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.

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

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

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.

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

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

Canada

On Weighted

Quadrature Amplitude Modulation:

Power-Bandwidth Performance,

Synchronization and Detection

by

Andy D. Kucar

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

Ottawa, Ontario

ISBN 0-315-46793-2

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

ACKNOWLEDGEMENT

The author would like to gratefully acknowledge the help provided by many individuals during the course of this research.

Sincere thanks go to his supervisor, Dr. Kamilo Feher, for his supervising, encouragement and support. A special appreciation goes to Dr. Willem Steenaart and Dr. Peter Galko for their invaluable help and understanding.

Fruitful discussions with the members of author's candidacy and thesis defence committees: Dr. A. Javed, Dr. J. Wight, Dr. W. McGee, Dr. W. Steenaart, and Dr. P. Goud are greatly appreciated.

Thanks is also extended to M.X. Chen, J. Coll, H. Girard, M. Hamze, D. Hatzinakos, P. Hill, Z.Z. Lei, P. Leung, J.X. Li, D. Makrakis, P. Mathiopoulos, M.S. Murthy, K. Nagaraj, H. Ohnishi, D. Prendergast, I. Sasase, M. Sato, R. Scott, K. Sreenath,M. St-Amand, I.Tezcan, J. Thorpe, K.T.Wu, A. Yongacoglu, and R. Zakarevicius for their assistance and support.

The help of many other individuals at the University of Ottawa, Telesat and Iskra, where the author was affiliated during his study, is deeply appreciated.

The author has been privileged to work with Dr. J. Vugrinec, whose honesty, hard dedicated work, broad knowledge and friendly human approach stimulated this contribution.

Finally,the author is deeply embeded to his spouse Mira for her timely support and understanding, and to their lovely daughters Dorothy and Kristina for their patience and cooperation. The author's brother Edward, parents, and the whole family deserve a special appreciation for their invaluable contributions.

ABSTRACT

This thesis investigates the performance characteristics of the newly introduced weighted quadrature amplitude modulation (WQAM) family and its corresponding new receivers.

A WQAM signal format consists of two quadrature components of the carriers that have generally unequal powers, and are modulated with independent data streams of different rates and different pulse shapes — composed of the rectangular and cosine waveforms, in general. Modulation schemes such as quadrature phase shift keying (QPSK), staggered QPSK (SQPSK), minimum shift keying (MSK), staggered quadrature overlapped raised cosine (SQORC), unbalanced QPSK, and M-ary QAM (MQAM) are all members of the WQAM family.

In order to reduce the adjacent channel interference, by using the gradient search technique, a few new pulse shapes giving a narrow mainlobe and minimal sidelobe levels are found.

The performance of 4-state WQAM schemes is evaluated in the adjacent channel interference linear and nonlinear channel environments. The results indicate that one of the new staggered WQAM schemes outperforms other known members of the WQAM family, i.e. (S)QPSK, MSK and (S)QORC in almost all practical situations.

The impact of amplitude and group delay distortions on MQAM schemes with more than 4 states is evaluated. Staggered QAM schemes perform better in the presence of linear group delay impairments, but nonstaggered schemes are less sensitive to linear amplitude gain impairments.

The correct detection of WQAM signals requires accurate carrier phase and symbol timing at the receiver. Therefore, a composite phase and timing estimation and data detection of WQAM signal sets is analyzed. By using a maximum likelihood approach, four new joint phase and timing estimators, for balanced and unbalanced WQAM schemes, are derived. At high signal-to-noise ratios, phase estimator and detector schemes merge into the classical decision feedback carrier recovery loop (DFCRL). The performance of this loop, employing an integrate-and-sample device in each quadrature arm (active loop), is evaluated in the presence of both carrier uncertainty ϕ and timing uncertainty λ.

The performance of overlapped schemes is degraded if classical carrier recovery schemes are employed, e.g. by 1.09 dB if SQORC is used. The theoretical results are experimentally verified on our Intersymbol-Jitter-Free (IJF) 64 kb/s modem. New equalized DFCRL which employs a simple 3-tap transversal equalizer in each quadrature arm is proposed for the carrier recovery and the detection of overlapped schemes. Our new loop does not exhibit the performance degradation associated with classical schemes.

Using a maximum-a posteriori probability approach and knowledge of the phase uncertainty ϕ, timing uncertainty λ, and nonlinearity surface of the loop $h(\phi, \lambda)$ — accumulated over corresponding observation intervals — improved loops, i.e. crosstalk cancellation DFCRL, timing intersymbol interference cancellation DFCRL, and FEEDLOOP, are introduced and analyzed. The theoretical results are verified by the Monte Carlo simulation. These loops

perform almost as well as a continuous wave (pilot tone) carrier recovery loop, do not exhibit quadrant ambiguities, and might be employed for phase and timing estimation and the detection of balanced and unbalanced WQAM schemes. The advantages of the new loops over the classical one become greater at lower probabilities of error and when higher state schemes are employed. For example at $P_e = 10^{-6}$ and in the presence of a normalized loop detuning of 0.7, the crosstalk cancellation loop (and FEEDLOOP) outperforms the classical one by 2.4 dB, assuming QPSK modulation is employed.

The impact of phase noise on the performance of MQAM, M-ary phase shift keying and M-ary quadrature partial response systems in a Gaussian noise environment is computed. For all cases the degradation due to phase noise is found to be less than 1 dB if a carrier-to-phase noise ratio in a double-sided Nyquist bandwidth $(C/N)_p$ is more than 10 dB higher than the carrier-to-thermal noise ratio $(C/N)_6$ required for the probability of error performance $P_e = 10^{-6}$. Performance graphs are presented which enable a fast first order approximation of the phase noise requirements of a system to be estimated. Our engineering rule-of-thumb approximations are in a close agreement with the results of measurements.

CONTENTS

LIST OF ABBREVIATIONS

ACI ..... Adjacent Channel Interference
AM/AM Amplitude Modulation to Amplitude Modulation Transfer due to Nonlinearity
AM/PM .. Amplitude Modulation to Phase Modulation Transfer due to Nonlinearity
BLQAM ..... Blackman Quadrature Amplitude Modulation (QAM scheme employing the Blackman window as an elementary pulse shape)
BPSK ..... Binary Phase Shift Keying
CCDFCRL ..... Crosstalk Cancellation Decision Feedback Carrier Recovery
CCI ..... Cochannel Interference
(C/N) ..... Carrier-to-noise ratio in a specified bandwidth
$(C/N)_6$ ..... Carrier-to-thermal noise ratio in a specified bandwidth required for the probability of error performance $P_e = 10^{-6}$
$(C/N)_p$ ..... Carrier-to-phase noise ratio in a specified bandwidth
CR ..... Carrier Recovery
CW ..... Continuous Wave
DA ..... Data Aided (Approach)
DFCRL ..... Decision Feedback Carrier Recovery Loop
DTTL ..... Data Transition Tracking Loop
FEC ..... Forward Error Correction (Codes)
FEEDLOOP Feedback,Estimation of phase,Estimation of timing, Decision LOOP (an acronym)
GD ..... Group Delay
HL ..... Hard Limiter
HMQAM ..... Hamming Quadrature Amplitude Modulation (QAM scheme employing the Hamming window as an elementary pulse shape)
HPA ..... High-Power Amplifier
IJF ..... Intersymbol Jitter Free (Feher's QPSK)
ISD ..... Integrate-Sample-and-Dump (Circuit)
ISI ..... Intersymbol Interference
LO ..... Local Oscillator
LPAM ..... L-ary Pulse Amplitude Modulation
MAP ..... Maximum a Posteriory
ML ..... Maximum Likelihood
MMSK ..... Modified Minimum Shift Keying

MPSK ..... M-ary Phase Shift Keying
MQAM ..... M-ary Quadrature Amplitude Modulation
MQPR ..... M-ary Quadrature Partial Response
MSK ..... Minimum Shift Keying
P/S ..... Parallel-to-Serial Converter
P_e ..... Probability of Error
$P(e|\phi)$ .... Conditional Probability of Error (conditioned on the phase uncertainty ϕ)
QAM ..... Quadrature Amplitude Modulation
QPSK ..... Quadrature Phase Shift Keying
QORC ..... Quadrature Overlapped Raised Cosine
RX ..... Receiver
SCPC ..... Single Channel per Carrier (Satellite System)
sign(a) ..... Signum of the Quantity a
SNR ..... Signal-to-Noise Ratio
S/N ..... Signal-to-Noise Ratio
SNR_{IF} .... Signal-to-Noise Ratio within the Intermediate Frequency (IF) Bandwidth
SP ..... Signal Processor, Signal Processing
S/P ..... Serial-to-Parallel Converter
SQAM ..... Staggered Quadrature Amplitude Modulation
SQPSK ..... Staggered Quadrature Phase Shift Keying
SQORC ..... Staggered Quadrature Overlapped Raised Cosine
SSB ..... Single Sideband
S3MQAM ..... Staggered (Offset) Quadrature Amplitude
Modulation scheme employing the class 3 minimal window as an elementary pulse shape
TDA ..... Transition Detector in the A Channel
TDB ..... Transition Detector in the B Channel
TR ..... Timing Recovery
TWTA ..... Travelling Wave Tube Amplifier
TX ..... Transmitter
UD ..... Uncertainty Diagram
$U\phi$ ..... Phase Uncertainty (Diagram)
VCO ..... Voltage Controlled Oscillator
VCC ..... Voltage Controlled Clock
WQAM ..... Weighted Quadrature Amplitude Modulation
3MQAM ..... Quadrature Amplitude
Modulation scheme employing the class 3 Minimal window as an elementary pulse shape
4MQAM ..... Quadrature Amplitude
Modulation scheme employing the class 4 minimal window as an elementary pulse shape

λ	timing uncertainty
ϕ	phase uncertainty
$h(\phi, \lambda)$	nonlinearity surface of the loop
$p(\phi, \lambda)$	probability density function of the process $\{\phi, \lambda\}$
$\rho \equiv SNR_L$	Signal-to-Noise Ratio within the loop bandwidth

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 DEMAND FOR POWER-BANDWIDTH EFFICIENCY

The available radio frequency spectrum is a limited natural resource, which demands that efficient transmission of information be done. In a satellite system the maximum available power utilization is of primary concern, which requires a power efficient modulation schemes to be employed [B1]-[B11]. An efficient power utilization requires a spacecraft traveling wave tube amplifier (TWTA), or solid-state power amplifier (SSPA), and possibly an earth station high-power amplifier (HPA) to be operated in a nonlinear mode, which causes intermodulation, AM/AM and AM/PM transfer, and spectrum regeneration. To avoid a spectrum spillover into the neighboring channels — which will cause adjacent channel interference (ACI) — a careful design of the signal shaping filters in the modem, the output filter which follows the HPA, transponder input and output multiplex filters, and possibly linearization of the HPA and TWTA needs to be done. However, tight filtering might cause increased intersymbol interference (ISI). Multipath propagation within adjacent transponders on the spacecraft and reuse of frequency by other satellite or terrestrial links will cause a further degradation of the system performance. In this environment, nearly constant envelope modulation schemes with 4 or 8 states and a spectral efficiency of about 1 bit/s/Hz are used.

In cable, telephone line, and terrestrial radio systems (except a mobile radio) the bandwidth efficiency is of primary concern, which leads to the use of high-level modulation schemes [S4], [S6]. 256 state schemes with a spectral efficiency in the range of 6 bit/s/Hz are the field leaders. Such schemes are sensitive to amplitude and group delay imperfections associated with practical (analog) filters, selective fading channels, ACI and cochannel interference (CCI), and phase and timing uncertainties.

During the past years,a tremendous amount of effort has been devoted to this subject. This includes novel modulation schemes,which perform well in a complex ACI and CCI environment [S1], [S3], [S5]; power efficient coding but at the expense of bandwidth efficiency [B19]; Ungerböck trellis coding in which an increased power efficiency is achieved without sacrifice of the bandwidth efficiency, but at the expense of the increased number of signal states and therefore increased sensitivity to carrier and timing uncertainties [P111], [S6]; combinations of modulation and coding, different linearization techniques to improve performance of a modulator-power amplifier combination [B10], [S3] —all these at the transmitter — and different carrier and timing recovery schemes with and without equalization at the receiver [B3], [S7]–[S8].

Many investigations into new modulation schemes (pulse shapes) have been based on trial-and-error methods. After a "new" pulse shape (modulation scheme) has been found, heuristically, the analysis — usually supported by computer simulation — followed. The procedure

was iterated until the "best" results were achieved. However, an optimal pulse shape for non-linearly amplified systems has not been described in the literature. Chapter 2 of this thesis deals with this problem in a different way. The idea and details are presented in the thesis outline and Chapter 2, respectively.

The bandwidth efficient modulation schemes usually operate in a linear channel environment, and the practical constraints differ significantly from those systems which operate in a power limited environment. Some of the typical problems, i.e. group delay and amplitude linearity impairments, estimation of the carrier phase and timing and detection, and degradation due to phase noise of the frequency sources are tackled in Chapter 3, Chapter 4 and Chapter 5, respectively.

1.2 DEMAND FOR ACCURATE CARRIER PHASE AND TIMING ESTIMATION, AND DETECTION

The correct detection of WQAM signals requires accurate carrier phase and symbol timing. It is desired to estimate these parameters directly from measurements on the received data signal. Since WQAM schemes are of the suppressed carrier type, a nonlinear operation on the receiver data is necessary. When the number of states or the imbalance increases, estimation becomes more complex and a higher sensitivity to any phase uncertainty ϕ and timing uncertainty λ might be expected. Some bandwidth and power efficient modulation schemes employ pulse weighting and overlapping to improve performance in nonlinear, ACI and CCI environments, which impose additional constraints on the receiver.

Numerous techniques for carrier and timing (clock) information recovery, based on a maximum likelihood (ML) or a maximum a posteriori (MAP) probability approach, have been analyzed. Among others, references [B3], [S7]–[S8] excellently summarize these efforts. Carrier recovery (CR) loops for an unbalanced QPSK scheme are analyzed in [P127]–[P130]. New CR loops for overlapped schemes are proposed in [P115]–[P116].However, studies [P131]-[P136] show the superiority of a joint estimation of phase and timing, to which attention is devoted in Chapter 4 of this contribution. Kobayashi [P131] proposed a joint estimator and detector for QAM and SSB schemes which employs a decision-directed feedback. The proposed ML receiver is a recursive type which minimizes the probability of error of an entire data sequence and estimates phase and timing. Gaussian and generally unknown channels which invoke an adaptive receiver are analyzed. Falconer and Salz [P132] showed that the receiver of [P131] does not necessarily produce minimum symbol error probability. An ML receiver is proposed where phase and timing estimates are updated on a symbol-per-symbol basis,rather than by an average taken over an entire data sequence. Mengali [P133] analyzed two different recursive methods for carrier phase and timing acquisition. The coupling between phase and timing is discussed in view of its effects upon the stability and convergence rate of the synchronization algorithm. Franks [P134], and Meyers and Franks [P135] examined a joint carrier phase and symbol timing estimation in SSB and MQAM schemes. The importance of cyclostationarity for CR and timing recovery (TR) is highlighted and new joint CR+TR ML based estimators are suggested. Poklemba [P136] presented a similar, although different, practical realization of a joint estimator-detector for the QPSK signal set. Kam [P83] pointed out some of the shortcomings of previous approaches and suggested a new ML receiver.

Most of the previous approaches dealt with optimization of phase and timing estimators and eventually the conditional probability of error for small uncertainties ϕ and λ. However, in practice, an average probability of error $P_e = \int_\phi \int_\lambda P(e|x,y)\, p(x,y)\, dx\, dy$ is more important. Optimization of the system performance corresponds to the minimization of

the average probability of error P_e, which is equivalent to the minimization of the product $P(e|\phi,\lambda)\,p(\phi,\lambda)$. Herein, the product of this conditional probability of error $P(e|\phi,\lambda)$ times the probability density function $p(\phi,\lambda)$ of a joint estimator shall be minimized over the entire range of ϕ and λ. The probability density function $p(\phi,\lambda)$ of the decision feedback loop is difficult to obtain in a closed form. Therefore, we describe $p(\phi,\lambda)$ by the nonlinearity surface of the loop $h(\phi,\lambda)$, the importance and meaning of which is explained further, later. By knowing $P(e|\phi,\lambda)$ and $h(\phi,\lambda)$ we are able to minimize P_e and therefore improve the overall system performance.

1.3 THESIS OUTLINE

The flowchart of the thesis is presented in **Fig.1.1.** In **Chapter 2** the WQAM family is introduced. First, attention is focused on the pulse shaping and overlapping at the transmitter to keep the spectral density main lobe as narrow as possible while at the same time minimizing the sidelobe level within the prescribed bandwidth. Although coding might be combined with the WQAM schemes to improve their performance even further, coding analysis is outside the scope of this contribution. Our signal shape consists of weighted rectangular and cosine functions only, which allows an easy practical implementation of the modem even at the highest data rates. We use the gradient search technique to minimize sidelobe levels within an equivalent baseband $|fT_s| = 0$ to 5 (this corresponds to 1 to 2 adjacent channels in practice). The results of this minimization are the values of the weighting coefficients in the WQAM modulator, the meaning of which will be further explained, later. It might be expected that such a modulation scheme with low sidelobes will perform well in an ACI environment. Results which follow show that this hypothesis is correct. A comparison of 4 state WQAM schemes — new and known — is given, including signal shapes, power spectral densities, and performance in the linear and nonlinear ACI environments — typical for satellite systems.

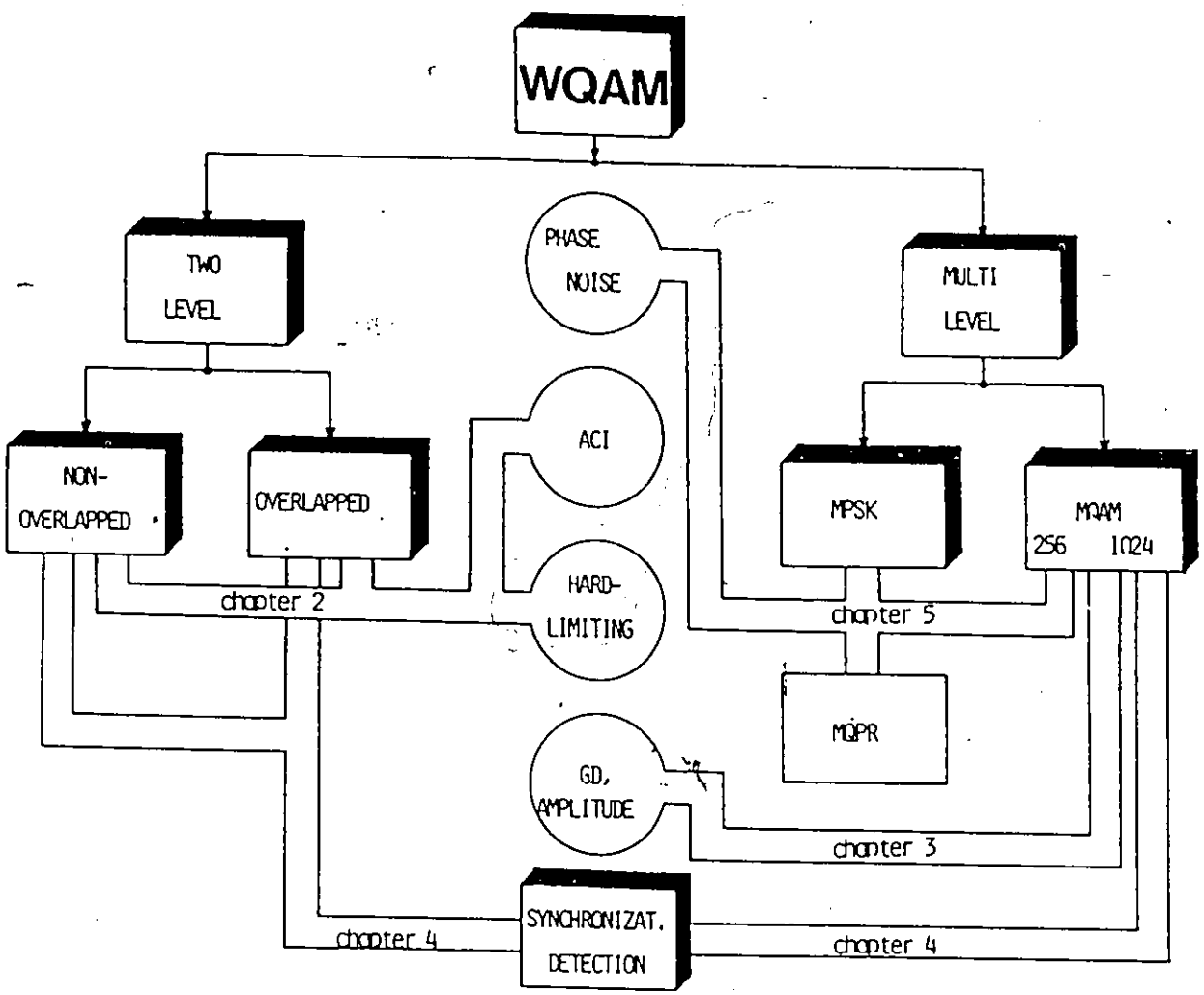


Fig.1.1. The flowchart of the thesis.

WQAM schemes with more than 4 states might combine different signal shapes at different signal levels to improve the performance in a particular environment. As an example of a possible application of high level WQAM schemes, in **Chapter 3**, a brief feasibility study of the transmission of North American T1 (DS1) 1544 kb/s or CCITT 2048 kb/s data stream over the analog 240 kHz wide CCITT supergroup is performed. The performance of the simplest, rectangular window, 256 and 1024 state, staggered and nonstaggered QAM schemes is analyzed in the presence of amplitude and group delay imperfections — typically associated with the supergroup filters.

In Chapter 2 and Chapter 3 a perfect knowledge of the carrier phase and symbol timing was assumed. However, this information is not available at the receiver in advance. Before any decision has to be made, the receiver must establish the correct phase and timing reference. This problem is considered in **Chapter 4**. First, by using a ML approach, four new joint phase and timing estimators are derived. Then, based on the knowledge of ϕ, λ, and estimator-detector behavior accumulated over the observation interval(s), MAP detectors are derived. Since both phase and timing estimates depend on the probability of correct decision, this improved decision leads to an improved estimation, etc. As a result, a few new structures are suggested which cancel (or at least significantly attenuate) data-dependent pattern jitter, i.e. these loops perform nearly as well as a continuous wave (CW) loop.

The performance of any coherent digital modulation system is degraded by an excessive amount of phase noise. In this contribution the term **phase noise** is synonymous with non-thermal random interference such as phase noise of the frequency sources (local oscillators,up/down converters,voltage controlled oscillators), etc. In the past a considerable effort has been devoted by various authors to this subject [B3], [B13], [B15], [B16], [B20]. However, an exact mathematical solution for modulation schemes with more than 4 states has not been published. In **Chapter 5** we adopt a practical approach, which yields useful performance curves in which the degradation due to phase noise can be readily seen.

Chapter 6 summarizes the thesis results. In the **Chapter 7** a brief research proposal for the future work is outlined. This contribution concludes with an extensive list of references, and a printout of the computer programs.

9

2

4-state WQAM schemes

A brief outline of the chapter follows. In Section 2.1 the WQAM system is introduced. A family of weighted cosine pulse shapes is defined. The general expressions for power spectral density are given. The new modulation schemes termed 3M-,4M-,HM- and BLQAM are introduced. In Section 2.2 the performance of the WQAM schemes is evaluated in linear and hardlimited channels and in the presence of ACI by means of a computer simulation. A comparison of new and known modulation schemes is given in the table and graphs.

2.1. SYSTEM DESCRIPTION AND SPECTRUM OPTIMIZATION

A weighted quadrature amplitude modulation signal $s(t)$ with data rates $R_A = 1/T_A$ and $R_B = 1/T_B$, where A and B represent the in-phase and quadrature channels respectively, Fig.2.1.(a),is

$$s(t) = \sqrt{P_A}\sum_i \alpha_i w_A(t - iT_A)\cos(2\pi f_c t) + \sqrt{P_B}\sum_j \beta_j w_B[t - (jT_B + \delta)]\sin(2\pi f_c t) \quad (2.1)$$

The quantities P_A and P_B are the signal powers, α_i and β_j are independent binary (± 1) random data sequences which could be obtained from an information sequence through a serial-to-parallel converter, $w_A(t)$ and $w_B(t)$ are the pulse shaping functions (weights, windows) in channels A and B, defined in the interval $(0, T_A)$ and $(0, T_B)$, respectively, f_c is the carrier frequency, and δ is an arbitrary delay in the B channel. The present concept might be applied to multilevel, multistate systems, where, in general, different pulse shapes might be employed on different baseband levels. As an example, M-ary QAM schemes employ $\alpha_i, \beta_j \in \{-(L-1), \ldots, -1, +1, \ldots, +(L-1)\}$ and rectangular signal shape on every baseband signal level. By allowing $P_A \neq P_B$ and $R_A \neq R_B$ an (intentionally) unbalanced system might be analyzed, or the effects of imbalance on the performance of balanced systems might be estimated. Generally $P_A \neq P_B$ and $R_A \neq R_B$;i.e.,the QAM system is unbalanced. In this chapter, we limit our attention to the 4-state balanced system with $P_A = P_B = P$, $R_A = R_B = R = 1/2T = 1/T_S$ and $w_A(t) = w_B(t) = w(t)$, particularly the nonstaggered schemes with $\delta = 0$ and the staggered schemes with $\delta = T = T_S/2$. The equivalent baseband power spectral density of the WQAM signal $s(t)$ is [B15,p.120]

$$S_b(f) = \frac{2}{T_S}|W(f)|^2 \quad (2.2)$$

where $W(f)$ is the Fourier transform of the baseband signal pulse shape $w(t)$.

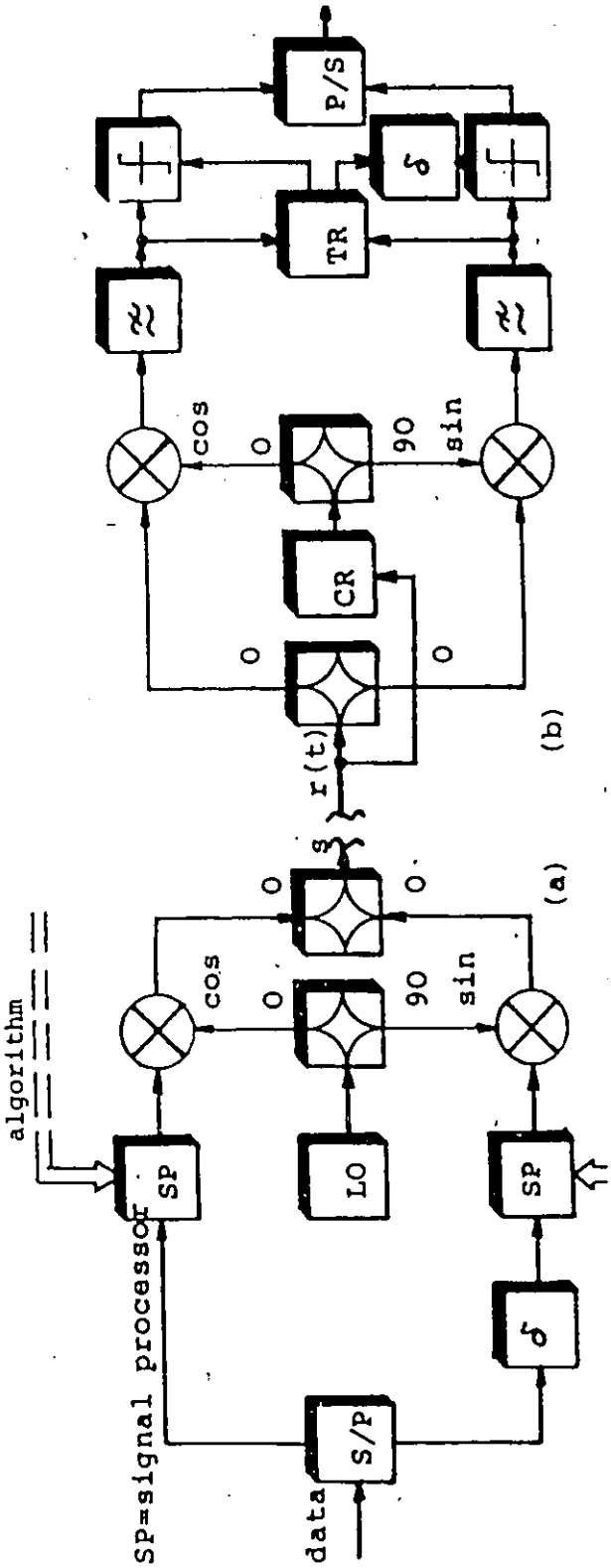


Figure 2.1. The weighted quadrature amplitude modulation system: a) Transmitter, b) Receiver.

We define the weighted cosine pulse shape as

$$w_n(t) \stackrel{\text{def}}{=} \begin{cases} \sum_{i=1}^{n} g_i \cos(\frac{\pi t K_i}{L_i}), & \text{for } |t| \le L_i/2, \\ 0, & \text{for } |t| > L_i/2, \end{cases} \qquad (2.3a)$$

where

$$K_1 = 0, \quad \sum_{i=1}^{n} g_i = 1. \qquad (2.3b)$$

The quantity g_i is the weighting factor ($g_i = a_i$,or $g_i = b_i$,refers to the A or B channel, respectively), K_i is the frequency parameter, L_i is the overlapping factor normalized to the symbol duration T_S and n is the class of the signal pulse shape. The Fourier transform of the weighted cosine signal is

$$\begin{aligned} W_n(f) &= \int_{-L_i/2}^{L_i/2} w_n(t) \exp(-j2\pi f t)\, dt \\ &= \frac{1}{2} \sum_{i=1}^{n} a_i L_i [W_{io}(fL_i - \frac{K_i}{2}) + W_{io}(fL_i + \frac{K_i}{2})] \end{aligned} \qquad (2.4)$$

where

$$W_{io} = \frac{\sin \pi x}{\pi x} = sinc(\pi x) \qquad (2.5)$$

The right choice of g_i, K_i and L_i parameters depends on the desirable goal. Here, our primary concern is the spectral density with narrow mainlobe and minimal sidelobe levels within $|fT_s| = 0$ to 5. Since the signal with larger overlapping factor L_i posseses the narrower spectrum,$L = L_i$ could be one of additional practical constraints. On the other hand, the signal with large L causes intersymbol interference at the sampling point.A further increase in L leads to a multilevel signal.The signal $w_n(t)$ with a smooth transitions from maximum to zero and a good compromise between the mainlobe width and the sidelobe levels suggests the K_i to be an even integer. As an example,assume the overlapping factor $L_i = L$,and $n = 3$. The Fourier transform $W_3(f)$ of the class 3 signals, $n = 3$ in (2.4), is

$$\begin{aligned} W_3(f) = &\underbrace{\left[\frac{\sin(\pi f L/2)}{(\pi f L/2)}\right]}_{\text{QPSK like}} \underbrace{\left[\frac{\cos(\pi f L/2)}{(fL)^2 - 1}\right]}_{\text{MSK like}} \\ &\times L \frac{4a_1 - (5a_1 - 4a_2 + a_3)(fL)^2 + (a_1 - a_2 + a_3)(fL)^4}{(fL)^2 - 1} \end{aligned} \qquad (2.6)$$

The spectral density consists of a QPSK-like factor, a MSK-like factor,and a third factor which can be optimized to obtain a better spectral performance. Our results, reported in

Section 2.2,indicate that the signals obtained for the linear channel also have a good performance in non-linearly amplified (hardlimited) systems. The performance of modulation schemes is summarized in **Table 2.1**. The pulse shape which gives minimum sidelobes within the $|fT_s| = 0$ to 5 frequency band is termed as the minimum. The choice of a pulse duration L will depend on an expected amount of ACI, frequency separation,and filter strategy. The subclass of windows termed as minimum, with $L = T_S$ is also known as the Blackman–Harris family [P63]. If $L > T_S$ the consecutive pulses will overlap, and the new elementary waveshape within the interval $(-T_S/2, T_S/2)$ will generally be asymmetric about the plane $t = 0$.The + and − pulses will also be asymmetric. However,the symmetry is maintained in average.

The QPSK, BPSK, and MQAM schemes analyzed in Chapter 3, are using rectangular signal shape (Dirichlet window). The MSK consists of Hann 1 signal in each of two quadrature channels,and the B channel is additionally staggered by $T_S/2$ to maintain a constant envelope of the signal $s(t)$,**Fig.2.1.** However,all other modulation schemes employ overlapping techniques (overlapping factor $L > 1$) to obtain a narrower spectrum. The envelope of signal $s(t)$ is not constant in general. But spectrum sidelobes of schemes with the nonconstant envelope are far below the spectrum sidelobes of schemes with the constant envelope.They remain low even after hardlimiting, **Fig.2.3.** The QORC scheme employs the Hann 2 signal window with the overlapping factor $L = 2T_S$.Further reduction of sidelobes level is achieved by employing the Hamming weighting and overlapping,i.e. the HMQAM.
In [P56] the MMSK signal family was introduced and the particular scheme with parameter A=0.8 was found to be the best,based on a trial-and-error method. However,the BLQAM which is equivalent to the MMSK scheme with A=0.84 seems to be the best.
The sidelobe level of the 3MQAM scheme is 12.7 dB lower than the BLQAM's and far below the QORC's.Additional sidelobe level reduction could be achieved at the expense of a wider mainlobe,i.e., by employing the 4MQAM.
The signal shapes $w_n(t)$ and the corresponding power spectral densities are given in Fig.2.2. and Fig.2.3, respectively.
The 3MQAM outperforms MSK in the whole frequency band,as well the QPSK, except around the first zero of the QPSK at $fT_S = 1$. From $fT_S = 1.17$ up,the sidelobes of 3MQAM are below the QORC. The 3MQAM represents a good compromise between the mainlobe width and the sidelobes level.

	Name of the window	1,zero width	Sidelobe level,dBr	Duration L/T	QAM name	a1 K1	a2 K2	a3 K3	a4 K4
1	Rectangular (Dirichlet)	1.00	-13.0	1.0	QPSK SQPSK	1.00000 0.00000	- -	- -	- -
2	Cosinel (Hann 1)	1.50	-23.0	1.0	MSK	- -	1.00000 1.00000	- -	- -
2	Cosine2 (Hann 2)	1.00	-31.4	2.0	QORC (IJF)	0.50000 0.00000	0.50000 2.00000	- -	- -
2	Hamming	1.17	-42.6	1.7	HMQAM	0.54000 0.00000	0.46000 2.00000	- -	- -
3	Blackman	1.50	-58.1	2.0	BLQAM	0.42000 0.00000	0.50000 2.00000	0.08000 4.00000	- -
3	Minimum Bl.-Harris3	1.50	-70.8	2.0	3MQAM	0.42323 0.00000	0.49755 2.00000	0.07922 4.00000	- -
4	Minimum Bl.-Harris4	2.00	-92.0	2.0	4MQAM	0.35875 0.00000	0.48829 2.00000	0.14128 4.00000	0.01168 6.00000

Table 2.1. The summary of WQAM system performance.

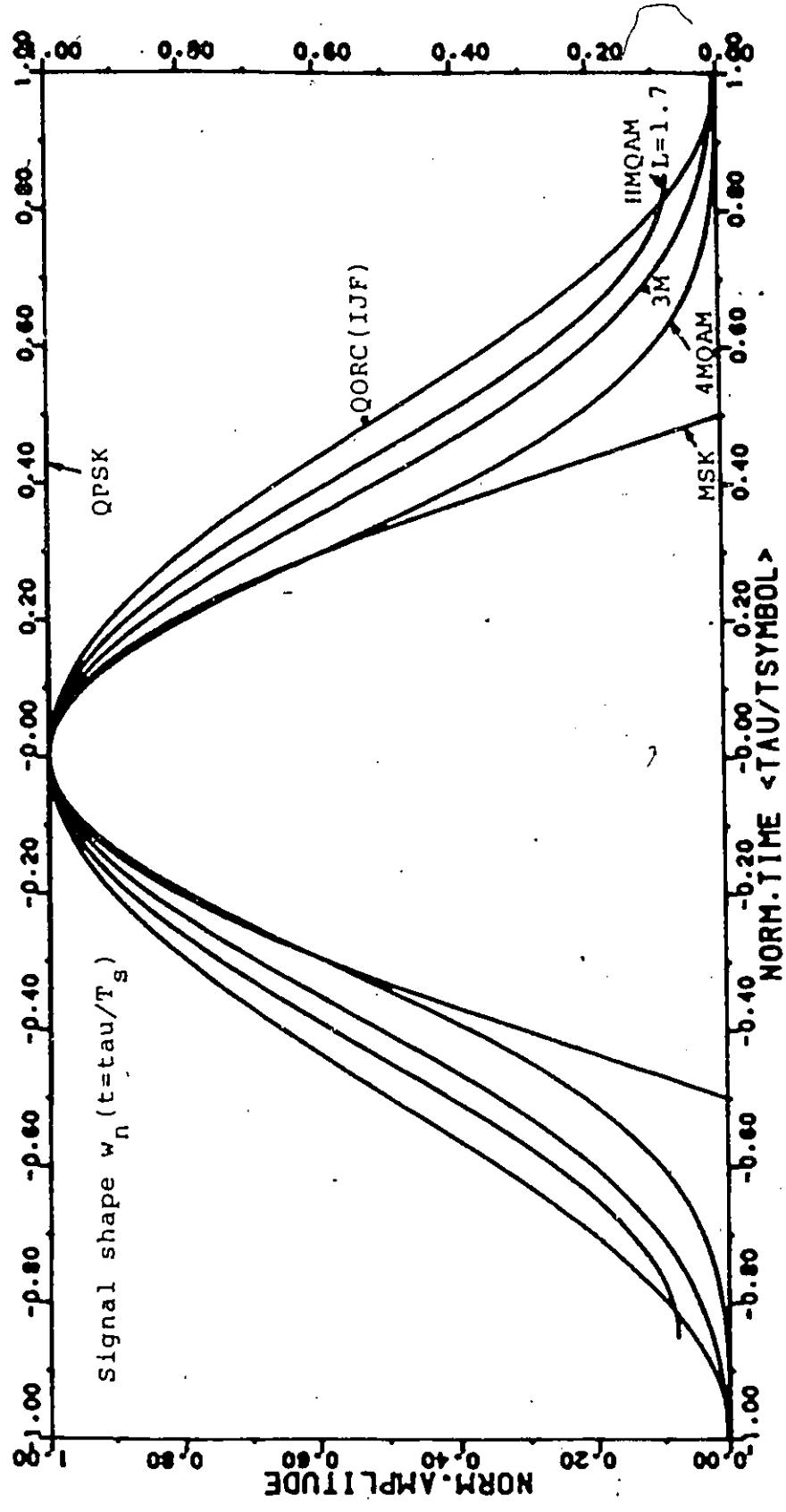


Figure 2.2. The signal shapes (windows) of the WQAM schemes: The QPSK (Dirichlet), MSK (cosine, Hann 1) — both the one symbol duration — , the HMQAM (Hamming) $L = 1.7\ T_S$ symbol duration, the QORC, i.e. IJF (cosine squared, Hann 2), the 3MQAM (minimum class 3), and the 4MQAM (minimum class 4) modulation schemes — all two symbols duration.

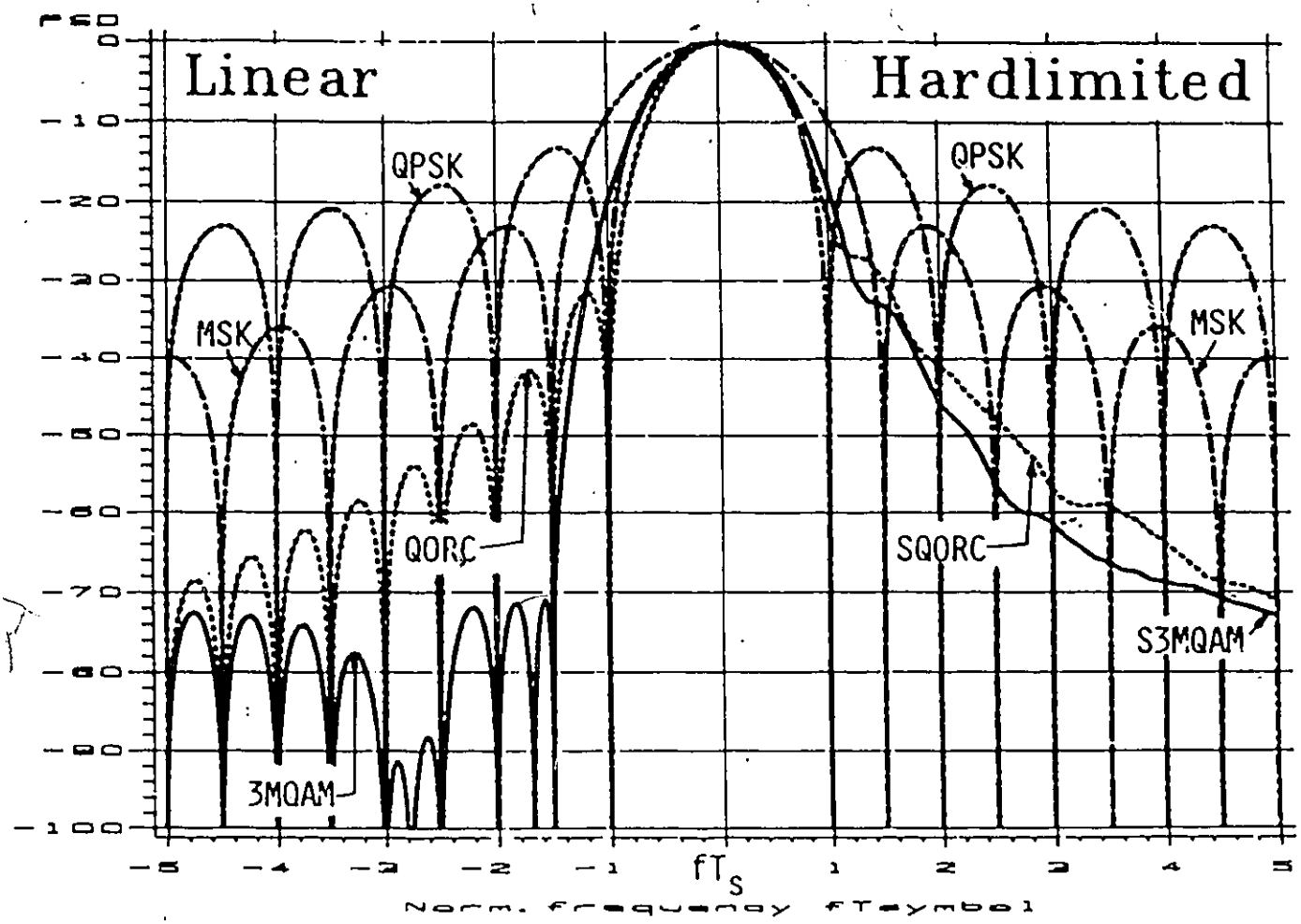


Figure 2.3. The power spectral densities of WQAM schemes. Ordinata in dB.

2.2 PERFORMANCE IN ACI ENVIRONMENT

To evaluate the performance of WQAM schemes in ACI environment the computer simulation procedure with the following assumptions is used. The WQAM receiver is illustrated in Fig.2.1(b).. Here ideal carrier recovery (CR) and symbol timing recovery (STR) are assumed. The simulation model of a system is given in Fig.2.4. A 5th order phase equalized (constant group delay) Butterworth filter with single-sided bandwidth $B_1T_S = 0.55$ is assumed in every channel. The interferer's phases and symbol timings are randomized over the interval $(0, 2\pi)$ and $(0, T_S)$ respectively.

2.2.1 Linear channel

In the linear channel system model the hardlimiter in Fig.2.4 is by-passed. The performance of the WQAM systems vs. the normalized channel spacing frequency in the presence of two ACIs with the power level $I_1 = I_2 = 0$ dBr, equal to the power of the main channel,is summarized in Fig.2.5.(a). The performance of WQAM systems vs. the ACI interference level,i.e. the relative fading depth,is given in Fig.2.5.(b). The staggered S3MQAM using simple Butterworth filter is performing well even in the presence of strong ACI and outperforms QPSK,SQPSK,SQORC(SIJF) [B8],[P52],and MMSK [P56].

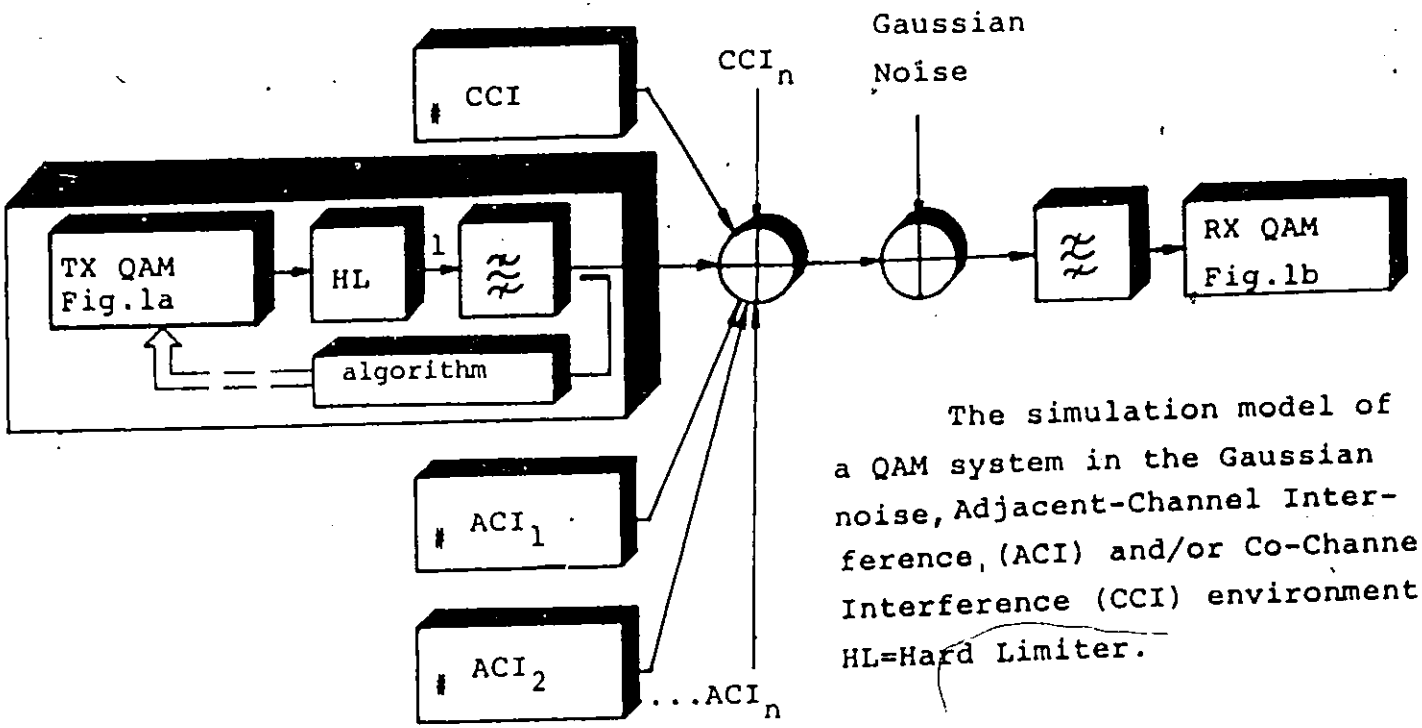
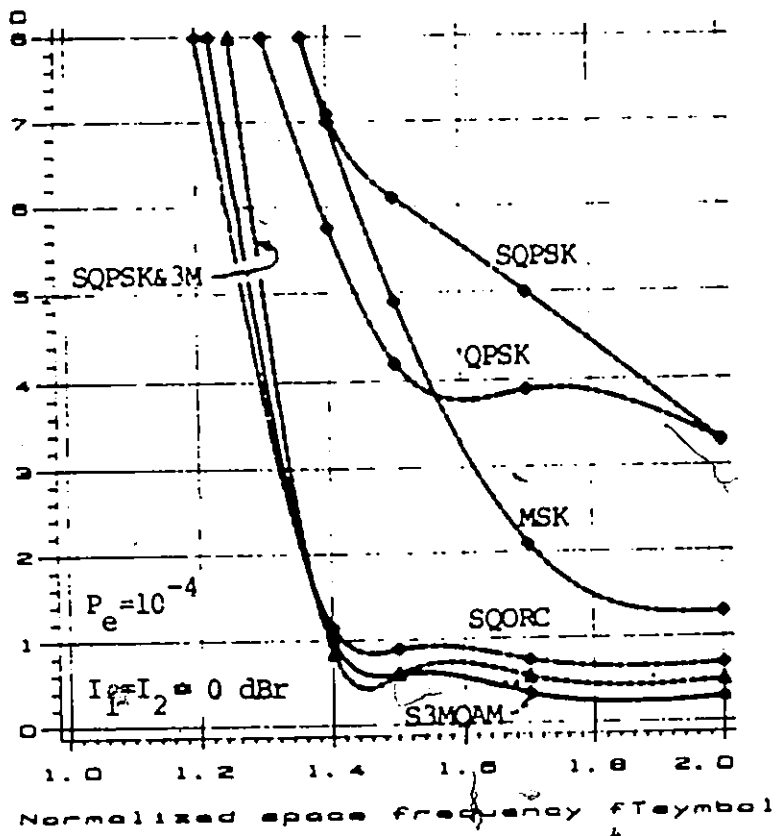
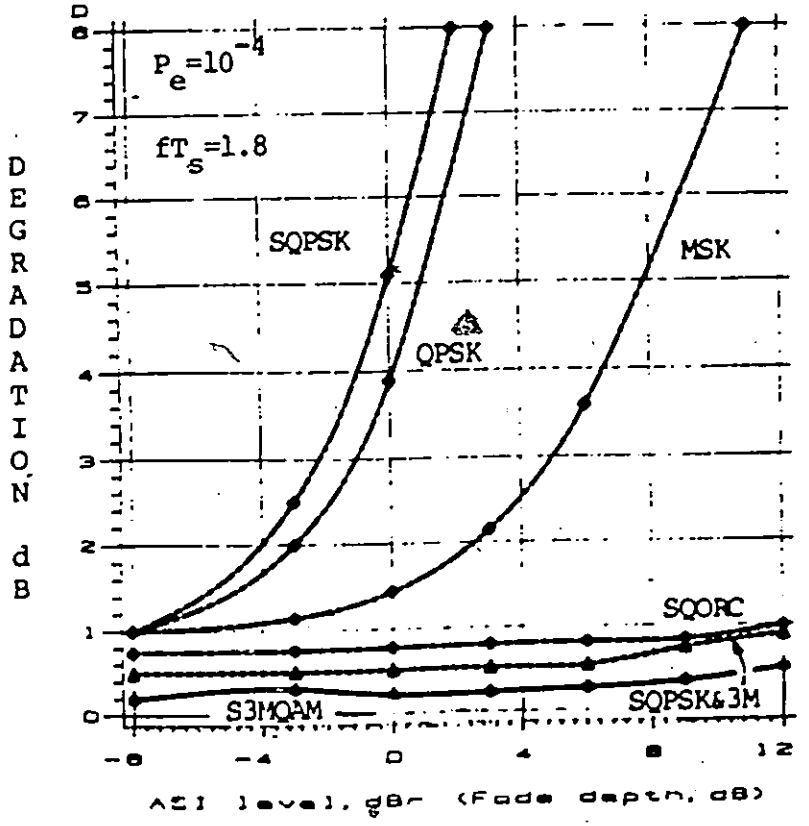


Figure 2.4. The simulation model of a QAM system.



a) The performance of the WQAM systems versus the normalized channel spacing frequency fT_S in the presence of two equal power adjacent channel interferers.



b) The performance of the WQAM systems versus the adjacent channel interferers' level in dBr (or flat fading depth in dB).

The ordinates represent the degradation in dB, relative to the theoretical value $E_b/N_o = 8.4$ dB at $P_e = 10^{-4}$. The (S)QPSK, MSK, (S)QORC, and (S)3MQAM — all, main channel and interferences are of the same type, correspondingly. The SQPSK & 3M labeled curve represents SQPSK main channel and two ACI interferences of the S3MQAM type.

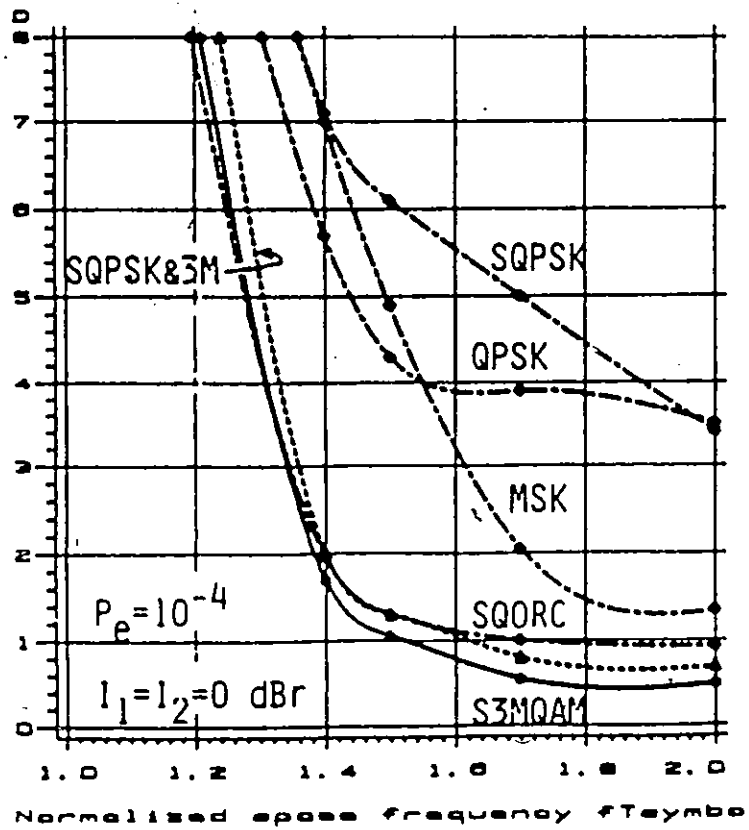
Figure 2.5. The linear channel performance of WQAM schemes.

2.2.2 Hardlimited channel

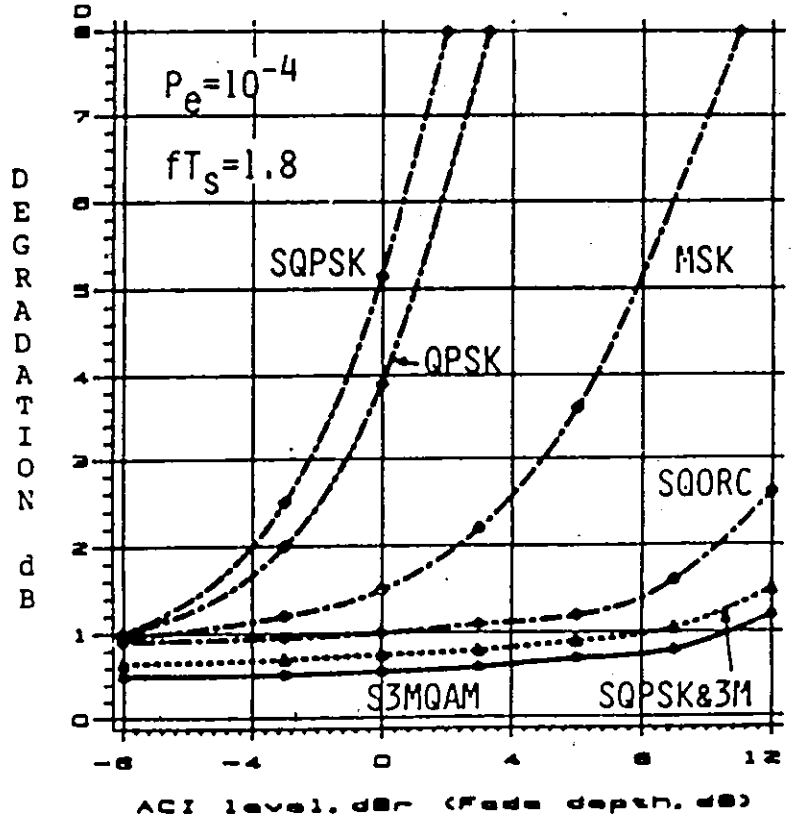
The power spectral density of the signal $s(t)$,Point 1 after the hardlimiter in **Fig.2.4**,is given in **Fig.2.3**. The sidelobes of S3MQAM scheme are about 10 dB below those of the SQORC and far below QPSK and MSK. In the present filtering strategy configuration,the constant envelope signals (QPSK,SQPSK and MSK) perform in exactly the same way as in the case of linear channel. The performance of the WQAM systems vs. the normalized channel space frequency and in the presence of two ACIs with the power level $I_1 = I_2 = 0$ dBr, equal to the level of the main channel, is summarized in **Fig.2.6.(a)**. The performance of the WQAM system vs. the ACI interference level,i.e. a relative flat fading depth,is illustrated in Fig.2.6.(b).

The S3MQAM outperforms all competitors if a normalized space frequency of adjacent channels is larger than 1.35 fT_S. As the fade depth increases the advantage of the S3MQAM becomes more obvious. As an example, at the fade depth of 12 dB, S3MQAM outperforms SQORC (IJF) modulation scheme by more than 1 dB, **Fig.2.6b**.

Single channel per carrier (SCPC) satellite systems above 10 GHz, and terrestrial and satellite systems for mobile, maritime, and aeronautical services, are some of the typical examples where these schemes might find an application. In practice, observed —main— channel might be faded due to the rain (above 10 GHz satellite SCPC systems) or due to the shadowing (mobile radio), while neighboring channels do not experience either fading or shadowing. Under this circumstances the main channel is experiencing an attenuation of the signal level, and in addition it must cope with an increased ACI. As it is shown, our new S3MQAM scheme outperforms its competitors under these practical conditions.



a) The performance of the WQAM systems versus the normalized channel spacing frequency fT_S in the presence of two equal power adjacent channel interferers.



b) The performance of the WQAM systems versus the adjacent channel interferers' level in dBr (or flat fading depth in dB).

The ordinates represent the degradation in dB, relative to the theoretical value $E_b/N_o = 8.4$ dB at $P_e = 10^{-4}$. The (S)QPSK, MSK, (S)QORC, and (S)3MQAM — all, main channel and interferences are of the same type, correspondingly. The SQPSK & 3M labeled curve represents SQPSK main channel and two ACI interferences of the S3MQAM type.

Figure 2.6. The hardlimited channel performance of WQAM schemes.

2.3 CONCLUSION

In this chapter the weighted quadrature amplitude modulation (WQAM) was introduced. By using the gradient search technique, a family of pulse shapes with a narrow mainlobe and minimal sidelobe levels within an equivalent baseband $|fT_s| = 0$ to 5 (this corresponds to 1 to 2 adjacent channels in practice) was found. The performance of 4-state WQAM schemes was evaluated in the ACI linear and nonlinear channel environments, with the channel spacing and fading depth (signal attenuation) as parameters, by means of the computer simulation. One of the new staggered WQAM schemes, termed **S3MQAM**, outperforms other known members of the WQAM family, i.e. (S)QPSK, MSK and (S)QORC in almost all practical situations. By using the S3MQAM scheme more compact spacing —**more efficient transmission, i.e. more b/s/Hz**— than before might be possible.

3

256 and 1024QAM schemes

The WQAM family was introduced in the previous chapter. The performance of 4 state WQAM schemes was evaluated in the presence of ACI. In this chapter the performance of the 256 and the 1024QAM schemes, which employ rectangular window (pulse shape) at each symbol level, is evaluated in the presence of amplitude and group delay impairments. These are the typical impairments associated with the practical (analog) filters employed in the radio and cable systems.

An outline of the chapter follows. In Section 3.1 an overview of system considerations is given. The computer simulation model is presented in Section 3.2. The chapter concludes with the results of the computer simulation.

3.1 SYSTEM CONSIDERATIONS

To transmit 1544 (2048) kb/s over a 240 kHz wide CCITT supergroup, a digital modulation scheme with a spectral efficiency better than 1544(2048)/240=6.43 (8.53) b/s is required. To achieve this, (S)256QAM and (S)1024QAM schemes, with a theoretically maximum spectral efficiency of 8 b/s/Hz and 10 b/s/Hz respectively,are considered in this chapter. Forward error correcting (FEC) coding might be employed,which will bring the corresponding channel rate to the 1.6 Mb/s (2.1 Mb/s). Throughout this chapter the information rate of 1544 (2048) kb/s is associated with a channel rate of 1.6 (2.1) Mb/s. The 1.6 Mb/s might be transmitted by the 256QAM scheme, which will give the symbol rate of 0.2 MBaud and the spectral efficiency of 6.67 b/s/Hz. Should 1024QAM scheme be used,the symbol rate will be 0.16 MBaud, **Table 3.1.** In this case the less steep cosine filter with $\alpha = 0.5$ would be sufficient ($\alpha = 0.2$ if 256QAM is employed). The 2.1 Mb/s bit rate requires the 1024QAM to be employed,which gives the symbol rate of 0.21 MBaud,noise bandwidth of 210 kHz,channel spectral efficiency of 8.75 b/s/Hz and $\alpha = 0.14$ filters,**Table 3.1.**

Table 1. The 256 and the 1024QAM system considerations.

