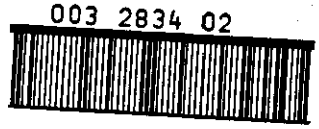


DETECTION OF CODED AND DISTORTED QAM SIGNALS

by

A-K. A-R. KADHIM, BSc, MSc

A Doctoral Thesis Submitted in Partial Fulfilment of the Requirements for the Award of Doctor of Philosophy of the Loughborough University of Technology

February 1989

Supervisor : Professor A. P. Clark

Department of Electronic and Electrical Engineering

CONTENTS

ABSTRACT

The aim of this study is to devise detection processes for digital modems operating at rates of 9600 bit/s and more over telephone lines, using coded QAM signals. The baseband representation of QAM systems and the signal distortion introduced by the telephone circuits are first investigated to derive a suitable model for the simulation of data transmission systems, using a digital computer.

The detection processes studied in the work include both novel and conventional detectors. The performance comparison of the detection processes is based on the tolerance to additive Gaussian noise. A new detection process is proposed for the detection of coded and distorted QAM signals and some near maximum likelihood detectors, previously described, are modified to suit a coded and distorted QAM signal. The modification is such that the detector takes account both of the coding applied at the transmitter and the signal distortion introduced by the channel. The results of the test show that some of these detectors can achieve a performance close to that of the optimum detection process. The complexity required by each detector is also considered.

The effect of noise correlation, introduced by processing the received signal at the receiver, is also investigated for each detector.

A simple interleaving technique for coded and distorted QAM signals is investigated as a solution for error extension effects normally encountered in the detection of a coded signal.

Finally, the most promising of the detectors investigated for 9600 bit/s modems, and for coded and uncoded signals, are investigated further for data rates of 14400, 16000 and 19200 bit/s.

ACKNOWLEDGEMENTS

The author would like to express sincere thanks to his supervisor, Professor A. P. CLark for his continuous guidance and constant encouragement throughout this work.

The financial support of the IRAQI government, without which my work here would not have been possible, is gratefully acknowledged.

I am grateful to all colleagues who have helped in various ways to ease my task.

I would also like to thank my wife, Nidal and my daughter Ruaa for their patience

GLOSSARY OF SYMBOLS

a(t), A(f)	Impulse response and transfer function of transmitter filter
b(t), B(f)	Impulse response and transfer function of receiver filter
c_i	(i+1)th component of the linear feedforward equalizer (chapter 2) or the cost of the vector Q_i
C	Sampled impulse response of the linear feedforward equalizer
C(z)	z-Transform of C
c(t), C(f)	Impulse response and transfer function of transmitter lowpass filter
d_i	(i+1)th component of the vector D
d_m	Minimum free distance of the coded signal
d_{um}	Minimum free distance of the uncoded signal
D	Sampled impulse response of the adaptive linear filter
D(f)	Transfer function of transmitter bandpass filter
D(z)	z-Transform of D
$D_0(f)$	Low frequency component of $D(f+f_c)$
e_i	Sample value of the equalized signal at the input of the detector, at time iT
e_i'	Sample value of the equalized signal at the output of the linear feedforward filter (chapter 2)
E(f)	Transfer function of receiver lowpass filter
$E_0(f)$	Low frequency component of $E(f+f_c)$
E_b	Average transmitted energy per data bit
f	Frequency in Hz
f(t)	Impulse response of receiver lowpass filter
f_c	Carrier frequency in Hz
F	Sampled impulse response of the linear baseband channel and linear feedforward equalizer
F(f)	Transfer function of receiver lowpass filter
F(z)	z-Transform of F
(g+1)	Number of components in the vector Y
G_c	Coding gain in dB
h(t)	Impulse response of transmission path
H(f)	Transfer function of transmission path
$H_0(f)$	Low frequency component of $H(f+f_c)$

j	$\sqrt{-1}$ when not used as subscript
k	Number of stored vectors $\{Q_i\}$ in the near-maximum likelihood detectors
(K+1)	Number of components in the vector V
L	Memory of the convolutional encoder
L(f)	Transfer function of the bandpass channel
$L_0(f)$	Low frequency component of $L(f+f_c)$
m	Number of transmitted data bits per modulation interval
M	Number of signal levels
n	Delay in detection (in terms of symbols)
n(t)	White Gaussian noise waveform
N	Interleaving depth
p(t)	Demodulated baseband waveform
p_i	Sample value of p(t) at time iT
P(f)	Fourier Transform of p(t)
P_{i+1}	(n+1)-component vector represents a possible transmitted data symbols
q(t)	Transmitted QAM signal (real-valued)
Q(f)	Fourier Transform of q(t)
$Q_0(f)$	Low frequency component of $Q(f+f_c)$
Q_i	n-component vector represents a possible transmitted data symbols
r_i	sample value of the received signal at the output of the adaptive linear filter
R_c	Rate of the convolutional encoder
s_i	Transmitted data symbol at time iT (coded or uncoded)
$s_{i,0}$	Real part of s_i
$s_{i,1}$	Imaginary part of s_i
$\hat{s}_i$	Detected value of s_i
T	Symbol period in seconds
u(t)	Gaussian random process with zero mean and fixed variance (noise component in p(t))
u_i	sample value of u(t) at time iT
v(t)	Impulse response of the linear baseband channel
v_i	(i+1)th component of the vector V
V	Sampled impulse response of the linear baseband channel
V(z)	z-Transform of V
w_i	Gaussian random process with zero mean and fixed variance (the noise component in r_i at time iT)

X_i	(i+1)-component vector represent a possible received data sequence
x_i	Possible value of s_i
y_i	(i+1)th component of the vector Y
Y	Sampled impulse response of the linear baseband channel and adaptive linear filter
z(t)	Received waveform at the output of the receiver bandpass filter
z_i	An estimate of the received sample p_i in the absence of noise
Z(f)	Fourier Transform of z(t)
$Z_0(f)$	Low frequency component of $Z(f+f_c)$
$\frac{1}{2}N_0$	Two-sided power spectral density of the additive white Gaussian noise
$\{\alpha_{i,h}\}$	Information digits (binary)
$\{\hat{\alpha}_{i,h}\}$	Detected or decoded values of $\{\alpha_{i,h}\}$
$\{\beta_{i,h}\}$	Convolutionally or differentialy coded binary digits
$\{\hat{\beta}_{i,h}\}$	Detected or decoded values of $\{\beta_{i,h}\}$
$\delta(t)$	The Dirac impulse function
λ	Mean square value of the transmitted data symbols $\{s_i\}$ (coded or uncoded)
σ^2	Variance of real or imaginary components of the noise samples $\{u_i\}$
ψ	Signal/noise ratio in dB.

System IMNC	Interleaving system using system MNC
System INC	Interleaving system using system NC
System LC	Linear equalizer with convolutionally encoded signal
System LU	Linear equalizer with an uncoded signal
System MNC	Modified nonlinear equalizer with convolutionally encoded signal
System NC	Simple nonlinear equalizer with convolutionally encoded signal
System NU	Nonlinear (decision feedback) equalizer with an uncoded signal

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF THE RESEARCH

For the past few decades, voiceband data communication has been an active area for both commercial development and scientific investigation. The advent of the digital computer in the early 1950's and the resulting commercial and military interest in large scale data processing systems led to an interest in using telephone lines for transmitting digital information. As a result, the demand for high speed data transmission increased and it became desirable to attempt a more efficient use of the telephone channels [1].

Efficient modulation methods, such as Quadrature Amplitude Modulation (QAM), have been used since the late 1960's to achieve reliable transmission of data at rates upto 9600 bit/s or more [2].

Channel coding (convolutional or block coding) is used to obtain a better tolerance of the data transmission system to noise, and the 1980's has seen the introduction of the first practical channel coding schemes for telephone line modems, where the coding and modulation processes are combined together [3-6]. In these schemes, the redundancy required by coding is carried by the corresponding increase in the signal alphabet, so that the bandwidth and the element rate of the transmitted signal are unchanged. Convolutional codes are normally used here to give a trellis coded modulation system. Coding gains of up to 6 dB in tolerance to additive white Gaussian noise can be achieved by such systems, at very low error rates [3-5]. In

1984 a trellis coded modulation scheme with a coding gain of 4 dB was adopted by the International Telegraph and Telephone Consultative Committee (CCITT) for use in high speed modems [5,7-9].

The performance of data transmission systems is normally measured under ideal conditions, when the transmission path does not distort the signal and when the receiver uses a maximum likelihood detector, implemented by means of Viterbi algorithm [3,10-16]. When the telephone circuit introduces significant signal distortion, the received coded and distorted signal is ideally detected by a Viterbi algorithm detector that takes due account of both the coding and distortion [13,14,17,18]. The detector in this case becomes considerably more complex and may no longer be cost effective [13].

The aim of this thesis is to develop more effective detectors for use with coded and distorted QAM signals received at 9600 bit/s or more over telephone circuits, and to assess the likely value of such detectors.

1.2 OUTLINE OF THE THESIS

In Chapter 2, descriptions of some related communication topics are given . The general model of a serial data transmission system is presented. The telephone circuits and the different types of noise and distortion introduced by them is then described. The available equalization and detection techniques for an uncoded QAM signal is also presented. Section 2.4 is concerned with convolutional codes. The general description of the encoder is given, followed by the recent developments of the combined coding and modulation. Finally, the detection (decoding) of convolutionally encoded signal is discussed

Chapter 3 describes the model of data transmission system when the data is transmitted at a rate of 9600 bit/s. The differential encoder, which is used with the uncoded (not convolutionally encoded) signal and the adopted trellis coded modulation schemes are described. The characteristics of the telephone circuits, which are used in the work, and their sampled impulse responses are given.

The detection processes used for an uncoded 16-level QAM signal are described in Chapter 4. These are the equalizers and near-maximum likelihood detectors. The detectors are used here as basic systems with which the proposed detectors for convolutionally encoded and distorted QAM signal are compared. The complexity of the detection processes are also investigated in this chapter.

Chapter 5 describes a wide range of detection processes that are suitable for coded and distorted signals. The complexity of the detectors are studied and their performances are compared with those of the corresponding uncoded systems by using the results of the computer simulation tests.

The effect of the noise correlation introduced by processing the received signal at the receiver, on the performance of the detectors which are described in Chapters 4 and 5, is investigated in Chapter 6.

In Chapter 7, a simple interleaver is used with the coded signal aiming to improve the performance of the coded systems.

The most promising detectors for encoded and uncoded signals together with the equalizers are tested further in Chapter 8, when the data is transmitted at rates of 14400, 16000 and 19200 bit/s. The model of data transmission systems and the mapping of the signals at these rates are also described in this chapter. The results of the computer simulation tests are then presented and discussed.

Chapter 9 gives the conclusions of the investigation and suggestions for further work.

CHAPTER 2

RELATED THEORY

2.1 INTRODUCTION

A serial data transmission system, in general, consists of three basic parts; the transmitter, the transmission path and the receiver, as shown in Fig. 2.1. The information to be transmitted is carried by a sequence of binary digits $\{\alpha_{i,h}\}$, where h=1,2,... m. These digits are fed to the encoder in groups, each group containing m binary digits. The goal of the encoder and its mapping process (Fig. 2.1) is to map the input digits $\{\alpha_{i,h}\}$ into channel input symbols such that the number of errors between the $\{\alpha_{i,h}\}$ and the corresponding digits at the output of the detector/decoder is minimized [19]. The $\{\alpha_{i,h}\}$ are statistically independent binary digits and equally likely to have any of their two possible values 0 and 1. The baseband signal at the input to the linear modulator in the transmitter, may be expressed as a sequence of impulses

$$s(t)=\sum_i s_i\,\delta(t-iT) \qquad \ldots \quad 2.1.1$$

where $\delta(t)$ is the Dirac function, s_i is the $(i+1)^{th}$ transmitted data symbol and T is the interval between the impulses, in seconds. Although in practice the signal fed into the modulator is always in the form of a rectangular or rounded waveform, the assumption in Eqn. 2.1 is used to simplify the theoretical analysis of the system.

The transmitter filter, with impulse response a(t), is employed to shape the spectrum of the signal fed to the transmission path, to match its available bandwidth and consequently to maximize the signal power at the receiver input for a given transmitted signal power. The transmission path could be a lowpass or bandpass channel with an impulse response h(t) which, for practical purposes, will be of a finite duration and time invariant. White Gaussian noise n(t), with zero mean and two sided power spectral density $\frac{1}{2}N_o$, is added to the signal at the transmission path output. Although the transmission path, in practice, introduces different kinds of additive noise, the tolerances of different data transmission systems to additive white

Gaussian noise is a good measure of their relative tolerances to most practical types of additive noise [20]. The receiver filter with impulse response b(t), removes the noise frequencies outside the signal band without excessively band limiting the signal itself. The transmitter filter, the transmission path and the receiver filter in cascade are assumed here to form a linear baseband channel with impulse response v(t) is given by

$$v(t) = a(t) * h(t) * b(t) \quad \ldots \quad 2.1.2$$

where * represents the operation of convolution.

The received signal at the output of the receiver filter is given by

$$p(t) = \sum_i s_i v(t - iT) + u(t) \quad \ldots \quad 2.1.3$$

where u(t) is the noise component in p(t) and is given by

$$u(t) = n(t) * b(t) \quad \ldots \quad 2.1.4$$

The waveform p(t) is sampled once per data symbol, at the time instants $\{iT\}$, to give the received samples $\{p_i\}$, where

$$p_i = \sum_{h=0}^{K} s_{i-h} v_h + u_i \quad \ldots \quad 2.1.5$$

and $p_i = p(iT)$, $v_i = v(iT)$ and $u_i = u(iT)$. Thus the sampled impulse response of the linear baseband channel is given by the $(K+1)$ -component row vector

$$V = [\, v_0 \quad v_1 \quad v_2 \quad . \quad . \quad . \quad v_K \,] \quad \ldots \quad 2.1.6$$

The transmitter and receiver filters are normally designed to have a bandwidth (measured over positive frequencies) that is close to 1/2T Hz, so that the sampling rate of 1/T samples/s is close to the Nyquist rate for p(t) [20], which ensures that most of the information carried by the received waveform p(t) is contained also in the samples $\{p_i\}$.

The detector/decoder in Fig. 2.1 operates on the samples $\{p_i\}$ to produce the detected values $\{\hat{s}_i\}$ of the data symbols $\{s_i\}$ and the corresponding detected values $\{\hat{\alpha}_{i,k}\}$ of the information digits $\{\alpha_{i,k}\}$.

2.2 TELEPHONE CIRCUITS

Telephone circuits are voiceband channels with a nominal bandwidth of 200-3400 Hz, and they can handle transmission rates of typically up to 19200 bit/s [20-22]. A telephone circuit connecting two subscribers is normally made up of two or more links connected in tandem. These links are usually of four different types; unloaded audio, loaded audio, PCM and carrier links. Microwave and HF radio links are also used [20]. Since each of these links has its own properties, different kinds of distortion and noise may be experienced. Multipath propagation is a significant source of the distortions in a telephone circuit, and results from the reflections at two or more points of mismatch in the circuit. Multipath propagation causes time dispersion of the received signal elements. Each element now constitutes a main pulse together with several echoes of this pulse that are dispersed in time [20,22]. The received signal elements therefore overlap each other, and so, any sample of the received waveform is a function of several elements (and hence of the data symbol carried by these elements). This effect contributes to intersymbol interference which becomes very serious at the higher transmission rates. The presence of echoes in the received signal corresponds to the appropriate combination of attention and delay distortions introduced by the circuit. The characteristics of an ideal telephone circuit are a constant group delay over the entire bandwidth and an attenuation-frequency response as shown in Fig. 2.2.a, whereas the attenuation and group delay characteristics of a poor telephone circuit could be as shown in Fig. 2.2.b [23]. The latter circuit is described in Chapter 3.

Telephone circuits may be divided into two groups: private (leased) and switched lines. The private lines are not connected through any of the automatic switches in the exchange and they are also disconnected from the exchange battery supplies.

The attenuation and group delay characteristics of the private lines are constrained to lie within specified limits [24,25]. The switched line, which is a line on the public network, is a circuit obtained when using an ordinary telephone to set up a call [20].

The noise experienced over the telephone circuits may be classified into additive and multiplicative noise. The most important type of additive noise is the impulsive noise. It comprises short busts of random additive noise and arises from the switching equipment in the telephone system [21,24]. White noise with low level and wide bandwidth may be experienced over telephone circuits. It produces errors in the detection only at very low signal level [20]. The noise introduced by the ADPCM links, which can be considered as a combination of additive and multiplicative noise, can impair the operation of the telephone modem at high transmission rates (around 9600 bit/s) [21]. The effects of the multiplicative noise in the telephone lines can be divided into amplitude and frequency modulation effects. Amplitude modulation effects occur such as transient interruption, sudden signal-level changes and modulation noise, which appear as amplitude modulation of the signal by bandlimited White noise [20].

Frequency modulation effects such as, phase jitter, frequency offset and sudden phase changes occur only over telephone circuits containing carrier links [20]. Phase jitter is basically frequency modulation of the transmitted signal by the low frequency harmonics of the mains frequency (50 or 60 Hz). This can seriously degrade the performance of the transmission system at rates of 9600 bit/s and above [22]. The frequency offset may have a value up to $\mp$ 5 Hz and is usually compensated for by the carrier recovery loop in the demodulator. Sudden carrier phase changes may occur quite regularly and involve a large change of phase. Differential encoding is used normally to limit the effect of the phase changes.
When a transient interruption exceeds a certain duration or when a sudden signal level or carrier phase change exceeds a certain level, errors are more likely to result in the detected data symbols [21].

The amplitude and frequency modulation effects can be predominant over the private lines, while the switched lines have in addition a high level of impulsive noise which may often mask the other types of noise [20].

2.3 EQUALIZATION AND DETECTION

In this section, the equalizers and different detection techniques are described briefly. The transmitted data signal is assumed here to be an M-level QAM signal. The model of the data transmission system for QAM signal is described in detail in Appendix A, and so, all the assumptions made in Appendix A are applicable here also. Fig. 2.3 shows the discreet time model of the data transmission system, where the sampled impulse response of the linear baseband channel is given by the $(K+1)$ -component vector

$$V = [v_0 \quad v_1 \quad v_2 \quad . \quad . \quad . \quad v_K] \qquad \ldots \quad 2.3.1$$

and the received sample at the detector input at time t=iT is given by

$$p_i = \sum_{h=0}^{K} s_{i-h} v_h + u_i \qquad \ldots \quad 2.3.2$$

where p_i, s_{i-h}, v_h and u_i are complex-valued samples, and the noise samples u_i are statistically independent Gaussian random variable with zero mean and variance $2\sigma^2$. When the channel introduces no intersymbol interference Eqn. 2.3.1 can be written as

$$V = [v_0 \quad 0 \quad 0 \quad . \quad . \quad . \quad . \quad 0] \qquad \ldots 2.3.3$$

and so Eqn. 2.3.2 becomes

$$p_i = s_i v_0 + u_i \qquad \ldots \quad 2.3.4$$

In this case, the detector in Fig. 2.3 will be an appropriate threshold detector for the given signal. For example, if the QAM signal has 16 levels, as shown in Fig. 2.4, the threshold levels of the detector are placed at -2, 0 and 2 for each of the real and imaginary axes, and a correct detection of s_i can be made so long as $|u_i/v_0| < 1$ [26]. When $v_0 = 1$, and for the 16-level QAM signal (Fig. 2.4), it can be shown that the average probability of s_i being incorrect is given by

$$P_e = 3Q\left(\frac{1}{\sigma}\right) \qquad \ldots \quad 2.3.5$$

where σ is the standard deviation of the Gaussian noise samples $\{u_i\}$, and Q(.) is the Gaussian error probability density function which is defined by [20]

$$Q(x) = \int_{x}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy \qquad \ldots \quad 2.3.6$$

Eqn. 2.3.2 can be rewritten as

$$p_i = s_i v_0 + \sum_{h=1}^{K} s_{i-h} v_h + u_i \qquad \ldots \quad 2.3.7$$

the term $\sum_{h=1}^{K} s_{i-h} v_h$ represents intersymbol interference and unless the inequality

$$\left| \sum_{h=1}^{K} s_{i-h} v_h + u_i \right| < | v_0 |$$

is true, the correct detection of s_i will not normally be obtained. $|x|$ is taken to be the absolute value of the scalar x, and $|X|$ is taken to be the unitary length of the vector X.

2.3.1 LINEAR EQUALIZER

The linear equalizer removes the intersymbol interference introduced by the channel from the received samples $\{p_i\}$. It may be implemented as a linear feedforward transversal filter with (q+1) tap gains, as shown in Fig. 2.5. The equalizer here operates on the received sequence $\{p_i\}$, which is held in stores (marked by T in Fig. 2.5) and then are shifted one place to the right for each new received sample. The output of each store is multiplied by the tap value and then summed to give the equalized sample e_i , at time iT, where

$$e_i = \sum_{h=0}^{q} p_{i-h} c_h \qquad \ldots \quad 2.3.8$$

The sampled impulse response of the equalizer is given by

$$C = [c_0 \quad c_1 \quad c_2 \quad . \quad . \quad . \quad c_q] \qquad \ldots \quad 2.3.9$$

and with z-transform

$$C(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} \quad . \quad . \quad . \quad + c_q z^{-q} \qquad \ldots \quad 2.3.10$$

where $\{c_i\}$ are, in general, complex valued samples.
The z-transform of V (Eqn. 2.3.1) is,

$$V(z) = v_0 + v_1 z^{-1} + v_2 z^{-2} \quad . \quad . \quad . \quad + v_K z^{-K} \qquad \ldots \quad 2.3.11$$

Then the sampled impulse response of the equalizer and the channel will be the $(K+q+1)$ -component vector.

$$F = [f_0 \quad f_1 \quad f_2 \quad . \quad . \quad . \quad f_{K+q}] \qquad \ldots \quad 2.3.12$$

with Z-transform

$$F(z) = f_0 + f_1 z^{-1} + f_2 z^{-2} \quad . \quad . \quad . \quad + f_{K+q} z^{-(K+q)} \qquad \ldots \quad 2.3.13$$

Clearly F(z) is given by

$$F(z) = V(z) C(z) \qquad \ldots \quad 2.3.14$$

If the equalizer achieves the exact equalization of the channel, Eqn. 2.3.12 becomes

$$F = [\overbrace{0\,0\,0\ldots0}^{h} \quad 1\,0\,.\,.\,0] \qquad \ldots \quad 2.3.15$$

where h is a positive integer in the range 0 to (K+q)
The value of h here indicates that the channel and the equalizer together introduce a delay of h sampling intervals. Therefore, F(z) (Eqn. 2.3.13) can be written as

$$F(z) \;=\; z^{-h} \qquad \ldots \quad 2.3.16$$

and from Eqn. 2.3.14

$$C(z) \;=\; z^{-h} V^{-1}(z) \qquad \ldots \quad 2.3.17$$

Clearly C(z) in Eqn. 2.3.17 is a polynomial in z^{-1} with an infinite number of coefficients, which is the condition for the exact equalization. Eqn, 2.3.17 assumes that there is no root (zero) of V(z), that lies on the unit circle in the z-plane.

If the exact equalization of the channel is assumed, the equalized sample at the input to the detector (Fig. 2.5) at time (i+h)T, is given by

$$e_{i+h} = s_i + u'_{i+h} \qquad \ldots \quad 2.3.18$$

where u'_{i+h} is the noise component in e_{i+h} , and is given by

$$u'_{i+h} = \sum_{j=0}^{q} c_j u_{i+h-j} \quad \ldots \quad 2.3.19$$

Since the noise samples $\{u_i\}$ are statistically independent Guassian random variables with zero mean and variance $2\sigma^2$ (as assumed before), $\{u'_{i+h}\}$ are Gaussian random variables with zero mean and variance η^2 [22,26], where

$$\eta^2 = 2\sigma^2 \sum_{j=0}^{q} | c_j |^2 = 2\sigma^2 | C |^2 \quad \ldots \quad 2.3.20$$

and $| C |$ is the length of the vector C (Eqn. 2.3.9). It can be shown that the average probability of error in detecting s_i from e_{i+h} (for the 16-level QAM signal) can be lower bounded by,

$$P_e = 3Q\left(\frac{1}{\sigma | C |}\right) \quad \ldots \quad 2.3.21$$

Where Q(.) is defined by Eqn. 2.3.6

When the channel introduces amplitude distortion $| C | > 1$ and so $\eta^2 > \sigma^2$ [22]. However, if the channel introduces only pure phase distortion, the equalizer performs pure phase equalization and it can achieve the best tolerance to additive white Gaussian noise. Under this condition, the equalizer is also matched to the channel [22,26].

When V(z), the z-transform of the linear baseband channel has no roots (zeros) outside or on the unit circle of the z-plane, the linear equalizer can be implemented as a linear feedback transversal filter with only K taps. This is described in Chapter 4.

In practice, only a finite number of taps (q+1) of the equalizer can be used, and the accuracy of equalization can be increased by increasing the number of taps. Consequently, Eqn. 2.3.17 becomes

$$C(z) \approx z^{-h} V^{-1}(z) \quad \ldots \quad 2.3.22$$

where ≈ means approximately equal.

And so the vector F (Eqn. 2.3.15) may have components with non-zero amplitude beside the h^{th} component whose value is approximately unity in this case. The

presence of these components represents residual interference in the equalized signal. Different techniques have been introduced to determine the tap gains of the equalizer that minimize the residual interference [22,24,27].

When the sampled impulse response of the channel varies with the time, it is necessary to keep adjusting the tap gains of the equalizer to hold it correctly set for the channel. The information needed for the optimum setting of tap gains are derived from the received samples $\{p_i\}$ [22]. The equalizer in this case is known as an adaptive linear equalizer. The adaptation techniques of the tap gains are based on the determination of the error in the equalized signal [2,13,24,28]. An adaptive linear equalizer is used in many medium and high speed ($\geq$ 4800 bit/s) voiceband telephone modems.

2.3.2 NONLINEAR EQUALIZER

The pure nonlinear equalizer has the structure shown in Fig. 2.6. Unlike the linear equalizer, the nonlinear equalizer uses the detected values of the previously received data symbols to form an estimate of the intersymbol interference components in the received samples. Furthermore, it operates correctly whether or not the roots of V(z) (Eqn. 2.3.11) lie on the unit circle in the z-plane.

The equalized sample at the input to the detector in Fig. 2.6, at time iT, is given by

$$e_i = \frac{1}{v_0}\left(p_i - \sum_{h=1}^{K} \hat{s}_{i-h} v_h \right) \qquad \ldots \quad 2.3.23$$

Where p_i is given by Eqn. 2.3.2 and $\{\hat{s}_{i-h}\}$ are the detected values of $\{s_{i-h}\}$ for h=1,2,...**K**, and when these are correctly detected,

$$e_i = s_i + \frac{u_i}{v_0} \qquad \ldots \quad 2.3.24$$

where $v_0 \neq 0$ as assumed before.

The detector selects as $\hat{s}_i$ the one of the possible values of s_i which is closest to e_i. Here $\{u_i/v_0\}$ are the noise components in $\{e_i\}$ and they are Gaussian random variables with zero mean and variance $2\sigma^2/|v_0|^2$, where $2\sigma^2$ is the variance of the noise samples $\{u_i\}$. It can be shown, that the probability of error in detecting s_i is given by [22]

$$P_e = 3Q\left(\frac{|v_0|}{\sigma}\right) \qquad \ldots \quad 2.3.25$$

Eqn. 2.3.25 assumes that the signal is a 16-level QAM (Fig. 2.4), and Q(.) is given by Eqn. 2.3.6. Both p_i/v_0 and e_i have the same wanted data symbol s_i and the same noise component u_i/v_0, so that the nonlinear equalizer removes the intersymbol interference without changing the signal/noise ratio [22]. When one or more of the previously received data symbols are incorrectly detected, their intersymbol interference in the following received samples are enhanced instead of being eliminated. Errors in the detection of s_i, therefore, tend to occur in bursts and the system suffers from the error extension effect. This increases the probability of error by a factor which depends on the signal distortion introduced by the channel and the level of the transmitted QAM signal. However, at high signal/noise ratios, when v_0 is one of the

larger components in the vector V, the error extension effects do not normally change the signal/noise ratio, for a given low error rate, by more than a factor of 1 dB [22].

When all the roots (zeros) of V(z) lie inside the unit circle of the z-plane, the nonlinear equalizer usually gives better tolerance to noise than the linear equalizer [22]. However, when all the roots of V(z) lie outside the unit circle in the z-plane, the error extension effects become serious and the nonlinear equalizer often gives a lower tolerance to additive Gaussian noise than the linear equalizer [22]. However, V(z) frequently has roots outside the unit circle, such that $|v_h| \gg |v_j|$ for $0 \le j \le h$ and $0 \le h \le K$. Under this condition, a better performance can sometimes be achieved with the nonlinear equalizer by detecting s_i from p_{i+h} and not from p_i as in Eqn. 2.3.24 [27]. Clearly, the latter arrangement leaves intersymbol interference terms which cannot be removed by the nonlinear equalizer. A further improvement in the tolerance to additive noise may be obtained by adding a linear filter at the input of the nonlinear equalizer to remove the intersymbol interference terms that are not eliminated by the pure nonlinear equalizer [22,27]. Let the z-transform of the required linear filter with n taps be

$$D(z) = C(z)Y(z) \qquad \ldots 2.3.26$$

where C(z) is the linear equalizer for the channel with (q+1) taps satisfying Eqn. 2.3.22, and

$$Y(z) = y_0 + y_1 z^{-1} + y_2 z^{-2} + \ . \ . \ . \ + y_g z^{-g} \qquad \ldots \ 2.3.27$$

The sampled impulse response of the linear filter is given by the n-component vector

$$D = [\, d_0 \quad d_1 \quad d_2 \quad . \ . \ . \ . \quad d_{n-1} \,] \qquad \ldots 2.3.28$$

with Z-transform

$$D(z) = d_0 + d_1 z^{-1} + d_2 z^{-2} + \ . \ . \ . \ + d_{n-1} z^{-n+1} \qquad \ldots \ 2.3.29$$

which satisfy Eqn. 2.3.26, where

$$n = q + g + 1 \qquad \ldots \ 2.3.30$$

The z-transform of the linear baseband channel and the filter is

$$V(z)D(z)=V(z)C(z)Y(z) \quad \ldots \quad 2.3.31$$

By using Eqn. 2.3.22, the above equation becomes

$$V(z)D(z)\approx z^{-h}Y(z) \quad \ldots \quad 2.3.32$$

When the delay of hT seconds is neglected, the sampled impulse response of the channel and the linear filter is given by the (g+1)-component vector

$$Y=[y_0 \quad y_1 \quad y_2 \quad . \quad . \quad . \quad . \quad y_g] \quad \ldots 2.3.33$$

The structure of the above arrangement is shown in Fig. 2.7. The sample value at the output of the linear filter (Fig. 2.7) at time (i+h)T is given by

$$e'_{i+h}=\sum_{j=0}^{n-1} p_{i+h-j}d_j \quad \ldots \quad 2.3.34$$

which can be written as (see Eqn. 2.3.32)

$$e'_{i+h}=\sum_{j=0}^{g} s_{i-j}y_j+w_{i+h} \quad \ldots \quad 2.3.35$$

where

$$w_{i+h}=\sum_{j=0}^{n-1} u_{i+h-j}d_j \quad \ldots \quad 2.3.36$$

and $\{w_{i+h}\}$ are Gaussian random variables with zero mean and variance

$$\eta^2=2\sigma^2\sum_{j=0}^{n-1}|\,d_i\,|^2=2\sigma^2|\,D\,|^2 \quad \ldots \quad 2.3.37$$

where $|D|$ is the length of the vector D and $2\sigma^2$ is the variance of the noise samples $\{u_i\}$.

The remaining intersymbol interference components in e'_{i+h} will be removed by the nonlinear filter (Fig. 2.7), and the sample value at the input to the detector is

$$e_{i+h}=\frac{1}{y_0}\left(e'_{i+h}-\sum_{j=1}^{g}\hat{s}_{i-j}y_j\right) \quad \ldots \quad 2.3.38$$

where $\{\hat{s}_{i-j}\}$ are the detected values of the data symbols $\{s_{i-j}\}$ and when these values are correct

$$e_{i+h} = s_i + \frac{w_{i+h}}{y_0} \qquad \ldots \quad 2.3.39$$

The detector then selects one of the possible values of s_i , which is closest to e_{i+h} as the detected value of data symbol s_i as before.

To minimize the probability of error in the detection of s_i , it is necessary to minimize $|D|$ (see Eqn. 2.3.37). The minimisation is subject to the constraints imposed by Eqn. 2.3.26. It has been shown [22], that the optimum choice of the linear filter, with z-transform D(z) (Eqn. 2.3.29), is that which replaces all the roots of V(z) (the z-transform of the linear baseband channel) which lie outside the unit circle of the z-plane, by the complex conjugates of their reciprocals. The resultant impulse response of the linear baseband channel and the linear filter is given by the vector Y (Eqn. 2.3.33), which now becomes a minimum phase sequence with all the roots of its z-transform lying inside or on the unit circle of the z-plane. So the linear filter here concentrates the energy of the sampled impulse response Y towards the earliest samples and at the same time removing the phase distortion introduced by the channel, without, however, changing any amplitude distortion [22,23]. Thus the signal/noise ratio at the output of the linear filter is not changed and so this arrangement is optimum in terms of its tolerance to additive white Gaussian noise (since the signal/noise ratio is unchanged through the pure nonlinear equalizer as described earlier). The optimization here is subject to the constraint that the equalizer achieves accurate equalization of the channel. When the sampled impulse response varies with the time, the linear filter must be adaptively adjusted. This involves the estimation of the sampled impulse response of the linear baseband channel (V) first and then the determination of the roots in the z-plane. The knowledge of these roots is now used to adjust the tap gains of the adaptive linear filter [29,30]. A comparison of the available nonlinear equalizers is given in [59] and [60], and it has been shown in the latter that the arrangement just described can give a better tolerance to additive Gaussian noise over all the other techniques at high signal/noise ratios.

2.3.3 THE VITERBI ALGORITHM DETECTOR

It is clear from the description of the equalizers, that only a portion of the received signal energy is used in the detection, the remaining part being eliminated by the equalization process. Since p_i (Eqn. 2.3.2) contains information about s_{i-h} as well as s_i for h=1,2,... K, a detector that operates simultaneously on a group of $\{p_i\}$ in such a way that the intersymbol interference is involved in the joint detection of the $\{s_i\}$ instead of being eliminated may achieve better tolerance to additive white Gaussian noise than the optimum equalizer [22,26]. An effective arrangement of such a detector is known as the maximum likelihood detector or maximum likelihood sequence estimator [31]. The detector here holds in store the distorted and noise corrupted received samples $\{p_i\}$. It also holds in store all possible combinations of the transmitted sequences $\{s_i\}$ based on the knowledge of the possible values of s_i . Noiseless version of the received sequences are then produced by convolving all the possible sequences of the transmitted data symbols with the sampled impulse response of the channel, which is assumed to be known to the detector. The detector then selects one of these sequences which is at the minimum unitary distance (see Eqn.2.4.5) from the actually received sequence, and the corresponding possible combination of the data sequence is taken to be the detected sequence of $\{s_i\}$.

When the transmitted data symbols are statistically independent and equally likely to have any of their M possible values, and the noise samples are statistically independent Gaussian random variables with zero mean (as assumed here), the detected sequence becomes the maximum likelihood sequence, and the selection of this minimizes the probability of error in the detection of the received message [26]. At high signal/noise ratios, the maximum likelihood detector can, for practical purposes, be taken to minimize the average probability of error in the detection of an individual data symbol [31,32].

Since a very long streams of data symbols are usually transmitted in any one message, the maximum likelihood detector can not in practice be implemented to operate on a complete message, in a single operation. A practical approach towards the maximum likelihood detection of the received message is the Viterbi

algorithm detector, which was first proposed for decoding a convolutionally encoded signal [61]. This detector will now be described. Just prior to the receipt of p_i at time iT, the Viterbi algorithm detector holds in store M^K vectors $\{Q_{i-1}\}$

$$Q_{i-1}=[x_{i-n} \quad x_{i-n+1} \quad . \quad . \quad . \quad . \quad x_{i-1}] \qquad \ldots \quad 2.3.40$$

where M is the number of levels (alphabet size) of the signal, $(K+1)$ is the number of components of the vector V which represents the sampled impulse response of the channel (Eqn. 2.3.1), x_h for h ≥ 0 may take on any one of the M-possible values of s_h and n> K .

Each vector Q_{i-1} is formed by the last n-components of the corresponding i-component vector X_{i-1} , which represents a possible received data sequence $\{s_i\}$,

$$X_{i-1}=[x_0 \quad x_1 \quad x_2 \quad . \quad . \quad . \quad x_{i-1}] \qquad \ldots \quad 2.3.41$$

Associated with each vector Q_{i-1} is stored its cost ;

$$c_{i-1}=\sum_{h=0}^{i-1} | \, p_h - z_h \, |^2 \qquad \ldots \quad 2.3.42$$

where

$$z_h=\sum_{j=0}^{K} x_{h-j} v_j \qquad \ldots \quad 2.3.43$$

$$p_h=z_h+\hat{u}_h \qquad \ldots \quad 2.3.44$$

where $x_h=0$ for h<0. Here, z_h is the value of p_h that would have been received in the absence of noise, had the given vector X_{i-1} been the actual transmitted sequence of data symbols. The value $\hat{u}_h$ is the corresponding estimate of the noise component in the received sample p_h . It is clear from Eqn. 2.3.42, that the cost c_{i-1} is the square of the unitary distance between the two sequences $\{p_i\}$ and $\{z_h\}$. The smaller the value of c_{i-1} the more likely is the corresponding sequence X_{i-1} to be correct [26].

The M^K vectors $\{Q_{i-1}\}$, at time (i-1)T, have all M^K different possible combinations of the last K components $x_{i-K+1} x_{i-K} \ldots x_{i-1}$, and each vector has the smallest cost for its given combination.

On the receipt of p_i at time t=iT, each vector Q_{i-1} is used to form M vectors $\{P_i\}$.

$$P_i=[x_{i-n} \quad x_{i-n+1} \quad . \quad . \quad . \quad . \quad x_{i-1} \quad x_i] \qquad \ldots \quad 2.3.45$$

The first n components of each vector P_i are given by the components of the original vector Q_{i-1}, and the last component x_i takes on its M possible values. The cost of each vector P_i is given by

$$c_i = c_{i-1} + | p_i - z_i |^2 \quad \ldots \quad 2.3.46$$

where c_{i-1} is the cost of the original vector Q_{i-1} given by Eqn. 2.3.42. Now the total number of the vectors $\{P_i\}$ is M^{K+1}. The detector next selects, for each of the M^K combinations of the components x_{i-K+1} x_{i-K+2} . . . x_i in the vectors $\{P_i\}$, the vector P_i with the smallest cost. The resulting M^K vectors $\{P_i\}$ are then stored together with their costs $\{c_i\}$. The vector P_i with the smallest cost c_i, forms the last (n+1) components of the maximum likelihood vector X_i [22,31], given by

$$X_i = [x_0 \quad x_1 \quad x_2 \quad . \quad . \quad . \quad x_i] \quad \ldots \quad 2.3.47$$

The value of the first component x_{i-n} of the vector P_i with the smallest cost is now taken as the detected value $\hat{s}_{i-n}$ of the data symbol s_{i-n}. So, there is a delay in detection of nT seconds, and this should be made as large as conveniently possible [22]. To avoid an unacceptable increase in the value of the costs $\{c_i\}$, the smallest of the costs of the M^K vectors $\{P_i\}$ is subtracted from every cost, thus the smallest cost is reduced to zero without changing the differences between the various costs [22]. The first component of each selected vector P_i is then omitted to give the corresponding vector Q_i as

$$Q_i = [x_{i-n+1} \quad x_{i-n+2} \quad . \quad . \quad . \quad . \quad x_{i-1} \quad x_i] \quad \ldots \quad 2.3.48$$

The process then continues in this way.

The error probability analysis for the viterbi algorithm detector was first performed in [31] and specific results for the QAM case are investigated in [33]. The analysis is based on the concept of the error event, which is defined as an interval during which the maximum likelihood vector (sequence) differs from the correct sequence [31]. The upper and the lower bounds on the probability of error event are ,respectively , given by [31]

$$P_{ev} \leq K_u Q(d_{un}/2\sigma) \quad \ldots \quad 2.3.49$$

and

$$P_{ev} \geq K_l Q(d_{un}/2\sigma) \quad \ldots \quad 2.3.50$$

where K_u and K_l are constants, and they are independent of the noise variance σ^2 [24,31,34], Q(.) is defined in Eqn. 2.3.6 and d_{um} is the minimum unitary distance between any two different received sequences in the absence of noise [28]. The components of these sequences are given by $\{z_i\}$ (Eqn. 2.3.43). It has been shown [34,35], that the value of d_{um} depends on the energy of the sampled impulse of the channel and the number of its components. Also it has been shown that the error probability differs in most cases only by a small factor from the best that can be achieved for a given channel, thus implying in many cases that the Viterbi algorithm detector uses all the received signal energy for its detection process, and is effectively as good as if there is no intersymbol interference [28,31].

Clearly, from the description of the detection process of the Viterbi algorithm detector, a storage of M^K vectors $\{Q_{i-1}\}$ and M^K costs $\{c_{i-1}\}$ are required, and the detector involves the evaluation of M^{K+1} costs and M^K searches through M costs, for each received data symbol. When M or/and K become large, the detector becomes unacceptably complex. One way to reduce the complexity of the detector, when K is large, is to use a linear filter , which may be adjusted adaptively, ahead of the detector in order to reduce the number of the components in the sampled impulse response of the channel and filter to some desired value [22,36,37]. Many techniques are presented in the published literature to determine the tap gains of the linear filter. When the filter performs some amplitude equalization of the channel, the noise samples at its output will be correlated and the noise variance will be enhanced. So the optimality of the detection process can no longer be achieved [22,36,38]. A nonlinear filter (equalizer) may be used instead of the linear filter. The nonlinear equalizer here makes tentative decisions and uses them to remove some of the intersymbol interference components, leaving the detector to deal with the remaining components [38]. Although no noise correlation will be introduced by the nonlinear equalizer, the arrangement suffers from the error extension effects and some of the signal energy is eliminated by the intersymbol interference cancellation process [39]. Another approach towards reducing the complexity of the Viterbi algorithm detector is to simplify the algorithm itself rather than pre-processing the received signal linearly or nonlinearly. This is described in the next section.

It is assumed throughout the above description that the noise samples $\{u_i\}$ at the detector input are uncorrelated. When these samples are correlated, the Viterbi algorithm detector does not achieve the optimum performance [31]. In this case, a whitening matched filter must be used ahead of the detector to achieve the optimum performance. The whitened matched filter consists of a linear filter that is matched to the channel, a sampler that samples the signal at the output of the linear filter at times $\{iT\}$ and a linear noise whitening network implemented as a linear feedforward transversal filter [26,31,40]. When the received signal is sampled at the Nyquist rate a noise-whitening matched filter is no longer required. It is now sufficient to use in its place, a low pass filter with a rectangular (or nearly rectangular) transfer function and a cut-off frequency equal to the half of the sampling rate followed by the sampler and then an adaptive linear feedforward transversal filter similar to that employed in the optimum (conventional) nonlinear equalizer (see the previous section) [26,41,42].

2.3.4 NEAR-MAXIMUM LIKELIHOOD DETECTORS

These are detectors operating with fewer stored vectors (possible data sequences) and, therefore, with less computation requirements than the Viterbi algorithm detector [23,43-57]. Many of these detectors come close to achieving the maximum likelihood detection (minimum distance) of the received signal, and so they are known as near maximum likelihood detectors. The principle of operation of these detectors, is to consider only k vectors $\{Q_{i-1}\}$ (Eqn. 2.3.40) which have usually the lowest costs $\{c_{i-1}\}$ (Eqn. 2.3.41). Since the greater the cost of a vector the less likely is it to be correct, the greater cost vectors are simply discarded. Here $k \ll M^K$ and typically $4 \le k \le 16$. The near-maximum likelihood detector is usually proceeded by an adaptive allpass linear filter that is adjusted to make the sampled impulse response of the channel and the filter minimum phase [23,57]. The adaptive linear filter here is similar to the linear filter which forms the first part of the optimum nonlinear equalizer [23,44-47]. In Chapter 4, some near-maximum likelihood detectors are described.

2.4 CONVOLUTIONAL ENCODING

2.4.1 GENERAL DESCRIPTION

A convolutional encoder consists of logic circuits that contain shift registers, logic gates and a set of connections between them. The encoder accepts a sequence of m binary digits and it produces at its output k binary coded digits at any time instant, where k>m. The code rate is defined as

$$R_c = m/k \qquad \ldots \quad 2.4.1$$

In general, a convolutional encoder with a memory of L bits may be viewed as a finite state machine with 2^L possible states. The state of the encoder at any time instant is determined by the contents of its store (shift register) at that time instant. Although a convolutional encoder may be described by its generator matrix [19,24,58] the description employed here is in terms of the states and state transition diagram of the encoder. Because the state transition diagram is usually called a trellis diagram, convolutional encoding can be said to be a type of trellis coding.

Let the m input digits form the m-component vector

$$\alpha_i = [\, \alpha_{i,1} \quad \alpha_{i,2} \quad . \quad . \quad . \quad \alpha_{i,m} \,] \qquad \ldots \quad 2.4.2$$

and let the k output digits form the k-component vector

$$\beta_i = [\, \beta_{i,0} \quad \beta_{i,1} \quad . \quad . \quad . \quad \beta_{i,k-1} \,] \qquad \ldots \quad 2.4.3$$

Also let the state of the encoder be defined as the L-component vector

$$\mu_i = [\, \mu_{i,0} \quad \mu_{i,1} \quad . \quad . \quad . \quad \mu_{i,L-1} \,] \qquad \ldots \quad 2.4.4$$

where the binary digits $\alpha_{i,h}$, $\beta_{i,h}$ and $\mu_{i,h}$ may take any one of their two possible values 0 and 1.

The operation of the encoder may now be described as follows. For each input sequence (vector) α_i , the encoder generates the sequence β_i at its output, while changing its state from μ_i to its next state μ_{i+1} . The coded binary sequences are then mapped onto a data symbols (signal elements) by using an appropriate modulation scheme.

When the channel introduces no intersymbol interference and the noise samples are statistically independent Gaussian random variables with zero mean and fixed variance, the received sequence is optimally decoded (detected) by using the maximum likelihood decoder [16,61]. The maximum likelihood decoder operates on the unquantized demodulated samples and it determines the correct transmitted information sequence by using its knowledge of the possible sequences of signal elements [3,4,16,61,62].

A coding gain results when the minimum unitary distance between valid coded signal sequences is greater than that of the corresponding uncoded system [4]. So the objective of the design is to maximize the minimum unitary distance between all the possible (valid) coded signal sequences. The minimum unitary distance is sometimes called the minimum free distance of the code, and can be defined as [63],

$$d_m^2 = \underset{S_h \neq S_j}{Minimum} \mid S_h - S_j \mid^2 \qquad for \quad \text{all} \quad j, h \qquad \ldots \quad 2.4.5$$

Where S_h and S_j assume all the valid pairs of coded symbols (sequences) that the encoder can produce and excludes all the cases where the two sequences are identical, and $\mid S_h - S_j \mid$ is the unitary distance between the two sequences S_h and S_j.
The asymptotic coding gain (in dB) of the coded system over the corresponding uncoded system is given by, [4,64]

$$G_c = 10 \log_{10} (d_m^2 / d_{un}^2) \qquad \ldots \quad 2.4.6$$

where d_m is given by Eqn. 2.4.5 and d_{un} is the minimum unitary distance of the uncoded system. Here Eqn. 2.4.6 assumes that the average transmitted signal energy of the coded and uncoded system is the same [65]. When dealing with binary sequences, the distances (d_m and d_{un}) in Eqn. 2.4.6 may be represented by the corresponding Hamming distance [61]. Usually, for higher signal levels (non-binary) the larger Hamming distance does not translate monotonically into larger unitary distance [8,61]. So d_m and d_{un} , the unitary distances, are considered here instead of the Hamming distances.

The important role played by the minimum free distance of the code becomes evident when the error events of the decoder are examined. The probability of an error event is defined as the probability of an error event starting at specific time instant, given that the decoder has correctly identified the encoder state at that time

instant [64]. The error event probability of the maximum likelihood decoder, assuming an additive white Gaussian noise channel and a high signal/noise ratio, can be lower bounded for the case, of the two dimensional signal, by [16]

$$P_e \geq N_m Q(d_m/2\sigma) \quad \ldots \quad 2.4.7$$

where d_m is defined in Eqn. 2.4.5, N_m is the average number of errors at distance d_m, σ is the standard deviation of the real and imaginary components of the additive noise samples and Q(.) is the Gaussian error probability density function (Eqn. 2.3.6). To ensure a good performance of the coded system, the encoder is designed to achieve a good minimum free distance and hence the modulation method that is used to transmit the coded symbols, becomes important.

2.4.2 COMBINED CODING AND MODULATION

If a code of rate R_c (Eqn. 2.4.1) is used in conjunction with an M-level modulation scheme where the modulator and encoder are separate, the bandwidth of the coded signal must be increased by a factor of $1/R_c$ over the uncoded signal that uses the same M-level modulation. In this case, the coding system may achieve an improvement in the probability of error for a particular signal/noise ratio at the cost of increased bandwidth. But for the systems where the modulation scheme has been designed to use the available bandwidth very efficiently (such as the voiceband telephone channels), an increase in the signal bandwidth is not tolerable. The solution here involves the joint design of the encoder and modulator, such that good distance properties of the coded signal and efficient use of the available bandwidth can be achieved. This was first considered in [66,67], and a more comprehensive solution to the design problem was given by Ungerboeck [3]. He devised a method for mapping the signal points, and called it "mapping by set partitioning". The redundancy required by the coding process is provided by an increased signal alphabet (more possible values of a data symbol) but with no change in the data symbol rate and hence no change in the bandwidth of the transmitted signal. A general description of the Ungerboeck technique is given below, and this is followed by some recent developments.

In this technique, and to transmit m information bits per modulation interval, a constellation of 2^{m+1} signal points is used. Fig. 2.8 shows the general structure of the encoder and its mapping process. The m' bits enter a rate $m'/(m'+1)$ binary convolutional encoder, where $m' \leq m$ and typically $m' = 2$ or 3 . The $m'+1$ coded bits are used to select one of the $2^{m'+1}$ subsets of the 2^{m+1} signal points. The remaining uncoded bits determine which of the $2^{m-m'}$ points in the subset is to be transmitted [68]. This is further described by considering an example, where a 32-level QAM signal constellation is partitioned into eight subsets, each subset representing four signal points, giving m=4, and m' =2. The encoder in this example is known as the G8 encoder [9], which is used throughout the work. The encoder is an eight state convolutional encoder, with a state transition diagram as shown in Fig. 2.7. The details of this encoder, including the circuit diagram and the truth table, are given in Chapter 3. In this chapter, only the partitioning of the signal points and the assignment of the signal subsets to the state transition diagram are described.

The set partitioning of the 32-level QAM signal constellation is shown in Fig. 2.10. This signal constellation is known as a 32-CROSS signal constellation. The signal points in Fig. 2.10 are first divided into two subsets B0 and B1, which are then partitioned into subsets C0, C1, C2 and C3, and the latter subsets partitioned further, to give the final eight subsets D0, D1, ... and D7. The minimum unitary distance between the points within each subset is increased at each partitioning step. So in Fig. 2.10

$$\Delta_0 < \Delta_1 < \Delta_2 < \Delta_3 \qquad \ldots \quad 2.4.8$$

and

$$\Delta_{j+1} = \sqrt{2}\,\Delta_j \qquad \ldots \quad 2.4.9$$

where Δ_j is the minimum unitary distance, at the j-th partitioning step. Thus, in this example, the square of the minimum unitary distance is doubled at each partitioning step. Each of the subsets D0 D1 D2 ..... D7 are defined by a decimal number which follow the letter D. This number specifies the binary coded digits $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$. The two uncoded bits $\alpha_{i,3}$ and $\alpha_{i,4}$ are used to choose a signal point from the selected subset. Fig. 2.11 gives the 32-CROSS signal constellation, where the bit assignment for each point is shown. The assignment of the subsets to the state transition is very important and must satisfy the following rules [3].

1) Any state transition originating from or merging to the same states are assigned signals from the subsets B0 or B1.

2) Any transition having the same originating and joining states (known as parallel transitions) are assigned signals from the same subset (D0,D1,.....D7).

3) All signals appear with equal frequency and the state transitions exhibit a reasonable degree of symmetry.

It can be shown (see Fig. 2.9 and 2.10) that the encoder in the given example satisfies all these rules, and its free distance is given by [9]

$$d_m = \sqrt{2\Delta_1^2 + \Delta_0^2} \qquad \ldots \quad 2.4.10$$

with an asymptotic coding gain of 4 dB (see Eqn. 2.4.6) over the uncoded 16-QAM scheme (Fig. 2.4), where $\Delta_0 = 2$. It has been shown [3], that coding gains of upto 6 dB can be achieved by using this technique.

Despite the advantage of the trellis coded modulation schemes, phase changes introduced by the transmission path cause a phase rotation of the received signal as compared to the transmitted signal [4,5]. This can cause catastrophic errors in the decoding of the received data [4,68]. The problem is solved by employing differential encoding techniques that remove these rotations [4,5,63]. It has been shown that the use of differential encoding does not change the distance properties of the code [5]. Many trellis coded schemes based on the Ungerboeck technique which use differential encoding, have been considered as international CCITT Standards for 9600 bit/s transmission over the public switched telephone network [5,9,68-71], and 14400 bit/s transmission over private lines [7]. The codes used are 8-state trellis codes with a nominal coding gain of 4 dB. The coding technique just described has been used with multidimensional signals (more than two dimensions) in recent years [72,73].

2.4.3 DETECTION (DECODING) OF A CONVOLUTIONALLY ENCODED SIGNAL

When a coded signal is transmitted over an additive white Gaussian noise channel, a maximum likelihood decoder, implemented by means of the Viterbi algorithm, gives the best available tolerance to noise [10,11,61]. If a trellis coded modulation scheme is used, the operation of the Viterbi algorithm detector may be described by using the trellis diagram of the code [74]. The Viterbi algorithm detector holds in store 2^L possible coded sequences (vectors), where L is the memory of the code. For the coding scheme considered in the example of the previous section, each sequence is associated with a different one of the 2^L states at time iT (Fig. 2.10), where 2^L is 8. Associated with each vector is stored its cost, which is the square of the unitary distance between the actual received sequence and the particular stored vector. On the receipt of the received sample at time (i+1)T, the detector expands each sequence into 2^m, where 2^m is the number of the permitted (valid) transitions that originate from each state at time iT (Section 2.4.2). The incremented cost of the expanded sequence is then given by the square of the difference between the received sample at time (i+1)T and the possible value of the transmitted coded symbol which is determined by the particular transition. The cost of an expanded sequence is then given by adding the incremented cost to the cost of the original sequence. For each state at time (i+1)T, the detector then selects one of the 2^m sequences which has the smallest cost. The selected sequences (vectors) are called survivors. The detected value of the coded symbol is then given by the earliest component of the survivor with the smallest cost. A delay in detection of typically 5L sampling intervals is normally assumed [75]. In Chapter 5, the Viterbi algorithm detector is described in detail .

There are two other methods of decoding convolutional codes, namely sequential decoding and threshold (majority logic) decoding [76]. The latter is considered as an algebraic method, and basically involve solving sets of algebraic equations [76]. Its performance is inferior to both the Viterbi algorithm decoding and the sequential decoding [76]. Sequential decoders search for the most probable sequence (path) through the trellis by examining one path at a given time instant. For any given state, each evaluated cost (for the given transition) is compared with a threshold. If the evaluated cost is below the threshold then the decoder is forced to go back and try

another transition [74,10]. Clearly, this requires a variable number of operations. It has been found that the performance of the Viterbi algorithm decoder is better than that of the sequential decoder over the real channel impairments [19].

For moderate values of L and m, the complexity of the Viterbi algorithm detector (decoder) can be accepted as the price that needs to be paid for the performance improvement arising from coding. But, when both the number of states and the number of signal levels are relatively large, the Viterbi algorithm detector becomes too complex and its implementation is impractical. Different techniques have been found to simplify the operation of the Viterbi algorithm decoder. Decoding methods that come close to achieving the performance of the maximum likelihood decoder without, however, requiring nearly as much storage or computation per decoded data symbol as does the Viterbi algorithm detector, have been proposed [55,56,74,77]. A simple method of evaluating the costs of the survivors is presented in [78], and a new approach has been proposed by considering a simple coding scheme, such that the required decoder is of low complexity [79].

When the channel introduces intersymbol interference, the maximum likelihood decoder (detector), implemented by means of the Viterbi algorithm, becomes considerably more complex. A total number of $2^L \times 2^{mK}$ states must now be considered by the detector/decoder [14,17,73], where L is the memory of the encoder and K is the number of interfering components in the sampled impulse response of the channel (see Eqn. 2.3.1). So , $2^L \times 2^{mK}$ possible coded sequences must be processed by the Viterbi algorithm detector/decoder. This number corresponds to the number of states of the encoder and the channel [81]. A solution to the problem of high complexity is to use a linear or nonlinear (decision feedback) equalizer to equalize the channel, followed by a Viterbi algorithm decoder which is suitable for the encoded, but not distorted signal [13,14,54,65,78,82-84]. Although, the latter arrangement is simple to implement, it may degrade the system performance such that no advantage is gained through coding [13,14,54]. It is the purpose of this work to develop more effective detection processes for use with a convolutionally encoded and distorted signal.

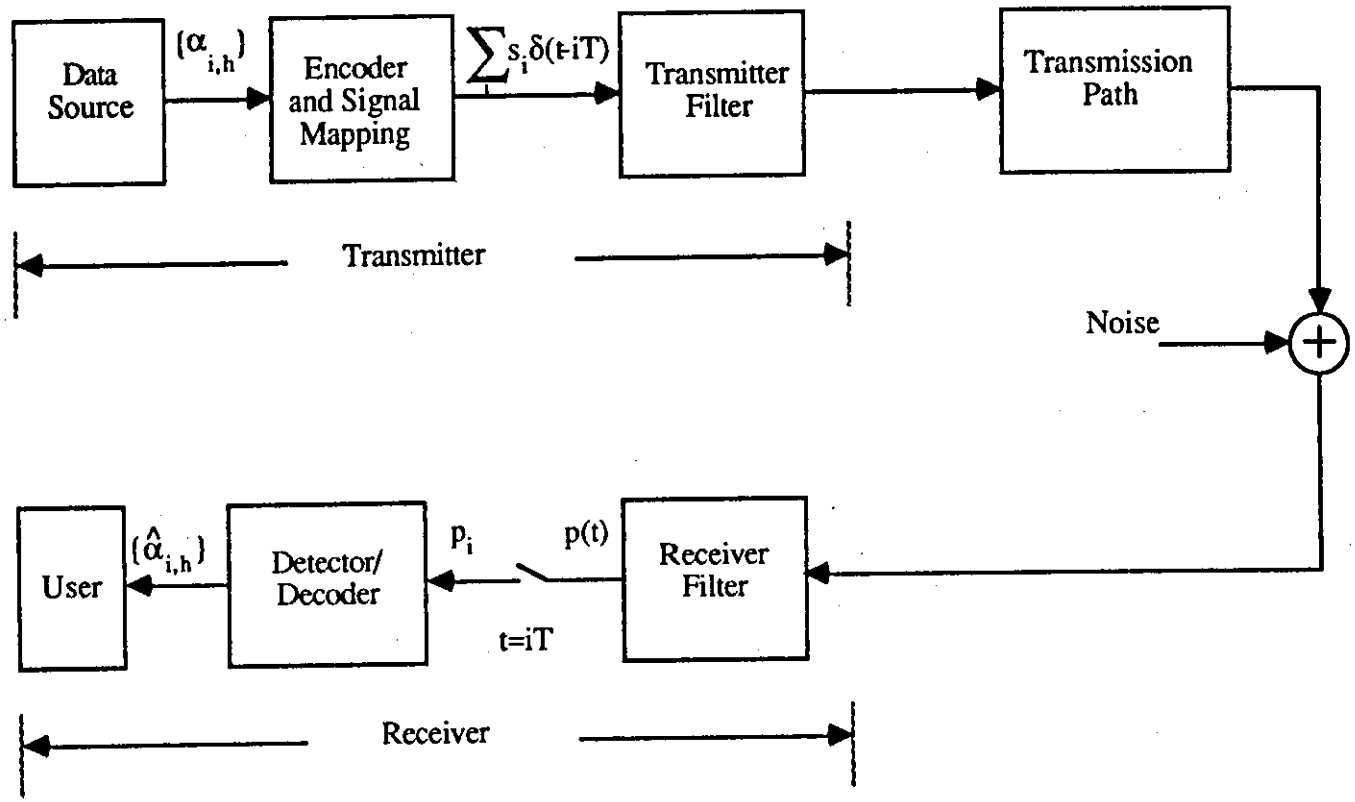


Fig.2.1 General model of data transmission system .

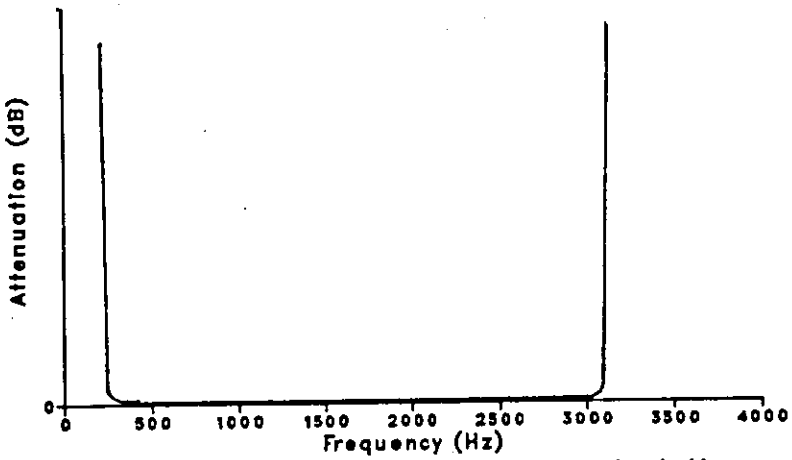
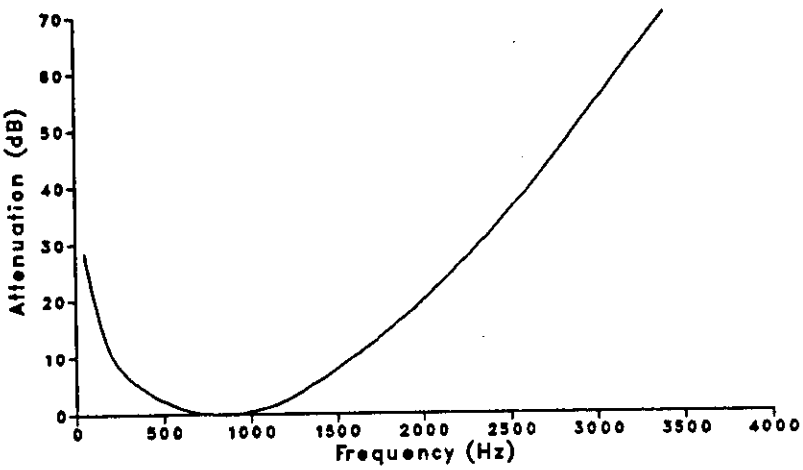
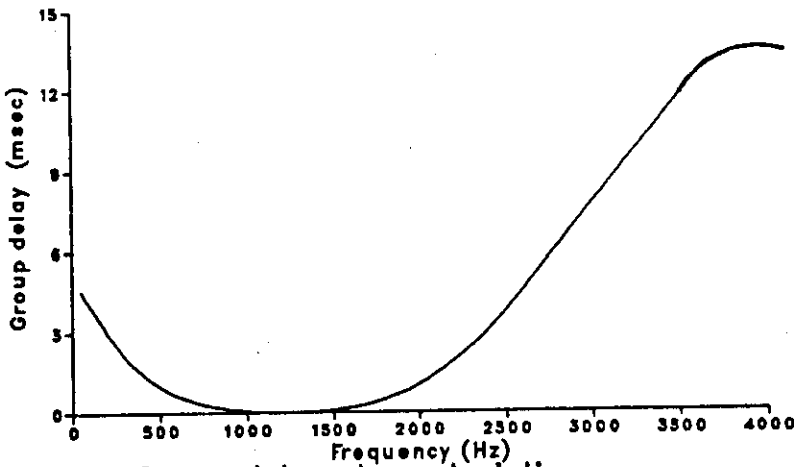


Fig.2.2.a– Ideal attenuation characteristic



b.1– Attenuation characteristic



b.2– Group delay characteristic

Fig.2.2.b Attenuation and group delay characteristics of a poor telephone circuit

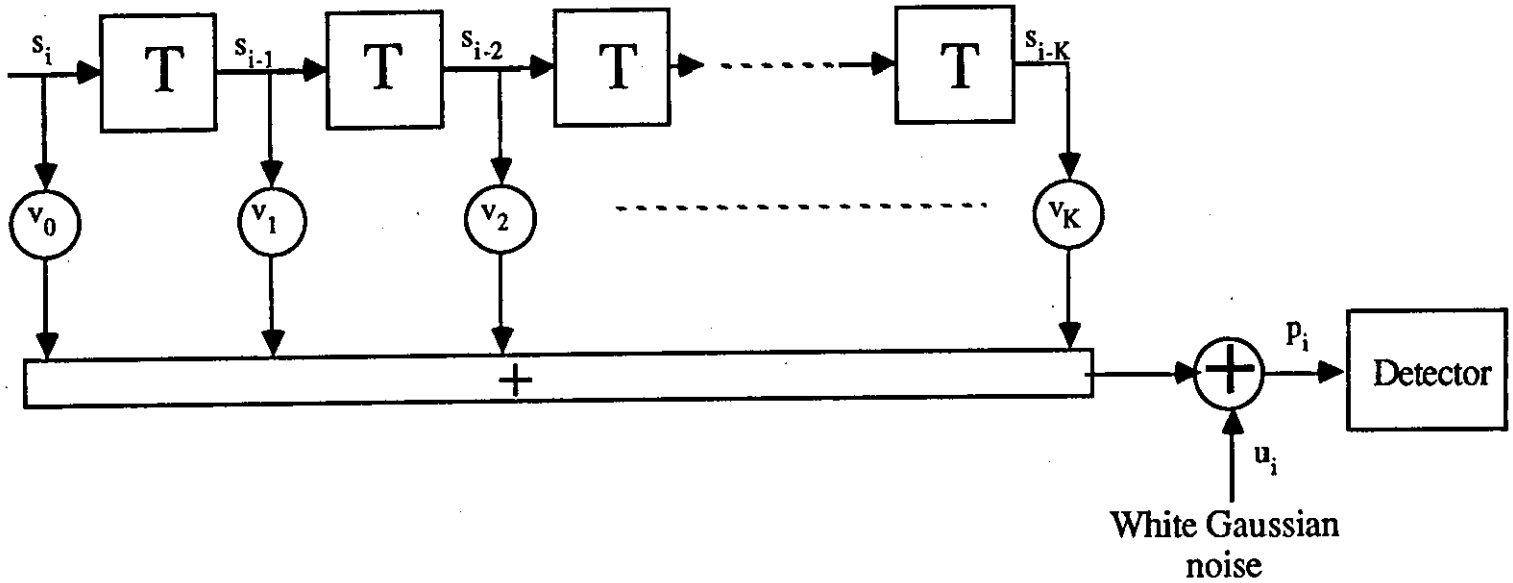


Fig. 2.3 Discreet time model of the data transmission system.

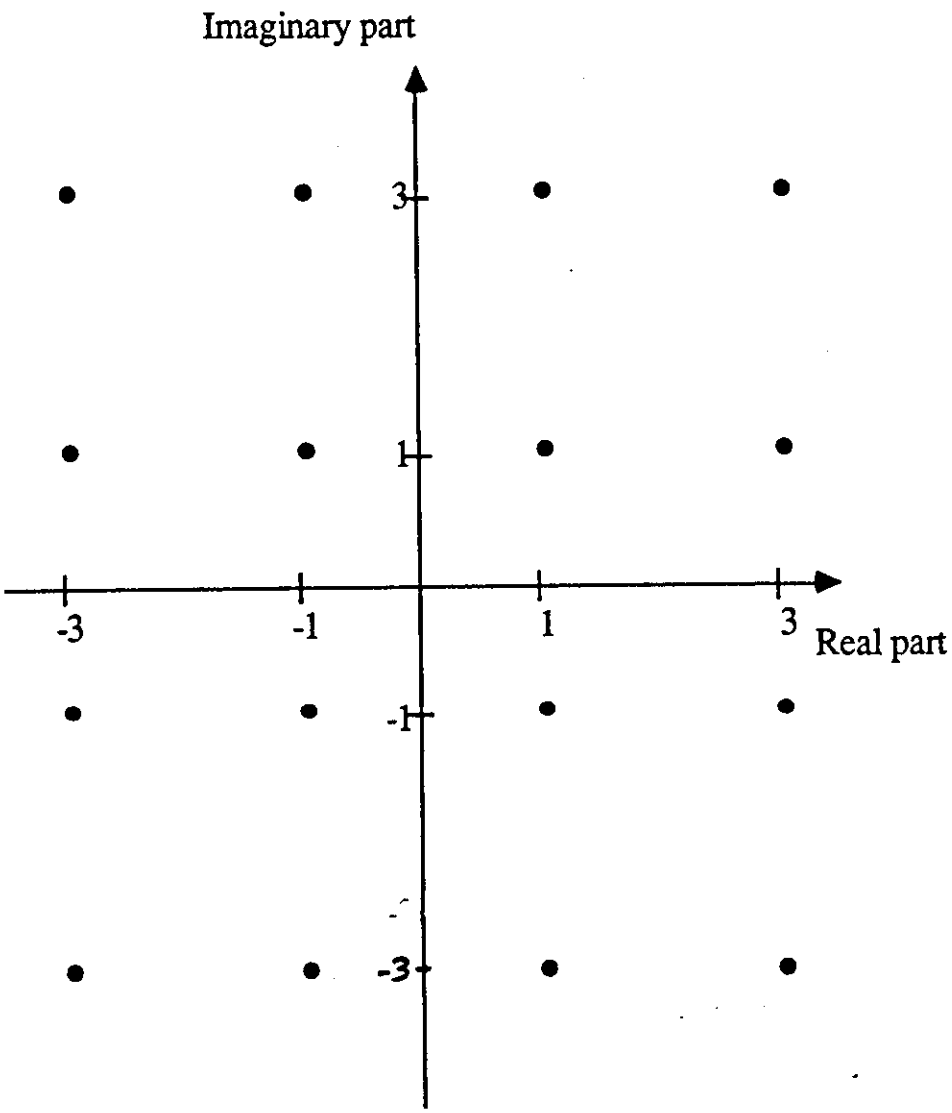


Fig.2.4 Signal constellation of 16-level QAM signal .

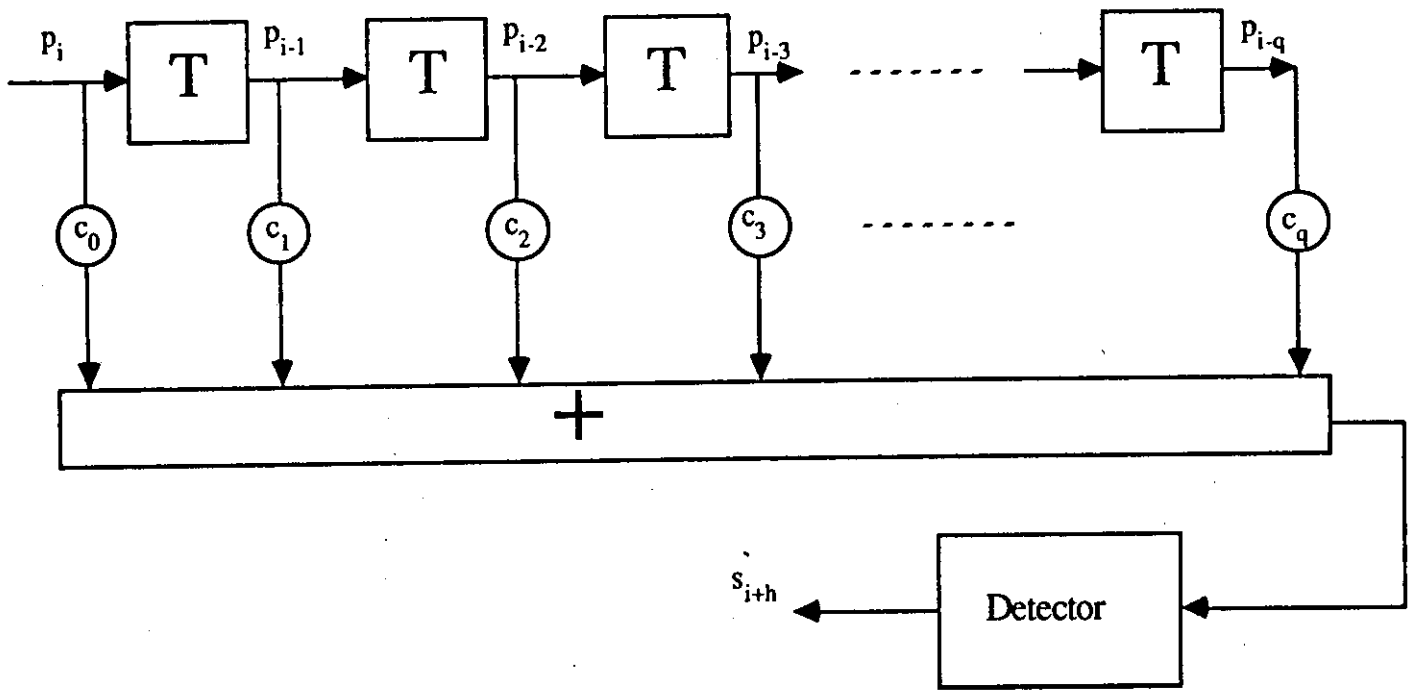


Fig. 2.5 Linear feedforward transversal equalizer and detector

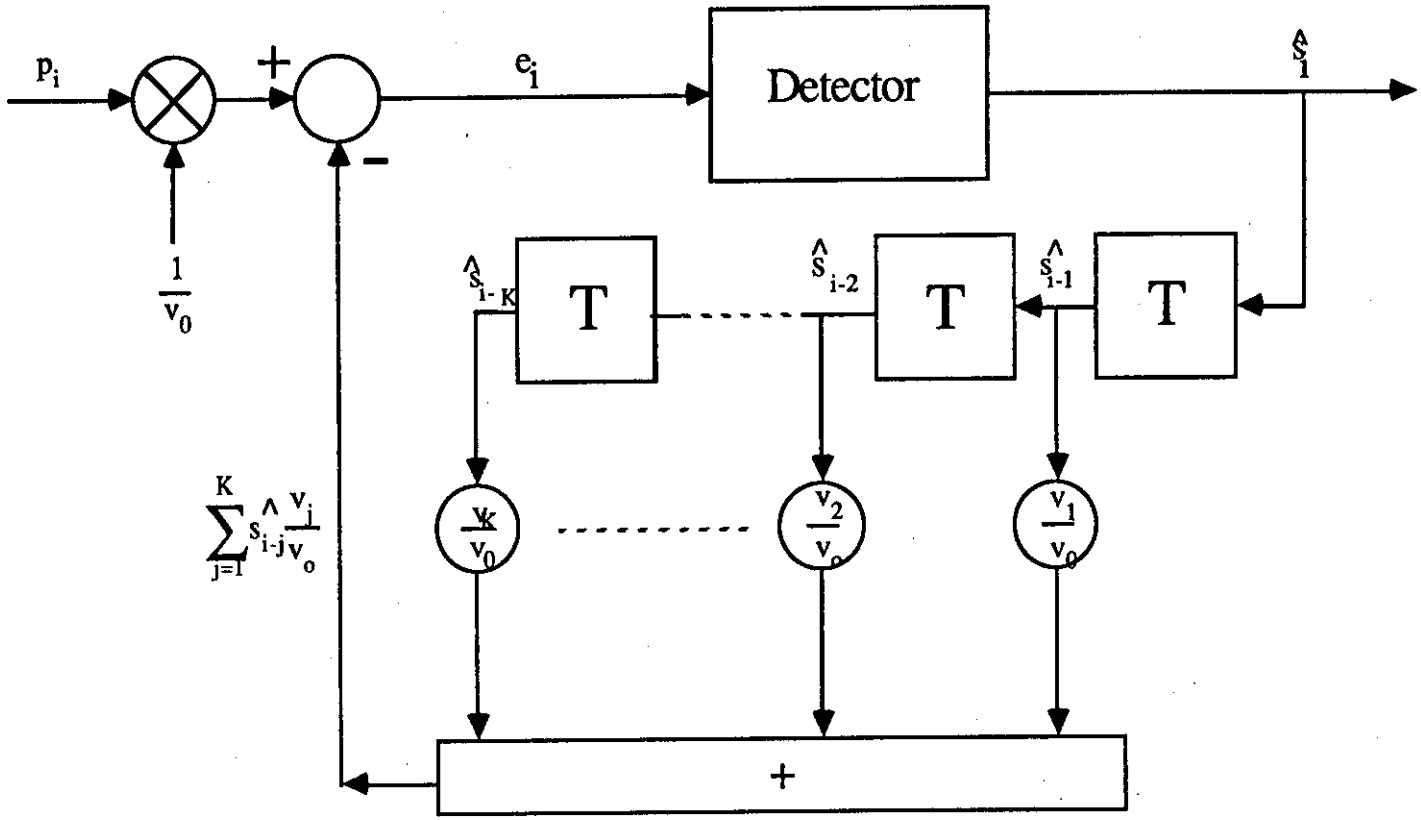


Fig.2.6 Nonlinear equalizer and detector .

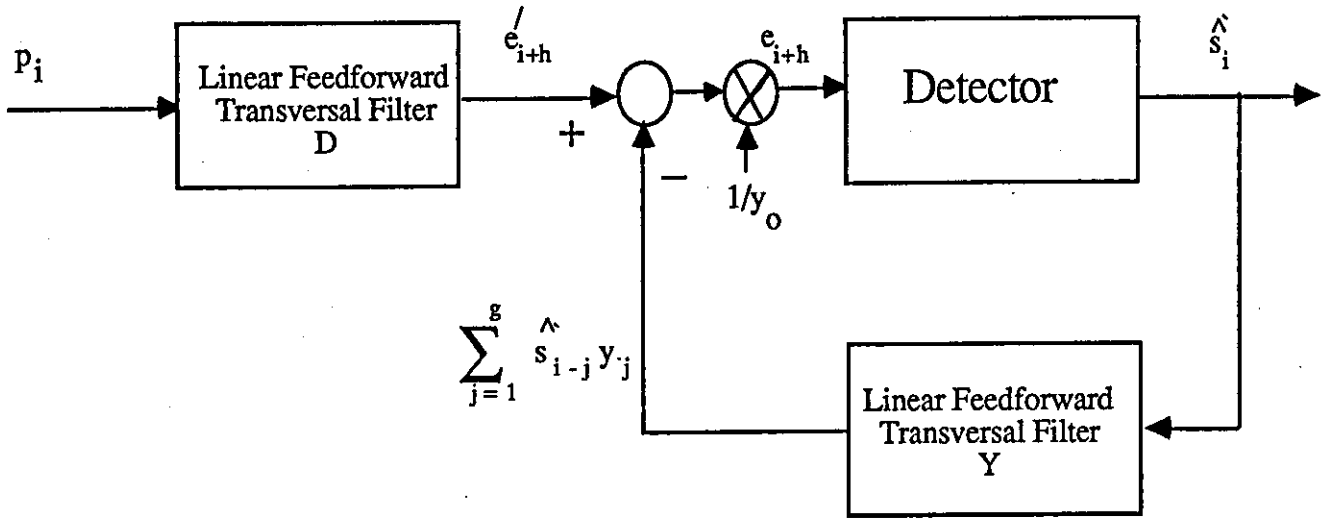


Fig.2.7 Decision feedback equalizer containing both a linear and nonlinear filter .

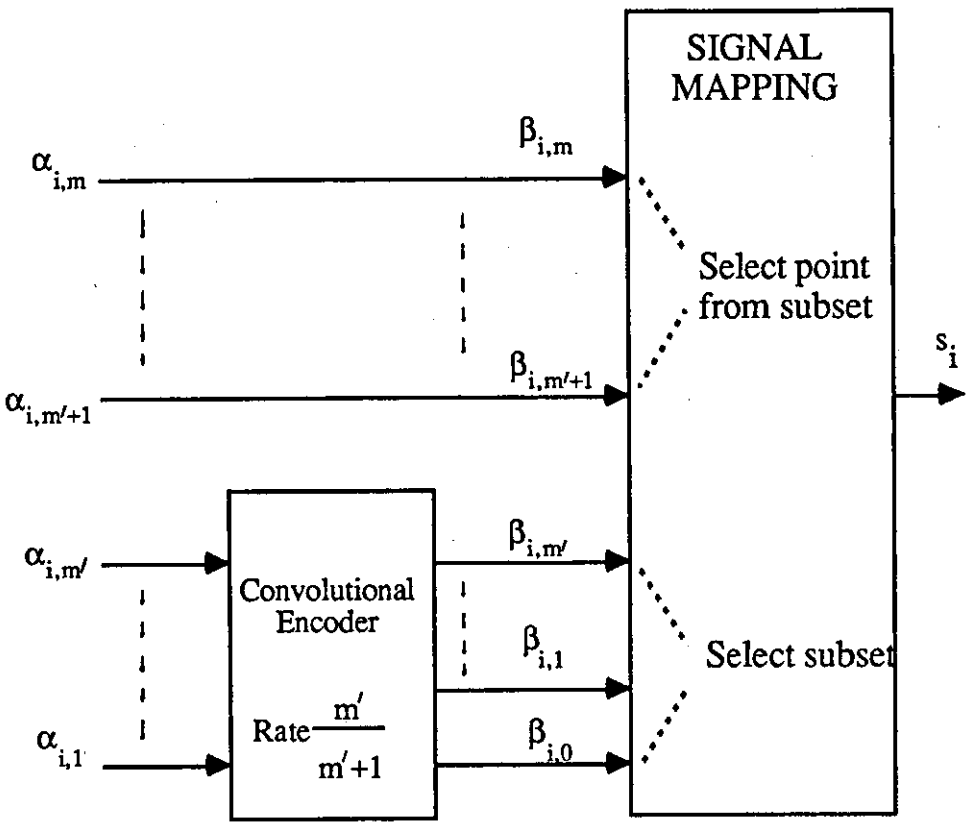


Fig.2.8 General structure of encoder and signal mapping process for trellis-coded modulation .

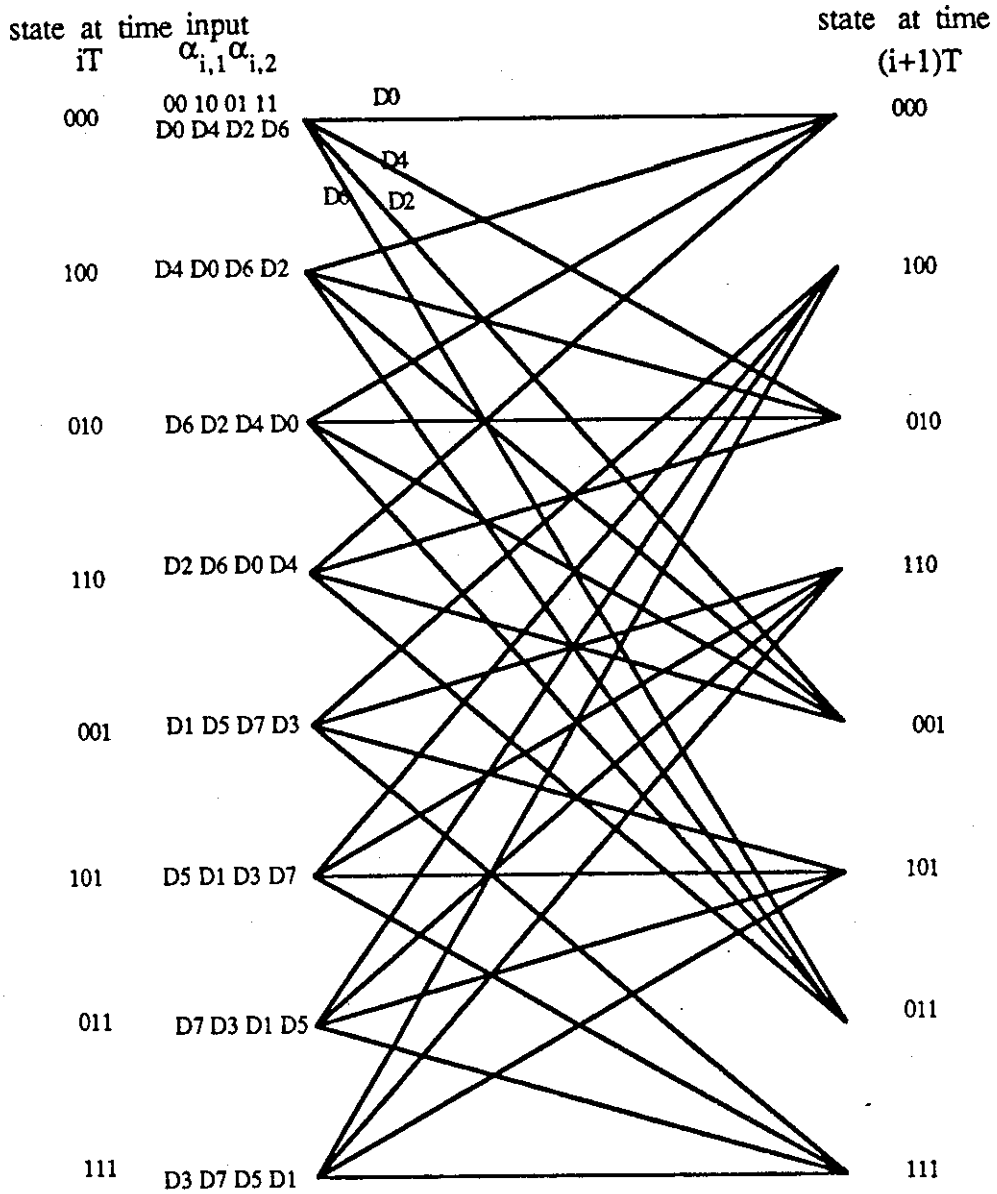


Fig.2.9 State transition diagrame of the encoder.

