FLOAT(I)*A1)/(SIN(FLOAT(I)*A1))
C     A=(3.1415926/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
C     TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
C     TF(J)=TF(J)*CMPLX(A3,0.0)
C   9 CONTINUE
C     JH=IF2+1
C     DO 10 I=JH,NO1
C       TF(I)=CMPLX(0.0,0.0)
C   10 CONTINUE
C     NO2=NO1+1
C     DO 5 I=NO2,LDIM
C       TF(I)=CONJG(TF(LDIM+2-I))
C   5 CONTINUE
C     RETURN
C     END
C*****
C   SUBROUTINE: RCOSRX
C
C   THIS SUBROUTINE SIMULATES THE TRANSFER FUNCTION OF A
C   RAISED COSINE FILTER WITH ARBITRARY ALPHA
C*****
C   SUBROUTINE RCOSRX(TF)
C     COMPLEX TF(1)
C     COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
C     COMMON /NUMB2/ NSNR,NSYMB,BAUD
C     SBANDW=BAUD*LSAMPL
C     NO=LDIM/2
C     NO1=NO+1
C     IF (ALPHA.EQ.0) ALPHA=0.0001
C     FN=LDIM*(FBW/SBANDW)
C     F1=(1.-ALPHA)*FN
C     F2=(1.+ALPHA)*FN

```

```

      IFN=IFIX(FN)
      IF1=IFIX(F1)+1
      IF2=IFIX(F2)+1
C
C      AMPLITUDE CHARACTERISTIC
C
      A1=3.141592/(2.*FLOAT(IFN))
      DO 8 I=2,IF1
      TF(1)=CMPLX(1.0,0.0)
      J=I-1
      TF(I)=CMPLX(1.,0.0)
8 CONTINUE
      JK=IF1+1
      DO 9 J=JK,IF2
      I=J-1
C
C      ROOT OF RAISED COSINE
C
      A=(3.141592/(2.0*ALPHA))*((FLOAT(I)/FLOAT(IFN))-1.)
      TF(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)
9 CONTINUE
      JH=IF2+1
      DO 10 I=JH,NO1
      TF(I)=CMPLX(0.0,0.0)
10 CONTINUE
      NO2=NO1+1
      DO 5 I=NO2,LDIM
      TF(I)=CONJG(TF(LDIM+2-I))
5 CONTINUE
      RETURN
      END
C*****
C      SUBROUTINE: FILTER
C
C      THIS SUBROUTINE PERFORMS THE FILTERING
C      PROCESS ON A DATA SEQUENCE
C*****
      SUBROUTINE FILTER(DATA,TF)
      COMPLEX DATA(1),TF(1)
      DIMENSION IWK(20)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      CALL FFT2C(DATA,19,IWK)
      DO 1 I=1,LDIM
      DATA(I)=CONJG(DATA(I)*TF(I))
      CALL FFT2C(DATA,19,IWK)
      DO 2 I=1,LDIM
      DATA(I)=CONJG(DATA(I))/FLOAT(LDIM)
      RETURN
      END
C*****
C      SUBROUTINE: HMGG
C
C      THIS SUBROUTINE COMPUTES THE EFFECTIVE NOISE POWER
C      AT THE OUTPUT OF THE RECEIVE FILTER
C*****
      SUBROUTINE HMGG(TF,PNOISE)
      COMPLEX TF(1)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      SUM=0.0
      SBANDW=BAUD*LSAMPL
      DO 1 L=1,LDIM
      HH=(CABS(TF(L)))**2
      SUM=SUM+HH
      PNOISE=SUM*SBANDW/FLOAT(LDIM)/2.
      WRITE (6,2) PNOISE
      FORMAT(5X,'PNOISE=',F7.3,/)
      RETURN
      END
C*****
C      SUBROUTINE: ENERGY
C

```

```

51202170
51202180
51202190
51202200
51202210
51202220
51202230
51202240
51202250
51202260
51202270
51202280
51202290
51202300
51202310
51202320
51202330
51202340
51202350
51202360
51202370
51202380
51202390
51202400
51202410
51202420
51202430
51202440
51202450
51202460
51202470
51202480
51202490
51202500
51202510
51202520
51202530
51202540
51202550
51202560
51202570
51202580
51202590
51202600
51202610
51202620
51202630
51202640
51202650
51202660
51202670
51202680
51202690
51202700
51202710
51202720
51202730
51202740
51202750
51202760
51202770
51202780
51202790
51202800
51202810
51202820
51202830
51202840
51202850
51202860
51202870
51202880

```