	256QAM			1024QAM		
Mb/s	MBaud	α	b/s/Hz	MBaud	α	b/s/Hz
1.6	0.20	0.20	6.67	0.16	0.50	6.67
2.1	—	—	—	0.21	0.14	8.75

3.2 COMPUTER SIMULATION MODEL

In practice, at the transmitter the 1.544 (2.048) Mb/s data stream might be interleaved and FEC encoded to 1.6 (2.1) Mb/s and split into two streams by a serial-to-parallel converter S/P. Each stream might be differentially and Gray encoded into the 16 (32) level signals with the symbol rate of 200 (210) kBaud. These signals are fed into the $\alpha = 0.2$ ($\alpha = 0.14$) cosine filters and modulated by the in-phase and quadrature signals.
256QAM (1024QAM) signal is fed to the in-phase and quadrature receiver. The in-phase signal is used to extract the symbol timing clock. After demodulation,the baseband adaptive transversal equalizer might be applied to equalize the impairments caused by the supergroup-through-connect filters and other system impairments. After Gray and differential decoding,the in-phase and quadrature signals are parallel-to-serial converted into a 1.6 (2.1) Mb/s signal. This signal is de-interleaved and FEC decoded and becomes the 1.544 (2.048) Mb/s replica of the original signal.
To be able to simulate the influence of amplitude and group delay impairments on the 256QAM and 1024QAM systems we adopted a simplified computer model without coding and equalization and with the following properties: the pseudo-random data has period of $2^{31} - 1$, a half million samples is used for every run, the carrier frequency is set to the zero,an ideal carrier recovery is assumed,Gaussian noise is included analytically and the timing recovery has the memory which is 256 symbols long, i.e. practically perfect timing exists. †
The computer model of the M-ary quadrature amplitude modulation,MQAM,system is given in Fig.3.1. The input data stream $d(t)$ with the data rate of 1.6 (2.1) Mb/s is split, by using the serial-to-parallel converter S/P,into the two independent data streams $A(t)$ and $B(t)$,respectively. The $B(t)$ could be staggered by a time delay $\delta = T_S/2$,where T_S is the time period of $B(t)$. $A(t)$ and $B(t)$ are processed in the $2 \rightarrow L$ level converter, $x/\sin x$ and $\sqrt{\alpha}$ cosine filter to give the corresponding signals $a(t)$ and $b(t)$, respectively. $M = L \times L$ and α is the roll-off factor of the cosine filter. The corresponding parameters are given in Table 3.1. The channel is represented by the filter H(f) and the white Gaussian noise. In the practice,the cascade of supergroup (SG)-through-connect filters might have a significant impact on the performance of transmission. At the receiver the in-phase and quadrature channels are fed into the $\sqrt{\alpha}$ cosine filters. Assuming wide H(f),this filtering will satisfy both Nyquist and matched filter criteria,i.e. an optimum reception. If this is not the case,an adaptive equalizer should be employed. The carrier recovery and timing recovery circuits provide the correct phase and timing instants to make a correct decision. Finally,by $L \rightarrow 2$ mapping and the parallel-to-serial conversion P/S an estimation of the original data, $\hat{d}(t)$, is

† The carrier recovery and timing recovery schemes are analyzed in the Chapter 4.

available.
The results previously achieved by this computer model were in a close agreement with the measured results made on different hardware models with up to 256 states. Here,we present the performance of 256QAM and 1024QAM modems in the presence of amplitude or group delay imperfections similar to those associated with practical/SG-through-connnect filters.

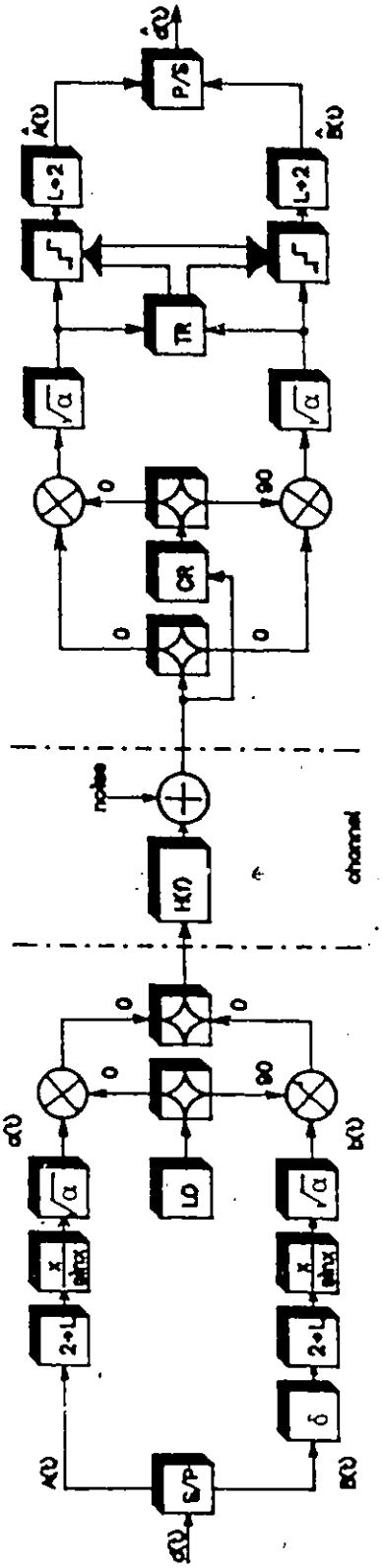


Figure 3.1. The computer model of MQAM system.

3.3. RESULTS

Three sets of results are presented. First, the performance of the 1.6 Mb/s modems using 256QAM schemes and a symbol rate of 0.2 MBaud is compared with the 1.6 Mb/s modems using 1024QAM schemes with a symbol rate of 0.16 MBaud.

- The degradation in dB is the difference between the carrier-to-noise ratio (C/N) required to achieve the $P_e = 10^{-6}$ performance when group delay or amplitude imperfections exist,and the $(C/N) \doteq 33$ dB needed to achieve the same probability of error performance P_e with the 256QAM scheme in an ideal Gaussian channel environment.

The performance of 256QAM and 1024QAM systems in the presence of linear group delay is summarized in **Fig.3.2.** ‡ Staggered schemes seem to be less sensitive to the linear group delay imperfections than their nonstaggered counterparts. In the presence of strong group delay impairments the S1024QAM,with more relaxed filter $\alpha = 0.5$,outperforms (S)256QAM schemes with $\alpha = 0.2$. In the Hermitian symmetric channel (parabolic group delay as given in **Fig.3.3** and parabolic amplitude gain as given in **Fig.3.5**) the staggered and nonstaggered schemes perform equally. However,in the presence of the linear amplitude gain impairments nonstaggered schemes outperform their staggered counterparts, **Fig.3.4**, i.e. 256QAM gives the best performance and it is followed by S256QAM,1024QAM and S1024QAM. Usually amplitude distortions are less difficult to be equalized,which might prefere staggered schemes to be employed.

‡ The definition of group delay and amplitude gain corresponds to that given in [B8, p.185].
† dB higher implementation margin for the 1024QAM schemes is assumed in figures which follow.

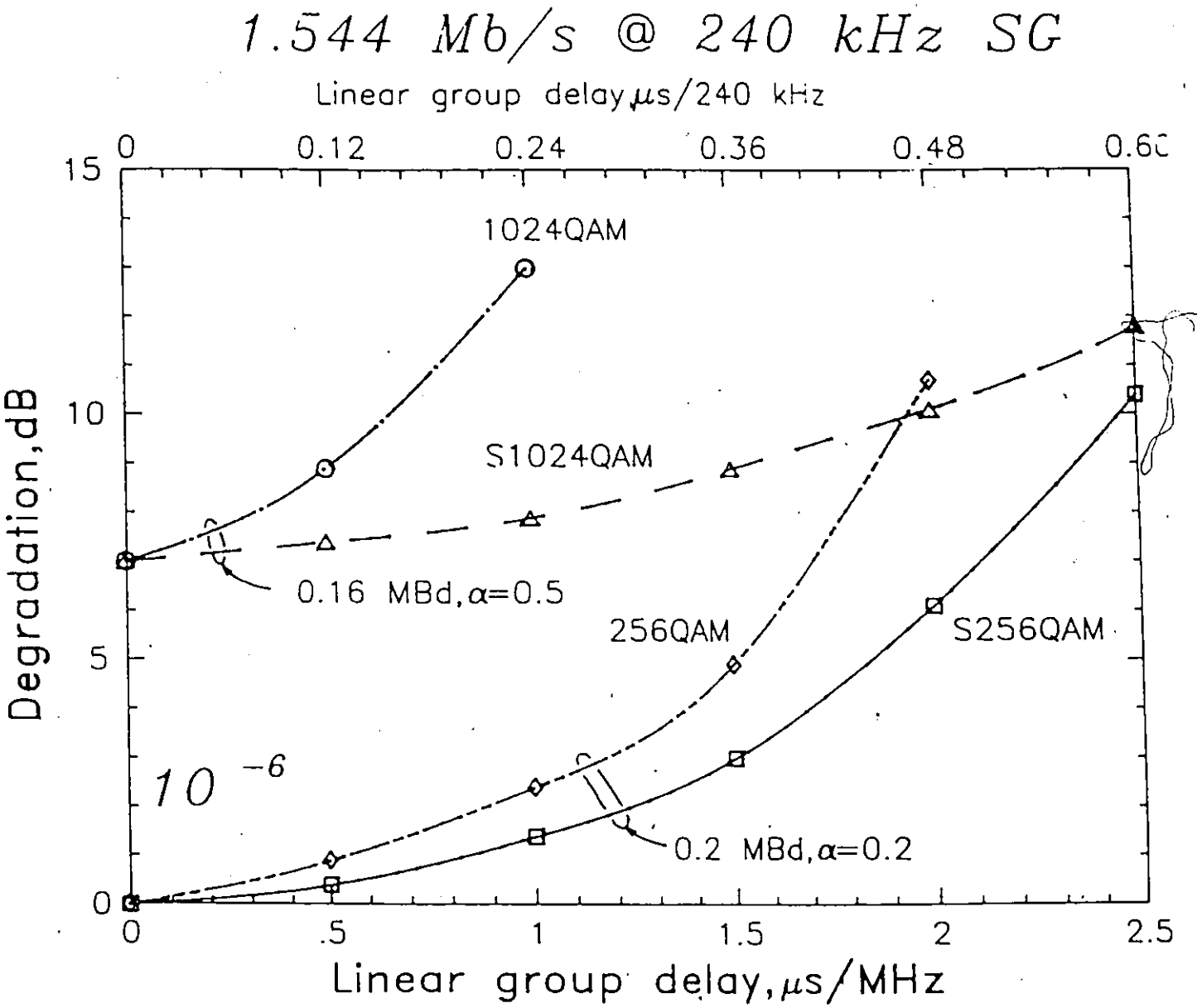


Figure 3.2. The performance degradation of the 256 and 1024QAM (1.544/1.6) Mb/s systems versus the linear group delay impairments. The degradation in dB is the difference between the C/N necessary to achieve the $P_e = 10^{-6}$ performance when impairments are present, and the C/N≈33 dB giving $P_e = 10^{-6}$ without impairments.

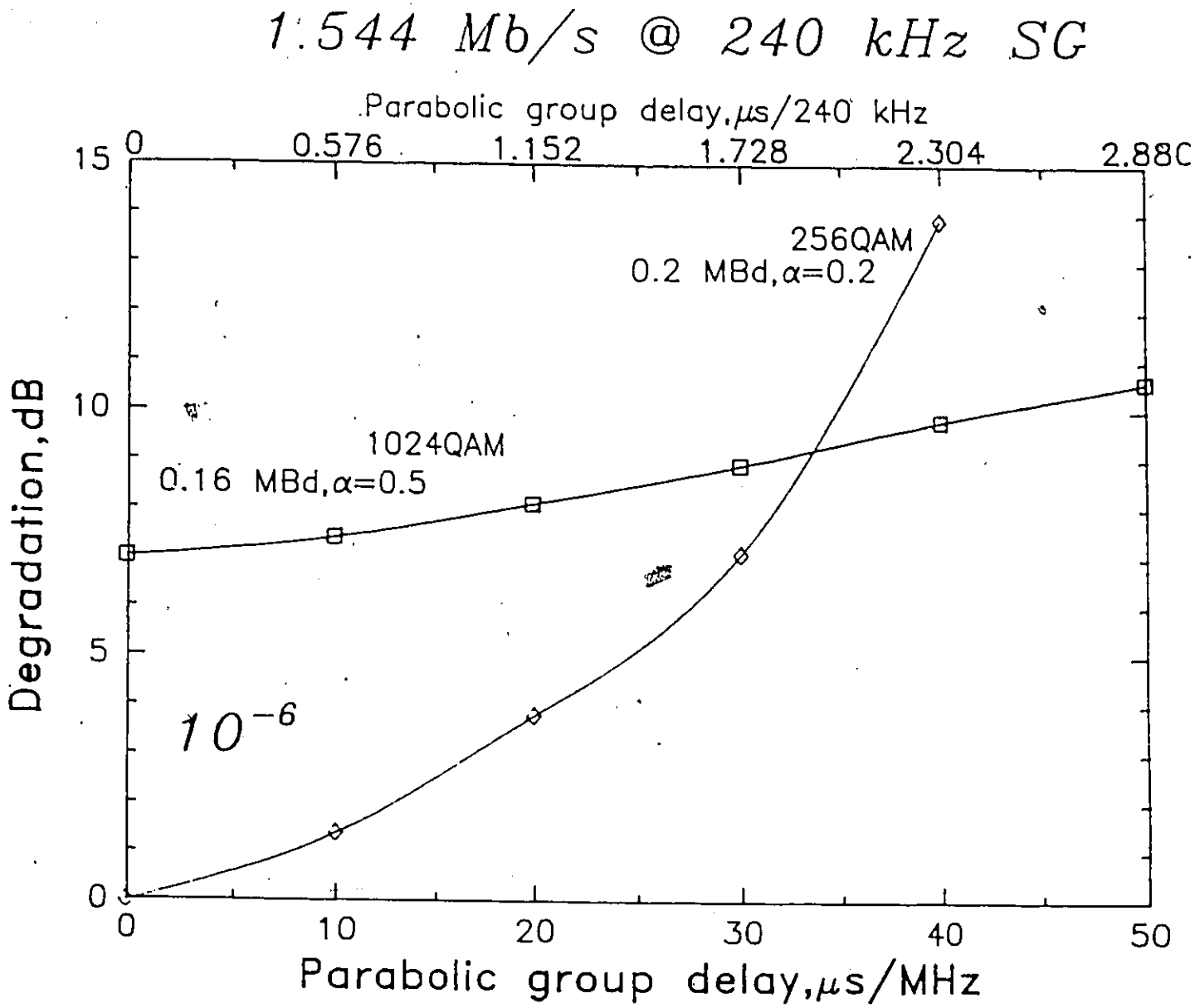


Figure 3.3. The performance degradation of the 256 and 1024QAM (1.544/1.6) Mb/s systems versus the parabolic group delay impairments. The degradation in dB is the difference between the C/N necessary to achieve the $P_e = 10^{-6}$ performance when impairments are present, and the C/N≈33 dB giving $P_e = 10^{-6}$ without impairments.

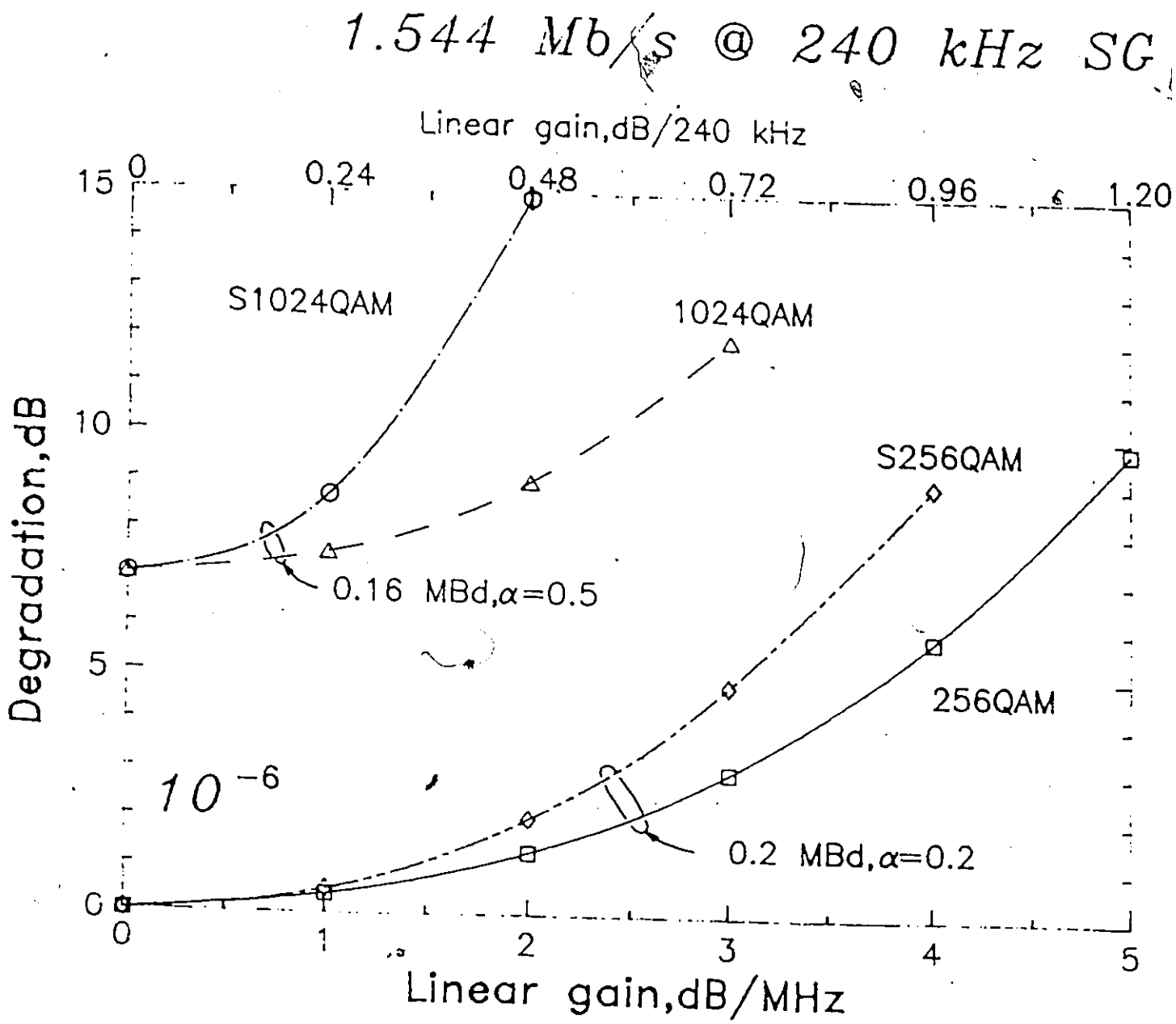


Figure 3.4. The performance degradation of the 256 and 1024QAM (1.544/1.6) Mb/s systems versus the linear amplitude gain impairments. The degradation in dB is the difference between the C/N necessary to achieve the $P_e = 10^{-6}$ performance when impairments are present, and the C/N≈33 dB giving $P_e = 10^{-6}$ without impairments.

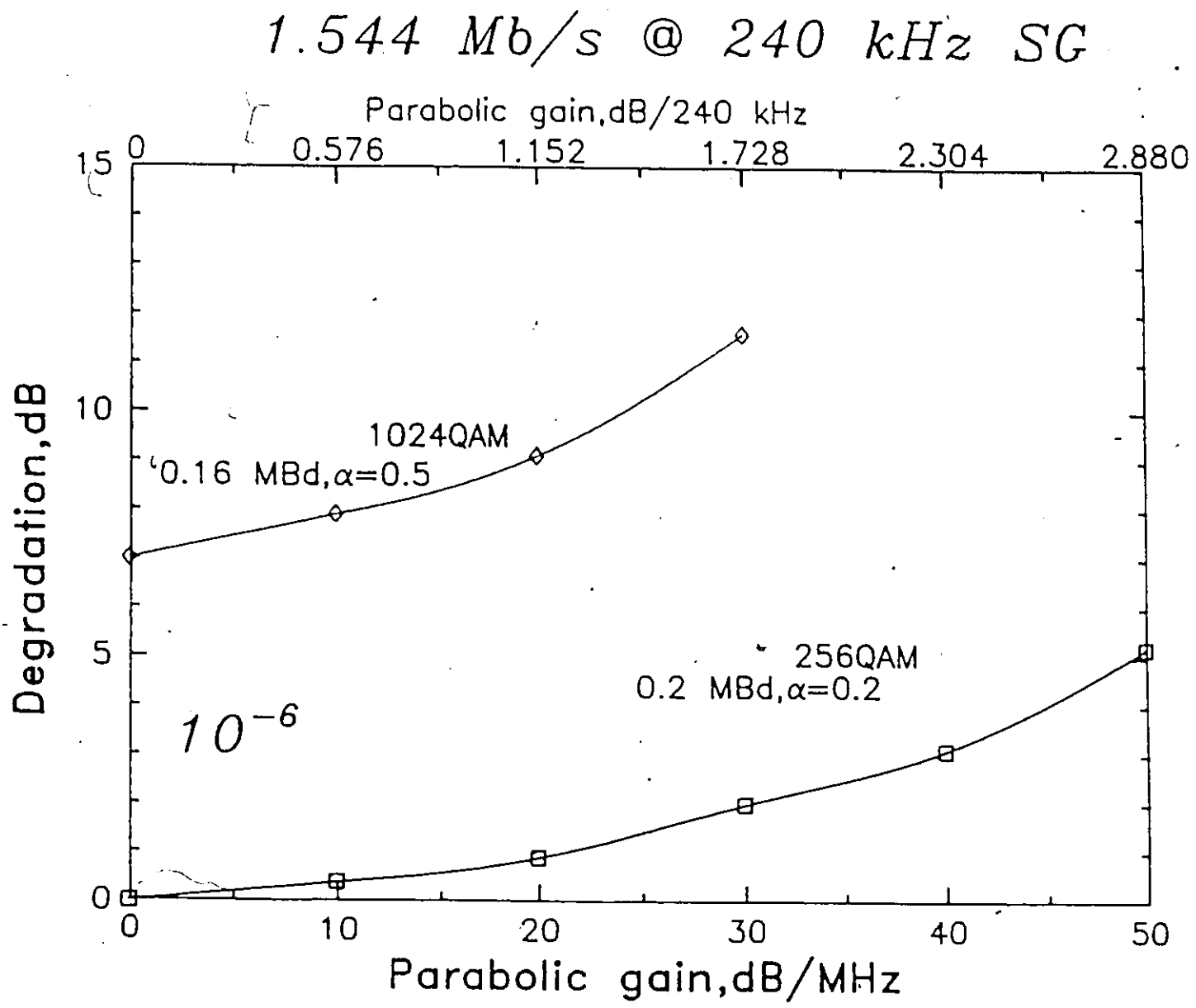


Figure 3.5. The performance degradation of the 256 and 1024QAM (1.544/1.6) Mb/s systems versus the parabolic amplitude gain impairments. The degradation in dB is the difference between the C/N necessary to achieve the $P_e = 10^{-6}$ performance when impairments are present, and the C/N≈33 dB giving $P_e = 10^{-6}$ without impairments.

Second,the performance of the 2.1 Mb/s modems using the 1024QAM schemes is given.

- The degradation in dB is the difference between the (C/N) required to achieve the $P_e = 10^{-6}$ performance when group delay or amplitude imperfections exist, and the $(C/N) \doteq 39$ dB needed to achieve the same P_e performance with the 1024QAM system in an ideal Gaussian noise channel.

The staggered schemes seem to be less sensitive to the linear group delay while there is no difference in the performance in the presence of parabolic distortions, **Fig.3.6.** However,the nonstaggered schemes perform better in the presence of linear amplitude gain distortion, Fig.3.7.
Third, the computer generated eye diagram of the 256QAM system in the presence of 1.5 μs/MHz group delay is given in **Fig.3.8 (a).** The central eyes are quite open,while the timing eyes are smeared, i.e.timing jitter exists. The eye diagram of the same system but without group delay distortion is given in **Fig.3.8 (b).** In the S1024QAM system with the 0.5 μs/MHz linear group delay and $\alpha = 0.14$ cosine filter middle eyes are much narrower,intersymbol interference is present,while the small timing eyes are totally smeared,i.e. a strong timing jitter exists, **Fig.3.9.** However,this will not be the case in an ideal $\alpha = 1.0$ Nyquist channel, i.e. both sampling and timing eyes are fully open, **Fig.3.10.** Obviously, high level modulation schemes are very sensitive to any timing recovery uncertainty.

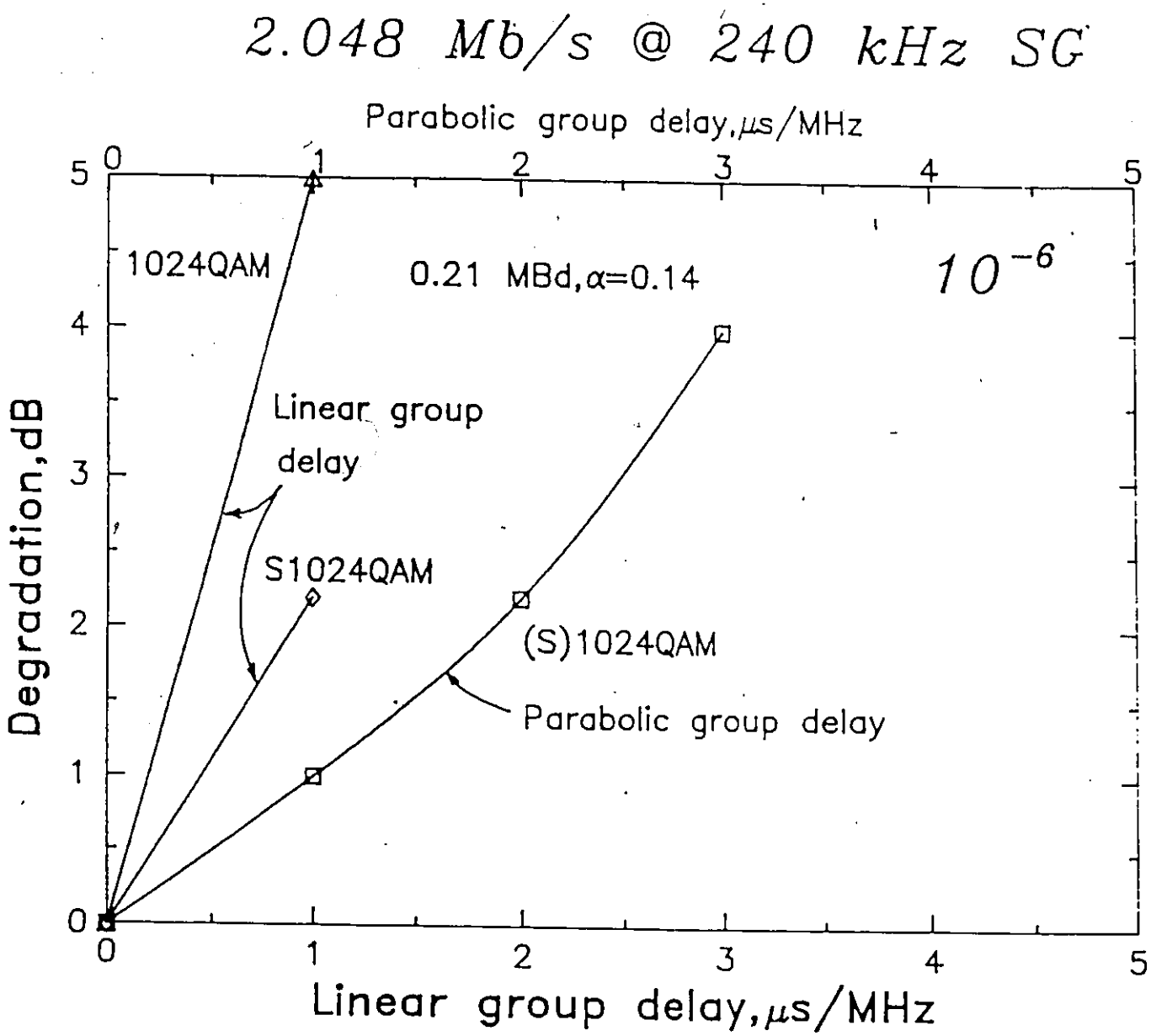


Figure 3.6. The performance degradation of 1024QAM (2.048/2.1) Mb/s systems versus the group delay impairments. The degradation in dB is the difference between the C/N necessary to achieve the $P_e = 10^{-6}$ performance when impairments are present, and the C/N≈39 dB giving $P_e = 10^{-6}$ without impairments.

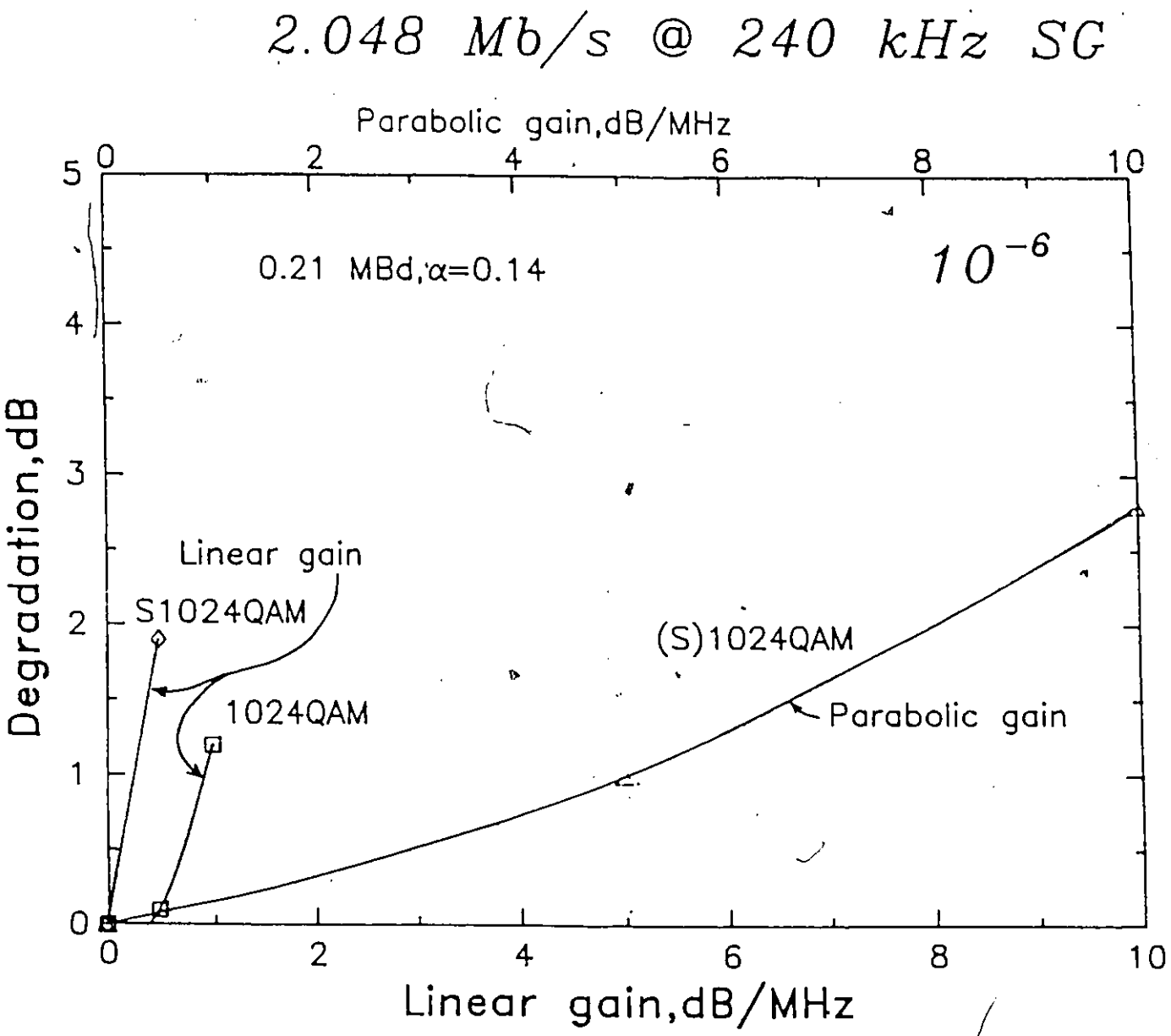


Figure 3.7. The performance degradation of the 1024QAM (2.048/2.1) Mb/s systems versus the amplitude gain impairments. The degradation in dB is the difference between the C/N necessary to achieve the $P_e = 10^{-6}$ performance when impairments are present, and the C/N≈39 dB giving $P_e = 10^{-6}$ without impairments.

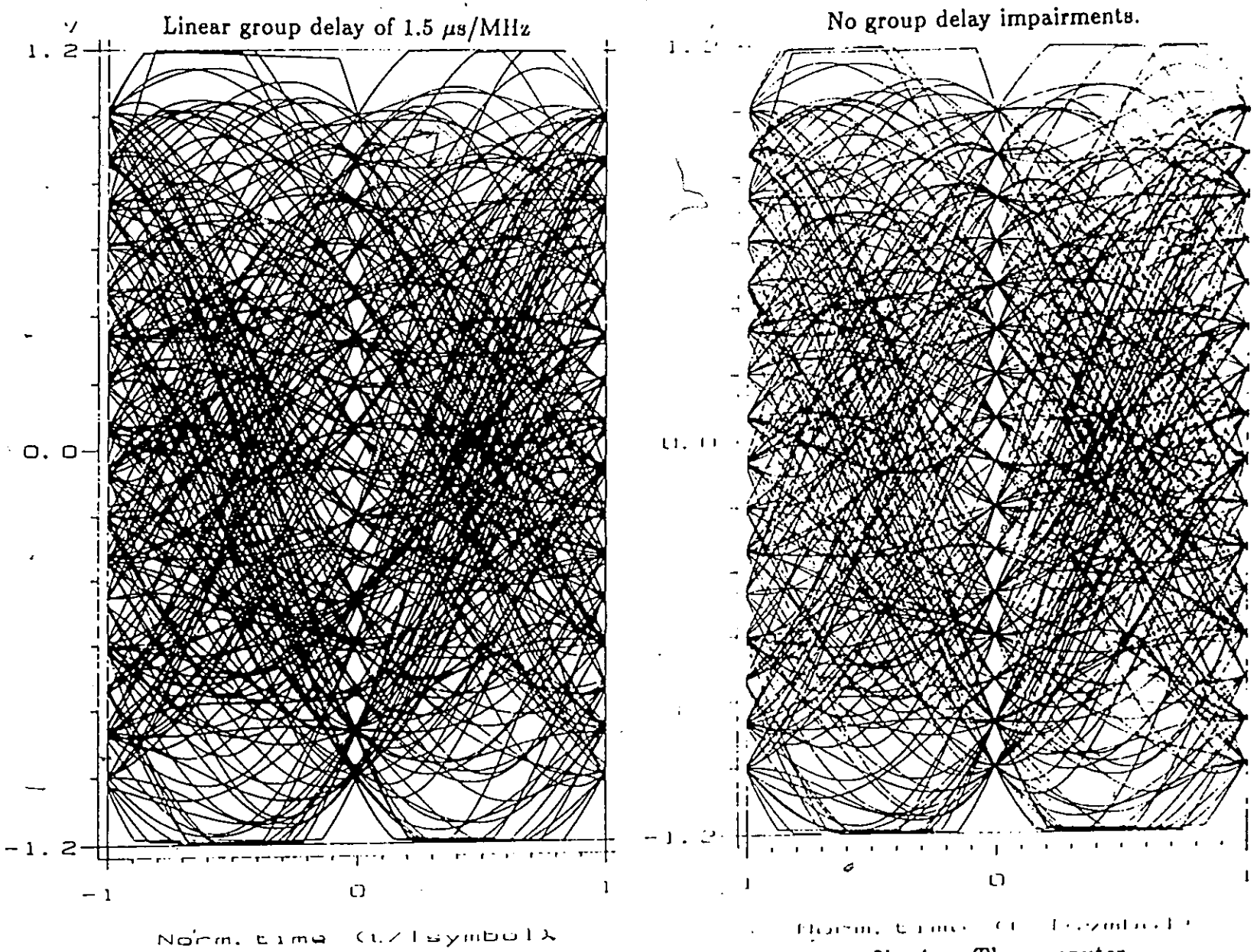


Figure 3.8. The 256QAM 0.2 MBd system with the $\alpha = 0.2$ filtering. The computer generated eye diagrams.

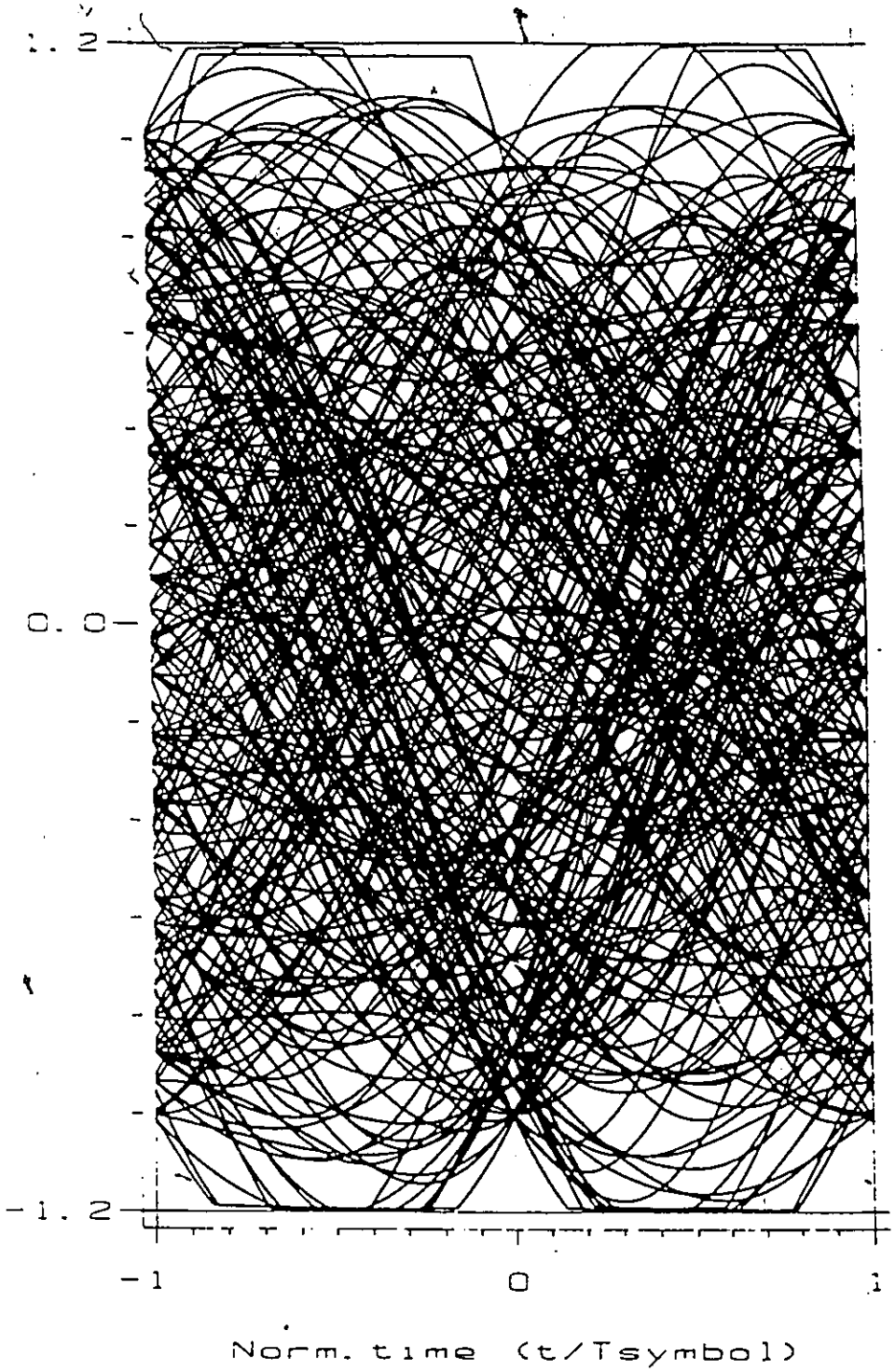


Figure 3.9. The S1024QAM 0.21 MBd system with the $\alpha = 0.14$ filtering. Linear group delay of 0.5 μs/MHz.The computer generated eye diagram.

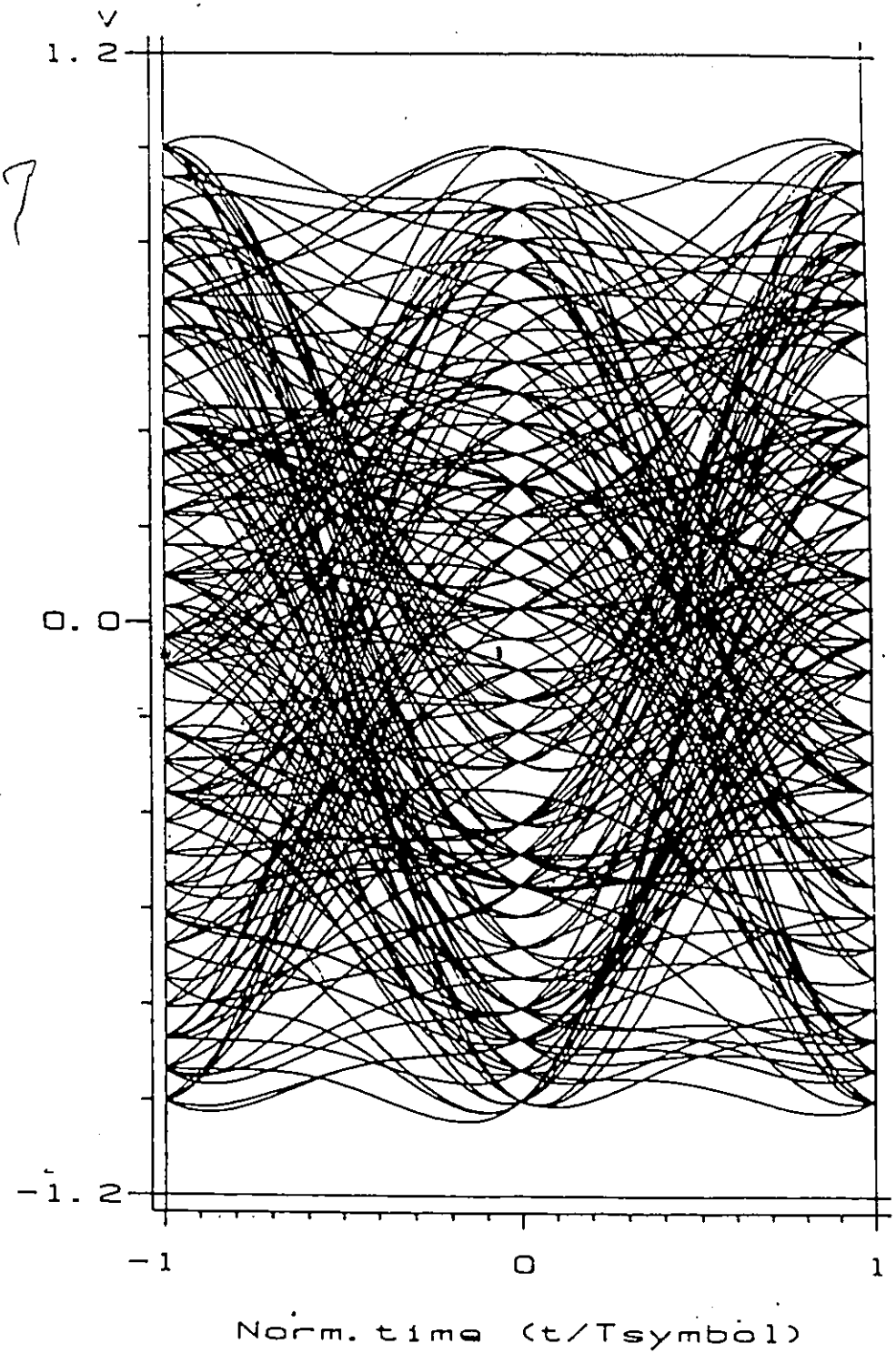


Figure 3.10. The 1024QAM 0.21 MBd system with the $\alpha = 1.0$ filtering. No group delay impairments. The computer generated eye diagram.

3.4. CONCLUSION

In this chapter the performance analysis of 256 and 1024QAM schemes is performed in the presence of amplitude and group delay impairments. A brief feasibility study of the transmission of North American T1 (DS1) 1544 kb/s or CCITT 2048 kb/s data stream over the analog 240 kHz wide CCITT supergroup is given. Staggered QAM schemes perform better in the presence of linear group delay impairments, but nonstaggered schemes are less sensitive to linear amplitude gain impairments.

In this, and the previous chapter, it has been assumed that the receiver is provided with a perfect knowledge of the carrier phase and the timing instants. However, in practice, this information is not available in advance. Before any decision is to be made, the receiver must establish, by itself, a proper phase and timing. This topic is the concern of our next chapter.

4

ON COMPOSITE PHASE AND TIMING ESTIMATION AND DETECTION OF WEIGHTED QAM SIGNALS

In Chapter 2 the WQAM family has been introduced. Its performance has been analyzed in the presence of different impairments such as: ACI, hardlimiter nonlinearity (Chapter 2), amplitude and group delay (Chapter 3) imperfections. An exact knowledge of the carrier phase and symbol timing(s) has been assumed. However, in practice, this is not the case and prior any decision has to be made, the proper carrier phase and symbol timing(s) need to be established. In this chapter, we search for new (improved) receivers for composite phase and timing estimation and detection of WQAM signals, and then, compare their performance with the known (conventional) ones.

An outline of the chapter follows. The WQAM signal has been defined in (2.1). However, for the purposes of this chapter a more precise definition is necessary. This, and the Bayes-based optimization strategy are given in the Section 4.1. Section 4.2 deals with ML-based classical- and equalized DFCRL. The performance of these loops is presented in the form of an uncertainty diagram (UD) and a nonlinearity surface. Finally, MAP-based crosstalk cancellation DFCRL,timing intersymbol interference (ISI) cancellation DFCRL, and FEEDLOOP are presented in Section 4.3.

4.1. SYSTEM DESCRIPTION AND OPTIMIZATION STRATEGY

A weighted quadrature amplitude modulation signal $s(t,\alpha,\beta,\epsilon_1,\epsilon_2,\theta)$ with data rates $R_A = 1/T_A$ and $R_B = 1/T_B$ in the in-phase and quadrature channels respectively, can be represented by

$$s(t,\alpha,\beta,\epsilon_1,\epsilon_2,\theta) = \sqrt{P_A}\sum_i \underbrace{\alpha_i w_A[t - iT_A + \epsilon_1(t)]}_{x} \cos[2\pi f_c t + \theta(t)]$$

$$+ \sqrt{P_B}\sum_j \underbrace{\beta_j w_B[t - (jT_B + \delta) + \epsilon_2(t)]}_{y} \sin[2\pi f_c t + \theta(t)],$$

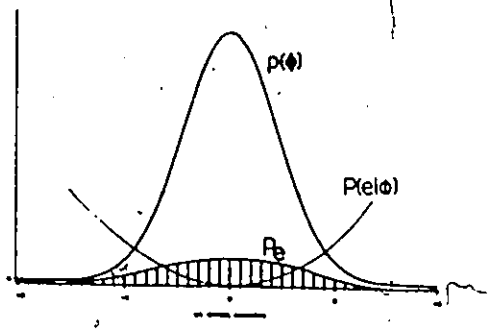
$$i,j \in \{I\} \quad (4.1)$$

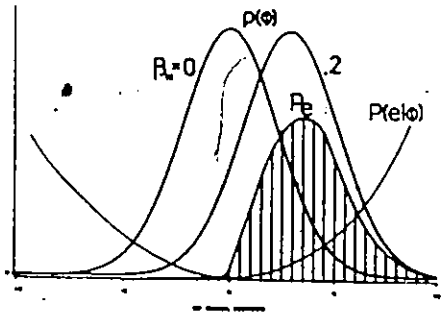
Quantities P_A and P_B are the signal powers, α and β are independent random data sequences, $w_A(t)$ and $w_B(t)$ are the pulse shaping functions (weights,windows) in channels A and B, defined on the interval $(-T_A/2,+T_A/2)$ and $(-T_B/2,+T_B/2)$, respectively,f_c is the carrier frequency, and δ is an arbitrary delay in the B channel. $\epsilon_1(t)$ and $\epsilon_2(t)$ are random timing offsets assumed to be uniformly distributed on the interval $(-T_A/2,+T_A/2)$ and $(-T_B/2,+T_B/2)$, respectively— and independent of each other in general, and $\theta(t)$ is the random carrier phase assumed to be uniformly distributed on the interval $(-\pi,+\pi)$. Summations over i,j extend over the set of all integers $\{I\}$. In the text which follows we use abbreviations $\epsilon_1 = \epsilon_1(t)$, $\epsilon_2 = \epsilon_2(t)$ and $\theta = \theta(t)$.
We would like to minimize the average probabilities of error P_{eA}, P_{eB},where

$$P_{eA} = \int_\phi \int_{\lambda_1} P(e|x,y)\, p(x,y)\, dx\, dy \quad (4.2a)$$

$$P_{eB} = \int_\phi \int_{\lambda_2} P(e|x,y)\, p(x,y)\, dx\, dy \quad (4.2b)$$

Quantities $P(e|x=\phi,y=\lambda_1)$ and $P(e|x=\phi,y=\lambda_2)$ are the conditional error rates — where errors are committed by the decision devices within the loop and are conditioned on the phase uncertainty $\phi = \theta - \hat{\theta}$ and the timing uncertainties $\lambda_1 = \epsilon_1 - \hat{\epsilon}_1$ and $\lambda_2 = \epsilon_2 - \hat{\epsilon}_2$, respectively. $\hat{\theta},\hat{\epsilon}_1$, and $\hat{\epsilon}_2$ are estimates of the phase and timings as provided by the corresponding loops. $p(x=\phi,y=\lambda_1)$ and $p(x=\phi,y=\lambda_2)$ are the corresponding probability density functions. Typical $P(e|\phi,\lambda=0)$ and $p(\phi,\lambda=0)$ functions are given in Fig.4.1.





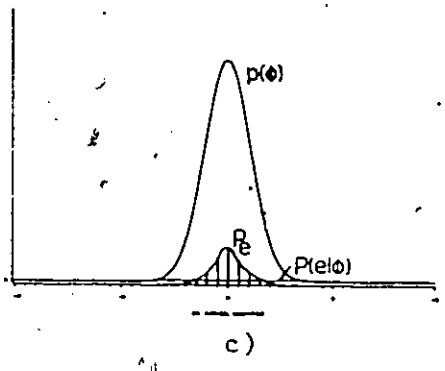


Figure 4.1. The typical conditional probability of error $P(e|\phi)$, probability density function $p(\phi)$, and and average probability of error P_e curves of the MQAM scheme employing: a) classical carrier recovery loop (e.g. classical decision feedback carrier recovery loop DFCRL) with normalized loop detuning $\beta_N = 0$, b) classical DFCRL with $\beta_N = 0.2$, and c) crosstalk cancellation DFCRL.
The values of corresponding $P(e|\phi)$ curves are enlarged for illustrative purposes. In practice they are as low as 10^{-12} at $\phi \approx 0$, and approach to 1 as $\phi \to 45°$.

The minimization of the P_{eA}, P_{eB} is equaivalent to the minimization of the corresponding products under the double integrals in (4.2a-b). Therefore, we would like to minimize these products over the entire range of $\{\phi, \lambda\}$. Both, yet unknown, functions depend on the known WQAM signal constellation and, yet unknown, circuit design which we would like to find. To achieve this, a Bayes estimate is used with the following cost functions: $C_{ii} = 0$ for a correct decision and $C_{ij} = 1,\ i \neq j$ for an erroneous decision [B22]. Although a similar analysis might be found in many books related to communications,our approach has a few differences which lead to new solutions.These differences are emphasized in the text which follows, including analysis of a few new receivers. A few preliminaries need to be established. The received signal $r(t)$ consists of the signal $s(t)$ and the band-limited ($-B_{IF}/2$ to $+B_{IF}/2$) white Gaussian noise with single sided spectral density N_o, i.e.

$$r(t) = s(t) + n(t) \tag{4.3}$$

where, for convenience, we have dropped the dependance of $s(t)$ and thus $r(t)$ on $\alpha, \beta, \epsilon_1, \epsilon_2, \theta$. We assume that the receiving IF filter H(f) is wide enough to cause no additional shaping of the signal pulse,and narrow enough to allow a narrowband representation of the noise $n(t)$,i.e.

$$n(t) = n_c(t)\cos(2\pi f_c t + \theta) + n_s(t)\sin(2\pi f_c t + \theta) \tag{4.4}$$

Furthermore,we assume that H(f) is symmetric around f_c,i.e. the in-phase noise component $n_c(t)$ and quadrature component $n_s(t)$ are statistically independent. In practice, a tight filtering might be imposed on both the transmitter and receiver side accompanied with nonlinearities,adjacent and cochannel interferences (ACI and CCI),selective fading, and other impairments. However,taking into account all of these factors at the same time leads to extremely complex expressions which remain to be solved. We limit our attention to a more tractable situation,where the WQAM signal is immersed in white Gaussian noise only. A responsibility of the receiver will be:based on observation of $r(t)$ as given in (4.3),decide which of data pairs (α, β) are sent.However first, both phase coherence and timing synchronization need to be established. Composite phase and timing estimation and detection analysis is performed as follows. We use the sampled data approach [B21]. $r(t)$ is uniformly sampled on the interval $(0, T_N)$ N times.The spacing between samples is $\Delta t = t_k - t_{k-1} = 1/2B_{IF}$, which creates independent samples. Now, (4.3) might be written in the sampled form

$$\vec{r} = \vec{s} + \vec{n} \tag{4.5}$$

Quantities $\vec{r}$ and $\vec{s}$ are the N-dimensional vectors of the received and transmitted signal,respectively,and $\vec{n}$ is the N-dimensional noise vector, the components of which are statistically independent zero-mean Gaussian random variables with variances $N_o B_{IF}$. Without loss of generality, we assume that a decision on $r(t)$ occurs at $t_k = k$,i.e. at the k-th data

symbol. Carrier phase is observed on the interval $[k-1-T_c, k-1]$ where,with very little loss of generality,it might be assumed that $T_c = V_{Ac}T_A$. V_{Ac} is an integer such that $V_{Ac} > 1$,where the first subscript A refers to the A channel and the second subscript c refers to the carrier phase estimation. We arbitrarily assume that $T_B = \xi T_A$ and $\xi \geq 1$,i.e. symbol rate R_A in the A channel is equal to or higher than R_B. During the interval $[k-1-T_c, k-1]$ there are V_{Ac} symbols $(\alpha_{k-1-T_c}, \alpha_{k-T_c}, \ldots, \alpha_{k-1})$ in the A channel if $\epsilon_1 = 0$, $V_{Ac}+1$ symbols (the right most part of $\alpha_{k-2-T_c}, \alpha_{k-1-T_c}, \ldots, \alpha_{k-2}$, the left most part of α_{k-1}) if $\epsilon_1 < 0$, and $V_{Ac}+1$ symbols (the right most part of $\alpha_{k-1-T_c}, \alpha_{k-T_c}, \ldots, \alpha_{k-1}$, the left most part of α_k) if $\epsilon_1 > 0$. The data stream in the B channel is generally not aligned to the stream in the A channel. Depending on the data rate ratio ξ,static offset δ,and amount of random offsets ϵ_1 and ϵ_2, different combinations exist. Generally,$V_{Bc}+1 = V_{Ac}/\xi + 1$ symbols in the B channel belong to the observation interval $[k-1-T_c, k-1]$. Timing epochs ϵ_1 and ϵ_2 are observed on the interval $[k-1-T_A, k-1]$ and $[k-1-T_B+\Delta\epsilon, k-1+\Delta\epsilon]$ respectively, where $\Delta\epsilon$ accounts for the relative offset between ϵ_1 and ϵ_2.

With these preliminaries,the Bayes estimate of ϵ_1, ϵ_2 and θ might be derived as follows: **Choose $\epsilon_1 = \hat{\epsilon}_1$, $\epsilon_2 = \hat{\epsilon}_2$ and $\theta = \hat{\theta}$ such that the conditional density function of the parameters $\alpha, \beta, \epsilon_1, \epsilon_2, \theta$ given observation vector $\vec{r}$, i.e. $f(\epsilon_1, \epsilon_2, \theta|\vec{r})$, is maximum.** Such an estimator is called nondata-aided (NDA),while a data-aided (DA) estimator employs data estimates $\hat{\alpha}$ and $\hat{\beta}$ instead. We use Bayes formula [B22]

$$f(\epsilon_1, \epsilon_2, \theta|\vec{r}) = \frac{\langle f(\vec{r}|\alpha, \beta, \epsilon_1, \epsilon_2, \theta)\rangle\, f(\epsilon_1, \epsilon_2, \theta)}{f(\vec{r})} \tag{4.6}$$

where $< \cdot >$ represents the statistical average on data α, β. Since $f(\vec{r})$ is independent of ϵ_1, ϵ_2, and θ,the Bayes estimate of ϵ_1, ϵ_2, and θ might be restated as follows: **Choose $\epsilon_1 = \hat{\epsilon}_1$, $\epsilon_2 = \hat{\epsilon}_2$, and $\theta = \hat{\theta}$ such that the product $< f(\vec{r}|\alpha, \beta, \epsilon_1, \epsilon_2, \theta) > \cdot f(\epsilon_1, \epsilon_2, \theta)$ is maximum.** Usually, $f(\epsilon_1, \epsilon_2, \theta)$ is assumed to be uniformly distributed over the $(\epsilon_1, \epsilon_2, \theta)$ hyperplane,which leads to the following criterion: **Choose $\epsilon_1 = \hat{\epsilon}_1$, $\epsilon_2 = \hat{\epsilon}_2$, and $\theta = \hat{\theta}$ such that $< f(\vec{r}|\alpha, \beta, \epsilon_1, \epsilon_2, \theta) >$ is maximum.** We call this estimator maximum likelihood (ML),since the ML estimate corresponds mathematically to the limiting case of a maximum a posteriori (MAP) estimate in which the a priori knowledge approaches zero,i.e. ϵ_1, ϵ_2 and θ are uniformly distributed,[B22,p.65]. Note that many authors treat ϵ_1, ϵ_2, and θ as deterministic but unknown quantities in their ML approaches. $f(\vec{r}|\alpha, \beta, \epsilon_1, \epsilon_2, \theta)$ is an a posteriori probability and very often such an estimator is (incorrectly) called MAP. Later, we use the knowledge of ϵ_1, ϵ_2, and θ accumulated over corresponding observation intervals to develop improved estimator-detectors,which will be called MAP.