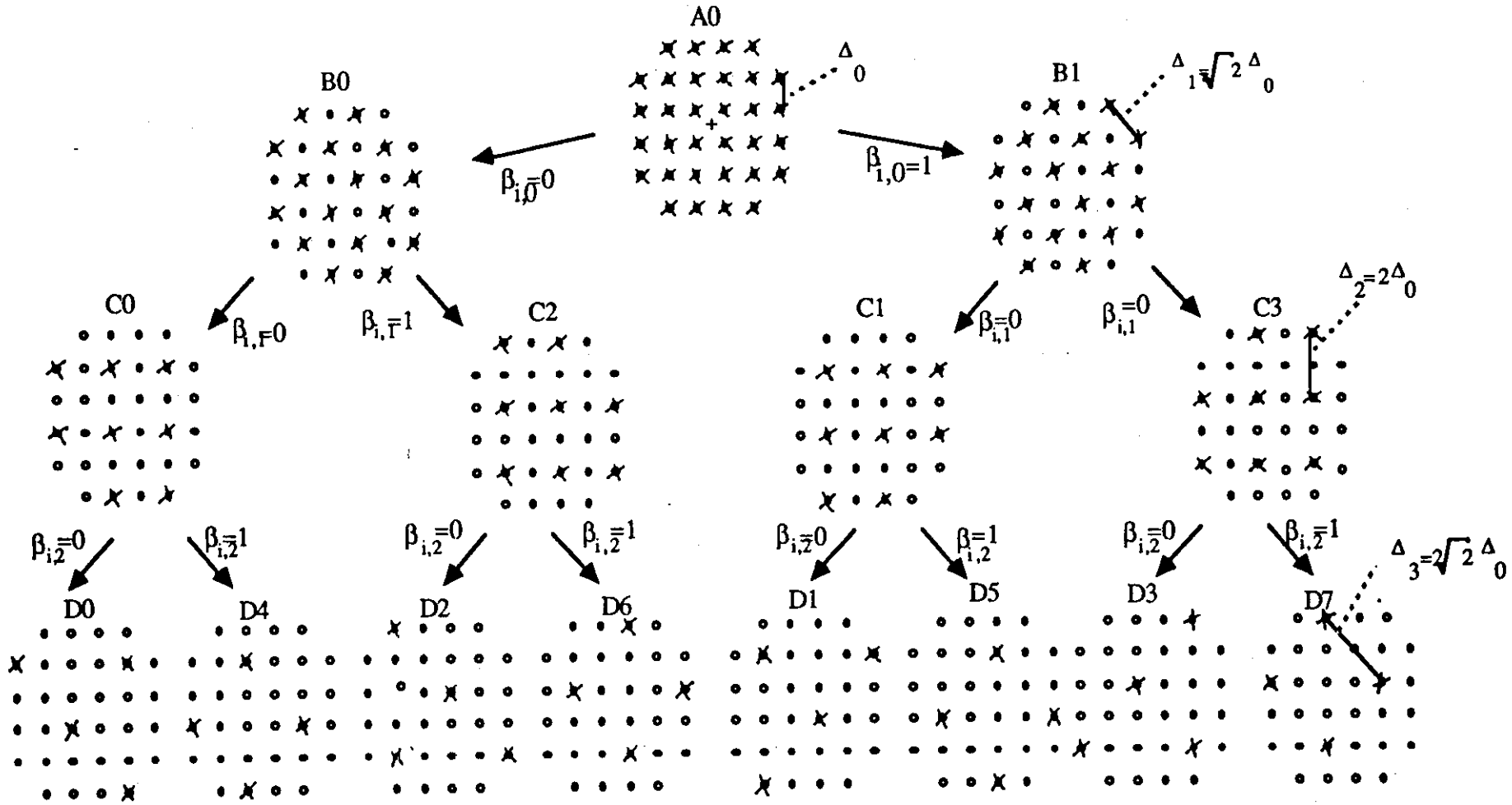


Fig.2.10 Set partitioning of a coded 32-level QAM signal.

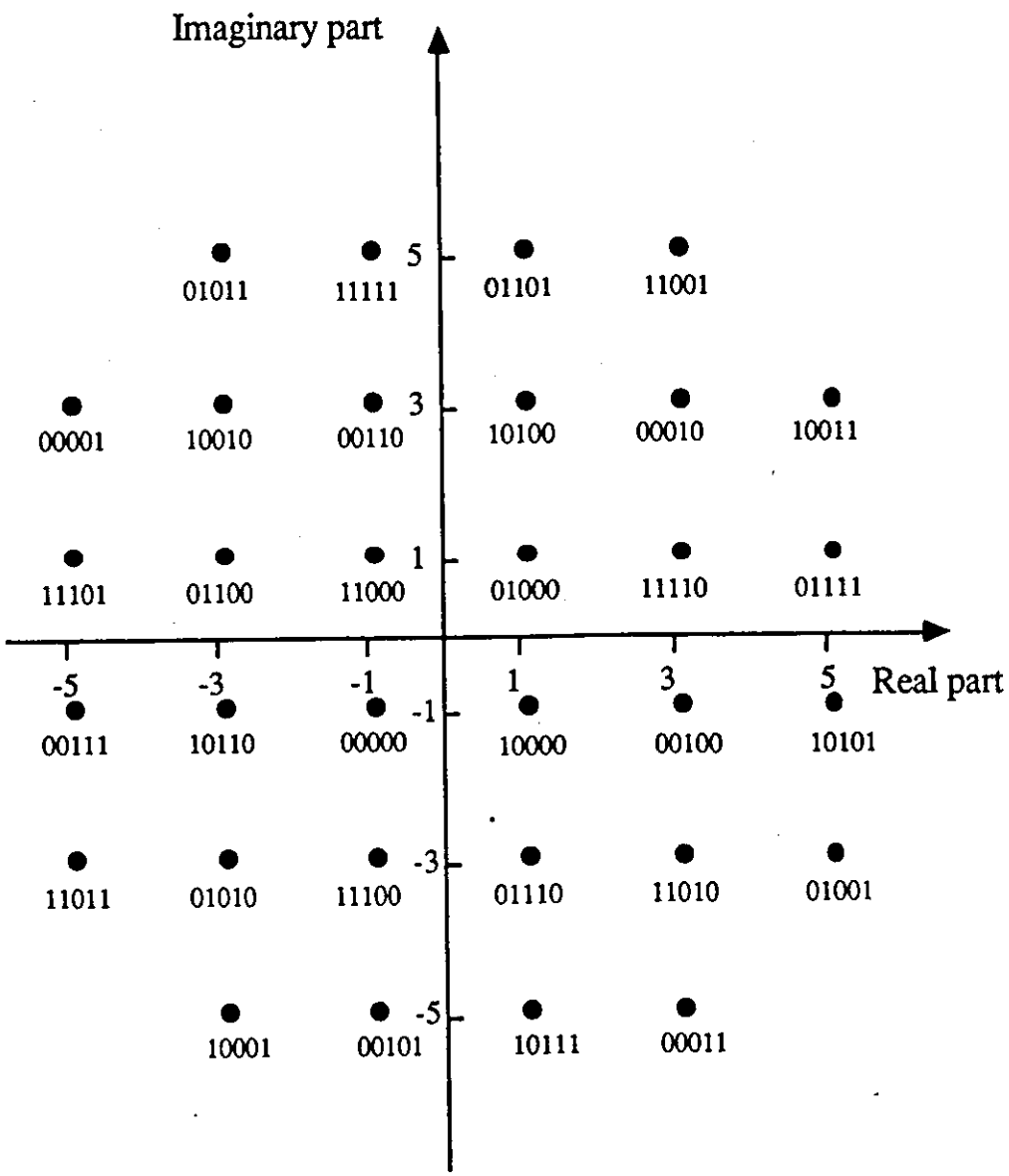


Fig.2.11 Signal constellation for a coded data symbol s_i.

(the binary coded number against each point is $\beta_{i,0}\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}$)

CHAPTER 3

MODEL OF DATA TRANSMISSION SYSTEM

In this chapter, the model of the data transmission system is described. The data transmission system is a synchronous serial system and may operate either with an uncoded 16-level QAM signal or else with a convolutionally encoded 32-level QAM signal, and is shown in Fig. 3.1. In either case the QAM signal has a carrier frequency of 1800 Hz and an element rate of 2400 symbols/s giving a useful transmission rate of 9600 bit/s.

The information to be transmitted is carried by the binary digits $\{\alpha_{i,h}\}$, which are statistically independent and equally likely to have any of their possible values 0 and 1, for $i \leq 0$. The $\{\alpha_{i,h}\}$ occur in separate groups of four adjacent digits $\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}$ and $\alpha_{i,4}$, which determine the corresponding coded or uncoded data symbol s_i . It is assumed that $\alpha_{i,h} = 0$ and $s_i = 0$ for $i<0$, so that s_i is the (i+1)th transmitted data symbol.

The lowpass filter and linear modulator at the transmitter, the transmission path and the linear demodulator at the receiver together form a linear baseband channel, as shown in Fig. 3.1 (Appendix A). The channel has a complex-valued impulse-response v(t), which, for practical purposes, has a finite duration and is time invariant, such that v(t-iT) is a time shifted version of v(t-jT), for any integers $i \neq j$. T here is the modulation interval (1/2400 seconds). The relationship between the resultant linear baseband channel and the bandpass transmission path is considered in Appendix A. The additive white Gaussian noise in Fig. 3.1 is complex valued. It has a zero mean and a constant two-sided power spectral density of $\frac{1}{2}N_0$ for each of its the real and imaginary parts.

3.1 DIFFERENTIAL ENCODING AND THE MAPPING OF THE UNCODED 16-LEVEL QAM SIGNAL

When the data is uncoded (not convolutionally encoded), the data symbol s_i is derived from the binary digits $\{\alpha_{i,k}\}$ by a process of differential encoding. Here, the data symbol s_i is equally likely to take on any one of its 16 possible values given by the signal constellation in Fig. 3.2, Thus

$$s_i = s_{i,0} + j\, s_{i,1} \qquad \ldots \quad 3.1.1$$

$$s_{i,0}\,, s_{i,1} = \mp 1 \quad \text{or} \quad \mp 3 \qquad \ldots \quad 3.1.2$$

and $j=\sqrt{-1}$. The $\{s_i\}$ are statistically independent.

The encoder in Fig. 3.1 is now taken as a differential encoder. The binary digits $\alpha_{i,1}$ and $\alpha_{i,2}$ are recoded to give the corresponding two binary digits $\beta_{i,1}$ and $\beta_{i,2}$ according to Table 3.1 [85], whereas

$$\beta_{i,3} = \alpha_{i,3} \qquad \ldots \quad 3.1.3$$

and

$$\beta_{i,4} = \alpha_{i,4} \qquad \ldots \quad 3.1.4$$

The resulting group of four binary digits $\beta_{i,1}, \beta_{i,2}, \beta_{i,3}$ and $\beta_{i,4}$ now determine the corresponding data symbol s_i, according to Fig. 3.2 [85]. The first two binary digits in any binary coded number determine the quadrant containing s_i, and the remaining two digits ($\beta_{i,3}$ and $\beta_{i,4}$) in any quadrant are the same as those in the all positive quadrant, if this is rotated to coincide with the given quadrant.

Following the detection of s_i, at the receiver, the detected values of the binary digits $\beta_{i,1}, \beta_{i,2}, \beta_{i,3}$ and $\beta_{i,4}$ are determined from Fig. 3.2, and then the detected values of $\alpha_{i,1}$ and $\alpha_{i,2}$ are determined from Table 3.1 by using the detected values of $\beta_{i-1,1}, \beta_{i-1,2}, \beta_{i,1}$ and $\beta_{i,2}$,whereas the detected values of $\alpha_{i,3}$ and $\alpha_{i,4}$ are given by the detected values of $\beta_{i,3}$ and $\beta_{i,4}$, respectively, (Eqn. 3.1.3 and 3.1.4).

It can be seen from Fig. 3.2 and Table 3.1 that a shift of a multiple of $\pi/2$ radians in the phase relationship between the reference carriers in the coherent demodulators and the received signal carrier, giving the corresponding rotation in the phase angle of a received sample r_i , does not change the detected values of $\alpha_{i,3}$ and $\alpha_{i,4}$, corresponding to any given value of s_i , nor can it lead to a prolonged burst of errors in the detected values of $\{\alpha_{i,h}\}$ (for h=1 and 2). This is shown in Appendix B. To reduce further the number of errors in the detection of $\{\alpha_{i,h}\}$, the mapping in Fig. 3.2 is as near as possible to Gray coding; the exact realization of this is not attainable with the given signal, when differential encoding is used [42,85].

3.2 THE CONVOLUTIONAL ENCODER AND MAPPING OF THE CODED 32-LEVEL QAM SIGNAL

Fig. 3.3 shows the convolutional encoder used throughout the work. The encoder carries out a process of convolutional encoding with differential encoding. This encoder is known as the G8 encoder [9], and is one of the trellis coded modulation schemes which have been considered as CCITT standards for V32 modems [9,68]. The sequence of the binary information digits $\{\alpha_{i,h}\}$ are fed to the encoder in groups, each contains four binary digits. For each group the resulting five binary digits $\beta_{i,0}, \beta_{i,1}, \beta_{i,2}, \beta_{i,3}$ and $\beta_{i,4}$ determine the corresponding coded symbol s_i , which may take on any one of the 32 possible values as shown in Fig. 3.4.

In the encoder, differential encoding and convolutional encoding are accomplished by a single finite state machine with eight states. Nonlinear elements (NOT and AND logic gates in Fig. 3.3) have been used in the encoder circuit to make the code invariant to 90° rotations, while maintaining its coding gain [5,9,68]. The differential encoder is used here to resolve the problem of the phase ambiguity [9]. The coded signal constellation has a four-way 90° symmetry and therefore it has an inherent phase ambiguity of integral multiples of $\pi/2$ [9,68,69]. The constellation of Fig. 3.4 is the same as that used in the example of Section 2.4. in Chapter 2, to describe the technique of mapping by set partitioning. The symmetry of the above signal can be

followed by examining the subsets C0, C1, C2 and C3 in Fig. 2.10, where the phase rotations by 90° or its multiples reflected only in the bits $\beta_{i,0}$ and $\beta_{i,1}$. Appendix C show the effect of phase change on the encoded 32-level QAM signal.

The state transition diagram (trellis diagram) is shown in Fig. 3.5. The state of the encoder at time iT is defined by the values of the binary digits $\mu_{i,0}$, $\mu_{i,1}$ and $\mu_{i,2}$ at the outputs of the delay elements in Fig. 3.3. The relations between the input (present) and output (next) state are given in Table 3.2. The encoder operates on $\alpha_{i,1}$ and $\alpha_{i,2}$ to determine $\beta_{i,0}$ and $\beta_{i,1}$, whereas

$$\beta_{i,2} = \alpha_{i,2} \quad \ldots \quad 3.2.1$$

$$\beta_{i,3} = \alpha_{i,3} \quad \ldots \quad 3.2.2$$

and

$$\beta_{i,4} = \alpha_{i,4} \quad \ldots \quad 3.2.3$$

The values of $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$, that correspond to any particular values of $\alpha_{i,1}$ and $\alpha_{i,2}$, depend upon the state of the encoder. Since the truth table (Table 3.2) is independent of $\alpha_{i,3}$ and $\alpha_{i,4}$, it is evident that, associated with each pair of values of $\alpha_{i,1}$ and $\alpha_{i,2}$, are all four combinations of $\alpha_{i,3}$ and $\alpha_{i,4}$, so that every row of the truth table holds for these four combinations. The state at time (i+1)T is uniquely determined by the state at time iT together with either $\alpha_{i,1}$ and $\alpha_{i,2}$ or else $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$. Again, $\alpha_{i,1}$, $\alpha_{i,2}$, $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$ are uniquely determined by the states at the time instants iT and (i+1)T [14]. Clearly, only the information bit $\alpha_{i,1}$ is represented non-systematically at the encoder output. As shown in Chapter 2, the asymptotic coding gain of the above encoder is about 4 dB.

3.3 THE EQUIPMENT FILTERS AND TELEPHONE CIRCUITS

The equipment filters include all filters in the transmission path such as those used in the modulation, demodulation, band limiting and main hum rejection [86]. The equipment filters used here are models of practical filters that have been designed for

an actual modem by M. J. Fairfield [23,47]. The attenuation and group-delay characteristics of the filters are shown in Fig. 3.6. These filters, when considered as operating on the transmitted bandpass signal, introduce an attenuation of about 15 dB at 600 and 3000 Hz, the attenuation increasing rapidly as the frequency is reduced below 600 Hz or increased above 3000 Hz, and being less than 1 dB over the frequency band from 1100 to 2200 Hz. The equipment filters (Fig. 3.6) are known here as equipment filters-1.

Models of four telephone circuits are used in the investigation. Figs. 3.7 to 3.10 show the attenuation and group delay characteristics of these circuits. Telephone circuit 1 (Fig. 3.7) and telephone circuit 2 (Fig. 3.8) introduce negligible and typical levels of distortion, respectively, whereas telephone circuits 3 and 4 (Figs. 3.9 and 3.10, respectively) are close to the typical worst circuits normally considered for the transmission of data at 9600 and 600-1200 bit/s [47]. Telephone circuit 3 introduces severe group delay distortion as well as considerable attenuation distortion, and telephone circuit 4 introduces very severe attenuation distortion [23,44,46,47]

Table 3.3 shows the sampled impulse responses of the above four circuits when combined with the filters. The derivation of these responses is considered in Appendix D. Although the telephone circuits normally introduce various types of additive and multiplicative noise (as described in Chapter 2), it is assumed here that the only noise introduced by the channel is stationary white Gaussian noise with zero mean and a flat power spectral density, which is added to the signal at the output of the telephone circuit. The tolerances of different data transmission systems to additive white Gaussian noise is a good measure of their relative tolerances to most practical types of additive noise [20].

The demodulated and filtered waveform, in Fig. 3.1, is a complex-valued baseband signal p(t), with a bandwidth extending from about -1200 to 1200 Hz, where (see Appendix A)

$$p(t)=\sum_i s_i v(t-iT)+u(t) \qquad \ldots \quad 3.3.1$$

and v(t) is the resultant impulse response of the filters and telephone circuit, and u(t) is the noise waveform in p(t). The sampler in Fig. 3.1 samples the received waveform p(t) once per data symbol, at time instants $\{iT\}$, to give the received samples $\{p_i\}$,

$$p_i = \sum_{h=0}^{K} s_i v_{i-h} + u_i \qquad \ldots \quad 3.3.2$$

where $\{v_{i-h}\}$ are the components of the vector

$$V = [\, v_0 \quad v_1 \quad v_2 \quad . \quad . \quad . \quad . \quad v_K \,] \qquad \ldots \quad 3.3.3$$

which represents the resultant sampled impulse response of the linear baseband channel (Fig. 3.1) formed by the filters and telephone circuit as in Table 3.3.
It is assumed here that the real and imaginary components of the noise samples $\{u_i\}$ (Eqn. 3.3.2) are statistically independent Gaussian random variables with zero mean and fixed variance. This assumption implies that a little more of the filtering is carried out at the transmitter than at the receiver.

3.4 THE ADAPTIVE LINEAR FILTER

The samples $\{p_i\}$ are fed to the adaptive linear filter in Fig. 3.1. The sampled impulse response of the linear baseband channel, the sampler and the adaptive linear filter in Fig. 3.1 is given by the (g+1)-component vector

$$Y = [\, y_0 \quad y_1 \quad y_2 \quad . \quad . \quad . \quad y_g \,] \qquad \ldots \quad 3.4.1$$

with the z-transform

$$Y(z) = y_0 + y_1 z^{-1} + y_2 z^{-2} \quad + \quad . \quad . \quad . \quad . \quad + y_g z^{-g} \qquad \ldots \quad 3.4.2$$

where $\{y_i\}$ are complex values, and the delay in transmission over the baseband channel and the filter, other than that involved in the time dispersion of the received signal, is neglected here and $y_i = 0$ for 0>i or i>g.

The adaptive linear filter is a linear feedforward transversal filter that operates so that all the roots (zeros) of Y(z) lie inside or on the unit circle in the z-plane. The zeros of Y(z) are derived from the zeros of the z-transform of the sampled impulse response of the linear baseband channel, shown in Table 3.3, by replacing all the zeros of the channel that lie outside the unit circle by the complex conjugate of their reciprocals, leaving the remaining zeros unchanged [22,23,47]. Thus, the adaptive linear filter is an all pass network, with ideally an infinite number of taps, that adjusts the sampled impulse response of the linear baseband channel and filter to be minimum phase, without changing any amplitude distortion introduced by the channel. This in turn concentrates the energy of the channel and filter towards the earlier samples in such a way as to maximize the ratio of $|\ y_0\ |$ to the noise variance at the output of the filter [22,26]. It also removes all phase distortion introduced by the channel other than any that assists in making the impulse response minimum phase [47]. This filter is identical to the linear transversal filter that forms the first part of a conventional nonlinear (decision feedback) equalizer, where the latter is adjusted to minimize the mean square error (and hence maximize the signal/noise ratio) in its output signal, subject to the exact equalization of the channel [22]. A technique has been developed in [29], whereby the adjustment of the adaptive linear filter can be implemented simply and accurately. Thus, since no serious inaccuracy need in principle be introduced into the operation of the detector on account of the adaptive adjustment of the receiver, the correct adjustment of the adaptive filter is assumed throughout the work. Furthermore, the gain introduced by the adaptive linear filter is adjusted so that

$$y_0 = 1 \qquad \ldots \quad 3.4.3$$

where y_0 is the first component of the vector Y, which represents the sampled impulse response of the linear baseband channel and the adaptive linear filter. When an allowance has been made for the change in level introduced by the filter, the noise samples $\{w_i\}$ at its output have the same statistical properties as the noise samples $\{u_i\}$ at its input (see appendix E for the justification of this statement), furthermore, there is no change in the signal/noise ratio [22]. Thus the signal at the output of the adaptive linear filter, at time t=iT, is the complex valued quantity

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_h + w_i \qquad \ldots \quad 3.4.4$$

where the real and imaginary components of the $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and fixed variance (see appendix E).

Table 3.4 shows the sampled impulse response Y for the four telephone circuits used here in the test. The telephone circuits 1, 2, 3 and 4 are referred to as channels C, D, E and F, respectively. Table 3.4 gives an idea about the signal distortion introduced by each channel. The two remaining channels (not shown in Table 3.4) are channel A, which introduces no intersymbol interference, and channel B, which is an idealized partial response channel.

The sampled impulse response Y (Eqn. 3.4.1) is given by

$$Y = [1 \quad 0 \quad 0 \quad . \quad . \quad . \quad . \quad 0] \qquad \ldots \quad 3.4.5$$

for channel A, and

$$Y = [1 \quad 1 \quad 0 \quad . \quad . \quad . \quad . \quad 0] \qquad \ldots \quad 3.4.6$$

for channel B.

Channel B is used to compare the proposed detectors with an optimum detector, which can here be implemented as a Viterbi algorithm detector in the computer simulation tests, without requiring an unduly large number of operations.

Appendix F considers the signal/noise ratio used in the investigation and comments on the computer simulation techniques.

$\alpha_{i,1}$	$\alpha_{i,2}$	$\beta_{i-1,1}$	$\beta_{i-1,2}$	$\beta_{i,1}$	$\beta_{i,2}$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	1	1	1
0	0	1	0	1	0
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	1	1	0
0	1	1	0	0	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	1	0	0
1	1	1	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	1	0	1
1	0	1	0	1	1

Table 3.1 Differential encoding of 16-level QAM signal

State at time iT			Input at time iT		Output at time iT			State at time (i+1)T		
$u_{i,0}$	$u_{i,1}$	$u_{i,2}$	$\alpha_{i,1}$	$\alpha_{i,2}$	$\beta_{i,0}$	$\beta_{i,1}$	$\beta_{i,2}$	$u_{i+1,0}$	$u_{i+1,1}$	$u_{i+1,2}$
0	0	0	0	0	0	0	0	0	0	0
			1	0	0	1	0	0	0	1
			0	1	0	0	1	0	1	0
			1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0	1	0
			1	0	0	1	0	0	1	1
			0	1	0	0	1	0	0	0
			1	1	0	1	1	0	0	1
0	1	0	0	0	0	1	0	0	1	0
			1	0	0	0	0	0	1	1
			0	1	0	1	1	0	0	0
			1	1	0	0	1	0	0	1
1	1	0	0	0	0	1	0	0	0	0
			1	0	0	0	0	0	0	1
			0	1	0	1	1	0	1	0
			1	1	0	0	1	0	1	1
0	0	1	0	0	1	1	0	1	1	1
			1	0	1	0	0	1	0	0
			0	1	1	1	1	1	0	1
			1	1	1	0	1	1	1	0
1	0	1	0	0	1	1	0	1	0	1
			1	0	1	0	0	1	1	0
			0	1	1	1	1	1	1	1
			1	1	1	0	1	1	0	0
0	1	1	0	0	1	0	0	1	0	1
			1	0	1	1	0	1	1	0
			0	1	1	0	1	1	1	1
			1	1	1	1	1	1	0	0
1	1	1	0	0	1	0	0	1	1	1
			1	0	1	1	0	1	0	0
			0	1	1	0	1	1	0	1
			1	1	1	1	1	1	1	0

Table 3.2 Truth table of the convolutional encoder

Channel C (Telephone circuit 1)		Channel D (Telephone circuit 2)		Channel E (Telephone circuit 3)		Channel F (Telephone circuit 4)	
Real Part	Imaginary Part	Real Part	Imaginary Part	Real Part	Imaginary Part	Real Part	Imaginary Part
0.0326	-0.0045	0.0145	-0.0006	0.0176	-0.0175	-0.0038	-0.0049
0.5483	-0.0255	0.0750	0.0176	0.1381	-0.1252	0.0077	-0.0044
0.8031	0.0659	0.3951	0.0033	0.4547	-0.1885	0.0094	0.0207
-0.2430	-0.0286	0.7491	-0.1718	0.5078	0.1622	-0.0884	0.0355
0.0066	-0.0176	0.1951	0.0972	-0.1966	0.3505	-0.1138	-0.2869
0.0307	0.0180	-0.2856	0.1894	-0.2223	-0.2276	0.5546	-0.2255
-0.0170	-0.0115	0.0575	-0.2096	0.2797	-0.0158	0.1903	0.5813
0.0052	0.0056	0.0655	0.1139	-0.1636	0.1352	-0.2861	-0.0892
-0.0041	-0.0028	-0.0825	-0.0424	0.0594	-0.1400	0.2332	-0.0384
0.0021	0.0017	0.0623	0.0085	-0.0084	0.1111	-0.0652	0.0428
-0.0001	0.0001	-0.0438	0.0034	-0.0105	-0.0817	0.0335	-0.0519
-0.0017	-0.0004	0.0294	-0.0049	0.0152	0.0572	-0.0323	0.0170
0.0010	-0.0002	-0.0181	0.0032	-0.0131	-0.0406	0.0044	-0.0023
0.0006	0.0001	0.0091	0.0003	0.0060	0.0255	0.0054	0.0076
-0.0013	0.0000	-0.0038	-0.0023	0.0003	-0.0190	0.0008	-0.0051
0.0004	0.0002	0.0019	0.0027	-0.0035	0.0116	-0.0056	0.0001
0.0004	0.0000	-0.0018	-0.0014	0.0041	-0.0078	0.0018	0.0032
-0.0002	0.0001	0.0006	0.0003	-0.0031	0.0038	-0.0009	-0.0015
0.0001	-0.0004	0.0005	0.0000	0.0018	-0.0005	-0.0022	-0.0026
-0.0005	0.0003	-0.0008	-0.0001	-0.0018	-0.0005	0.0029	0.0019

Table 3.3 Sampled impulse responses of the linear baseband channels C-F in Fig. 3.1, for each of the four telephone circuits.

Channel C (Telephone circuit 1)		Channel D (Telephone circuit 2)		Channel E (Telephone circuit 3)		Channel F (Telephone circuit 4)	
Real Part	Imaginary Part	Real Part	Imaginary Part	Real Part	Imaginary Part	Real Part	Imaginary Part
1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
0.3412	0.0667	0.5091	0.1960	0.4861	1.0988	0.2544	1.9941
-0.1298	-0.0358	-0.1465	0.0000	-0.5980	0.0703	-1.7394	-0.2019
0.0263	0.0051	0.0323	-0.0171	0.1702	-0.1938	0.6795	-0.8086
0.0015	0.0008	0.0125	0.0200	-0.0245	0.1000	0.0408	0.5113
-0.0019	-0.0016	-0.0099	-0.0109	0.0100	-0.0258	-0.1189	-0.1463
-0.0017	0.0006	0.0046	0.0074	-0.0134	0.0110	0.0343	0.0420
-0.0011	-0.0004	-0.0069	-0.0083	0.0056	-0.0042	-0.0185	-0.0364
0.0018	0.0010	0.0059	0.0076	0.0003	0.0003	0.0139	0.0216
-0.0014	0.0000	-0.0025	-0.0053	-0.0008	0.0041	-0.0102	-0.0009
-0.0008	-0.0004	-0.0013	0.0040	0.0000	-0.0061	-0.0019	-0.0034
0.0016	-0.0001	0.0024	-0.0028	0.0007	-0.0007	0.0037	-0.0046
-0.0006	0.0001	-0.0009	0.0018	0.0037	0.0002	-0.0028	-0.0006
-0.0006	0.0003	-0.0006	-0.0006	-0.0019	-0.0025	-0.0019	-0.0046
0.0005	-0.0001	0.0001	-0.0003	0.0020	0.0008	0.0083	0.0022
0.0000	0.0004	0.0002	0.0008	0.0005	-0.0002	-0.0056	0.0059
0.0002	-0.0003	0.0000	-0.0006	-0.0022	0.0002	-0.0046	-0.0028
-0.0004	0.0000	-0.0003	-0.0001	0.0007	-0.0005	0.0049	-0.0019
-0.0001	0.0002	-0.0002	0.0003	-0.0008	0.0002	-0.0009	0.0037
0.0004	-0.0001	0.0003	-0.0002	0.0005	0.0005	-0.0009	0.0003

Table 3.4 Sampled impulse responses of the linear baseband channel and adaptive linear filter in Fig. 3.1, for each of the four telephone circuits.

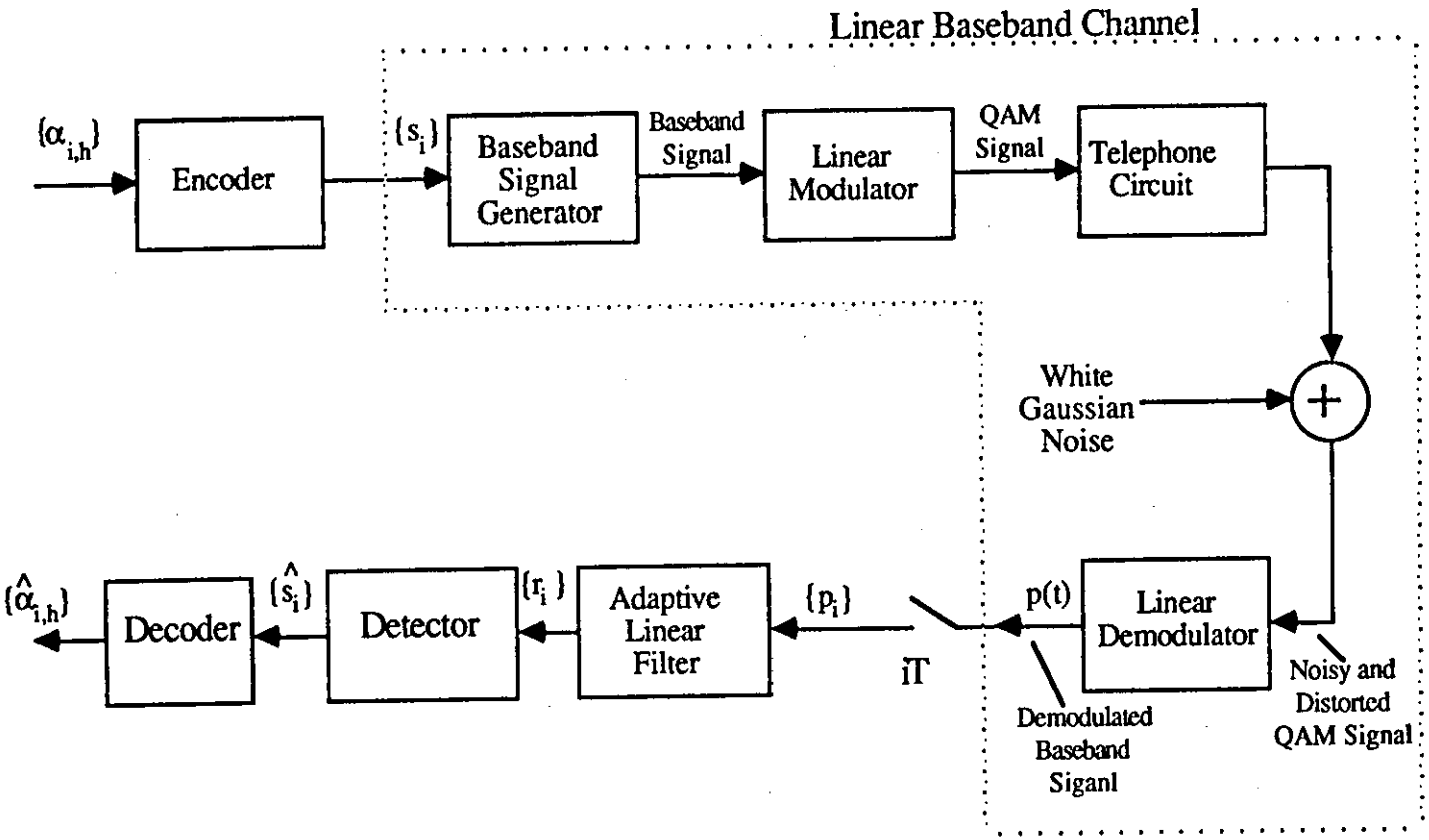


Fig.3.1 Model of data transmission system

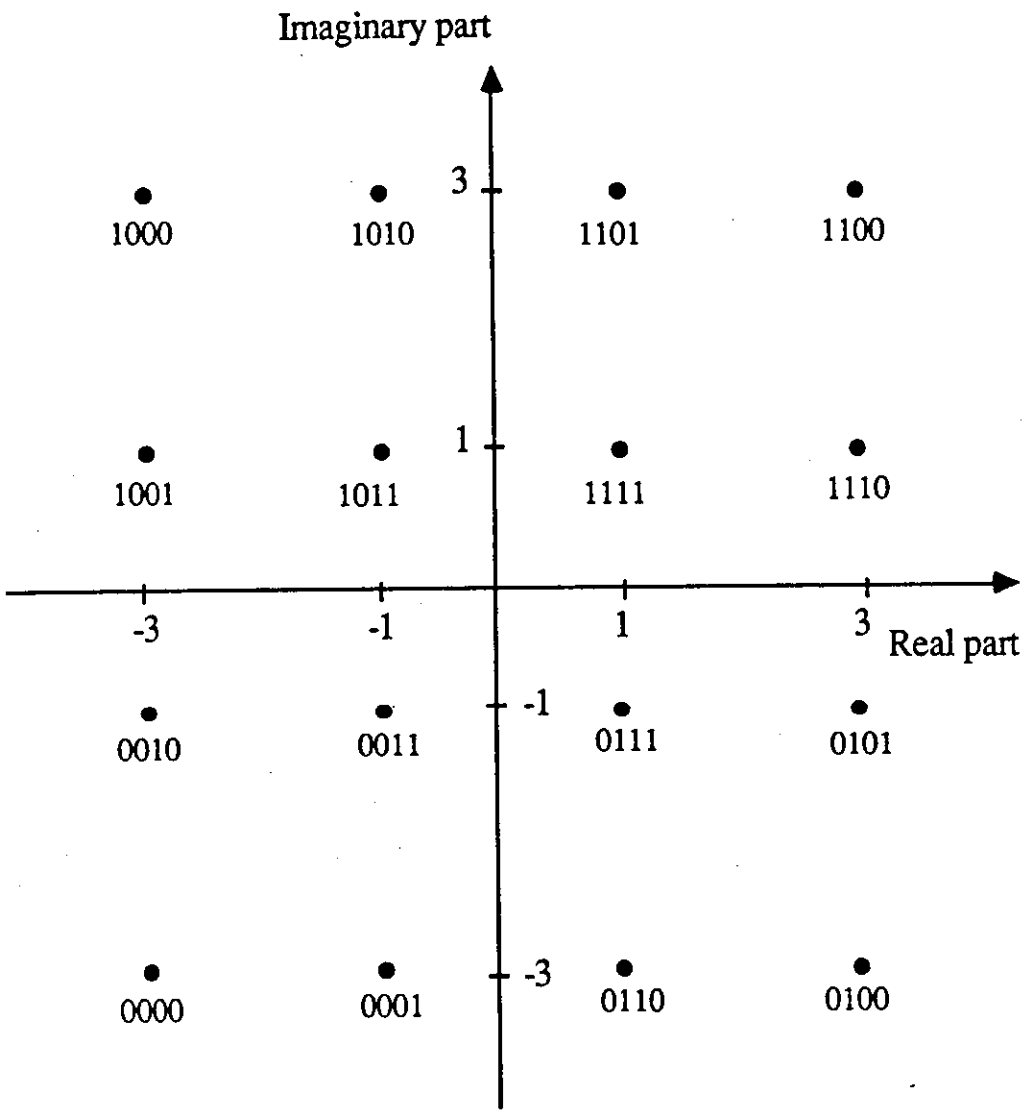


Fig.3.2 Signal constellation for an uncoded data symbol s_i

(the binary coded number against each point is $\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}$)

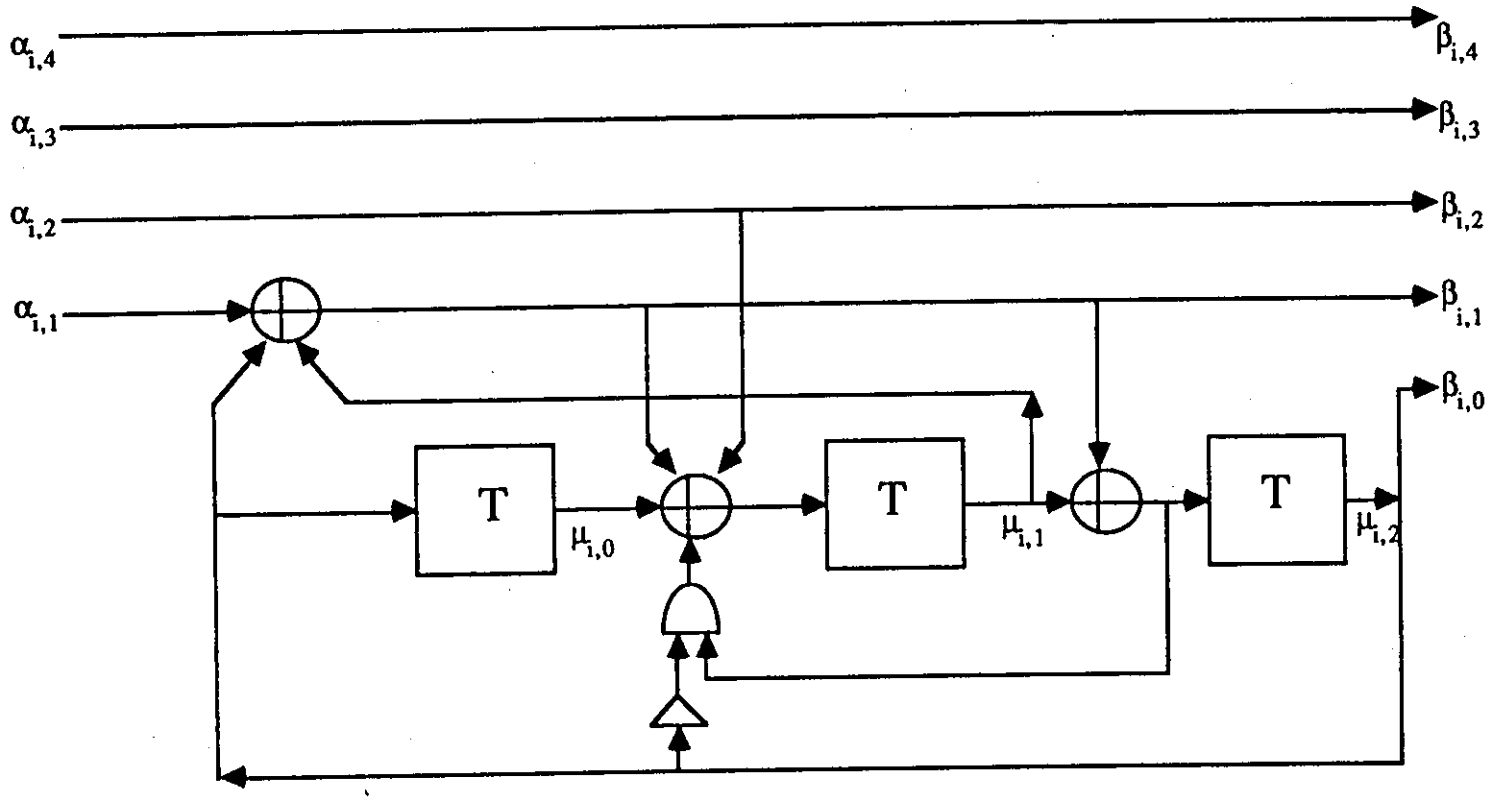


Fig.3.3 G8 convolutional encoder

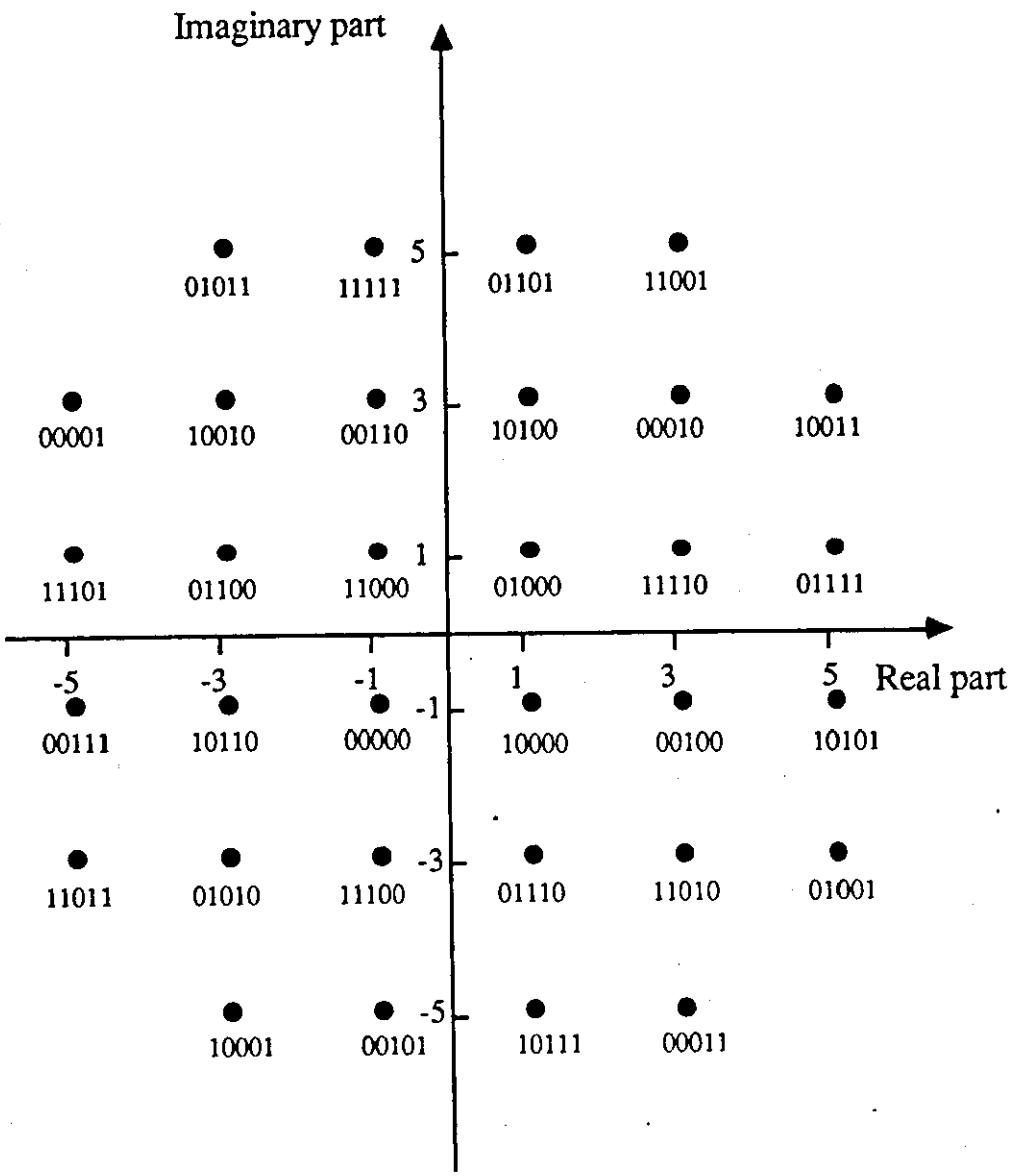


Fig.3.4 Signal constellation for a coded data symbol s_i .

(the binary coded number against each point is $\beta_{i,0}\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}$)

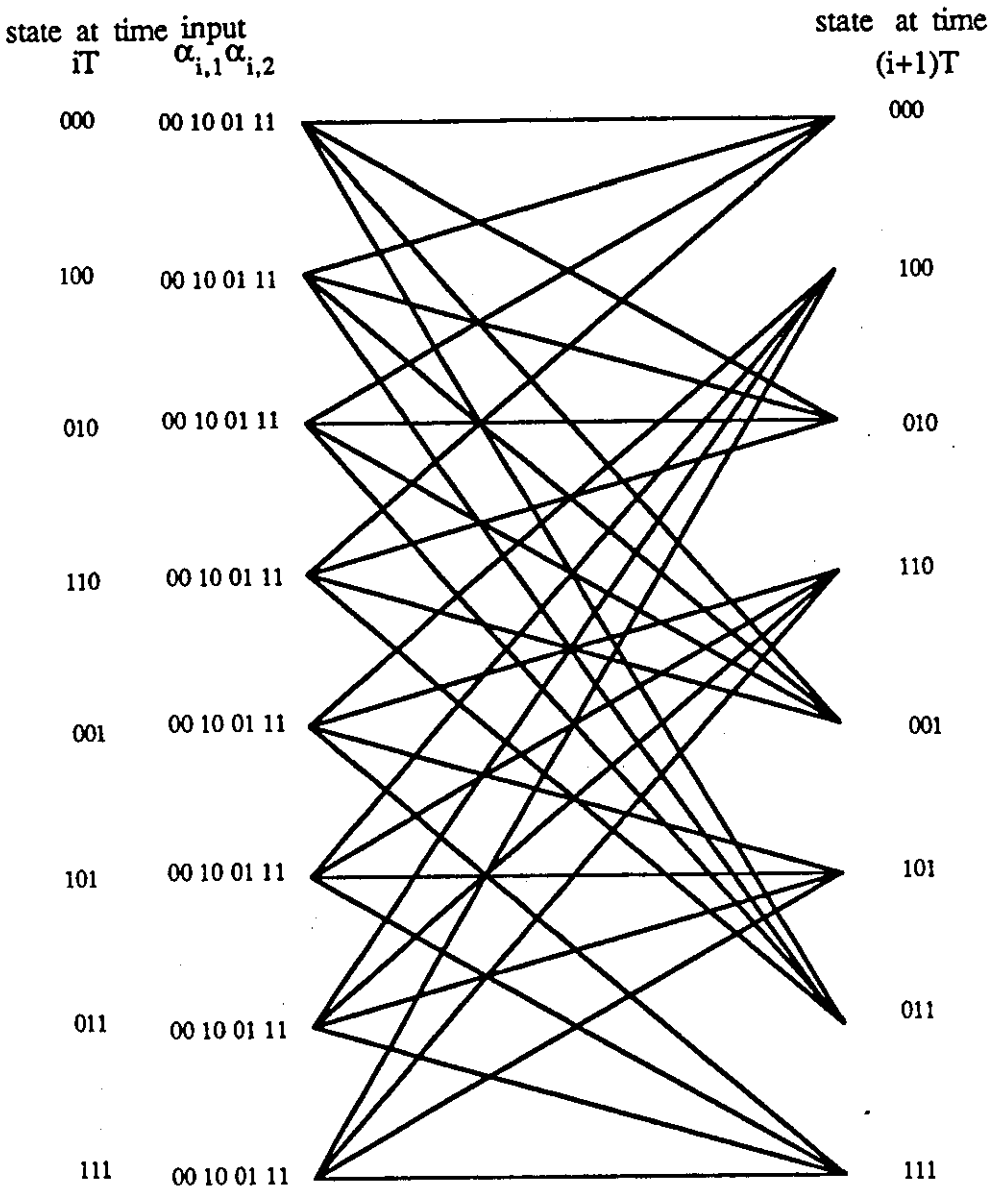
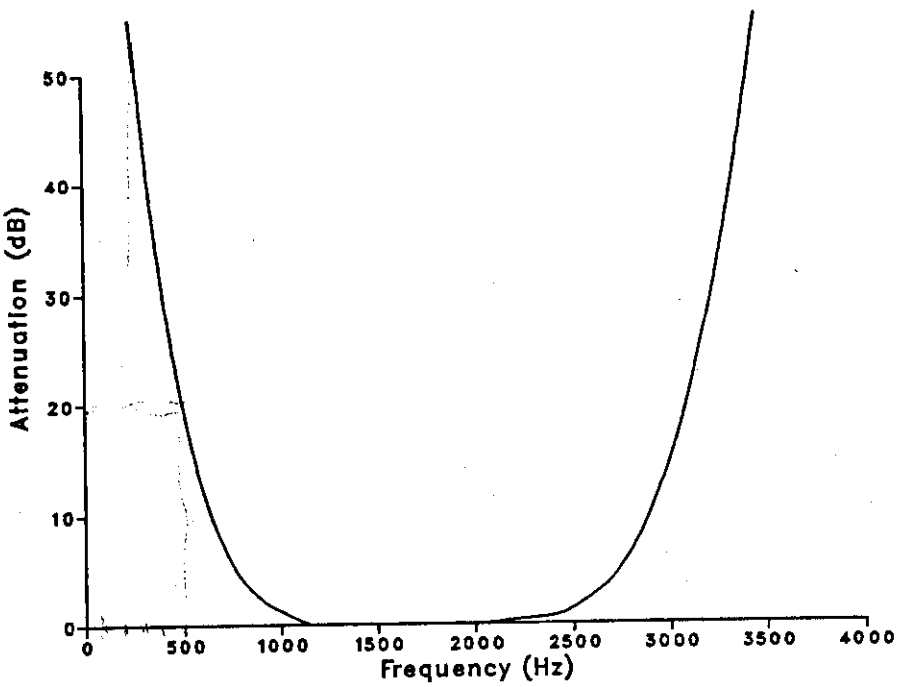
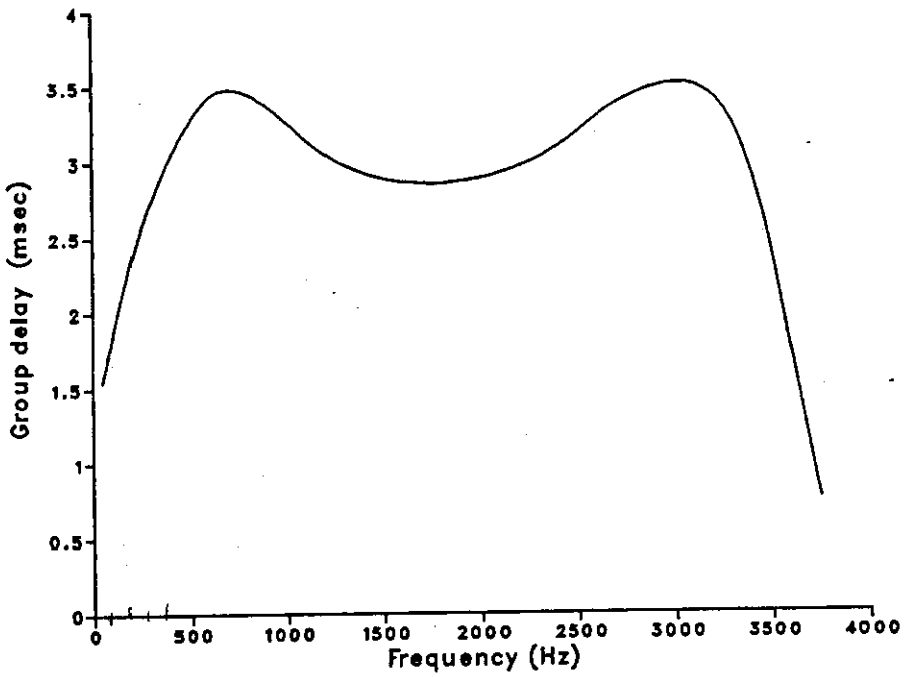


Fig.3.5 State transition diagram of G8 encoder.

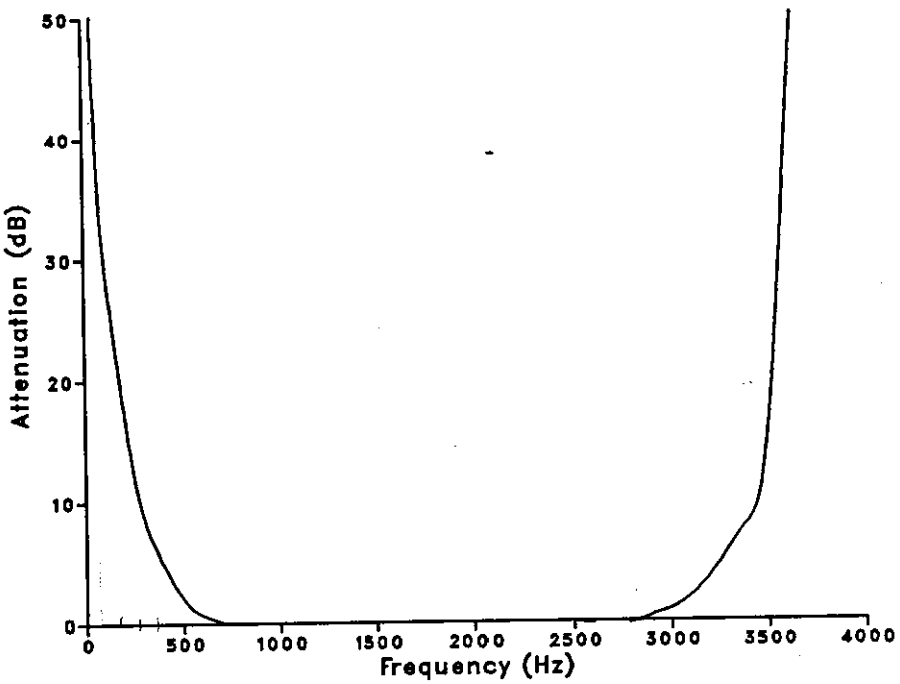


a – Attenuation characteristic

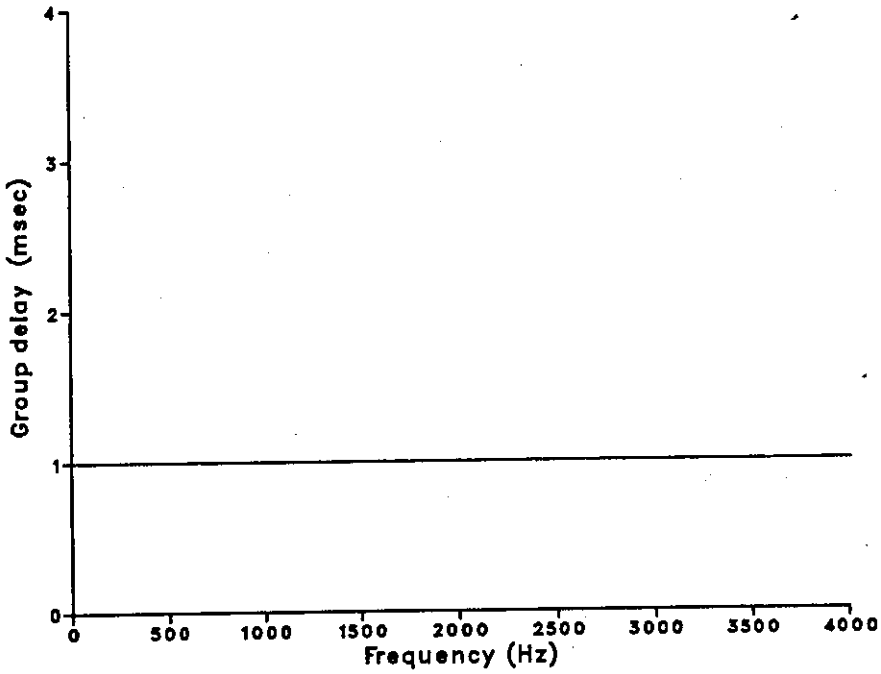


b – Group delay characteristic

Fig.3.6 Attenuation and group delay characteristics of the equipment filters–1

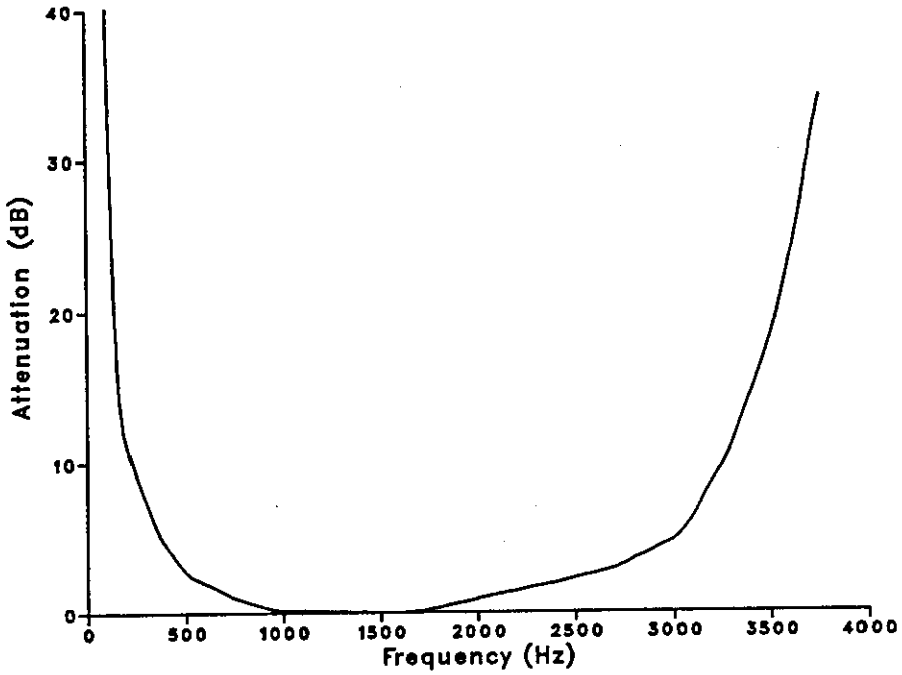


a – Attenuation characteristic

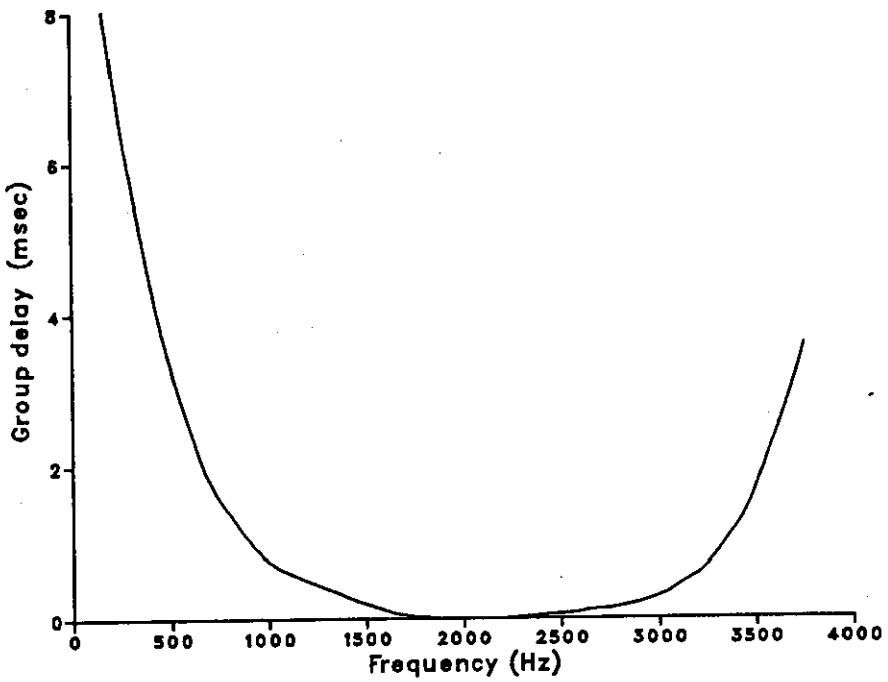


b – Group delay characteristic

Fig.3.7 Attenuation and group delay characteristics of telephone circuit 1.

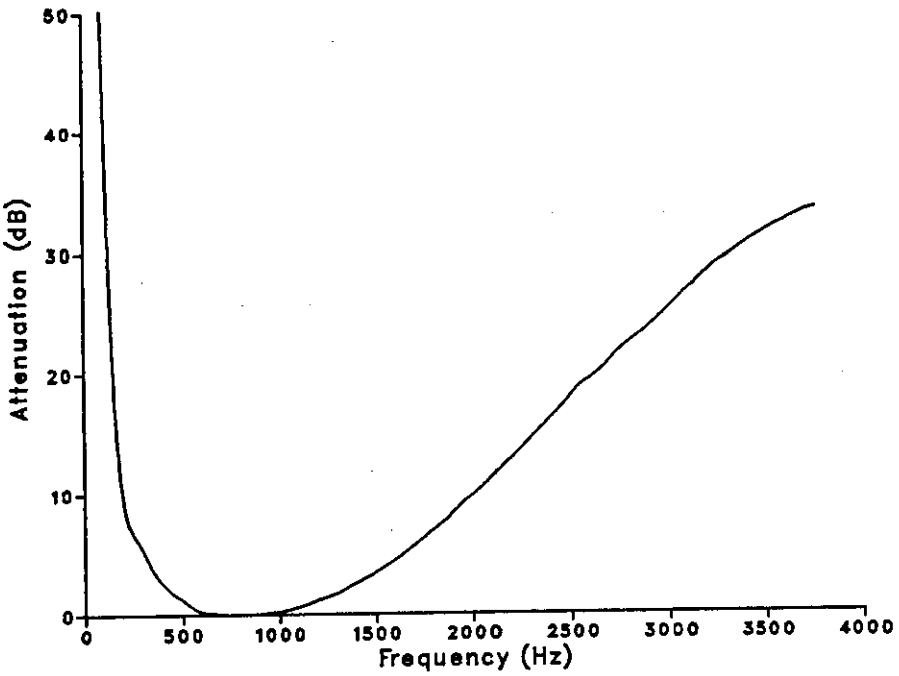


a – Attenuation characteristic

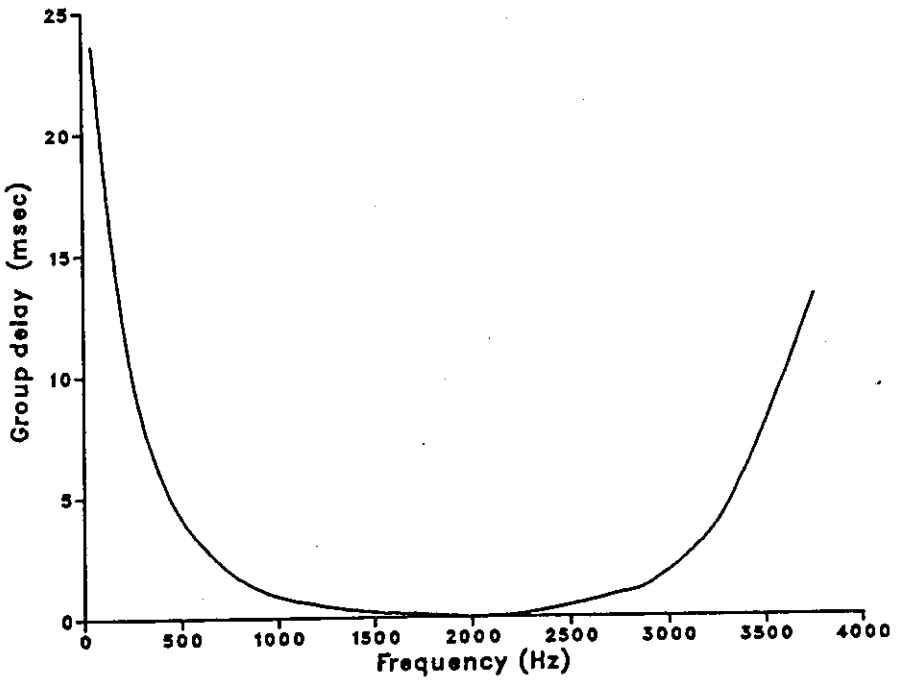


b – Group delay characteristic

Fig.3.8 Attenuation and group delay characteristics of telephone circuit 2.

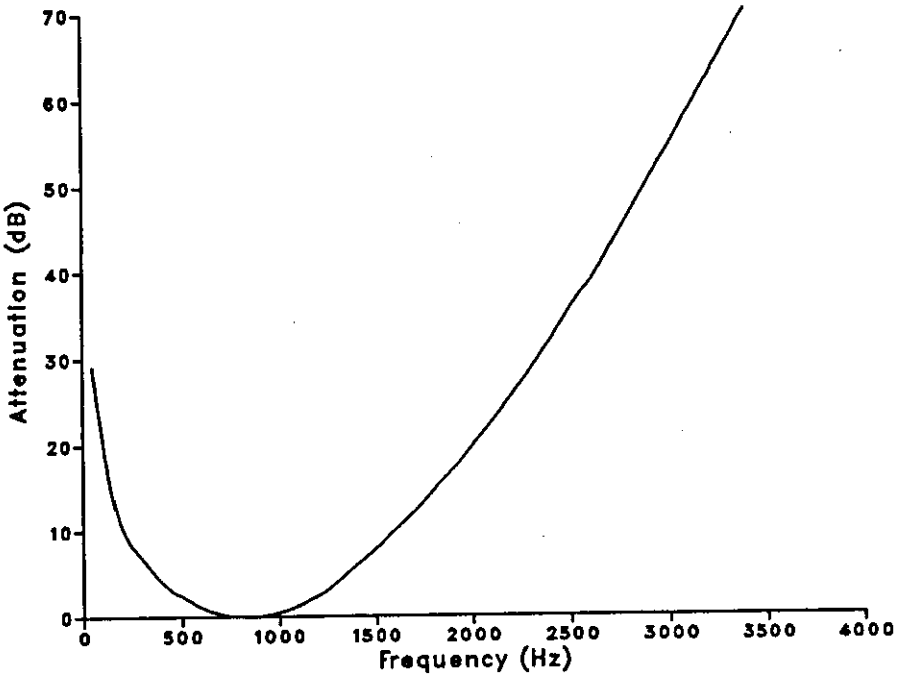


a – Attenuation characteristic

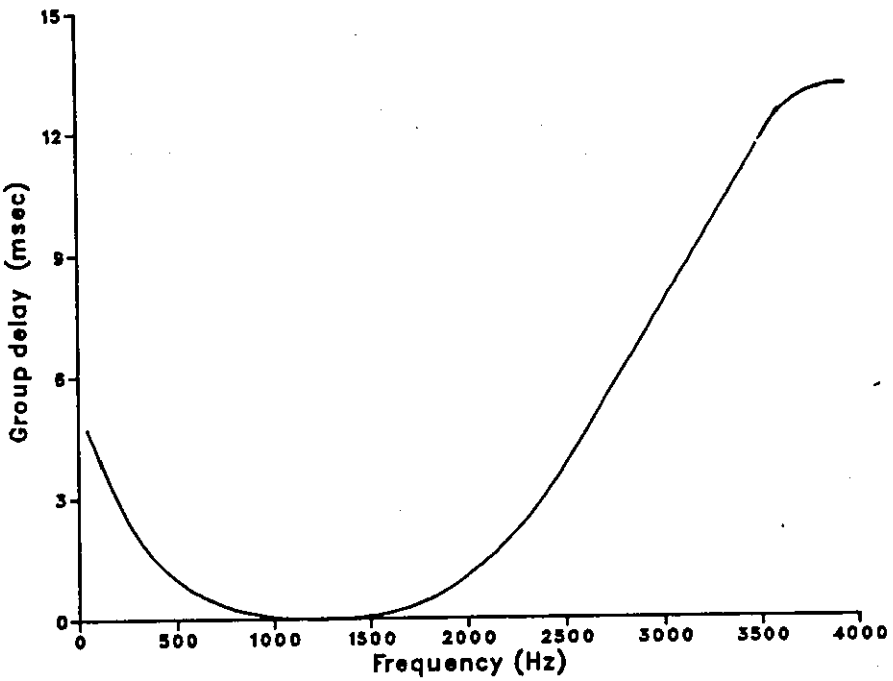


b – Group delay characteristic

Fig.3.9 Attenuation and group delay characteristics of telephone circuit 3.



a – Attenuation characteristic



b – Group delay characteristic

Fig.3.10 Attenuation and group delay characteristics of telephone circuit 4.

CHAPTER 4

DETECTION PROCESSES FOR AN UNCODED 16-LEVEL QAM SIGNAL

4.1 INTRODUCTION

In this chapter, the equalizer and near-maximum likelihood detectors, used in the detection of the uncoded 16-level QAM signal are described and some results of the computer simulation tests are presented. The detectors for the uncoded system are used here as references against which proposed detection processes for a convolutionally encoded signal are compared. The model of the data transmission system is as described in Chapter 3. The information digits $\{\alpha_{i,h}\}$ are differentialy encoded, and then mapped into data symbols $\{s_i\}$, which form a 16-level QAM signal as shown in Fig. 3.2. The sampled impulse response of the linear baseband channel and the adaptive linear filter is given by the (g+1)-component vector Y

$$Y = [y_0 \quad y_1 \quad y_2 \quad . \quad . \quad . \quad . \quad y_g \quad] \qquad \ldots \quad 4.1.1$$

where $y_0 = 1$.

The sample value of the received signal at the output of the adaptive linear filter, at time t=iT, is given by

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_h + w_i \qquad \ldots \quad 4.1.2$$

where the real and imaginary components of the noise samples $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and fixed variance (Chapter 3 and Appendix A). The detectors operate on the received sequence $\{r_i\}$ to produce the detected sequence $\{\hat{s}_i\}$, using a knowledge of both the sampled impulse response Y and of the possible values of s_i. As stated in Chapter 3, it is assumed that the sampled impulse response of the channel is time invariant, and the adaptive linear filter is ideally adjusted.

4.2 LINEAR FEEDBACK EQUALIZER (SYSTEM LU)

Although, the linear equalizer considered in Chapter 2 is a linear feedforward transversal filter, which is the type normally used in practice, the linear equalization of the channel may alternatively be achieved by means of a feedback transversal filter, provided that all the roots of the z-transform of the sampled impulse response of the channel lie inside the unit circle of the z-plane [22]. The latter condition is satisfied here by the ideal adjustment of the adaptive linear filter (Chapter 3). The object here is to compare the performances of the detectors, described later, with that of an ideal linear equalizer and not to investigate the adjustment of the equalizer itself. The structure of the linear feedback equalizer is shown in Fig. 4.1. If the z-transform of Y (Eqn. 4.1.1) is Y(z), the z-transform of the equalizer in Fig. 4.1 is 1/Y(z). It can be shown that such equalizer is stable if and only if all the roots (zeros) of Y(z) lie inside the unit circle in the z-plane, and when this condition is satisfied ,

$$\frac{1}{Y(z)} = (y_0 + y_1 z^{-1} + y_2 z^{-2} + . \quad . \quad . \quad + y_g z^{-g})^{-1} \qquad \ldots \quad 4.2.1$$

where $y_0 = 1$. Under this condition the equalizer achieves the accurate equalization of the channel with only g taps [22]. On the other hand, the corresponding feedforward equalizer can in principle only achieve the exact equalization when the number of taps is infinity.

The equalized signal at the input to the threshold detector, (shown in Fig. 4.1) at time t=iT is given by

$$e_i = r_i - \sum_{h=1}^{g} e_{i-h}\, y_h \qquad \ldots \quad 4.2.2$$

where r_i is given by Eqn. 4.1.2. The threshold levels of the detector, for the 16-level QAM signal, are placed at the values of -2, 0 and 2 for each of the real and imaginary axes. The detector then selects one of the 16 possible values of s_i which is closest to the equalized sample e_i and it takes this value as the detected value of the data symbol s_i . The detected value of s_i uniquely determines the detected binary digits $\beta_{i,1}, \beta_{i,2}, \beta_{i,3}$ and $\beta_{i,4}$. The later digits together with the values of $\beta_{i-1,1}, \beta_{i-1,2},$

which are already known at time t=(i-1)T, are used to determine the detected binary digits $\hat{\alpha}_{i,1}, \hat{\alpha}_{i,2}, \hat{\alpha}_{i,3}$ and $\hat{\alpha}_{i,4}$ of the binary digits $\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}$ and $\alpha_{i,4}$ respectively. This is done with the help of Table.3.1 as shown Chapter 3.

4.3 NONLINEAR EQUALIZER (SYSTEM NU)

The nonlinear equalizer used here is the one described in Chapter 2. The linear feedforward transversal filter that forms the first part of this equalizer, is the adaptive linear filter. The latter is assumed to be ideally adjusted to perform its operation as described in Chapter 3 so that all the roots of Y(z) lie inside the unit circle of the z-plane. As shown in Fig. 4.2, the tap gains of the feedback filter in the nonlinear equalizer are given by the last g components of the vector Y (Eqn. 4.1.1), where $y_0 = 1$. The equalized sample at the input to the threshold level detector in Fig. 4.2, at time t=iT, is given by

$$e_i = r_i - \sum_{h=1}^{g} \hat{s}_{i-h}\, y_h \qquad \ldots \quad 4.3.1$$

where $\hat{s}_{i-h}$ is the detected value of s_{i-h}, for h=1,2, . . . g.
The operation of the detector and the decoding process are identical to that described in the linear feedback equalizer (Section 4.2)

4.4 NEAR-MAXIMUM LIKELIHOOD DETECTORS

4.4.1 SYSTEM 1U

The near-maximum likelihood detector system 1U is known as a pseudoquaternary detection process [23,45,46]. Just prior to the receipt of r_i (Eqn. 4.1.2) the detector in system 1U holds in store k n-component vectors $\{Q_{i-1}\}$, where k can take any suitable integer and

$$Q_{i-1} = [x_{i-n} \quad x_{i-n+1} \quad . \quad . \quad . \quad x_{i-1}] \qquad \ldots \quad 4.4.1$$

and x_{i-k} takes any one of the 16 possible values of s_{i-k} (Fig. 3.2). It is assumed here that $n \geq g$ and when i<0 , $x_i = 0$. . The n-component vector Q_{i-1} forms the last n components of the i-component vector

$$X_{i-1} = [x_0 \quad x_1 \quad . \quad . \quad . \quad x_{i-1}] \qquad \ldots \quad 4.4.2$$

which represents a possible sequence of the transmitted data symbols $\{s_j\}$. Associated with each vector Q_{i-1} is stored its cost, which is taken to be the cost of the corresponding vector X_{i-1}, given by

$$c_{i-1} = \sum_{k=0}^{i-1} \left| r_k - \sum_{j=0}^{g} x_{k-j}\, y_j \right|^2 \qquad \ldots \quad 4.4.3$$

where $|u|$ is the absolute value of u. It can be shown [23,26] that, under the assumed conditions and for the given received sequence $\{r_i\}$, the vector X_{i-1} most likely to be correct is that which has the smallest cost (value of c_{i-1}), over all combinations of the possible values of its i components.

On the receipt of r_i , at time t=iT, each vector Q_{i-1} is used to form 4 (n+1)-component vectors $\{P_i\}$, where

$$P_i = [x_{i-n} \quad x_{i-n+1} \quad . \quad . \quad . \quad x_{i-1} \quad x_i] \qquad . \qquad \ldots \quad 4.4.4$$

The first n components of each of four vectors $\{P_i\}$, derived from any one Q_{i-1} , are as in the original vector Q_{i-1} , and the last components of the vectors $\{P_i\}$, take on the four different values of their 16 possible values. The cost of each vector P_i , is given by

$$c_i = c_{i-1} + \left| r_i - \sum_{k=0}^{g} x_{i-k}\, y_k \right|^2 \qquad \ldots \quad 4.4.5$$

where c_{i-1} is the cost of the original vector Q_{i-1} as in Eqn. 4.4.3. The four values of x_i in the four vectors $\{P_i\}$, which are derived from any one vector Q_{i-1} , are determined by all combinations of the two adjacent possible real parts of x_i giving the smallest costs $\{c_i\}$, for any fixed value of the imaginary part, and the two adjacent possible imaginary parts of x_i giving the smallest costs, for any fixed value of the real part [46]. The selection of the four values of x_i here can be achieved very simply, by threshold level comparisons without requiring the evaluation of any cost [23]. After expanding each vector Q_{i-1} , the detector holds 4k vectors $\{P_i\}$ together with their costs $\{c_i\}$. The detector then selects the vector P_i with the smallest cost and takes its

first component x_{i-n} as the detected value $\hat{s}_{i-n}$ of the data symbol s_{i-n}. The value of $\hat{s}_{i-n}$ is then used to determine the detected values of the information digits $\alpha_{i-n,1}, \alpha_{i-n,2}, \alpha_{i-n,3}$ and $\alpha_{i-n,4}$ by using the truth table of the differential encoder as before. Clearly, the delay in the detection is nT seconds, where T is the symbol interval. All vectors $\{P_i\}$ for which $\hat{s}_{i-n} \neq x_{i-n}$ are now discarded from any future processing, and the first components of all remaining vectors $\{P_i\}$ (including that with the smallest cost) are then omitted, to give the corresponding n-component vectors $\{Q_i\}$ where

$$Q_i = [x_{i-n+1} \quad x_{i-n+2} \quad . \quad . \quad . \quad x_{i-1} \quad x_i] \qquad \ldots \quad 4.4.6$$

The cost of the vector Q_i is the same as that of the vector P_i from which it was derived. The detector then selects from the resulting vectors $\{Q_i\}$ the k vectors with the lowest costs $\{c_i\}$. The k vectors Q_i together with their costs are stored, ready for the next detection process. To prevent overflow due to the steady increase in costs over any one transmission, the smallest cost is subtracted from the cost of each vector after each detection process. Thus the value of the smallest cost is always reduced to zero. This process is carried out for all the near-maximum likelihood detectors investigated here. The discarding of the given vectors $\{P_i\}$ is used here to prevent the merging (becoming the same) of any of the stored vectors, since it ensures that if these are all different at the start of transmission, no two or more of them can subsequently become the same. A suitable starting up procedure, for the detector and the differential decoder, is to begin with k stored vectors Q_{i-1} that are all the same and correct. In practice a synchronization procedure must always be used at the start of transmission, with the data symbols taking on at least some of their possible values and, therefore, not being set to zero. A zero cost is allocated to one of the vectors and a very high cost to each of the remaining vectors. After a few received samples, the detector must hold k vectors which are all different and are all derived from the original vector Q_{i-1} with zero cost. The starting up procedure, just described, and the process of discarding of the vectors, are carried out in all near-maximum likelihood detectors considered here.

4.4.2 SYSTEM 2U

The near-maximum likelihood detector here operates as the previous detector (system 1U), the only exception being that the number of vectors $\{P_i\}$ derived from each vector Q_{i-1} becomes a function of the relative cost of that vector Q_{i-1}. This is a development of the technique previously proposed for the given application [45].

Just prior to the receipt of r_i , the detector holds in store k n-component vectors $\{Q_{i-1}\}$ (Eqn. 4.4.1) together with their costs $\{c_{i-1}\}$ (Eqn. 4.4.3), where k here is an integral multiple of 4. The vectors are arranged according to the value of their costs $\{c_{i-1}\}$, so that the first stored vector Q_{i-1} is the one which has the smallest cost and the second vector has the second smallest cost and so on.

On the receipt of r_i , each vector Q_{i-1} is expanded into a number of the corresponding vectors $\{P_i\}$ (Eqn. 4.4.4) as follows: the first group of k/4 vectors, with the lowest k/4 costs, are each expanded into four vectors $\{P_i\}$. The second group of k/4 vectors $\{Q_{i-1}\}$ are each expanded into three vectors $\{P_i\}$. Each of the third group of k/4 vectors $\{Q_{i-1}\}$ (i.e the (k/2 +1)th upto the (3k/2)th vector) is expanded into two vectors $\{P_i\}$, and the last group of k/4 vectors $\{Q_{i-1}\}$ in the store are each expanded into only one vector P_i . In any case the vectors $\{P_i\}$ derived from any particular vector Q_{i-1} are those with the smallest cost. The expansion of the vectors in system 2U, just described, is shown in Fig. 4.3. For example, when k=8 the detector expands each of the first and the second vectors $\{Q_{i-1}\}$ into four vectors $\{P_i\}$, each of the third and the fourth into three vectors, while the fifth and the sixth are each expanded into two vectors and the last two vectors, i.e the seventh and the eighth, are each expanded into one vector only. For each vector P_i , the detector then evaluates the cost c_i according to Eqn. 4.4.5. And from the resulting 5k/2 vectors $\{P_i\}$, the detector then selects the vector P_i with the smallest cost and it takes the first component x_{i-n} of this vector as the detected value of $\hat{s}_{i-n}$ of the data symbol s_{i-n} . As before the detected value of s_{i-n} is then used to determine the detected value of the information digits $\alpha_{i-n,1}, \alpha_{i-n,2}, \alpha_{i-n,3}$ and $\alpha_{i-n,4}$. After discarding of all vectors $\{P_i\}$ for which $x_{i-n} \neq \hat{s}_{i-n}$, the detector omits the first component of each of the remaining vectors $\{P_i\}$ to give the corresponding n-component vector Q_i (Eqn. 4.4.6). The detector then selects k vectors $\{Q_i\}$ from the remaining vectors and it arranges them according to

the values of their costs. Clearly, the detector in system 2U operates with a smaller number of vectors $\{P_i\}$, and therefore it evaluates a smaller number of costs $\{c_i\}$, than that required by system 1U for a given value of k (the number of the stored vectors $\{Q_{i-1}\}$ in the detector).

4.4.3 SYSTEM 3U

The detector in system 3U operates in a rather different manner from that of system 1U or 2U and it is a modification of a detector known as "system B" in [45].

Just prior to the receipt of r_i , the detector holds in store k n-component vectors $\{Q_{i-1}\}$ (Eqn. 4.4.1) together with their costs $\{c_{i-1}\}$ (Eqn. 4.4.3), where k here is a multiple of 8. The k vectors $\{Q_{i-1}\}$ are arranged into k/2 groups, as follows; Each of the first k/8 groups has four vectors, with the same values of $x_{i-n}, x_{i-n+1}, \ldots . x_{i-2}$ in any one group, but different values of x_{i-1} . Each of the next k/8 groups has two vectors, again with the same values of $x_{i-n}, x_{i-n+1}, \ldots . x_{i-2}$, in any one group, but different values of x_{i-1} . Each of the last k/4 groups has just one vector. The arrangement of these groups is shown in Fig. 4.4.

On the receipt of r_i at time t=iT, each vector Q_{i-1} is expanded into the vector P_i (Eqn. 4.4.4) with the smallest cost, resulting in k vectors $\{P_i\}$ together with their cost $\{c_i\}$ (Eqn. 4.4.5). From these vectors, the detector next selects the vector P_i with the smallest cost and takes its first component x_{i-n} as the detected value $\hat{s}_{i-n}$ of the data symbol s_{i-n} . As before, the detector then uses the value of $\hat{s}_{i-n}$ to determine the detected values of the corresponding information digits. Any vector for which $x_{i-n} \neq \hat{s}_{i-n}$ is now discarded and from the remaining vectors the detector then selects k/2 vectors $\{P_i\}$ with the smallest costs. The first components of the selected vectors are then omitted to give the corresponding vectors $\{Q_i\}$. The vectors $\{Q_i\}$ are then arranged according to the values of their costs. To each of the first k/8 vectors $\{Q_i\}$, the detector adds three vectors $\{Q_i\}$ differing only in the last component and, usually, with the smallest costs [23]. For each of the second k/8 vectors $\{Q_i\}$ is added one

vector Q_i differing only in the last component and with the smallest cost [47]. No vectors are added to the last k/4 vectors $\{Q_i\}$, of the original k/2 vectors. There are now k vectors $\{Q_i\}$, together with their costs, ready for the next detection process.

The simulation program of this detector (system 3U) is given in Appendix G1.

4.5 THE COMPLEXITY AND THE EFFECTS OF THE PARAMETERS k AND n ON THE PERFORMANCE OF THE DETECTORS

In this section, the complexity of the near-maximum likelihood detectors, described in this Chapter, are examined and the performances of the detectors are presented with different values of k (the number of stored vectors held by the detector) and n (the delay in detection, in terms of symbol intervals). Only channel E (see Chapter 3) is considered here. This channel includes telephone circuit 3, which represents a typical worst case circuit normally considered for the transmission of data at a rate of 9600 bit/s [23,47]. The complete results of the computer simulation tests for all systems, including the equalizers, are presented later in Chapter 5, for all channels considered in this work.

An approximate assessment of the complexity requirements of the three detectors (systems 1U, 2U and 3U) and the Viterbi algorithm detector (system VU) are given in Table 4.1. The systems are listed in decreasing order of their complexity and starting with the most complex detector, which is system VU. System VU has already been described in Chapter 2. The evaluation of a cost here (Eqn. 4.4.5) is a considerably more complex process than is one of the operations involved in a search through the costs. It is clear (Table 4.1) that the near-maximun likelihood detectors, described in this chapter, involve a great reduction in the complexity requirements when compared to that required by the Viterbi algorithm detector, for moderate values of g.

Figs. 4.5, 4.6 and 4.7 show the performances of systems 1U, 2U and 3U, respectively. System 1U and 2U are tested here with k= 4, 8 and 16, whereas system 3U is tested with k= 8 and 16. The delay in the detection is 32 symbol intervals

(n=32). The performance of the nonlinear equalizer (system NU) is shown in each figure for the sake of comparison. The definition of the signal/noise ratio (ψ) and the accuracy of the results are described in Appendix F, where $\lambda = 10$ and m=4 in Eqn.F.6. It is clear from the above figures, that an advantage of about 4 dB in tolerance of the detectors to additive white Gaussian noise can be gained over the nonlinear equalizer (system NU) at bit error rate of 10^{-4} . When k=1 and n=0, the near-maximum likelihood detector degenerates into a nonlinear equalizer. The performance of the linear equalizer (system LU) for channel E is much worse than that of the nonlinear equalizer, and is presented later in Chapter 5. The above figures also show that, for each detector, when the value of k is doubled, an advantage of about 0.3 to 0.5 dB can be achieved in tolerance of the system to additive noise. And for a given value of k, the performance of the three systems (1U, 2U and 3U) is approximately the same when k=8 or 16. But according to Table 4.1, system 3U requires about three eighths the number of cost evaluations required by system 1U and three fifths of that required by system 2U. Furthermore, for the same value of k, system 3U requires a smaller number of searches through the costs than that required by system 1U or 2U. It follows that, system 3U can give better performance than that of system 1U or 2U for a given complexity.