```

C      THIS SUBROUTINE COMPUTES THE CARRIER POWER          * 51202880
C      AND THE AVERAGE BIT ENERGY                        * 51202900
C*****                                                    51202910
      SUBROUTINE ENERGY(DATA,EB)                          51202920
      COMPLEX DATA(1)                                    51202930
      WATTS=0.                                             51202940
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                     51202950
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL           51202960
      DO 1 I=1,LDIM                                       51202970
      WATTS=WATTS+((CABS(DATA(I)))**2.)                   51202980
1      CONTINUE                                           51202990
      WATTS=WATTS/(FLOAT(LDIM))                           51203000
      EB=WATTS/(9.*BAUD)                                   51203010
      WRITE (6,40) WATTS,EB                               51203020
40     FORMAT(5X,'POWER=',F15.6,'BIT ENERGY=',E15.8)   51203030
      RETURN                                              51203040
      END                                                  51203050
C*****                                                    51203060
C      SUBROUTINE: SYNCRO                                  * 51203070
C                                                         * 51203080
C      THIS SUBROUTINE SYNCHRONIZES THE RECEIVED DATA    * 51203090
C*****                                                    51203100
      SUBROUTINE SYNCRO(DATA,PNOISE,NI,NQ,WI,MQ,EB)        51203110
      INTEGER Q7FLAG                                       51203120
      COMPLEX DATA(1),AMP                                  51203130
      DIMENSION NI(1),NQ(1)                                51203140
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL           51203150
      COMMON /NUMB2/ NSNR,NSYMB,BAUD                     51203160
      NERROR=0                                             51203170
      IF (IOFF.EQ.0) GO TO 111                             51203180
      DO 6 K=1,IOFF                                         51203190
      XX=AIMAG(DATA(LDIM))                                  51203200
      DO 7 KK=2,LDIM                                        51203210
7      DATA(KK)=CMPLX(REAL(DATA(KK)),AIMAG(DATA(KK-1)))  51203220
6      DATA(1)=CMPLX(REAL(DATA(1)),XX)                   51203230
C                                                         51203240
C      SYNCHRONIZE THE RECEIVED DATA                     51203250
C                                                         51203260
111     NOLD=0                                              51203270
      NOF=0                                                 51203280
      K=1                                                    51203290
300     CONTINUE                                           51203300
      NEW=0                                                 51203310
      DO 200 J=1,NSYMB                                       51203320
      J1=K+(J-1)*LSAMPL                                     51203330
      IF(J1.GT.LDIM) J1=J1-LDIM                             51203340
      AXBAR=REAL(DATA(J1))                                   51203350
      AYBAR=AIMAG(DATA(J1))                                 51203360
      SS=AXBAR*NI(J)                                         51203370
      IF (SS.GT. 0.) NEW=NEW+1                               51203380
      SS=AYBAR*NQ(J)                                         51203390
      IF (SS.GT. 0.) NEW=NEW+1                               51203400
200     CONTINUE                                           51203410
      IF(NOLD.GE.NEW) GO TO 399                             51203420
      NOLD=NEW                                              51203430
      NOF=K                                                 51203440
399     K=K+1                                               51203450
      LFU=2*NSYMB                                           51203460
C                                                         51203470
C      SHIFT RECEIVED DATA LEFT UNTIL ALL SYMBOLS ARE LINED UP 51203480
C                                                         51203490
      NSFT=LSAMPL*3                                         51203500
      IF (NOLD.LT.LFU.AND.K.LE.NSFT) GO TO 300             51203510
      NOF=NOF-1                                             51203520
      WRITE(6,56) NOLD,K                                    51203530
56     FORMAT(1X,'NO OF CORRECT SYMBOLS = ',I6,5X,'NO OF SHIFT IN SAMPLES', 51203540
      *',I6,/)                                              51203550
      IF (NOF.EQ.0) GO TO 230                               51203560
      LO=LDIM-1                                             51203570
      DO 250 I=1,NOF                                       51203580
      AMP=DATA(1)                                           51203590
      DO 240 J=1,LO                                         51203600

```

```

DATA(J)=DATA(J+1)
240 CONTINUE
DATA(LDIM)=AMP
250 CONTINUE
C
C      OPTIMIZE THE SAMPLING INSTANT
C
230 MI=1
    MQ=1
    EOI=FLOAT(NSYMB) + 1.
    EQ=FLOAT(NSYMB) + 1.
    M=25
    VARIAN=PNOISE*EB*(10.**(-0.1*FLOAT(M)))
    SIGMA=SQRT(VARIAN)
    DO 80 J=1,LSAMPL
        EI=0.
        EQ=0.
        DO 70 K=1,NSYMB
C
C      VERIFY THAT NO SAMPLED SYMBOL IS IN ERROR
C
            J1=(K-1)*LSAMPL+J
            AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
            AYBAR=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.
            INDEXI=0
            SS=FLOAT(NI(K))*AXBAR
            IF (SS .LT. 0.) INDEXI=1
            AMPX=ABS(AXBAR)
            AMPI=ABS(FLOAT(NI(K)))
            THR1I=AMPI - 1.0
            THR2I=AMPI + 1.0
            I7FLAG=0
            IF (AMPI.EQ. 23.) GO TO 30
            IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
            GO TO 40
30 IF (AMPX.LE.THR1I) INDEXI=1
            I7FLAG=1
40 CONTINUE
            INDEXQ=0
            SS=FLOAT(NQ(K))*AYBAR
            IF (SS .LT. 0.) INDEXQ=1
            AMPY=ABS(AYBAR)
            AMPQ=ABS(FLOAT(NQ(K)))
            THR1Q=AMPQ - 1.0
            THR2Q=AMPQ + 1.0
            Q7FLAG=0
            IF (AMPQ.EQ. 23.) GO TO 50
            IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
            GO TO 60
50 IF (AMPY.LE.THR1Q) INDEXQ=1
            Q7FLAG=1
60 CONTINUE
            IF (INDEXI.EQ.1) EI=EI + 1.
            IF (INDEXQ.EQ.1) EQ=EQ + 1.
C
C      COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL AT EB/NO=22dB
C
            D1=ABS(AMPX-THR1I)
            ARG=D1/(SIGMA*SQRT(2.))
C
C      CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C
            IF (ARG.GT.12.) ARG=12.
            EI=EI +ERFC(ARG)/2.
            IF (I7FLAG .EQ. 1) GO TO 65
            D2=ABS(THR2I-AMPX)
            ARG=D2/(SIGMA*SQRT(2.))
            IF (ARG.GT.12.) ARG=12.
            EI=EI + ERFC(ARG)/2.
65 D1=ABS(AMPY-THR1Q)
            ARG=D1/(SIGMA*SQRT(2.))
            IF (ARG.GT.12.) ARG=12.

```

51203610
51203620
51203630
51203640
51203650
51203660
51203670
51203680
51203690
51203700
51203710
51203720
51203730
51203740
51203750
51203760
51203770
51203780
51203790
51203800
51203810
51203820
51203830
51203840
51203850
51203860
51203870
51203880
51203890
51203900
51203910
51203920
51203930
51203940
51203950
51203960
51203970
51203980
51203990
51204000
51204010
51204020
51204030
51204040
51204050
51204060
51204070
51204080
51204090
51204100
51204110
51204120
51204130
51204140
51204150
51204160
51204170
51204180
51204190
51204200
51204210
51204220
51204230
51204240
51204250
51204260
51204270
51204280
51204290
51204300
51204310
51204320

