INVESTIGATION INTO PRS-PRECODED, CONSTANT-ENVELOPE, CONTINUOUS-PHASE DIGITAL MODULATION SCHEMES

Prepared by: **John N. A. Golby**
B.Sc. Eng. (Elec)

September 1990

A half-dissertation submitted to the Faculty of Engineering, University of Cape Town, in partial fulfilment of the requirements of the degree of Master of Science in Electrical Engineering.

TERMS OF REFERENCE

This thesis investigation was initiated by Dr. Robin Braun of the Department of Electrical Engineering, University of Cape Town, in February 1989. His instructions were:

1. To research angle modulation schemes that employ Partial Response Signaling (PRS) to achieve improved spectral efficiency and power efficiency, while maintaining constant signal envelopes.
2. To select and review a particular scheme of this type.
3. To perform computer simulations of system performance.
4. To build a reduced-complexity modem.
5. To perform testing and evaluation of overall system performance.

ACKNOWLEDGEMENTS

The work presented in this thesis would not have been possible without the assistance of my supervisor, Dr. Robin Braun.

Grateful thanks are also due to Mr. Jan Hesselink, for valuable encouragement and support.

Also, to Jan Schoonees, for review comments, and to all those associated with the development of this project.

SYNOPSIS

Partial response signaling (PRS) has been used successfully to improve the spectral properties of Pulse Amplitude Modulated (PAM) digital transmission systems. This thesis investigation studied the effect of PRS on frequency- and phase-modulated carrier systems, in particular on their spectral performance and their maintenance of constant envelope.

Because of their high immunity to nonlinear signal operations, constant-envelope modulation schemes are attractive for use in digital radio environments (such as mobile and satellite). Traditional power-efficient systems such as QPSK and MSK have high out-of-band signal levels. Filtering of these signals is ineffective because any further nonlinear processing causes spectral spreading.

A literature survey has been conducted, showing the development of constant-envelope modulation formats that employ some form of correlative encoding. The most promising systems to date are those that employ Continuous Phase Frequency Shift Keying (CPFSK) with some form of baseband pulse shaping. Systems that fall into this category are Continuous Phase Modulation (CPM) and Correlative Phase Shift Keying (CORPSK).

CORPSK has been studied as a typical scheme from this class. Computer simulations were performed to analyse spectral behaviour, time domain characteristics and the performance of synchronization systems. These results agreed with those reported in the literature.

A reduced-complexity CORPSK modem was developed, employing coherent symbol-by-symbol detection instead of the Viterbi algorithm. The former experiences a 4.5 dB degradation over the Viterbi approach.

Extensive testing was performed to determine system performance. Results show that a spectral efficiency of only 0.5 bps/Hz is achieved. However, this modulation format has no spectral sidelobes, and so the out-of-band signal levels are much lower than those yielded by systems such as unfiltered QPSK. CORPSK is thus able to satisfy more stringent spectral masks than constant-envelope OQPSK and MSK. Measured error rate characteristics are comparable to the performance of a 4 PSK system.

Carrier recovery is straightforward — good performance is achieved with fairly simple circuitry. Symbol timing recovery, while possible, requires more sophisticated synchronization, and was not attempted. However, the existence of suitable spectral components was verified.

The effect of hardlimited amplification on the system spectrum was simulated. No spectral sidelobes were generated, demonstrating this system's immunity to spectral spreading.

Conclusions are that CORPSK yields a hardware solution that is practical both in terms of system complexity and performance. This modulation format exhibits spectral properties that are easily predicted by simulation.

The in-depth study of actual system performance under nonlinear conditions (bandpass limiting, TWTA nonlinearity, and even multipath) is recommended for further research.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

INTRODUCTION

Data transmission theory strives to achieve the transmission of digital information using the smallest possible signal powers occupying the least possible spectral space, yielding the lowest possible data error rates. Power efficiency and spectral efficiency have usually been seen as mutually exclusive goals of the design process and, in practice, some form of compromise or trade-off has had to be made between them.

With the increasing diversity of the global telecommunications network, efficient satellite communications and mobile (cellular) communications are becoming increasingly important. As a result, there is increasing demand for power efficient digital modulation schemes with high spectral efficiency. Current endeavours in increasing the spectral efficiency of existing power efficient systems are an attempt to satisfy this need. However, a further constraint imposed by the practical limitations of technology is that such signal formats have a constant envelope.

Constant-envelope modulation schemes are very attractive for use in digital radio because of their high tolerance to the nonlinear operations convenient in high-power amplifiers and receiver IF stages. Traditionally, four-phase systems such as OQPSK and MSK have been used because they represent a good compromise between necessary bandwidth and power efficiency (the former because stringent limitations on out-of-band power must often be met, and the latter because good error performance must be achieved with only moderate power levels). However, any improvement in out-of-band signal level can only be achieved by filtering of the signal, and this introduces envelope fluctuations and therefore reduces these scheme's immunity to the effects of nonlinearities. In many instances, filtering of the output signal is simply not possible, because of the high filter Q-factors and signal powers involved. Any further nonlinear amplification of the signal simply results in spectral spreading.

Clearly, there exists a need to develop constant-envelope modulation schemes whose spectral efficiency is higher than that yielded by power-efficient systems such as QPSK. Spectrally-efficient modems are currently being developed in the amplitude modulation (PAM) area, employing Partial Response Signaling (PRS). Their signals, however, have fluctuating envelopes. This investigation studies the effect of applying PRS to frequency- and phase-modulated systems with the aim of realising the same benefits in angle modulation as are yielded in the amplitude domain.

This study focuses on one scheme in particular - Correlative Phase Shift Keying (CORPSK). Tamed Frequency Modulation, or TFM, has been shown to be an example of this system.

A literature survey is presented, showing the development of PRS precoded systems that exhibit constant signal envelopes. A review of partial response signaling is then followed by a description of the Correlative Phase Shift Keyed modulation scheme. Computer simulation results are then presented for a particular case of this system, including baseband eye diagrams, RF waveform generation, baseband and RF spectrum analyses, and signal space diagrams. Also, various carrier regeneration methods are simulated to assess their viability. The results of a symbol error rate software simulation are presented. The effect of hardlimiting on the signal spectrum is also simulated, using an approximation of this process, to yield a qualitative appreciation of this scheme's immunity to the effects of system nonlinearities.

The design of a CORPSK modem is described. Hardware testing and evaluation is performed, and some suggestions for further research are presented.

CHAPTER 2
LITERATURE REVIEW

The scope of this thesis investigation is limited to the design, testing and evaluation of a Correlative Phase Shift Keyed modem, and a survey of *constant-envelope* modulation schemes whose signals are of the form

$$s(t) = A \cdot \cos(\omega_c t + \Phi(t)) \qquad (2.1)$$

Here $\Phi(t)$, the phase function, is derived from the input data sequence by correlation of successive symbols according to some rule, and filtered at baseband level. Further, $\Phi(t)$ is a *continuous* function of time, and $\frac{d\Phi(t)}{dt}$ has no discontinuities ie. there are no steps or impulses in the frequency-modulating function. Since the correlation process produces symbol pulses longer than one symbol interval, these systems are referred to as being of a "partial response" type.

However, because of the historical development of such systems, it is necessary to mention some schemes that are either "full response," some that do not exhibit constant envelopes, or some that have only "linear" phase functions.

In 1963, Lender developed "Correlative Level Coding" (Partial Response Signaling) as a means of achieving higher data packing densities, improved spectral shape and error-detection capability in PAM-type modulation schemes [2.1], [2.2]. He also applied this coding principle to FSK data transmission, yielding performance superior to the conventional, differentially phase-modulated systems.

The principles of partial response signaling were reviewed by Kabal and Pasupathy in 1975 [2.3]. They developed a theoretical model to analyse PRS systems in terms of bandwidth, spectral nulls and continuity of derivatives. They compared several schemes on the basis of speed tolerance, minimum eye width and SNR

degradation over binary systems. The effect of precoding on system performance was also addressed.

Over the period from 1977 to 1979, Taylor and Cheung developed a Duobinary-shaped offset-QPSK system for use on the Canadian RD-3 long-haul network [2.4], [2.5]. This system allowed the transmission of two DS3 channels (44.7 Mb.p.s.) in a bandwidth less than 45 MHz, with a signal-to-noise penalty of only 3.2 dB over conventional QPSK. A substantial gain is achieved over sixteen-phase modulation, which represents the most power-efficient phase-modulated scheme capable of this speed. (The resulting spectral efficiency was 2 b.p.s./Hz, placing this form of QPSK in the spectrally-efficient category). A difficulty with this system, however, was the complex method required for carrier recovery at the receiver since, unlike with QPSK, a frequency quadrupler and phase-locked loop were unable to process the partly amplitude-modulated and partly phase-modulated signal.

Garrison [2.6] analysed the power spectral densities of various FSK and digital FM systems, some with correlation between input bits. He evaluated the spectra of, amongst others, Minimum Shift Keying (MSK) and Duobinary coded MSK. The effect of the correlation is to reduce the size of the main lobe and to decrease the out-of-band signal level.

Tamed Frequency Modulation (TFM) [2.7] ensured that the phase-modulating function was smoothed and correlated, yielding not only improved spectral properties over systems such as MSK, but only a 1 dB penalty compared to four-phase methods. Elnoubi and Gupta [2.8] analysed the error rate performances of noncoherent detection of Duobinary-encoded MSK and TFM in mobile radio environments characterized by fast Rayleigh-type fading and cochannel interference. When differential detection of these systems is employed, the MSK scheme exhibits improved error performance over TFM.

A major difficulty in the design of constant-envelope, continuous

(and linear) phase systems has been the characterization and theoretical analysis of their spectra and error performances. In many cases, an exact formula cannot be found.

Deshpande and Wittke derived an explicit expression for the spectrum of a correlative encoded digital FM signal [2.9]. Making use of an encoding polynomial similar to that used in TFM, a spectral efficiency improvement of 47% over MSK resulted. They also investigated the use of different baseband pulse shapes (eg. raised cosine as opposed to rectangular, which creates a discontinuity in the frequency-modulating function).

Anderson *et al* studied similar schemes and compared them in the power-bandwidth plane [2.10]. Their results indicate that, within the constraints imposed by Shannon's channel capacity theorem, any power-bandwidth combination may be achieved with phase codes. Further, channel nonlinearity is no longer seen as an obstacle to spectrum efficiency. On the other hand, efficient phase-modulated schemes employ low modulation indices, and so high-quality phase synchronization is required.

Continuous Phase Modulation (CPM) was reported in 1981 [2.11], [2.12]. Aulin and Sundberg showed that *M*-ary full response CPM systems have improved spectral properties and gain in E_b/N_0, compared to MSK. The use of multilevel data inputs as well as smoothing of the phase trajectory at transition instants yield a far more attractive trade-off between error performance and spectrum efficiency.

Further improvements result from using baseband pulses longer than one symbol interval ie. partial response CPM. Using this approach, modulation schemes can be found that are both bandwidth- and power-efficient.

Aulin, Rydbeck and Sundberg [2.12] reported power-bandwidth comparisons for a number of partial response CPM (PRCPM) schemes, as well as evaluating their minimum Euclidean distances for

different modulation indices. An algorithm is described for computation of the minimum distance.

By 1982, interest had developed in the use of "multi-*h*" CPM — modulation indices are rotated cyclically from symbol interval to symbol interval in a known sequence. The effect is to introduce more variety between phase shifts, and this improves the system's power efficiency [2.20]. Aulin and Sundberg studied the minimum Euclidean distances of multi-*h* partial response phase codes [2.13], as well as spectra [2.14]. Their results indicate that multilevel input systems provide good power-bandwidth trade-offs, and that multilevel input, fixed-*h* systems are preferable to multi-*h*, binary input schemes. Better spectral efficiencies are yielded by employing partial response signaling.

Finally, Aulin and Sundberg [2.15] derived a numerical method with application to computer use for calculating accurate power spectral densities for these smoothed-phase modulation schemes.

Considerable attention has been paid to the detection methods used in PRCPM. Svensson, Aulin and Sundberg reported a robust, reduced-complexity approach to implementing the Viterbi receiver [2.16]. They also derived the optimum receiver for this case. Svensson extended this study to general CPM signals in the presence not only of white Gaussian noise, but also multiple interfering signals [2.17].

The working group went on to study the use of both the serial and parallel MSK receivers in PRCPM [2.18]. Both perform equally well when synchronization is perfect, although the serial receiver is easier to implement. The serial receiver is less sensitive to phase errors but more so to bit timing errors than the parallel. They also calculated the optimum filters for the parallel and serial cases. In short, the performance and implementation advantages of the serial receiver over the parallel are the same as for conventional MSK as they are here.

Bellini and Tartara [2.19] described the use of discriminator detection of PRCPM, with results approaching those yielded by coherent methods. They also reported the development of a moderate-complexity baseband processor for use with binary systems.

Muilwijk [2.20] reported the development of Correlative Phase Shift Keying in 1981. This work was a generalization of smoothed-phase modulation systems with PRS precoding, and included TFM. This system yielded necessary bandwidths similar to unfiltered (ie. constant-envelope) QPSK and MSK, but far lower out-of-band signal levels. Using four phase positions, a power gain of 2 dB may be achieved over MSK and DQPSK. It was shown that room existed for trade-offs between power efficiency, bandwidth and circuit complexity. Also, practical results for TFM were reported.

It should be noted that, while full response multi-*h* CPM systems introduce variety in symbol-to-symbol phase shifts to enhance power efficiency, CORPSK achieves it by using correlative coding — more phase shifts are created than the number of input levels, the actual number of phase shifts depending on the correlation rule used.

Maseng [2.21] showed in 1985 that CPM signals could be approximated in practice by a class of overlapped-pulse phase-modulated signals, whose pulse shaping is achieved by finite impulse response (FIR) digital filters. This system is particularly suited to discrete processing and VLSI design.

REFERENCES

[2.1] A. Lender, *"The Duobinary Technique for High-speed Data Transmission,"* IEEE Trans. Commun. Electron., Vol. 82, May 1963, pp. 214-218.

[2.2] S. Pasupathy, *"Correlative Coding: A Bandwidth-efficient Signaling Scheme,"* IEEE Communications Magazine, July 1977, pp. 4-11.

[2.3] P. Kabal and S. Pasupathy, *"Partial-Response Signaling,"* IEEE Trans. Commun., Vol. COM-23, No. 9, September 1975, pp. 921-934.

[2.4] D. P. Taylor and D. Cheung, *"The Effect of Carrier Phase Error on the Performance of a Duobinary Shaped QPSK Signal,"* IEEE Trans. Commun., Vol. COM-25, July 1977, pp. 738-744.

[2.5] D. P. Taylor and D. Cheung, *"A Decision-directed Carrier Recovery Loop for Duobinary Encoded Offset QPSK Signals,"* IEEE Trans. Commun., Vol. COM-27, No. 2, February 1979, pp.461-468.

[2.6] G. J. Garrison, *"A Power Spectral Density Analysis for Digital FM,"* IEEE Trans. Commun., Vol. COM-23, No. 11, November 1975, pp. 1228-1243.

[2.7] F. de Jager and C. B. Dekker, *"Tamed Frequency Modulation, A Novel Method to Achieve Spectrum Economy in Digital Transmission,"* IEEE Trans. Commun., Vol. COM-26, No. 5, May 1978, pp. 534-542.

[2.8] S. Elnoubi and S. C. Gupta, *"Error Rate Performance of Noncoherent Detection of Duobinary Coded MSK and TFM in Mobile Radio Communication Systems,"* IEEE Trans. Veh. Technol., Vol. VT-30, No. 2, May 1981, pp. 62-76.

[2.9] G. S. Deshpande and P. H. Wittke, *"Correlative Encoded Digital FM,"* IEEE Trans. Commun., Vol. COM-29, No. 2, February 1981, pp. 156-162.

[2.10] J. B. Anderson, C-E. W. Sundberg, T. Aulin, N. Rydbeck, *"Power-bandwidth Performance of Smoothed Phase Modulation Codes,"* IEEE Trans. Commun., Vol. COM-29, No. 3, March 1981, pp. 187-195.

[2.11] T. Aulin and C-E. W. Sundberg, *"Continuous Phase Modulation — Part I: Full Response Signaling,"* IEEE Trans. Commun., Vol. COM-29, No. 3, March 1981, pp. 196-209.

[2.12] T. Aulin, N. Rydbeck and C-E. W. Sundberg, *"Continuous Phase Modulation — Part II: Partial Response Signaling,"* IEEE Trans. Commun., Vol. COM-29, No. 3, March 1981, pp. 210-215.

[2.13] T. Aulin and C-E. Sundberg, *"On the Minimum Euclidean*

Distance for a Class of Signal Space Codes," IEEE Trans. Information Theory, Vol. IT-28, No. 1, January 1982, pp. 43-55.

[2.14] T. Aulin and C-E. Sundberg, *"Minimum Euclidean Distance and Power Spectrum for a Class of Smoothed Phase Modulation Codes with Constant Envelope,"* IEEE Trans. Commun., Vol. COM-30, No. 7, July 1982, pp 1721-1729.

[2.15] T. Aulin and C-E. Sundberg, *"Exact Asymptotic Behaviour of Digital FM Spectra,"* IEEE Trans. Commun., Vol. COM-30, No. 11, November 1982, pp. 2438-2449.

[2.16] A. Svensson, C-E. Sundberg and T. Aulin, *"A Class of Reduced-complexity Viterbi Detectors for Partial Response Continuous Phase Modulation,"* IEEE Trans. Commun., Vol. COM-32, No. 10, October 1984, pp. 1079-1087.

[2.17] N. A. B. Svensson, *"Error Probability Analysis for Continuous Phase Modulation with Viterbi Detection on a Gaussian Channel with Multiple Signal Interference,"* IEEE Trans. Commun., Vol. COM-37, No. 3, March 1989, pp. 230-237.

[2.18] A. Svensson and C-E. Sundberg, *"Serial MSK-type Detection of Partial Response Continuous Phase Modulation,"* IEEE Trans. Commun., Vol. COM-33, No. 1, January 1985, pp. 44-52.

[2.19] S. Bellini and G. Tartara, *"Efficient Discriminator Detection of Partial-Response Continuous Phase Modulation,"* IEEE Trans. Commun., Vol. COM-33, No. 8 August 1985, pp. 883-886.

[2.20] D. Muilwijk, *Correlative Phase Shift Keying — A Class of Constant Envelope Modulation Techniques,"* IEEE Trans. Commun., Vol. COM-29, No. 3, March 1981, pp. 226-236.

[2.21] T. Maseng, *"Digitally Phase Modulated (DPM) Signals,"* IEEE Trans. Commun., Vol. COM-33, No. 9, September 1985, pp. 911-918.

Chapter 3
Partial Response Signaling - A review

A constraint imposed on the design of Pulse-Amplitude Modulation (PAM) systems is that signal waveshapes should contain no intersymbol interference (ISI); that is, at any symbol instant, the received waveform sample should contain no information from preceding symbols. Such systems are referred to as "memoryless". This constraint is summarised in Nyquist's criteria for zero ISI.

It is possible, however, to allow a given quantity of ISI to occur, thereby causing reshaping of the system spectrum and increases in spectral efficiency. Because the extent of the ISI is specified, its effect can be removed at the receiver.

This technique, called "partial response signaling," was developed by Lender [3.1], [3.2] in 1963 primarily as a means of doubling the transmission speeds of binary communication systems. Lender achieved Nyquist-rate signaling, a feat not possible with conventional PAM systems.

3.1 Generalised Partial Response Signaling System

Kabal and Pasupathy [3.3] suggest a model for a synchronous data transmission system as shown in Fig. 3.1.

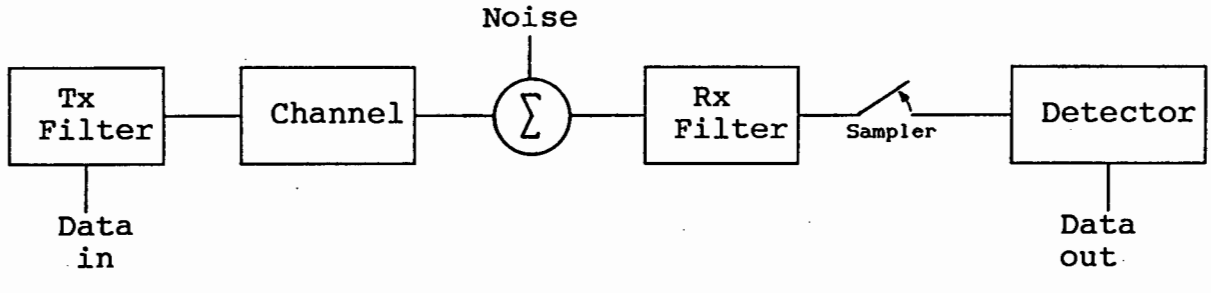


Fig. 3.1. Model for synchronous data communications.

If an ideal, noiseless system is assumed, it may be characterised by the samples of the desired system impulse response, h(t), where $H(\omega)$ would be the system transfer function. $H(\omega)$ encompasses the

transmit filter, equivalent baseband channel and the receive filter.

If it is assumed that there be N adjacent impulse response samples, spanning the entire set of non-zero samples, then

$$F(D) = \sum_{n=0}^{N-1} f_n D^n \qquad (3.1)$$

where $\{f_n\}$, n = 0, 1, 2..., are the sample values, and D is the delay operator. Further, if an input data sequence $\{x_n\}$ is assumed, the output sequence, $\{y_n\}$, will be given by

$$Y(D) = X(D)\, F(D) \qquad (3.2)$$

Here $X(D) = \sum_{k=-\infty}^{\infty} x_n D^n$ and $Y(D) = \sum_{n=0}^{\infty} y_n D^n$

and the $\{x_n\}$ are assumed to be independent *m*-ary symbols, taking values of { -(m-1), -(m-3),... (m-3), (m-1) } with equal likelihood.

Fig. 3.2 shows a scheme for implementation of partial response signaling.

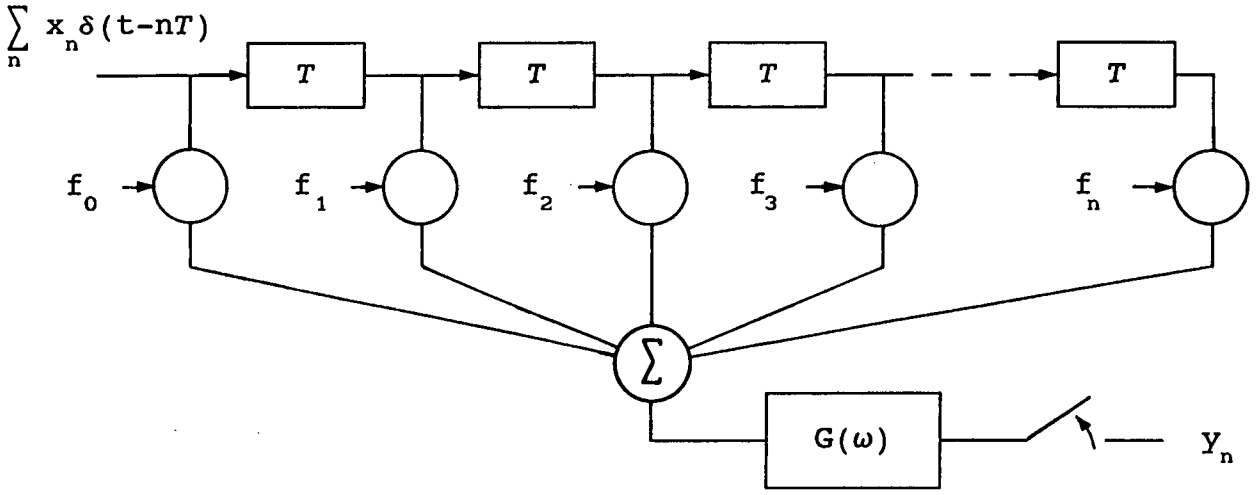


Fig. 3.2. Generalised partial response signaling scheme.

The system comprises a transversal filter (tapped delay line) with filter coefficients $\{f_n\}$, cascaded with a second filter whose frequency transfer function is $G(\omega)$. The transversal filter has a transfer function $F(\omega)$, where

$$F(\omega) = F(D)\Big|_{D = e^{-j\omega T}}$$

$$= \sum_{n=0}^{N-1} f_n \, e^{-j\omega nT} \qquad (3.3)$$

Here T is the symbol spacing, and $F(\omega)$ is assumed to be periodic. The system impulse response, h(t), will have the sample values $\{f_n\}$ if and only if $G(\omega)$ satisfies Nyquist's first criterion

$$\sum_{k=-\infty}^{\infty} G(\omega - \frac{2\pi k}{T}) = T \qquad (3.4)$$

Different choices of $G(\omega)$ are possible, and each yield a different system function, each with its own characteristics.

3.2 Choosing the PRS system polynomial

Because of the ease of using the delay operator (eg. in (3.1)), it is convenient to describe PRS systems by polynomials in D. If $G(\omega)$ is a high-order (approaching "brick-wall") lowpass filter, then the polynomial will accurately describe the transversal filter, $F(\omega)$, and any such filter and its characteristic polynomial can be seen by inspection to be equivalent.

The selection of different system polynomials lends itself to the tailoring of system performance. Factors to be considered are overall transmission bandwidth, location of spectral nulls and the number of output levels. Table 3.1 describes some popular PRS systems in terms of their system polynomials, F(D), and shows their respective frequency and impulse responses, and number of output levels, L [3.3].

3.2.1. Location of spectral nulls

In the design of PRS systems, it is convenient to choose $G(\omega)$ to be a high-order lowpass filter, since the system response is then defined purely by choice of $F(\omega)$. Because $G(\omega)$ will then have a discontinuity at the edge of its passband (at $\omega = \pi/T$), it is necessary that $F(\omega)$ have a transmission zero at the same place for $H(\omega)$ to be a continuous function.

The reason for this approach is that if $H(\omega)$ and its first K-1 derivatives are continuous, and its Kth is not, the function $|h(t)|$ decays asymptotically as $1/|t|^{K+1}$. Continuity of the function and its derivatives ensures that the signal energy in the

Table 3.1. Characteristics of minimum-bandwidth PRS systems

F(D)	H(ω) for $\|\omega\| \le \pi/T$	h(t)	L
$1 + D$	$2T \cos \frac{\omega}{2} T$	$\frac{4T^2}{\pi} \frac{\cos(\pi t/T)}{T^2 - 4t^2}$	2m-1
$1 - D$	$j2T \sin \frac{\omega}{2} T$	$-\frac{8T\ t}{\pi} \frac{\cos(\pi t/T)}{T^2 - 4t^2}$	2m-1
$1 - D^2$	$j2T \sin \omega T$	$-\frac{2T^2}{\pi} \frac{\sin(\pi t/T)}{T^2 - t^2}$	2m-1
$1 + 2D + D^2$	$4T \cos^2 \frac{\omega}{2} T$	$\frac{2T^3}{\pi t} \frac{\sin(\pi t/T)}{T^2 - t^2}$	4m-3

tails of h(t) remains small compared to its total energy. Hence, for minimum bandwidth systems (those in which $G(\omega)$ has this discontinuity), $F(\omega)$ should place a zero wherever discontinuities are likely to arise.

Since the first K-1 derivatives of a minimum bandwidth $H(\omega)$ will be continuous if and only if F(D) has (1+D) as a factor, it is desirable that such a factor be included in the system polynomial. As the number of such factors increases ie. as K increases, so the roll-off of $H(\omega)$ at $\omega = \pi/T$ decreases, and the design of practical system filters becomes easier. The trade-off is that, as K increases, so does the number of output levels, and hence the system noise performance is degraded. Also, more intersymbol interference is introduced, and so the system becomes more sensitive to timing errors.

In systems that are not restricted by bandlimiting, it is convenient to have a null at $\omega = \pi/T$, since this allows the insertion of a pilot tone for clock recovery at the receiver.

A similar argument applies to the introduction of a transmission zero at zero frequency. The system response rolls off at dc with introduction of (1-D) factors into the system polynomial, but the system noise performance is again degraded, as is the timing sensitivity. The major advantage is that after modulation (frequency shifting) of the baseband signal, there will be little signal power in the region of the carrier frequency — this allows a low-amplitude carrier to be inserted, for recovery at the receiver.

3.2.2. Number of output levels

A PRS system with M nonzero impulse samples (as defined in §3.1) and an m-ary input will have a number of output levels, L, where

$$M(m - 1) + 1 \leq L \leq m^M \qquad (3.5)$$

The minimum value occurs when the impulse values are the same, and the maximum occurs when no special relationship exists between sample values [3.3]. For example, Duobinary has M = 2. For binary inputs, $m = 2$, so the number of output levels is seen to be 3 or 4. Table 3.1 shows that this system has 3 output levels (L = 3).

It has been found that the introduction of (1+D) factors into the

system polynomial causes some of the output levels to merge, and so the SNR degradation due to the increased number of levels is not so severe.

3.3 Error Detection

PRS encoding creates a signal that contains more levels or states than the number of input levels. This causes an inherent redundancy that can be used to monitor system error performance. The mechanism will be illustrated in the following sections.

3.4 The Baseband Duobinary Technique

The Duobinary signaling scheme permits data transmission at the Nyquist rate, which is not possible with conventional pulse amplitude modulated (PAM) systems. Here "Duo-" signifies a doubling of the bit capacity of the conventional binary system.

This system is a particular case of the generalised system described in §3.1. The system polynomial (refer to Table 3.1) is given by

$$F(D) = 1 + D \tag{3.6}$$

This yields a system transfer function of

$$H(\omega) = (1 + e^{-j\omega T})_{LP}$$

$$= \begin{cases} 2\, e^{-j\omega T/2} \cos\frac{\omega T}{2} & |\omega| \le \frac{\pi}{T} \\ 0 & \text{Elsewhere} \end{cases} \tag{3.7}$$

and an impulse response of

$$h(t) = \frac{4 \cos(\pi(t-T/2)/T)}{\pi T \{1-4(t-T/2)^2/T^2\}} \tag{3.8}$$

The system frequency and impulse responses are shown in Fig. 3.3, and Fig. 3.4 shows the implementation of such a system.

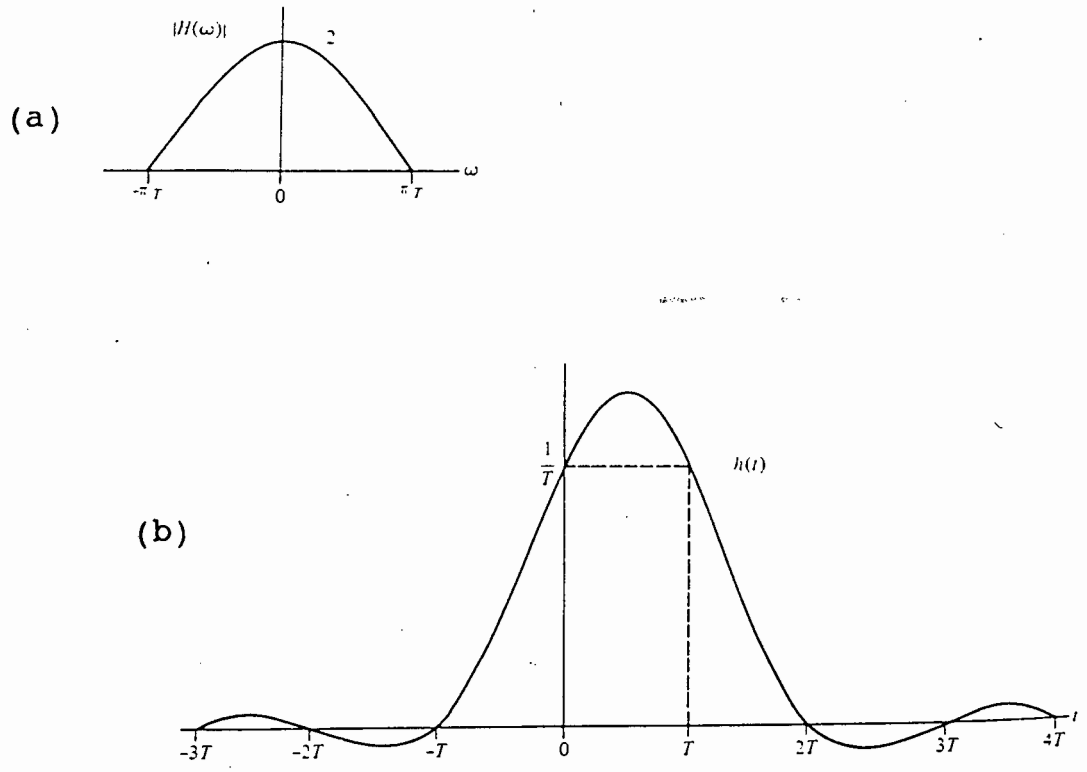


Fig. 3.3: (a) Magnitude of system frequency response
(b) Unit impulse response.

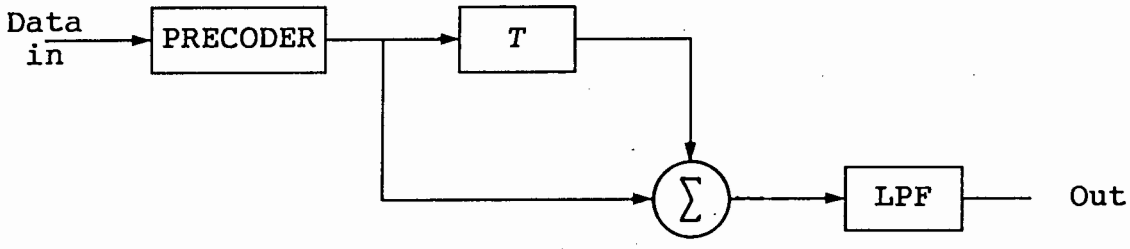


Fig. 3.4. Transversal filter implementation of Duobinary scheme.