Figs. 4.8 to 4.10 show the effect of changing the delay in the detection on the performance of the above three systems, when operating with 8 stored vectors (k=8). It is clear from these figures and the performance of the nonlinear equalizer that there is still an advantage gained by the near-maximum likelihood detectors, tested here, over the nonlinear equalizer even when there is no delay in detection (n=0). Generally this advantage is of about 1 dB at bit error rate of 10^{-4} . When $n<g$ (for channel E, g=19), the length of the vector Q_{i-1} must be at least 19 components, so that the cost c_{i-1} can be evaluated successfully according to Eqn. 4.4.5. So, in obtaining the results in Figs. 4.8 to 4.10, the length of the vector is fixed at 32 components, and the detection is performed at a delay of n symbol intervals. The value of n is varied from 0 to 32. Although, the best performance of each system is obtained when n is 32, which is the largest value of n tested here, an increase of n beyond 8 does not show great improvement in the performances of the systems over the range of the bit error rates tested here.

Detector	No. of stored vectors $\{Q_i\}$	No. of expanded vectors $\{P_{i+1}\}$	No. of cost evaluations	No. of searches	No. of costs searched through
Viterbi algorithm (system VU)	16^k	16^{k+1}	16^{k+1}	16	16^{k+1}
System 1U	k	4k	4k	k	4k
System 2U	k	2.5k	2.5k	k	2.5k
System 3U	k	k	1.5k	0.5k	k

Table 4.1 Complexities of different detectors for uncoded 16-level QAM signal.

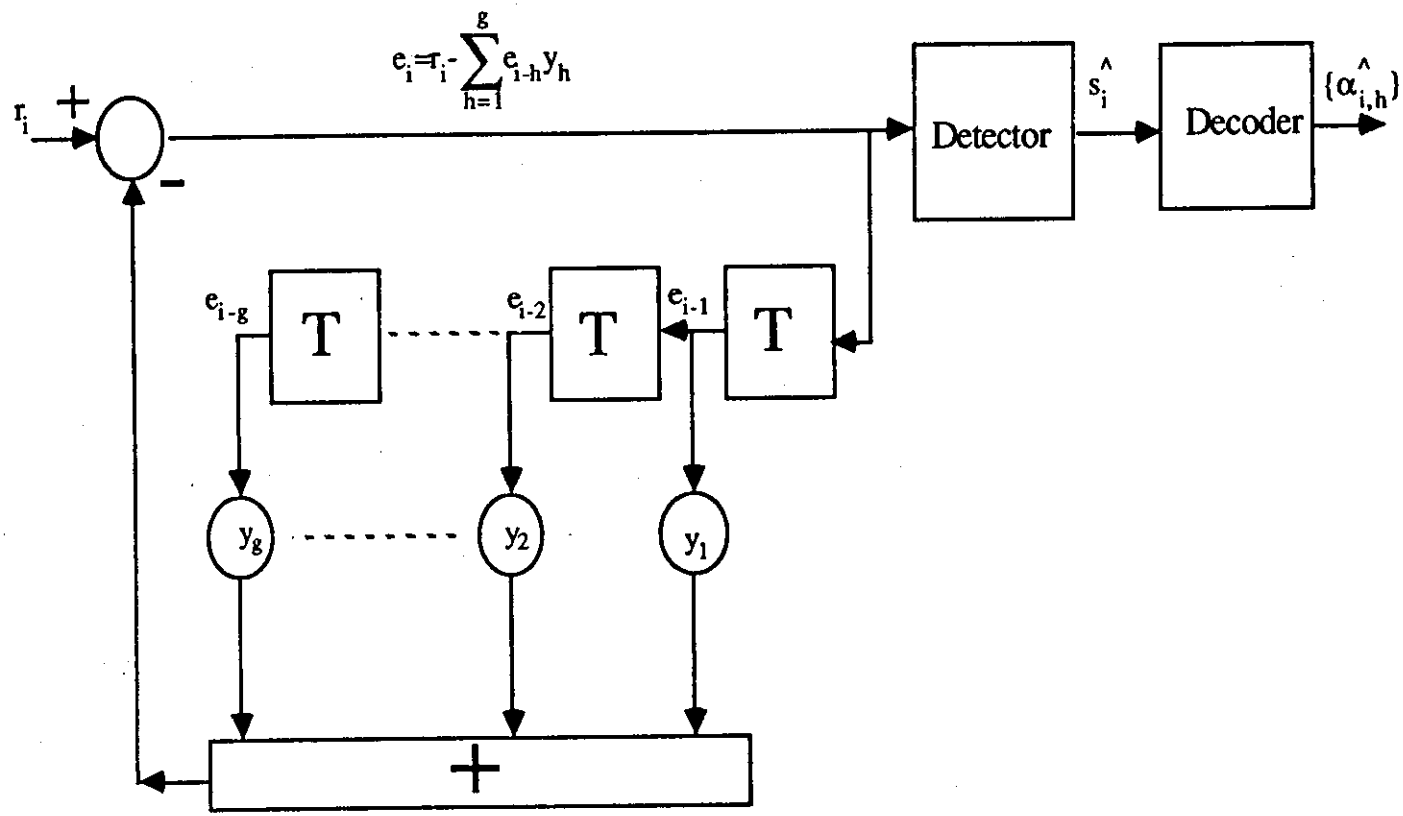


Fig.4.1 The linear feedback equalizer and detector .

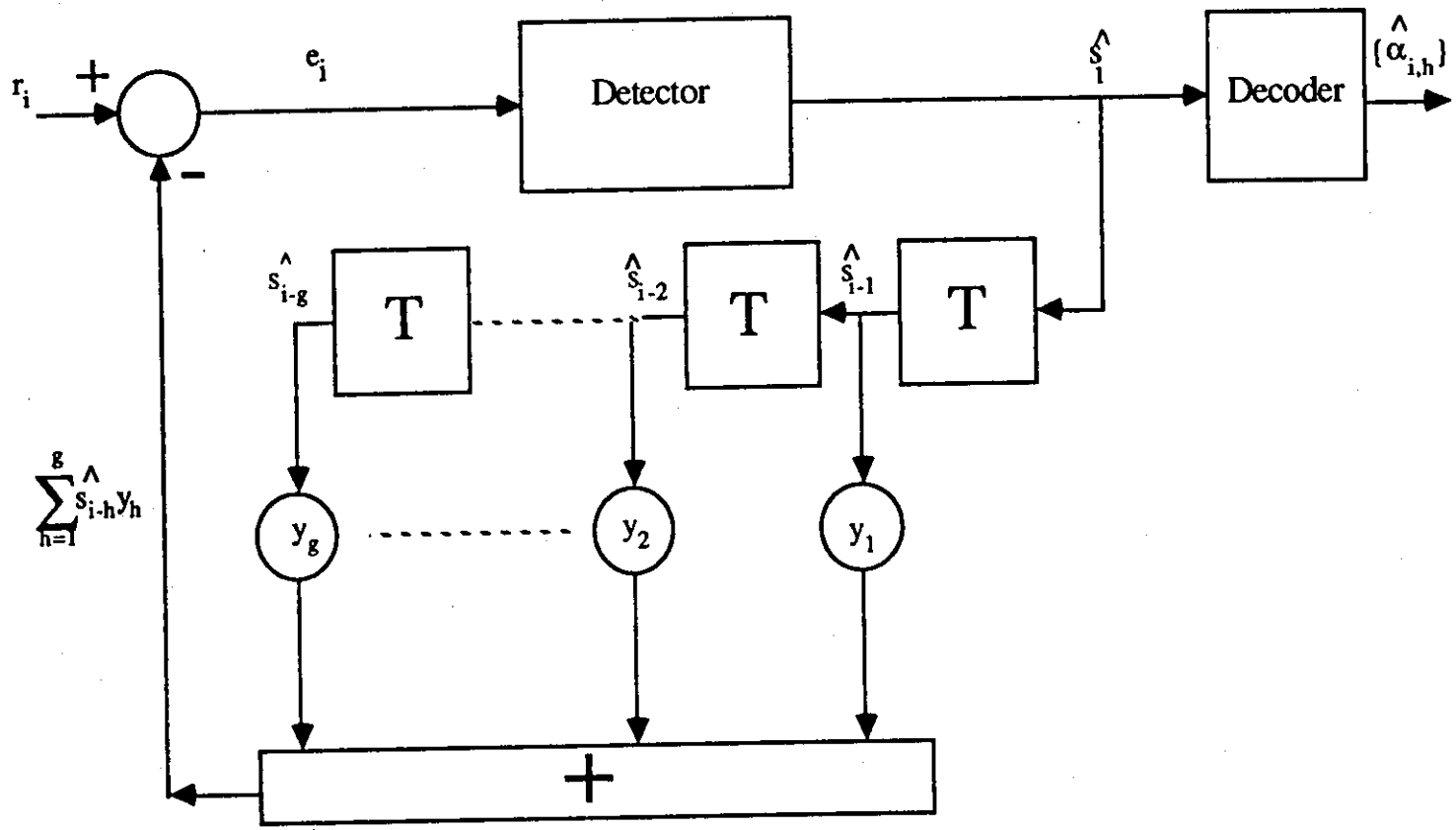


Fig.4.2 The nonlinear (decision feedback) equalizer and detector .

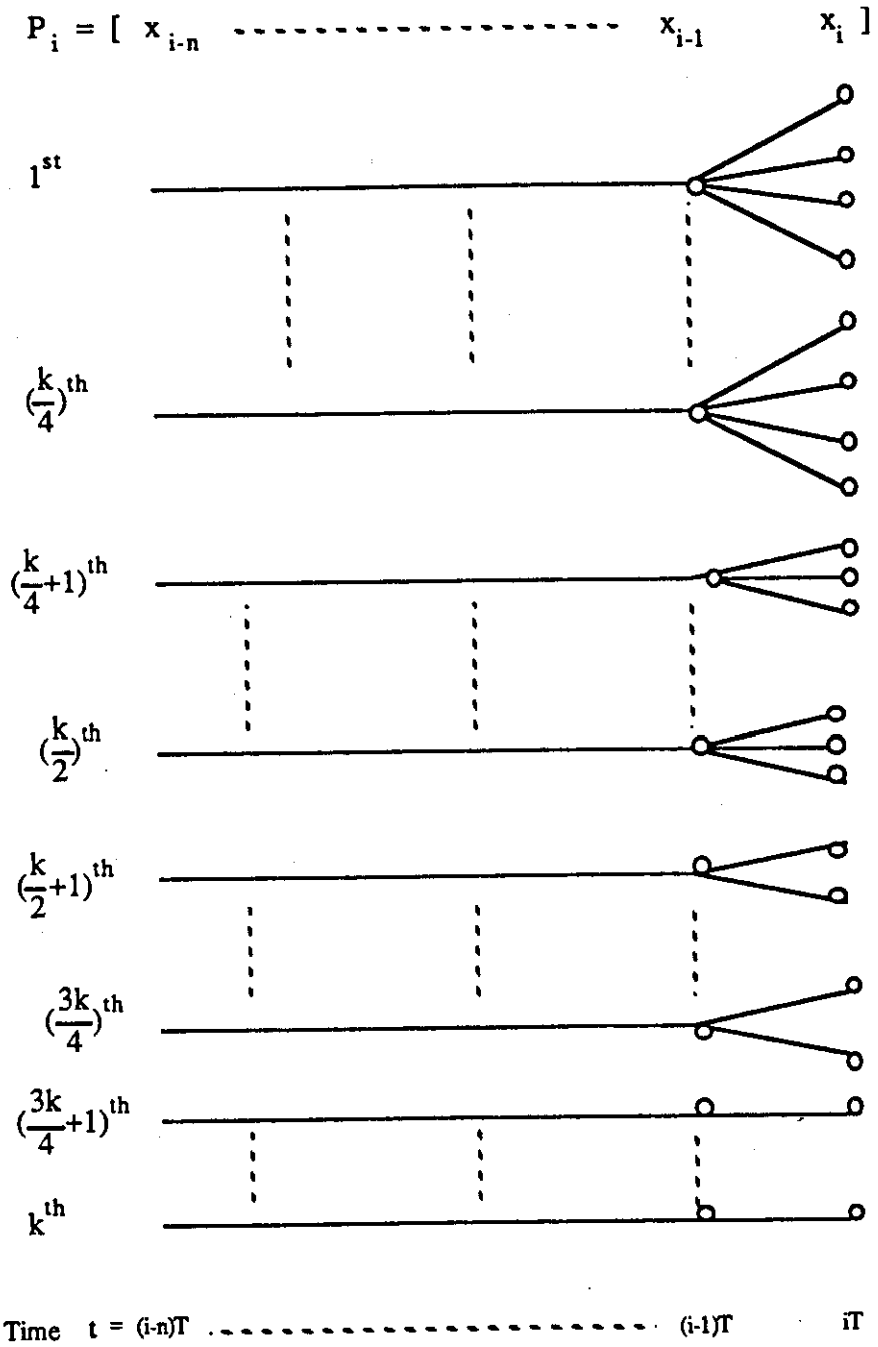


Fig.4.3 Expansion of the vectors { P_i } in system 2U at time t=iT .

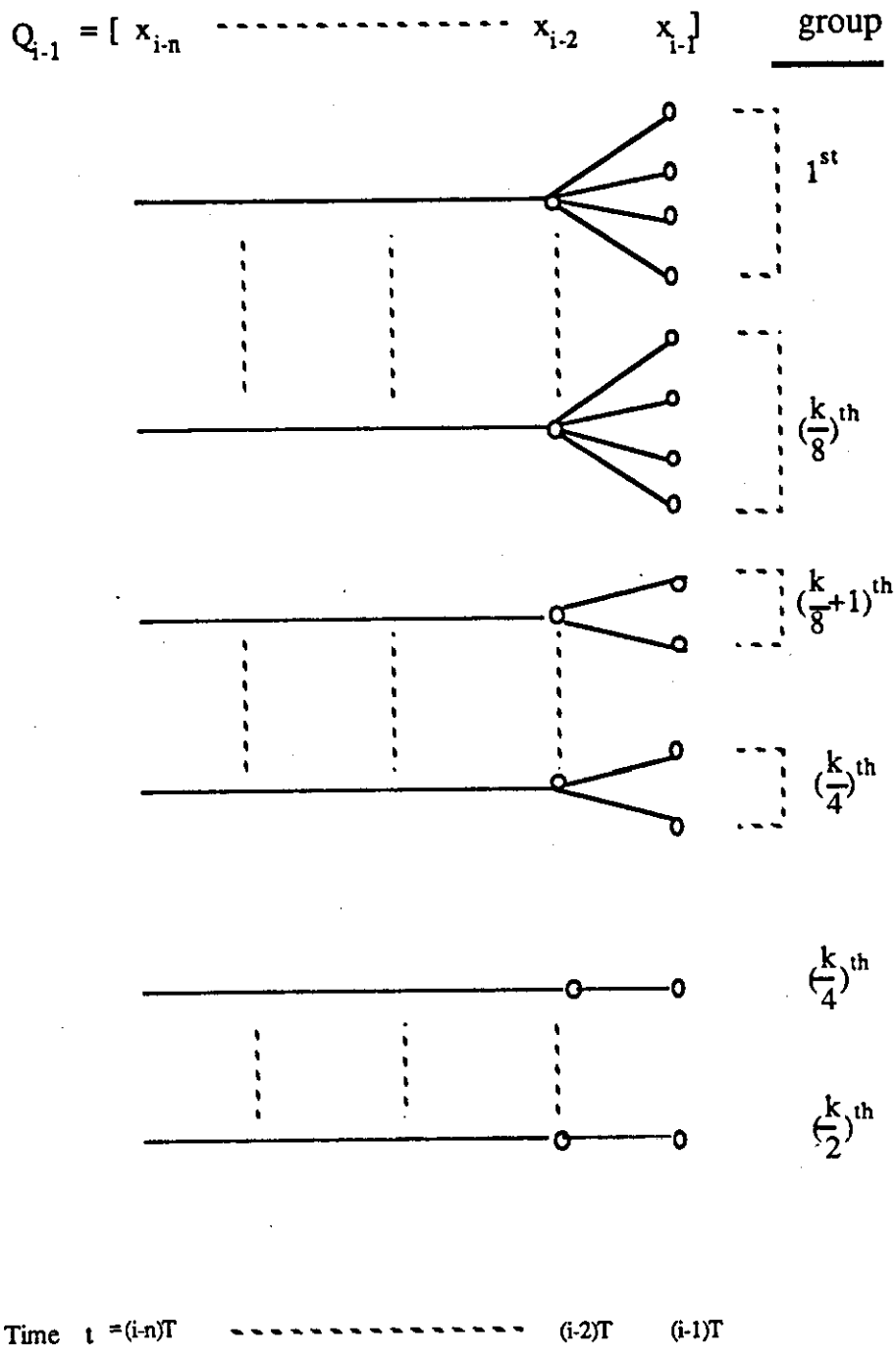


Fig.4.4 Arrangement of the vectors $\{Q_{i-1}\}$ in system 3U at time t=(i-1)T .

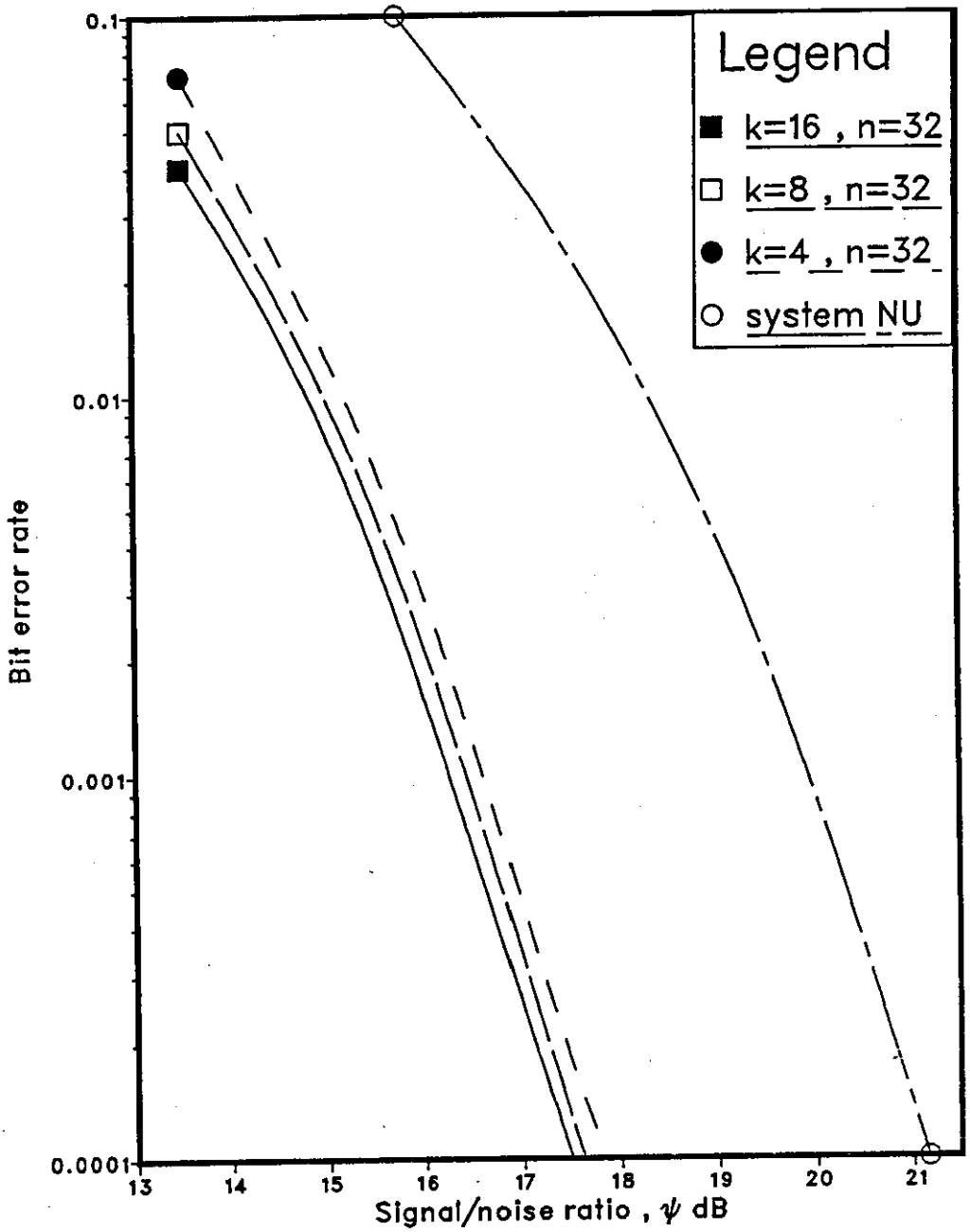


Fig.4.5 Performance of system 1U and the nonlinear equalizer over channel E

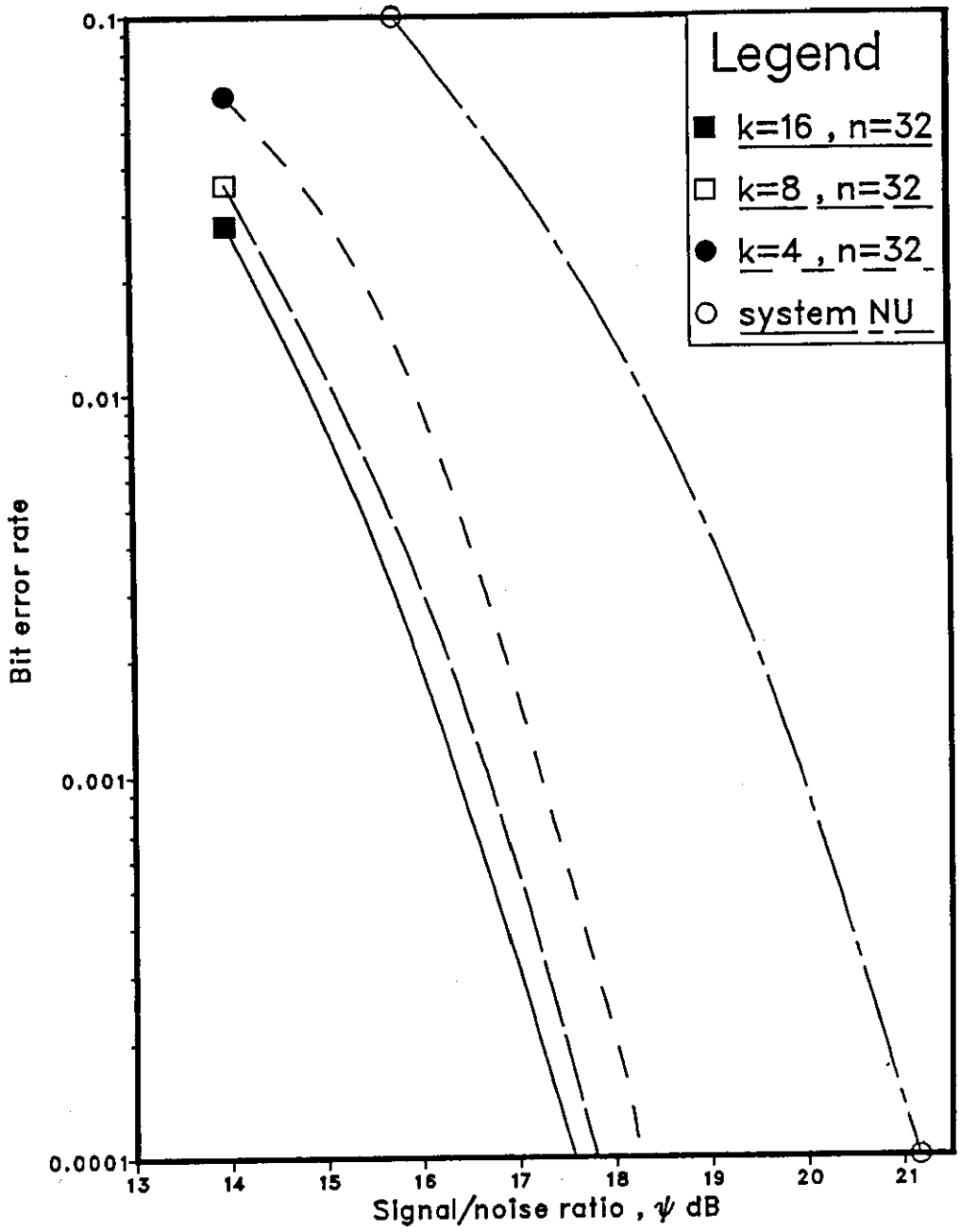


Fig.4.6 Performance of system 2U and the nonlinear equalizer over channel E

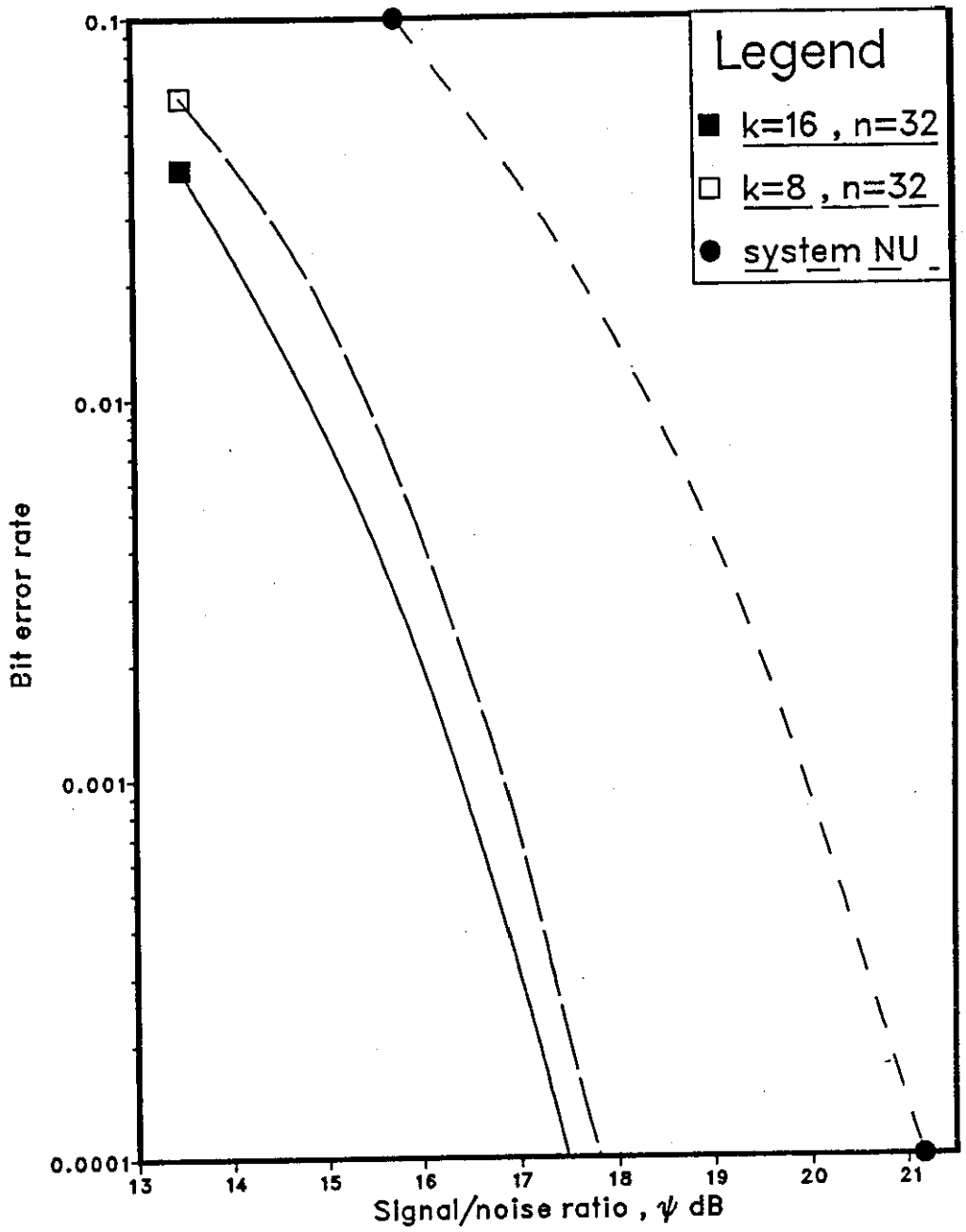


Fig.4.7 Performance of system 3U and the nonlinear equalizer over channel E

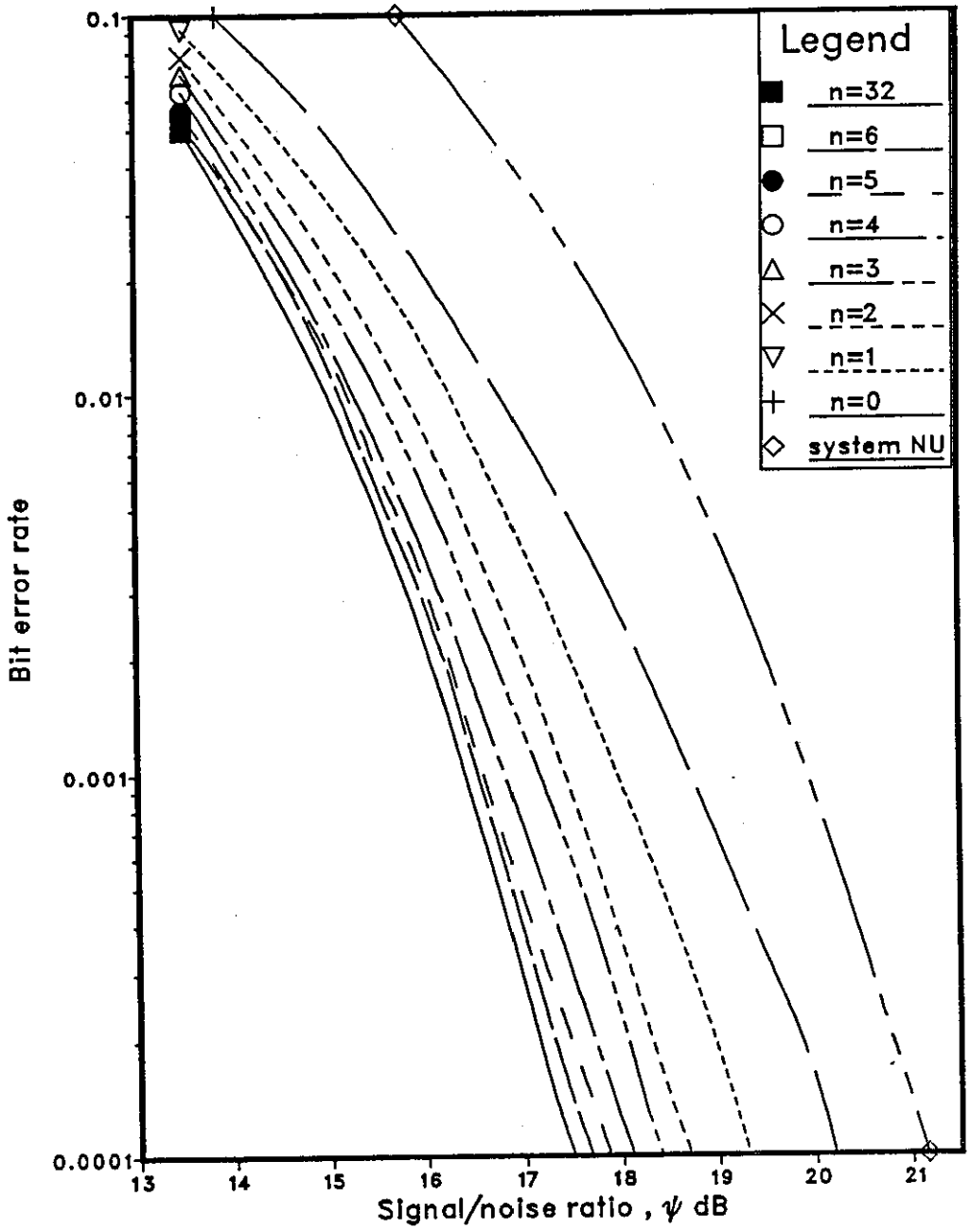


Fig.4.8 Performance of system 1U8 and the nonlinear equalizer over channel E

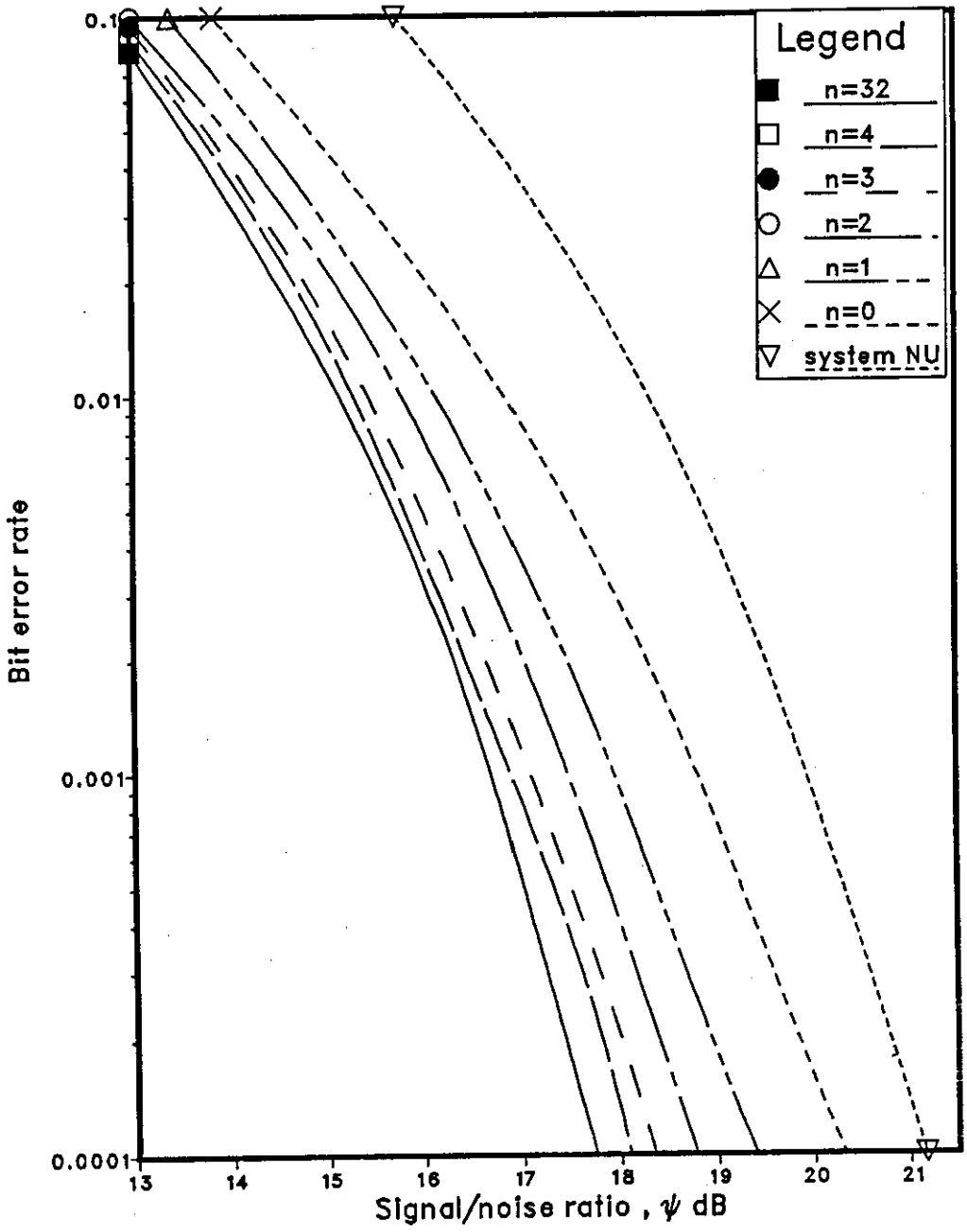


Fig.4.9 Performance of system 2U8 and the nonlinear equalizer over channel E

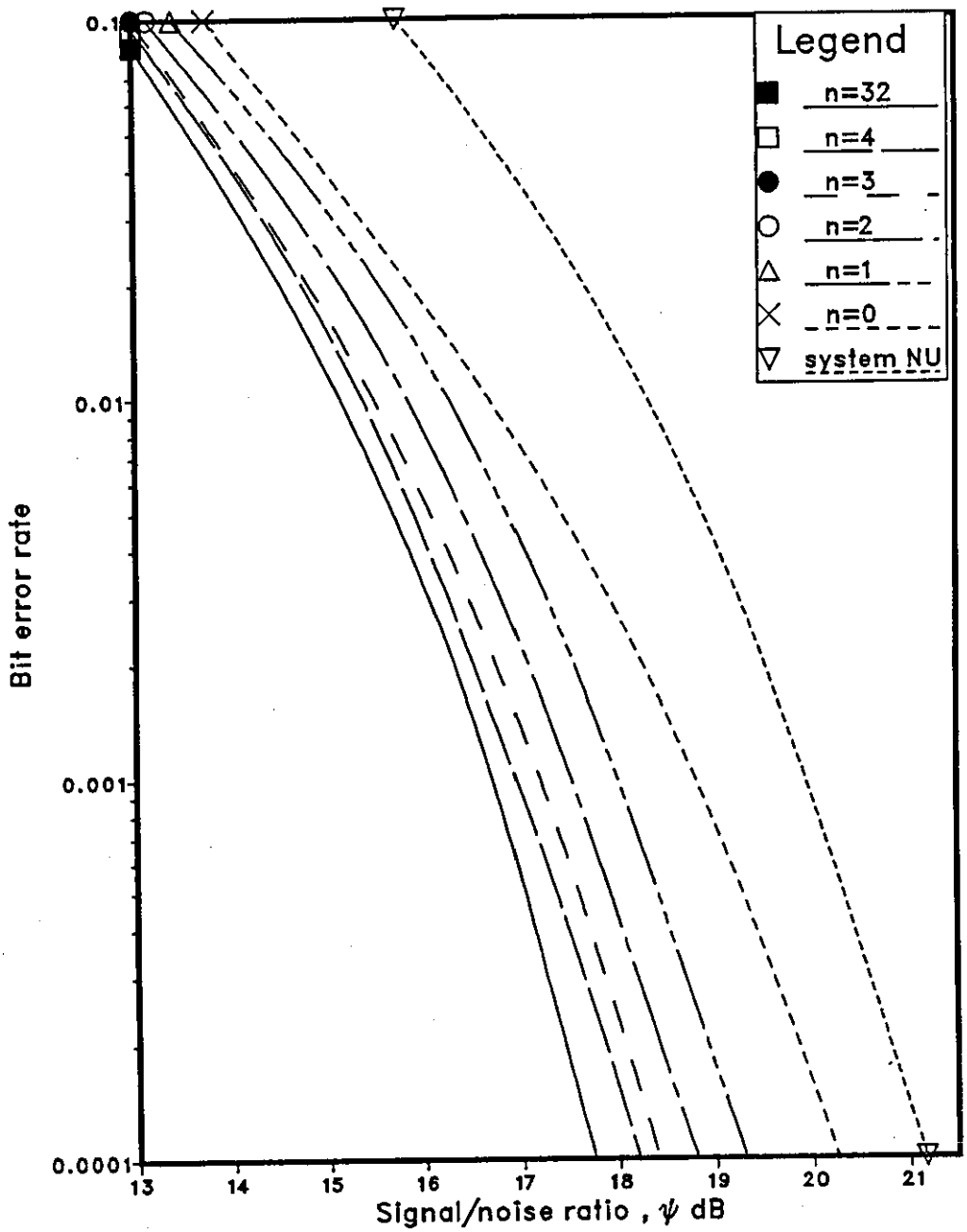


Fig.4.10 Performance of system 3U8 and the nonlinear equalizer over channel E

CHAPTER 5

DETECTION PROCESSES FOR CONVOLUTIONALLY ENCODED 32-LEVEL QAM SIGNAL

5.1 INTRODUCTION

Various detection processes that are suitable for detecting (decoding) a convolutionally encoded and distorted QAM signal are described in this chapter. The Viterbi algorithm detector is described first for the case where the channel introduces no intersymbol interference and for the case where the channel introduces limited distortion. The use of the equalizers (linear or nonlinear) to equalize the channel ahead of the detector is also described. Some of the near-maximum likelihood detectors, described in Chapter 4, are modified here to suit the encoded signal and a new detector, which is a direct modification of the Viterbi algorithm detector is described. The complexity of the systems and the effects of the delay in detection and the number of the stored vectors held by each detector are next investigated. Finally the results of the computer simulation tests are presented for all systems (detectors) studied here together with the uncoded systems described in the previous chapter.

The coding scheme used here is described in Section 3.2, and all assumptions made earlier in Chapter 3 are also valid here. The sampled impulse response of the linear baseband channel and the adaptive linear filter is given by the (g+1)-component row vector

$$Y = [\, y_0 \quad y_1 \quad y_2 \quad . \quad . \quad . \quad y_g \,] \qquad \ldots \quad 5.1.1$$

where $y_0 = 1$ and the received sample at the output of the filter, at time t=iT, is given by

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_h + w_i = s_i + \sum_{h=1}^{g} s_{i-h}\, y_h + w_i \qquad \ldots \quad 5.1.2$$

where $\{s_{i-k}\}$ are the encoded data symbols with possible values as shown in Fig. 5.1, and the real and imaginary components of the noise samples $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and fixed variance $2\sigma^2$ (Chapter 3 and Appendix E).

5.2 THE VITERBI ALGORITHM DETECTOR (DECODER)

The Viterbi algorithm detector (decoder) to be described operates on a 32-level QAM signal that has been convolutionally encoded according to Section 3.2 (Chapter 3), and has been transmitted over channel A, which introduces no intersymbol interference, so that the vector Y (Eqn. 5.1.1) is given by

$$Y=[1 \quad 0 \quad 0 \quad . \quad . \quad . \quad 0] \qquad \ldots \quad 5.2.1$$

and the received sample at time t=iT is now given by

$$r_i = s_i + w_i \qquad \ldots \quad 5.2.2$$

Just prior to the receipt of r_{i+1} , at time t=(i+1)T, the detector holds in store 8 different n-component vectors $\{Q_i\}$, where

$$Q_i=[x_{i-n+1} \quad x_{i-n} \quad . \quad . \quad . \quad x_i] \qquad \ldots \quad 5.2.3$$

and x_{i-k} takes on any of a subset of 16 of the 32 possible values of s_{i-k} in Fig. 5.1. The two subsets are complementary and each contains 16 possible values. They are known as subsets B0 and B1 in Fig. 2.10 in Chapter 2, and are shown again in Fig. 5.2.

When i<0, $x_i=0$. Each vector Q_i is associated with a different one of the eight states of the encoder at time t=iT in Table 5.1, in the sense that $\beta_{i,0}, \beta_{i,1}$ and $\beta_{i,2}$, which form the first three of the five binary digits determining x_i in Fig. 5.1, corresponds to a transition from a different state at time t=iT. Associated with each transition are the four possible combinations of $\beta_{i,3}$ and $\beta_{i,4}$ (Table 5.1). The state of a vector is identified by its storage location. The n-component vector Q_i forms the last n components of the (i+1)-component vector

$$X_i=[x_0 \quad x_1 \quad x_2 \quad . \quad . \quad . \quad x_i] \qquad \ldots \quad 5.2.4$$

which represents a possible sequence of the encoded data symbols $\{s_h\}$. It is evident from the description of the encoder that, there is a unique one-to-one relationship between X_i and the corresponding possible sequence of the binary digits $\{\alpha_{i,h}\}$, so that the correct detection of $\{s_i\}$ results in the correct detection of the $\{\alpha_{i,h}\}$.

Associated with each vector Q_i is stored its cost, which is taken to be the cost of the corresponding vector X_i , given by

$$c_i = \sum_{h=0}^{i} | r_h - x_h |^2 \qquad \ldots \quad 5.2.5$$

where $| u |$ is the absolute value of u. The cost c_i is simply the square of the unitary distance between the two sequences $\{r_h\}$ and $\{x_h\}$.

On the receipt of the received sample r_{i+1} , each of the eight vectors $\{Q_i\}$ is expanded to give 16 vectors $\{P_{i+1}\}$, where

$$P_{i+1} = [\, x_{i-n+1} \quad x_{i-n} \quad . \quad . \quad . \quad x_i \quad x_{i+1} \,] \qquad \ldots \quad 5.2.6$$

The first n components of P_{i+1} are given by the original vector Q_i , and the last component x_{i+1} takes on its 16 different permitted values. The latter are determined by the four different permitted combinations of $\beta_{i+1,0}, \beta_{i+1,1}$ and $\beta_{i+1,2}$, corresponding to the given state at time (i+1)T, each associated with the four different combinations of $\beta_{i,3}$ and $\beta_{i,4}$ (Fig. 5.2). The cost of the resulting 128 vectors $\{P_{i+1}\}$ is given by

$$c_{i+1} = c_i + | r_{i+1} - x_{i+1} |^2 \qquad \ldots \quad 5.2.7$$

where c_i is the cost of the original vector Q_i .

The detector now has 16 different vectors $\{P_{i+1}\}$, for each of the eight states at time (i+1)T in Table 5.1 and it selects, for each state at time (i+1)T the one of the 16 associated vectors $\{P_{i+1}\}$ with the smallest cost c_{i+1} , the remaining vectors being discarded. The detected value $\hat{s}_{i-n+1}$ of the data symbol s_{i-n+1} is next given by the value of x_{i-n+1} in the selected vector P_{i+1} with the smallest cost. The detected symbol $\hat{s}_{i-n+1}$, in turn determines the detected values $\hat{\alpha}_{i-n+1,1}, \hat{\alpha}_{i-n+1,2}, \hat{\alpha}_{i-n+1,3}$ and $\hat{\alpha}_{i-n+1,4}$ of the binary digits $\alpha_{i-n+1,1}, \alpha_{i-n+1,2}, \alpha_{i-n+1,3}$ and $\alpha_{i-n+1,4}$, respectively. In the simulation, this is done by storing for each vector the corresponding sequence of $\{\hat{\alpha}_{i,h}\}$.

The first component x_{i-n+1} of each of the eight vectors $\{P_{i+1}\}$ is then omitted to give the corresponding vector Q_{i+1} , without changing its cost. Finally the resulting eight

vectors $\{Q_{i+1}\}$ are stored together with their costs. To prevent overflow due to the steady increase in costs over any one transmission, the smallest cost is subtracted from the cost of each vector Q_{i+1}, thus always reducing the smallest cost to zero.

The detection process just described is referred to asaVA detector.

When the convolutional encoder is used together with a distorting channel, in which the sampled impulse response Y has (g+1) components as in Eqn. 5.1.1, the true Viterbi algorithm detector in this case operates with 8×16^{g} states and therefore with 8×16^{g} stored vectors. This number results from the fact that the encoder has 8 states (Table 5.1) and the sampled impulse response Y has (g+1) components, while the value (16) represents the 16 possible values in each subset in Fig. 5.2.

The operation of the Viterbi algorithm detector is described below for the case where the data is transmitted over channel B (Chapter 3). This is a simple case where g=1 and $y_0=y_1=1$, so that $8\times16^{g} = 128$, and $y_i=0$ for i<0 and i>1. Now

$$Y=[1 \quad 1 \quad 0 \quad 0 \quad . \quad . \quad . \quad 0] \qquad \ldots \quad 5.2.8$$

and the received sample at the input of the detector, at time t=iT, is

$$r_i = s_i + s_{i-1} + w_i \qquad \ldots \quad 5.2.9$$

Just prior to the receipt of r_{i+1}, the detector holds in store 128 vectors $\{Q_i\}$ (Eqn. 5.2.3) together with their costs $\{c_i\}$, where

$$c_i = \sum_{h=0}^{i} | r_h - x_h - x_{h-1} |^2 \qquad \ldots \quad 5.2.10$$

Since each vector Q_i is associated with a different one of the 128 states at time iT, the vectors $\{Q_i\}$ occur in eight groups, each of which contains 16 vectors and is associated with a different one of the original eight states of the encoder (Table 5.1). Furthermore, the 16 vectors in a group have the 16 different permitted values of x_i that are associated with the given state of the encoder [14].

On the receipt of r_{i+1}, each vector Q_i is expanded into 16 vectors $\{P_{i+1}\}$, as described previously in the VA detector. The cost of each of the resulting 2048 vectors is evaluated as

$$c_{i+1} = c_i + | r_{i+1} - x_{i+1} - x_i |^2 \qquad \ldots \quad 5.2.11$$

where c_i is the cost of the original vector Q_i (Eqn. 5.2.10). The vectors $\{P_{i+1}\}$ are in 128 groups of 16 each, every group corresponding to a different one of the 128 states at time iT and being determined by the combination of the state at time iT and the value of x_{i+1} . The vectors $\{P_{i+1}\}$ also occur in 128 groups of 16 vectors each, every group now corresponding to a different one of the 128 states at time (i+1)T, and being determined by the combination of the state at time (i+1)T and the value of x_i [14].

The detector next selects, for each of the 128 states at time (i+1)T, the one of the 16 associated vectors $\{P_{i+1}\}$ with the smallest cost, the remaining vectors being discarded. The detected values of the binary digits $\alpha_{i-n+1,1}, \alpha_{i-n+1,2}, \alpha_{i-n+1,3}$ and $\alpha_{i-n+1,4}$ are then determined, as in VA detector, from the selected vector P_{i+1} with the smallest cost. The first component of each vector P_{i+1} is then omitted to give the corresponding vector Q_{i+1}, without changing its cost as before. The resulting 128 vectors together with their cost are then stored after the smallest cost has been subtracted from all costs.

It is clear from the above description that the detector now requires a large number of operations and a large store. Therefore, when g (the number of intersymbol interference components introduced by the channel) is greater than 1, the true Viterbi algorithm detector becomes impractical to implement.

5.3 LINEAR EQUALIZER WITH VITERBI ALGORITHM DETECTOR (SYSTEM LC)

The structure of this system is shown in Fig. 5.3. The linear feedback equalizer here is identical to that used in the uncoded system (Chapter 4), and it can be used without risk of instability, because the sequence $y_0, y_1, \ldots, y_g$ has been made minimum phase by the adaptive linear filter that proceeds the detector (see Fig. 3.1). The detector in Fig. 5.3 operates in the same way as does the Viterbi algorithm detector where the channel introduces no intersymbol interference. The input signal to the detector is the sequence of the equalized samples $\{e_i\}$, where e_i , at time iT, is given by

$$e_i = r_i - \sum_{h=1}^{g} e_{i-h}\, y_h = x_i + w_i' \qquad \ldots \quad 5.3.1$$

where r_i is given by Eqn. 5.1.2, x_i is the possible value of s_i, w_i' is the noise component in e_i and $y_0 = 1$.

As described in the beginning of the previous section, just prior to the receipt of r_{i+1}, the VA detector holds in store eight different vectors $\{Q_i\}$ (Eqn. 5.2.3) together with their costs $\{c_i\}$, where the value of the cost in this system is given by

$$c_i = \sum_{h=0}^{i} | e_h - x_h |^2 \qquad \ldots \quad 5.3.2$$

On the receipt of r_{i+1}, at time (i+1)T, each vector Q_i is expanded into 16 vectors $\{P_{i+1}\}$ as described in the previous section. The cost c_{i+1} of each vector P_{i+1} is evaluated as

$$c_{i+1} = c_i + | e_{i+1} - x_{i+1} |^2 \qquad \ldots \quad 5.3.3$$

where e_{i+1} is the equalized sample at time (i+1)T and c_i is the cost of the original vector Q_i.

The process then continues as in the VA detector (Section 5.2).

5.4 NONLINEAR EQUALIZER WITH VITERBI ALGORITHM DETECTOR

5.4.1 SIMPLE NONLINEAR EQUALIZER WITH VITERBI ALGORITHM DETECTOR (SYSTEM NC)

The arrangement of this system is shown in Fig. 5.4. The feedback filter here is identical to that used for the uncoded signal (system NU) in Chapter 4, where the tap gains of the filter are given by the last g components of the vector Y (Eqn. 5.1.1). The threshold level detector in Fig. 5.4 is suitable for the 32-point constellation of Fig. 5.1. The Viterbi algorithm detector (VA) operates on the sequence of the equalized samples $\{e_i\}$ exactly in the same manner as in system LC (Section 5.3). Here the equalized sample at time iT, is given by

$$e_i = r_i - \sum_{h=1}^{g} \hat{s}_{i-h}\, y_h \qquad \ldots \quad 5.4.1$$

where r_i is given by Eqn. 5.1.2 and $\{\hat{s}_{i-k}\}$ are the detected values of $\{s_{i-k}\}$ provided by the threshold level detector (Fig. 5.4), with correct detection, $e_i = x_i + w_i$, bearing in mind that $y_0 = 1$. The threshold level detector simply selects one of the 32 possible values of s_i , which is closest (in the unitary distance sense) to the equalized sample e_i .

5.4.2 MODIFIED NONLINEAR EQUALIZER WITH VITERBI ALGORITHM DETECTOR (SYSTEM MNC)

The detection process in system MNC is similar to that in system NC, described in the previous section, with only one exception that the values of $\{\hat{s}_{i-k}\}$ are provided here by the Viterbi algorithm detector (VA) itself and not by the threshold level detector. So that the latter is no longer required here. The arrangement of this system is shown in Fig. 5.5

After the receipt of r_{i+1} , at time (i+1)T, the VA detector operates as in system LC or NC, and the value of x_{i-n+1} in the vector P_{i+1} with the smallest cost is then taken as the detected value of the data symbol s_{i-n+1} . The latter is then used to determine the detected values of the transmitted information digits as before. Now, the value of the last component (x_{i+1}) of the vector P_{i+1} with the smallest cost, is used as an early detected value of the data symbol s_{i+1} for the equalizer to cancel its intersymbol interference terms in the next received samples. The process then continues in this way.

The simulation program of this system is given in Appendix G2.

5.5 NEAR-MAXIMUM LIKELIHOOD DETECTORS FOR CONVOLUTIONALLY ENCODED SIGNALS

5.5.1 SYSTEM 1C

System 1C is a near-maximum likelihood detector (decoder) that is a direct modification of system 1U, which is used for the detection of uncoded QAM signal in Chapter 4.

Just prior to the receipt of r_{i+1} , at time (i+1)T, the detector in system 1C holds in store k vectors $\{Q_i\}$ together with their costs $\{c_i\}$, where Q_i is given by Eqn. 5.2.2 and c_i is given by

$$c_i = \sum_{h=0}^{i} \left| r_h - \sum_{j=0}^{g} x_{h-j} y_j \right|^2 \quad \ldots \quad 5.5.1$$

and the value of r_h is given by Eqn. 5.1.2. k may be any suitable integer.
On the receipt of r_{i+1} , at time (i+1)T, each vector Q_i is used to form four vectors $\{P_{i+1}\}$. The first n components of each vector P_{i+1} (Eqn. 5.2.6) are given by the n components of the original vector Q_i , and the last component x_{i+1} is determined as follows ; for each vector Q_i , the detector first calculates the quantity z_{i+1} , where

$$z_{i+1} = r_{i+1} - \sum_{h=1}^{g} x_{i-h+1} y_h \quad \ldots \quad 5.5.2$$

and the $\{x_{i-h+1}\}$ are given by the particular vector Q_i . The detector then determines the four possible values of x_{i+1} , that are closest to z_{i+1}. The particular values of x_{i+1} are given by one or other of the two subsets in Fig. 5.2, the subset being determined by the state of the original vector Q_i at time iT. So, in fact, the state of each vector is also stored. The process just described ensures, that only the permitted (valid) transitions are considered by the detector. The cost of each vector P_{i+1} is given by

$$c_{i+1} = c_i + \left| r_{i+1} - \sum_{h=0}^{g} x_{i-h+1} y_h \right|^2 \quad \ldots \quad 5.5.3$$

where c_i is the cost of the original vector Q_i . Clearly, the four values of x_{i+1} are selected such that only the four vectors $\{P_{i+1}\}$ with the smallest costs $\{c_{i+1}\}$, that have originated from any vector Q_i , are used by the detector. The detector then selects from the resulting 4k vectors $\{P_{i+1}\}$, the one with the smallest cost and it takes the first component x_{i-n+1} of this vector to be the detected value $\hat{s}_{i-n+1}$ of the data symbol s_{i-n+1} . The detected values of the binary digits $\alpha_{i-n+1,1}, \alpha_{i-n+1,2}, \alpha_{i-n+1,3}$ and $\alpha_{i-n+1,4}$ are then determined as before. Any vector P_{i+1} for which $x_{i-n+1} \neq \hat{s}_{i-n+1}$ is now discarded from any future selection. The detector then selects from the remaining vectors $\{P_{i+1}\}$, including the one with the smallest cost, the k vectors $\{P_{i+1}\}$ with the lowest costs $\{c_{i+1}\}$. The first component x_{i-n+1} of each vector P_{i+1} is then omitted to give the corresponding vector Q_{i+1} . As in the previous systems, the smallest cost of the vectors is then subtracted from all costs to bring the smallest cost

to zero. Finally the k vectors $\{Q_{i+1}\}$ together with their states and the values of their costs $\{c_{i+1}\}$ are stored for the next detection.
The simulation program of this system is shown in Appendix G3.

5.5.2 SYSTEM 2C

The detector in system 2C operates in the same way as system 1C, with only one exception, which is that the number of vectors $\{P_{i+1}\}$ derived from any vector Q_i become a function of the relative cost of that Q_i . This detector is also a modification of system 2U used for the detection of the uncoded QAM signal in Chapter 4.
The detector here holds in store k vectors $\{Q_i\}$, together with their costs $\{c_i\}$, where k is an integral multiple of 4. These vectors are arranged according to the values of their costs, where the first vector in store has the smallest cost and the second vector has the second smallest cost and so on.

On the receipt of r_{i+1} , at time (i+1)T, each vector Q_i is expanded into a number of corresponding vectors $\{P_{i+1}\}$ as follows ; The first k/4 vectors $\{Q_i\}$ are each expanded into four vectors $\{P_{i+1}\}$. The second k/4 vectors are each expanded into three vectors $\{P_{i+1}\}$. The third k/4 vectors are each expanded into two vectors $\{P_{i+1}\}$ and the last k/4 vectors $\{Q_i\}$ in the store are each expanded into one vector P_{i+1} only. Clearly, the method of expanding the vectors here is similar to that used in system 2U for the case where the signal is uncoded as in Fig. 4.3 (Chapter 4). In every case the vectors $\{P_{i+1}\}$ derived from any particular vector Q_i are those with the smallest cost. According to the state of the vector Q_i at time iT, the value of x_{i+1} (the last component of the vector P_{i+1}) may take any one of its 16 possible (permitted) values defined by the corresponding one of the two subsets in Fig. 5.2. The detector then evaluates the costs $\{c_{i+1}\}$ of the vectors $\{P_{i+1}\}$ according to Eqn. 5.5.3. There are now 5k/2 vectors $\{P_{i+1}\}$. The detector then selects from these vectors the one with the smallest cost c_{i+1} , and it takes the value of its first component x_{i-n+1} as the detected value $\hat{s}_{i-n+1}$ of the data symbol s_{i-n+1} . This value in turn, gives the detected values of the binary digits $\alpha_{i-n+1,1}, \alpha_{i-n+1,2}, \alpha_{i-n+1,3}$ and $\alpha_{i-n+1,4}$ as before. As in the previous detector (system 1C), any vector P_{i+1} for which $x_{i-n+1} \neq \hat{s}_{i-n+1}$ is discarded from any future selection. The detector then selects from the remaining vectors $\{P_{i+1}\}$, including that with the lowest cost, the k vectors with lowest costs and the smallest

cost is then subtracted from the cost of each vector. The first components of the vectors $\{P_{i-1}\}$ are now omitted to give the corresponding vectors $\{Q_{i+1}\}$. The k vectors $\{Q_{i+1}\}$ together with their costs are then stored after being arranged according to their costs ready for the next detection. The simulation program of this detector is shown in Appendix G4.

5.5.3 SYSTEM 3C

System 3C is a near-maximum likelihood detector that is a direct modification of the Viterbi algorithm detector for a convolutionally encoded and distorted signal, involving a considerable reduction in the number of stored vectors [14].

Just prior to the receipt of r_{i+1} , at time (i+1)T, the detector holds in store k vectors $\{Q_i\}$, together with their costs $\{c_i\}$, where k here is an integral multiple of 8, the latter represents the number of states of the encoder. Each vector Q_i is given by Eqn. 5.2.3 and its cost is given by Eqn. 5.5.1. The k vectors $\{Q_i\}$ are arranged in eight groups, each of which has k/8 vectors and corresponds to a different state of the encoder at time iT (Table 5.1). On the receipt of r_{i+1} each vector Q_i is expanded into 16 vectors $\{P_{i+1}\}$. The first n components of each vector P_{i+1} (Eqn. 5.2.6) are given by the n components of the original vector Q_i , and the last component x_{i+1} takes on the 16-possible values defined by one of the two subsets in Fig. 5.2. The selection of the particular subset is determined by the state of the vector Q_i at time iT. The resulting 16k vectors $\{P_{i+1}\}$ occur in eight groups, each with 2k vectors and associated with a different state of the encoder at time (i+1)T. The cost c_{i+1} of the vector P_{i+1} is evaluated according to Eqn. 5.5.3. The detector then selects from the 16k vectors $\{P_{i+1}\}$ the vector P_{i+1} with the smallest cost c_{i+1} , and uses the value of its first component x_{i-n+1} to determine the detected values of the binary digits $\alpha_{i-n+1,1}, \alpha_{i-n+1,2}, \alpha_{i-n+1,3}$ and $\alpha_{i-n+1,4}$ as before. To avoid possible merging, as in the previous detectors, any vector P_{i+1} for which $x_{i-n+1} \neq s_{i-n+1}$ is now discarded from any future selection. From the remaining vectors $\{P_{i+1}\}$ (including that with the smallest cost) the detector next selects, for each of the eight states of the encoder at time (i+1)T, the k/8 vectors $\{P_{i+1}\}$ with the smallest costs. The first component of each of the selected vectors is now omitted to give the corresponding vector Q_{i+1} , without changing its cost. The smallest cost of all vectors is then subtracted from the value of

each cost, and the resulting k vectors $\{Q_{i+1}\}$ are then stored together with their costs. Unlike system 1C or 2C, the storage of the state for each vector in system 3C is not required. Instead the location of the vector in the store is used as a guide for its state. The simulation program of this system is shown in Appendix G5.

5.6 COMPLEXITY AND THE EFFECTS OF THE PARAMETERS k AND n ON THE PERFORMANCE OF THE DETECTORS

In this section the complexity required by each near-maximum likelihood detector for the encoded signal (described in the previous section) are compared to that required by the Viterbi algorithm detector. The effects of the detection delay and the number of the stored vectors on the performance of these detectors are also examined here.

Table 5.2 shows the complexities of different detectors (systems). System VC is the Viterbi algorithm detector, that is suitable for the convolutionally encoded signal transmitted over a channel whose sampled impulse response has (g+1) components (Eqn. 5.1.1). This detector is tested here with channel B (defined by Eqn. 5.2.8) and its performance is presented in the next section. The systems in Table 5.2 are listed according to their complexity and starting with the most complex detector (the Viterbi algorithm detector). It is clear, that the number of the stored vectors k, held by the near-maximum likelihood detector, is independent of the value of g, whereas in the case of system VC (the Viterbi algorithm detector) this number is equivalent to 8×16^{g} . As in all maximum or near-maximum likelihood detectors, the most complex process of the individual processes is the evaluation of the costs. It is clear (Table 5.2) that the complexities of systems 1C, 2C and 3C are well below that of the true Viterbi algorithm detector. The number of cost evaluations and the number of searches through the costs of system 1C or 2C are smaller than those of system 1C, for a given value of k.

Channel E, which includes telephone circuit 3, (Chapter 3) is used again to show the effect of changing the values of k and n on the performance of the near-maximum likelihood detectors proposed here. Figs. 5.6, 5.7 and 5.8 show the performance of the three systems 1C, 2C and 3C, respectively, when operating with different

numbers of stored vectors $\{Q_i\}$. System 1C and 2C are tested here with k=4,8 and 16, whereas system 3C is tested with k=8,16,32 and 64. The delay in the detection is 32 symbol intervals (n=32), and the definition of the signal/noise ratio ψ is given by Eqn. F.6 (Appendix F), where for the encoded 32-level QAM signal λ =20 and m=4. An advantage of about 0.5 dB can be gained, at bit error rate of 10^{-4} , in tolerance of the system to additive white Gaussian noise when the value of k is increased by a factor of two for system 1C, and for the particular channel tested here, (Fig. 5.6). The corresponding advantage for system 2C is about 0.3 dB (Fig. 5.7). An increase of the value of k from 8 to 16 for system 3C gives an advantage of about 2.6 dB at bit error rate of 10^{-4} , while an increase of k from 16 to 32 (or from 32 to 64) gives an advantage of about 0.5 dB at the same bit error rate (Fig. 5.8).

Figs. 5.9, 5.10 and 5.11 show the performances of systems 1C, 2C and 3C, respectively, when operating with 16 stored vectors (k=16) and different values of n (the delay in the detection in terms of symbols interval). Channel E is used here. As in the similar tests carried out for the case where the signal is uncoded in Chapter 4, the number of the components of the vectors $\{Q_i\}$ is fixed at 32 components and the detection is performed with different values of n. where n varies from 0 to 32. Figs. 5.9-5.11 show that, in general, an increase of n from 8 to 32 gain an advantage for a particular system, in tolerance to additive white Gaussian noise, of about 0.25 dB at bit error rate of 10^{-4} . Furthermore, it was found that an increase of n beyond 16 has no effect on the performance of the systems tested here. On the other hand, an increase of n beyond 8 has no effect on the performance of the corresponding uncoded systems, as is shown in Chapter 4.

5.7 ASSESSMENT OF VARIOUS SYSTEMS FOR CODED- AND UNCODED QAM SIGNALS

The results of the computer simulation tests on the different detectors described in this chapter and in the previous chapter (Chapter 4) are shown in Figs. 5.12-5.24. A total of typically 2.5×10^6 data symbols were transmitted in plotting any one curve, and the 95% confidence limits for the relative positions of any one curve is generally better than ± 0.5 dB. Each curve in the above figures represents the variation of the bit error rate against the signal/noise ratio ψ . Here ψ is defined according to Eqn. F.6

in Appendix F, where the value of λ is 10 for the uncoded signal and 20 for the convolutionally encoded signal, and m=4 for both signals. The computer simulation programs for different detection processes are given in Appendix G.

The performance curves of the detectors are labelled by the corresponding systems, and the numeral at the end of the label represents the number of the stored vectors $\{Q_i\}$ used by the particular detector (system). Thus, for example, the curve labelled by 1U16 is system 1U (which is used for the uncoded signal in Chapter 4) operating with 16 stored vectors, and system 2C8 is system 2C (used for convolutionally encoded signal) operating with 8 stored vectors. The letter C or U in each label specify the type of the signal used, C for a convolutionally encoded signal and U for the uncoded signal. In Fig. 5.12, the curve labelled by VA represents the performance of the Viterbi algorithm detector that is suitable for the encoded signal when transmitted over channel A (defined by Eqn. 5.2.1), which introduces no intersymbol interference. This detector is used together with the equalizers (systems LC, NC and MNC) as described in Section 5.3 and 5.4. The curves labelled by VU and VC in Fig. 5.18 represent the performance of the Viterbi algorithm detectors for the uncoded and convolutionally encoded signals when transmitted over channel B (defined by Eqn. 5.2.8), respectively. System VC for this channel is described in Section 5.2 and, system VU is described in Section 2.3.3 in Chapter 2.

It is clear from Fig. 5.12 that, in the absence of any signal distortion (channel A), an improvement in tolerance to noise between 2 and 3 dB is achieved, at bit error rates of 10^{-3} to 10^{-4}, through the use of the convolutional encoder. A simple threshold level detector is used here for the uncoded signal. The above improvement is reduced by about 0.5 dB, when system 1C8 (or 2C8) replaces the Viterbi algorithm detector for the encoded signal (system VA). And the corresponding reduction for system 1C4 (or 2C4) is about 1 dB. Although, systems 1C8 and 2C8 use the same number of vectors (8) as that for system VA, for the given channel (channel A). The amount of operations required by systems 1C and 2C is much less than that required by the Viterbi algorithm detector as shown in Table 5.2, where g=0.

Figs. 5.13-5.17 show the performance of the nonlinear equalizers together with the VA detector for the case where the signal is convolutionally encoded, when operating over channels B-F. These figures show that the modified nonlinear equalizer together with VA detector (system MNC) consistently has a better

performance than system NC. The latter uses a threshold detector to determine the detected values of the data symbols, which are used by the equalizer to perform an estimate of the intersymbol interference in the received samples. It is clear from the above figures that the improvement of system MNC in tolerance to additive white Gaussian noise over system NC is relatively higher over channels B, E and F. This results from the fact that there are relatively longer error bursts in system NC, with these channels, than that in system MNC.

Figs. 5.18-5.24 show the performance of different arrangements of the systems (detectors). The linear equalizer has not been tested with channel B (Fig. 5.18), since the z-transform Y(z) of the vector Y (Eqn. 5.2.8) has a root on the unit circle of the z-plane, which prevents satisfactory operation being achieved with linear equalizer. It is evident from Figs. 5.19, 5.20, 5.22 and 5.24 that a linear equalizer gives a relatively poor performance, compared with other systems, whether or not the signal is convolutionally encoded. The Viterbi algorithm detector (VA) is used with the equalizer when the signal is coded. The additional degradation in the performance that occurs with channel E and F, when the signal is coded, is at least partly due to the noise correlation that is introduced by the linear equalizer into the neighbouring noise components $\{w_i\}$. The Viterbi algorithm detector (VA) used here assumes that the noise samples at its input are uncorrelated. This is investigated further in the next chapter (Chapter 6).

Although the nonlinear equalizer has a much better performance than the linear equalizer, particularly with the poorer channels (channel E and F), it is inferior to all remaining systems tested here. Again with the poorer channels, the use of coding degrades the performance of the equalizer. There is, however, no correlation of the noise components here. The additional degradation in the performance with a coded signal here is due to the fact that the detector with the coded signal (VA detector) is more sensitive to the error extension effects than that used with the uncoded signal which uses a threshold detector.

Over channel B and at bit error rates in the range of 10^{-3} to 10^{-4}, the best performance is given by a Viterbi algorithm detector, operating on a coded signal, (system VC). This gains an advantage in tolerance to noise of between 1.5 and 2 dB over a Viterbi algorithm detector operating on an uncoded signal (system VU), as shown in Fig. 5.18. System 3C64 is only about 0.25 dB inferior to the Viterbi algorithm detector

(system VC), and system 3U16 is only about 0.25 dB inferior to the Viterbi algorithm detector (system VU). The systems 3C64 and 3U16 are, therefore, likely to have performances that are not greatly inferior to those of the corresponding Viterbi algorithm detectors, over the channels C to F.

System 3U16 has consistently the best performance of all systems, tested with uncoded signal, over channels C to F. Although a performance almost as good as that of this system can be achieved by system 1U16 (Chapter 4), the complexity of system 3U16 is less than that of any other near-maximum likelihood detector, studied here with an uncoded signal. It, therefore, follows that the most promising of the systems for an uncoded signal is system 3U16. This achieves a better performance over the channels tested here than that obtained with any equalizer, regardless of whether the equalizer is used with a coded or uncoded signal. Fig. 5.18 show that over channel B, system 3C64 has the best performance of all systems tested here, except the Viterbi algorithm detector (system VC). However, system 3C64 is very complex as shown in Table 5.2. Therefore, a potentially more cost effective arrangement is system 1C16, which has a tolerance to noise generally less than 1 dB inferior to that of system 1C64. Although system 2C is slightly less complex than system 1C, the latter has a better performance than system 2C for the same number of stored vectors and over the channels tested here. At bit error rates not greater than 10^{-3} , system 1C16 has a performance better than that of any near-maximum likelihood detector or equalizer tested with uncoded signal.

There does not appear to be much purpose in using a simple detector with a coded signal, such as systems 1C, 2C and 3C with 8 stored vectors or less. The use of the equalizers with the coded signal also does not appear to be very promising. However, these arrangement are relatively simple to implement. Thus, the preferred arrangement for the coded signal is system 1C16 and for the uncoded signal is system 3U16. These systems, together with the equalizers, are tested further in Chapter 8 where the information rate is more than 9600 bit/s. It is to be further noted that system 1C16 has already been tested by L. A. Alfakhri over an ADPCM link having nonlinear distortion and has been found to perform well.

The expansion of the vectors $\{Q_i\}$ in the systems for the coded signal is restricted to the signal points which result in valid transitions according to the states of the vectors. This make the near-maximum likelihood detectors for the convolutionally encoded signal more complex than those used for the uncoded signal. Furthermore, in the coded systems extra storage is required to store the states of the vectors.

State at time iT			Input at time iT		Output at time iT			State at time (i+1)T		
$u_{i,0}$	$u_{i,1}$	$u_{i,2}$	$\alpha_{i,1}$	$\alpha_{i,2}$	$\beta_{i,0}$	$\beta_{i,1}$	$\beta_{i,2}$	$u_{i+1,0}$	$u_{i+1,1}$	$u_{i+1,2}$
0	0	0	0	0	0	0	0	0	0	0
			1	0	0	1	0	0	0	1
			0	1	0	0	1	0	1	0
			1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0	1	0
			1	0	0	1	0	0	1	1
			0	1	0	0	1	0	0	0
			1	1	0	1	1	0	0	1
0	1	0	0	0	0	1	0	0	1	0
			1	0	0	0	0	0	1	1
			0	1	0	1	1	0	0	0
			1	1	0	0	1	0	0	1
1	1	0	0	0	0	1	0	0	0	0
			1	0	0	0	0	0	0	1
			0	1	0	1	1	0	1	0
			1	1	0	0	1	0	1	1
0	0	1	0	0	1	1	0	1	1	1
			1	0	1	0	0	1	0	0
			0	1	1	1	1	1	0	1
			1	1	1	0	1	1	1	0
1	0	1	0	0	1	1	0	1	0	1
			1	0	1	0	0	1	1	0
			0	1	1	1	1	1	1	1
			1	1	1	0	1	1	0	0
0	1	1	0	0	1	0	0	1	0	1
			1	0	1	1	0	1	1	0
			0	1	1	0	1	1	1	1
			1	1	1	1	1	1	0	0
1	1	1	0	0	1	0	0	1	1	1
			1	0	1	1	0	1	0	0
			0	1	1	0	1	1	0	1
			1	1	1	1	1	1	1	0

Table 5.1 Truth table of the convolutional encoder

Detector	No. of stored vectors $\{Q_i\}$	No. of expanded vectors $\{P_{i+1}\}$	No. of cost evaluations	No. of searches through the costs
Viterbi algorithm (system VC)	8×16^{g}	$8\times16^{g+1}$	$8\times16^{g+1}$	8x(16 searches through 16^{g+1} costs)
System 3C	k	16k	16k	8x(k/8 searches through 2k costs)
System 1C	k	4k	4k	k searches through 4k costs
System 2C	k	2.5k	2.5k	k searches through 2.5k costs

Table 5.2 Complexities of different detectors for coded 32-level QAM signal.

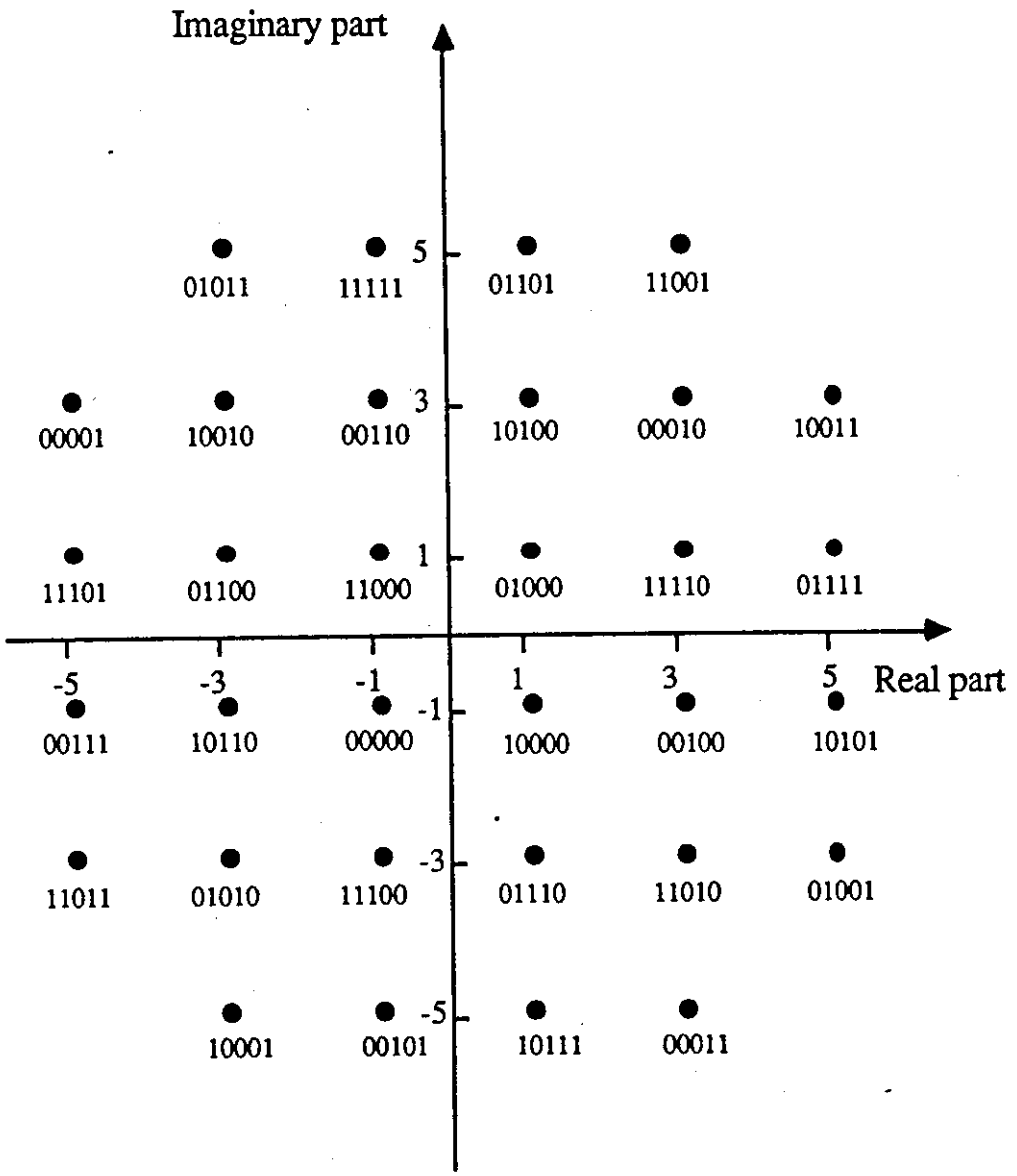


Fig.5.1 Signal constellation for a coded data symbol s_i .

(the binary coded number against each point is $\beta_{i,0}\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}$)

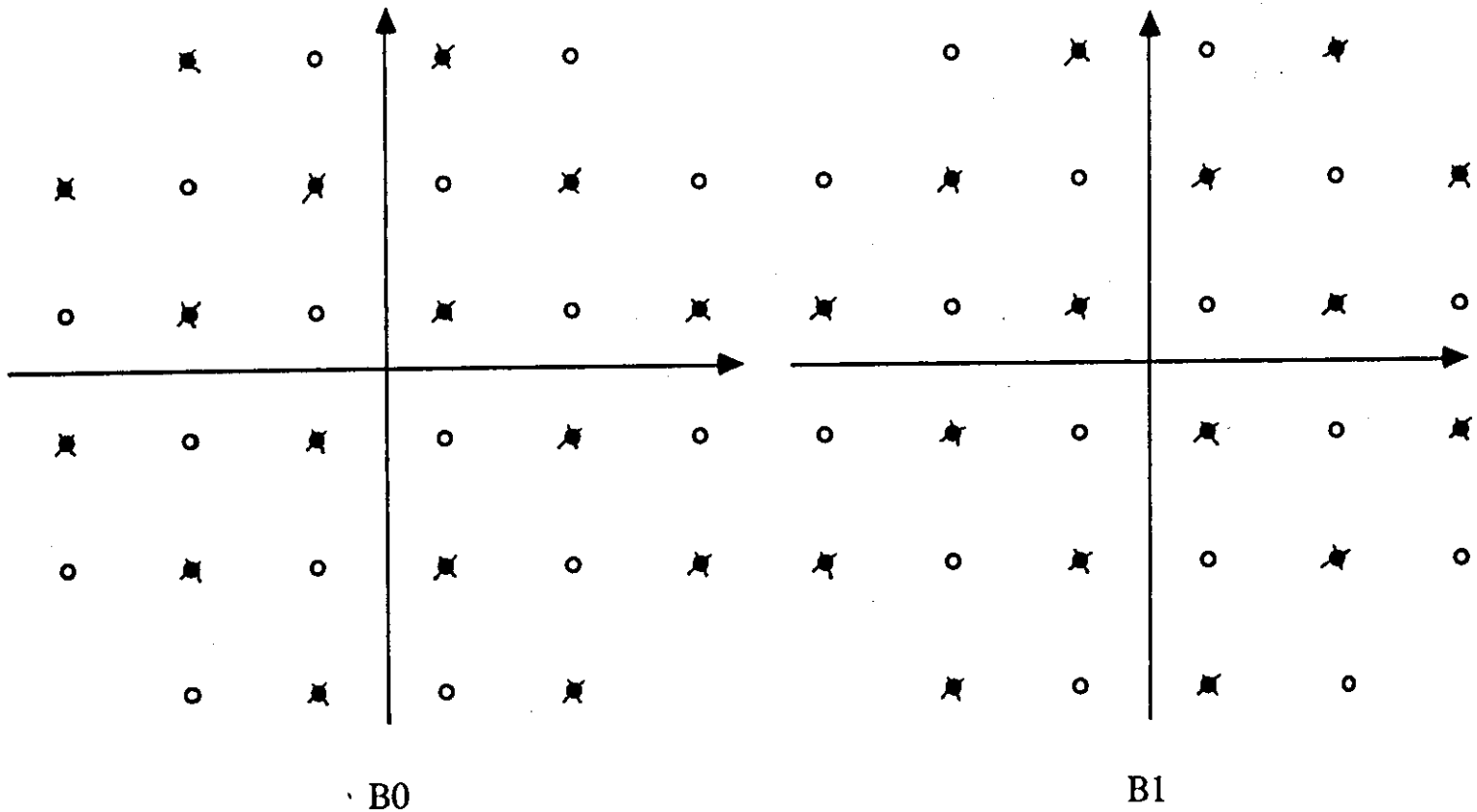


Fig.5.2 The two subsets of a coded 32-level QAM signal.

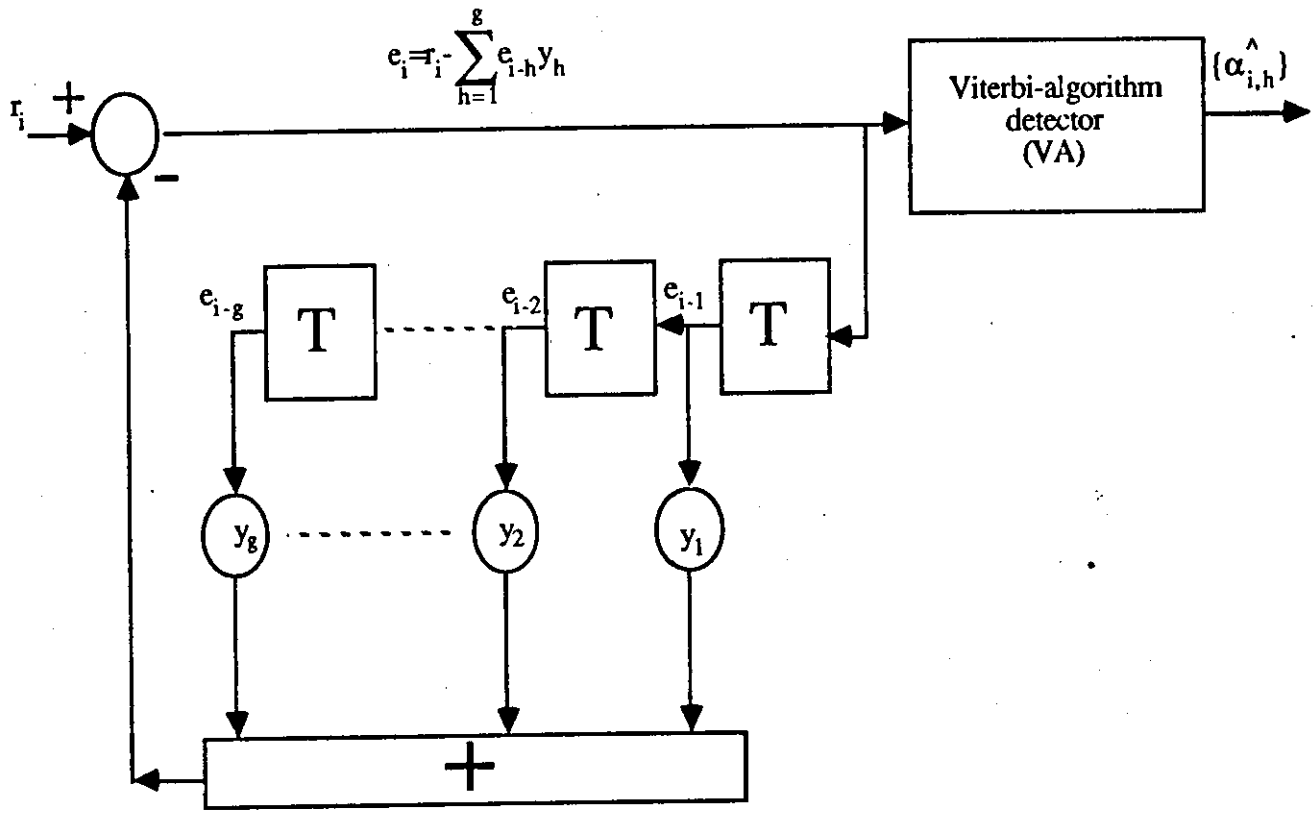


Fig.5.3 Linear feedback equalizer and the Viterbi-algorithm detector (system LC).