```

EQ=EQ + ERFC(ARG)/2.
IF (Q7FLAG.EQ. 1) GO TO 70
D2=ABS(THR2Q-AMPY)
ARG=D2/(SIGMA*SQRT(2.))
IF (ARG.GT.12.) ARG=12.
EQ=EQ + ERFC(ARG)/2.
70 CONTINUE
IF (EOI.LE.EI) GO TO 75
EOI=EI
MI=J
75 CONTINUE
IF (EQQ.LE.EQ) GO TO 80
EQQ=EQ
MQ=J
80 MOFF=IABS(MI-MQ)
IF (MOFF.NE.0) WRITE(6,90) MOFF
OFF=(FLOAT(MI) + FLOAT(NQ))/FLOAT(LSAMPL)
90 FORMAT(5X,'SAMPLING POINTS FOR I AND Q CHANNELS DIFFER BY',
#12,' SIXTEENTHS OF THE SYMBOL INTERVAL',/)
WRITE(6,95) OFF
95 FORMAT(5X,'RECEIVED DATA IS DELAYED BY ',F7.3,' SYMBOLS',/)
RETURN
END
C*****
C SUBROUTINE: DECODE *
C *
C THIS SUBROUTINE DECODES THE SYNCHRONIZED DATA *
C*****
SUBROUTINE DECODE(DATA,PNOISE,MI,MQ,NI,NQ,EBNO,PE,EB)
INTEGER Q7FLAG
COMPLEX DATA(1),AMP
DIMENSION EBNO(1),PE(1),NI(1),NQ(1)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
NERROR=0
NNSNR=NSNR+3
DO 1 I=1,NNSNR
1 PE(I)=0.
C
C CHECK FOR AN ERROR AT THE SAMPLING INSTANT, ONLY THOSE
C ERRORS IN ADJACENT LEVELS ARE CONSIDERED
C
DO 2 K=1,NSYMB
J1=(K-1)*LSAMPL +MI
J2=(K-1)*LSAMPL +MQ
AXBAR=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.
AYBAR=(AIMAG(DATA(J2))+AIMAG(DATA(J2+1)))/2.
INDEXI=0
SS=FLOAT(NI(K))*AXBAR
IF (SS.LT. 0.) INDEXI=1
AMPX=ABS(AXBAR)
AMPI=ABS(FLOAT(NI(K)))
THR1I=AMPI - 1.0
THR2I=AMPI + 1.0
I7FLAG=0
IF (AMPI.EQ. 23.) GO TO 30
IF (( AMPX.GE.THR2I).OR.(AMPX.LE.THR1I)) INDEXI=1
GO TO 40
30 IF (AMPX.LE.THR1I) INDEXI=1
I7FLAG=1
40 CONTINUE
INDEXQ=0
SS=FLOAT(NQ(K))*AYBAR
IF (SS.LT. 0.) INDEXQ=1
AMPY=ABS(AYBAR)
AMPQ=ABS(FLOAT(NQ(K)))
THR1Q=AMPQ - 1.0
THR2Q=AMPQ + 1.0
Q7FLAG=0
IF (AMPQ.EQ. 23.) GO TO 50
IF (( AMPY.GE.THR2Q).OR.(AMPY.LE.THR1Q)) INDEXQ=1
GO TO 60

```