A two-level impulse stream is passed through a digital transversal filter modelled by a delay element and a summer. The two-level input is therefore transformed into a three-level signal, thereby introducing correlation between successive digits, and hence a controlled amount of ISI. When cascaded with a lowpass filter, this system yields the frequency transfer function of (3.7).

Because each duobinary symbol contains interference from only the previous two-level bit, decoding at the receiver is merely a matter of relating pairs of symbols to one another in the correct fashion. Since the Duobinary filter adds two successive bits for each output symbol, the receiver could decode the data by subtracting from each received symbol the effect of the previous (decoded) bit.

If any symbol should be decoded incorrectly, that error will propagate throughout the remainder of the data stream. Lender [3.2] proposed a scheme, called "precoding," to eliminate this problem. By performing (modulo-2) the inverse function of the system polynomial before the transversal filter, it is possible to decode data at the receiver without reference to previous data bits, and error propagation does not occur. The precoder in Fig. 3.4, in effect, provides the function

$$b_k = (x_k - b_{k-1}) \bmod 2 \qquad (3.9)$$

where $\{x_k\}$ is the input binary sequence and b_k is the output of the precoder. The object is to remove, at the transmitter where all bits are known, the effect of the preceding bit on a symbol, without altering the spectral characteristics of the system. Because modulo-2 addition and subtraction are equivalent, (3.9) can be expressed as

$$b_k = (x_k + b_{k-1}) \bmod 2 \qquad (3.10)$$

$$= x_k \oplus b_{k-1}, \qquad (\text{"}\oplus\text{" the EXCLUSIVE-OR function}) \qquad (3.11)$$

and the precoder in Fig. 3.4 is therefore implemented by means of an EX-OR gate.

The signal from the transversal filter has three levels ie. $\{Y_k\} = \{ 0, 1, 2 \}$. These are level-shifted to yield $\{Y_k\} = \{ -1, 0, 1 \}$ before lowpass filtering.

In the case of multilevel duobinary (*m*-ary input instead of binary), (3.9) is generalised to

$$b_k = (a_k - b_{k-1}) \bmod m \qquad (3.12)$$

where a_k are the *m*-ary input symbols. (The term "duo-*m*-ary" will be used for the *m*-ary input-level duobinary scheme). The output Y_k's are derived in the same way as for the Duobinary case, yielding the same spectral shape.

Such a scheme will still exhibit a doubling of signaling speed compared to the conventional *m*-ary case. Table 3.1 shows that a two-level input duobinary system (*m*=2) creates three output levels. Similarly, a four-level input is transformed into seven output levels. Because duo-*m*-ary schemes have fewer signaling levels than *m*-ary PAM systems of equivalent speeds, they have a signal-to-noise advantage. This is true of many PRS schemes.

Duobinary encoding results in a set of rules governing level transitions:

1. No two *adjacent* signal samples may be of opposite sign.
2. If a nonzero sample is followed by one of opposite sign, there must be an odd number of zero (centre) samples between them.
3. If a nonzero sample is followed by another of the same sign, there must be an even number of zero samples between them.

Any violation indicates a sample error, resulting in a symbol error. By monitoring the sampled signal before decoding, such violations may be detected, and some form of error signal created. Such systems are described in [3.4] and [3.5, Ch. 7].

3.5 The Baseband Duoquaternary Technique

The duoquaternary technique is an extension of that used in Duobinary systems. The implementation of the system polynomial is

the same as for the duobinary case, (3.6), but the precoder is now specified, for four-level input data, by

$$b_k = (a_k - b_{k-1}) \bmod 4 \qquad (3.13)$$

Duo-*m*-ary systems yield 2*m*-1 output levels, and so this scheme has seven. The spectral properties are the same as for the duobinary case insofar as a halving of transmission bandwidth occurs relative to the quaternary system. Properties such as speed tolerance and eye openings are not relevant to the PRS model used in the modem described in this volume, because the PRS signal undergoes a transformation from the amplitude domain into the frequency/phase domain, and it is only at that point that such time-domain characteristics affect system performance. Rules concerning level transitions apply here as before, and so the possibilities for error detection remain.

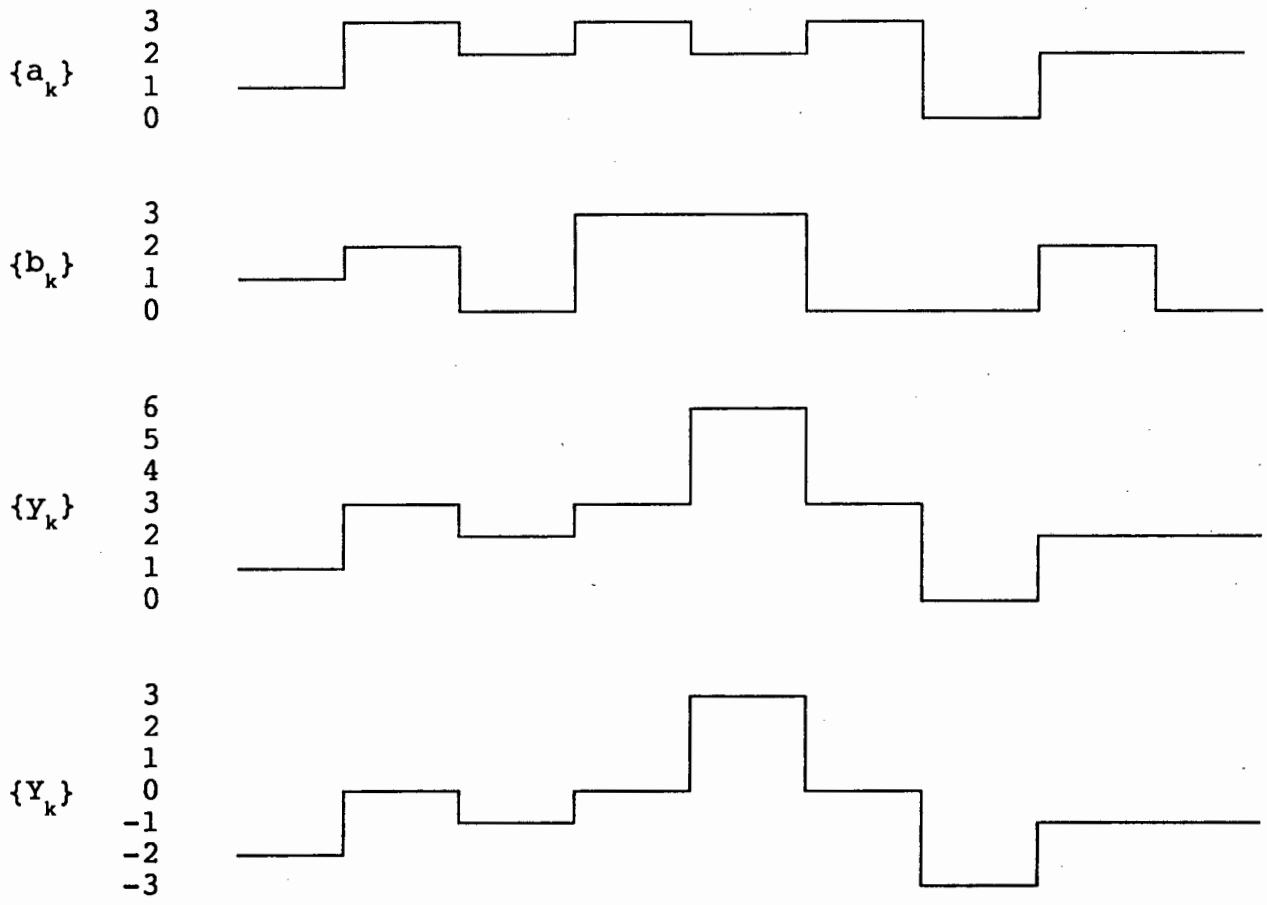


Fig. 3.5. Duoquaternary system signal waveforms.

Fig. 3.5 shows examples of the signals a_k, b_k, y_k and Y_k in a duoquaternary implementation.

Table 3.2 compares output symbol values, Y_k, with input symbols, a_k.

Table 3.2. Decoding table for Duoquaternary system

Y_k	a_k
3	2
2	1
1	0
0	3
-1	2
-2	1
-3	0

The transmitted data values, a'_k, may be decoded from the received Y'_k by the following rule:

$$a'_k = (Y'_k - 1)_{\text{mod } 4} \qquad (3.14)$$

Because of precoding, only one sample is required per decoded symbol.

Table 3.2 describes the relationship between original data bytes and their transmitted symbol values. Clearly, any given data value may be sent as one of two symbols, except for a data value of 3, which may only be sent as a symbol 0. Quite which of the two symbols is used depends on the previous data values and the initial conditions in the encoder, all of which are random. Because the original data are random, all transmitted values (except 0) are equiprobable, with the zero symbol having twice the probability of any other symbol. The zero symbol has a probability of transmission of 0.25, while all other symbols have probability 0.125.

The following chapter describes the application of partial-response signaling to frequency- and phase-modulated systems, specifically Correlative Phase-Shift Keying (CORPSK).

REFERENCES

[3.1] A. Lender, *"Faster Digital Communications with Duobinary Techniques,"* Electronics, March 22, 1963, pp. 61-65.

[3.2] A. Lender, *"The Duobinary Technique for High-speed Data Transmission,"* IEEE Trans. Commun. Electr., Vol. 82, May 1963, pp. 214-218.

[3.3] P. Kabal and S. Pasupathy, *"Partial-Response Signaling,"* IEEE Trans. Commun., Vol. COM-23, No. 9, September 1975, pp. 921-934.

[3.4] J. Gunn and J. Lombardi, *"Error Detection for Partial Response Systems,"* IEEE Trans. Commun. Technol., Vol. COM-17, No. 6, December 1969, pp. 734-737.

[3.5] K. Feher, *Digital Communications, Microwave Applications,* Prentice-Hall, 1981, Ch. 7.

CHAPTER 4
CORRELATIVE PHASE SHIFT KEYING

4.1 MODULATION DEFINITION

Correlative Phase Shift Keying [4.1], or CORPSK, has a signal of the form

$$s(t) = \mathrm{Cos}(\omega_c t + \phi(t)) \qquad (4.1)$$

where the signal behaviour is defined by manipulation of $\phi(t)$ according to an input sequence $\{a_m\} = \{a_0, a_1, \ldots a_m, \ldots\}$; successive phase shifts $\Delta\phi_m$ occur at $t = T_s$, T_s the symbol period, according to three criteria:

(1) the phase is continuous, and $\Delta\phi_m$ is given by

$$\Delta\phi_m = C_m \cdot 2\pi/n \qquad (4.2)$$

$$= \phi((m+1)T_s) - \phi(mT_s); \qquad (4.3)$$

C_m is the information-carrying term, and n is the number of phase states, n an integer.

(2) successive phase paths follow some correlation rule, so that each shift is determined by the current input level and that from at least one previous interval. Because of this type of correlative coding, there must be more possible phase shifts than input levels.

(3) the phase path must be smoothed. This ensures good spectral efficiency, since the derivative of the phase function is then continuous.

Many different forms of CORPSK exist, depending on the correlation rule whereby C_m in criterion (1) is derived. The notation CORPSK(M-N, F(D)) is used, where F(D) is the PRS system polynomial of the type described in Chapter 3. M is the number of input levels, and N is the number of possible phase shifts. For example, CORPSK(4-7, 1+D) represents a four-level input system with seven possible output phase shifts, following the Duobinary rule. The

number of phase states, n, is four.

The basic CORPSK modulator consists of a PRS encoder with precoder, followed by a lowpass premodulation filter and angle modulator (Fig. 4.1).

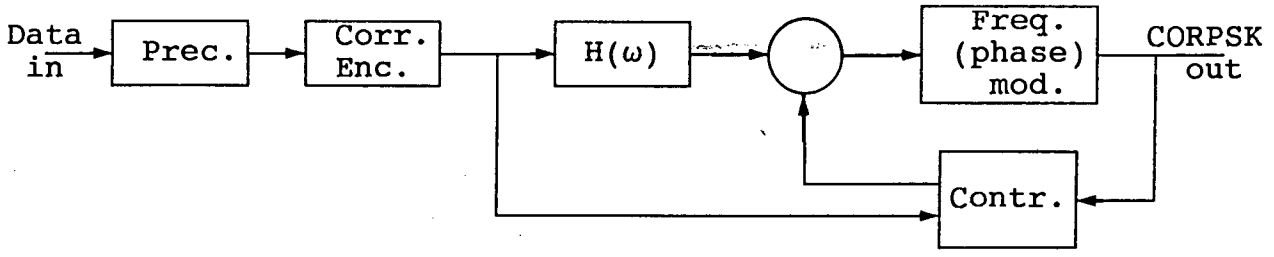


Fig. 4.1. Basic CORPSK modulator structure

This is equivalent to a baseband PRS system followed by angle modulation. The premodulation filter ensures that the angle modulator follows a smooth phase path. If frequency modulation is used, this is equivalent to a phase modulator whose input is the integral of the PRS signal. This is because frequency modulation has an inherent integrating effect.

The precoder and PRS encoder are identical to those used in the baseband PRS case. The CORPSK process is just like Continuous-phase Frequency Shift Keying (CPFSK) in that the baseband signal to an angle modulator is a continuous, shaped function — in CORPSK, the shaping is by a baseband partial response filter.

By trigonometric identities, (4.1) can be modified to be

$$s(t) = \mathrm{Cos}(\omega_c t)\mathrm{Cos}(\phi(t)) - \mathrm{Sin}(\omega_c t)\mathrm{Sin}(\phi(t)) \qquad (4.4)$$

and s(t) can therefore be represented by an in-phase and a quadrature carrier modulated by the cosine and the sine, respectively, of the phase function, $\phi(t)$.

The input sequence $\{a_m\} = \{a_0, a_1, \ldots a_m, \ldots\}$ can be written, using the delay operator D, as

$$a(D) = a_0 + a_1 D + a_2 D^2 + a_3 D^3 + \ldots \quad (4.5)$$

The output of the correlative encoder is given by

$$c(D) = b(D) \cdot F(D) \quad (4.6)$$

where b(D) is the precoder output and F(D) is the encoding rule.

Referring to Fig. 4.1, the lowpass filter $H(\omega)$ ensures that the phase is not just smoothed, but that it passes through the stationary phase points at each symbol transition instant. In order to achieve this in conjunction with a phase modulator, it must satisfy Nyquist's first criterion for zero ISI:

$$n_I(kT_s) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0, \text{ k an integer} \end{cases} \quad (4.7)$$

Here $n_I(t)$ is the impulse response of the filter that satisfies this criterion. If the input to the filter is described as

$$c(t) = \sum_m c_m \delta(t - mT_s) \quad (4.8)$$

then the signal phase, $\phi_p(t)$, will be

$$\phi_p(t) = \frac{2\pi}{n} \cdot c(t) * n_I(t) \quad (4.9)$$

If, however, a frequency modulator is used instead, Nyquist's third criterion must be satisfied —

$$\int_{(2k-1)(T_s/2)}^{(2k+1)(T_s/2)} n_{III}(t)\, dt = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0, \text{ k an integer} \end{cases} \quad (4.10)$$

The signal phase from the frequency modulator is then

$$\phi_f(t) = \frac{2\pi}{nT_s} \int_{-\infty}^{t} c(t) * n_{III}(t)\, dt \quad (4.11)$$

where $2\pi/nT_s$ is the transfer constant of the modulator.

The duality that arises between (4.9 and (4.11) is because of the integrating effect of the frequency modulator. It is of little practical consequence, however, since encoding with F(D) followed by a phase modulator is equivalent to encoding with F(D)(1-D) followed by a frequency modulator, except for a time delay of $T_s/2$. The differentiating effect of the (1-D) factor cancels the frequency modulator integration.

Equations (4.9) and (4.11) show how CORPSK produces PRS signals identical to those created in PAM systems, but transfers them into the phase domain.

4.2 SYSTEM SPECTRA

Muilwijk analysed the performance of a number of signals in the CORPSK class. Fig. 4.2 shows the spectra of three systems, CORPSK(2-3, 1+D), CORPSK(2-3, $1-D^2$) and CORPSK(2-5, $(1+D)^2$) (TFM), compared to those of Minimum-shift Keying (MSK) and Duobinary-encoded MSK.

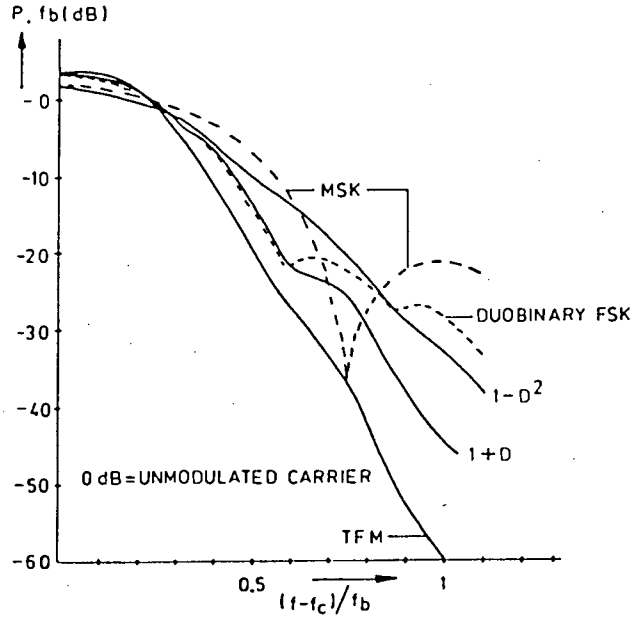


Fig. 4.2. Spectra of CORPSK(2-3, 1+D), CORPSK(2-3, $1-D^2$), CORPSK(2-5, $(1+D)^2$) (TFM), MSK and Duobinary MSK.

Comparing the spectra of MSK and Duobinary MSK, the effect of correlative encoding is a reduction of out-of-band signal levels and size of main lobe, and hence a reduction of necessary bandwidth. Duobinary MSK is similar to CORPSK(2-3, 1+D), except that the frequency-modulating signal is made up of square pulses, and is not smoothed, as in the CORPSK case. The effect of the premodulation filter, therefore, is a lessening of out-of-band signal level.

Fig. 4.3 compares the spectra of CORPSK(4-5), which employs a nonlinear encoding rule, CORPSK(4-7, 1+D) and unfiltered, or constant-envelope, QPSK for reference.

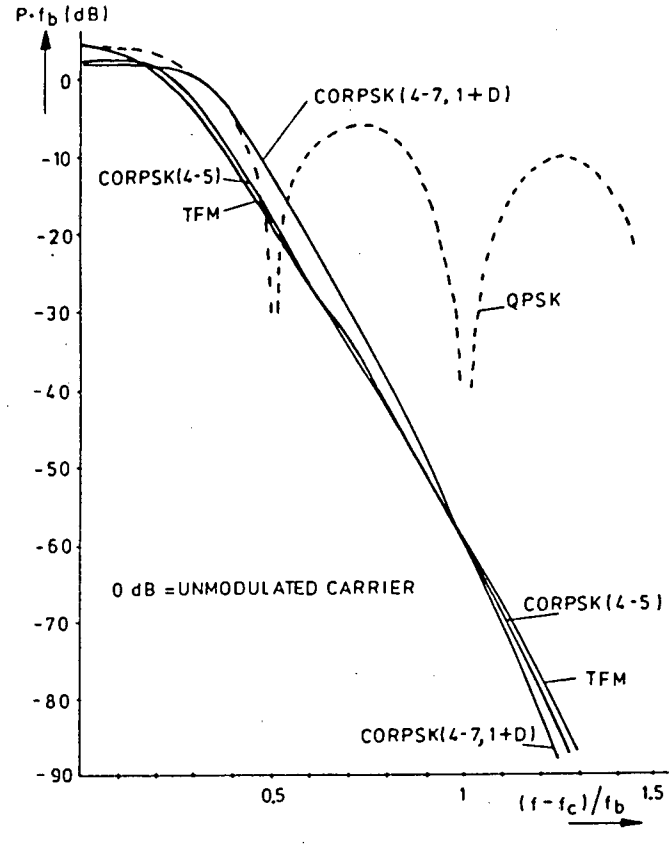


Fig. 4.3. Spectra of CORPSK(4-5), CORPSK(4-7, 1+D) and QPSK.

CORPSK(4-7, 1+D) has a slightly wider main lobe than CORPSK(4-5). However, compared to QPSK, they both have far lower out-of-band signal levels.

4.3 DEMODULATION

4.3.1 Synchronization

A coherent correlation receiver for CORPSK is the optimum. Symbol-by-symbol sampling, however, yields a more practical, but suboptimum, solution.

In either case, secure recovery of the carrier is required in order to demodulate the received signal synchronously. CORPSK is an angle modulated system with n phase states, and so carrier recovery is achieved by processing the received signal with a nonlinearity of order n, and extracting the resulting discrete frequency component that arises at n times the carrier frequency. This signal is then divided in frequency by n to give a frequency- and phase-stable carrier reference. The initial n-fold phase ambiguity that results is overcome by encoding the signal differentially at the transmitter.

The carrier regeneration process is successful because the carrier phase remains in the region of one of the n phase states for some portion of time when the transmitted data require no actual phase change. This will occur when the PRS input to the modulator is zero over one symbol period. Upon frequency multiplication by n, all zero phase shift portions of the signal add in-phase and create nonzero average power at $n \cdot \omega_c$, ω_c the carrier frequency. (This is similar to the approach used for carrier recovery in QPSK and MSK systems [4.2]).

In addition, all other phase shifts that occur become multiples of $2 \cdot \pi$ when frequency-multiplied by n. All equal-valued phase shifts then add in-phase to give discrete frequency components situated symmetrically about $n \cdot \omega_c$, at a distance proportional to the symbol rate.

4.3.2 Practical Detector Model

Symbol-by-symbol detection offers the most practical implementation of a CORPSK receiver but, as is shown in Chapter 5, introduces a signal-to-noise degradation of approximately 3 dB relative to the ideal correlation receiver. The baseband analysis model shown in Fig. 4.4 is adopted for the simulations performed in Chapter 5 and the tests reported in Chapter 7.

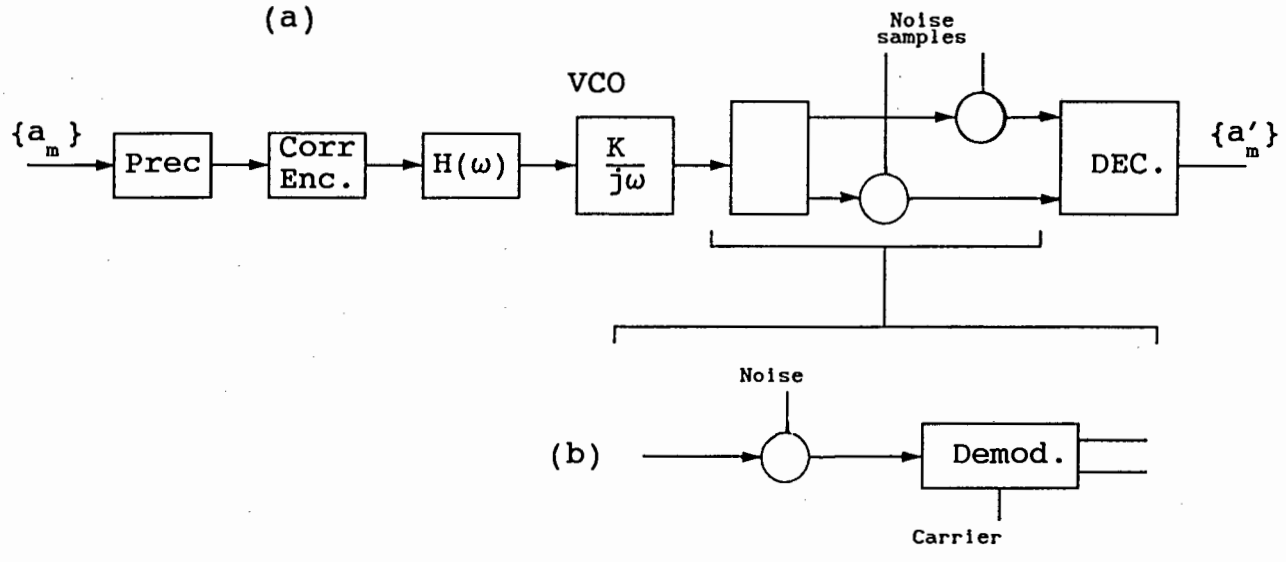


Fig. 4.4(a) Analysis model for CORPSK systems (b) equivalent.

An input sequence $\{a_m\}$ is generated and precoded, to yield the sequence $\{b_m\}$. The PRS rule (for example, Duobinary) is encoded to give $\{c_m\}$. This is then smoothed by a premodulation filter which satisfies Nyquist's first criterion for zero ISI. The frequency modulator is modelled as an integrator. The in-phase and quadrature signal components are derived by simple trigonometric functions. Gaussian channel noise is added to each signal independently, and the resulting waveforms sampled and decoded to give an estimate of the transmitted symbol value.

Note that because the modulating signal is integrated by the frequency modulator, successive values of $\{c_m\}$, the PRS symbol values, are encoded differentially onto the phase of the transmitted carrier. This requires that successive phase samples be compared in the decoder to give the PRS value, which is then

decoded into data words by a simple mapping (Table 3.2).

The results reported in [4.1] for system error rates are all analysed in terms of E_b/N_0, the ratio of bit energy to noise density. This approach is only valid when the analysis is applied to the correlation receiver, which is just an implementation of the matched filter [4.3, p. 505]. In the case of symbol-by-symbol detectors, comparisons must be made on the basis of signal-to-noise ratio (SNR) at the input of the sampler. Equation (4.4) shows that carrier-to-noise ratio (CNR) may be used instead of SNR, since the in-phase and quadrature carriers are scaled directly by the values of the in-phase and quadrature signals, respectively.

Values for E_b/N_0 may be related to those of CNR according to

$$\frac{E_b}{N_0} = \frac{C}{N}\cdot\frac{W}{R} \qquad (4.12)$$

where W is the baseband signal bandwidth, and R the bit rate [4.4], [4.5]. For CORPSK(4-7, 1+D) the data rate and the signal bandwidth are equal, and therefore E_b/N_0 and CNR are equal. Nevertheless, all comparisons will be in terms of CNR from now on.

The signal space diagram for CORPSK(4-7, 1+D) is shown in Fig. 4.5.

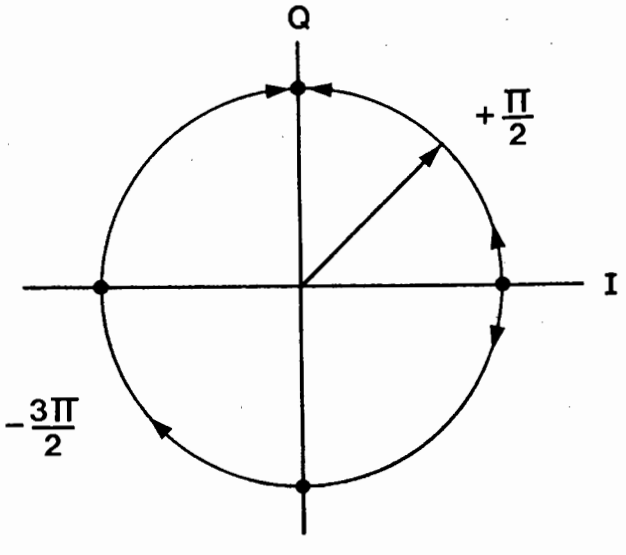


Fig. 4.5. Signal space for CORPSK(4-7, 1+D).

The signal phasor remains on the unit circle at all times. At sampling instants, it may be located only at one of the four stationary phase points. Although the symbol-to-symbol phase shifts are not equiprobable (since they are derived by the Duoquaternary rule), there is no absolute phase coherence in the signal, and so, at the receiver, the signal phasor may lie on any stationary phase point with equal probability. Each of the in-phase and quadrature signal vectors will therefore have value +V (probability of 0.25), -V (probability of 0.25) or 0 (probability of 0.5). The average signal power at the input of each channel's sampler is then

$$P_{ave} = P_0 \cdot V_0^2 + P_1 \cdot V_1^2 + P_2 \cdot V_2^2 + P_3 \cdot V_3^2 \qquad (4.13)$$

where P_i is the probability of the phasor being at phase point i, ie. a probability of 0.25. So, for the in-phase channel, with V_i = V volts,

$$P_I = P_0 \cdot V^2 + P_1 \cdot 0^2 + P_2 \cdot V^2 + P_3 \cdot 0^2 \qquad (4.13)$$

$$= 0.5 \cdot V^2 \qquad (4.14)$$

Similarly for the quadrature channel,

$$P_Q = P_0 \cdot 0^2 + P_1 \cdot V^2 + P_2 \cdot 0^2 + P_3 \cdot V^2 \qquad (4.15)$$

$$= 0.5 \cdot V^2 \qquad (4.16)$$

This means that the RMS voltage at the input of each sampler is

$$V_S = \frac{V}{\sqrt{2}} \qquad (4.17)$$

The detection threshold is a phase value situated between adjacent stationary phase points — in the case of CORPSK(4-7, 1+D) it lies at odd multiples of $\pi/4$. The receiver converts the received in-phase and quadrature signals into a phase value and a decision

is made as to which phase point is represented by the coordinates. The effect of channel noise is to turn each stationary point into a region or cluster of possible sample points. Given that the relevant measure of channel noise power is then the variance of the noise, on a line joining adjacent stationary points, the in-phase and quadrature noise components are given, respectively, by

$$\sigma_{nI}=\frac{\sigma_n}{\sqrt{2}} \qquad (4.18a)$$

and

$$\sigma_{nQ}=\frac{\sigma_n}{\sqrt{2}} \qquad (4.18b)$$

For normalized baseband signals $\cos(\phi(t))$ and $\sin(\phi(t))$ (ie. V = 1),

$$CNR=20.\log_{10}\left(\frac{V_s}{\sigma_n}\right) \qquad (4.19)$$

CNR in [dB]. Therefore,

$$\frac{V_s}{\sigma_n} = 10^{CNR/20} \qquad (4.20)$$

and

$$\sigma_n = V_s \cdot 10^{-CNR/20} \qquad (4.21)$$

$$= 0.707 \cdot 10^{-CNR/20} \qquad (4.22)$$

From (4.18) and (4.19),

$$\sigma_{nI} = 0.5 \cdot 10^{-CNR/20} \qquad (4.23)$$

and

$$\sigma_{nq} = 0.5 \cdot 10^{-CNR/20} \qquad (4.24)$$

These are then the values used for scaled noise voltages added to

the sampler inputs in the noise performance simulations of Chapter 5, as well as the error rate measurements of Chapter 7.

4.3.3 Error Detection

CORPSK employs correlative coding to yield a PRS-shaped phase trajectory in the transmitted carrier. As mentioned in Chapter 3, such coding schemes exhibit a natural coding redundancy, which can be used for the detection, and sometimes even the correction, of errors in the received data. Chapter 3.4 outlines the coding rules whereby errors may be detected in conventional PRS systems. The equivalent rules in CORPSK(4-7, 1+D) are:

1. No two *adjacent* phase shifts may be of opposite sign.
2. If a nonzero phase shift is followed by another of opposite sign, there must be an odd number of zero phase shifts between them.
3. If a nonzero phase shift is followed by another of the same sign, there must be an even number of zero phase shifts between them.

In the symbol-by-symbol detection process, only the start- and end-points of the phase shifts are used to decode data. For a given input data word, the two possible Duoquaternary phase shifts are complementary — they traverse the unit circle in opposite directions, but reach the same end-point. The direction of the phase transition contains the inherent redundancy of correlative coding. Only if accompanied by an efficient frequency discriminator scheme (to detect transition direction) is symbol-by-symbol detection able to realize the use of this coding redundancy.

REFERENCES

[4.1] D. Muilwijk, *"Correlative Phase Shift Keying - A Class of Constant Envelope Modulation Techniques,"* IEEE Trans. Commun., Vol. COM-29, No. 3, March 1981, pp. 226-236.

[4.2] R. de Buda, *"Coherent Demodulation of FSK with Low Deviation Ratio,"* IEEE Trans. Commun., vol. COM-20, No. 3,

June 1972, pp. 429-435.

[4.3] B. P. Lathi, *Modern Digital and Analog Communication Systems,* Holt, Rinehart and Winston, 1983.

[4.4] K. Feher and the Engineers of Hewlett-Packard, *Telecommunications Measurements, Analysis, and Instrumentation,* Prentice-Hall, Inc., Englewood Cliffs, N. J.

[4.5] B. Sklar, *"A structured Overview of Digital Communications — A Tutorial Review — Part 1,"* IEEE Communications Magazine, August 1983, pp. 4-17.

CHAPTER 5

CORPSK SIMULATION RESULTS

Performance of the system CORPSK(4-7, 1+D) was simulated both to verify and to extend the results given by Muilwijk [5.1], who reported only on the output spectrum and error performance. All of the following simulation results were produced on MathCad 2.50, (Appendix C) except for the noise error simulation, which was by means of a program written in Turbo Pascal (Appendix D). The MathCad listings are commented, so no further explanation of method will be given here.

5.1 Baseband eye diagrams

Fig. 5.1 shows the eye diagram for one channel (in-phase or quadrature).