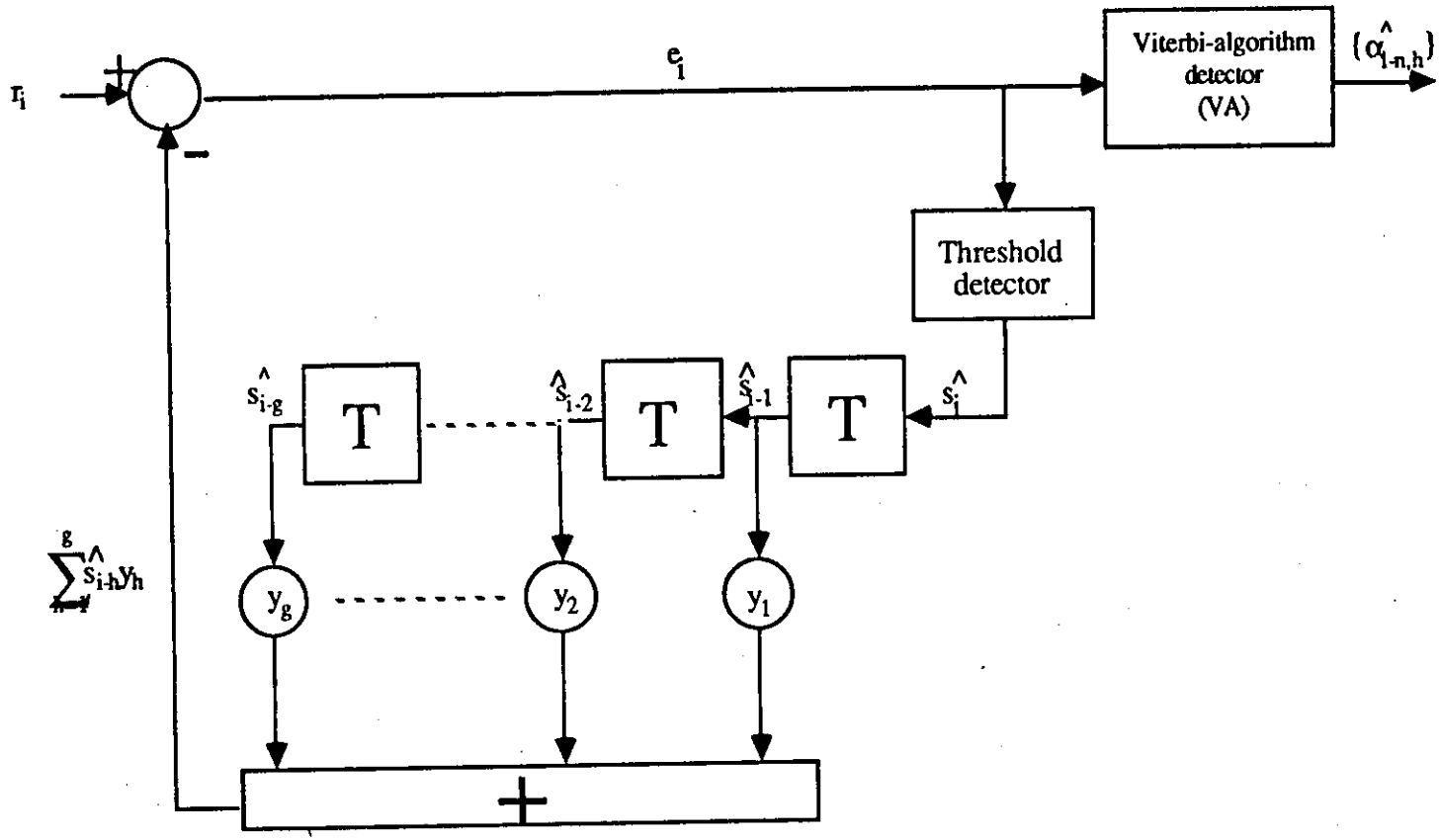


Fig.5.4 Nonlinear (decision feedback) equalizer and Viterbi algorithm detector (system NC).

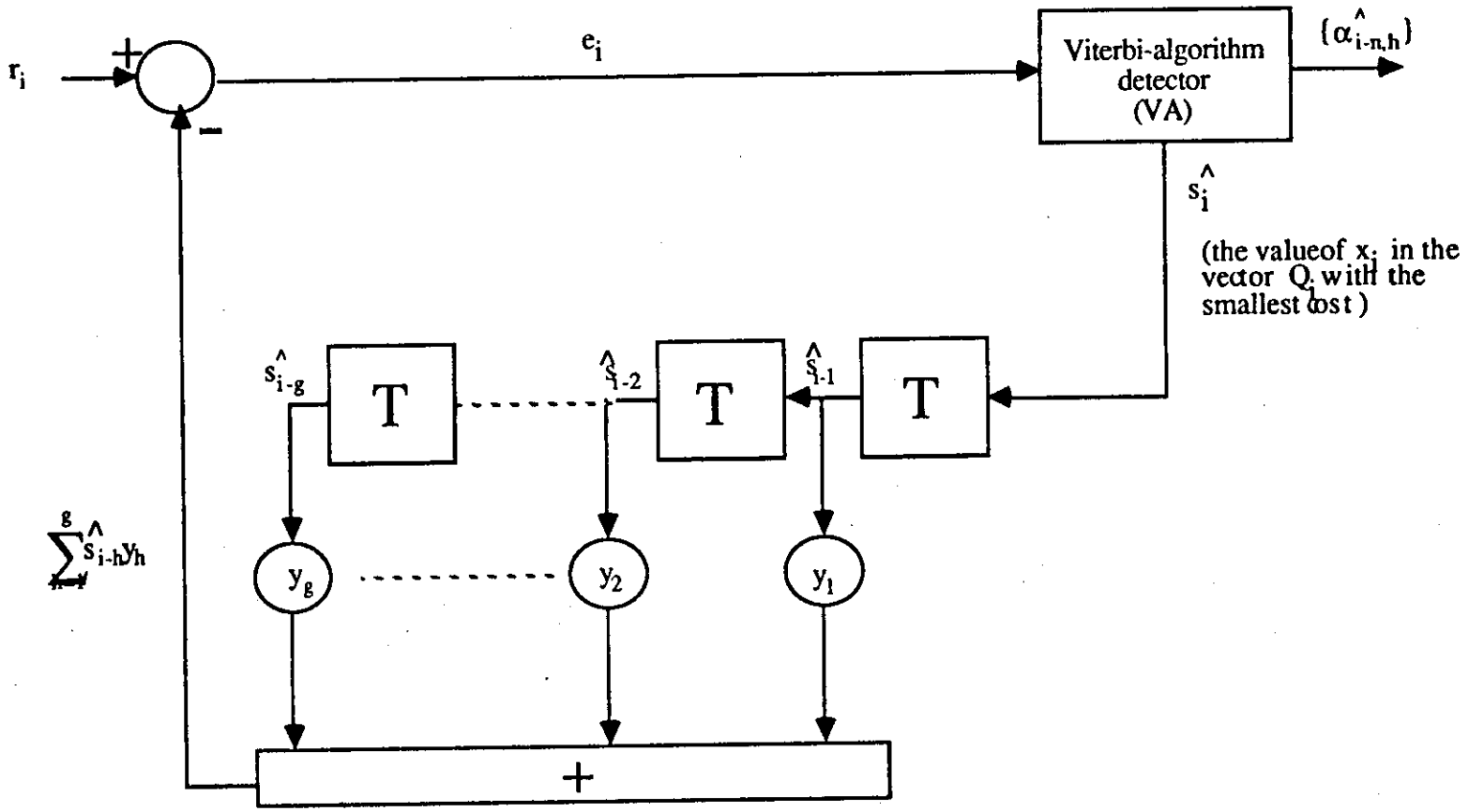


Fig5.5 The modified nonlinear equalizer with Viterbi-algorithm detector (system MNC) .

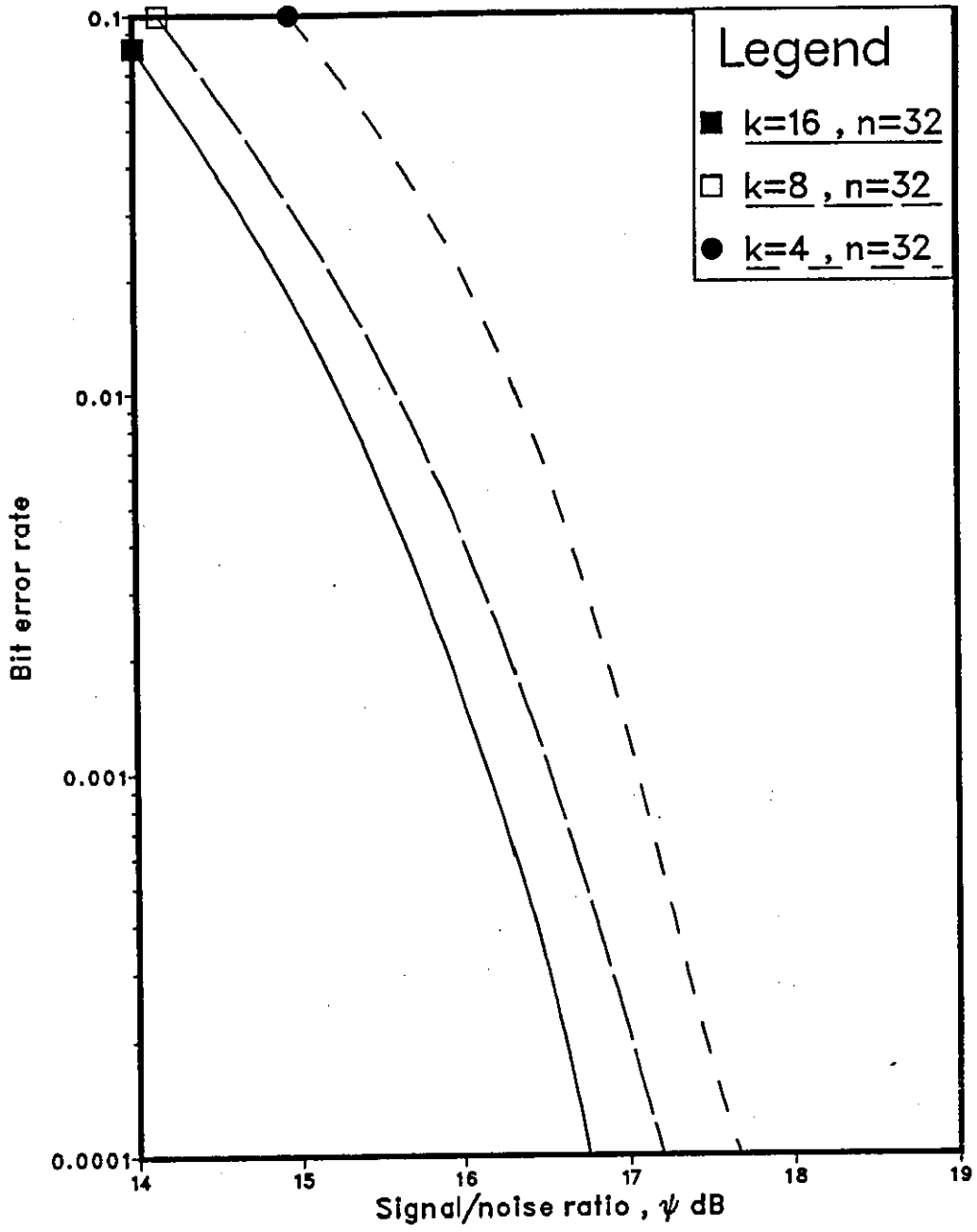


Fig.5.6 Performance of system 1C over channel E .

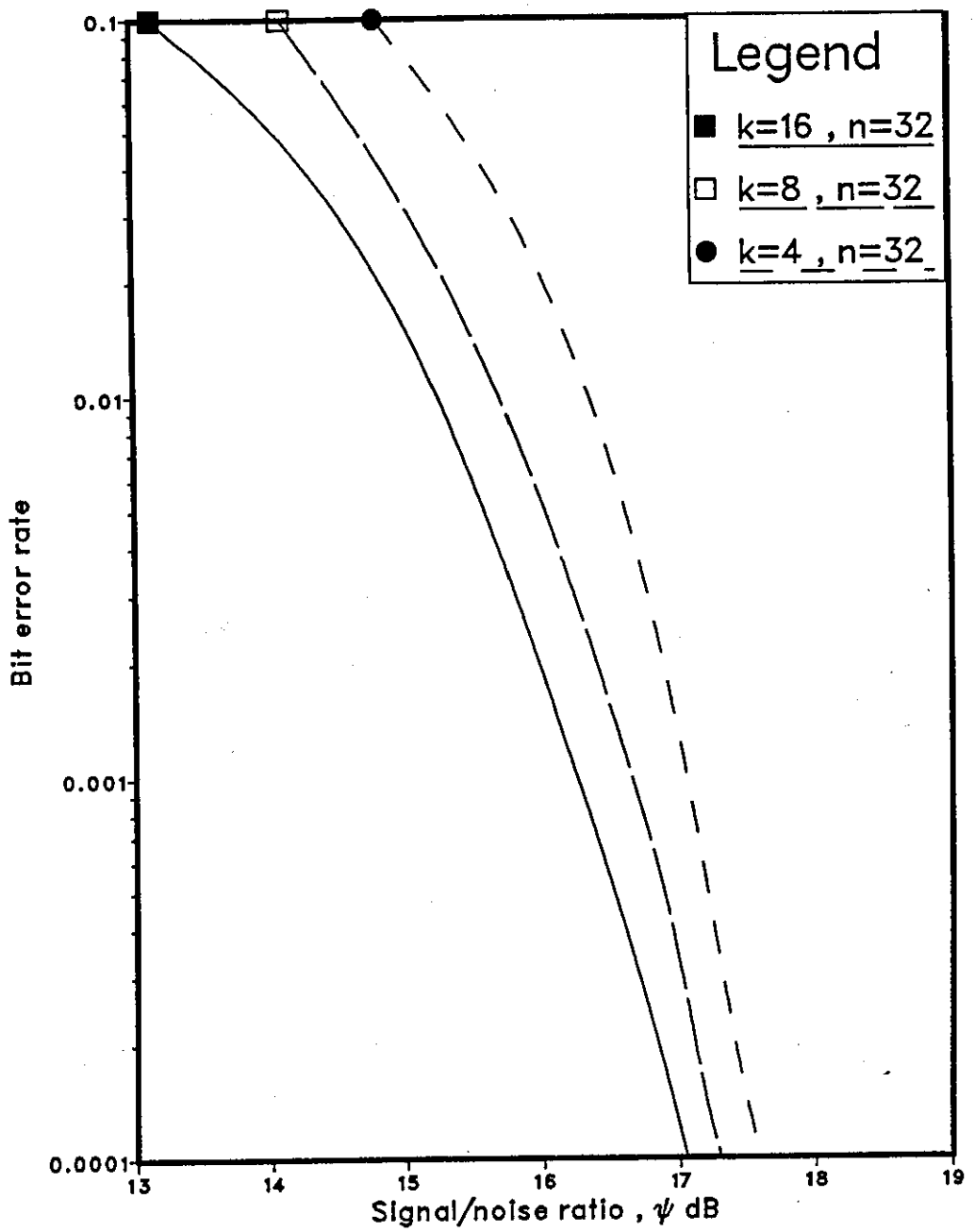


Fig.5.7 Performance of system 2C over channel E .

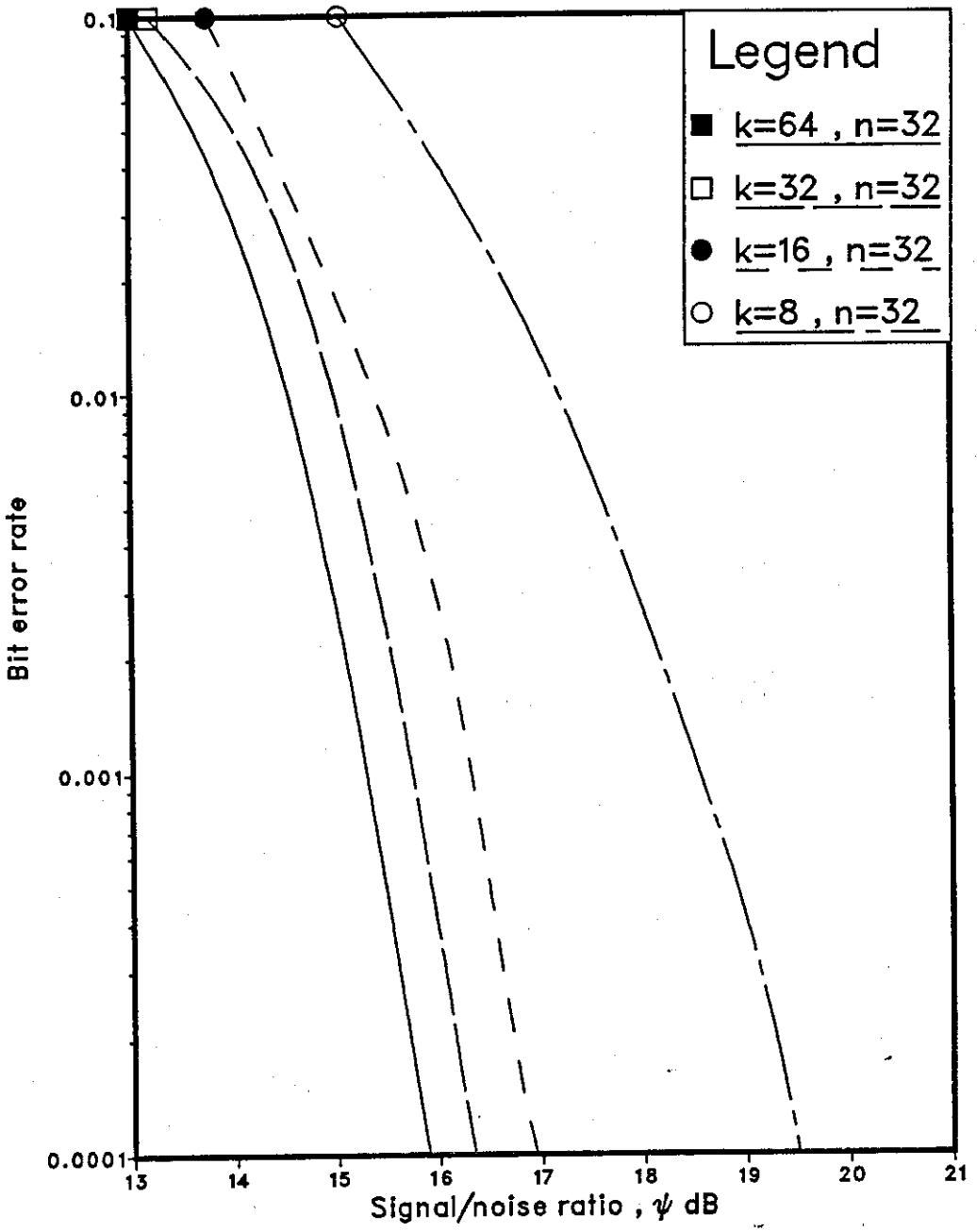


Fig.5.8 Performance of system 3C over channel E .

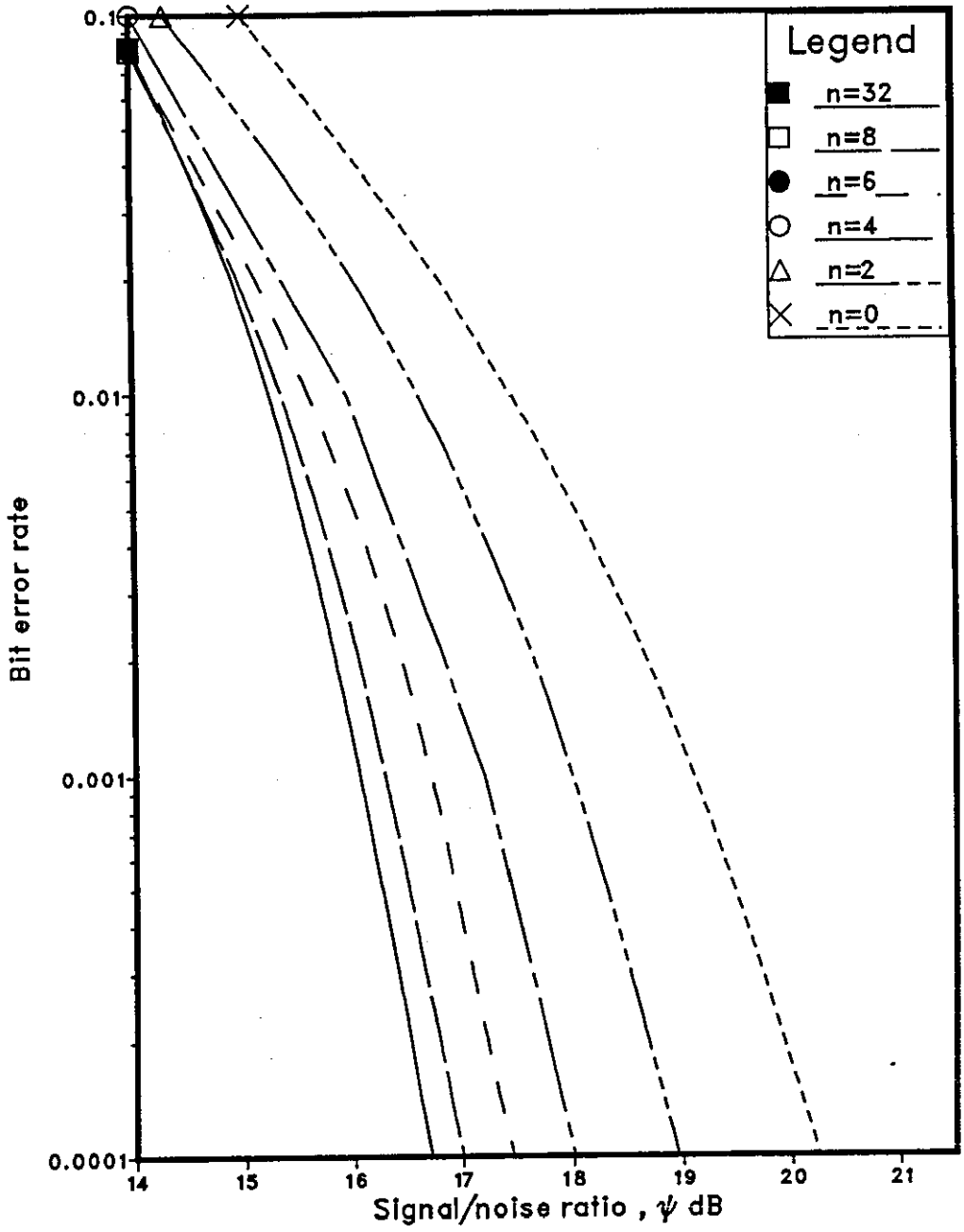


Fig.5.9 Performance of system 1C16 over channel E .

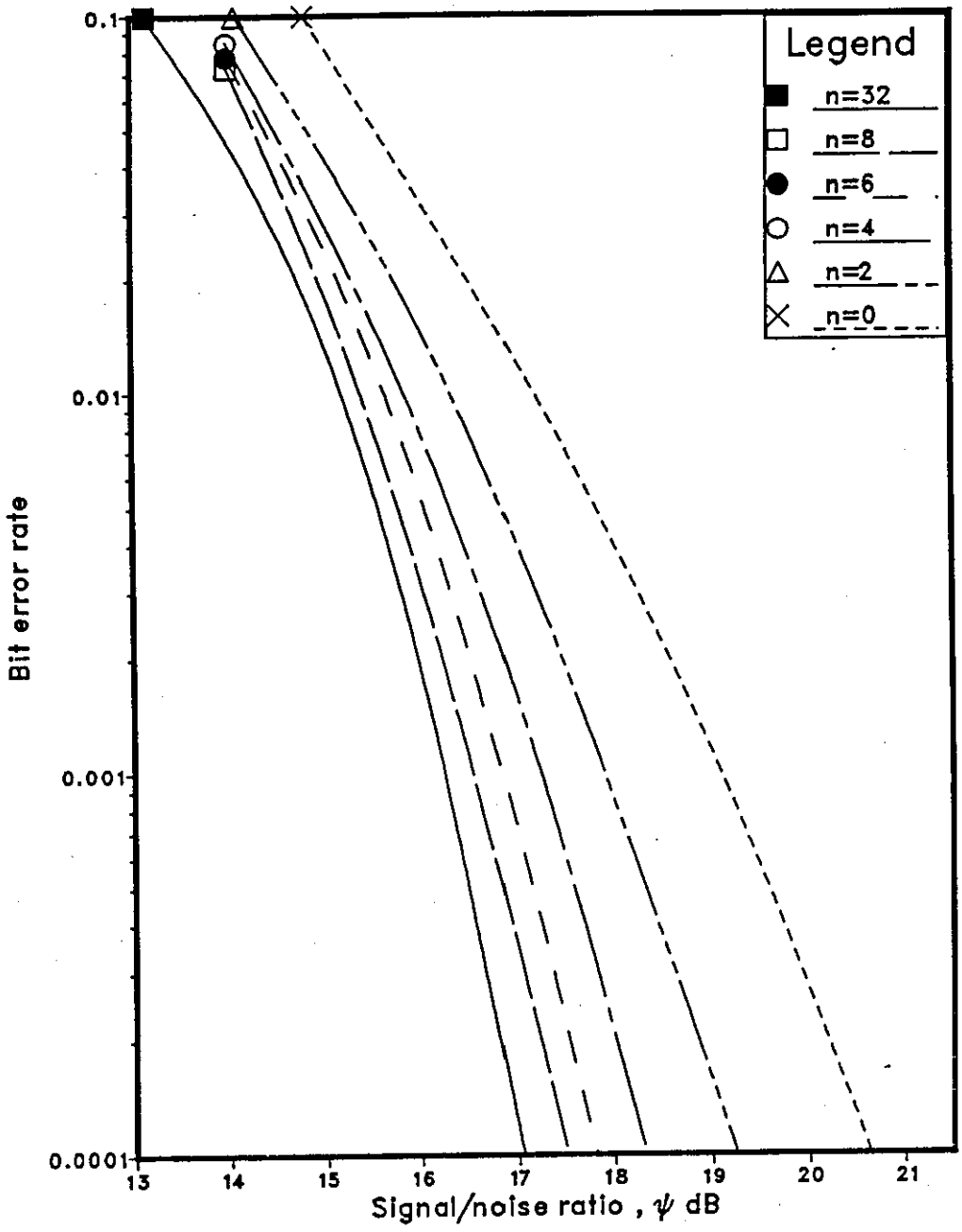


Fig.5.10 Performance of system 2C16 over channel E .

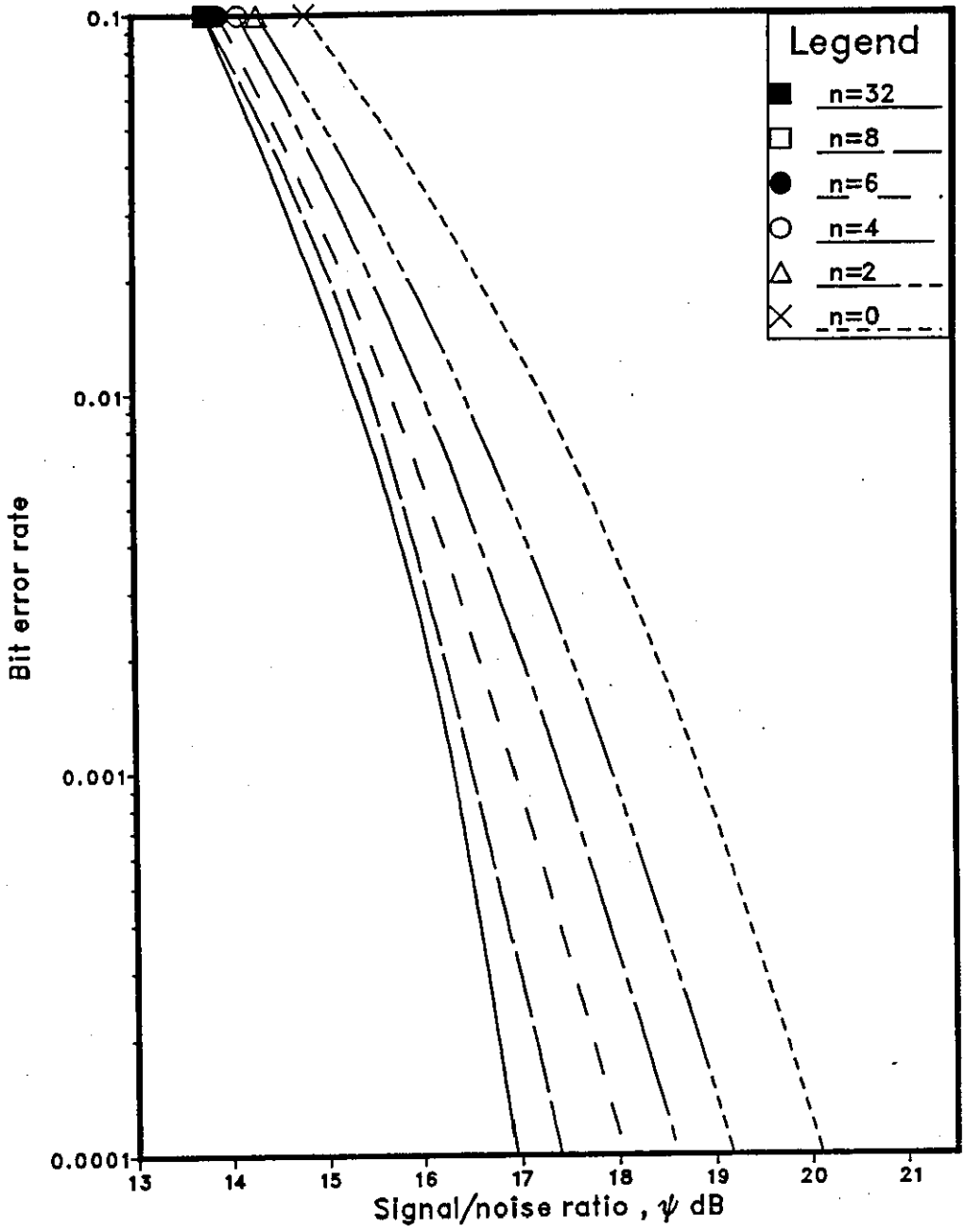


Fig.5.11 Performance of system 3C16 over channel E .

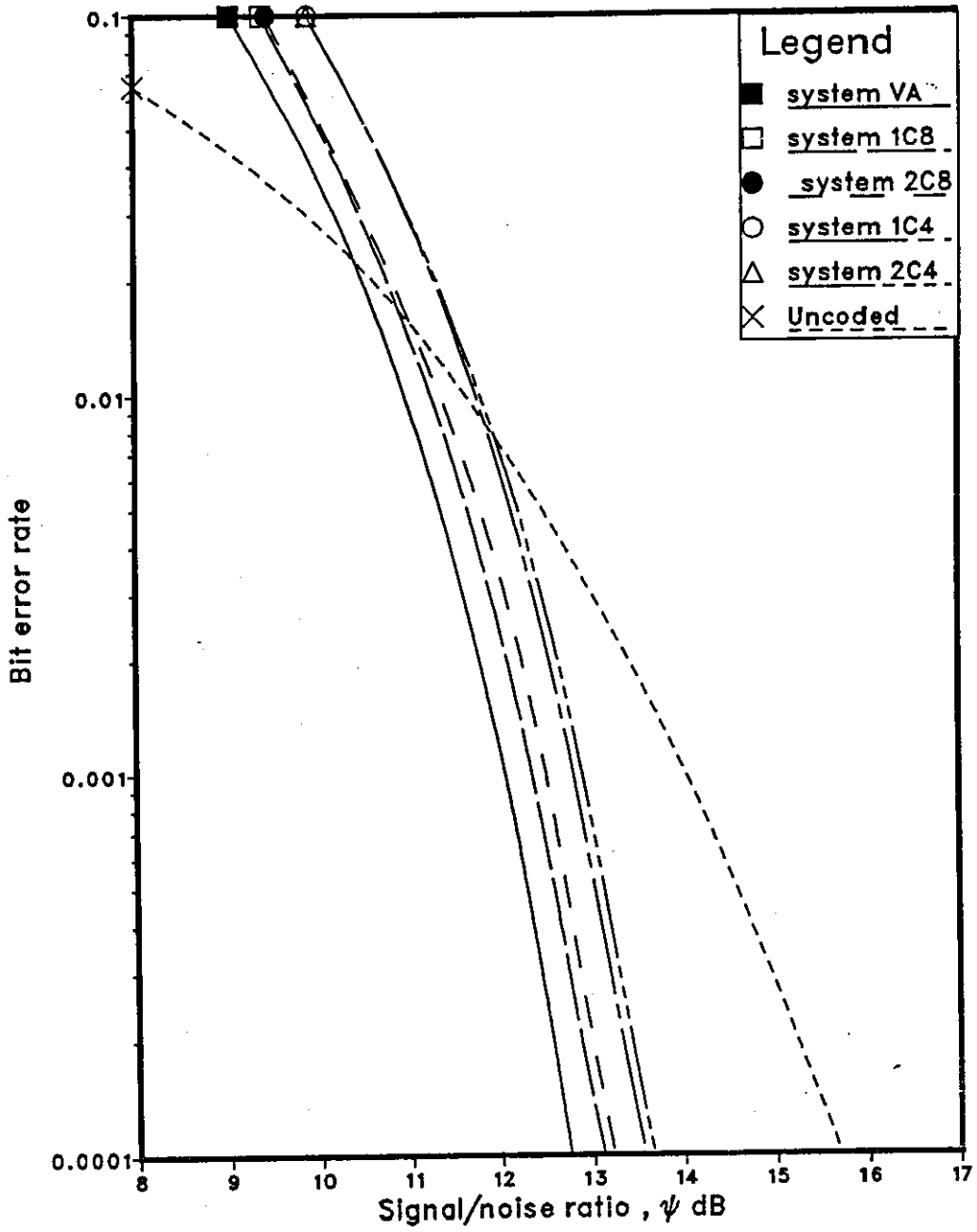


Fig.5.12 Performance of various systems over channel A.

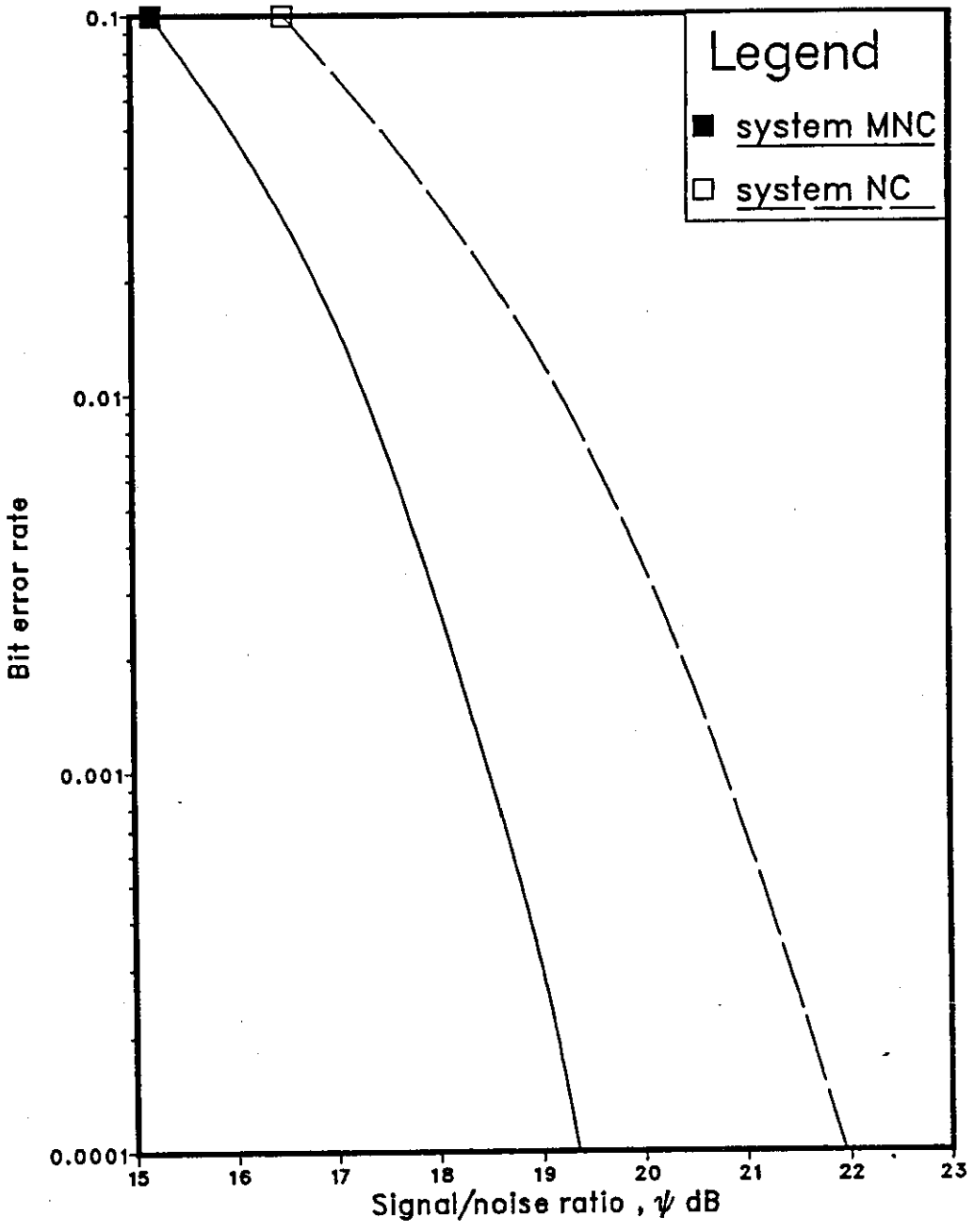


Fig.5.13 Performance of nonlinear equalizers with a coded signal over channel B .

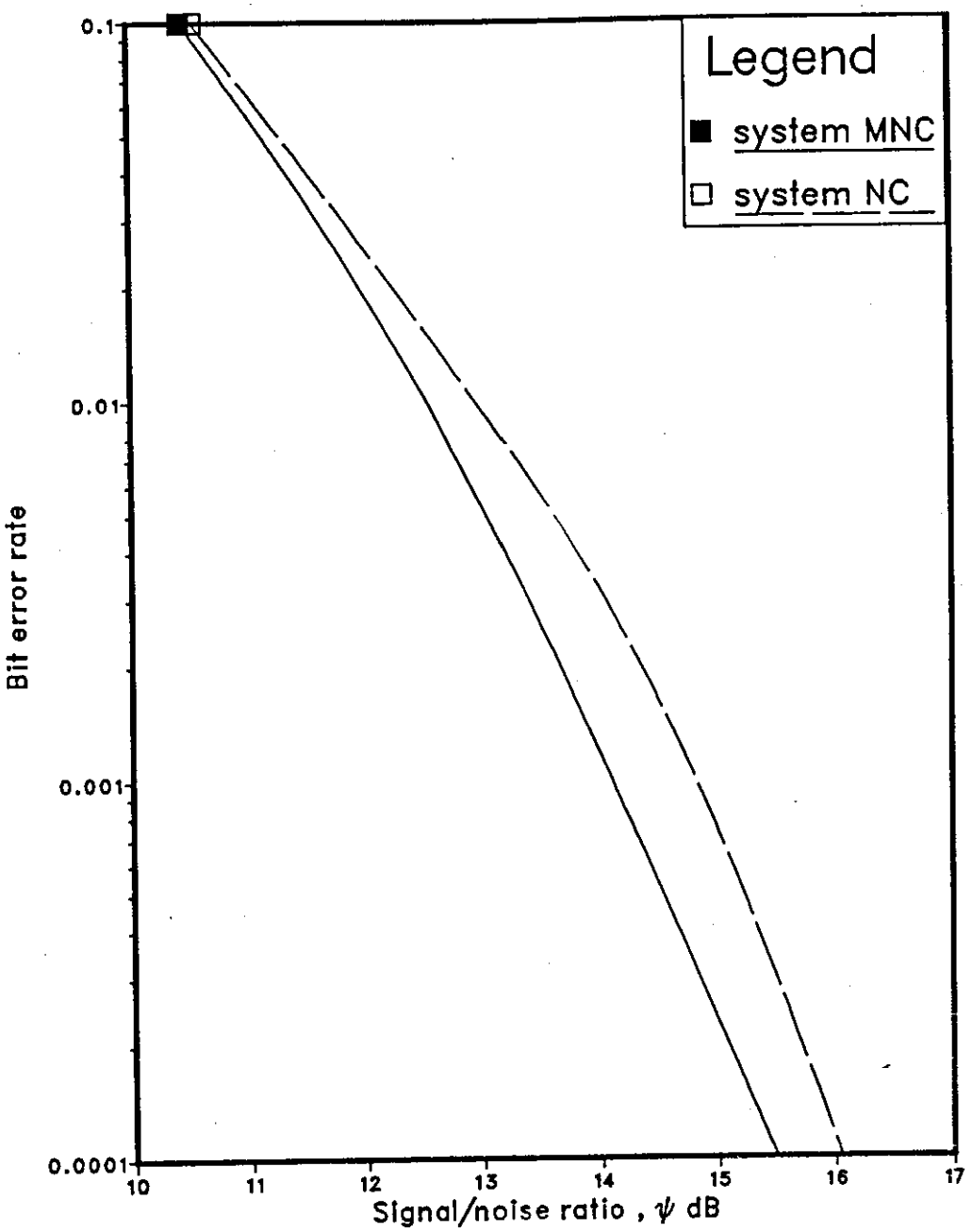


Fig.5.14 Performance of nonlinear equalizers with a coded signal over channel C .

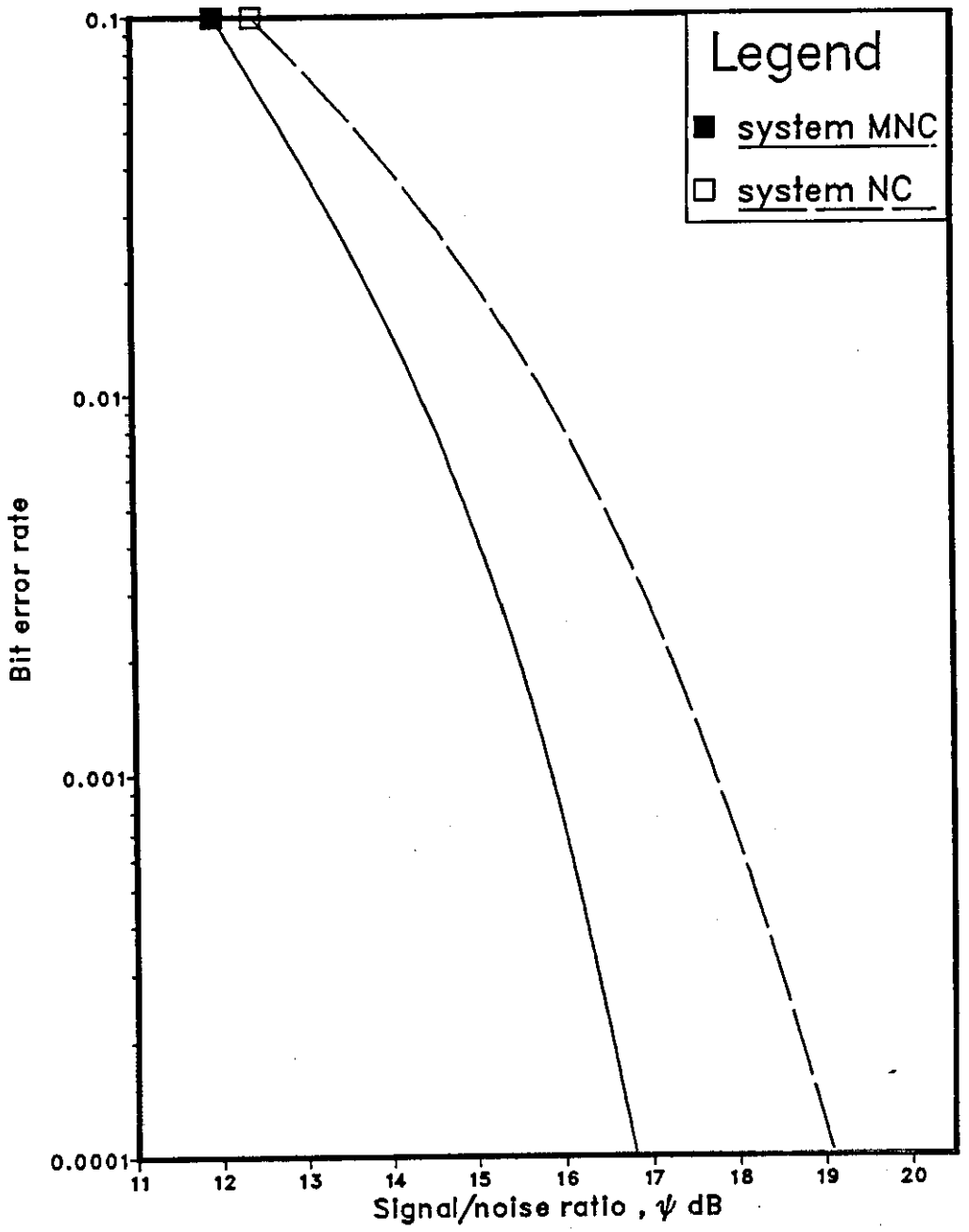


Fig.5.15 Performance of nonlinear equalizers with a coded signal over channel D .

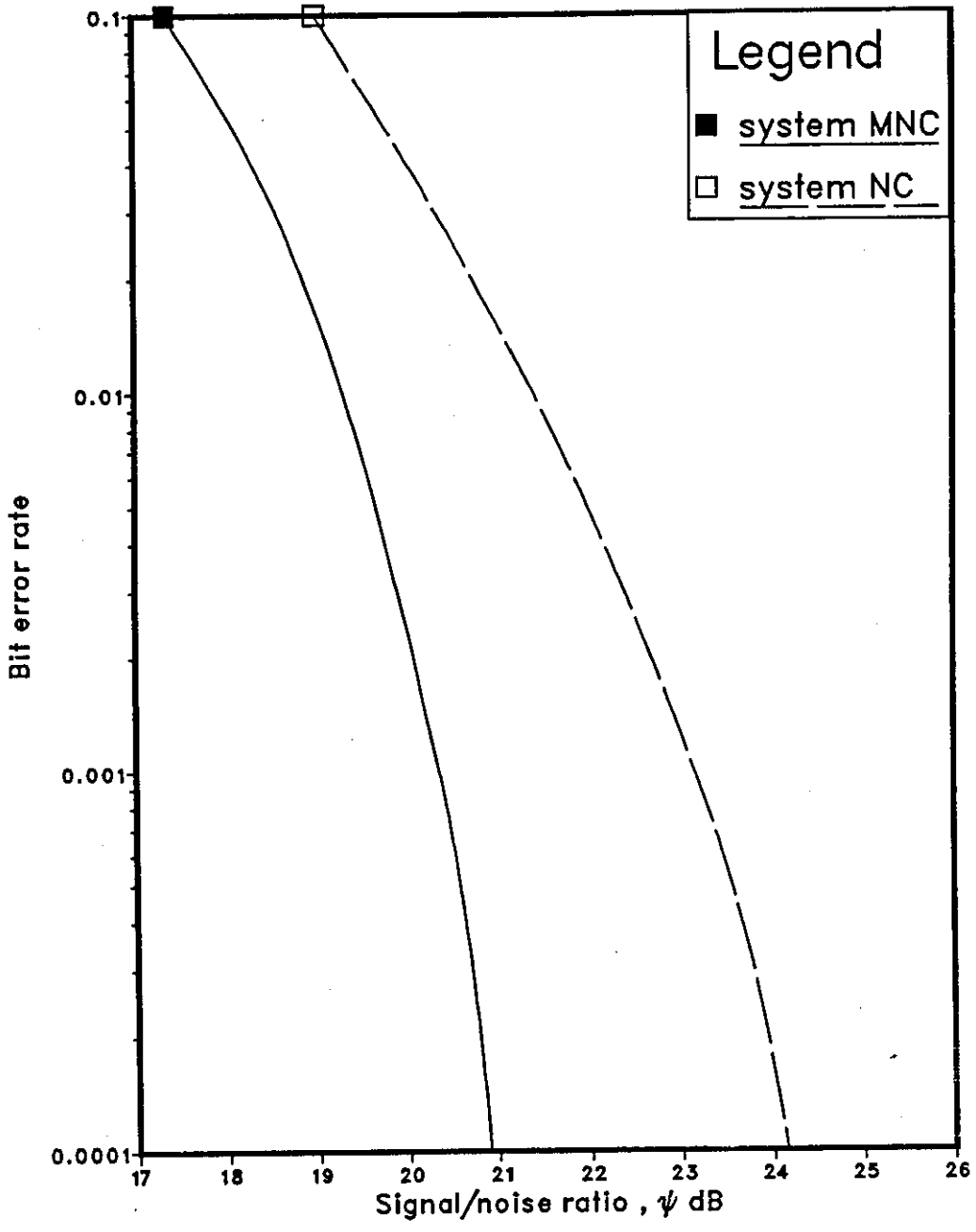


Fig.5.16 Performance of nonlinear equalizers with a coded signal over channel E .

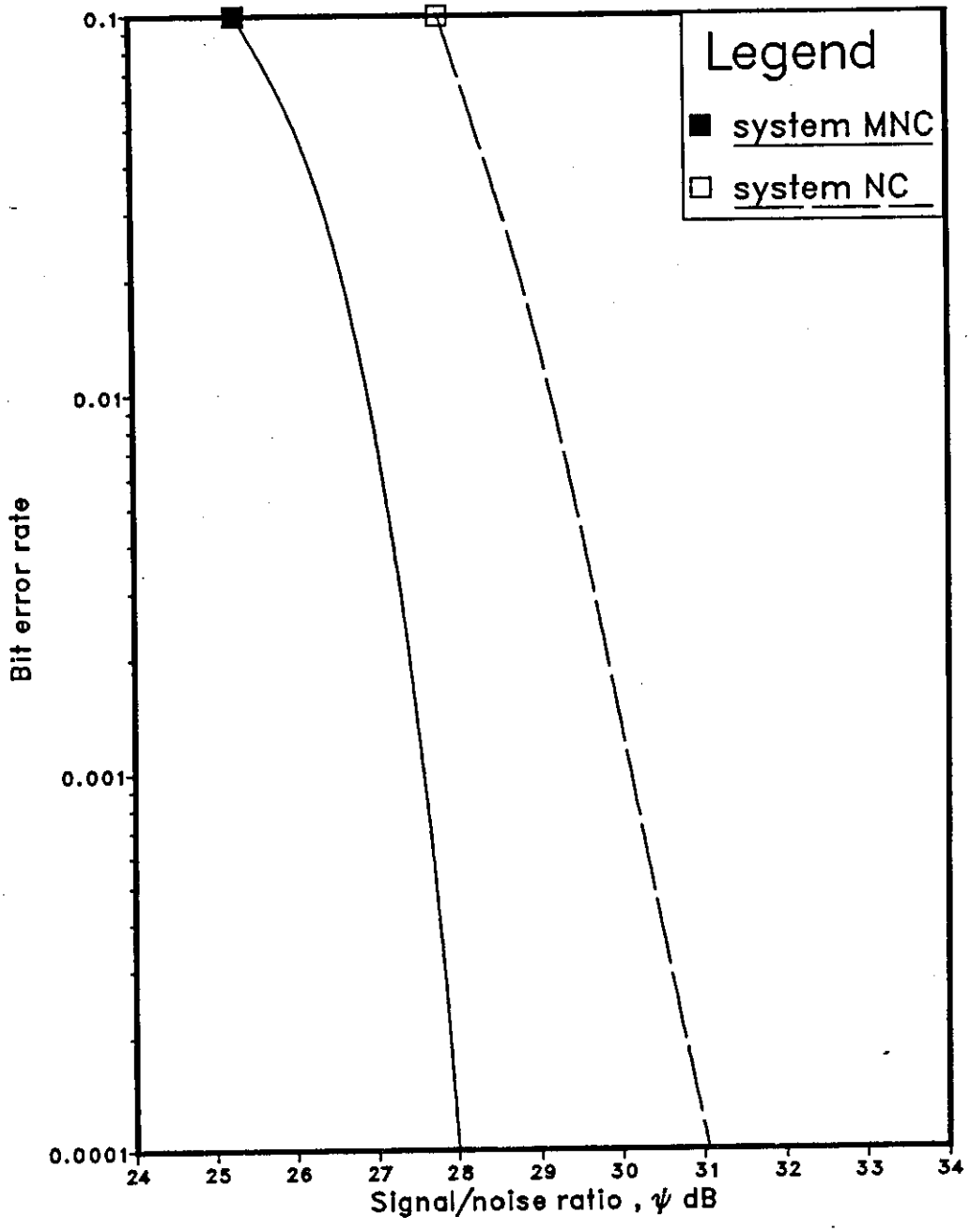


Fig.5.17 Performance of nonlinear equalizers with a coded signal over channel F .

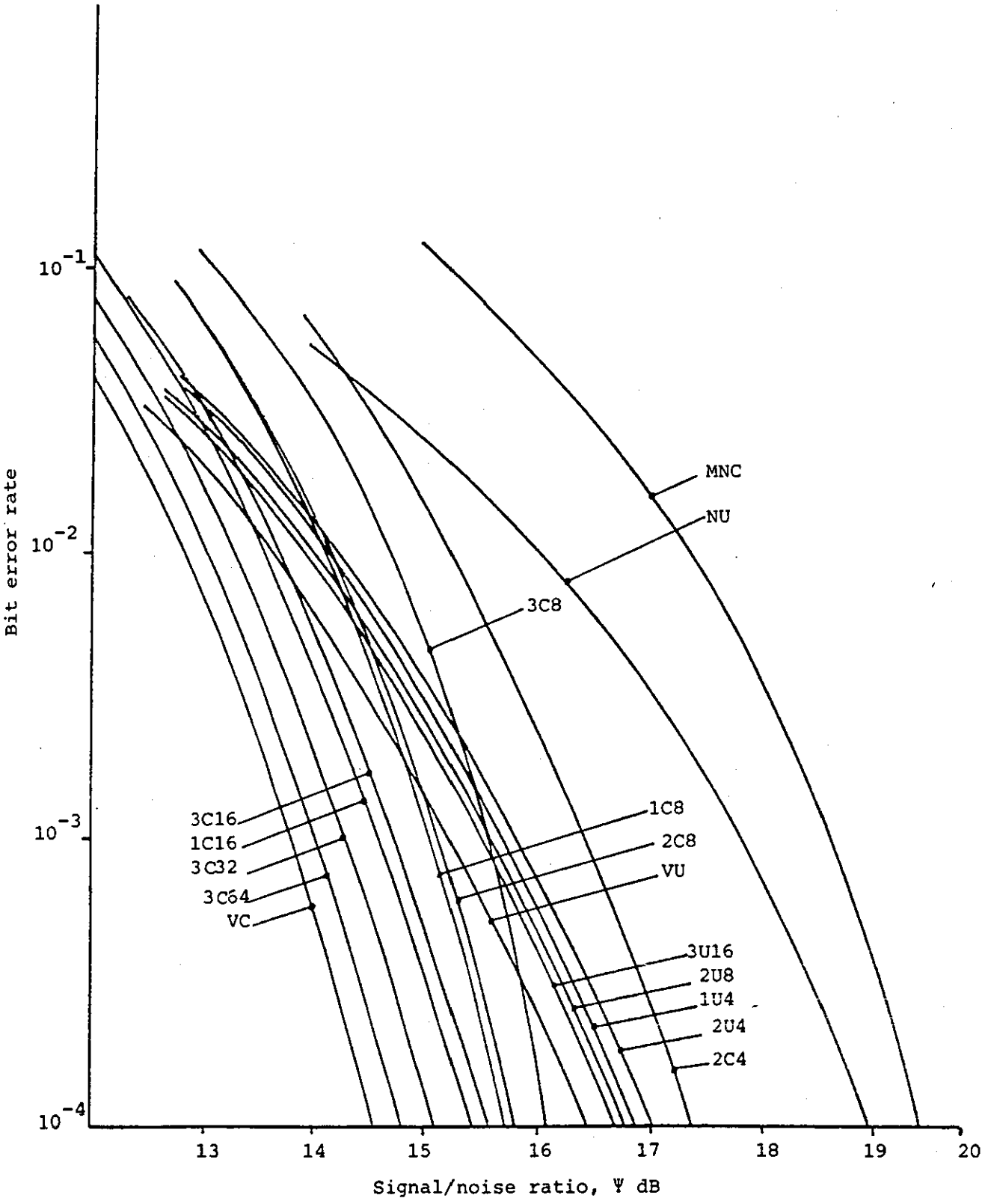


Fig. 5.18 Performance of various systems over Channel B

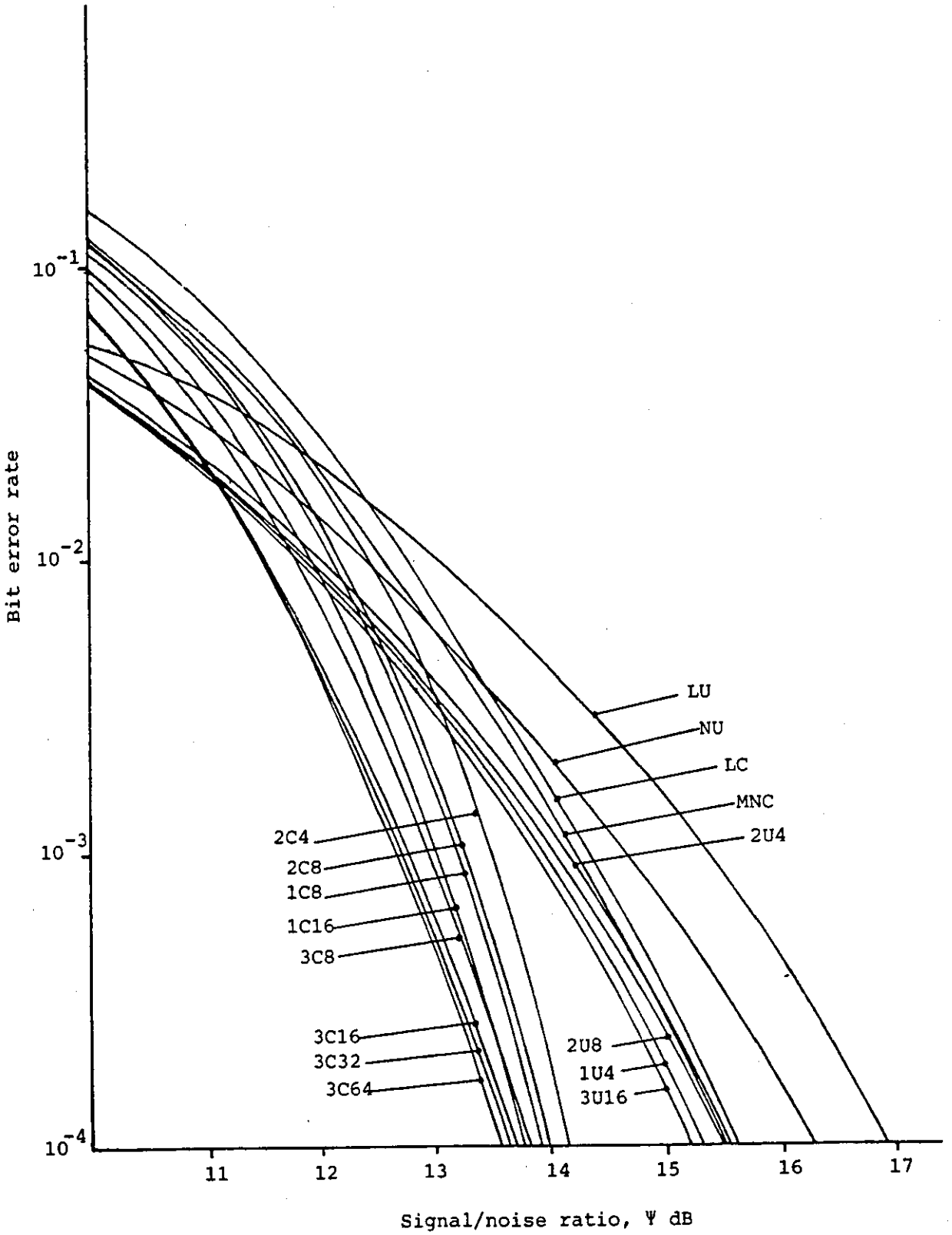


Fig. 5.19 Performance of various systems over channel C.

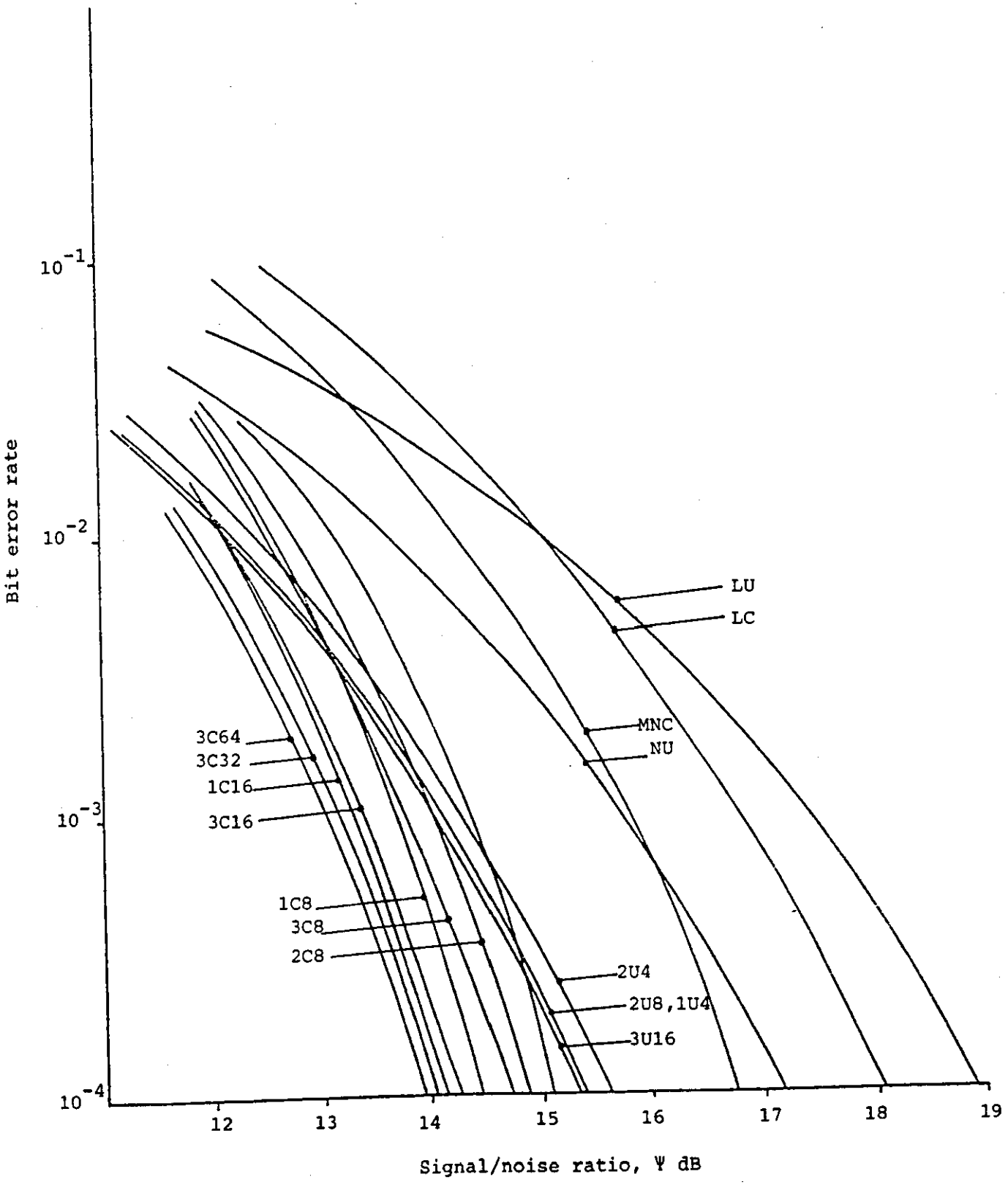


Fig. 5.20 Performance of various systems over channel D.

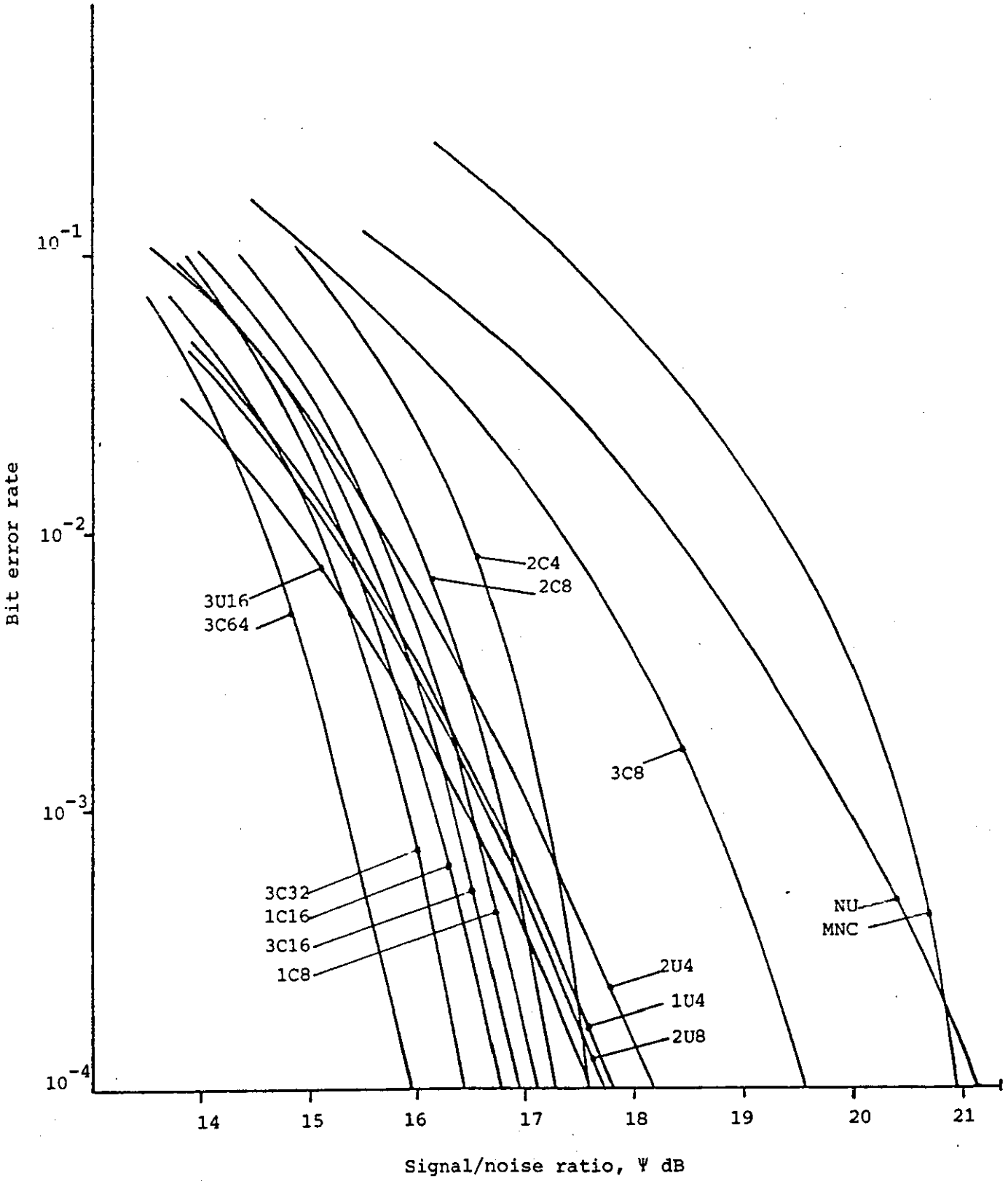


Fig. 5.21 Performance of various systems over channel E.

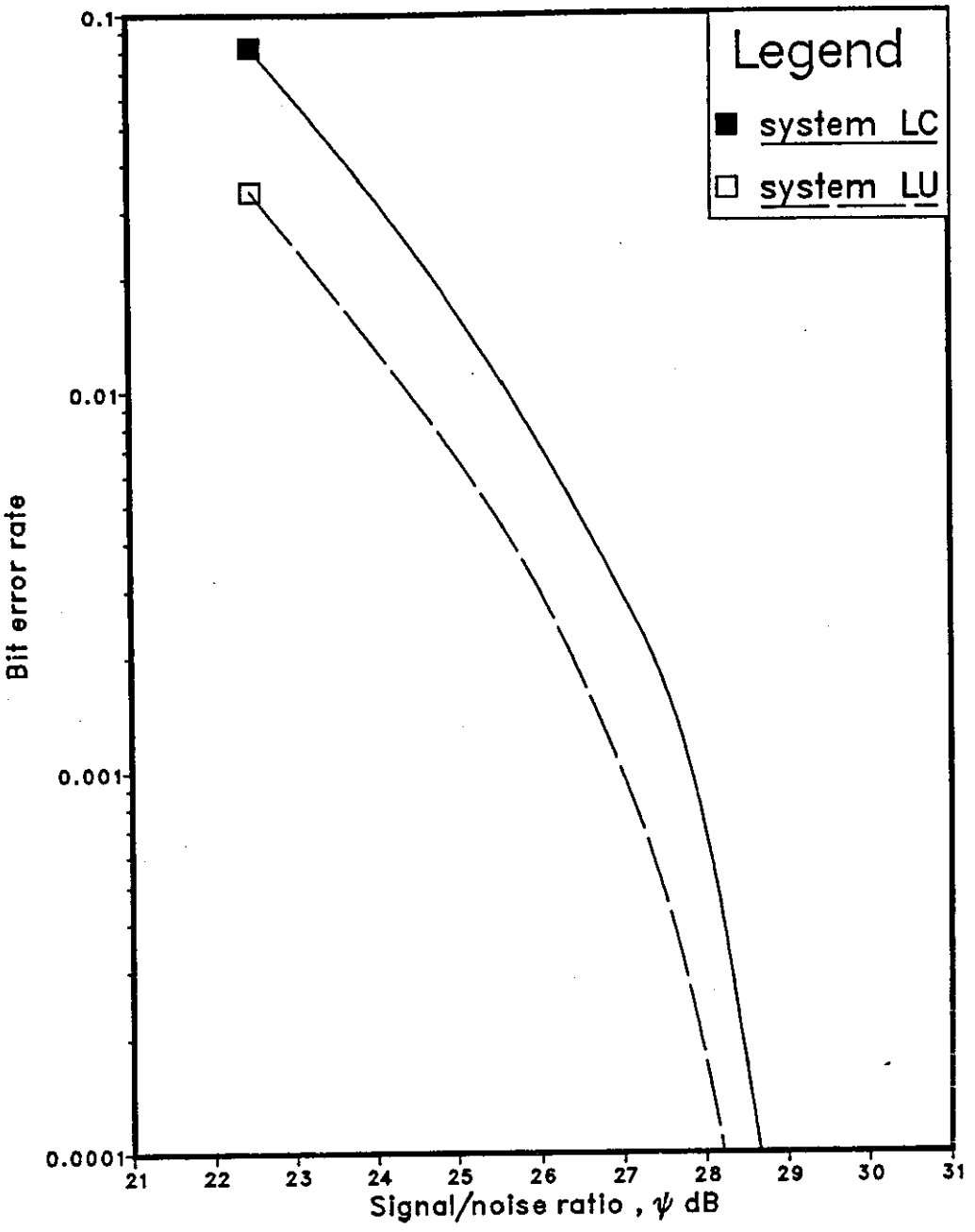


Fig.5.22 Performance of linear equalizers over channel E

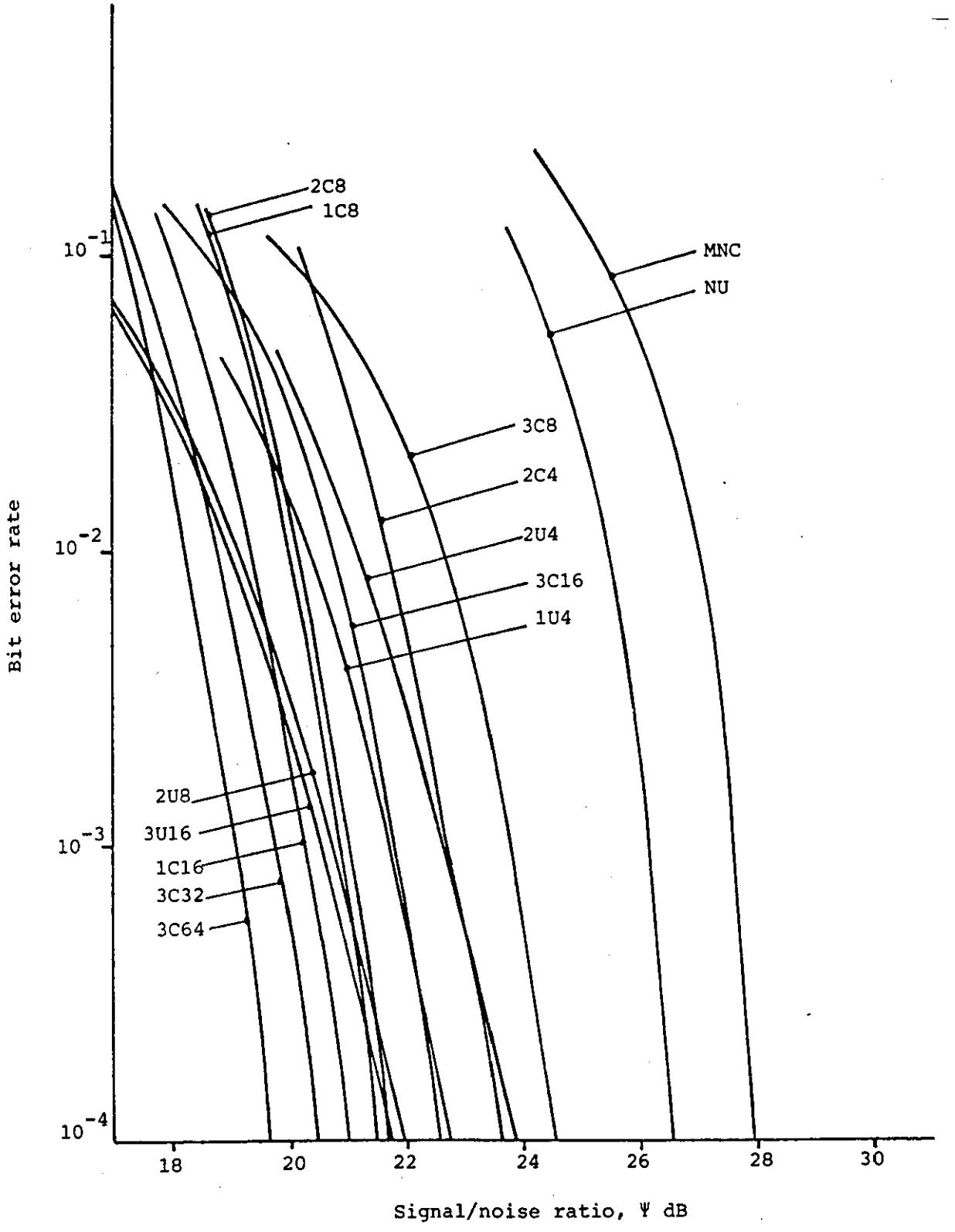


Fig. 5.23 Performance of various systems over channel F.

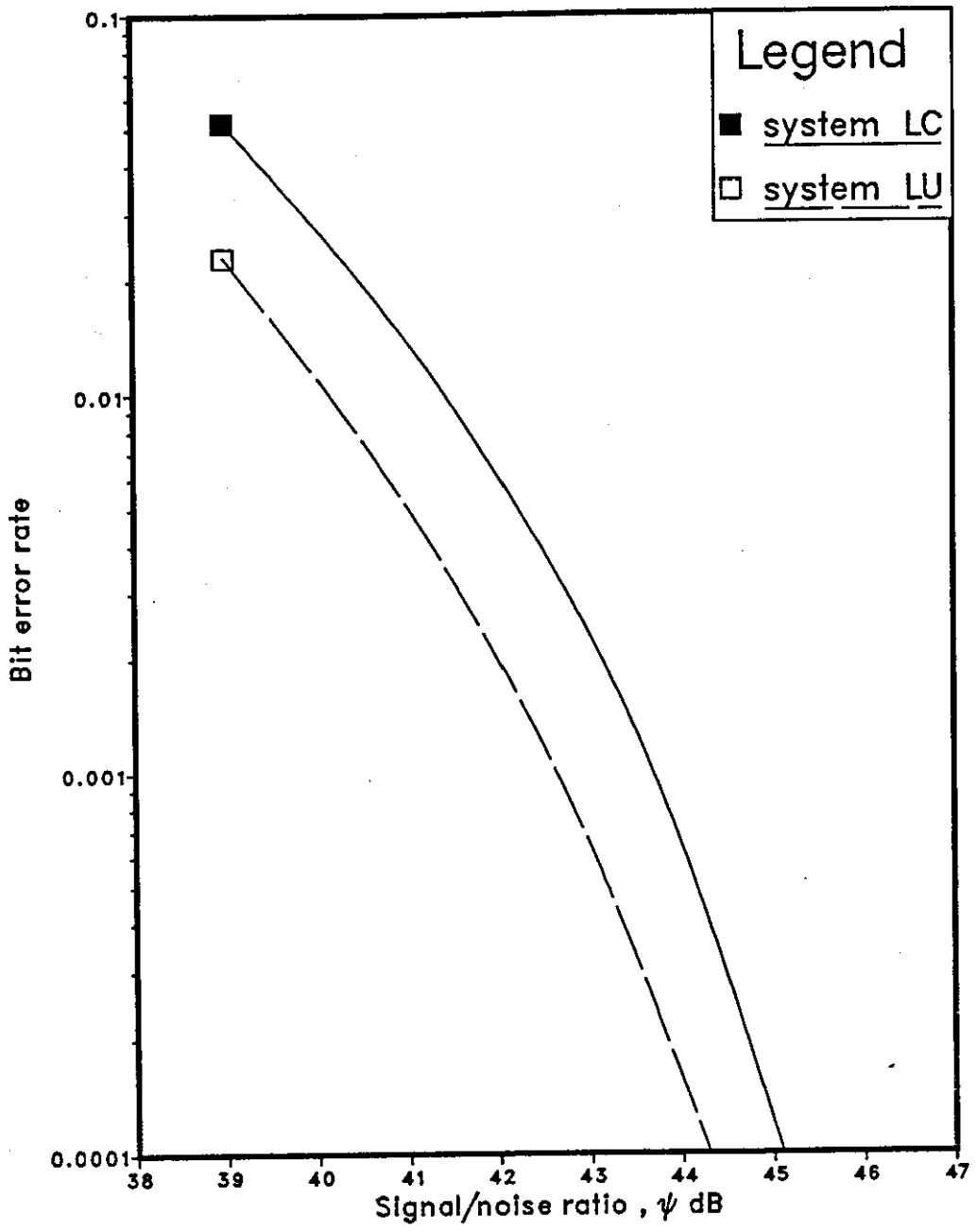


Fig.5.24 Performance of linear equalizers over channel F

CHAPTER 6

THE EFFECT OF NOISE CORRELATION ON THE PERFORMANCE OF THE DETECTORS

The effect of noise correlation on the performances of the various detectors (systems) described in Chapters 4 and 5, is investigated here. The noise correlation considered here is introduced when the uncorrelated noise samples, at the output of the adaptive linear filter, are fed through a linear equalizer. The noise samples at the receiver filter output are uncorrelated [24,39].

6.1 THE EFFECT OF NOISE CORRELATION INTRODUCED BY THE LINEAR EQUALIZER

The model of data transmission system is described in Chapter 3, and is shown again here in Fig. 6.1 and all assumptions made earlier are valid in this section also. The received sample, at time iT, at the output of the adaptive linear filter is given by

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_h + w_i \qquad \ldots \quad 6.1.1$$

where $\{s_{i-h}\}$ are the transmitted data symbols (coded or uncoded), and the real and imaginary parts of the noise samples $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and fixed variance, and so they are uncorrelated [87]. The above properties of the noise samples follow from the fact that the absolute value of the square of the receiver filter transfer function is an even function and satisfies Nyquist's Vestigial Symmetry Theorem, about the frequency 1/2T Hz, and the adaptive linear filter in Fig. 6.1 does not change the statistical properties of the noise samples as shown in Appendix E, where the filter here is assumed to be ideally adjusted to perform its function as described in Chapter 3.

When the linear equalizer is used at the receiver to equalize the distortion introduced by the channel, the equalized sample, at time iT, is given by

$$e_i = x_i + w_i' \qquad \ldots \quad 6.1.2$$

where x_i is the possible value of s_i and w_i' is the noise components in e_i. Clearly, Eqn. 6.1.2 can be used in both the coded and uncoded system, where x_i take the corresponding value of the coded or uncoded symbol.

In the case where the linear equalizer is implemented as a feedback transversal filter, as assumed here, the noise component w_i' in Eqn. 6.1.2 can be written as

$$w_i' = w_i - \sum_{h=1}^{g} w_{i-h}' y_h \qquad \ldots \quad 6.1.3$$

Since the noise samples $\{w_i\}$ are processed by the linear equalizer, the level of the noise samples $\{w_i'\}$ is increased whenever one or more of the components $y_1, y_2, \ldots, y_g$ are nonzero. The increase in the level of the noise is determined by the sum of the squares of the components of the sampled impulse response of the equalizer. Furthermore, in this case, the noise components in the equalized sequence are correlated (as can be seen from Eqn. 6.1.3). In the particular case where the channel introduces pure phase distortion (no amplitude distortion), $y_1 = y_2 = \ldots = y_g = 0$, and so no equalizer is required.

It has been shown in Chapter 5 that the use of a linear equalizer may degrade the performance of the coded system to a level no better than that of the uncoded system, when the latter also uses a linear equalizer. Since, for a given channel, the linear equalizer has the same impulse response whether the signal is coded or not, the change in the noise level is the same in the two systems. As mentioned above, another effect of the linear equalizer in the two systems is the introduction of a correlation between the neighbouring noise samples in the equalized signal. In the case of the uncoded signal, the linear equalizer is followed by a threshold level detector. This detector performs separate decisions on the equalized samples at its input and produces the corresponding detected values of the transmitted data symbols one at a time. So its operation is not significantly affected by whether or not the noise samples, at its input, are correlated. On the other hand the equalized samples in the coded system (system LC) is processed by the Viterbi algorithm detector (VA), where the operation of the latter is optimum only when the channel and the equalizer introduces no intersymbol interference, and the noise samples in

the equalized signal are uncorrelated [31]. Computer simulation tests have been carried out to show the amount of degradation in the performance of system LC due to the noise correlation introduced by the equalizer.

The absolute value of the autocorrelation function of the noise samples $\{w_i'\}$ at the output of the linear equalizer is plotted in Fig. 6.2, for channels C to F. 100000 noise samples $\{w_i\}$, which are statistically independent Gaussian random variables with zero mean, are generated and passed through the equalizer for each channel. The absolute values of the normalized autocorrelation function are determined as follows. First are evaluated the quantities $\{A_h\}$, where

$$A_h = \left| \sum_{i=1}^{100000} w_i' w_{i+h}'^{*} \right| \qquad \ldots \quad 6.1.4$$

for all integer values of h. $|x|$ is the absolute value of the complex quantity x.
The values $\{A_h\}$ are then normalized by setting to unity the maximum value, which occurs at h=0.
Fig. 6.2 shows that the correlation of the noise samples $\{w_i'\}$, introduced by the linear equalizer, is relatively higher in the cases of channels F and E. The four channels (C to F) are arranged according to their level of amplitude distortion (Chapter 3). A study of Fig. 6.2 and the performance of systems LC and LU over these four channels, shows that the greater the amplitude distortion that is equalized by the linear equalizer, the greater is the correlation of the neighbouring noise samples, and hence the greater the degradation of the performance of system LC over system LU. In order to show how much degradation in the performance of system LC is caused by this correlation, further tests have been carried out. The results of these tests are presented in Figs. 6.3 to 6.6, where the performances of system LC over channels C to F are given under two conditions. In the first condition, system LC operates exactly as described in Chapter 5 (Section 5.3), and the corresponding performance is labelled "system LC", whereas, in the second condition (which is rather unrealistic) the noise components in the equalized samples are assumed to be uncorrelated. Here the equalizer is assumed only to change the amplitude level of the noise samples rather than introducing correlation. The performance curves for the second condition are labelled "system LC (uncorrelated)". Figs. 6.3 to 6.6 show that the degradation in the performance of system LC due to the noise correlation is a function of the amplitude distortion introduced by

the channel. At bit error rates of 10^{-3} to 10^{-4}, the amount of this degradation is about 3 to 3.8 dB over channel F, while over the same range of bit error rates the corresponding degradation is about 2.5 to 3 dB over channel E. In the case of channel D and C, the degradation is reduced to about 1.3 to 1.8 dB and 0.7 to 1.4 dB, respectively. It is clear from the above figures that the effect of the noise correlation at bit error rates above 10^{-2} is relatively small over the channels tested here.

6.2 THE EFFECT OF NOISE CORRELATION INTRODUCED BY THE RECEIVER FILTER

It is assumed throughout this work that the transfer function of the receiver filter is deliberately chosen so that the noise components at the receiver filter output are uncorrelated. In this section it assumed that the filtering carried out by the data transmission system (Fig. 6.1) is equally shared between the transmitter and the receiver filters. The attenuation and the group delay characteristics of the equipment filters used here is shown in Fig. 6.7. This is different from that used in the model of the data transmission system described in Chapter 3, and is here called equipment filters-2. These filters are actually a combination of equipment filters-1 (Chapter 3) and a radio filter, which has been used for the transmission of data at a rate of 9600 bit/s over a model of an HF radio link [39,88]. The equipment filters-2 introduce slightly higher levels of signal distortion than the equipment filters-1. When the filtering is equally shared between the transmitter and receiver filters, the noise at the output of the receiver filter becomes correlated. In fact, this is the main reason behind the usage of the equipment filters-2 in this section. The sampled impulse responses of the transmitter and the receiver filters, when the filtering is equally shared between them, are shown in Table 6.1, and these are derived in the manner described in Appendix D.

The assumptions made in the model of data transmission system (Fig. 6.1) are the same as that given in Chapter 3. The sampled impulse response of the linear baseband channel and the adaptive linear filter of Fig. 6.1 is given in Table 6.2, where the transmission path is assumed to be ideal and introduces no signal distortion. The received sample at the output of the adaptive linear filter (Fig. 6.1), at time iT, is given by

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_h + w_i^c \qquad \ldots \quad 6.2.1$$

where $\{w_i^c\}$ are the noise samples at the output of the adaptive linear filter, and they are correlated as described above. The absolute value of the autocorrelation function, defined in Eqn. 6.1.4, of the noise samples $\{w_i^c\}$ is shown in Fig. 6.8.