4.2. MAXIMUM LIKELIHOOD (ML) APPROACH

We maximize the likelihood function

$$\Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta) = \ln[\langle f(\vec{r}|\alpha,\beta,\epsilon_1,\epsilon_2,\theta)\rangle], \tag{4.7}$$

denote the corresponding ϵ_1, ϵ_2, and θ with $\hat{\epsilon}_1, \hat{\epsilon}_2$, and $\hat{\theta}$,and name the respective estimator as ML. The analysis proceeds as follows: Conditioned on α, β, ϵ_1, ϵ_2, and θ, the random vector $\vec{r}$ is N-dimensional Gaussian with mean value which is equal to the vector $\vec{s}$,i.e.

$$f(\vec{r}|\alpha,\beta,\epsilon_1,\epsilon_2,\theta) = \frac{\exp[-(1/2)\,(\vec{r}-\vec{s})^T\,K_n^{-1}(N)\,(\vec{r}-\vec{s})]}{(2\pi)^{N/2}\,|K_n(N)|^{1/2}} \tag{4.8}$$

Quantity $K_n(N) = I(N)N_oB_{IF}$ is the noise covariance matrix, $I(N)$ is the N-dimensional identity matrix,superscript T denotes transpose, and dependence of $f(\vec{r}|\alpha,\beta,\epsilon_1,\epsilon_2,\theta)$ on α,β,ϵ_1,ϵ_2, and θ occurs in the vector $\vec{s}$. A more complex channel might be represented by a complex matrix $K(N)$, while a corresponding equalizer will be represented by its inverse $K^{-1}(N)$. However, analysis of such a system is out of the scope of this contribution. Since the components of vector $\vec{r}$ are mutually independent

$$\begin{aligned} f(\vec{r}|\alpha,\beta,\epsilon_1,\epsilon_2,\theta) = (2\pi N_oB_{IF})^{-N/2} \exp\left(-\frac{1}{N_o}\sum_{k=1}^{N}|r_k|^2\Delta t\right) \\ \times \exp\left(-\frac{1}{N_o}\sum_{k=1}^{N}|s_k|^2\Delta t + \frac{2}{N_o}\sum_{k=1}^{N} r_k s_k \Delta t\right) \end{aligned} \tag{4.9}$$

where $r_k = r(t_k)$ and $s_k = s(t_k)$. r_k does not depend on $\alpha, \beta, \epsilon_1, \epsilon_2$, and θ and might be absorbed within a constant. Very often, $s(t)$ has been assumed to be a constant envelope process and a maximization of the terms in the last sum of (4.9) followed. This is valid for a rather limited number of systems such as nonfiltered QPSK, MSK, and MPSK. Any filtering of such a system introduces envelope fluctuations, which require an equalization to be employed at the receiver end. Here,we do not consider problems related to the filtering and equalization. However,an envelope fluctuation is inherited within the elementary structure of the multilevel and overlapped schemes and this must be taken into consideration. We explore this problem in Section 4.2.2. In this section a constant envelope process is assumed. Using (4.9), the likelihood function $\Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta)$ might be written as

$$\Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta) = \sum_{V} \ln \sum_{i,j} e^{\Gamma} + C_1 \tag{4.10}$$

where summation over V denotes estimation on interval $[k-(V+2), k-1]$, summation over i, j denotes averaging over all possible data patterns $\{\alpha_i, \beta_j\}$, and Γ will be described in (4.13), and C_1 is a constant which includes the a priory probabilities of the sequencies α and β. Necessary but not sufficient conditions that must be satisfied by ϵ_1, ϵ_2, and θ in order to be an ML estimate are

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \theta}\Big|_{\theta=\hat{\theta},\, \epsilon_1=\hat{\epsilon}_1,\, \epsilon_2=\hat{\epsilon}_2} = 0 \qquad (4.11a)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \epsilon_1}\Big|_{\theta=\hat{\theta},\, \epsilon_1=\hat{\epsilon}_1,\, \epsilon_2=\hat{\epsilon}_2} = 0 \qquad (4.11b)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \epsilon_2}\Big|_{\theta=\hat{\theta},\, \epsilon_1=\hat{\epsilon}_1,\, \epsilon_2=\hat{\epsilon}_2} = 0 \qquad (4.11c)$$

In addition to (4.11a-c) the following conditions apply

$$\begin{vmatrix} \partial^2\Lambda/\partial\epsilon\partial\theta & \partial^2\Lambda/\partial\epsilon^2 \\ \partial^2\Lambda/\partial\theta^2 & \partial^2\Lambda/\partial\epsilon\partial\theta \end{vmatrix} < 0. \qquad (4.11d)$$

Λ has a maximum if $\partial^2\Lambda/\partial\epsilon^2 < 0$ and $\partial^2\Lambda/\partial\theta^2 < 0$ at (ϵ_o, θ_o)-this corresponds to a stable and correct (desirable) lock point of a recovery loop. Λ has a minimum if $\partial^2\Lambda/\partial\epsilon^2 > 0$ $\partial^2\Lambda/\partial\theta^2 > 0$ at (ϵ_o, θ_o)-this corresponds to a false lock point of a recovery loop. Here we have tacitly assumed that the function $\Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)$ is differentiable over all $\{\epsilon_1, \epsilon_2, \theta\}$,i.e. the first and second derivatives with respect to the $\epsilon_1, \epsilon_2, \theta$ exist. However,this might not be true, in general, and any singularity must be taken into consideration. As an example:The derivative of a rectangular pulse must be precisely defined for $t = +T_S/2$ as $(a_k - a_{k+1})/2$; the derivative of $++$ $(--)$ sequence of MSK,(Hann1 pulse) has a discontinuity in the vicinity of $t = T_S/2$, i.e. the step from -1 to +1 (+1 to -1), and must be defined as zero for $t = T_S/2$ for both cases;etc. Then, (4.11a-c) become

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \theta} = \sum_V \frac{\sum_{i,j}(\partial\Gamma/\partial\theta)\, e^{\Gamma}}{\sum_{i,j} e^{\Gamma}} \qquad (4.12a)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \epsilon_1} = \sum_V \frac{\sum_{i,j}(\partial\Gamma/\partial\epsilon_1)\, e^{\Gamma}}{\sum_{i,j} e^{\Gamma}} \qquad (4.12b)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \epsilon_2} = \sum_V \frac{\sum_{i,j}(\partial\Gamma/\partial\epsilon_2)\, e^{\Gamma}}{\sum_{i,j} e^{\Gamma}} \qquad (4.12c)$$

By using (4.1 and 4.8), a nondata-aided approach gives

$$\Gamma = -\frac{1}{N_o}\int [P_A x^2(V,\epsilon_1) + P_B y^2(V,\epsilon_2)] + [P_A x^2(V,\epsilon_1) - P_B y^2(V,\epsilon_2)]\cos 2(2\pi f_c t + \theta)\,dt$$
$$+ \frac{2}{N_o}\int [\sqrt{P_A} r(t) x(V,\epsilon_1)\cos(2\pi f_c t + \theta) + \sqrt{P_B} r(t) y(V,\epsilon_2)\sin(2\pi f_c t + \theta)]\,dt \quad (4.13)$$

$$\frac{\partial \Gamma}{\partial \theta} = \frac{1}{N_o}\int [P_A x^2(V,\epsilon_1) - P_B y^2(V,\epsilon_2)]\sin 2(2\pi f_c t + \theta)\,dt$$
$$- \frac{2}{N_o}\int [\sqrt{P_A} r(t) x(V,\epsilon_1)\sin(2\pi f_c t + \theta) - \sqrt{P_B} r(t) y(V,\epsilon_2)\cos(2\pi f_c t + \theta)]\,dt \quad (4.14)$$

$$\frac{\partial \Gamma}{\partial \epsilon_1} = -\frac{P_A}{N_o}\int x(V,\epsilon_1)\frac{\partial x(V,\epsilon_1)}{\partial \epsilon_1}[1 + \cos 2(2\pi f_c t + \theta)]\,dt$$
$$+ \frac{2\sqrt{P_A}}{N_o}\int r(t)\frac{\partial x(V,\epsilon_1)}{\partial \epsilon_1}\cos(2\pi f_c t + \theta)\,dt + L_{AS} + L_{AC} \quad (4.15)$$

$$\frac{\partial \Gamma}{\partial \epsilon_2} = -\frac{P_B}{N_o}\int y(V,\epsilon_2)\frac{\partial y(V,\epsilon_2)}{\partial \epsilon_2}[1 - \cos 2(2\pi f_c t + \theta)]\,dt$$
$$+ \frac{2\sqrt{P_B}}{N_o}\int r(t)\frac{\partial y(V,\epsilon_2)}{\partial \epsilon_2}\sin(2\pi f_c t + \theta)\,dt + L_{BS} + L_{BC} \quad (4.16)$$

where the summation over k is approximated with an integral

$$\sum_{k=1}^{N} r(t_k)\, w(t - iT_s + \epsilon) \begin{matrix}\cos\\ \sin\end{matrix}(2\pi f_c t + \theta)\,\Delta t =$$
$$\int_{-T_s/2+iT_s-\epsilon T_s/2}^{+T_s/2+iT_s-\epsilon T_s/2} r(t_k)\, w(t + \epsilon) \begin{matrix}\cos\\ \sin\end{matrix}(2\pi f_c t + \theta)\,dt, \qquad S = A, B \quad (4.17)$$

and for convenience the integral limits are omitted. $L_{AS}, L_{AC}, L_{BS}, L_{BC}$ are terms due to Leibniz' rule given as

$$L_{AS} = \frac{2\sqrt{P_A}}{N_o}\frac{T_A}{\pi} r(t)\cos(2\pi f_c t + \theta)$$

$$[x(iT_A - T_A/2 - \epsilon_1 T_A/\pi) - x(iT_A + T_A/2 - \epsilon_1 T_A/\pi)]\, p_A \qquad (4.18a)$$

$$L_{AC} = \frac{P_A}{N_o}\,[x^2(iT_A - T_A/2 - \epsilon_1 T_A/\pi) - x^2(iT_A + T_A/2 - \epsilon_1 T_A/\pi)]\, p_A \qquad (4.18b)$$

$$L_{BS} = \frac{2\sqrt{P_B}}{N_o}\frac{T_B}{\pi} r(t)\sin(2\pi f_c t + \hat{\theta})$$
$$[y(jT_B - T_B/2 - \epsilon_2 T_B/\pi) - y(jT_B + T_B/2 - \epsilon_2 T_B/\pi)]\, p_B \qquad (4.19a)$$

$$L_{BC} = \frac{P_B}{N_o}\,[y^2(jT_B - T_B/2 - \epsilon_2 T_B/\pi) - y^2(jT_B + T_B/2 - \epsilon_2 T_B/\pi)]\, p_B \qquad (4.19b)$$

where p_A and p_B are the probabilities of symbol transitions in the A and B channel, respectively. Note that for $\epsilon_i > 0$, two symbols, the k-th and $(k-1)$-th,are involved in (4.18a–4.19b).

Equations (4.12a-c) are highly nonlinear, and explicit solutions for ϵ_1, ϵ_2, and θ seem to be, at least, very complex if not impossible. Therefore, an ML estimator might not be practically realizable in general. Furthermore, these equations are not independent,which suggests that a joint carrier and timing(s) solution might offer better results than separate solutions. In the following, we concentrate on some simplifications, which might lead to easier practical realizations. The integrals in (4.14–4.16) which contain double frequency terms equal zero if the carrier frequency f_c is an integer multiple of data rates R_A, R_B. They are negligible if $f_c \gg R_A, R_B$ and will be ignored in our analysis. Usually, the signal set in each quadrature channel is symmetric. Then, (4.12a-c) might be written as

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \theta} = 0 = \sum_V \frac{N(\theta)}{D_o} \qquad (4.20a)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \epsilon_1} = 0 = \sum_V \frac{N(\epsilon_1)}{D_o} \qquad (4.20b)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1, \epsilon_2, \theta)}{\partial \epsilon_2} = 0 = \sum_V \frac{N(\epsilon_2)}{D_o} \qquad (4.20c)$$

$$D_o = \sum_{i,j} e^{\Gamma} \qquad (4.21a)$$

$$\Rightarrow \sum_{i,j} W_i \cosh(C_x)\, W_j \cosh(S_y) \qquad (4.21b)$$

In (4.21a), $i, j \in \{I\}$, and the Γ term is given in (4.13). However,due to symmetry of the signal set,summations over i, j in (4.21b) and the equations which follow extend over

nonnegative integers only.

$$N(\theta) = 4 \sum_{i,j} [-S_x \sinh(C_x) \cosh(S_y) + C_y \cosh(C_x) \sinh(S_y)] \tag{4.22}$$

$$N(\epsilon_1) = 4 \sum_{i,j} \Big([-\frac{P_A}{N_o} \int x(V,\epsilon_1) \frac{\partial x(V,\epsilon_1)}{\partial \epsilon_1} dt + L_{AC}] W_i \cosh(C_x) W_j \cosh(S_y) + [\frac{2\sqrt{P_A}}{N_o} \int \frac{\partial x(V,\epsilon_1)}{\partial \epsilon_1} r(t) \cos(2\pi f_c t + \theta)\, dt + L_{AS}] W_i \sinh(C_x) W_j \cosh(S_y) \Big) \tag{4.23}$$

$$N(\epsilon_2) = 4 \sum_{i,j} \Big([-\frac{P_B}{N_o} \int y(V,\epsilon_2) \frac{\partial y(V,\epsilon_2)}{\partial \epsilon_2} dt + L_{BC}] W_i \cosh(C_x) W_j \cosh(S_y) + [\frac{2\sqrt{P_B}}{N_o} \int \frac{\partial y(V,\epsilon_2)}{\partial \epsilon_2} r(t) \cos(2\pi f_c t + \theta)\, dt + L_{BS}] W_i \cosh(C_x) W_j \sinh(S_y) \Big) \tag{4.24}$$

where

$$W_i = \exp[-\frac{P_A i^2}{2N_o} \int x^2(V,\epsilon_1)\, dt] \tag{4.25a}$$

$$W_j = \exp[-\frac{P_B j^2}{2N_o} \int y^2(V,\epsilon_2)\, dt] \tag{4.25b}$$

are weighting factors proportional to the negative exponent of the signal-to-noise ratio (SNR) in the corresponding baseband levels,and

$$C_x = \frac{2\sqrt{P_A} i}{N_o} \int r(t) x(V,\epsilon_1) \cos(2\pi f_c t + \theta)\, dt \tag{4.26a}$$

$$C_y = \frac{2\sqrt{P_B} j}{N_o} \int r(t) y(V,\epsilon_2) \cos(2\pi f_c t + \theta)\, dt \tag{4.26b}$$

$$S_x = \frac{2\sqrt{P_A} i}{N_o} \int r(t) x(V,\epsilon_1) \sin(2\pi f_c t + \theta)\, dt \tag{4.26c}$$

$$S_y = \frac{2\sqrt{P_B} j}{N_o} \int r(t) y(V,\epsilon_2) \sin(2\pi f_c t + \theta)\, dt \tag{4.26d}$$

Leibnitz's terms and derivatives are now related to the nonnegative i, j.Then, (4.20a-c) suggest a closed-loop structure for the joint carrier phase estimate θ and timing estimates ϵ_1, ϵ_2

given in Fig.4.2. In this and following figures,timing information paths to the integrals and delay elements necessary for a proper operation of the loops are omitted for the reason of clarity. Note that the timing loop consists of two essential parts: the first one which represents the first continuous part in (4.15–4.16) and the second one which represents Leibniz's transitional parts (4.18a–4.19b). The transitional part only is the essence of a digital data transition tracking loop (DTTL) [B16]. The probability of no transition of MQAM schemes equals $1/L$,where L is the number of baseband levels,which suggests that this part might be essential for a TR device for multilevel schemes. The transitional part of rectangular schemes is independent of ϵ,whereas it increases as ϵ increases for MSK, an example of a nonrectangular scheme.

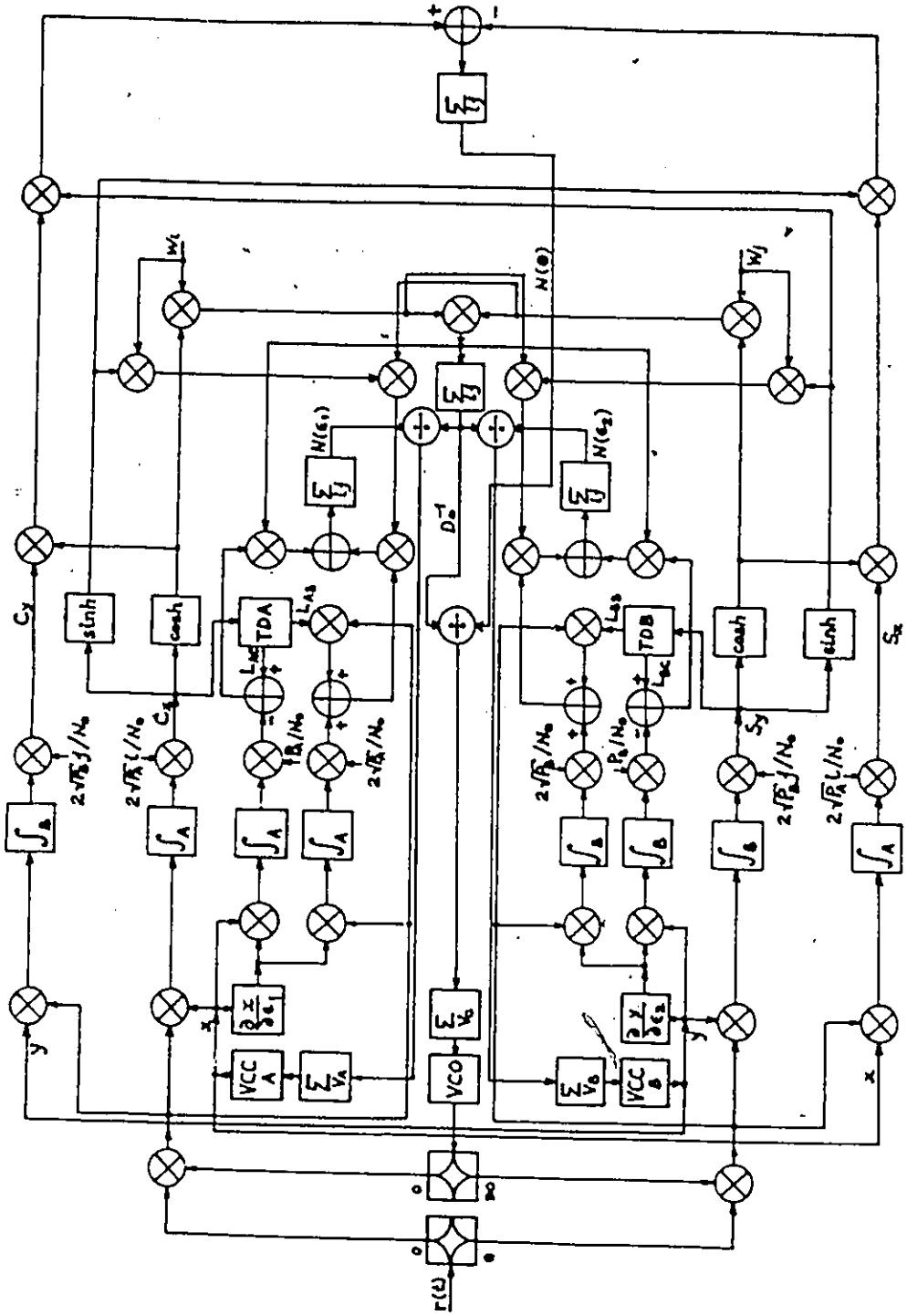


Figure 4.2. The joint phase and timing estimator for WQAM schemes, which corresponds to eqs. (4.11 a-c) in the text.

By approximating the continuous derivative by a finite difference, i.e.,

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta)}{\partial \epsilon} \doteq \frac{\Lambda(\epsilon+\Delta\epsilon/2)-\Lambda(\epsilon-\Delta\epsilon/2)}{\Delta\epsilon}$$

$$= \sum_V \frac{\ln\sum_{i,j}\exp(+\Delta\epsilon)-\ln\sum_{i,j}\exp(-\Delta\epsilon)}{\Delta\epsilon} \qquad (4.27a)$$

$$\Rightarrow \sum_V \frac{\ln\sum_{i,j}\cosh(+\Delta\epsilon)-\ln\sum_{i,j}\cosh(-\Delta\epsilon)}{\Delta\epsilon} \qquad (4.27b)$$

where $\{\pm\Delta\epsilon\}$ correspond to $\Gamma(\epsilon\pm\Delta\epsilon/2)$ respectively, an early-late gate loop [B3] is suggested. The summations in (4.27a) assume that $i,j \in \{I\}$ while the summations in (4.27b) assume nonnegative i,j.

The structure given in **Fig.4.2** seems to be complex, and the hyperbolic sine and hyperbolic cosine functions might not be practical to realize. The approximations $\sinh(x) \doteq x$, $\cosh(x) \doteq 1+x^2/2$ for small x (low SNR) and $\cosh(x) \doteq sign(x)\sinh(x)$ for large x (high SNR) might be employed to simplify the loop. However,we proceed in a different way. For 4-state 2x2 baseband levels schemes, (4.20a-c) reduce to

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta)}{\partial \theta} = 0 = \sum_V [-S_x\tanh(C_x)+C_y\tanh(S_y)] \qquad (4.28a)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta)}{\partial \epsilon_1} = 0 = \sum_V \Big(-\frac{P_A}{N_o}\int x(V,\epsilon_1)\frac{\partial x(V,\epsilon_1)}{\partial \epsilon_1}\,dt$$

$$+\Big[\frac{2\sqrt{P_A}}{N_o}\int \frac{\partial x(V,\epsilon_1)}{\partial \epsilon_1} r(t)\cos(2\pi f_c t+\theta)\,dt + L_{AS}\Big]\tanh(C_x)\Big) \qquad (4.28b)$$

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta)}{\partial \epsilon_2} = 0 = \sum_V \Big(-\frac{P_B}{N_o}\int y(V,\epsilon_2)\frac{\partial y(V,\epsilon_2)}{\partial \epsilon_2}\,dt$$

$$+\Big[\frac{2\sqrt{P_B}}{N_o}\int \frac{\partial y(V,\epsilon_2)}{\partial \epsilon_2} r(t)\sin(2\pi f_c t+\theta)\,dt + L_{BS}\Big]\tanh(S_y)\Big) \qquad (4.28c)$$

(4.28a-c) suggest a device as given in **Fig.4.3.** Without the TR part (i.e. 4.28b and 4.28c) and for rectangular signal shapes, this becomes a CR loop as given in [P127].

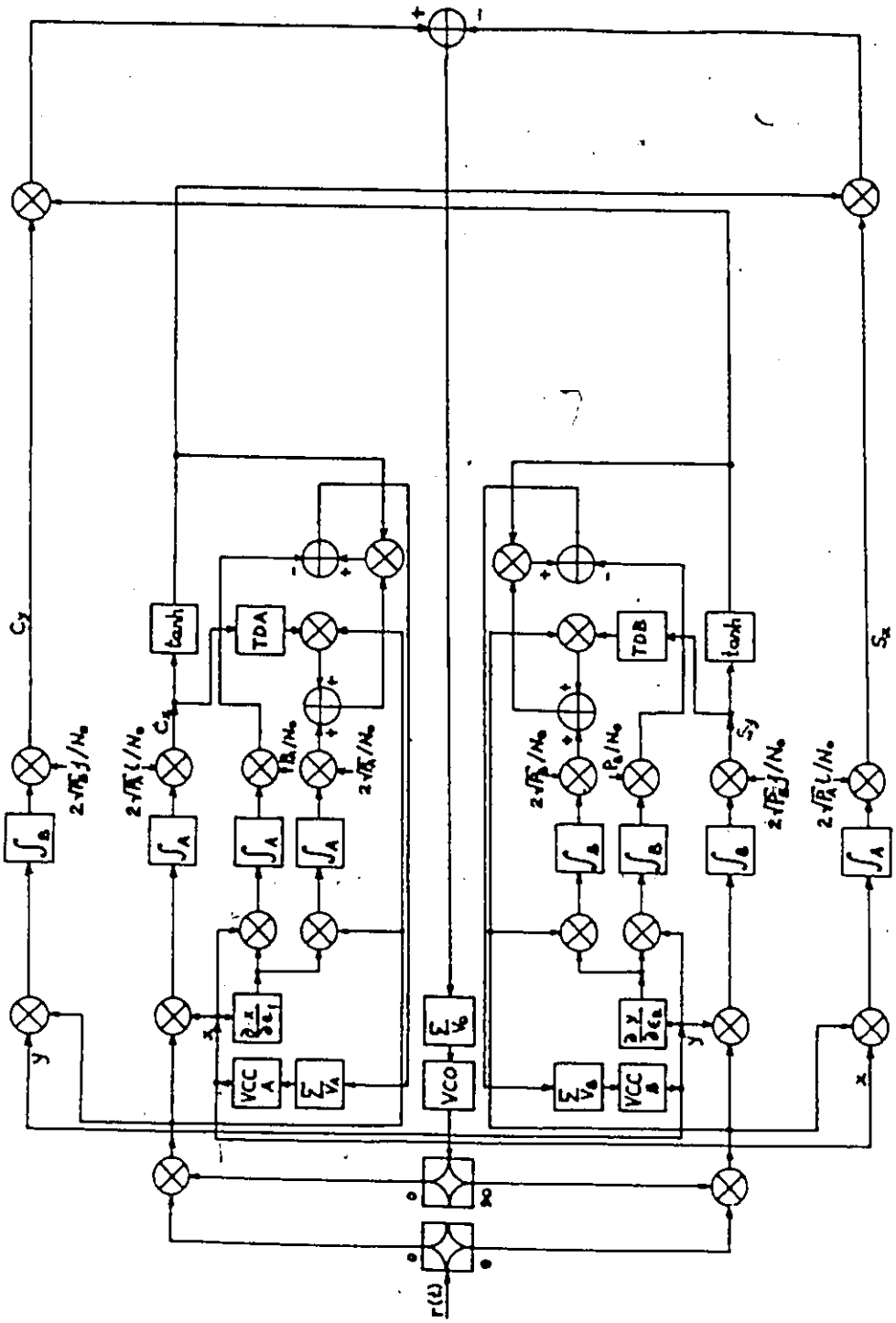


Figure 4.3. The joint phase and timing estimator for 4 state 2x2 baseband level WQAM schemes, which corresponds to eqs. (4.28 a-c) in the text.

We will not pursue unbalanced schemes further. However,note that QAM schemes with more than 4 states are balanced only on average i.e. the average power in the A channel is equal to the average power in the B channel,but they are not balanced on a per-symbol-basis. As an example,a 256QAM scheme exhibits the maximum imbalance of 15/1 when a baseband level of 1 in the A channel is accompanied with the level of 15 in the B channel and vice versa. Similarities between unbalanced schemes and MQAM schemes may be observed on the corresponding nonlinearity curves for unbalanced schemes [P127]-[P130] and MQAM schemes [P122]-[P125]. For 4 state balanced schemes with $P_A = P_B, R_A = R_B = 1/T_S, C_x = C_y, S_x = S_y$ and rectangular signal shapes, (4.28a-c) reduce to

$$\frac{\partial \Lambda(\vec{r}|\epsilon_1,\epsilon_2,\theta)}{\partial \theta} = 0 = \sum_V [-S_x \tanh(C_x) + C_x \tanh(S_x)] \tag{4.29a}$$

$$\begin{aligned}\frac{\partial \Lambda(\vec{r}|\epsilon_1,\epsilon_2;\theta)}{\partial \epsilon} = 0 = \sum_V \Big(& -\sqrt{P_A}\int x(V,\epsilon)\frac{\partial x(V,\epsilon)}{\partial \epsilon}\,dt \\ & + \tanh(C_x)\int \frac{\partial x(V,\epsilon)}{\partial \epsilon} r(t)\cos(2\pi f_c t+\theta)\,dt \\ & + \tanh(S_x)\int \frac{\partial y(V,\epsilon)}{\partial \epsilon} r(t)\sin(2\pi f_c t+\theta)\,dt \\ & + L_{AS}\tanh(C_x) + L_{BS}\tanh(S_x)\Big)\end{aligned} \tag{4.29b}$$

which suggests the device given in **Fig.4.4.** The nonlinearity $\tanh(x)$ might be approximated by $x - x^3/3! + \ldots$ for low SNRs and by $sign(x)$ for high SNRs. The high SNR loop approximation without the timing part is called a **DFCRL** since decisions on α, β,i.e. $\hat{\alpha}, \hat{\beta}$ are used to (partly) wipe off the data. This type of loop is often called data-aided. However, a DA approach is not used here and this solution is, therefore, not necessary optimal in a DA sense. The DA approach calls for data estimates $\hat{\alpha}$ and $\hat{\beta}$ in (4.6–4.7). Both $\hat{\alpha}$ and $\hat{\beta}$ depend on ϵ_1, ϵ_2, in a yet unknown way and a strict DA approach seems to be rather difficult. Later we discuss this dependence and a few improved circuits are suggested.

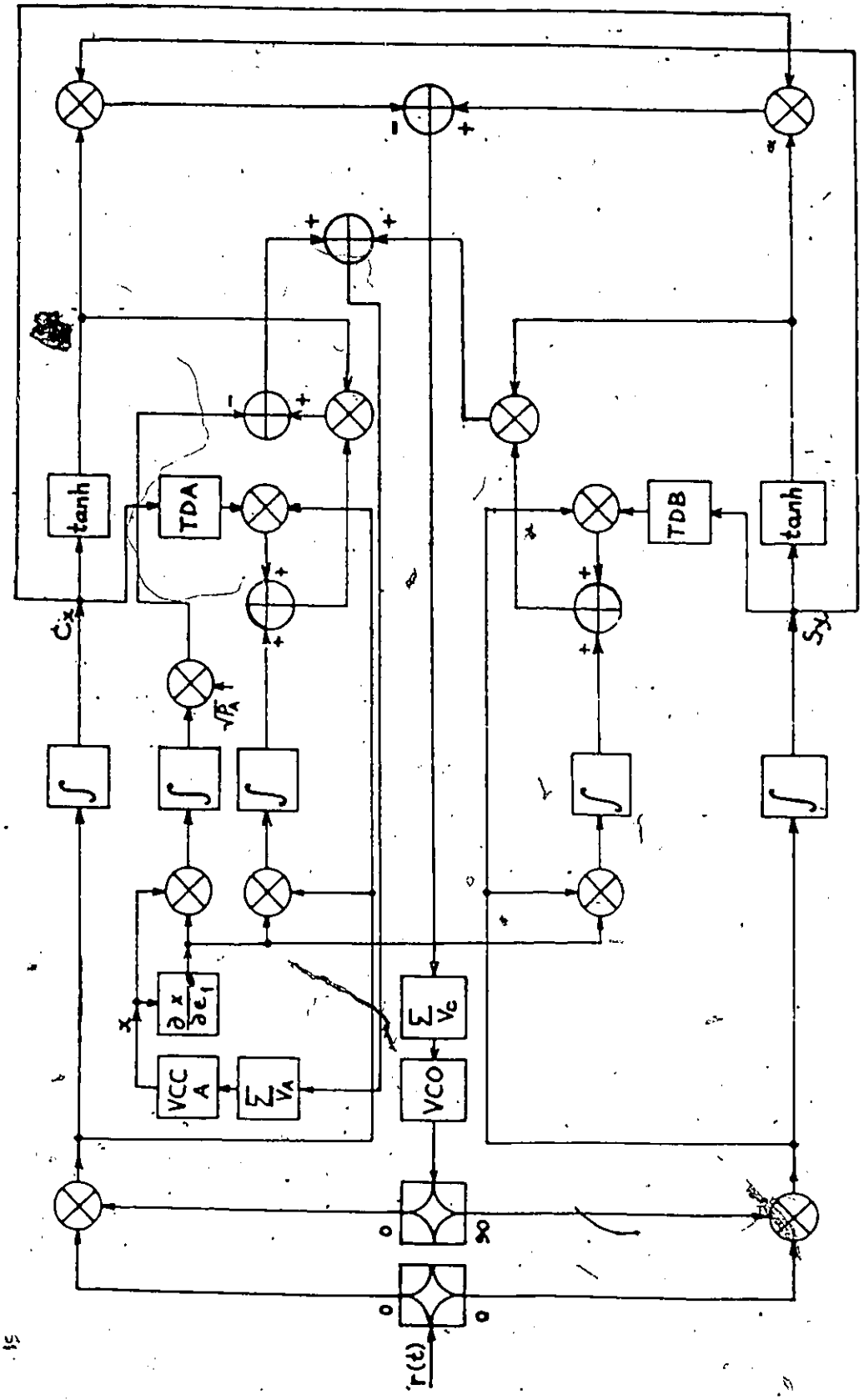


Figure 4.4. The joint phase and timing estimator for QPSK schemes, which corresponds to eqs. (4.29 a-c) in the text.

MSK is an example of schemes with a nonrectangular signal shape. By using (4.26a-d) and (4.28a-c) the phase and timing estimation equations become

$$\frac{\partial\Lambda(\vec{r}|c_1,\epsilon_2,\theta)}{\partial\theta} = 0 = \sum_V \Big(-\int \cos(\frac{\pi t}{T}+\epsilon) r(t) \sin(2\pi f_c t+\theta)\, dt$$
$$\times \tanh[\frac{2\sqrt{P_A}}{N_o}\int \cos(\frac{\pi t}{T}+\epsilon) r(t)\cos(2\pi f_c t+\theta)\, dt]$$
$$+\int \sin(\frac{\pi t}{T}+\epsilon) r(t)\cos(2\pi f_c t+\theta)\, dt$$
$$\times \tanh[\frac{2\sqrt{P_B}}{N_o}\int \sin(\frac{\pi t}{T}+\epsilon) r(t)\sin(2\pi f_c t+\theta)\, dt]\Big) \qquad (4.30a)$$

$$\frac{\partial\Lambda(\vec{r}|c_1,\epsilon_2,\theta)}{\partial\epsilon} = 0 = \sum_V \Big(-[\int \sin(\frac{\pi t}{T}+\epsilon) r(t)\cos(2\pi f_c t+\theta)\, dt + L_{AS}]$$
$$\times \tanh[\frac{2\sqrt{P_A}}{N_o}\int \cos(\frac{\pi t}{T}+\epsilon) r(t)\cos(2\pi f_c t+\theta)\, dt]$$
$$+[\int \cos(\frac{\pi t}{T}+\epsilon) r(t)\sin(2\pi f_c t+\theta)\, dt + L_{BS}]$$
$$\times \tanh[\frac{2\sqrt{P_B}}{N_o}\int \sin(\frac{\pi t}{T}+\epsilon) r(t)\sin(2\pi f_c t+\theta)\, dt]\Big) \qquad (4.30b)$$

The corresponding joint estimator is given in **Fig.4.5.** Note the similarity of our estimator with that given in [P138,Fig.3]. Our scheme uses additional transition detectors TDA and TDB to improve timing performance by accounting for the Leibniz terms. Since MSK is using $\cos(x)$ and $\sin(x)$ pulse shapes in A and B channels respectively,the derivatives suggested in (4.28a-c) are realized by employing the respective crossarm signals. In a data-aided approach, at high SNR, $\tanh(x) \doteq sign(x)$ is used. The extension of this concept to other WQAM schemes is straightforward.

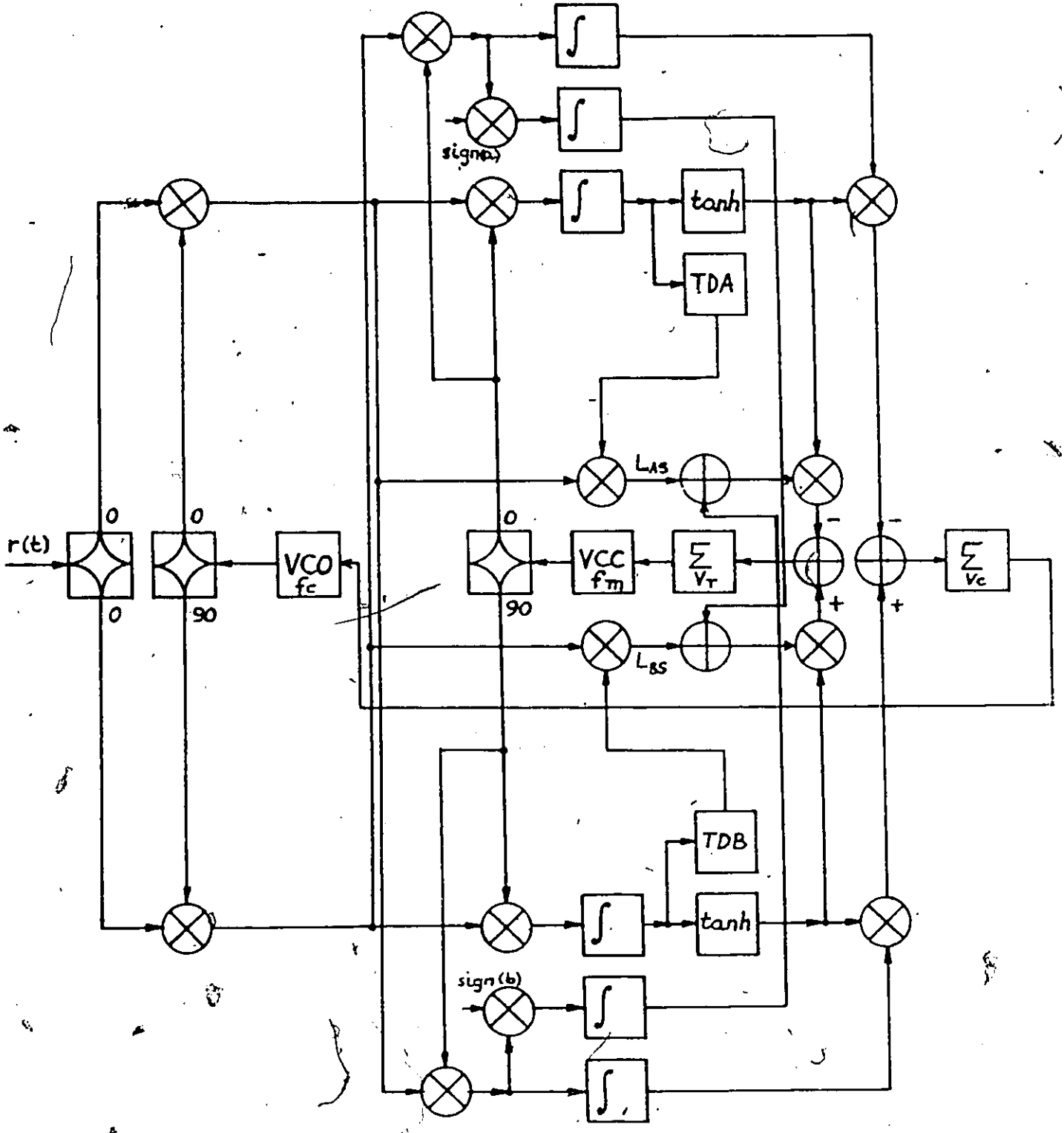


Figure 4.5. The joint phase and timing estimator for MSK schemes, which corresponds to eqs. (4.30 a-c) in the text.

Now, we proceed with the CR structures analysis. When schemes with more than 4 states are employed, a term within the second sum in (4.9) suggests that a quantizer

$$Q_k(x) = \begin{cases} kd, & \text{for } (k-1)d \le x < (k+1)d; \\ (L-1)d, & \text{for } (L-2)d \le x < \infty; \\ -(L-1)d, & \text{for } -\infty < x \le -(L-2)d. \end{cases} \qquad k = 0 \pm 2, \pm 4, \ldots \pm (L-3); \tag{4.31}$$

be used as a decision device. This loop, Fig.4.6, which we call the classical DFCRL, serves as a reference to which all new schemes are compared. This type of loop with an integrate-sample-and-dump (ISD) circuit (box noted with an integral) within the loop arms is called active,while the other one, employing a lowpass filter instead, is called passive. We analyze the former one only and note that an active loop performs better than a passive one,at least for $\phi, \lambda \doteq 0$. Usually, an active loop with ideal sampling is assumed. Since this might not be a practical case, we analyze the loop performance in the presence of both CR uncertainty ϕ and TR uncertainty λ, which to the best of our knowledge has not been published in the open literature, yet.

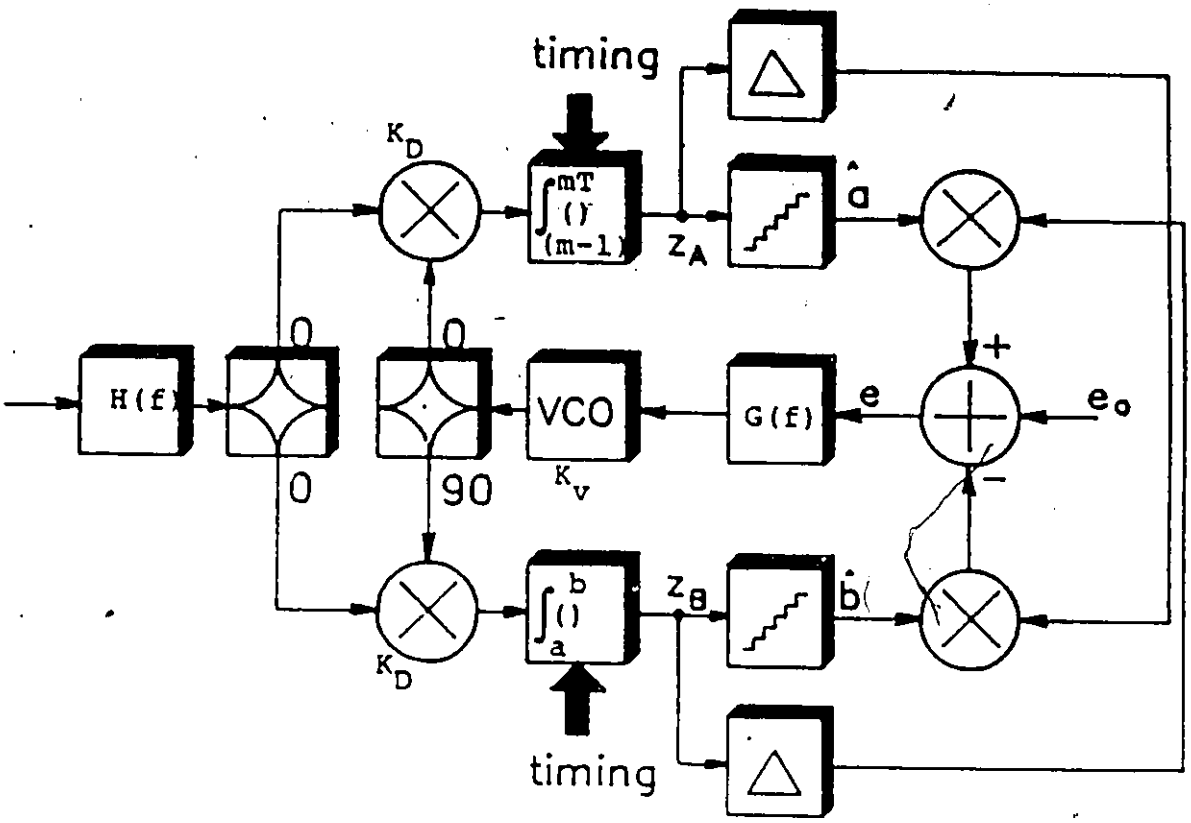


Figure 4.6. The classical decision feedback carrier recovery loop.

4.2.1. CLASSICAL DECISION FEEDBACK CR LOOP

The block diagram of the classical DFCRL is given in **Fig.4.** Combining both CR and TR uncertainties, † signal $z_A(t)$ at the decision device input is

$$z_A(t) = a_m w_A[t - (m-1)T - \epsilon]\cos\phi + b_n w_B[t-(n-1)T-\epsilon]\sin\phi + \nu_A \qquad (4.32)$$

where ν_A is the sample of Gaussian noise in the A channel. The conditional probability of error of an MQAM scheme is

$$U(\phi,\lambda) = P(e|\phi,\lambda) = P[e|\phi,\lambda,a_m = m = -(L-1)] + P[e|\phi,\lambda,a_m = m = +(L-1)] + P[e|\phi,\lambda,|a_m = m| < (L-1)] \qquad (4.33)$$

$$P[e|\phi,\lambda,a_m = m = \pm(L-1), a_{m-1} = u, b_n = n, b_{n-1} = v] = Q[\![\gamma\{-(L-2) + [(L-1)R_A(|\lambda|) + uR_A(1-|\lambda|)]\cos\phi + [nR_B(|\lambda|) + vR_B(1-|\lambda|)]\sin\phi\}]\!] \qquad (4.34)$$

$$P[e|\phi,\lambda,|a_m = m| < (L-1), a_{m-1} = u, b_n = n, b_{n-1} = v] = Q[\![\gamma\{-(m-1) + [mR_A(|\lambda|) + uR_A(1-|\lambda|)]\cos\phi + [nR_B(|\lambda|) + vR_B(1-|\lambda|)]\sin\phi\}]\!] + Q[\![\gamma\{+(m+1) - [mR_A(|\lambda|) + uR_A(1-|\lambda|)]\cos\phi - [nR_B(|\lambda|) + vR_B(1-|\lambda|)]\sin\phi\}]\!] \qquad (4.35)$$

The quantities $P(e|x,y,\ldots)$ are the conditional probabilities of error conditioned on $x,y,\ldots$ respectively,$Q[x]$ is the Gaussian probability function [B26,26.2.3],γ is the average signal-to-noise ratio at the decision device input,and $R_A(x)$ and $R_B(x)$ are the crosscorrelation functions of reference signal and data in A and B channel respectively, in the corresponding TR loops. To distinguish between different conditional probabilities,the conditional probability of error $P(e|\phi,\lambda)$ is called the **uncertainty diagram** $U(\phi,\lambda)$, the meaning and importance of which will be explained further, later on. The conditional probability of error of staggered schemes is given by (4.33-4.35) if $R_B(|\lambda|)$ and $R_B(1-|\lambda|)$ are replaced by $R_B(1/2+|\lambda|)$ and $R_B(1/2-|\lambda|)$, respectively.

The conditional probability of error performance $P(e|\phi)$ of MQAM schemes, conditioned on a CR uncertainty ϕ is summarized in **Fig.4.7.** For comparison purposes, a probability of

† We assume that the receiving filter H(f) is wide enough to cause no significant pulse reshaping,and narrow enough to allow the narrow-band representation of additive white Gaussian noise of the channel. The loop bandwidth is much narrower than the data bandwidth. The loop is initially locked. As a consequence,the phase process ϕ in the loop varies much more slowly than the signal and the noise process correlation time is much less than the length of the symbol interval,i.e. the noise is practically white.

error of 10^{-10} is assumed. This corresponds to a typical non-faded value in terrestrial radio. To emphasize threshold crossings,the absolute value of arguments within the $Q(x)$ function, i.e. $Q(|x|)$, is used. The corresponding angles represent phase errors which cause threshold boundary crossings,i.e. erroneous decisions.
We will attempt to relate these theoretically obtained curves to some practical cases as follows. As an example, an absolute CR phase uncertainty of 2 degrees causes a minimal degradation of the conditional probability of error, if BPSK is employed, see **Fig.4.7**. However, the same 2 degrees uncertainties causes the degradation from 10^{-10} to 10^{-3} if 256QAM scheme is concerned. The corresponding average probability of error P_e depends further on the probability density function $p(\phi)$, which is dictated by the loop design. For this type of loop and an MQAM signal constellation, an exact expression for $p(\phi)$ is very difficult to obtain, and — in our best knowledge — is not known. As a consequence, P_e is not known either. Therefore, for the practical purposes we will give some examples. If $p(\phi)$ is uniformly distributed, a receiver does not have the information about the carrier phase and coherent detection is not possible. If $p(\phi) = \delta(\phi)$, where δ is the delta function, the conditional probability of error $P(e|\phi)$ equals the average P_e and the ordinate in **Fig.4.7** reads as P_e. This occurs when the impairment dynamic is faster than the loop bandwidth, but slower than the data bandwidth, i.e. the phase error is virtually constant over a number of data symbols. A practical loop emphasizes the angles close to zero degrees. Therefore, P_e results better than those given in **Fig.4.7** might be expected. **Fig.4.7** gives an estimate of maximum tolerable imperfections of the loop components, such as phase detector, amplifier and filter responses, d.c. wandering, etc. Obviously, as the number of levels (states) increases, these imperfections cause an increased degradation in the P_e performance. In Section 4.3 we search for loops which are less prone to these effects. There, the P_e performance of the classical DFCRL is given and compared with new improved loops.

The L-ary pulse amplitude modulation,LPAM (M=L×L),conditional probability of error performance $P(e|\lambda)$ is summarized in **Fig.4.8**. To emphasize threshold crossings the absolute value of arguments within the $Q(x)$ function, i.e. $Q(|x|)$, is used. The corresponding fractions represent timing errors,normalized to the symbol duration,which cause erroneous decisions. A timing uncertainty of 1/30 of the symbol duration causes a performance degradation in probability of error from 10^{-10} to 10^{-9} in a two level baseband signal modulation scheme (BPSK,QPSK),but an erroneous decision in a 16 (256QAM) or higher level scheme.
The meaning of this figure is similar to that of **Fig.4.7**.

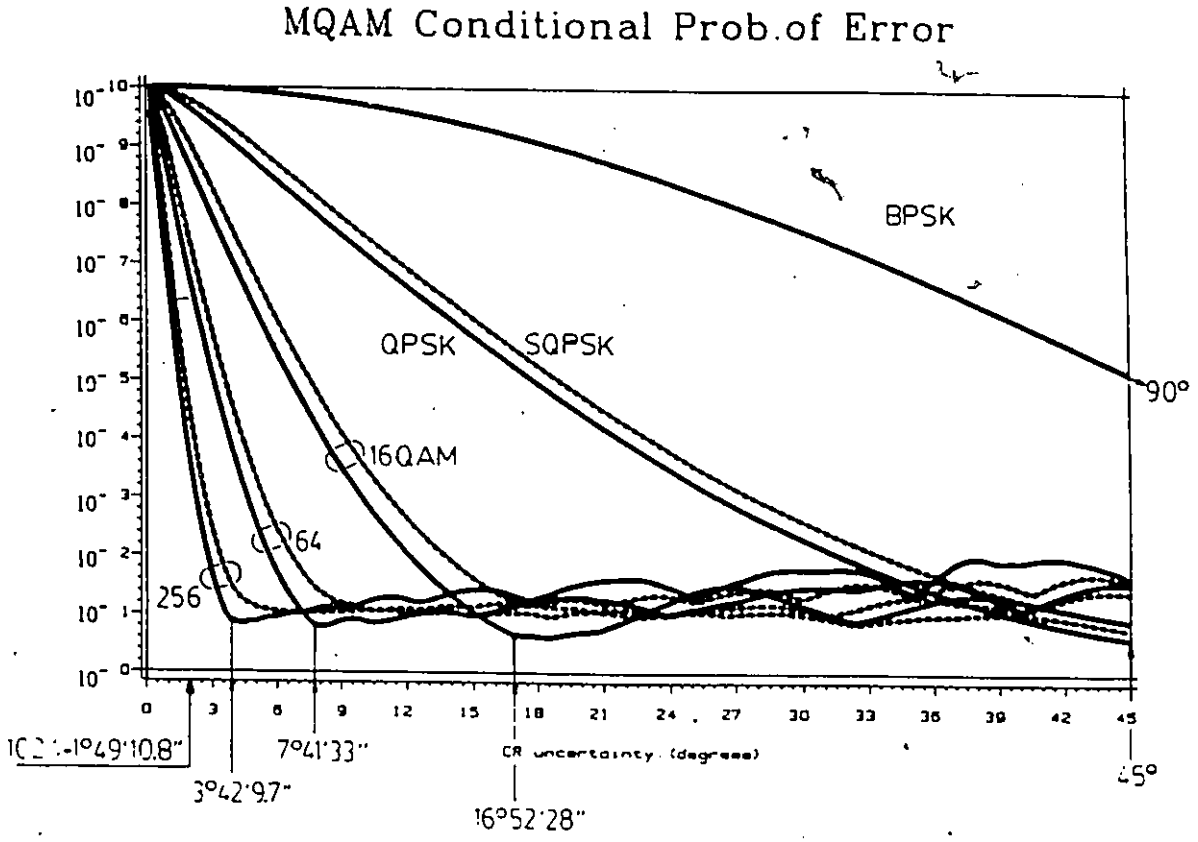


Figure 4.7. The $P(e|\phi)$ performance of MQAM schemes. The performance of 10^{-10} is taken as the reference. To emphasize threshold crossings, the absolute value of arguments within the $Q(x)$ function, i.e. $Q(|x|)$, is used.

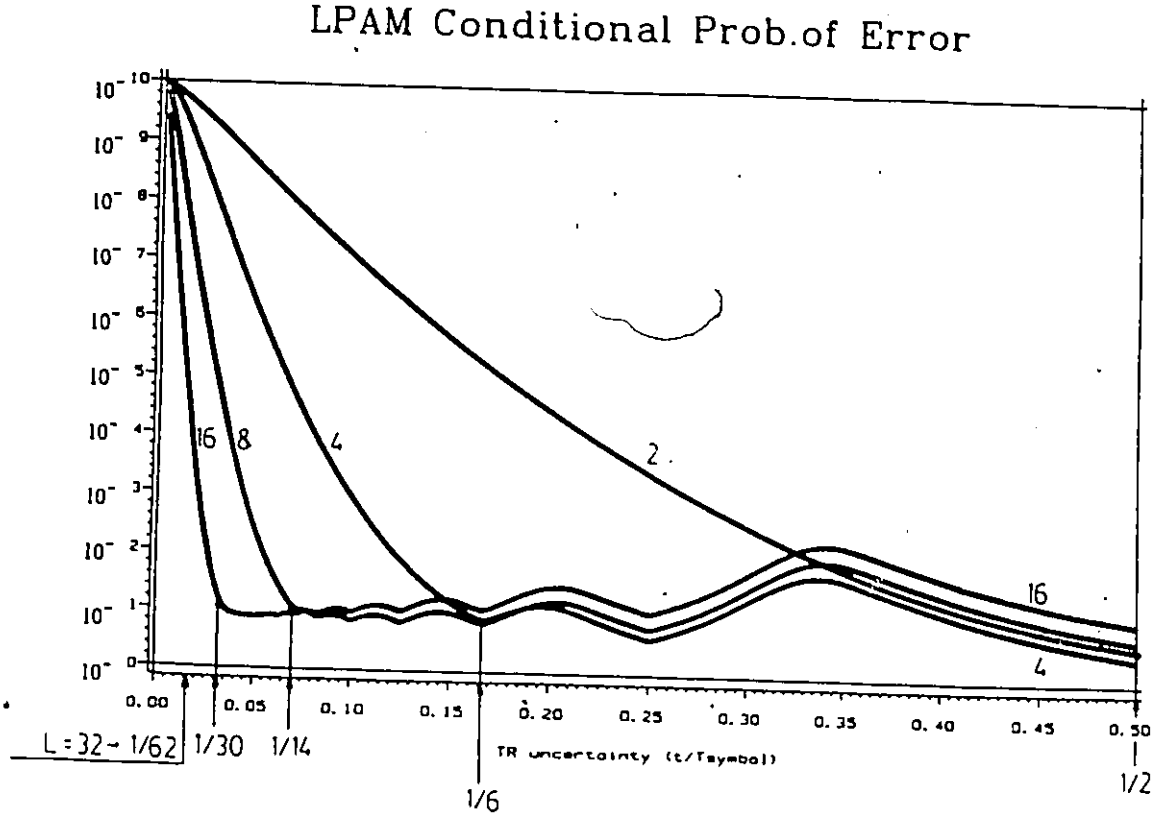


Figure 4.8. The $P(e|\lambda)$ performance of the LPAM schemes. The performance of 10^{-10} is taken as the reference. To emphasize threshold crossings, the absolute value of arguments within the $Q(x)$ function, i.e. $Q(|x|)$, is used.

The combined effect of the phase and timing uncertainties on the system performance is given by (4.33–4.35). To get a better understanding of the nature of (4.33–4.35), we plotted them using two different forms:the **uncertainty diagram (UD)** $U(\phi, \lambda)$ and,the equal probability of error or the **isoper curves.** To emphasize threshold crossings, $Q(x)$ is sometimes replaced by $Q(|x|)$. $U(\phi, \lambda)$ exhibits a periodicity of 2π radians, while $U(|\phi, \lambda|)$ of the BPSK and QAM schemes have periods of π and $\pi/2$, respectively. The UD and isoper curves of the 256QAM scheme are given as an example in **Fig.4.9a-b.** These figures show a high sensitivity to any phase and timing uncertainty assuming that a classical DFCRL is used. The UD and isoper curves of the BPSK, 4PAM,QPSK,SQPSK and 16QAM schemes are given in [P117]. Similar curves for four level modulation schemes are generated by Harris and Kristiansen [P118] using a computer simulation technique. Further results are given in [P119]–[P121].

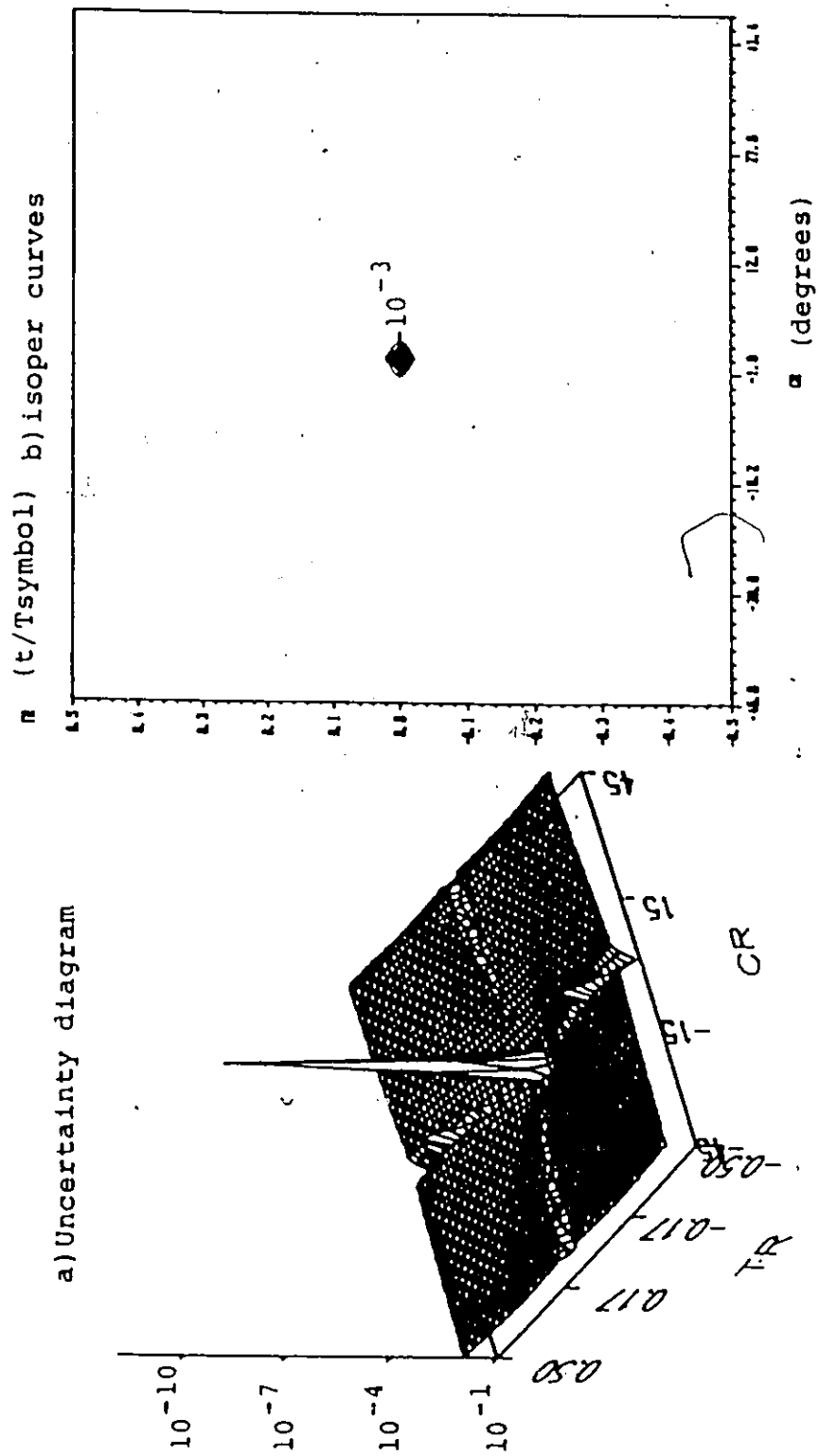


Figure 4.9. The 256QAM uncertainty diagram. The performance of 10^{-10} is taken as the reference.

More detailed analysis of the loop behavior including the probability density function, average probability of error,slipping rate, and acquisition properties is out of the scope of this presentation. However,most of these quantities, including $p(\phi,\lambda)$ are functions of a nonlinearity surface which will be briefly assessed. The integro-differential equation of the loop is

$$\dot{\phi}(t) = \dot{\theta}(t) - K_o F(p)\, e^{-p\Delta}\, P_{av} \left([-g_1(\phi,\lambda)\sin\phi + g_2(\phi,\lambda)\cos\phi] + \nu\sqrt{N_e(\phi,\lambda)N_o} \right) \quad (4.36)$$

where $P_{av} = 2(M-1)/3$ is the average signal power, the dot represents the derivative with respect to the time variable t, $F(p)$ is the transfer function of the loop filter, K_o is the equivalent loop gain, Δ is the delay within the loop,functions g_1, g_2 and N_e are

$$g_1(\phi,\lambda) = < a\hat{a} + b\hat{b} | \phi,\lambda > / P_{av} \quad (4.37)$$

$$g_2(\phi,\lambda) = < \hat{a}b - \hat{b}a | \phi,\lambda > / P_{av} \quad (4.38)$$

$$N_e(\phi,\lambda) = < \hat{a}^2 + \hat{b}^2 | \phi,\lambda > / P_{av} \quad (4.39)$$

ν is a Gaussian random variable,and $< \cdot >$ represents the time average. The function

$$h(\phi,\lambda) = 2\,\frac{-g_1(\phi,\lambda)\sin\phi + g_2(\phi,\lambda)\cos\phi}{N_e(\phi,\lambda)N_o} \quad (4.40)$$

is termed the normalized **nonlinearity surface** or (two dimensional) restoring force of the CR loop,and becomes the nonlinearity S curve when $\lambda \to 0$. An integral of $-h(\phi,\lambda)$ represents a potential function. The nonlinearity surface QPSK has period of $\pi/2$ radians, **Fig.4.10.** However,the period of staggered QPSK is extended to π radians, **Fig.4.11.** The nonlinearity surface of the SQPSK scheme has a favorable shape in a comparison with the shape of the QPSK scheme, because of an extended linear region near the origin ($\phi = 0$, $\lambda = 0$) and the larger period. Therefore, SQPSK will perform better (will have lower P_e) than QPSK in the presence of phase uncertainties. Note that $p(\phi,\lambda)$ depends on $h(\phi,\lambda)$,i.e.

$$p(\phi,\lambda) = C\,\frac{exp[-\int_\phi \int_\lambda h(\phi,\lambda)\,dx\,dy]}{N_e(\phi,\lambda)\,N_o} + \text{Ito terms} \quad (4.41)$$

$p(\phi,\lambda)$ is a nonelementary function due to a nonelementary integral in its exponent. By using (4.36–4.41),solutions of (4.2a-b) might be obtained, in principal, by a numerical integration. However, for more than 4 state schemes,an excessive amount of computer time is required. Nevertheless, for comparison purposes, the nonlinearity surface $h(\phi,\lambda)$ and uncertainty diagram $U(\phi,\lambda)$ are better figures of merit than the corresponding average probability of error P_e. An ideal $U(\phi,\lambda)$ is independent of ϕ and λ, while $h(\phi) = \sin\phi$ is a good choice.