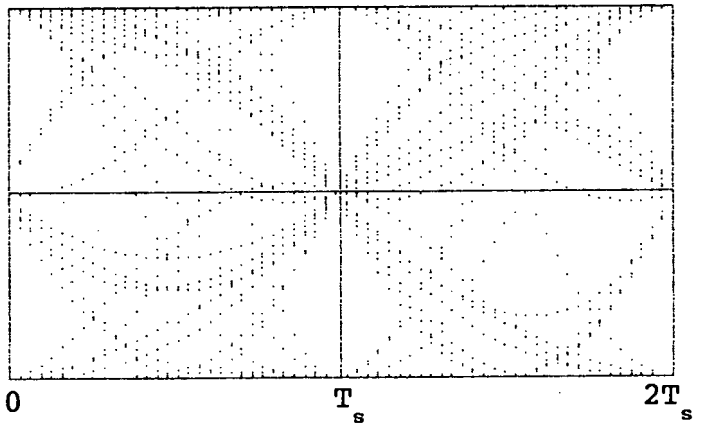


Fig. 5.1. Eye diagram for one channel

This result forms a useful standard by which to assess the performance of the final digital implementation. Such comparisons are made on the basis of degradations in vertical and horizontal eye size.

5.1.1 Vertical eye opening (VEO)

At each sampling instant, the phase of the CORPSK(4-7, 1+D) signal can only be a multiple of $\pi/2$. Therefore, since each channel

represents either the (scaled) cosine or sine of the phase angle, the sampled value can only be 0 or $\pm c$, c real (for purposes of the simulation, $c = 1$). Both the upper and lower eyes are at their maximum opening. Any degradation in VEO in the practical implementation would be from hardware imperfections (resulting in uncontrolled amounts of intersymbol interference), or from channel noise. This degradation is expressed, in dB, as a ratio between the practical VEO and the theoretical.

5.1.2 Horizontal eye opening (HEO)

The horizontal eye width (at the widest point) is estimated to be 0.4 of a symbol period. The HEO value can be degraded, in practice, by the effect of imperfect transmission filters, and the effect of channel group delay (which includes a contribution from hardware). The greater the HEO, the greater the system's tolerance to sample timing errors at the detection stage.

5.2 Constellation diagram

Fig. 5.2(a) shows the X-Y diagram, and 5.2(b) the constellation diagram of the signal.

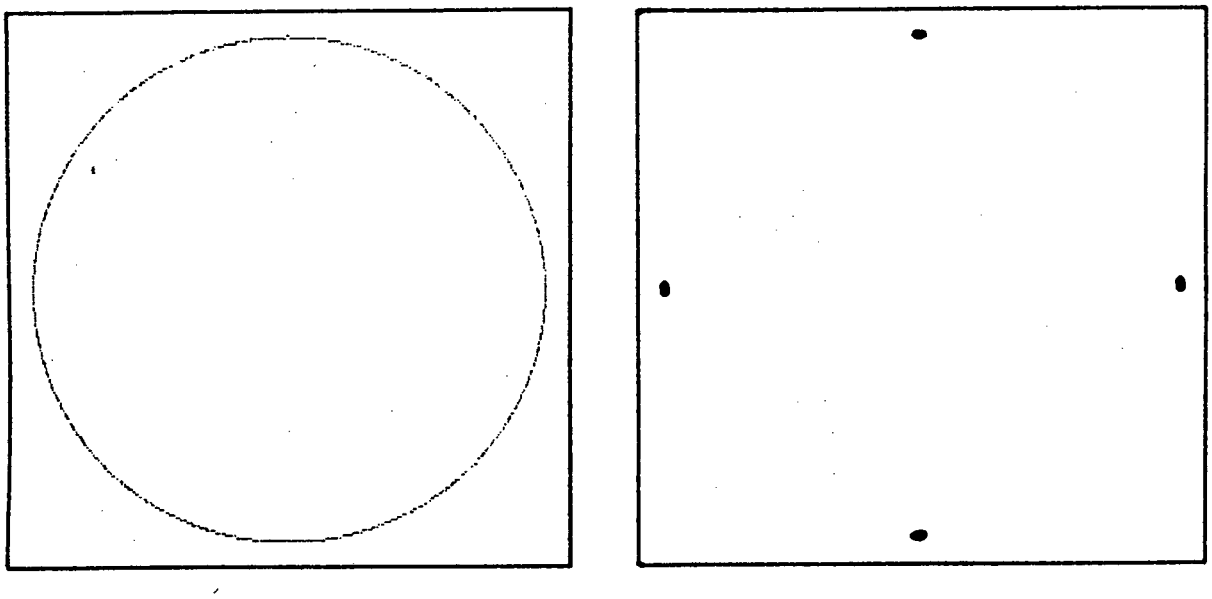


Fig. 5.2. (a) X-Y diagram and (b) constellation diagram for CORPSK(4-7, 1+D)

The circular X-Y plot (Lissajous figure) indicates that CORPSK has a constant envelope (the phase of the CORPSK carrier is seen as a rotating phasor of unit length). In practice, any fluctuations induced in the envelope will be indicated by a broadening of the circular band. Again, this degradation may be expressed as a dB ratio relative to the average carrier envelope.

The constellation diagram (Fig. 5.2.(b)) shows the four phase states. The effect of channel noise is that each of the four dots becomes a "cluster" whose diameter is related to the noise power. The decision distance in the signal space is then no longer the distance between the four individual dots, but between some region within each of the clusters, and so the noise immunity is impaired. No cluster has an "edge" because noise with an infinite crest factor can theoretically generate infinitely large values, whatever the variance.

5.3 Signal spectra

The simulated baseband spectrum (Fig. 5.3(a).) corresponds well with the results of Muilwijk [5.1], shown in Chapter 4.

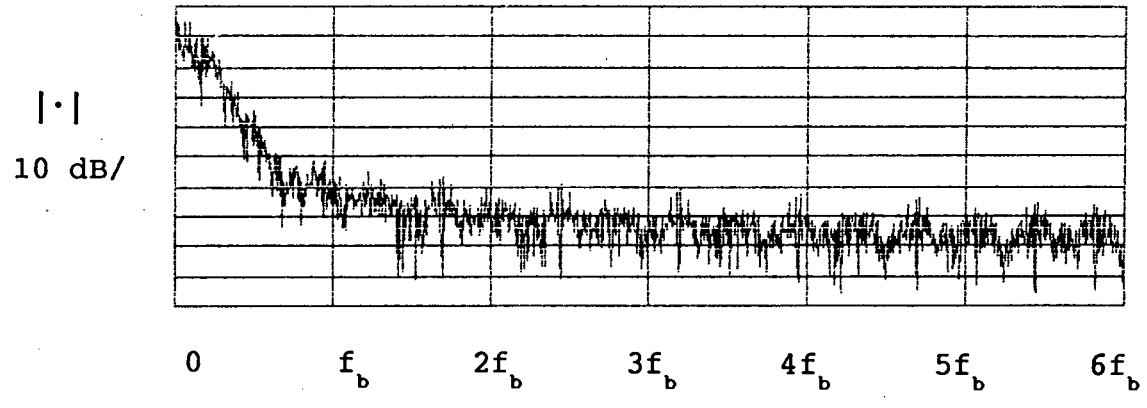


Fig. 5.3(a). Simulated normalised CORPSK(4-7,1+D) signal spectrum

There are no periodic sidelobes - this is the effect of pulse shaping before the angle modulator (the frequency pulse therefore has no discontinuities). The final bandpass spectrum (in-phase and quadrature modulated carriers added) is shown in

Fig. 5.3(b).

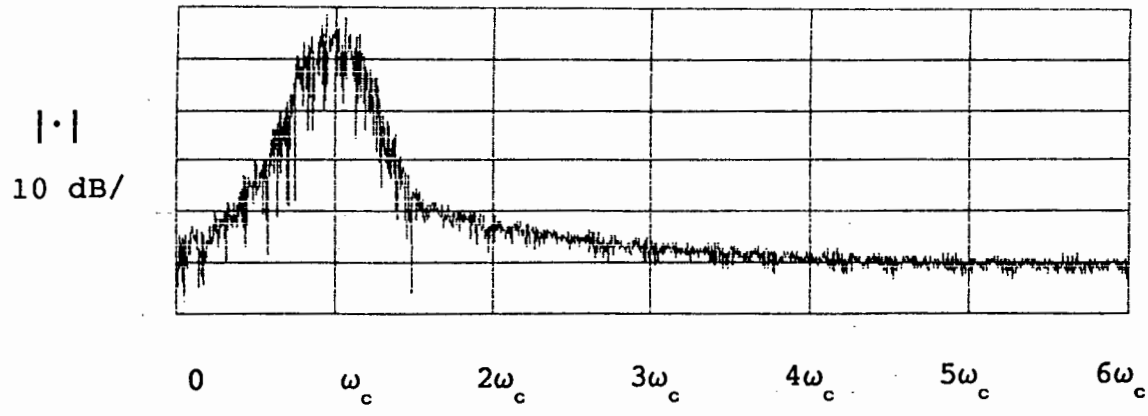


Fig. 5.3(b). Simulated bandpass spectrum for CORPSK(4-7, 1+D)

5.4 Carrier regeneration and symbol timing recovery

Simulations were performed to investigate the use of circuit nonlinearities for the regeneration of a carrier signal at the receiver.

The standard approach is to use a nonlinearity of order n, where n is the number of signal phase states located symmetrically in the signal space [5.2]. In our case, n is equal to four. A second-order nonlinearity is inadequate for this purpose, as Fig. 5.4(a) shows. Only the second-order signal harmonic appears, with no discernible carrier information.

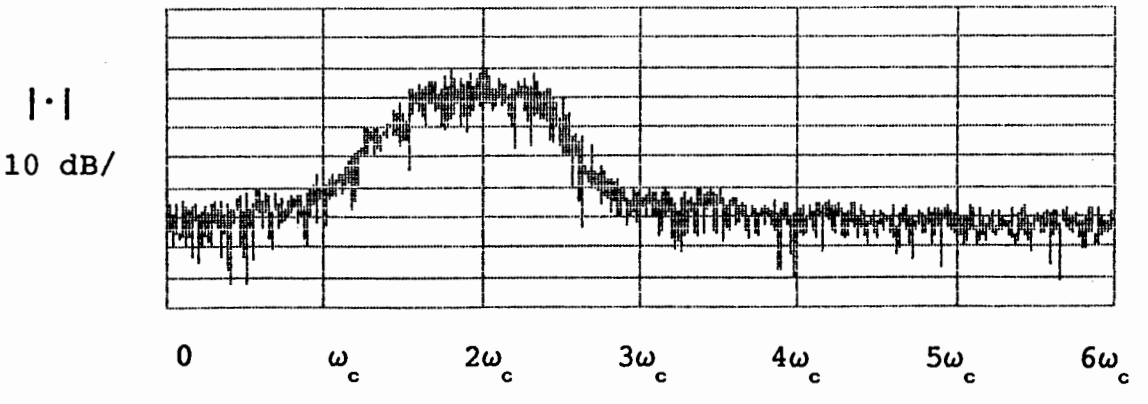


Fig. 5.4(a). Second-order spectrum for CORPSK(4-7, 1+D)

Use of a fourth-order nonlinearity yields good results (Fig.

5.4(b)). In practice, this would be implemented with two balanced mixers cascaded, each acting as a squarer. A simpler approach is to use a full-wave rectifier, which will yield all orders of signal. Simulation of this method (taking the absolute value of the signal) yields identical results over the frequency range of interest, with no discernible forms of aliasing.

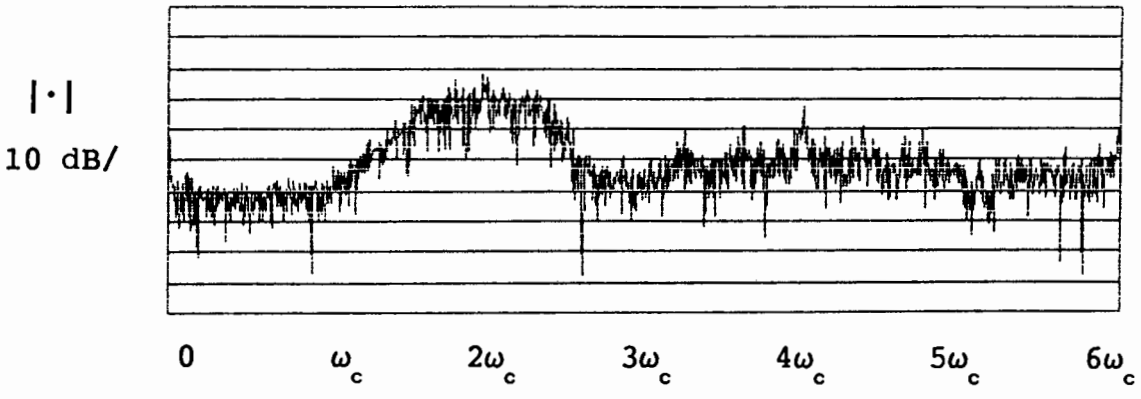


Fig. 5.4(b). Fourth-order spectrum for CORPSK(4-7, 1+D)

Symbol-timing components are located symmetrically about $4\omega_c$; their frequency separation is proportional to the symbol rate. A symbol-rate clock may be created by locking onto two components (or one component and the fourth-order carrier) and multiplying them to give a difference frequency. It can be seen from Fig. 5.4(b) that the symbol-timing components are small in amplitude, and that there appears to be some phase noise around them. These two factors may, in practice, impair the success of this symbol-timing recovery approach.

5.5 Effects of hard limiting on the output spectrum

Hard limiting is often used in, for example, satellite transponders as signal conditioners prior to power amplification, to prevent large amplitude swings. The usual approach is to follow the hard limiter by a bandpass filter, tuned to the carrier frequency concerned. This configuration is referred to as a "bandpass limiter" (BPL). Any modulation schemes used in such systems must obviously exhibit as high an immunity to such operations as possible. Signals with nonconstant envelopes, such

as filtered QPSK or MSK (filtered to reduce out-of-band signal levels), usually experience spectral spreading as well as coupling between in-phase and quadrature channels.

Appendix A describes the effect of a bandpass limiter on the system carrier-to-noise ratio and the mechanisms of spectral spreading and quadrature crosstalk in QPSK and MSK. Constant-envelope systems such as CORPSK are more immune to these last two effects, and can therefore enjoy the benefits of using bandpass limiting.

Fig. 5.5 shows a hardlimited CORPSK spectrum (compare to Fig. 5.3(b)). No sidelobes have been generated, but the out-of band noise level has increased remarkably. This is because the simulation modelled a pure hardlimiter — the Heaviside step function was used, and this is just a mathematical zero-crossing detector. In practice, a bandpass limiter (BPL) would be used. The effect would be to reduce the out-of-band signal level, leaving the still well-defined main spectral lobe.

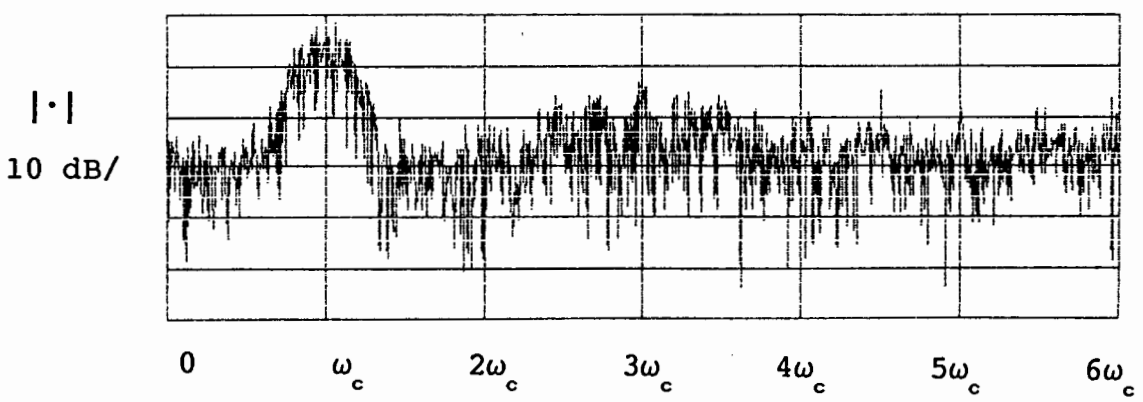


Fig. 5.5. Hardlimited CORPSK(4-7, 1+D) spectrum

5.6 Error performance in the presence of noise

The effect of Gaussian channel noise on the performance of CORPSK(4-7, 1+D) was simulated with a Pascal program (listed in Appendix D). Symbol-by-symbol sampling was assumed, and the results were used to predict the performance of the final hardware implementation. Fig. 5.6 is the program flowchart.

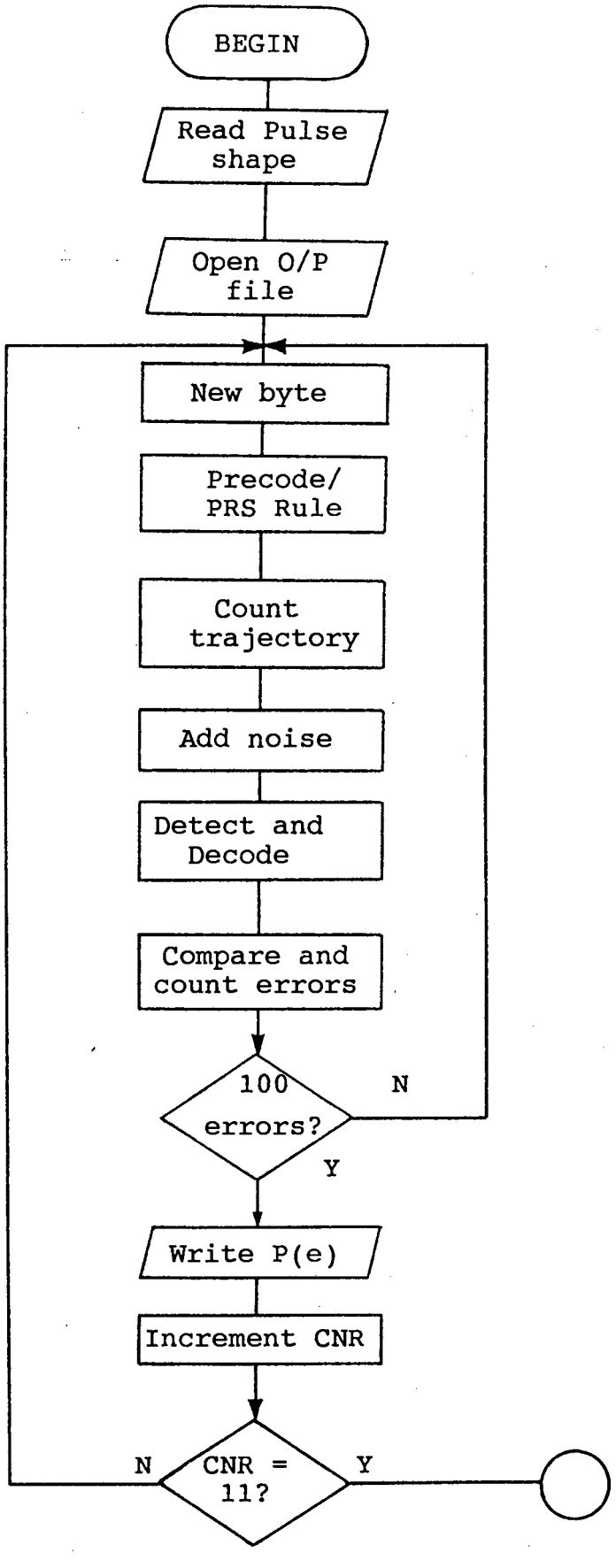


Fig. 5.6. Noise simulation program flowchart

The phase pulse sample values are read in from disk and a data file is opened for program output. Simulations begin for a SNR value of 1 dB. A random four-level word is generated and precoding is performed. The Duoquaternary function is performed on the precoder output, yielding a seven-level word. Based on this value, the previous two values, and the phase state three symbols previously, a phase trajectory is calculated using the phase pulse, scaled accordingly. The in-phase and quadrature channels are created by taking the cosine and the sine, respectively, of the phase function. Gaussian-distributed noise values are added to these signals — separate values are used for each channel to ensure that there be no correlation between in-phase and quadrature channel noise.

The I and Q channels are then sampled separately and these "Cartesian" coordinates converted into a phase angle. A decision is then made as to which of the four signal space points is represented by this noise-corrupted phase value. Fig. 5.7 illustrates.

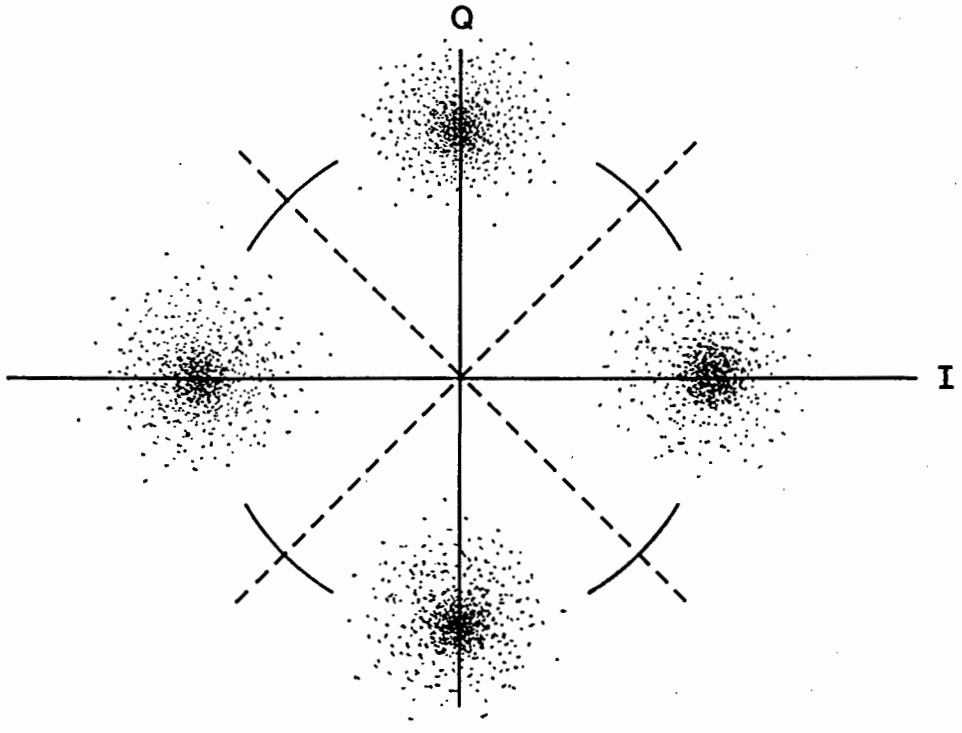


Fig. 5.7. Simulation signal space diagram.

Successive phase samples are then compared to give the original Duoquaternary symbol value and the original data byte is decoded from this.

Finally, the received data byte is compared with the transmitted value. A running total is kept of the number of mismatches and this is used, with the transmitted symbol count, to compute a symbol error rate, which is continually updated on the screen. Once one hundred errors have been registered, the symbol error rate is written to a disk file. The channel SNR value is incremented and simulation begins again.

It should be noted that for low channel SNR, the simulation size is approximately 10^3, while at higher SNR, it is of the order of 10^5 or 10^6. For each value of CNR, 100 error events are registered — this ensures an error rate statistic whose variance is sufficiently small [5.1], [5.5, p. 139].

Fig. 5.8 shows a plot of the results of the simulation, against the theoretical values yielded by Muilwijk's Viterbi receiver simulation [5.1].

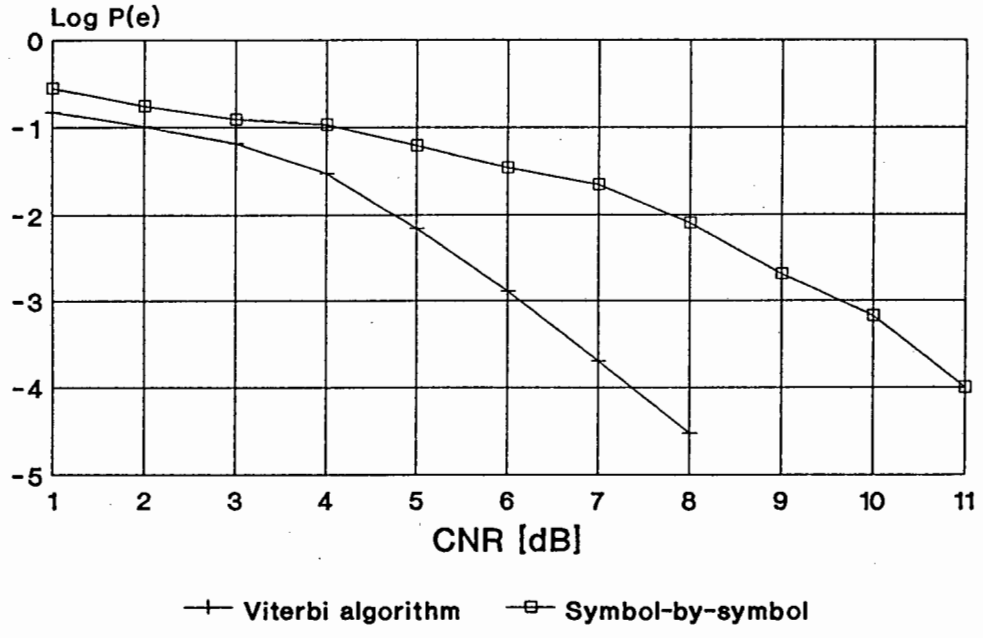


Fig. 5.8. Error rates for CORPSK(4-7, 1+D) employing (a) the Viterbi algorithm, and (b) symbol-by-symbol detection.

As predicted, because symbol-by-symbol detection is a suboptimum method, it yields higher error rates than the Viterbi, or Maximum-likelihood (MLD), detection method. It is well-known that the MLD approach yields a coding gain of 2.5 to 3 dB over the symbol-by-symbol approach in conventional PRS and PAM systems [5.3], [5.4]. It is therefore not surprising that Muilwijk's results are only 2 to 3.5 dB better than our simulation, since coherent demodulation of CORPSK yields a set of received signals of the PAM type.

A possible source of error in the noise simulation relates to the finite crest factor of the computer-generated noise. A Gaussian noise value is created by summing twelve random numbers (each in the range from 0 to 1) and subtracting 6 — this yields a Gaussian value of zero mean and unit variance. The rms value is 1, but the peak is limited to 6. The crest factor (ratio of peak to rms) is then 6, or 15.5 dB. This value is then scaled appropriately for the particular CNR in question. Whatever the final rms noise value, the crest factor remains the same. Feher [5.5] calculated the effect of finite crest factor noise on error probabilities for 4 PSK, 8 PSK and 16 PSK. The 4 PSK system provides a good comparison since CORPSK(4-7, 1+D) uses four phase states. His results showed that 15 dB crest factor noise provided valid error rate results down to 10^{-8}. The noise simulation only reaches 10^{-4}.

REFERENCES

[5.1] D. Muilwijk, *"Correlative Phase Shift Keying — A Class of Constant Envelope Modulation Techniques,"* IEEE Trans. Commun., Vol. COM-29, No. 3, March 1981, pp. 226-236.

[5.2] R. Gagliardi, *Satellite Communications,* Van Nostrand Reinhold Company, 1984, Appendix B.3.

[5.3] H. Kobayashi, *"Correlative Level Coding and Maximum-likelihood Decoding,"* IEEE Trans. Inform. Theory, Vol. IT-17, No. 5, September 1971, pp. 586-594.

[5.4] H. Kobayashi, *"A Survey of Coding Schemes for Transmission*

or Recording of Digital Data," IEEE Trans. Commun. Technol., Vol. COM-19, No. 6, December 1971, pp. 1087-1100.

[5.5] Feher, K. and the Engineers of Hewlett-Packard, *Telecommunications Measurements, Analysis, and Instrumentation*, Prentice-Hall, Englewood Cliffs, N.J., 1987.

CHAPTER 6
CORPSK MODEM DESIGN

The hardware described in this chapter does not implement the CORPSK(4-7, 1+D) modem fully. All the necessary modules required for baseband testing were completed, as were all the bandpass-level circuits. Of the synchronization subsystem, only the carrier recovery section was completed. The recovery of symbol timing, although simple in approach, proved unsuccessful using only simple phase-locked loop techniques. This aspect of symbol-timing recovery is covered in Chapter 7.

All circuit diagrams referred to in the text are contained in Appendix B, and will be referenced by their page numbers there.

6.1 System Description

The CORPSK(4-7, 1+D) modulator is shown in block form in Fig. 6.1(a), and the demodulator in Fig. 6.1(b).

The transmitter comprises a partial-response (PRS) encoder followed by a phase-trajectory shaping module, which creates an in-phase and quadrature symbol channel. Each channel contains a digital-to-analog converter, lowpass filter and modulator. The modulators have an in-phase and a quadrature carrier as inputs, respectively. The outputs of the modulators are then summed to give the final bandpass signal.

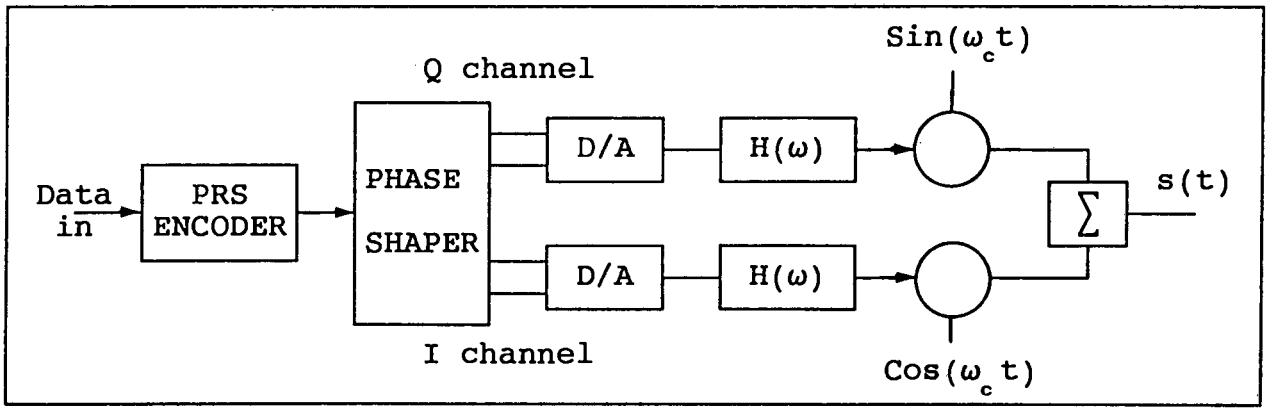


Fig. 6.1(a). CORPSK(4-7, 1+D) modulator block diagram

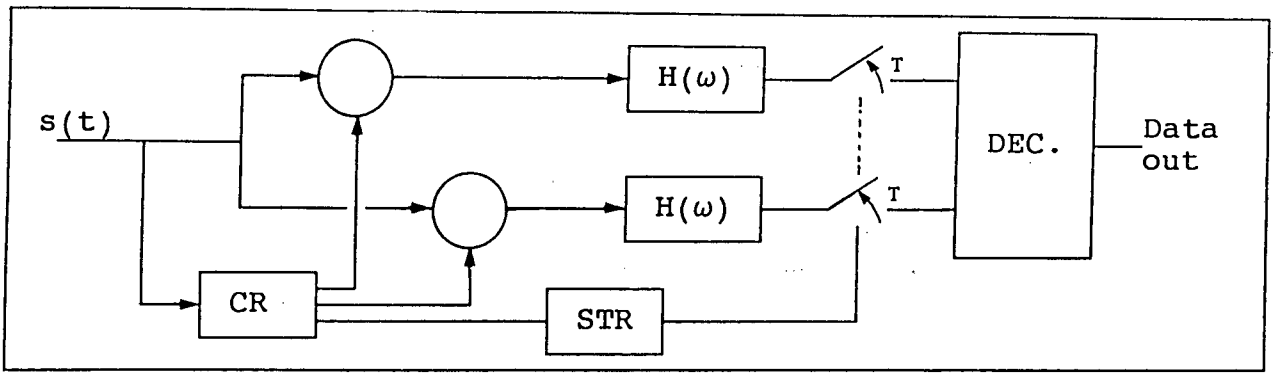


Fig. 6.1(b). CORPSK(4-7, 1+D) demodulator block diagram

The receiver demodulates the received signal coherently into in-phase and quadrature components and lowpass filters them to remove high-order product terms. The I and Q channels are then sampled separately, at the symbol rate. Successive sample-pairs are compared by the decision circuitry to derive the value of the transmitted symbol, and this is converted into a serial binary data stream.

Carrier and symbol timing recovery are required for the receiver to demodulate and sample the received signal correctly. Both these functions are carried out by use of a circuit nonlinearity and a phase-locked loop (PLL).

6.2 Pseudorandom bit sequence generator (PRBSG) (B.1)

"Out-of-service" testing is required to evaluate the performance of the final modem prototype [6.1] — for example, to generate eye diagrams or to perform bit error rate performance tests. This requires that the modem be supplied with a data stream that best represents a real-world sequence type. This can be achieved with a "pseudorandom" bit sequence, generated by an *n*-stage shift register and an EXCLUSIVE-OR gate.