```

50 IF(AMPY.LE.THR1Q) INDEXQ=1
   Q7FLAG=1
60 CONTINUE
   IF((INDEXI.EQ.1).OR.(INDEXQ.EQ.1)) NERROR=NERROR+1
C
C   COMPUTE THE PROBABILITY OF ERROR FOR THIS SYMBOL.
C   THE VARIABLES ARE AS FOLLOWS: M IS EB/NO IN dB; PNOISE IS
C   1/2*INTERGAL (0 H(F)**2" DF); AND VARIAN = NO*PNOISE.
C   FOR THE END POINTS (7FLAG=1) THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2)))
C   AND FOR THE INNER POINTS THE PE=1/2*ERFC(D1/(SIGMA*SQRT(2))) +
C   1/2*ERFC(D2/(SIGMA*SQRT(2))).
C
   DO 4 M=4,NSNR
     VARIAN=PNOISE*ES*(10.**(-0.1*FLOAT(M)))
     SIGMA=SQRT(VARIAN)
57 D1=ABS(AMPX-THR1I)
     ARG=D1/(SIGMA*SQRT(2.))
C
C   CHECK IF ARG IS LARGE INWHICH CASE PE IS INSIGNIFICANT
C   CHECK IF SYMBOL IS IN ERROR. IF IT IS TERMINATE RUN
C
     IF (INDEXI.EQ.1) GO TO 153
     IF (ARG.GT.12.) ARG=12.
     PEI=ERFC(ARG)/2.
     IF (I7FLAG.EQ.1) GO TO 100
     D2=ABS(THR2I-AMPX)
     ARG=D2/(SIGMA*SQRT(2.))
     IF (ARG.GT.12.) ARG=12.
     PEI=PEI + ERFC(ARG)/2.
100 D1=ABS(AMPY-THR1Q)
     ARG=D1/(SIGMA*SQRT(2.))
     IF (INDEXQ.EQ.1) GO TO 153
     IF (ARG.GT.12.) ARG=12.
     PEQ=ERFC(ARG)/2.
     IF (Q7FLAG.EQ.1) GO TO 110
     D2=ABS(THR2Q-AMPY)
     ARG=D2/(SIGMA*SQRT(2.))
     IF (ARG.GT.12.) ARG=12.
     PEQ=PEQ + ERFC(ARG)/2.
110 IF(PEI.LT.1.E-15) PEI=0.
     IF(PEQ.LT.1.E-15) PEQ=0.
     4 PE(M)=PE(M)+(PEI+PEQ)/2.
     2 CONTINUE
     DO 5 I=1,NSNR
       PE(I)=PE(I+3)/FLOAT(NSYMB)
     5 EBNO(I)=FLOAT(I+3)
     PRINT 150
150 FORMAT(5X,'EB/NO',10X,'PROB. OF ERROR',/)
     WRITE (6,151) (EBNO(I),PE(I),I=1,NSNR)
151 FORMAT(5X,F5.1,10X,E13.6)
     WRITE (6,152) NERROR
152 FORMAT(/5X,'ERRORS=',15,/)
     GO TO 155
153 PRINT 154
154 FORMAT (10X,'SYMBOL WAS IN ERROR, RUN WAS TERMINATED',/)
155 CONTINUE
     RETURN
     END
C*****
C   SUBROUTINE: DRAWEB
C
C   THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR
C   P(E) AS LOW AS 1.0E-8 AND A C/N RATIO AS HIGH AS 33 DB
C*****
C   SUBROUTINE DRAWEB(XARRAY,YARRAY,JCURV,JLAST)
C   DIMENSION XARRAY( 1),YARRAY( 1),X(22),Y(22)
C   COMMON /NUMB2/ NSNR,NSYMB,BAUD
C   M=NSNR
C   DO 1 K=1,NSNR
     IF(YARRAY(K).LE.1.0E-8) GOTO 3
     GOTO 1
3   M=K-1

```

```

      GOTO 4                                51205770
1     CONTINUE                             51205780
4     CONTINUE                             51205790
      I=M+1                                51205800
      II=I+1                               51205810
      XARRAY(I)=0.0                        51205820
      XARRAY(II)=2.0                       51205830
      YARRAY(I)=1.E-8                      51205840
      YARRAY(II)=0.4                       51205850
      IF(JCURV.GT.1) GOTO 2                51205860
C                                           51205870
C     ESTABLISH THE SURFACE AREA.           51205880
C                                           51205890
      CALL PLOTS(31.0,27.5)                51205900
      CALL FACTOR(.7)                      51205910
C                                           51205920
C     ESTABLISH THE ORIGIN.                51205930
C                                           51205940
      CALL PLOT(2.0,1.5,-3)                51205950
C                                           51205960
C                                           51205970
      CALL SYMBOL(6.,23.,0.35,32HBIT ERROR PROBABILITY OF 512-QAM,0., 51205980
&32)                                       51205990
C                                           51206000
C                                           51206010
C     DRAW THE LOGARITHMIC Y-AXIS.           51206020
C                                           51206030
      CALL LGAXS(0.0,0.0,20HPROBABILITY OF ERROR,20,20.5, 51206040
+90.,1.0E-8,.4)                          51206050
C                                           51206060
C     DRAW THE LINEAR X-AXIS.               51206070
C                                           51206080
      CALL AXIS(0.0,0.0,11HEB/NO IN dB,-11, 51206090
+20.0,0.0,0.0,2.0)                       51206100
2     CONTINUE                             51206110
C                                           51206120
C     PLOT THE DATA IN LOG-LINEAR MODE.     51206130
C                                           51206140
      CALL LGLIN(XARRAY,YARRAY,M,1,2,JCURV,1) 51206150
      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999) 51206160
      RETURN                                51206170
      END                                    51206180
C*****                                     51206190
C     SUBROUTINE: DRAWCN                     * 51206200
C                                           * 51206210
C     THIS SUBROUTINE DRAWS PROBABILITY OF ERROR CURVES FOR * 51206220
C     P(E) AS LOW AS 1.0E-8 AND A C/N RATIO AS HIGH AS 35 DB * 51206230
C                                           * 51206240
C     ID = 1 FOR AMPLITUDE DISTORTIONS        * 51206250
C     ID = 2 FOR GROUP DELAY DISTORTIONS      * 51206260
C*****                                     51206270
      SUBROUTINE DRAWCN(ID,XARRAY,YARRAY,JCURV,JLAST) 51206280
      DIMENSION XARRAY(34),YARRAY(34),X(29),Y(29) 51206290
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL 51206300
      COMMON /NUMB2/ NSNR,NSYMB,BAUD 51206310
      M=NSNR 51206320
      CF=9. 51206330
      DO 1 K=1,NSNR 51206340
      XARRAY(K)=XARRAY(K) + 10*(ALOG10(CF)) 51206350
      IF(YARRAY(K).GE.1.0) YARRAY(K)=1.0 51206360
      IF(YARRAY(K).LE.1.0E-8) GOTO 3 51206370
      GOTO 1 51206380
3     M=M-1 51206390
      GOTO 4 51206400
1     CONTINUE 51206410
4     CONTINUE 51206420
      I=M+1 51206430
      II=I+1 51206440
      XARRAY(I)=12.0 51206450
      XARRAY(II)=2.0 51206460
      YARRAY(I)=1.E-8 51206470
      YARRAY(II)=0.4 51206480

```