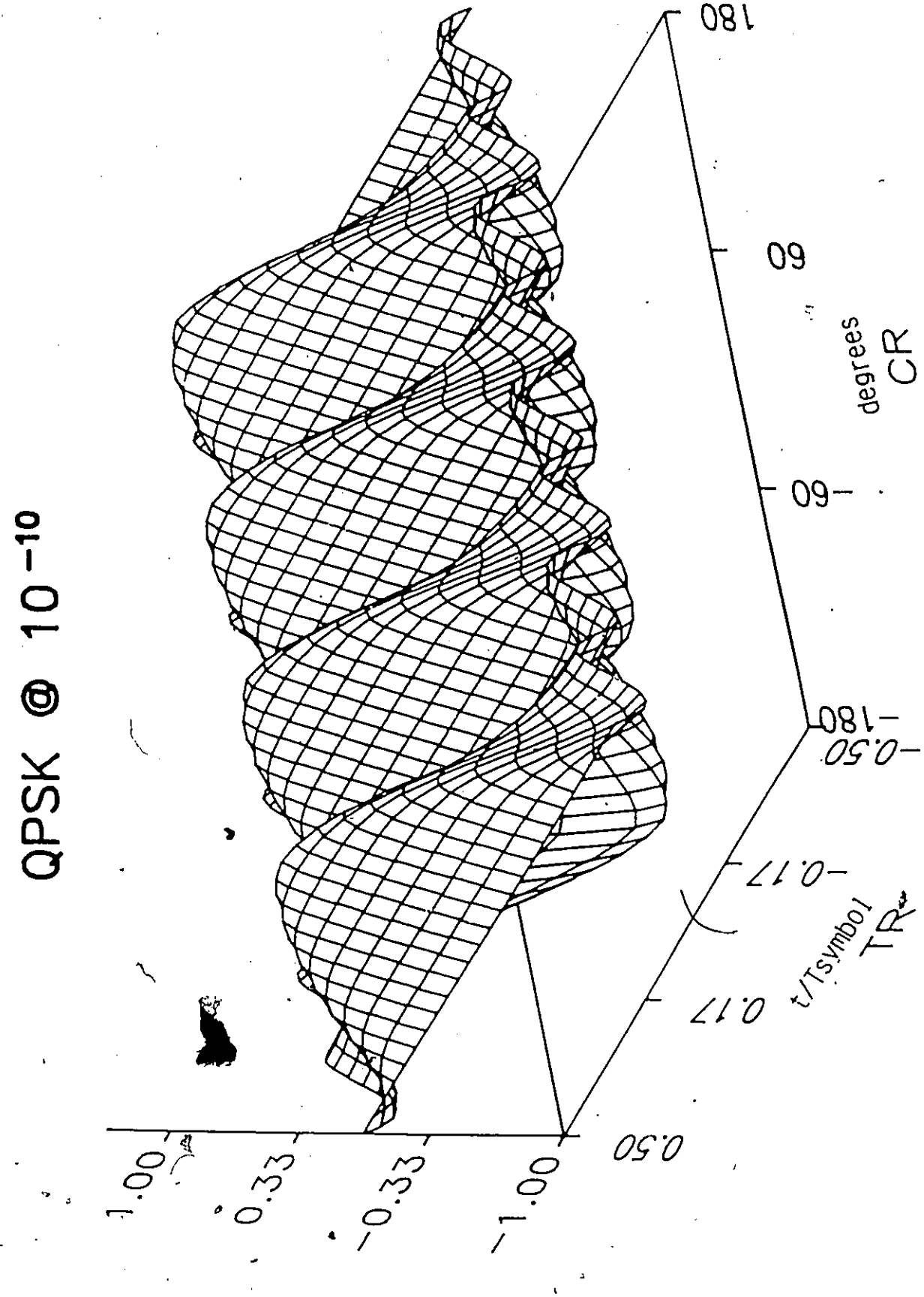


Figure 4.10. The QPSK nonlinearity surface.

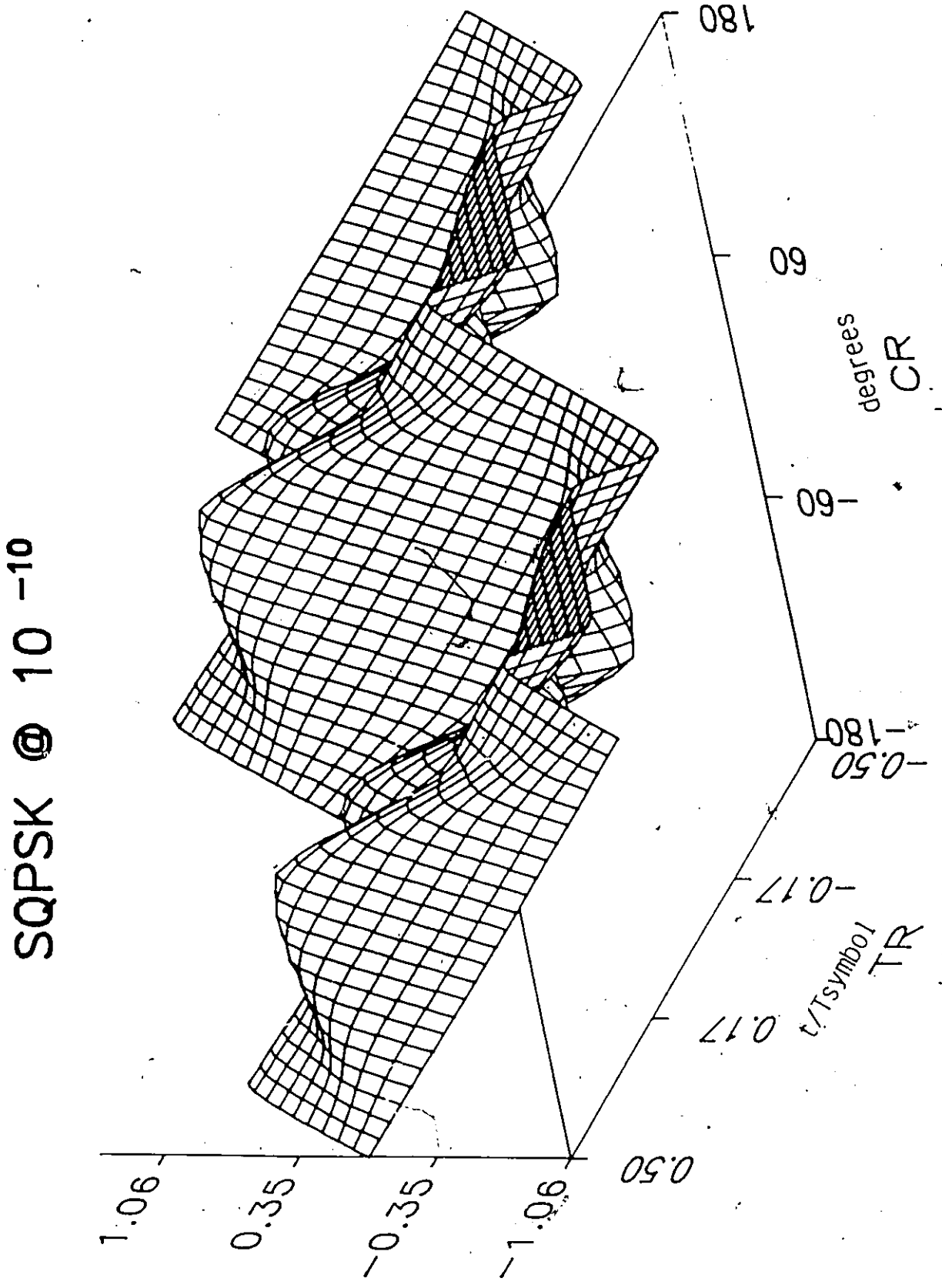


Figure 4.11. The SQPSK nonlinearity surface.

4.2.2. EQUALIZED DFCRL

Although an equalization within the loop might be employed in any narrowband system to improve performance,here we analyze a wideband system employing an overlapped modulation scheme. Due to overlapping pulses extending beyond the $(-T/2, +T/2)$ interval, Fig.2.2, ISI is present at the receiver. We analyze this ISI,which might be reduced by employing a simple equalizer in each of the quadrature arms of the DFCRL. Our analysis uses the SQORC (also known as Feher's IJF, [BS]) modulation scheme, Table 2.1,Fig.2.2, eqs.(2.1–2.3), as a model which might be straightforwardly applied to any other overlapped scheme. From (2.1–2.3), it is easy to show that SQORC does not have a constant envelope,i.e. the factor within the second sum in (4.9) is not a constant. Therefore,a ML approach should take this into consideration. By applying the previous, rather lengthy procedure, an ML-based receiver might be derived. Instead,we use a more effective approach based on the following logical reasoning: The autocorrelation function of an overlapped scheme spans more than a $2T$ second interval, **Fig.4.12**, i.e. neighboring pulses cause ISI at the sampling point. We assume that an early-late gate loop, like a TR crosscorrelation device, is used to establish a timing reference,[B3], and sampling is made every T seconds. The crosscorrelation function of a presumably rectangular reference signal and input pulses (described by eqs. 2.3a-b), rather than the autocorrelation function given in **Fig.4.12**, are used for our purposes. A T seconds long pulse might have four different forms, **Fig.4.13a**, and an equal number of forms with opposite signum. The signal power after the ISD device (denoted with an integral in **Fig.4.6**), i.e. at the decision device input, varies from symbol to symbol because of non-equal pulses. The probability of error performance is dictated by a pulse with the lowest energy,i.e a degradation of 1.09 dB might be expected. This theoretical result is experimentally verified on our IJF modem, **Fig.4.13b**. This translates into a conditional probability of error degradation from 10^{-10} to $10^{-8.59}$. See the UD in **Fig.4.14**.

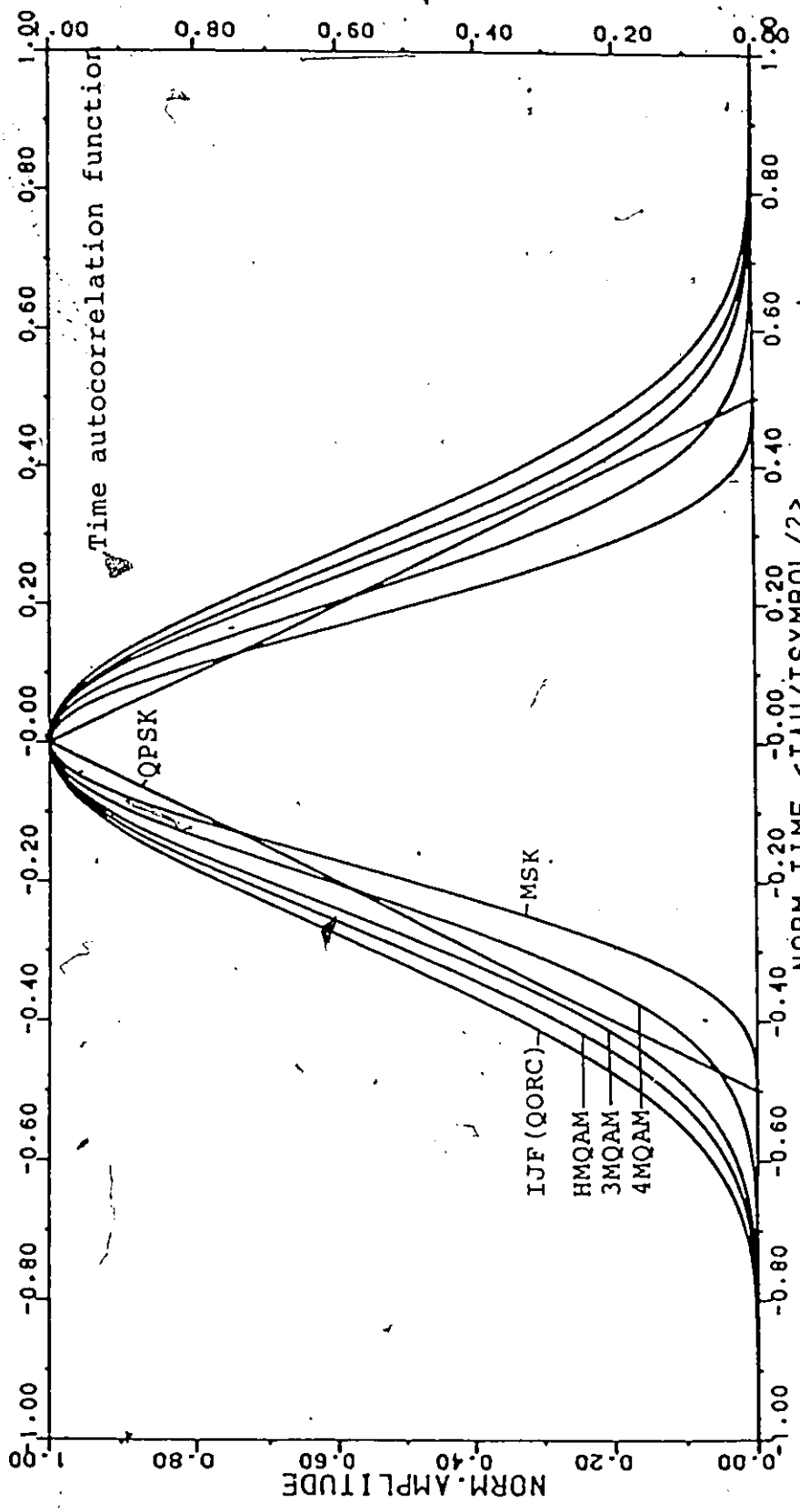


Figure 4.12. The autocorrelation functions of WQAM schemes.

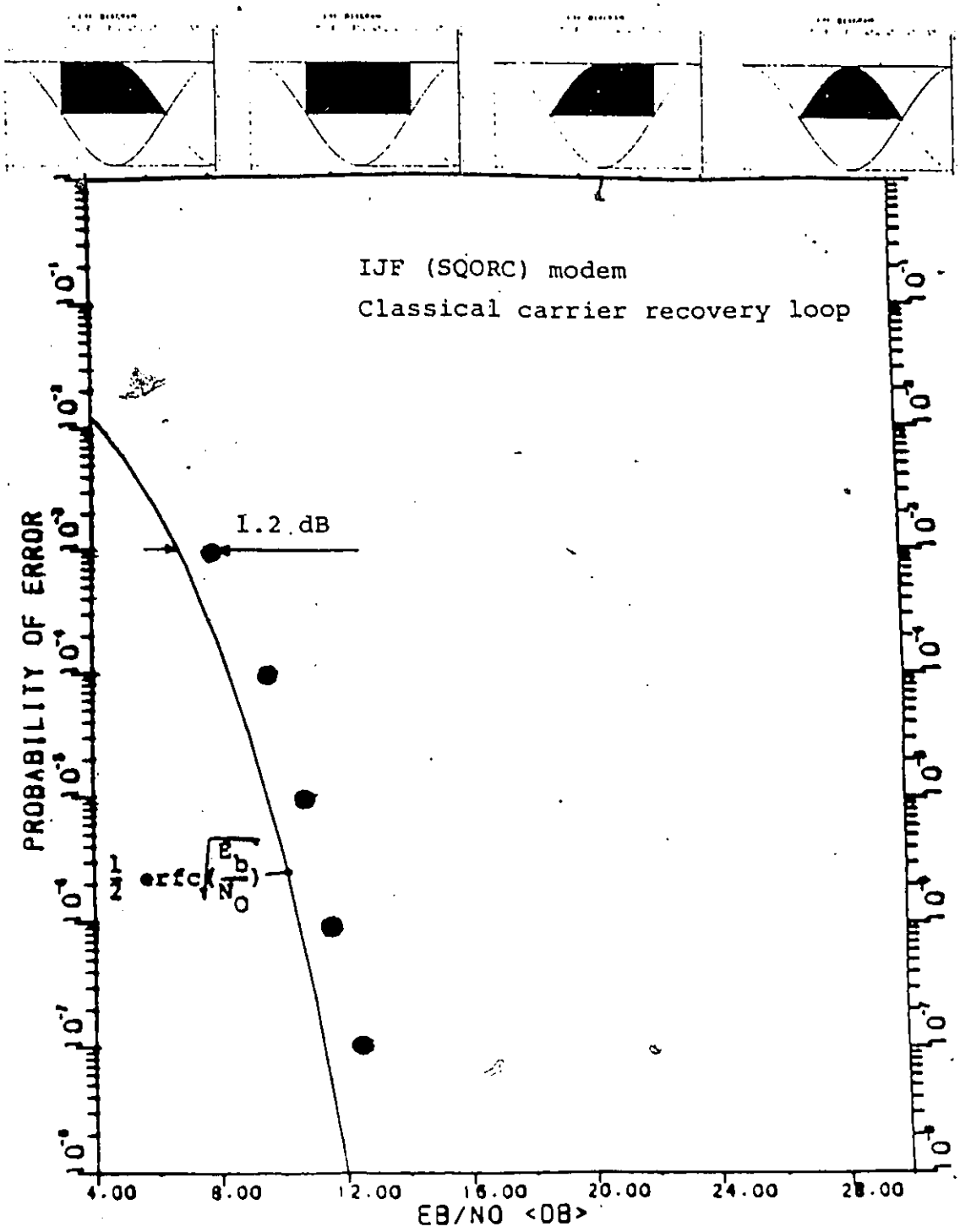


Figure 4.13. a) The SQORC signal shapes. b) The average probability of error performance measurements (•) of the IJF (SQORC) modem employing the classical carrier recovery loop.

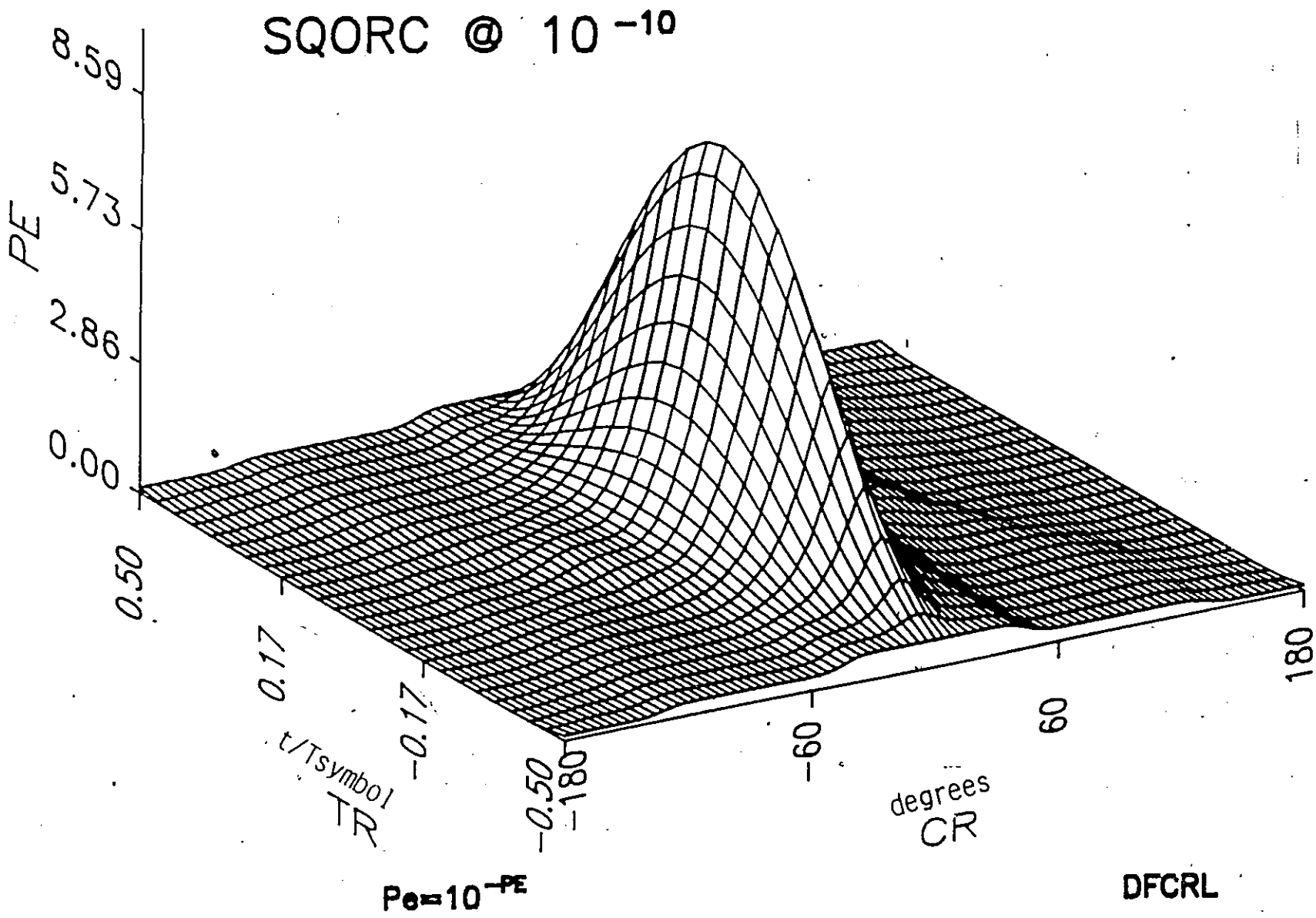


Figure 4.14. The SQORC uncertainty diagram employing the classical DFCRL. The performance of 10^{-10} is taken as the reference.

To smooth the signal fluctuations of overlapped schemes at the decision device input a simple 3-tap equalizer is placed between the ISD and the decision device inside the DFCRL in **Fig.4.6**, which gives a new loop called **equalized DFCRL,Fig.4.15**. The noise after the equalizer is Gaussian but not white with variance $\sigma_n^2 = (c_o^2 + 2c_1^2)N_o$,where c_o, c_1 are tap coefficients. As a consequence,the performance of SQORC is improved to $10^{-9.97}$ which is a minor, rather insignificant degradation (0.03 dB) from the 10^{-10} reference. The UD of SQORC using this transversal type of equalizer within the DFCRL is illustrated in **Fig.4.16**. Although the pulse shape of QORC schemes extends over parts of three T intervals— the right half of the $(k-1)$-th,all of the k-th, and the left half of the $(k+1)$-th interval, **Fig.2.2**, the energy in the $(k-1)$-th interval depends also on the pulse in the $(k-2)$-th interval, etc. Total ISI cancellation requires an infinitive number of taps. However,a simple 3-tap equalizer cancels ISI almost completely. Tap coefficients actually depend on the data streams in both quadrature channels,except for $\phi = 0$. By employing a more complex feedforward and decision feedback equalizer, we might be able to achieve a slightly better performance than given in **Fig.4.16**. However, a detailed analysis of the canceller optimization is out of the scope of this contribution. An ML-based device for SQORC (and other overlapped schemes) might also be realized in a parallel form. Three pairs of in-phase and quadrature arms at frequencies f_c and $f_c + n/T_S$ will be followed by the ISD and decision devices plus a baseband combiner. No equalizers will be needed.

We have concluded the analysis of the ML approach and proceed with the MAP approach, in which a search for new improved loops is pursued.

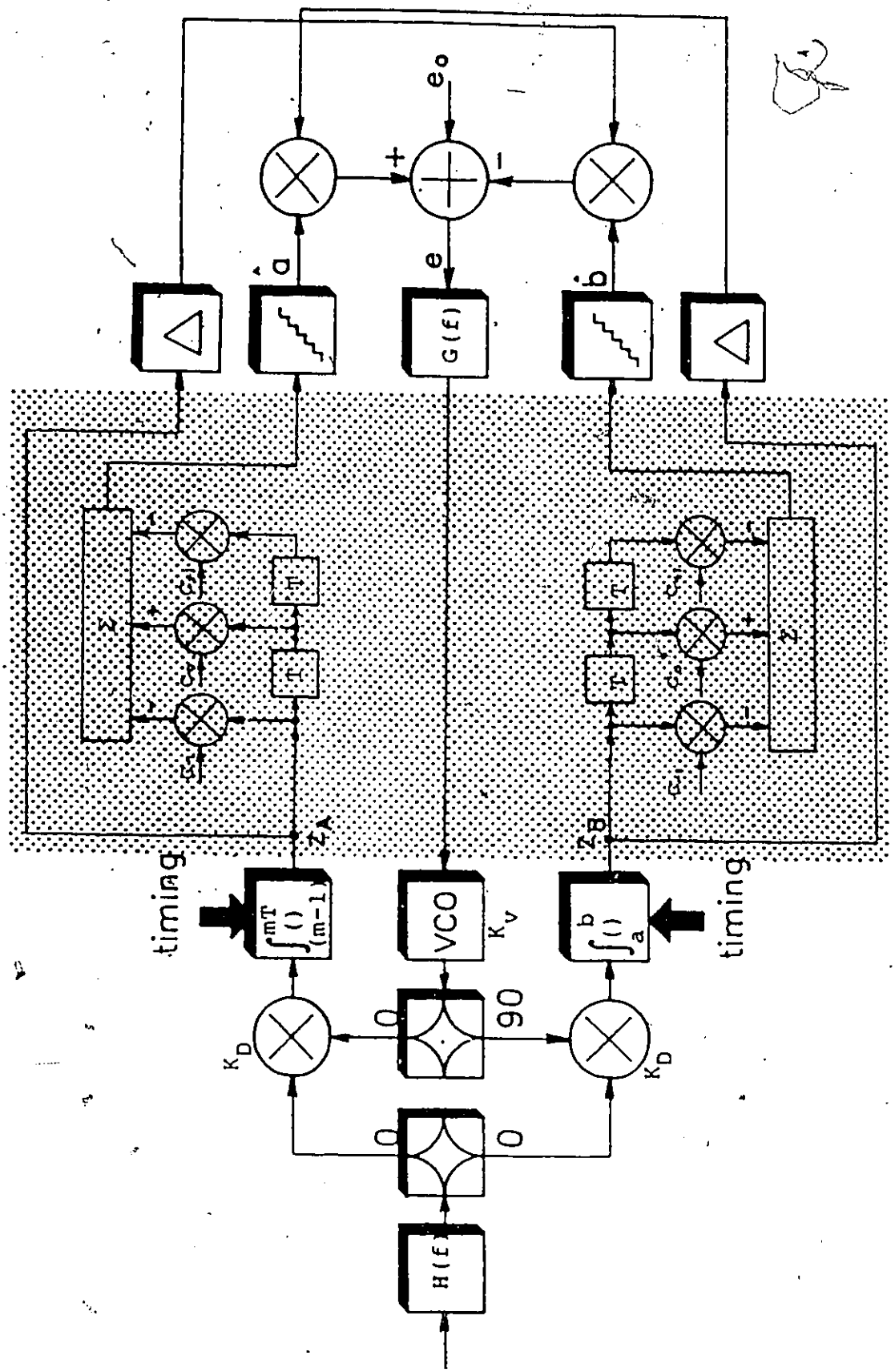


Figure 4.15. The equalized decision feedback carrier recovery loop.

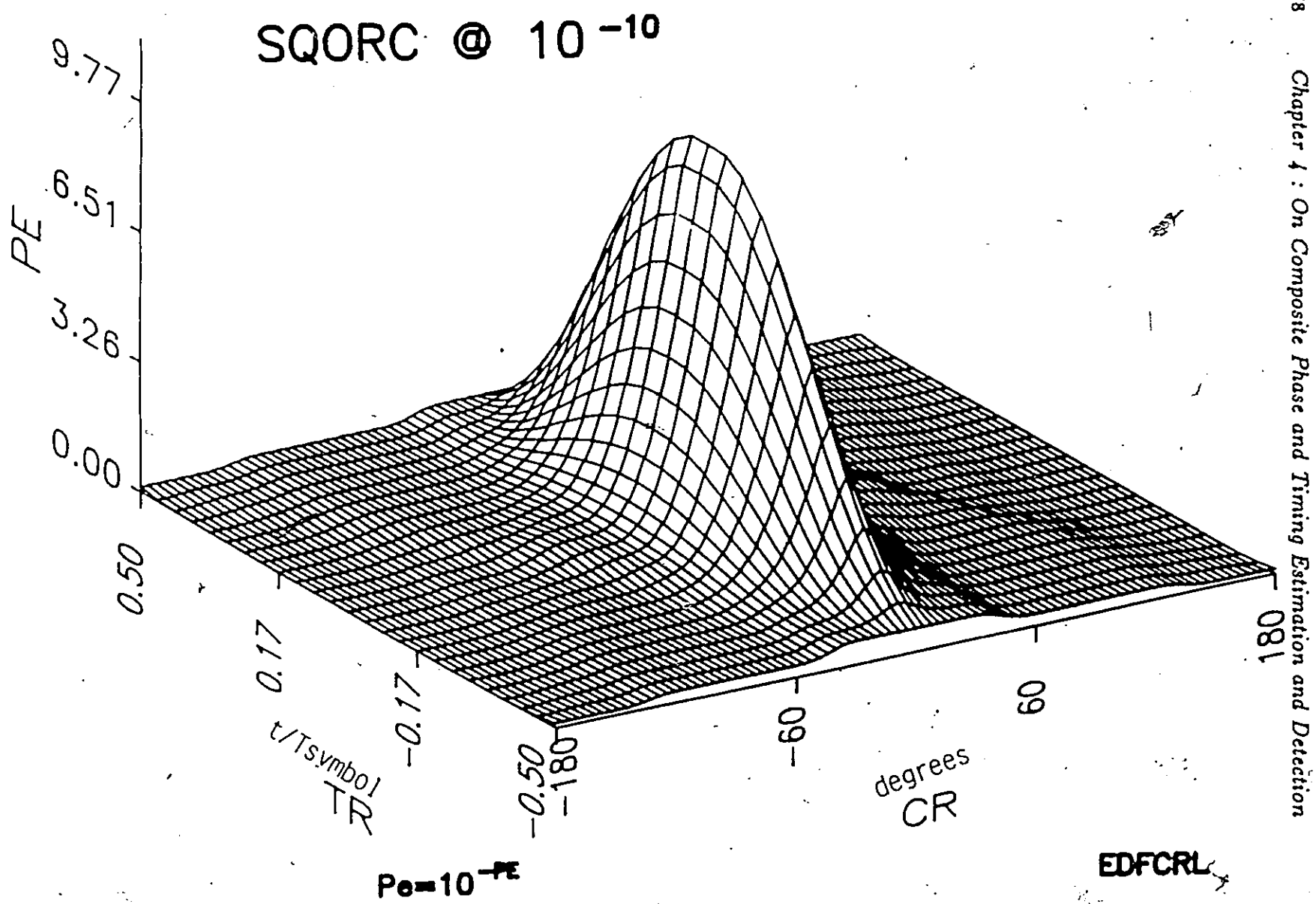


Figure 4.16. The SQORC uncertainty diagram employing equalized DFCRL. The performance of 10^{-10} is taken as the reference.

4.3 MAXIMUM A POSTERIORI (MAP) APPROACH

Previously,a uniform distribution of $f(\epsilon_1, \epsilon_2, \theta)$ in (4.6) was assumed and ML optimization followed. A structure with the $\tanh(x)$ nonlinearity was found, which was approximated with $sign(x)$ for high SNR and realized with a quantizer characteristic (4.31) for schemes with more than 2-baseband levels. The corresponding loop is called classical DFCRL, **Fig.4.6.** However,as shown by (4.31–4.39) and **Figs.4.7–4.11**, this loop seems to be optimal for $\phi = \lambda = 0$ only. In general,we might use the knowledge of $f(\epsilon_1; \epsilon_2, \theta)$ accumulated over previous V_T and V_c symbols to improve our decision on the k-th symbol. This value of $f(\epsilon_1, \epsilon_2, \theta)$ might be introduced in (4.6) and an MAP optimization procedure might follow. However, $f(\epsilon_1, \epsilon_2, \theta)$ and the related $p(\phi, \lambda)$ are nonelementary functions, which are rather difficult if not impossible to manipulate. Therefore,we use a more practical approach based on the knowledge of ϕ, λ and $h(\phi, \lambda)$ accumulated over previous V_c and V_T symbols respectively,as follows: According to (4.32–4.35),the phase uncertainty ϕ of the loop causes an attenuation of the in-phase signal proportional to $\cos\phi$ and a crosstalk proportional to $\sin\phi$. The timing uncertainty λ causes an attenuation of the k-th pulse (on which the decision has to be made) proportional to $R(|\lambda|)$ and an ISI of the $(k-1)$-th or $(k+1)$-th neighboring pulse proportional to $R(1-|\lambda|)$. A model of these degradations is shown in **Fig.4.17a.** Since we know the V_c symbols long estimate of ϕ,say $\hat{\phi}$, the V_T symbols long estimate of λ,say $\hat{\lambda}$,and the (V_c, V_T) symbols long estimate of $h(\phi, \lambda)$,say $\hat{h}(\phi, \lambda)$,we might be able to compensate for these degradations, **Fig.4.17b.** For the purpose of this contribution,we assume perfect estimates of ϕ, λ and $h(\phi, \lambda)$, i.e. $\phi = \hat{\phi}$, $\lambda = \hat{\lambda}$, and $h(\phi, \lambda) = \hat{h}(\phi, \lambda)$, which might be reasonable since high SNRs within the loop bandwidth are assumed. A compensation for the ϕ-caused degradation is explained in Section 4.3.1, compensation for the λ-caused degradation is explained in Section 4.3.2, while a hybrid compensation scheme follows in Section 4.3.3.

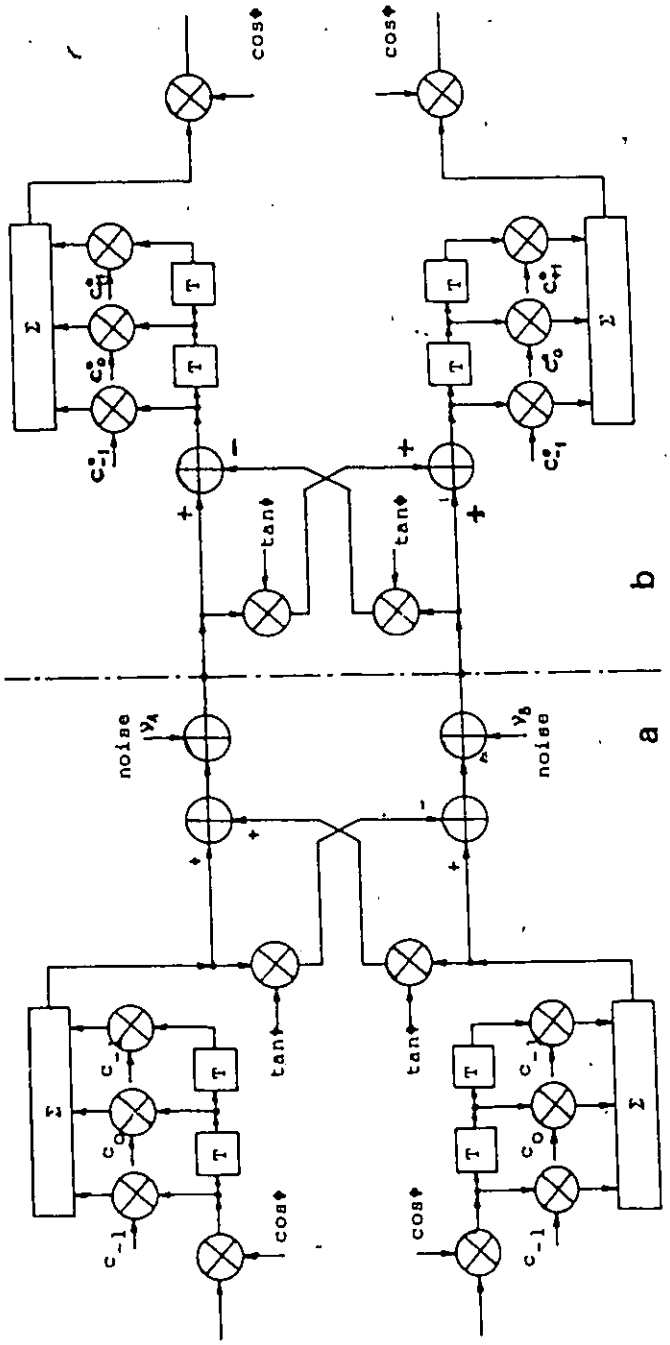


Figure 4.17. a) The degradation model for the loop caused impairments due to the phase uncertainty ϕ and the timing uncertainty λ. b) The corresponding equalization model.

4.3.1 Crosstalk Cancellation DFCR Loop

The following conclusions might be drawn from (4.31-4.40): The quantizer represented by the characteristic in (4.31) is an optimum nonlinearity for $\phi = 0$ only. However,to minimize an average probability of error P_e for any ϕ, a device for which the decision is independent of ϕ must be found. In addition to the already applied ML criterium, different criteria (constraints) might apply,e.g. minimum mean square error,etc. We decide to minimize $P(e|\phi)$ over the entire range of ϕ, which, as shown later, corresponds to the cancellation of pattern jitter. Although an analysis might be quite complex regarding the difficulties of an exact calculation of the noise statistics at the VCO input, we ignore this noise, which is reasonable as long as the loop bandwidth is small in comparison with the data bandwidth. Then,a new **crosstalk cancellation (CC) DFCRL** for M-ary quadrature amplitude modulation (MQAM) signal sets is given in **Fig.4.18**. The loop employs two adders,two baseband amplifiers with gains equal to $\tan\phi$, and two adaptive quantizers (or a pair of amplifier-fix quantizer combination), which are governed by a phase error ϕ of the loop, to cancel (or at least attenuate) the crosstalk. As a result,the pattern dependent jitter is cancelled (at least attenuated) and the loop performs as a continuous wave (CW) CR loop. Here,we optimize the loop performance for $\lambda = 0$, while a more general case $\phi \neq 0$, $\lambda \neq 0$ is analyzed in Section 4.3.3. If $K_A(\phi) = K_B(\phi) = \tan\phi$,a signal at the A decision device input is

$$\begin{aligned}\zeta_A &= (+a_m \cos\phi + b_n \sin\phi + \nu_A) \\ &\quad - (-a_m \sin\phi + b_n \cos\phi + \nu_B)\tan\phi \\ &= \frac{a_m}{\cos\phi} + (\nu_A - \nu_B \tan\phi)\end{aligned} \tag{4.42}$$

i.e. no crosstalk occurs and the signal-to-noise ratio remains constant,since both the desired signal and noise powers are increased in the same proportion. A decision is independent of CR uncertainties,i.e. the loop operates as a CW CR loop and the nonlinearity curves equal $\sin\phi$, **Fig.4.19a** (here an ideal timing is assumed). Note that the curve labeled $\lambda = 0.0$ corresponds also to any MQAM scheme (within a constant factor), assuming CCDFCRL is employed. The nonlinearity curves ($\lambda = 0.0$) of the MQAM schemes employing classical DFCRL are summarized in **Fig.4.19b**. The advantage of our new crostalk cancellation loop over the classical one is obvious.

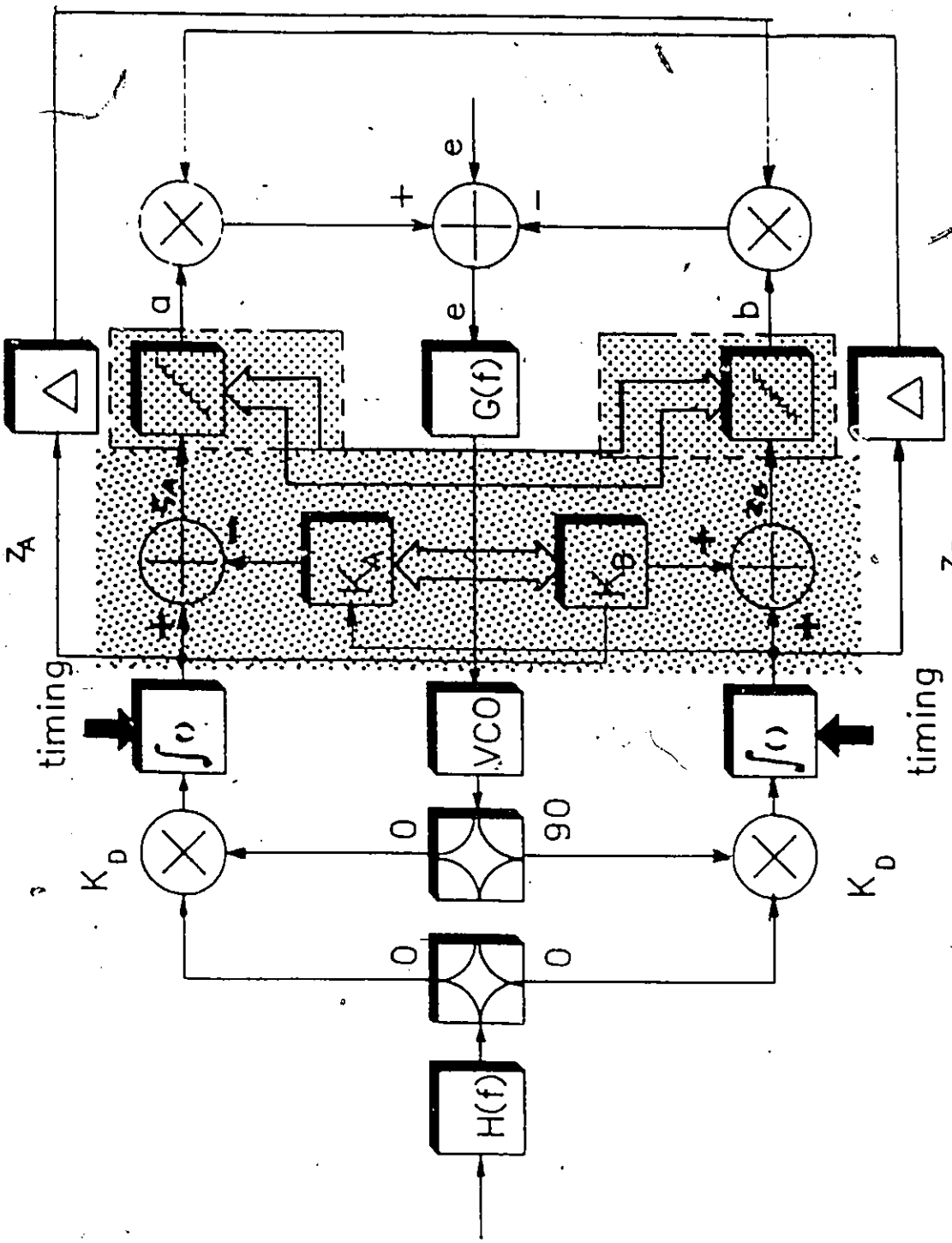


Figure 4.18. The crosstalk cancellation DFCRL.

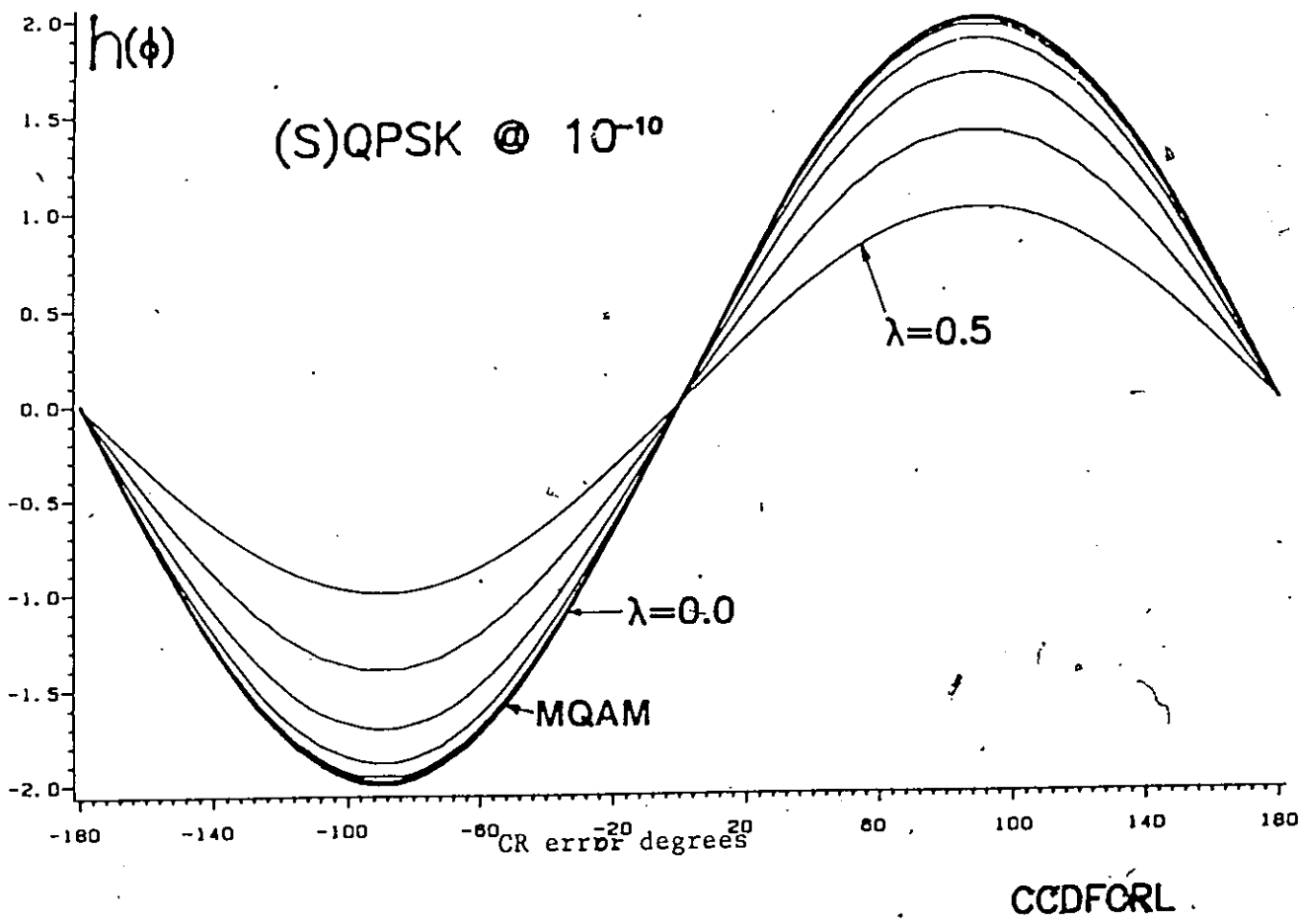


Figure 4.19a . The (S)QPSK nonlinearity curves employing the crosstalk cancellation decision feedback carrier recovery loop. The $\lambda = 0.0$ labelled curve corresponds to the MQAM schemes also.

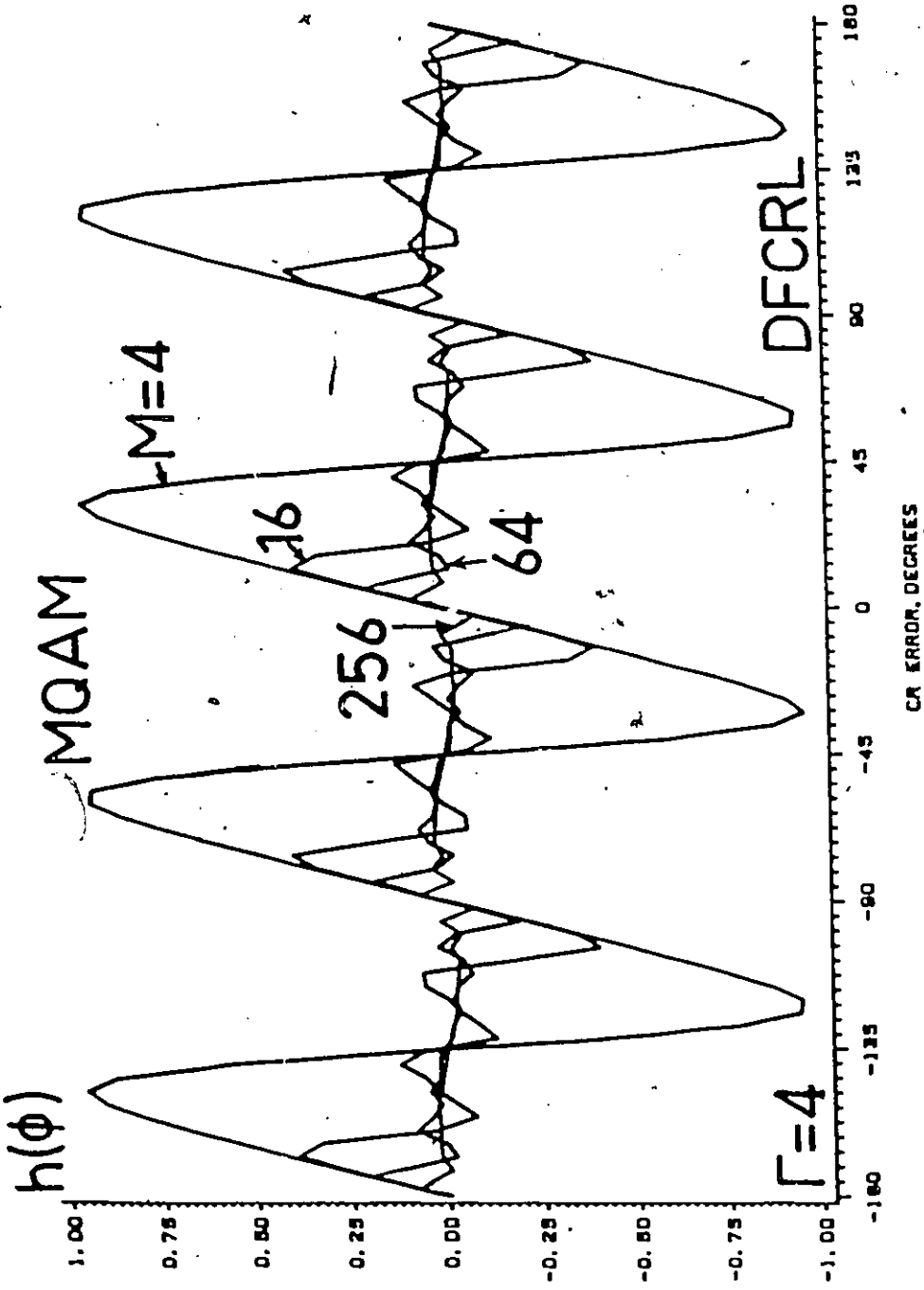


Figure 4.19b. The MQAM nonlinearity curves employing classical decision feedback carrier recovery loop. The parameter $\Gamma = 4$ corresponds to the $P(\epsilon|\phi) \approx 10^{-8}$.

In our analysis of the CCDFCRL we have assumed an optimal estimation of ϕ, i.e. $\phi = \hat{\phi} = [h(\phi, \lambda = 0)]^{-1}$ where $[h(\phi, \lambda = 0)]^{-1}$ equals the inverse of $h(\phi, \lambda = 0)$. A detailed analysis of this estimator is out of the scope of this presentation. However,a brief heuristic explanation follows. Let's assume that the loop is initially locked and $\phi = 0$. As input symbols of data are arriving into the loop arms, the phase uncertainty $\phi(t)$ due to Gaussian noise fluctuates slowly (relative to the symbol rate) and might take any value between $(-\pi, +\pi)$. For small ϕ, $h(\phi) \doteq \phi$, i.e. ϕ is directly proportional to the voltage $h(\phi)$. The shaded part of the loop in **Fig.4.18** feeds back this (presumably correct) information on ϕ accumulated over the previous V_c symbols and improves the decision at the k-th symbol,which is now virtually independent of ϕ. This improved decision changes $h(\phi)$ from the complex form given in equation (4.40) and **Fig.4.19b** to $h(\phi) = \sin\phi$, **Fig.4.19a**. As a final result,the pattern jitter is cancelled (practically,it is attenuated by an amount proportional to the ratio of data and loop bandwidths $R_\rho = SNR_{IF}/SNR_L$) and, more importantly,the quadrant ambiguities previously present due to the combination of signal constellation and fixed threshold decision devices,have vanished. Similar improvement might be expected in the tracking of unbalanced schemes while employing proper gains in the quadrature arms.

The average probability of error P_e of the WQAM schemes employing CCDFCRL equals

$$P_e = \underbrace{P(e|\phi)}_{\text{constant}} \underbrace{\int_{-\pi}^{+\pi} p(\phi)\, d\phi}_{=1} = P(e|\phi = 0) \tag{4.43}$$

i.e. no degradation (pattern jitter) exists due to the presence of data, assuming $P_e < 10^{-3}$. At high P_e (above 10^{-2}) the signal $h(\phi)$ which governs the loop is attenuated (by approximatelly 0.175 dB at $P_e \approx 10^{-1}$ when 16QAM is conterned) since the decision principal is employed within the loop. However, further analysis of these effects at high P_e is out of the scope of this presentation.

In order to compare the average probability of error P_e performances of the WQAM schemes employing the classical DFCRL and those schemes employing our new crosstalk cancellation DFCRL the knowledge of the probability density function $p(\phi)$ of the classical DFCRL is necessary. This is rather difficult task, which requires a tedious mathematical procedure to be employed in an attempt to solve the stochastic partial differential equation of the Fokker-Planck-Kolmogorov type, or a time consuming computer simulation needs to be done. On the contrary, the probability density function $p(\phi)$ of the crosstalk cancellation DFCRL is practically equivalent to that of the CW loop and is given by [B16], **Fig.4.1**,

$$p(\phi) \overset{\rho \geq 10}{\approx} C \frac{\exp[\rho(\cos\phi + \beta_N \phi)]}{2\pi I_0(\rho)} \tag{4.44}$$

ρ is the signal-to-noise ratio within the loop, C is the normalization constant, $\beta_N = \beta/\rho$ is the normalized loop detuning, β is the absolute detuning, and $I_0(\rho)$ is the modified Bessel

function of zero order and argument ρ. We assume that the classical DFCRL has the same probability density function $p(\phi)$ as the CW loop (and the crosstalk cancellation DFCRL) — except for the factor four and the periodicity of $\pi/2$ radians. The true probability density function of the classical DFCRL will have less desirable form. Therefore, our estimation might be biased in favor of the classical loop.

We were able to perform the Monte Carlo simulation of the performance of the QPSK schemes — employing both classical and crosstalk cancellation loops — at $P_e > 10^{-4}$ and that of the 16QAM at $P_e \approx 10^{-2}$. We also generated the theoretical performance curves of QPSK and 16QAM schemes at $\rho \leq 174$ (22.4 dB). The results are summarized in **Figs.20–21**. In **Fig.20** the QPSK $\log_{10} P_e$ performance versus the carrier-to-noise ratio C/N in dB in the double-sided Nyquist bandwidth and the normalized loop detuning (loop stress) β_N as a parameter is presented. The ratio $R_\rho = \rho/SNR_{IF} = 10$ is assumed. This is the lower end practical value, which allows us to generate these curves. Note that typical satellite systems operate at $SNR_{IF} > 5$ dB, i.e. $\rho > 15$ dB. The typical terrestrial (and cable) systems operate at even higher SNR_{IF} ($\rho \gg 15$ dB). An estimation error of 0.46 dB is assumed when the crosstalk cancellation DFCRL is employed. An ideal ambiguity resolution is assumed when the classical DFCRL is employed — a bias in the favor of the classical loop. In practice,a degradation of 0.3–0.5 dB might be expected due to the ambiguity resolution. A random d.c. wandering within the loop, channel asymmetry (e.g. during a selective fading event), or any other dynamic impairment of such kind will cause a stress in the loop and consequently a degradation of the P_e performance. The classical DFCRL suffers significant degradation, while our new loop is practically insensitive to these kind of impairments, **Fig.4.1**, **Fig.4.20**. At $P_e = 10_{-6}$ and $\beta_N = 0.7$ our new loop outperforms the classical one by 2.5 dB, assuming QPSK is employed. Results in **Fig.4.7**, and **Fig.4.21** indicate that at lower P_e and for higher state modulation schemes our new loop will outperform the classical one by even higher margin.

• E.g., let's brief the necessary performance of the $K_A(\phi)$ and $K_B(\phi)$ devices (amplifiers). Let's assume the symbol rate equals 10 Msymbols/s. Then, the necessary bandwidth of the $K_A(\phi)$ and $K_B(\phi)$ devices is in the range of the tens of MHz. Usually, the loop bandwidth is a one hundred or more times narrower than the bandwidth of the data. Therefore, the phase changes ϕ and the necessary dynamic changes of the $K_A(\phi)$ and $K_B(\phi)$ amplifiers are in the range of one hundred of kHz, or lower. The necessary gain is proportional to the $\tan\phi$. If $|\phi| < 45°$, $K_A(\phi)$ and $K_B(\phi)$ perform as attenuators. If the maximal available gains of $K_A(\phi)$, $K_B(\phi)$ equal 100, the $\tan\phi$ function might be modeled for any ϕ except in the 1° wide strips around ±90°. Therefore, the crosstalk cancellation DFCRL might be realized with the practically available components.

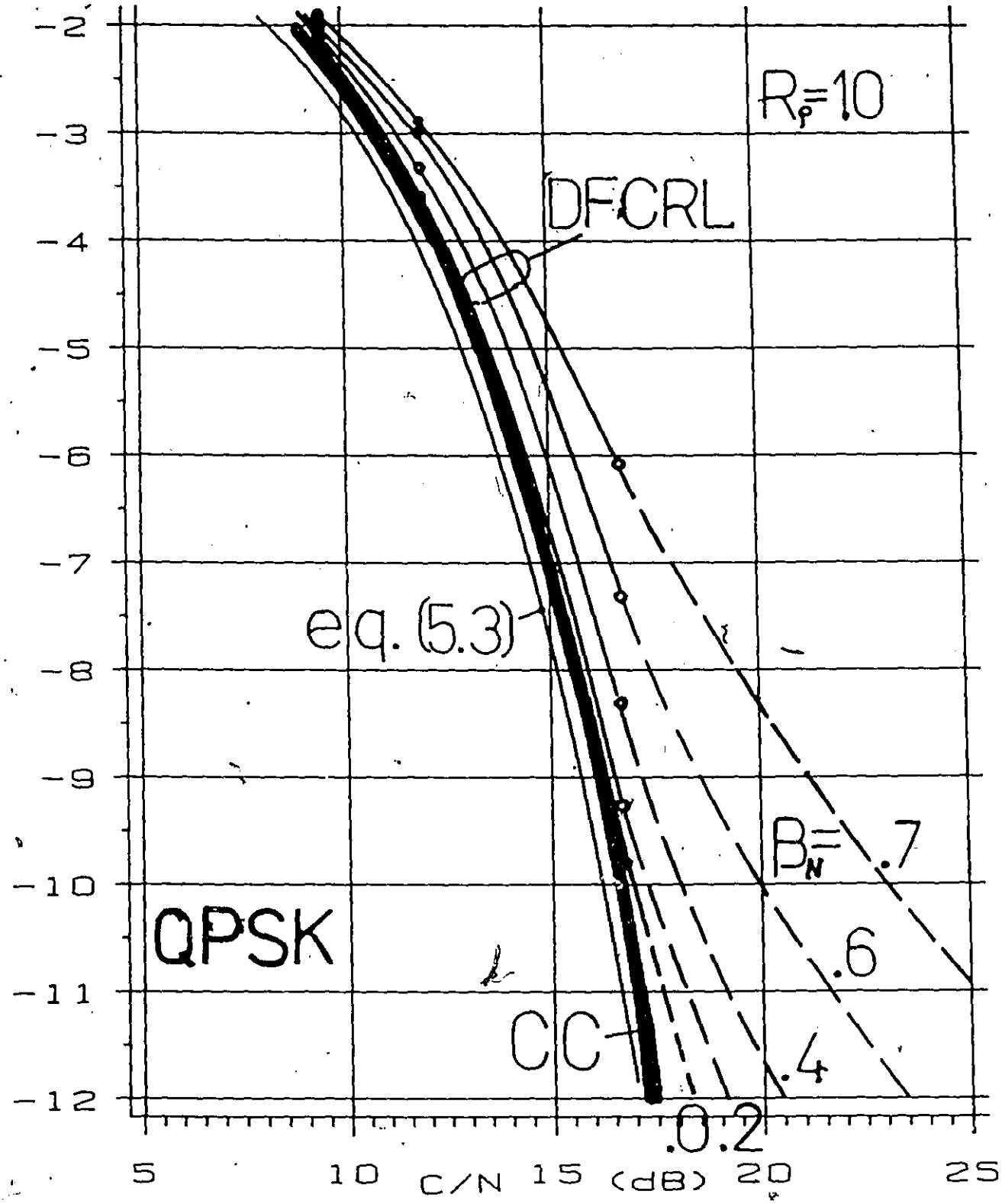


Figure 4.20. The QPSK $\log_{10} P_e$ performance versus the carrier-to-noise ratio C/N in dB in the double-sided Nyquist bandwidth and the normalized loop detuning (loop stress) β_N as a parameter, employing classical or crosstalk cancellation (CC) decision feedback carrier recovery loop (DFCRL). The ratio $R_\rho = SNR_L/SNR_{IF} = 10$ is assumed.