The pseudorandom bit sequence pattern standardised by the CCITT for data transmission rates of 2.048 Mbps requires the use of a fifteen-stage pseudorandom bit sequence generator. A spectral line

spacing Δf results (Fig. 6.2.), where

$$\Delta f = \frac{f_b}{2^n - 1} \qquad (6.1)$$

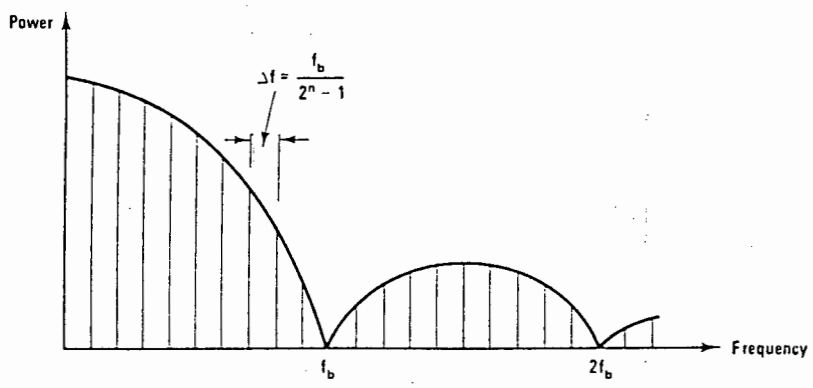


Fig. 6.2. Spectrum of binary pseudorandom bit sequence

For a 2.048 Mbps system, Δf is 62.5 Hz. The data transmission rate of the CORPSK(4-7, 1+D) modem is 120 kbps; the most suitable specification for spectral line spacing may be derived by scaling down the CCITT-recommended value. The required Δf is then 3.66 Hz. The use of a twenty-stage shift register yields Δf of 0.11 Hz.

The prototype PRBSG employs four 7496 five-stage shift registers (U1-U4). Feedback is by means of a 7486 EXCLUSIVE-OR gate (U6) fed from stages 17 and 20. A 7404 inverter (U5) and R-C combination presets the outputs of the shift registers to ensure that binary one's are present in the loop at power-on (if only zero's exist, there can be no feedback, and hence no data output).

6.3 PRS Encoder (B.2)

A block diagram of the partial-response encoder module is shown in Fig. 6.3.

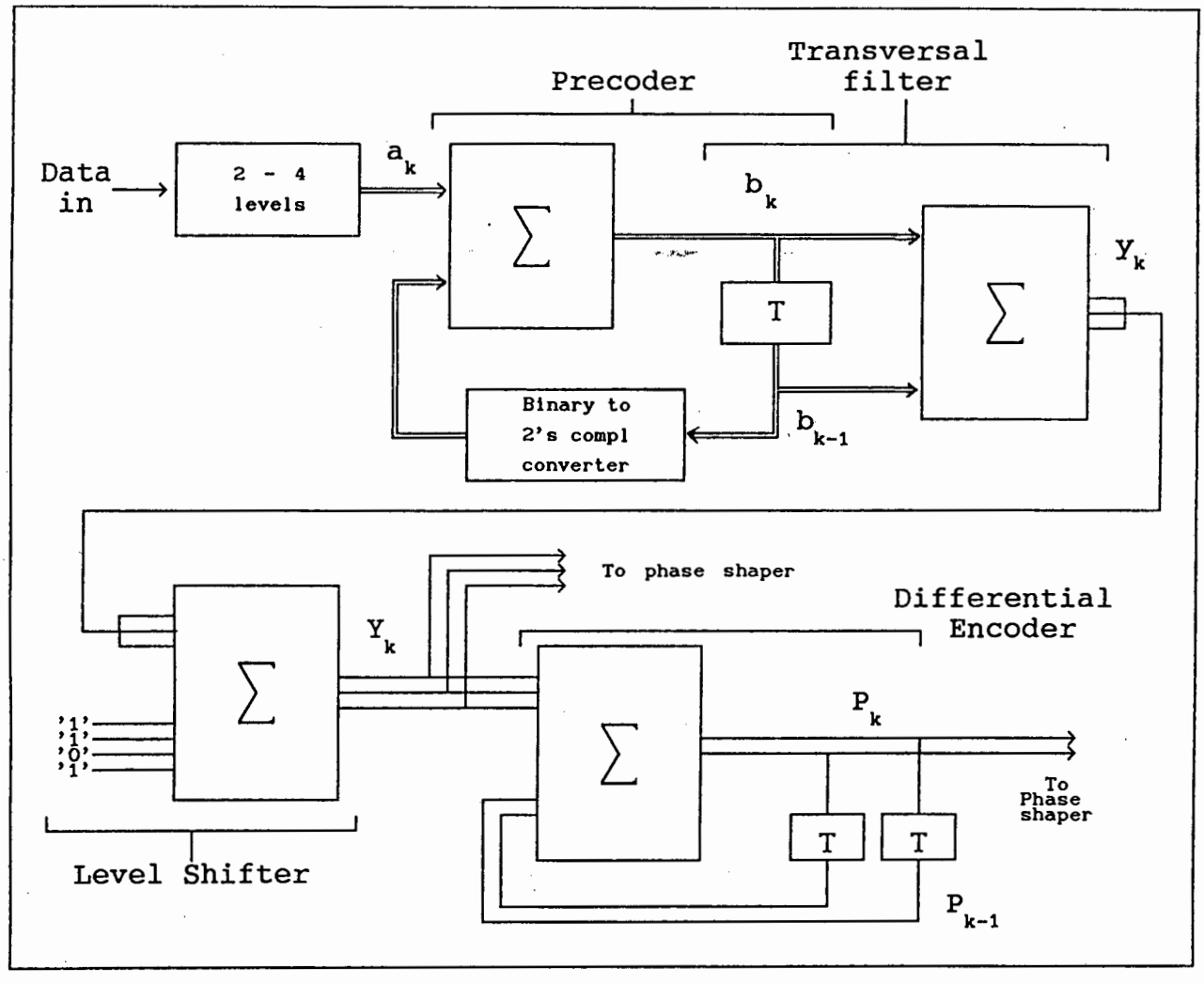


Fig. 6.3. PRS encoder module block diagram

The input data stream is in binary format, and has to be converted to a four-level signal $\{a_k\}$. The precoder output is given by

$$b_k = a_k - b_{k-1} \qquad (6.2)$$

and can be formed by adding the two's complement of the $\{b_{k-1}\}$ to the $\{a_k\}$. The system polynomial, F(D) = 1+D, indicates the addition of successive pairs of precoder output words, b_k's and b_{k-1}'s, ie.

$$y_k = b_k + b_{k-1} \qquad (6.3)$$

The output of the transversal filter, y_k, is a three-bit word

ranging from 0 to 6. This is to be placed symmetrically about zero by subtraction, yielding

$$Y_k = y_k - 3 \qquad (6.4)$$

Therefore, $\{Y_k\} = \{ -3, -2, -1, 0, 1, 2, 3 \}$. The output carrier phase change over one symbol, $\Delta\phi_m$, is given by

$$\Delta\phi_m = Y_k \cdot \frac{\pi}{2} \qquad (6.5)$$

A differential encoder then adds successive Y_k (mod 4) to give the actual phase end-point of the transmitted phase symbol, P_k. The phase trajectory shaping module uses an appropriately delayed replica of P_k as the starting phase point for the current symbol phase function and the tails of the preceding phase functions.

6.3.1. Two to four level converter (U1, U2)
A 7496 shift register is clocked at the bit rate to provide a two-bit binary word at half the bit rate (ie. the symbol rate, $f_b/2$). A 74175 latches this word at the symbol rate (a two-bit word is the code representation of a four-level signal).

6.3.2. Precoder (U3, U4, U5)
The two's complement converter is an exclusive-OR gate (7486); a 7483 summer performs the addition, while a 74174, clocked at the symbol rate, acts as a delay element.

6.3.3. Transversal filter (U6)
This is effected by means of a 7483 summer and U4.

6.3.4. Level Shifter (U8)
This subtraction (6.4) is also performed by two's complement addition, with a 7483 summer. The second input to the summer is the codeword '1101', ie. -3.

6.3.5. Differential encoder (U9, U10)
Again a 7483 and 74174 D-type flip-flop array are used.

The current Y_k is added to the output of the encoder, yielding a running total. The modulo-4 property is provided by feeding back only two bits of the output word to the input of the summer.

6.3.6 Symbol clock (U7)

All latches used in the two-to-four level converter, precoder and differential encoder operate at the symbol rate — U7A is a D-type flip-flop configured as a single-stage binary counter to double the period of the bit-rate clock.

6.4 Phase shaper (B.3, B.4)

The purpose of this module is to trace the carrier phase trajectory over each symbol period. Each symbol trajectory is made up of the current PRS input symbol convolved in time with a lowpass filter step response, added to the tails of the previous two PRS-convolved step responses, and all added to the final phase point of the third-last symbol (ie P_{k-3}). A further six phase points are to be interpolated on the trajectory, giving a total of eight phase points per symbol. A large-capacity read-only memory (ROM) can store the necessary phase path values, being addressed, in addition, by a modulo-8 counter to count through the interpolated points of the specified phase trajectory. The outputs of this module are the sine and the cosine of the phase trajectory in analog form. Fig. 6.4 is the block diagram.

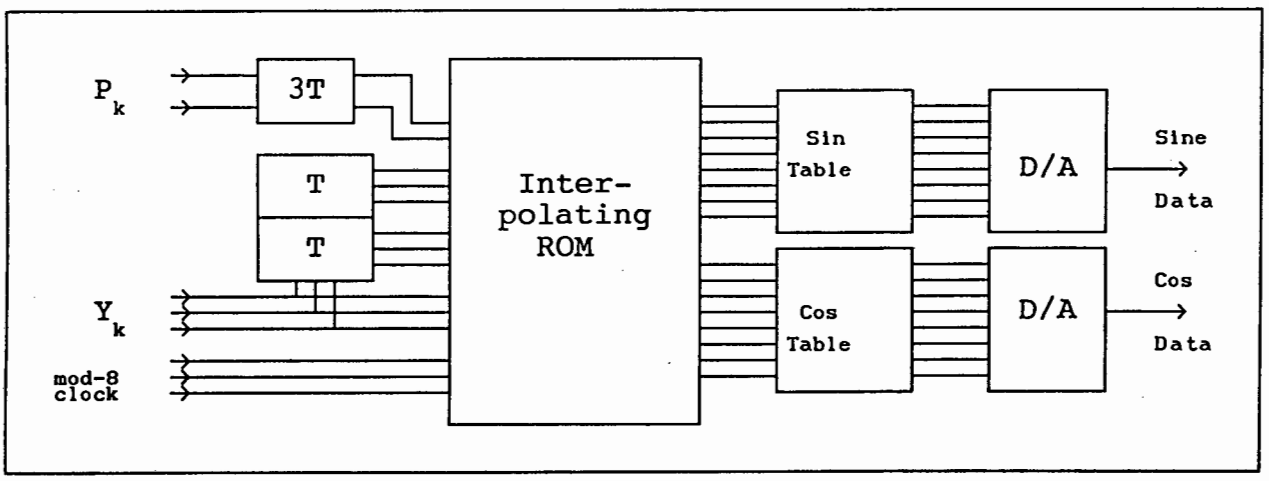


Fig. 6.4. Phase trajectory shaping module block diagram

Each possible phase trajectory is uniquely specified by three criteria:

- the symbol pulse truncation length, ν, in symbol intervals;
- the number of signal phase states, n;
- the number of possible phase shifts, N.

The number of possible phase trajectories is therefore

$$N_{PT} = N^{\nu} \cdot n \qquad (6.6)$$

So, for CORPSK(4-7, 1+D) with an interpolation factor of eight and a pulse truncation length of three symbol intervals, 1372 possible trajectories result. To calculate the required ROM capacity, these factors, ν, n and N have to be powers of 2 – a value of 8 is therefore assumed for N, and not 7. The ROM capacity is then given by

$$\begin{aligned} R &= N^{\nu} \cdot n \cdot \eta \qquad (6.7) \\ &= 16384, \text{ or } 16 \text{ kbytes}, \end{aligned}$$

where η is the oversampling ratio, $\eta = 8$.

It is clear from (6.7) that the required memory capacity increases exponentially with ν, the pulse truncation length – consequently, truncation lengths greater than three symbol intervals, while desirable, yield increasingly impractical implementations.

The delay elements in Fig. 6.4 are implemented with sets of 7496 shift registers, clocked at the symbol rate. Two registers are required for the P_k, and three for the Y_k, to give P_{k-3}, Y_{k-2} and Y_{k-1} in addition to Y_k. These four codeword inputs together account for eleven ROM address bits. A further three ROM bits are required for the modulo-8 counter input, yielding a total of fourteen. The ROM is a 27128 (16k x 8) EPROM device.

The contents of the phase trajectory ROM were created by means of a Pascal program (described in Appendix D).

The sine and cosine look-up EPROMs (27C16's) are driven directly by the output of the phase trajectory EPROM. Their contents were also generated by Pascal software, listed in Appendix D. Two DAC0800 D/A converters convert the sine and cosine digital words into an oversampled analog signal. The differential current outputs of these devices drive LF347-based current-to-voltage converters. The in-phase and quadrature signals represent the instantaneous Cartesian coordinates of the moving unit signal vector.

6. Lowpass transmission filters (B.5)

In Pulse Amplitude Modulation (PAM) systems, it is desirable that the transmission pulse shape have a relatively large response at the symbol instant, and small response at subsequent multiples of the symbol period. This minimizes intersymbol interference (ISI). Such an all-pole lowpass filter solution is still applicable in the design of the CORPSK modem. Although PRS encoding deliberately introduces ISI, it is a controlled amount, and subsequent ISI contributions must still be kept to a minimum. Any excess ISI will cause a degradation in the system's horizontal eye opening, and hence reduced immunity to receiver timing errors. Lind and Nader [6.2] specify pole locations for such data transmission filters.

For a normalized squared error, E, of 10^{-4}, a fourth-order transmission filter has an insertion loss (A_N) of 3.72 dB at ω_N, the Nyquist frequency. At a frequency of $4.42 \cdot \omega_N$ the attenuation is 60 dB. Assuming a roll-off of $6N$ dB/octave from the point at ω_N, N=4, an attenuation of approximately 50 dB can be expected at $4 \cdot \omega_N$.

The output spectrum of each channel's D/A converter will look similar to that in Fig. 6.5. If ω_N is chosen to be twice the symbol rate (ie. equal to the *bit* rate), very little signal power will fall within the transition region — filter group delays tend

to be large here — and the repeated spectrum of the oversampled waveform will fall at the filter's -50 dB point.

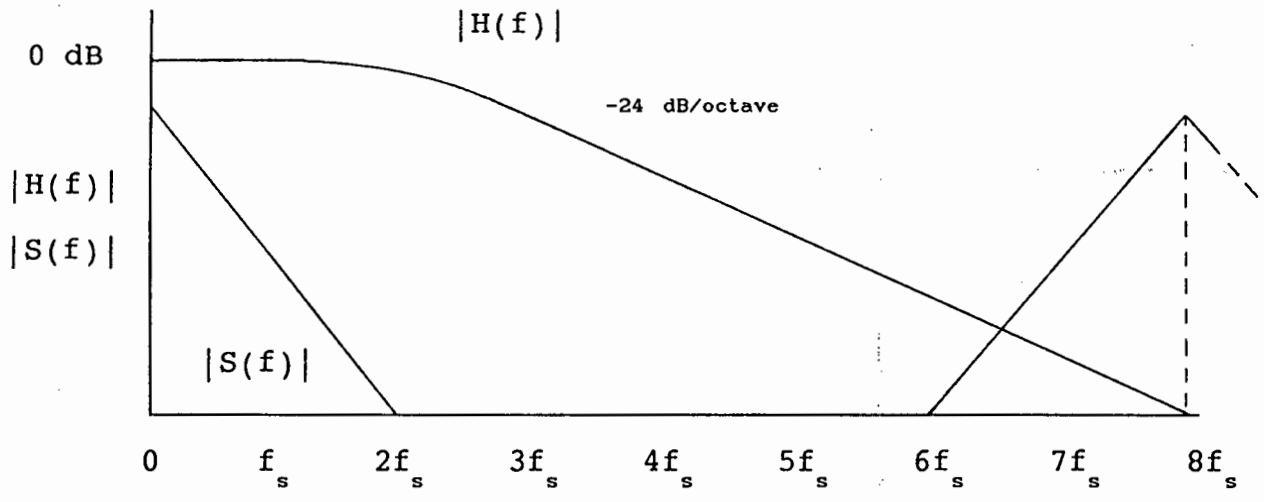


Fig. 6.5. D/A spectrum and transmission filter characteristic.

The normalized pole positions are given by

$$p_{1,2} = -0.488949 \pm j0.377055$$
$$p_{3,4} = -0.243995 \pm j0.952931 \qquad (6.8)$$

These are implemented with a two-stage state-variable active lowpass filter (Fig. 6.6.).

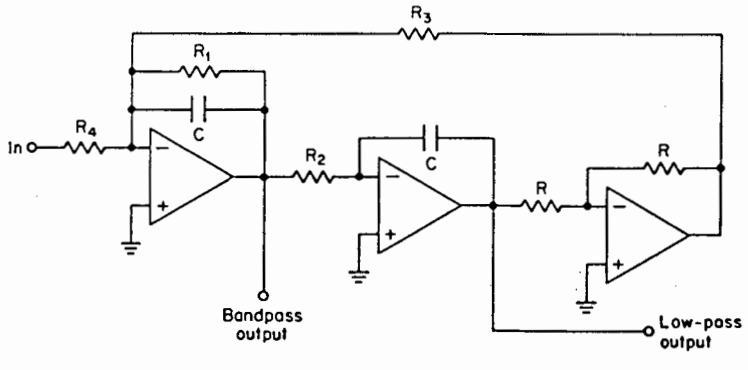


Fig. 6.6. State-variable all-pole lowpass configuration.

The following design equations apply:

$$R_1 = \frac{1}{2\alpha C} \tag{6.9}$$

$$R_2 = R_3 = \frac{1}{C\sqrt{\alpha^2 + \beta^2}} \tag{6.10}$$

$$R_4 = \frac{1}{AC\sqrt{\alpha^2 + \beta^2}} \tag{6.11}$$

where A is the required dc gain. The denormalized component values are found by employing denormalized α' and β' values, from

$$\begin{aligned} \alpha' &= \alpha \times (2\pi f_0) \\ \beta' &= \beta \times (2\pi f_0) \end{aligned} \tag{6.12}$$

f_0 the new cutoff frequency. Choosing C = 0.1 nF with a unity-gain configuration, the first pole-pair is created with R_1 = 6780 Ω, R_2 = R_3 = R_4 = 1074 Ω. For the second pole-pair, R_1 = 1359 Ω, R_2 = R_3 = R_4 = 674.2 Ω.

6. In-phase and quadrature carrier generation (B.6)

A carrier frequency of 432 kHz was chosen for convenience. A TTL crystal oscillator (7404) generates a 3.456 MHz pulse train which is divided down (by 8) to give the carrier frequency (74193). A high-Q LC bandpass filter (LF347) isolates the fundamental frequency, and its output drives the in-phase output driver and a 90° phase shifter, implemented with a unity-gain LF347-based pole-zero element. This quadrature-shifted carrier signal then feeds the quadrature output driver.

The 90° phase shifter is adjustable. The correct setting is obtained by viewing both carriers on an X-Y oscilloscope, and adjusting for a perfectly circular trace. Any deviation from this

indicates not only a relative phase shift between the two carriers, but also unequal carrier amplitudes.

6. Modulator (B.7)

Two SBL-1 balanced mixers are used to multiply the in-phase and quadrature baseband signals by their respective carriers. The outputs are added in a virtual-earth configuration, thus reducing the possibility of crosstalk. This final bandpass signal is then amplified by a line driver stage, which isolates the output channel by means of transformer coupling.

6. Receiver front end (B.8)

The channel is transformer-coupled into a pair of high-input impedance noninverting amplifiers (LF347). The first provides a buffered signal to supply the carrier and symbol timing recovery module. The second supplies a line driver/power splitter stage, whose two (identical) outputs drive the RF inputs of the coherent demodulator.

6. Carrier and symbol timing recovery module (B.9, B.10)

Simulations indicated that a fourth-order nonlinearity was required for recovery of the receive carrier from the received bandpass signal (Chapter 5, §4). In addition, results showed that a multi-order nonlinearity (eg. full-wave rectifier) would not cause undue aliasing error in the recovery spectrum.

A phase-locked loop (PLL) with a free-running frequency of $4\omega_0$ is preceded by a precision full-wave rectifier and a Q-adjustable resonant circuit [6.3].

The PLL is an EXAR device, the XR-215, configured as a tracking filter. A free-running frequency of 1.728 MHz is required. A fairly narrow capture range is required, since the carrier component at 1.728 MHz is surrounded by symbol timing components

60 kHz away. A capture range, $\Delta\omega_c$, of 10 kHz was chosen. The lock range, $\Delta\omega_L$, must be no less than the capture range, and a value of 20 kHz is therefore chosen.

6. Detector and decoder (B.11 - B.13)

A block diagram of the detector and decoder module is shown below, in Fig. 6.7.

The information-carrying function in CORPSK is the carrier phase, encoded differentially. Symbol-by-symbol sampling requires that the phase be sampled at the end of each symbol period, and successive samples compared to give the value of the original PRS symbol.

The in-phase and quadrature signals represent the x and y Cartesian coordinates of a moving signal vector of unit length. Each sample-pair therefore represents the coordinates of the

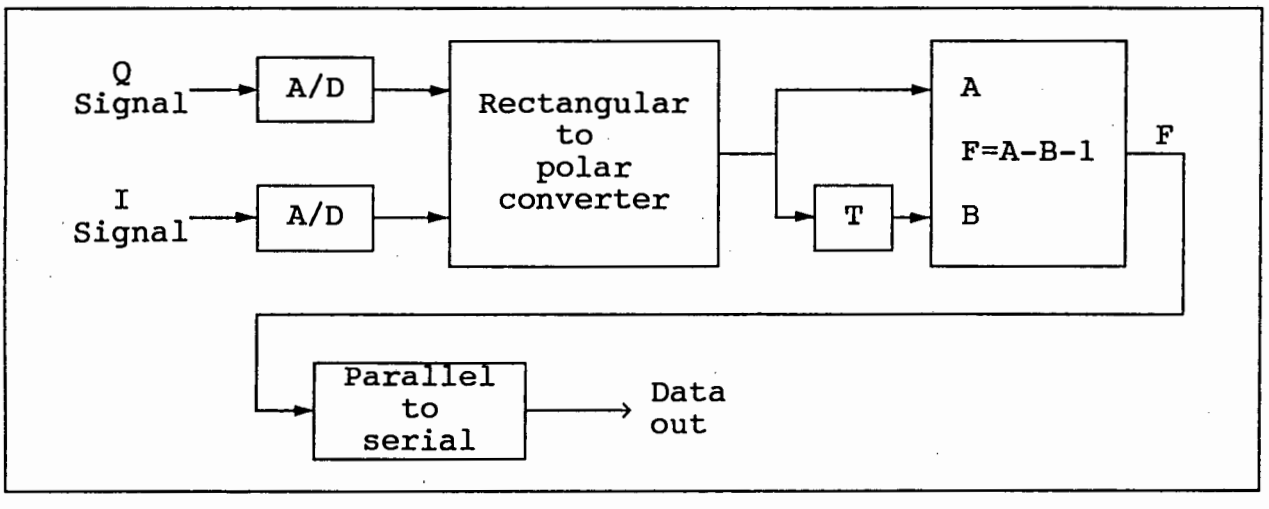


Fig. 6.7. Detector and decoder block diagram

vector at the end of each symbol period. A rectangular-to-polar conversion yields the symbol-end phase values.

The phase values from the current and the last symbol interval are compared according to the function

$$a'_k = \left[\left[\left[\Phi_k - \Phi_{k-1} \right] \cdot \frac{2}{\pi} \right] - 1 \right]_{\text{mod } 4} \qquad (6.13)$$

to yield an estimate of the original four-level data value (See Ch. 3). This is then converted into serial form.

The incoming in-phase and quadrature signals are sampled at the symbol rate by an ADC0820 analog-to-digital converter (ADC). The incoming signals are symmetrical about zero volts, and so level translation and scaling are performed by operational amplifiers (LF347). The ADC's are negative-edged triggered by a 600 nS pulse, provided by a 74123 monostable multivibrator.

The two ADC's have eight-bit outputs, and these are combined into a sixteen-bit address for a pair of 27256 (32k × 8) EPROM's. The Cartesian coordinate input is converted into an eight-bit phase value. Fig. 6.8 shows the range of output phase values relative to the (noise-corrupted) constellation diagram.

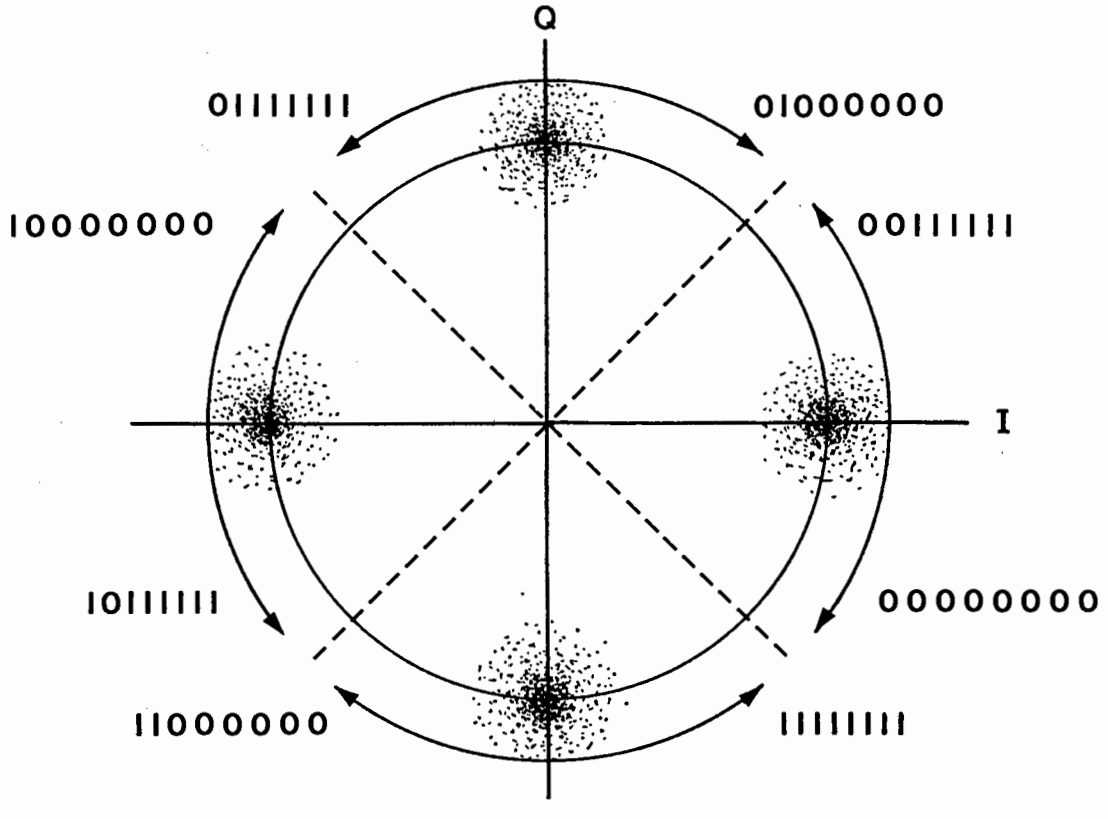


Fig. 6.8. Rectangular-to-polar conversion.

The phase of each received sample will be in the region of multiples of $\pi/2$ because of the "spreading" resulting from channel noise. The decoder makes a decision as to the original transmitted phase value by truncating the eight-bit phase value to two, thereby representing each of the four states by a unique codeword.

Clearly, the slightest corruption of the received signal by noise could cause decoding errors because of the rounding-down effect of truncation. For this reason, the rectangular-to-polar transformation deliberately introduces a rotation of the coordinate axes by $\pi/4$ so that the codeword output can represent a phase *region*, and not just a state. The Pascal program that created the EPROM look-up table is listed in Appendix D.

A 74181 arithmetic logic unit (ALU) implements the subtraction function expressed in (6.13). The inputs are the current and the last phase region codewords, provided by a pair of 7496 shift registers clocked at the symbol rate. The two-bit output of the ALU is the parallel representation of the two binary data bits transmitted. A 74151 multiplexer has, as one address input, the symbol rate clock which provides two logic states per symbol. The multiplexer output is therefore switched cyclically between the two ALU output lines, and this gives the final serial data stream.

Every output bit is compared (by a 7486 EX-OR gate) with a delayed replica of the transmitted bit to give an error signal when there is a mismatch. A 74175 latch ensures that the source and decoded bits are exactly synchronized. An estimate of the system bit error rate can be made by averaging the frequency of error signals, relative to the number of bits sent. An optical error signal is also provided, for system adjustment.

REFERENCES

[6.1] Feher, K. and the Engineers of Hewlett-Packard, *Telecommunications Measurements, Analysis, and Instrumentation*, Prentice-Hall, Englewood Cliffs, N.J., 1987.

[6.2] L. F. Lind, S. E. Nader, *"Design Tables for a Class of Data-Transmission Filters,"* Electronics Letters, 15 September, 1977, Vol. 13, No. 19, pp. 564-566.

[6.3] Feher, K., *Advanced Digital Communications Systems and Signal Processing Techniques,* Prentice-Hall, Englewood Cliffs, N. J., 1987.

CHAPTER 7
SYSTEM TESTING AND EVALUATION

7.1 Test requirements

The primary performance parameter in any digital transmission system is the rate at which errors occur in the received data stream. The major degradations affecting this are [7.1]

1. Power amplifier AM to PM conversion.
2. Compression and expansion of signals by nonlinear amplifiers.
3. The introduction of excessive amounts of intersymbol interference (ISI) into the signal.
4. Reduced orthogonality in quadrature modulators.
5. Timing errors.
6. Carrier phase offset and jitter.

Any evaluation of a modulation scheme's performance should assess at least some, if not all, of these factors. In the case of the CORPSK(4-7, 1+D) system implemented, the first two are not measured explicitly, although the behaviour of the system spectrum under conditions of hardlimiting is simulated.

All-pole baseband transmission filters minimize the introduction of uncontrolled ISI, and a qualitative assessment is made of their efficacy.

The effect of reduced modulator orthogonality is the introduction of envelope fluctuations as well as jitter in the recovered carrier. Both of these are measured here. An assessment of quadrature carrier orthogonality is performed.

The synchronization subsystem does not address directly the recovery of symbol timing, but the peak-to-peak jitter of the recovered carrier is measured.

Also, system performance has been characterized by the measurement of intermediate parameters such as baseband spectra.

All time-domain measurements were performed with a Hewlett-Packard 54501A 100 MHz digitizing oscilloscope, while an *HP* 8590A spectrum analyser was used for spectral measurements. All printouts from these instruments were by means of an *HP* Thinkjet printer.

7.2 Baseband channels

7.2.1 Eye diagrams

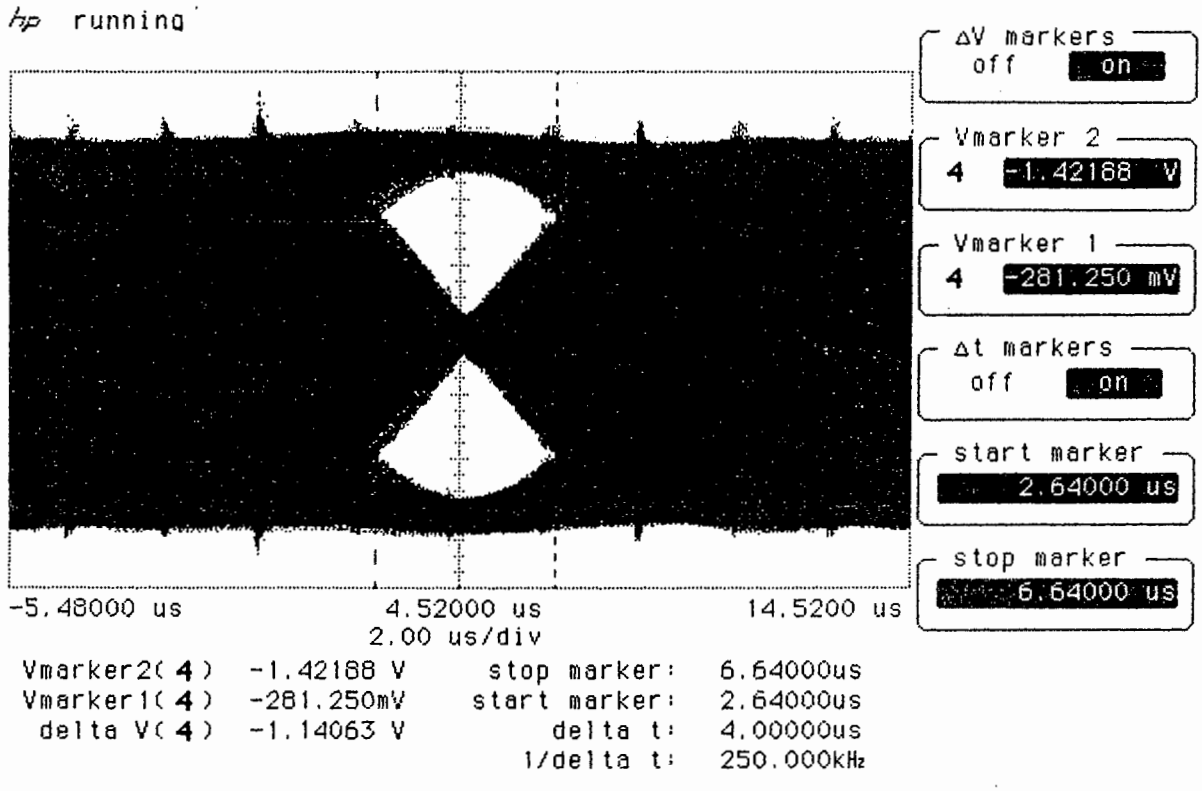


Fig. 7.1(a). Baseband eye diagram for one channel (measured after transmission filter)

Fig. 7.1(a) shows the baseband eye diagram for one channel, measured after the transmission filters. The vertical eye opening (VEO) has been degraded by circuit imperfections, such that the three signaling levels have been broadened into regions. The oscilloscope voltage markers have been placed at the upper and lower edges of the lower eye. The baseband signal has a peak-to-peak voltage of 2.9V. Hence a *performance degradation*

[7.1, p. 23] is introduced, of $20 \cdot \log_{10}(1.45/1.14) = 2.1$ dB.