Computer simulation tests have been carried out to evaluate the performances of the different detectors described in Chapters 4 and 5 for the case where there is no telephone circuit (these are referred here to as the back-to-back tests). The tests have been carried out under two conditions. In the first, the filtering in the system is assumed to be equally shared between the transmitter and receiver filters, so that the noise samples $\{w_i^c\}$ are correlated as described above, and the results of the tests are shown in Fig. 6.9. In the second condition the receiver filter introduces no correlation between the neighbouring noise samples $\{w_i^c\}$ as in the remaining parts of this work and the results are shown in Fig. 6.10. The results in Figs. 6.9 and 6.10 do not include the linear equalizers which are shown in Fig. 6.11. Each curve in the above figures is labelled by the corresponding detector (system) as in Chapters 4 and 5. It is clear from Figs. 6.9 and 6.10 that in the case where the receiver filter introduces correlation the performances of the systems, tested here, irrespective of whether the signal is convolutionally coded or not, are slightly better than the corresponding systems in the case where the receiver filter introduces no correlation. On the other hand, the improvement in the performances of the coded systems, in tolerance to additive noise, over the uncoded systems remains unchanged. For example, in Fig. 6.10 system 3C64 (which is system 3C used for the encoded signal with 64 stored vectors) has an advantage in tolerance to additive noise of about 2 dB over system 3U16 (which is system 3U for the uncoded signal with 16 stored vectors) at bit error rate of 10^{-4} , and the corresponding advantage of system 3C64 over system 3U16 in Fig. 6.9 is also about 2 dB. Systems LU and LC in Fig. 6.11 give a relatively poor performance when compared to all other systems for the case where the receiver filter introduces no correlation between the neighbouring noise samples $\{w_i^c\}$. When the filtering is equally shared between the transmitter and receiver filters, however, the linear equalizers give surprisingly better performances, as can be seen in Fig. 6.11. In this case system LC is only about 1 dB below system 3C64 at bit error rate of 10^{-4} , while the performance of system LU is about 1.1 dB

below system 3U16 at the same bit error rate, as it is clear from Figs. 6.9 and 6.11. These figures also show that the performance of system LC is as good as system 1C16, with a slightly better performance at bit error rates above 10^{-2}. System LC gains an advantage of about 1 and 1.3 dB over systems 2C8 and 3C8, respectively, at bit error rate of 10^{-4}. It also gives a better performance than system MNC (the modified nonlinear equalizer for a coded signal). Figs. 6.9 and 6.11 show that, in the case where the signal is uncoded, the performance of the linear equalizer for the uncoded signal (system LU) is better than that of the nonlinear equalizer (system NU). Furthermore, as can be seen from Fig. 6.11, the use of a linear equalizer with the convolutionally encoded signal (system LC) gains an advantage of about 1.5 to 2.1 dB over the corresponding uncoded system over the range of bit error rates of 10^{-3} to 10^{-4}, when the receiver filter introduces noise correlation. In the case where the receiver filter introduces no noise correlation there is no such advantage and the performance of system LC is only better than that of system LU at bit error rates below 2×10^{-4}. So the linear equalizers, unlike the other systems, are affected by the presence of the noise correlation. In fact, the correlation introduced by the receiver filter works in favour of the coded system (system LC). But it has been shown in the previous section that the correlation introduced by the linear equalizer itself may degrade the performance of the coded system. As mentioned earlier in the back-to-back tests there is no telephone circuit included in the model, and so the sampled impulse response of the linear baseband channel in Fig. 6.1 is the resultant response of the transmitter and the receiver filters (Table 6.1). Now, since the adaptive linear filter has no effect on the amplitude distortion introduced by the linear baseband channel when perfectly adjusted, as assumed here, the linear equalizer here equalizes the amplitude distortion introduced by the resultant response of the transmitter and receiver filters. When the filtering is shared equally between the transmitter and receiver filters, the equalizer equalizes twice the distortion introduced by the receiver filters, and so it tends to reduce the levels of the noise components at the output of the receiver filter, and at the same time it reduces the correlation between the noise samples. Fig. 6.12 shows the absolute value of the autocorrelation function of the noise samples at the outputs of the receiver filter and the linear equalizer, for the case where the filtering is equally shared between the transmitter an receiver filters. The autocorrelation function of the noise samples at the output of the linear equalizer for the case where the receiver filter introduces no

correlation between the neighbouring noise samples is also shown in Fig. 6.12. It is clear from this figure that the equalizer reduces the correlation between the neighbouring noise samples, when the filtering is equally shared between the transmitter and receiver filters. This explains why there is an improvement in the performance of the linear equalizers (systems LC and LU) over the corresponding systems for the case where the receiver filter introduces no noise correlation. It is clear that the improvement of system LC here is greater than that of system LU. This is due to the fact that system LC performs, relatively, better than system LU when the noise correlation is reduced.

Finally, to show the effect of the noise correlation (introduced by the receiver filter) on the performance of the linear equalizers in the presence of a telephone circuit, further tests have been carried out. In these tests, telephone circuit 3 (Chapter 3) is used. The sampled impulse response of the linear baseband channel and the adaptive linear filter is given in Table 6.3. Equipment filters-2 (Fig. 6.2) is also used here. The results of the tests are shown in Fig. 6.13, for the two cases where the receiver filter introduces noise correlation between the neighbouring noise samples (the filtering is equally shared between the transmitter and receiver filters) and when the receiver filter introduces no noise correlation. Clearly, Fig. 6.13 shows that there is no significant change in the relative performances between system LC and LU in the two cases. Unlike the results of the back-to-back tests, the equalizers here have no role in reducing the correlation introduced by the receiver filter. Instead, it actually introduces further correlation between the neighbouring noise components in the equalized samples, and it also increases the noise level. Although no results of other systems are presented here, it is clear from the results of the back-to-back tests that their performances are not effected by the presence of the noise correlation, introduced by the receiver filter.

Transmitter filter		Receiver filter	
Real Part	Imaginary Part	Real Part	Imaginary Part
-0.0037334	0.04893466	-0.04079517	0.02862715
-0.206655	0.94708762	-0.73830338	0.57427385
-0.0719658	0.18095437	-0.23593839	0.15276818
0.06370091	-0.11603890	0.15783024	-0.10705166
-0.03112757	0.03610002	-0.07081625	0.03986585
0.01077012	-0.00268702	0.02202331	-0.01014814
-0.0065853	-0.00483293	-0.00652528	0.00415777
-0.00092260	0.00081502	0.00155247	0.00197557

Table 6.1 Sampled impulse responses of the transmitter and receiver filters (Equipment filters-2) at sampling rate of 2400 samples/s.

Real Part	Imaginary Part
1.0000000	0.0
0.6280652	0.0670085
-0.2245447	-0.0292450
0.0200149	0.0061246
0.0321720	0.0095349
0.0189756	-0.0071542
0.0069501	0.0069183
-0.00022625	-0.0034449
-0.004014	0.0009599
0.0002131	-0.0001827
-0.0000618	0.0000209
0.0000204	0.0000208
0.0000047	-0.0000008
0.0000003	-0.0000002

Table 6.2 Sampled impulse responses of the linear baseband channel (without telephone circuit) and adaptive linear filter in Fig. 6.1.

Real Part	Imaginary Part
1.0000000	0.0
0.54635	0.10526
-0.60866	0.13043
0.13559	-0.22501
-0.01326	0.10803
0.00260	-0.01811
-0.01624	0.01245
0.00605	-0.01110
-0.00080	0.00740
-0.00069	0.00382
-0.00091	-0.00760
0.00123	0.00481
0.00197	-0.00111
-0.00116	-0.00856
0.00203	0.00562
0.00284	0.00385
-0.00616	-0.00531
-0.00220	-0.00410
-0.00266	-0.00169
-0.00063	0.00065

Table 6.3 Sampled impulse response of the linear baseband channel and adaptive linear filter in Fig. 6.1, for telephone circuit 3 (channel E).

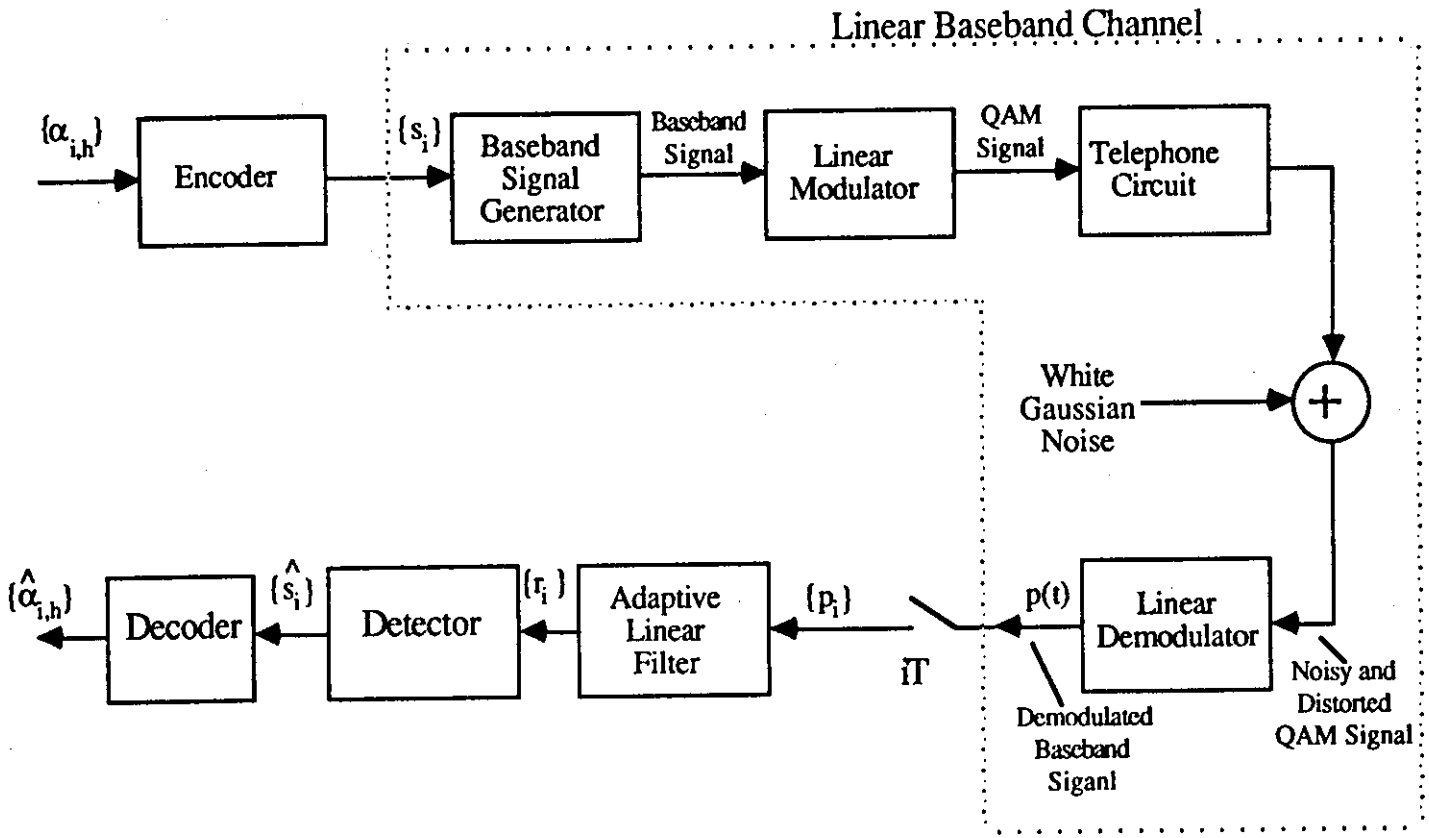


Fig.6.1 Model of data transmission system

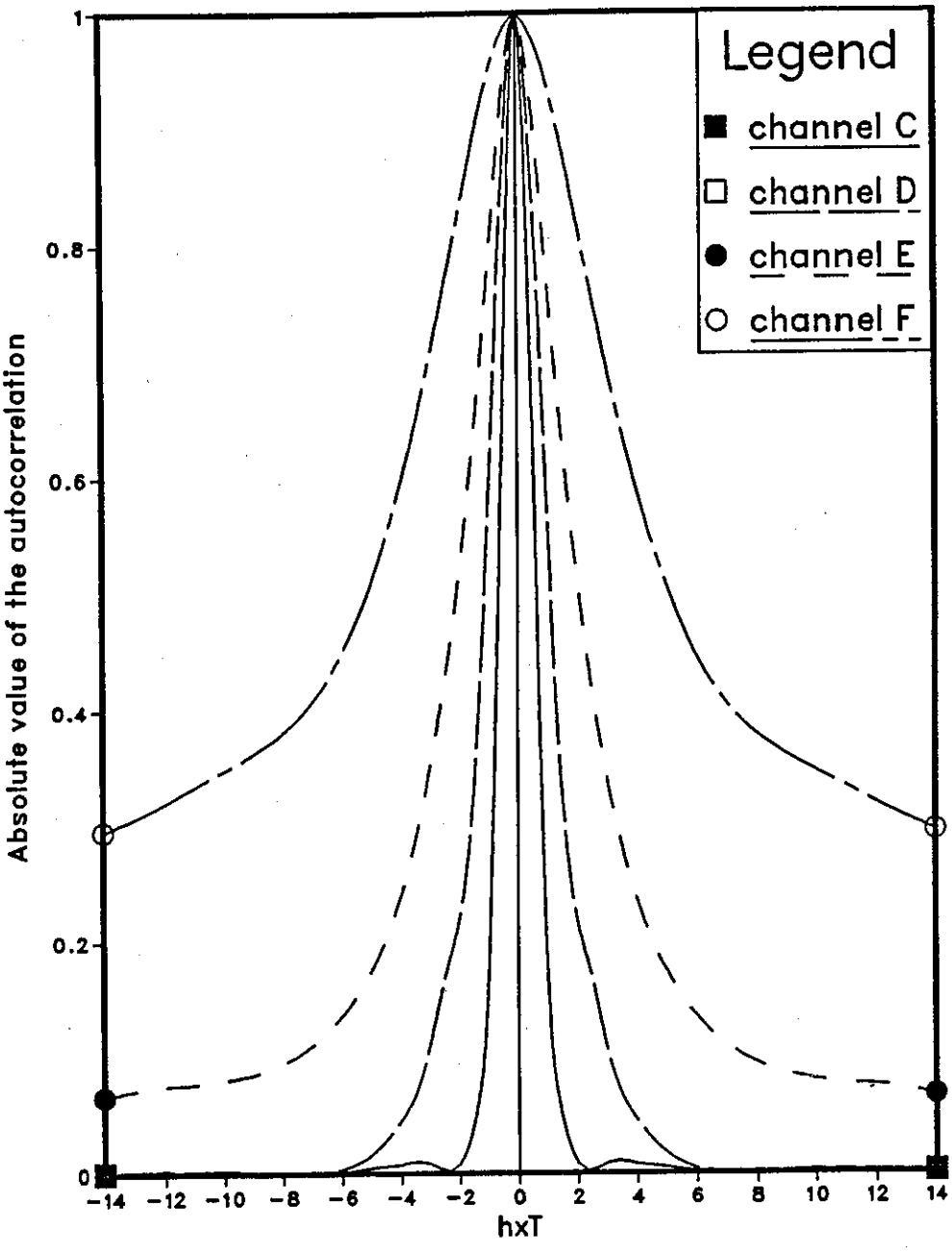


Fig.6.2 The autocorrelation function of the noise samples at the output of the linear equalizer

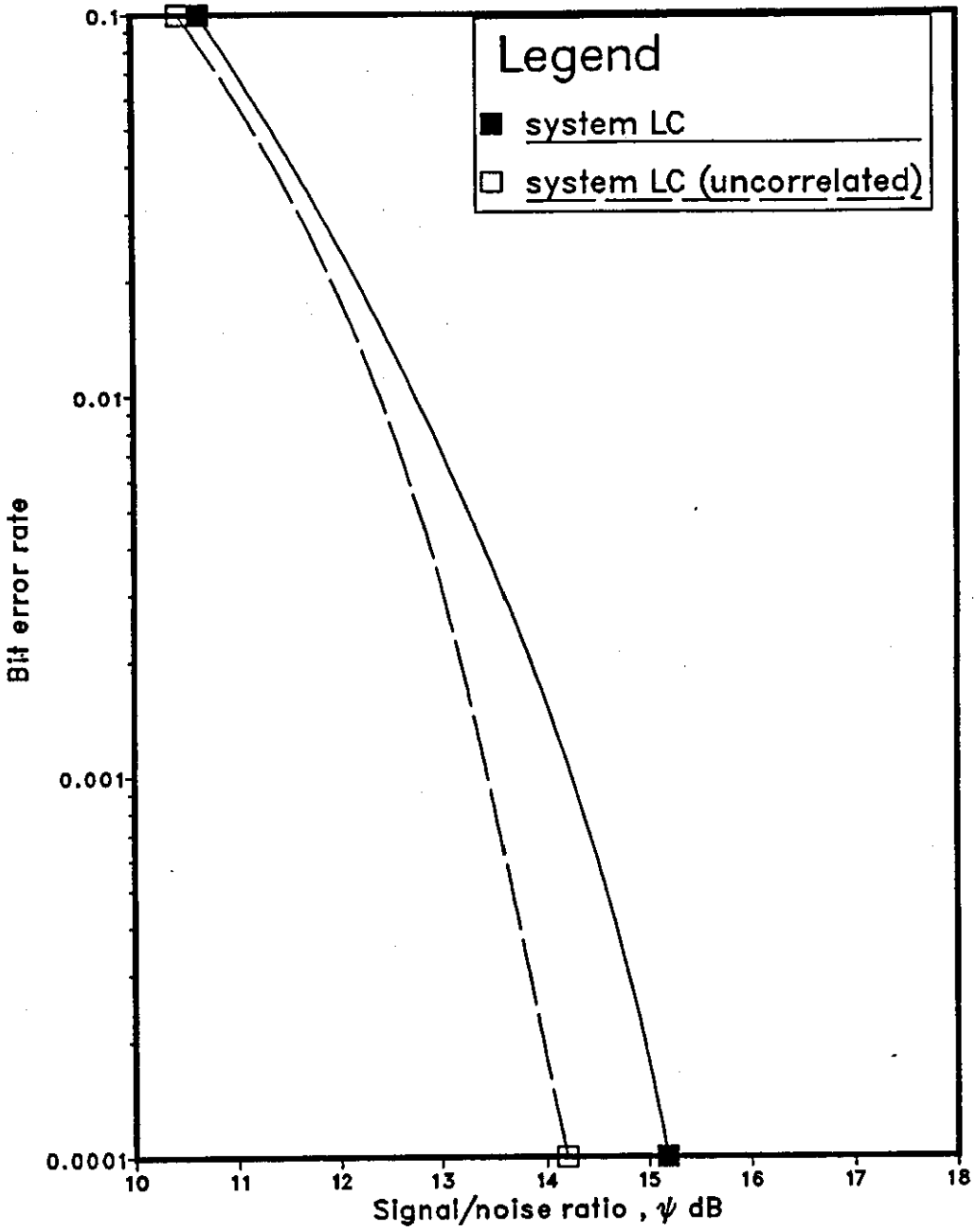


Fig6.3 Performance of system LC over channel C .

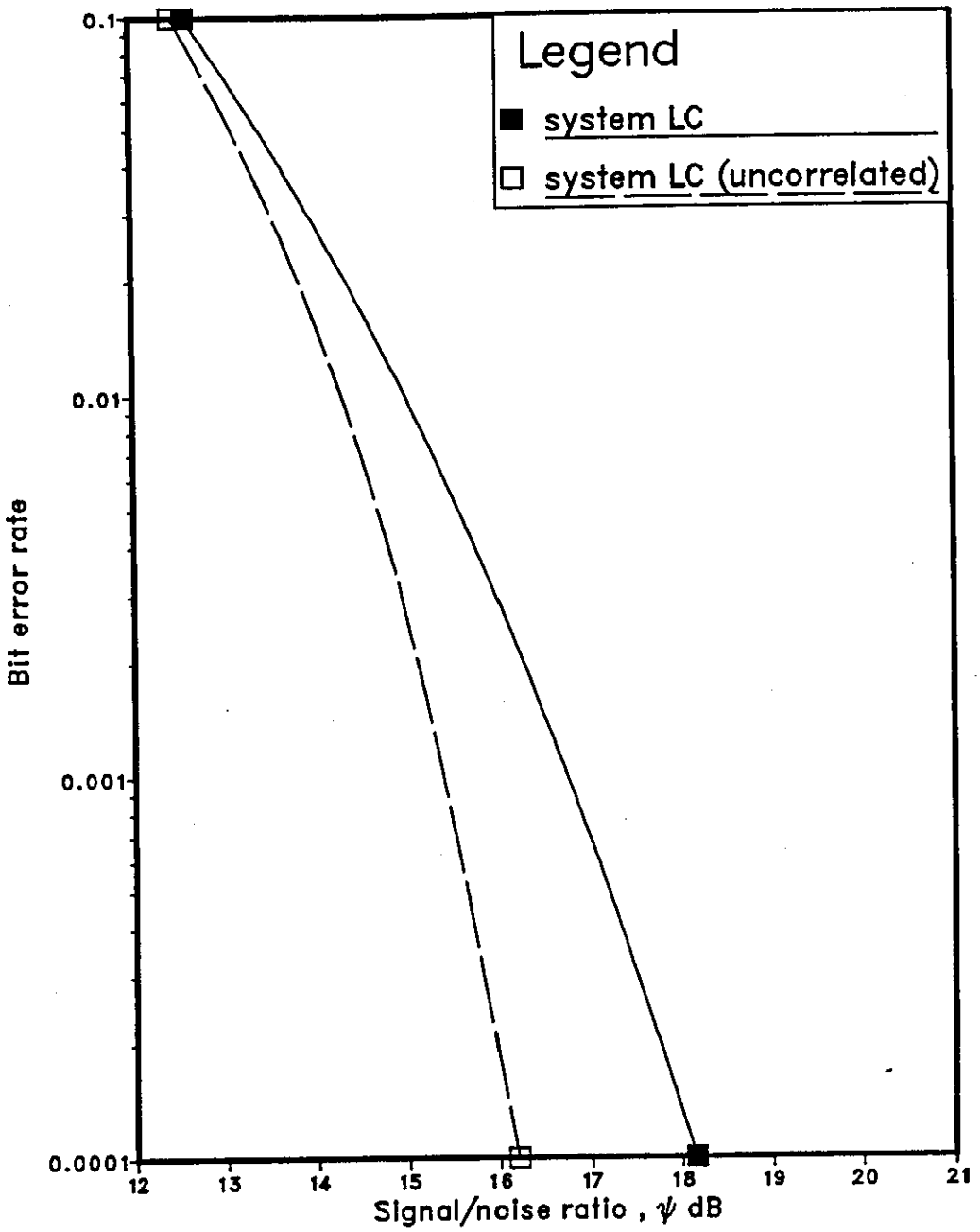


Fig6.4 Performance of system LC over channel D .

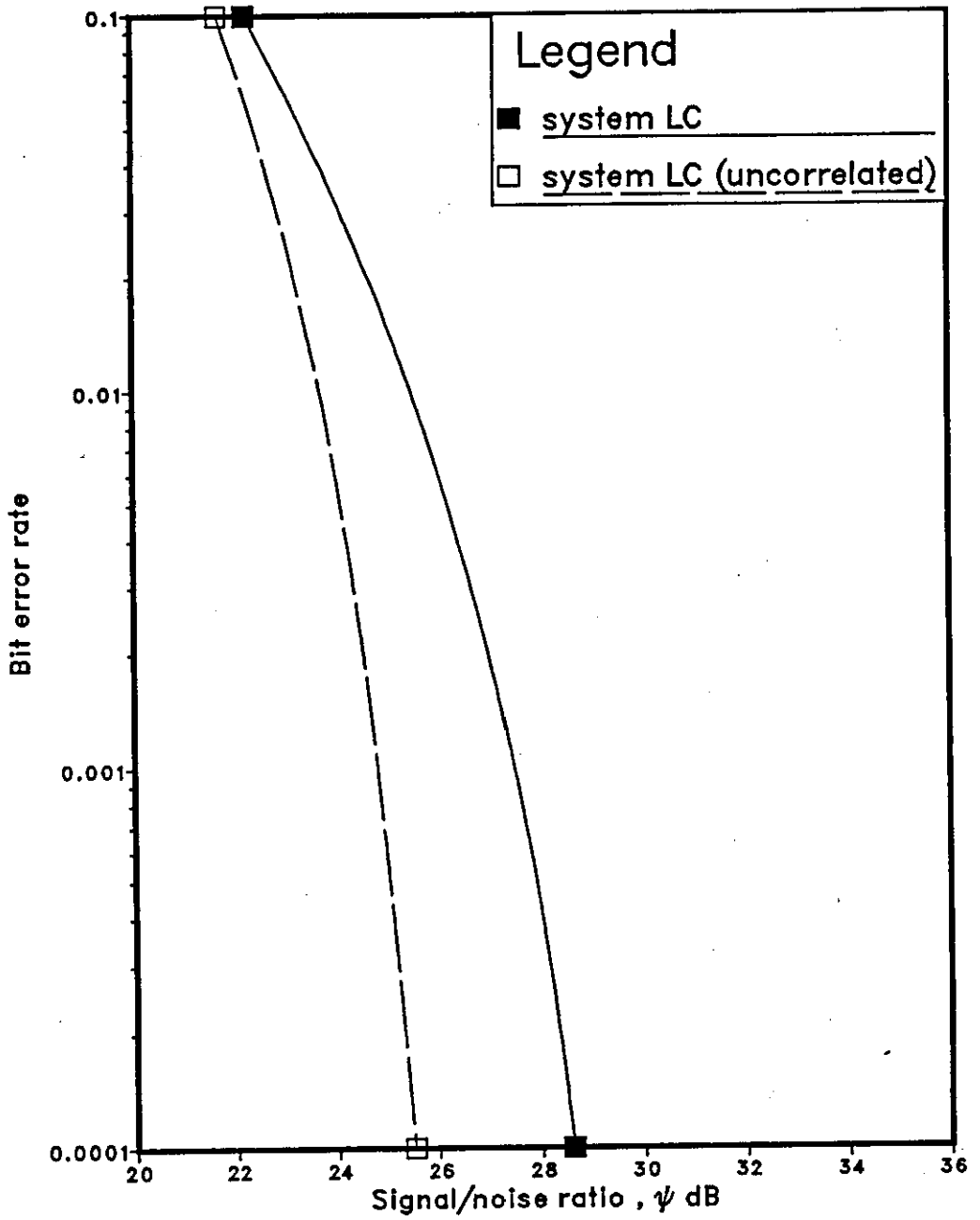


Fig6.5 Performance of system LC over channel E .

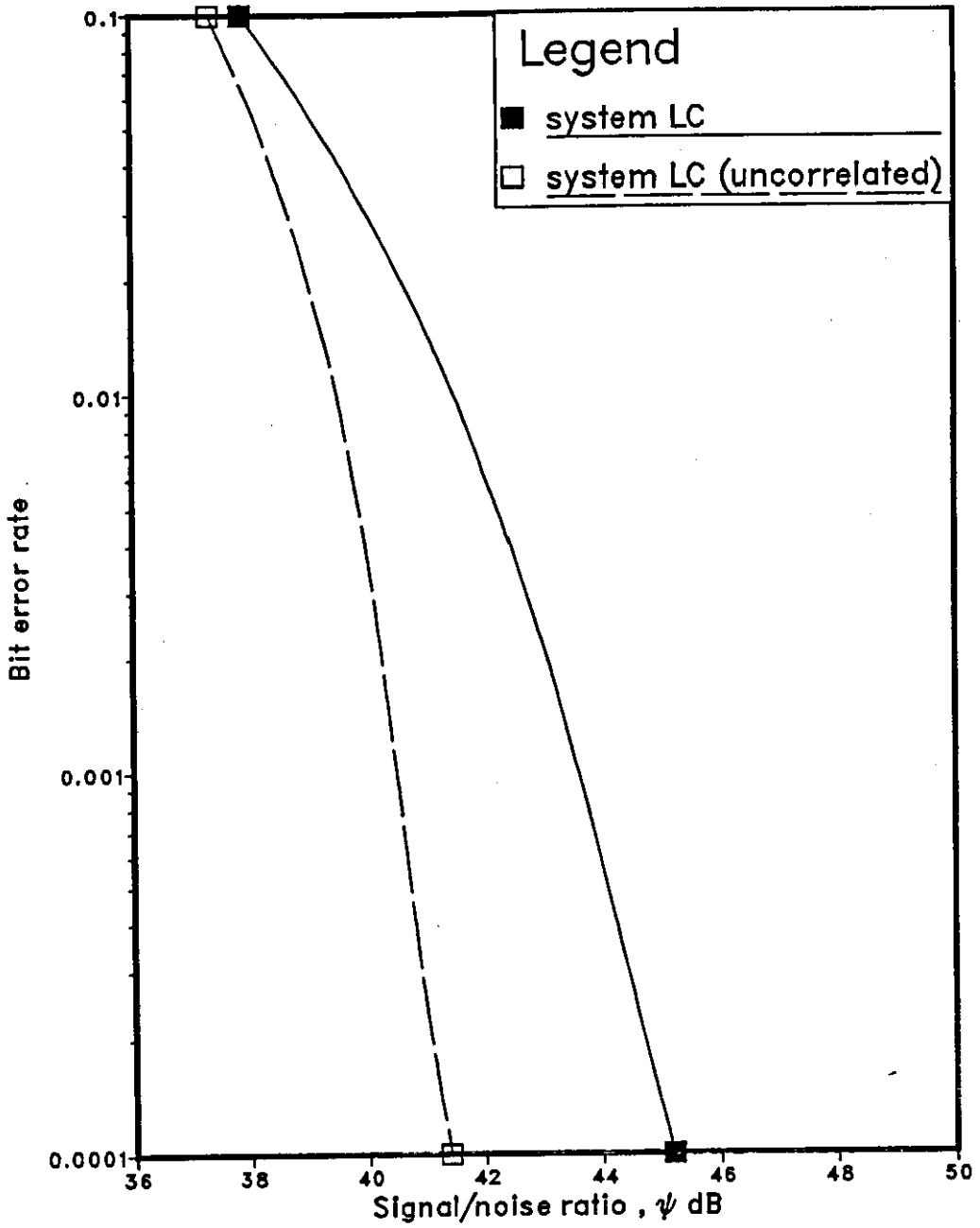
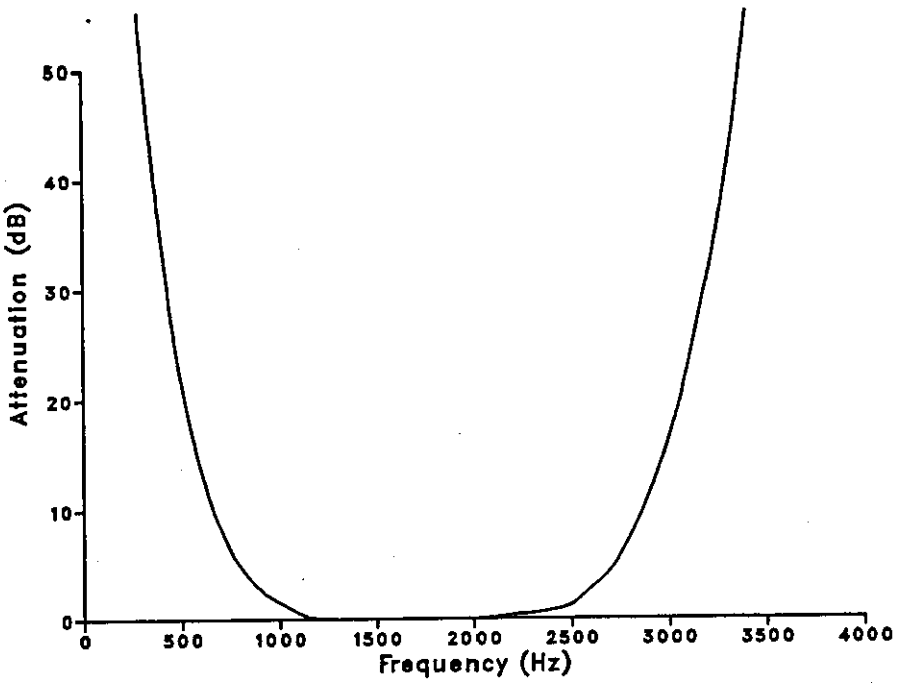
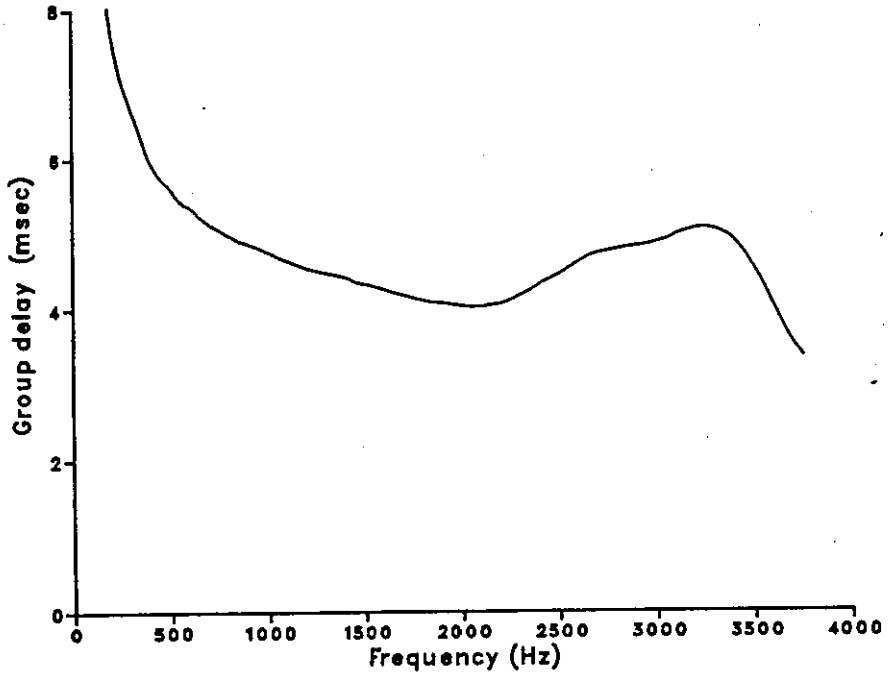


Fig6.6 Performance of system LC over channel F .



a – Attenuation characteristic



b – Group delay characteristic

Fig.6.7 Attenuation and group delay characteristics of the equipment filters–2

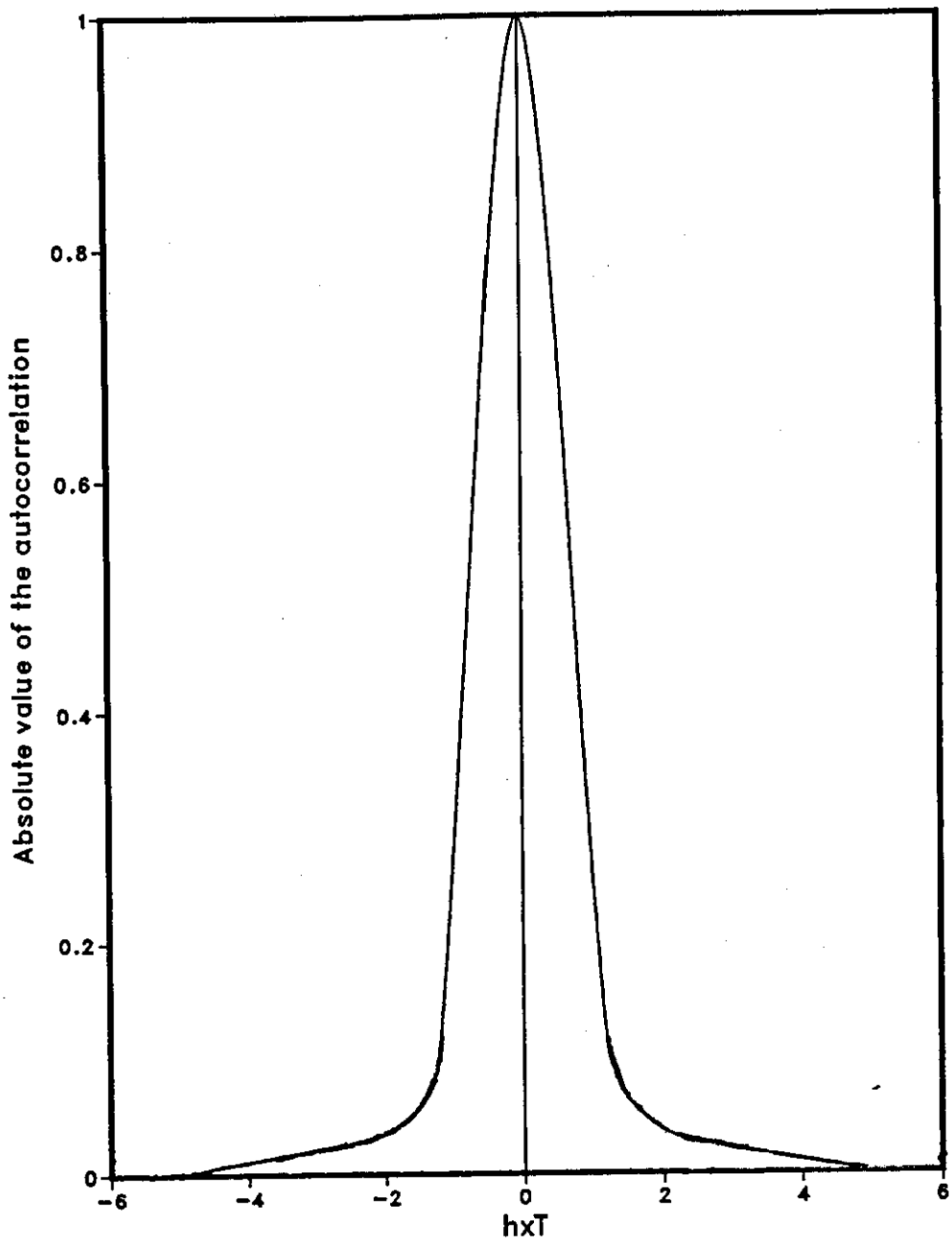


Fig.6.8 The autocorrelation function of the noise samples at the output of the receiver filter

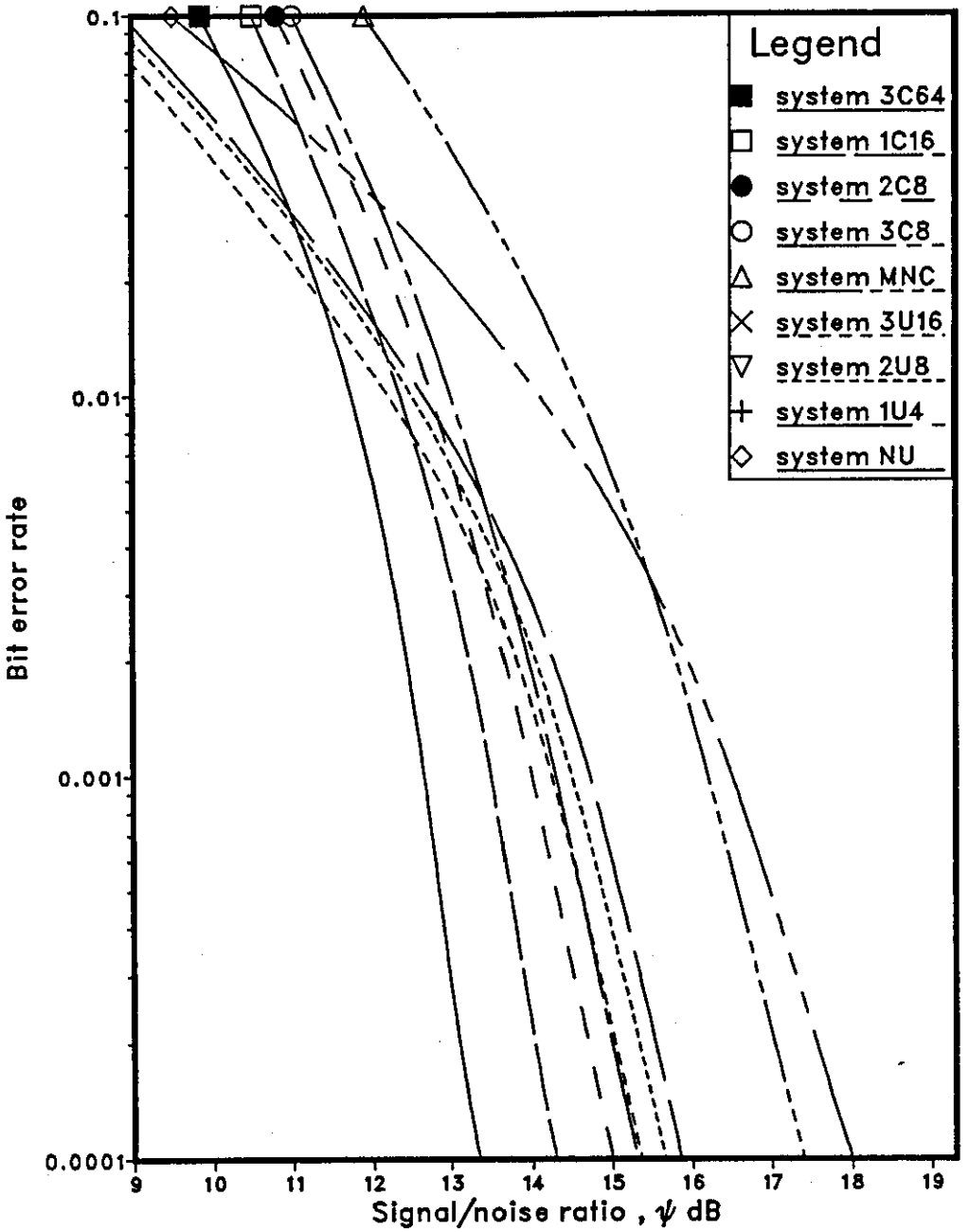


Fig.6.9. Performance of various systems in the back-to-back tests , where the filtering is equally shared between the transmitter and receiver filters .

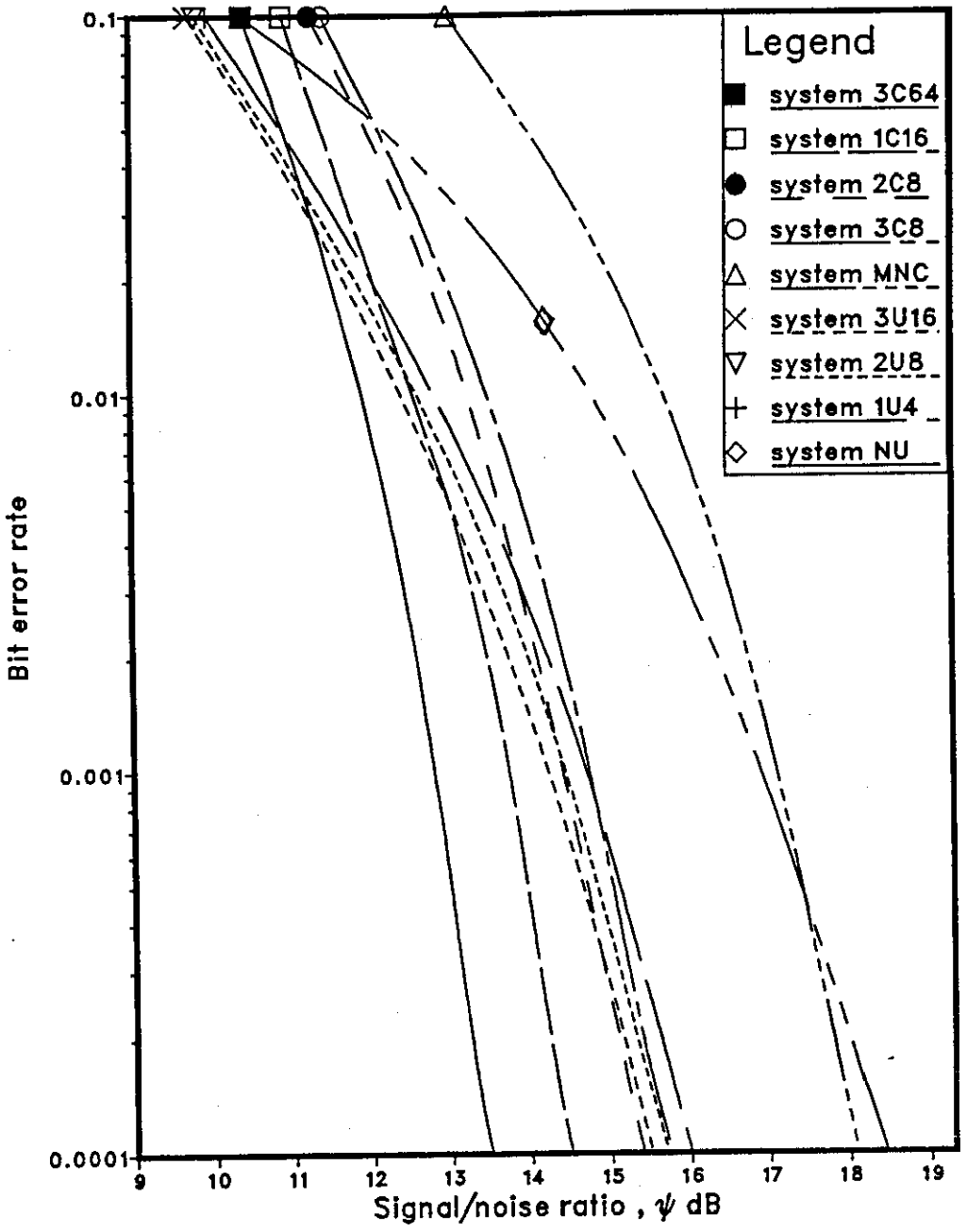


Fig.6.10. Performance of various systems in the back-to-back tests , where the receiver filter introduces no correlation between the neighbouring noise components

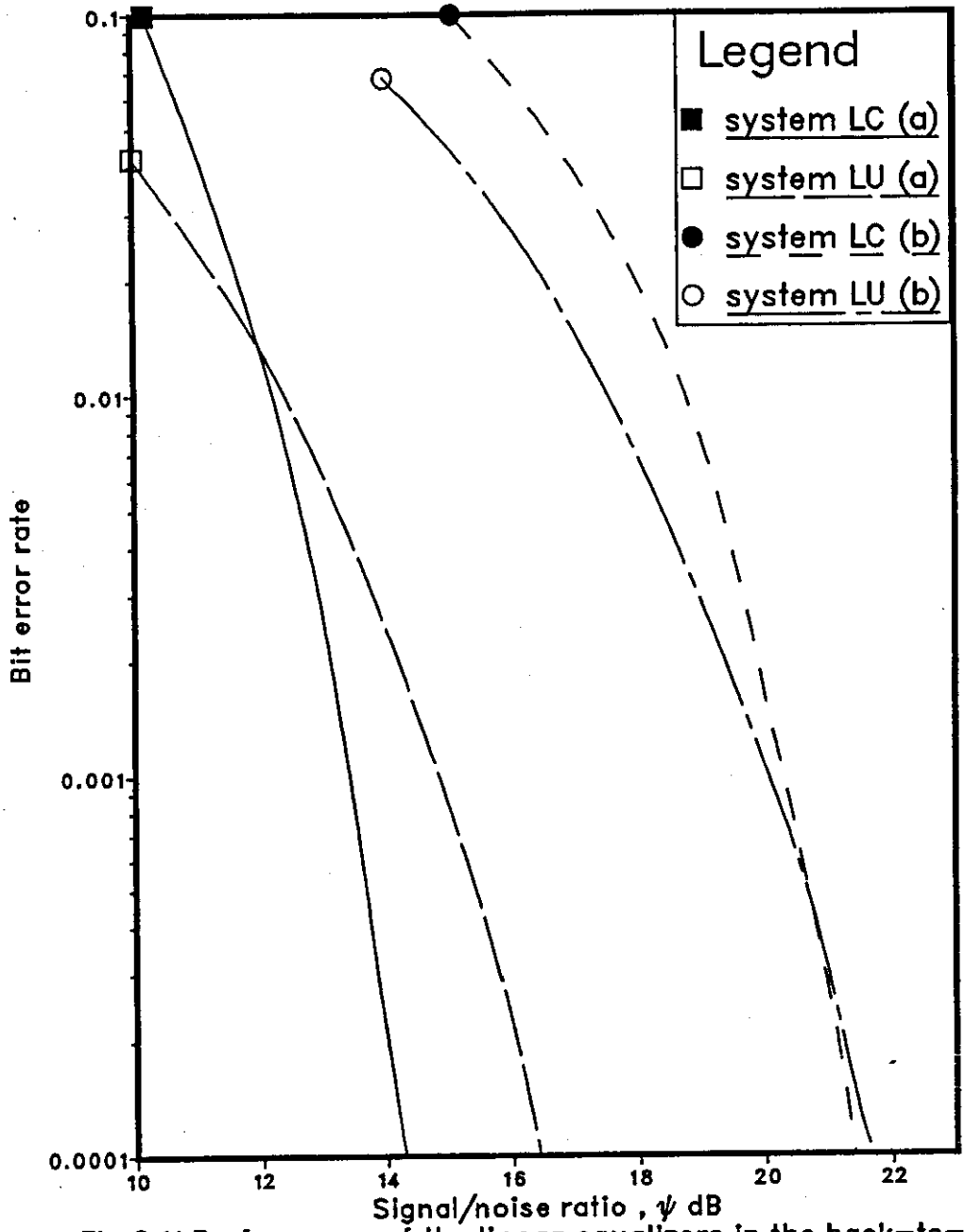


Fig.6.11 Performance of the linear equalizers in the back-to-back tests

a. The filtering is equally shared between the transmitter and receiver filters
b. The receiver filter introduces no correlation between the neighbouring noise components

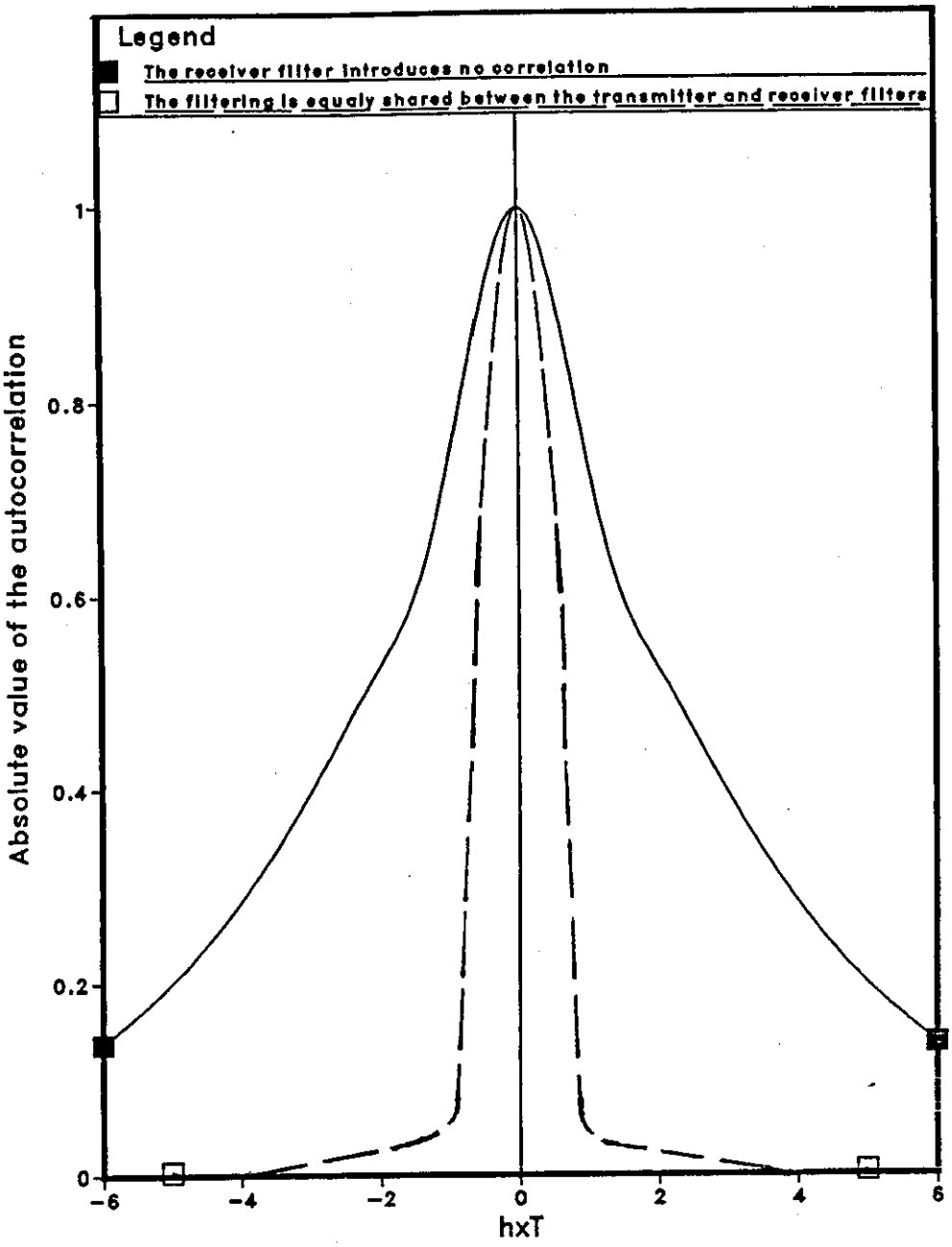


Fig.6.12 The autocorrelation function of the noise samples at the output of the linear equalizer in the back-to-back tests

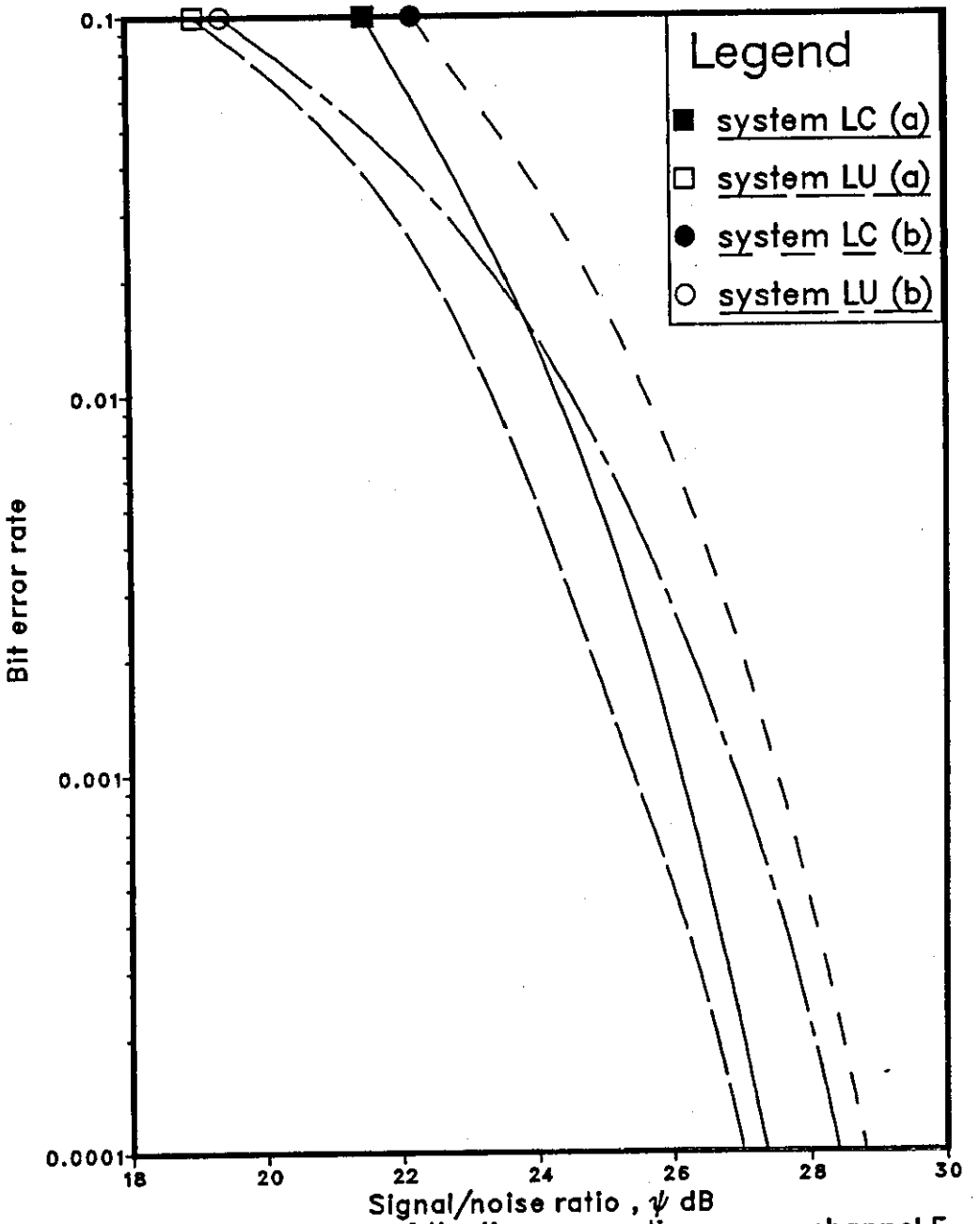


Fig.6.13 Performance of the linear equalizers over channel E (Equipment filters–2 with telephone circuit 3)

a. The filtering is equally shared between the transmitter and receiver filters
b. The receiver filter introduces no correlation between the neighbouring noise components

CHAPTER 7

SIMPLE INTERLEAVER FOR CONVOLUTIONALLY ENCODED QAM SIGNAL

7.1 INTRODUCTION

It is clear from the results of Chapter 5, that the use of a nonlinear (decision feedback) equalizer together with the Viterbi algorithm detector (VA) for the encoded signal (system NC or MNC) does not appear to be a very promising arrangement. The performance of such a system can be considerably inferior to that of the corresponding arrangement with an uncoded signal over the poorer channels tested here. The main reason behind the poor performances of the above systems (system NC and MNC) when compared to other systems is their sensitivity to the error extension effects. When the detected values of the previously received data symbols are incorrect, they lead to a series of incorrect detectionsof the next received symbols. As described in Chapter 5, the detected values, which are used to cancel the intersymbol interference terms in the received samples, are provided by the threshold detector in system NC and by the early decisions of the Viterbi algorithm detector (VA) in system MNC. So any attempt to improve the performances of such systems should be concentrated towards reducing the error extension effects. This may be achieved by a suitable interleaver/deinterleaver arrangement [19,76,89,90].

An interleaver is a device at the transmitter that rearranges the ordering of a sequence of symbols in a deterministic manner. Associated with an interleaver is a deinterleaver at the receiver that applies the inverse ordering to the received symbols to restore the sequence to its original ordering form. In such an arrangement, the encoded data symbols at the output of the encoder are interleaved prior to transmission and deinterleaved prior to the detection. This may reduce the lengths of error bursts that introduced by the channel and so the errors are distributed more uniformly at the detector input [76]. The above is known as an external interleaver,

while an internal interleaver interleaves the information digits prior to the encoding process and the corresponding detected (decoded) digits at the detector output are then deinterleaved; the latter is not considered here.

Different interleaving systems are described in the published literatures [76,89]. In most of these arrangements the interleaving/deinterleaving process introduces a delay of many symbol intervals. Usually, the number of errors in a burst, which are needed to cause an error in the interleaved system, increase with the delay caused by interleaving [89]. Unfortunately, the direct application of any of these arrangements to system MNC may not lead to a successful operation of the detector, due to the fact that system MNC employs the early decisions of the Viterbi algorithm detector (VA) to cancel the intersymbol interference terms in the received samples, and no delay is tolerable in these decisions. The preferred arrangement of the interleaver/deinterleaver here, is the one which is suitable for the system MNC, since the performance of this system is better than that of system NC, over all channels considered here (Chapter 5).

7.2 MODEL OF DATA TRANSMISSION SYSTEM

Fig. 7.1 shows a simplified model of data transmission system. The interleaver and deinterleaver consist of commutator switches. The information to be transmitted is carried by the binary digits $\{\alpha_{i,k}\}$. Each encoder at the transmitter operates exactly in the same manner as the encoder described in Chapter 3. The linear baseband channel and the function of the adaptive linear filter in Fig. 7.1 are as described in Chapter 3. The received sample at the output of the adaptive linear filter, at time iT, is given by

$$r_i = \sum_{k=0}^{g} s_{i-k}\, y_k + w_i \qquad \ldots \quad 7.2.1$$

where $\{s_{i-k}\}$ are the encoded data symbols, the real and imaginary components of the noise samples $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and fixed variance σ^2 (Chapter 3 and Appendix D) and the sampled impulse response of the linear baseband channel and the adaptive linear filter is given by the (g+1)-component vector

$$Y = [\, y_0 \quad y_1 \quad y_2 \quad . \quad . \quad . \quad y_g \,] \qquad \ldots \quad 7.2.2$$

where $y_0 = 1$ as described in Chapter 3.

The commutator switches A, B, C and D in Fig. 7.1 are assumed to be perfectly synchronized, so that when switch A, for example, is at the jth position, all the remaining switches (B, C and D) are also at the jth position. The arrangement in Fig. 7.1 uses N encoders at the transmitter and N decoders at the receiver. N here represents the interleaving depth. Each encoder (or decoder) operates every N symbol intervals. So, when the binary digits $\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}$ and $\alpha_{i,4}$,at time iT, are fed to the first encoder, the latter produces the encoded data symbol s_i at its output as described in Chapter 3. No signal is fed or produced from this encoder at the time instants (i+1)T , (i+2)T , . . . (i+N-1)T , but at time (i+N)T a new set of four binary information digits are applied to the first encoder. These digits are given by $\alpha_{i+N,1}, \alpha_{i+N,2}, \alpha_{i+N,3}$ and $\alpha_{i+N,4}$, and the corresponding encoded symbol s_{i+N} is then produced at the output of the particular encoder.

When the interleaving arrangement uses system NC as a detection process, as shown in Fig. 7.1, the system is called here "system INC", and when the arrangement uses system MNC it is called "system IMNC", where the detectors and the equalizer in the latter system are shown in Fig. 7.2.

7.3 DETECTION PROCESSES IN THE INTERLEAVING SYSTEMS

The detection processes here ,in principle, are the same as those of systems NC and MNC. The only exception here is that, each detector in Fig. 7.1 or Fig. 7.2 operates every N sampling intervals. This requires a slight modification in the detection process and will be described here.

Assume that the data symbol s_i is produced at the output of the first encoder at time iT. Thus at time (i+j-1)T each of the four commutator switches is at the jth position, where $1 \geq j \geq N$. When the encoded symbol s_{i-j+1} is produced at the output of the jth encoder, the corresponding received sample at the output of the adaptive linear filter ,at time (i+j-1)T, is given by

$$r_{i+j-1} = \sum_{k=0}^{g} s_{i+j-k-1} y_k + w_{i+j-1} \qquad \ldots \quad 7.3.1$$

and the corresponding equalized sample at the jth detector input is given by

$$e_{i+j-1} = r_{i+j-1} - \sum_{h=1}^{g} \hat{s}_{i+j-h-1}\, y_h \qquad \ldots \quad 7.3.2$$

where $\hat{s}_{i+j-h-1}$ is the detected value of the data symbol $s_{i+j-h-1}$ for h=1, 2, . . . g. These values are provided by the threshold detector in system INC and by the Viterbi algorithm detector (VA) in system IMNC. The equalized sample e_{i+j-1} is then processed by the jth detector. Just prior to the receipt of e_{i+j-1} at time (i+j-1)T the jth detector holds in store eight n-component vectors $\{Q^j_{i-N+j-1}\}$ (the superscript j specifies the detector), where

$$Q^j_{i-N+j-1} = [x_{i-nN+j-1}\, x_{i-(n-1)N+j-1} \cdot \quad \cdot \quad \cdot \quad x_{i-2N+j-1}\, x_{i-N+j-1}] \qquad \ldots \quad 7.3.3$$

and $x_{i-nN+j-1}$ takes on any one of the 16 of the 32 possible values of $s_{i-nN+j-1}$, as described in Chapter 5. Each vector $\{Q^j_{i-N+j-1}\}$ is associated with a different one value of the eight states of the jth encoder at time (i-N+j-1)T. And since each encoder and each detector operates every N symbol intervals, these states are also the states at time (i+j-2)T. Associated with each vector $Q^j_{i-N+j-1}$ is stored its cost, which is given by

$$c^j_{i-N+j-1} = \sum_{h=1}^{n} | e_{i-hN+j-1} - x_{i-hN+j-1} |^2 \qquad \ldots \quad 7.3.4$$

On the receipt of r_{i+j-1}, at time (i+j-1)T, each vector $Q^j_{i-N+j-1}$ is expanded into 16 vectors $\{P^j_{i+j-1}\}$

$$P^j_{i+j-1} = [x_{i-nN+j-1}\, x_{i-(n-1)N+j-1} \cdot \quad \cdot \quad \cdot \quad x_{i-N+j-1}\, x_{i+j-1}] \qquad \ldots \quad 7.3.5$$

where the first n-components of each vector P^j_{i+j-1} are given by the original vector $Q^j_{i-N+j-1}$ and the last component x_{i+j-1} takes on its 16 different permitted values. The latter are determined by the state of the particular vector $Q^j_{i-N+j-1}$ at time (i-N+j-1)T. The cost of each of the resulting vectors $\{P^j_{i+j-1}\}$ is given by

$$c^j_{i+j-1} = c^j_{i-N+j-1} + | e_{i+j-1} - x_{i+j-1} |^2 \qquad \ldots \quad 7.3.6$$

where $c^j_{i-N+j-1}$ is the cost of the original vector $Q^j_{i-N+j-1}$.

There are now 16 different vectors $\{P^j_{i+j-1}\}$, for each of the states at time (i+j-1)T and the detector selects, for each state, the one of the 16 associated vectors $\{P^j_{i+j-1}\}$ with the smallest cost c^j_{i+j-1}, the remaining vectors being discarded. The detected value $\hat{s}_{i-nN+j-1}$ of the data symbol $s_{i-nN+j-1}$ is next given by the value of $x_{i-nN+j-1}$ of the

vector P^j_{i+j-1} with the smallest cost. The detected value $\hat{s}_{i-nN+j-1}$ in turn determines the detected values of the binary digits $\alpha_{i-nN+j-1,1}$, $\alpha_{i-nN+j-1,2}$, $\alpha_{i-nN+j-1,3}$ and $\alpha_{i-nN+j-1,4}$ as described in Chapter 5. Since $s_{i-nN+j-1}$ is detected here after the receipt of r_{i+j-1} there is a delay of nN symbol intervals in the detection.

The first component $x_{i-nN+j-1}$ of each of the eight vectors P^j_{i+j-1} is then omitted, to give the corresponding vector Q^j_{i+j-1}, without changing its cost. The smallest cost is then subtracted from the cost of each vector Q^j_{i+j-1}, to bring the smallest cost to zero. The eight vectors $\{Q^j_{i+j-1}\}$ together with their costs $\{c^j_{i+j-1}\}$ are then stored. As in the original systems (system NC and MNC) the detected value of the data symbol s_{i+j-1} is either given by the threshold detector in system INC or by the jth detector as is the case in system IMNC.

7.4 COMPUTER SIMULATION TESTS AND ASSESSMENT OF THE SYSTEMS

Computer simulation tests have been carried out on the proposed interleaving systems, when the convolutionally encoded signal is transmitted over channels B to F. These channels are described in Chapter 3. The results of the tests are given in Figs. 7.3 to 7.12, and each curve in these figures is labelled by the corresponding system together with the value of N (the interleaving depth). The systems are tested here with N=4 and 8. The performance of system NC and MNC are also shown in each figure for the purpose of comparison. Systems NC and MNC can be considered as a special case of the interleaving systems where N=1. The curves in Figs. 7.3 to 7.7 show the variation of bit error rate against the signal/noise ratio ψ, and ψ is defined by Eqn. F.6 (λ =20 and m=4) in Appendix F. In Figs. 7.8 to 7.12 the performances of the systems are presented in terms of the variation of the average probability of error event against ψ, to show the capability of the interleaving systems in reducing the number of error bursts. The event error rate is given by the number of error events divided by the total number of the transmitted data symbols $\{s_i\}$. An error event is counted here, when at least 32 successive data symbols are correctly detected followed by any incorrect detection of one or more data symbols. The delay in the detection involved in each detector is n symbol intervals in the interleaving systems, where n is fixed at a value of 32.

As can be seen from Fig. 7.4 or 7.5, the interleaving systems INC and IMNC have a better performance than the corresponding systems NC and MNC, respectively, when operating over channels C or D. Over channel C (Fig. 7.4) an advantage of about 1.8 dB in tolerance to additive white Gaussian noise can be achieved, at bit error rate of 10^{-4} , by system IMNC with N=8 over system MNC, and an advantage of the similar order also achieved here by system INC with N=8 over system NC. The corresponding advantages of system IMNC with N=4 is about 1.4 dB over system MNC and that of system INC (with N=4) over system NC is about 1.3 dB. Fig. 7.5 shows the performances of the systems when operating over channel D, where the advantages gained by interleaving are relatively less than those achieved over channel C. The improvement in the performances of the interleaving systems becomes evident when the variation of the event error rates against the signal/noise ratio is examined as in Figs. 7.9 and 7.10. It can be seen from these figures that the interleaving systems (IMNC and INC) reduces the number of error events when compared to that of the original systems (NC and MNC), since for a given value of ψ the probability of error event is reduced by the process of interleaving. The above figures also show that system IMNC, which is the interleaving system based on the modified nonlinear equalizer with the Viterbi algorithm detector (VA), has a better performance than that of system INC for the same value of N (the interleaving depth).

It can be seen from Figs. 7.3, 7.6 and 7.7 that there is no improvement in the performances of the interleaving systems over the original systems (NC and MNC) when operating over channels B, E and F. A similar behaviour of the systems over these channels can also be seen from Figs. 7.8, 7.11 and 7.12, where the performances of the systems are given in terms of event error rates. Channel E and F include the poorest telephone circuits (circuit 3 and 4 in Chapter 3) considered in this work, and channel B has a relatively high level of signal distortion. In fact the behaviour of the performance of the interleaving systems over these three channels suggest that an improvement may be achieved at error rates below 10^{-4} .

It is clear from the results, that the interleaving systems ,tested here, do not appear to be very promising over channels that introduce severe amplitude distortion as in the case of channels B, E and F. This is due to the fact that the number of errors in a burst is relatively large so that N=8 or less (as is the case here) seems to be not

enough to reduce the number of errors in a burst over such channels. For large values of N, the arrangement of interleaving systems requires a relatively large storage. On the other hand and over channels C and D, the interleaving system IMNC with N=8 comes close to achieving the performance of the near-maximum likelihood detector system 3C with 64 vectors (system 3C64). The latter system has the best performance over all other systems, tested over channel C and D (see Chapter 5).

Although, the interleaving systems require as many encoders and detectors as the interleaving depth (N), they involve the same number of arithmetic operations per received data symbol as that of system NC or MNC .This is due to the fact that there is only one encoder (and the corresponding detector) in operation at any one time.

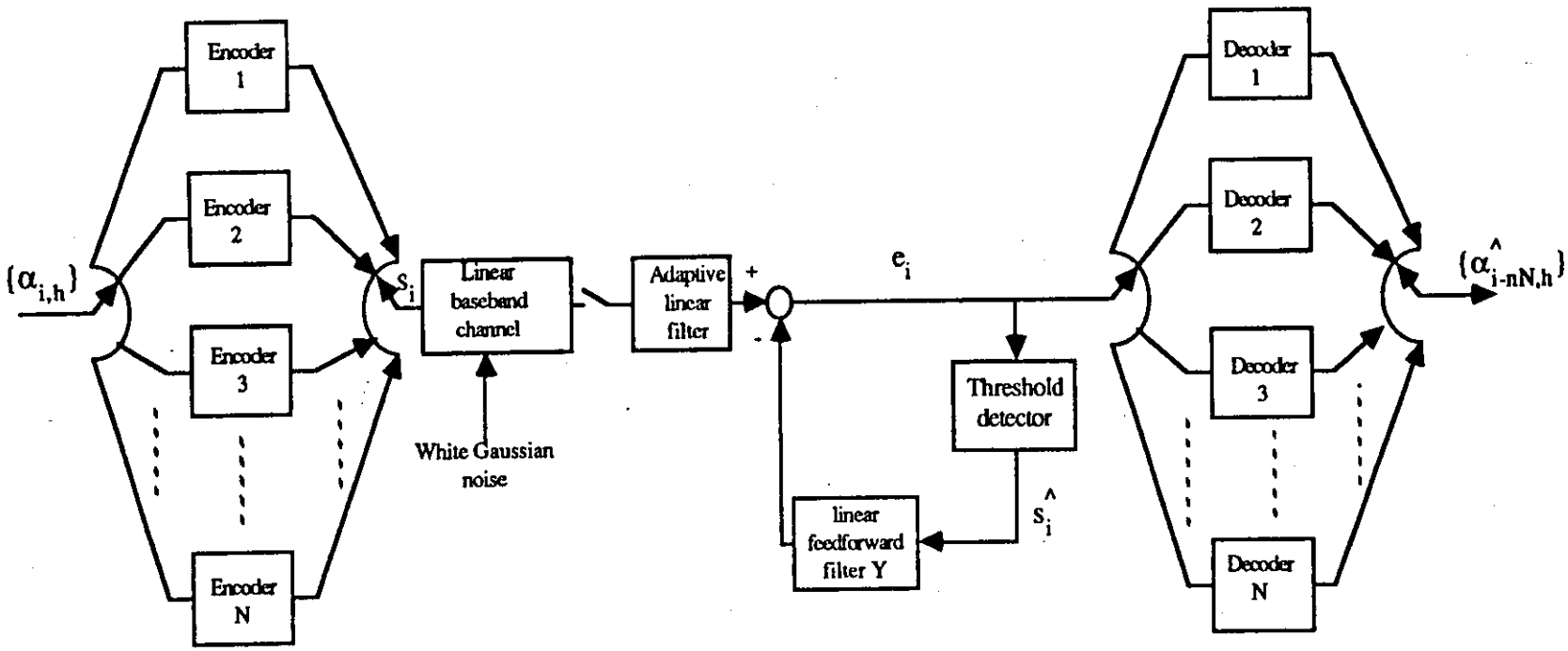


Fig.7.1 Model of interleaving system (system INC) .

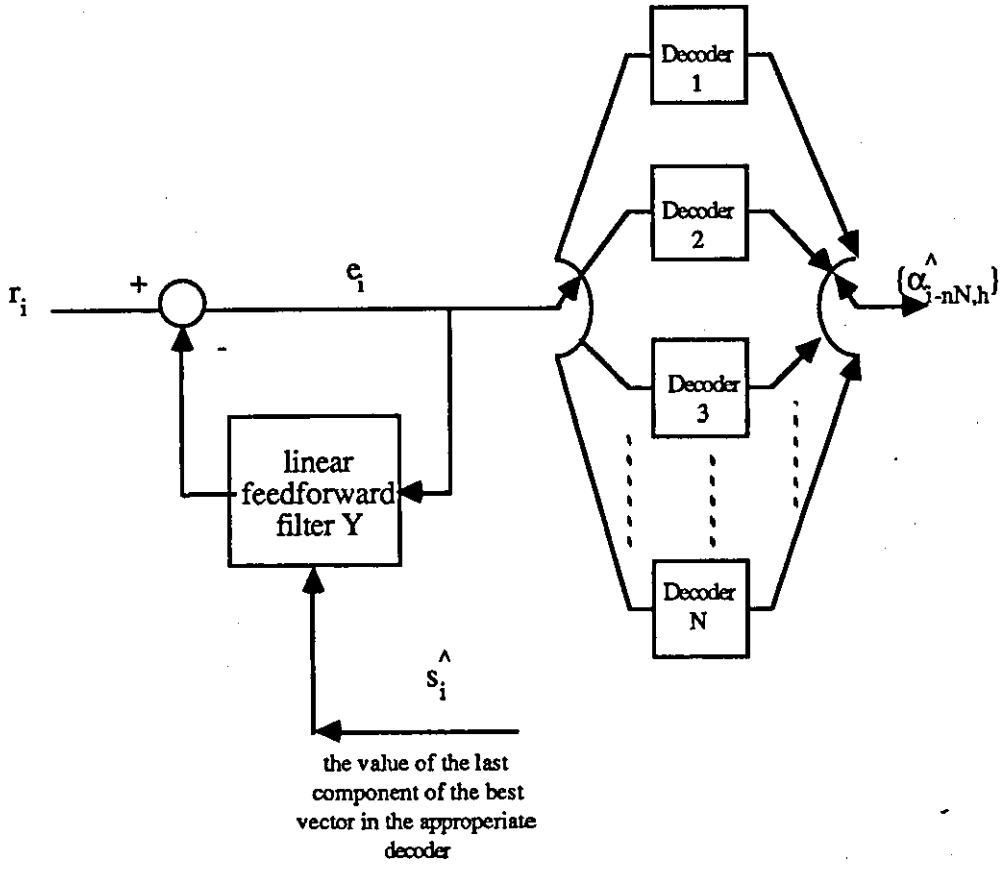


Fig.7.2 The arrangement of the nonlinear equalizer and decoders in system IMNC

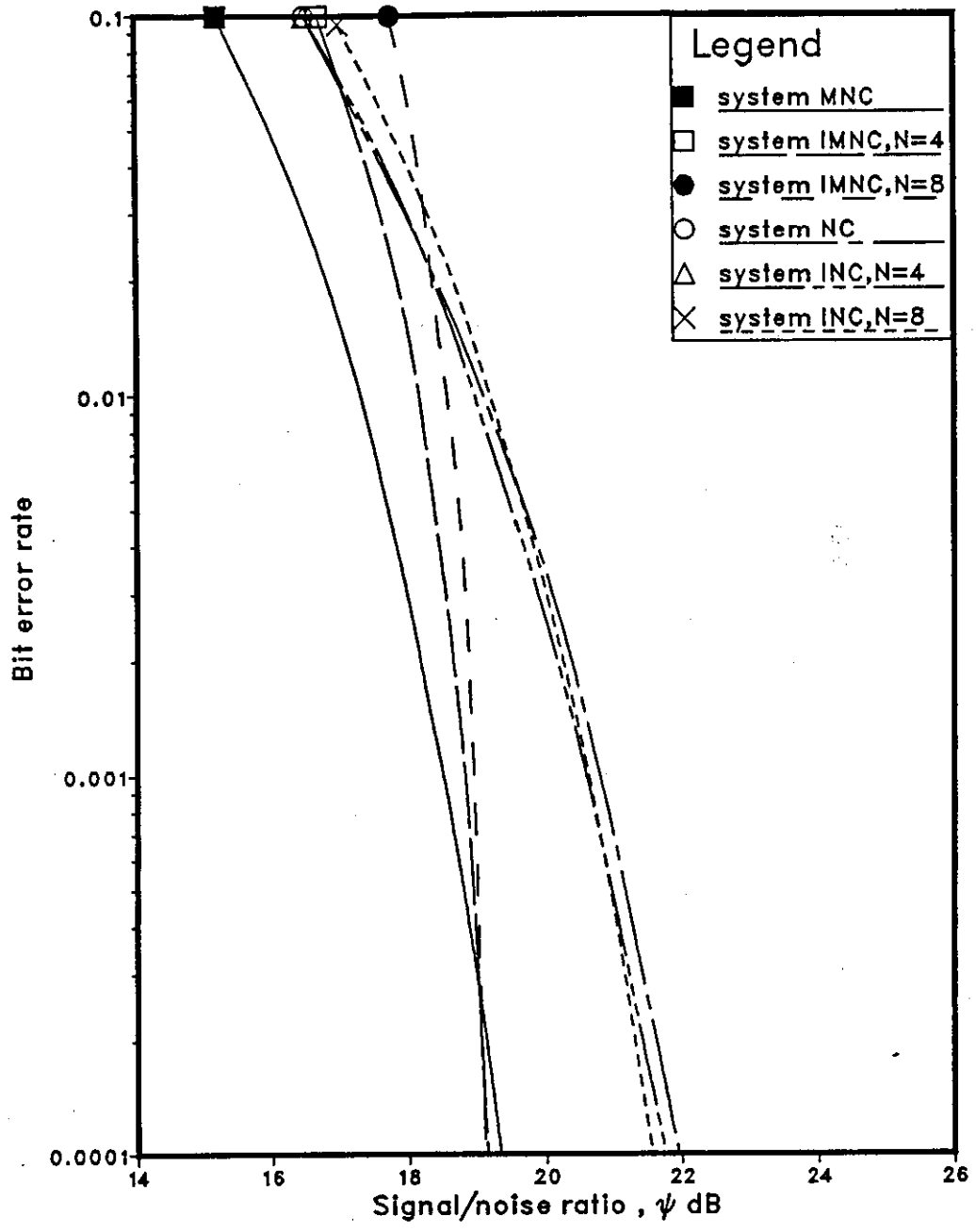


Fig.7.3 Performance of various systems over channel B.

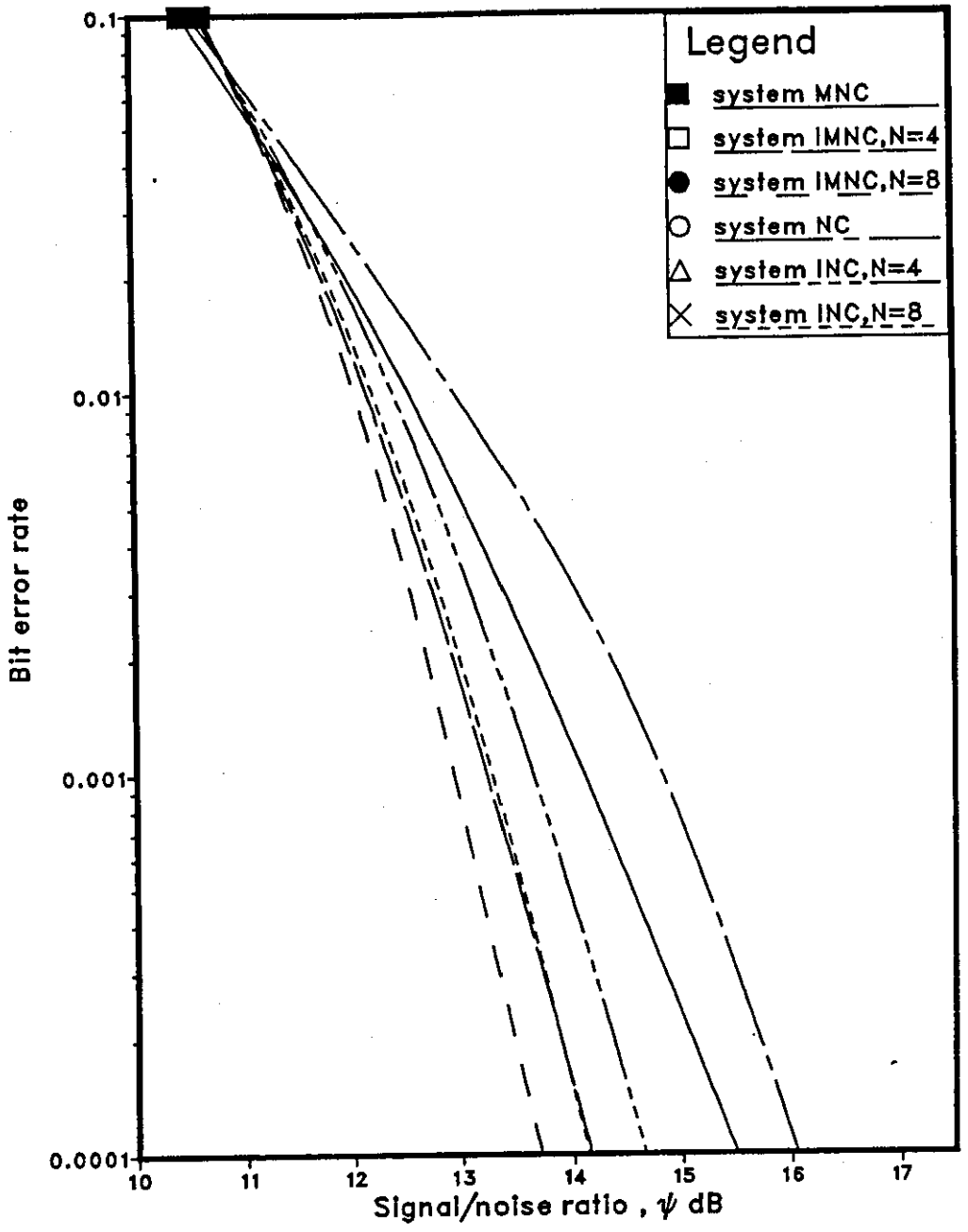


Fig.7.4 Performance of various systems over channel C.

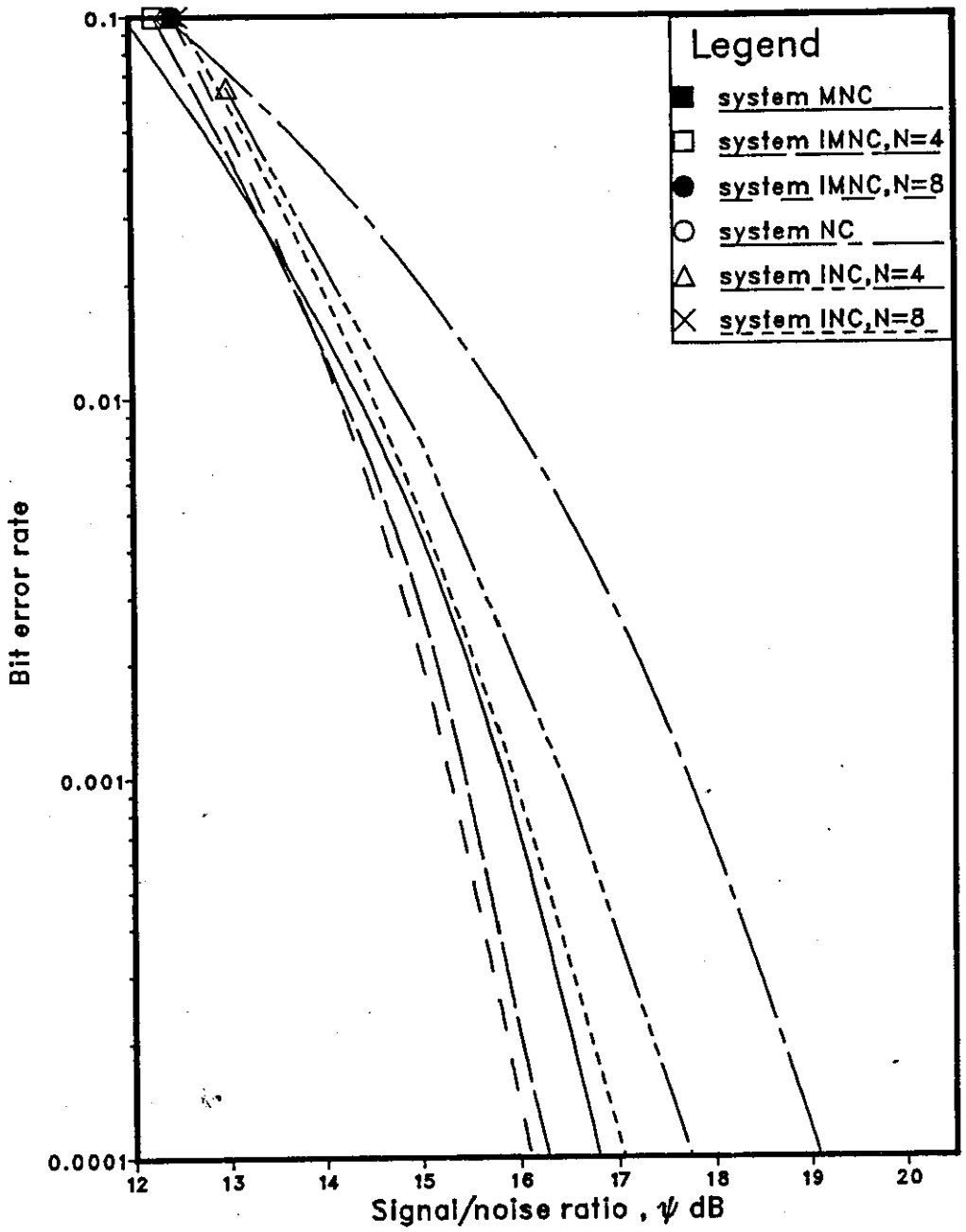


Fig.7.5 Performance of various systems over channel D.