```

      IF(JCURV.GT.1) GOTO 2
C
C ESTABLISH THE SURFACE AREA.
C
      CALL PLOTS(30.0,27.5)
      CALL FACTOR(.8)
C
C ESTABLISH THE ORIGIN.
C
      CALL PLOT(3.0,1.5,-3)
C
C DRAW THE LOGARITHMIC Y-AXIS.
C
      CALL LGAXS(0.0,0.0,29HPROBABILITY OF A SYMBOL ERROR,29,20.,
+90.,1.0E-8,.4)
C
C DRAW THE LINEAR X-AXIS.
C
      CALL AXIS(0.0,0.0,9HC/N IN dB,-9,
+18.0,0.0,12.0,2.0)
C
C WRITE THE TITLE OF THE GRAPH.
C
      CALL SYMBOL(6.,23.,.35,'PARABOLIC AMPLITUDE DISTORTION',0.,30)
C
C PLOT LEGEND
C*****
C      GO TO (11,22), ID
C 22 CALL SYMBOL(2.,13.,.35,'NO DISTORTION',0.,13)
C      CALL SYMBOL(2.,12.,.35,'L = 0.095 NS/MHZ',0.,16)
C      CALL SYMBOL(2.,11.,.35,'L = 0.190 NS/MHZ',0.,16)
C      CALL SYMBOL(2.,10.,.35,'P = 0.270 NS/MHZ**2',0.,19)
C      CALL SYMBOL(2.,9.,.35,'P = 0.540 NS/MHZ**2',0.,19)
C      CALL SYMBOL(2.,8.,.35,'S = 6.00 NS',0.,12)
C      CALL SYMBOL(2.,7.,.35,'S = 12.0 NS',0.,12)
C      CALL SYMBOL(2.,4.5,.35,22HGROUP DELAY DISTORTION,0.,22)
C      GO TO 2
C*****
C 11 CALL SYMBOL(2.,13.,.35,'NO DISTORTION',0.,13)
C      CALL SYMBOL(2.,12.,.35,'L = 0.038 DB/MHZ',0.,16)
C      CALL SYMBOL(2.,11.,.35,'L = 0.076 DB/MHZ',0.,16)
C      CALL SYMBOL(2.,10.,.35,'P = 0.004 DB/MHZ**2',0.,19)
C      CALL SYMBOL(2.,9.,.35,'P = 0.008 DB/MHZ**2',0.,19)
C      CALL SYMBOL(2.,8.,.35,'S = 0.2 DB',0.,12)
C      CALL SYMBOL(2.,7.,.35,'S = 0.4 DB',0.,12)
C      CALL SYMBOL(2.,4.5,.35,20HAMPLITUDE DISTORTION,0.,20)
C 2 CONTINUE
C      YPAGE=6.5 - (JCURV-1)
C      YPGE=YPAGE-.1
C      CALL SYMBOL(2.,YPAGE,.2,JCURV,0.,-1)
C      CALL SYMBOL(2.5,YPGE,.28,2Hz=,0.,2)
C      CALL SYMBOL(5.0,YPAGE,.28,'IN dB/MHz**2',0.,12)
C
C PLOT THE DATA IN LOG-LINEAR MODE.
C
      CALL LGLIN(XARRAY,YARRAY,M,1,0.0 ,1)
      IF(JCURV.EQ.JLAST) CALL PLOT(0.0,0.0,999)
      RETURN
      END
C*****
C SUBROUTINE: EYEQ
C
C THIS SUBROUTINE DRAWS THE EYE DIAGRAM
C FOR A TWO SYMBOL DURATION
C*****
C SUBROUTINE EYEQ(DATA)
C DIMENSION DATA2(66),DATA3(66)
C COMPLEX DATA(1)
C COMMON /NUMB1/ FBW,ALPHA,LDIM,ICFF,LSAMPL
C COMMON /NUMB2/ NSNR,NSYMB,BAUD
C ESTABLISH THE SURFACE AREA.
C CALL PLOTS(35.0,27.5)

```

51206490
 51206500
 51206510
 51206520
 51206530
 51206540
 51206550
 51206560
 51206570
 51206580
 51206590
 51206600
 51206610
 51206620
 51206630
 51206640
 51206650
 51206660
 51206670
 51206680
 51206690
 51206700
 51206710
 51206720
 51206730
 51206740
 51206750
 51206760
 51206770
 51206780
 51206790
 51206800
 51206810
 51206820
 51206830
 51206840
 51206850
 51206860
 51206870
 51206880
 51206890
 51206900
 51206910
 51206920
 51206930
 51206940
 51206950
 51206960
 51206970
 51206980
 51206990
 51207000
 51207010
 51207020
 51207030
 51207040
 51207050
 51207060
 51207070
 51207080
 51207090
 51207100
 51207110
 51207120
 51207130
 51207140
 51207150
 51207160
 51207170
 51207180
 51207190
 51207200