The horizontal eye opening (HEO) is 3.8 μseconds. This represents $(4.0 \times 10^{-6} \times 60 \times 10^{3}) = 0.24$ of a symbol interval. Our simulation results (Chapter 5 §1.2) predicted a HEO of 0.4 of a symbol interval. A degradation of $20 \cdot \log_{10}(0.4/0.24) = 4.4$ dB has therefore resulted (if the symbol-by symbol sampling occurs at the eye midpoint, then this degradation in HEO is of little consequence).

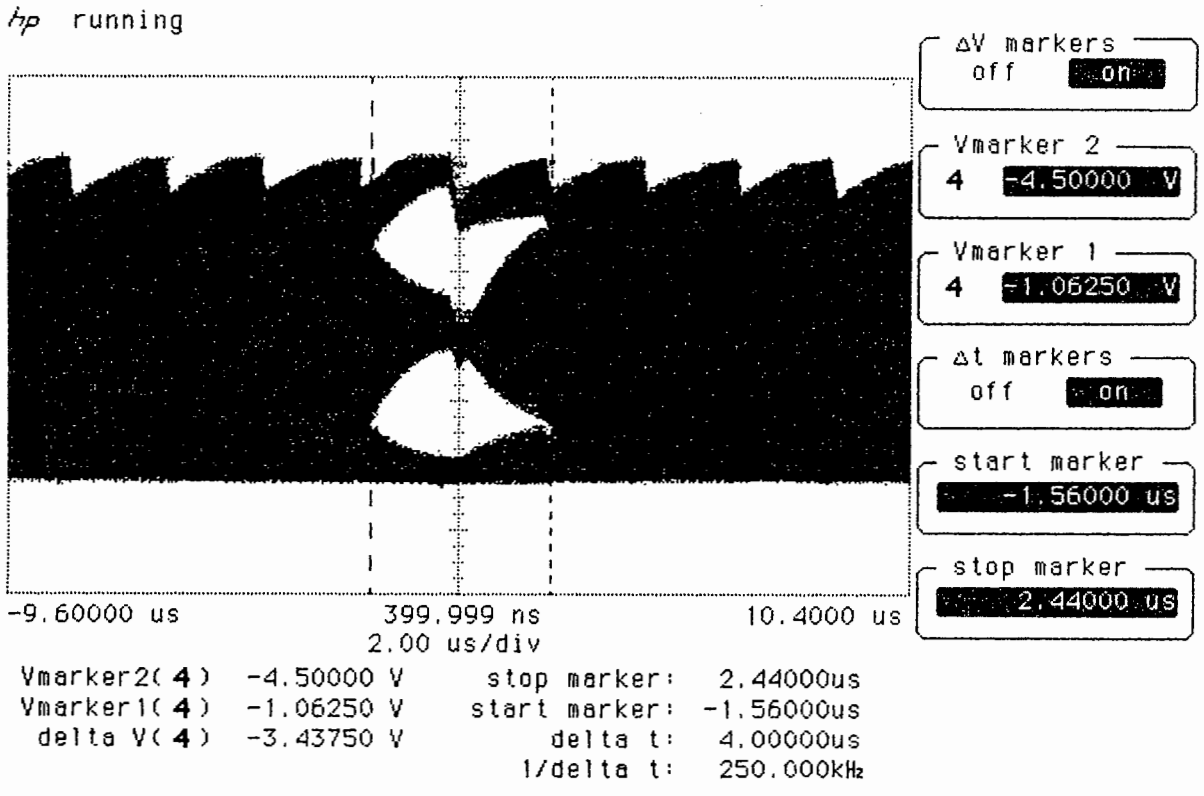


Fig. 7.1(b). Baseband eye diagram for one channel (measured after D/A).

Fig. 7.1(b) shows the same channel's eye diagram, measured before the transmission filter, at the output of the digital-to-analog converter. The D/A output signal exhibits a first-order transient of the order of 1 μsecond, most likely caused by circuit layout and hardware imperfections. The result is that the eye is severely distorted, and the width of the signaling levels is similar to those at the transmission filter outputs ie. the VEO has been

impaired. Also, the HEO has been reduced. The eye opening is, in fact, slightly distorted compared to that at the filter output, showing that the transmission filters actually "clean up" the signal before transmission.

7.2.2 Baseband spectrum

The output of the D/A converter of each baseband channel is an oversampled waveform, 8 samples per symbol period. The baseband spectrum, centred at 0 Hz, is repeated at multiples of the sampling rate. Fig. 7.2. shows the spectrum analyser plot.

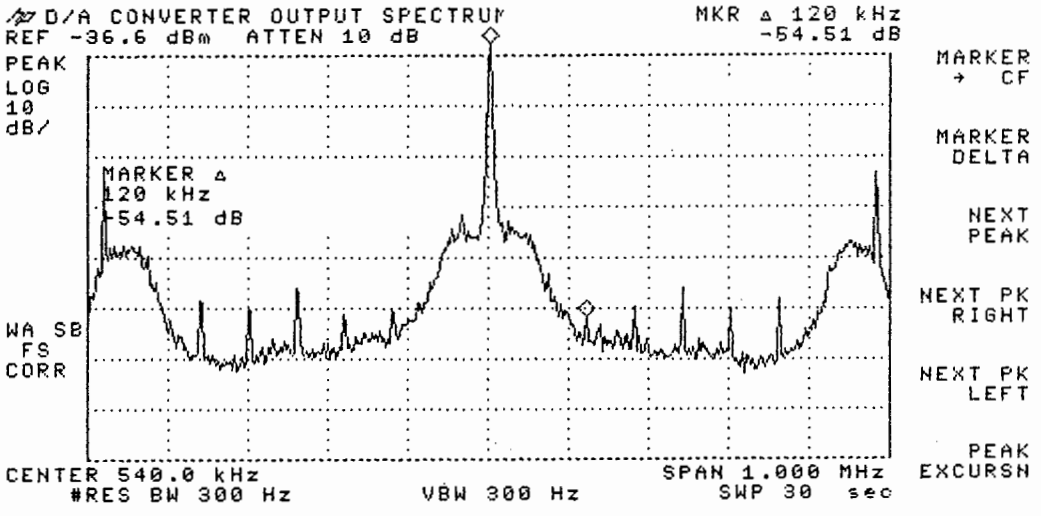


Fig. 7.2. Spectrum of one unfiltered baseband channel.

This spectrum is composed of a continuous part and discrete components at multiples of the symbol rate. Similar results were found in [7.3], where a system comparable to CORPSK is described.

The spectrum in Fig. 7.2. appears to have a high residual noise level surrounding the main lobe, so that it does not fall to the low value predicted by simulations. The probable cause of this is

D/A converter noise — a similar effect occurs in the work of Maseng in [7.3], where measured values are 10 to 15 dB worse than predicted. The spectral efficiency is approximately 0.5 bps/Hz.

The filtered baseband spectrum is shown in Fig. 7.3.

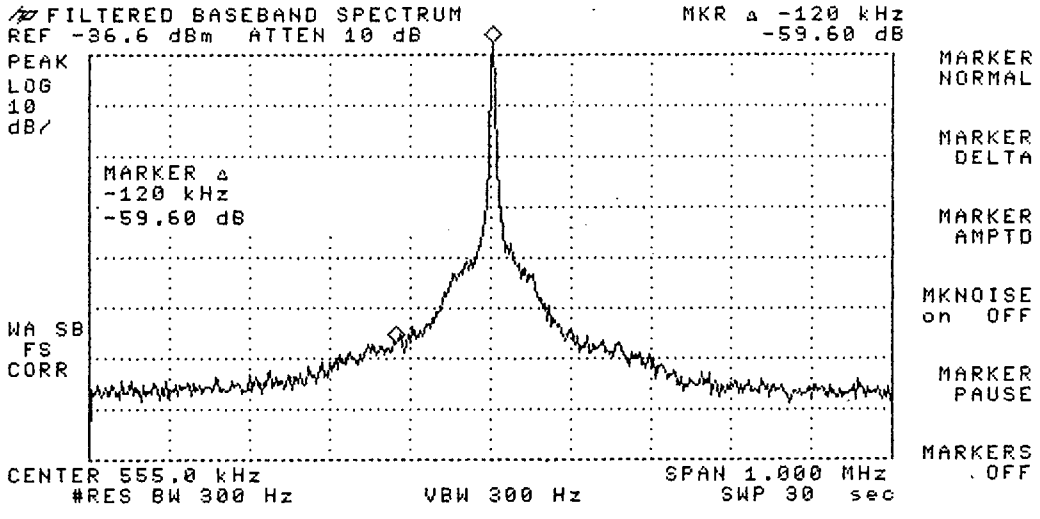


Fig. 7.3. Spectrum of one filtered baseband channel.

Although the filtered spectrum lacks the definition of the unfiltered case, all higher-order sampled spectral lobes have been eliminated. Some insertion loss is apparent in the region from 60 to 120 kHz — this causes a "rounding" of the edges of the spectrum. (Note also that the transmission filters trade off their improved ISI characteristics against a reduction in spectral roll-off).

7.3 Transmission filters

Below is the frequency response of the baseband transmission filter.

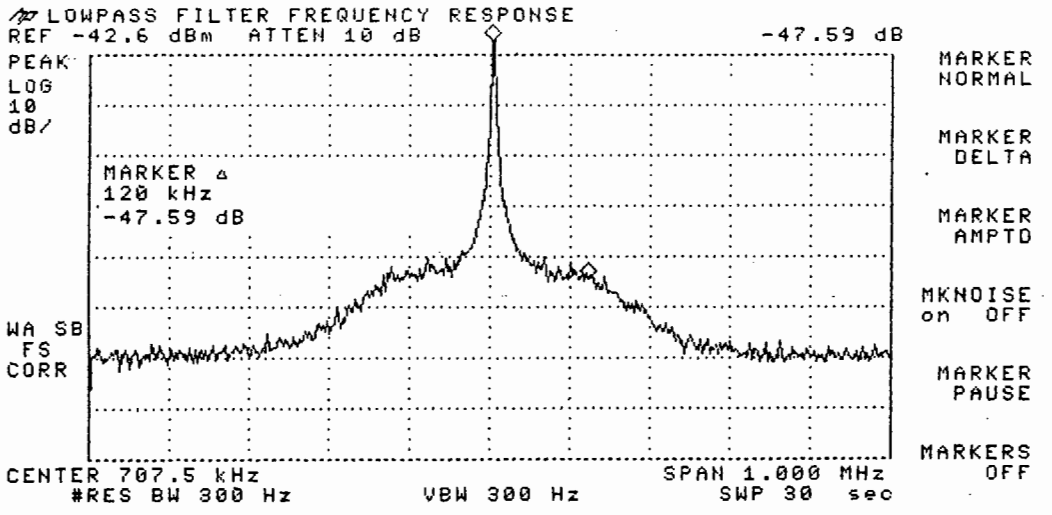


Fig. 7.4. Frequency response of baseband transmission filter.

An insertion loss of approximately 3.7 dB is expected at 120 kHz. A 24 dB/octave rolloff is expected from that point. There is, indeed a loss at 120 kHz, estimated to be 2 to 3 dB. From 120 kHz to 240 kHz, the spectrum rolls off about 16 to 18 dB and flattens out.

7.4 In-phase and Quadrature carrier generators

7.4.1 Spectra

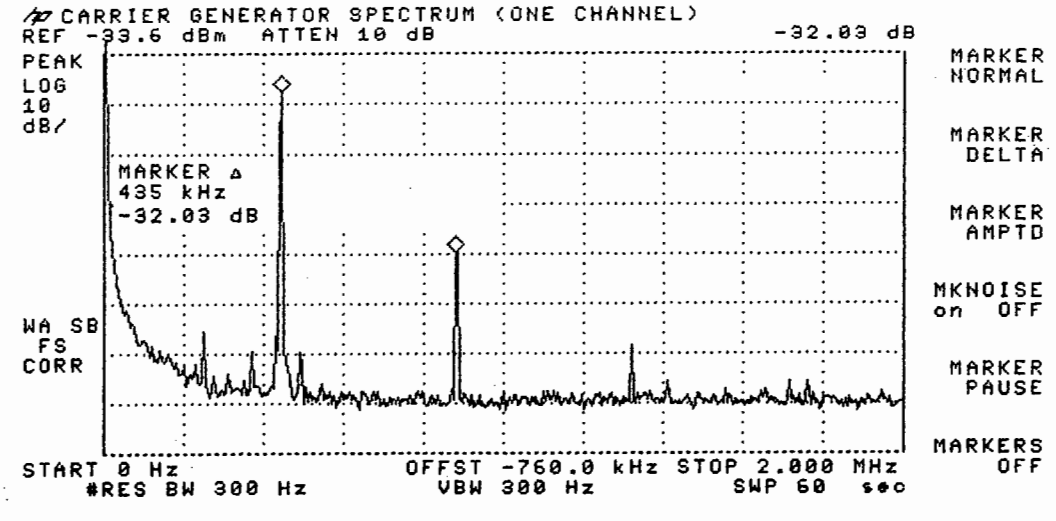


Fig. 7.5. Spectrum of transmit carrier (one channel only).

The spectrum for one channel of the in-phase and quadrature carrier generator is shown in Fig. 7.5. The first harmonic is over 32 dB down in amplitude from the fundamental. The third harmonic is approximately 50 dB down. Also, there is little phase noise on the fundamental, any components being approximately 50 dB less than the component at this frequency. Both channels exhibit similar characteristics.

7.4.2 X-Y diagram

The Lissajous figure of the in-phase and quadrature carriers will show imperfections in orthogonality as well as carrier waveform distortion. Two equal-amplitude sinusoid carriers, perfectly orthogonal and distortion-free, will generate a perfectly circular trace. Since their relative phase is adjustable in this design, a perfectly circular trace was obtained, with no discernible bulges or dents whatsoever — these would indicate waveform distortions.

7.5 Modulator

7.5.1 Output spectrum

This is shown in Fig. 7.6.

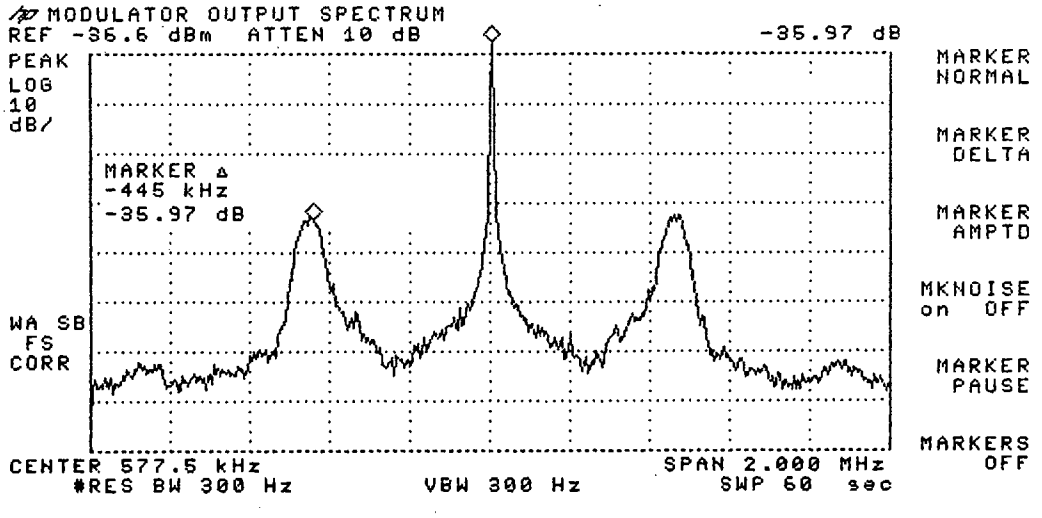


Fig. 7.6. Modulator output spectrum.

The carrier frequency is 432 kHz. The spectrum analyser exhibits some frequency drift, and so the carrier frequency appears higher on longer sweeps.

7.5.2 Signal envelope

Any fluctuation in the output signal envelope represents a degradation in system performance. Two approaches are available for assessing envelope fluctuations, each method highlighting different causes. They are, firstly, by analysing the X-Y diagram (Lissajous figure) of the baseband in-phase and quadrature modulating signals, and, secondly, by measuring the envelope fluctuations directly. The former method will show how much degradation is introduced by the effects of phase pulse truncation, as well as D/A and transmission filter imperfections (unequal group delays in each channel, for example). The latter will show the contribution not only from these, but also from the mixers and bandpass-summer circuitry of the transmitter.

7.5.2(a) X-Y diagram

The XY diagram created by the in-phase and quadrature channels should, in an ideal system, be a perfect circle with an infinitely narrow trace width. Since the trace represents the final signal envelope in a baseband form, envelope fluctuations are seen as a spreading of the trace into a band. See Fig. 7.7.

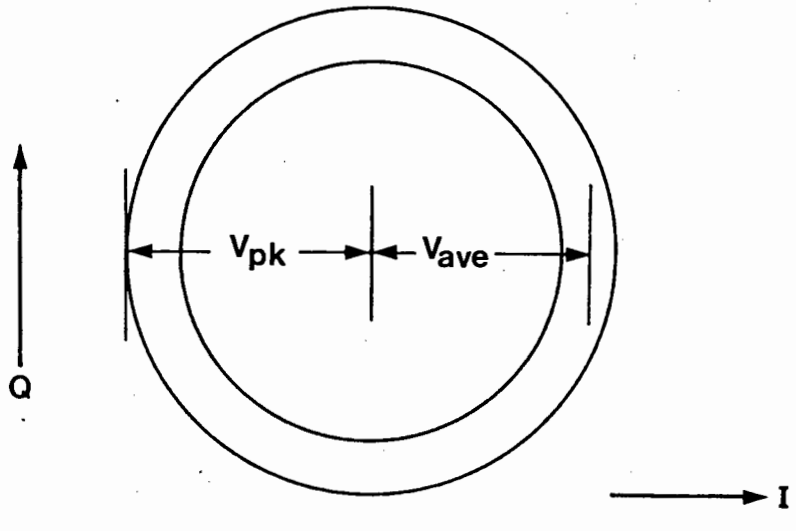


Fig. 7.7. Envelope fluctuation measurement by baseband analysis.

If envelope fluctuation can be described as a ratio between the envelope peak and the average (ie. V_{pk} vs. V_{ave}), then the measured fluctuation is $20 \cdot \log_{10}$(4.8 divs / 4.4 divs) = 0.76 dB.

7.5.2(b) Direct measurement of envelope fluctuations

An assessment is also obtained by viewing the modulated signal in time and estimating the width of the fluctuation "band" that appears on the top and the bottom of the signal (Fig. 7.8.).

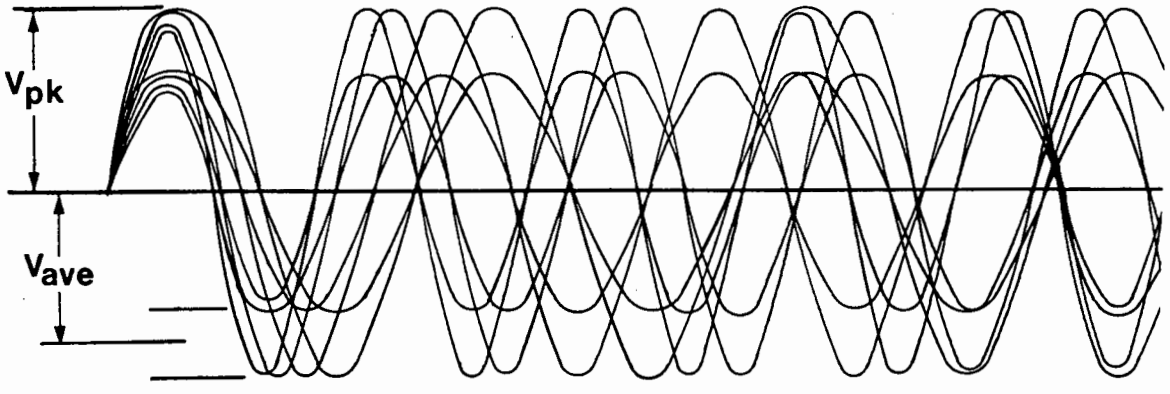


Fig. 7.8. Time-domain estimation of envelope fluctuation.

Using the same definition as above, an envelope fluctuation of $20 \cdot \log_{10}$(3 divs / 2.6 divs) = 1.2 dB is found. This indicates that the baseband circuitry (D/A's, transmission filters) introduce approximately 0.8 dB, while the carrier circuitry (modulators and summer) contribute just over 0.4 dB.

7.6 Carrier and symbol timing recovery system

7.6.1 Spectra

Fig. 7.9. is a plot of the output spectrum of the synchronization nonlinearity.

Little information can be recovered from the lower-order spectra, but the fourth-order spectrum contains a carrier component approximately 20 dB larger than the surrounding noise. This component is tracked by a PLL to yield the recovered receive carrier.

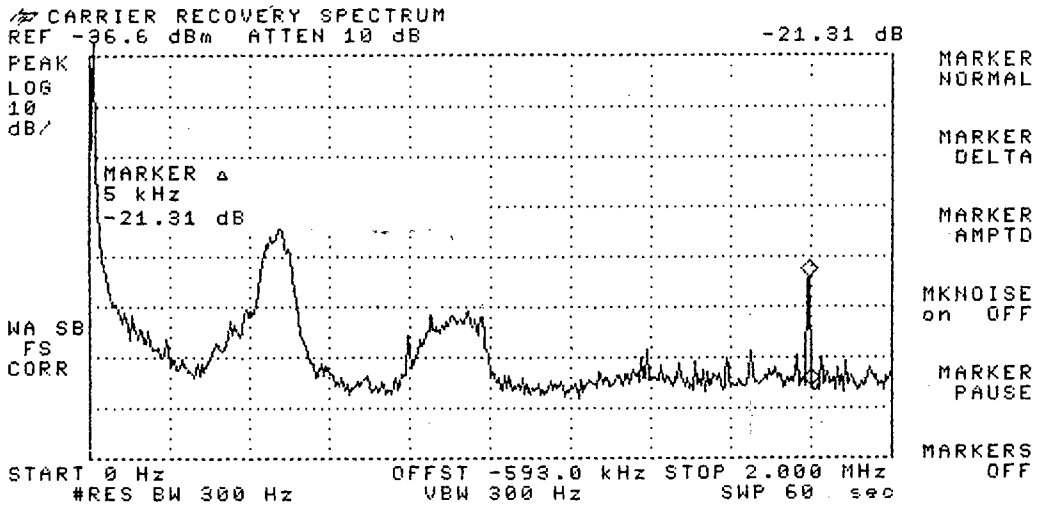


Fig. 7.9. Carrier and symbol-timing recovery nonlinearity output spectrum.

The carrier component is surrounded by spectral components at distances that are multiples of the symbol rate. However, they are up to 20 dB smaller than the carrier component. They are also only about 2 to 3 dB above the surrounding noise, and therefore require more complex tracking circuitry. The simulation of Chapter 5 (§4) predicted that these components would be small. (Recovery of these components is feasible, but the added circuit complexity is beyond the scope of this investigation).

The final recovered carrier spectrum of one channel is plotted in Fig. 7.10.

Comparing this figure with the spectrum of the original transmit carrier (Fig. 7.5.), there is far more phase noise on the recovered fundamental. Also, the harmonics are larger in amplitude — they, too, have larger phase noise components. The first harmonic is now approximately 29 dB down in amplitude from the fundamental.

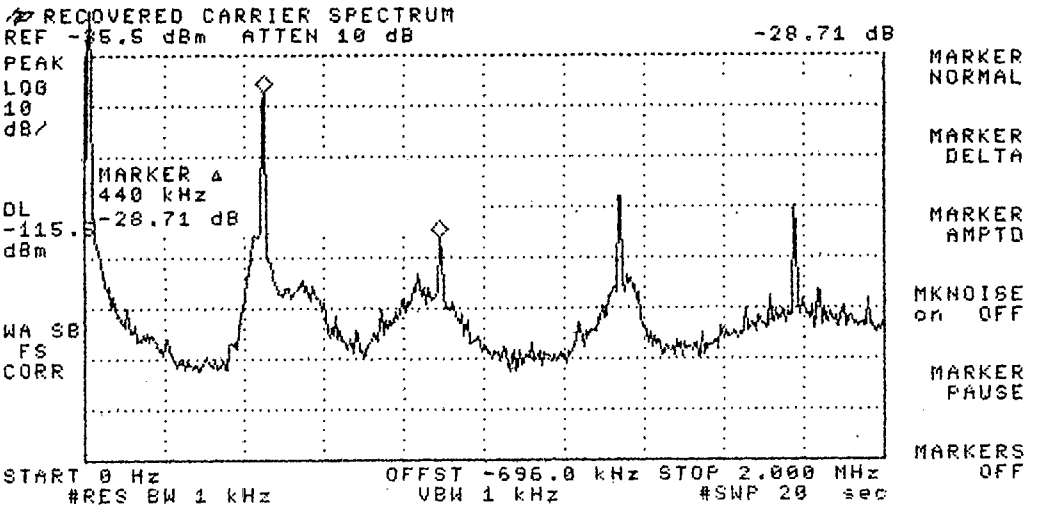


Fig. 7.10. Recovered carrier spectrum (one channel).

7.6.2 Timing jitter

The peak-to-peak timing jitter of the recovered carrier clock (before filtering out the fundamental, for ease of measurement) was evaluated. Fig. 7.11. shows the procedure.

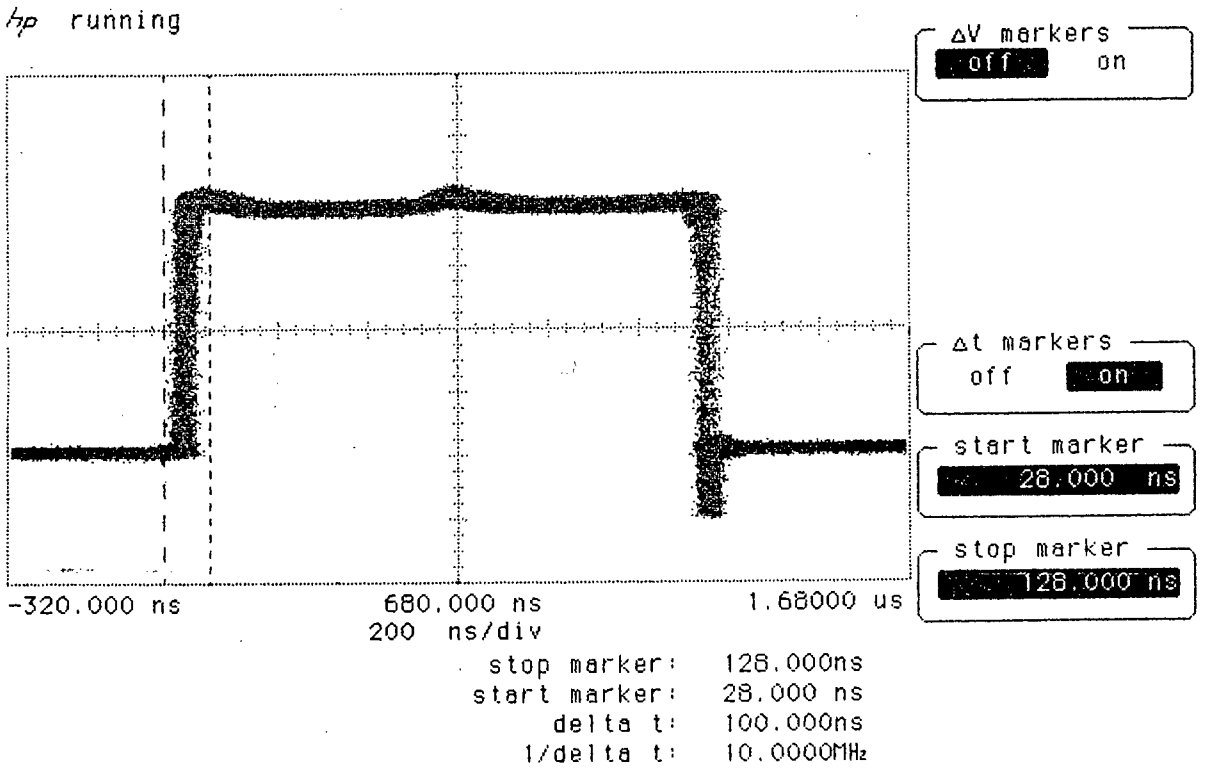


Fig. 7.11. Recovered carrier clock jitter measurement.

The carrier has a period of $1/(432 \times 10^3) = 2.315$ μseconds. The peak-to-peak jitter, J_{pp}, is measured to be 100 nseconds, or 4.3% of a carrier cycle. This translates into a random carrier phase jitter of up to 7.7° (peak) either side of the central phase point.

7.7 Decoder

7.7.1 Noise generator (B.14)

The output spectrum of the noise generator is shown below.

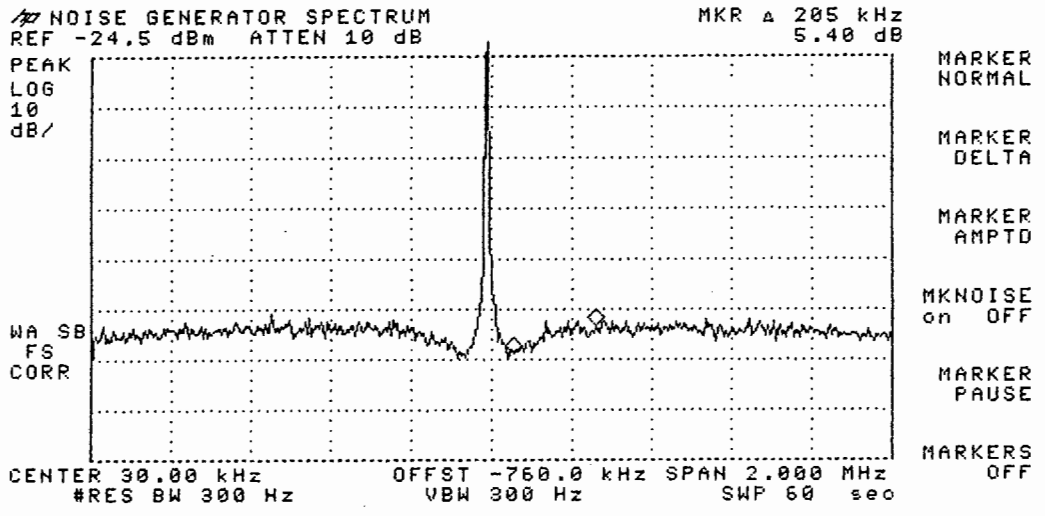


Fig. 7.12. Output spectrum of noise generator.

The spectrum is flat to within about 4 or 5 dB across the baseband channel width, and almost completely flat across the span of 1 MHz (up to several MHz, in fact). It is therefore assumed that this approximates a true white noise source.

The crest factor of the noise generated varies as the inverse of the rms value, since the peak values are limited by supply rail voltages. Over the range of noise tests, the crest factor varies from 15 dB, for low CNR, to 25 dB for high CNR. These figures, however, are an *upper bound* only — the characteristics of the noise source itself could reduce this figure of merit

substantially.

7.7.2 Symbol error rate results

Bit error rate tests were performed according to the scheme illustrated in Fig. 7.13.

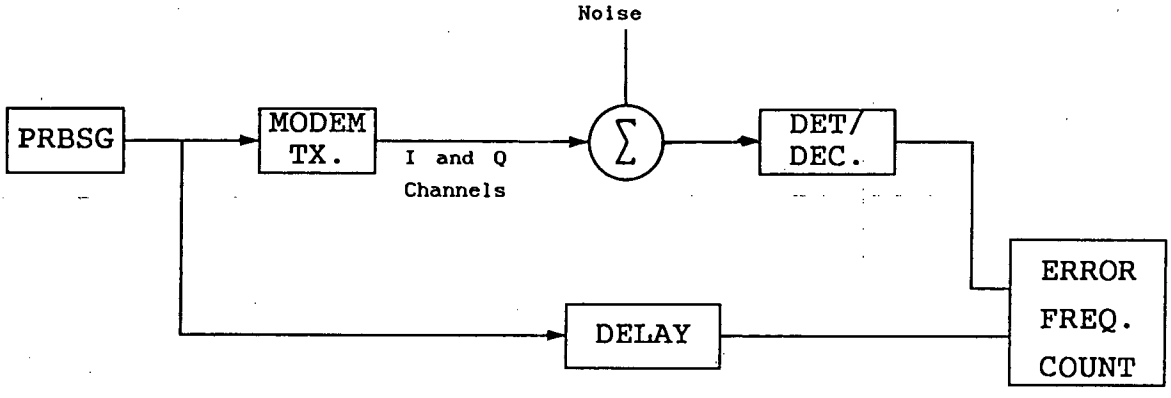


Fig. 7.13. Bit error rate test setup.