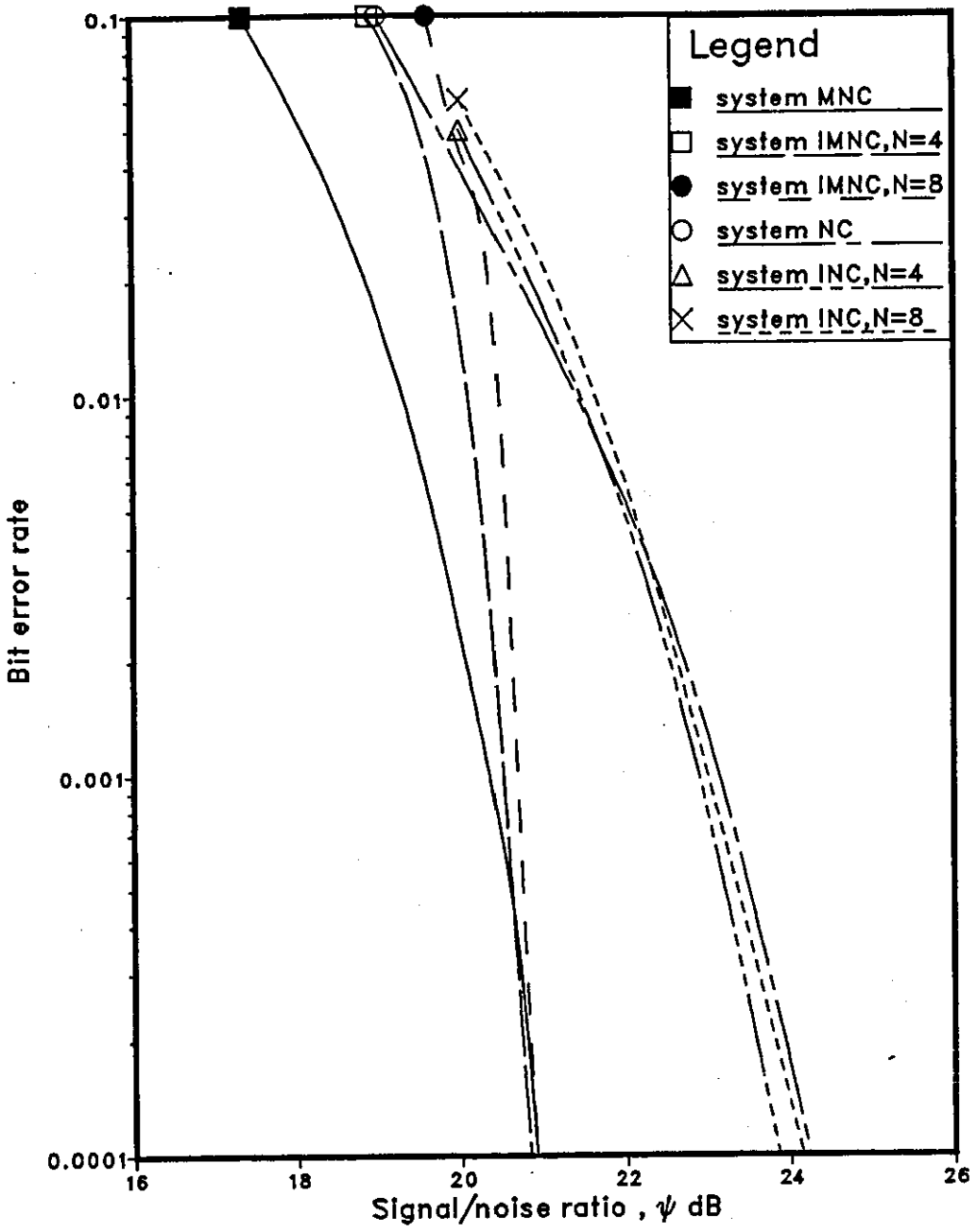


Fig.7.6 Performance of various systems over channel E.

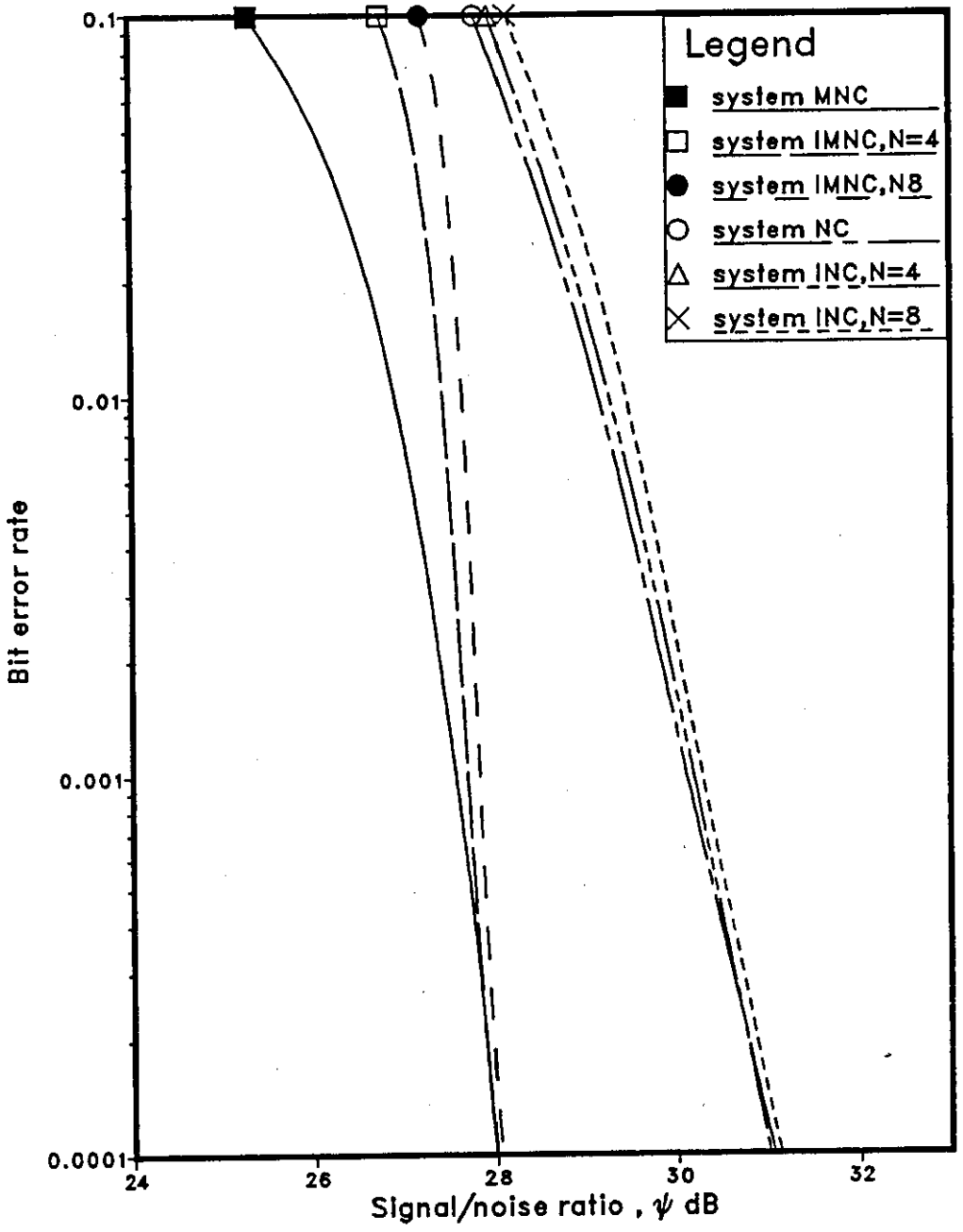


Fig.7.7 Performance of various systems over channel F.

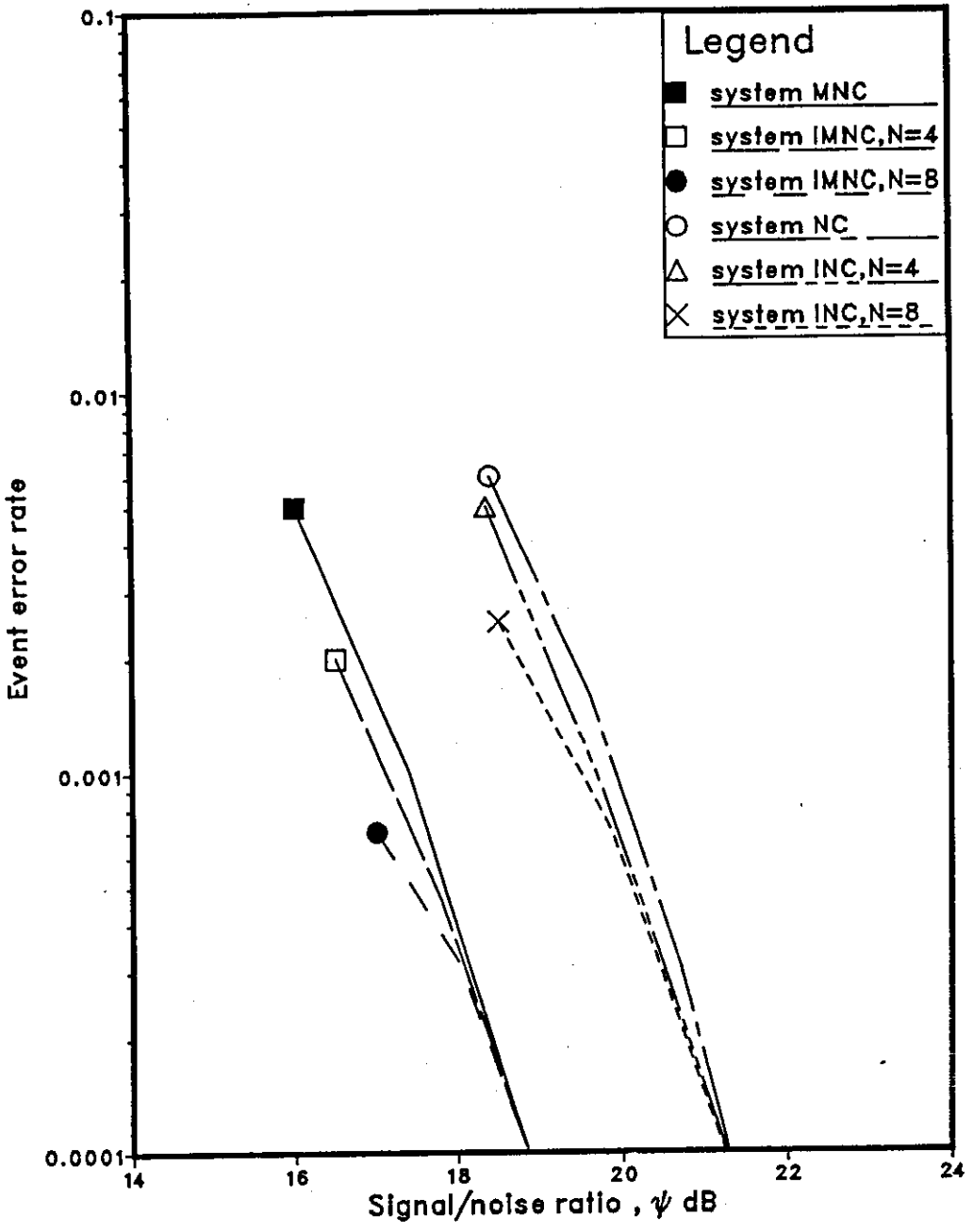


Fig.7.8 Performance of various systems over channel B.

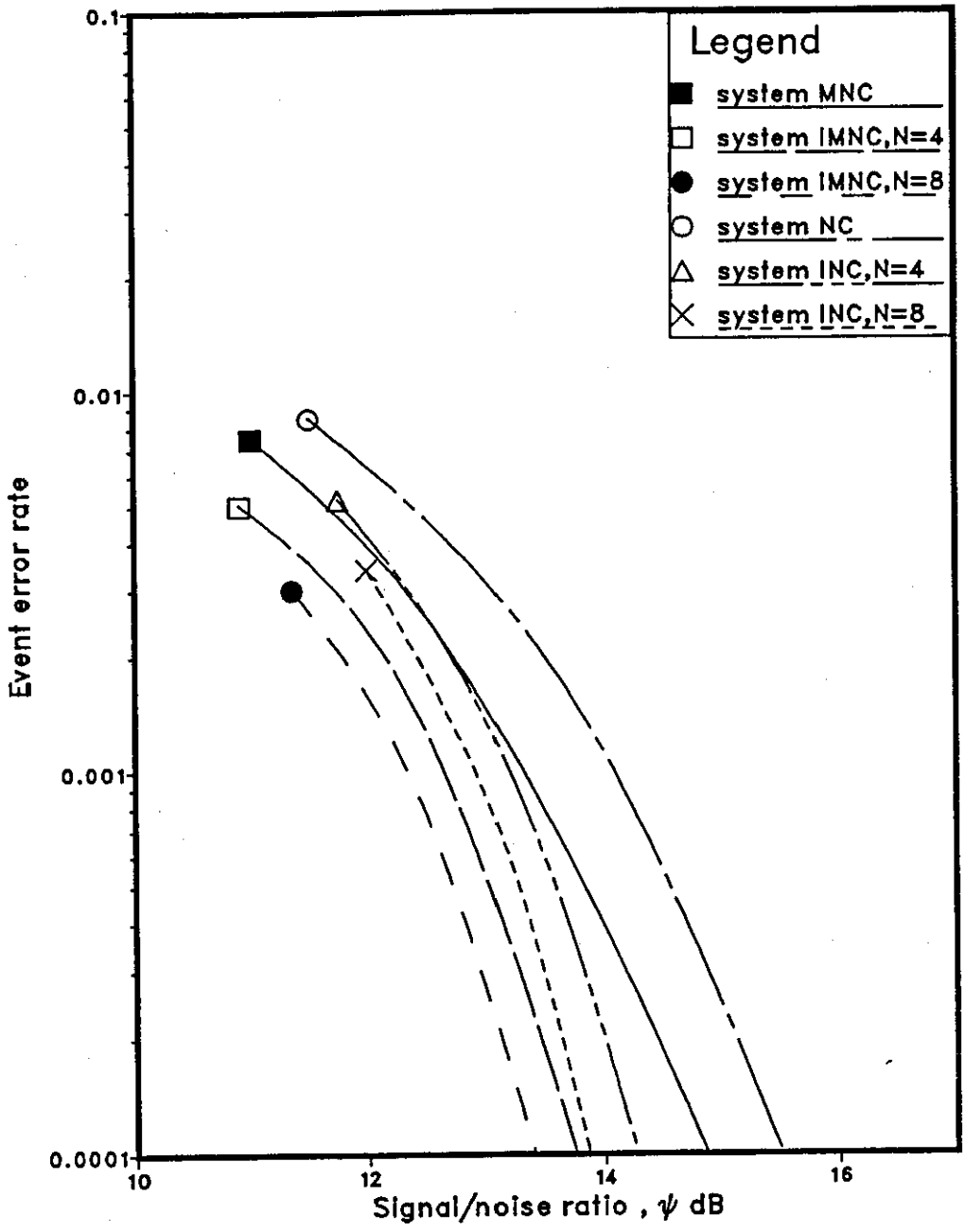


Fig.7.9 Performance of various systems over channel C.

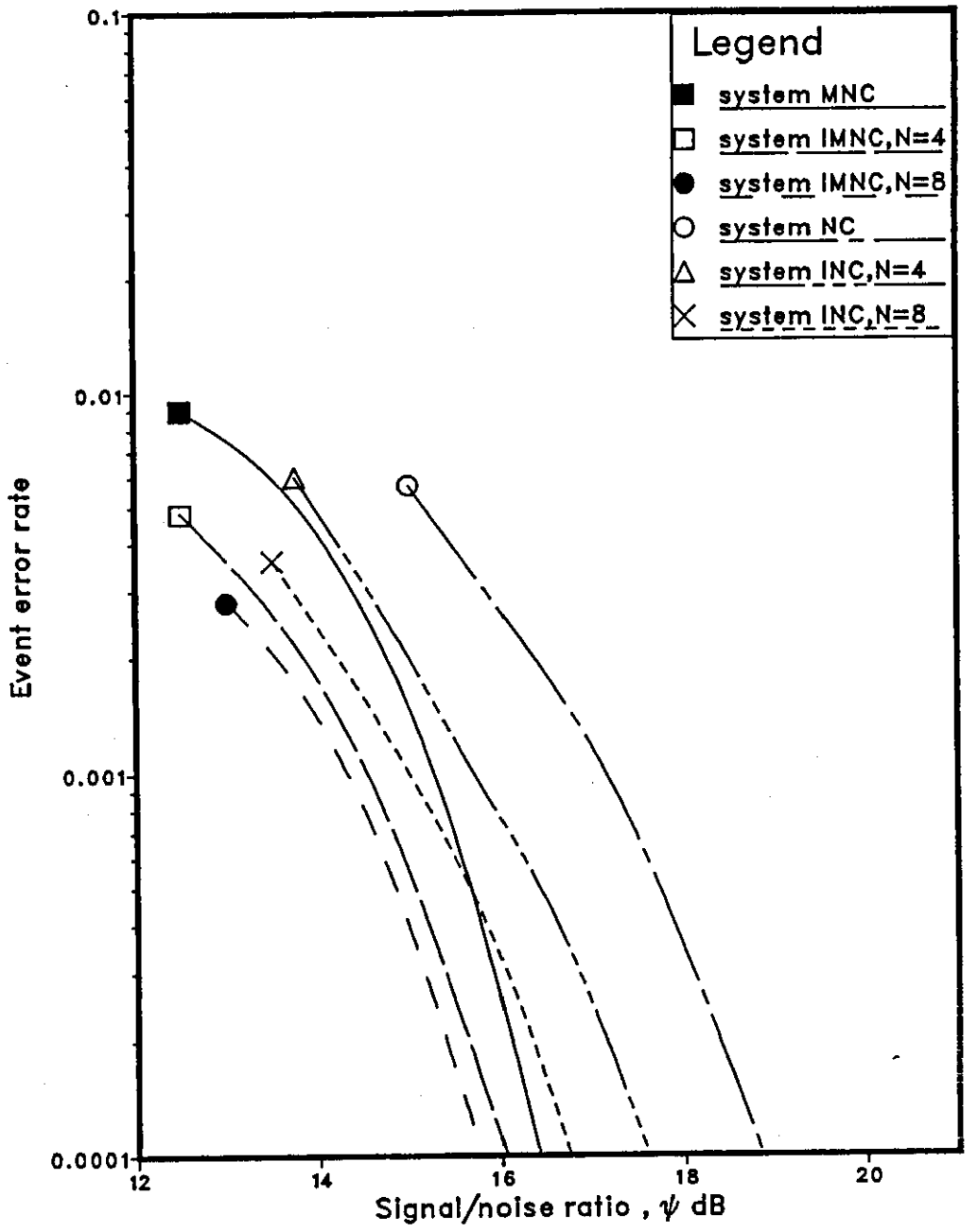


Fig.7.10 Performance of various systems over channel D.

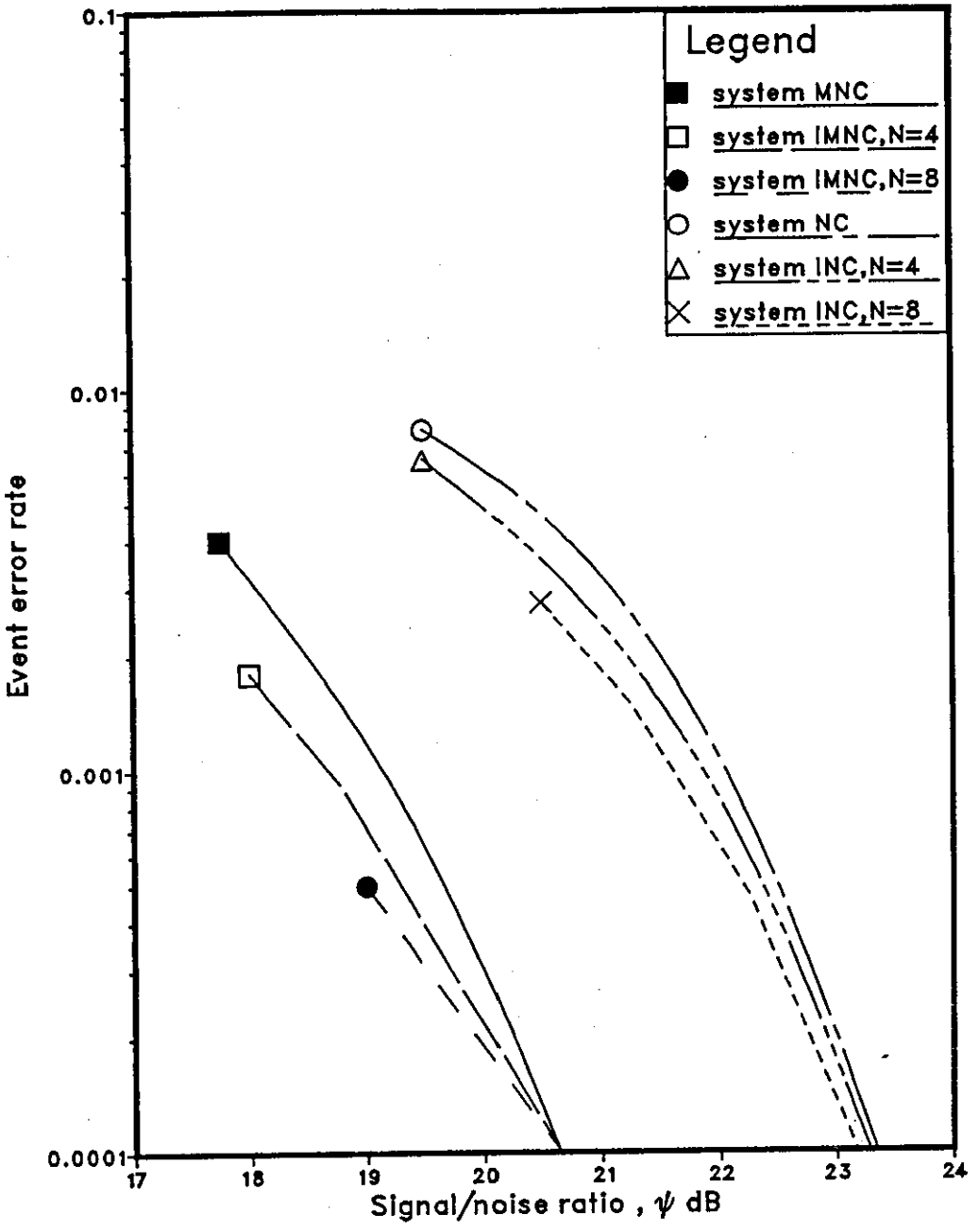


Fig.7.11 Performance of various systems over channel E.

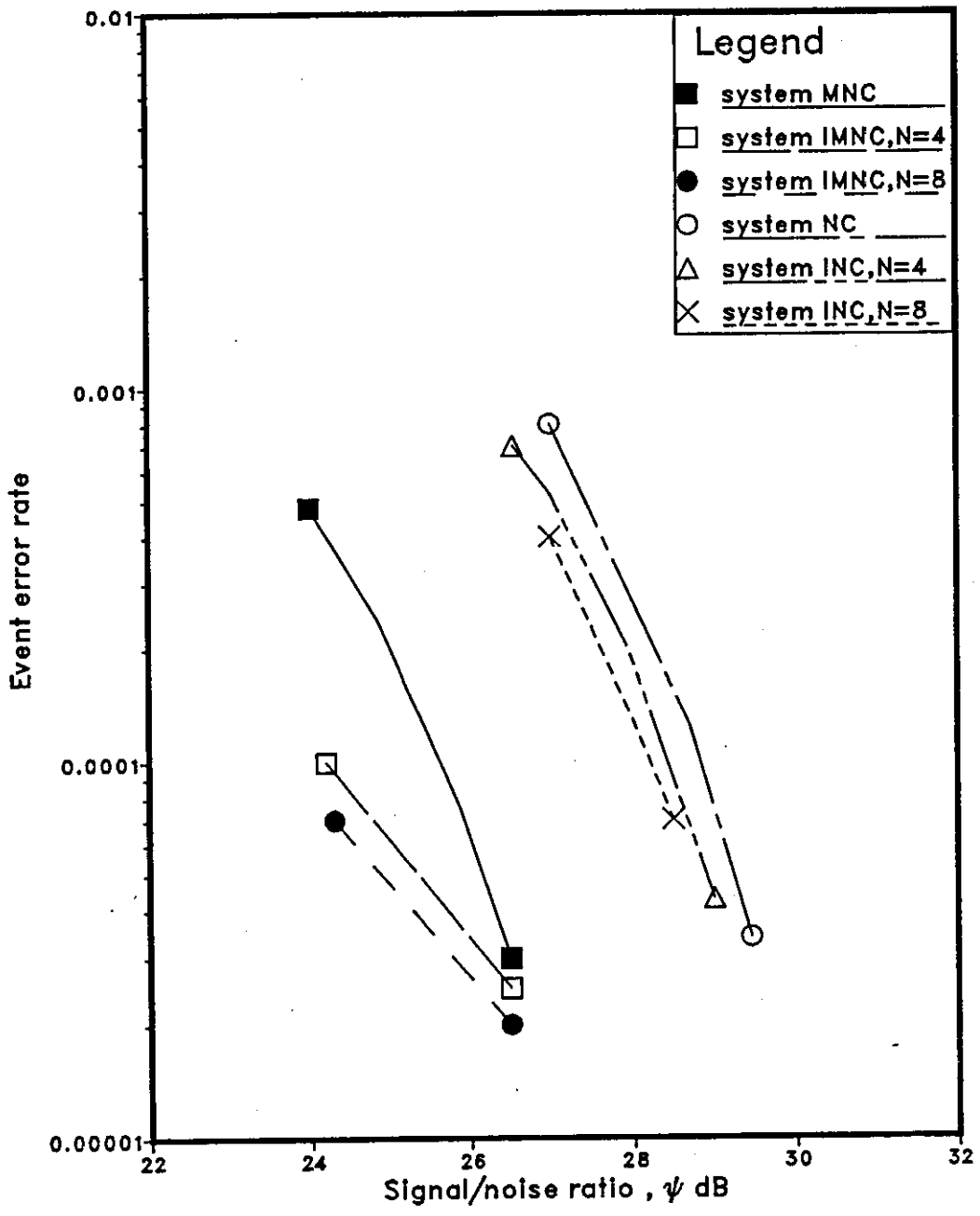


Fig.7.12 Performance of various systems over channel F.

CHAPTER 8

DETECTION OF QAM SIGNALS FOR TRANSMISSION RATES HIGHER THAN 9600 bit/s

8.1 INTRODUCTION

In this chapter, the most promising near-maximum likelihood detectors for coded and uncoded QAM signals described in Chapters 4 and 5, are tested further for application where the data are transmitted at rates of 14400, 16000 and 19200 bit/s. These detectors are known as system 3U16 and 1C16 in Chapters 4 and 5, respectively. The equalizers are also tested here.

An increase of transmission rate is achieved here either by increasing the signal alphabet (the possible values of the transmitted data symbols), so that more information digits can be transmitted per data symbol, or by increasing the modulation rate (number of transmitted signal elements per second). A possible increase in transmission rate can also be achieved by increasing both the number of signal levels and the data symbol rate.

8.2 MODEL OF 14400 bit/s SYSTEM

A model of a synchronous serial data transmission system is shown in Fig. 8.1. The system may operate either with an uncoded 64-level QAM signal or else with a coded 128-level QAM signal. In each case the QAM signal has a carrier frequency of 1800 Hz and an element rate of 2400 bauds, giving a useful transmission rate of 14400 bit/s. The information to be transmitted is carried by the binary digits $\{\alpha_{i,k}\}$, which are statistically independent and equally likely to have either one of their possible values 0 or 1. Furthermore, the information digits $\{\alpha_{i,k}\}$ occur in separate groups of six adjacent digits $\alpha_{i,1}\alpha_{i,2}\ldots\alpha_{i,6}$, which determine the corresponding uncoded or coded symbol s_i. As before, it is assumed that $\{\alpha_{i,k}\}=0$ and $s_i=0$ for $i<0$, so that s_i is the (i+1)th transmitted data symbol.

When s_i is uncoded (not convolutionally encoded), it is derived from the $\{\alpha_{i,h}\}$ by a process of differential encoding. In the differential encoder, the binary digits $\alpha_{i,1}$ and $\alpha_{i,2}$ are recoded to give the corresponding binary digits $\beta_{i,1}$ and $\beta_{i,2}$ according to Table 8.1 (as in Chapter 3), while

$$\beta_{i,h} = \alpha_{i,h} \qquad \text{for } h = 3,4,5,6 \qquad \ldots 8.2.1$$

The resulting group of the six binary digits $\beta_{i,1}\beta_{i,2}\ldots\beta_{i,6}$ now determines the appropriate data symbol s_i according to Fig. 8.2 [42]. As in the case of the 16-level QAM signal in Chapter 3, the first two binary digits in any binary coded number determine the quadrant containing s_i . The remaining four digits determine the position of s_i in the quadrant, the latter digits in any quadrant are the same as those in the all-positive quadrant, if this is rotated to coincide with the given quadrant [14,42,85]. The data symbol s_i is equally likely to take on any one of its 64 possible values given by the signal constellation in Fig. 8.2. Following the detection of s_i , at the receiver, the corresponding detected values of the six binary digits $\beta_{i,1}\beta_{i,2}\ldots\beta_{i,6}$ are determined first from Fig. 8.2. The detected values of $\alpha_{i,1}$ and $\alpha_{i,2}$ are then determined from Table 8.1, using the detected values of $\beta_{i-1,1}$, $\beta_{i-1,2}$, $\beta_{i,1}$ and $\beta_{i,2}$, while the detected values of $\alpha_{i,3}$, $\alpha_{i,4}$, $\alpha_{i,5}$ and $\alpha_{i,6}$ are given by the corresponding detected values of $\beta_{i,3}$, $\beta_{i,4}$, $\beta_{i,5}$ and $\beta_{i,6}$, respectively, as in Eqn. 8.2.1. Any phase rotation of a multiple of $\pi/2$ radians, introduced into the received samples during the transmission, can not lead to an extended sequence of errors in the detected binary digits $\{\hat{\alpha}_{i,h}\}$. This can be verified in the same way as that for the 16-level QAM signal in Appendix B.

When s_i is coded, it is derived from the $\{\alpha_{i,h}\}$ by a process of convolutional encoding. The convolutional encoder used here is the same as that used in Chapter 3 and is shown again here in Fig. 8.3, where, in the case of 14400 bit/s system, m=6, so that the rate of the encoder is 6/7. It can be shown that the asymptotic coding gain in this case is also 4 dB as in the case where the encoder operates at a rate of 4/5 (Chapter 3) [8,68,91]. The encoded symbol s_i has a total of 128 possible values, given by the signal constellation of Fig. 8.4. For each set of six input digits $\alpha_{i,1}\alpha_{i,2}\ldots\alpha_{i,6}$, the resulting seven digits $\beta_{i,0}\beta_{i,1}\ldots\beta_{i,6}$ at the output of the encoder determine the appropriate point in the signal constellation of Fig. 8.4. This gives the encoded symbol s_i , which, for any value of i, can take on any 64 of the 128 possible values in Fig. 8.4, the permitted 64 values belonging to one or other of two disjoint sets that

together include all 128 values. The truth table for the encoder is given in Table 8.2, which is the same as that given in Chapter 3. Each of the eight states of the encoder, at time iT, is determined by the values of $\mu_{i,0}, \mu_{i,1}$ and $\mu_{i,2}$ (Fig. 8.3). The encoder operates on $\alpha_{i,1}$ and $\alpha_{i,2}$ to determine $\beta_{i,0}$ and $\beta_{i,1}$, whereas

$$\beta_{i,h} = \alpha_{i,h} \qquad for \quad h = 2,3,4,5,6 \qquad \ldots 8.2.2$$

Clearly, the values of $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$,that correspond to any particular values of $\alpha_{i,1}$ and $\alpha_{i,2}$, depend upon the state of the encoder. Since the truth table is independent of $\alpha_{i,3}$, $\alpha_{i,4}$, $\alpha_{i,5}$ and $\alpha_{i,6}$, it is evident that, associated with each pair of values of $\alpha_{i,1}$ and $\alpha_{i,2}$, are all sixteen combinations of $\alpha_{i,3}$, $\alpha_{i,4}$, $\alpha_{i,5}$ and $\alpha_{i,6}$, so that every row of the truth table holds for these sixteen combinations. The state at time (i+1)T is uniquely determined by the state at time iT, together with either the values $\alpha_{i,1}$ and $\alpha_{i,2}$ or else by the values $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$. The 16 possible values of s_i (Fig. 8.4), for any given set of values of $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$, are selected according to Ungerboeck's technique of set partitioning [3], in order to maximize the minimum distance between any two of the sixteen $\{s_i\}$, over all sets of values of $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$. This corresponds to the arrangement described Chapter 2 (Section 2.4.2). Fig. 8.5 shows the set partitioning of the encoded signal constellation into eight final states (D0, D1, D2, D3, ... D7), for different numbers of signal levels [9].

The linear baseband channel in Fig. 8.1 is the same as that described in Chapter 3 for the transmission of data at 9600 bit/s. The noise introduced in transmission is stationary white Gaussian noise, which is added to the data signal at the output of the transmission path to give the complex valued noise waveform u(t) at the output of the receiver filter. The resulting waveform at the output of the receiver filter is, therefore the complex valued baseband signal (Chapter 3 and Appendix A)

$$p(t) = \sum_i s_i v(t - iT) + u(t) \qquad \ldots \quad 8.2.3$$

where v(t) is the impulse response of the linear baseband channel, which consists of the baseband signal generator, the linear modulator, the telephone circuit and the linear demodulator.

The waveform p(t) has a bandwidth extending from about -1200 to 1200 Hz and is sampled once per data symbol, at time instants $\{iT\}$, to give the received samples $\{p_i\}$ which are fed to the adaptive linear filter (Fig. 8.1). The sampling rate is 2400

samples/s and is close to the Nyquist rate for p(t). The receiver filter, as in Chapter 3, is such that the real and imaginary components of the noise samples at its output are statistically independent Gaussian random variables with zero mean and fixed variance [20]. The adaptive linear filter, in Fig. 8.1, is an allpass network, with ideally an infinite number of taps, that adjusts the sampled impulse response of the linear baseband channel and filter to be minimum phase, thus concentrating the energy of the sampled impulse response of the channel and filter towards the earlier samples and at the same time removing the phase distortion introduced by the channel other than any that tends to make the sampled impulse response of the channel and filter minimum phase, without however changing any amplitude distortion (Chapter 3 and [22,23]). The noise components $\{w_i\}$ at the output of the linear filter have the same statistical properties as the noise samples at its input (Appendix E). The sampled impulse response of the linear baseband channel and the linear filter in Fig. 8.1 is given by the (g+1)-component row vector

$$Y = [\, y_0 \quad y_1 \quad y_2 \quad . \quad . \quad . \quad y_g \,] \qquad \ldots \quad 8.2.4$$

Where $y_0 = 1$, and Y(z) (the z-transform of Y) have no roots outside the unit circle in the z-plane.

The delay in transmission over the baseband channel and the filter, other than that involved in the time dispersion of the received signal is neglected here and $y_i = 0$ for $i<0$ and $i>g$. Thus the signal at the filter output, at time iT, is given by the complex-valued sample

$$r_i = \sum_{h=0}^{g} s_{i-h}\, y_h + w_i \qquad \ldots \quad 8.2.5$$

where the real and imaginary components of the noise samples $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and fixed variance (Appendix E).

8.3 MODEL OF 16000 bit/s SYSTEM

When the data transmission system operates at a rate of 16000 bit/s, the information digits $\{\alpha_{i,h}\}$ occur in separate groups of five adjacent digits $\alpha_{i,1}\alpha_{i,2}\ldots\alpha_{i,5}$. These digits determine the corresponding uncoded or coded data symbol s_i . The system,

here, may operate either with an uncoded 32-level QAM signal or else with a coded 64-level QAM signal. In each case the QAM signal has a carrier frequency of 1800 Hz and an element rate of 3200 bauds, giving a useful transmission rate of 16000 bit/s.

The uncoded data symbol s_i is derived from $\{\alpha_{i,h}\}$ by a process of differential encoding, where the latter is the same as that described in the previous section (and in Chapter 3). The binary digits $\alpha_{i,1}$ and $\alpha_{i,2}$ are used to determine $\beta_{i,1}$ and $\beta_{i,2}$ (Table 8.1), whereas

$$\beta_{i,h} = \alpha_{i,h} \qquad for \quad h = 3,4,5 \qquad ...8.3.1$$

The resulting group of the five binary digits $\beta_{i,1}, \beta_{i,2}, \beta_{i,3}, \beta_{i,4}$ and $\beta_{i,5}$ now determine the appropriate data symbol s_i according to Fig. 8.6. The mapping of these digits follow the same principle as that used in the previous section and it is different from that used in Chapter 3 for the convolutionally encoded 32-level QAM signal.

When s_i is convolutionally encoded, it is derived from $\{\alpha_{i,h}\}$ by using the encoder in Fig. 8.3, where here m=5 and the coding rate is 5/6. The encoded symbol s_i has a total of 64 possible values given by the signal constellation in Fig. 8.7. For each set of the five input digits, the encoder determines the resulting six output digits which in turn determine the appropriate point in the signal constellation of Fig. 8.7. In the truth table of the encoder (Table 8.2), each of the eight states at time iT is determined by the values of $\mu_{i,0}, \mu_{i,1}$ and $\mu_{i,2}$ (Fig. 8.3) as before. The encoder operates on $\alpha_{i,1}$ and $\alpha_{i,2}$ to determine $\beta_{i,0}$ and $\beta_{i,1}$, whereas

$$\beta_{i,h} = \alpha_{i,h} \qquad for \quad h = 2,3,4,5 \qquad ...8.3.2$$

Associated with each pair of values of $\alpha_{i,1}$ and $\alpha_{i,2}$, in Table 8.2, are all eight combinations of $\alpha_{i,3}$, $\alpha_{i,4}$ and $\alpha_{i,5}$, so that every row in the truth table holds for these eight combinations. The eight possible values of the encoded symbol s_i , for any given set of values of $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$, are selected such that the minimum unitary distance between any of the eight $\{s_i\}$ is maximized, as in Fig. 8.5 (here m=5).

The linear baseband channel and the adaptive linear filter (Fig. 8.1) are as described in the previous section. The only difference here is the value of the data-symbol rate, which is 3200 symbol/s here. The sampling rate of the sampler in Fig. 8.1 is 3200 samples/s. The sampled impulse response of the linear baseband channel and adaptive linear filter is given by Eqn. 8.2.4 and the received sample at the output of the linear filter at time iT is given by Eqn. 8.2.5.

8.4 MODEL OF 19200 bit/s SYSTEM

When the data transmission system (Fig. 8.1) operates at a rate of 19200 bit/s, it uses either an uncoded 64 level QAM signal or else a coded 128-level QAM signal here. The carrier frequency of the QAM signal in either case is 1800 Hz; the element rate of the system is 3200 bauds. The information to be transmitted is carried by the binary digits $\{\alpha_{i,k}\}$. As before these digits are statistically independent and equally likely to have any of their two possible values 0 or 1. The $\{\alpha_{i,k}\}$ occur in a separate groups of six adjacent digits $\alpha_{i,1}\,\alpha_{i,2}\ldots\alpha_{i,6}$, which determine the corresponding uncoded or coded data symbol s_i .

The differential encoder used throughout the work is used here to determine the uncoded data symbol s_i . The operation of the differential encoder, the mapping of the uncoded data symbol and the operation of the differential decoder at the receiver are exactly the same as those described in Section 8.2 for the 14400 bit/s system. And when the transmitted data are convolutionally encoded, the operation of the encoder is also exactly the same as that described for the 14400 bit/s system. The only difference here between the 14400 and 19200 bit/s systems is the value of the data-symbol rate, which is 3200 symbol/s in the 19200 bit/s system and 2400 symbol/s in the 14400 bit/s system.

8.5 TELEPHONE CIRCUITS AND EQUIPMENT FILTERS

Models of two telephone circuits are used here to evaluate the performance of the detectors, when the data are transmitted at different rates. The telephone circuits are the telephone circuits 2 and 3 in Chapter 3. As described before, telephone circuit 2 introduces typical levels of signal distortion, whereas telephone circuit 3 is close to the poorest circuit normally considered for the transmission of data at a rate of 9600 bit/s [23,47]. The attenuation and group delay characteristics of telephone circuit 2 and 3 are shown in Fig.3.8 and Fig. 3.9, respectively, in Chapter 3.

When the data are transmitted at a rate of 14400 bit/s, with a data-symbol rate of 2400 symbol/s, the equipment filters (which are the combination of the transmitter and receiver filters) are the equipment filters-1, which are used in Chapter 3 for the transmission of data at a rate of 9600 bit/s. The attenuation and group-delay characteristics of the filters are shown in Fig.3.6 in Chapter 3. For data transmission systems operating at data-symbol rates greater than 2400 symbol/s and with telephone circuits which introduce typical levels of signal distortion (such as telephone circuit 2), the use of the equipment filters-1 result in excessive signal distortion [42]. This can be verified by inspecting the attenuation and group-delay characteristics of the equipment filters-1 and telephone circuit 2. Thus, a reduction in signal distortion, accompanied by the corresponding improvement in tolerance to noise, can be achieved by increasing the bandwidth of these filters [39,42]. Thus, wider-band equipment filters (called here equipment filters-3) are used for the transmission systems operating at rates of 16000 and 19200 bit/s. These systems operate with a data-symbol rate of 3200 symbol/s. The attenuation and group-delay characteristics of equipment filters-3 are shown in Fig. 8.8, where it can be seen that the bandwidth of the filters is wider than that of equipment filters-1 (Chapter 3) by about 200 Hz (at the -6 dB frequencies) at both the low and high frequency limits of the characteristics. Equipment filters-2 have been used successfully for transmission of an uncoded QAM signal (identical to the signal constellation in Fig. 8.2) over telephone lines at a rate of 19200 bit/s [46,92].

The sampled impulse responses of the linear baseband channel (Fig. 8.1) are derived in the same way as for the corresponding channel in Chapter 3 and described in

Appendix D. Clearly the sampled impulse responses of the linear baseband channels for the case where the data is transmitted at a rate of 14400 bit/s are the same as those used in Chapter 3, where equipment filters-1 and a symbol rate of 2400 symbol/s are used. The sampled impulse responses of the linear baseband channels which include telephone circuits 2 and 3 (known here as channel D and E, respectively, as in Chapter 3), are shown in Tables 8.3 and 8.4 for the cases where the data-symbol rates are 2400 and 3200 symbol/s, respectively. As described above, equipment filters-3 is used here when the data-symbol rate is 3200 symbol/s.

The sampled impulse responses of the linear baseband channel and the adaptive linear filter (given by the vector Y in Eqn. 8.2.4) for the two circuits are shown in Tables 8.5 and 8.6 for the case where the data-symbol rates are 2400 and 3200 symbol/s, respectively.

8.6 DETECTION PROCESSES

In Chapters 4 and 5, different detectors have been proposed for the detection of an uncoded and coded (convolutionally encoded) QAM signals. The preferred arrangements of these are used here for the data transmission systems operating at rates of 14400, 16000 and 19200 bit/s. These detectors are system 3U16, which is system 3U with 16 stored vectors as described in Chapter 4, and system 1C16, which is system 1C with 16 stored vectors as described in Chapter 5. The equalizers are also tested here.

Although, the principle of operation of a detector here is the same as that for the corresponding detector, described in Chapter 4 or 5, its operations here is slightly modified due to the larger numbers of signal levels in the data transmission system operating at a rate of 14400, 16000 or 19200 bit/s. For example, when a linear or nonlinear equalizer is used together with the appropriate threshold detector for the uncoded 32-level QAM signal, which is used by the 16000 bit/s system, the threshold levels are placed at -4, -2, 0, 2 and 4, for each dimension (real or imaginary part) in the constellation in Fig. 8.6 and the corresponding levels for the uncoded 64-level QAM signal (in the 14400 or 19200 bit/s system) are -6, -4, -2, 0, 2, 4 and 6

(Fig. 8.2). These levels are also used by system 3U16 to determine the possible values of the received data symbol for which the expanded vectors have the lowest costs, as described in Chapter 4.

When the transmitted signal is convolutionally encoded, the Viterbi algorithm detector (VA), which is used together with the equalizers in system LC or MNC, is modified here to consider more possible transitions than those used in Chapter 5 for the case where the transmission rate is 9600 bit/s. Although, the number of states (or the number of stored vectors) is 8, which is not changed here, the number of transitions from each state at time iT to a valid states at time (i+1)T vary according to the number of the transmitted signal levels. When the transmission system operates at a rate of 14400 or 19200 bit/s, the above number of transitions is 64 (Section 8.2), whereas the corresponding number for the 16000 bit/s system is 32 (Section 8.3).

When the near-maximum likelihood detector system 1C16 is used for the coded signal, each vector is expanded into four new vectors as described in Chapter 5. The expansion of these vectors must correspond to permitted transitions according to the truth table of the encoder, and such that the four vectors, which are derived from any original vector have the four lowest costs. This implies, that the search for the above permitted (valid) expansions is more complicated than that required for the case of the coded 32-level QAM signal, which is used in the 9600 bit/s system in Chapter 5.

8.7 RESULTS OF COMPUTER SIMULATION TESTS

Computer simulation tests have been carried out to evaluate the performances of the different detectors (systems), considered in the previous section, when the data are transmitted at rates of 14400, 16000 and 19200 bit/s over channel D and E. The results of the tests are shown in Figs. 8.9 to 8.21, where the performances are given in terms of the variation of bit error rate against the signal/noise ratio ψ. The latter is defined by Eqn. F.6 in Appendix F and the value of λ (the mean square value of the transmitted data symbols $\{s_i\}$), and m (the number of transmitted information digits carried by each data symbol) used in this equation for the different data transmission

systems considered in this Chapter are given in Table 8.7. Information about each system, such as the signal constellations for the uncoded and coded signal, the equipment filters and the modulation rates are also given in Table 8.7.

In Figs. 8.9 to 8.21, a total of at least 2.5 $\times 10^6$ data symbols were transmitted in plotting any one curve, and the 95% confidence limits for the relative positions of any two curves is generally better than $\pm$ 0.5 dB.

The performances of the optimum detectors for the uncoded 32-level and coded 64-level QAM signals in the absence of any signal distortion (channel A) are shown in Fig. 8.9. The corresponding performances of the uncoded 64-level and coded 128-level QAM signals are also shown in Fig. 8.10. The optimum detector for the uncoded signal, over channel A, is the threshold detector and that for the coded signal is the Viterbi algorithm detector (VA). Fig. 8.9 shows that an improvement in tolerance to additive white Gaussian noise of about 1.8 to 2.8 dB is achieved at bit error rates of 10^{-3} to 10^{-4} through the use of coding, whereas the corresponding improvement in Fig. 8.10 is about 1.6 to 2.6 dB.

In Figs. 8.11 to 8.21, the curves are labelled by the corresponding systems as in Chapters 4 and 5. These figures show that the linear equalizers (systems LC and LU) give relatively poor performances when compared to the other systems tested here. Although the nonlinear equalizers have a much better performances over the two channels tested, and particularly when the data are transmitted at symbol rates of 3200 symbol/s, than the linear equalizers, their performances are inferior to those of the near-maximum likelihood detectors, as expected. Figs. 8.12, 8.14, 8.16, 8.18 and 8.20 show that the use of the modified nonlinear equalizer with the Viterbi algorithm detector (system MNC) for the coded signal degrades the performance of the coded system below that of the uncoded system, when the latter also uses a nonlinear equalizer.

The results here show that system 3U16 gains a relatively large advantage in tolerance to noise over all systems which use an equalizer, whether the signal is convolutionally encoded or not. The best performance, of the systems tested here, is given by system 1C16, which is the near-maximum likelihood detector used for the encoded signal. Table 8.8 shows the relative advantages, in tolerance to additive noise, (in dB) gained by system 1C16 over system 3U16 over channel D and E,

when the data are transmitted at different rates. A comparison between this table and the results of Chapter 5, show that the advantages gained here are approximately similar to those gained when the data are transmitted at a rate of 9600 bit/s.

It is clear from the above results that the proposed detection processes (tested in this chapter) have a behaviour similar to that when the data are transmitted at a rate of 9600 bit/s.

$\alpha_{i,1}$	$\alpha_{i,2}$	$\beta_{i-1,1}$	$\beta_{i-1,2}$	$\beta_{i,1}$	$\beta_{i,2}$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	1	1	1
0	0	1	0	1	0
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	1	1	0
0	1	1	0	0	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	1	0	0
1	1	1	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	1	0	1
1	0	1	0	1	1

Table 8.1 Truth table of the differential encoder

State at time iT			Input at time iT		Output at time iT			State at time (i+1)T		
$u_{i,0}$	$u_{i,1}$	$u_{i,2}$	$\alpha_{i,1}$	$\alpha_{i,2}$	$\beta_{i,0}$	$\beta_{i,1}$	$\beta_{i,2}$	$u_{i+1,0}$	$u_{i+1,1}$	$u_{i+1,2}$
0	0	0	0	0	0	0	0	0	0	0
			1	0	0	1	0	0	0	1
			0	1	0	0	1	0	1	0
			1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0	1	0
			1	0	0	1	0	0	1	1
			0	1	0	0	1	0	0	0
			1	1	0	1	1	0	0	1
0	1	0	0	0	0	1	0	0	1	0
			1	0	0	0	0	0	1	1
			0	1	0	1	1	0	0	0
			1	1	0	0	1	0	0	1
1	1	0	0	0	0	1	0	0	0	0
			1	0	0	0	0	0	0	1
			0	1	0	1	1	0	1	0
			1	1	0	0	1	0	1	1
0	0	1	0	0	1	1	0	1	1	1
			1	0	1	0	0	1	0	0
			0	1	1	1	1	1	0	1
			1	1	1	0	1	1	1	0
1	0	1	0	0	1	1	0	1	0	1
			1	0	1	0	0	1	1	0
			0	1	1	1	1	1	1	1
			1	1	1	0	1	1	0	0
0	1	1	0	0	1	0	0	1	0	1
			1	0	1	1	0	1	1	0
			0	1	1	0	1	1	1	1
			1	1	1	1	1	1	0	0
1	1	1	0	0	1	0	0	1	1	1
			1	0	1	1	0	1	0	0
			0	1	1	0	1	1	0	1
			1	1	1	1	1	1	1	0

Table 8.2 Truth table of the convolutional encoder

Channel D (Telephone circuit 2)		Channel E (Telephone circuit 3)	
Real Part	Imaginary Part	Real Part	Imaginary Part
0.0145	-0.0006	0.0176	-0.0175
0.0750	0.0176	0.1381	-0.1252
0.3951	0.0033	0.4547	-0.1885
0.7491	-0.1718	0.5078	0.1622
0.1951	0.0972	-0.1966	0.3505
-0.2856	0.1894	-0.2223	-0.2276
0.0575	-0.2096	0.2797	-0.0158
0.0655	0.1139	-0.1636	0.1352
-0.0825	-0.0424	0.0594	-0.1400
0.0623	0.0085	-0.0084	0.1111
-0.0438	0.0034	-0.0105	-0.0817
0.0294	-0.0049	0.0152	0.0572
-0.0181	0.0032	-0.0131	-0.0406
0.0091	0.0003	0.0060	0.0255
-0.0038	-0.0023	0.0003	-0.0190
0.0019	0.0027	-0.0035	0.0116
-0.0018	-0.0014	0.0041	-0.0078
0.0006	0.0003	-0.0031	0.0038
0.0005	0.0000	0.0018	-0.0005
-0.0008	-0.0001	-0.0018	-0.0005

Table 8.3 Sampled impulse responses of the linear baseband channels D and E, at sampling rate of 2400 samples/s.

Channel D (Telephone circuit 2)		Channel E (Telephone circuit 3)	
Real Part	Imaginary Part	Real Part	Imaginary Part
0.001966	0.000544	0.001155	-0.001275
0.016707	0.003912	0.011734	-0.016035
0.067284	0.008166	0.057043	-0.059210
0.221772	0.000227	0.170236	-0.103845
0.502139	-0.056595	0.289941	-0.041411
0.548069	-0.110307	0.203587	0.140088
0.046425	0.064073	-0.106333	0.175431
-0.294318	0.210038	-0.188929	-0.059493
0.006498	-0.101885	0.098031	-0.135501
0.145968	-0.128412	0.116619	0.098686
-0.070121	0.184022	-0.141349	0.062653
-0.051347	-0.055243	0.004742	-0.126885
0.084641	-0.087113	0.115057	0.052773
-0.017438	0.127455	-0.106427	0.058291
-0.048800	-0.070772	0.013733	-0.100340
0.059209	-0.008137	0.067016	0.061483
-0.029553	0.055785	-0.085331	0.007381
-0.007553	-0.058468	0.049681	-0.055936
0.028324	0.035002	0.000236	0.061144
-0.031788	-0.007764	-0.035259	-0.039069

Table 8.4 Sampled impulse responses of the linear baseband channels D and E, at sampling rate of 3200 samples/s.

Channel D (Telephone circuit 2)		Channel E (Telephone circuit 3)	
Real Part	Imaginary Part	Real Part	Imaginary Part
1.0000	0.0000	1.0000	0.0000
0.5091	0.1960	0.4861	1.0988
-0.1465	0.0000	-0.5980	0.0703
0.0323	-0.0171	0.1702	-0.1938
0.0125	0.0200	-0.0245	0.1000
-0.0099	-0.0109	0.0100	-0.0258
0.0046	0.0074	-0.0134	0.0110
-0.0069	-0.0083	0.0056	-0.0042
0.0059	0.0076	0.0003	0.0003
-0.0025	-0.0053	-0.0008	0.0041
-0.0013	0.0040	0.0000	-0.0061
0.0024	-0.0028	0.0007	-0.0007
-0.0009	0.0018	0.0037	0.0002
-0.0006	-0.0006	-0.0019	-0.0025
0.0001	-0.0003	0.0020	0.0008
0.0002	0.0008	0.0005	-0.0002
0.0000	-0.0006	-0.0022	0.0002
-0.0003	-0.0001	0.0007	-0.0005
-0.0002	0.0003	-0.0008	0.0002
0.0003	-0.0002	0.0005	0.0005

Table 8.5 Sampled impulse responses of the linear baseband channel and adaptive linear filter in Fig. 8.1, for telephone circuits 2 and 3, at sampling rate of 3200 samples/s.

Channel D (Telephone circuit 2)		Channel E (Telephone circuit 3)	
Real Part	Imaginary Part	Real Part	Imaginary Part
1.0000	0.0000	1.0000	0.0000
1.2281	0.2098	1.2234	1.3398
-0.1585	0.1141	-0.8484	1.0919
-0.1567	-0.0519	-0.4543	-0.6497
0.2092	0.0299	0.4840	0.0337
-0.1170	0.0069	-0.2570	0.2106
-0.0086	-0.0100	0.0002	-0.2023
0.0473	0.0075	0.0982	0.0929
-0.0227	0.0048	-0.0705	0.0089
0.0115	0.0044	0.0238	-0.0229
-0.0040	-0.0045	-0.0022	0.0164
0.0000	0.0036	-0.0054	-0.0032
0.0002	0.0030	0.0046	-0.0016
-0.0018	-0.0059	-0.0040	-0.0002
0.0005	0.0048	0.0048	-0.0052
-0.0002	-0.0022	0.0012	0.0023
0.0009	0.0005	-0.0011	-0.0033
-0.0006	0.0015	0.0077	-0.0025
-0.0007	-0.0024	-0.0041	0.0030
0.0012	0.0015	0.0041	-0.0035

Table 8.6 Sampled impulse response of the linear baseband channel and adaptive linear filter in Fig. 8.1, for telephone circuit 2 and 3, at sampling rate of 3200 samples/s.

System	Modulation rate (bauds)	No. of data bits per symbol m	Signal constellation		Mean square value of $\{s_i\}$ λ		Equipment filters
			Uncoded	Coded	Uncoded	Coded	
14400 bit/s	2400	6	Fig. 8.2	Fig. 8.4	42	84	1
16000 bit/s	3200	5	Fig. 8.6	Fig. 8.7	20	42	3
19200 bit/s	3200	6	Fig. 8.2	Fig. 8.4	42	84	3

Table 8.7 Information about each system tested in chapter 8.

Bit error rates / System	10^{-2}	10^{-3}	10^{-4}
14400 bit/s	-0.10	0.70	1.30
16000 bit/s	-0.10	0.55	1.30
19200 bit/s	-0.10	0.60	1.30

Table 8.8.a Relative tolerance to additive white Gaussian noise (dB) of detector 1C16 to that of 3U16 at different bit error rates, when operating over telephone circuit 2 (channel D).

Bit error rates / System	10^{-2}	10^{-3}	10^{-4}
14400 bit/s	-0.30	0.25	0.80
16000 bit/s	-0.45	0.15	0.80
19200 bit/s	-0.40	0.20	0.75

Table 8.8.b Relative tolerance to additive white Gaussian noise (dB) of detector 1C16 to that of 3U16 at different bit error rates, when operating over telephone circuit 3 (channel E).

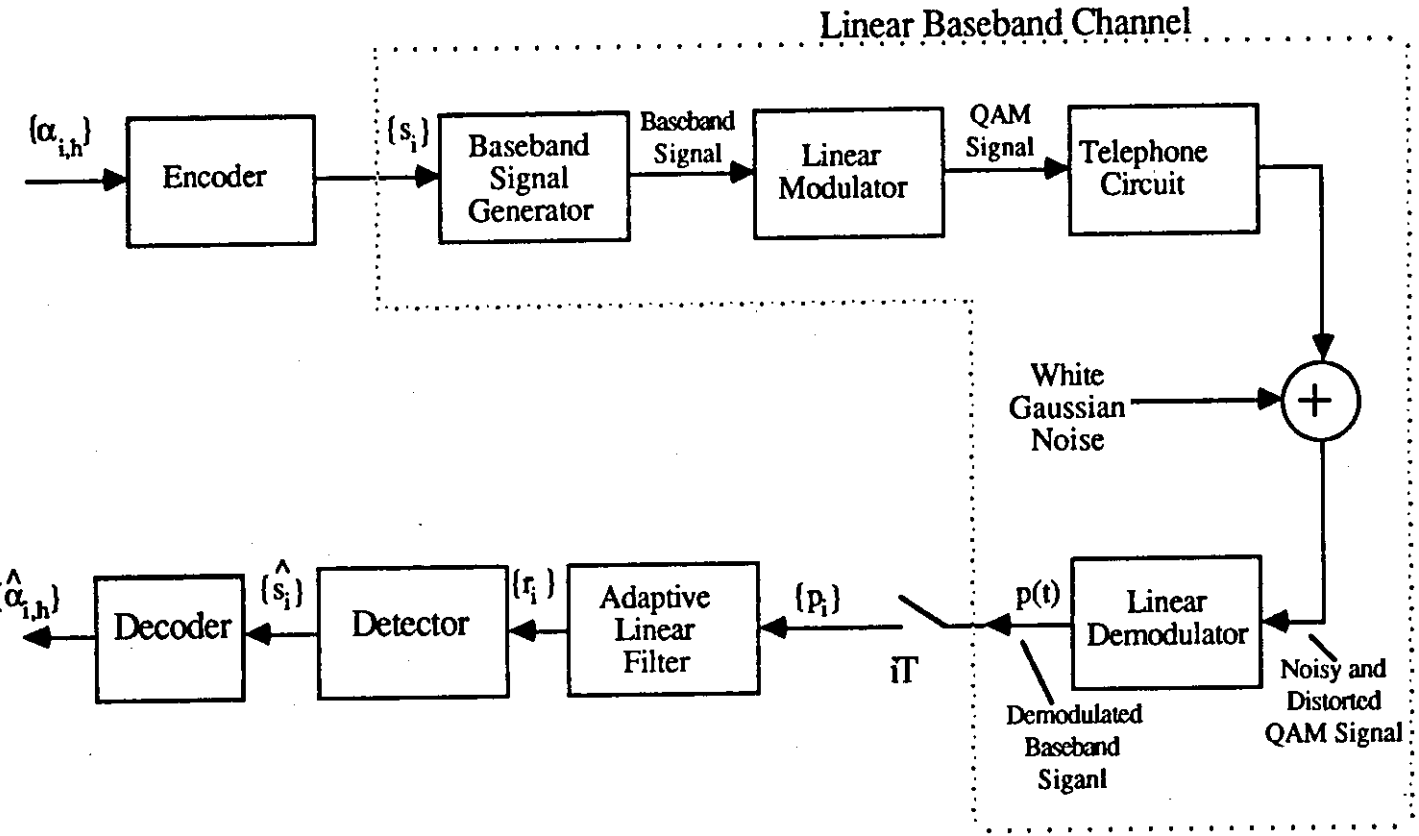


Fig.8.1 Model of data transmission system

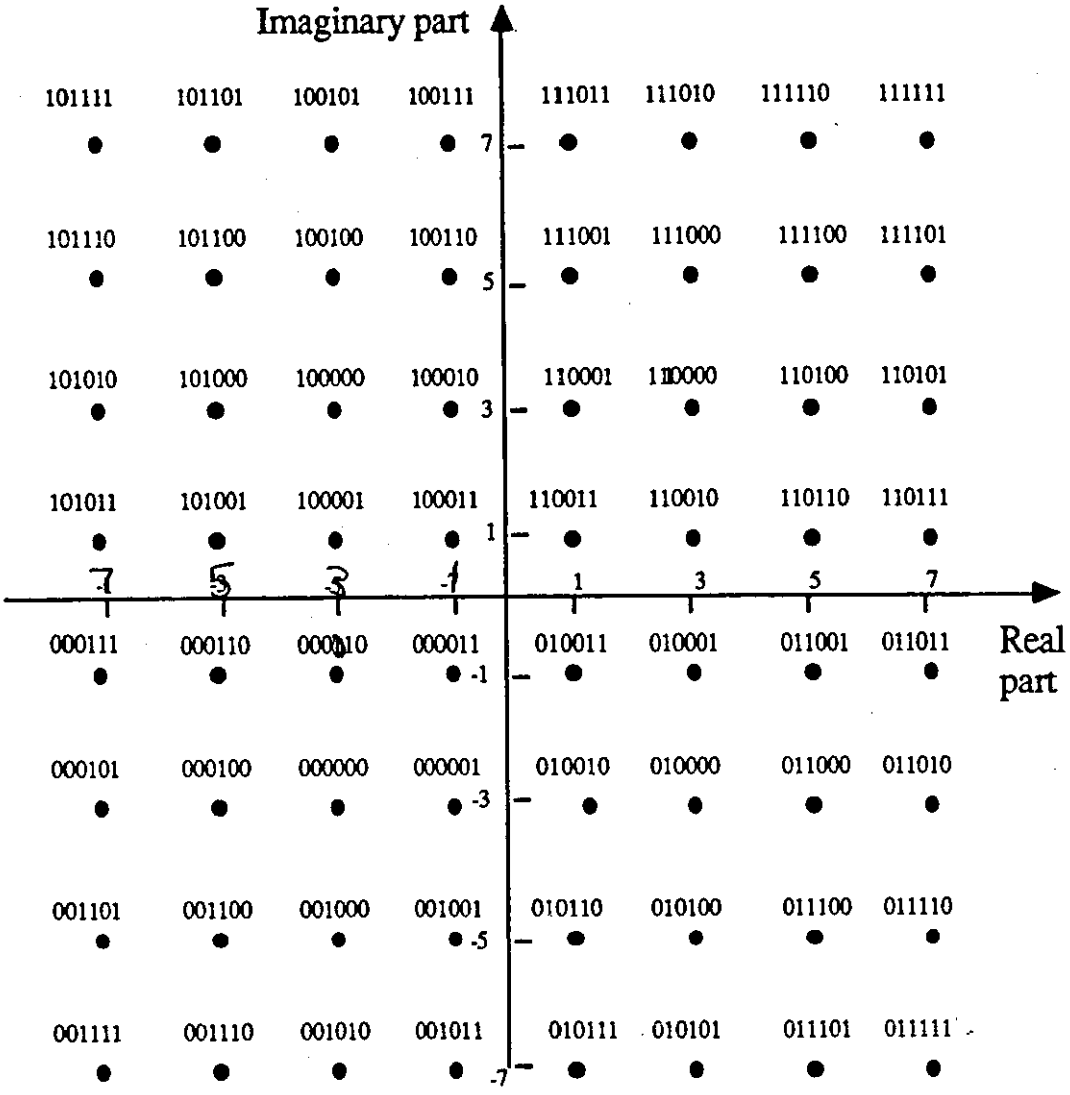


Fig.8.2 Signal constellation for uncoded 64-level QAM sognal .

(th binary coded number against each point is $\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}\beta_{i,5}\beta_{i,6}$)

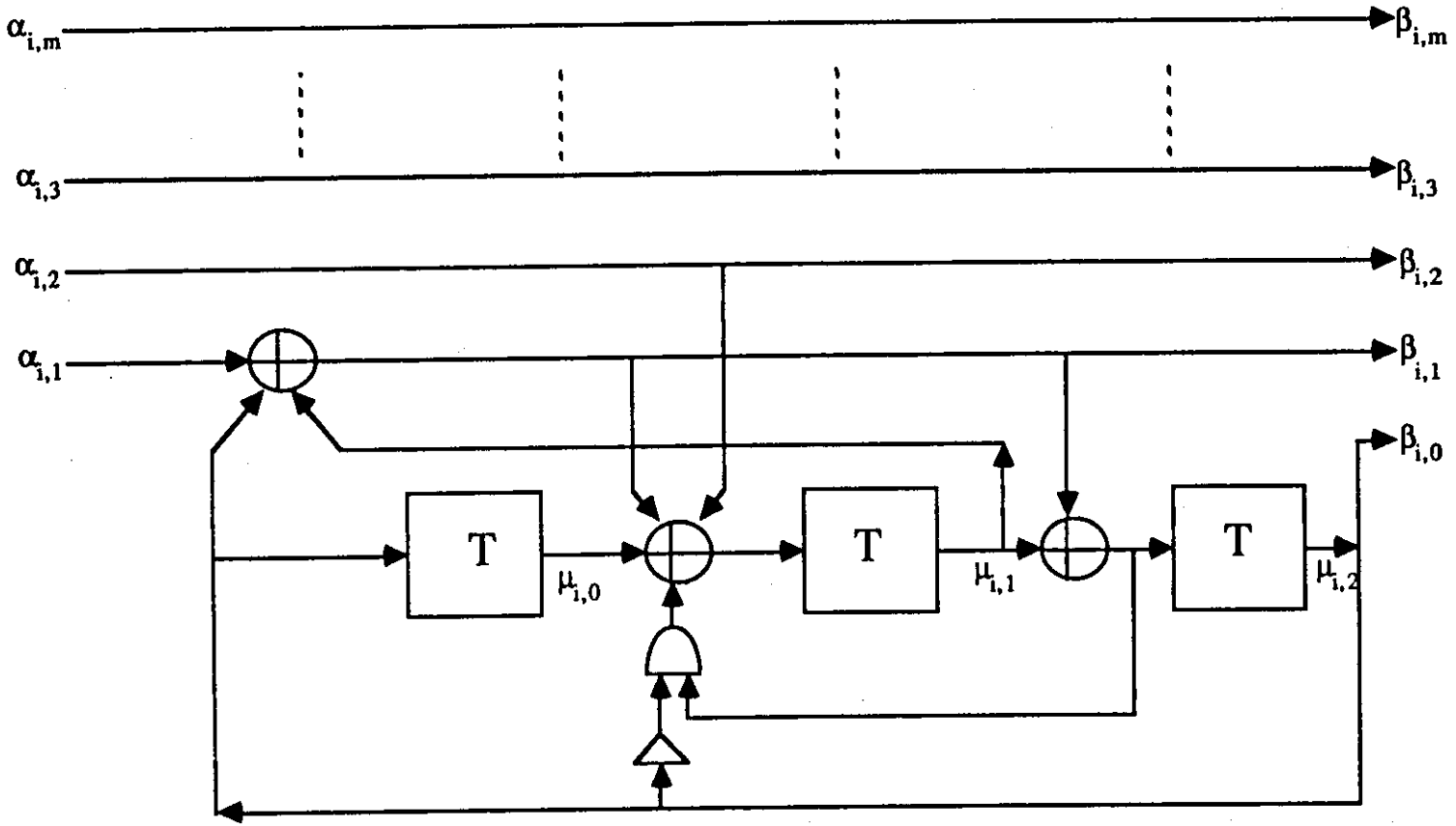


Fig.8.3 The convolutional encoder

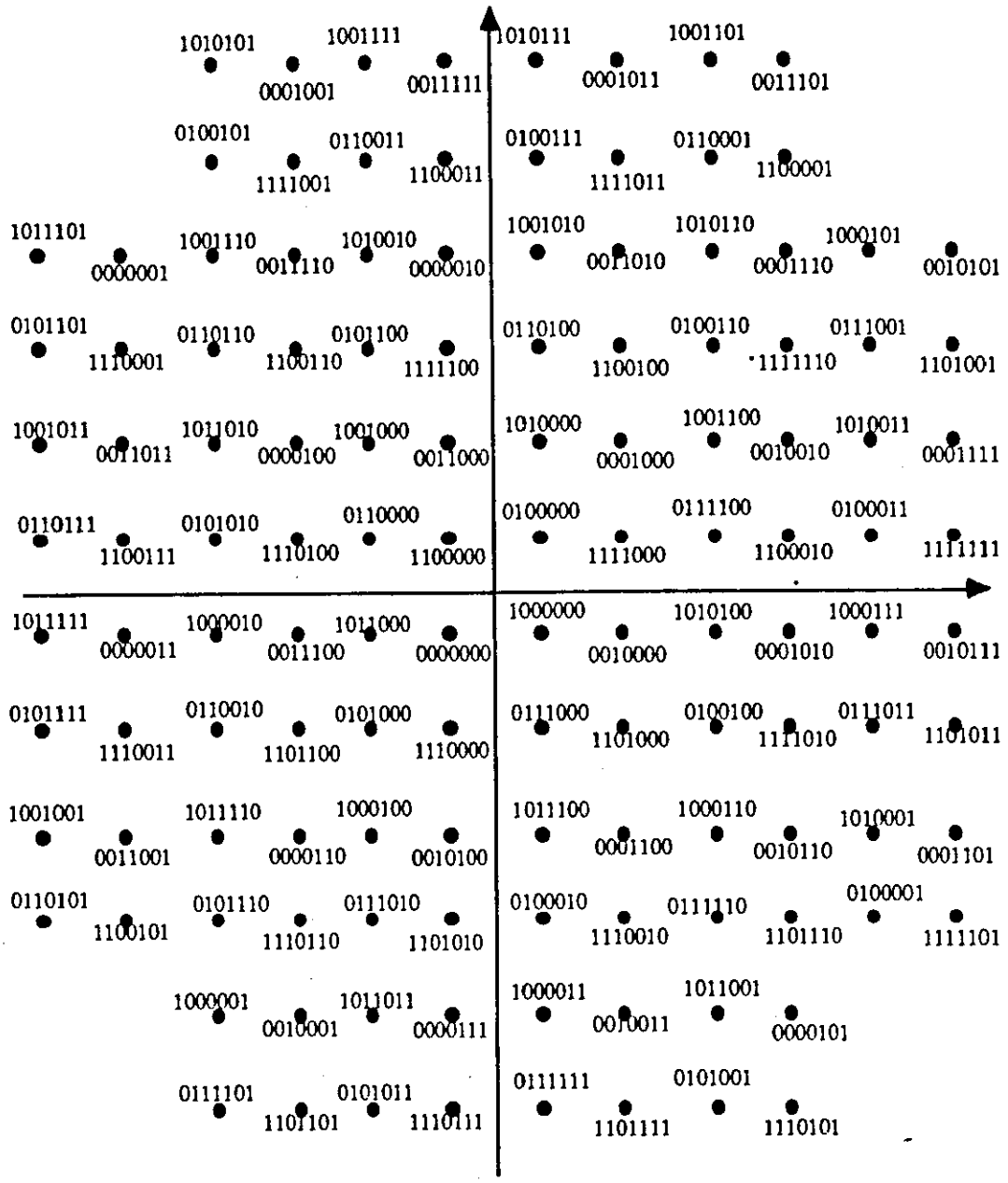


Fig.8.4 Signal constellation for coded 128-level QAM signal.

(the binary coded number against each point is $\beta_{i,0}\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}\beta_{i,5}\beta_{i,6}$)

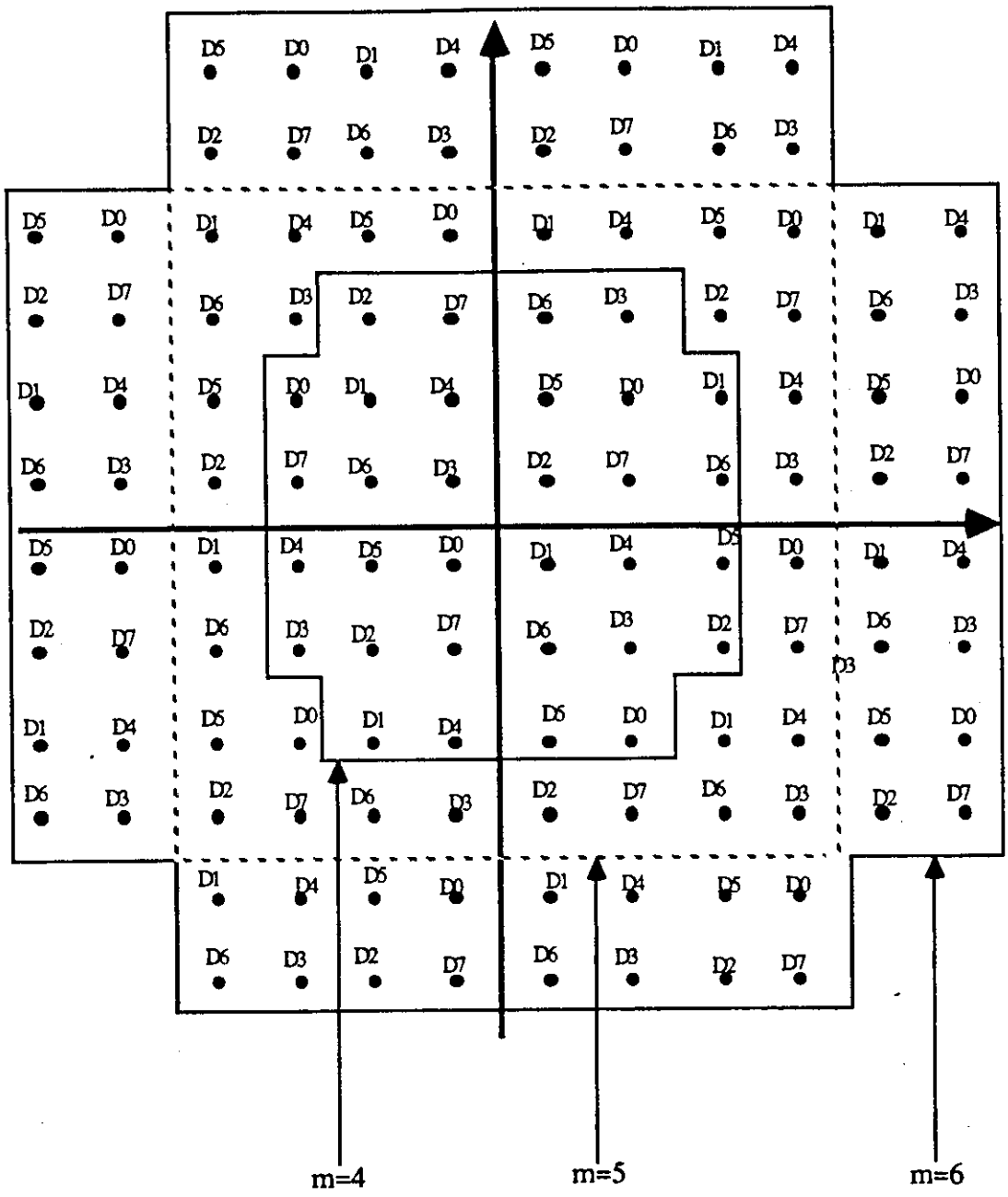


Fig.8.5 Mapping of the coded symbols for different signal levels.

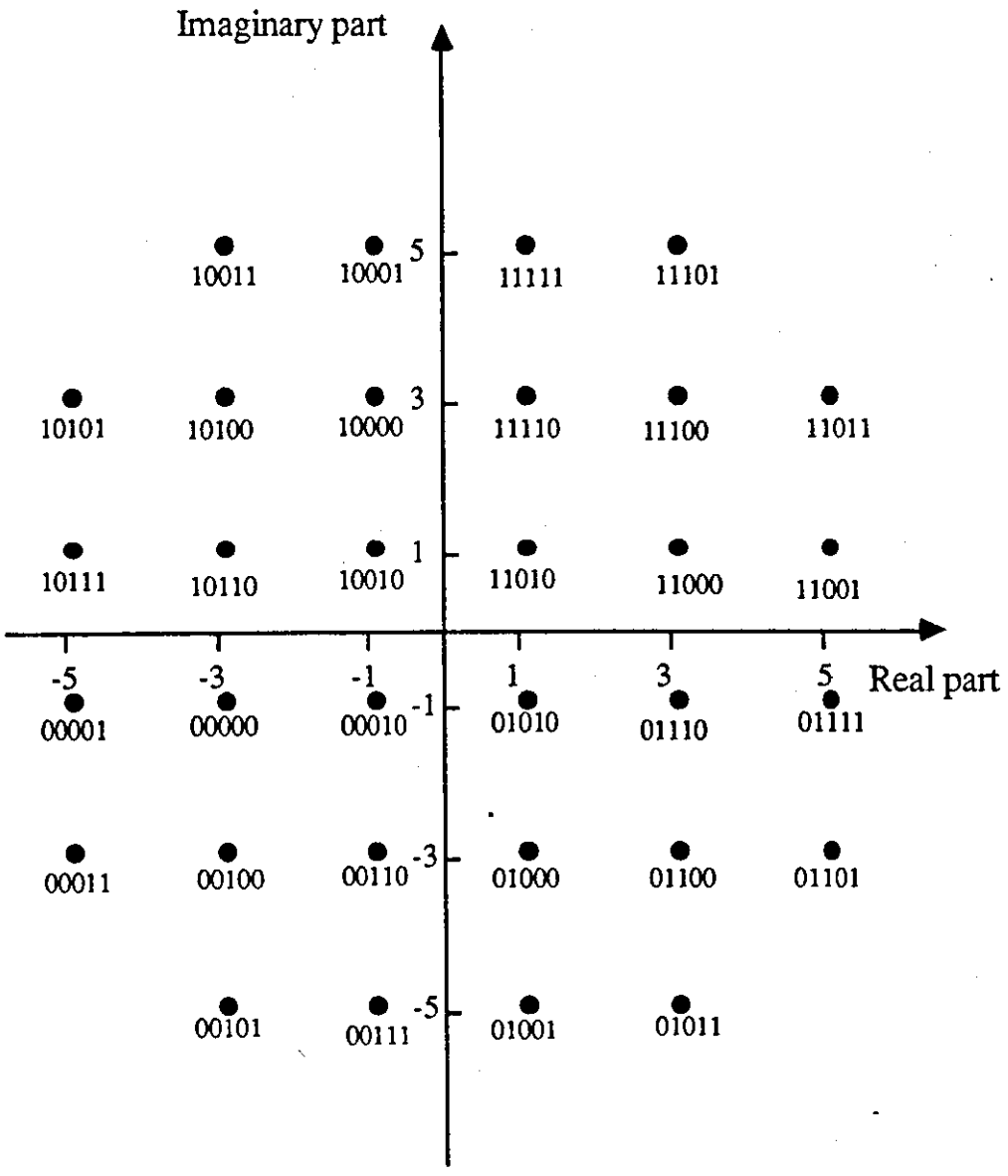


Fig.8.6 Signal constellation for uncoded 32-level QAM signal .

(the binary coded number against each point is $\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}\beta_{i,5}$)

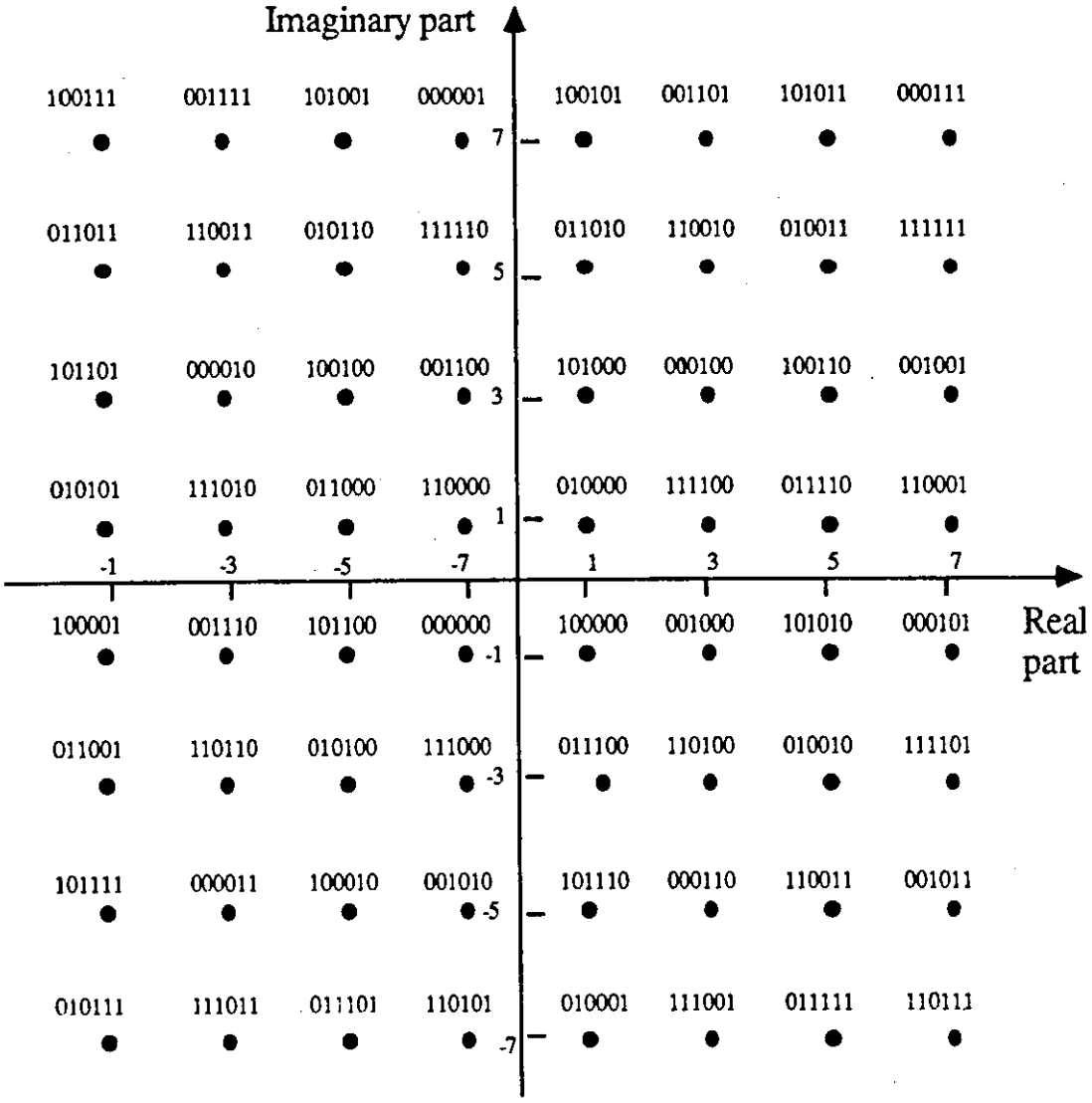
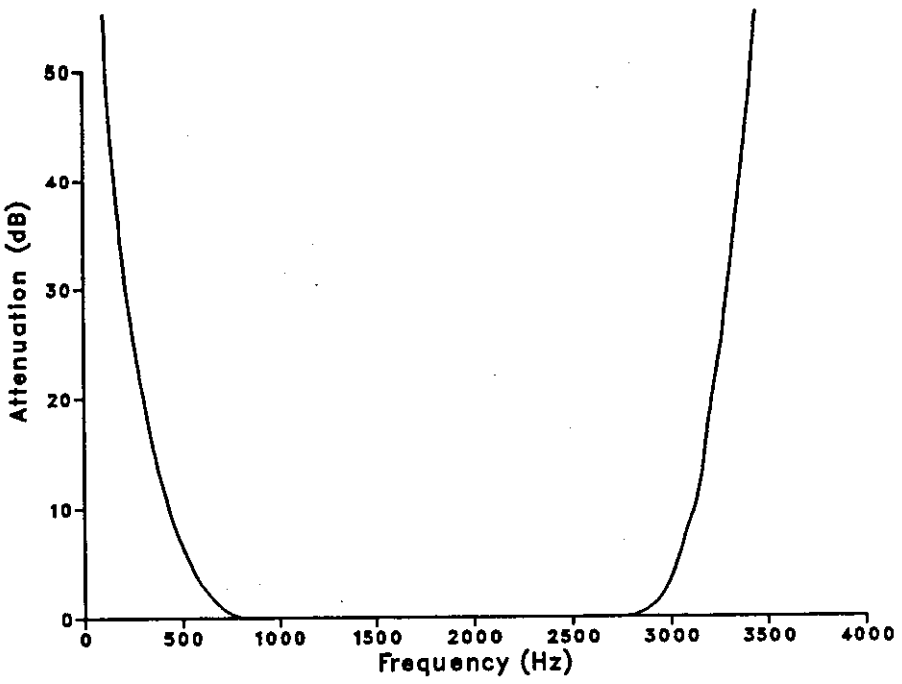
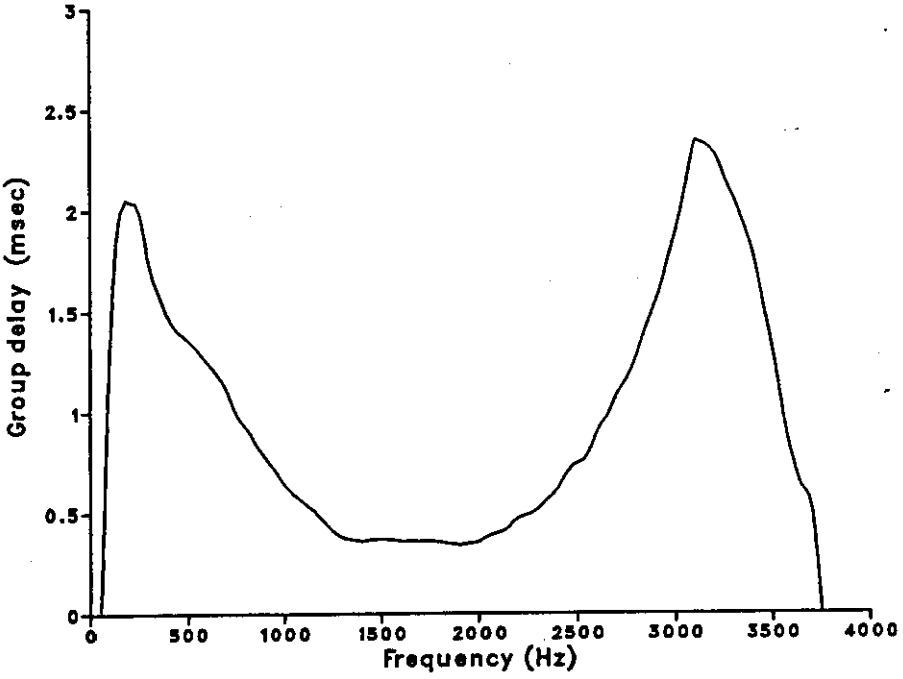


Fig.8.7 Signal constellation for coded 64-level QAM signal .