```

C      ESTABLISH THE ORIGIN.                                51207210
C      CALL PLOT(3.0,13.0,-3)                                51207220
C      WRITE THE TITLE OF THE GRAPH.                         51207230
C      CALL SYMBOL(10.0,12.0,0.49,16HQPSK EYE DIAGRAM,0.0,16) 51207240
C      CALL SCALE (DATA3(1),31.0,16,1)                      51207250
C      CALL SCALE (DATA2(1),26.0,16,1)                      51207260
C      DRAW THE TIME AXIS.                                    51207270
C      CALL AXIS(0.0,0.0,1H,-1,31.,0.0,1.0,2.0)             51207280
C      DRAW THE AMPLITUDE AXIS.                               51207290
C      CALL AXIS(0.0,-13.0,1H,1,27.0,90.0,-25.0,2.5)        51207300
C      PLOT THE DATA.                                        51207310
C      JJ=2*LSAMPL                                           51207320
C      DATA2(JJ+1)=0.0                                       51207330
C      DATA2(JJ+2)=2.5                                       51207340
C      DATA3(JJ+1)=1.0                                       51207350
C      DATA3(JJ+2)=2.0                                       51207360
C      M2=NSYMB/2                                             51207370
C      DO 4 KK=1,M2                                           51207380
C      DO 2 I=1,JJ                                           51207390
C      II=(KK-1)*2*LSAMPL+I                                   51207400
C      DATA2(I)=REAL(DATA(II))                               51207410
C      DATA3(I)=FLOAT(I)                                     51207420
C      CALL LINE(DATA3,DATA2,JJ,1,0,0)                       51207430
C      4 CONTINUE                                             51207440
C      CALL PLOT(0.0,0.0,999)                                 51207450
C      RETURN                                                 51207460
C      END                                                    51207470
C*****                                                    51207480
C      SUBROUTINE: PARGD                                     * 51207490
C                                                           * 51207500
C      THIS SUBROUTINE SIMULATES PARABOLIC GROUP DELAY DISTORTION * 51207510
C      IN NANOSECONDS = S*F*F                               * 51207520
C*****                                                    51207530
C      SUBROUTINE PARGD(S,TF)                                51207540
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL             51207550
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD                       51207560
C      COMPLEX TF(1)                                         51207570
C      SBANDW=FLOAT(LSAMPL)*BAUD                             51207580
C      NO1=LDIM/2 + 1                                       51207590
C      NO2=NO1 + 1                                           51207600
C      DO 20 I=1,NO1                                         51207610
C      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                     51207620
C      PHI=2.*3.141592/3.*((S*(AI**3.))/(10.**3.))          51207630
C      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                51207640
C      20 CONTINUE                                           51207650
C      DO 21 I=NO2,LDIM                                     51207660
C      TF(I)=CONJG(TF(LDIM+2-I))                             51207670
C      21 CONTINUE                                           51207680
C      RETURN                                                 51207690
C      END                                                    51207700
C*****                                                    51207710
C      SUBROUTINE: SINCGD                                    * 51207720
C                                                           * 51207730
C      THIS SUBROUTINE SIMULATES SINE OR COSINE GROUP DELAY * 51207740
C      DISTORTION WITH ARBITRARY AMPLITUDE AND NUMBER OF RIPPLES * 51207750
C      GROUP DELAY IN NANOSECONDS = C*COS(2*PIE*K*F/FNYQ + ISIN*PIE/2) * 51207760
C*****                                                    51207770
C      SUBROUTINE SINCGD(ISIN,C,K,TF)                       51207780
C      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL             51207790
C      COMMON /NUMB2/ NSNR,NSYMB,BAUD                       51207800
C      COMPLEX TF(1)                                         51207810
C      SBANDW=FLOAT(LSAMPL)*BAUD                             51207820
C      NO1=LDIM/2 + 1                                       51207830
C      NO2=NO1 + 1                                           51207840
C      C1=BAUD/K                                             51207850
C      IF (ISIN.EQ.1) GO TO 50                               51207860
C      DO 10 I=1,NO1                                         51207870
C      AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)                     51207880
C      PHI=(C*C1/(10.**3.))*SIN(2.*3.141592*AI/C1)          51207890
C      TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))                51207900
C      10 CONTINUE                                           51207910
C      50 CONTINUE                                           51207920

```

```

DO 11 I=NO2,LDIM
TF(I)=CONJG(TF(LDIM+2-I))
11 CONTINUE
GO TO 60
50 DO 12 I=1,NO1
AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)
PHI=(C*C1/(10.**3.))*COS(2.*3.141592*AI/C1)
TF(I)=TF(I)*CMPLX(COS(PHI),SIN(PHI))
12 CONTINUE
DO 13 I=NO2,LDIM
TF(I)=TF(LDIM+2-I)
13 CONTINUE
60 RETURN
END
C*****
C SUBROUTINE: LINGCD *
C *
C THIS SUBROUTINE SIMULATES LINEAR GROUP DELAY *
C IN NANoseconds = B*F *
C*****
SUBROUTINE LINGD(B,TF)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
COMPLEX TF(1)
SBANDW=FLOAT(LSAMPL)*BAUD
NO1=LDIM/2 + 1
NO2=NO1 + 1
DO 10 I=1,NO1
AI=FLOAT(I-1)*SBANDW/FLOAT(LDIM)
PHI=3.141592*(B/(10.**3.))*(AI**2.)
TF(I)=TF(I)*CMPLX(COS(PHI),-SIN(PHI))
10 CONTINUE
DO 20 I=NO2,LDIM
TF(I)=TF(LDIM+2-I)
20 CONTINUE
RETURN
END
C*****
C SUBROUTINE: LINAMP *
C *
C THIS SUBROUTINE SIMULATES LINEAR AMPLITUDE DISTORTION *
C*****
SUBROUTINE LINAMP(Z,TF)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
COMPLEX TF(1)
SBANDW=FLOAT(LSAMPL)*BAUD
NO1=LDIM/2 + 1
NO2=NO1 + 1
AI=Z*SBANDW/FLOAT(LDIM)
DO 10 I=1,NO1
AJ=(I-1)*AI/20.
TF(I)=TF(I)*(10.**AJ)
10 CONTINUE
DO 20 I=NO2,LDIM
AZ=(LDIM+2-I)*AI/20.
TF(I)=TF(I)/(10.**AZ)
20 CONTINUE
RETURN
END
C*****
C SUBROUTINE: PARAMP *
C *
C THIS SUBROUTINE SIMULATES PARABOLIC AMPLITUDE DISTORTION *
C*****
SUBROUTINE PARAMP(P,TF)
COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
COMMON /NUMB2/ NSNR,NSYMB,BAUD
COMPLEX TF(1)
SBANDW=FLOAT(LSAMPL)*BAUD
NO1=LDIM/2 + 1
NO2=NO1 + 1

```