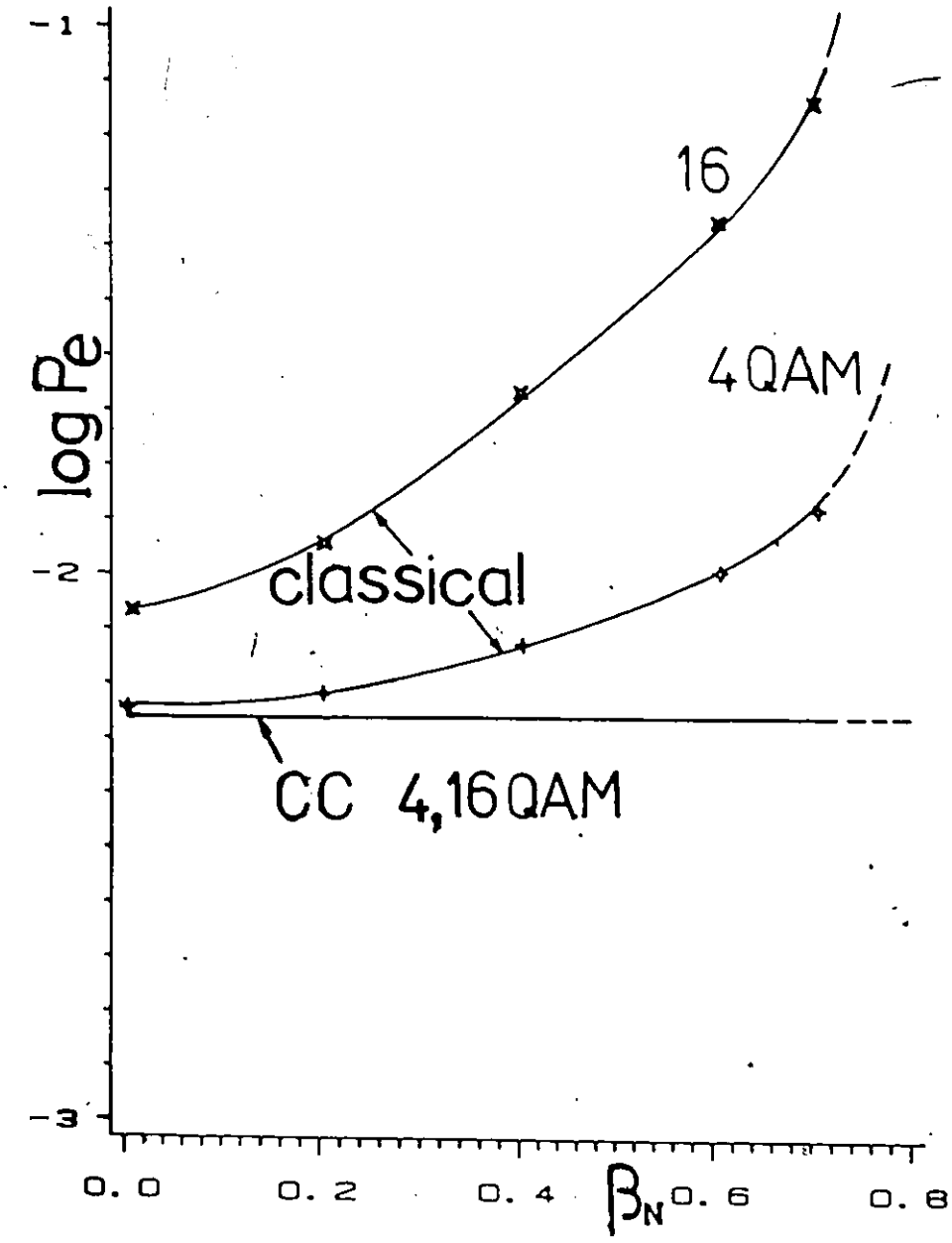


Figure 4.21. The 4 and 16QAM $\log_{10} P_e$ performance versus the normalized loop detuning (loop stress) β_N, employing classical or crosstalk cancellation (CC) decision feedback carrier recovery loop (DFCRL). The ratio $R_\rho = SNR_L/SNR_{IF} = 10$ is assumed.

4.3.2 Timing ISI Cancellation DFCR Loop

By observing equation (4.34), it is easy to see that a timing uncertainty λ causes complex ISI. As an example, these negative effects might be partially overcome by employing a simple 3-tap equalizer,whose central tap gain is proportional to $+1/R(|\lambda|)$ and the gains of two neighboring taps are proportional to the product of $-R(1-|\lambda|)$ and the corresponding step function ($c_{-1} = 0$ for $\lambda \geq 0$, while $c_{+1} = 0$ for$\lambda \leq 0$), **Fig.4.17b.** In a system employing narrowband filtering,an equalizer with more taps will be necessary. The resultant CR loop is called **timing ISI cancellation DFCRL,Fig.4.22.** The loops in **Fig.4.15** and **Fig.4.22** are the same except for the different tap gains of the corresponding equalizers. Here we have assumed perfect knowledge of λ and have ignored crosstalk terms proportional to $\sin\phi$. The noise after the equalizer is Gaussian but not white with variance $\sigma_n^2 = [R^{-2}(|\lambda|) + R^2(1-|\lambda|)]N_o$. However,SNR is proportional to

$$\frac{R^2(|\lambda|)}{1 + R^2(|\lambda|)R^2(1-|\lambda|)} \tag{4.45}$$

Timing ISI pattern jitter is cancelled but SNR and the conditional probability of error deteriorate as λ increases. However,this deterioration is not as severe as in the classical DFCRL. The relative improvement increases as the number of baseband levels increases. A decision feedback equalizer which cancels timing ISI on the previous symbol(s) and a transversal equalizer which acts on future symbol(s) might give better results. Further improvements might be expected with an optimization of the TR circuit.

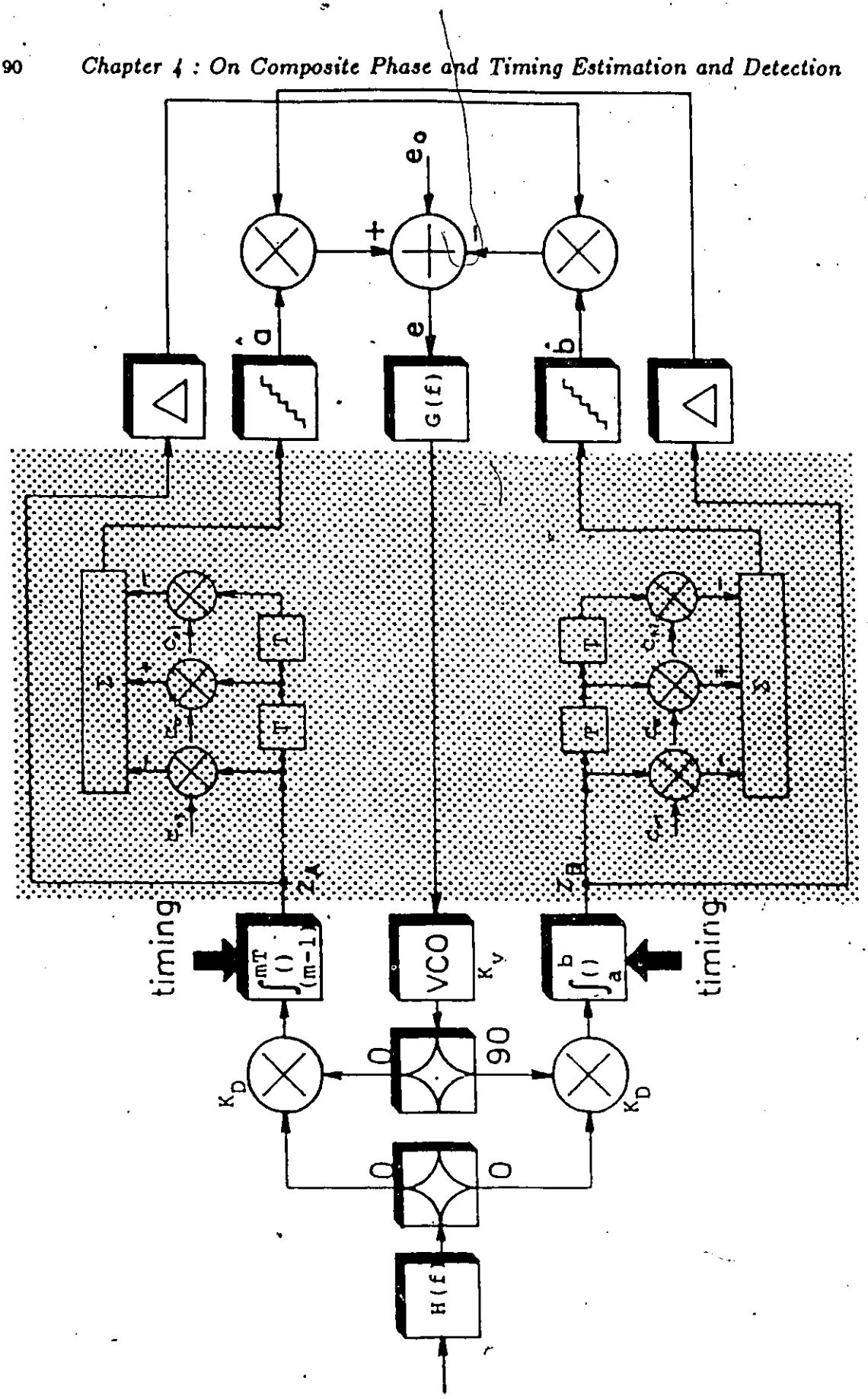


Figure 4.22. The timing intersymbol interference cancellation DFCRL.

4.3.3 FEEDLOOP

By using knowledge of both ϕ and λ accumulated over V_c and V_T symbols respectively,a new feedback structure which uses an estimation of phase and an estimation of timing to improve detection is suggested in **Fig.4.23.** We call this loop **FEEDLOOP**, as an acronym for Feedback of data,Estimation of phase, Estimation of timing, and Detection **LOOP**. The uncertainty diagram of FEEDLOOP is independent of ϕ and dependance on λ is significantly reduced,which becomes particularly important for multilevel schemes. SNR at the decision device input is proportional to

$$\frac{R^2(|\lambda|)\cos\phi}{\cos\phi + R^2(|\lambda|)R^2(1-|\lambda|)} \tag{4.46}$$

for a FEEDLOOP employing a transversal equalizer, **Fig.4.23.** Based on the results of the Monte Carlo simulation, the UD of 256QAM scheme employing a FEEDLOOP is given in **Fig.4.24.** Note the deeps arround the singular points at $\phi = \pm$ 90 degrees. The width of these deeps depend on the practical limitations of the components within the loop. The advantages of the FEEDLOOP,as seen in the results presented in **Fig.4.24**, and **Fig.4.1c**, over the classical DFCRL, **Fig.4.10a-b**, and **Fig.4.1a**, are obvious.
The SNR for the FEEDLOOP employing a decision feedback equalizer is rather difficult to calculate,in general. However,regarding λ as a slow varying process relative to the symbol rate, i.e. λ is constant during one symbol period,

$$\text{ISI} \propto a_{k\pm1} R(1-|\lambda|)P(e_k|\lambda) \tag{4.47a}$$

$$\text{signal} \propto a_k\{1-[1-R(|\lambda|)]P(e_k|\lambda)\} \tag{4.47b}$$

$$\text{noise} \propto \sigma_n^2 \tag{4.47c}$$

where $\propto$ means proportional to. Crosstalk is reduced by $P(e_k|\lambda)$,the drop in signal is reduced in the same proportion,and noise remains the same.

A few more sentences are devoted to the choice of UD and nonlinearity surface as figures of merit: An UD independent of phase uncertainty ϕ and timing uncertainty λ represents the best possible performance,i.e. a CW loop. Any dependence on ϕ and λ represents a degradation caused by the presence of data,which has been called in the literature "data introduced ISI," "data dependent pattern jitter," "squaring loss," "self noise," etc. The nonlinearity surface determines the probability of error function of the loop and other properties such as acquisition,skipping rate,etc. Therefore,both the UD and nonlinearity surface describe the performance of the particular signal format-carrier recovery loop combination.

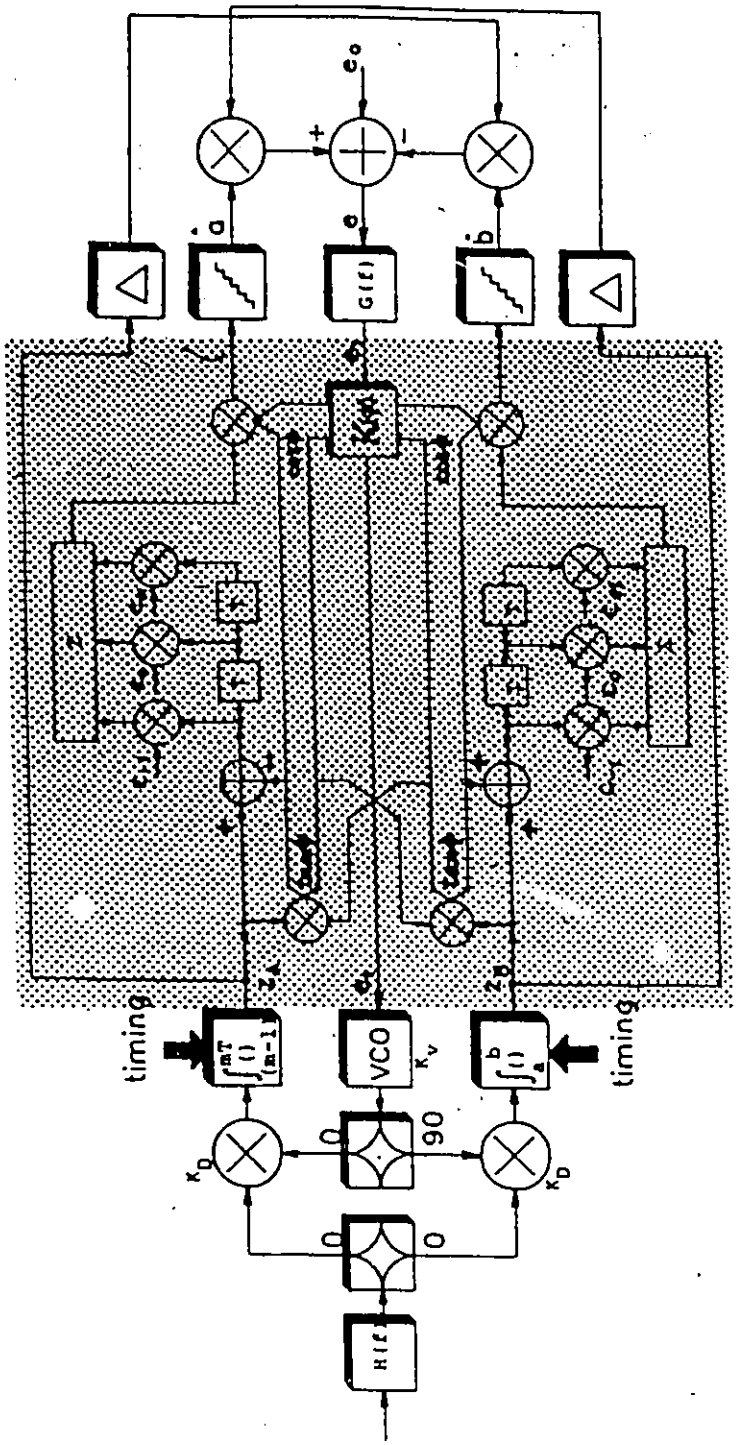


Figure 4.23. The FEEDLOOP.

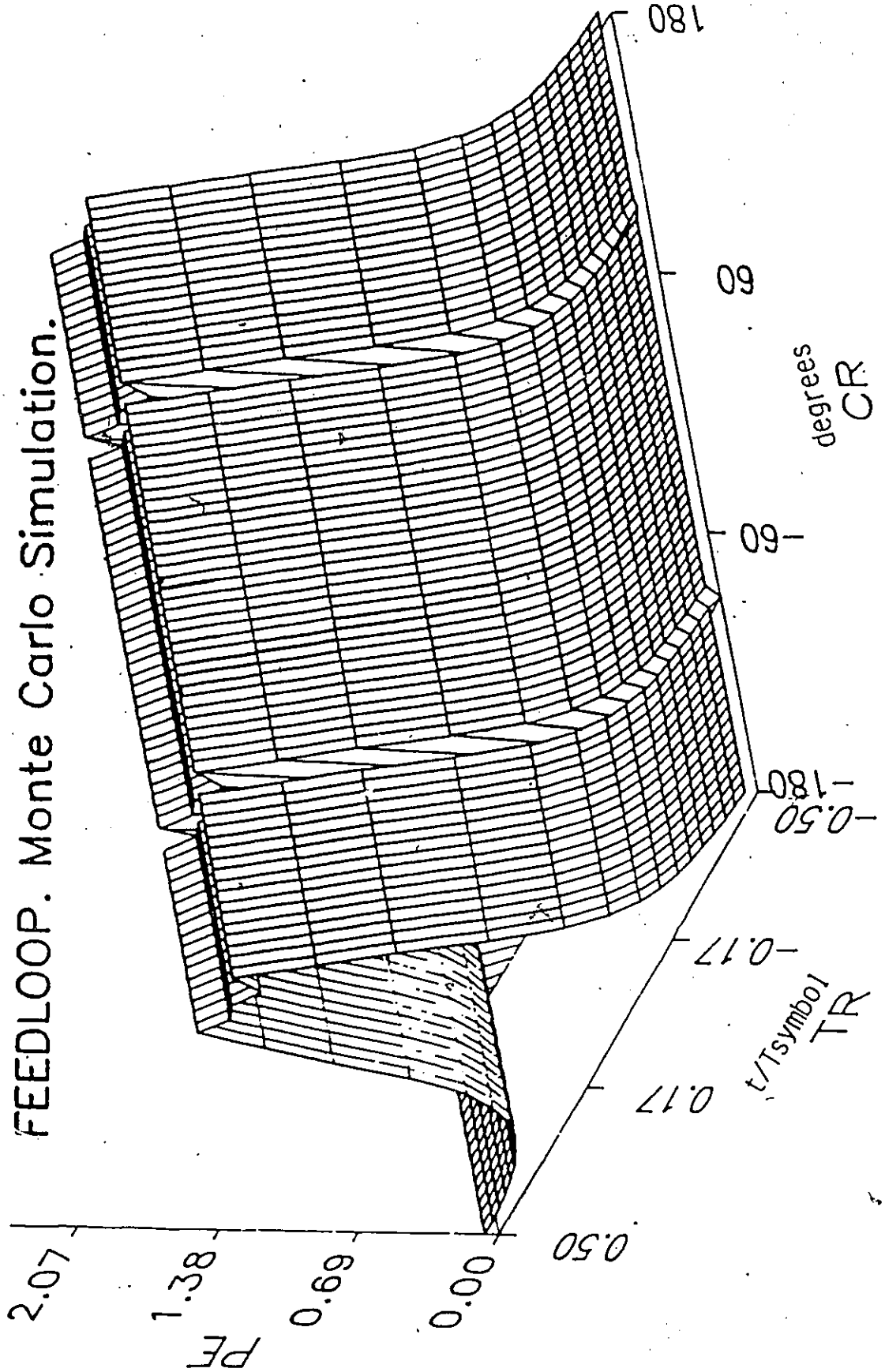


Figure 4.24. The 256QAM uncertainty diagram employing FEEDLOOP. The results of Monte Carlo simulation with 8124 symbols. $P(\epsilon|\phi,\lambda) = -10\log(PE)$. TR ($t/T_{\text{symbols}}$). CR (degrees).

4.4 CONCLUSION

In this chapter a composite phase and timing estimation and data detection of WQAM signal sets is analyzed. By using a maximum likelihood (ML) approach, four new joint phase and timing estimators are derived. The performance of the classical decision feedback carrier recovery loop (DFCRL) employing an integrate-and-sample device in each quadrature arm (active loop) is evaluated in the presence of both carrier uncertainty ϕ and timing uncertainty λ. Results are presented in the following new forms: the **Uncertainty diagram** $U(\phi, \lambda)$, the equal (iso) probability of error **isoper curves**, and the **nonlinearity surface** $h(\phi, \lambda)$ which becomes loop's nonlinearity S-curve when $\lambda \to 0$. The performance of overlapped schemes is degraded if classical carrier recovery (CR) schemes are employed. The theoretical results are experimentally verified on our Intersymbol-Jitter-Free (IJF) 64 kb/s modem. New equalized **DFCRL** which employs a simple 3-tap transversal equalizer in each quadrature arm is proposed for CR and the detection of overlapped schemes. Our new loop does not exhibit performance degradation associated with classical CR loops. By using the MAP probability approach and knowledge of ϕ, λ, and $h(\phi, \lambda)$ accumulated over corresponding observation intervals, improved loops,i.e. **crosstalk cancellation DFCRL, timing intersymbol interference cancellation DFCRL**, and **FEEDLOOP**, are introduced and analyzed. The theoretical results are verified by the Monte Carlo simulation. These loops perform almost as well as a continuous wave CR loop, do not exhibit quadrant ambiguities, and might be employed for phase and timing estimation and the detection of balanced and unbalanced WQAM schemes. The advantages of the new loops over the classical one become greater at lower probabilities of error and when higher state schemes are employed. E.g. at $P_e = 10^{-6}$ and in the presence of a normalized loop detuning of 0.7, the crosstalk cancellation loop (and FEEDLOOP) outperform the classical one by 2.4 dB, assuming QPSK modulation scheme is employed.

However, we are aware that our analysis is far to be complete. In practice, the total phase error might be composed of many (usually independent) components such as Doppler shift, angle modulation, instabilities of the transmitter and receiver frequency sources, additive noise (usually Gaussian), etc. Herein we dealt with the digital data modulated signals with suppressed carrier in the presence of white Gaussian noise. We analyzed the interaction of data and noise and introduced a method to wipe-off the data — in order to improve the performance. The analysis of effects of Doppler,ACI, discrete spuriouses, and other

impairments on the tracking and detection performance of the loop was not the part of this presentation. In the next chapter, we present a simplified performance analysis of the multistate modulation schemes influenced by the phase noise, i.e. instabilities of the frequency sources.

5

PERFORMANCE OF M-ARY MODULATION SYSTEMS IN THE PRESENCE OF PHASE NOISE

In the Chapter 2 WQAM family is introduced. The performance of 4 state WQAM schemes was evaluated in the presence of different impairments typical for the satellite channels such as ACI and hardlimiter nonlinearity. In the Chapter 3 the performance of 256QAM and 1024QAM schemes is evaluated in the presence of the group delay and amplitude response impairments typical for the terrestrial radio and cable systems. An a priori knowledge of the carrier phase and symbol timing has been assumed. Chapter 4 deals with the estimation of the carrier phase and symbol timing of WQAM schemes, and a few new receivers are introduced and analyzed. In this chapter, the performance of WQAM and M-ary quadrature partial response (MQPR) systems is analyzed in the presence of phase noise. Herein we adopt a practical approach, which yields useful performance curves in which the degradation due to phase noise can be readily seen.

An outline of the chapter follows. In Section 5.1 the phase noise characterisation is given. In Section 5.2 the results for probability of error performance of MPSK,MQAM and MQPR modulation systems in the presence of thermal and phase noise are presented.

5.1 PHASE NOISE CHARACTERIZATION

The expressions for the performance of coherent digital modulation systems in a Gaussian channel are well known [B15]. However,in practice we rarely have a perfect knowledge of the local phase reference. The output signals from carrier and symbol timing recovery circuits are really random processes and produce errors in the estimation of phase and time,respectively. These random processes consist of two components: first contributed by thermal white Gaussian noise ,and second contributed by all other random sources which is termed phase noise. The effects on the error probability performance due to this phase noise can be obtained in principle. However,this is a formidable task with only a few rather complex solutions existing [B3], [B12], [B16] for two- and four-state modulation systems only.
We compute the degradation of a system in the presence of phase noise in the following manner. Considering the sources of phase noise as independent random variables, the central-limit theorem says that,under certain general conditions, the resultant equivalent phase noise probability density function approaches a normal Gaussian curve as number of sources increase,[B25,p.267]. Since the number of individual sources tends to be large in practice, and their magnitudes are of the comparable order, the assumptions of the central limit theorem should apply. This is the case of a well designed system. However,in a particular example, an individual source might dictate the overall system performance and Gaussianity might be destroyed, which might result in an increased degradation of the performance. The phase noise is combined with assumed white Gaussian noise channel to produce a total carrier-to-noise ratio $(C/N)_T$ given by

$$(C/N)_T = [(N/C) + (N/C)_p]^{-1} \tag{5.1a}$$

$$(N/C)_p = (N/C)_1 + (N/C)_2 + \ldots + (N/C)_n \tag{5.1b}$$

where (N/C) is the thermal noise-to-carrier ratio, $(N/C)_i$ $(i = 1, ..., n)$ is the noise-to-carrier ratio of the i-th random source and $(N/C)_p$ is the equivalent phase noise-to-carrier ratio. We use known expressions [B15] to calculate performance of coherent digital modulation systems in the presence of phase noise by replacing (C/N) by $(C/N)_T$ and having $(C/N)_p$ as a variable parameter.
Because of a simplistic phase noise model,the accuracy of results might depend how close the assumed Gaussian probability density function fits the probability density function of the real phase noise. However,in practice we are dealing with the small amount of phase noise which causes degradations less than 1 dB at $P_e = 10^{-6}$. In that case and for our purpose the Gaussian probability density function seems to be a good aproximation for a real but unknown distribution of a phase noise. Our results of the measurement presented in Section 5.2 are in a close agreement with previous assumptions.

5.2 PERFORMANCE EVALUATION

The probability of error performance of Binary PSK (BPSK=2PSK),MPSK ($M > 2$),MQAM and MQPR modulation schemes are given by the following expresions [B15]

$$P_B \doteq \frac{1}{2} erfc(\sqrt{\gamma_b}), \qquad \text{(BPSK)} \quad (5.2)$$

$$P_M \doteq erfc(\sqrt{k\gamma_b} \sin \frac{\pi}{M}), \qquad \text{(MPSK, } M > 2\text{)} \quad (5.3)$$

$$P_L \doteq (1 - \frac{1}{L}) erfc\Big(\sqrt{\frac{3}{M-1}\frac{1}{2}\gamma_{av}}\Big), \qquad \text{(MQAM)} \quad (5.4a)$$

$$P_L \doteq (1 - \frac{1}{M}) erfc\Big(\sqrt{\frac{3}{M-1}\frac{1}{2}(\frac{\pi}{4})^2\gamma_{av}}\Big), \qquad \text{(MQPR)} \quad (5.5a)$$

$$P_M = 2P_L(1 - \frac{1}{2}P_L) \qquad \text{MQAM and MQPR} \quad (5.4b, 5.5b)$$

where [B8,p.254]

$$\gamma_b = \frac{E_b}{N_o} = \frac{C}{N}\frac{B}{f_b} \qquad (5.6)$$

$\doteq$ stands for the approximatelly equal, P_B is the probability of error performance of BPSK, P_M is the corresponding symbol error rate for the MPSK ($M > 2$), MQAM and MQPR systems, while P_L corresponds to the probability of error of the baseband signal in each of the two quadrature components of QAM or QPR. γ_{av} is the average signal-to-noise ratio per k-bit symbol, where $k = \log M$ and M is the number of levels. γ_b is the energy per bit-to-noise ratio, (C/N) is the carrier-to-thermal noise ratio, f_b is the bit rate bit/s and B is the double-sided noise bandwidth in Hz.

We assume B is equal to the double-sided Nyquist bandwidth. The expressions (5.2)-(5.5) give the probability of error performance curves illustrated in **Fig.5.1**. Inclusion of $(C/N)_p$ as a variable parameter produces a series of curves,one of which (256QAM) is shown for illustration in **Fig.5.2**. The degradations $(C/N)_T - (C/N)$ for $P_e = 10^{-6}$,dBr, vs. the $(C/N)_p$ are summarized in **Figs.5.3—5.5**. **Fig.5.3** applies for the MPSK systems, the MQAM systems are shown in **Fig.5.4** and the MQPR systems in **Fig.5.5**.

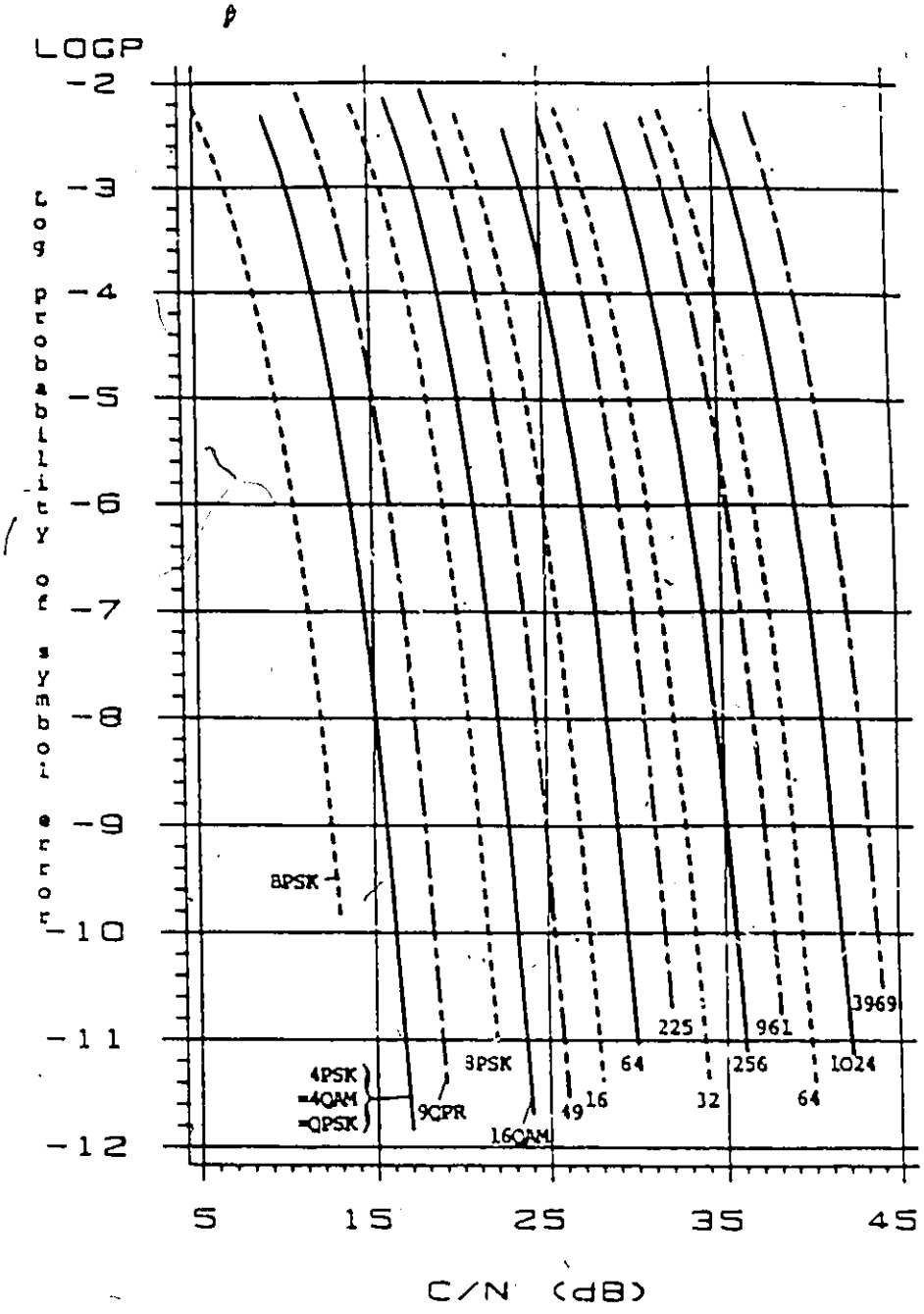


Figure 5.1. The average probability of error performance of the M-ary schemes versus the carrier–to–thermal noise ratio. The white Gaussian channel only (no phase noise). The double–sided Nyquist bandwidth. ——— MQAM schemes,

– – – – – MPSK schemes,

—·—·— MQPR schemes.

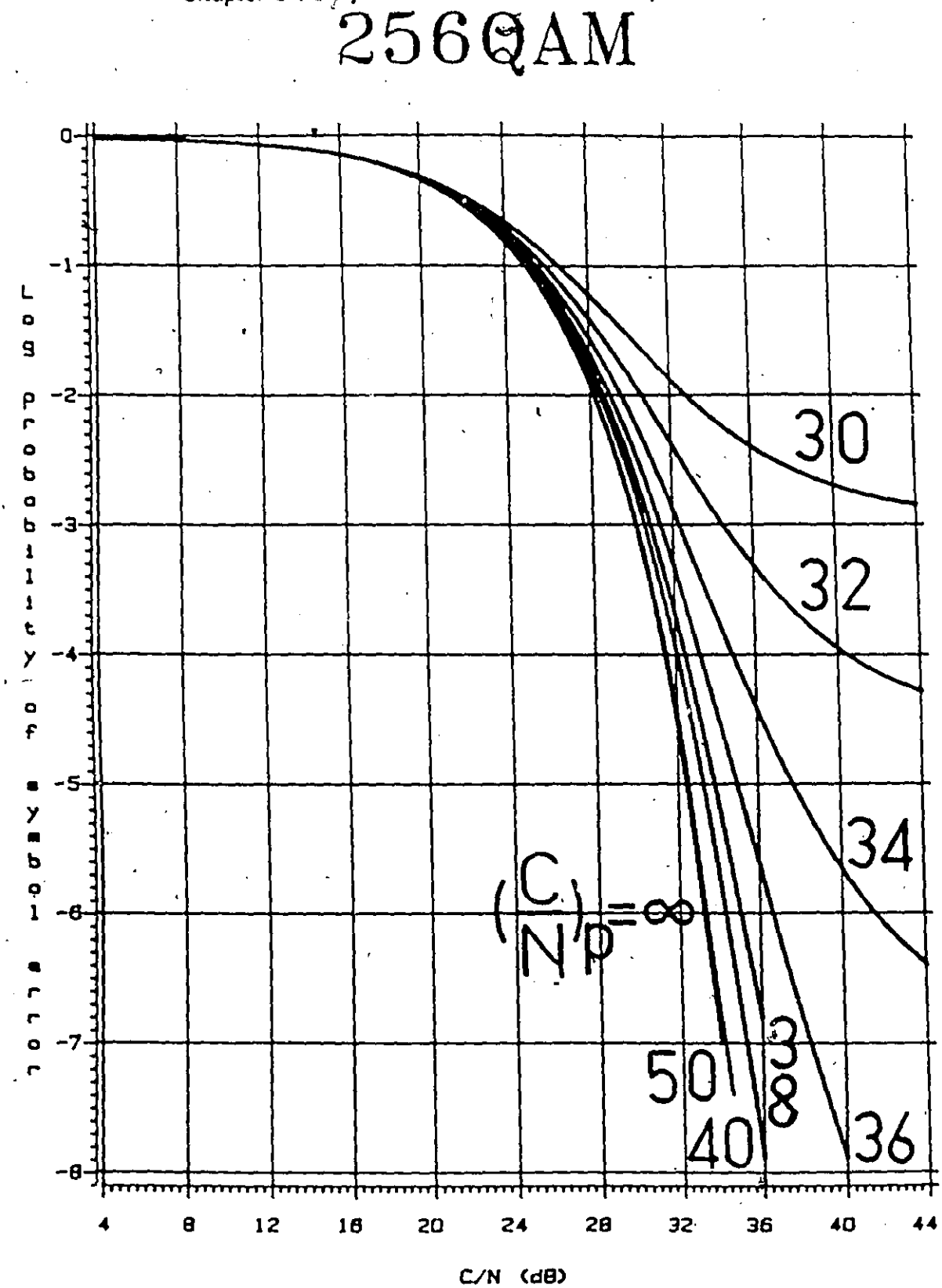


Figure 5.2. The 256QAM average probability of error performance versus the carrier-to-thermal noise ratio with the carrier-to-phase noise ratio as a parameter. The double-sided Nyquist bandwidth.

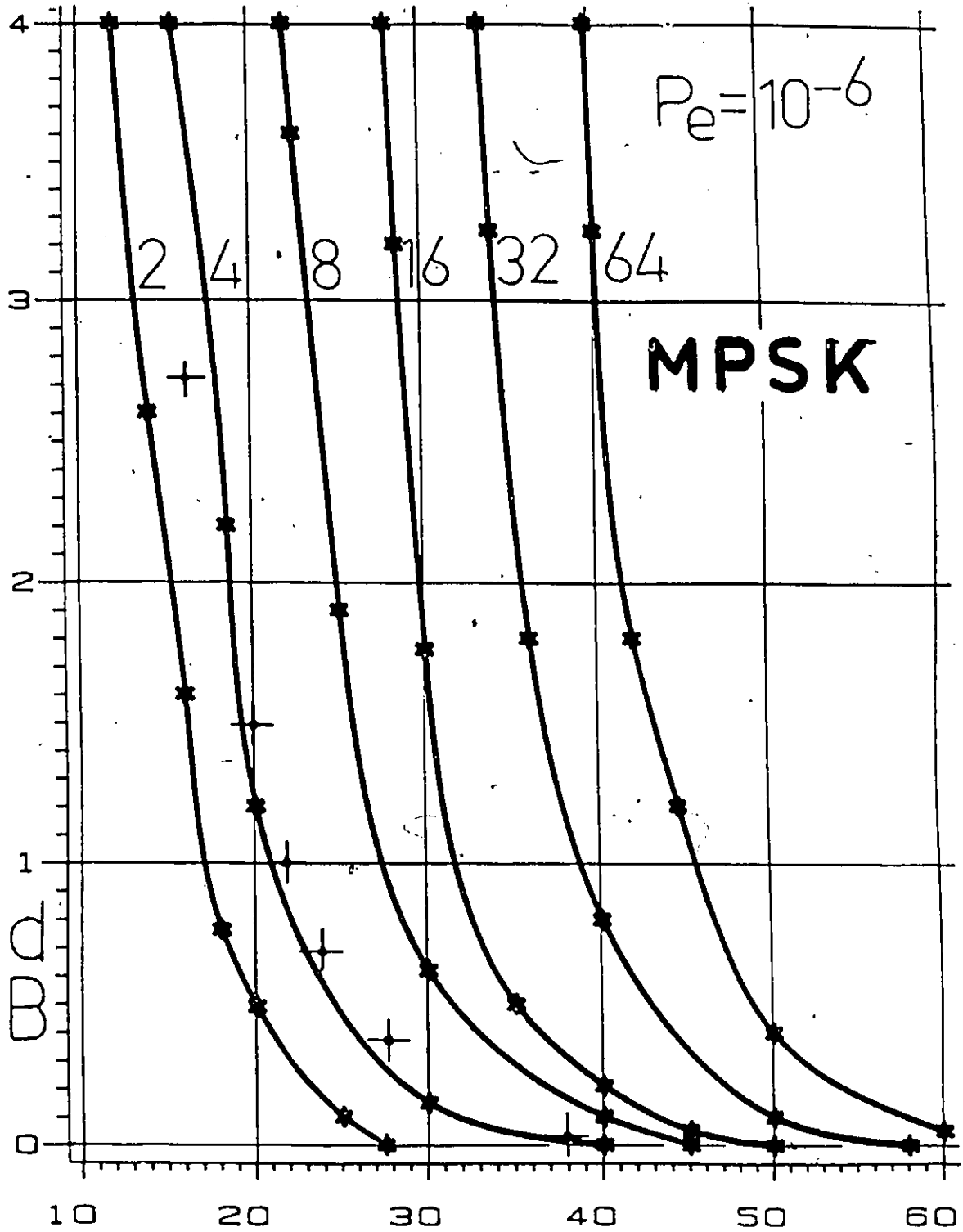


Figure 5.3. The degradation of the MPSK systems in dB, with respect to the theoretical value necessary to achieve the $P_e = 10^{-6}$ performance, versus the carrier-to-phase noise ratio in the double-sided Nyquist bandwidth in dB. The pluses (+ + +) represent the results of measurements on 4 state SCPC satellite modems.

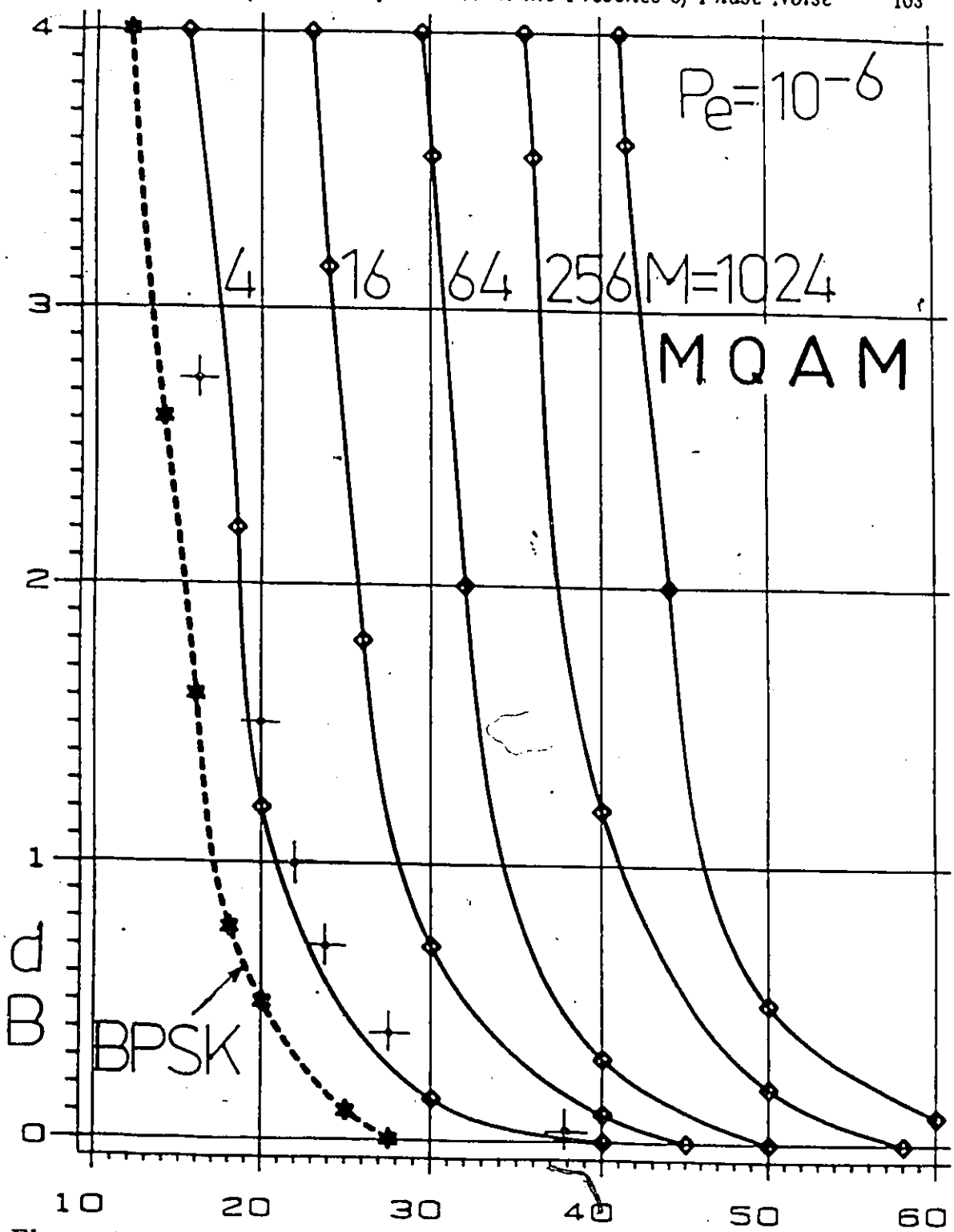


Figure 5.4. The degradation of the MQAM systems in dB, with respect to the theoretical value necessary to achieve the $P_e = 10^{-6}$ performance, versus the carrier-to-phase noise ratio in the double-sided Nyquist bandwidth in dB. The pluses (+ + +) represent the results of measurements on 4 state SCPC satellite modems.

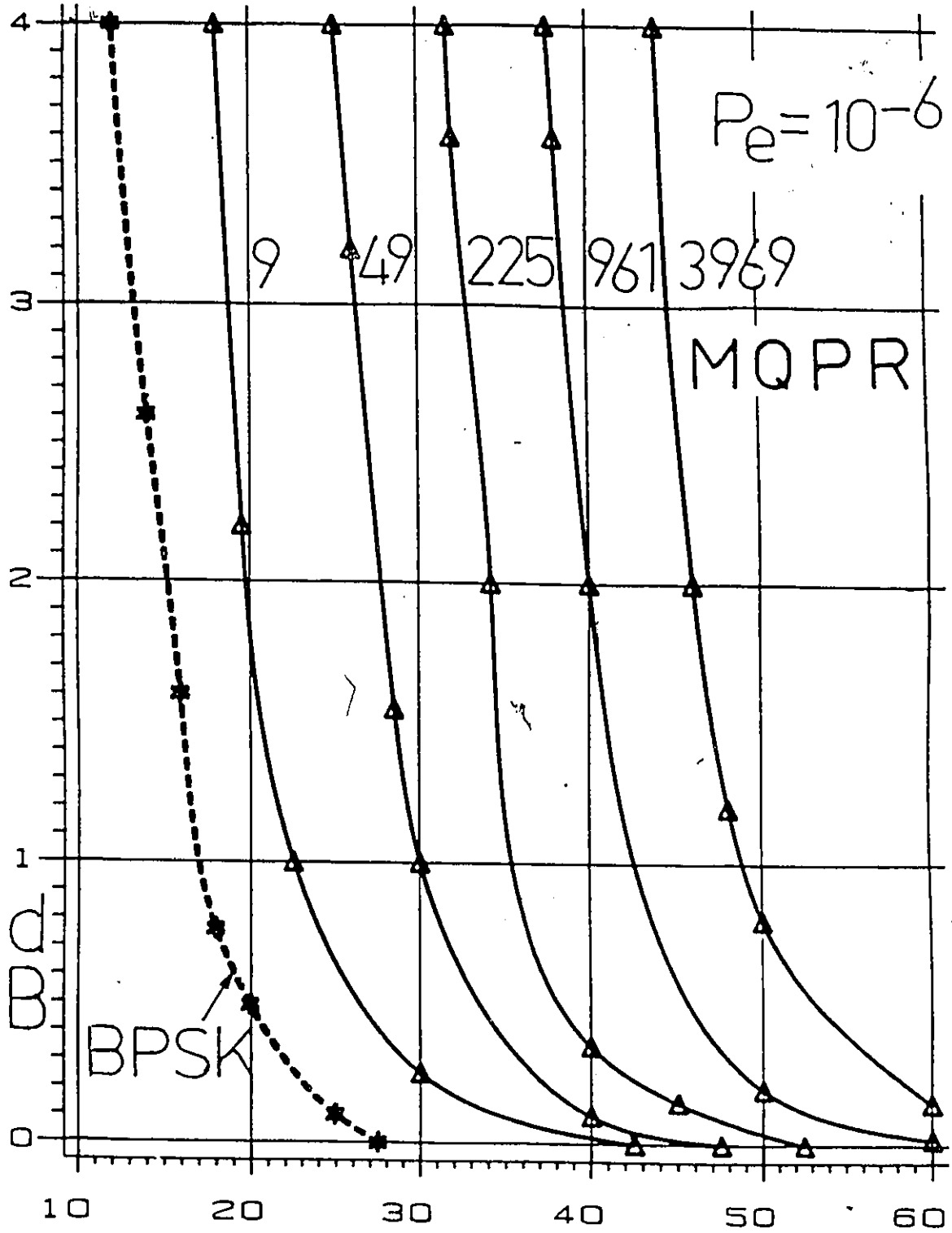


Figure 5.5. The degradation of the MQPR systems in dB, with respect to the theoretical value necessary to achieve the $P_e = 10^{-6}$ performance, versus the carrier-to-phase noise ratio in the double-sided Nyquist bandwidth in dB.

In the BPSK system,**Fig.5.3**, a $(C/N)_p > 20$ dB causes the degradation to be less than 0.5 dB. If $(C/N)_p > 30$ dB the degradation is negligible. This is an easily achievable goal,except in very low speed systems, where the spectral purity of signal source(s) could be critical [B28]. However,the 256QAM system — **Fig.5.2** and **Fig.5.4** — requires $(C/N)_p >$ 45 dB to limit the phase noise caused degradation to the amount below 1 dB. Additionally,the high-ary modulation schemes are more fragile in a multiple fading and interference environment. Further,the interference equalizer,carrier and symbol timing recovery and data generation and decision circuits are more complex for the case of high-ary modulation schemes. Consequently,the performance required of each component in the high-ary system should be higher. In all practical cases we find that if we take $(C/N)_p$ to be 10 dB higher than the $(C/N)_6$,the degradation is less than 1 dB (**Figs.5.2—5.5**). If $(C/N)_p > (C/N)_6 + 20$ dB the degradation is negligible.

Results of measurements, **Figs.5.3—5.4**, which will be briefly assessed,are in a close agreement with our engineering rule-of-thumb approximation. In a low speed data system (e.g. single-channel-per-carrier SCPC satellite link) a phase noise of frequency sources might limit an overall performance of the system. To verify this,we performed sets of measurements on SCPC satellite links consisting of two different 70 MHz 32 ksymbols/s 4-state modems (staggered QPSK and IJF), University of Ottawa earth station with 70 MHz/14 GHz up-converter and 12 GHz/70 MHz down-converter and two different "space segments". (Telesat Canada ANIK 14/12 GHz satellite transponder and a laboratory transponder with a similar performance). Each link consisted of five local oscillators of a comparable quality, which phase noises were measured by an automated spectrum analyzer. Phase noise power of each oscillator was integrated over the error transfer function [1-H(f)] of modem's carrier recovery loop (Costas loop type with a second order passive filter). The average of multiple measurements gave a noise power within the 16 kHz Nyquist bandwidth,which after multiplying by 2 and dividing by a total carrier power gave an $(N/C)_i$ of a particular phase noise contributor. Although a phase noise of frequency source itself is nonwhite (it might by represented by $\sum c_i f^{-\nu_i}$), after it passes through an "whitening filter" (i.e. error transfer function of the loop,which might be approximated by the cf^{ν} near the carrier where the noise contribution is the most important) it becomes near white,but not necessary Gaussian. To get different values of $(N/C)_i$,the frequency stability of one of five local oscillators was intentionally degraded. Even with these unbalanced phase noise sources the measured results were in a close resemblance with previously assumed curves, **Fig.5.3—5.4**.

5.3 CONCLUSION

In this chapter, the performance of MPSK,MQAM and MQPR modulation systems in the presence of phase noise and additive white Gaussian noise is presented in-graphic forms. If a carrier-to-phase noise ratio is at least 10 dB higher than a carrier-to-thermal noise ratio required for the $P_e = 10^{-6}$, the degradation due to phase noise will be less than 1 dB. If $(C/N)_p > (C/N)_6 + 20$ dB the degradation is negligible. The analysis is supported with results of measurements on 4-state SCPC satellite modems. Based on previous results a first order estimation of components requirements might be made. To minimize the degradation in the higher-ary modulation schemes the rigorous selection of components and use of advanced technologies is necessary.

6

THESIS SUMMARY

In the **Introduction**, the purpose of this study= was oulined. It was the author's goal to find power-bandwidth efficient modulation schemes, which will perform superiorly in composite ACI, linear and nonlinear, channel environments. Synthesis and analysis of both, transmitter and receiver, were performed. The results, achievements,and our own contributions which are presented in the chapters 2—5 are summarized as follows.

In the **Chapter 2**, the weighted quadrature amplitude modulation (WQAM) was introduced. By using the gradient search technique, a family of pulse shapes with a narrow mainlobe and minimal sidelobe levels within an equivalent baseband $|fT_s| = 0$ to 5 (this corresponds to 1 to 2 adjacent channels in practice) was found. The performance of 4-state WQAM schemes was evaluated in the ACI linear and nonlinear channel environments, with the channel spacing and fading depth (signal attenuation) as parameters, by means of the computer simulation. One of the new staggered WQAM schemes, termed **S3MQAM**, outperforms other known members of the WQAM family, i.e. (S)QPSK, MSK and (S)QORC in almost all practical situations.

In the **Chapter 3** the performance analysis of 256 and 1024QAM schemes is performed in the presence of amplitude and group delay impairments. A brief feasibility study of the transmission of North American T1 (DS1) 1544 kb/s or CCITT 2048 kb/s data stream over the analog 240 kHz wide CCITT supergroup is given. Staggered QAM schemes perform better in the presence of linear group delay impairments, but nonstaggered schemes are less sensitive to linear amplitude gain impairments.

In **Chapter 4**, a composite phase and timing estimation and data detection of WQAM signal sets is analyzed. By using a maximum likelihood (ML) approach, four new joint phase and timing estimators are derived. The performance of the classical decision feedback carrier recovery loop (DFCRL) employing an integrate-and-sample device in each quadrature arm (active loop) is evaluated in the presence of both carrier uncertainty ϕ and timing uncertainty λ. Results are presented in the following new forms: the **Uncertainty diagram** $U(\phi, \lambda)$, the equal (iso) probability of error **isoper curves**, and the **nonlinearity surface** $h(\phi, \lambda)$ which becomes loop's nonlinearity S-curve when $\lambda \to 0$. The performance of overlapped schemes is degraded if classical carrier recovery (CR) schemes are employed. The theoretical results are experimentally verified on our Intersymbol-Jitter-Free (IJF) 64 kb/s modem. New **equalized DFCRL** which employs a simple 3-tap transversal equalizer in each quadrature arm is proposed for CR and the detection of overlapped schemes. Our new loop does not exhibit performance degradation associated with classical CR loops. By using the MAP probability approach and knowledge of ϕ, λ, and $h(\phi, \lambda)$ accumulated over corresponding observation intervals, improved loops,i.e. **crosstalk cancellation DFCRL, timing intersymbol interference cancellation DFCRL**, and **FEEDLOOP**, are introduced and analyzed. The theoretical results are verified by the Monte Carlo simulation. These loops perform almost as well as a continuous wave CR loop, do not exhibit quadrant ambiguities, and might be

employed for phase and timing estimation and the detection of balanced and unbalanced WQAM schemes. The advantages of the new loops over the classical one become greater at lower probabilities of error and when higher state schemes are employed. E.g. at $P_e = 10^{-6}$ and in the presence of a normalized loop detuning of 0.7, the crosstalk cancellation loop (and FEEDLOOP) outperform the classical one by 2.4 dB, assuming QPSK modulation scheme is employed.

In Chapter 5, the impact of phase noise on the performance of MQAM,M-ary phase shift keying and M-ary quadrature partial response systems in a Gaussian noise environment is computed. For all cases the degradation due to phase noise is found to be less than 1 dB if a carrier-to-phase noise ratio in a double-sided Nyquist bandwidth $(C/N)_p$ is at least 10 dB higher than the carrier-to-thermal noise ratio $(C/N)_6$ required for the probability of error performance $P_e = 10^{-6}$. Performance graphs are presented which enable a fast first order approximation of the phase noise requirements of a system to be estimated. Our engineering rule-of-thumb approximations are in a close agreement with the results of measurements performed on the single-channel-per-carrier (SCPC) satellite links consisting of two different 70 MHz 32 ksymbols/s 4-state modems (SQPSK and IJF), University of Ottawa earth station with 70 MHz/14 GHz up-converter and 12 GHz/70 MHz down-converter and two different "space segments" (Telesat Canada ANIK 14/12 GHz satellite transponder and a laboratory transponder with a similar performance).

Thesis concludes with an extensive list of references, and a copy of the computer programs printout.

We highlighted some of the so far unresolved problems associated with data transmission over power and bandwidth limited nonlinear channels, we proposed new WQAM schemes at the transmitter and new CR loops at the receiver, and we have shown that our novel schemes outperform previously known devices.

7

RESEARCH PROPOSAL

This thesis outlined some of the so far unresolved problems related to the generation and reception of power-bandwidth efficient coherent digital modulation schemes. Several new modulation schemes were proposed and analyzed, including several new receivers for the corresponding WQAM schemes. In this chapter, a brief research proposal for future studies is given.

- The optimization of the signal shapes (i.e. the minimization of the spectral sidelobes) might be extended to include a cascade of nonlinearities, filters, and channel noise sources —typical in the satellite link. This might allowe one to match the modulation scheme to the particular power amplifier nonlinearity, and as a result, improve the power and spectral efficiency of the system. Modulation might be combined with coding type — block, convolutional, or Ungerböck trellis.
- The synthesis and analysis of the WQAM schemes might be extended to the multipath selective fading channels — typical for terrestrial radio systems, and to the Doppler effected channels — typical for mobile radio communications.
- The detailed study of different receivers might include stochastic Fokker-Planck-Kolmogorov partial differential equation analysis of the receiver behaviour in the presence of channel noise and intersymbol interferences, for acquisition performance during transients such as selective fading, discrete occasional interferences, etc.
- Posible application of the new decision feedback loops to coherent fiber optics and radionavigation, and further optimization of these devices might be investigated.

8

REFERENCES

A list of references consists of books (presented in the Section 8.1 and referred as Bxx), special issues (presented in the Section 8.2 and referred as Sxx) and other general publications (presented in the Section 8.3 and referred as Pxxx).
Further references on synchronous communications systems, particularly phase locked loops, might be found in

[a] Kucar, A.: Synchronous Communications Systems: Introduction and Bibliography. Elektrotehniski vestnik, Vol.49, pp:165-187, No.4, 1982. In Sloven and Croato-Serb with English abstract.

8.1 Books

[B1] Bennett, W.R., J.R.Davey: Data Transmission. McGraw-Hill Book Company, New York,1965.

[B2] Lucky, R.W., J. Salz, E.J. Weldon Jr.: Principles of Data Communication. McGraw-Hill Book Company, New York,1968.

[B3] Lindsey, W.C., M.K. Simon: Telecommunication Systems Engineering. Prentice-Hall, Inc., Englewood Cliffs, N.J.,1973.

[B4] Spilker, J.J., Jr.: Digital Communications by Satellite. Prentice-Hall, Inc., Englewood Cliffs, N.J.,1977.

[B5] Shanmugam, K.S.: Digital and Analog Communication Systems. John Wiley & Sons, Inc., New York,1979.

[B6] Bhargava, V.K., D. Haccoun, R. Matyas, P.P. Nuspl: Digital Communications by Satellite. Modulation, Multiple Access and Coding. John Wiley & Sons, Inc., New York,1981.

[B7] Feher, K.: Digital Communications, Microwave Applications. Prentice-Hall, Inc., Englewood Cliffs, N.J.,1981.

[B8] Feher, K.: Digital Communications, Satellite/Earth Station Engineering. Prentice-Hall, Inc., Englewood Cliffs, N.J.,1983.

[B9] Gagliardi, R.M.: Satellite Communications. Lifetime Learning Publications, Belmont, California, 1984.

[B10] Wu, W.W.: Elements of Digital Satellite Communications. Volume I. Computer Science Press, Rockville, Maryland, 1984.

[B11] Wu, W.W.: Elements of Digital Satellite Communications. Volume II. Computer Science Press, Rockville, Maryland, 1985.

[B12] Yuen, J.H. (Editor): Deep Space Telecommunications Systems Engineering. New York: Plenum Press, 1983.

[B13] Feher, K.: Digital Modulation Techniques in an Interference Environment. Don White Consultants, Inc., Germantown, Maryland, 1977.