This scheme is just a form of "out-of-service" testing [7.1]. The PRBSG generates the necessary test pattern for transmission over the link. The effect of channel noise is evaluated by injection of Gaussian noise, at baseband level, into the in-phase and quadrature symbol channels of the transmitter. One independent noise source is used per channel, to prevent correlation between in-phase and quadrature noise components. The relationship between signal and noise amplitudes per baseband channel and effective transmission link carrier-to-noise ratio is described in Chapter 4 §3.2.

The detector circuit performs an estimate of the value of the received symbol and converts this into a binary data stream. The error detector circuit compares these bits to replicas of the original data bits (from the PRBSG) with an EXCLUSIVE-OR gate, which only gives a logical "high" output when the two compared bits are different ie. when there is an error in the decoded bit. A frequency counter (set to a long-period count — a maximum of 10 seconds) evaluates the frequency of errors. The outputs of the counter are averaged and divided by the system bit rate to arrive

at an average system bit error rate. Given a data rate of 120 kbps, this period allows 1.2×10^6 bits to be transmitted. Assuming that the simulation results of Chapter 5 §6 set an upper bound on practical system noise performance (ie. the hardware system will perform only as well as or worse than the simulation model), the expected lower limit on bit error rate is approximately 10^{-4}. This means that over a 10-second averaging period, *at least* 120 bit errors will be registered. Most sources consider 100 error events to be sufficient to arrive at an error rate estimate of sufficiently small statistical variance [7.1, p. 139], [7.2].

Table 7.1. lists the measured results, over the (effective) CNR range from 1 to 11 dB.

TABLE 7.1. Measured bit error rate results for CORPSK(4-7, 1+D)

CNR [dB]	S/N [·]	σ_n [V]	f_{error} [kHz]	P(e)
1	1.12	2.12	13.8	1.2×10^{-1}
2	1.26	1.88	11.0	9.0×10^{-2}
3	1.41	1.68	12.6	1.1×10^{-1}
4	1.58	1.50	9.48	7.9×10^{-2}
5	1.78	1.33	7.28	6.1×10^{-2}
6	2.00	1.18	5.40	4.5×10^{-2}
7	2.24	1.06	2.31	1.9×10^{-2}
8	2.51	0.94	1.42	1.2×10^{-2}
9	2.82	0.84	0.53	4.4×10^{-3}
10	3.16	0.75	0.23	1.9×10^{-3}
11	3.55	0.67	0.076	6.3×10^{-4}

These results are plotted in Fig. 7.14 and compared to the results simulated for the ideal correlation receiver [7.2], and the symbol-by-symbol detection method simulated in Chapter 5 §6.

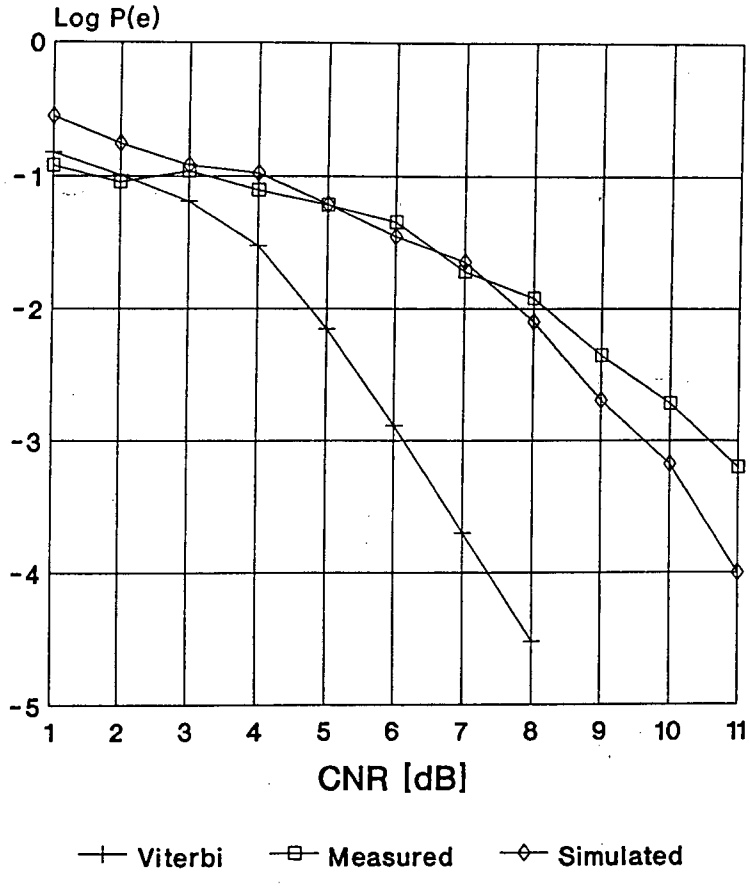


Fig. 7.14. Measured bit error rates compared to correlation receiver and symbol-by-symbol detection simulation.

It should be borne in mind that the simulation of Chapter 5 models directly the hardware system tested here — the hardware phase pulse shape is, in fact, derived from the simulation program. The simulation results set an upper bound on actual system performance — any degradation in the measured results are because of hardware imperfections. The results above indicate that, at most, 1 dB degradation occurs in practice. Compared to the ideal Viterbi (correlation) receiver, up to 4.5 dB CNR penalty is incurred, but a far less complex receiver implementation has been achieved, employing only symbol-by-symbol detection. These bit error rate results are comparable to the performance of 4 PSK [7.1, p. 196].

A cause of the CNR degradation is the relatively poor performance of the modulator digital-to-analog converters (resulting in reduced vertical eye opening). The relatively short tails of the (theoretically infinitely-long) phase pulse causes small discontinuities in the phase trajectory, and hence in the I and Q channels. The transmission filters tend to smooth these out, but a degraded trajectory portion will result. The fact that a degradation of 2 dB is introduced in the VEO by the D/A process (§ 7.2.1) suggests that the measured error rates are actually optimistic by about 1 dB. This would be because of the finite crest factor of the injected noise, as well as the fact that the noise source is only approximately white.

REFERENCES

[7.1] Feher, K. and the Engineers of Hewlett-Packard, *Telecommunications Measurements, Analysis, and Instrumentation*, Prentice-Hall, Englewood Cliffs, N.J., 1987.

[7.2] D. Muilwijk, "*Correlative Phase Shift Keying — A Class of Constant Envelope Modulation Techniques,*" IEEE Trans. Commun., Vol. COM-29, No. 3, March 1981, pp. 226-236.

[7.3] T. Maseng, "*Digitally Phase Modulated (DPM) Signals,*" IEEE Trans. Commun., Vol. COM-33, No. 9, September 1985, pp. 911-918.

CHAPTER 8
CONCLUSIONS

Because of their high immunity to nonlinear signal operations, constant-envelope modulation schemes are attractive for use in digital radio environments. Traditional systems such as OQPSK and MSK have high out-of-band signal levels. Filtering of these signals is ineffective because any further nonlinear processing causes spectral spreading.

Partial response signaling (PRS) has been used successfully to improve the spectral properties of PAM systems. This thesis investigation studied the effect of PRS on frequency- and phase-modulated carrier systems, in particular on their spectral performance and their maintenance of constant envelope.

A literature survey has been conducted, showing the development of constant-envelope modulation formats that employ some form of correlative encoding. The most promising systems to date are those that employ Continuous Phase Frequency Shift Keying (CPFSK) with some form of baseband pulse shaping. Systems that fall into this category are Continuous Phase Modulation (CPM) and Correlative Phase Shift Keying (CORPSK).

CORPSK has been studied as a typical scheme from this class. Computer simulations were performed to analyse spectral behaviour, time domain characteristics and the performance of synchronization systems. These results agreed with those reported in the literature.

A reduced-complexity CORPSK modem was developed, employing symbol-by-symbol detection instead of the Viterbi algorithm. The results indicate that a good, but suboptimum, solution is possible using this approach.

Extensive testing was performed to determine system performance. Results show that a spectral efficiency of only 0.5 bps/Hz is

achieved. However, this modulation format has no spectral sidelobes, and so the out-of-band signal levels are much lower than those yielded by systems such as unfiltered QPSK. Measured error rate characteristics are comparable to the performance of a 4 PSK system.

Carrier recovery is straightforward — a peak-to-peak jitter of about 4% is achieved with fairly simple circuitry. Symbol timing recovery, while possible, requires more sophisticated synchronization, and was not attempted. However, the existence of suitable spectral components was verified.

The effect of hardlimited amplification on the system spectrum was simulated. No spectral sidelobes were generated, demonstrating this system's immunity to spectral spreading. The in-depth study of actual system performance under nonlinear conditions (bandpass limiting, TWTA nonlinearity, and even multipath) is recommended for further research.

BIBLIOGRAPHY

Feher, K., *Digital Communications, Satellite-Earth Station Engineering,* 1983, Prentice-Hall, Englewood Cliffs, N.J.

Feher, K., *Digital Communications, Microwave Applications,* 1987, Prentice-Hall, Englewood Cliffs, N.J.

Feher, K. and the Engineers of Hewlett-Packard, *Telecommunications Measurements, Analysis, and Instrumentation,* Prentice-Hall, Englewood Cliffs, N.J., 1987.

Feher, K., *Advanced Digital Communications Systems and Signal Processing Techniques,* 1987, Prentice-Hall, Englewood CLiffs, N.J.

Franks, L. E., *"Carrier and Bit Synchronization in Data Communications — A Tutorial Review,"* IEEE Trans. Commun., Vol. COM-28, No. 8, August 1980, pp. 1107-1121.

Gagliardi, Robert M., *Satellite Communications,* 1984, Van Nostrand Reinhold Company, N.Y.

Horowitz, P. and Hill, W., *The Art of Electronics,* 1983, Cambridge University Press.

Lathi, B. P., *Modern Digital and Analog Communication Systems,* 1983, Holt-Saunders.

Lowdon, E., *Practical Transformer Design Handbook,* Second Edition, Howard W. Sams & Co., Inc., 1980.

National Semiconductor, *Data Conversion/Acquisition Data Book,* 1984.

National Semiconductor, *Linear Data Book,* 1984.

Saal, R., *Handbook of filter Design,* 1979, AEG-Telefunken.

Snijders, Verhoeckx, Van Essen and Van Gerwen, *"Digital Generation of Linearly Modulated Data Waveforms,"* IEEE Trans. Commun., Vol. COM-23, No. 11, November 1975, pp. 1259-1270.

Sklar, B., *"A Structured Overview of Digital Communications — A Tutorial Review — Part 1,"* IEEE Communications Magazine, August 1983, pp. 4-17.

Sklar, B. *"A Structured Overview of Digital Communications — A Tutorial Review — Part 2,"* IEEE Communications Magazine, October 1983, pp. 6-21.

Stremler, F. G., *Introduction to Communication Systems,* Second Edition, Addison-Wesley, 1982.

Williams, *Electronic Filter Design Handbook,* 1981, McGraw-Hill.

Ziemer and Peterson, *Digital Communications and Spread Spectrum Systems,* 1985, McMillan.

APPENDIX A

THE EFFECT OF HARDLIMITING ON CARRIER SYSTEM PERFORMANCE

APPENDIX A

THE EFFECT OF HARD LIMITING ON CARRIER SYSTEM PERFORMANCE

Hard limiting is often used in, for example, satellite transponders as signal conditioners prior to power amplification, to prevent large amplitude swings. The usual approach is to follow the hard limiter by a bandpass filter, tuned to the carrier frequency concerned. This configuration is referred to as a "bandpass limiter" (BPL). Any modulation schemes used in such systems must obviously exhibit as high an immunity to such operations as possible. Signals with nonconstant envelopes, such as filtered QPSK or MSK (filtered to reduce out-of-band signal levels), usually experience spectral spreading as well as coupling between in-phase and quadrature channels.

Presented here is an analysis of the performance of bandpass limiters (from Gagliardi [A.1]). Fig. A.1 shows the bandpass limiter model.

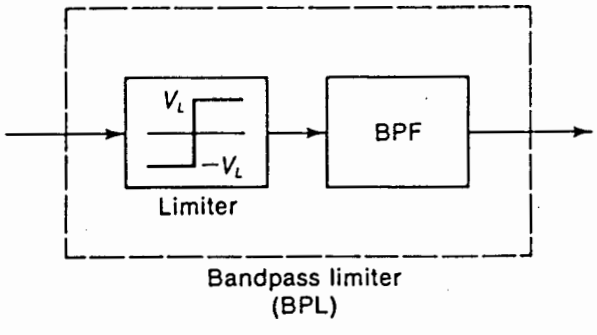


Fig. A.1. Bandpass Limiter model

A.1 Effect of BPL on System CNR

The limiter is a simple threshold device, with output voltage limits $+V_L$ and $-V_L$, respectively.

Given that the input is a carrier and noise combined, in the form

$$x(t) = \alpha(t)\cos(\omega_c t + \theta(t)) \qquad (A.1)$$

the BPL output is then given by

$$y(t) = \left(\frac{4V_L}{\pi} \right) \cos(\omega_c t + \theta(t)) \qquad (A.2)$$

The effect of the BPL is clear — namely to remove the amplitude variations in the RF signal, while preserving phase. The envelope amplitude is proportional to the limiter level V_L and so the limiter can provide a convenient and accurate power adjustment. The output power is

$$P_L = \frac{8V_L^2}{\pi^2} \qquad (A.3)$$

It can be shown that the carrier power at the BPL output is also given by

$$P_{co} = \left(\frac{2V_L^2}{\pi} \right) CNR_i e^{-CNR_i} \left[I_0\left(\frac{CNR_i}{2} \right) + I_1\left(\frac{CNR_i}{2} \right) \right]^2 \qquad (A.4)$$

$I_0(x)$ and $I_1(x)$ are first and second order imaginary Bessel functions and CNR_i is the input carrier-to-noise ratio of the BPL, such that

$$CNR_i = \frac{P_c}{N_0 B_{RF}} \qquad (A.5)$$

with $N_0 B_{RF}$ the input channel noise power. The output noise power is the difference between P_L, the total output power (A.3), and the carrier output power, P_{co}, in (A.4). Hence

$$P_{no} = P_L - P_{co} \qquad (A.6)$$

yielding an output BPL CNR of

$$CNR_{BL} = \frac{P_{co}}{P_{no}} \qquad (A.7)$$

$$= \frac{P_{co} / P_L}{1 - (P_{co} / P_L)} \qquad (A.8)$$

From (A.4),

$$\frac{P_{co}}{P_L} = \left(\frac{\pi}{4}\right) CNR_i e^{-CNR_i} \left[I_0\left(\frac{CNR_i}{2}\right) + I_1\left(\frac{CNR_i}{2}\right) \right]^2 \qquad (A.9)$$

Fig. A.2 shows a plot of the (normalized) ratio

$$\Gamma = \frac{CNR_{BL}}{CNR_i} \qquad (A.10)$$

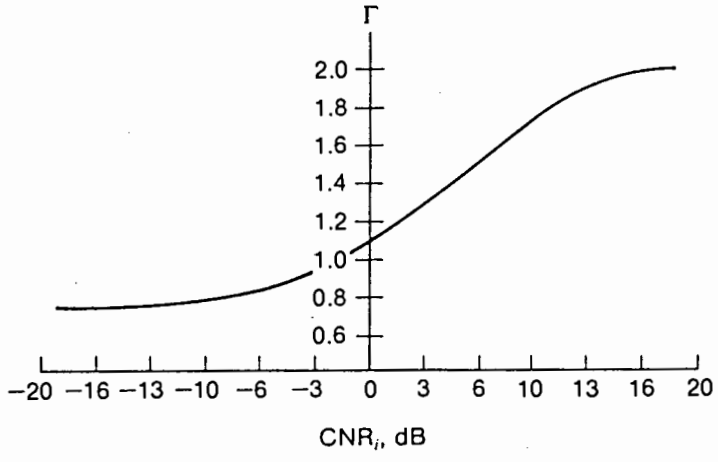


Fig. A.2. Output/Input CNR vs. input CNR for BPL

The curve illustrates how, for low CNR of the BPL input signal, little or no change occurs between input and output CNR's. However, for fairly large input CNR's, the BPL actually "cleans up" the signal and a CNR gain of up to 2 dB results. This altering of the CNR can be seen as a modification of the carrier and noise powers at the output. Carrier and noise *suppression* factors, α_s^2 and α_n^2, can be defined according to

$$\begin{aligned} P_{co} &= \alpha_s^2 P_c \\ P_{no} &= \alpha_n^2 (N_0 B_{RF}) \end{aligned} \qquad (A.11)$$

From (A.7),

$$CNR_{BL} = \left(\frac{\alpha_s^2}{\alpha_n^2} \right) CNR_i \qquad (A.12)$$

The suppression factor ratio, Γ, is then defined as

$$\Gamma = \frac{\alpha_s^2}{\alpha_n^2} \qquad (A.13)$$

This is just the ratio plotted in Fig. A.2, and so the suppression factors themselves are seen to be a function of the input CNR_i. From (A.6) and (A.11),

$$P_L = \alpha_s^2 P_c + \alpha_n^2 (N_0 B_{RF}) \qquad (A.14)$$

From (A.13), the carrier suppression factor can be written as

$$\alpha_s^2 P_c = P_L \left(\frac{CNR_{BL}}{1 + CNR_{BL}} \right) \qquad (A.15a)$$

Similarly, the noise suppression factor is given by

$$\alpha_n^2 (N_0 B_{RF}) = P_L \left(\frac{1}{1 + CNR_{BL}} \right) \qquad (A.15b)$$

These suppression factors therefore divide the available output power between signal and noise, in the ratios defined in (A.15).

A.2 Effect of BPL on System Spectrum

An RF signal such as QPSK may be represented as a quadrature carrier of the form

$$c(t) = A\, m_c(t) \cos(\omega_c t + \psi) + A\, m_s(t) \sin(\omega_c t + \psi) \qquad (A.16)$$

The BPL output is again given by (A.2), with

$$\theta(t) = \tan^{-1}\left[\frac{m_s(t)}{m_c(t)} \right] \qquad (A.17)$$

Writing the BPL output, (A.2), as a quadrature signal,

$$y(t) = f_c(t) \cos(\omega_c t + \psi) + f_s(t) \sin(\omega_c t + \psi) \qquad (A.18)$$

it can be deduced that the envelope amplitude is given by

$$[f_c^2(t) + f_s^2(t)]^{1/2} = \frac{4V_L}{\pi} \tag{A.19}$$

and

$$\frac{m_s(t)}{m_c(t)} = \frac{f_c(t)}{f_s(t)} \tag{A.20}$$

A solution requires that

$$f_c(t) = \frac{m_c(t)}{[m_c^2(t) + m_s^2(t)]^{1/2}} \left[\frac{4V_L}{\pi}\right] \tag{A.21a}$$

$$f_s(t) = \frac{m_s(t)}{[m_c^2(t) + m_s^2(t)]^{1/2}} \left[\frac{4V_L}{\pi}\right] \tag{A.21b}$$

Both $f_c(t)$ and $f_s(t)$ depend on both data bit sequences $m_c(t)$ and $m_s(t)$, and so the effect of the BPL is seen to be to introduce undesirable coupling between the in-phase and quadrature components. If the value of the in-phase data channel, $m_c(t)$, changes bit sign, obviously $f_c(t)$ will change sign, too. However,

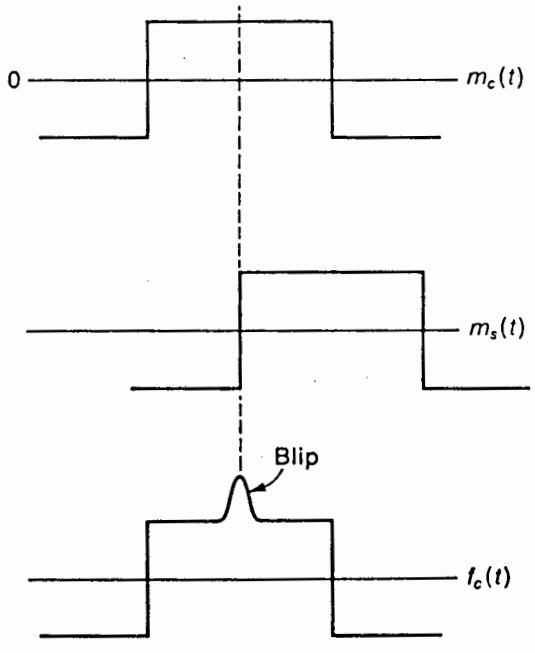


Fig. A.3. Blip development in bandpass-limited OQPSK

during the bit transition, $m_c(t)$ passes through zero, and so (A.21b) shows that $f_s(t)$ also reflects the change, as a perturbation or "blip" — see Fig. A.3. This can lead to errors in detection of the affected channel.

In the case of QPSK, all bit transitions occur simultaneously. As a result, bit errors are not so significant, but the existence of blips in the waveforms leads to bandwidth increases in the bandpass-limited carrier.

Fig. A.4 shows the effect of bandpass limiting on filtered QPSK and OQPSK [A.2], while Fig. A.5 shows the effect for MSK.

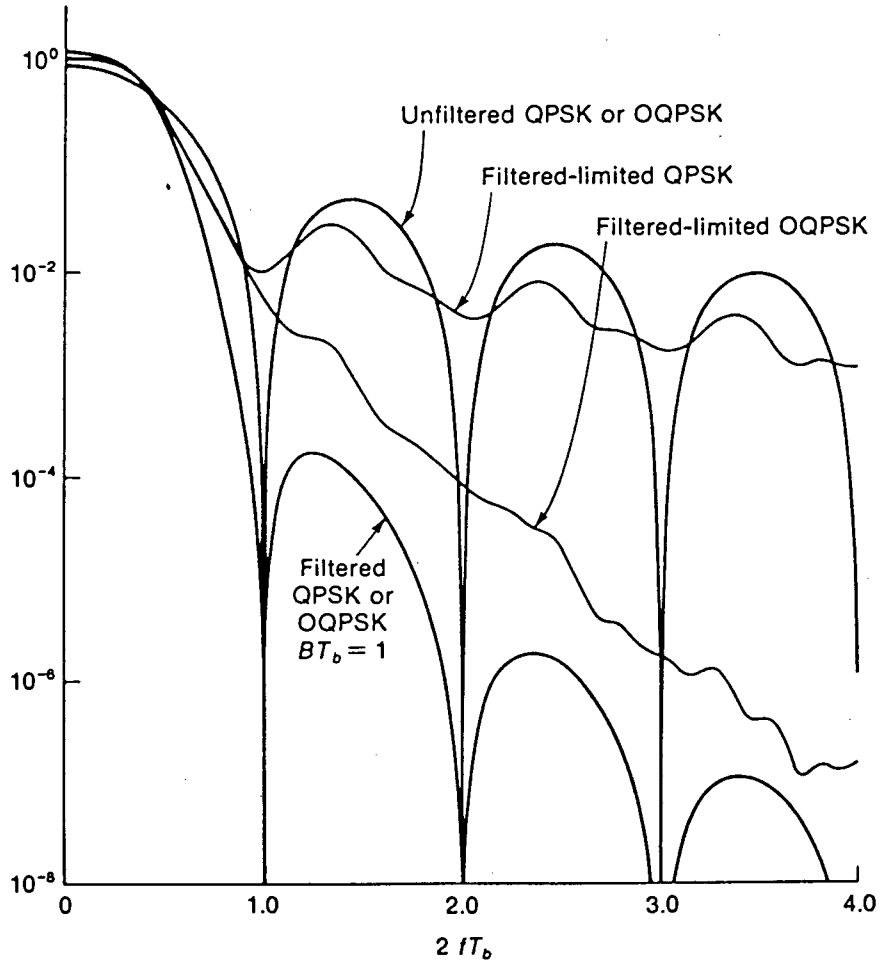


Fig. A.4. Spectral restoration due to hard limiting of filtered QPSK and OQPSK

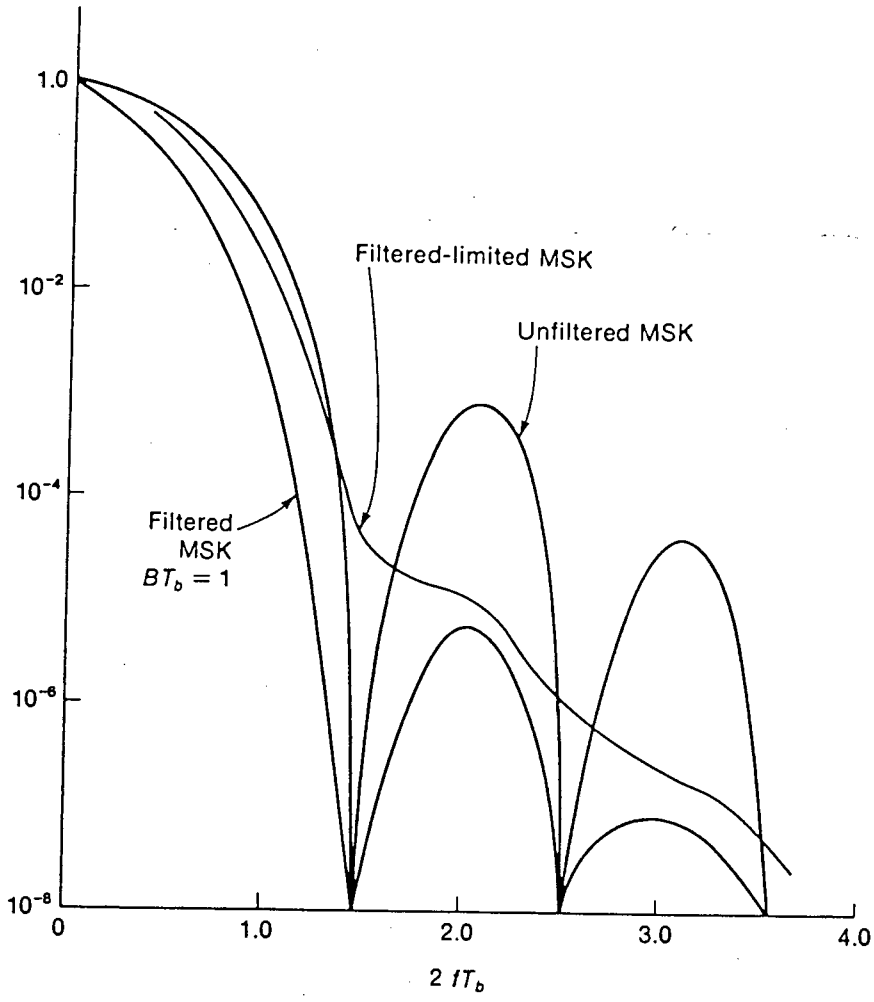


Fig. A.5. Spectral restoration due to hard limiting of filtered MSK

Filtering of the RF signals clearly causes reduction of the out-of-band signal levels, but hardlimiting tends to restore the spectral tails to their prefiltered values, also destroying their spectral nulls. In the case of MSK, the main spectral lobe that has been reduced by filtering has been almost completely restored. If these carriers had been filtered to satisfy spectral masks defining spectral occupancy, hardlimiting makes them unsuitable for use from there on.

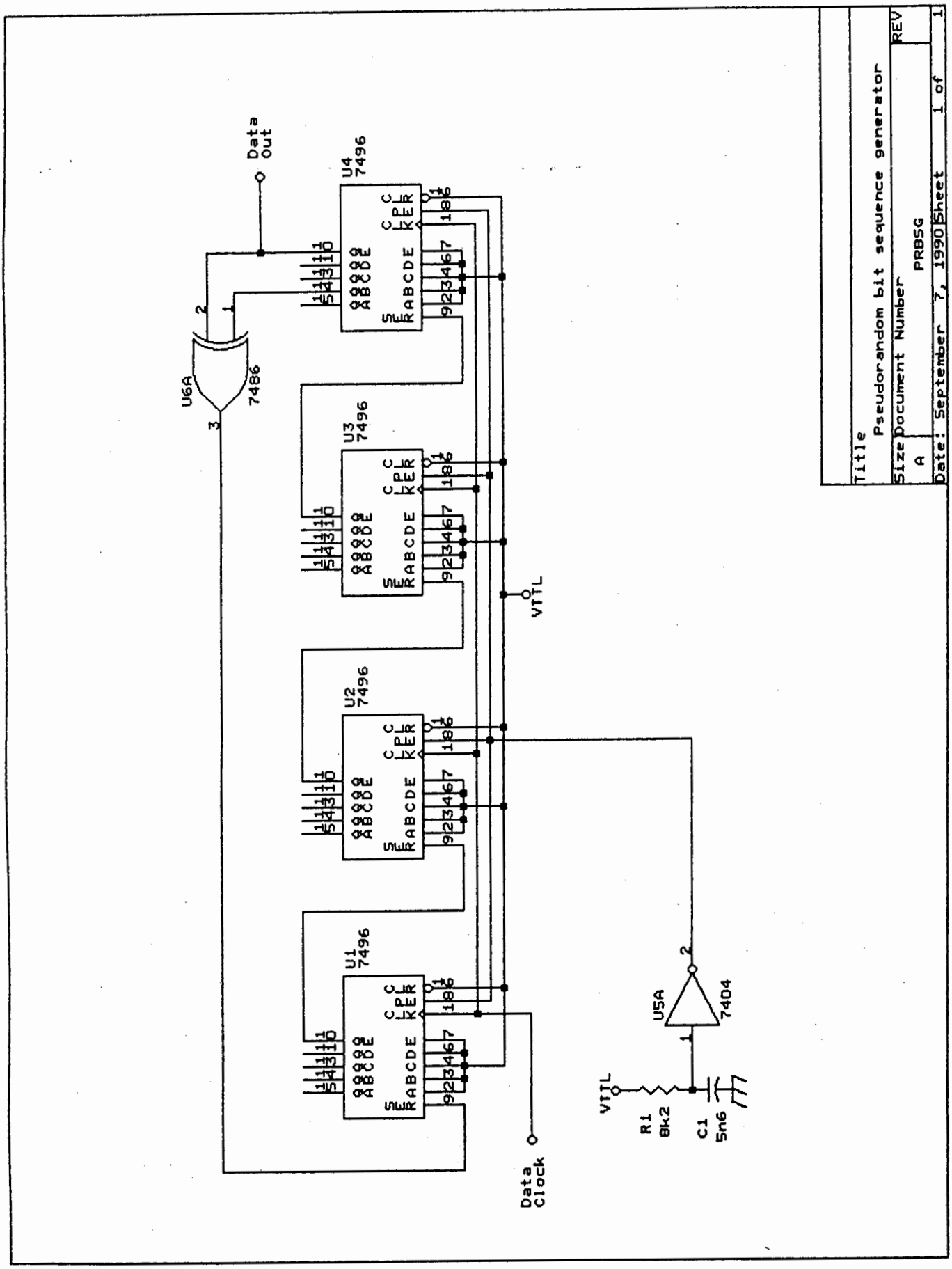
REFERENCES

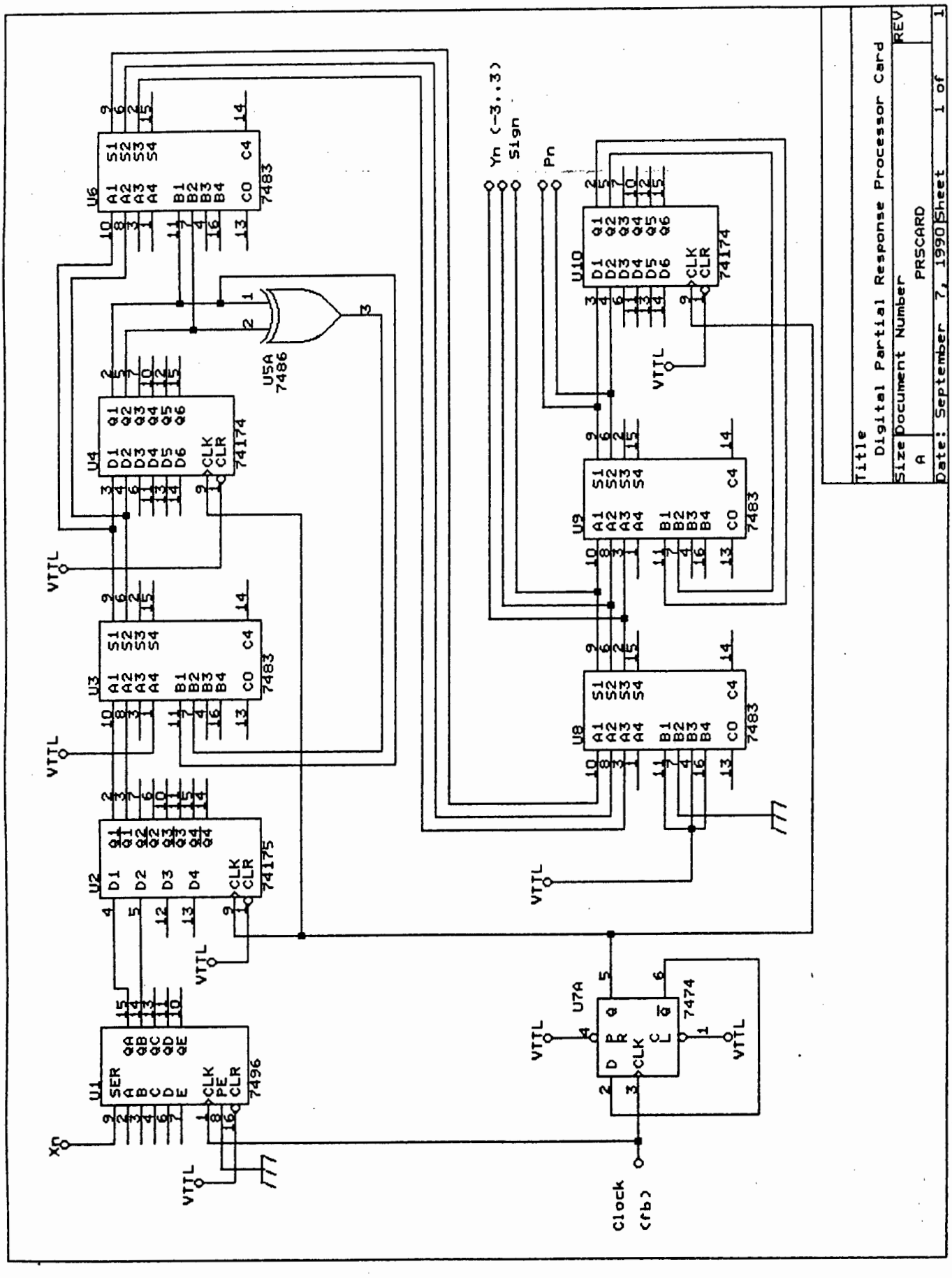
[A.1] Robert M. Gagliardi, *Satellite Communications,* Van Nostrand Reinhold Company, 1984, Chapter 4.