(the binary coded number against each point is $\beta_{i,0}\beta_{i,1}\beta_{i,2}\beta_{i,3}\beta_{i,4}\beta_{i,5}$)



a – Attenuation characteristic



b – Group delay characteristic

Fig.8.8 Attenuation and group delay characteristics of the equipment filters–3

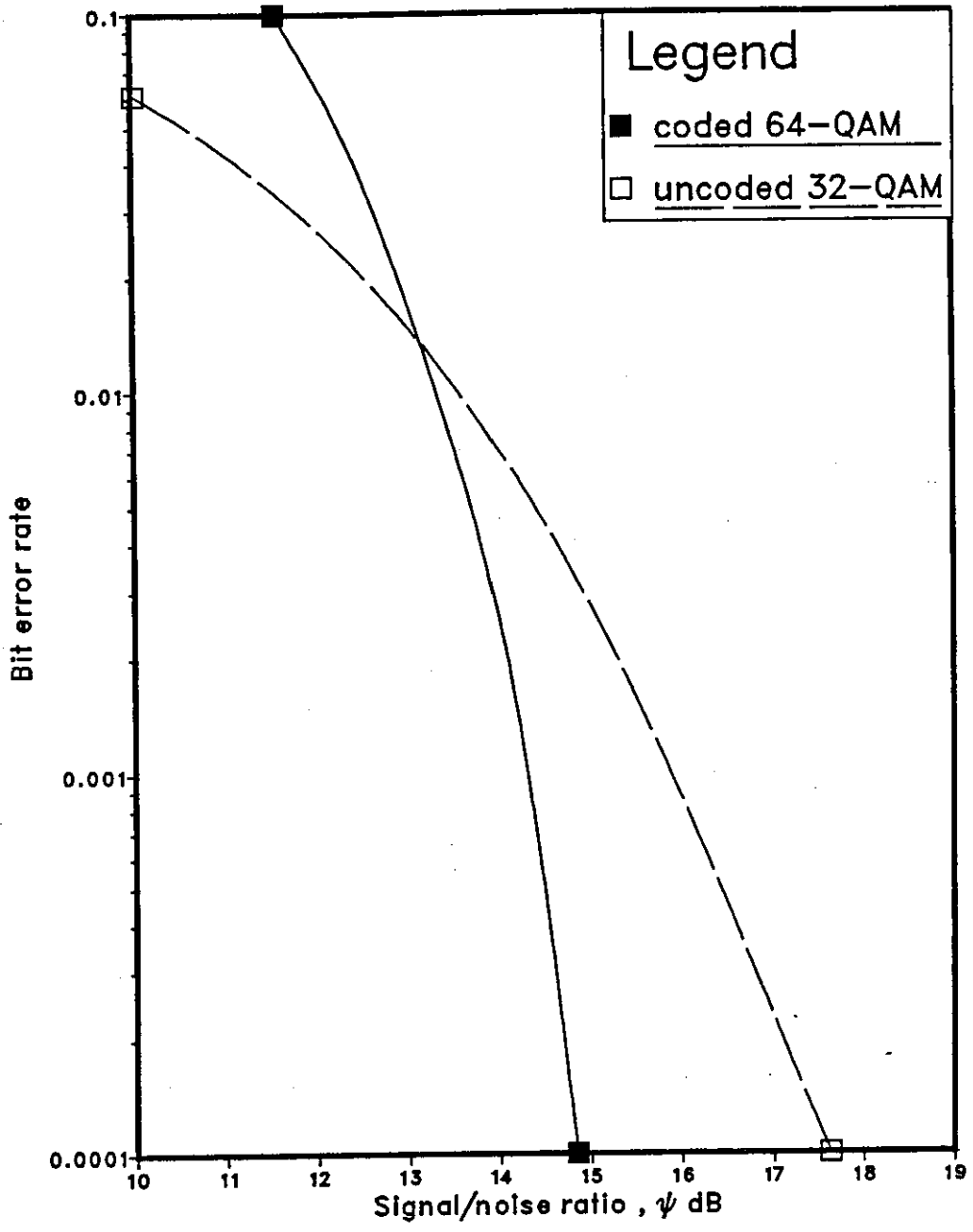


Fig.8.9 Performance of optimum detectors over channel A.

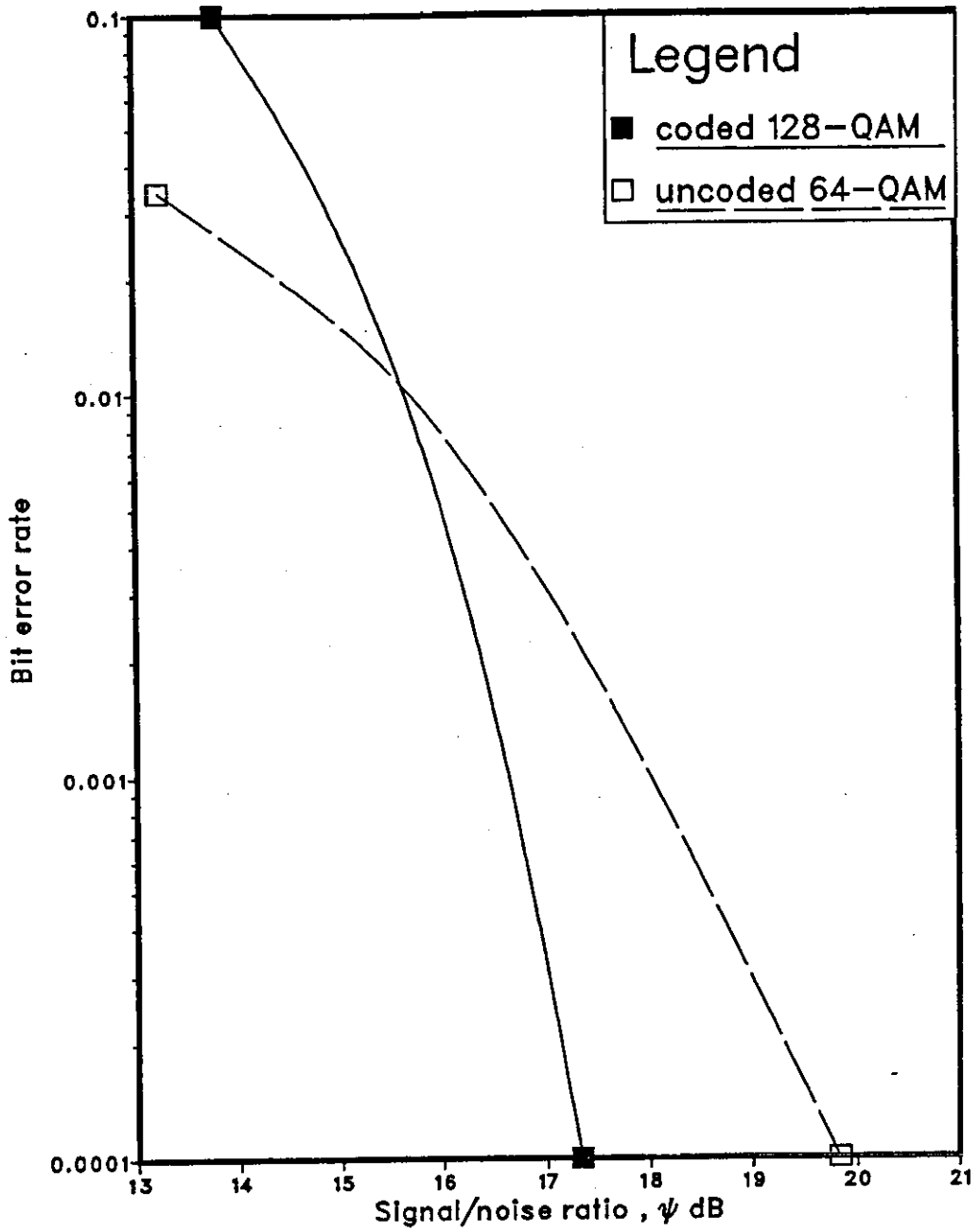


Fig.8.10 Performance of optimum detectors over channel A.

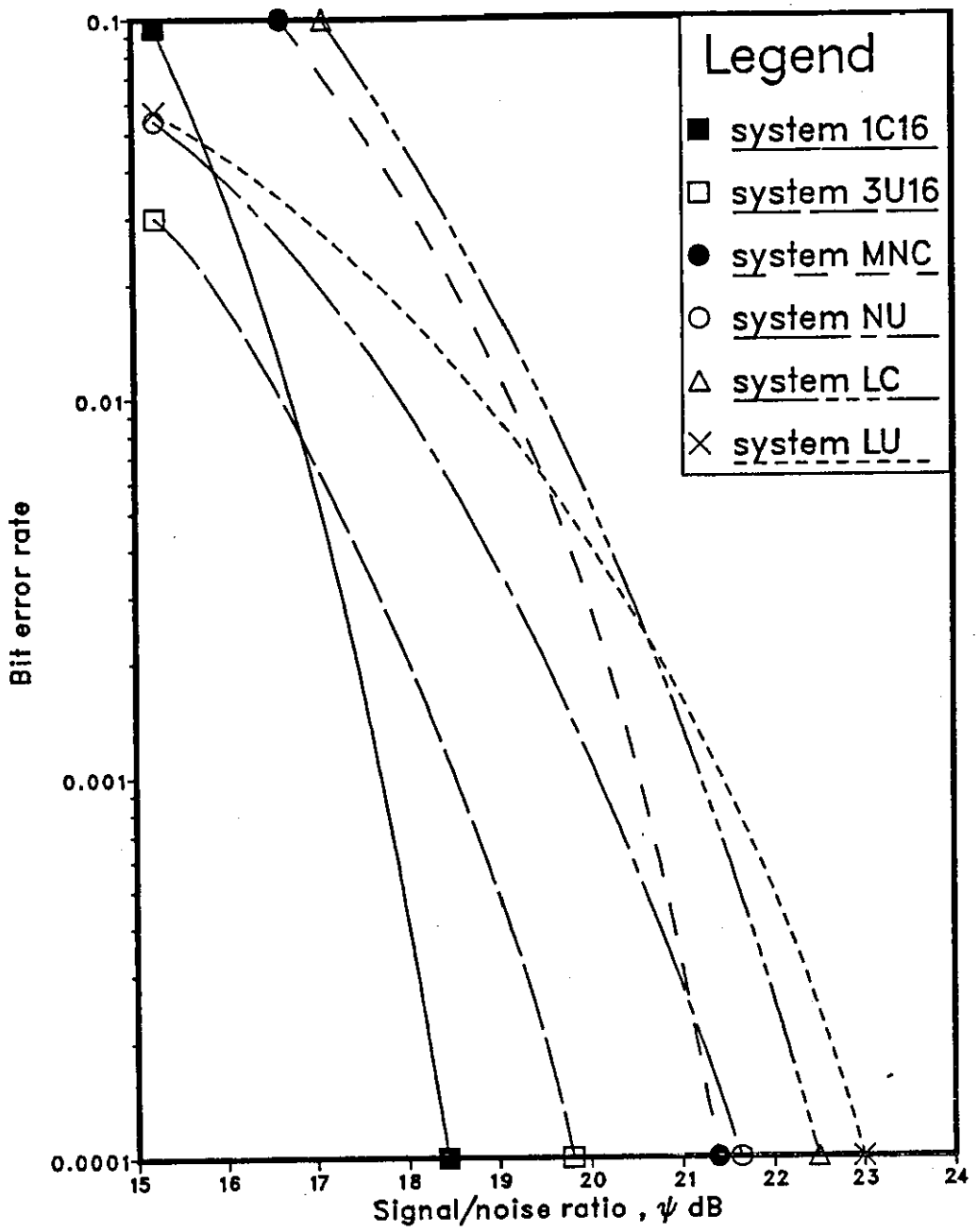


Fig.8.11 Performance of various detectors at 14400 bit/s over channel D.

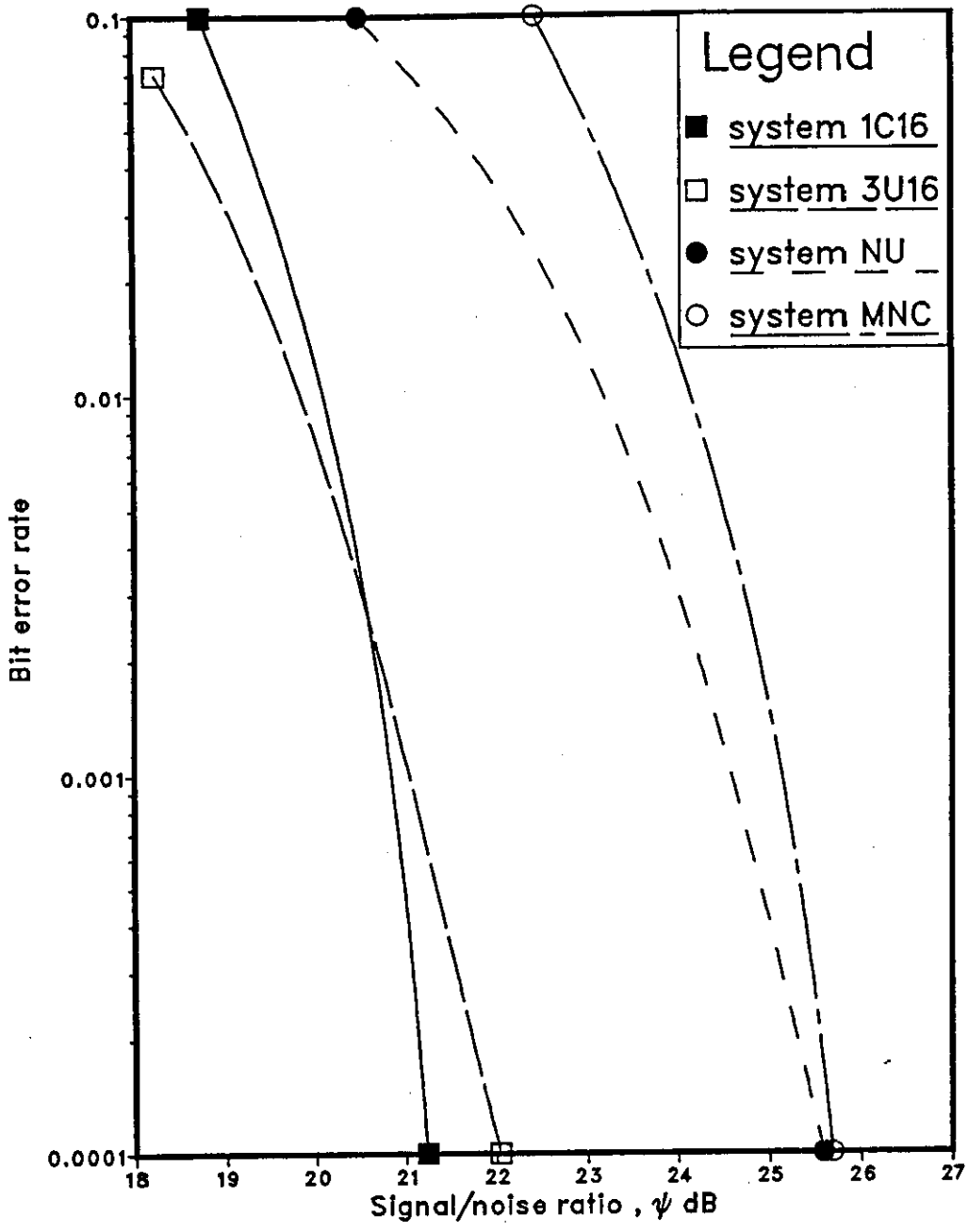


Fig.8.12 Performance of various detectors at 14400 bit/s over channel E.

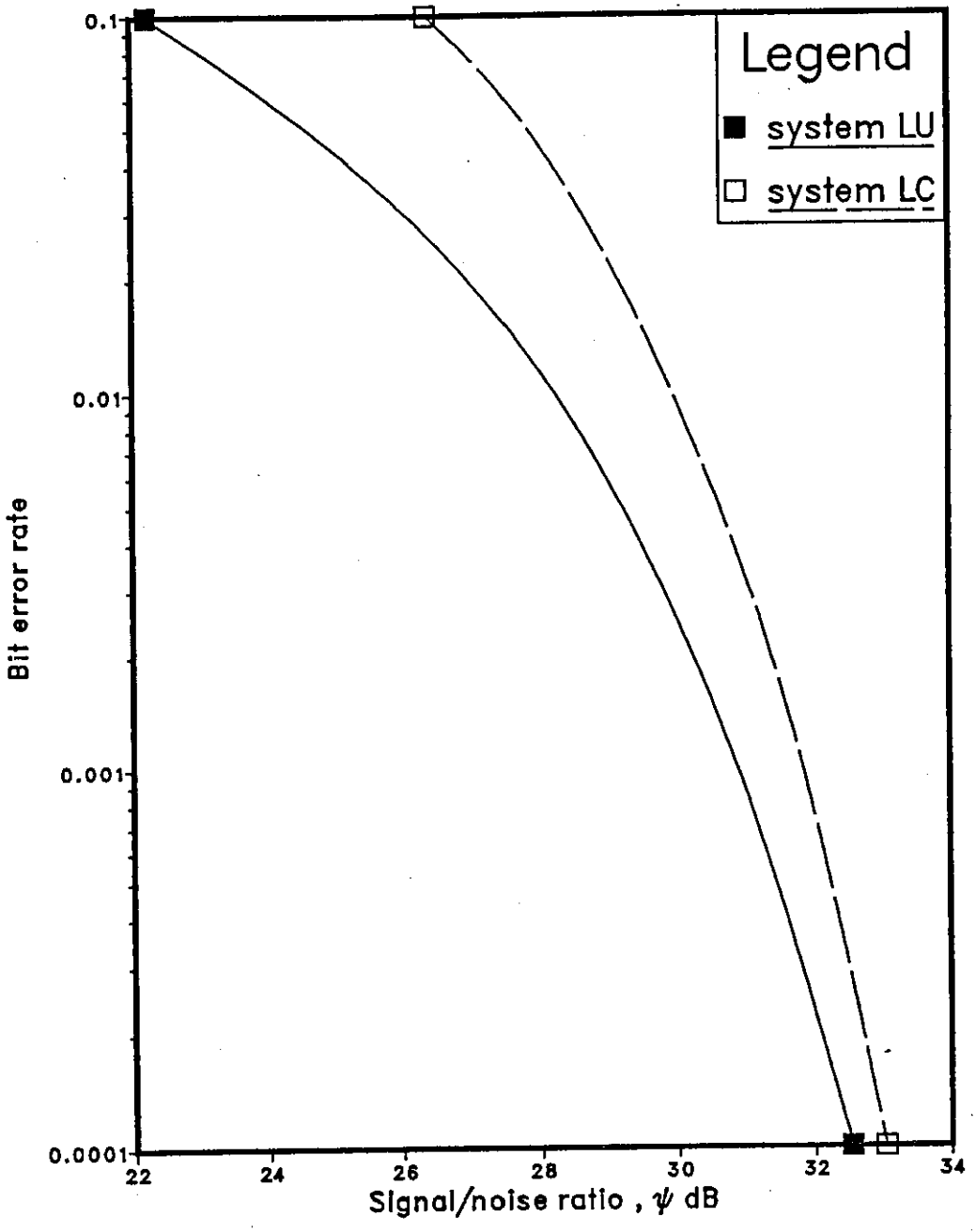


Fig.8.13 Performance of linear equalizers at 14400 bit/s over channel E.

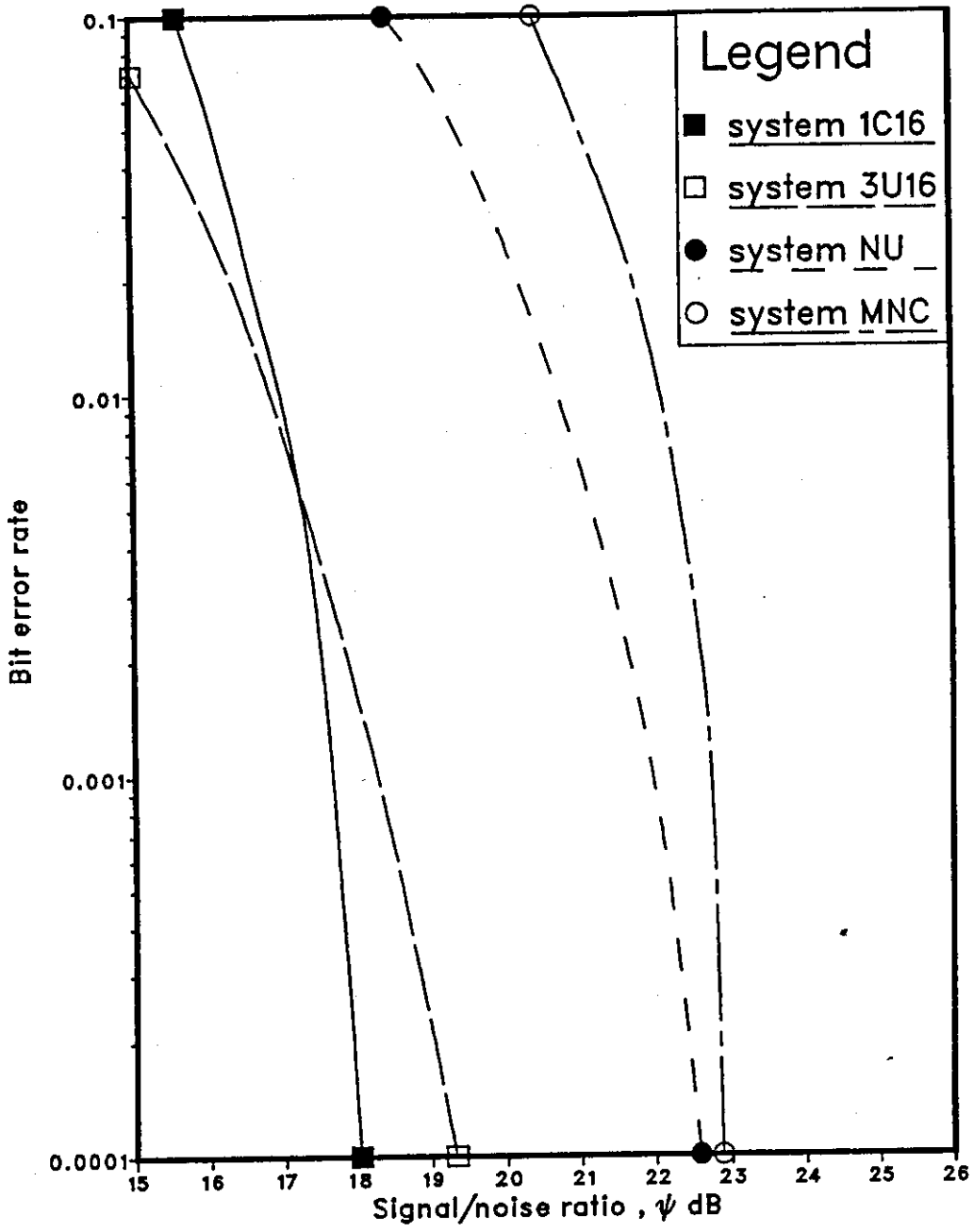


Fig.8.14 Performance of various detectors at 16000 bit/s over channel D.

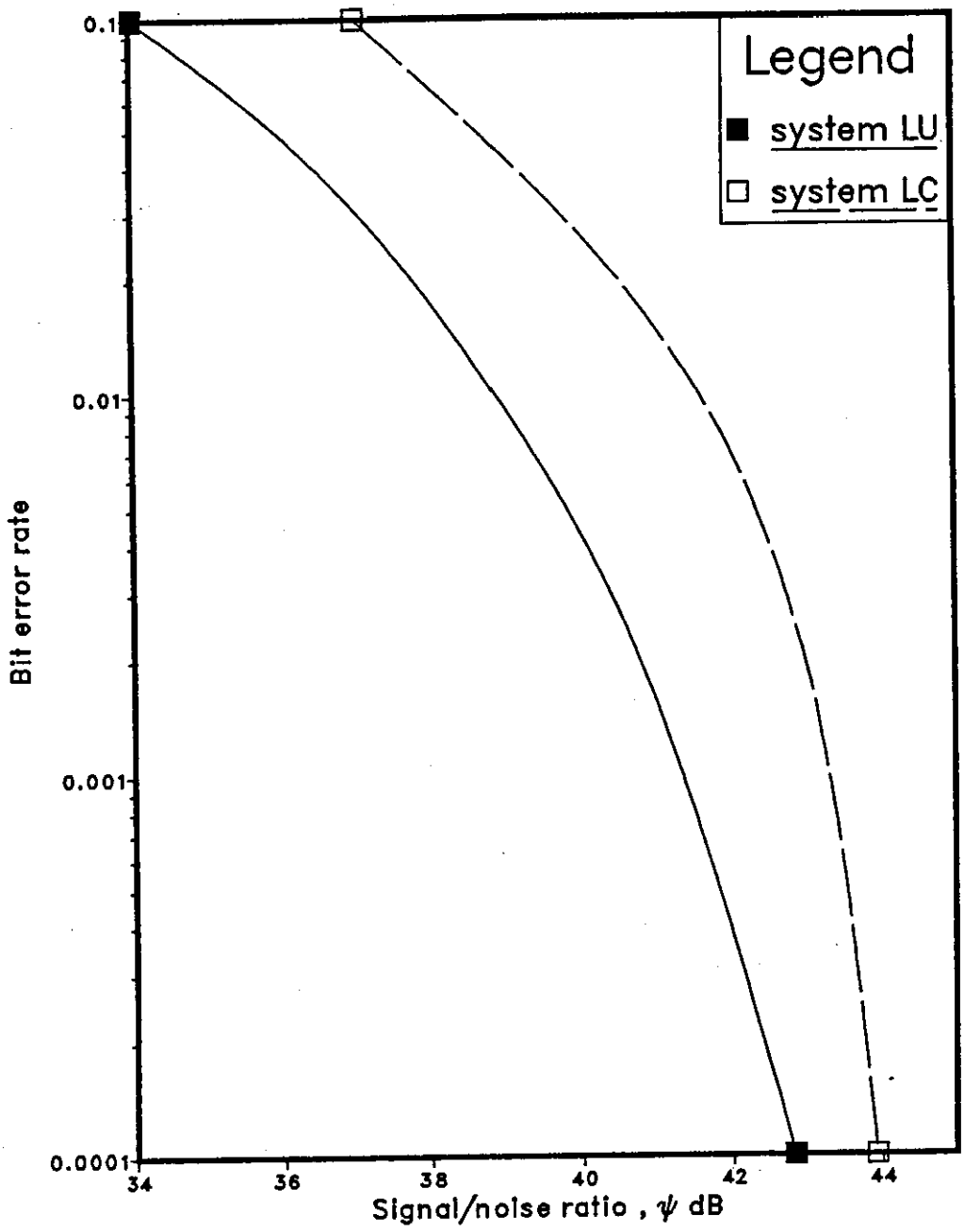


Fig.8.15 Performance of linear equalizers at 16000 bit/s over channel D.

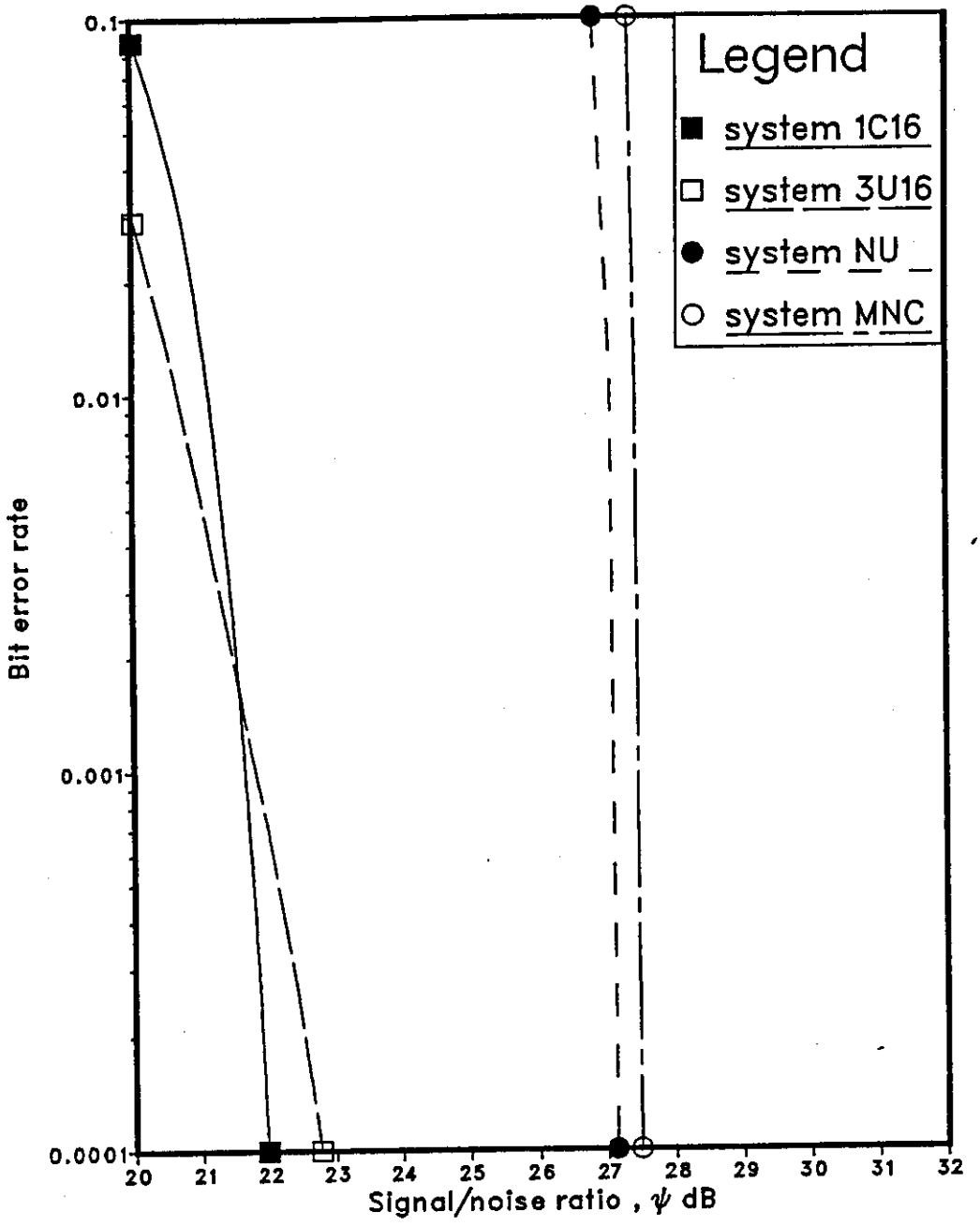


Fig.8.16 Performance of various detectors at 16000 bit/s over channel E.

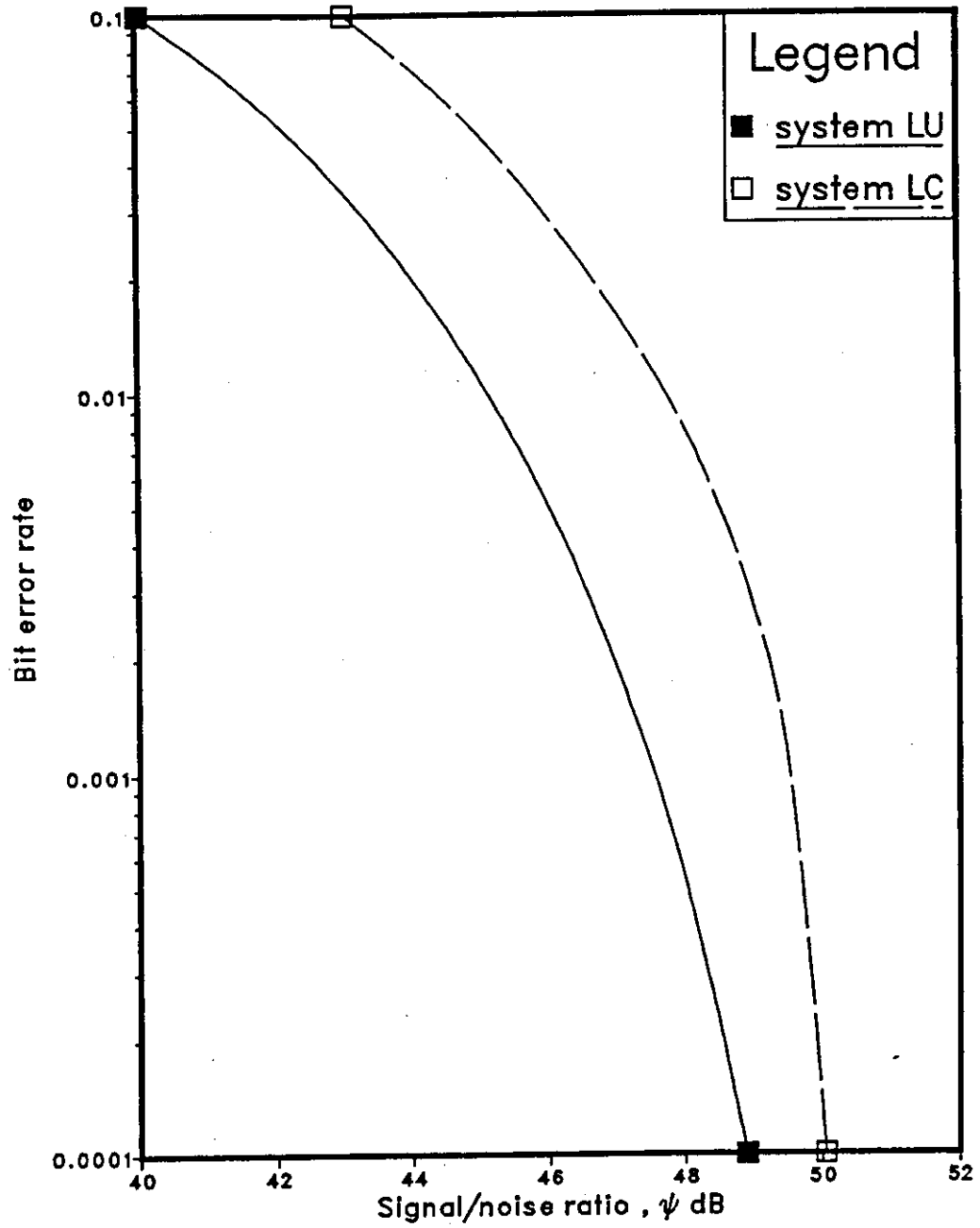


Fig.8.17 Performance of linear equalizers at 16000 bit/s over channel E.

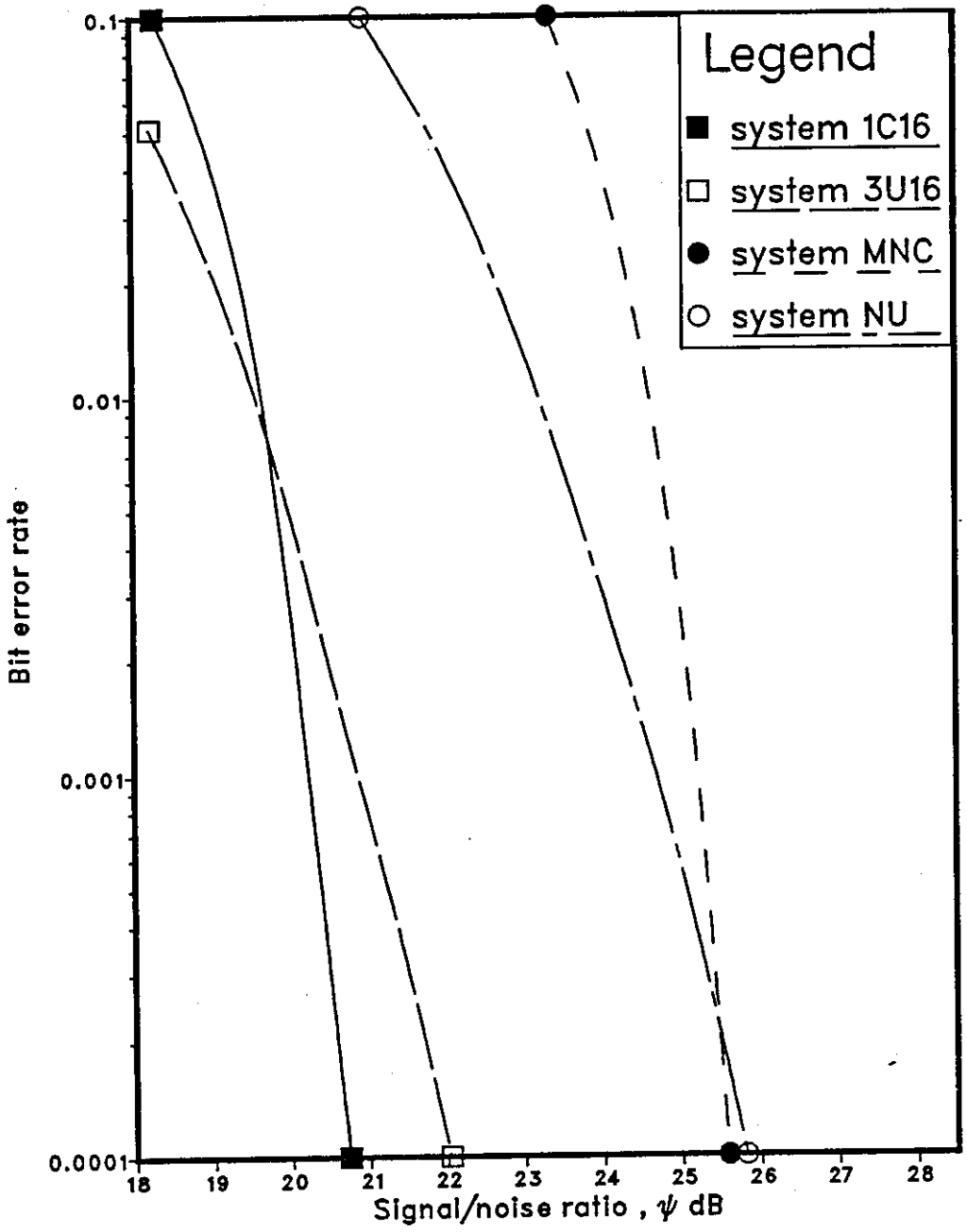


Fig.8.18 Performance of various detectors at 19200 bit/s over channel D.

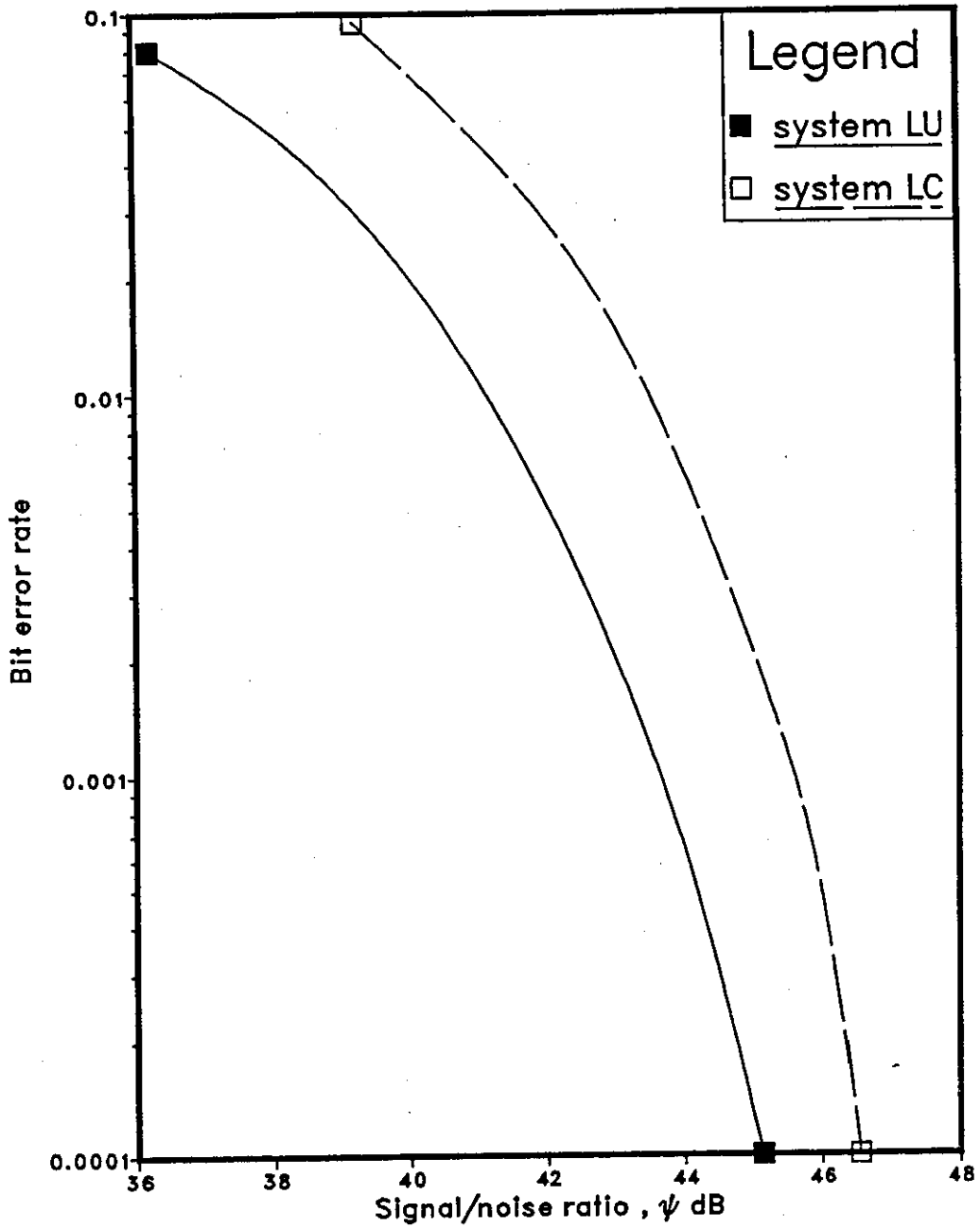


Fig.8.19 Performance of linear equalizers at 19200 bit/s over channel D.

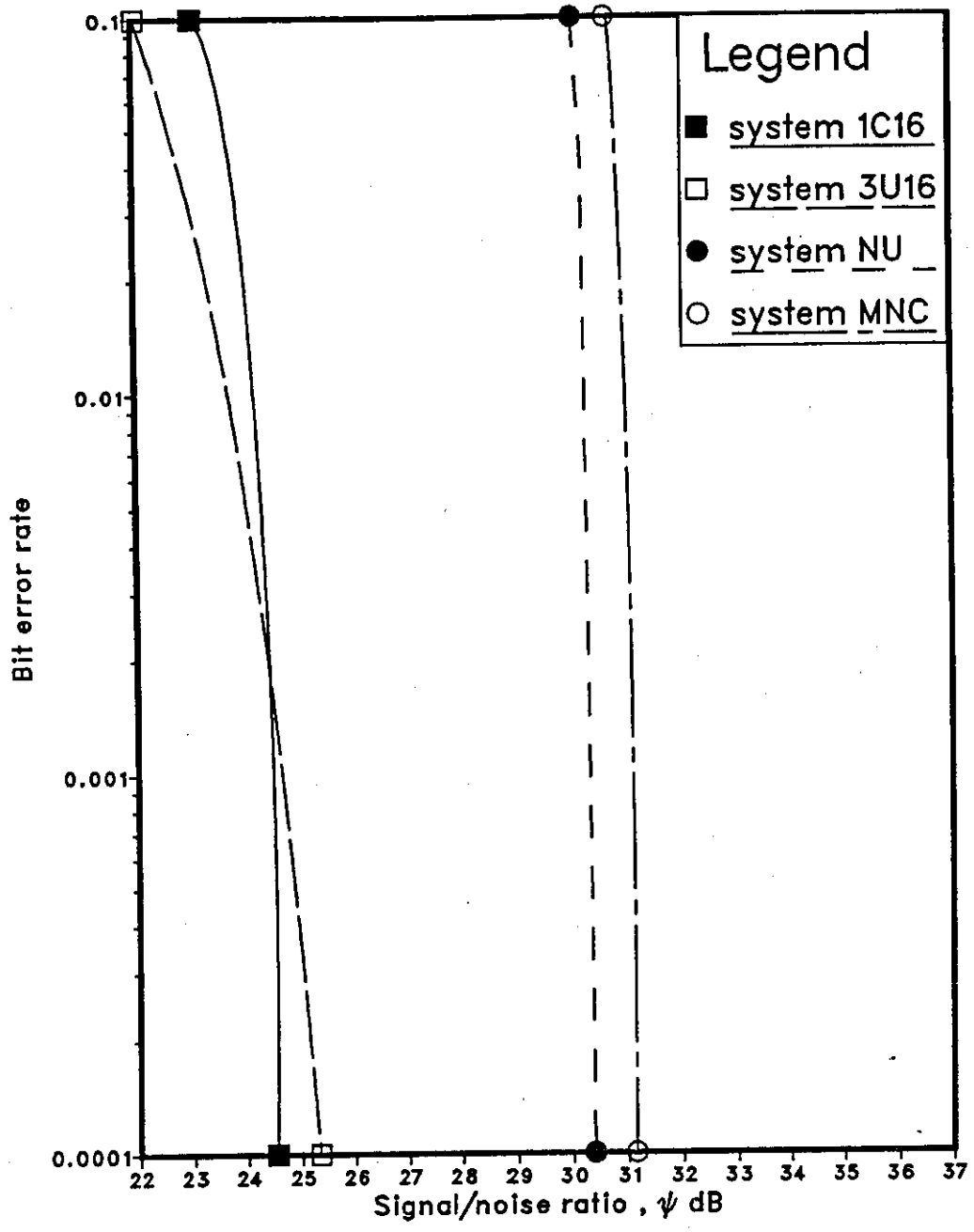


Fig.8.20 Performance of various detectors at 19200 bit/s over channel E.

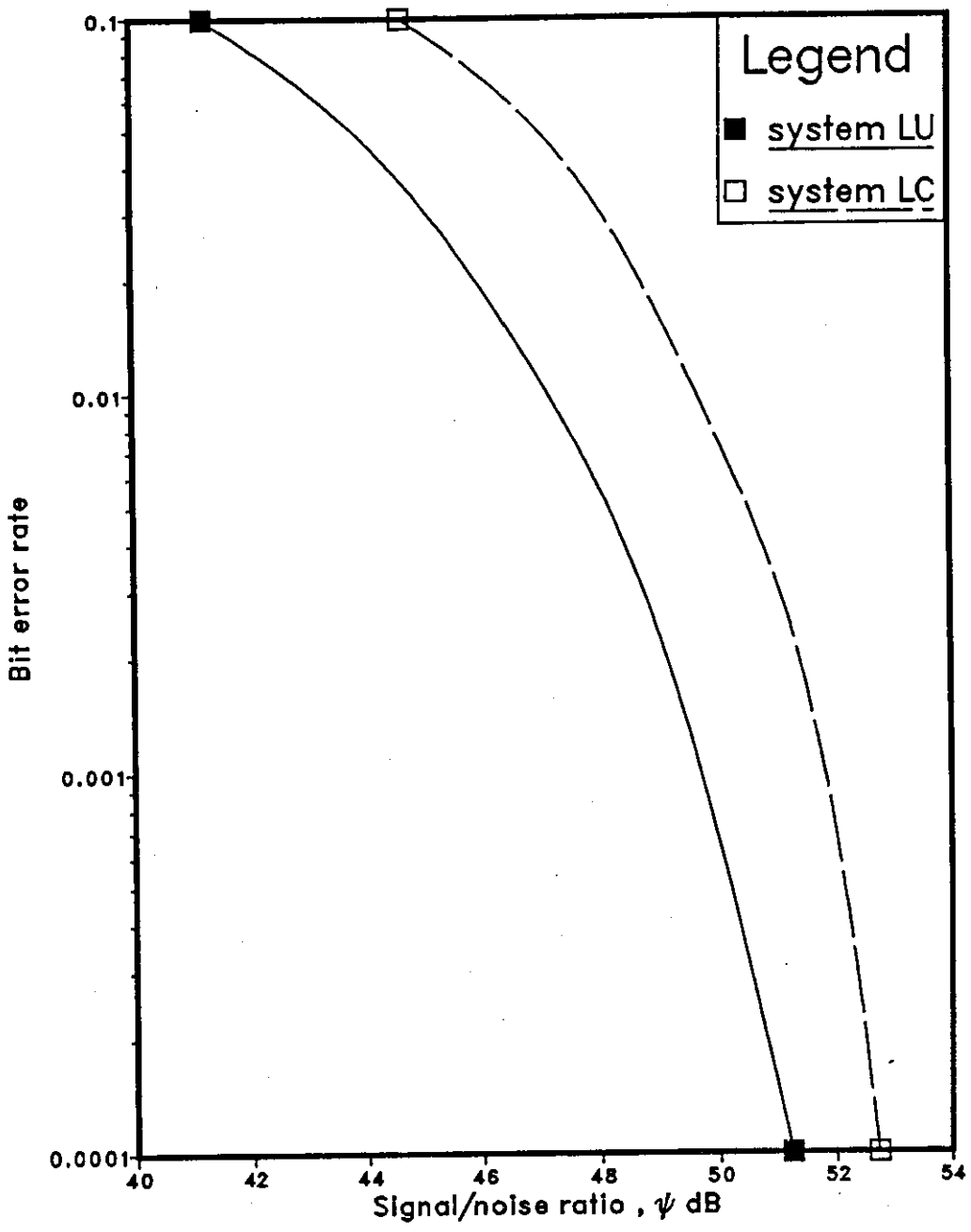


Fig.8.21 Performance of linear equalizers at 19200 bit/s over channel E.

CHAPTER 9

CONCLUSION

9.1 SUGGESTIONS FOR FURTHER WORK

It has been assumed in this thesis that the channel is known and time invariant. However, some of the systems (detectors) developed here appear to be well suited for use over a time-varying channel. These systems could, therefore, be studied further for use over a time-varying channel.

The proposed arrangements are tested here for the case where QAM signals are transmitted over telephone lines. Further tests may be carried out on the promising system for other modulation schemes or when the data signal is transmitted over other transmission paths, such as HF or satellite links.

The interleaving arrangement studied in this thesis is quite simple. This could be studied further and modified such that an improvement in the performance of the coded system can be achieved over channels that introduce severe distortion.

The detailed hardware designs of the more promising systems (detectors) developed in this thesis could be produced to assess the cost effectiveness of these systems for use in a synchronous serial data transmission systems.

9.2 CONCLUSION

For the convolutionally coded and distorted signal, a detector employing the maximum likelihood detection (implemented by means of the Viterbi algorithm detector), which takes account of both the coding and distortion, achieves the best tolerance to additive white Gaussian noise, but it suffers from an unduly complex implementation in practice.

Near-maximum likelihood detectors have been proposed. These detectors take account of both the coding and the distortion introduced by the channel. Some arrangements of these (such as system 3C with 64 vectors) come close to achieving the performance of the optimum detector (the Viterbi algorithm detector). The complexity of the proposed detectors is considerably less than that required by the corresponding Viterbi algorithm detector.

The results of the study show that the use of a convolutionally encoded signal together with a linear or nonlinear (decision feedback) equalizer and Viterbi algorithm detector (known in the thesis as systems LC, NC and MNC), does not appear to be very promising arrangement. The performance of such systems can be considerably inferior to that of the corresponding uncoded signal and an appropriate near-maximum likelihood detector at the receiver.
The poor performance of the arrangement which uses the linear equalizer (system LC) is at least partly due to the noise correlation, which is introduced by the linear equalizer between the neighbouring noise samples in the received signal. On the other hand and when the nonlinear equalizer is used with the coded signal (system NC or MNC), relatively longer error bursts occur over the channels that introduce a high level of amplitude distortion.

The preferred arrangement with a coded signal is system 1C (of Chapter 5) with 16 stored vectors (system 1C16). This system (detector) has a better tolerance to noise (at high signal/noise ratios) than the preferred arrangement with an uncoded signal,

which is system 3U (of Chapter 4) with 16 vectors (system 3U16). System 1C16 also give a better performance than system 3U16 when the data is transmitted at rates of 14400, 16000 and 19200 bit/s.

Unlike the case with the uncoded signal, there does not appear to be much purpose in using a near-maximum likelihood detector operating with less than 16 vectors for the coded signal. The performance of such a system is at best only a little better than that of the preferred system with an uncoded signal (system 3U16).

The simple interleaving arrangement, described in Chapter 7, can improve the performance of the coded system. Over channels which introduce low and typical levels of attenuation distortion, such an arrangement can achieve a performance which is close to that of the best near-maximum likelihood detector proposed here (system 3C64). However, the performance of such an arrangement is not encouraging over channels which introduce higher levels of signal distortion.

APPENDIX A

MODEL OF DATA TRANSMISSION SYSTEM FOR QAM SIGNAL

Fig. A.1 shows the detailed model of the data transmission system for the case where a QAM signal is transmitted over a linear bandpass channel. The two multipliers at the transmitter form a linear modulator, while the two multipliers together with the receiver lowpass filters form a linear demodulator. The white Gaussian noise in Fig. A.2 is real valued and has a two-sided power spectral density of $\frac{1}{2}N_0$. Each of the lowpass filters in the transmitter has an impulse response c(t) and transfer function C(f), and each of the two lowpass filters in the receiver has an impulse response f(t) and transfer function of F(f). The two bandpass filters together with bandpass transmission path form a linear bandpass channel, with transfer function L(f).

For an M-level QAM signal, $s_{i,0}$ and $s_{i,1}$ are $\sqrt{M}$ -level polar signals. The signal at the output of the adder (at the transmitter) is the real-valued waveform q(t),

$$q(t) = \sqrt{2}\sum_i s_{i,0}\, c(t - iT)\cos(2\pi f_c t)$$

$$-\sqrt{2}\sum_i s_{i,1}\, c(t - iT)\sin(2\pi f_c t) \qquad \ldots \quad A.1$$

where f_c is the carrier frequency in Hz and T is the modulation interval in seconds. It can be shown [21] that the signal q(t) can be written as

$$q(t) = \frac{1}{\sqrt{2}}\sum_i s_i c(t - iT)\exp(j2\pi f_c t)$$

$$+\frac{1}{\sqrt{2}}\sum_i s_i^* c(t - iT)\exp(-j2\pi f_c t) \qquad \ldots \quad A.2$$

where s_i^* is the complex conjugate of s_i , $j = \sqrt{-1}$ and

$$s_i = s_{i,0} + j\, s_{i,1} \qquad \ldots \quad A.3$$

Since the Fourier transform of $s_i c(t-iT)$ is $s_i C(f)\exp(-j2\pi fiT)$, the Fourier transform of $\frac{1}{\sqrt{2}}s_i c(t-iT)\exp(j2\pi f_c t)$ is $\frac{1}{\sqrt{2}}s_i C(f-f_c)\exp(-j2\pi(f-f_c)iT)$ and the Fourier transform of $\frac{1}{\sqrt{2}}s_i^* c(t-iT)\exp(-j2\pi f_c t)$ is $\frac{1}{\sqrt{2}}s_i^* C(f-f_c)\exp(-j2\pi(f+f_c)iT)$, the Fourier transform of q(t) is given by [21],

$$Q(f)=\frac{1}{\sqrt{2}}\sum_i s_i\, C(f-f_c)\exp(-j2\pi(f-f_c)iT)$$

$$+\frac{1}{\sqrt{2}}\sum_i s_i^*\, C(f+f_c)\exp(-j2\pi(f+f_c)iT) \qquad \ldots \quad A.4$$

The Fourier transform of the received signal at the output of the receiver bandpass filter in the absence of noise, is

$$Z(f)=Q(f)L(f) \qquad \ldots \quad A.5$$

where L(f) is the transfer function of the bandpass transmission channel, formed by the two bandpass filters and bandpass transmission path in Fig. A.1.

If imaginary values are now ascribed to the signals following the lower multiplier in the demodulator of Fig. A.1, the two multipliers in the demodulator together multiply the signal z(t) (at the output of the receiver bandpass filter) by

$$\sqrt{2}\cos(2\pi f_c t+\phi)-j\sqrt{2}\sin(2\pi f_c t+\phi)$$

$$=\sqrt{2}\exp(-j(2\pi f_c t+\phi)) \qquad \ldots \quad A.6$$

where ϕ is the carrier phase at the demodulator.

The demodulator can therefore be represented by Fig. A.2, where the lowpass filter is the same as each receiver lowpass filter in Fig. A.1 [21]. When there is no phase difference between the carriers at the modulator and demodulator the value of ϕ is zero and this will be assumed here.

The multiplication of z(t) by $\sqrt{2}\exp(-j2\pi f_c t)$ causes Z(f) to be shifted in the negative direction by f_c Hz and hence to be replaced by $Z(f+f_c)$.

Let $Z_0(f)$ be the low frequency component of $Z(f+f_c)$, and since in the absence of noise

$$Z(f+f_c)=Q(f+f_c)\,L(f+f_c) \qquad \ldots \quad A.7$$

it follows that

$$Z_0(f) = Q_0(f) L_0(f) \quad \ldots \; A.8$$

where $Q_0(f)$ and $L_0(f)$ are the low frequency components of $Q(f+f_c)$ and $L(f+f_c)$, respectively. Thus the Fourier transform P(f) of the demodulated baseband signal p(t) at the output of the lowpass filter in Fig. A.2 is given by [21],

$$P(f) = \sqrt{2} Z_0(f) F(f) \quad \ldots \; A.9$$

Using Eqn. A.8 gives

$$P(f) = \sqrt{2} Q_0(f) L_0(f) F(f) \quad \ldots \; A.10$$

Changing the variable f in Eqn. A.4 to $(f+f_c)$, gives

$$Q(f+f_c) = \frac{1}{\sqrt{2}} \sum_i s_i C(f) \exp(-j2\pi fiT)$$

$$+ \frac{1}{\sqrt{2}} \sum_i s_i^* C(f+2f_c) \exp(-j2\pi(f+2f_c)iT) \quad \ldots \; A.11$$

Thus

$$Q_0(f) = \frac{1}{\sqrt{2}} \sum_i s_i C(f) \exp(-j2\pi fiT) \quad \ldots \; A.12$$

and so Eqn. A.10 become

$$P(f) = \sum_i V(f) \exp(-j2\pi fiT) \quad \ldots \; A.13$$

where

$$V(f) = C(f) L_0(f) F(f) \quad \ldots \; A.14$$

the demodulated baseband waveform in the absence of noise is given by

$$p(t) = \sum_i \delta(t - iT) * v(t) = \sum_i s_i v(t - iT) \quad \ldots A.15$$

where * indicates convolution and V(f) is the Fourier transform of v(t), which assumed here to be time invariant. Furthermore, Eqn. A.14 can be written as

$$V(f) = C(f) D_0(f) H_0(f) E_0(f) F(f) \quad \ldots \; A.16$$

where $D_0(f), H_0(f)$ and $E_0(f)$ are the low frequency components of $D(f+f_c), H(f+f_c)$ and $E(f+f_c)$, respectively. It follows that a linear bandpass transmission path, proceeded by a linear modulator at the transmitter and followed by a linear demodulator at the receiver can be considered as the corresponding baseband channel. The transfer function of the baseband channel can be obtained as follows. The frequency characteristic of each of the separate bandpass filters or channels is converted into the corresponding baseband characteristic, by deleting the negative part of the characteristic and shifting the positive frequency part down in frequency by f_c Hz, to give a baseband characteristic, which is the required transfer function.

Fig. A.3 shows the baseband model of the QAM system of Fig. A.1. The transfer function of the transmitter filter A(f) and the receiver filter B(f) in Fig. A.3 are given by

$$A(f) = C(f) D_0(f) \quad \ldots \quad A.17$$

and

$$B(f) = E_0(f) F(f) \quad \ldots \quad A.18$$

and the transfer function of the transmission path in Fig. A.3 is given by $H_0(f)$. So the transfer function of the linear baseband channel in Fig. A.3 may be written as

$$V(f) = A(f) H_0(f) B(f) \quad \ldots \quad A.19$$

It is assumed that the impulse response of the transmitter and receiver lowpass filters in Fig. A.1 are real valued, so they introduce no coupling between the in-phase and quadrature channels. But, since the bandpass filters (and the channel) in Fig. A.1 can introduce coupling between the in-phase and quadrature channels, so also can the transmitter and the receiver filters in Fig. A.3. Thus the impulse response of each of these filters may be complex valued, and so may the impulse response v(t) of the linear baseband channel in Fig. A.3.

It has been assumed in the previous analysis that there is no additive noise. In the presence of the noise the received waveform p(t) at the output of the linear baseband channel in Fig. A.3 is given by

$$p(t) = \sum_i v(t - iT) + u(t) \quad \ldots \quad A.19$$

where u(t) is bandlimited complex valued Gaussian noise waveform.
The two-sided power spectral density $\frac{1}{2}N_0$ of the real-valued white Gaussian noise n(t) in Fig. A.1 is increased to N_0 for the complex-valued white Gaussian noise in Fig. A.3, because in the demodulation process, the noise is multiplied by $\sqrt{2}\exp(-j2\pi f_c t)$ (where ϕ =0 in Eqn. A.6) which doubles the noise power spectral density.

Since the impulse response of the receiver lowpass filters in Fig. A.1 are real valued, it can be shown [21,22,39,93] that, if the absolute value of the transfer function of the receiver bandpass filter in Fig A.1 is symmetrical about the carrier frequency or else if the absolute value of the transfer function of the receiver filter in Fig. A.3 is an even function (symmetrical about f=0), any sample of the real part of u(t) and any sample of the imaginary part of u(t) must now be statistically independent Gaussian random variables with zero mean and fixed variance. The two sided power spectral density of each of the real and imaginary parts of the bandlimited Gaussian noise waveform u(t), at the output of the receiver lowpass filter in Fig. A.3 is $\frac{1}{2}N_0|B(f)|^2$. If ,in addition, the absolute value sequared of the transfer function of the receiver filter in Fig. A.3 satisfies Nyquist vestigial symmetry theorem, about the frequency 1/2T Hz, then the real parts of the noise samples $\{u_i\}$ are statistically independent, as are the imaginary parts [20]. The above conditions are assumed to be hold throughout the thesis.

The waveform p(t) in Fig. A.1 is sampled once per data symbol, at the time instants $\{iT\}$ to give the received samples $\{p_i\}$, where

$$p_i = \sum_{h=0}^{K} s_{i-h} v_h + u_i \qquad \ldots \quad A.20$$

and $p_i = p(iT)$, $v_h = v(hT)$ and $u_i = u(ih)$.
Here the delay in transmission is neglected so that $v_0 \neq 0$ and v(hT)=0 fr h>K . Thus the sampled impulse response of the linear baseband channel in Fig. A.3 is given by (K+1)-component vector

$$V = [v_0 \quad v_1 \quad v_2 \quad . \quad . \quad . \quad . \quad v_K] \qquad \ldots \quad A.21$$

The transmitter and receiver filters are normally designed to have a bandwidth close to 1/2T Hz, so that the sampling rate of 1/T samples/s is quite close to the Nyquist

rate for p(t). This ensures that most of the information carried by the received waveform p(t) is contained also in the samples $\{p_i\}$, so that there is no serious loss in performance induced through operating on the samples $\{p_i\}$ in place of p(t).

It is shown in this appendix that the QAM system can be represented as a linear baseband model. This leads to a simple representation of the data transmission system and therefore it eases any attempt to improve the performance of the system, which is usually carried out by the computer simulation as is the case in this work.

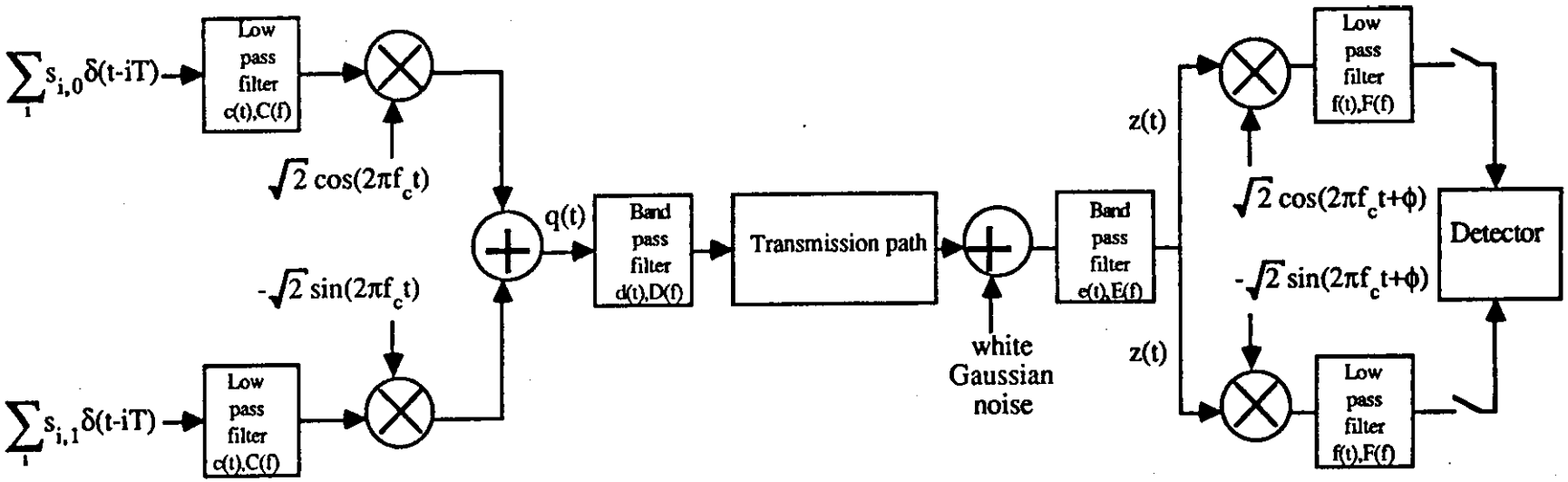


Fig.A.1 Bandpass model of QAM system.

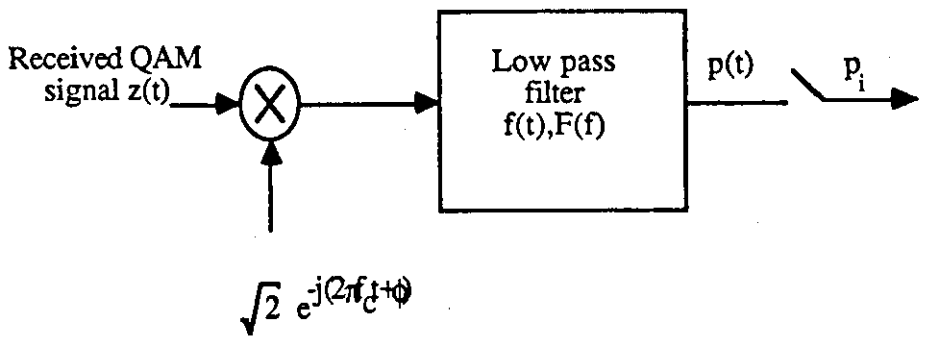


Fig.A.2 Alternative representation of the demodulator in Fig.A.1.

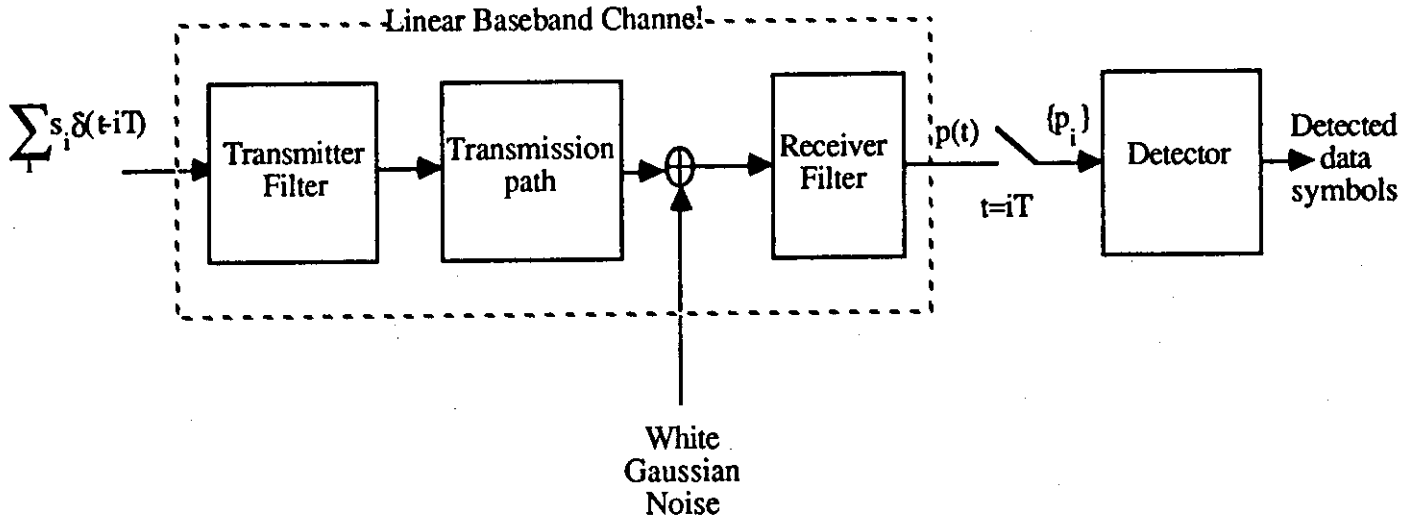


Fig.A.3 Baseband model of data transmission system.

APPENDIX B

THE EFFECT OF PHASE ROTATION ON THE UNCODED SIGNAL

In this section, the effect of phase rotation in the uncoded signal is examined. The transmitted data symbol may take any one of the 16 possible values, as shown in Fig. 3.2. The information digits are generated randomly and then differentialy encoded according to Section 3.1. The resulting coded digits are then mapped into the corresponding data symbols and transmitted over the linear baseband channel. Here it is assumed that the channel introduces no distortion or noise, so that the signal/noise ratio is infinity. Phase changes of integral multiples of $\pi/2$ of the received signal as compared to the transmitted signal element,are introduced by the transmission path.

Table B.1 shows the transmitted information digits, the transmitted data symbols, the phase introduced by the transmission path and the received samples. The differentialy decoded digits and the number of erroneous digits in the detection (decoding) of each data symbol is also shown in Table B.1. It is clear from this table that errors occur only in two successive symbols following the phase rotation, furthermore, these errors effect the detection (decoding) of the first two digits only.

I	Data bits 1234	Data symbol Re.	Im.	Phase rotation Deg.	Received sample Re.	Im.	Decoded bits 1234	No. of incorrect bits
1	1000	-3	3	0	-3	3	1000	0
2	1101	3	-1	0	3	-1	1101	0
3	0000	3	-3	0	3	-3	0000	0
4	1100	-3	3	0	-3	3	1100	0
5	0000	-3	3	90	3	3	1000	1
6	0101	-1	-3	0	-1	-3	1101	1
7	0001	-1	-3	0	-1	-3	0001	0
8	1000	-3	3	0	-3	3	1000	0
9	0111	-1	-1	0	-1	-1	0111	0
10	0010	-3	-1	0	-3	-1	0010	0
11	1000	-3	3	0	-3	3	1000	0
12	1000	3	3	0	3	3	1000	0
13	1101	-1	-3	0	-1	-3	1101	0
14	1100	3	3	0	3	3	1100	0
15	0010	3	1	180	-3	-1	1110	2
16	0100	-3	3	0	-3	3	1000	2
17	1101	3	-1	0	3	-1	1101	0
18	1011	-1	-1	0	-1	-1	1011	0
19	1100	3	3	0	3	3	1100	0
20	1001	3	-1	0	3	-1	1001	0
21	1001	-1	-3	0	-1	-3	1001	0
22	1111	1	1	0	1	1	1111	0
23	1010	1	-3	0	1	-3	1010	0
24	0101	1	3	0	1	3	0101	0
25	0100	-3	3	0	-3	3	0100	0
26	0101	-1	-3	0	-1	-3	0101	0
27	1110	3	1	0	3	1	1110	0
28	1010	1	-3	0	1	-3	1010	0
29	1110	-1	3	0	-1	3	1110	0
30	1101	3	-1	-90	1	3	0101	1
31	1101	-3	1	0	-3	1	0101	1
32	0110	-3	-1	0	-3	-1	0110	0
33	1110	3	1	0	3	1	1110	0
34	0011	1	1	0	1	1	0011	0
35	1111	-1	-1	0	-1	-1	1111	0
36	1000	-3	3	0	-3	3	1000	0
37	1111	1	-1	0	1	-1	1111	0
38	1011	-1	-1	0	-1	-1	1011	0
39	0110	1	-3	0	1	-3	0110	0
40	0100	3	3	0	3	3	0100	0

Table B.1 The effect of phase rotation in the uncoded system

APPENDIX C

THE EFFECT OF PHASE ROTATION ON THE CONVOLUTIONALLY ENCODED SIGNAL

The effect of phase rotation of the encoded signal is examined in this section. The transmitted digits $\{\alpha_{i,h}\}$ are generated randomly and then convolutionally encoded to give the corresponding encoded data symbols $\{s_i\}$ according to Section 3.2. The initial state of the encoder is taken to be (000). As in Appendix B, the only effect of the transmission path is the introduction of phase rotations of integral multiples of $\pi/2$ of the received signal as compared to the transmitted signal.
The decoder here is the Viterbi algorithm detector (VA) (Chapter 5)

As can be seen from Table C.1, any phase rotation of $\pi/2$ or its multiple results in an incorrect detection (decoding) of the first two data digits only.

I	Data bits 1234	Data symbol Re.	Im.	Phase rotation Deg.	Received sample Re.	Im.	Decoded bits 1234	No. of incorrect bits
1	1000	1	1	0	1	1	1000	0
2	1101	5	-1	0	5	-1	1101	0
3	0000	1	1	0	1	1	0000	0
4	1100	-3	1	0	-3	1	1100	0
5	0000	1	-1	90	-1	-1	1100	2
6	0101	-5	1	0	-5	1	0101	0
7	0001	-3	-5	0	-3	-5	0001	0
8	1000	-1	1	0	-1	1	1000	0
9	0111	-5	-1	0	-5	-1	0111	0
10	0010	3	3	0	3	3	0010	0
11	1000	1	1	0	1	1	1000	0
12	1000	1	-1	0	1	-1	1000	0
13	1101	1	5	0	1	5	1101	0
14	1100	1	3	0	1	3	1100	0
15	0010	-3	-3	180	3	3	1010	1
16	0100	3	-1	0	3	-1	1000	2
17	1101	-1	-5	0	-1	-5	0101	1
18	1011	5	3	0	5	3	0011	1
19	1100	-3	1	0	-3	1	1100	0
20	1001	-3	-5	0	-3	-5	1001	0
21	1001	5	-3	0	5	-3	1001	0
22	1111	-1	5	0	-1	5	1111	0
23	1010	-3	-3	0	-3	-3	1010	0
24	0101	5	-1	0	5	-1	0101	0
25	0100	1	3	0	1	3	0100	0
26	0101	-5	1	0	-5	1	0101	0
27	1110	3	1	0	3	1	1110	0
28	1010	3	3	0	3	3	1010	0
29	1110	-3	-1	0	-3	-1	1110	0
30	1101	-1	-5	0	-1	-5	1101	0
31	1101	-5	1	0	-5	1	1101	0
32	0110	-1	3	0	-1	3	0110	0
33	1110	1	-3	0	1	-3	1110	0
34	0011	5	3	0	5	3	0011	0
35	1111	1	-5	-90	5	1	0011	2
36	1000	1	1	0	1	1	1000	0
37	1111	-1	5	0	-1	5	1111	0
38	1011	-3	5	0	-3	5	1011	0
39	0110	-3	-1	0	-3	-1	0110	0
40	0100	1	3	0	1	3	0100	0

Table C.1 The effect of phase rotation in the coded system.

APPENDIX D

DERIVATION OF THE SAMPLED IMPULSE RESPONSE FROM THE ATTENUATION AND GROUP-DELAY CHARACTERISTICS

The impulse responses are derived here with respect to the carrier frequency 1800 Hz. The values of the attenuation and group-delay of the equipment filters and the telephone circuit are read from the corresponding graphs (Chapters 3, 6 and 8) every 50 Hz over the frequency range 50 - 3750 Hz. The sample values of the attenuation and group-delay for the telephone circuit and the equipment filters are added together for each frequency. The combined response is then normalized so that the attenuation and group delay at the carrier frequency are both zero. The amplitude and phase angle at each frequency are then calculated from the combined response as follows [86].

The amplitude is given by

$$A = 10^{-\frac{ATT}{20}} \qquad \ldots \quad D.1$$

where A is the value of the amplitude and ATT is the value of the attenuation in dB, which is given by the frequency characteristics of the combined response.

The phase angle at each frequency is derived with respect to the phase angle at the carrier frequency, which is taken to be zero here. The relation between the phase angle and the group delay is given by [24]

$$MGD = -\frac{d\phi}{d\omega} \qquad \ldots \quad D.2$$

where MGD is the mean group delay and ϕ is the phase angle and $\omega = 2\pi f$.

Now Eqn. D.2 can be expressed numerically as

$$MGD = (\phi_{i-1} - \phi_i)/2\pi\Delta f \qquad \ldots \quad D.3$$

where Δf is 50 Hz and, ϕ_i and ϕ_{i-1} are the phase angle at the frequencies f_i and f_{i-1} ,respectively.

Eqn. D.3 can be written as

$$\phi_i = \phi_{i-1} - 2\pi.\Delta f.MGD \qquad \ldots \quad D.4$$

which gives the value of the phase angle ϕ_i at frequency f_i assuming the phase angle ϕ_{i-1} at f_{i-1} is known.

The definition of the mean group delay MGD is

$$MGD = (d_i + d_{i-1})/2 \qquad \ldots \quad D.5$$

where d_i and d_{i-1} are the values of the group delay (in seconds) at the frequencies f_i and f_{i-1} , respectively. These values are available from the group-delay characteristics. As a result Eqn. D.4 becomes

$$\phi_i = \phi_{i-1} - \pi.\Delta f.(d_i + d_{i-1}) \qquad \ldots \quad D.6$$

Thus the phase angles at frequencies above and below the carrier frequency can be derived from the group delay by using Eqn. D.6, where the phase angle at the carrier frequency is zero.

The resultant complex valued frequency response of the telephone circuit and the equipment filter are then determined by calculating the values of the real and imaginary components of the response at each frequency, as follows

$$R_i = A_i \cos(\phi_i) \qquad \ldots \quad D.7$$

$$I_i = A_i \sin(\phi_i) \qquad \ldots \quad D.8$$

where A_i and ϕ_i are determined by Eqn. D.1 and D.6, respectively, for the given value of frequency (f_i). In the above equations R_i and I_i are the real and imaginary values of the resultant frequency response, respectively. These values are then formatted for the inverse discreet Fourier transform (IDFT) routine. To ensure finely spaced samples of the calculated impulse response, the frequency range is extended upto 48 kHz by assuming the amplitude of all the values outside the known range to be zero. The resultant impulse response is then sampled at the required rate, which is 2400 sample/s for the 9600 and 14400 bit/s systems, and 3200 sample/s for the 16000 and 19200 bit/s systems.

NOTE

It was found, in the final stage of the work,that the sampled impulse responses of the linear baseband channel derived here are different from that of the actual responses. However, the difference is only in the sign of the imaginary parts of the responses, so that the sampled impulse responses presented in this thesis are the complex conjugates of the actual responses. To avoid any change in the sampled impulse responses, it is assumed here that the negative sign of the term $-\sqrt{2}\sin(2\pi f_c t)$ in the model of QAM system described in Appendix A is changed into plus sign. The corresponding changes in Figs. A.1 and A.3 (Appendix A) of the above sign are also assumed here.

APPENDIX E

STATISTICAL PROPERTIES OF THE NOISE SAMPLES AT THE OUTPUT OF THE ADAPTIVE LINEAR FILTER

In this appendix, the statistical properties of the noise samples at the output of the adaptive linear filter, are examined, for the case where the noise samples at its input are statistically independent Gaussian random variables with zero mean and fixed variance.

Let the sampled impulse response of the filter be given by the (n+1)-component vector

$$D = [d_0 \quad d_1 \quad d_2 \quad . \quad . \quad . \quad . \quad d_n] \qquad \ldots \quad E.1$$

where $\{d_k\}$ are the tap gains of the filter, and they are complex-valued quantities in general.

The properties of the filter are described in Chapters 2 and 3. For the ideal operation of the filter the number of its tap gains is infinity, and so the value of n here is assumed to be very large. It is also assumed throughout the work that the gain introduced by the filter is adjusted such that the first component y_0 of the sampled impulse response of the linear baseband channel and the filter is unity (Chapter 3). This assumption is ignored in the following analysis, and it is assumed that the filter introduces no gain or attenuation, thus

$$|D|^2 = \sum_k d_k d_k^* = 1 \qquad \ldots \quad E.2$$

where $|D|$ is the unitary length of the vector D.

It has been shown [22], that under this condition and when the filter is ideally adjusted, if the sequence D is reversed in time, the reversal being pivoted about its component at time t=0 and if each component is replaced by its complex conjugate, the resultant sequence is the inverse of the original sequence, and so

$$\sum_k d_k d_{k+h}^* \approx 0 \qquad \textit{for} \quad h \neq 0 \qquad \ldots \quad E.3$$

Also Eqn. E.3 implies that

$$Re.\left(\sum_k d_k d_{k+h}^*\right) \approx 0 \qquad for \quad h \neq 0 \qquad \ldots \quad E.4$$

and

$$Im.\left(\sum_k d_k d_{k+h}^*\right) \approx 0 \qquad for \quad h \neq 0 \qquad \ldots \quad E.5$$

where Re.(x) and Im.(x) are the real and imaginary components of the quantity x, respectively.

The real and the imaginary components of the noise samples $\{u_i\}$ at the filter input are statistically independent Gaussian random variables with zero mean and variance σ^2, and therefore they are uncorrelated [87], thus

$$E[u_i] = E[u_{i,0}] = E[u_{i,1}] = 0 \qquad \ldots \quad E.6$$

where E[x] is the expected value of x, and $u_{i,0}$ and $u_{i,1}$ are the real and imaginary components of the noise sample u_i, respectively.

Also

$$\begin{aligned} E[u_{i,0}.u_{i-h,0}] &= 0 \qquad for \quad h \neq 0 \\ &= \sigma^2 \qquad for \quad h = 0 \end{aligned} \qquad \ldots \quad E.7$$

$$\begin{aligned} E[u_{i,1}.u_{i-h,1}] &= 0 \qquad for \quad h \neq 0 \\ &= \sigma^2 \qquad for \quad h = 0 \end{aligned} \qquad \ldots \quad E.8$$

$$E[u_{i,1}.u_{i-h,0}] = 0 \qquad for \quad \text{all} \quad h \qquad \ldots \quad E.9$$

and

$$\begin{aligned} E[u_i.u_i^* - h] &= 0 \qquad for \quad h \neq 0 \\ &= 2\sigma^2 \qquad for \quad h = 0 \end{aligned} \qquad \ldots \quad E.10$$

where * indicates a complex conjugate.

The noise samples at the output of the filter, at time t=iT, is given by

$$w_i = \sum_k u_{i-k} d_k \qquad \ldots \quad E.11$$

The mean value of $\{w_i\}$ is given by

$$\begin{aligned} E[w_i] &= E\left[\sum_k u_{i-k} d_k\right] \\ &= \sum_k d_k . E[u_{i-k}] \\ &= 0 \end{aligned} \qquad \ldots \quad E.12$$

So, the mean value of $\{w_i\}$ is zero. Also, it can be shown that each of the real and imaginary components of the samples $\{w_i\}$ have zero mean.
The variance of $\{w_i\}$ is given by

$$\begin{aligned} E[w_i . w_i^*] &= E\left[\left(\sum_k u_{i-k} d_k\right).\left(\sum_k u_{i-k} d_k\right)^*\right] \\ &= E[(u_i d_0 + u_{i-1} d_1 + \quad . \quad . \quad . \quad + u_{i-n} d_n) \\ &\times (u_i^* d_0^* + u_{i-1}^* d_1^* + \quad . \quad . \quad . \quad + u_{i-n}^* d_n^*)] \end{aligned} \qquad \ldots \quad E.13$$

and by using the properties of the noise samples $\{u_i\}$ given in Eqns. E.7 to E.10, the above equation become

$$\begin{aligned} E[w_i . w_i^*] &= 2\sigma^2 . \sum_k d_k d_k^* \\ &= 2\sigma^2 \end{aligned} \qquad \ldots \quad E.14$$

So, the variance of $\{w_i\}$ is $2\sigma^2$. It can be shown, in a similar way, that the variance of each of the real and imaginary components of the noise samples $\{w_i\}$ is σ^2.