```

      AI=P*((SBANDW/FLOAT(LDIM))**2.)
      DO 10 I=1,NO1
      AJ=((I-1)**2.)*AI/20.
      TF(I)=TF(I)*(10.**AJ)
10    CONTINUE
      DO 11 I=NO2,LDIM
      TF(I)=TF(LDIM+2-I)
11    CONTINUE
      RETURN
      END
C*****
C    SUBROUTINE:  SINAMP
C
C    THIS SUBROUTINE SIMULATES SINUSOIDAL AMPLITUDE DISTORTION
C*****
      SUBROUTINE SINAMP(ISIN,D,K,TF)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      COMMON /NUMB2/ NSNR,NSYMB,BAUD
      COMPLEX TF(1)
      SBANDW=FLOAT(LSAMPL)*BAUD
      NO1=LDIM/2 + 1
      NO2=NO1 + 1
      AJ=SBANDW/FLOAT(LDIM)
      C1=BAUD/K
      IF(ISIN.EQ.1) GO TO 50
      DO 10 I=1,NO1
      AZ=D*COS(2.*3.141592*AJ*(FLOAT(I)-1.)/C1)/20.
      TF(I)=TF(I)*(10.**AZ)
10    CONTINUE
      DO 20 I=NO2,LDIM
      TF(I)=TF(LDIM+2-I)
20    CONTINUE
      RETURN
50    DO 30 I=1,NO1
      AZ=D*SIN(2.*3.141592*AJ*(FLOAT(I)-1.)/C1)/20.
      TF(I)=TF(I)*(10.**AZ)
30    CONTINUE
      DO 40 I=NO2,LDIM
      AZ=D*SIN(2.*3.141592*AJ*(FLOAT(LDIM+2-I)-1.)/C1)/20.
      TF(I)=TF(I)/(10.**AZ)
40    CONTINUE
      RETURN
      END
C*****
C    SUBROUTINE:  LD512
C
C    THIS SUBROUTINE FORMS A BASEBAND 512-QAM SIGNAL
C*****
      SUBROUTINE LD512 (DATA,NI,NQ)
      COMPLEX DATA(1),A(128)
      DIMENSION NX(8192),NI(1),NQ(1),NY(8192)
      COMMON /NUMB1/ FBW,ALPHA,LDIM,IOFF,LSAMPL
      DATA NX(1),NX(2),NX(3),NX(4),NX(5),NX(6),NX(7),NX(8),NX(9),NX(10)
      &,NX(11),NX(12),NX(13) /-1,1,1,-1,-1,1,1,-1,-1,1,-1,1,1/
      DATA JLAST /13/
      KKK=2**JLAST
      KK=KKK-(1+JLAST)
C
C
C    GENERATE (2**JLAST-1) LENGTH SEQUENCE USING GIVEN DATA
C    AND GENERATOR POLYNOMIAL
C
      DO 1 J=1,KK
      I=J+JLAST
      NX(I)= NX(J)*NX(J+9)*NX(J+10)*NX(J+12)
      NX(I)=0-NX(I)
1    CONTINUE
C
C    CHANGE THE NUMBER OF BITS TO 8190 (I.E.:910 SYMBOLS TIMES 9 BITS)
C
      MM=KKK
      MM=FLOAT (KKK/ 4)

```

```

C      ARANGE THE 8190 BITS INTO SYMBOLS OF 9 BITS EACH AND MAP EACH
C      SYMBOL INTO 1 STATE OF THE 512 QAM SYSTEM
C
      K=0
      DO 3 J=1,8
      DO 20 I=1,12
      K=K+1
      M1=2*J-1
      N1=2*I-1
      A(K)=CMPLX(FLOAT(M1),FLOAT(N1))
20 CONTINUE
      3 CONTINUE
      M=96
      DO 4 J=9,12
      DO 5 I=1,8
      M=M+1
      A(M)=CMPLX(FLOAT(2*J-1),FLOAT(2*I-1))
      5 CONTINUE
      04 CONTINUE
C
C      CHANGE THE VALUES OF -1 TO 0 FOR THE DATA OF NX( )
C
      DO 12 I=1,KKK
      NY(I)=NX(I)
      IF (NY(I).EQ. -1) NY(I)=0
12 CONTINUE
      DO 10 J=1,MM
      NI(J)=0
      NQ(J)=0
10 CONTINUE
C
C      GET THE DECADE NUMBER FOR THE MATRIX A( )
C
      DO 11 I=1,MM
      L=NY(I)+NY(I+1)*2+NY(I+2)*4+NY(I+3)*8+NY(I+4)*16+NY(I+5)*32+NY(I+
      &6)*64
      IF (L.EQ.0) L=1
      NI(I)=IFIX(REAL(A(L)))
      NQ(I)=IFIX(AIMAG(A(L)))
      IF (NY(I+7).EQ.0) NI(I)=-NI(I)
      IF (NY(I+8).EQ.0) NQ(I)=-NQ(I)
11 CONTINUE
C
C      LOAD INTO SAMPLE ARRAY "LSAMPL" SAMPLES PER SYMBOL
C
      DO 15 I=1,MM
      J1=(I-1)*LSAMPL+1
      J2=I*LSAMPL
      DO 15 J3=J1,J2
15 DATA(J3)= CMPLX(FLOAT(NI(I)),FLOAT(NQ(I)))
      RETURN
      END
C*****
C      SUBROUTINE: LAD
C
C      THIS SUBROUTINE SIMULATES LINEAR AMPLITUDE DISTORTION
C*****
      SUBROUTINE LAD(NR,TF)
      COMPLEX TF(1)
      GO TO (74,1,2,3,4,5),NR
1      Z=.01
      GO TO 22
2      Z=.02
      GO TO 22
3      Z=.03
      GO TO 22
4      Z=.04
      GO TO 22
5      Z=.05
22 WRITE (6,11) Z
11 FORMAT (10X,'LINEAR AMPLITUDE SLOPE IN DB/MHZ=',F8.5/)
      CALL LINAMP(Z,TF)

```