[B14] Lukatela, G., D. Drajic, G. Petrovic: Digital Telecommunications. Gradjevinska knjiga, Beograd 1978. In Serbo-Croat: Digitalne Telekomunikacije.

[B15] Proakis, J.G.: Digital Communications. McGraw-Hill Book Company, New York,1983.

[B16] Lindsey, W.C.: Synchronization Systems in Communication and Control. Prentice-Hall, Inc., Englewood Cliffs, N.J.,1972.

[B17] Balakrishnan, A.V. (Series Editor), A.J. Viterbi (Volume Editor): Advances in Communication Systems. Theory and applications. Volume 4. Academic Press, Inc., New York, 1975.

[B18] Viterbi, A.J., J.K. Omura: Principles of Digital Communication and Coding. McGraw-Hill Book Company, New York,1979.

[B19] Clark, G.C., Jr., J.B.Cain: Error — Correction Coding for Digital Communications. Plenum Press, New York, 1981.

[B20] Robins, W.P.: Phase Noise in Signal Sources (Theory and Applications). London: Peter Peregrinus, 1982.

[B21] Helstrom, C.W.: Statistical Theory of Signal Detection. Second Edition. Pergamon Press, Hungary 1968.

[B22] Van Trees,H.L.: Detection, Estimation, and Modulation Theory. Part I. John Wiley & Sons, Inc., New York,1968.

[B23] Van Trees,H.L.: Detection, Estimation, and Modulation Theory. Part III. John Wiley & Sons, Inc., New York,1971.

[B24] Papoulis, A.: The Fourier Integral and its Applications. McGraw-Hill Book Company, New York,1962.

[B25] Papoulis,A.: Probability, Random Variables, and Stochastic Processes. McGraw-Hill Book Company, New York,Kogakusha, 1965.

[B26] Abramowitz, M., I.A.Stegun: Handbook of Mathematical Functions. Dover Publications, Inc., New York 1972.

[B27] Middleton, D.: An Introduction to Statistical Communication Theory. McGraw-Hill Book Company, New York,1960.

[B28] Jakes,W.C.,Jr. (Editor): Microwave Mobile Communications. John Wiley & Sons, Inc., New York,1974.

[B29] Lee,W.C.Y.: Mobile Communications Engineering. McGraw-Hill Book Company, New York,1982. 1967.

8.2 Special Issues

[S1] Lesh,J.R. (Guest Editor): Special Section on Communications over Nonlinear Channels. IEEE Trans. on Communications, Vol.COM-29, pp:537-620, May 1981.

[S2] Kullstam, P. (Guest Editor): Special Issue on Satellite Communications and Direct Broadcast Satellite (DBS) Systems. IEEE Communications Magazine, Vol.22, No.3, March 1984.

[S3] Feher, K., L.G.Greenstein, D.Lombard, L.Pollack (Guest Editors) : Special Issue on Digital Satellite Communications. IEEE Journal on Selected Areas in Communications, Vol.SAC-1, No.1, Jan. 1983.

[S4] Feher, K., R.P. Tetarenko, P.R. Hartmann, V.K. Prabhu (Guest Editors) : Special Issue on Digital Radio. IEEE Trans. on Communications, Vol.COM-27, No.12, Dec. 1979.

[S5] Anderson, J.B., J.R. Lesh (Guest Editors): Special Section on Combined Modulation and Encoding. IEEE Trans. on Communications, Vol.COM-29, No.3, March 1981.

[S6] Falconer, D.D., R.D. Gitlin (Guest Editors): Special Issue on Voiceband Telephone Data Transmission. IEEE Journal on Selected Areas in Communications, Vol.SAC-2, No.5, September 1984.

[S7] Gardner, F.M., W.C. Lindsey (Guest Editors) : Special Issue on Synchronization. IEEE Trans. on Communications, Part I, Vol.COM-28, Aug. 1980.

[S8] Lindsey, W.C., C.M.Chie (Guest Editors) : Special Issue on Phase-Locked Loops. IEEE Trans. on Communications, Vol.COM-30, Oct. 1982.

[S9] Davis, J.H. (Guest Editor) , J.J. Mikulski and P.T. Porter (Associated Guest Editors) , B.L. King (Guest Editorial Assistant) : Special Issue on Mobile Radio Communications. IEEE Journal on Selected Areas in Communications, Vol.SAC-2, No.4, July 1984.

[S10] Okumura,Y., M.Shinji (Guest Editors): Mobile Radio Communications. IEEE Communications Magazine, Vol.24, No.2, February 1986.

8.3 Other Publications

[P1] Nyquist, H.: Certain Topics in Telegraph Transmission Theory. Trans. AIEE, Vol.47, pp: 617-644, February 1928.

[P2] Cahn, C.R.: Performance of Digital Phase-Modulation Communication Systems. IRE Trans. Communication Systems, Vol.CS-7, pp: 3-6, May 1959.

[P3] Arthurs, E., H. Dym : On the Optimum Detection of Digital Signals in the Presence of White Gaussian Noise-A Geometric Interpretation and a Study of Three Basic Data Transmission Systems. IRE Trans. Communication Systems, Vol.CS-10, pp: 336-372, Dec. 1962.

[P4] Lender,A.: Correlative Digital Communication Techniques. IEEE Tr. on Communication Technology, Vol.COM-12, pp: 128-135, Dec. 1964.

[P5] DeBuda, R.: Coherent Demodulation of Frequency-Shift Keying with Low Deviation Ratio. IEEE Trans. on Communications, Vol.COM-20, pp: 429-435, June 1972.

[P6] Thomas, C.M., M.Y. Weidner, S.H. Durrani: Digital Amplitude-Phase Keying with M-ary Alphabets. IEEE Trans. on Communications, Vol.COM-22, pp: 168-180, Feb. 1974.

[P7] Mathwich, H.R., J.F. Balcewicz, M. Hecht: The Effect of Tandem Band and Amplitude Limiting on the E_b/N_o Performance of Minimum (Frequency) Shift Keying (MSK). IEEE Trans. on Communications, Vol.COM-22, pp: 1525-1540, Oct. 1974.

[P8] Gitlin, R.D., E.Y. Ho: The Performance of Staggered Quadrature Amplitude Modulation in the Presence of Phase Jitter. IEEE Trans. on Communications, Vol.COM-23, pp: 348-352, March 1975.

[P9] Hill, F.S. Jr.: Optimum Pulse Shapes for Pulse-Amplitude Modulation Data Transmission Using Vestigial Sideband Modulation. IEEE Trans. on Communications, Vol. COM-23, pp: 352-361, March 1975.

[P10] Thomas,C.M., C.L. May, G.R. Welti: Hybrid Amplitude-and-Phase Modulation for Analog Data Transmission. IEEE Trans. on Communications, Vol.COM-23, pp: 634-645, June 1975.

[P11] Amoroso, F.: Pulse and Spectrum Manipulation in the Minimum (Frequency) Shift Keying (MSK) Format. IEEE Trans. on Communications, Vol.COM-24, pp: 381-384, March 1976.

[P12] Gronemeyer, S.A., A.L. McBride: MSK and Offset QPSK Modulation. IEEE Trans. on Communications, Vol.COM-24, pp: 809-820, Aug. 1976.

[P13] Simon, M.K.: A Generalization of Minimum-Shift-Keying (MSK) Type Signaling Based Upon Input Data Symbol Pulse Shaping. IEEE Trans. on Communications, Vol.COM-24, pp: 845-856, Aug. 1976.

[P14] Amoroso, F., J.A. Kivett: Simplified MSK Signaling Technique. IEEE Trans. on Communications, Vol. COM-25, pp: 433-441, April 1977.

[P15] Prabhu, V.K.: PSK Type Modulation with Overlapping Baseband Pulses. IEEE Trans. on Communications, Vol. COM-25, pp: 980-990, Sept.1977.

[P16] Rabzel, M., S. Pasupathy: Spectral Shaping in Minimum Shift Keying (MSK)-Type Signals. IEEE Trans. on Communications, Vol. COM-26, pp: 189-195, Jan. 1978.

[P17] DeJager, F., C.B. Dekker: Tamed Frequency Modulation,A novel method to achieve spectrum economy in digital transmission. IEEE Trans. on Communications, Vol.COM-26, pp:534-542, May 1978.

[P18] Feher, K., M.Gendron: Une nouvelle famille de filtre non-lineaire. R. Can. Genie Elec. Vol.4, pp: 26-32, No.1, 1979.

[P19] Pasupathy, S.: Minimum Shift Keying:A Spectrally Efficient Modulation. IEEE Communications Magazine, Vol.17, pp: 14-22, July 1979.

[P20] Amoroso, F.: The Use of Quasi-Bandlimited Pulses in MSK Transmission. IEEE Trans. on Communications, Vol. COM-27, pp: 1616-1624, Oct. 1979.

[P21] Huang, J.C.Y., K.Feher: Techniques to Generate ISI and Jitter — Free Bandlimited Nyquist Signals and a Method to Analyze Jitter Effects. IEEE Trans. on Communications, Vol. COM-27, pp: 1700-1711, Nov. 1979.

[P22] Ryan, C.R., A.R. Hambley, E. Vogt: 760 Mbit/s Serial MSK Microwave Modem. IEEE Trans. on Communications, Vol. COM-28, pp: 771-777, May 1980.

[P23] Yazdani, H., K. Feher, W. Steenaart: Constant Envelope Bandlimited BPSK Signal. IEEE Trans. on Communications, Vol. COM-28, pp: 889-897, June 1980.

[P24] Amoroso, F.: The Bandwidth of Digital Data Signals. IEEE Communications Magazine, Vol.18, No.6, pp: 13-24, Nov. 1980.

[P25] Rhodes, S.A.: FSOQ, a New Modulation Technique that Yields a Constant Envelope. NTC'80, pp: 5111-5117, Houston, Nov-Dec. 1980.

[P26] Morais, D.H., K. Feher: The Effects of Filtering and Limiting on the Performance of QPSK, Offset QPSK, and MSK Systems. IEEE Trans. on Communications, Vol. COM-28, pp: 1999-2009, Dec. 1980.

[P27] Deshpande, G.S., P.H. Wittke: Optimum Pulse Shaping in Digital Angle Modulation. IEEE Trans. on Communications, Vol. COM-29, pp: 162-168, Feb. 1981.

[P28] Boutin, N., S. Morissette: Useful Signaling Waveforms and Related Transmit Filter Functions in Bandwidth Limited Channels. IEEE Trans. on Communications, Vol. COM-29, pp: 177-180, Feb. 1981.

[P29] Anderson, J.B., C.-E. Sundberg, T. Aulin, N. Rydbeck: Power Bandwidth Performance of Smoothed Phase Modulation Codes. IEEE Trans. on Communications, Vol. COM-29, pp: 187-195, March 1981.

[P30] Aulin, T., C.-E. Sundberg: Continuous Phase Modulation — Part I: Full Response Signaling. IEEE Trans. on Communications, Vol. COM-29, pp: 196-209. — Part II: Partial Response Signaling. pp: 210-225, March 1981.

[P31] Muilwijk, D.: Correlative Phase Shift Keying-A Class of Constant Envelope Modulation Techniques. IEEE Trans. on Communications, Vol. COM-29, pp: 226-236, March 1981.

[P32] Austin, M., M.U. Chang: Quadrature Overlapped Raised-Cosine Modulation. IEEE Trans. on Communications, Vol. COM-29, pp: 237-249, March 1981.

[P33] Wilson, S.G., R.C. Gauss: Power Spectra of Multi-h Phase Codes. IEEE Trans. on Communications, Vol. COM-29, pp: 250-256, March 1981.

[P34] Mazur, B.A., D.P. Taylor: Demodulation and Carrier Synchronization of Multi-h Phase Codes. IEEE Trans. on Communications, Vol. COM-29, pp: 257-266, March 1981.

[P35] Taylor, D.P., H.C. Chan: A Simulation Study of Two Bandwidth-Efficient Modulation Techniques. IEEE Trans. on Communications, Vol. COM-29, pp: 267-275, March 1981.

[P36] Johnston, D.A., S.K. Jones: Spectrally Efficient Communication via Fading Channels Using Coded Multilevel DPSK. IEEE Trans. on Communications, Vol. COM-29, pp: 276-284, March 1981.

[P37] Divsalar, D., M.K. Simon: The Power Spectral Density of Digital Modulations Transmitted Over Nonlinear Channels. IEEE Trans. on Communications, Vol. COM-30, pp: 142-151, Jan. 1982.

[P38] Le-Ngoc, T., K. Feher, H.P. Van: New Modulation Techniques for Low-Cost Power and Bandwidth Efficient Satellite Earth Stations. IEEE Trans. on Communications, Vol. COM-30, pp: 275-283, Jan. 1982.

[P39] Ziemer, R.E., C.R. Ryan, J.H. Stilwell: Conversion and Matched Filter Approximations for Serial Minimum-Shift Keyed Modulation. IEEE Trans. on Communications, Vol. COM-30, pp: 495-509, March 1982.

[P40] Morais, D.H., K. Feher: NLA-QAM:A Method for Generating High-Power QAM Signals Through Nonlinear Amplification. IEEE Trans. on Communications, Vol. COM-30, pp: 517-522, March 1982.

[P41] Aulin, T., C.-E. Sundberg: Partially Coherent Detection of Digital Full Response Continuous Phase Modulated Signals. IEEE Trans. on Communications, Vol. COM-30, pp: 1096-1117, May 1982.

[P42] Aulin, T., C.-E. Sundberg: Minimum Euclidian Distance and Power Spectrum for a Class of Smoothed Phase Modulation Codes with Constant Envelope. IEEE Trans. on Communications, Vol. COM-30, pp: 1721-1729, July 1982.

[P43] Aulin, T., C.-E. Sundberg: Exact Asymptotic Behavior of Digital FM Spectra. IEEE Trans. on Communications, Vol. COM-30, pp: 2438-2449, Nov. 1982.

[P44] Oshita, S., K. Feher: P_e Performance of Coherent M-ary PSK Systems in an Impulsive and Gaussian Noise Environment. IEEE Trans. on Communications, Vol. COM-30, pp: 2458-2464, Nov. 1982.

[P45] Oshita, S., K. Feher: Combined Effect of the Carrier Recovery and Symbol Timing Recovery Error on the P_e Performance of QPR and Offset QPR Systems. IEEE Trans. on Communications, Vol. COM-30, pp: 2534-2540, Dec. 1982.

[P46] Oshita, S., K. Feher: Performance of Coherent PSK and DPSK Systems in an Impulsive and Gaussian Noise Environment. IEEE Trans. on Communications, Vol. COM-30, pp: 2540-2546, Dec. 1982.

[P47] Dickerson, E.T., W.B. Warren: Generation of Constant Envelope Signals. IEEE Trans. on Communications, Vol. COM-30, pp: 2554-2556, Dec. 1982.

[P48] Le-Ngoc, T., K. Feher: Performance of IJF-OQPSK Modulation Schemes in a Complex Interference Environment. IEEE Trans. on Communications, Vol. COM-31, pp: 137-144, Jan. 1983.

[P49] Amoroso, F.: Experimental Results on Constant Envelope Signaling with Reduced Spectral Sidelobes. IEEE Trans. on Communications, Vol. COM-31, pp: 157-160, Jan. 1983.

[P50] Korn, I., B. Seth: Adjacent-Channel and Quadrature-Channel Interference in Minimum Shift Keying. IEEE Journal on Selected Areas in Communications, Vol. SAC-1, No.1, pp: 21-28, Jan. 1983.

[P51] Vandamme, P., T. Le-Ngoc, K. Feher: Performance of IJF-OQPSK and Partial Responce (PR) IJF-OQPSK Modems in a Nonlinearly Amplified and Adjacent-Channel Interference Satellite Environment. IEEE Journal on Selected Areas in Communications, Vol. SAC-1, No.1, pp: 29-38, Jan. 1983.

[P52] Austin, M.C., M.U. Chang, D.F. Horwood, R.A. Maslov: QPSK, Staggered QPSK, and MSK-A Comparative Evaluation. IEEE Trans. on Communications, Vol. COM-31, pp: 171-182, Feb. 1983.

[P53] Le-Ngoc, T., K. Feher: Performance of an IJF-OQPSK Modem in Cascaded Nonlinear and Regenerative Satellite Systems. IEEE Trans. on Communications, Vol. COM-31, pp: 296-301, Feb. 1983.

[P54] Van, H.P., K. Feher: A Class of Two-Symbol-Interval Modems for Nonlinear Radio Systems. IEEE Trans. on Communications, Vol. COM-31, pp: 433-441, March 1983.

[P55] Kato, S., K. Feher: XPSK: A New Cross-Correlated Phase-Shift Keying Modulation Technique. IEEE Trans. on Communications, Vol. COM-31, pp: 701-707, May 1983.

[P56] Seo, J.S., K. Feher: Modified Minimum Shift Keyed (MMSK) Modem for Mobile Satellite Systems. 1st Canadian Satellite Comm. Conference SCC'83, pp: 551-556.North Holland. Ottawa, June 1983.

[P57] Abe, R., I. Sasase, S. Mori: Bandwidth Efficient Quadrature Overlapped Sqared Raised-Cosine Modulation. 1st Canadian Satellite Comm. Conference SCC'83, pp: 2471-2474. Ottawa, June 1983. Distibuted by Nort Holland.

[P58] Sklar, B.: A Structured Overview of Digital Communications — a Tutorial Review — Part I. IEEE Communications Magazine, Vol.21, No.5, pp: 4-17, Aug. 1983. Part II. No.7, pp: 6-21, Oct. 1983.

[P59] Sundberg, C.-E.: On Continuous Phase Modulation in Cellular Digital Mobile Radio Systems. B.S.T.J., Vol.62, pp: 2067-2089, Sept. 1983.

[P60] Ziemer, R.E., C.R. Ryan: Minimum-Shift Keyed Modem Implementations for High Data Rates. IEEE Communications Magazine, Vol.21, No.7, pp: 28-37, Oct. 1983.

[P61] Vaisey, D.J., P.J. McLane: Passive Arm Filters in I and Q Receivers for MSK-Type Continuous Phase Modulations. IEEE Trans. on Communications, Vol. COM-31, pp: 1235-1240, Nov. 1983.

[P62] Boutin, N., S. Morissette, L. Dussault: Optimum Discrete Pulse Shaping in MSK-Type Signals. IEEE Trans. on Communications, Vol. COM-31, pp: 1251-1253, Nov. 1983.

[P63] Harris, F.J.: On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proceedings IEEE, Vol.66, No.1, pp: 51-83, Jan. 1978.

[P64] Kabal, P., S. Pasupathy: Partial-Response Signaling. IEEE Trans. on Communications, Vol. COM-23, pp: 921-934, Sept. 1975.

[P65] Pasupathy, S.: Correlative Coding. A Bandwidth Efficient Signalling Scheme. IEEE Communications Magazine, Vol. 15, pp: 4-11, July 1977.

[P66] Halpern, P.H.: Optimum Finite Duration Nyquist Signals. IEEE Trans. on Communications, Vol. COM-27, pp: 884-888, June 1979.

[P67] Korn, I.: Error Probability and Bandwidth of Digital Modulation. IEEE Trans. on Communications, Vol. COM-28, pp: 287-290, Feb. 1980.

[P68] Palmer, L.C., S.A. Rhodes, S.H. Lebowitz: Synchronization for QPSK Transmission via Communications Satellites. IEEE Trans. on Communications, Vol. COM-28, pp: 1302-1314, Aug. 1980.

[P69] Wilson, S.G.: Joint MAP Data/Phase Sequence Estimation for Trellis Phase Codes. ICC'80, Seattle, pp: 2611-2615, June 1980.

[P70] Cruz, J.R., R.S. Simpson: Cochannel and Intersymbol Interference in Quadrature-Carrier Modulation Systems. IEEE Trans. on Communications, Vol. COM-29, pp: 285-297, March 1981. Comments on... COM-30, p. 2482, Nov. 1982.

[P71] Ekanayake, N., D.P. Taylor: A Decision Feedback Receiver Structure for Bandlimited Nonlinear Channels. IEEE Trans. on Communications, Vol. COM-29, pp: 539-548, May 1981.

[P72] Devieux, C. Jr., M.E. Jones: A Practical Optimization Approach for QPSK/TDMA Satellite Channel Filtering. IEEE Trans. on Communications, Vol. COM-29, pp: 556-566, May 1981.

[P73] Fang, R.J.: Quaternary Transmission Over Satellite Channels with Cascaded Nonlinear Elements and Adjacent Channel Interference. IEEE Trans. on Communications, Vol. COM-29, pp: 567-582, May 1981.

[P74] Kennedy, D.J., O. Shimbo: Cochannel Interference in Nonlinear QPSK Satellite Channel. IEEE Trans. on Communications, Vol. COM-29, pp: 582-592, May 1981.

[P75] Huang, T.-C., J.K. Omura, W.C. Lindsey: Analysis of Coherent Satellite Communication Systems in the Presence of Interference and Noise. IEEE Trans. on Communications, Vol. COM-29, pp: 593-604, May 1981.

[P76] Korn, I., Y. Tsang: Effect of Intersymbol and Quadrature Channel Interference on Error Probability of 16-ary Offset Quadrature Amplitude Modulation with Sinusoidal and Rectangular Shaping. IEEE Trans. on Communications, Vol. COM-31, pp: 264-269, Feb. 1983.

[P77] Lim, T.L., J.K. Omura: Error Rate Estimates in Digital Communication Over a Nonlinear Channel with Memory. IEEE Trans. on Communications, Vol. COM-31, pp: 407-412, March 1983.

[P78] Laurent, P.A.: Exact and Approximate Construction of Digital Phase Modulation by Superposition of Amplitude Modulated Pulses (AMP). IEEE Trans. on Communications, Vol. COM-34, No.2, pp: 150-160, Feb. 1986. IEEE Log Number 8406879.

[P79] Natali,F.D.: Noise Performance of a Cross-Product AFC with Decision Feedback for DPSK Signals. IEEE Trans. on Communications, Vol. COM-34, No.3, pp: 303-307, March 1986. IEEE Log Number 8407419.

[P80] Helstrom,C.W.: Calculating Error Probabilities for Intersymbol and Cochannel Interference. IEEE Trans. on Communications, Vol. COM-34, No.5, pp: 430-435, May 1986. IEEE Log Number 8607998.

[P81] Padovani,R.,J.K.Wolf: Coded Phase/Frequency Modulation. IEEE Trans. on Communications, Vol. COM-34, No.5, pp: 446-453, May 1986. IEEE Log Number 8607995.

[P82] Lee,P.J.: Computation of the Bit Error Rate of Coherent M-ary PSK with Gray Code Bit Mapping. IEEE Trans. on Communications, Vol. COM-34, No.5, pp: 488-491, May 1986. IEEE Log Number 8707997.

[P83] Kam,P.Y.: Maximum Likelihood Carrier Phase Recovery for Linear Suppressed-Carrier Digital Data Modulation. IEEE Trans. on Communications, Vol. COM-34, No.6, pp: 522-527, June 1986. IEEE Log Number 8608511.

[P84] Hirosaki B.,S.Hasegawa,A.Sabato: Advanced Groupband Data Modem Using Orthogonally Multiplexed QAM Technique. IEEE Trans. on Communications, Vol. COM-34, No.6, pp: 587-592, June 1986. IEEE Log Number 8608514.

[P85] Forney,G.D.,Jr., R.G.Gallager, G.R.Lang, F.M.Longstaff, S.U.Quereshi: Efficient Modulation for Band-Limited Channels. IEEE Journal on Selected Areas in Communications, Vol. SAC-2, No.5, pp: 632-647, September 1984.

[P86] Thapar,H.K.: Real-Time Application of Trellis Coding to High-Speed Voiceband Data Transmission. IEEE Journal on Selected Areas in Communications, Vol. SAC-2, No.5, pp: 648-658, September 1984.

[P87] Hirosaki,B.: A Maximum Likelihood Receiver for an Orthogonally Multiplexed QAM System. IEEE Journal on Selected Areas in Communications, Vol. SAC-2, No.5, pp: 757-764, September 1984.

[P88] Pawula,R.F.: Asymptotics and Error Rate Bounds for M-ary DPSK. IEEE Trans. on Communications, Vol. COM-32, No.1, pp: 93-94, January 1984.

[P89] Das,J.: A Technique for Improving the Efficiency of M-ary Signaling. IEEE Trans. on Communications, Vol. COM-32, No.2, pp: 199-201, February 1984.

[P90] Pawula,R.F.: Offset DPSK and a Comparison of Conventional and Symmetric DPSK with Noise Correlation and Power Imbalance. IEEE Trans. on Communications, Vol. COM-32, No.3, pp: 233-240, March 1984.

[P91] Taylor,D.P.,M.Shafi: Decision Feedback Equalization for Multipath Induced Interference in Digital Microwave LOS Links. IEEE Trans. on Communications, Vol. COM-32, No.3, pp: 267-279, March 1984.

[P92] Hambley,A.R.,O.Tanaka: Generalized Serial MSK Modulation. IEEE Trans. on Communications, Vol. COM-32, No.3, pp: 305-308, March 1984.

[P93] Ishizuka,M.,Y.Yasuda: Improved Coherent Detection of GMSK. IEEE Trans. on Communications, Vol. COM-32, No.3, pp: 308-311, March 1984.

[P94] McMillen, G.R., M. Shafi, D.P. Taylor: Simultaneous Adaptive Estimation of Carrier Phase, Symbol Timing, and Data for a 49-QPRS DFE Radio Receiver. IEEE Trans. on Communications, Vol. COM-32, No.4, pp: 429-443, April 1984.

[P95] Boutin, N.: Constant Envelope Signal Generator with Improved Phase Linearity. IEEE Trans. on Communications, Vol. COM-32, No.4, pp: 488-490, April 1984.

[P96] Ekanayake, N.: DPSK Signaling Over Hard Limiting Channel in the Presence of Intersymbol Interference. IEEE Trans. on Communications, Vol. COM-32, No.5, pp: 503-510, May 1984.

[P97] Bello, P.A., K.Pahlavan: Adaptive Equalization for SQPSK and SQPR Over Frequency Selective Microwave LOS Channels. IEEE Trans. on Communications, Vol. COM-32, No.5, pp: 609-615, May 1984.

[P98] Biglieri, E.: High-Level Modulation and Coding for Nonlinear Satellite Channels. IEEE Trans. on Communications, Vol. COM-32, No.5, pp: 616-626, May 1984.

[P99] Sasase, I., Y.Harada, S.Mori: Bandwidth Efficient Quadrature Overlaped Modulation. IEEE Trans. on Communications, Vol. COM-32, pp: 638-640, May 1984.

[P100] Pawula, R.F.: On M-ary DPSK Transmission Over Terrestrial and Satellite Channels. IEEE Trans. on Communications, Vol. COM-32, No.7, pp: 752-761, July 1984.

[P101] Van, H.P., K.Feher: TSI-OQPSK fo. Multiple Carrier Satellite Systems. IEEE Trans. on Communications, Vol. COM-32, No.7, pp: 818-825, July 1984.

[P102] Moeneclaey, M.: A Fundamental Lower Bound on the Performance of Practical Joint Carrier and Bit Synchronizers. IEEE Trans. on Communications, Vol. COM-32, No.9, pp: 1007-1012, September 1984.

[P103] Moeneclaey, M.: Synchronizability of a General Class of PCM Formats, Including NRZ, Manchester, and Miller Coding. IEEE Trans. on Communications, Vol. COM-32, No.9, pp: 1020-1024, September 1984.

[P104] Meidan, R.: Comment on 'Optimal Design of PLL with Two Separate Phase Detectors'. IEEE Trans. on Communications, Vol. COM-32, No.9, pp: 1056-1058, September 1984. Heiman, A., Y.Bar-Ness: Autors' Reply. IEEE Trans. on Communications, Vol. COM-32, No.9, pp: 1058-1060, September 1984.

[P105] Jeruchim, M.C.: On the Application of Importance Sampling to the Simulation of Digital Satellite and Multihop Links. pp: 1088-1092, October 1984.

[P106] Moeneclaey, M.: Two Maximum – Likelihood Symbol Synchronizers with Superior Tracking Performance. IEEE Trans. on Communications, Vol. COM-32, No.11, pp: 1178-1185, November 1984.

[P107] Moeneclaey, M.: The Influence of Four Types of Symbol Synchronizers on the Error Probability of a PAM Receiver. IEEE Trans. on Communications, Vol. COM-32, No.11, pp: 1186-1190, November 1984.

[P108] Sandoz, J.P., W.Steenaart: Performance Improvement of a Binary Quantized All-Digital Phase-Locked Loop with a New Aided-Acquisition Technique. IEEE Trans. on Communications, Vol. COM-32, No.12, pp: 1269-1276, December 1984.

[P109] Wilson, S.G., H.A.Sleeper, II, P.J.Schottler, M.T.Lyons: Rate 3/4 Convolutional Coding of 16-PSK: Code Design and Performance Study. IEEE Trans. on Communications, Vol. COM-32, No.12, pp: 1308-1315, December 1984.

[P110] Aulin, T., C.-E.Sundberg: On the Minimum Euclidean Distance for a Class of Signal Space Codes. IEEE Tr. on Information Theory, Vol.IT-28, No.1, pp: 43-55, January 1982.

[P111] Ungerboeck, G.: Channel Coding with Multilevel/Phase Signals. IEEE Tr. on Information Theory, Vol.IT-28, No.1, pp: 55-67, January 1982.

[P112] Prabhu, V.K.: Modified Chernoff Bounds for PAM Systems with Noise and Interference. IEEE Tr. on Information Theory, Vol.IT-28, No.1, pp: 95-100, January 1982.

[P113] Stallings, W.: Digital Signaling Techniques. IEEE Communications Magazine, Vol.22, No.12, pp: 21-25, December 1984.

[P114] Sundberg, C.-E.: Continuous Phase Modulation. IEEE Communications Magazine, Vol.24, No.4, pp: 25-38, April 1986.

[P115] Hagmann, W.C.: Carrier synchronizer for overlapped raised cosine pulse amplitude modulation. COMSAT Tech. Review, Vol.14, No.1, pp: 25-52, Spring 1984.

[P116] Kucar, A.D., K.Feher: Performance of Weighted QAM in the Presence of Phase and Timing Error: Uncertainty Diagrams and Isoper Curves. ICDSC-7, Munchen, W. Germany, May 11-17, 1986.

[P117] Kucar, A.D., K.Feher: Performance of MQAM in the presence of CR and TR uncertainty: Part I — Wideband systems. IEEE ELECTRONICOM'85, pp: 542-545, Toronto, Oct.7-9, 1985.

[P118] Harris, R.A., P.Kristiansen: The Generation and Use of Bit Error Rate Surface Plots. IEEE Journal on Selected Areas in Comm., Vol. SAC-2, pp: 185-190, Jan.1984.

[P119] Rhodes, S.A.: Effect of Noisy Phase Reference on Coherent Detection of Offset-QPSK Signals. IEEE Trans. on Communications, Vol. COM-22, pp: 1046-1055, Aug.1974.

[P120] Davarian, F.: Imperfect phase and timing transfer effects on MSK and OQPSK signals. National Telesystems Conf., NTC'82, Galveston Texas, pp: E231-5, Nov. 7-10, 1982.

[P121] Lindsey, W.C., M.K.Simon: Carrier Synchronization and Detection of Polyphase Signals. IEEE Trans. on Communications, Vol. COM-20, pp: 441-454, June 1972.

[P122] Simon, M.K., J.G.Smith: Carrier Synchronization and Detection of QASK Signal Sets. IEEE Trans. on Communications, Vol. COM-22, pp: 98-106, Feb.1974.

[P123] Simon, M.K., J.G.Smith: Offset Quadrature Communications with Decision-Feedback Carrier Synchronization. IEEE Trans. on Communications, Vol. COM-22, pp: 1576-1584, Oct. 1974.

[P124] Simon, M.K.: Further Results on Optimmum Receiver Structures for Digital Phase and Amplitude Modulated Signals. ICC'78, Toronto, pp: 4211-4217, June 4-7, 1978.

[P125] Leclert, A., P.Vandamme: Universal Carrier Recovery Loop for QASK and PSK Signal Sets. IEEE Trans. on Communications, Vol. COM-31, pp: 130-136, Jan. 1983.

[P126] Qshita, S., K.Feher: Combined Effect of the Carrier Recovery and Sybmol Timing Recovery Error on the P_e Performance of QPR and Offset QPR Systems. IEEE Trans. on Communications, Vol. COM-30, pp: 2534-2540, Dec. 1982.

[P127] Braun, W.R., W.C.Lindsey: Carrier Synchronization Techniques for Unbalanced QPSK Signals-Part I. IEEE Trans. on Communications, Vol. COM-26, No.9, pp: 1325-1333, September 1978.

[P128] Braun, W.R., W.C.Lindsey: Carrier Synchronization Techniques for Unbalanced QPSK Signals-Part II. IEEE Trans. on Communications, Vol. COM-26, No.9, pp: 1334-1341, September 1978.

[P129] Simon, M.K., W.K.Alem: Tracking Performance of Unbalanced QPSK Demodulators: Part I-Biphase Costas Loop with Passive Arm Filters. IEEE Trans. on Communications, Vol. COM-26, No.8, pp: 1147-1156, August 1978.

[P130] Simon, M.K.: Tracking Performance of Unbalanced QPSK Demodulators: Part II-Biphase Costas Loop with Active Arm Filters. IEEE Trans. on Communications, Vol. COM-26, No.8, pp: 1157-1166, August 1978.

[P131] Kobayashi, H.: Simultaneous Adaptive Estimation and Decision Algorithm for Carrier Modulated Data Transmission Systems. IEEE Tr. on Comm. Technology, Vol. COM-19, pp: 268-280, June 1971.

[P132] Falconer, D.D., J.Salz: Optimal Reception of Digital Data over the Gaussian Channel with Unknown Delay and Phase Jitter. IEEE Tr. on Information Theory, Vol.IT-23, pp: 117-126, Jan. 1977.

[P133] Mengali, U.: Joint Phase and Timing Acquisition in Data-Transmission. IEEE Trans. on Communications, Vol. COM-25, No.10, pp: 1174-1185, October 1977.

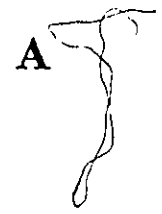
[P134] Franks, L.E.: Carrier and Bit Synchronization in Data Communication-A Tutorial Review. IEEE Trans. on Communications, Vol. COM-28, No.8, pp: 1107-1121, August 1980.

[P135] Meyers, M.H., L.E.Franks: Joint Carrier Phase and Symbol Timing Recovery for PAM Systems. IEEE Trans. on Communications, Vol. COM-28, No.8, pp: 1121-1129, August 1980.

[P136] Poklemba, J.J.: A joint estimator-detector for QPSK data transmission. COMSAT Technical Review, Vol.14, No.2, pp: 211-259, Fall 1984.

[P137] Foschini, G.T., R.D.Gitlin, S.B.Weinstein: Optimization of Two-Dimensional Signal Constellations in the Presence of Gaussian Noise. IEEE Trans. on Communications, Vol. COM-22, No.1, pp: 28-38, Jan.1974.

[P138] Booth, R.W.D.: An Illustration of the MAP Estimation Method for Deriving Closed-Loop Phase Tracking Topologies: The MSK Signal Structure. IEEE Trans. on Communications, Vol. COM-28, No.8, pp: 1137-1142, August 1980.



APPENDIX A

COMPUTER PROGRAMS PRINTOUT

This appendix contains a copy of the computer programs printout, which has been used in this thesis for the simulation purposes. Since joining Dr. Feher's Digital Communications Group at the Department of Electrical Engineering, University of Ottawa, in September 1982, the author has been developing computer software for the Amdahl 470 (IBM 370 compatible) system in the CMS environment.

The functional software has been written in the FORTRAN 66 and FORTRAN 77 languages. PLOTCOMM plotter, Hewlett-Packard hp 7470 plotter, and Tektonix 4105 color graphic terminal have been used as the output devices. The author wrote the software in the corresponding languages. In addition, SAS language has been used for graphic outputs.

The software is organized in the functional subroutines, whose model particular black-boxes (modulator, filter, channel, nonlinearity, demodulator, synchronizer, equalizer, etc. — in practice). The main programs use these subroutines (black-boxes) and make appropriate connections, according to the author's wishes. Most of the programs have been written in an interactive form, which allowes an user-friendly approach to the simulation. Program description is given in the program printout captions (comment lines).

It is assumed that a potential user of these programs is familiar with the mentioned computer languages and the CMS environment.

FILE: ASINC FORTRAN A UNIV D'/OP OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1983 08 10                                                    ASI00010
CCC   ***************************************************************     ASI00020
C     Subroutine calculates the Arcsin(x)/x function                        ASI00030
CC    Singular point Asinc(0)=1.0                                           ASI00040
CCC   ***************************************************************     ASI00050
C                                                                           ASI00060
      SUBROUTINE ASINC(X)                                                   ASI00070
      IF (X.EQ.0.) GO TO 1                                                  ASI00080
      X=ARSIN(X)/X                                                          ASI00090
      RETURN                                                                ASI00100
    1 X=1.                                                                  ASI00110
      RETURN                                                                ASI00120
      END                                                                   ASI00130
```

```
FILE: ATT      FORTRAN  A1  UNIV D'/OF OTTAWA CMS RELEASE 8

C     A.KUCAR,1984 01 04,12 08                                                  ATT00010
CC    ************************************************************************ATT00020
CC    Subroutine ATTENUATES the signal power                                    ATT00030
C     ************************************************************************ATT00040
      SUBROUTINE ATT(LD,DATA,K)                                                 ATT00050
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)                ATT00060
      COMPLEX DATA(LD)                                                          ATT00070
      AT=10.**(TIMEK(K,11)/20.)                                                 ATT00080
      DO 1 I=1,LD                                                               ATT00090
    1 DATA(I)=DATA(I)*AT                                                        ATT00100
      RETURN                                                                    ATT00110
      END                                                                       ATT00120
```

FILE: BERPHP FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1983 10 10,1985 02 25                                    BER00010
CCCCC ***************************************************************BER00020
      Subroutine DRAWS Probability of error curves for a Pe(Eb/No) as  BER00030
      low as TH                                                        BER00040
      ***************************************************************BER00050
      SUBROUTINE BERPHP(IPEN,PEI,IFR,ITO,TH)                           BER00060
      DIMENSION PEI(50)                                                BER00070
      M=ITO-IFR+1                                                      BER00080
      DO 1 K=1,M                                                       BER00090
      J=K+IFR-1                                                        BER00100
      IF(PEI(J).LT.TH) RETURN                                          BER00110
      A2=FLOAT(J)                                                      BER00120
      A1=ALOG10(PEI(J))                                                BER00130
    1 WRITE(2,2)A2,A1,IPEN                                             BER00140
    2 FORMAT(2E15.6,I2)                                                BER00150
      RETURN                                                           BER00160
      END                                                              BER00170
```

```
FILE: CHANN   . FORTRAN  A    UNIV D'/OF OTTAWA CMS RELEASE 4

CCCCCC  A.KUCAR,1984 12 20                                                      CHA00010
        ********************************************************************CHA00020
        Subroutine GENERATES the noise of                                       CHA00030
        JTYPE=1.../.GAUSSIAN                                                    CHA00040
        and ADD it to the data                                                  CHA00050
        ********************************************************************CHA00060
        SUBROUTINE CHANN(LD,DATA,SNR,JTYPE)                                     CHA00070
        COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),IT(4,4096)              CHA00080
        COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)              CHA00090
        COMPLEX DATA(LD)                                                        CHA00100
        REAL R(2)                                                               CHA00110
        DOUBLE PRECISION PID,DSEED                                              CHA00120
        PID=3.1415926535897930                                                  CHA00130
        DSEED=4569286.D0                                                        CHA00140
        DO 1 I=1,LD                                                             CHA00150
        CALL GGNML(DSEED,2,R)                                                   CHA00160
        R(1)=R(1)/SNR                                                           CHA00170
        R(2)=R(2)/SNR                                                           CHA00180
      1 DATA(I)=DATA(I)+CMPLX(R(1),R(2))                                        CHA00190
        RETURN                                                                  CHA00200
        END                                                                     CHA00210
```

FILE: CURPHP FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1984 01 26,12 08                                          CUR00010
CCC   ***************************************************************CUR00020
C     Subroutine PLOTS:1)Pe=0.5*erfc(SQRT(Eb/No))                       CUR00030
C     ***************************************************************CUR00040
      SUBROUTINE CURPHP(I,IFROM,ITO)                                    CUR00050
      DO 1 J=IFROM,ITO                                                  CUR00060
      A1=FLOAT(J)                                                       CUR00070
      E0=10.**(.1*A1)                                                   CUR00080
      A2=ERFC(SQRT(E0))/2.                                              CUR00090
      A2=ALOG10(A2)                                                     CUR00100
    1 WRITE(2,2)A1,A2,IPEN                                              CUR00110
    2 FORMAT(2E15.5,I2)                                                 CUR00120
      RETURN                                                            CUR00130
      END                                                               CUR00140
```

```
C     A.KUCAR,1984 11 19,1985 03 11                                          DEC00010
CCCCC ************************************************************************DEC00020
CCCCC Subroutine DECODRS:LARY=1...MQAM data                                   DEC00030
CCCCC                    >1...MQAM data                                       DEC00040
CCCC  CNEB=0. vs.Eb/No                                                        DEC00050
CCC      =1. vs. C/N                                                          DEC00060
C     ************************************************************************DEC00070
      SUBROUTINE DECODE(LD,DATA,PNOISE,PEI,IFR,ITO,EB,MI,MQ,LARY,CN,NER)      DEC00080
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)              DEC00090
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMER(4,4),TIMER(4,30)              DEC00100
      COMPLEX DATA(LD)                                                        DEC00110
      DIMENSION PEI(50)                                                       DEC00120
      ARY=FLOAT(LARY)                                                         DEC00130
      IF (LARY.EQ.1) ARY=2.                                                   DEC00140
      HARY=(ARY*ARY-1.)/3.0                                                   DEC00150
      IF (CN.EQ.0.) EB=EB/ALOG(ARY)*ALOG(2.)                                  DEC00160
      SNR=SQRT(2.*PNOISE*EB)                                                  DEC00170
      PEP=(1.-1./ARY)                                                         DEC00180
      NER=0                                                                   DEC00190
      DO 1 M=IFR,ITO                                                          DEC00200
    1 PEI(M)=0.0                                                              DEC00210
      IF (LARY.GT.1) GOTO 4                                                   DEC00220
C     ------------------------------------------------------------------      DEC00230
      DO 2 K=1,NSYMB                                                          DEC00240
      JI=(K-1)*LSAMPL+MI                                                      DEC00250
      JQ=(K-1)*LSAMPL+MQ                                                      DEC00260
      XB=REAL(DATA(JI))                                                       DEC00270
      YB=AIMAG(DATA(JQ))                                                      DEC00280
      INDI=0                                                                  DEC00290
      IF((NI(3,K)*XB).LE.0.) INDI=1                                           DEC00300
      INDQ=0                                                                  DEC00310
      IF((NI(4,K)*YB).LE.0.) INDQ=1                                           DEC00320
      IF(INDI.EQ.1.OR.INDQ.EQ.1) NER=NER+1                                    DEC00330
      XB=ABS(XB)/SNR                                                          DEC00340
      YB=ABS(YB)/SNR                                                          DEC00350
C     ---------Compute the probability of error for this symbol--------      DEC00360
      DO 2 M=IFR,ITO                                                          DEC00370
      XM=10.**(FLOAT(M)/20.)                                                  DEC00380
      ARG=XB*XM                                                               DEC00390
      IF (ARG.GT.12.) ARG=12.                                                 DEC00400
      EI=ERFC(ARG)/2.                                                         DEC00410
      IF (INDI.EQ.1) EI=1.-EI                                                 DEC00420
      ARG=YB*XM                                                               DEC00430
      IF (ARG.GT.12.) ARG=12.                                                 DEC00440
      EQ=ERFC(ARG)/2.                                                         DEC00450
      IF (INDQ.EQ.1) EQ=1.-EQ                                                 DEC00460
    2 PEI(M)=PEI(M)+(EI+EQ)/2.                                                DEC00470
      GOTO 9                                                                  DEC00480
C     ------------------------------------------------------------------      DEC00490
    4 HIGH=FLOAT(LARY-1)                                                      DEC00500
      DO 7 K=1,NSYMB                                                          DEC00510
      JI=(K-1)*LSAMPL+MI                                                      DEC00520
      JQ=(K-1)*LSAMPL+MQ                                                      DEC00530
      XB=REAL(DATA(JI))                                                       DEC00540
      YB=AIMAG(DATA(JQ))                                                      DEC00550
```

```
      DX1=ABS(XB)                                                       DEC00560
      DY1=ABS(YB)                                                       DEC00570
      INDI=0                                                            DEC00580
      AI=FLOAT(NI(3,K))                                                 DEC00590
      IF((YB*AI).LE.0.) INDI=1                                          DEC00600
      AI=ABS(AI)                                                        DEC00610
      THRI1=AI-1.                                                       DEC00620
      THRI2=AI+1.                                                       DEC00630
      LAGI=0                                                            DEC00640
      IF (AI.EQ.HIGH) GOTO 30                                           DEC00650
      IF (DX1.GE.THRI2.OR.DX1.LE.THRI1) INDI=1                          DEC00660
      GOTO 40                                                           DEC00670
   30 IF (DX1.LE.THRI1) INDI=1                                          DEC00680
      LAGI=1                                                            DEC00690
   40 INDQ=0                                                            DEC00700
      AI=FLOAT(NI(4,K))                                                 DEC00710
      IF((YB*AI).LE.0.) INDQ=1                                          DEC00720
      AI=ABS(AI)                                                        DEC00730
      THRQ1=AI-1.                                                       DEC00740
      THRQ2=AI+1.                                                       DEC00750
      LAGQ=0                                                            DEC00760
      IF (AI.EQ.HIGH) GOTO 50                                           DEC00770
      IF (DY1.GE.THRQ2.OR.DY1.LE.THRQ1) INDQ=1                          DEC00780
      GOTO 60                                                           DEC00790
   50 IF (DY1.LE.THRQ1) INDQ=1                                          DEC00800
      LAGQ=1                                                            DEC00810
   60 IF(INDI.EQ.1.OR.INDQ.EQ.1) NER=NER+1                              DEC00820
      IF (NER.GT.3) GOTO 9                                              DEC00830
C     ---------Compute the probability of error for this symbol--------DEC00840
      DX2=ABS(DX1-THRI2)/SNR                                            DEC00850
      DX1=ABS(DX1-THRI1)/SNR                                            DEC00860
      DY2=ABS(DY1-THRQ2)/SNR                                            DEC00870
      DY1=ABS(DY1-THRQ1)/SNR                                            DEC00880
      DO 7 M=IFR,ITO                                                    DEC00890
      XM=10.**(FLOAT(M)/20.)                                            DEC00900
      ARG=DX1*XM                                                        DEC00910
      IF (ARG.GT.12.) ARG=12.                                           DEC00920
      EI=ERFC(ARG)/PEF                                                  DEC00930
      IF (LAGI.EQ.1) GOTO 5                                             DEC00940
      ARG=DX2*XM                                                        DEC00950
      IF (ARG.GT.12.) ARG=12.                                           DEC00960
      EI=EI+ERFC(ARG)/PEF                                               DEC00970
    5 IF (INDI.EQ.1) EI=1.-EI                                           DEC00980
      ARG=DY1*XM                                                        DEC00990
      IF (ARG.GT.12.) ARG=12.                                           DEC01000
      EQ=ERFC(ARG)/PEF                                                  DEC01010
      IF (LAGQ.EQ.1) GOTO 6                                             DEC01020
      ARG=DY2*XM                                                        DEC01030
      IF (ARG.GT.12.) ARG=12.                                           DEC01040
      EQ=EQ+ERFC(ARG)/PEF                                               DEC01050
    6 IF (INDQ.EQ.1) EQ=1.-EQ                                           DEC01060
    7 PEI(M)=PEI(M)+(EI+EQ)/2.                                          DEC01070
    9 WRITE(6,3)NER                                                     DEC01080
      WRITE(8,3)NER                                                     DEC01090
    3 FORMAT(' ....ERRORS=',I5,' Try again')                            DEC01100
```

```
      RETURN                                                        DEC01110
      END                                                           DEC01120
```

FILE: DECODO FORTRAN * UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1984 11 19,1987 01 01                                          DEC00010
CCC   ********************************************************************** DEC00020
CCC   Subroutine DECODES:LARY=1...MQAM data                                  DEC00030
CCC                          >1...MQAM data                                  DEC00040
C     LD is the total number of samples                                      DEC00050
CCC   DATA is the complex data matrix                                        DEC00060
CCC   MI (MQ) is the offset in samples in the I (Q) channel                  DEC00070
C     ********************************************************************** DEC00080
      SUBROUTINE DECODO(LD,DATA,MI,MQ,LARY)                                  DEC00090
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             DEC00100
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIME1(4,4),TIMER(4,30)             DEC00110
      COMPLEX DATA(LD)                                                       DEC00120
      ARY=FLOAT(LARY)                                                        DEC00130
      IF (LARY.EQ.1) ARY=2.                                                  DEC00140
      IF (LARY.GT.2) GOTO 4                                                  DEC00150
C.....2-level schemes.....                                                   DEC00160
      DO 2 I=1,LSAMPL                                                        DEC00170
      NI(1,I)=0                                                              DEC00180
      NI(2,I)=0                                                              DEC00190
      IMI=I+MI                                                               DEC00200
      IMQ=I+MQ                                                               DEC00210
      DO 2 K=1,NSYMB                                                         DEC00220
      JI=(K-1)*LSAMPL+IMI                                                    DEC00230
      JQ=(K-1)*LSAMPL+IMQ                                                    DEC00240
      XB=REAL(DATA(JI))                                                      DEC00250
      YB=AIMAG(DATA(JQ))                                                     DEC00260
      INDI=0                                                                 DEC00270
      IF((NI(3,K)*XB).LE.0.) INDI=1                                          DEC00280
      INDQ=0                                                                 DEC00290
      IF((NI(4,K)*YB).LE.0.) INDQ=1                                          DEC00300
      NI(1,I)=NI(1,I)+INDI                                                   DEC00310
      NI(2,I)=NI(2,I)+INDQ                                                   DEC00320
C.....IF error is detected WRITE sample number,generated,received samp.      DEC00330
C     IF (INDI.EQ.1.OR.INDQ.EQ.1) WRITE(2,10) JI,NI(3,K),XB,JQ,NI(4,K)       DEC00340
C    $,YB                                                                    DEC00350
    2 CONTINUE                                                               DEC00360
      RETURN                                                                 DEC00370
C.....More than 2-level schemes.....                                        DEC00380
    4 HIGH=ARY-1                                                             DEC00390
      DO 7 I=1,LSAMPL                                                        DEC00400
      NI(1,I)=0                                                              DEC00410
      NI(2,I)=0                                                              DEC00420
      IMI=I+MI                                                               DEC00430
      IMQ=I+MQ                                                               DEC00440
      DO 7 K=1,NSYMB                                                         DEC00450
      JI=(K-1)*LSAMPL+IMI                                                    DEC00460
      JQ=(K-1)*LSAMPL+IMQ                                                    DEC00470
      XB=REAL(DATA(JI))                                                      DEC00480
      YB=AIMAG(DATA(JQ))                                                     DEC00490
      DX1=ABS(XB)                                                            DEC00500
      DY1=ABS(YB)                                                            DEC00510
      INDI=0                                                                 DEC00520
      AI=FLOAT(NI(3,K))                                                      DEC00530
      IF((XB*AI).LE.0.) INDI=1                                               DEC00540
      AI=ABS(AI)                                                             DEC00550
```

```
FILE: DECODO    FORTRAN  *   UNIV D'/OF OTTAWA CMS RELEASE 8

      THRI1=AI-1.                                                          DEC00560
      THRI2=AI+1.                                                          DEC00570
      IF (AI.EQ.HIGH) GOTO 30                                              DEC00580
      IF (DI1.GE.THRI2.OR.DI1.LE.THRI1) INDI=1                             DEC00590
      GOTO 40                                                              DEC00600
   30 IF (DI1.LE.THRI1) INDI=1                                             DEC00610
   40 INDQ=0                                                               DEC00620
      AI=FLOAT(NI(4,K))                                                    DEC00630
      IF((YB*AI).LE.0.) INDQ=1                                             DEC00640
      AI=ABS(AI)                                                           DEC00650
      THRQ1=AI-1.                                                          DEC00660
      THRQ2=AI+1.                                                          DEC00670
      IF (AI.EQ.HIGH) GOTO 50                                              DEC00680
      IF (DY1.GE.THRQ2.OR.DY1.LE.THRQ1) INDQ=1                             DEC00690
      GOTO 5                                                               DEC00700
   50 IF (DY1.LE.THRQ1) INDQ=1                                             DEC00710
    5 NI(1,I)=NI(1,I)+INDI                                                 DEC00720
      NI(2,I)=NI(2,I)+INDQ                                                 DEC00730
C.....IF error is detected WRITE sample number,generated,received samp.    DEC00740
C     IF (INDI.EQ.1.OR.INDQ.EQ.1) WRITE(2,10) JT,NI(3,K),YB,JO,NI(4,K)     DEC00750
C    *,YB                                                                  DEC00760
    7 CONTINUE                                                             DEC00770
   10 FORMAT(2(I10,I6,F10.2))                                              DEC00780
      RETURN                                                               DEC00790
      END                                                                  DEC00800
```

FILE: ENCODE FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1983 07 01,1985 03 10                                          ENC00010
CCC   ********************************************************************** ENC00020
CCC   Subroutine GENERATES two channels of:LARY=1...MQAM signal              ENC00030
CCC                                         =L...MQAM signal (M=L*L)         ENC00040
CCC   ********************************************************************** ENC00050
C                                                                            ENC00060
      SUBROUTINE ENCODE(LD,DATA,N,LARY)                                      ENC00070
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             ENC00080
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),ATMPR(4,30)             ENC00090
      COMPLEX DATA(LD)                                                       ENC00100
      DOUBLE PRECISION PID,DSEED                                             ENC00110
      REAL RI(512),RQ(512)                                                   ENC00120
      PID=3.14159265358979 3D0                                               ENC00130
      DSEED=4569286.D0                                                       ENC00140
      IF (LARY.GT.1) GOTO 9                                                  ENC00150
C     ---------------------------------------------------------------------- ENC00160
      J=1                                                                    ENC00170
      DO 1 I=1,NSYMB                                                         ENC00180
      IF (GGUBFS(DSEED).LE..5) J=-J                                          ENC00190
      NI(1,I)=J                                                              ENC00200
      IF (GGUBFS(DSEED).LT..5) J=-J                                          ENC00210
    1 NI(2,I)=J                                                              ENC00220
      DO 5 K=1,NSYMB                                                         ENC00230
      IF (TIMEL(1,1).LE.1.) GOTO 3                                           ENC00240
      KM1=K-1                                                                ENC00250
      KM2=K-2                                                                ENC00260
      IF (KM2.EQ.0) KM2=NSYMB                                                ENC00270
      IF (KM2.EQ.-1) KM2=NSYMB-1                                             ENC00280
      IF (KM1.EQ.0) KM1=NSYMB                                                ENC00290
      MI1=NI(1,KM1)                                                          ENC00300
      MQ1=NI(2,KM1)                                                          ENC00310
      MI2=NI(1,KM2)                                                          ENC00320
      MQ2=NI(2,KM2)                                                          ENC00330
      KM1=K+1                                                                ENC00340
      KM2=K+2                                                                ENC00350
      IF ((NSYMB-K).EQ.1) KM2=1                                              ENC00360
      IF (NSYMB.EQ.K) KM2=2                                                  ENC00370
      IF (NSYMB.EQ.K) KM1=1                                                  ENC00380
      LI1=NI(1,KM1)                                                          ENC00390
      LQ1=NI(2,KM1)                                                          ENC00400
      LI2=NI(1,KM2)                                                          ENC00410
      LQ2=NI(2,KM2)                                                          ENC00420
    3 IP=NI(1,K)                                                             ENC00430
      CALL WINDOW(LD,MI1,LI1,MI2,LI2,IP,RI,N,K)                              ENC00440
      IP=NI(2,K)                                                             ENC00450
      CALL WINDOW(LD,MQ1,LQ1,MQ2,LQ2,IP,RQ,N,K)                              ENC00460
      J1=(K-1)*LSAMPL                                                        ENC00470
      DO 5 I=1,LSAMPL                                                        ENC00480
    5 DATA(J1+I)=CMPLX(RI(I),RQ(I))                                          ENC00490
      GOTO 8                                                                 ENC00500
C     ---------------------------------------------------------------------- ENC00510
    9 M1=LARY-1                                                              ENC00520
      DO 2 I=1,NSYMB                                                         ENC00530
      NI(1,I)=2*INT(LARY*GGUBFS(DSEED))-M1                                   ENC00540
    2 NI(2,I)=2*INT(LARY*GGUBFS(DSEED))-M1                                   ENC00550
```

FILE: ENCODE FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
      DO 6 K=1,NSYMB                                                    ENC00560
      J1=(K-1)*LSAMPL                                                   ENC00570
      DO 6 I=1,LSAMPL                                                   ENC00580
    6 DATA(J1+I)=CMPLX(FLOAT(NI(1,K)),FLOAT(NI(2,K)))                   ENC00590
C     ------------------------------------------------------------------ ENC00600
    8 IF (N.NE.1) GOTO 4                                                ENC00610
      DO 7 I=1,NSYMB                                                    ENC00620
      NI(3,I)=NI(1,I)                                                   ENC00630
    7 NI(4,I)=NI(2,I)                                                   ENC00640
    4 IF (IOFF.EQ.0) RETURN                                             ENC00650
      L2=LD-1                                                           ENC00660
      DO 11 I=1,IOFF                                                    ENC00670
      A=AIMAG(DATA(1))                                                  ENC00680
      DO 12 L=1,L2                                                      ENC00690
   12 DATA(L)=CMPLX(REAL(DATA(L)),AIMAG(DATA(L+1)))                     ENC00700
   11 DATA(LD)=CMPLX(REAL(DATA(LD)),A)                                  ENC00710
      RETURN                                                            ENC00720
      END                                                               ENC00730
```

FILE: ENCODO FORTRAN * UNIV OF/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1987 01 01                                                    ENC00010
CCC   ***************************************************************     ENC00020
CCC   Subroutine GENERATES two channels of:MQAM signal.M=LARY**2            ENC00030
CC    ***************************************************************     ENC00040
C                                                                           ENC00050
      SUBROUTINE ENCODO(LD,DATA,M,LARY)                                     ENC00060
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)            ENC00070
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMPK(4,30)            ENC00080
      COMPLEX DATA(LD)                                                      ENC00090
      DOUBLE PRECISION PID,DSEED                                            ENC00100
      PID=3.14159265358979 3D0                                              ENC00110
      DSEED=4569286.D0                                                      ENC00120
      M1=LARY-1                                                             ENC00130
      DO 2 I=1,NSYMB                                                        ENC00140
      NI(1,I)=2*INT(LARY*GGUBFS(DSEED))-M1                                  ENC00150
    2 NI(2,I)=2*INT(LARY*GGUBFS(DSEED))-M1                                  ENC00160
      DO 6 K=1,NSYMB                                                        ENC00170
      J1=(K-1)*LSAMPL                                                       ENC00180
      DO 6 I=1,LSAMPL                                                       ENC00190
    6 DATA(J1+I)=CMPLX(FLOAT(NI(1,K)),FLOAT(NI(2,K)))                       ENC00200
C     ------------------------------------------------------------------    ENC00210
    8 IF (M.NE.1) GOTO 4                                                    ENC00220
      DO 7 I=1,NSYMB                                                        ENC00230
      NI(3,I)=NI(1,I)                                                       ENC00240
    7 NI(4,I)=NI(2,I)                                                       ENC00250
    4 IF (IOFF.EQ.0) RETURN                                                 ENC00260
      L2=LD-1                                                               ENC00270
      DO 11 I=1,IOFF                                                        ENC00280
      A=AIMAG(DATA(1))                                                      ENC00290
      DO 12 L=1,L2                                                          ENC00300
   12 DATA(L)=CMPLX(REAL(DATA(L)),AIMAG(DATA(L+1)))                         ENC00310
   11 DATA(LD)=CMPLX(REAL(DATA(LD)),A)                                      ENC00320
      RETURN                                                                ENC00330
      END                                                                   ENC00340
```