To prove that the real and imaginary components of the noise samples $\{w_i\}$ are statistically independent random variables, and therefore they are uncorrelated, it is sufficient to show that for any non zero integer h [87],

$$E[w_{i,0} . w_{i-h,0}] = 0 \qquad \ldots \quad E.15.a$$

,

$$E[w_{i,0} . w_{i-h,1}] = 0 \qquad \ldots \quad E.15.b$$

and

$$E[w_{i,1}.w_{i-h,1}]=0 \qquad \ldots \quad E.15.c$$

where $w_{i,0}$ and $w_{i,1}$ are the real and imaginary components of w_i , respectively, and w_{i-h} is the noise sample at the output of the filter at time (i-h)T, which is given by

$$w_{i-h}=\sum_k u_{i-h-k}d_k \qquad \ldots \quad E.16$$

The left hand side of Eqn. E.15.a can be written as

$$E[w_{i,0}.w_{i-h,0}]$$

$$=E\left[Re.\left(\sum_k u_{i-k}d_k\right).Re.\left(\sum_k u_{i-h-k}d_k\right)^*\right]$$

$$=E[(u_{i,0}d_{0,0}-u_{i,1}d_{0,1}+u_{i-1,0}d_{1,0}-u_{i-1,1}d_{1,1}\ldots+u_{i-n,0}d_{n,0}-u_{i-n,1}d_{n,1})\times$$

$$(u_{i-h,0}d_{0,0}-u_{i-h,1}d_{0,1}+u_{i-h-1,0}d_{0,1}-u_{i-h-1,1}d_{1,1}\ldots+u_{i-h-n,0}d_{n,0}-u_{i-h-n,1}d_{n,0})]$$

$$\ldots \quad E.17$$

where $d_{k,0}$ and $d_{k,1}$ are the real and imaginary components, respectively, of the (k+1)-th tap (d_k) of the filter.

By using the properties of the noise samples $\{u_i\}$ and Eqn. E.4, Eqn. E.17 becomes

$$E[w_{i,0}.w_{i-h,0}]$$

$$=E[(u^2_{i-h,0}d_{0,0}d_{h,0}+u^2_{i-h,1}d_{0,1}d_{h,1}+\quad u^2_{i-h-1,0}d_{1,0}d_{h+1,0}+$$

$$+u^2_{i-h-1}d_{1,1}d_{h+1,1}+\quad .\quad .\quad .\quad .\quad .\quad +u^2_{i-n,0}d_{n,0}+u^{2i}_{i-n,1}\quad d_{n-h,1}d_{n,1})]$$

$$=\sigma^2.Re.\left(\sum_k d_k d^*_{k+h}\right)$$

$$\approx 0 \qquad \ldots \quad E.18$$

so the real components of the noise samples w_i and w_{i-h} (for $h \neq 0$) are uncorrelated.

In a similar way and by using Eqn. E.5, it can be shown that the left hand side of Eqn. E.15.b is given by

$$E[w_{i,0}.w_{i-h,1}] = \sigma^2.Im.\left(\sum_k d_k d^*_{k+h}\right)$$

$$\approx 0 \qquad \ldots \quad E.19$$

so the real and imaginary components of the noise samples w_i and w_{i-h} at the output of the filter, for non zero values of h are uncorrelated. Also it can be shown that Eqn. E.19 is satisfied for h=0.

Again by using Eqn. E.4 and by following the above steps, the left hand side of Eqn. E.15.c can be written as

$$E[w_{i,1}.w_{i-h,1}] = \sigma^2.Re.\left(\sum_k d_k d_{k+h}^*\right)$$

$$\approx 0 \qquad \ldots \quad E.20$$

and so, the imaginary components of the noise samples w_i and w_{i-h} at the output of the filter, for non zero values of h, are uncorrelated.

It follows that, the real and imaginary components of the noise samples $\{w_i\}$ are statistically independent Gaussian random variables with zero mean and variance σ^2, just like the noise samples $\{u_i\}$.

So, the adaptive linear filter used in the model of data transmission system throughout the work does not change the statistical properties of the noise samples at its output.

APPENDIX F

SIGNAL/NOISE RATIO DEFINITION AND NOTES ON THE COMPUTER SIMULATION

The signal/noise ratio in the received samples $\{p_i\}$ at the output of the demodulator may be defined generally as

$$SNR = 10\log_{10}(K_r E[|s_i|^2]/E[|u_i|^2]) \quad \ldots \quad F.1$$

where E[x] is the expected value of x, s_i is the transmitted complex valued data symbol (coded or uncoded), K_r is constant determined by the resultant transfer function of the transmitter filter and u_i is the complex valued noise sample in p_i. The real and imaginary components of the noise samples $\{u_i\}$ are sample values of statistically independent Gaussian random variables with zero mean and fixed variance.

Let λ be the mean square value of the coded or uncoded data symbols $\{s_i\}$, then

$$\lambda = E[|s_i|^2]$$

$$= E[s_{i,0}^2] + E[s_{i,1}^2] \quad \ldots \quad F.2$$

where $s_{i,0}$ and $s_{i,1}$ are the real and imaginary components of s_i, respectively.

If the data transmission system transmits m information bits per modulation interval, the average transmitted energy per bit is given by

$$E_b = K_r.\lambda/m \quad \ldots F.3$$

where λ is given by Eqn. F.2.

The variance of each of the real and imaginary components of the noise samples $\{u_i\}$ in the received samples $\{p_i\}$ is given by [22,14]

$$\sigma^2 = \frac{1}{2}N_0.K_r \quad \ldots \quad F.4$$

where $\frac{1}{2}N_0$ is the two-sided power spectral density of the additive white Gaussian noise at the output of the telephone circuit (see Chapter 3), and K_r is a constant determined by the resultant transfer function of the receiver filters. Thus

$$\frac{E_b}{\frac{1}{2}N_0} = K_t.K_r.\frac{\lambda}{m.\sigma^2} \qquad \ldots \quad F.5$$

Since the values of K_t and K_r do not change from one telephone circuit to another for the particular transmission rate, the signal/noise ratio (measured in dB) is taken to be

$$\psi = 10\log_{10}\frac{\lambda}{m.\sigma^2} \qquad \ldots \quad F.6$$

The definition given by Eqn. F.6 is used throughout the work. Furthermore, the attenuation or gain introduced by the model of telephone circuit is adjusted such that the vector, which represents the sampled impulse response of the channel, has a unit gain.

The value of λ/m for an uncoded 16-level QAM signal is 2.5 and for a coded 32-level QAM signal is 5.0 (see the corresponding signal constellations in Chapter 3). In Chapter 8, the values of λ/m for different signal constellations used for transmitting the data at rates of 14400, 16000 and 19200 bit/s are given in Table 8.8.

All simulation programs used in this work are written in Fortran 77. The computer simulation tests carried out in this work employ the standard Numerical Algorithm Generator (NAG) random number generator subroutines to generate the information digits $\{\alpha_{i,k}\}$ and the noise samples. The information digits are generated with a uniform distribution, and the real and imaginary parts of the noise samples are generated as Gaussian random number generator with zero mean and a standard deviation σ. The latter is determined by the particular value of the signal/noise ratio ψ from the following relation (see Eqn. F.6)

$$\sigma^2 = \frac{\lambda}{m}.10^{-\frac{\psi}{10}} \qquad \ldots \quad F.7$$

In Chapter 7, the performance of the interleaving systems is given in terms of the probability of error events against the signal/noise ratio ψ. The error event defined as follows, following an incorrect decoded data symbols s_i, if N_e or more subsequent detected symbol are correct, the next incorrect detected symbol will be considered as the start of a new error event, otherwise this error will be taken as part of the present

error event. The selected value of N_e must be sufficiently large such that the first error in an error event is independent of all errors in the previous event. The value of N_e is taken here to be 32.

A total of typically 2.5×10^6 data symbols were transmitted in plotting any one curve that represents the error rate performance. This number is increased to around 1×10^7 in Chapter 7 for the interleaving systems. To increase the accuracy in the measured performance of the systems, the bit error rate is determined after a sufficient number (N_e) of independent error events occur at the particular value of the signal/noise ratio. It has been shown [21,94], when this number is greater than 30, the 95% confidence limits in the values of the error probability can be approximated by $p \pm 2p/\sqrt{N_e}$, where p is the probability of error. In the computer simulation tests, the value of N_e is taken to be around 40, whenever it is possible, and so the 95% confidence limits for the values of p are approximately $p(1 \pm 0.3)$.

APPENDIX G

LISTINGS OF COMPUTER PROGRAMS

G1 - SIMULATION OF SYSTEM 3U.

G2 - SIMULATION OF SYSTEM MNC.

G3 - SIMULATION OF SYSTEM 1C.

G4 - SIMULATION OF SYSTEM 2C.

G5 - SIMULATION OF SYSTEM 3C.

APPENDIX G1
THE FOLLOWING PROGRAM SIMULATES THE OPERATION OF SYSTEM 3U16 .
THE DATA IS AN UNCODED 16-LEVEL QAM SIGNAL AND THE INFORMATION
DIGITS ARE DIFFERENTIALLY ENCODED.

```
/*JOB S3U16,EUELAKK,ST=MFX,C=S,TI=1280,
/* PW=K
FTN5,DB=0/PMD,L=0.
LIBRARY,PROCLIB.
NAG(FTN5)
LGO.
####S
      PROGRAM S3U16
      REAL G05DDF,G05DAF,P,W1,W2
      REAL C1(16),C2(16),C3(8,32),C22(16),Z1(16),Z2(16),C11(8)
     C      ,E1(16),E2(16),E11(16),E22(16),Y1(20),Y2(20)
      INTEGER IS1(33),IS2(33),ISS1(16),ISS2(16),IX1(16,33)
     C         ,IX2(16,33),IXX1(24,33),IXX2(24,33),IXT1(16)
     C         ,IXT2(16),IXXX1(16,33),IXXX2(16,33),IA(33,4)
     C         ,IB(33,4),IAD(4),IBD(4),IPRES(16),IPREV(16)
     C         ,INA(16),KK(8),ISER,IBER,IEVA,IEVK

      DATA KK/4,4,2,2,1,1,1,1/
C     TABLE FOR DIFF. ENCODING /DECODING AND SIGNAL MAPPING.
      DATA INA/0,2,3,1,0,2,3,1,0,2,3,1,0,2,3,1/
      DATA IPREV/0,0,0,0,2,2,2,2,3,3,3,3,1,1,1,1/
      DATA IPRES/0,2,3,1,2,3,1,0,3,1,0,2,1,0,2,3/
      DATA ISS1/-3,-3,3,3,-3,-1,1,3,-1,-3,3,1,-1,-1,1,1/
      DATA ISS2/-3,3,-3,3,-1,3,-3,1,-3,1,-1,3,-1,1,-1,1/
C     LOAD THE SAMPLED IMPULSE RESPONSE OF TH CHANNEL.
      DATA   Y1/ 1.0000, 0.4861,-0.5980, 0.1702,-0.0245, 0.0100
     C          ,-0.0134, 0.0056, 0.0003,-0.0008, 0.0000, 0.0007
     C          , 0.0037,-0.0019, 0.0020, 0.0005,-0.0022, 0.0007
     C          ,-0.0008, 0.0005/
      DATA   Y2/ 0.0000, 1.0988, 0.0703,-0.1938, 0.1000,-0.0258
     C          , 0.0110,-0.0042, 0.0003, 0.0041,-0.0061,-0.0007
     C          , 0.0002,-0.0025, 0.0008,-0.0002, 0.0002,-0.0005
     C          , 0.0002, 0.0005/
C      SETTING OF INITIAL VALUES
       IBIT=4
       LEV=2**IBIT
       M=6
       IQ=23
       K=16
       L=100000
       N=33
       N1=N-1
       N2=N+1
       LN=L+N1
       SNR=17.5
       SNRD=0.25
       CALL G05CBF(IQ)
C      SCALING THE SMPLED IMPULSE RESPONSE.
       YY=0.0
       DO 2 I=1,20,1
```

```
      YY=YY+Y1(I)**2+Y2(I)**2
2     CONTINUE
      Y=SQRT(YY)
      DO 3 I=1,20,1
      Y1(I)=Y1(I)/Y
3     Y2(I)=Y2(I)/Y
      OPEN(1,FILE='OUTPUT')

C     MAIN DO LOOP (SELECT SNR)
      DO  1000 MM=1,M,1
      SNR=SNR+SNRD
      P=SQRT(2.5/(10.0**(SNR/10.0)))
C     SET INITIAL VALUES OF ALL ARRAYS
      IDPRES=0
      IDPREV=0
      DO 10 I=1,N,1
      ISER=0
      IBER=0
      IEVA=0
      IEVK=0
      IS1(I)=-3
      IS2(I)=-3
      DO 10 J=1,K,1
      IX1(J,I)=-3
10    IX2(J,I)=-3
      DO 20 I=2,K,1
20    C1(I)=100000.0

C     ZERO COST FOR THE FIRST VECTOR.
      C1(1)=0.0
      DO 30 I=1,N,1
      DO 30 J=1,IBIT,1
      IA(I,J)=0
30    IB(I,J)=0

C     SECOND DO LOOP (TRANSMISSION).
      DO 440 LLL=1,LN,1
C     SHIFT ALL ARRAYS
      DO 40 I=1,N1,1
      I1=I+1
      IS1(I)=IS1(I1)
      IS2(I)=IS2(I1)
      DO 40 J=1,K,1
      IX1(J,I)=IX1(J,I1)
40    IX2(J,I)=IX2(J,I1)
      DO 50 I=1,N1,1
      I1=I+1
      DO 50 J=1,IBIT,1
      IA(I,J)=IA(I1,J)
50    IB(I,J)=IB(I1,J)
C     CALCULATE ISI FOR ALL VECTORS
      DO 70 I=1,K,1
      ZZ1=0.0
      ZZ2=0.0
```

```
      DO 60 J=2,20,1
      J1=N-J+1
      ZZ1=ZZ1+REAL(IX1(I,J1))*Y1(J)-REAL(IX2(I,J1))*Y2(J)
60    ZZ2=ZZ2+REAL(IX1(I,J1))*Y2(J)+REAL(IX2(I,J1))*Y1(J)
      Z1(I)=ZZ1
70    Z2(I)=ZZ2
C     GENERATE INFORMATION DIGITS
      DO 80 I=1,IBIT,1
80    IA(N,I)=NINT(G05DAF(0.0,1.0))
C     DIFF. ENCODING AND SIGNAL MAPPING
      ITPREV=IB(N1,1)+2*IB(N1,2)
      ITINA=IA(N,1)+2*IA(N,2)
      DO 110 I=1,K,1
      IF(ITINA.EQ.INA(I)) THEN
      IF(ITPREV.EQ.IPREV(I)) THEN
      II=I
      ENDIF
      ENDIF
110   CONTINUE
      DO 111 I=3,IBIT,1
111   IB(N,I)=IA(N,I)
      IF(IPRES(II).LT.2) THEN
      IF(IPRES(II)) 112,112,113
112   IB(N,1)=0
      IB(N,2)=0
      GO TO 120
113   IB(N,1)=1
      IB(N,2)=0
      GO TO 120
      ELSE
      IF(IPRES(II)-3) 114,115,115
114   IB(N,1)=0
      IB(N,2)=1
      GO TO 120
115   IB(N,1)=1
      IB(N,2)=1
      ENDIF
120   IN=IPRES(II)+4*IA(N,3)+8*IA(N,4)
      IS1(N)=ISS1(IN+1)
      IS2(N)=ISS2(IN+1)
C     CALCULATE THE RECEIVED SAMPLE AND ADD THE NOISE
      R1=0.0
      R2=0.0
      DO 130 I=1,20,1
      J=N-I+1
      R1=R1+REAL(IS1(J))*Y1(I)-REAL(IS2(J))*Y2(I)
130   R2=R2+REAL(IS1(J))*Y2(I)+REAL(IS2(J))*Y1(I)
      W1=G05DDF(0.0,P)
      W2=G05DDF(0.0,P)
      R1=R1+W1
      R2=R2+W2
C     EXPAND EACH VECTOR INTO  ONE VECTOR
      DO 150 I=1,K,1
      E1(I)=R1-Z1(I)
      E2(I)=R2-Z2(I)
```

```
          E1(I)=E1(I)*Y
          E2(I)=E2(I)*Y
C         CALCULATE TH COST OF EACH VECTOR.
          D=100000.0
          DO 140 J=1,LEV,1
          D1=E1(I)-REAL(ISS1(J))
          D2=E2(I)-REAL(ISS2(J))
          DD=D1*D1+D2*D2
          IF(DD.LT.D) THEN
          D=DD
          JJ=J
          ENDIF
140       CONTINUE
          C2(I)=C1(I)+D
          IXT1(I)=ISS1(JJ)
150       IXT2(I)=ISS2(JJ)
C         SELECT THE BEST VECTOR (WITH THE SMALLEST COST)
          CC=500000.0
          DO 160 I=1,K,1
          IF(C2(I).LT.CC) THEN
          CC=C2(I)
          II=I
          ENDIF
160       CONTINUE
C         DETECTION AND ERRORS CALCULATION
          IEVK=IEVK+1
          IF(IS1(1)-IX1(II,1)) 180,170,180
170       IF(IS2(1)-IX2(II,1)) 180,190,180
180       ISER=ISER+1

          IF(IEVK-32) 185,185,186
185       IEVA=IEVA+1
186       IEVK=0
190       CONTINUE

C         DIFFERANTIAL DECODING. and BER calculation.
          DO 220 I=1,LEV,1
          IF(ISS1(I)-IX1(II,1)) 220,200,220
200       IF(ISS2(I)-IX2(II,1)) 220,210,220
210       IND=I
          GO TO 230
220       CONTINUE
230       CONTINUE

          IBB=IND-1
          DO 240 I=IBIT,1,-1
          J=2**(I-1)
          IF(IBB.LT.J) THEN
          IBD(I)=0
          ELSE
          IBD(I)=1
          IBB=IBB-J
          ENDIF
240       CONTINUE
```

```
      IDPREV=IDPRES
      IDPRES=IBD(1)+2*IBD(2)
      DO 270 I=1,K,1
      IF(IDPREV.EQ.IPREV(I)) THEN
      IF(IDPRES.EQ.IPRES(I)) THEN
      III=I
      GO TO 280
      ENDIF
      ENDIF
270   CONTINUE
280   IAA=INA(III)+4*IBD(3)+8*IBD(4)
      DO 290 I=IBIT,1,-1
      J=2**(I-1)
      IF(IAA.LT.J) THEN
      IAD(I)=0
      ELSE
      IAD(I)=1
      IAA=IAA-J
      ENDIF
290   CONTINUE
      DO 310 I=1,IBIT,1
      IF(IAD(I)-IA(1,I)) 300,310,300
300   IBER=IBER+1
310   CONTINUE
C     DISCARD ALL VECTORS WHICH DISAGREE IN THE FIRST COMP.
      CC=500000.0
      DO 360 I=1,K,1
      IF(IX1(I,1)-IX1(II,1)) 350,340,350
340   IF(IX2(I,1)-IX2(II,1)) 350,360,350
350   C1(I)=CC
      C2(I)=CC
360   CONTINUE
C     SELECT THE BEST 8 VECTORS,ADD 3,3,1 AND 1 TO THE 1ST,2ND
C     ,3RD AND 4TH VECTORS AND CALCULATE THEIR COSTS.
      I1=0
      DO 410 I=1,8,1
      CC=500000.0
      DO 370 J=1,K,1
      IF(C2(J).LT.CC) THEN
      CC=C2(J)
      JJJ=J
      ENDIF
370   CONTINUE
      C11(I)=C1(JJJ)
      E11(I)=E1(JJJ)
      E22(I)=E2(JJJ)
      C2(JJJ)=600000.0
      DO 380 J1=1,N1,1
      IXX1(I,J1)=IX1(JJJ,J1)
380   IXX2(I,J1)=IX2(JJJ,J1)
      DO 390 J=1,LEV,1
390   C3(I,J)=C11(I)+(E11(I)-REAL(ISS1(J)))**2
     C                +(E22(I)-REAL(ISS2(J)))**2
      DO 410 J1=1,KK(I),1
      I1=I1+1
```

```
          CC=500000.0
          DO 400 J=1,LEV,1
          IF(C3(I,J).LT.CC) THEN
          CC=C3(I,J)
          JJ=J
          ENDIF
400       CONTINUE
          C22(I1)=C3(I,JJ)
          C3(I,JJ)=700000.0
          IXXX1(I1,N)=ISS1(JJ)
          IXXX2(I1,N)=ISS2(JJ)
          DO 410 I2=1,N1,1
          IXXX1(I1,I2)=IXX1(I,I2)
          IXXX2(I1,I2)=IXX2(I,I2)
410       CONTINUE
          DO 415 I=1,K,1
          DO 415 J=1,N,1
          IX1(I,J)=IXXX1(I,J)
415       IX2(I,J)=IXXX2(I,J)
C         SUBTRACT THE SMALLEST COST FROM ALL COST.
          CC=500000.0
          DO 420 I=1,K,1
          IF(C22(I).LT.CC) THEN
          II=I
          CC=C22(I)
          ENDIF
420       CONTINUE
          CCC=CC
          DO 430 I=1,K,1
430       C1(I)=C22(I)-CCC

C         THE PROCESS CONTINUE WITH THE NEXT SYMBOL.
440       CONTINUE
C         CALCULATE ERROR RATES AND PRINTOUT ALL RESULTS
          WRITE(1,*)
          WRITE(1,500) SNR,P**2
          SER=REAL(ISER)/REAL(L)
          EVE=REAL(IEVA)/REAL(L)
          BER=REAL(IBER)/REAL((L)*IBIT)
          WRITE(1,600) SER,EVE,BER
500       FORMAT(1X,4HSNR=,F12.6,2X,10HNOISE VAR=,F12.6)
600       FORMAT(1X,4HSER=,F9.7,1X,4HEVE=,F9.7,1X,4HBER=,F9.7)
1000      CONTINUE
          END
####S
```

APPENDIX G2
THE FOLLOWING PROGRAM SIMULATES THE OPERATION OF THE MODIFIED NONLINEAR EQUALIZER WITH VITERBI ALGORITHM DETECTOR(SYSTEM MNC) THE TRANSMITTED SIGNAL IS A CONVOLUTIONALLY ENCODED 32-LEVEL QAM SIGNAL

```
/*JOB SMNC,EUELAKK,ST=MFX,C=S,TI=1280,
/* PW=K
FTN5,DB=0/PMD,L=0.
LIBRARY,PROCLIB.
NAG(FTN5)
LGO.
####S
       PROGRAM SMNC
       INTEGER IS1(33),IS2(33),ISS(34),IA(33,4),IAA(4)
     C,IXXX1(128),IXXX2(128),IXXX3(128),IXXX4(128),IXXX5(128)
     C,IADD(8,33),IAD(8,33),ISTATP(8),IEVA,IEVK
     C,ISER,IBER,IX1(8,33),IX2(8,33),IXX1(8,33),IXX2(8,33)
       REAL  Y1(20),Y2(20),E1,E2,C1(8),C2(8,128),CC1(8)
       REAL P,SNR,SNRD,G05DDF,G05DAF,B1(20),B2(20)
C      LOAD THE SAMPLED IMPULSE RESPONSE OF THE CHANNEL
       DATA   Y1 / 1.0000, 0.4861,-0.5980, 0.1702,-0.0245, 0.0100
     C           ,-0.0134, 0.0056, 0.0003,-0.0008, 0.0000, 0.0007
     C           , 0.0037,-0.0019, 0.0020, 0.0005,-0.0022, 0.0007
     C           ,-0.0008, 0.0005/
       DATA   Y2 / 0.0000, 1.0988, 0.0703,-0.1938, 0.1000,-0.0258
     C           , 0.0110,-0.0042, 0.0003, 0.0041,-0.0061,-0.0007
     C           , 0.0002,-0.0025, 0.0008,-0.0002, 0.0002,-0.0005
     C           , 0.0002, 0.0005/
C      LOAD TABLE FOR ENCODER/DECODER
       DATA IXXX1/0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1
     C           ,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
     C           ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4
     C           ,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5
     C           ,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7
     C           ,7,7,7,7,7,7,7,7/
       DATA IXXX2/0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2
     C           ,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6
     C           ,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,1,1,1,1,3,3,3,3
     C           ,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7
     C           ,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3
     C           ,5,5,5,5,7,7,7,7/
       DATA IXXX3/-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
     C           , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3
     C           , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
     C           ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
     C           , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C           ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
     C           , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
     C           ,-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5/
       DATA IXXX4/-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5
     C           , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C           ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
     C           , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
```

```
C           ,-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
C           ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
C           , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
C           , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3/
      DATA IXXX5/ 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12
C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
C           , 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12/
      OPEN(1,FILE='OUTPUT')
C     SET ALL PARAMETERS .
      MM=3
      N =33
      N1=N-1
      N2=N+1
      L=100000
      LN=L+N1
      MK=8
      IQ=10
      SNR=18.5
      SNRD=0.25
      CALL G05CBF(IQ)
C     SCALING THE SAMPLED IMPULSE RESPONSE
      YYY=0.0
      DO 10 I=1,20,1
10    YYY=YYY+Y1(I)*Y1(I)+Y2(I)*Y2(I)
      YY=SQRT(YYY)
      DO 20 I=1,20,1
      Y1(I)=Y1(I)/YY
20    Y2(I)=Y2(I)/YY
C     MAIN DO LOOP (SELECT SNR)
      DO 1000 MMM=1,MM,1
      DO 1 I=1,MK,1
      DO 1 J=1,N,1
      IXX1(I,J)=0
1     IXX2(I,J)=0
      SNR=SNR+SNRD
      P=SQRT(5.0/(10.0**(SNR/10.0)))
C     INITIALIZATION OF ALL ARRAY AND VARIABLES.
      ISER=0
      IBER=0
      IEVA=0
      IEVK=0
      DO 40 I=1,N,1
      IS1(I)=-1
      IS2(I)=-1
      ISS(I)=0
      DO 30 J=1,4,1
30    IA(I,J)=0
      DO 40 J1=1,MK,1
      IX1(J1,I)=-1
      IX2(J1,I)=-1
```

```
40      IAD(J1,I)=0
        DO 50 I=2,MK,1
        DO 45 I=1,20,1
        B1(I)=-1.0
45      B2(I)=-1.0
50      C1(I)=100000.0
C       ASSIGN DIFFERENT STATES TO THE VECTORS WITH ZERO COST
C       TO THE VECTOR WITH THE CORRECT STATE .
        DO 60 I=0,7,1
        J=I+1
60      ISTATP(J)=I
        C1(1)=0.0
        ISS(N2)=0
C       SECOND DO LOOP ( TRANSMISSION )
        DO 900 LLL=1,LN,1
C       SHIFT ALL ARRAYS .
        DO 100 J=1,N1,1
        JA=J+1
        IS1(J)=IS1(JA)
        IS2(J)=IS2(JA)
        ISS(J)=ISS(JA)
        DO 90 I=1,4,1
90      IA(J,I)=IA(JA,I)
        DO 100 K=1,MK,1
        IX1(K,J)=IX1(K,JA)
        IX2(K,J)=IX2(K,JA)
100     IAD(K,J)=IAD(K,JA)
        DO 105 J=20,2,-1
        B1(J)=B1(J-1)
105     B2(J)=B2(J-1)
        ISS(N)=ISS(N2)
C       GENERATE INFORMATION DIGITS.
        DO 110 I=1,4,1
110     IA(N,I)=NINT(G05DAF(0.0,1.0))
C       CONVOLUTIONAL ENCODING AND SIGNAL MAPPING
        INPUT=IA(N,1)+2*IA(N,2)+4*IA(N,3)+8*IA(N,4)
        DO 140 I=1,128,1
        IF(ISS(N).EQ.IXXX2(I).AND.INPUT.EQ.IXXX5(I)) THEN
        II=I
        ENDIF
140     CONTINUE
        ISS(N2)=IXXX1(II)
        IS1(N)=IXXX3(II)
        IS2(N)=IXXX4(II)
C       CALCULATION OF THE RECEIVED SAMPLE
        RR1=0.0
        RR2=0.0
        DO 150 I=1,20,1
        J=N+1-I
        RR1=RR1+REAL(IS1(J))*Y1(I)-REAL(IS2(J))*Y2(I)
150     RR2=RR2+REAL(IS1(J))*Y2(I)+REAL(IS2(J))*Y1(I)
C       ADD THE NOISE.
        W1=G05DDF(0.0,P)
        W2=G05DDF(0.0,P)
        RR1=RR1+W1
```

```
      RR2=RR2+W2
C     CALCULATE THE EQUALIZED SAMPLE.
      ZZ1=0.0
      ZZ2=0.0
      DO 160 J=2,20,1
      ZZ1=ZZ1+B1(J)*Y1(J)-B2(J)*Y2(J)
160   ZZ2=ZZ2+B1(J)*Y2(J)+B2(J)*Y1(J)
      E1=(RR1-ZZ1)*YY
      E2=(RR2-ZZ2)*YY
C     SEARCH FOR ALL VALID TRANSITION AND COST CALCULATIONS FOR
C     ALL VECTORS.
      DO 180 I=1,128,1
      DO 180 J=1,MK,1
180   C2(J,I)=1000000.0
      DO 210 I=1,MK,1
      DO 210 J=1,128,1
      IF(ISTATP(I).EQ.IXXX2(J)) THEN
      D1=E1-REAL(IXXX3(J))
      D2=E2-REAL(IXXX4(J))
      C2(I,J)=C1(I)+D1*D1+D2*D2
      ENDIF
210   CONTINUE
C     FIND THE VECTOR WITH THE SMALLEST COST .
      CC=500000.0
      DO 220 I=1,MK,1
      DO 220 J=1,128,1
      IF(C2(I,J).LT.CC) THEN
      CC=C2(I,J)
      II=I
      JJ=J
      ENDIF
220   CONTINUE
      IX1(II,N)=IXXX3(JJ)
      IX2(II,N)=IXXX4(JJ)
      IAD(II,N)=IXXX5(JJ)
C     ERROR CALCULATION
      IEVK=IEVK+1
      IF(IS1(1)-IX1(II,1)) 250,240,250
240   IF(IS2(1)-IX2(II,1)) 250,290,250
250   ISER=ISER+1
      IF(IEVK-32) 260,260,270
260   IEVA=IEVA+1
270   IEVK=0
290   CONTINUE
      IAAD=IAD(II,1)
      DO 300 I=4,1,-1
      JJ=2**(I-1)
      IF(IAAD.LT.JJ) THEN
      IAA(I)=0
      ELSE
      IAA(I)=1
      IAAD=IAAD-JJ
      ENDIF
300   CONTINUE
      DO 320 I=1,4,1
```

```
      IF(IAA(I)-IA(1,I)) 310,320,310
310   IBER=IBER+1
320   CONTINUE
C     PASS THE DETECTED VALUE OF THE CODED SYMBOL TO
C     FEEDBACK LOOP OF THE NONLINEAR EQUALIZER .
      B1(1)=REAL(IX1(II,N))
      B2(1)=REAL(IX2(II,N))
C     DISCARDING OF ALL VECTOR WHICH DISAGREE IN THE
C     FIRST COMPONENT AND SELECTING OF MK VECTOR FOR
C     THE NEXT SAMPLE
      CC=1000000.0
      DO 360 I=1,MK,1
      IF(IX1(II,1)-IX1(I,1)) 340,330,340
330   IF(IX2(II,1)-IX2(I,1)) 340,360,340
340   CONTINUE
      DO 350 J=1,128,1
350   C2(I,J)=CC
360   CONTINUE
      J1=0
      QQ1=0
      DO 430 I1=0,7,1
      CC=1000000.0
      J1=J1+1
      DO 380 I=1,128,1
      DO 380 J=1,MK,1
      IF(I1.EQ.IXXX1(I)) THEN
      IF(C2(J,I).LT.CC) THEN
      CC=C2(J,I)
      JJ=J
      II=I
      ENDIF
      ENDIF
380   CONTINUE
      QQ1=QQ1+1
      IXX1(J1,N)=IXXX3(II)
      IXX2(J1,N)=IXXX4(II)
      IADD(J1,N)=IXXX5(II)
      DO 420 K=1,N1,1
      IXX1(J1,K)=IX1(JJ,K)
      IXX2(J1,K)=IX2(JJ,K)
420   IADD(J1,K)=IAD(JJ,K)
      CC1(J1)=C2(JJ,II)
      C2(JJ,II)=1000000.0
      ISTATP(J1)=IXXX1(II)
430   CONTINUE
C     TRANSFER ALL ARRAYS
      DO 440 I=1,MK,1
      DO 440 J=1,N,1
      IX1(I,J)=IXX1(I,J)
      IX2(I,J)=IXX2(I,J)
440   IAD(I,J)=IADD(I,J)
      CC=2000000.0
      DO 450 I=1,MK,1
      IF(CC1(I).LT.CC) THEN
      CC=CC1(I)
```

```
      II=I
      ENDIF
450   CONTINUE
      DO 460 I=1,MK,1
460   C1(I)=CC1(I)-CC1(II)
900   CONTINUE
C     CALCULATE ERROR RATES AND PRINTOUT ALL RESULTS
      WRITE(1,*)
      WRITE(1,500) SNR,P**2
      SER=REAL(ISER)/REAL(L)
      EVE=REAL(IEVA)/REAL(L)
      BER=REAL(IBER)/REAL((L)*IBIT)
      WRITE(1,600) SER,EVE,BER
500   FORMAT(1X,4HSNR=,F12.6,2X,10HNOISE VAR=,F12.6)
600   FORMAT(1X,4HSER=,F9.7,1X,4HEVE=,F9.7,1X,4HBER=,F9.7)
1000  CONTINUE
      END
####S
```

APPENDIX G3
THE FOLLOWING PROGRAM SIMULATES THE OPERATION OF SYSTEM 1C16.
THE DATA IS CONVOLUTIONALLY ENCODED (32-LEVEL QAM SIGNAL).

```
/*JOB S1C16,EUELAKK,ST=MFX,C=S,TI=1280,
/* PW=K
FTN5,DB=0/PMD,L=0.
LIBRARY,PROCLIB.
NAG(FTN5)
LGO.
####S
      PROGRAM S1C16
      INTEGER IS1(33),IS2(33),ISS(34),IA(33,4),IAA(4)
     C,IXXX1(128),IXXX2(128),IXXX3(128),IXXX4(128),IXXX5(128)
     C,IADD(64,33),IAD(16,33),ISTATP(16),IEVA,IEVK,ISTATPP(64)
     C,ISER,IBER,IX1(16,33),IX2(16,33),IXX1(64,33),IXX2(64,33)
      REAL  Y1(20),Y2(20),Z1(16),Z2(16),C1(16),C2(64),CC1(16)
      REAL P,SNR,SNRD,G05DDF,G05DAF,C3(16,128)

C     LOAD THE SAMPLED IMPULSE RESPONSE OF THE CHANNEL
      DATA   Y1 / 1.0000, 0.4861,-0.5980, 0.1702,-0.0245, 0.0100
     C          ,-0.0134, 0.0056, 0.0003,-0.0008, 0.0000, 0.0007
     C          , 0.0037,-0.0019, 0.0020, 0.0005,-0.0022, 0.0007
     C          ,-0.0008, 0.0005/
      DATA   Y2 / 0.0000, 1.0988, 0.0703,-0.1938, 0.1000,-0.0258
     C          , 0.0110,-0.0042, 0.0003, 0.0041,-0.0061,-0.0007
     C          , 0.0002,-0.0025, 0.0008,-0.0002, 0.0002,-0.0005
     C          , 0.0002, 0.0005/
C     LOAD TABLE FOR ENCODER/DECODER AND SIGNAL MAPPING
      DATA IXXX1/0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1
     C          ,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
     C          ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4
     C          ,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5
     C          ,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7
     C          ,7,7,7,7,7,7,7,7/
      DATA IXXX2/0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2
     C          ,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6
     C          ,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,1,1,1,1,3,3,3,3
     C          ,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7
     C          ,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3
     C          ,5,5,5,5,7,7,7,7/
      DATA IXXX3/-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
     C          , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3
     C          , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
     C          ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
     C          , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C          ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
     C          , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
     C          ,-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5/
      DATA IXXX4/-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5
     C          , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C          ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
     C          , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
     C          ,-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
```

```
C           ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
C           , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
C           , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3/
      DATA IXXX5/ 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12
C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
C           , 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12/
      OPEN(1,FILE='OUTPUT')

C     SET ALL PARAMETERS.
      MM=4
      N =33
      N1=N-1
      N2=N+1
      L=100000
      LN=L+N1
      MK=16
      MKK=2
      IQ=22
      SNR=12.5
      SNRD=1.0
      CALL G05CBF(IQ)
C     SCALING THE SAMPLED IMPULSE RESPONSE
      YYY=0.0
      DO 10 I=1,20,1
10    YYY=YYY+Y1(I)*Y1(I)+Y2(I)*Y2(I)
      YY=SQRT(YYY)
      DO 20 I=1,20,1
      Y1(I)=Y1(I)/YY
20    Y2(I)=Y2(I)/YY
C     MAIN DO LOOP (SELECTION OF SNR)
      DO 1000 MMM=1,MM,1
      DO 1 I=1,MK,1
      DO 1 J=1,N,1
      IXX1(I,J)=0
1     IXX2(I,J)=0
      SNR=SNR+SNRD
      P=SQRT(5.0/(10.0**(SNR/10.0)))
C     INITIALIZATION OF ALL ARRAY AND VARIABLES.
      ISER=0
      IBER=0
      IEVA=0
      IEVK=0
      DO 40 I=1,N,1
      IS1(I)=-1
      IS2(I)=-1
      ISS(I)=0
      DO 30 J=1,4,1
30    IA(I,J)=0
      DO 40 J1=1,MK,1
      IX1(J1,I)=-1
```

```
      IX2(J1,I)=-1
40    IAD(J1,I)=0
      DO 50 I=2,MK,1
50    C1(I)=100000.0
C     ASSIGN DIFFERENT STATES FOR THE VECTORS WITH ZERO COST
C     TO THE VECTOR WITH THE CORRECT STATE.
      DO 60 I=0,7,1
      DO 60 J=2*I+1,2*(I+1),1
60    ISTATP(J)=I
      C1(1)=0.0
      ISS(N2)=0
C     SECOND DO LOOP ( TRANSMISSION )
      DO 900 LLL=1,LN,1
C     SHIFT ALL ARRAYS .
      DO 100 J=1,N1,1
      JA=J+1
      IS1(J)=IS1(JA)
      IS2(J)=IS2(JA)
      ISS(J)=ISS(JA)
      DO 90 I=1,4,1
90    IA(J,I)=IA(JA,I)
      DO 100 K=1,MK,1
      IX1(K,J)=IX1(K,JA)
      IX2(K,J)=IX2(K,JA)
100   IAD(K,J)=IAD(K,JA)
      ISS(N)=ISS(N2)
C     GENERATE INFORMATION DIGITS.
      DO 110 I=1,4,1
110   IA(N,I)=NINT(G05DAF(0.0,1.0))
C     CONVOLUTIONAL ENCODING AND SIGNAL MAPPING .
      INPUT=IA(N,1)+2*IA(N,2)+4*IA(N,3)+8*IA(N,4)
      DO 140 I=1,128,1
      IF(ISS(N).EQ.IXXX2(I).AND.INPUT.EQ.IXXX5(I)) THEN
      II=I
      ENDIF
140   CONTINUE
      ISS(N2)=IXXX1(II)
      IS1(N)=IXXX3(II)
      IS2(N)=IXXX4(II)
C     CALCULATION OF THE RECEIVED SAMPLE
      RR1=0.0
      RR2=0.0
      DO 150 I=1,20,1
      J=N+1-I
      RR1=RR1+REAL(IS1(J))*Y1(I)-REAL(IS2(J))*Y2(I)
150   RR2=RR2+REAL(IS1(J))*Y2(I)+REAL(IS2(J))*Y1(I)
C     GENERATE AND ADD THE NOISE SAMPLES.
      W1=G05DDF(0.0,P)
      W2=G05DDF(0.0,P)
      RR1=RR1+W1
      RR2=RR2+W2
C     CALCULATE ISI FOR EACH STORED VECTOR
      DO 170 I=1,MK,1
      ZZ1=0.0
      ZZ2=0.0
```

```
      DO 160 J=2,20,1
      J1=N+1-J
      ZZ1=ZZ1+REAL(IX1(I,J1))*Y1(J)-REAL(IX2(I,J1))*Y2(J)
160   ZZ2=ZZ2+REAL(IX1(I,J1))*Y2(J)+REAL(IX2(I,J1))*Y1(J)
      Z1(I)=(RR1-ZZ1)*YY
170   Z2(I)=(RR2-ZZ2)*YY
C     SEARCH FOR ALL VALID TRANSITION AND COST CALCULATIONS FOR
C     ALL VECTORS.
      DO 180 I=1,128,1
      DO 180 J=1,MK,1
180   C3(J,I)=1000000.0
      J2=0
      DO 210 I=1,MK,1
      DO 215 J=1,128,1
      IF(ISTATP(I).EQ.IXXX2(J)) THEN
      D1=Z1(I)-REAL(IXXX3(J))
      D2=Z2(I)-REAL(IXXX4(J))
      C3(I,J)=C1(I)+D1*D1+D2*D2
      ENDIF
215   CONTINUE
      DO 216 J1=1,4,1
      J2=J2+1
      CC=1000001.0
      DO 217 J=1,128,1
      IF(C3(I,J).LT.CC) THEN
      CC=C3(I,J)
      JJ=J
      ENDIF
217   CONTINUE
      ISTATPP(J2)=IXXX1(JJ)
      IADD(J2,N)=IXXX5(JJ)
      IXX1(J2,N)=IXXX3(JJ)
      IXX2(J2,N)=IXXX4(JJ)
      DO 225 K1=1,N1,1
      IXX1(J2,K1)=IX1(I,K1)
      IXX2(J2,K1)=IX2(I,K1)
225   IADD(J2,K1)=IAD(I,K1)
      C2(J2)=CC
      C3(I,JJ)=1000001.0
216   CONTINUE
210   CONTINUE
C     FIND THE VECTOR WITH THE SMALLEST COST .
      CC=500000.0
      DO 220 I=1,MK*4,1
      IF(C2(I).LT.CC) THEN
      CC=C2(I)
      II=I
      ENDIF
220   CONTINUE
C     ERROR CALCULATION
      IEVK=IEVK+1
      IF(IS1(1)-IXX1(II,1)) 250,240,250
240   IF(IS2(1)-IXX2(II,1)) 250,290,250
250   ISER=ISER+1
      IF(IEVK-32) 260,260,270
```

```
260     IEVA=IEVA+1
270     IEVK=0
290     CONTINUE
        IAAD=IAD(II,1)
        DO 300 I=4,1,-1
        JJ=2**(I-1)
        IF(IAAD.LT.JJ) THEN
        IAA(I)=0
        ELSE
        IAA(I)=1
        IAAD=IAAD-JJ
        ENDIF
300     CONTINUE
        DO 320 I=1,4,1
        IF(IAA(I)-IA(1,I)) 310,320,310
310     IBER=IBER+1
320     CONTINUE
C       DISCARDING OF ALL VECTOR WHICH DISAGREE IN THE
C       FIRST COMPONENT AND SELECTING OF MK VECTOR FOR
C       THE NEXT SAMPLE
        CC=1000000.0
        DO 360 I=1,MK*4,1
        IF(IXX1(II,1)-IXX1(I,1)) 340,330,340
330     IF(IXX2(II,1)-IXX2(I,1)) 340,360,340
340     C2(I)=CC
360     CONTINUE
        DO 430 J1=1,MK,1
        CC=1000000.0
        DO 380 I=1,MK*4,1
        IF(C2(I).LT.CC) THEN
        CC=C2(I)
        II=I
        ENDIF
380     CONTINUE
        DO 420 K=1,N,1
        IX1(J1,K)=IXX1(II,K)
        IX2(J1,K)=IXX2(II,K)
420     IAD(J1,K)=IADD(II,K)
        CC1(J1)=C2(II)
        C2(II)=1000000.0
        ISTATP(J1)=ISTATPP(II)
430     CONTINUE
        CC=2000000.0
        DO 450 I=1,MK,1
        IF(CC1(I).LT.CC) THEN
        CC=CC1(I)
        II=I
        ENDIF
450     CONTINUE
C       SUBTRACT THE SMALLEST COST.
        DO 460 I=1,MK,1
460     C1(I)=CC1(I)-CC1(II)
C       THE PROCESS CONTINUE WITH THE NEXT SYMBOL.
900     CONTINUE
C       CALCULATE ERROR RATES AND PRINTOUT ALL RESULTS
```

```
      WRITE(1,*)
      WRITE(1,500) SNR,P**2
      SER=REAL(ISER)/REAL(L)
      EVE=REAL(IEVA)/REAL(L)
      BER=REAL(IBER)/REAL((L)*4)
      WRITE(1,600) SER,EVE,BER
500   FORMAT(1X,4HSNR=,F12.6,2X,10HNOISE VAR=,F12.6)
600   FORMAT(1X,4HSER=,F9.7,1X,4HEVE=,F9.7,1X,4HBER=,F9.7)
1000  CONTINUE
      END
####S
```

APPENDIX G4
THE FOLLOWING PROGRAM SIMULATES THE OPERATION OF SYSTEM 2C16
THE TRANSMITTED DATA IS CONVOLUTIONALLY ENCODED 32-LEVEL QAM
SIGNAL. THE DETECTOR OPERATES WITH 16 VECTORS.

```
/*JOB S2C16,EUELAKK,ST=MFX,C=S,TI=1280,
/* PW=K
FTN5,DB=0/PMD,L=0.
LIBRARY,PROCLIB.
NAG(FTN5)
LGO.
####S
       PROGRAM S2C16
       INTEGER IS1(33),IS2(33),ISS(34),IA(33,4),IAA(4)
     C,IXXX1(128),IXXX2(128),IXXX3(128),IXXX4(128),IXXX5(128)
     C,IADD(64,33),IAD(16,33),ISTATP(16),IEVA,IEVK,ISTATPP(64)
     C,ISER,IBER,IX1(16,33),IX2(16,33),IXX1(64,33),IXX2(64,33)
     C,KK1(16),KK2(16)
       REAL  Y1(20),Y2(20),Z1(16),Z2(16),C1(16),C2(64),CC1(16)
       REAL P,SNR,SNRD,G05DDF,G05DAF,C3(16,128)

       DATA KK1/1,5,9,13,17,20,23,26,29,31,33,35,37,38,39,40/
       DATA KK2/4,8,12,16,19,22,25,28,30,32,34,36,37,38,39,40/
C      LOAD THE SAMPLED IMPULSE RESPONSE OF THE CHANNEL
       DATA   Y1 / 1.0000, 0.4861,-0.5980, 0.1702,-0.0245, 0.0100
     C           ,-0.0134, 0.0056, 0.0003,-0.0008, 0.0000, 0.0007
     C           , 0.0037,-0.0019, 0.0020, 0.0005,-0.0022, 0.0007
     C           ,-0.0008, 0.0005/
       DATA   Y2 / 0.0000, 1.0988, 0.0703,-0.1938, 0.1000,-0.0258
     C           , 0.0110,-0.0042, 0.0003, 0.0041,-0.0061,-0.0007
     C           , 0.0002,-0.0025, 0.0008,-0.0002, 0.0002,-0.0005
     C           , 0.0002, 0.0005/
C      LOAD TABLE FOR ENCODER/DECODER AND SIGNAL MAPPING
       DATA IXXX1/0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1
     C           ,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
     C           ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4
     C           ,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5
     C           ,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7
     C           ,7,7,7,7,7,7,7,7/
       DATA IXXX2/0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2
     C           ,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6
     C           ,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,1,1,1,1,3,3,3,3
     C           ,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7
     C           ,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3
     C           ,5,5,5,5,7,7,7,7/
       DATA IXXX3/-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
     C           , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3
     C           , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
     C           ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
     C           , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C           ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
     C           , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
     C           ,-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5/
```

```
      DATA IXXX4/-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5
     C           , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C           ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
     C           , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
     C           ,-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
     C           ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
     C           , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
     C           , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3/
      DATA IXXX5/ 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12
     C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
     C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
     C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
     C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
     C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
     C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
     C           , 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12/
      OPEN(1,FILE='OUTPUT')
C     SET ALL PARAMETERS.
      MM=3
      N =33
      N1=N-1
      N2=N+1
      L =100000
      LN=L+N1
      MK=16
      MKK=2
      IQ=1111
      SNR=14.0
      SNRD=0.5
      CALL G05CBF(IQ)
C     SCALING OF THE SAMPLED IMPULSE RESPONSE
      YYY=0.0
      DO 10 I=1,20,1
10    YYY=YYY+Y1(I)*Y1(I)+Y2(I)*Y2(I)
      YY=SQRT(YYY)
      DO 20 I=1,20,1
      Y1(I)=Y1(I)/YY
20    Y2(I)=Y2(I)/YY
C     MAIN DO LOOP (SELECTION OF SNR)
      DO 1000 MMM=1,MM,1
      DO 1 I=1,MK,1
      DO 1 J=1,N,1
      IXX1(I,J)=0
1     IXX2(I,J)=0
      SNR=SNR+SNRD
      P=SQRT(5.0/(10.0**(SNR/10.0)))
C     INITIALIZATION OF ALL ARRAY AND VARIABLES.
      ISER=0
      IBER=0
      IEVA=0
      IEVK=0
      DO 40 I=1,N,1
      IS1(I)=-1
      IS2(I)=-1
      ISS(I)=0
```

```
      DO 30 J=1,4,1
30    IA(I,J)=0
      DO 40 J1=1,MK,1
      IX1(J1,I)=-1
      IX2(J1,I)=-1
40    IAD(J1,I)=0
      DO 50 I=2,MK,1
50    C1(I)=100000.0
C     ASSIGN DIFFERENT STATES FOR THE VECTORS WITH ZERO COST
C     TO THE VECTOR WITH CORRECT STATE.
      DO 60 I=0,7,1
      DO 60 J=2*I+1,2*(I+1),1
60    ISTATP(J)=I
      C1(1)=0.0
      ISS(N2)=0
C     SECOND DO LOOP (TRANSMISSION )
      DO 900 LLL=1,LN,1
C     SHIFT ALL ARRAYS .
      DO 100 J=1,N1,1
      JA=J+1
      IS1(J)=IS1(JA)
      IS2(J)=IS2(JA)
      ISS(J)=ISS(JA)
      DO 90 I=1,4,1
90    IA(J,I)=IA(JA,I)
      DO 100 K=1,MK,1
      IX1(K,J)=IX1(K,JA)
      IX2(K,J)=IX2(K,JA)
100   IAD(K,J)=IAD(K,JA)
      ISS(N)=ISS(N2)
C     GENERATE INFORMATION DIGITS.
      DO 110 I=1,4,1
110   IA(N,I)=NINT(G05DAF(0.0,1.0))
C     CONVOLUTIONAL ENCODING.
      INPUT=IA(N,1)+2*IA(N,2)+4*IA(N,3)+8*IA(N,4)
      DO 140 I=1,128,1
      IF(ISS(N).EQ.IXXX2(I).AND.INPUT.EQ.IXXX5(I)) THEN
      II=I
      ENDIF
140   CONTINUE
      ISS(N2)=IXXX1(II)
      IS1(N)=IXXX3(II)
      IS2(N)=IXXX4(II)
C     CALCULATION OF THE RECEIVED SAMPLE
      RR1=0.0
      RR2=0.0
      DO 150 I=1,20,1
      J=N+1-I
      RR1=RR1+REAL(IS1(J))*Y1(I)-REAL(IS2(J))*Y2(I)
150   RR2=RR2+REAL(IS1(J))*Y2(I)+REAL(IS2(J))*Y1(I)

C     GENERATE AND ADD TH NOISE.
      W1=G05DDF(0.0,P)
      W2=G05DDF(0.0,P)
      RR1=RR1+W1
```

```
      RR2=RR2+W2
C     CALCULATE ISI FOR EACH STORED VECTOR
      DO 170 I=1,MK,1
      ZZ1=0.0
      ZZ2=0.0
      DO 160 J=2,20,1
      J1=N+1-J
      ZZ1=ZZ1+REAL(IX1(I,J1))*Y1(J)-REAL(IX2(I,J1))*Y2(J)
160   ZZ2=ZZ2+REAL(IX1(I,J1))*Y2(J)+REAL(IX2(I,J1))*Y1(J)
      Z1(I)=(RR1-ZZ1)*YY
170   Z2(I)=(RR2-ZZ2)*YY
C     SEARCH FOR ALL VALID TRANSITION AND COST CALCULATIONS FOR
C     ALL VECTORS.
      DO 180 I=1,128,1
      DO 180 J=1,MK,1
180   C3(J,I)=1000000.0
      J2=0
      DO 210 I=1,MK,1
      DO 215 J=1,128,1
      IF(ISTATP(I).EQ.IXXX2(J)) THEN
      D1=Z1(I)-REAL(IXXX3(J))
      D2=Z2(I)-REAL(IXXX4(J))
      C3(I,J)=C1(I)+D1*D1+D2*D2
      ENDIF
215   CONTINUE
      DO 216 J1=KK1(I),KK2(I),1
      J2=J2+1
      CC=1000001.0
      DO 217 J=1,128,1
      IF(C3(I,J).LT.CC) THEN
      CC=C3(I,J)
      JJ=J
      ENDIF
217   CONTINUE
      ISTATPP(J2)=IXXX1(JJ)
      IADD(J2,N)=IXXX5(JJ)
      IXX1(J2,N)=IXXX3(JJ)
      IXX2(J2,N)=IXXX4(JJ)
      DO 225 K1=1,N1,1
      IXX1(J2,K1)=IX1(I,K1)
      IXX2(J2,K1)=IX2(I,K1)
225   IADD(J2,K1)=IAD(I,K1)
      C2(J2)=CC
      C3(I,JJ)=1000001.0
216   CONTINUE
210   CONTINUE
C     FIND THE VECTOR WITH THE SMALLEST COST .
      CC=500000.0
      DO 220 I=1,KK2(16),1
      IF(C2(I).LT.CC) THEN
      CC=C2(I)
      II=I
      ENDIF
220   CONTINUE
C     ERROR CALCULATION
```

```
      IEVK=IEVK+1
      IF(IS1(1)-IXX1(II,1)) 250,240,250
240   IF(IS2(1)-IXX2(II,1)) 250,290,250
250   ISER=ISER+1
      IF(IEVK-32) 260,260,270
260   IEVA=IEVA+1
270   IEVK=0
290   CONTINUE
      IAAD=IAD(II,1)
      DO 300 I=4,1,-1
      JJ=2**(I-1)
      IF(IAAD.LT.JJ) THEN
      IAA(I)=0
      ELSE
      IAA(I)=1
      IAAD=IAAD-JJ
      ENDIF
300   CONTINUE
      DO 320 I=1,4,1
      IF(IAA(I)-IA(1,I)) 310,320,310
310   IBER=IBER+1
320   CONTINUE
C     DISCARDING OF ALL VECTOR WHICH DISAGREE IN THE
C     FIRST COMPONENT AND SELECT MK VECTORS FOR
C     THE NEXT SAMPLE
      CC=1000000.0
      DO 360 I=1,KK2(16),1
      IF(IXX1(II,1)-IXX1(I,1)) 340,330,340
330   IF(IXX2(II,1)-IXX2(I,1)) 340,360,340
340   C2(I)=CC
360   CONTINUE
      DO 430 J1=1,MK,1
      CC=1000000.0
      DO 380 I=1,KK2(16),1
      IF(C2(I).LT.CC) THEN
      CC=C2(I)
      II=I
      ENDIF
380   CONTINUE
      DO 420 K=1,N,1
      IX1(J1,K)=IXX1(II,K)
      IX2(J1,K)=IXX2(II,K)
420   IAD(J1,K)=IADD(II,K)
      CC1(J1)=C2(II)
      C2(II)=1000000.0
      ISTATP(J1)=ISTATPP(II)
430   CONTINUE
      CC=2000000.0
      DO 450 I=1,MK,1
      IF(CC1(I).LT.CC) THEN
      CC=CC1(I)
      II=I
      ENDIF
450   CONTINUE
C     SUBTRACT THE SMALLEST COST.
```

```
      DO 460 I=1,MK,1
460   C1(I)=CC1(I)-CC1(II)
C     THE PROCESS CONTINUE WITH THE NEXT SYMBOL.
900   CONTINUE
C     CALCULATE ERROR RATES AND PRINTOUT ALL RESULTS
      WRITE(1,*)
      WRITE(1,500) SNR,P**2
      SER=REAL(ISER)/REAL(L)
      EVE=REAL(IEVA)/REAL(L)
      BER=REAL(IBER)/REAL((L)*4)
      WRITE(1,600) SER,EVE,BER
500   FORMAT(1X,4HSNR=,F12.6,2X,10HNOISE VAR=,F12.6)
600   FORMAT(1X,4HSER=,F9.7,1X,4HEVE=,F9.7,1X,4HBER=,F9.7)
1000  CONTINUE
      END
####S
```

APPENDIX G5
THE FOLLOWING PROGRAM SIMULATES THE OPERATION OF SYSTEM 3C16 .
THE TRANSMITTED SIGNAL IS A CONVOLUTIONALLY CODED 32-LEVEL QAM
SIGNAL THE DETECTOR OPERATES WITH 16 STORED VECTORS .

```
/*JOB S3C16,EUELAKK,ST=MFX,C=S,TI=1280,
/* PW=K
FTN5,DB=0/PMD,L=0.
LIBRARY,PROCLIB.
NAG(FTN5)
LGO.
####S
      PROGRAM S3C16
      INTEGER IS1(33),IS2(33),ISS(34),IA(33,4),IAA(4)
     C,IXXX1(128),IXXX2(128),IXXX3(128),IXXX4(128),IXXX5(128)
     C,IADD(16,33),IAD(16,33),ISTATP(16),IEVA,IEVK,IXX2(16,33)
     C,ISER,IBER,IX1(16,33),IX2(16,33),IXX1(16,33)
      REAL Y1(20),Y2(20),Z1(16),Z2(16),C1(16),C2(16,128),CC1(16)
      REAL P,SNR,SNRD,G05DDF,G05DAF

C     LOAD THE SAMPLED IMPULSE RESPONSE OF THE CHANNEL
      DATA   Y1 / 1.0000, 0.4861,-0.5980, 0.1702,-0.0245, 0.0100
     C           ,-0.0134, 0.0056, 0.0003,-0.0008, 0.0000, 0.0007
     C           , 0.0037,-0.0019, 0.0020, 0.0005,-0.0022, 0.0007
     C           ,-0.0008, 0.0005/
      DATA   Y2 / 0.0000, 1.0988, 0.0703,-0.1938, 0.1000,-0.0258
     C           , 0.0110,-0.0042, 0.0003, 0.0041,-0.0061,-0.0007
     C           , 0.0002,-0.0025, 0.0008,-0.0002, 0.0002,-0.0005
     C           , 0.0002, 0.0005/
C     LOAD TABLE FOR ENCODER/DECODER AND SIGNAL MAPPING
      DATA IXXX1/0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1
     C          ,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
     C          ,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4
     C          ,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5
     C          ,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7
     C          ,7,7,7,7,7,7,7,7/
      DATA IXXX2/0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2
     C          ,4,4,4,4,6,6,6,6,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6
     C          ,0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6,1,1,1,1,3,3,3,3
     C          ,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7
     C          ,1,1,1,1,3,3,3,3,5,5,5,5,7,7,7,7,1,1,1,1,3,3,3,3
     C          ,5,5,5,5,7,7,7,7/
      DATA IXXX3/-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
     C          , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3
     C          , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
     C          ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
     C          , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C          ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
     C          , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
     C          ,-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5/
      DATA IXXX4/-1, 3, 3,-5, 1,-3, 5, 1,-1, 3,-5,-1, 1,-3,-3, 5
     C          , 1,-3,-3, 5,-1, 3,-5,-1, 1,-3, 5, 1,-1, 3, 3,-5
     C          ,-1, 3,-5,-1, 1,-3,-3, 5,-1, 3, 3,-5, 1,-3, 5, 1
```

```
C           , 1,-3, 5, 1,-1, 3, 3,-5, 1,-3,-3, 5,-1, 3,-5,-1
C           ,-1, 3,-5, 3,-3, 1, 1, 5, 3,-1,-1,-5, 1,-3, 5,-3
C           ,-3, 1, 1, 5,-1, 3,-5, 3, 1,-3, 5,-3, 3,-1,-1,-5
C           , 3,-1,-1,-5, 1,-3, 5,-3,-1, 3,-5, 3,-3, 1, 1, 5
C           , 1,-3, 5,-3, 3,-1,-1,-5,-3, 1, 1, 5,-1, 3,-5, 3/
      DATA IXXX5/ 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12
C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
C           , 1, 5, 9,13, 3, 7,11,15, 3, 7,11,15, 1, 5, 9,13
C           , 2, 6,10,14, 0, 4, 8,12, 0, 4, 8,12, 2, 6,10,14
C           , 3, 7,11,15, 1, 5, 9,13, 1, 5, 9,13, 3, 7,11,15
C           , 0, 4, 8,12, 2, 6,10,14, 2, 6,10,14, 0, 4, 8,12/
C     SET ALL PARAMETERS.
      MM=2
      N =33
      N1=N-1
      N2=N+1
      L=100000
      LN=L+N1
      MK=16
      MKK=2
      IQ=10
      SNR=17.25
      SNRD=0.25
      CALL G05CBF(IQ)
C     SCALING OF THE SAMPLED IMPULSE RESPONSE
      YYY=0.0
      DO 10 I=1,20,1
10    YYY=YYY+Y1(I)*Y1(I)+Y2(I)*Y2(I)
      YY=SQRT(YYY)
      DO 20 I=1,20,1
      Y1(I)=Y1(I)/YY
20    Y2(I)=Y2(I)/YY
C     MAIN DO LOOP (SELECTION OF SNR)
      DO 1000 MMM=1,MM,1
      DO 1 I=1,MK,1
      DO 1 J=1,N,1
      IXX1(I,J)=0
1     IXX2(I,J)=0
      SNR=SNR+SNRD
      P=SQRT(5.0/(10.0**(SNR/10.0)))
C     INITIALIZATION OF ALL ARRAY AND VARIABLES.
      ISER=0
      IBER=0
      IEVA=0
      IEVK=0
      DO 40 I=1,N,1
      IS1(I)=-1
      IS2(I)=-1
      ISS(I)=0
      DO 30 J=1,4,1
30    IA(I,J)=0
      DO 40 J1=1,MK,1
      IX1(J1,I)=-1
```

```
      IX2(J1,I)=-1
40    IAD(J1,I)=0
      DO 50 I=2,MK,1
50    C1(I)=100000.0
C     ASSIGN DIFFERENT STATES FOR THE VECTORS WITH ZERO COST
C     TO THE VECTOR WITH THE CORRECT STATE.
      DO 60 I=0,7,1
      DO 60 J=2*I+1,2*(I+1),1
60    ISTATP(J)=I
      C1(1)=0.0
      ISS(N2)=0
C     SECOND DO LOOP (TRANSMISSION)
      DO 900 LLL=1,LN,1
C     SHIFT ALL ARRAYS .
      DO 100 J=1,N1,1
      JA=J+1
      IS1(J)=IS1(JA)
      IS2(J)=IS2(JA)
      ISS(J)=ISS(JA)
      DO 90 I=1,4,1
90    IA(J,I)=IA(JA,I)
      DO 100 K=1,MK,1
      IX1(K,J)=IX1(K,JA)
      IX2(K,J)=IX2(K,JA)
100   IAD(K,J)=IAD(K,JA)
      ISS(N)=ISS(N2)
C     GENERATE INFORMATION DIGITS.
      DO 110 I=1,4,1
110   IA(N,I)=NINT(G05DAF(0.0,1.0))
C     CONVOLUTIONAL ENCODING AND SIGNAL MAPPING
      INPUT=IA(N,1)+2*IA(N,2)+4*IA(N,3)+8*IA(N,4)
      DO 140 I=1,128,1
      IF(ISS(N).EQ.IXXX2(I).AND.INPUT.EQ.IXXX5(I)) THEN
      II=I
      ENDIF
140   CONTINUE
      ISS(N2)=IXXX1(II)
      IS1(N)=IXXX3(II)
      IS2(N)=IXXX4(II)
C     CALCULATION OF THE RECEIVED SAMPLE
      RR1=0.0
      RR2=0.0
      DO 150 I=1,20,1
      J=N+1-I
      RR1=RR1+REAL(IS1(J))*Y1(I)-REAL(IS2(J))*Y2(I)
150   RR2=RR2+REAL(IS1(J))*Y2(I)+REAL(IS2(J))*Y1(I)
C     GENERATE AND ADD TH NOISE.
      W1=G05DDF(0.0,P)
      W2=G05DDF(0.0,P)
      RR1=RR1+W1
      RR2=RR2+W2
C     CALCULATE ISI FOR EACH STORED VECTOR
      DO 170 I=1,MK,1
      ZZ1=0.0
      ZZ2=0.0
```

```
      DO 160 J=2,20,1
      J1=N+1-J
      ZZ1=ZZ1+REAL(IX1(I,J1))*Y1(J)-REAL(IX2(I,J1))*Y2(J)
160   ZZ2=ZZ2+REAL(IX1(I,J1))*Y2(J)+REAL(IX2(I,J1))*Y1(J)
      Z1(I)=(RR1-ZZ1)*YY
170   Z2(I)=(RR2-ZZ2)*YY
C     SEARCH FOR ALL VALID TRANSITION AND COST CALCULATIONS FOR
C     ALL VECTORS.
      DO 180 I=1,128,1
      DO 180 J=1,MK,1
180   C2(J,I)=1000000.0
      DO 210 I=1,MK,1
      DO 210 J=1,128,1
      IF(ISTATP(I).EQ.IXXX2(J)) THEN
      D1=Z1(I)-REAL(IXXX3(J))
      D2=Z2(I)-REAL(IXXX4(J))
      C2(I,J)=C1(I)+D1*D1+D2*D2
      ENDIF
210   CONTINUE
C     FIND THE VECTOR WITH THE SMALLEST COST .
      CC=500000.0
      DO 220 I=1,MK,1
      DO 220 J=1,128,1
      IF(C2(I,J).LT.CC) THEN
      CC=C2(I,J)
      II=I
      JJ=J
      ENDIF
220   CONTINUE
      IX1(II,N)=IXXX3(JJ)
      IX2(II,N)=IXXX4(JJ)
      IAD(II,N)=IXXX5(JJ)
C     ERROR CALCULATION
      IEVK=IEVK+1
      IF(IS1(1)-IX1(II,1)) 250,240,250
240   IF(IS2(1)-IX2(II,1)) 250,290,250
250   ISER=ISER+1
      IF(IEVK-32) 260,260,270
260   IEVA=IEVA+1
270   IEVK=0
290   CONTINUE
      IAAD=IAD(II,1)
      DO 300 I=4,1,-1
      JJ=2**(I-1)
      IF(IAAD.LT.JJ) THEN
      IAA(I)=0
      ELSE
      IAA(I)=1
      IAAD=IAAD-JJ
      ENDIF
300   CONTINUE
      DO 320 I=1,4,1
      IF(IAA(I)-IA(1,I)) 310,320,310
310   IBER=IBER+1
320   CONTINUE
```

```
C       DISCARDING OF ALL VECTOR WHICH DISAGREE IN THE
C       FIRST COMPONENT.
        CC=1000000.0
        DO 360 I=1,MK,1
        IF(IX1(II,1)-IX1(I,1)) 340,330,340
330     IF(IX2(II,1)-IX2(I,1)) 340,360,340
340     CONTINUE
        DO 350 J=1,128,1
350     C2(I,J)=CC
360     CONTINUE
C       SELECT 2 VECTORS FOR EACH STATE .
        J1=0
        QQ1=0
        DO 430 I1=0,7,1
        DO 430 JJ1=1,MKK,1
        CC=1000000.0
        J1=J1+1
        DO 380 I=1,128,1
        DO 380 J=1,MK,1
        IF(I1.EQ.IXXX1(I)) THEN
        IF(C2(J,I).LT.CC) THEN
        CC=C2(J,I)
        JJ=J
        II=I
        ENDIF
        ENDIF
380     CONTINUE
        QQ1=QQ1+1
        IXX1(J1,N)=IXXX3(II)
        IXX2(J1,N)=IXXX4(II)
        IADD(J1,N)=IXXX5(II)
        DO 420 K=1,N1,1
        IXX1(J1,K)=IX1(JJ,K)
        IXX2(J1,K)=IX2(JJ,K)
420     IADD(J1,K)=IAD(JJ,K)
        CC1(J1)=C2(JJ,II)
        C2(JJ,II)=1000000.0
        ISTATP(J1)=IXXX1(II)
430     CONTINUE
C       TRANSFER ALL ARRAYS
        DO 440 I=1,MK,1
        DO 440 J=1,N,1
        IX1(I,J)=IXX1(I,J)
        IX2(I,J)=IXX2(I,J)
440     IAD(I,J)=IADD(I,J)
        CC=2000000.0
        DO 450 I=1,MK,1
        IF(CC1(I).LT.CC) THEN
        CC=CC1(I)
        II=I
        ENDIF
450     CONTINUE
C       SUBTRACT THE SMALLEST COST.
        DO 460 I=1,MK,1
460     C1(I)=CC1(I)-CC1(II)
```

```
C       THE PROCESS CONTINUE WITH THE NEXT SYMBOL.
900     CONTINUE
C       CALCULATE ERROR RATES AND PRINTOUT ALL RESULTS
        WRITE(1,*)
        WRITE(1,500) SNR,P**2
        SER=REAL(ISER)/REAL(L)
        EVE=REAL(IEVA)/REAL(L)
        BER=REAL(IBER)/REAL((L)*IBIT)
        WRITE(1,600) SER,EVE,BER
500     FORMAT(1X,4HSNR=,F12.6,2X,10HNOISE VAR=,F12.6)
600     FORMAT(1X,4HSER=,F9.7,1X,4HEVE=,F9.7,1X,4HBER=,F9.7)
1000    CONTINUE
        END
####S
```

REFERENCES

1. Pahlavan, K. and Holsinger, J. L., "Voiceband data communication modems- A historical review : 1919-1988", IEEE Comm. Mag., Vol. 26, No. 1, pp. 16-27, Jan. 1988.

2. Falconer, D. D., "Bandlimited digital communications; recent trends and applications to voiceband modems and digital radio", Proc. of the 2nd Inter. Workshop on Digital Communications, Tirrenia, Italy, Sept. 1985.

3. Ungerboeck, G., "Channel coding with multi-level / phase signals", IEEE Trans. on Inf. Theory, Vol. IT-28, No. 1, pp. 55-67, Jan. 1982.

4. Wei, L. F., "Rotationally invariant convolutional channel coding with expanded signal space- part I : 180 ", IEEE Selected Area in Comm., Vol. SAC-2, No. 2, pp. 659-671, Sept. 1984.

5. Wei, L. F., "Rotationally invariant convolutional channel coding with expanded signal space - part I : Nonlinear codes", IEEE Selected Area in Comm., Vol. SAC-2, No. 2, pp. 672-686, Sept. 1984.

6. Brownlie, J. D. and Cusack, E. L., "Duplex transmission at 4800 and 9600 bit/s on the PSTN and use of channel coding with partitioned signal constellation", British Telecom. Tech. J., Vol. 2, No. 4, pp. 64-73, Sept. 1984.

7. Rapporteur on 14400 bit/s Modem, "Working draft recommendation V.cc for a 14400 bits/s modem standardized for use on point-to-point 4-wire leased telephone-type circuits", CCITT Study Group XVII, 1984.

8. Ungerboeck, G., "Trellis-coded modulation with redundant signal sets, part I : Introduction", IEEE Comm. Mag., Vol. 25, No. 2, pp. 5-11, Feb. 1987.

9. IBM Europe, "Trellis-coded modulation scheme with 8-state systematic encoder and 90 symmetry for use in data modems transmitting 3-7 bits per modulation interval", CCITT, COM. XVII-D180, Oct. 1983.

10. Viterbi, A. J., "Convolutional codes and their performance in communication systems", IEEE Trans. on Comm. Tech., Vol. COM-19, No. 5,pp. 751-772, Oct. 1971.

11. Omura, J. K., "On the Viterbi decoding algorithm", IEEE Trans. on Inf. Theory (corresp.), Vol. 15, No. 1, pp. 177-179, Jan. 1969.

12. Borelli, W. C., Rashvand, H. F. and Farrell, P. G., "convolutional codes for multi-level modems", Elect. Lett., Vol. 17, No. 9, pp. 331-333, April 1981.

13. Fagan, A. D., and Okeane, F. D., "Performance comparison of detection methods derived from maximum-likelihood sequence estimation", IEE Proc., Vol. 133, Pt. F, No. 6, pp. 535-542, Oct. 1986.

14. Clark, A. P., and Kadhim, A-K. A-R., "Detection of coded and distorted QAM signal", J. IERE, Vol. 58, No. 4, pp. 187-196, June 1988.

15. Acampora, A. S., "Maximum-likelihood decoding of binary convolutional codes on band-limited satellite channels", IEEE Trans. on Comm., Vol. COM-26, No. 6,pp. 766-775, June 1978.

16. Forney, G. D. Jr., "The Viterbi algorithm", IEEE Proc., Vol. 61, pp. 268-278, March 1973.

17. Wolf, J. K. and Ungerboeck, G., "Trellis coding for partial response channels", IEEE Trans. on Comm., Vol. COM-34, No. 8, pp. 765-773, Aug. 1986.

18. Clark, A. P., "Combined coding and modulation of digital signals", IEEE Int. Symp. on Information Theory, Brighton, England, pp. 29, June 1985.

19. Viterbi, A. J. and Omura, J. K., "Principles of digital communication and coding", Mcgraw-Hill, 1979.

20. Clark, A. P., "Principles of digital data transmission", 2nd Edition, Pretech Press, 1983.

21. Clark, A. P., "Modeling of digital communication systems", International Seminar on Mathematical Modelling and Applications, Arab School of Science and Technology, Damascuse, Syria, April 1986.

22. Clark, A. P., "Equalizers for digital modems", Pentech Press, 1985.

23. Clark, A. P. and Fairfield, M. J., "Detection processes for a 9600 bit/s modem", J. IERE, VOL. 51, No. 9, pp. 455-465, Sept. 1981.

24. Proakis, J. G., "Digital communication", McGraw-Hill, 1983.

25. CCITT Recommendation, "Yellow book, M-series Recs. M1020, M1040", CCITT, Geneva, 1980.

26. Clark, A. P., "Advanced data transmission system", Pretech Press, 1977.

27. Clark, A. P. and Tint, U. S., "Linear and nonlinear transversal equalizers for baseband channels", The Radio and Electronic Eng., Vol. 45, No. 6, pp. 271-283, June 1973.

28. Lucky, R. W., "A survey of the communication theory literature: 1968-1973", IEEE Trans. on Inf. Theory, Vol. IT-19, No. 5, pp. 725-739, Nov. 1973.

29. Clark, A. P. and Hau, S. F., "Adaptive adjustment of receiver for distorted digital signals", IEE Proc., Pt. F, Vol. 131, pp. 526-536, Aug. 1984.

30. Clark, A. P., Kwong, C. P. and McVerry, F., "Estimation of the sampled impulse response of a channel", Signal Processing, Vol. 2, No. 1, pp. 39-53, Jan. 1980.

31. Forney, G. D., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference", IEEE Trans. on Inf. Theory, Vol. IT-18, No. 3, pp. 363-378, May 1972.

32. Hayes, J. F., Cover, T. M. and Riera, J. B., "Optimal sequence detection and optimal symbol-by-symbol detection: similar algorithms", IEEE Trans. on Comm. Tech., Vol. COM-30, No. 1, pp. 152-157, Jan. 1982.

33. Acompora, A. S., "Analysis of maximum-likelihood sequence estimation performance for quadrature amplitude modulation", Bell sys. Tech. J., Vol. 60, No. 6, pp. 865-885, July-Aug. 1981.

34. Foschini, G. J., "Performance bound for maximum-likelihood reception of digital data", IEEE Trans. Inf. Theory, Vol. IT-21, No. 19, PP. 47-50, Jan. 1975.

35. Magee, F. R. and Proakis, J. G., "An estimate of the upper bound on error probability for maximum-likelihood sequence estimation on channel having a finite duration pulse response", IEEE Trans. Inf. Theory (corresp.), Vol. IT-19, No. 5, pp. 699-702, Sept. 1973

36. Beare, C. T., "The choice of the desired impulse response in combined linear Viterbi algorithm equalizer", IEEE Trans. on Comm., Vol. COM-26, No. 8, pp. 1301-1307, Aug. 1978.

37. Qureshi, S. U. and Newhau, E. E., "An adaptive receiver for data transmission over time dispersive channels", IEEE Trans. Inf. Theory, Vol. IT-19, No. 4,pp. 448-457, July 1973.

38. Lee, W. U. and Hill, F. S. Jr., "A maximum-likelihood sequence estimator with decision-feedback equalization", IEEE Trans. on Comm., Vol. COM-25, pp. 971-979, 1977.

39. Najdi, H. Y., "Digital data transmission over voice channels", Ph.D. Thesis, Loughborough University of Technology, Loughborough, Leicester, England, 1982.

40. Anderson, I. N., "Sampled-whitened matched filters", IEEE Trans. Inf. Theory, Vol. IT-19, No. 5, pp. 653-660, Sept. 1073.

41. Clark, A. P., Harvy, J. D. and Driscoll, J. P., "Near-maximum likelihood detection processes for distorted digital signals", The Radio and Electronic Eng., Vol. 48, pp. 301-309, June 1978.

42. Clark, A. P., Najdi, H. Y. and Fairfield, M. J., "Data transmission at 19.2 kbit/s over telephone circuits", J. IERE, Vol. 53, No. 4, pp. 157-166, April 1983.

43. Ser, W. and Clark, A. P., "Near-maximum likelihood detectors for binary signals", IEE Proc., Vol. 132, Pt. F, No. 6, pp. 485-490, Oct. 1985.

44. Clark, A. P., Ip. S. F. A. and Soon, C. W., "Pseudobinary detection processes for 9600 bit/s modems", IEE Proc., Vol. 129,Pt. F, No. 5, pp. 305-314, Oct. 1982.

45. Clark, A. P., "Near-maximum likelihood detectors", IEE Colloquium on Adaptive Filters, London, England, pp. 2/1-6, Oct. 1985.

46. Clark, A. P. , Abdullah, S. N., Jaysinghe, S. G. and Sun, K. H., "Pseudobinary and pseudoquaternary detection processes for linearly distorted multilevel QAM signals", IEEE Trans. on Comm., Vol. COM-33, No. 7, pp. 639-645, July 1985.

47. Clark, A. P. and Clayden, M., "Pseudobinary Viterbi detector", IEE Proc. , Vol. 131, Pt. F, No. 2, pp. 208-218, April 1984.

48. Clark, A. P., Harvy, J. D. and Driscoll, J. P., "Improved detection processes for digital signals", IERE Conf. Proc. 37, pp. 125-130, ... 1977.

49. Mclane, P. J., "A residual intersymbol interference error bound for truncated-state Viterbi detectors", IEEE Trans. Inf. Theory, Vol. IT-26, No. 5, pp. 548-553, Sept. 1980.

50. Clark, A. P. and Najdi, H. Y., "Detection process of a 9600 bit/s serial modem for HF radio links", IEE Proc., Vol. 130, No. 5, pp. 368-376, Aug. 1983.

51. Clark, A. P., Najdi, H. Y. and McVerry, F., "Performance of a 9600 bit/s serial modem over a model of an HF radio link", IEE Conf. Publication No. 224, Radio Spectrum Conservation Techniques, pp. 151-155, Sept. 1983.

52. Clark, A. P., "Near-maximum likelihood detectors", Jordan International Electrical and Electronic Eng. Conf., Amman, Jordan, 28 April - 1 May 1985.

53. Clark, A. P., Kwong, C. P. and Harvy J. D., "Detection processes for severely distorted digital signals", Elect. circuits and sys., Vol. 3, No. 1, pp. 27-37, Jan. 1979.

54. Driscoll, J. P. and Karia, N., "Detection processes for V32 modems using trellis coding", IEE Proc., Vol. 135, Pt. F, April 1988.

55. Aftelak, S. B. and Clark, A. P., "Adaptive reduced-state Viterbi algorithm detector", J. IERE, Vol. 56, No. 5, pp. 197-206, May 1986.

56. Zhu, Z. C. and Clark, A. P., "Near-maximum likelihood decoding of convolutional codes", IEE Proc., Vol. 135, Pt. F, No. 1, pp. 33-42, Feb. 1988.

57. Clark, A. P. and Abdullah, S. N., "Near-maximum likelihood detectors for voiceband channels", IEE Proc., Vol. 134, Pt. F, No. 3, pp. 217-226, June 1987.

58. Blahut, R. G., "The theory and practice of error control coding", Addison-Wesley, 1983.

59. Belfore, C. A. and Park, J. H., "Decision feedback equalization", IEEE Proc., Vol. 67, No. 8, pp. 1143-1156, Aug. 1979.

60. Clark, A. P., Abdullah, S. N. and Ameen, S. Y., "A comparison of decision feed-back equalizers for a 9600 bit/s modem", J. IERE, Vol. 58, No. 2, pp. 74-83, April 1988.

61. Viterbi, A. J., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE Trans. on Inf. Theory, Vol. IT-13, No. 2, pp. 260-270, April 1967.

62. Jensen, J. M. and Reed, I. S., "Bounded distance coset decoding of convolutional codes", IEE Proc., Vol. 133, Pt. F, No. 5, pp. 488-492, Aug. 1986.

63. Zhu, Z. C. and Clark, A. P., "Rotationally invariant coded PSK signals", IEE Proc., Vol. 134, Pt. F, No. 1, pp. 43-52, Feb. 1987.

64. Bendetto, S., Marsan, M. A., Albertengo, G. and Giachin, E., "Combined coding and modulation: Theory and applications", IEEE Trans. on Inf. Theory, Vol. IT-34, No. 2, pp. 223-236, March 1988.

65. Thapar, H. K., "Real-time application of trellis coding to high-speed voiceband data transmission", IEEE J. of Selected Area in Comm., Vol. SAC-2, No. 5, pp. 648-658, Sept. 1984.

66. Digeon, A., "On improving bit-error probability of QPSK and 4-level amplitude modulation systems" IEEE Trans. on Comm., Vol. COM-25, No. 10, pp. 1238-1239, Oct. 1977.

67. Perry, K. E. and Wozencraft, J. M., "SECO: A self-regulating error correcting coder-decoder", IRE Trans. on Inf. Theory, Vol. IT-8, pp. 128-135, Sept. 1962.

68. Ungerboeck, G., "Trellis-coded modulation with redundant signal sets, part II : State of the art", IEEE Comm. Mag., Vol. 25, No. 2, pp. 12-21, Feb. 1987.

69. AT and T Information System, "A trellis-coded modulation scheme that includes differential encoding for 9600 bit/s, full duplex, two-wire modem", CCITT COM XVII-D159, Aug. 1983.

70. Rapporteur, "Draft recommendation V32 for a family of 2-wire, duplex modems operating at data signalling rates of up to 9600 bit/s for use on the general switched telephone network and on leased telephone-type circuits", CCITT Qu. 16/XVII, Nov. 1983.

71. IBM Technical Disclosure Bulletin, "Sixteen-state forward convolutional encoder", Vol. 28, No. 6, pp. 2466-2468, Nov. 1985.

72. Forney, G. D., Gallager, R. G., Lang, Longstaff, F. M. and Qureshi, S. U., "Efficient modulation for band-limited channels", IEEE J. Selected Area in Comm., Vol. SAC-2, No. 5, pp. 632-647, Sept. 1984.

73. Gersho, A. and Lawrance, V. B., "Multidimensional signal constellations for voiceband data transmission", IEEE J. Selected Area in Comm., Vol. SAC-2, No. 5, pp. 687-702, Sept. 1984.

74. Lee, L. H. C. and Farrell, P. G., "Error performance of maximum-likelihood trellis decoding of (n,n-1) convolutional codes: a simulation study", IEE Proc., Vol. 134, Pt. F, No. 7, pp. 673-680, Dec. 1987.

75. Hemmati, F. and Costello, D. J., "Truncation error probability in Viterbi decoding", IEEE Trans. on Comm., Vol. COM-25, pp. 530-532, May 1977.

76. Clark, G. C. and Cain, J. B., "Error-correction coding for digital communications", Plenum Press, 1981.

77. Clark, A. P., "Minimum-distance decoding of binary convolutional codes", Computer and Digital Techniques, Vol. 1, No. 7, pp. 190-196, 1978.

78. Farrell, P. G., Tait, D. J. and Borelli, W. C., "A convolutional code using soft decision for practical line transmission", IEE Colloquium on Data Transmission Codes, Nov. 1980.

79. Hemmati, F. and Fang, R. J., "Low-complexity coding methods for high data rate channels", COMSAT Tech. Rev., Vol. 16, No. 2, pp. 425-447, 1986.

80. Immink, K. A. S., "Coding techniques for partial response channels", IEEE Trans. on Comm., Vol. 36, No. 10, pp. 1163-1165, Oct. 1988.

81. Marsan, M. A., Albertengo and Benedetto, S., "Combined coding and modulation: performance over real channels", Proc. of the 2nd Inter. Workshop on Digital Communications, Tirrenia, Italy, Sept. 1985.

82. Kohno, R., Amai, H. and Hatori, M., "Design of automatic equalizer including a decoder of error-correcting code", IEEE Trans. on Comm., Vol. COM-33, No. 10, pp. 1142-1146, Oct. 1985.

83. Kohno, R., Pasupathly, S., Amai, H. and Hatori, M., "Combination of cancelling intersymbol interference and decoding of error-correcting code", IEE Proc., Vol. 133, Pt. F, No. 3, pp. 224-231, June 1986.

84. Ishida, K., Oka, I. and Endo, I., "Cochannel interference cancelling effects of MSE processing and Viterbi decoding in on-board processing satellites", IEEE Trans. on Comm., Vol. COM-34, No. 10, pp. 1049-1053, Oct. 1986.

85. Akashi, F., Tatsui, N., Sato, Y., Koike, S., and Marumo, Y., "A high performance digital QAM 9600 bit/s modem", NEC Research and Development, Pt. 45, pp. 38-48, (Japan), April 1977.

86. Fairfield, M. J., "Analysis of telephone connection", Internal Communication, Loughborough University of Technology, 1978.

87. Papoulis, A., "Probability, random variables and stochastic processes", McGraw-Hill, New York , 1965.

88. Abdullah, S. N., "Data transmission at 9600 bit/s over an HF radio link", Ph.D. Thesis, Loughborough University of Technology, Loughborough, England, 1986.

89. Ramsey, J. L., "Realization of optimum interleavers", IEEE Trans. on Inf. Theory, Vol. IT-16, No. 3, pp. 338-345, May 1970.

90. Clark, A. P. and Ser, W., "Improvement in tolerance to noise through the transmission of multilevel coded signals", IERE Conference on Digital Processing of signals in Communications, Loughborough, England, pp. 129-141, April 1981.

91. Calderbank, R., and Mazo, J. E., "A new description of trellis codes", IEEE Trans. on Inf. Theory, Vol. IT-30, No. 6, pp. 784-791, Nov. 1984.

92. Bateman, S. C., "Data transmission at 19,200 bit/s over telephone channels", Ph.D. Thesis, Loughborough University of Technology, Loughborough, Leicester, England, 1985.

93. Wozencraft, J. M. and Jacobs, I. M., "Principles of communication engineering", John Wiley and Sons, New York 1967.

94. Portny, S. E., "Large sample confidence limits for binary error probabilities", IEEE Proc., Vol. 54, P. 1993, Dec. 1966.