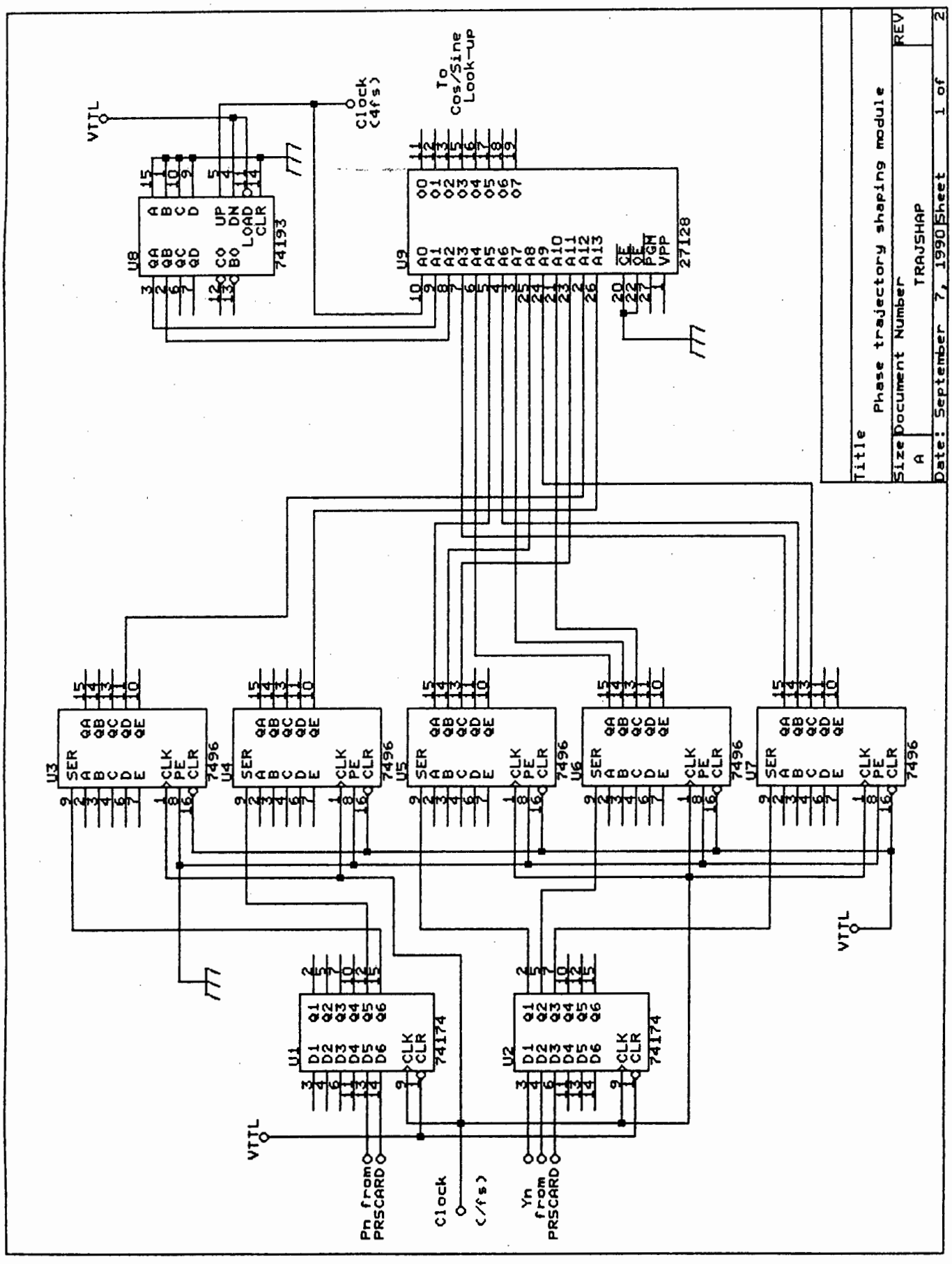
[A.2] D. Morais and K. Feher, *"The Effects of Filtering and Limiting on the Performance of QPSK and MSK Systems,"* IEEE Trans. Commun., Vol. COM-28, No. 12, December 1980, pp. 1999-2009.

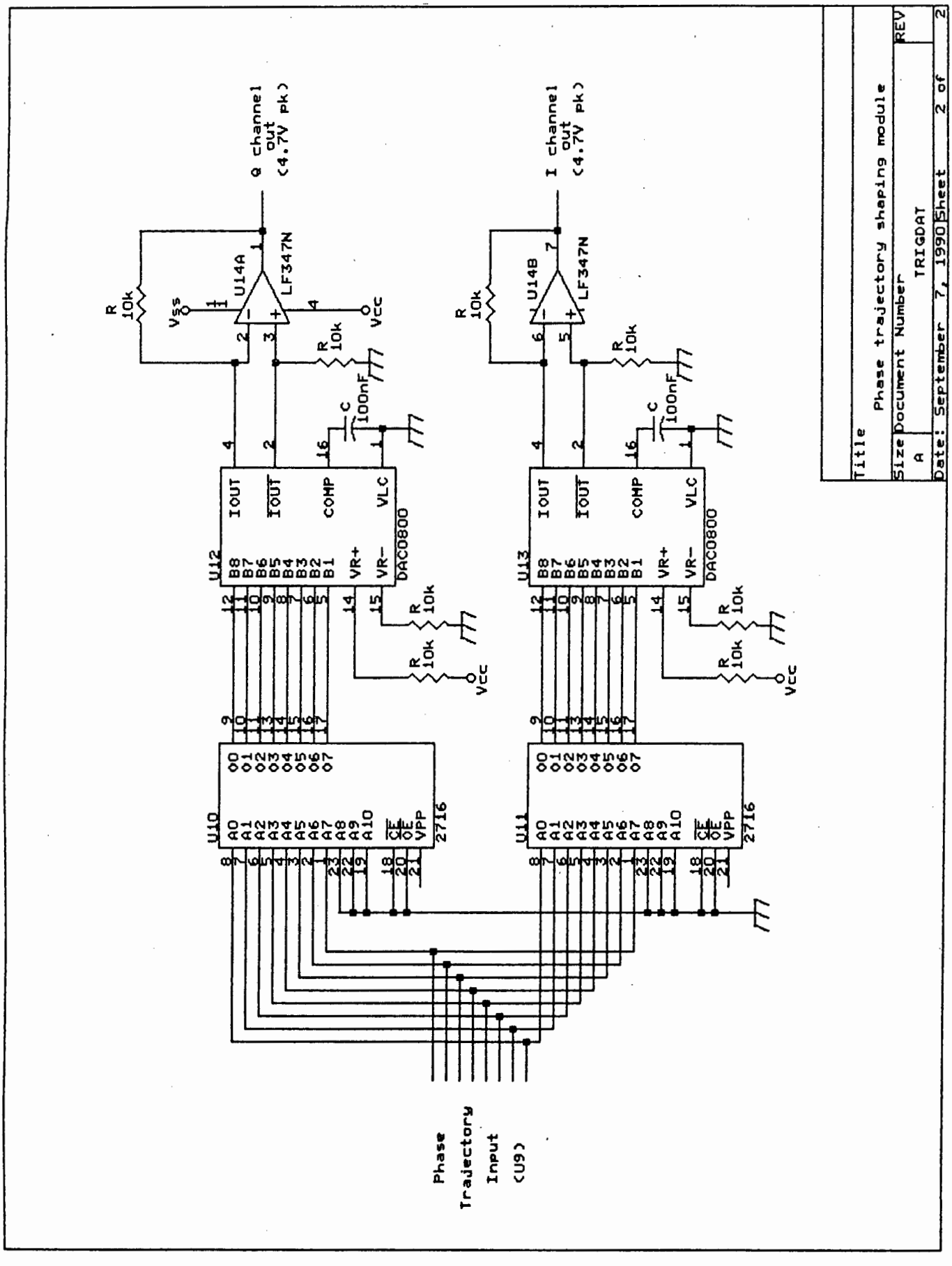
APPENDIX B

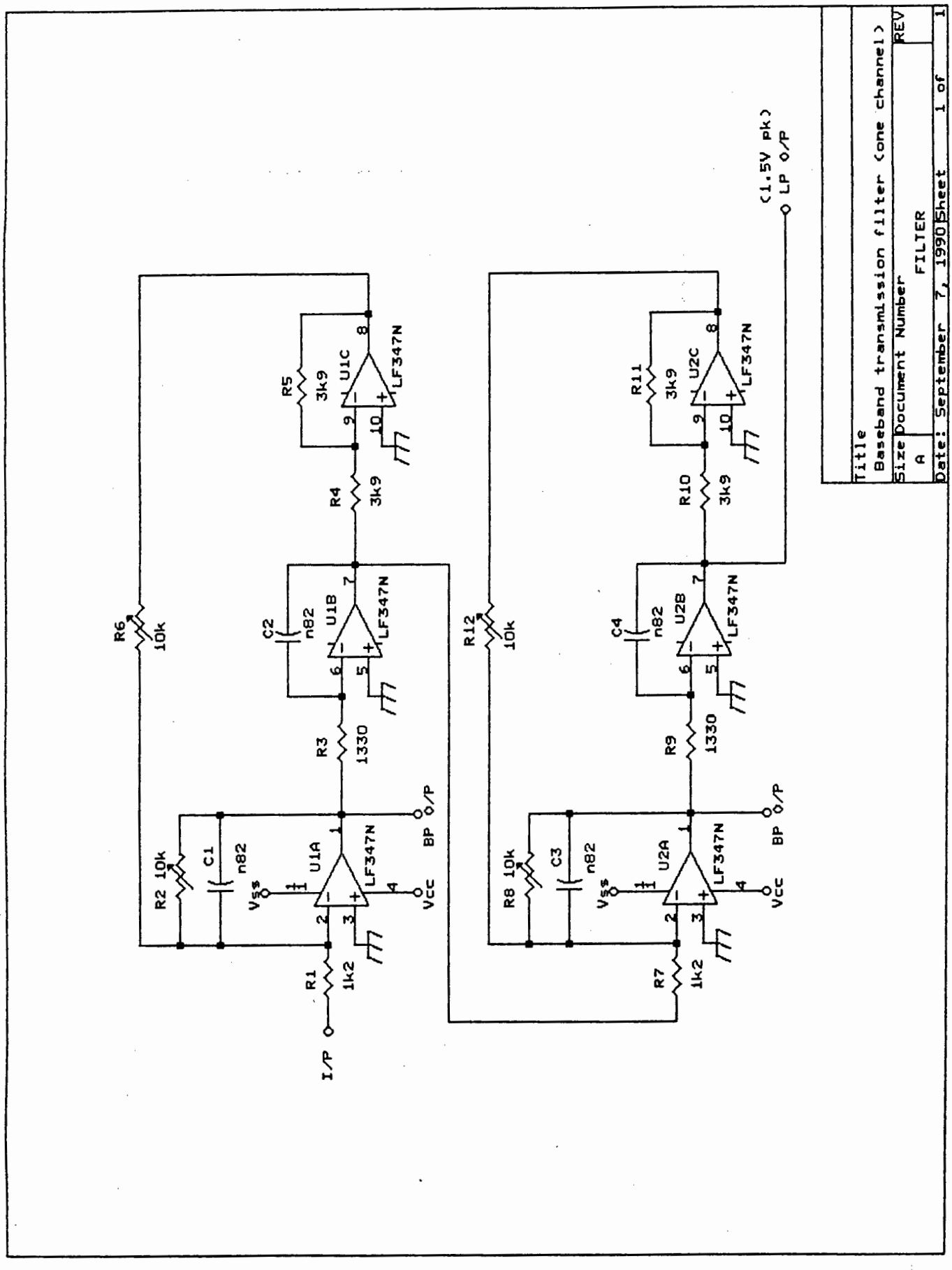
MODEM CIRCUIT DIAGRAMS

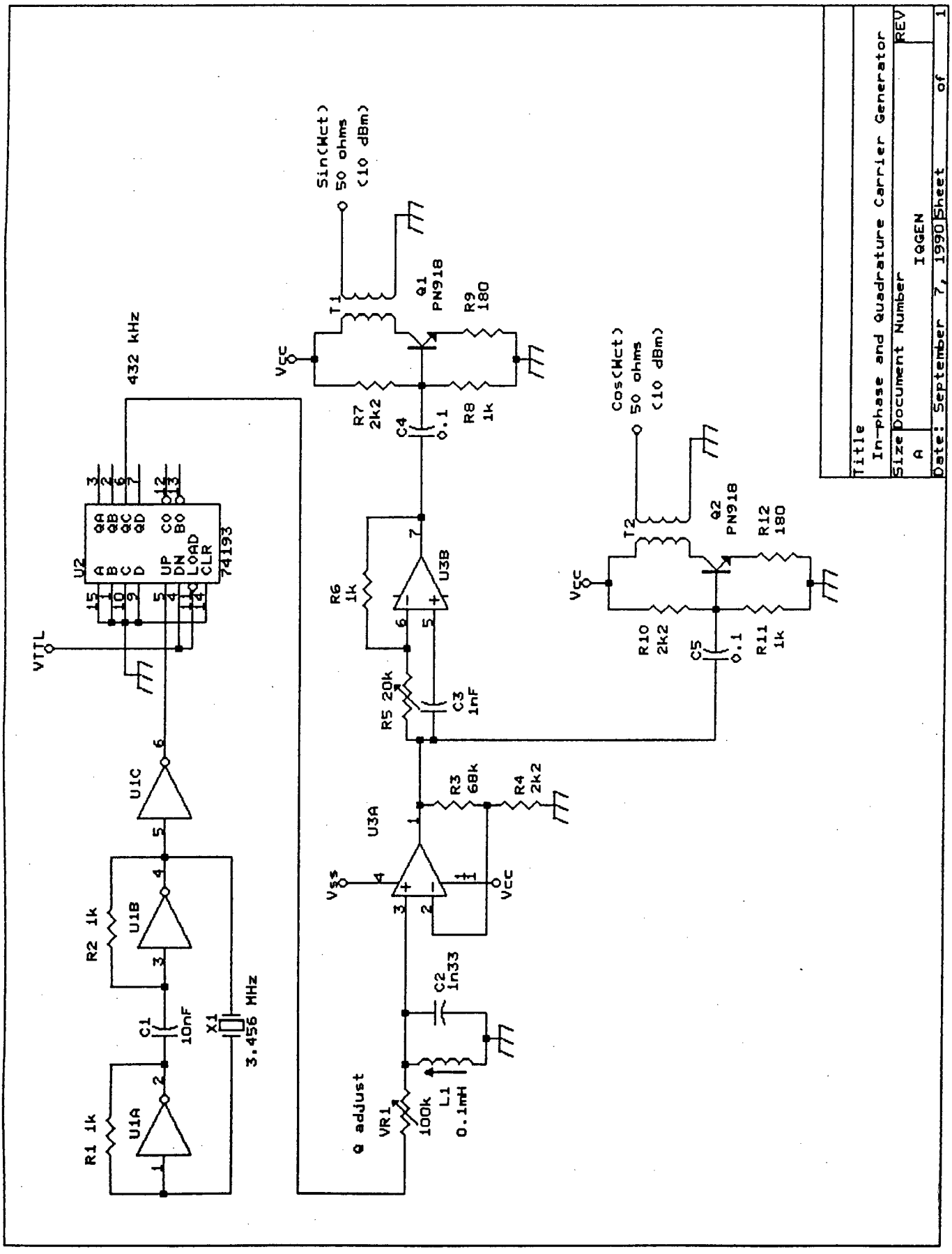

Data Out
U6A
7486
U4
7496
U3
7496
U2
7496
U1
7496
VTTL
Data Clock
U5A
7404
R1
8k2
C1
5n6
Title
Pseudorandom bit sequence generator
Size
A
Document Number
PRBSG
REV
Date: September 7, 1990
Sheet 1 of 1

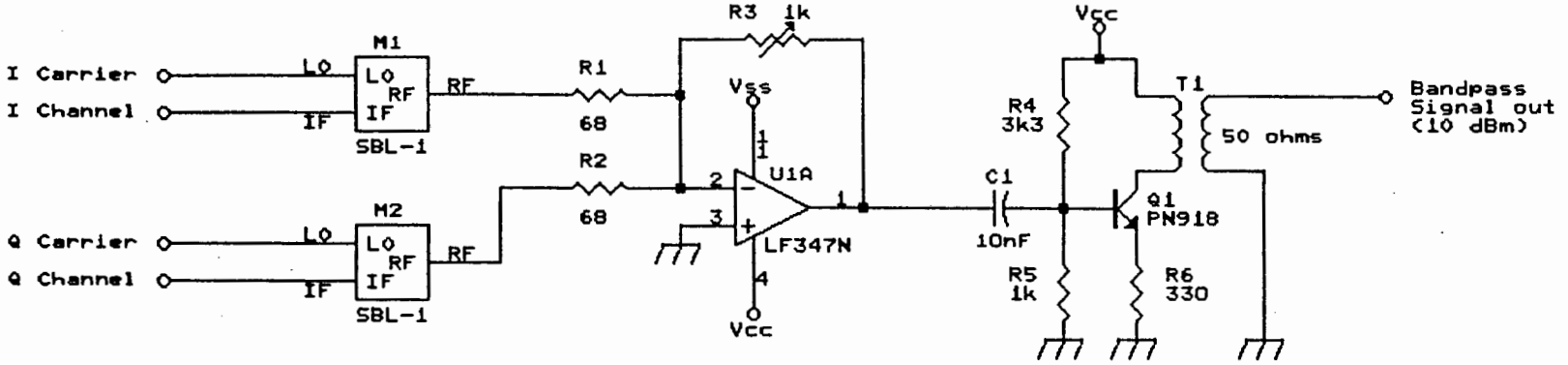

Title
Digital Partial Response Processor Card
Size
A
Document Number
PRSCARD
REV
Date: September 7, 1990
Sheet
1 of
1
Yn (-3..3)
Sign
Pn
Xn
Clock
(fb)
VTTL
U1
7496
U2
74175
U3
7483
U4
74174
U5A
7486
U6
7483
U7A
7474
U8
7483
U9
7483
U10
74174


VTTL
Clock (4fs)
To Cos/Sine Look-up
U8 74193
U9 27128
U1 74174
U2 74174
U3 7496
U4 7496
U5 7496
U6 7496
U7 7496
Pn from PRSCARD
Clock (/fs)
Yn from PRSCARD
Title
Phase trajectory shaping module
Size A
Document Number TRAJSHAP
REV
Date: September 7, 1990 Sheet 1 of 2

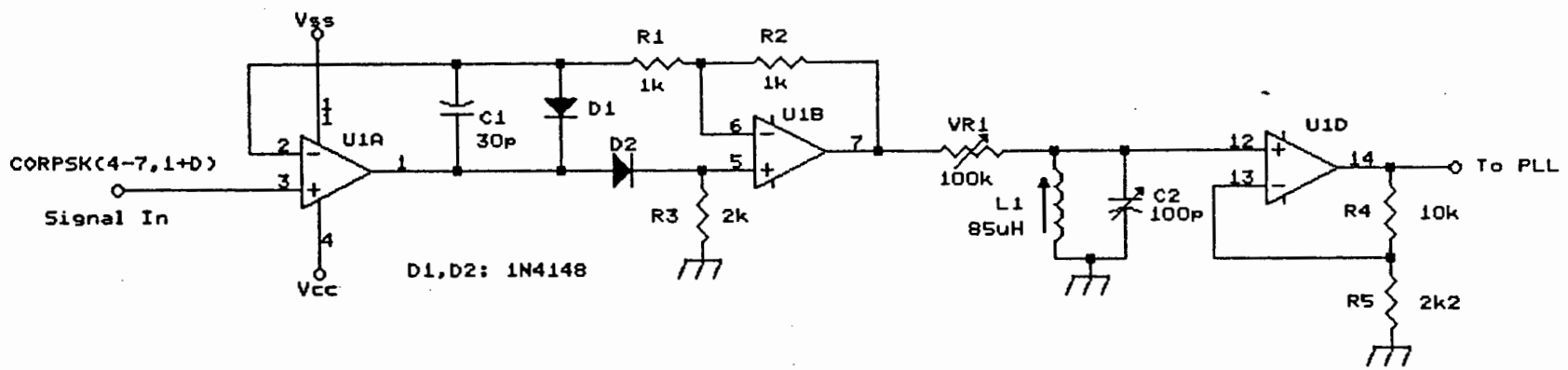

Q channel out (4.7V pk)
I channel out (4.7V pk)
U14A
U14B
LF347N
R 10k
Vss
Vcc
C 100nF
U12
U13
DAC0800
IOUT
IOUT
COMP
VLC
VR+
VR-
B8
B7
B6
B5
B4
B3
B2
B1
U10
U11
2716
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
CE
OE
VPP
O0
O1
O2
O3
O4
O5
O6
O7
Phase Trajectory Input (U9)
Title
Phase trajectory shaping module
Size
A
Document Number
TRIGDAT
REV
Date: September 7, 1990
Sheet 2 of 2


Title
Baseband transmission filter (one channel)
Size
A
Document Number
FILTER
REV
Date: September 7, 1990
Sheet 1 of 1
I/P
R1 1k2
R2 10k
C1 n82
R3 1330
C2 n82
R4 3k9
R5 3k9
R6 10k
U1A LF347N
U1B LF347N
U1C LF347N
Vss
Vcc
BP O/P
R7 1k2
R8 10k
C3 n82
R9 1330
C4 n82
R10 3k9
R11 3k9
R12 10k
U2A LF347N
U2B LF347N
U2C LF347N
(1.5V pk)
LP O/P

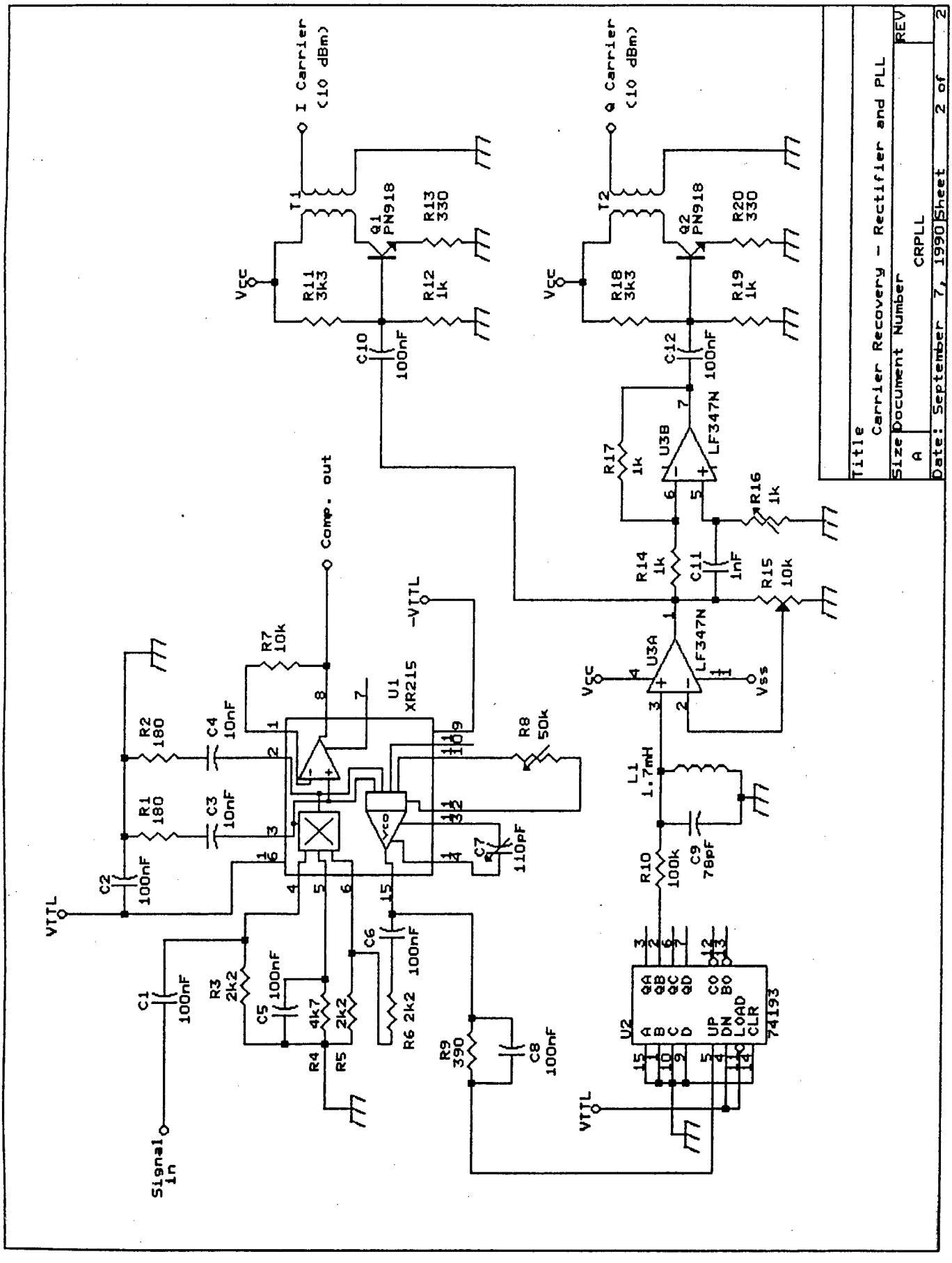

Sin(Wct)
50 ohms
(10 dBm)
Cos(Wct)
50 ohms
(10 dBm)
T1
T2
Q1
PN918
Q2
PN918
R9
180
R12
180
R7
2k2
R8
1k
R10
2k2
R11
1k
C4
0.1
C5
0.1
Vcc
Vcc
U2
74193
QA
QB
QC
QD
CO
BO
A
B
C
D
UP
DN
LOAD
CLR
432 kHz
VTTL
U1A
U1B
U1C
R1 1k
R2 1k
C1
10nF
X1
3.456 MHz
U3A
U3B
R3
68k
R4
2k2
R5 20k
C3
1nF
R6
1k
Vss
Vcc
C2
1n33
L1
0.1mH
VR1
100k
Q adjust
Title
In-phase and Quadrature Carrier Generator
Size
A
Document Number
IQGEN
REV
Date: September 7, 1990
Sheet
of
1

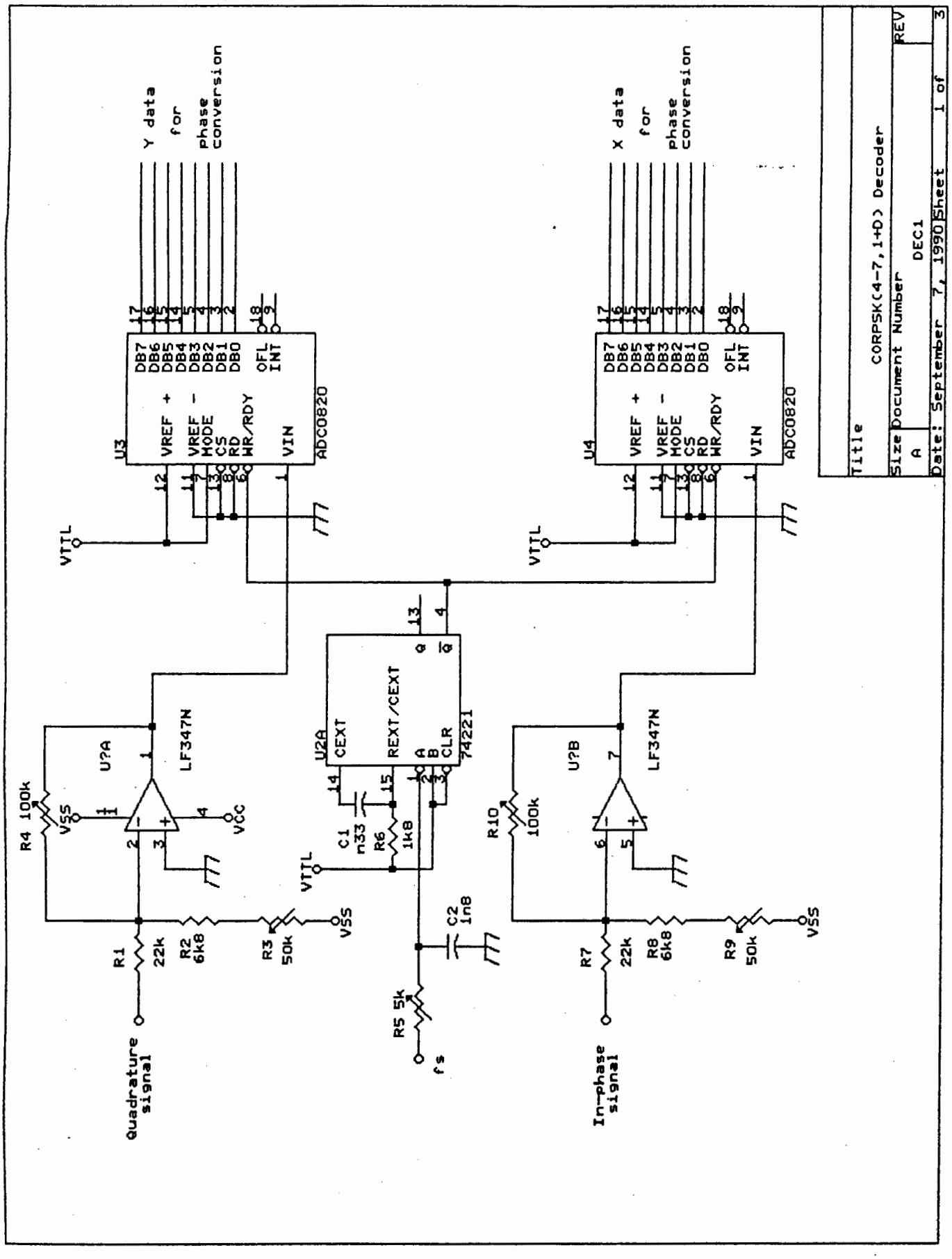


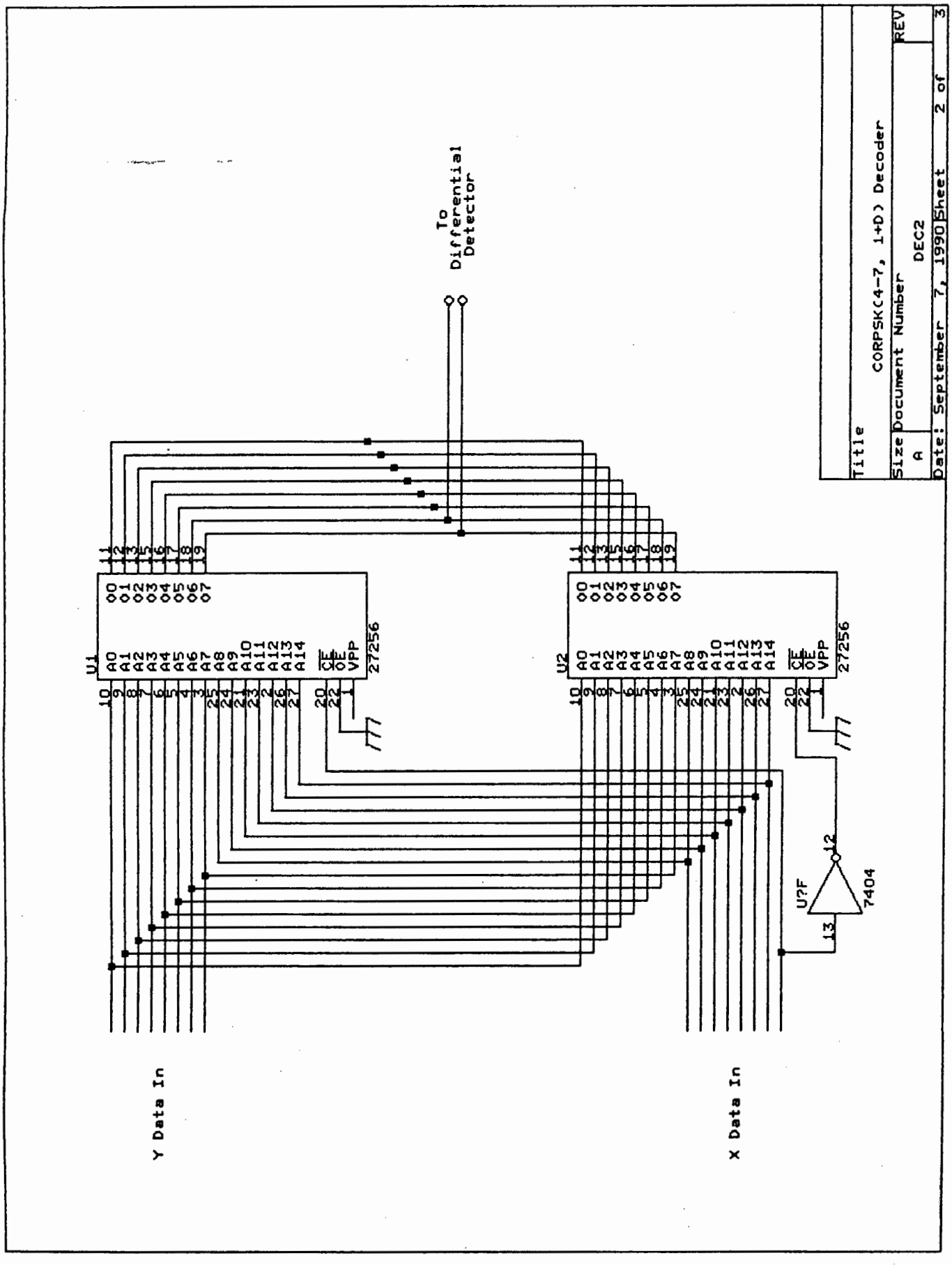
Title: CORPSK(4-7, 1+D) modulator and driver		
Size: A	Document Number: MIXSUM	REV
Date: September 7, 1990	Sheet 1 of 1	