```
C     KUCAR,1986 11 09,1987 01 04                                          FEE00010
CCCCC ***************************************************************** FEE00020
CCCCC FEEDLOOP evaluates performance of FEEDLOOP by Monte Carlo method   FEE00030
CCCCC JACK=0 No  crosstalk compensation                                   FEE00040
CCCCC        1 Yes crosstalk compensation                                 FEE00050
CCCCC JIM =0 No  ISI compensation                                         FEE00060
CCCCC        1 Yes ISI compensation                                       FEE00070
CCCCC JOE =0 No coefficients definition within TREQ subrontine            FEE00080
CCCCC     =1 NRZ signal ISI due to timing uncertainty                     FEE00090
CCCCC     =2 NRZ signal ISI compensation                                  FEE00100
CCCCC ESTFI>999 Perfect estimation of the angle FI                        FEE00110
CCCCC       =X<1000 Estimation with SNR=X                                 FEE00120
CCCCC ESTLM>999 Perfect estimation of the timing SAMP                     FEE00130
CCCCC       =Y<1000 Estimation with SNR=Y                                 FEE00140
CCCCC ***************************************************************** FEE00150
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)          FEE00160
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)          FEE00170
      COMPLEX DATA(65536),TF(65536),DATA1(65536),XA(514),C1,D1            FEE00180
      DOUBLE PRECISION PID,DSEED                                          FEE00190
      DIMENSION R(2),C(129)                                               FEE00200
      PID=3.14159265358979300                                             FEE00210
      DSEED=4569286.D0                                                    FEE00220
      PI=3.14159265358979300                                              FEE00230
      PIR=180./PI                                                         FEE00240
      LSAMPL=32                                                           FEE00250
      NSYMB=2048                                                          FEE00260
      MI=LSAMPL/2                                                         FEE00270
      MI=0                                                                FEE00280
      MQ=MI                                                               FEE00290
      IOFF=0                                                              FEE00300
      SAM=FLOAT(LSAMPL)                                                   FEE00310
      BAUD=100.                                                           FEE00320
      BW=SAM*BAUD                                                         FEE00330
      LD=LSAMPL*NSYMB                                                     FEE00340
      LS2=LSAMPL/2                                                        FEE00350
      SAM1=1./SAM                                                         FEE00360
      ALD=ALOG10(FLOAT(2*NSYMB))                                          FEE00370
      JACK=1                                                              FEE00380
      JIM=1                                                               FEE00390
      JOE=1                                                               FEE00400
      ESTFI=1000.                                                         FEE00410
      ESTLM=1000.                                                         FEE00420
C.....Window definition.....                                             FEE00430
      NUMB(1,1)=1                                                         FEE00440
      TIMEL(1,1)=1.                                                       FEE00450
      TIMEK(1,1)=0.                                                       FEE00460
      AMPL(1,1)=1.                                                        FEE00470
C     Number of levels                                                    FEE00480
      LARY=2                                                              FEE00490
      ARY=FLOAT(LARY)                                                     FEE00500
C     Number of channels                                                  FEE00510
      NCH=1                                                               FEE00520
C     Calculate the average S/N for chosen symbol error rate SER          FEE00530
      SER=1.E-02                                                          FEE00540
                                                                          FEE00550
```

FILE: FEEDLOOP FORTRAN *' UNIV D'/OF OTTAWA CMS RELEASE 8

```
      CALL MERFCI(SER,EB1,IER)                                          FEE00560
      EB=EB1*SQRT(2.)                                                   FEE00570
C     Generate data                                                     FEE00580
      CALL ENCODO(LD,DATA,NCH,LARY)                                     FEE00590
C.....Generate Gaussian noise,add it to the DATA and store it in DATA1  FEE00600
      DO 1 I=1,LD                                                       FEE00610
      CALL GGNML(DSEED,2,R)                                             FEE00620
      TF(I)=CMPLX(R(1)/EB,R(2)/EB)                                      FEE00630
    1 DATA(I)=DATA(I)+TF(I)                                             FEE00640
C     Introduce ISI                                                     FEE00650
      CALL TREQ(3,LSAMPL,LD,DATA,TF,C,LS2,JOB)                          FEE00660
      DO 3 I=1,LD                                                       FEE00670
    3 DATA1(I)=DATA(I)                                                  FEE00680
C     5 degrees phase steps                                             FEE00690
      DO 99 J=1,73                                                      FEE00700
      DEG1=FLOAT(J-37)*5.                                               FEE00710
      PI=DEG1/PIR                                                       FEE00720
      CX=COS(PI)                                                        FEE00730
      SX=SIN(PI)                                                        FEE00740
C     Calculate crosstalk                                               FEE00750
      DO 2 I=1,LD                                                       FEE00760
    2 DATA(I)=CMPLX(CX*REAL(DATA1(I))+SX*AIMAG(DATA1(I)),               FEE00770
     $              CX*AIMAG(DATA1(I))-SX*REAL(DATA1(I)))              FEE00780
C.....Crosstalk and timing ISI compensation.............................FEE00790
      IF (JACK.EQ.0) GOTO 12                                            FEE00800
      IF (PI.NE.90.OR.PI.EQ.-90.) TFI=TAN(PI)                           FEE00810
      IF (PI.EQ.90..OR.PI.EQ.-90.) TFI=1.E+33                           FEE00820
      IF (ESTFI.GT.999.) GOTO 21                                        FEE00830
      CALL GGNML(DSEED,1,R)                                             FEE00840
      TFI=TFI+R(1)/SQRT(ESTFI)                                          FEE00850
   21 DO 11 I=1,LD                                                      FEE00860
   11 DATA(I)=CMPLX(REAL(DATA(I))-TFI*AIMAG(DATA(I)),                   FEE00870
     $AIMAG(DATA(I))+TFI*REAL(DATA(I)))                                 FEE00880
   12 IF (JIM.EQ.1) CALL TREQ(3,LSAMPL,LD,DATA,TF,C,LS2,2)              FEE00890
      IF (JACK.EQ.0) GOTO 14                                            FEE00900
      DO 15 I=1,LD                                                      FEE00910
   15 DATA(I)=DATA(I)*CX                                                FEE00920
   14 CONTINUE                                                          FEE00930
C     Make decision once per sample and compare with sent sample        FEE00940
      CALL DECODO(LD,DATA,MI,MQ,LARY)                                   FEE00950
C     Calculate signal e(t) before loop filter,then filter it           FEE00960
CC    DATA(I)=CMPLX(REAL(DATA(I))*AD-AIMAG(DATA(I))*RD,0.)              FEE00970
C     CALL FILTER(LD,L3,DATA,TF,IWK,N,NFIL)                             FEE00980
      DO 99 K=1,LSAMPL                                                  FEE00990
      SAMB=FLOAT(K-1)/SAM-0.5                                           FEE01000
      PEA=-ALOG10(FLOAT(NI(1,K)+NI(2,K)))+ALD                           FEE01010
      WRITE(1,7) SAMB,DEG1,PEA                                          FEE01020
      IF (K.EQ.LSAMPL) WRITE(1,7) SAMB+SAM1,DEG1,PEA                    FEE01030
   99 CONTINUE                                                          FEE01040
    7 FORMAT(9E10.3)                                                    FEE01050
   10 FORMAT (2F5.0,5X,2F5.0,I3)                                        FEE01060
      END                                                               FEE01070
```

```
FILE: FILTER   FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 8
C
C     A.KUCAR,1984 01 04,1985 05 08                                          FIL00010
C     ***********************************************************************FIL00020
C     Subroutine performs the FILTERING of data                              FIL00030
C     ***********************************************************************FIL00040
      SUBROUTINE FILTER(LD,L3,DATA,TF,IWK,N,NFIL)                            FIL00050
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(8,4096)             FIL00060
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(8,4),TIMEK(4,4),TIMEF(8,30)             FIL00070
      COMPLEX DATA(LD),TF(LD)                                                FIL00080
      INTEGER IWK(L3)                                                        FIL00090
      NF=IABS(NFIL)                                                          FIL00100
      M1=NF*3+2                                                              FIL00110
      M2=NF*2+3                                                              FIL00120
      DIN=FLOAT(LD)                                                          FIL00130
      BW1=BW/DIN                                                             FIL00140
      TWOPI=2.*PI                                                            FIL00150
      NO=NUMB(N,M1)                                                          FIL00160
      CALL FFT2C(DATA,L3,IWK)                                                FIL00170
      IFOF=IFIX(TIMEK(N,13)/BW1)                                             FIL00180
      CALL TIMESH(LD,DATA,TF,IFOF)                                           FIL00190
      GOTO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),NO                       FIL00200
    1 RETURN                                                                 FIL00210
C     ***********************************************************************FIL00220
C     Phase equalized BUTTERWORTH filter                                     FIL00230
C                          1                                                 FIL00240
C     Mod(H(f))=--------------                                               FIL00250
C                        /------------                                       FIL00260
C                 \/ 1+(f/FN)**2n                                            FIL00270
C                                                                            FIL00280
C     Arg(H(f))=0                                                            FIL00290
C     FN=3 dB point                                                          FIL00300
C     n =filter order                                                        FIL00310
C     -------------------------------------------------------------------    FIL00320
    2 FN=TIMEK(N,M2+1)/BW1                                                   FIL00330
      NA1=2*NUMB(N,M1+1)                                                     FIL00340
      TF(1)=CMPLX(1.0,0.0)                                                   FIL00350
      DO 100 I=2,NO1                                                         FIL00360
      J=I-1                                                                  FIL00370
      A2=1./SQRT(1.+(FLOAT(J)/FN)**NA1)                                      FIL00380
  100 TF(I)=CMPLX(A2,0.)                                                     FIL00390
      GOTO 99                                                                FIL00400
    3 RETURN                                                                 FIL00410
C     ***********************************************************************FIL00420
C     Phase equalized  SQUARE ROOT COSINE FILTER with arbitrary alpha        FIL00430
C                                                                            FIL00440
C     |TF(f)|=1,                                  0<=f<=(F1=FN(1-alpha))    FIL00450
C                          /-------------------------                       FIL00460
C     |TF(f)|=.5\/(1-sin(PI/2/alpha*(f/FN-1))),  F1<=f<=(F2=FN(1+alpha)     FIL00470
C                                                                            FIL00480
C     |TF(f)|=0,                                  f>=F2                     FIL00490
C     +PIL>0...x/sinx equalizer included                                     FIL00500
C     -------------------------------------------------------------------    FIL00510
    4 IF (TIMEK(N,M2).EQ.0.) TIMEK(N,M2)=0.0001                              FIL00520
      FN=TIMEK(N,M2+1)/BW1                                                   FIL00530
      F1=(1.-TIMEK(N,M2))*FN                                                 FIL00540
```

```
      P2=(1.+TIMEK(N,M2))*PN                                                    FIL00560
      IPN=IPIX(PN)                                                              FIL00570
      PN1=PLOAT(IPN)                                                            FIL00580
      IP1=IPIX(P1)+1                                                            FIL00590
      IP2=IPIX(P2)+1                                                            FIL00600
      A1=PI/2./PLOAT(IPN)                                                       FIL00610
      A2=1.                                                                     FIL00620
      A3=PI/2./TIMEK(N,M2)                                                      FIL00630
      TP(1)=CMPLX(1.0,0.0)                                                      FIL00640
      DO 110 I=2,IP1                                                            FIL00650
      IF (NPIL.LT.0) GOTO 110                                                   FIL00660
      PI=PLOAT(I-1)*A1                                                          FIL00670
      A2=PI/SIN(PI)                                                             FIL00680
  110 TP(I)=CMPLX(A2,0.0)                                                       FIL00690
      JK=IP1+1                                                                  FIL00700
      DO 120 J=JK,IP2                                                           FIL00710
      PI1=PLOAT(J-1)                                                            FIL00720
      A=A3*((PI1/PN1)-1.)                                                       FIL00730
      TP(J)=CMPLX(SQRT(0.5*(1.0-SIN(A))),0.0)                                   FIL00740
      IF (NPIL.LT.0) GOTO 120                                                   FIL00750
      PI=PI1*A1                                                                 FIL00760
      A2=PI/SIN(PI)                                                             FIL00770
      TP(J)=TP(J)*CMPLX(A2,0.0)                                                 FIL00780
  120 CONTINUE                                                                  FIL00790
      JK=IP2+1                                                                  FIL00800
      DO 130 I=JK,NO1                                                           FIL00810
  130 TP(I)=CMPLX(0.0,0.0)                                                      FIL00820
      GOTO 99                                                                   FIL00830
    5 RETURN                                                                    FIL00840
    6 RETURN                                                                    FIL00850
    7 RETURN                                                                    FIL00860
C     ************************************************************************FIL00870
C     Linear Group Delay                                                        FIL00880
    8 PI1=PI*1.E-3*TIMEK(N,M2)                                                  FIL00890
      DO 31 I=1,NO1                                                             FIL00900
      PHI=(I-1)*BW1                                                             FIL00910
      PHI1=PI1*(PHI*PHI)                                                        FIL00920
      PHI1=AMOD(PHI1,TWOPI)                                                     FIL00930
   31 TP(I)=CMPLX(COS(PHI1),-SIN(PHI1))                                         FIL00940
      GOTO 98                                                                   FIL00950
C     ************************************************************************FIL00960
C     Parabolic Group Delay                                                     FIL00970
    9 PI2=TWOPI/3.*1.E-3*TIMEK(N,M2)                                            FIL00980
      DO 32 I=1,NO1                                                             FIL00990
      PHI=(I-1)*BW1                                                             FIL01000
      PHI1=PI2*(PHI**3.)                                                        FIL01010
      PHI1=AMOD(PHI1,TWOPI)                                                     FIL01020
   32 TP(I)=CMPLX(COS(PHI1),-SIN(PHI1))                                         FIL01030
      GOTO 99                                                                   FIL01040
C     ************************************************************************FIL01050
C     Linear Gain                                                               FIL01060
   10 A1=BW1*TIMEK(N,M2)/20.                                                    FIL01070
      DO 34 I=1,NO1                                                             FIL01080
   34 TP(I)=CMPLX(10.**((I-1)*A1),0.)                                           FIL01090
      DO 41 I=NO2,LD                                                            FIL01100
```

```
FILE: FILTER   FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

   41 TF(I)=CMPLX(10.**([I-2-LD)*A1),0.)                                FIL01110
      GOTO 199                                                          FIL01120
C     ****************************************************************FIL01130
C     Parabolic Gain                                                    FIL01140
   11 A1=BW1*BW1*TIMEK(N,M2)/20.                                        FIL01150
      DO 36 I=1,NO1                                                     FIL01160
      FI=FLOAT(I-1)                                                     FIL01170
      AM1=(A1*FI*FI)                                                    FIL01180
   36 TF(I)=CMPLX(10.**AM1,0.)                                          FIL01190
      GOTO 99                                                           FIL01200
C     ****************************************************************FIL01210
   12 RETURN                                                            FIL01220
   13 RETURN                                                            FIL01230
   14 RETURN                                                            FIL01240
   15 RETURN                                                            FIL01250
C     ****************************************************************FIL01260
CCCCC RUMMLER's SELECTIVE FADING model                                  FIL01270
CCCCC H(f)=a(1-b*exp[+-j2PI(f-fo)tau])                                  FIL01280
CCCCC a=flat fading depth (normalized to 1)                             FIL01290
CCCCC B(dB)=20log[(1+b)/(1-b)] is a notch depth                         FIL01300
CCCCC b=0 (flat fading)...1 (infinite deep)                             FIL01310
CCCCC +- nonminimum,minimum phase                                       FIL01320
CCCCC fo=notch offset                                                   FIL01330
CCCCC tau=delay of second path                                          FIL01340
C     ----------------------------------------------------------------  FIL01350
   16 AR0=TWOPI*TIMEK(N,15)*1.E-3                                       FIL01360
      TT=TIMEK(N,M2+1)                                                  FIL01370
      B=10.**(ABS(TT)/20.)                                              FIL01380
      B=(B-1.)/(B+1.)                                                   FIL01390
      DO 57 I=1,NO1                                                     FIL01400
      AR1=AR0*(FLOAT(I-1)/BW1-TIMEK(N,M2))                              FIL01410
      AR2=-SIGN(1.,TT)*B*SIN(AR1)/(1.-B*COS(AR1))                       FIL01420
      AR2=ATAN(AR2)                                                     FIL01430
   57 TF(I)=SQRT(1.+B*B-2.*B*COS(AR1))*                                 FIL01440
     *CMPLX(COS(AR2),-SIN(AR2))                                         FIL01450
C     ****************************************************************FIL01460
   99 DO 20 I=NO2,LD                                                    FIL01470
   20 TF(I)=CONJG(TF(LD+2-I))                                           FIL01480
      GOTO 199                                                          FIL01490
   98 DO 21 I=NO2,LD                                                    FIL01500
   21 TF(I)=TF(LD+2-I)                                                  FIL01510
  199 DO 111 I=1,LD                                                     FIL01520
  111 DATA(I)=CONJG(DATA(I)*TF(I))                                      FIL01530
      CALL FFT2C(DATA,L3,IWK)                                           FIL01540
      DO 222 I=1,LD                                                     FIL01550
  222 DATA(I)=CONJG(DATA(I))/DIN                                        FIL01560
      RETURN                                                            FIL01570
      END                                                               FIL01580
```

```
FILE: INP1     FORTRAN A   UNIV D'/OF OTTAWA CMS RELEASE 4

C     A.KUCAR,1983 11 02,1985 04 21                                          INP00010
CCCCC ***********************************************************************INP00020
CCCCC Input parameters for WQAM,TRIANGULAR and KAISER-BESSEL signals          INP00030
CCCCC N is the channel number 1...4                                          INP00040
CCCCC AMPL(N,1-4)-amplitude parameter of the WQAM signal                     INP00050
CCCCC            -parameter of the Kaiser-Bessel signal for channels 1-4     INP00060
CCCCC TIMEL(N,1-4)-symbol duration of the signal normalized to Tsymbol       INP00070
CCCCC TIMEK(N,1-4)-frequency parameter                                       INP00080
CCCCC ***********************************************************************INP00090
C                                                                            INP00100
      SUBROUTINE INP1(N,IWAY)                                                INP00110
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NT(4,4096)             INP00120
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)             INP00130
    2 FORMAT(I1)                                                             INP00140
      DO 4 K=1,N                                                             INP00150
      DO 4 I=1,4                                                             INP00160
    4 TIMEL(K,I)=0.                                                          INP00170
      DO 55 J=1,N                                                            INP00180
      A=0.                                                                   INP00190
      TIMEK(J,1)=0.                                                          INP00200
      WRITE(8,3)J                                                            INP00210
    3 FORMAT('channel ',I1,':window TYPE:W=1 TO 4,0=Kaiser-Bessel')          INP00220
      READ(7,2)NUMB(J,1)                                                     INP00230
      KN=NUMB(J,1)                                                           INP00240
      IF (KN.GT.0) GOTO 9                                                    INP00250
      WRITE(8,8)J                                                            INP00260
    8 FORMAT('channel ',I1,':PARAMETER 1.<alpha<9.')                         INP00270
      READ(7,5)AMPL(J,1)                                                     INP00280
      WRITE(8,7)J                                                            INP00290
    7 FORMAT('channel ',I1,' L.')                                            INP00300
      READ(7,5)TIMEL(J,1)                                                    INP00310
      TIMEK(J,1)=PI*AMPL(J,1)                                                INP00320
      GOTO 55                                                                INP00330
    9 DO 55 I=1,KN                                                           INP00340
      IF (KN.EQ.I) GO TO 31                                                  INP00350
      WRITE(8,18)J,I                                                         INP00360
   18 FORMAT(I1,5X,'AMPL(',I1,').')                                          INP00370
      READ(7,5)AMPL(J,I)                                                     INP00380
      A=A+AMPL(J,I)                                                          INP00390
      GO TO 29                                                               INP00400
   31 AMPL(J,KN)=1.-A                                                        INP00410
   29 WRITE(8,25)J,I                                                         INP00420
   25 FORMAT(I1,5X,'L(',I1,').')                                             INP00430
      READ(7,5)TIMEL(J,I)                                                    INP00440
      IF (I.EQ.1) GO TO 55                                                   INP00450
      WRITE(8,11)J,I                                                         INP00460
   11 FORMAT(I1,5X,'K(',I1,').')                                             INP00470
      READ(7,5)TIMEK(J,I)                                                    INP00480
   55 CONTINUE                                                               INP00490
    5 FORMAT (3F13.7)                                                        INP00500
      WRITE(6,1)                                                             INP00510
      WRITE(8,1)                                                             INP00520
    1 FORMAT(9X,'Ampl',8X,'timeL',8X,'timeK')                                INP00530
      DO 6 J=1,N                                                             INP00540
      KN=NUMB(J,1)                                                           INP00550
```

```
FILE: INP1     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

      WRITE(8,5)(AMPL(J,I),TIMEL(J,I),TIMEK(J,I),I=1,KN)                    INP00560
    6 CONTINUE                                                              INP00570
      IF (IWAY.EQ.1) WRITE(6,5)(AMPL(J,I),TIMEL(J,I),TIMEK(J,I),I=1,KN)     INP00580
      RETURN                                                                INP00590
      END                                                                   INP00600
```

```
FILE: INP2     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

C     A.KUCAR,1983 12 13,1985 05 02                                          INP00010
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCINP00020
C     Input data for the MAIN and ACI/CCI channels                           INP00030
C     N is the number of channels,max 4                                      INP00040
C     NUMB(N,1)-modulation (window) type                                     INP00050
C          (N,4)-number of filters in each of max 4 systems,max 4            INP00060
C          (N,5)-filter type #1                                              INP00070
C          (N,6)-filter order #1                                             INP00080
C          (N,8-10)-filter #2                                                INP00090
C          (N,11-13)-filter #3                                               INP00100
C          (N,14-16)-filter #4                                               INP00110
C          (N,20)-Nonlinearity type                                          INP00120
C     TIMEK(N,5)-1st filter parameter,                                       INP00130
C          or linear GD:ns/MHz,                                              INP00140
C          or parabolic GD:ns*ns/MHz,                                        INP00150
C          or linear DG:dB/MHz,                                              INP00160
C          or parabolic DG:dB*dB/MHz,                                        INP00170
C          or notch offset:MHz                                               INP00180
C          (N,6)-filter bandwidth MHz #1, or notch depth:dB,                 INP00190
C          (N,7-8)-filter #2,                                                INP00200
C          (N,9-10)-filter #3,                                               INP00210
C          (N,11-12)-filter #4,                                              INP00220
C          (N,13)-frequency offset MHz                                       INP00230
C          (N,14)-channel level dBr                                          INP00240
C          (N,15)-2nd path delay:ns                                          INP00250
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCINP00260
C                                                                            INP00270
      SUBROUTINE INP2(N)                                                     INP00280
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             INP00290
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)             INP00300
      TIMEK(1,13)=0.                                                         INP00310
      TIMEK(1,14)=0.                                                         INP00320
      IF (N.EQ.1) GOTO 22                                                    INP00330
      DO 2 I=2,N                                                             INP00340
      WRITE(8,3)I                                                            INP00350
    3 FORMAT('Ch.',I1,':OFFSET-MHz.')                                        INP00360
      READ(7,12)TIMEK(I,13)                                                  INP00370
      WRITE(8,5)I                                                            INP00380
    5 FORMAT('Ch.',I1,':LEVEL-dBr.')                                         INP00390
    2 READ(7,12)TIMEK(I,14)                                                  INP00400
   22 IF (NUMB(2,4).GT.0) GO TO 13                                           INP00410
      DO 8 I=1,N                                                             INP00420
      WRITE(8,21)I                                                           INP00430
   21 FORMAT('Ch.',I1,' NONLINEARITY:0=linear',/,18X,'1=hardlimiter',/,      INP00440
     *18X,'2=TWTA (Intelsat)',/,18X,'3=HP',/,18X,'4=GaAs FET')               INP00450
      READ(7,4)NUMB(I,20)                                                    INP00460
      WRITE(8,7)I                                                            INP00470
    7 FORMAT('Ch.',I1,':NUMB.of filters-max 4')                              INP00480
      READ(7,4)NUMB(I,4)                                                     INP00490
      KK=NUMB(I,4)                                                           INP00500
      IF (KK.EQ.0) GO TO 8                                                   INP00510
      DO 8 J=1,KK                                                            INP00520
      M=J*3+2                                                                INP00530
      M1=J*2+3                                                               INP00540
      NUMB(I,1+M)=0                                                          INP00550
```

```
      TIMEK(I,M1)=0.                                                    INP00560
      WRITE(8,9)I,J                                                     INP00570
    9 FORMAT('Ch.',I1,'filter ',I1,' :01=Bessel',T45,'11=Parabolic gainINP00580
     #',/,15X,'02=Butterworth',T45,'12=Sine phase ripple',              INP00590
     #/,15X,'03=Chebyshev',T45,'13=Cosine phase ripple',                INP00600
     #/,15X,'04=Cosine',T45,'14=Sine amplitude ripple',                 INP00610
     #/,15X,'05=Elliptical',T45,'15=Cosine amplitude ripple',           INP00620
     #/,15X,'06=Gauss',T45,'16=Selective fading',                       INP00630
     #/,15X,'07=S-parameter',                                           INP00640
     #/,15X,'08=Linear GD',                                             INP00650
     #/,15X,'09=Parabolic GD',                                          INP00660
     #/,15X,'10=Linear gain')                                           INP00670
      READ(7,44)NUMB(I,M)                                               INP00680
      WRITE(8,57)I,J                                                    INP00690
   57 FORMAT('Ch.',I1,' filter ',I1)                                    INP00700
      IF (NUMB(I,M).EQ.8) WRITE(8,108)                                  INP00710
  108 FORMAT(15X,'linear GD:ns/MHz')                                    INP00720
      IF (NUMB(I,M).EQ.9) WRITE(8,109)                                  INP00730
  109 FORMAT(15X,' parabolic GD:ns/MHz')                                INP00740
      IF (NUMB(I,M).EQ.10) WRITE(8,110)                                 INP00750
  110 FORMAT(15X,' linear DG:dB/MHz')                                   INP00760
      IF (NUMB(I,M).EQ.11) WRITE(8,111)                                 INP00770
  111 FORMAT(15X,' parabolic DG:dB/MHz')                                INP00780
      IF (NUMB(I,M).EQ.12.OR.NUMB(I,M).EQ.14) WRITE(8,112)              INP00790
  112 FORMAT(15X,'Ripple:ns')                                           INP00800
      IF (NUMB(I,M).EQ.13.OR.NUMB(I,M).EQ.15) WRITE(8,113)              INP00810
  113 FORMAT(15X,'Ripple:dB')                                           INP00820
      IF (NUMB(I,M).EQ.12) WRITE(8,114)                                 INP00830
  114 FORMAT(15X,'Number of periods within Nyquist bandwidth')          INP00840
      IF (NUMB(I,M).EQ.16) WRITE(8,116)                                 INP00850
  116 FORMAT(15X,'Notch offset:MHz')                                    INP00860
      IF (NUMB(I,M).EQ.4) WRITE(8,106)                                  INP00870
  106 FORMAT(15X,'PARAMETER')                                           INP00880
      IF (NUMB(I,M).EQ.2) GOTO 15                                       INP00890
      READ(7,12)TIMEK(I,M1)                                             INP00900
      IF (NUMB(I,M).GT.4.AND.NUMB(I,M).LT.16) GOTO 8                    INP00910
      IF (NUMB(I,M).EQ.16) WRITE(8,118)                                 INP00920
  118 FORMAT(15X,'Notch depth:dB.Negative for nonminimum phase fading') INP00930
   15 IF (NUMB(I,M).NE.16) WRITE(8,20)BAUD                              INP00940
   20 FORMAT(5X,F9.2,' MBd---BANDWIDTH:MHz.')                           INP00950
      READ(7,12)TIMEK(I,M1+1)                                           INP00960
      IF (NUMB(I,M).EQ.2) WRITE(8,19)I,J                                INP00970
   19 FORMAT('Ch.',I1,' filter ',I1,':ORDER-max 9')                     INP00980
      IF (NUMB(I,M).LT.4) READ(7,4)NUMB(I,1+M)                          INP00990
      IF (NUMB(I,M).NE.16) GOTO 8                                       INP01000
      WRITE(8,117)                                                      INP01010
  117 FORMAT(15X,'2nd path delay:ns')                                   INP01020
      READ(7,12) TIMEK(I,15)                                            INP01030
    8 CONTINUE                                                          INP01040
   13 DO 10 I=1,N                                                       INP01050
      WRITE(6,1)                                                        INP01060
      WRITE(8,1)                                                        INP01070
    1 FORMAT(6X,'ch.NUMBER  ch.OFFSET-MHz   ch.LEVEL-dBr')              INP01080
    6 FORMAT(4X,'filter TYPE',10X,'ORDER',2X,'BANDWIDTH-MHz',12X,'PAR') INP01090
      WRITE(6,14)I,TIMEK(I,13),TIMEK(I,14)                              INP01100
```

FILE: INP2 FORTRAN A UNIV D*/OF OTTAWA CMS RELEASE 4

```
      WRITE(8,14)I,TIMEK(I,13),TIMEK(I,14)                                  INP01110
      WRITE(6,6)                                                            INP01120
      WRITE(8,6)                                                            INP01130
      K=NUMB(I,4)                                                           INP01140
      DO 10 J=1,K                                                           INP01150
      M=J*3+2                                                               INP01160
      M1=J*2+3                                                              INP01170
      WRITE(6,214)NUMB(I,M),NUMB(I,M+1),TIMEK(I,M1+1),TIMEK(I,M1)           INP01180
      WRITE(8,214)NUMB(I,M),NUMB(I,M+1),TIMEK(I,M1+1),TIMEK(I,M1)           INP01190
   10 CONTINUE                                                              INP01200
    4 FORMAT(I1)                                                            INP01210
   44 FORMAT(I2)                                                            INP01220
   12 FORMAT (F13.7)                                                        INP01230
   14 FORMAT(I15,2F15.7)                                                    INP01240
  214 FORMAT(2I15,2F15.7)                                                   INP01250
      RETURN                                                                INP01260
      END                                                                   INP01270
```

FILE: MCARL4 FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1984 12 19,1985 04 18                                          MCA00010
CCCC  ***************************************************************       MCA00020
      Subroutine DECODES the 4QAM data using MONTE CARLO techniques           MCA00030
      ***************************************************************       MCA00040
      SUBROUTINE MCARL4(LD,DATA,MC4,NERROR,MI,MQ)                             MCA00050
      COMMON /I1/IOPP,LSAMPL,NSYMB,NO1,NO2,NOMB(4,20),NI(4,4096)              MCA00060
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEI(4,4),TIMEK(4,30)              MCA00070
      COMPLEX DATA(LD)                                                        MCA00080
      DO 1 K=1,NSYMB                                                          MCA00090
      JI=(K-1)*LSAMPL+MI                                                      MCA00100
      JQ=(K-1)*LSAMPL+MQ                                                      MCA00110
      XB=NI(3,K)*REAL(DATA(JI))                                               MCA00120
      YB=NI(4,K)*AIMAG(DATA(JQ))                                              MCA00130
      IF(XB.GT.0.) GOTO 2                                                     MCA00140
      WRITE (6,4) K,MC4                                                       MCA00150
      WRITE (6,4) K,MC4                                                       MCA00160
    2 IF(YB.LT.0.) GOTO 6                                                     MCA00170
      WRITE (6,5) K,MC4                                                       MCA00180
    6 IF(XB.LE.0..OR.YB.LE.0.) NERROR=NERROR+1                                MCA00190
    1 CONTINUE                                                                MCA00200
    4 FORMAT('I channel error in symbol',I5,'   RUN',I5)                      MCA00210
    5 FORMAT('Q channel error in symbol',I5,'   RUN',I5)                      MCA00220
      RETURN                                                                  MCA00230
      END                                                                     MCA00240
```

```
FILE: MDFL1     FORTRAN  A    UNIV D'/OF OTTAWA CMS RELEASE 8
C       A.KUCAR,1985 10 27,1986 04 25                                          MDF00010
C       Program MDFL FORTRAN                                                   MDF00020
C       This program calculates the NONLINEARITY SURFACE and CONDITIONAL       MDF00030
C       PROBABILITY OF ERROR of decision feedback carrier recovery loop        MDF00040
C       for any M-ary QAM (staggered and non-staggered) scheme                 MDF00050
C       Pe=Pea is calculated rather than Pe=Pea+Peb-Pea*Peb                    MDF00060
C       Pe(PI=carrier recovery error,SAMB=timing recovery error)               MDF00070
C       The output of this program is used as input to the SAS programs        MDF00080
C       DFL,G1,G2,GPI,NI and GPIN                                              MDF00090
C       For M>4 the argument in "erfc(arg)" function is out of range for       MDF00100
C       particular values of PI and AMB,which causes an error message          MDF00110
        COMMON L,L2,LM1,LM2,LM1SQ,L2L,L4L                                      MDF00120
        COMMON PEA,PEB,EBN,RA,RA1,RB,RB1,G1,G2,AN1,PL,P1,P2,P3                 MDF00130
        COMMON PLM2,PEA1,PEB1,PEI1,PEI2,PRJ1,RAL,RBL,GPI,GPIN                  MDF00140
        INTEGER IER                                                            MDF00150
        REAL DBLIN,F,AX,AY,BX,BY,AERR,ERROR,MO                                 MDF00160
        EXTERNAL F,AY,BY                                                       MDF00170
        AERR=0.001                                                             MDF00180
        AY=-0.5                                                                MDF00190
C       CALL ERRSET(208,700,-1,1,1,208)                                        MDF00200
        PI=3.14159265358979 3                                                  MDF00210
        PIR=180./PI                                                            MDF00220
C       WRITE(8,10)                                                            MDF00230
C       READ(7,2)IXP                                                           MDF00240
        IXP=10                                                                 MDF00250
        MIXP=100*IXP                                                           MDF00260
C       WRITE(8,1)                                                             MDF00270
C       READ(7,2)L                                                             MDF00280
        L=2                                                                    MDF00290
C       WRITE(7,101)                                                           MDF00300
C       READ(7,2)ISTAGG                                                        MDF00310
        ISTAGG=0                                                               MDF00320
        L2=L*2                                                                 MDF00330
        LM1=L-1                                                                MDF00340
        LM2=L-2                                                                MDF00350
        PL=FLOAT(L)                                                            MDF00360
        PM=PL*PL                                                               MDF00370
        P1=3.0/(PM-1.)                                                         MDF00380
        P2=1./PM/PM/2.                                                         MDF00390
        P3=P2*P1                                                               MDF00400
        PEP=PL                                                                 MDF00410
        PEP=10.**(-IXP)*PEP                                                    MDF00420
        CALL MERFCI(PEP,EBNO,IER)                                              MDF00430
        EBN=EBNO                                                               MDF00440
        IF (L.GT.0) GOTO 77                                                    MDF00450
        PEA=FLOAT(IXP)                                                         MDF00460
        PEPA=PEA                                                               MDF00470
    4   EBN=EBN*PEPA/PEA                                                       MDF00480
        PI=0.                                                                  MDF00490
        AMB=0.                                                                 MDF00500
        RA1=1.-AMB                                                             MDF00510
        RA=AMB                                                                 MDF00520
        RB=RA                                                                  MDF00530
        RB1=RA1                                                                MDF00540
        X=P                                                                    MDF00550
```

```
      NPE=INT(100.*PEA)                                                 MDP00560
C     WRITE(8,3)NPE,EBN                                                 MDP00570
      IF (NPE.NE.MIXP) GOTO 4                                           MDP00580
C     WRITE(8,8)EBN,IXP                                                 MDP00590
   77 DO 5 I=1,25                                                       MDP00600
C     WRITE(8,6)I                                                       MDP00610
      SAMB=FLOAT(I-13)/24.                                              MDP00620
      BX=SAMB                                                           MDP00630
      AMB=ABS(SAMB)                                                     MDP00640
      RA1=1.-AMB                                                        MDP00650
      RA=AMB                                                            MDP00660
      IF (ISTAGG.EQ.0) RB=RA                                            MDP00670
      IF (ISTAGG.EQ.0) RB1=RA1                                          MDP00680
      IF (ISTAGG.EQ.1) RB=0.5+AMB                                       MDP00690
      IF (ISTAGG.EQ.1) RB1=0.5-AMB                                      MDP00700
      DO 5 J=46,46                                                      MDP00710
      DEG1=FLOAT(J-46)*4.                                               MDP00720
      PI=DEG1/PIR                                                       MDP00730
      Y=P                                                               MDP00740
    5 WRITE(1,7)SAMB,DEG1,PEA,G1,G2,AN1,GPI,GPIN                        MDP00750
    1 FORMAT('# of levels L=04')                                        MDP00760
    2 FORMAT(I2)                                                        MDP00770
    3 FORMAT('NPE and EbNo=',I9,E16.3)                                  MDP00780
    6 FORMAT('I=',I4)                                                   MDP00790
    7 FORMAT(8E10.3)                                                    MDP00800
    8 FORMAT('EbNo=',F7.2,' at Pe=1E-',I2)                              MDP00810
   10 FORMAT('BER exp=06')                                              MDP00820
  101 FORMAT('Staggered=01,Non-staggered=10')                           MDP00830
      END                                                               MDP00840
C     ================================================================  MDP00850
      REAL FUNCTION F(FI,AMB)                                           MDP00860
      COMMON L,L2,LM1,LM2,LM1SQ,I2L,L4L                                 MDP00870
      COMMON PEA,PEB,EBN,RA,RA1,RB,RB1,G1,G2,AN1,PL,P1,P2,P3            MDP00880
      COMMON PLM2,PEA1,PEB1,PEI1,PEI2,PEJ1,PAL,PBL,GPI,GPIN             MDP00890
      PEB=0.                                                            MDP00900
      PEI2=0.                                                           MDP00910
      S1=0.                                                             MDP00920
      S2=0.                                                             MDP00930
      C1=0.                                                             MDP00940
      C2=0.                                                             MDP00950
      CX=COS(FI)                                                        MDP00960
      SX=SIN(FI)                                                        MDP00970
C     ----------------------------------------------------------------  MDP00980
      DO 14 M0=1,L2,2                                                   MDP00990
      M=M0-L                                                            MDP01000
      FM1=FLOAT(M-1)                                                    MDP01010
      FP1=FM1+2.                                                        MDP01020
      RAM1=M*RA1                                                        MDP01030
      IM=IABS(M)                                                        MDP01040
      PEN=0.                                                            MDP01050
      PEIN=0.                                                           MDP01060
      DO 12 N0=1,L2,2                                                   MDP01070
      N=N0-L                                                            MDP01080
      RAN1=N*RB1                                                        MDP01090
      PEA1=0.                                                           MDP01100
```

```
      DO 11 M9=1,L2,2                                                    MDP01110
      M1=M9-L                                                            MDP01120
      RAM=M1*RA                                                          MDP01130
      YA1=(RAM1+RAM)*CX                                                  MDP01140
      DO 11 N9=1,L2,2                                                    MDP01150
      N1=N9-L                                                            MDP01160
      RAN=N1*RB                                                          MDP01170
      YB1=(RAN1+RAN)*SX                                                  MDP01180
      YAB1=YA1+YB1                                                       MDP01190
      IF (M0.EQ.1) PEA=ERFC((FP1-YAB1)*EBN)                              MDP01200
      IF (M.EQ.LM1) PEA=ERFC((YAB1-FM1)*EBN)                             MDP01210
      IF (M.LT.LM1) PEA=ERFC((YAB1-FM1)*EBN)-ERFC((FP1-YAB1)*EBN)        MDP01220
   11 PEA1=PEA1+PEA                                                      MDP01230
      PEN=PEN+PEA1                                                       MDP01240
   12 PEIN=PEIN+M*PEA1                                                   MDP01250
      PEB=PEB+PEN                                                        MDP01260
      G2=G2+M*PEIN                                                       MDP01270
      C1=C1+M*M*PEN                                                      MDP01280
      DO 14 I9=1,L2,2                                                    MDP01290
      IF (I9.EQ.M0) GOTO 14                                              MDP01300
      I1=I9-L                                                            MDP01310
      FPI1=FLOAT(I1+1)                                                   MDP01320
      FMI1=FPI1-2.                                                       MDP01330
      IA=IABS(I1)                                                        MDP01340
      PENI=0.                                                            MDP01350
      PEJ1=0.                                                            MDP01360
      DO 13 N0=1,L2,2                                                    MDP01370
      N=N0-L                                                             MDP01380
      RAN1=N*RB1                                                         MDP01390
      PEI1=0.                                                            MDP01400
      DO 16 M9=1,L2,2                                                    MDP01410
      M1=M9-L                                                            MDP01420
      RAM=M1*RA                                                          MDP01430
      YA1=(RAM1+RAM)*CX                                                  MDP01440
      DO 16 N9=1,L2,2                                                    MDP01450
      N1=N9-L                                                            MDP01460
      RAN=N1*RB                                                          MDP01470
      YB1=(RAN1+RAN)*SX                                                  MDP01480
      YAB1=YA1+YB1                                                       MDP01490
      IF (I9.EQ.1) PEI=ERFC((YAB1-FPI1)*EBN)                             MDP01500
      IF (I1.EQ.LM1) PEI=ERFC((FMI1-YAB1)*EBN)                           MDP01510
      IF ((IA.LT.LM1).AND.(I1.GT.M))                                     MDP01520
     %PEI=ERFC((FMI1-YAB1)*EBN)-ERFC((FPI1-YAB1)*EBN)                    MDP01530
      IF ((IA.LT.LM1).AND.(I1.LT.M))                                     MDP01540
     %PEI=ERFC((YAB1-FPI1)*EBN)-ERFC((YAB1-FMI1)*EBN)                    MDP01550
   16 PEI1=PEI1+PEI                                                      MDP01560
      PENI=PENI+PEI1                                                     MDP01570
   13 PEJ1=PEJ1+N*PEI1                                                   MDP01580
      PEI2=PEI2+I1*I1*PENI                                               MDP01590
      G2=G2-I1*PEJ1                                                      MDP01600
      C2=C2+I1*M*PENI                                                    MDP01610
   14 CONTINUE                                                           MDP01620
      G1=C1-C2                                                           MDP01630
      AN1=C1-PEI2                                                        MDP01640
C-----------------------------------------------------------------       MDP01650
```

```
      DO 24 M0=1,L2,2                                                    MDP01660
      M=M0-L                                                             MDP01670
      FM1=FLOAT(M-1)                                                     MDP01680
      FP1=FM1+2.                                                         MDP01690
      RAM1=M*RA1                                                         MDP01700
      IM=IABS(M)                                                         MDP01710
      PEIN=0.                                                            MDP01720
      PEN=0.                                                             MDP01730
      DO 22 N0=1,L2,2                                                    MDP01740
      N=N0-L                                                             MDP01750
      RAN1=N*RB1                                                         MDP01760
      PEA1=0.                                                            MDP01770
      DO 21 M9=1,L2,2                                                    MDP01780
      M1=M9-L                                                            MDP01790
      RAM=M1*RA                                                          MDP01800
      YA1=(RAM1+RAM)*CX                                                  MDP01810
      DO 21 N9=1,L2,2                                                    MDP01820
      N1=N9-L                                                            MDP01830
      RAN=N1*RB                                                          MDP01840
      YB1=(RAN1+RAN)*SX                                                  MDP01850
      YAB1=YA1-YB1                                                       MDP01860
      IF (M0.EQ.1) PEA=ERFC((FP1-YAB1)*EBN)                              MDP01870
      IF (M.EQ.LM1) PEA=ERFC((YAB1-FM1)*EBN)                             MDP01880
      IF (IM.LT.LM1) PEA=ERFC((YAB1-FM1)*EBN)-ERFC((FP1-YAB1)*EBN)       MDP01890
   21 PEA1=PEA1+PEA                                                      MDP01900
      PEN=PEN+PEA1                                                       MDP01910
   22 PEIN=PEIN+N*PEA1                                                   MDP01920
      G2=G2-M*PEIN                                                       MDP01930
      DO 24 I9=1,L2,2                                                    MDP01940
      IF (I9.EQ.M0) GOTO 24                                              MDP01950
      I1=I9-L                                                            MDP01960
      FPI1=FLOAT(I1+1)                                                   MDP01970
      FMI1=FPI1-2.                                                       MDP01980
      IA=IABS(I1)                                                        MDP01990
      PEMI=0.                                                            MDP02000
      PEJ1=0.                                                            MDP02010
      DO 23 N0=1,L2,2                                                    MDP02020
      N=N0-L                                                             MDP02030
      RAN1=N*RB1                                                         MDP02040
      PEI1=0.                                                            MDP02050
      DO 26 M9=1,L2,2                                                    MDP02060
      M1=M9-L                                                            MDP02070
      RAM=M1*RA                                                          MDP02080
      YA1=(RAM1+RAM)*CX                                                  MDP02090
      DO 26 N9=1,L2,2                                                    MDP02100
      N1=N9-L                                                            MDP02110
      RAN=N1*RB                                                          MDP02120
      YB1=(RAN1+RAN)*SX                                                  MDP02130
      YAB1=YA1-YB1                                                       MDP02140
      IF (I9.EQ.1) PEI=ERFC((YAB1-FPI1)*EBN)                             MDP02150
      IF (I1.EQ.LM1) PEI=ERFC((FMI1-YAB1)*EBN)                           MDP02160
      IF ((IA.LT.LM1).AND.(I1.GT.M))                                     MDP02170
     *PEI=ERFC((FMI1-YAB1)*EBN)-ERFC((FPI1-YAB1)*EBN)                    MDP02180
      IF ((IA.LT.LM1).AND.(I1.LT.M))                                     MDP02190
     *PEI=ERFC((YAB1-FPI1)*EBN)-ERFC((YAB1-FMI1)*EBN)                    MDP02200
```

```
FILE: MDFL1    FORTRAN  A    UNIV D'/OF OTTAWA CMS RELEASE 4
                                                                      MDF02210
   26 PEI1=PEI1+PEI                                                   MDF02220
      PENI=PENI+PEI1                                                  MDF02230
  '23 PEJ1=PEJ1+N*PEI1                                                MDF02240
      G2=G2+I1*PEJ1                                                   MDF02250
   24 CONTINUE                                                        MDF02260
C-----------------------------------------------------------------    MDF02270
      G2=G2*P3/2.                                                     MDF02280
      G1=1.-G1*P3                                                     MDF02290
      AN1=1.-AN1*P3                                                   MDF02300
      PEA=-ALOG10(PEB*P2)                                             MDF02310
      GFI=G1*SI+G2*CI                                                 MDF02320
      GFIN=2.*GFI/AN1                                                 MDF02330
      F=GFIN                                                          MDF02340
      RETURN                                                          MDF02350
      END
```

```
FILE: MQAM     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

C     KUCAR,1983 10 19,1985 04 19                                                MQA00010
C     ***************************************************************************MQA00020
C     PROGRAM TO EVALUATE THE  PERFORMANCE OF MQAM MODEM IN                      MQA00030
C     A NONLINEARLY AMPLIFIED MULTICHANNEL ENVIRONMENT.                          MQA00040
C  4  ***************************************************************           MQA00050
C                                                                                MQA00060
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NT(4,4096)                 MQA00070
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMBK(4,30)                 MQA00080
      COMPLEX DATA(32768),TF(32768),DATA1(32768),TA(514)                         MQA00090
      DOUBLE PRECISION PID,DSEED                                                 MQA00100
      DIMENSION XB(65536),PSI(257),WWK(9),IWK(15),WK(256)                        MQA00110
      DIMENSION PEI(50),BER(50)                                                  MQA00120
      DIMENSION NAME(10)                                                         MQA00130
      CALL ERRSET(208,700,-1,1,1,208)                                            MQA00140
      IJK=0                                                                      MQA00150
      JKL=0                                                                      MQA00160
      PI=3.14159265358979 3                                                      MQA00170
      PI2=PI*2.                                                                  MQA00180
      P2=PI/2.                                                                   MQA00190
      PID=3.14159265358979 3D0                                                   MQA00200
      DSEED=4569286.D0                                                           MQA00210
      IPROM=38                                                                   MQA00220
      ITO=50                                                                     MQA00230
      NR=1                                                                       MQA00240
      LSAMPL=64                                                                  MQA00250
      NSYMB=512                                                                  MQA00260
      MI=LSAMPL/2                                                                MQA00270
      MQ=MI                                                                      MQA00280
      IOFF=0                                                                     MQA00290
      SYM=FLOAT(NSYMB)                                                           MQA00300
      RUNS=FLOAT(NR)                                                             MQA00310
      CNEB=1.                                                                    MQA00320
      WRITE(6,101)                                                               MQA00330
  101 FORMAT('Baud rate:MBd')                                                    MQA00340
      READ(7,222)BAUD                                                            MQA00350
      SAM=FLOAT(LSAMPL)                                                          MQA00360
      BW=SAM*BAUD                                                                MQA00370
      LD=LSAMPL*NSYMB                                                            MQA00380
      LD2=2*LD                                                                   MQA00390
      L1=NSYMB                                                                   MQA00400
      L2=(L1/2)+1                                                                MQA00410
C     L(L)3=L2G(L1)...2**LL3=L1 IN P1,BUT 2**L3=LD IN FILTER                     MQA00420
      L3=15                                                                      MQA00430
      LL3=9                                                                      MQA00440
      L4=L1/2                                                                    MQA00450
      L5=L1+2                                                                    MQA00460
      NO1=LD/2+1                                                                 MQA00470
      NO2=NO1+1                                                                  MQA00480
      NUMB(2,4)=0                                                                MQA00490
  100 WRITE(6,3)                                                                 MQA00500
    3 FORMAT('number of channels')                                               MQA00510
      READ(7,102)NCH                                                             MQA00520
      WRITE(6,8)                                                                 MQA00530
    8 FORMAT('number of levels L,L=01 means shaping')                            MQA00540
      READ(7,12)LARY                                                             MQA00550
```

```
      IF (LARY.EQ.1) CALL INP1(NCH,1)                                          MQA00560
    2 CALL INP2(NCH)                                                           MQA00570
      WRITE(8,4)                                                               MQA00580
    4 FORMAT(' pen type:1 TO 9')                                               MQA00590
      READ(7,102)IPEN                                                          MQA00600
      LPEN=IPEN+4                                                              MQA00610
C     WRITE(6,6) IPEN,IOFF,NUMB(1,20),LARY                                     MQA00620
      WRITE(8,6) IPEN,IOFF,NUMB(1,20),LARY                                     MQA00630
    6 FORMAT(/,'----------------PEN,OFFSET,NONL,LARY',4I5)                     MQA00640
      DO 5 I=IFROM,ITO                                                         MQA00650
    5 BER(I)=0.                                                                MQA00660
      ANORM=SQRT(1.)                                                           MQA00670
      DO 99 J=1,NR                                                             MQA00680
      CALL ENCODE(LD,DATA,1,LARY)                                              MQA00690
C     CALL P1(LD,LD2,L1,L2,LL3,L4,L5,DATA,XA,XB,PSY,NWK,WK,3,IPEN,1)           MQA00700
      IF (J.NE.1) GOTO 7                                                       MQA00710
C     CALL PLTSAS(LD,DATA,60,MI,MQ,JKL,IJK,LARY,2)                             MQA00720
C     CALL PLTSAS(LD,DATA,250,MI,MQ,JKL,IJK,LARY,1)                            MQA00730
    7 CALL NONLIN(LD,DATA,ANORM,1)                                             MQA00740
C     IF (IPEN.GT.2)                                                           MQA00750
C    *CALL P1(LD,LD2,L1,L2,LL3,L4,L5,DATA,XA,XB,PSY,NWK,WK,3,IPEN,1)           MQA00760
      CALL FILTER(LD,L3,DATA,TF,IWK,1,1)                                       MQA00770
      IF (NUMB(1,4).GT.1) CALL FILTER(LD,L3,DATA,TF,IWK,1,2)                   MQA00780
      IF (NUMB(1,5).EQ.4) CALL FILTER(LD,L3,DATA,TF,IWK,1,-1)                  MQA00790
      CALL NOISE1(LD,TF,PNOISE)                                                MQA00800
C     CALL CHANN(LD,DATA,SNR,1)                                                MQA00810
      IF (NCH.LT.2) GOTO 66                                                    MQA00820
      DO 10 K=2,NCH                                                            MQA00830
      CALL ENCODE(LD,DATA1,K,LARY)                                             MQA00840
      CALL NONLIN(LD,DATA1,ANORM,K)                                            MQA00850
      PSH=PI2*GGUBFS(DSEED)                                                    MQA00860
      CALL PHASSH(LD,DATA1,PSH)                                                MQA00870
      ITSH=INT(SAM*(GGUBFS(DSEED)-.5))                                         MQA00880
      CALL TIMESH(LD,DATA1,TF,ITSH)                                            MQA00890
      CALL ATT(LD,DATA1,K)                                                     MQA00900
C     CALL FILTER(LD,L3,DATA1,TF,IWK,K,N)                                      MQA00910
      DO 10 I=1,LD                                                             MQA00920
   10 DATA(I)=DATA(I)+DATA1(I)                                                 MQA00930
      CALL POWER(LD,DATA,PMEA,PPEA,EB,0)                                       MQA00940
   66 CALL POWER(LD,DATA,PAVP,PEAK,EB,1)                                       MQA00950
      CALL SYNCH(LD,DATA,TF,PNOISE,MI,MQ,EB,LARY)                              MQA00960
C     CALL PLTSAS(LD,DATA,60,MI,MQ,JKL,IJK,LARY,2)                             MQA00970
      CALL PLTSAS(LD,DATA,250,MI,MQ,JKL,IJK,LARY,1)                            MQA00980
      CALL DECODE(LD,DATA,PNOISE,PEI,IFROM,ITO,EB,MI,MQ,LARY,CNEB,NER)         MQA00990
      IF (NER.GT.0) GOTO 123                                                   MQA01000
      DO 99 I=IFROM,ITO                                                        MQA01010
      PEI(I)=PEI(I)/SYM                                                        MQA01020
   99 BER(I)=BER(I)+PEI(I)                                                     MQA01030
C     WRITE(6,1)                                                               MQA01040
      IF (CNEB.EQ.0.) WRITE(8,1)                                               MQA01050
    1 FORMAT('----------Eb/No----------Prob.of symbol error----------')       MQA01060
      IF (CNEB.EQ.1.) WRITE(8,111)                                             MQA01070
  111 FORMAT('----------C/N----------Prob.of symbol error----------')         MQA01080
      DO 9 I=IFROM,ITO                                                         MQA01090
      PSH=FLOAT(I)                                                             MQA01100
```

```
FILE: MQAM     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 8

      PEI(I)=BER(I)/PUNS                                                    MQA01110
    9 WRITE(8,172)PSH,PEI(I)                                                MQA01120
  172 FORMAT(2E15.6,I7)                                                     MQA01130
C     CALL BERPHP(IPEN,PEI,IFROM,ITO,1.E-12)                                MQA01140
  123 WRITE(8,173)                                                          MQA01150
  173 FORMAT('==========================================================',/MQA01160
     #'NEW PARAM>0,NEW BB SIGNAL<0...')                                     MQA01170
      READ(7,222)GO                                                         MQA01180
      IF (GO) 100,11,2                                                      MQA01190
   11 CONTINUE                                                              MQA01200
   12 FORMAT(I2)                                                            MQA01210
  102 FORMAT(I1)                                                            MQA01220
  222 FORMAT(F15.7)                                                         MQA01230
      END                                                                   MQA01240
```

```
FILE: NOISE1   FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 8
                                                                                NOI00010
C     A.KUCAR,1984 01 04,1985 04 18                                             NOI00020
C     ***************************************************************           NOI00030
C     Subroutine COMPUTES the effective noise (PNOISE)                          NOI00040
C     PNOISE=No*Integral(|TF(f)|**2*df)                                         NOI00050
C     No=0.5          (White Gaussian Noise)                                    NOI00060
C     ***************************************************************           NOI00070
C                                                                               NOI00080
      SUBROUTINE NOISE1(LD,TF,PNOISE)                                           NOI00090
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),MI(4,4096)                NOI00100
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TMPR(4,30)                 NOI00110
      COMPLEX TF(LD)                                                            NOI00120
      SUM=0.0                                                                   NOI00130
      DO 1 L=1,LD                                                               NOI00140
    1 SUM=SUM+(CABS(TF(L)))**2.                                                 NOI00150
      PNOISE=SUM*BW/FLOAT(LD)/2.                                                NOI00160
      RETURN                                                                    NOI00170
      END
```

```
FILE: NONLIN   FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

C      A.KUCAR,1984 01 04,1985 04 18                                          NON00010
CCC    ***************************************************************       NON00020
CCC    Subroutine simulates an ideal HARDLIMITER:Out=||Inp||                  NON00030
CCC    ***************************************************************       NON00040
C                                                                             NON00050
       SUBROUTINE NONLIN(LD,DATA,ANORM,N)                                     NON00060
       COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             NON00070
       COMPLEX DATA(LD)                                                       NON00080
       IF (NUMB(N,20).EQ.0) RETURN                                            NON00090
       DO 1 I=1,LD                                                            NON00100
     1 DATA(I)=ANORM*DATA(I)/CABS(DATA(I))                                    NON00110
       RETURN                                                                 NON00120
       END                                                                    NON00130
```

FILE: PHASSH · FORTRAN A UNIV D*/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1984 01 04,12985 04 18                                        PHA00010
CCCCC ************************************************************************PHA00020
      Subroutine COMPENSATES the PHASE SHIFT due to the carrier offset       PHA00030
      ************************************************************************PHA00040
C                                                                            PHA00050
      SUBROUTINE PHASSH(LD,DATA,PSHIFT)                                      PHA00060
      COMPLEX DATA(LD),EPS                                                   PHA00070
      EPS=CMPLX(COS(PSHIFT),-SIN(PSHIFT))                                    PHA00080
      DO 1 I=1,LD                                                            PHA00090
    1 DATA(I)=DATA(I)*EPS                                                    PHA00100
      RETURN                                                                 PHA00110
      END                                                                    PHA00120
```

FILE: PLTSAS FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1984 01 20,1985 04 18                                          PLT00010
CCC   ********************************************************************PLT00020
CCC   Subroutine DRAWS:IWHAT=1...EYE-PATTERN for 2 symbol duration           PLT00030
CCC                         2...POLAR (state space) DIAGRAM                  PLT00040
CCC   ********************************************************************PLT00050
C                                                                            PLT00060
      SUBROUTINE PLTSAS(LD,DATA,M2,MI,MQ,IKL,IJK,LARY,IWHAT)                 PLT00070
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NT(4,4096)             PLT00080
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),ATMEL(4,4),ATMEP(4,30)             PLT00090
      COMPLEX DATA(LD)                                                       PLT00100
      ARY=FLOAT(LARY)-1.                                                     PLT00110
      IF (ARY.EQ.0.) ARY=1.                                                  PLT00120
      GOTO (1,2),IWHAT                                                       PLT00130
C     ------------------------------- E Y E ----------------------------     PLT00140
    1 JJ=2*LSAMPL+1                                                          PLT00150
      J1=JJ-1                                                                PLT00160
      DO 11 KK=1,M2                                                          PLT00170
      K1=KK-1                                                                PLT00180
      JKL=JKL+1                                                              PLT00190
      DO 11 I=1,JJ                                                           PLT00200
      I1=I-1                                                                 PLT00210
      II=K1*J1+I+MI                                                          PLT00220
      A2=REAL(DATA(II))/ARY                                                  PLT00230
      A1=(FLOAT(I1)-SAM)/SAM                                                 PLT00240
   11 WRITE(1,12)A1,A2,JKL                                                   PLT00250
   12 FORMAT(2F10.6,I8)                                                      PLT00260
      RETURN                                                                 PLT00270
C     ------------------------------- P O L A R ------------------------     PLT00280
    2 JJ=LSAMPL+1                                                            PLT00290
      IF (IJK.GT.0) JJ=1                                                     PLT00300
      DO 13 KK=1,M2                                                          PLT00310
      K=KK*LSAMPL                                                            PLT00320
      K1=K+MI-1                                                              PLT00330
      K2=K+MQ-1                                                              PLT00340
      IJK=IJK+1                                                              PLT00350
      DO 13 I=1,JJ                                                           PLT00360
      A1=REAL(DATA(K1+I))/ARY                                                PLT00370
      A2=AIMAG(DATA(K2+I))/ARY                                               PLT00380
   13 WRITE(9,12)A1,A2,IJK                                                   PLT00390
      RETURN                                                                 PLT00400
      END                                                                    PLT00410
```