```

74 RETURN 51210090
END 51210100
C***** 51210110
C SUBROUTINE: PAD * 51210120
C * 51210130
C THIS SUBROUTINE SIMULATES PARABOLIC AMPLITUDE DISTORTION * 51210140
C***** 51210150
SUBROUTINE PAD(NR,TF) 51210160
COMPLEX TF(1) 51210170
GO TO (74,1,2,3,4,5), NR 51210180
1 P=.002 51210190
GO TO 22 51210200
2 P=.003 51210210
GO TO 22 51210220
3 P=.004 51210230
GO TO 22 51210240
4 P=.005 51210250
GO TO 22 51210260
5 P=.006 51210270
22 WRITE(6,11) P 51210280
11 FORMAT(5X,'PARABOLIC DISTORTION IN DB/MHZ**2=',F8.5/) 51210290
CALL PARAMP(P,TF) 51210300
74 RETURN 51210310
END 51210320
C***** 51210330
C SUBROUTINE: SAD * 51210340
C * 51210350
C THIS SUBROUTINE SIMULATES SINUSOIDAL AMPLITUDE DISTORTION * 51210360
C***** 51210370
SUBROUTINE SAD(NR,TF,ISIN) 51210380
COMPLEX TF(1) 51210390
GO TO (74,1,2,3,4,5), NR 51210400
1 D=.1 51210410
GO TO 22 51210420
2 D=.15 51210430
GO TO 22 51210440
3 D=.2 51210450
GO TO 22 51210460
4 D=.25 51210470
GO TO 22 51210480
5 D=.3 51210490
22 K=4 51210500
WRITE(6,77) K,D 51210510
77 FORMAT(5X,'# OF RIPPLES=',I3,'SINE AMPL. DIST. IN DB=',F8.5/) 51210520
CALL SINAMP(ISIN,D,K,TF) 51210530
74 RETURN 51210540
END 51210550
C***** 51210560
C SUBROUTINE: PAD * 51210570
C * 51210580
C THIS SUBROUTINE SIMULATES LINEAR GROUP DELAY DISTORTION * 51210590
C***** 51210600
SUBROUTINE LGD(NR,TF) 51210610
COMPLEX TF(1) 51210620
GO TO (74,1,2,3,4,5), NR 51210630
1 B=.05 51210640
GO TO 22 51210650
2 B=.095 51210660
GO TO 22 51210670
3 B=.14 51210680
GO TO 22 51210690
4 B=.19 51210700
GO TO 22 51210710
5 B=.23 51210720
22 WRITE(6,11) B 51210730
11 FORMAT(10X,'LINEAR GROUP DELAY IN NS/MHZ= ',F8.5/) 51210740
CALL LINGD(B,TF) 51210750
74 RETURN 51210760
END 51210770
C***** 51210780
C SUBROUTINE: PGD * 51210790
C * 51210800

```

C	THIS SUBROUTINE SIMULATES PARABOLIC GROUP DELAY DISTORTION	*	51210810
C*****			51210820
	SUBROUTINE PGD(NR,TF)		51210830
	COMPLEX TF(1)		51210840
	GO TO (74,1,2,3,4,5), NR		51210850
1	S=.05		51210860
	GO TO 22		51210870
2	S=.1		51210880
	GO TO 22		51210890
3	S=.15		51210900
	GO TO 22		51210910
4	S=.2		51210920
	GO TO 22		51210930
5	S=.25		51210940
22	WRITE(6,11) S		51210950
11	FORMAT(10X,'PARABOLIC GROUP DELAY IN NS/MHZ**2=',F8.5/)		51210960
	CALL PARGD(S,TF)		51210970
74	RETURN		51210980
	END		51210990
C*****			51211000
C	SUBROUTINE: SAGD	*	51211010
C		*	51211020
C	THIS SUBROUTINE SIMULATES SINUSOIDAL GROUP DELAY DISTORTION	*	51211030
C*****			51211040
	SUBROUTINE SAGD(NR,TF,ISIN)		51211050
	COMPLEX TF(1)		51211060
	GO TO (74,1,2,3,4,5), NR		51211070
1	C=2.0		51211080
	GO TO 22		51211090
2	C=4.0		51211100
	GO TO 22		51211110
3	C=6.0		51211120
	GO TO 22		51211130
4	C=8.0		51211140
	GO TO 22		51211150
5	C=10.0		51211160
	GO TO 22		51211170
22	K=4		51211180
	WRITE(6,11) K,C		51211190
11	FORMAT(5X,'# OF RIPPLES =',I3,'SINE AMPL. GROUP DELAY =',F8.5/)		51211200
	CALL SINCGD(ISIN,C,K,TF)		51211210
74	RETURN		51211220
	END		51211230