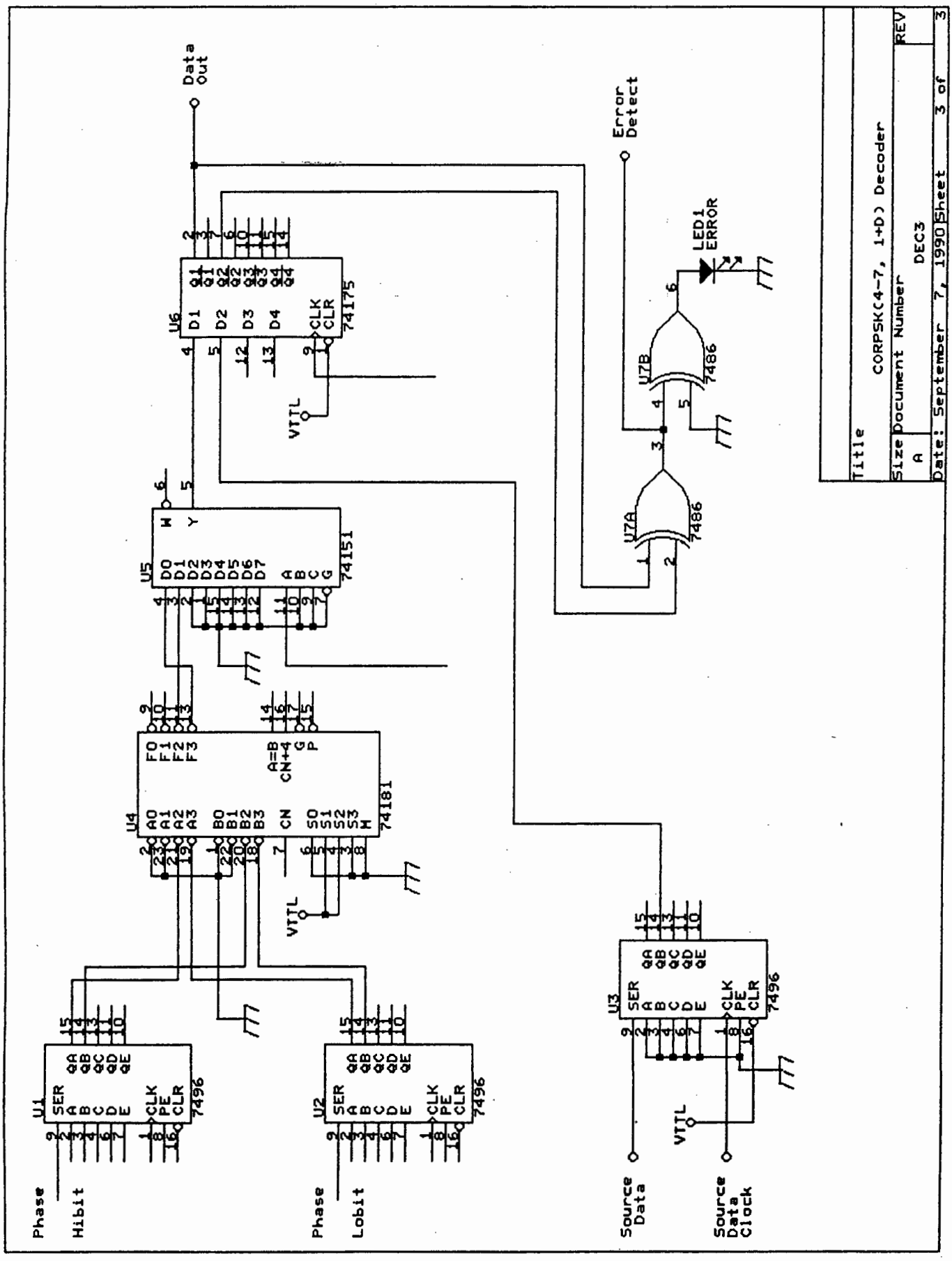


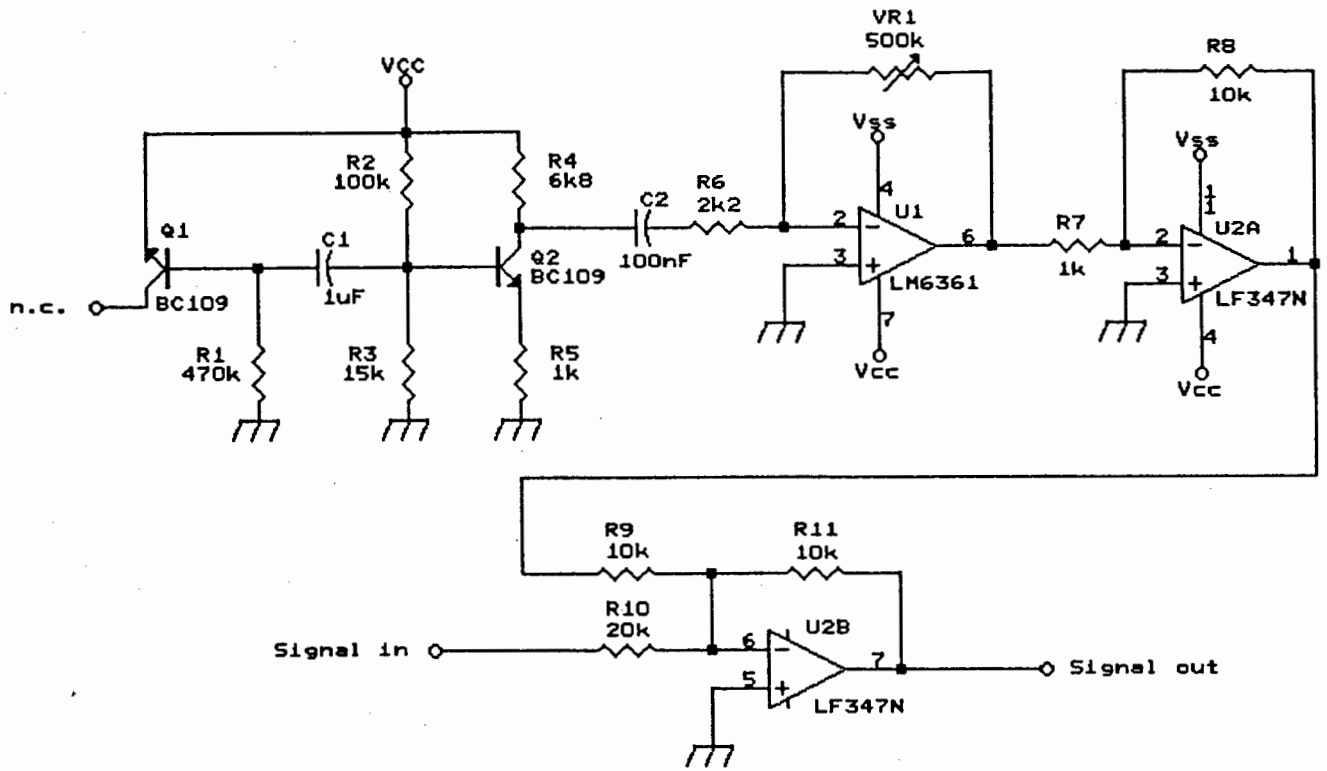
Title: Carrier Recovery - Rectifier and PLL		
Size: A	Document Number: CRCCT	REV
Date: September 7, 1990	Sheet 1 of 2	


I Carrier
(10 dBm)
Q Carrier
(10 dBm)
T1
T2
Q1 PN918
Q2 PN918
R13 330
R20 330
R11 3k3
R12 1k
R18 3k3
R19 1k
Vcc
C10 100nF
C12 100nF
U3B LF347N
R17 1k
R16 1k
R14 1k
C11 1nF
R15 10k
U3A LF347N
Vss
Comp. out
-VTTL
R7 10k
U1 XR215
R8 50k
R2 180
C4 10nF
R1 180
C3 10nF
C2 100nF
VTTL
C7 110pF
L1 1.7mH
C9 78pF
R10 100k
C1 100nF
R3 2k2
C5 100nF
4k7
2k2
C6
R6 2k2 100nF
R4
R5
R9 390
C8 100nF
Signal In
U2 74193
QA QB QC QD
A B C D
UP DN LOAD CLR
CO BO
Title
Carrier Recovery - Rectifier and PLL
Size A
Document Number
CRPLL
REV
Date: September 7, 1990
Sheet 2 of 2


Y data for phase conversion
X data for phase conversion
U3
U4
ADC0820
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
INT
VREF +
VREF -
MODE
CS
RD
WR/RDY
VIN
VTTL
U2A
74221
CEXT
REXT/CEXT
CLR
U?A
U?B
LF347N
R1 22k
R2 6k8
R3 50k
R4 100k
R5 5k
R6 1k8
R7 22k
R8 6k8
R9 50k
R10 100k
C1 n33
C2 1n8
VSS
VCC
fs
Quadrature signal
In-phase signal
CORPSK(4-7,1+D) Decoder
Document Number
DEC1
REV
Size
A
Date: September 7, 1990
Sheet 1 of 3


To
Differential
Detector
U1
U2
27256
U?F
7404
Y Data In
X Data In
Title
CORPSK(4-7, 1+D) Decoder
Size
A
Document Number
DEC2
REV
Date: September 7, 1990
Sheet
2 of
3


Data Out
Error Detect
LED1 ERROR
U6 74175
U5 74151
U4 74181
U7A 7486
U7B 7486
U1 7496
U2 7496
U3 7496
VTTL
Phase Hibit
Phase Lobit
Source Data
Source Data Clock
Title
CORPSK(4-7, 1+D) Decoder
Size A
Document Number DEC3
REV
Date: September 7, 1990 Sheet 3 of 3



Title: Noise Generator / Summer (One channel only)		
Size: A	Document Number: NOISEGEN	REV
Date: September 7, 1990	Sheet 1 of 1	

APPENDIX C

MATHCAD LISTINGS

CREATE THE FREQUENCY PULSE SHAPE

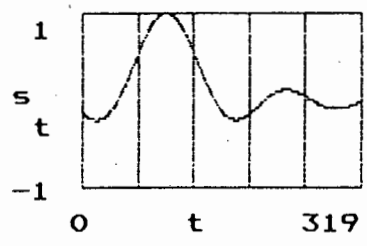
$\alpha := 1.0 \qquad \beta := 112$

$t := 0 \,..\, 319$

$$Wt_t := \frac{t - 96}{\beta} \cdot 2 \cdot \pi$$

$$s_t := \frac{\sin\left[Wt_t\right]}{Wt_t} \qquad s_{96} := 1 \quad \text{(Remove singularity)}$$

Sinx/x waveshape

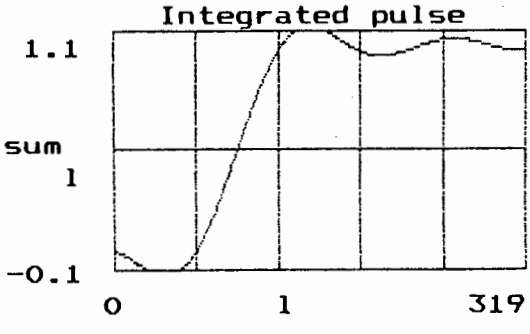


$sum_0 := 0$

$l := 1 \,..\, 319 \qquad scale := 0.01880$

$sum_l := \left[s_l \cdot scale\right] + sum_{l-1}$

Integrated pulse



****

CREATE THE DATA STREAM

$d := 0 \,..\, 140 \qquad k := 0 \,..\, 8$

$b_d := ceil(rnd(3) - 0.5)$ Random input sequence

$i := 2 \,..\, 140$

$x_i := \left[b_i + b_{i-1}\right] - 3 \qquad x_k := 0 \qquad y_0 := 0$ Duobinary Rule

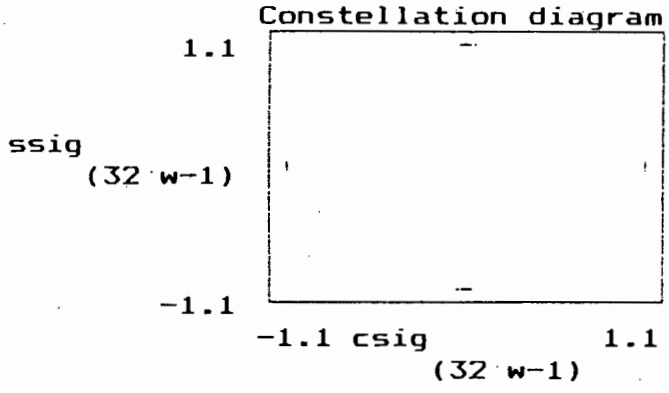
$y_i := y_{i-1} + x_i$ Present/next state

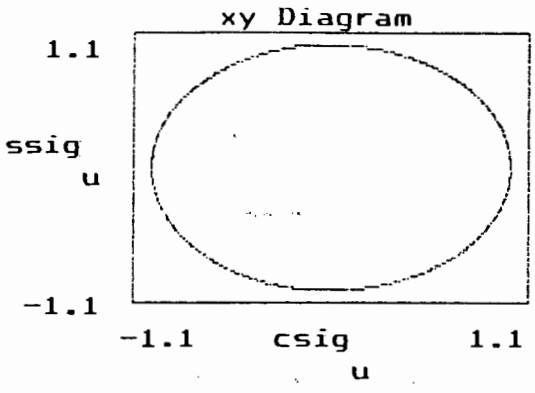
****

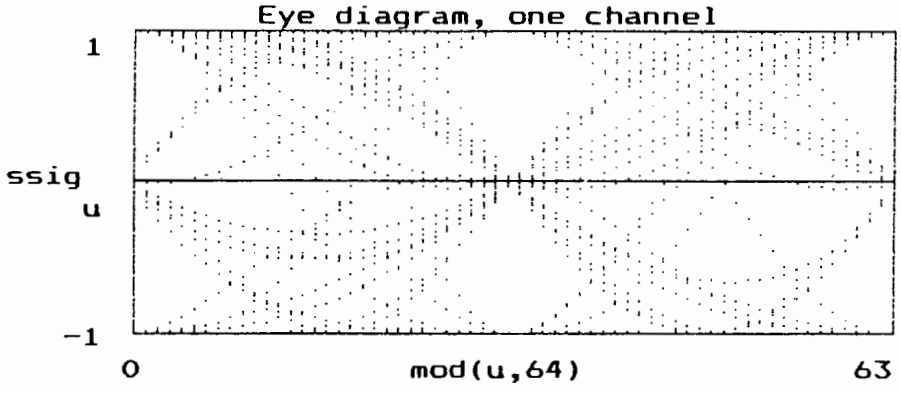
CREATE THE PHASE TRAJECTORY

$L := 32$ Interpolation factor

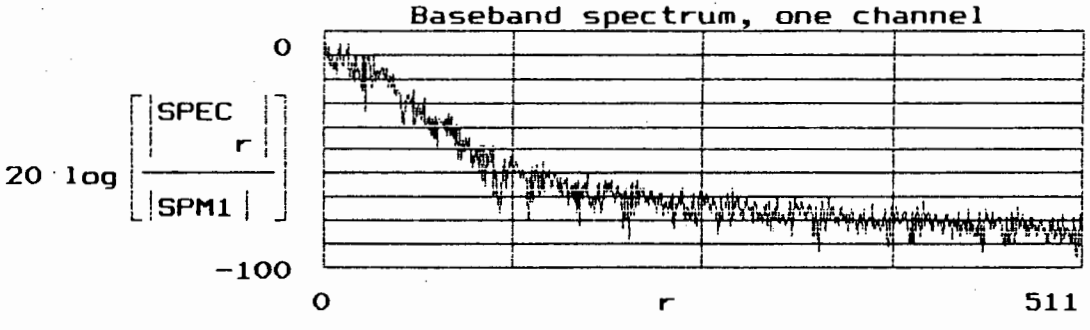
$m := 0 \,..\, 5 \cdot L - 1 \qquad path_m := sum_{(2 \cdot m)+1} \qquad k := 1 \,..\, 16$







SPEC := cfft(csig) r := 0 ..1023 SPM1 := max(SPEC)



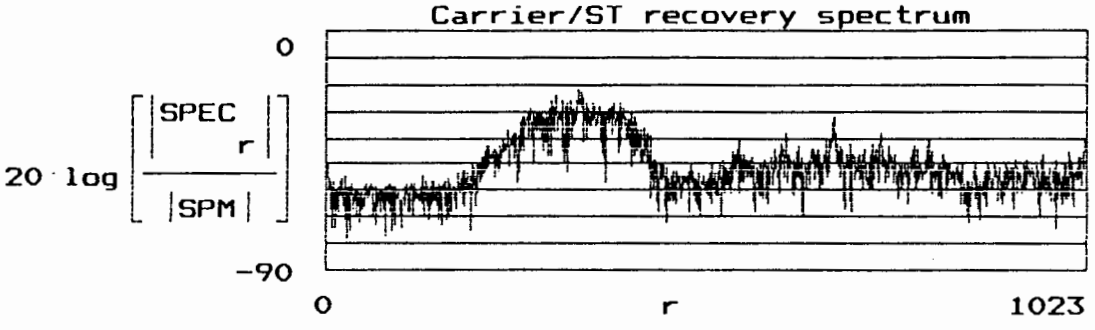
****

****

CREATE THE BANDPASS SIGNAL AND SIMULATE CARRIER / SYMBOL TIMING RECOVERY

$RFSIG_u := \left| \left[RFCI_u \cdot csig_u \right] + \left[RFCQ_u \cdot ssig_u \right] \right|$ Absolute value function simulates nonlinearity

$SPEC := cfft(RFSIG)$ $r := 0 \ldots 1023$ $SPM := \max(SPEC)$



APPENDIX D

PASCAL PROGRAM LISTINGS

NOISE SIMULATION PROGRAM

The Pascal program "Noise_Sim" generates baseband in-phase and quadrature CORPSK(4-7, 1+D) signals and simulates the effect of additive Gaussian channel noise. The signal is generated and decoded in the same way as in the hardware implementation; decoding is on a symbol-by-symbol basis. A running total of errors is kept, and error probabilities calculated for a range of signal-to-noise ratios. Simulation output is to a data file.

PHASE TRAJECTORY ROM CREATION PROGRAM

This program is not listed here, as it is based on the routine "Count_Traj" of the noise simulation program. All possible input combinations of State_minus_3, Yn_minus_2, Yn_minus_1, Yn_minus_0 and Int_Counter are generated in parallel as an integer ranging from 0 to 16383. The variable Int_Counter forms the least significant part of the counter word, as this is used to step through the eight interpolation points of that symbol period.

The program "Decoder" is listed on page D.8.

```
Program Noise_Sim;
{*****************************************************************}
{*  Generates baseband I and Q CORPSK(4-7, 1+D) signals and       *}
{*  simulates the effect of additive Gaussian channel noise.     *}
{*  The signal is generated and decoded in the same way as in    *}
{*  the hardware; decoding is on a symbol-by-symbol basis. A     *}
{*  running total of errors is kept, and P(e) is calculated for  *}
{*  a range of signal-to-noise ratios.                           *}
{*****************************************************************}

Uses CRT;

Const   Pi              =       3.141592654;

Var     Ch:                     Char;
        Pathfile:               Text;
        ROMWord,
        Mem_Counter:            Longint;
        Int_counter:            Integer;
        Yn_minus_2,
        Yn_minus_1,
        Yn_minus_0,
        State_minus_3:          Integer;
        ROMValue:               Real;
        Patharray:              Array[0..39] of real;
        ROMByte:                Byte;
        Bytefile:               File of Byte;
        Continue:               Boolean;
        i,
        j,
        Byte3back,
        Byte2back,
        Oldbyte,
        Newbyte,
        Oldprec,
        Prec,
        OldPRS,
        PRSOut,
        Noise_Counter,
        Noise_Channel,
        EBoverN0,
        Delta,
        Decoded_symbol,
        Error_Count,
        Count_Cycles,
        Symbol_Count:           LongInt;
        Noise_value:            Array[1..2] of Real;
        Noise_scale,
        PathValue,
        Cossig,
        Sinesig,
        Angle,
        NewPhase,
        OldPhase,
        Errorprob:              Real;
```

```
        Asciival:           String;
        Noisefile:          Text;
(*=============================================================*)
Procedure Initialise_Variables;        { Clear variables        }

Begin
  Continue:= true;                     { Loop control variable.  }
  Yn_minus_2:= 0;                      { Last-but-one PRS value. }
  Yn_minus_1:= 0;                      { Last seven-PRS value.   }
  Yn_minus_0:= 0;                      { Current seven-PRS value.}
  State_minus_3:= 0;                   { Phase three symbols ago.}
  Oldbyte:= 0;                         { Last four-level input   }
  Newbyte:= 0;                         { New input word variable.}
  Oldprec:= 0;                         { Last precoder value.    }
  Prec:= 0;                            { New precoder output.    }
  OldPRS:= 0;                          { Last seven-PRS value.   }
  PRSOut:= 0;                          { New seven-PRS value.    }
  Noise_Counter:= 0;                   { Noise  control variable.}
  Noise_Channel:= 0;                   { Noise control variable. }
  Delta:= 0;                           { Temp. phase difference  }
  Decoded_symbol:= 0;                  { Value of received word. }
  Error_Count:= 0;                     { Number of errors.       }
  Errorprob:= 0;                       { Computed P(e).          }
  Symbol_Count:= 0;                    { Number of symbols sent. }
  Count_Cycles:= 0;                    { Loop control variable.  }
  Noise_scale:= 0;                     { Noise magnitude scaler. }
  PathValue:= 0;                       { Phase trajectory sample.}
  Cossig:= 0;                          { In-phase signal.        }
  Sinesig:= 0;                         { Quadrature signal.      }
  Angle:= 0;                           { Temporary phase value.  }
  NewPhase:= 0;                        { New phase value.        }
  OldPhase:= 0;                        { Last phase value.       }
  For i:= 0 to 40 do
  Begin
    Patharray[i]:= 0;                  { Clear pulse shape array.}
  End;
End;
(*=============================================================*)

Procedure Open_Bytefile;

Begin
  Assign( Bytefile, 'ROMTRAJ.DAT');
  Rewrite( Bytefile );
End;
(*=============================================================*)

Procedure Read_disk;                   { Read in phase pulse.    }

Var       Value:            Real;
          Valcode,
          Pointcount:       Integer;
          Pathword:         String;

Begin
  Assign(Pathfile, 'PRSTRAJ2.DAT');
```

```
  Reset(Pathfile);
  For Pointcount:= 0 to 39 do
  Begin
    Readln(Pathfile, Pathword);
    Val(Pathword, Value, Valcode);
    Patharray[Pointcount]:= Value;
  End;
  Close(Pathfile);
End;
(*=============================================================*)

Procedure Open_Datafile;                    { Open simulation O/P file}
                                            { output values.          }
Begin
  Assign( Noisefile, 'Noisedat.dat' );
  Rewrite( Noisefile );
End;
(*=============================================================*)

Procedure Close_Datafile;                   { Closes output data file.}

Begin
  Close( Noisefile );
End;
(*=============================================================*)

Procedure New_byte;                         { Generates four-level I/P}
                                            { and records last three. }
Begin
  Byte3back:= Byte2back;
  Byte2back:= Oldbyte;
  Oldbyte:= Newbyte;
  NewByte:= Round(Random(4));
End;
(*=============================================================*)

Procedure Precode;                          { Performs precoding to   }
                                            { prevent error propo-    }
                                            { gation at the receiver. }
Begin
  Oldprec:= Prec;
  If Newbyte + Oldprec >= 0 then Prec:= (Newbyte + Oldprec) mod 4
  Else
  Prec:= (Newbyte + Oldprec + 4) mod 4;
End;
(*=============================================================*)

Procedure PRSFilter;                        { Performs 1+D addition of}
                                            { precoder outputs.       }

Begin
  OldPRS:= PRSOut;
  PRSOut:= Prec - Oldprec;
End;
(*=============================================================*)
```

```
Procedure Addnoise;                          { Generate Gaussian values}
                                             { for each channel.       }

Begin
  Noise_scale:= 0.5*exp(-(EBoverN0 * (ln(10))/20));
  For Noise_Channel:= 1 to 2 do
  Begin
  Noise_value[Noise_Channel]:= 0;
    For Noise_Counter:= 1 to 12 do
    Begin
      Noise_value[Noise_Channel]:= Noise_value[Noise_Channel] +
        Random;
    End;
    Noise_value[Noise_Channel]:= Noise_value[Noise_Channel] - 6;
  End;
  Cossig:= Cossig + (Noise_scale * Noise_Value[1]);
  Sinesig:= Sinesig + (Noise_Scale * Noise_value[2]);

End;
(*=============================================================*)

Procedure Count_Traj;                        { Creates  trajectory for }
                                             { each channel, eight     }
                                             { samples per symbol.     }

Begin
  State_minus_3:= State_minus_3 + Yn_minus_2;
  Yn_minus_2:= Yn_minus_1;
  Yn_minus_1:= Yn_minus_0;
  Yn_minus_0:= PRSOut;
  For Int_Counter:= 0 to 7 do
  Begin
    PathValue:= State_minus_3 + (Yn_minus_0 *
      Patharray[Int_counter]) +
      (Yn_minus_1 * Patharray[Int_counter + 8]) +
      (Yn_minus_2 * Patharray[Int_counter + 16]);
    Cossig:= Cos(Pathvalue * Pi/2);
    Sinesig:= Sin(Pathvalue * Pi/2);
  End;
  Addnoise;
End;
(*=============================================================*)

Procedure Detect;                            { Converts I and Q sample }
                                             { values to polar format  }
                                             { to yield signal phase.  }

Begin
  OldPhase:= NewPhase;
  If (Cossig=0) and (Sinesig>0) then NewPhase:= Pi/2
  Else
  If (Cossig=0) and (Sinesig<0) then NewPhase:= 3*Pi/2
  Else
  If (Cossig=0) and (Sinesig=0) then NewPhase:= 0
  Else
  If (Cossig<0) and (Sinesig=0) then NewPhase:= Pi
```

```
  Else
  If (Cossig>0) and (Sinesig=0) then NewPhase:= 0
  Else
  Begin
    Angle:= Arctan(Abs(Sinesig)/Abs(Cossig));
    If (Cossig > 0) and (Sinesig > 0) then NewPhase:= Angle
    Else
    If (Cossig > 0) and (Sinesig < 0) then NewPhase:= (2*Pi)
      - Angle
    Else
    If (Cossig < 0) and (Sinesig < 0) then NewPhase:= Pi + Angle
    Else
    NewPhase:= Pi - Angle;
  End;
End;
(*=============================================================*)

Procedure Decode;                          { Compares phase shifts   }
                                           { to give PRS value of Tx }
                                           { and decodes it into a   }
                                           { four-level data stream. }

Begin
  Delta:= Round((NewPhase - OldPhase) * 2 / Pi);
  If ( Delta - 1 ) >= 0 then
  Decoded_symbol:= (Delta - 0) mod 4
  Else
  Decoded_symbol:= (Delta - 0 + 8) mod 4;
  Gotoxy( 1,24 );
  Write( Byte3back, '    ', Decoded_Symbol );
End;
(*=============================================================*)

Procedure Count_Errors;                    { Compares received word  }
                                           { with the original value,}
                                           { counts the number of    }
                                           { errors and computes the }
                                           { symbol error rate.      }

Begin
  Count_Cycles:= Count_Cycles + 1;
  If Count_Cycles > 5 then
  Begin
    Symbol_Count:= Symbol_Count + 1;
    If Decoded_symbol <> Byte3back then
      Error_Count:= Error_Count + 1;
    Errorprob:= Error_Count/Symbol_Count;
    Write( '    ', Symbol_Count, '   ', Error_Count,
      '   ', Errorprob );
  End;
End;
(*=============================================================*)

Begin                                      { ***  MAIN PROGRAM  *** }
  Randomize;
  Clrscr;
```

```
  Read_Disk;                              { Read in phase pulse     }
  Open_Datafile;                          { Open output data file   }
  For EBOverN0:= 1 to 10 do               { Compute for Eb/No range }
  Begin
    Initialise_Variables;                 { Set up program variables}
    Gotoxy( 1,24 );
    ClrEol;
    Gotoxy( 1,10 );
    Write( 'Eb/No = ', EBOverN0 );        { Display current Eb/No.  }
    While Error_Count < 101 do            { Register 100 errors for }
    Begin                                 { each Eb/No value.       }
      New_byte;                           { Generate a new input.   }
      Precode;                            { Perform the precoding   }
      PRSFilter;                          { Implement PRS function. }
      Count_traj;                         { Do signal trajectories  }
                                          { and add noise.          }
      Detect;                             { Rect. to pol. conversion}
                                          { of corrupted signal.    }
      Decode;                             { Compare phase values,   }
                                          { decode into data stream.}
      Count_errors;                       { Compare received and    }
    End;                                  { sent data, compute P(e) }
    Str( Errorprob, Asciival );
    Writeln( Noisefile, Asciival );       { Write P(e) to file.     }
  End;
  Close_datafile;                         { Close output data file. }
End.
```

```
Program Decoder;
{*************************************************************}
{*  Generates table for EPROM to convert from rectangular to    *}
{*  polar coordinates (phase only). The signal space is modelled*}
{*  as a 256 x 256 matrix. The signal phasor falls on one of    *}
{*  65536 points - the EPROM creates a phase angle for the      *}
{*  decoder to extract the differentially-encoded symbol        *}
{*  information. The phase output ranges from 0 to 255. Since   *}
{*  two EPROM chips are used in the hardware, the program       *}
{*  creates two byte files, each of 32768 bytes - each is       *}
{*  chosen by altering program parameters and running again.    *}
{*************************************************************}
Uses DOS, CRT;

Var
   Filevar:          Text;
   Line:             String[3];
   J,
   K,
   I:                Longint;
   Code,
   Value:            Integer;
   Word:             Byte;
   Bytefile:         File of byte;
   Ch:               Char;

{=============================================================}
Procedure Assign_Filename;

Begin
  Assign( Bytefile, 'A:DecodeB.dat' );
End;
{=============================================================}
Procedure Write_Data;

Var
  A,
  B,
  C,
  Angle,
  Phi,
  Pi:               Real;
  Romvalue:         Byte;
  Cosbyte,
  Sinebyte:         Integer;

Begin
  Rewrite(Bytefile);
  Pi:= 4*Arctan(1);
  Write(Pi);
  For I:= 32768 to 65535 do
  Begin
    Cosbyte:= Hi(I)-128;
    Sinebyte:= Lo(I)-128;
    If (Cosbyte=0) and (Sinebyte>0) then Phi:= Pi/2
    Else
```

```
    If (Cosbyte=0) and (Sinebyte<0) then Phi:= 3*Pi/2
    Else
    If (Cosbyte=0) and (Sinebyte=0) then Phi:= 0
    Else
    If (Cosbyte<0) and (Sinebyte=0) then Phi:= Pi
    Else
    If (Cosbyte>0) and (Sinebyte=0) then Phi:= 0
    Else
    Begin
      Angle:= Arctan(Abs(Sinebyte)/Abs(Cosbyte));
      If (Cosbyte > 0) and (Sinebyte > 0) then Phi:= Angle
      Else
      If (Cosbyte > 0) and (Sinebyte < 0) then
        Phi:= (2*Pi) - Angle
      Else
      If (Cosbyte < 0) and (Sinebyte < 0) then Phi:= Pi + Angle
      Else
      Phi:= Pi - Angle;
    End;
    Phi:= Phi + (Pi/4);
    If (Phi > (2*Pi)) then Phi:=Phi - (2*Pi);
    Romvalue:= Round(255*Phi/(2*Pi));
    Write( Bytefile, Romvalue );
  End;
  Close(Bytefile);
End;
{================================================================}
Begin
  Assign_Filename;
  Write_Data;
End.
```