FILE: POWER FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 8

```
C     A.KUCAR,1984 01 22,1985 04 18                                          POW00010
CCCCC ***************************************************************      POW00020
      Subroutine COMPUTES the MEAN and PEAK POWER and EB                     POW00030
      ***************************************************************      POW00040
C                                                                            POW00050
      SUBROUTINE POWER(LD,DATA,PAVE,PEAK,EB,N)                               POW00060
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             POW00070
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMES(4,30)             POW00080
      COMPLEX DATA(LD)                                                       POW00090
      PEAK=0.                                                                POW00100
      PAVE=0.                                                                POW00110
      DO 1 I=1,LD                                                            POW00120
      A=CABS(DATA(I))                                                        POW00130
      PAVE=PAVE+A*A                                                          POW00140
      IF (PEAK.LT.A) PEAK=A                                                  POW00150
    1 CONTINUE                                                               POW00160
      PAVE=PAVE/FLOAT(LD)/2.                                                 POW00170
      PEAK=10.*ALOG10(PEAK*PEAK/PAVE)                                        POW00180
      IF (N.EQ.1) EB=PAVE/BAUD*2.                                            POW00190
      RETURN                                                                 POW00200
      END                                                                    POW00210
```

```
FILE: P1       FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 8

C     A.KUCAR,1984 02 02,1985 04 17                                          P1  00010
C     ***************************************************************** P1  00020
C     Subroutine COMPUTES,PRINTS AND PLOTS                                   P1  00030
C     1.)POWER SPECTRUM DENSITY AND OUT-OF-BAND ENERGY                       P1  00040
C     IWHAT=0 -->OUT-OF-BAND ENERGY                                          P1  00050
C          =1 -->SPECTRUM                                                    P1  00060
C          =2 -->BOTH                                                        P1  00070
C     2.)POWER SPECTRUM DENSITY OF Weighted Cosine signal                    P1  00080
C                                  Kaiser-Bessel signal                      P1  00090
C     IWHAT=3                                                                P1  00100
C     3.)N=-1 -->TRIANGULAR SIGNAL SHAPE                                     P1  00110
C     ---------------------------------------------------------------------- P1  00120
      LD2=2*LD                                                               P1  00130
      L1=LD/64                                                               P1  00140
      L2=(L1/2)+1                                                            P1  00150
      L3=L2G(L1)...2**L3=L1                                                  P1  00160
      L4=L1/2                                                                P1  00170
      L5=L1+2                                                                P1  00180
C     ***************************************************************** P1  00190
C                                                                            P1  00200
      SUBROUTINE P1(LD,LD2,L1,L2,L3,L4,L5,DATA,XA,XB,PSX,NWK,WK,IW,IP,N)P1  00210
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             P1  00220
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)             P1  00230
      COMPLEX DATA(LD),XA(L5)                                                P1  00240
      DIMENSION XB(LD2),PSX(L2),NWK(L3),WK(L4)                               P1  00250
      IF (IW.EQ.3) GOTO 98                                                   P1  00260
      LL1=L2+1                                                               P1  00270
      LL2=L2-2                                                               P1  00280
      P=0.                                                                   P1  00290
      DO 1 I=1,LD                                                            P1  00300
      I1=I-IOFF                                                              P1  00310
      IF(I1.LE.0) I1=I1+LD                                                   P1  00320
      XB(I)=REAL(DATA(I))                                                    P1  00330
      XB(I+LD)=AIMAG(DATA(I1))                                               P1  00340
    1 P=P+XB(I)+XB(I+LD)                                                     P1  00350
      P=P/LD2                                                                P1  00360
      DO 9 I=1,LD2                                                           P1  00370
    9 XB(I)=XB(I)-P                                                          P1  00380
      WRITE(6,10)P                                                           P1  00390
   10 FORMAT('The MEAN of I+Q channel:',E17.5)                               P1  00400
      CALL PTPPS(XB,Y,LD2,L1,0,PSX,PSY,XPS,NWK,WK,XA,IRP)                    P1  00410
      P=PSX(1)                                                               P1  00420
      P1=PSX(L2)                                                             P1  00430
      DO 2 I=1,L2                                                            P1  00440
      Y=PSX(I)/P                                                             P1  00450
      IF (Y.LT.1.E-10) Y=1.E-10                                              P1  00460
      PSX(I)=10.*ALOG10(Y)                                                   P1  00470
    2 XB(I)=0.                                                               P1  00480
      XB(L2)=P1                                                              P1  00490
      DO 3 I=1,LL2                                                           P1  00500
    3 XB(L2-I)=XB(LL1-I)+2.*10.**(PSX(L2-I)*.1)                              P1  00510
      XB(1)=XB(2)+10.**(PSX(1)*.1)                                           P1  00520
      P1=XB(1)                                                               P1  00530
      DO 4 I=1,L2                                                            P1  00540
      Y=XB(I)/P1                                                             P1  00550
```

```
FILE: P1       FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

      IF (Y.LT.1.E-10) Y=1.E-10                                          P1 00560
    4 XB(I)=10.*ALOG10(Y)                                                P1 00570
C     WRITE(6,5)                                                         P1 00580
    5 FORMAT('Spectrum X      Out-of-band energy')                       P1 00590
C     WRITE(6,6)(PSX(I),XB(I),I=1,L2)                                    P1 00600
    6 FORMAT(2F16.7)                                                     P1 00610
      M=INT(FLOAT((L2-1)*10)/SAM)+1                                      P1 00620
      SS=SAM/FLOAT(NSYMB)                                                P1 00630
      DO 22 I=1,M                                                        P1 00640
      Y=(I-1)*SS                                                         P1 00650
   22 WRITE(3,23)Y,PSX(I),IP                                             P1 00660
   23 FORMAT(2E15.6,I1)                                                  P1 00670
      RETURN                                                             P1 00680
   98 TRIANG=0.                                                          P1 00690
      IF (NUMB(N,1).EQ.-1) TRIANG=1.                                     P1 00700
      TIMEL(N,1)=TIMEL(N,1)/(1.+TRIANG)                                  P1 00710
      NNN=IABS(NUMB(N,1))                                                P1 00720
      F=0.                                                               P1 00730
      CALL PSD(N,NNN,TRIANG,F,SLOG,SZ)                                   P1 00740
      J1=2001                                                            P1 00750
      IF (IP.GT.2) J1=1001                                               P1 00760
      DO 12 J=1,J1                                                       P1 00770
      F=FLOAT(J-1001)/200.                                               P1 00780
      CALL PSD(N,NNN,TRIANG,F,SLOG,SZ)                                   P1 00790
   12 WRITE(3,23)F,SLOG,IP                                               P1 00800
      RETURN                                                             P1 00810
      END                                                                P1 00820
C     ------------------------------------------------------------------ P1 00830
C                                                                        P1 00840
      SUBROUTINE PSD(N,NNN,TRIANG,F,SLOG,SZ)                             P1 00850
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)         P1 00860
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)         P1 00870
      IF (NNN.GT.0) GOTO 77                                              P1 00880
      W0=TIMEK(N,1)/SINH(TIMEK(N,1))                                     P1 00890
      PL=F*TIMEL(N,1)                                                    P1 00900
      X=PI*SQRT(ABS(AMPL(N,1)*AMPL(N,1)-PL*PL))                          P1 00910
      IF (AMPL(N,1).LE.PL) GOTO 66                                       P1 00920
      X=SINH(X)/X                                                        P1 00930
      GOTO 67                                                            P1 00940
   66 CALL SINC(X)                                                       P1 00950
   67 SLIN=W0*X                                                          P1 00960
      SLIN=SLIN*SLIN                                                     P1 00970
      IF (F.EQ.0.) SZ=SLIN                                               P1 00980
      SLIN=SLIN/SZ                                                       P1 00990
      GOTO 78                                                            P1 01000
   77 X=PI*TIMEL(N,1)*F                                                  P1 01010
      CALL SINC(X)                                                       P1 01020
      GM=AMPL(N,1)*TIMEL(N,1)*X                                          P1 01030
      GN=0.                                                              P1 01040
      IF (TRIANG.EQ.1.) GM=GM*GM                                         P1 01050
      IF (NNN.EQ.1) GO TO 17                                             P1 01060
      DO 16 K=2,NNN                                                      P1 01070
      DO 16 L=1,3,2                                                      P1 01080
      X=PI*TIMEL(N,K)*(F+(FLOAT(L)-2.)*TIMEK(N,K)/TIMEL(N,K)/2.)         P1 01090
      CALL SINC(X)                                                       P1 01100
```

```
FILE: P1       FORTRAN   A    UNIV D'/OF OTTAWA CMS RELEASE 8

      GN=GN+TIMEL(N,K)*AMPL(N,K)*X                                      P1 01110
   16 CONTINUE                                                          P1 01120
   17 G=GM+GN/2.                                                        P1 01130
      IF (F.EQ.0.) SZ=2.*G*G                                            P1 01140
      SLIN=2.*G*G/SZ                                                    P1 01150
   78 IF (SLIN.LT.1E-10) SLIN=1.0E-10                                   P1 01160
      SLOG=10.*ALOG10(SLIN)                                             P1 01170
      RETURN                                                            P1 01180
      END                                                               P1 01190
```

```
FILE: SI3PHP   FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

C     A.KUCAR,1984 01 29,1985 04 18                                          SIG00010
C     ***************************************************************      SIG00020
C     Subroutine PLOTS (JOE=0) Y=SUM(A*cos(PI*t*K/L)) signal                 SIG00030
C                                 KBY=Io(w*SQRT(L*L-t*t))/To(wL)             SIG00040
C                              (JOE=1) I and Q ch.20 symbols                 SIG00050
C     ***************************************************************      SIG00060
C                                                                            SIG00070
      SUBROUTINE SIGPHP(LD,DATA,JOE,IPEN,N)                                  SIG00080
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             SIG00090
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)             SIG00100
      COMPLEX DATA(LD)                                                       SIG00110
      REAL MMBSIO                                                            SIG00120
      IF (JOE.EQ.1) GO TO 4                                                  SIG00130
      A0=AMPL(N,1)                                                           SIG00140
      DO 3 K=1,N                                                             SIG00150
      KNN=NUMB(K,1)                                                          SIG00160
      KPEN=IPEN+K-1                                                          SIG00170
      KN=KNN                                                                 SIG00180
      IF (KN.EQ.0) KN=1                                                      SIG00190
      A2=0.                                                                  SIG00200
      DO 3 J=1,KN                                                            SIG00210
      DO 3 L=1,1081                                                          SIG00220
      I=L-541                                                                SIG00230
      T=FLOAT(I)*TIMEL(K,J)/1080.                                            SIG00240
      IF (KNN.GT.0) GOTO 11                                                  SIG00250
      A1=AMPL(K,1)*SQRT(1.-4*T*T/TIMEL(K,1)/TIMPL(K,1))                      SIG00260
      A2=MMBSIO(1,A1,IER)/MMBSIO(1,A0,IER)                                   SIG00270
      GOTO 3                                                                 SIG00280
   11 A2=A2+AMPL(K,J)*COS(PI*T*TIMEK(K,J)/TIMEL(K,J))                        SIG00290
    3 WRITE(10,8)T,A2,KPEN                                                   SIG00300
    8 FORMAT(2F10.4,I2)                                                      SIG00310
      RETURN                                                                 SIG00320
    4 K=LSAMPL*10+1                                                          SIG00330
      DO 5 I=1,K                                                             SIG00340
      A1=AIMAG(DATA(I))                                                      SIG00350
      A2=REAL(DATA(I))                                                       SIG00360
      A4=1.5+ATAN2(A1,A2)*6./PI                                              SIG00370
      A1=FLOAT(I-1)/SAM                                                      SIG00380
      A2=7.5+REAL(DATA(I))                                                   SIG00390
      A3=4.0+CABS(DATA(I))                                                   SIG00400
    5 WRITE(10,6)A1,A2,A3,A4,IPEN                                            SIG00410
    6 FORMAT(4F10.4,I2)                                                      SIG00420
      RETURN                                                                 SIG00430
      END                                                                    SIG00440
```

```
FILE: SINC     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

C     A.KUCAR,1983 07 31                                                    SIN00010
CCCCC ********************************************************************SIN00020
      Subroutine CALCULATES sin(x)/x                                        SIN00030
      Singular point x=0 is included,i.e. sin(0)/0=1.                       SIN00040
      ********************************************************************SIN00050
C                                                                           SIN00060
      SUBROUTINE SINC(X)                                                    SIN00070
      IF (X.EQ.0.0) GO TO 1                                                 SIN00080
      X=SIN(X)/X                                                            SIN00090
      RETURN                                                                SIN00100
    1 X=1.                                                                  SIN00110
      RETURN                                                                SIN00120
      END                                                                   SIN00130
```

```
FILE: SYNCH     FORTRAN  A    UNIV D*/OF OTTAWA CMS RELEASE 8

C      A.KUCAR,1984 10 21,1985 04 18                                          SYN00010
CCCC   ***************************************************************       SYN00020
       Subroutine SYNCHRONIZES the M-ary QAM received data                    SYN00030
       ***************************************************************       SYN00040
C                                                                             SYN00050
       SUBROUTINE SYNCH(LD,DATA,TF,PNOISE,MI,MQ,ER,LARY)                      SYN00060
       COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             SYN00070
       COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMPL(4,4),TIMPR(4,30)             SYN00080
       COMPLEX DATA(LD),TF(LD)                                                SYN00090
       IF (IOFF.EQ.0) GOTO 111                                                SYN00100
       DO 13 I=1,IOFF                                                         SYN00110
       A=AIMAG(DATA(LD))                                                      SYN00120
       DO 14 J=2,LD                                                           SYN00130
       J1=LD-J+2                                                              SYN00140
    14 DATA(J1)=CMPLX(REAL(DATA(J1)),AIMAG(DATA(J1-1)))                       SYN00150
    13 DATA(1)=CMPLX(REAL(DATA(1)),A)                                         SYN00160
   111 NOLD=0                                                                 SYN00170
       NOF=0                                                                  SYN00180
       NS2=2*NSYMB                                                            SYN00190
       K=1                                                                    SYN00200
     3 NEW=0                                                                  SYN00210
       DO 4 J=1,NSYMB                                                         SYN00220
       J1=K+(J-1)*LSAMPL                                                      SYN00230
       IF(J1.GT.LD) J1=J1-LD                                                  SYN00240
       XB=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.                                SYN00250
       YB=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.                              SYN00260
       ABC=NI(3,J)*XB                                                         SYN00270
       IF (ABC.GT.0.) NEW=NEW+1                                               SYN00280
       ABC=NI(4,J)*YB                                                         SYN00290
       IF (ABC.GT.0.) NEW=NEW+1                                               SYN00300
     4 CONTINUE                                                               SYN00310
       IF(NOLD.GE.NEW) GOTO 5                                                 SYN00320
       NOLD=NEW                                                               SYN00330
       NOF=K                                                                  SYN00340
     5 K=K+1                                                                  SYN00350
       IF(NOLD.LT.NS2.AND.K.LE.LD) GOTO 3                                     SYN00360
       NOF=NOF-1                                                              SYN00370
       CALL TIMESH(LD,DATA,TF,NOF)                                            SYN00380
       EOI=FLOAT(NSYMB)+1.                                                    SYN00390
       EOQ=EOI                                                                SYN00400
       MI=1                                                                   SYN00410
       MQ=1                                                                   SYN00420
       SNR=SQRT(PNOISE*EB/10.)                                                SYN00430
C      ---------------------------------------------------------------        SYN00440
       IF (LARY.GT.1) GOTO 2                                                  SYN00450
       DO 7 J=1,LSAMPL                                                        SYN00460
       EI=0.                                                                  SYN00470
       EQ=0.                                                                  SYN00480
       DO 6 K=1,NSYMB                                                         SYN00490
       J1=(K-1)*LSAMPL+J                                                      SYN00500
       XB=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.                                SYN00510
       YB=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.                              SYN00520
       IF(SIGN(1.,XB).NE.NI(3,K)) EI=EI+1.                                    SYN00530
       IF(SIGN(1.,YB).NE.NI(4,K)) EQ=EQ+1.                                    SYN00540
       XB=ABS(XB)/SNR                                                         SYN00550
```

```
      IF (XB.GT.12.) XB=12.                                              SYN00560
      EI=EI+ERFC(XB)/2.                                                  SYN00570
      XB=ABS(YB)/SNR                                                     SYN00580
      IF (XB.GT.12.) XB=12.                                              SYN00590
      EQ=EQ+ERFC(XB)/2.                                                  SYN00600
    6 CONTINUE                                                           SYN00610
      IF(EOI.LE.EI) GO TO 8                                              SYN00620
      EOI=EI                                                             SYN00630
      MI=J                                                               SYN00640
    8 IF (EOQ.LE.EQ) GO TO 7                                             SYN00650
      EOQ=EQ                                                             SYN00660
      MQ=J                                                               SYN00670
    7 CONTINUE                                                           SYN00680
      GOTO 1                                                             SYN00690
C     ------------------------------------------------------------------ SYN00700
    2 HIGH=FLOAT(LARY-1)                                                 SYN00710
      DO 17 J=1,LSAMPL                                                   SYN00720
      EI=0.                                                              SYN00730
      EQ=0.                                                              SYN00740
      DO 16 K=1,NSYMB                                                    SYN00750
      J1=(K-1)*LSAMPL+J                                                  SYN00760
      XB=(REAL(DATA(J1))+REAL(DATA(J1+1)))/2.                            SYN00770
      YB=(AIMAG(DATA(J1))+AIMAG(DATA(J1+1)))/2.                          SYN00780
      DX1=ABS(XB)                                                        SYN00790
      DY1=ABS(YB)                                                        SYN00800
      INDI=0                                                             SYN00810
      AI=FLOAT(NI(3,K))                                                  SYN00820
      IF((XB*AI).LE.0.) INDI=1                                           SYN00830
      AI=ABS(AI)                                                         SYN00840
      THRI1=AI-1.                                                        SYN00850
      THRI2=AI+1.                                                        SYN00860
      LAGI=0                                                             SYN00870
      IF (AI.EQ.HIGH) GOTO 30                                            SYN00880
      IF (DX1.GE.THRI2.OR.DX1.LE.THRI1) INDI=1                           SYN00890
      GOTO 40                                                            SYN00900
   30 IF (DX1.LE.THRI1) INDI=1                                           SYN00910
      LAGI=1                                                             SYN00920
   40 INDQ=0                                                             SYN00930
      AI=FLOAT(NI(4,K))                                                  SYN00940
      IF((YB*AI).LE.0.) INDQ=1                                           SYN00950
      AI=ABS(AI)                                                         SYN00960
      THRQ1=AI-1.                                                        SYN00970
      THRQ2=AI+1.                                                        SYN00980
      LAGQ=0                                                             SYN00990
      IF (AI.EQ.HIGH) GOTO 50                                            SYN01000
      IF (DY1.GE.THRQ2.OR.DY1.LE.THRQ1) INDQ=1                           SYN01010
      GOTO 60                                                            SYN01020
   50 IF (DY1.LE.THRQ1) INDQ=1                                           SYN01030
      LAGQ=1                                                             SYN01040
   60 DX2=ABS(DX1-THRI1)/SNR                                             SYN01050
      DY2=ABS(DY1-THRQ1)/SNR                                             SYN01060
      IF (DX2.GT.12.) DX2=12.                                            SYN01070
      EI=EI+ERFC(DX2)/2.                                                 SYN01080
      IF (LAGI.EQ.1) GOTO 10                                             SYN01090
      DX2=ABS(DX1-THRI2)/SNR                                             SYN01100
```

FILE: SYNCH FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
      IF (DX2.GT.12.) DX2=12.                                          SYN01110
      EI=EI+ERFC(DY2)/2.                                               SYN01120
   10 IF(INDI.EQ.1) EI=EI+1.                                           SYN01130
      IF (DY2.GT.12.) DY2=12.                                          SYN01140
      EQ=EQ+ERFC(DY2)/2.                                               SYN01150
      IF (LASQ.EQ.1) GOTO 12                                           SYN01160
      DY2=ABS(DY1-THRQ2)/SNR                                           SYN01170
      IF (DY2.GT.12.) DY2=12.                                          SYN01180
      EQ=EQ+ERFC(DY2)/2.                                               SYN01190
   12 IF (INDQ.EQ.1) EQ=EQ+1.                                          SYN01200
   16 CONTINUE                                                         SYN01210
      IF(EOI.LE.EI) GO TO 18                                           SYN01220
      EOI=EI                                                           SYN01230
      MI=J                                                             SYN01240
   18 IF (EOQ.LE.EQ) GO TO 17                                          SYN01250
      EOQ=EQ                                                           SYN01260
      MQ=J                                                             SYN01270
   17 CONTINUE                                                         SYN01280
    1 EI=FLOAT(MI)/SAM                                                 SYN01290
      EQ=FLOAT(MQ)/SAM                                                 SYN01300
      WRITE(6,9)EI,EQ                                                  SYN01310
      WRITE(8,9)EI,EQ                                                  SYN01320
    9 FORMAT('Sampling points for I and Q channels',2F8.2)             SYN01330
      RETURN                                                           SYN01340
      END                                                              SYN01350
```

FILE: TIMESH FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1984 01 04,1985 04 18                                          TIM00010
CCCCC ***************************************************************      TIM00020
      Subroutine SHIFTS the signal in TIME DOMAIN                            TIM00030
      ***************************************************************      TIM00040
C                                                                            TIM00050
      SUBROUTINE TIMESH(LD,DATA,TF,IFOF)                                     TIM00060
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             TIM00070
      COMPLEX DATA(LD),TF(LD)                                                TIM00080
      IF (IFOF.EQ.0) RETURN                                                  TIM00090
      J=LD-IABS(IFOF)                                                        TIM00100
      K=J+1                                                                  TIM00110
      IF (IFOF.LT.0) GOTO 4                                                  TIM00120
      DO 1 I=1,J                                                             TIM00130
    1 TF(I)=DATA(I)                                                          TIM00140
      DO 2 I=K,LD                                                            TIM00150
    2 DATA(I-J)=DATA(I)                                                      TIM00160
      DO 3 I=1,J                                                             TIM00170
    3 DATA(I+IFOF)=TF(I)                                                     TIM00180
      RETURN                                                                 TIM00190
    4 L=1-IFOF                                                               TIM00200
      DO 5 I=L,LD                                                            TIM00210
    5 TF(I+IFOF)=DATA(I)                                                     TIM00220
      DO 6 I=1,IFOF                                                          TIM00230
    6 DATA(I+J)=DATA(I)                                                      TIM00240
      DO 7 I=1,J                                                             TIM00250
    7 DATA(I)=TF(I)                                                          TIM00260
      RETURN                                                                 TIM00270
      END                                                                    TIM00280
```

FILE: TREQ FORTRAN * UNIV D'/OF OTTAWA CMS RELEASE 4

```
C     A.KUCAR,1986 11 10,1987 01 04                                          TRE00010
CCCCC ************************************************************          TRE00020
CCCCC Subroutine TREQ models TRANSVERSAL EQUALIZER with an odd number        TRE00030
CCCCC of taps NTAP=2*N+1                                                     TRE00040
CCCCC NEQSAM is the number of samples of each of delay elements,e.g.         TRE00050
CCCCC if NEQSAM=LSAMPL a full tap equalizer is assumed,                      TRE00060
CCCCC if NEQSAM=LSAMPL/2 an 1/2 fractional equalizer is assumed,etc.         TRE00070
CCCCC LD is the total number of samples                                      TRE00080
CCCCC DATA is the complex data matrix                                        TRE00090
CCCCC C is the tap coefficients matrix                                       TRE00100
CCCCC TF is the dummy matrix                                                 TRE00110
CCCCC LS2=LSAMPL/2                                                           TRE00120
CCCCC JOE=0 tap coefficients defined previously                              TRE00130
CCCCC    =1 NRZ signal ISI due to timing uncertainty                         TRE00140
CCCCC    =2 NRZ signal ISI compensation                                      TRE00150
CCCCC ************************************************************          TRE00160
C                                                                            TRE00170
      SUBROUTINE TREQ(NTAP,NEQSAM,LD,DATA,TF,C,LS2,JOE)                      TRE00180
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)             TRE00190
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TAPE(4,30)              TRE00200
      COMPLEX DATA(LD),TF(LD),D1(1)                                          TRE00210
      DIMENSION C(NTAP)                                                      TRE00220
      NTA=(NTAP-1)/2                                                         TRE00230
      NTAP0=NTA+1                                                            TRE00240
      DO 2 I=1,NSYMB                                                         TRE00250
      I1=(I-1)*LSAMPL                                                        TRE00260
      DO 2 J=1,LSAMPL                                                        TRE00270
      K=J+I1                                                                 TRE00280
      IF (JOE.EQ.0) GOTO 4                                                   TRE00290
C     Define tap coefficients                                                TRE00300
      FL=ABS(FLOAT(J-1)/SAM-0.5)                                             TRE00310
      C(2)=1.-FL                                                             TRE00320
      C(1)=0.                                                                TRE00330
      C(3)=0.                                                                TRE00340
      IF (J.LE.LS2) C(1)=FL                                                  TRE00350
      IF (J.GE.LS2) C(3)=FL                                                  TRE00360
      IF (JOE.EQ.1) GOTO 4                                                   TRE00370
      C(2)=1./C(2)                                                           TRE00380
      C(1)=-C(1)                                                             TRE00390
      C(3)=-C(3)                                                             TRE00400
    4 D1(1)=CMPLX(0.,0.)                                                     TRE00410
      DO 1 L=1,NTA                                                           TRE00420
      L1=L*NEQSAM                                                            TRE00430
      KM=K-L1                                                                TRE00440
      KP=K+L1                                                                TRE00450
      IF (KM.LT.1) KM=LD+KM                                                  TRE00460
      IF (KP.GT.LD) KP=KP-LD                                                 TRE00470
    1 D1(1)=D1(1)+C(NTAP0-L)*DATA(KM)+C(NTAP0+L)*DATA(KP)                    TRE00480
    2 TF(K)=D1(1)+C(NTAP0)*DATA(K)                                           TRE00490
      DO 3 I=1,LD                                                            TRE00500
    3 DATA(I)=TF(I)                                                          TRE00510
      RETURN                                                                 TRE00520
      END                                                                    TRE00530
```

FILE: WINDOW FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE A

```
C     A.KUCAR,1983 10 28,1985 04 18                                    WIN00010
CCCCC ***************************************************************WIN00020
CCCCC Subroutine calculates WINDOW (WEIGHT) functions                  WIN00030
CCCCC Sum<Ai*cos(PI*Ki/Li*t)>, or KAISER-BESSEL window                 WIN00040
CCCCC Max overlapping L=4Ts                                            WIN00050
CCCCC ***************************************************************WIN00060
C                                                                      WIN00070
      SUBROUTINE WINDOW(LD,M1,L1,M2,L2,IP,R,N,LOOP)                    WIN00080
      COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)       WIN00090
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMES(4,30)       WIN00100
      REAL R(512)                                                      WIN00110
      IF (LOOP.GT.1) GOTO 4                                            WIN00120
      L8=LSAMPL/2                                                      WIN00130
      L9=LSAMPL*2                                                      WIN00140
      KNN=NUMB(N,1)                                                    WIN00150
      KN=KNN                                                           WIN00160
      IF (KNN.EQ.0) KN=1                                               WIN00170
      TL=TIMEL(1,1)                                                    WIN00180
    4 DO 3 I=1,LSAMPL                                                  WIN00190
    3 R(I)=0.                                                          WIN00200
      DO 6 J=1,KN                                                      WIN00210
      IF (TIMEL(N,J).GT.TL) TL=TIMEL(N,J)                              WIN00220
C     ---------------------------------------------------------------- WIN00230
      DO 6 I=1,LSAMPL                                                  WIN00240
      U=-.5+(.5+FLOAT(I-1))/SAM                                        WIN00250
      CALL WIND1(J,U,Y,N,KNN,IP)                                       WIN00260
    6 R(I)=R(I)+Y                                                      WIN00270
C     ---------------------------------------------------------------- WIN00280
      IF (TL.LE.1.) RETURN                                             WIN00290
      DO 1 J=1,KN                                                      WIN00300
      ITL=(INT(TIMEL(N,J)*SAM)-LSAMPL)/2                               WIN00310
      IF (ITL.EQ.0) ITL=1                                              WIN00320
      IF (ITL.GT.L8) ITL=L8                                            WIN00330
      DO 1 I=1,ITL                                                     WIN00340
      I1=LSAMPL-I+1                                                    WIN00350
      U=.5+(.5+FLOAT(I-1))/SAM                                         WIN00360
      CALL WIND1(J,U,Y,N,KNN,M1)                                       WIN00370
      R(I)=R(I)+Y                                                      WIN00380
      CALL WIND1(J,U,Y,N,KNN,L1)                                       WIN00390
    1 R(I1)=R(I1)+Y                                                    WIN00400
C     ---------------------------------------------------------------- WIN00410
      IF (TL.LE.2.) RETURN                                             WIN00420
      DO 2 J=1,KN                                                      WIN00430
      ITL=(INT(TIMEL(N,J)*SAM)-L9)/2                                   WIN00440
      IF (ITL.EQ.0) ITL=1                                              WIN00450
      IF (ITL.GT.L8) ITL=L8                                            WIN00460
      DO 2 I=1,ITL                                                     WIN00470
      U=.5+(.5+FLOAT(LSAMPL-I))/SAM                                    WIN00480
      CALL WIND1(J,U,Y,N,KNN,L1)                                       WIN00490
      R(I)=R(I)+Y                                                      WIN00500
      U=1.+(.5+FLOAT(I-1))/SAM                                         WIN00510
      CALL WIND1(J,U,Y,N,KNN,M1)                                       WIN00520
    2 R(L8+I)=R(L8+I)+Y                                                WIN00530
C     ---------------------------------------------------------------- WIN00540
      IF (TL.LE.3.) RETURN                                             WIN00550
```

FILE: WINDOW FORTRAN A UNIV D'/OF OTTAWA CMS RELEASE 6

```
      DO 7 J=1,KN                                                          WIN00560
      ITL=(INT(TIMEL(N,J)*SAM)-L8-LSAMPL)/2                                WIN00570
      IF (ITL.EQ.0) ITL=1                                                  WIN00580
      DO 7 I=1,ITL                                                         WIN00590
      I1=LSAMPL-I+1                                                        WIN00600
      U=1.5+(.5+FLOAT(I-1))/SAM                                            WIN00610
      CALL WIND1(J,U,Y,N,KNN,M2)                                           WIN00620
      R(I)=R(I)+Y                                                          WIN00630
      CALL WIND1(J,U,Y,N,KNN,L2)                                           WIN00640
    7 R(I1)=R(I1)+Y                                                        WIN00650
      RETURN                                                               WIN00660
      END                                                                  WIN00670
C                                                                          WIN00680
CCCC  ********************************************************************WIN00690
CCCC  KAISER-BESSEL and WEIGHTED COSINE windows--->Subr.WINDOW             WIN00700
CCCC  ********************************************************************WIN00710
C                                                                          WIN00720
      SUBROUTINE WIND1(J,T,Y,N,KNN,JIP)                                    WIN00730
      COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)           WIN00740
      REAL MMBSIO                                                          WIN00750
      IF (KNN.GT.0) GOTO 1                                                 WIN00760
      Y=AMPL(N,1)                                                          WIN00770
      A1=AMPL(N,1)*SQRT(1.-4.*T*T/TIMEL(N,1)/TIMEL(N,1))                   WIN00780
      Y=MMBSIO(1,A1,IER)/MMBSIO(1,Y,IER)                                   WIN00790
      Y=JIP*Y                                                              WIN00800
      RETURN                                                               WIN00810
    1 Y=JIP*AMPL(N,J)*COS(PI*TIMEK(N,J)/TIMEL(N,J)*T)                      WIN00820
      RETURN                                                               WIN00830
      END                                                                  WIN00840
```

```
FILE: WDPL     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

C       A.KUCAR,1985 10 27,1986 02 26                                          WDP00010
C       Program WDPL FORTRAN                                                   WDP00020
C       This program calculates the NONLINEARITY SURFACE and CONDITIONAL       WDP00030
C       PROBABILITY OF ERROR of decision feedback carrier recovery loop        WDP00040
C       for any WQAM (staggered and non-staggered) scheme                      WDP00050
C       Pe=Pea is calculated rather than Pe=Pea+Peb-Pea*Peb                    WDP00060
C       Pe(PI=carrier recovery error,SAMB=timing recovery error)               WDP00070
C       The output of this program is used as input to the SAS programs        WDP00080
C       DFL,G1,G2,GFI,NI and GFIN                                              WDP00090
        COMMON L,L2,LM1,LM2,LM1SQ,L2L,L4L                                      WDP00100
        COMMON FI,AMB,PEA,PEB,EBN,RA,RA1,RB,PB1,G1,G2,AN1,FL,P1,P2,P3          WDP00110
        COMMON PEA1,PEB1,PEI1,PEI2,PEJ1,RAL,RBL,GFI,GFIN,BD1,BD2,BD3,BD4       WDP00120
        COMMON PIP1,PIP3                                                       WDP00130
C       CALL ERRSET(208,700,-1,1,1,208)                                        WDP00140
        PI=3.1415926535897 93                                                  WDP00150
        PIR=180./PI                                                            WDP00160
        PIP2=PI+2.                                                             WDP00170
        PIP1=2.*PI/PIP2                                                        WDP00180
        PIP3=4./PIP2                                                           WDP00190
        PIP1=SQRT(PIP1)                                                        WDP00200
        PIP3=SQRT(PIP3)                                                        WDP00210
        WRITE(8,10)                                                            WDP00220
        READ(7,2)IXP                                                           WDP00230
        MIXP=100*IXP                                                           WDP00240
        L=2                                                                    WDP00250
        L2=L*2                                                                 WDP00260
        LM1=L-1                                                                WDP00270
        LM2=L-2                                                                WDP00280
        FL=FLOAT(L)                                                            WDP00290
        FM=FL*FL                                                               WDP00300
        P1=3.0/(FM-1.)                                                         WDP00310
        FM=FM*FM                                                               WDP00320
        P2=1./FM/FM/2.                                                         WDP00330
        P3=P2*P1                                                               WDP00340
        PEP=FL                                                                 WDP00350
        PEP=10.**(-IXP)*PEP                                                    WDP00360
        CALL MBRQCI(PEP,EBNO,IER)                                              WDP00370
        EBN=EBNO                                                               WDP00380
        DO 5 I=1,25                                                            WDP00390
        WRITE(8,6)I                                                            WDP00400
        SAMB=FLOAT(I-13)/24.                                                   WDP00410
        AMB=ABS(SAMB)                                                          WDP00420
        RA1=(1.-AMB)*PIP1                                                      WDP00430
        RA=AMB                                                                 WDP00440
        RB1=(0.5-AMB)*PIP1                                                     WDP00450
        RB=(0.5+AMB)*PIP1                                                      WDP00460
        BB=PI*AMB                                                              WDP00470
        BD3=(1.-SIN(BB))/2.                                                    WDP00480
        BD4=(1.+SIN(BB))/2.                                                    WDP00490
        BB=BB/2.                                                               WDP00500
        BD1=COS(BB)**2.                                                        WDP00510
        BD2=1.-BD1                                                             WDP00520
        DO 5 J=1,91                                                            WDP00530
        DEG1=FLOAT(J-46)*4.                                                    WDP00540
        FI=DEG1/PIR                                                            WDP00550
```

```
FILE: WDPL     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

      CALL UD1                                                          WDP00560
    5 WRITE(1,7)SAMB,DEG1,PEA,G1,G2,AN1,GFI,GFIN                        WDP00570
    1 FORMAT('# of levels L=04')                                        WDP00580
    2 FORMAT(I2)                                                        WDP00590
    3 FORMAT('NPE and EbNo=',I9,E16.3)                                  WDP00600
    6 FORMAT('I=',I4)                                                   WDP00610
    7 FORMAT(8E10.3)                                                    WDP00620
    8 FORMAT('EbNo=',F7.2,' at Pe=1E-',I2)                              WDP00630
   10 FORMAT('BER exp=06')                                              WDP00640
  101 FORMAT('Staggered=01,Non-staggered=00')                           WDP00650
      END                                                               WDP00660
C     ================================================================  WDP00670
      SUBROUTINE UD1                                                    WDP00680
      COMMON L,L2,LM1,LM2,LM1SQ,L2L,L4L                                 WDP00690
      COMMON FI,AMB,PEA,PEB,EBN,RA,RA1,RB,RB1,G1,G2,AN1,FL,F1,F2,F3     WDP00700
      COMMON PEA1,PEB1,PEI1,PEI2,PEJ1,RAL,RBL,GFI,GFIN,BD1,BD2,BD3,BD4  WDP00710
      COMMON PIP1,PIP3                                                  WDP00720
      PEB=0.                                                            WDP00730
      PEI2=0.                                                           WDP00740
      G1=0.                                                             WDP00750
      G2=0.                                                             WDP00760
      C1=0.                                                             WDP00770
      C2=0.                                                             WDP00780
      CX=COS(FI)                                                        WDP00790
      SX=SIN(FI)                                                        WDP00800
C     ----------------------------------------------------------------  WDP00810
      DO 14 M0=1,L2,2                                                   WDP00820
      M=M0-L                                                            WDP00830
      FM1=FLOAT(M-1)                                                    WDP00840
      FP1=FM1+2.                                                        WDP00850
      IM=IABS(M)                                                        WDP00860
      PEN=0.                                                            WDP00870
      PEIN=0.                                                           WDP00880
      DO 12 N0=1,L2,2                                                   WDP00890
      N=N0-L                                                            WDP00900
      PEA1=0.                                                           WDP00910
      DO 11 M9=1,L2,2                                                   WDP00920
      M1=M9-L                                                           WDP00930
      DO 11 M8=1,3,2                                                    WDP00940
      M7=M8-L                                                           WDP00950
      I3=IABS(M+M1+M7)                                                  WDP00960
      IF (I3.EQ.3) RAM1=M*RA1                                           WDP00970
      IF (M1.NE.M7) RAM1=M*BD1                                          WDP00980
      IF ((M7.EQ.M1).AND.(M7.NE.M)) RAM1=M*BD1*PIP3                     WDP00990
      DO 11 M6=1,3,2                                                    WDP01000
      M5=M6-L                                                           WDP01010
      I4=IABS(M+M1+M5)                                                  WDP01020
      IF (I4.EQ.3) RAM=M1*RA*PIP1                                       WDP01030
      IF ((M.NE.M5).AND.(M1.NE.M)) RAM=M1*BD2                           WDP01040
      IF ((M.NE.M5).AND.(M1.EQ.M)) RAM=M1*RA                            WDP01050
      IF ((M.EQ.M5).AND.(M1.NE.M)) RAM=M1*BD2*PIP3                      WDP01060
      RA1=(RAM1+RAM)*CX                                                 WDP01070
      DO 11 N9=1,L2,2                                                   WDP01080
      N1=N9-L                                                           WDP01090
      DO 11 N8=1,3,2                                                    WDP01100
```

```
      N7=N8-L                                                                   WDP01110
      J3=IABS(N+N1+N7)                                                          WDP01120
      IF (J3.EQ.3) RAN1=N*RB1                                                   WDP01130
      IF ((N7.EQ.N1).AND.(N7.NE.N)) RAN1=N*BD3*PIP3                             WDP01140
      IF ((N7.NE.N1).AND.(N.EQ.N7)) RAN1=N*BD3                                  WDP01150
      IF ((N7.NE.N1).AND.(N.NE.N7)) RAN1=N*RB1/PIP1                             WDP01160
      DO 11 N6=1,3,2                                                            WDP01170
      N5=N6-L                                                                   WDP01180
      J4=IABS(N+N1+N5)                                                          WDP01190
      IF (J4.EQ.3) RAN=N1*RB                                                    WDP01200
      IF ((N.EQ.N5).AND.(N1.NE.N)) RAN=N1*BD4*PIP3                              WDP01210
      IF (N.NE.N5) RAN=N1*BD4                                                   WDP01220
      YB1=(RAN1+RAN)*SX                                                         WDP01230
      YAB1=YA1+YB1                                                              WDP01240
      IF (M0.EQ.1) PEA=ERFC((FP1-YAB1)*EBN)                                     WDP01250
      IF (M.EQ.LM1) PEA=ERFC((YAB1-FM1)*EBN)                                    WDP01260
      IF (M.LT.LM1) PEA=ERFC((YAB1-FM1)*EBN)-ERFC((FP1-YAB1)*EBN)               WDP01270
   11 PEA1=PEA1+PEA                                                             WDP01280
      PEN=PEN+PEA1                                                              WDP01290
   12 PEIN=PEIN+N*PEA1                                                          WDP01300
      PEB=PEB+PEN                                                               WDP01310
      G2=G2+M*PEIN                                                              WDP01320
      C1=C1+M*M*PEN                                                             WDP01330
      DO 14 I9=1,L2,2                                                           WDP01340
      IF (I9.EQ.M0) GOTO 14                                                     WDP01350
      I1=I9-L                                                                   WDP01360
      FPI1=FLOAT(I1+1)                                                          WDP01370
      FMI1=FPI1-2.                                                              WDP01380
      IA=IABS(I1)                                                               WDP01390
      PENI=0.                                                                   WDP01400
      PEJ1=0.                                                                   WDP01410
      DO 13 N0=1,L2,2                                                           WDP01420
      N=N0-L                                                                    WDP01430
      PEI1=0.                                                                   WDP01440
      DO 16 M9=1,L2,2                                                           WDP01450
      M1=M9-L                                                                   WDP01460
      DO 16 M8=1,3,2                                                            WDP01470
      M7=M8-L                                                                   WDP01480
      I3=IABS(M+M1+M7)                                                          WDP01490
      IF (I3.EQ.3) RAM1=M*RA1                                                   WDP01500
      IF (M1.NE.M7) RAM1=M*BD1                                                  WDP01510
      IF ((M7.EQ.M1).AND.(M7.NE.M)) RAM1=M*BD1*PIP3                             WDP01520
      DO 16 M6=1,3,2                                                            WDP01530
      M5=M6-L                                                                   WDP01540
      I4=IABS(M+M1+M5)                                                          WDP01550
      IF (I4.EQ.3) RAM=M1*RA*PIP1                                               WDP01560
      IF ((M.NE.M5).AND.(M1.NE.M)) RAM=M1*BD2                                   WDP01570
      IF ((M.NE.M5).AND.(M1.EQ.M)) RAM=M1*RA                                    WDP01580
      IF ((M.EQ.M5).AND.(M1.NE.M)) RAM=M1*BD2*PIP3                              WDP01590
      YA1=(RAM1+RAM)*CX                                                         WDP01600
      DO 16 N9=1,L2,2                                                           WDP01610
      N1=N9-L                                                                   WDP01620
      DO 16 N8=1,3,2                                                            WDP01630
      N7=N8-L                                                                   WDP01640
      J3=IABS(N+N1+N7)                                                          WDP01650
```

```
FILE: WDPL     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

      IF (J3.EQ.3) RAN1=N*RB1                                          WDP01660
      IF ((N7.EQ.N1).AND.(N7.NE.N)) RAN1=N*BD3*PIP3                    WDP01670
      IF ((N7.NE.N1).AND.(N.EQ.N7)) RAN1=N*BD3                         WDP01680
      IF ((N7.NE.N1).AND.(N.NE.N7)) RAN1=N*RB1/PIP1                    WDP01690
      DO 16 N6=1,3,2                                                   WDP01700
      N5=N6-L                                                          WDP01710
      J4=IABS(N+N1+N5)                                                 WDP01720
      IF (J4.EQ.3) RAN=N1*RB                                           WDP01730
      IF ((N.EQ.N5).AND.(N1.NE.N)) RAN=N1*BD4*PIP3                     WDP01740
      IF (N.NE.N5) RAN=N1*BD4                                          WDP01750
      YB1=(RAN1+RAN)*SX                                                WDP01760
      YAB1=YA1+YB1                                                     WDP01770
      IF (I9.EQ.1) PEI=ERFC((YAB1-FPI1)*EBN)                           WDP01780
      IF (I1.EQ.LM1) PEI=ERFC((FMI1-YAB1)*EBN)                         WDP01790
      IF ((IA.LT.LM1).AND.(I1.GT.M))                                   WDP01800
     *PEI=ERFC((FMI1-YAB1)*EBN)-ERFC((PPI1-YAB1)*EBN)                  WDP01810
      IF ((IA.LT.LM1).AND.(I1.LT.M))                                   WDP01820
     *PEI=ERFC((YAB1-FPI1)*EBN)-ERFC((YAB1-FMI1)*EBN)                  WDP01830
   16 PEI1=PEI1+PEI                                                    WDP01840
      PENI=PENI+PEI1                                                   WDP01850
   13 PEJ1=PEJ1+N*PEI1                                                 WDP01860
      PEI2=PEI2+I1*I1*PENI                                             WDP01870
      G2=G2-I1*PEJ1                                                    WDP01880
      C2=C2+I1*M*PENI                                                  WDP01890
   14 CONTINUE                                                         WDP01900
      G1=C1-G2                                                         WDP01910
      AN1=C1-PEI2                                                      WDP01920
C-----------------------------------------------------------------     WDP01930
      DO 24 M0=1,L2,2                                                  WDP01940
      M=M0-L                                                           WDP01950
      FM1=FLOAT(M-1)                                                   WDP01960
      FP1=FM1+2.                                                       WDP01970
      IM=IABS(M)                                                       WDP01980
      PEIN=0.                                                          WDP01990
      PEN=0.                                                           WDP02000
      DO 22 N0=1,L2,2                                                  WDP02010
      N=N0-L                                                           WDP02020
      PEA1=0.                                                          WDP02030
      DO 21 M9=1,L2,2                                                  WDP02040
      M1=M9-L                                                          WDP02050
      DO 21 M8=1,3,2                                                   WDP02060
      M7=M8-L                                                          WDP02070
      I3=IABS(M+M1+M7)                                                 WDP02080
      IF (I3.EQ.3) RAM1=M*RA1                                          WDP02090
      IF (M1.NE.M7) RAM1=M*BD1                                         WDP02100
      IF ((M7.EQ.M1).AND.(M7.NE.M)) RAM1=M*BD1*PIP3                    WDP02110
      DO 21 M6=1,3,2                                                   WDP02120
      M5=M6-L                                                          WDP02130
      I4=IABS(M+M1+M5)                                                 WDP02140
      IF (I4.EQ.3) RAM=M1*RA*PIP1                                      WDP02150
      IF ((M.NE.M5).AND.(M1.NE.M)) RAM=M1*BD2                          WDP02160
      IF ((M.NE.M5).AND.(M1.EQ.M)) RAM=M1*RA                           WDP02170
      IF ((M.EQ.M5).AND.(M1.NE.M)) RAM=M1*BD2*PIP3                     WDP02180
      YA1=(RAM1+RAM)*CX                                                WDP02190
      DO 21 N9=1,L2,2                                                  WDP02200
```

```
      M1=N9-L                                                                   WDP02210
      DO 21 N8=1,3,2                                                            WDP02220
      M7=N8-L                                                                   WDP02230
      J3=IABS(N+N1+M7)                                                          WDP02240
      IF (J3.EQ.3) RAN1=N*RB1                                                   WDP02250
      IF ((M7.EQ.N1).AND.(M7.NE.N)) RAN1=N*BD3*PIP3                             WDP02260
      IF ((M7.NE.N1).AND.(N.EQ.M7)) RAN1=N*BD3                                  WDP02270
      IF ((M7.NE.N1).AND.(N.NE.M7)) RAN1=N*RB1/PIP1                             WDP02280
      DO 21 N6=1,3,2                                                            WDP02290
      N5=N6-L                                                                   WDP02300
      J4=IABS(N+N1+N5)                                                          WDP02310
      IF (J4.EQ.3) RAN=N1*RB                                                    WDP02320
      IF ((N.EQ.N5).AND.(N1.NE.N)) RAN=N1*BD4*PIP3                              WDP02330
      IF (N.NE.N5) RAN=N1*BD4                                                   WDP02340
      YB1=(RAN1+RAN)*SX                                                         WDP02350
      YAB1=YA1-YB1                                                              WDP02360
      IF (M0.EQ.1) PEA=ERFC((PP1-YAB1)*EBN)                                     WDP02370
      IF (M.EQ.LM1) PEA=ERFC((YAB1-PM1)*EBN)                                    WDP02380
      IF (IM.LT.LM1) PEA=ERFC((YAB1-PM1)*EBN)-ERFC((PP1-YAB1)*EBN)              WDP02390
   21 PEA1=PEA1+PEA                                                             WDP02400
      PEN=PEN+PEA1                                                              WDP02410
   22 PEIN=PEIN+N*PEA1                                                          WDP02420
      G2=G2-M*PEIN                                                              WDP02430
      DO 24 I9=1,L2,2                                                           WDP02440
      IF (I9.EQ.M0) GOTO 24                                                     WDP02450
      I1=I9-L                                                                   WDP02460
      PPI1=FLOAT(I1+1)                                                          WDP02470
      PMI1=PPI1-2.                                                              WDP02480
      IA=IABS(I1)                                                               WDP02490
      PENI=0.                                                                   WDP02500
      PEJ1=0.                                                                   WDP02510
      DO 23 N0=1,L2,2                                                           WDP02520
      N=N0-L                                                                    WDP02530
      PEI1=0.                                                                   WDP02540
      DO 26 M9=1,L2,2                                                           WDP02550
      M1=M9-L                                                                   WDP02560
      DO 26 M8=1,3,2                                                            WDP02570
      M7=M8-L                                                                   WDP02580
      I3=IABS(M+M1+M7)                                                          WDP02590
      IF (I3.EQ.3) RAM1=M*RA1                                                   WDP02600
      IF (M1.NE.M7) RAM1=M*BD1                                                  WDP02610
      IF ((M7.EQ.M1).AND.(M7.NE.M)) RAM1=M*BD1*PIP3                             WDP02620
      DO 26 M6=1,3,2                                                            WDP02630
      M5=M6-L                                                                   WDP02640
      I4=IABS(M+M1+M5)                                                          WDP02650
      IF (I4.EQ.3) RAM=M1*RA*PIP1                                               WDP02660
      IF ((M.NE.M5).AND.(M1.NE.M)) RAM=M1*BD2                                   WDP02670
      IF ((M.NE.M5).AND.(M1.EQ.M)) RAM=M1*RA                                    WDP02680
      IF ((M.EQ.M5).AND.(M1.NE.M)) RAM=M1*BD2*PIP3                              WDP02690
      RA1=(RAM1+RAM)*CX                                                         WDP02700
      DO 26 N9=1,L2,2                                                           WDP02710
      N1=N9-L                                                                   WDP02720
      DO 26 N8=1,3,2                                                            WDP02730
      M7=N8-L                                                                   WDP02740
      J3=IABS(N+N1+M7)                                                          WDP02750
```

```
FILE: WDPL     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 8

      IF (J3.EQ.3) RAN1=N*RB1                                                  WDP02760
      IF ((N7.EQ.N1).AND.(N7.NE.N)) RAN1=N*BD3*PIP3                            WDP02770
      IF ((N7.NE.N1).AND.(N.EQ.N7)) RAN1=N*BD3                                 WDP02780
      IF ((N7.NE.N1).AND.(N.NE.N7)) RAN1=N*RB1/PIP1                            WDP02790
      DO 26 N6=1,3,2                                                           WDP02800
      N5=N6-L                                                                  WDP02810
      J4=IABS(N+N1+N5)                                                         WDP02820
      IF (J4.EQ.3) RAN=N1*RB                                                   WDP02830
      IF ((N.EQ.N5).AND.(N1.NE.N)) RAN=N1*BD4*PIP3                             WDP02840
      IF (N.NE.N5) RAN=N1*BD4                                                  WDP02850
      YB1=(RAN1+RAN)*SX                                                        WDP02860
      YAB1=YA1-YB1                                                             WDP02870
      IF (I9.EQ.1) PEI=ERFC((YAB1-FPI1)*EBN)                                   WDP02880
      IF (I1.EQ.LM1) PEI=ERFC((FMI1-YAB1)*EBN)                                 WDP02890
      IF ((IA.LT.LM1).AND.(I1.GT.M))                                           WDP02900
     *PEI=ERFC((FMI1-YAB1)*EBN)-ERFC((FPI1-YAB1)*EBN)                          WDP02910
      IF ((IA.LT.LM1).AND.(I1.LT.M))                                           WDP02920
     *PEI=ERFC((YAB1-FPI1)*EBN)-ERFC((YAB1-FMI1)*EBN)                          WDP02930
   26 PEI1=PEI1+PEI                                                           WDP02940
      PENI=PENI+PEI1                                                           WDP02950
   23 PEJ1=PEJ1+N*PEI1                                                        WDP02960
      G2=G2+I1*PEJ1                                                            WDP02970
   24 CONTINUE                                                                 WDP02980
C-----------------------------------------------------------------------      WDP02990
      G2=G2*F3/2.                                                              WDP03000
      G1=1.-G1*F3                                                              WDP03010
      AN1=1.-AN1*F3                                                            WDP03020
      PEA=-ALOG10(PEB*F2)                                                      WDP03030
      GFI=G1*SX+G2*CX                                                          WDP03040
      GFIN=2.*GFI/AN1                                                          WDP03050
      RETURN                                                                   WDP03060
      END                                                                      WDP03070
```

```
FILE: 4QAM     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 4

C      KUCAR,1983 10 19,1985 04 29                                          4QA00010
CCCCC  ************************************************************        4QA00020
       PROGRAM TO EVALUATE THE  PERFORMANCE OF 4QAM MODEM IN                4QA00030
       A NONLINEARLY AMPLIFIED MULTICHANNEL ENVIRONMENT.                    4QA00040
       ************************************************************        4QA00050
C                                                                           4QA00060
       COMMON /I1/IOFF,LSAMPL,NSYMB,NO1,NO2,NUMB(4,20),NI(4,4096)           4QA00070
       COMMON /R1/SAM,PI,BAUD,BW,AMPL(4,4),TIMEL(4,4),TIMEK(4,30)           4QA00080
       COMPLEX DATA(4096),TF(4096),DATA1(4596),TA(514)                      4QA00090
       DOUBLE PRECISION PID,DSEED                                           4QA00100
       DIMENSION XB(8192),PSX(65),NWK(7),IWK(12),WK(64)                     4QA00110
       DIMENSION PRI(50),BER(50)                                            4QA00120
       DIMENSION NAME(10)                                                   4QA00130
C      CALL ERRSET(208,700,-1,1,1,208)                                      4QA00140
       IJK=0                                                                4QA00150
       JKL=0                                                                4QA00160
       PI=3.141592653589793                                                 4QA00170
       PI2=PI*2.                                                            4QA00180
       P2=PI/2.                                                             4QA00190
       PID=3.141592653589793D0                                              4QA00200
       DSEED=4569286.D0                                                     4QA00210
       IFROM=5                                                              4QA00220
       ITO=30                                                               4QA00230
       NR=1                                                                 4QA00240
       LSAMPL=32                                                            4QA00250
       NSYMB=128                                                            4QA00260
       MI=LSAMPL/2                                                          4QA00270
       MQ=MI                                                                4QA00280
       IOFF=MI                                                              4QA00290
       SYM=FLOAT(NSYMB)                                                     4QA00300
       RUNS=FLOAT(NR)                                                       4QA00310
       ONEB=1.                                                              4QA00320
       WRITE(8,101)                                                         4QA00330
  101  FORMAT('Baud rate:MBd')                                              4QA00340
       READ(7,222)BAUD                                                      4QA00350
       SAM=FLOAT(LSAMPL)                                                    4QA00360
       BW=SAM*BAUD                                                          4QA00370
       LD=LSAMPL*NSYMB                                                      4QA00380
       LD2=2*LD                                                             4QA00390
       L1=NSYMB                                                             4QA00400
       L2=(L1/2)+1                                                          4QA00410
C      L3=L2G(L1)...2**L3=L1 in P1,but 2**L3=LD in FILTER                   4QA00420
       L3=12                                                                4QA00430
       LL3=7                                                                4QA00440
       L4=L1/2                                                              4QA00450
       L5=L1+2                                                              4QA00460
       NO1=LD/2+1                                                           4QA00470
       NO2=NO1+1                                                            4QA00480
       NUMB(2,4)=0                                                          4QA00490
  100  WRITE(8,3)                                                           4QA00500
    3  FORMAT('number of channels')                                         4QA00510
       READ(7,102)NCH                                                       4QA00520
       WRITE(8,8)                                                           4QA00530
    8  FORMAT('number of levels L,L=01 means shaping')                      4QA00540
       READ(7,12)LARY                                                       4QA00550
```

```
      IF (LARY.EQ.1) CALL INP1(NCH,1)                                          4QA00560
    2 CALL INP2(NCH)                                                           4QA00570
      WRITE(8,4)                                                               4QA00580
    4 FORMAT('pen type:1 TO 9')                                                4QA00590
      READ(7,102)IPEN                                                          4QA00600
      LPEN=IPEN+4                                                              4QA00610
C     WRITE(6,6) IPEN,IOFF,NUMB(1,20),LARY                                     4QA00620
      WRITE(8,6) IPEN,IOFF,NUMB(1,20),LARY                                     4QA00630
    6 FORMAT(/,'----------------PEN,OFFSET,NONL,LARY',4I5)                     4QA00640
      DO 5 I=IFROM,ITO                                                         4QA00650
    5 BER(I)=0.                                                                4QA00660
      ANORM=SQRT(1.)                                                           4QA00670
      DO 99 J=1,NR                                                             4QA00680
      CALL ENCODE(LD,DATA,1,LARY)                                              4QA00690
C     CALL P1(LD,LD2,L1,L2,LL3,L4,L5,DATA,XA,XB,PSX,NWK,WK,3,IPEN,1)           4QA00700
C     IF (J.NE.1) GOTO 7                                                       4QA00710
C     CALL PLTSAS(LD,DATA,60,MI,MQ,JKL,IJK,LARY,2)                             4QA00720
C     CALL PLTSAS(LD,DATA,60,MI,MQ,JKL,IJK,LARY,1)                             4QA00730
    7 CALL NONLIN(LD,DATA,ANORM,1)                                             4QA00740
C     IF (IPEN.GT.2)CALL P1(LD,LD2,L1,L2,LL3,L4,L5,DATA,XA,XB,PSX,NWK,WK       4QA00750
C    $,1,LPEN,1)                                                               4QA00760
      CALL FILTER(LD,L3,DATA,TF,IWK,1,1)                                       4QA00770
      IF (NUMB(1,4).GT.1) CALL FILTER(LD,L3,DATA,TF,IWK,1,2)                   4QA00780
      IF (NUMB(1,5).EQ.4) CALL FILTER(LD,L3,DATA,TF,IWK,1,-1)                  4QA00790
      CALL NOISE1(LD,TF,PNOISE)                                                4QA00800
C     CALL CHANN(LD,DATA,SNR,1)                                                4QA00810
      IF (NCH.LT.2) GOTO 66                                                    4QA00820
      DO 10 K=2,NCH                                                            4QA00830
      CALL ENCODE(LD,DATA1,K,LARY)                                             4QA00840
      CALL NONLIN(LD,DATA1,ANORM,K)                                            4QA00850
      PSH=PI2*GSUBFS(DSEED)                                                    4QA00860
      CALL PHASSH(LD,DATA1,PSH)                                                4QA00870
      ITSH=INT(SAM*(GSUBFS(DSEED)-.5))                                         4QA00880
      CALL TIMESH(LD,DATA1,TF,ITSH)                                            4QA00890
      CALL ATT(LD,DATA1,K)                                                     4QA00900
C     CALL FILTER(LD,L3,DATA1,TF,IWK,K,N)                                      4QA00910
      DO 10 I=1,LD                                                             4QA00920
   10 DATA(I)=DATA(I)+DATA1(I)                                                 4QA00930
      CALL POWER(LD,DATA,PMEA,PPEA,EB,0)                                       4QA00940
   66 CALL POWER(LD,DATA,PAVE,PEAK,EB,1)                                       4QA00950
      CALL SYNCH(LD,DATA,TF,PNOISE,MI,MQ,SB,LARY)                              4QA00960
C     CALL PLTSAS(LD,DATA,60,MI,MQ,JKL,IJK,LARY,2)                             4QA00970
C     CALL PLTSAS(LD,DATA,250,MI,MQ,JKL,IJK,LARY,1)                            4QA00980
      CALL DECODE(LD,DATA,PNOISE,PEI,IFROM,ITO,EB,MI,MQ,LARY,CNEB,NER)         4QA00990
      IF (NER.GT.0) GOTO 123                                                   4QA01000
      DO 99 I=IFROM,ITO                                                        4QA01010
      PEI(I)=PEI(I)/SYM                                                        4QA01020
   99 BER(I)=BER(I)+PEI(I)                                                     4QA01030
C     WRITE(6,1)                                                               4QA01040
      IF (CNEB.EQ.0.) WRITE(8,1)                                               4QA01050
    1 FORMAT('----------Eb/No----------Prob.of symbol error----------')       4QA01060
      IF (CNEB.EQ.1.) WRITE(8,111)                                             4QA01070
  111 FORMAT('----------C/N----------Prob.of symbol error----------')         4QA01080
      DO 9 I=IFROM,ITO                                                         4QA01090
      PSH=FLOAT(I)                                                             4QA01100
```

```
FILE: 4QAM     FORTRAN  A   UNIV D'/OF OTTAWA CMS RELEASE 0

      PEI(I)=BER(I)/RUNS                                                    4QA01110
      WRITE(8,172)PSH,PEI(I)                                                4QA01120
  172 FORMAT(2E15.6,I2)                                                     4QA01130
      CALL BERPSP(IPEN,PEI,IFROM,ITO,1.E-12)                                4QA01140
  123 WRITE(8,173)                                                          4QA01150
  173 FORMAT('=================================================',/         4QA01160
     *'NEW PARAM>0,NEW BB SIGNAL<0...')                                     4QA01170
      READ(7,222)GO                                                         4QA01180
      IF (GO) 100,11,2                                                      4QA01190
   11 CONTINUE                                                              4QA01200
   12 FORMAT(I2)                                                            4QA01210
  102 FORMAT(I1)                                                            4QA01220
  222 FORMAT(F15.7)                                                         4QA01230
      END                                                                   4QA01240
```

The last page